Developing an integrated scoring and statistics program for baseball

Paul Brennan
Information Systems

2004/2005

SUMMARY

This project shall look at scoring baseball in the UK. Currently there is not a lot of methods to do this and baseballsoftballuk want to automate the process in order to make the game more enjoyable for scorers and fans alike. There needs to be some software developed that allows users to score a baseball game as they currently do manually, but quicker and with more efficiency. There then needs to be a database set up in order for this data to be stored on a central server. Finally, a website needs to be developed, which has access to this database and which users can navigate to and use with ease to find out statistics on their favourite players or find in depth details about a match.

All these systems shall be integrated and work with each other so that information can seemingly flow from the pitch side to the fans viewing it on the Internet. It shall be developed alongside BSUK in order to keep requirements as close to the company's specification as possible.

NB: Whilst the content of this electronic submission is the same as the hard copy submitted, the page numbering has altered due to the collaboration of several documents. Also some appendices are not available because they were hand written and not available on the PC. The contents of this document have been altered to reflect these two discrepancies.

Understanding The Environment

1.1 Overview of environment

Baseball is a game created in America approximately 300 years ago. It is a ball game involving two teams that try and score more runs than the opposing team. Each team comprises of nine players, who play both offensive and defensive roles within a team. When a player is "at bat" (AB), he can either be putout or is safe and becomes a runner. When a runner crosses home plate, a run is scored. For more details rule explanation see appendix 1. In America, baseball is a huge sport and is attended by approximately a quarter of a million people every day for more than half the year. The TV and press coverage in the states is also large with audiences worldwide and salaries of players and coaches reaching eight figures a year. It is a completely different scenario in the UK and is considered a minority sports with little or no coverage in the press and with a weekly attendance not even reaching one thousand nationwide.

Baseball is an Olympic sport and is one that Britain is hoping to compete in within twenty years and so recently has been given more funding to help raise the profile of the game. This project is one stage in a number of stages for bringing the game more into the public eye and helps raise the awareness and potential number of players and fans for the sport on a local and national level.

1.2 Immediate environment

BaseballSoftballUK (BSUK, the company) is the company that this project is being developed for. BSUK is the governing body for baseball and softball in the UK and moderates all games both on an adult and youth level. It is the managing agency for both the British baseball Federation and the British Softball Federation. It is a London based company and has two seats on the British Olympic board and therefore receives it's funding from the Olympic organisation with a view to become competitive sooner on an international scale. BSUK is a non-profit making organisation and is staffed mostly by volunteers and fans of the sport.

This project is going to concentrate on the adult baseball game, with scope to expand to youth baseball. There are forty clubs at this level and play on a weekly basis with quantity of players expanding constantly over the past years and now over one thousand players are competing on a regular basis.

The centre of this organisation is the website (www.baseballsoftballuk.com), it contains all the information about clubs, is a way for players to find new clubs and look at league

standings and scores from around the league. This website is becoming more and more popular due to the number of people playing and watching the local game and also the fact that it has been promoted on Channel Five for viewers of the American game.

1.3 Current Solution

1.3.1 Overview of current solution

Currently there is a particularly complicated process to get record the scores and the general public being able to get access to these. This is shown in the UML diagram in appendix 2. The main stages are that the scores are recorded at the game by an official scorer and then later on recorded in exactly the same manner onto an online system and uploaded onto the website. Then there is the option to write extra details or just to leave it as a "box score" (see appendix 3). Another method currently in place is to record the scores on a score sheet (see appendix 4) and then enter the details on the online system as shown in the UML diagram (appendix 2). Currently the website contains very little information about some games and yet a whole ball for ball report on others. This could be the difficulty in going from a manual score-sheet to a more meaningful representation.

1.3.2 Users of current solution

The solution that is in place at the moment is not used a great deal due to the complexity of it and the fact that the value of the end product is not that great. The main users of the current system are the implementers of it, the official scorers of the league. The system is not used by any other group of people to actually produce something, but is used by others to view information and data recorded. The players and fans are probably the next largest user of this system followed by league and specific team officials. Finally there are some casual users that will hardly ever use the system

1.3.2.1 *Official scorers:* These use the system for recording the scores onto an online server from either their pre-recorded electronic scoreboard or manual score-sheet. It is important that this group of people are a priority when using this system; they are responsible for ensuring that data is kept up to date and correct. They are not interested in manipulating the results and do not care about the effect that these results can have. Although they are interested in the accuracy of the results, they are not interested in the actual content of them as their capacity of a scorer. In terms of the database, they are responsible for populating it and nothing more. They do not need to be able to execute queries, produce reports or even delete from the database.

1.3.2.2 Players and fans: These people use the system for everything that the scorer does not. They are interested in looking at the results of matches on the website, reading reports about

their team and other teams within the league. The players are also interested in looking at what their performance was like compared to others around the league. Since the website is the only place that the results of every game is published in the public domain, this is the main functionality that it is used for. With the current solution players or fans are not allowed write access to the database but can execute queries to obtain some information.

1.3.2.3 Officials: Officials will hardly use the system at the moment but will use an admin facility to correct errors, raise complaints and advertise their team to the public. They are obviously interested in the team that they are responsible for and they will then have the same interest as a player or fan. League officials will have full access to the database and can manipulate it in whatever way possible.

1.3.2.4 Casual users: These include the British Olympic Association (BOA), some government officials (for auditing purposes) and casual Internet Surfers. The BOA are interested in how well the organisation is run and so may often visit the website to get a "feel" of what is going on.

1.3.3 Problems with the current system

There are a number of problems in the system at the moment, the one that stands out the most is the duplication of effort of the scorers, having to input scores of each game twice, once live and then again later to upload onto an Internet server. This is not just time consuming but can also cause inconsistency and inaccuracy of data when copying between the two applications. This process could easily be simplified by using an online scoring system or a simpler uploading process. Another problem with the system as it stands is the inconsistency of results. Each game can range from 18 numbers to a full two or three page report. By making each result the same and the fields mandatory, it can be resolved and each game will have the same output.

At the moment there is not a great deal of statistics held about players or teams and where this is not a feature needed for full functionality, it would increase the popularity of the game and also keep the interest of current players. This was a problem that arose when speaking to some of the players that were disappointed that they could not keep track of the performance throughout the season without doing it manually by themselves or looking at each game in turn and adding up the particular sections. This is clearly a big opportunity for improvement and can be easily achieved by simple mathematical functions. Another thing that could be improved upon is the details of each ball. If each ball thrown, or hit is recorded then a fan could "re-live" the game after it has happened and again improves the experience for fans and players alike.

Finally a big problem is the actual recording of the scores. As appendix 4 shows it is very complicated process and new scorers may have to be trained for a long time before they

are able to produce such a document accurately. This is time consuming, expensive and off-putting to potential scorers who may not want to have to concentrate too hard when they are effectively just fans that want a bit of involvement in the game. Clearly this is an opportunity to help these fans become more interested without having to learn a complex scoring system and can just learn a point and click interface on a laptop or hand-held device.

1.3.4 Business UML model

Appendix 2 shows the business UML model and details the flow of information for the current situation. As this model shows there seems to be a lot of different methods to the various stages and is a clear point that could be improved upon. Also the number of stages seems to be excessive and not as accurate or quick as it could be. In general the model seems to be "messy" with a lot of repeated processes and users. By making this model into a more streamlined and simpler model, it is possible to make this system a lot easier to use and eliminate the faults outlined in section 1.3.3.

1.4 Literacy search

1.4.1 Analysis and Design

There are a number of different methods of analysis and design, such as SQIRO and UML. I am going to use an object-oriented approach, which there is a lot of books written about. As well as basing my analysis on RUP's vision document [1], I shall incorporate a number of other ideas into this section of my report. Hackos et al [2] looks at how to analyse tasks and users in order to design an interface. This book describes object-oriented approaches and puts specific emphasis on flow charts and hierarchies. It also looks at most of the stages in a SQIRO approach in detail, giving advice on how to conduct interviews etc. Finally, this book gives a good, clear chapter on how to present this information formally and how to integrate the design with the analysis.

Barclay and Savage [3] Go into detail about designing using object-oriented methods, for programs developed in C++, they put forward the idea of integrated the design with the implementation of a product and believe that the production of a product should be very simple as long as the design is conducted properly. This book describes the basic forms of UML to an extent and also expands upon these methods by introducing LOOM, which is very similar to C++ but is used for design purposes. I shall use the ideas put forward in this book to ensure that the production of the product shall be quite straightforward and a simple step from design.

1.4.2 Databases

This project has a number of different requirements, but S.Roberts [4] has laid out what should be included within the report, this includes normalisation and an E-R model. Although these are laid out in Roberts' document, these are fairly straightforward exercises and should be included anyway.

The basis of all relational databases is Codd's model [17], which outlines the basic steps of making a database relational and this paper is an important part of my database design and implementation.

Kahn et al [5] goes into a lot of different ways in which databases can be designed and implemented and after considering these options I think it is best to stick to the "normal" method of ER diagrams and SQL. Chen [5] looks at entities and relations, which shall be a major part of this project and looks at hierarchical databases and logical design. His work shall form the basis of the database schema, by using these concepts it means that the database will be logically designed and presented and so may be easier to maintain either by myself or a third party in the future.

Hubbard [5] looks at using computers to aid the design of databases, whilst this method is a good idea in principle; it shall not be carried out due to the cost and complexity involved.

Lien [5] puts forward algorithms for designing databases in a relational manner, which seems a fresh idea in terms of design and I have considered this and it seems that this is too complex an idea to implement in the time constraints.

Pratt and Adamski [6] take the reader through the process of designing databases in a regular fashion, from the relational model, through to normalisation and physical design. This book shall be the one that I will use to follow the basic steps although not necessarily the methods that it uses.

Whereas Pratt [6] introduces SQL, it is very brief and does not look at the different languages available; therefore Dietrich's book [7] is a good start to deciding on which language to use when querying the database. This book also goes into details about how to use the languages and shall be used in the decision of which language to be used as well as a reference book when implementing the query language code.

1.4.3 Web-design and production

Mott [8] shows us how to use Adobe to connect Internet software with software and looks in detail about the technologies involved in this. This is the basis of what I shall be using in this project to ensure that all parts of the project can communicate with each other. Thatcher et al [9] discusses many issues to do with building a web site including the legal aspects and testing, this shall be discussed later in this project.

Experience from previous courses and academic qualifications means that in-depth research of this topic is not as important as other topics. I shall also use online reference libraries to ensure that the coding of the website is accurate and efficient. These shall be cited on use.

1.4.4 Interface design and production

Mullet and Sano [10] puts forward a different method, which could be looked at, based around communication rather than objects. Their book looks at how to design effective visual interfaces that are very simple to use. It also concentrates on structure and organisation of the designs. It is based on a GUI approach and supports this method highly above other recognised methods. This book is a good starting point for looking at the interface design and getting a good idea of how it shall function.

Whereas Sano et al [10] concentrates on communication oriented methods, Lea et al [11] looks at the object oriented methods of analysis, design and development. It guides the user all the way through the process of developing a system in an object-oriented way. It is particularly good at looking at the intermediary steps between the phases and how to get from one to the other.

Lea et al [11] is the book that shall be used in the development of this product and the stages shall be followed rigidly to ensure that the project is constructed in a feasible manner. This is the method that is used by most projects at the moment and so shall be a good guideline for how projects are developed in the professional world.

This project should also meet some criteria in software development [12] to make sure that it is implemented to a good standard; this shall be a constant reference point and be consulted at each stage of the design and implementation stage.

1.4.5 Baseball

Although there are not a lot of sources that can cited here, the American software is a good example of how the end software shall work [13,14] on the website and how the user shall interact with the software. MLB [14] shows a good example of a statistics program that could form the backbone of this projects stats section. Further information on these products can be found in section 1.8.

1.4.6 Evaluation

Sterne [15] is a good book for evaluating web sites and shows a good framework on how to do this effectively, without being biased. He looks at heuristic evaluation as well as other techniques available. Heuristics is also explored in greater detail in Kahneman et al [16], which concentrates largely on how to make effective evaluation using heuristics in uncertain circumstances and so can prove valuable when the result is not guaranteed to be successful.

The majority of literature read seems to think that heuristics is the best method of evaluation and so this is the method that shall be used in this project. By setting out the success criteria before hand it shall be simple to judge the success or failure of this project.

1.5 Problem to be solved

Having looked at the problems raised in section 1.3.3, it is clear that the current solution is not sufficient if this company is going to grow and attract new people to the sport of baseball; therefore there needs to be an improvement or replacement of the current system. This project looks on expanding the current solution and incorporating it with a new system to ensure that it has the maximum functionality and appeal to the general public.
This project will need to address three areas in detail:

- *Recording of scores and statistics –*
  This will be simpler and more regulated than the current method. It will be easy to use and be the same method over all games to create a consistent format of results.
- *Maintaining statistics and game details –*
  This shall expand on the current database that is in place to include new statistics and fields. It is also a priority to force referential integrity.
- *Returning statistics and game details on a web interface –*
  This shall be an entire new website where users can query the database and return results on certain games, teams or players in much greater detail that is currently available.

It is also very important that these three factors fit together and are not three separate systems but seemingly one large system that can do three functions.

1.6 Stakeholders and users

Although the stakeholders and users are very similar to that of the current system it is important to outline them, their responsibilities, and expectations of the new system.
The main group of stakeholders are users, which can be again split down into further groups:

1. Official scorers (stakeholder and user)

They will use the new system as they did the old one, but expect it to be easier to use and more consistent over all games. They will use the system to record scores as they did previously but will no longer have to record them twice as it could be uploaded via an Internet link. They will still not use the facility of the web site to query the database, as this is not needed to complete their job. However they may use the website for other

information such as player names and details of teams. The new system will heavily affect this group of users, as they will need to learn the new system and be able to work it before the product can have any value. They are very interested in the development and shall be consulted throughout the development of the system. Since the new system is going to replace the old one, and be used on a frequent basis (i.e. at least once a week) it needs to be tailored to their needs and they should have the number one priority when it comes to decisions.

2. Players and fans (stakeholder and user)

Currently using the system to check on scores and schedules, this usage will be extended to include the details of games, player stats and team stats. By extending the functionality of the current system and adding so many more fields to the database the new system will become used more frequently and probably by more people than the old one. They will not care how the system works and the stats get there so will not be interested in the structure of the database or the input program functionality. They will, however want the website to be suit their needs and be simple and attractive.

Although the development of this system will not directly affect this group of people, they will have to learn how to use the system, but this shall be done by self-teaching methods (i.e. there is no need to train them).

The development of this system, if successful in its aim, will make this group a much larger one. Hopefully the popularity of the sport will increase and people that enjoy the American game on TV will become aware of the British game and the website is there to help them follow the league week-by-week.

3. Officials (stakeholder and casual user)

Although will still not use the system a great deal will be greatly interested in the development of the new system. They are aware that the current system is inefficient and want to raise the profile of the game. The development of the new system will not affect them in any way. However, there may be functionality for more administration functions to be added if time permits. They will probably use the correction facility less often, as there will be more input validation and fewer stages where error could occur.

The website, upon becoming more in the public eye will give officials of the league chance to portray any messages to the public and also moderate the leagues' activity.

After talking to John Lee (Business Development manager for BSUK – stakeholder number 4), it was discovered that officials have been looking for a solution to this problem for a while but have been put off by the cost of such an operation. Therefore

officials will be looking for a decent solution whilst keeping monetary and other costs as low as possible.

This group of people is a representation of the organisation of BSUK as a whole and is therefore going to be the most interested in the success or failure of this product overall and how well it works with the current architecture, values and technology that are currently in place.

4. John Lee (External contact, general user and stakeholder)

John is my point of contact with BSUK and is overseeing this project externally. He is ultimately responsible for the project for the company and will have the last say whether it is to go live or not. His job is to promote baseball in the UK and help expand it is much as possible. This project is therefore a good step forward and he wants it to become a success. John will not use the system except to test it in its various stages of production. It is also important that I stay in contact with John throughout and work with him as much as possible as he as a clear idea of what the final functionality shall be.

5. Five (TV channel) (stakeholder)

A long-term partner of BSUK, they promote the local sport on the American coverage and are extending to cover some British games. They will also be promoting the finished product on this show and clearly want the product to be a success. By agreeing to promote the web site to its viewers, Five is giving its support and will therefore be interested in the result of the web site production process.

6. British Government (stakeholder)

Since the end product is going to be in the public domain, it is important that it complies with regulations that are in place. These are explained in more detail in section 2.9. Although this group of stakeholders probably will never see the system and may not even be aware of it, if there are illegal activities taking place then they will correct these out immediately. They will not use the system and if the product is successful then this stakeholder should not be affected at all.

7. Paul Brennan (The developer, stakeholder and user)

I am responsible for designing and developing this product and project. I will determine the amount of functionality of each part of the system, how it is being developed and what should be done when and how. I shall be greatly affected by the development of the new system, as it shall take a lot of my time and resources to complete the project. I shall also benefit from developing the project by acquiring new skills and techniques (see

evaluation). I shall also be using the system constantly as a developer, a tester, maintaining it once live and using it as a fan. Because of my numerous roles I shall be using the whole of the system, both as a finished product and "behind the scenes". Since I shall be the only person to get involved with the coding of this project, it is important that I am aware of how the system works completely and is able to explain it in detail to third parties such as the University (stakeholder 8) in my assessment.

8. University of Leeds (stakeholder and casual user)

The university is interested in this project, as they shall need to evaluate it and judge how successful it has been on a whole. They will be more interested in how the product was developed than the actual product itself. Although they will look at the functionality of the system, they will look in more detail at how this functionality was decided on and developed. They will also use the system in the future when future students are developing similar projects. There are two main contacts at the university:

8.1 Professor Martin Dyer (project supervisor)

Will help in the development of the project in terms of giving feedback and advice on a weekly basis. He will act as my contact to the university and will help me as much as possible with queries and other aspects of the project. Although Professor Dyer shall not use the system, he shall probably see it working and be able to help test the product as an independent third party.

8.2 Dr Natasha Shakhlevich (project assessor)

Ultimately has the say on whether this project is a success or not and how it could be improved upon. She will be looking at how well the product meets the aims and requirements, how well it functions and the methods used in getting to the final product. Although Dr Shakhlevich shall be an important stakeholder, she does not have a long-term interest in the product as some others mentioned above do.

9. British Olympic Committee (stakeholder)

Will not use the system, but will be interested in the success of the product because they have targeted baseball as a priority sport for development over the coming years. They will have similar interests to officials of BSUK (stakeholder 3) but not as concentrated to the actual functionality but simply how many more fans and players this product can attract to help them achieve their aim of competing in the Olympics in this sport.

1.7 Stakeholder/User Needs

Over the next pages is a list of stakeholder and user needs. Each need is applicable to one or more user or stakeholder and it has a level of importance assigned to it for each in particular group. The user/stakeholder number is representative of that given in section 1.6. Since each need is not necessarily as important as each, it was necessary to give each individual need a score, against which it can be evaluated later in this project and ultimately determine the success as a results of user and stakeholder satisfaction.
The weighting is calculated by the following formula

(Number of "W" * 4)+(Number of "S" * 3)+(Number of "C" * 2)+(Number of "M" * 1)

This is then multiplied by the importance factor (IF) of each user/stakeholder:

1: The most important user of the input system and the most frequent user. IF: 2.1
2: System developed for the benefit of this group. Needs satisfying. IF: 1.7
3: This group is not as important as other users but still needs satisfying. IF 1.1
4: John is not a beneficiary from the system but rather the results of it. IF 1.1
5: Five is not important at all. IF 0.8
6: This is the most important group of all, although not involved in the project. It is imperative that the law is not broken in this project IF 4.5
7: I am not as important as other users or stakeholders as this project is developed for them, not me. IF 1.0
8.1: This is not important at all, as he is not responsible for anything within the system and shall not be a factor in how successful it is. IF 0.2
8.2: Very important as this is what determines how well this project has been developed and the functionality of it as a success factor. IF 2.6
9: This group is not important, but still has a fair interest in the project. IF: 1.0

The weightings range from 4.1 to 45.1 with an average of 15.8 and a total of 994.3. These will be compared with the actual results in the evaluation section of this report. The higher the final score is (out of 994.3), the higher the success will be considered in terms of this group of people. There will also be a result for each of the user and stakeholder groups in turn to evaluate the success within each particular group

			Applicable to user: (W:will have,S:should have,C:could have,M: Might Have)										
num	**Need**	**How test**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8.1**	**8.2**	**9**	**Weighting**
	INPUT SYSTEM												
1.1	System shall be easy to use	3rd Party	S			M			M		C		13.6
1.2	System shall not allow invalid input	Test Data	S		C	C			S		S		21.5
1.3	System shall be attractive	3rd Party	C			M			C		C		12.5
1.4	System shall use appropriate colours	3rd Party	C			M			C		C		12.5
1.5	System shall minimize text input	Observation	C			M			S				8.3
1.6	System shall be able to record each ball	Observation	S			C			S				11.5
1.7	System shall record where ball was hit	Observation	S			C			W				12.5
1.8	System shall record all events that happen	Observation	S			C			S				11.5
1.9	System shall have no "lag"	Observation	S			C			S		S		19.3
1.10	System shall not allow contradicting events	Test Data	W		C	S			S		S		24.7
1.11	System shall update events when input	Observation	S						S				9.3
1.12	System shall have appropriate error messages	Observation	S						W		S		18.1
1.13	System shall automatically enter inevitable events	Test Data	W			C			S				13.6
1.14	System shall recognise possible input errors	Test Data	M		C	M			S				8.4
1.15	System should allow correction of errors	Observation	C			S			S				10.5
1.16	System shall be easy to install	3rd Party	C			S			M		S		16.3
1.17	System shall be a .EXE file	Observation	S			W			C				12.7
1.18	System shall save data to an external file	Test Data	W			W			W			M	17.8

1.19	Coding shall be well commented	Observation					S	M	W			13.6
1.20	Coding shall be split into appropriate functions	Observation					S	M	W			13.6
1.21	Coding shall be efficient	Test Data					S	M	W			13.6
1.22	There shall be a user manual	Observation	S			W		M	C	W		22.5
1.23	There shall be a one click interface to upload data	Observation	C		S	W		C				13.9
1.24	There shall be a troubleshooting section	Observation	C			W		M	C	W		20.4
1.25	Uploading shall be password protected	Test Data			S	W		C				9.7
1.26	The system shall include pictures of players	Observation	M					M			M	4.1
1.27	The system shall allow notes to be added to plays	Test Data	C		M	M		M				7.4
1.28	The system shall automatically rotate through players	Observation	W					S				11.4
1.29	The system shall allow substitutions of batters	Test Data	W			C		S				13.6
1.30	The system shall allow substitutions of fielders	Test Data	C					M				5.2
	DATABASE											
2.1	Data shall be stored in a relational database	Observation			C	S		W	C	W	M	21.3
2.2	Data shall not be stored more than once	Observation			S	S		W	M	W		21.2
2.3	Data shall be updated when it changes	Test Data			C	S	W	S	M	W		45.1
2.4	Data shall not be directly manipulated	Test Data			M	C		W				7.3
2.5	Data shall have clear field names	Observation						W	C	W		14.8
2.6	Data shall be accessible via Internet	Test Data		S	C	S		M			M	12.6
2.7	Data shall be compatible with current system	Test Data			S	S		S				9.6
2.8	There shall be no installation needed	Observation			S	S		S				9.6

	WEBSITE													
3.1	Website shall be attractive	3rd Party			C		C	W		M			C	M	16
3.2	Website shall be easy to use	3rd Party			S		C	W		M			C		16.7
3.3	There shall be a FAQ section	Observation			S		C	W		M			S		19.3
3.4	Website shall fit it with current site	3rd Party			C	S	C	S		S				M	15.3
3.5	Website shall allow queries to database	Test Data			W	M	S	S		S			S	M	25.4
3.6	Website shall allow generation of statistics	Test Data			W	M	S	S		S			W	M	28
3.7	Website shall not contain any sensitive information	Observation				W	S	W	W						36.9
3.8	Website shall allow users to see full details of all games	Observation			W		C	W		C				M	15.2
3.9	Website shall allow users to see stats on one player	Observation			C		M	S		C				M	9.9
3.10	Website shall allow users to see stats on all players	Observation	S			C	S		C			M		12.7	
3.11	Website shall allow production of comparisons	Test Data	M			C	S		C			M		9.3	
3.12	Website shall contain fixtures	Observation	C	S	S	S		C			M		15.4		
3.13	Website shall contain league tables	Observation	C	S	S	S		C			M		15.4		
3.14	Website shall not include copyrighted information	Observation		W	S	W	W						36.9		
3.15	Website shall increase in hits/day	Counter		S	W	C		S					12.3		
3.16	Website shall have no errors or missing pages	Observation	S	S	W	S		W		W	C		31.6		
3.17	Website shall only allow queries on games in past	Test Data			M			W		W			15.5		
3.18	Website shall not allow changing of data	Test Data		S	S			W					10.6		
3.19	Website shall include password protected admin page	Observation		W	W			M					9.8		
3.20	Admin page shall not be authorised by general public	Test Data		W	W		W	M					35.8		

3.21	Admin page shall allow manipulation of data in database	Test Data		S	W			M				8.7
	GENERAL											
4.1	Number of fans and players shall go up	Observation		W	S	C					W	13.3
4.2	Systems shall have similar aesthetics	Observation		M	M			C	M	S		12.2
4.3	Systems shall be developed with a recognised manner	Observation						S	C	W		13.8
4.4	Systems shall be consistant in style and functionality	Observation		M	M			S	M	S		13.2

TOTAL WEIGHTING:	147	65	81	149	44	104	150	3.4	231	20	994.3
Total interested in:	26	14	29	53	17	4	60	12	27	16	63
Average Weighting per need:	5.7	4.6	2.8	2.8	2.6	26	2.5	0.3	8.6	1.3	15.78254

1.8 Competition and alternatives

There are a number of products available to solve this product as well as the one that is being developed in this project. These shall be discussed in this chapter and compared to the proposed solution.

- *American systems*

  Currently there are a number of systems used by American companies to do a very similar function for the game in the US. They are bespoke to each company that uses them and so it is not possible (or certainly monetarily excessive) to use these systems. There are three main companies that use these systems but are all very similar and we can only see the general user view of these products and not how the details were arrived upon. CBS Sportsline [13] is one of the most widely used sports websites in the world. It has been updated every year and gets more functionality every season. By looking at this system it is possible to get an idea of how to present information, what information is looked at, which can help me to decide which fields to include in the database. MLB.com [14] has a very good statistics page and is the official page from which all records are extracted. The way that this is presented and the wide range of statistics available give us a good idea on what can be achieved with the new system. Finally, Fox [18] has a system similar to the previous two but not in as much detail. All three will be used when extracting features that could be used in this project, but one single system shall not be "copied". One feature that shall not be implemented due to hardware constraints that is common to these systems is that it shall not be a real-time system. Although this is desirable to fans, it is simply not possible to implement.

  For annotated examples of these systems and features that shall be considered see appendix 5.

- *Previously built input systems*

  There are a number of systems that are on the market to be able to record the scores of games as they happen. One such system is Score-it (see appendix 6 and [19]). This is a system for recording scores as they happen and printing out score sheets. However there are a number of disadvantages to this particular product; it has limited memory and so will not be bale to work with a database, it cannot produce certain stats and cannot be customized to fit the needs of this product. These limitations are common for most of the markets that have been developed for this purpose and so it is not feasible to use this method.

One of the better products on the market is one that is for a palmtop, which does allow storage of details in a database and is customisable. This is called ScorePad PAK and is available online [20]. Some screenshots can be seen in appendix 7. However this system cannot display the results as is requested in this project and cannot be used on laptops or PC creating a problem for the scorers, who do not all possess palmtop PCs. There is also a large monetary outlay of $150. Others similar to Score-it can be purchased cheaper (approx $30-$90).

- *Off-the-shelf package*
  An alternative to using a bespoke package, these shall be discussed further in section 1.11.1.

- *Update on current system*
  By keep using the current system and updating so that it is more consistent and can include some statistics is a viable option. However this shall include as much work as creating a whole new system and will not be as reliable and may confuse scorers on how to use the system. An update in the way results are presented on the website is also an option, although not a long term one because the results can only show what is recorded and if there is no information recorded about certain games then there shall not be sufficient information to display on the web site.

By taking a look at all these current available methods of solution and extracting the relevant features from each, it is possible to come up with a system that meets all the requirements of the company and does not have as much costs involved as some of the options mentioned above. Appendices 5 6 and 7 show the features that shall be adopted for the new system.

1.9 Constraints

The company has specified a number of constraints that need to be considered when designing and developing this product. There are also some that I have identified by looking at resources available and the current situation within the company.

The main constraint is that of money. There is no budget set out for this project and so the costs must be kept as low as possible. This means that it is not possible to buy extra hardware such as servers or laptops, although this is a consideration for the future if the product proved to be successful in the future. This, in itself, is another constraint; it means that the new system must be able to be run on the systems currently held (see section 1.10). There is also the constraint that whichever format the program to be developed for inputting scores must be platform independent and not need any additional software to be purchased. Ideally the

output of the program should be an .exe file so that it is easily installed and run on a Windows operating system, which is what is currently run on the laptops.
Finally there are some more general constraints that need to be adhered to such as legal constraints (section 2.9) and the constraints led out by Leeds University and the school of computing.

1.10 Current hardware situation

At the moment there is not a lot of hardware owned by the company, as there is not a great deal of demand for the use of computers. Currently, BSUK outsource most of the IT issues that need solving and are therefore not confident in obtaining more hardware for this project, unless it proves a success and requires this action. The website is currently hosted on a web based server and there are a few PCs in head office, which are used for simple maintenance and updating of information.
With regards to the scorers, most of them are in possession of a portable computer (either a laptop or palmtop). Those that do not have access to a laptop, have access to a PC and so it is safe to assume that this product shall be developed for use on a computer that runs Windows 95 or higher.

1.11 Available methods

There are a number of methods that could be used to solve this problem and each has its advantages and disadvantages.
The first of these is an off the shelf package application software such as MS access or MS excel. This would mean that new documents would have to be created and modified to reflect the needs and requirements of this project. This would mean creating macros and similar functions to record certain situations. This could include some programming in visual basic and using certain features that are included within these packages. There are advantages to using these packages and the already built in functions; the efficiency of such functions and the fact that these do not have to be programmed separately means that more time can be spent on different things. However, these functions are not as bespoke as they could be and do not do exactly as specified, this could prove to be very difficult and perhaps involve more effort than if it were programmed from scratch.
There is an advantage of using a generic package, it is available on most machines and most people are familiar with the layout of these packages and how they work so the training needed is minimal. Also a generic package developed system can be easily transferred between computers by copying and pasting the file.
The main disadvantage is the inability to fully personalise the system and this is solved by the second available method. This is to use a programming language that can be fully customised

to incorporate the needs of this system. This is a big advantage because anything is possible and any functionality can be achieved with some effort. A disadvantage of this is the difficulty of making such a system, it means that the amount of text that needs to be written is a lot and there will be no point and click interface that can be used to develop this system.
A program language shall be used to develop this system because the system that needs to be developed would be hard to implement on a generic piece of software, it would also not be able to do the wide range of things that are required by this project. The language that shall be used in this project is discussed further in section 2.4
The development of this system shall be done in a very structured way. It shall have clear chapter defined, as defined in the contents of this report and it shall flow from analysis of the problem and looking at the problem in depth, forward to designing and developing a solution and finally on to the evaluation of the product and how well it has performed. It shall also be supported by literature as books and the Internet and will use methods that are recognised and described in such sources. These are explored in more detail in section 1.4. Throughout this project there shall be evidence of cross-referencing to the different sections to stop the repetition of information. Also appendices shall be used for the inclusion of data and supporting sources because these would not fit in with the format of the report and are not necessary for the completion of this project.
It is important that Object oriented methods are used throughout this project, as stated in the requirements in section 2.2 and these methods will be sued whenever possible so that it is clear to see what is happening in the project and also to make it easy to follow when developing the system.

2. Producing a Solution

2.1 Minimum Requirements

The following are the minimum that this project needs to achieve in order to be able to start to evaluate it. If these are not met then the project is a failure.

- An input interface to allow users to input scores remotely
- An online system to allow uploading of scores by authorised personnel
- An online system to allow users to view scores and auto-generated stats
- A maintained database that keeps information about scores and statistics
- A detailed report which explains this development

2.2 Aim and objectives

The aim of this project as set out in the mid-project report is
Design and develop a system that records scores for baseball, which allows third party read back and produces ordered statistics.
The objectives of this project have not changed since the production of the mid project report and remain the as given below:

1. To establish the aims and requirements of the project
2. To produce an in-depth analysis and design for the above systems using relevant, object oriented methods
3. Create a simple input program. Usable on a laptop or PC, to record the events of a baseball game, as it happens.
4. This program shall be a "point and click" interface, making it very simple to use and therefore minimize training.
5. Create an input program, similar to that above, which allows input from a user that has recorded the details earlier and wants to write them to the system.
6. A simple uploading process, which the user can follow easily to save details to the database.
7. Produce a fully normalised database that can store the data input via the methods mentioned above.
8. Ensure that the data held in the database is kept up to date and complies with the data protection act (1998). Also ensure that data integrity runs throughout the database.

9. Produce a simple website that allows general users to look at the data held within the stored database, but not write to it.
10. To ensure data integrity throughout the whole system, input validation shall be implemented, wherever input is needed.
11. To ensure that the result of any queries on the website is not of a poor format. But is in a readable form, such as that used on American sites.
12. To ensure that the product does not contain any bugs or problems.
13. To critically evaluate the process and product with input from the company and independent third parties.
14. To meet with the company on a regular basis

2.3 Possible Enhancements

The following is a list of enhancements that could be implemented to extend the functionality or aesthetics of the programs. They shall only be implemented if time permits and shall not be a priority in this project. They have remained the same as shown in the mid project report.

- To make the input program usable on a palm top PC to make the input more versatile to the scorers
- A one-click upload process
- Extend the website so that it is more attractive and perhaps uses technologies such as Flash to increase the appeal of the website.
- To include sophisticated encryption within the data, to protect it.
- To offer a membership service – so that some features are only available to users that have logged on.

2.4 Hardware and software to be used

This software shall be developed on a system independent on the system that shall be running it. Therefore careful consideration needs to be taken when selecting hardware and software because of compatibility issues and also the wide range of devices that should be able to run the final product. The database can be stored in a number of ways, either on a server that is connected to the Internet all the time. It can also be stored on a central space on the Internet, provided by an ISP, or on a single PC. Storing on the PC is not really an option as queries can be executed on the DB at any time and so the information must be available all the time. Also the processing power of a standard PC is probably not enough to handle such a large amount of queries. The option of having a dedicated central server that we look after seems a good idea in terms of security and it certainly has enough processing power to handle and queries. However the major drawback in this approach is the cost that this would incur.

There would need to be a constant connection to the Internet, of high speed and would probably need a leased line for such a connection. Also the initial outlay of the server and maintenance of the hardware can rise over time and become uneconomical. This leaves us with the option of having the database stored on an ISP provided Internet based storage space. This option is less expensive than having a dedicated server but more economical than running it on a standalone PC. This also ties in with hosting the actual website, since there is already a website on the web that is hosted and up-kept by the company, this cost is already absorbed and the hosting of the database can be added to this domain. Since the input program is not going to be a live one and there may be simultaneous input, there is a requirement for some data to be stored on local machines. For example when in a game the scorer needs to be able to select the teams and players from the database but is not connected to the Internet. This means that a list of players and teams need to be stored on each scorer's machine and updated regularly. To reduce the need for extra storage and help keep data as consistent as possible, the data that is held locally shall be updated every time the scorer updates with only the minimum amount of information needed for the time until the next update.

There is also the issue of developing this database and testing it in a non-live situation. This cannot be done using the web-environment but must resemble this environment for realism. It is sensible to develop the database on a local PC and test locally using the other two elements of this project to bring it together. The database could be built and queried in a number of different query languages. The most commonly used language is SQL and as this is a universally accepted language, this is what I shall use. There are, however, many versions of this. Both proprietary and open source solutions are possible here but I shall use the free software, mysql so that when installing it on the company's machines no licences need to be purchased and it can be available on all computers without increasing monetary expense.

The actual input program also needs to be created using a programming language. Since the analysis and design of this project is using object-oriented methods, it makes sense that it is developed using OOP. There are a number of languages that I could use and will look at each in turn.

Java – A very good OO programming language but can be complicated at times, may need several files to be developed in order for it to work. Java has a good record of efficiency and many large software houses use this language, it is very versatile and good at handling files, however the handling of databases is not as good. My knowledge in java is very limited and believe it to be a hard language to learn and so will not use this language.

C/C++ – A high level language that can be messy at times. Very difficult to learn and implement a GUI with, however it is a very versatile language that is used a lot in industry, which would be a good reason to develop the project in this language.

Delphi/Pascal – Probably the best language in terms of my current knowledge, and also has the advantage of the final product being an executable file on any platform. However, it is not a good language for online documentation and support, so if I were to have some problems, it would be difficult to fix.

Visual Basic – This is a good language, which I am familiar with and has the advantage that it has a visual interface for programming, which helps in an object-oriented project. However, the main drawback of this language is that it is proprietary and therefore there will be an outlay of resources that the other languages will not have.

Python – A very simple, high-level language that has the possible extension of using wxpython to create a GUI. The online resources are very good for reading about things that are not in my knowledge. Another advantage of Python is that it has a facility to allow it to be converted into an .exe file, which is generic to all Windows platforms and so it means that the machine running the completed system will not need any specific software in order to run the program. Platform independence is a big plus in this project as a number of different computers with different specifications should be able to run the software. It also has the advantage that it can be embedded within a HTML script and could prove useful when connecting the number of applications together. One disadvantage of Python is that I am not overly confident of programming in it, although the online manuals and some previous experience do help a bit. This is the language that shall be used.

The middleware in this project is an important aspect, as it has to link all three applications together. There is the option of Apache and other similar products, or a manual system, in which the files are managed locally to the system that is running the program. Apache could prove to be difficult to program and may need some extensive research and learning, this is what I shall do before deciding on how to connect the systems.

The website also needs to be developed somehow and there is also a number of options for doing this. Firstly I could use a GUI development program such as MS Front Page or one given online by some ISPs. The other option is to hard-code the web site. The latter is probably the best option in this project as it is more customisable and I can use the features of HTML and JavaScript to the full extent to get exactly what I want from the system. Learning the basics of these languages is not necessary as I have a large amount of experience in building websites.

2.5 Product perspective

This project is not going to be a complete new system in its own right, but merely part of an overall system already used by BSUK. It is therefore not imperative that it works 100%, as there is already a system in place that can take over in the improbable case of failure of the new system.

The input program is to take over from the current input system and do the same job but faster, easier and more efficient. When this new system is in place, the old one will become obsolete and no longer be used by the company as there will be no need because the new system shall be capable of all that the old system was capable of doing.

The database that is being developed shall coincide with the current database and the old system shall not be phased as like the input system, it shall be extended to include new fields and become a more established (and larger) database, with the ability to create statistics and other analytical features that are required by this project. It is therefore not important to concentrate too much on the design of such a database, as one already exists (Although there shall be design given in section 2.12.1, this will incorporate the currents system's design and expand it as mentioned above), but to improve the way data is input, retrieved from and displayed by the database.

The website that shall be developed for this system is going to be independent of the website that is currently in place at this time. Although it shall form part of the website of BSUK, it shall not need to interact with the rest of the site. It may, however, replace some of the functions that the site is already used for such as league tables and/or fixture listings.

The most important aspect of the three systems that are going to be developed is that they integrate with each other and form a good backbone that future systems can be developed around, and the current system brought into line with.

2.6 Legal requirements

There are not a lot of particular laws that need to be adhered to in this particular project, however, since part of it is in the public domain and it is a company that deals with the public some regulation must be followed. As with all companies, it is important that the data protection act [21] is looked at and shall be observed in the development of this project. The way that this shall work is that no personal information shall be stored in the database and so sensitive information shall not have to be updated or kept in track with the DPA. Another aspect of legislation that must be considered is copyright, it shall not be the case that plagiarism takes place within this project. All sources that are used within this project shall be cited and referenced accordingly and listed in an appendix.

The information, illustrations and text on the website shall be as original as possible, in order to minimise the need for citing such information in the public domain. However, any such data will be properly cited and referenced on the website as well as in this report.
Further legislation that the company needs to consider but is outside the scope of this project includes tax laws, the computer misuse act [22] and competition laws.

2.7 Environmental and ethical requirements

Whilst there may seem to be no issues in this area, there is a few that need consideration when developing this system. One that may be overlooked is the affect on the environment of printing. Whilst it is not possible to eliminate paper output completely from this project, it is possible to minimize documents that may be printed by the public to one page. These could include individual statistics, certain game details and comparisons of players from the website.
It is important that this project is ethically correct and fits in with the values of BSUK. Therefore the products that shall be developed within this project will not be used for other purposes than that given so that they cannot be abused. There shall also be constraints put on the system to ensure fair play within the game of baseball and that no cheating can take place as a result of the new system being implemented. It is also important that this system cannot be used to gather information on players, fans, officials or the organisation itself and BSUK prides itself in the confidentiality and security of all its stakeholders. For more information on this see section 2.6 and 2.8.

2.8 Functional Requirements

This chapter concentrates on the functionality of the system and the errors that can be prevented by simply enforcing some constraints. The requirements below ensure that the system works and minimises human errors and the logic behind each requirement.
The main reasons that stand out are to reduce human input; this is a good reason because by reducing the amount that the user is responsible for, it reduces the possibility of the user making an error and improves the value of the data in the database. Another main reason for enforcing these constraints is to reduce errors further on in the system. By sorting out the inconsistencies at input and checking the input against certain properties, errors that could be more serious later on can be stopped easily. This will reduce the costs associated with putting these errors right. Finally the other main reason behind these functional requirements is to keep with baseball scores. This means that the rules of baseball are incorporated within the system and will not scorers break these rules, which will make the product more appealing to BSUK and other future users of the system.

Num	Requirment	Reason	
	INPUT PROGRAM		
1.1	Can not put details in before selected teams	Stop confusion / ensure some teams are entered / Helps 1.3, 1.8	1
1.2	Can not have two players on same base	Reduces redundancy / eliminates overlapping labels / Keeps with baseball rules	1
1.3	System should cycle through players automatically	Reduces human activity / reduces errors / ensures players bat in order	1
1.4	System should remember the place of batting order	ensure players bat in order / reduce human activity / keeps with baseball rules	1
1.5	System should automatically increase inning	Reduces human activity / speeds up system / keeps with baseball rules	1
1.6	System should automatically increase scores	Reduces human activity / reduces errors / can use to validate	1
1.7	Should not be able to manually increase scores	Eliminates possibility of unsportsmanlike conduct / Eliminates possible errors	1
1.8	Should not be able to insert "non-real" players	Eliminates errors / stops biased scorers	1
1.9	System should not take more than 5 seconds to load	Quality of program / eliminates feeling of "has crashed"	0
1.10	Data should not take more than 10 seconds to upload	Quality of program / eliminates feeling of "has crashed"	1
1.11	Should not be able to activate inactive window	eliminates confusion / stops errors of two events conflicting	0
1.12	Should be cancel button on all pages	To allow easy navigation / To be able to correct errors	0.5
1.13	Should confirm before writing to file	To ensure what is written is correct / eliminate future errors	0.5
1.14	Should automatically record out when 3rd strike	Reduce human activity / keeps with baseball rules / sppeds up system	1
1.15	Should automatically advance batters when 4th ball	Reduce human activity / keeps with baseball rules / sppeds up system	1
1.16	Foul should increase strikes except on 2 strikes	Reduce human activity / keeps with baseball rules / sppeds up system	1
	WEBSITE		
2.1	Should be no text input	Minimise ambiguity / reduce errors when querying DB	1
2.2	Automatically populate player combo box when team selcted	Ease of use / Eliminate confusion/ Helps with 2.7	0.5
2.3	Should give all stats on one particular player	Quality of website	1
2.4	Should have page for each match taken place	Quality of website	1
2.5	Can not write to database	Stop errors / eliminate "cheating"	1
2.6	Should be two pages with search facilities – player + match	Reduce confusion / quality of website / reduce errors when querying DB	1
2.7	Can not select player without first selecting a team	Reduce confusion / stops redundancy / ensures player exists	1
2.8	Should be a button to produce printer friendly sheet.	As in environmental requierments / Ease of use of website	1

2.9 Summary of capabilities of proposed system

Since this project is not a complete system in its own right, it must have some limitations that are in place. These limitations mean that there are only so much that this system can accomplish. The new system that shall be developed shall not be able to do everything that the company does on a regular basis and is just limited to the scoring aspect of the business. It shall not be able to deal with the financial aspects of BSUK or the human resources of the company. Is shall also not be possible for the details of a game to be updated and recorded in real time; this is due to the hardware capabilities and the lack of internet connection available to all scorers at games.

It shall not be possible with the new system for new players to be added remotely, this must be done centrally via the website. The website itself shall also have limited capabilities; it is only possible for the public to query the database from the website with the given parameters and will not be given a free query builder to execute as they wish. This is to ensure that the queries sent to the database will always be valid.

2.10 Costs

There shall not be a lot of costs associated with this product and it should be produced with a minimal budget in mind. The development of this project shall be done on a system that is already in place using software that is free or already acquired. Therefore there are no costs in actually developing the system in terms of money. There is obviously the issue of time and it is estimated that this shall equate to approximately 500 hours. The software required to run the new system shall have some costs involved. In order to run the system, the computer must be running a version of windows. This is the only additional piece of software that is needed and most of the current computers are already equipped with this and so there is not much cost in total. The cost of buying a few more computers may have to be absorbed to ensure that all scorers have access to this product; this shall not be a great deal.

Another cost, perhaps the main one is that of storing the data in the database, as there needs to be a large server to cope with the large amounts of data that shall be stored. Although this could be a large outlay, there is already a server in place that can cope with most of the requirements of the new system and perhaps a smaller server needs to be purchased if this system is successful.

The final cost that needs to be considered is the cost of actually using the system. Where this cost is not high, it must be considered. The main contributor to this is the cost of uploading and downloading from the Internet to the input program. This is minimal and the facilities for this are already in place so this can be dismissed.

2.11 Installation and training

Because this system shall be very simple to use, it is probable that not much training is needed. Also because the people that shall be using the input system are familiar with the general rules of baseball and how it is scored, there is no need for generic system to teach them this. The system shall be created in such a way that anyone familiar with the game of baseball can pick it up and use it. However, there will be a user manual developed for the input system (appendix 9), as a step-by-step guide to rookie scorers and people that are trying to learn the game.

However, I shall supervise some of the scoring within the first few weeks of the system going live in order to look at potential future problems and to help with any small queries that the scorers may have. This is also for my own personal gain, as I would like to see the system working in a live environment.

There shall be minimal training for the use of the website, as the users are undefined as such and it would be difficult to teach all baseball fans how to use the system. The solution to this problem is to offer an FAQ section and troubleshooting guide, as well as making the website as easy to use with the facility to eliminate errors before they happen (see section 2.8).

The installation of the input system shall not be performed by an installation program, but will be a zipped file in which all the files are contained. This means that the installation process shall be quite straightforward. The website shall not need to be installed, but merely the HTML and image files uploaded onto BSUK's server or simply a link to the server that I shall be developing the web site on.

The database will need more complex installation as it will be developed on the University's server, it must be transferred to BSUK's site and used on their server to avoid misuse. This process may lead to other problems as well in terms of retrieving data from the database from the website or input program as the location of the database will already be stored in the system and so this will need to be tweaked in order for the system to work effectively. This shall be done personally, by altering the code when the system has been fully tested and gone live.

2.12 System Design

2.12.1 Database design

The central part of this project is the database, it needs to be properly designed, normalised and be as efficient as possible in terms of disk space and retrieval time. The fact that this project is based on a sport gives us the most important table of the database, PLAYERS. The

attributes that need to be included are listed and described in table 1, appendix 17. This could be represented in 0NF as below, but is very inefficient and will result in a great deal of data repetition and does not allow for referential integrity. Therefore, we need to normalise this further, as shown in 1NF, where it is broken down further so that no data is unnecessarily repeated and the data is represented in more defined tables that more accurately represent the data.

0NF

PLAYERS (player_id, forename, surname, team_name, team_location, Team_colour1, team_colour2, contact_for_team, Contact_email, Contact_tel, league, position, team_wins, team_losses, for, against, player_pos, hits, RBI, HR, 2B, 3B, HR, BB, IBB, E, SB, CS, Free1B, AVG, Slugging, OBP, AB, ERA, player_wins, player_losses, saves, IP, CG, Shutouts, Pitch_BB, K, R, ER, HR_allowed, Opponent, home_score, away_score, winning_pitcher, losing_pitcher, save, date, inning, inning_home_runs, inning_away_runs, 1stBase, 2ndBase, 3rdBase, Home, Out, HitTo, RBI_atbat, replacement_id, replacement_place, count, pitcher_id)

1NF

PLAYERS (player_id, forename, surname, pos, hits, RBI, HR, 2B, 3B, HR, BB, IBB, E, SB, CS, Free1B, AVG, Slugging, OBP, AB, ERA, wins, losses, saves, IP, CG, Shutouts, Pitch_BB, K, R, ER, HR_allowed)
TEAM (team_id, name, location, colour1, colour2, contact, Contact_email, Contact_tel, league, position, wins, losses, for, against)
MATCH (match_id, home_team, away_team, home_score, away_score, winning_pitcher, losing_pitcher, save, date)
INNING (inning_id, home_runs, away_runs, match_id)
AT_BAT (at_bat_id, inning_id, player_id, batter_num, 1B, 2B, 3B, 4B, Out, HitTo, RBI, replacement_id, replacement_place, count, pitcher_id)

This is much better and is also reduces ambiguity and so some names can be shortened, e.g. inning_home_runs becomes home_runs because it is already associated with the INNING table. This still needs normalising further to reduce certain ambiguous fields such as at_bat_id, which are meaningless except to preserve uniqueness, this is done with composite keys. Also we can remove some fields to make each record smaller in size, these fields can now be calculated by using other fields and Query language/mathematical functions, as shown in table 2, appendix 17. This must be done in order to fully normalise the database, as

these could be dependant on other values from other tables. However, there is the issue of retrieval time that shall be discussed later in this paper.

2NF / 3NF

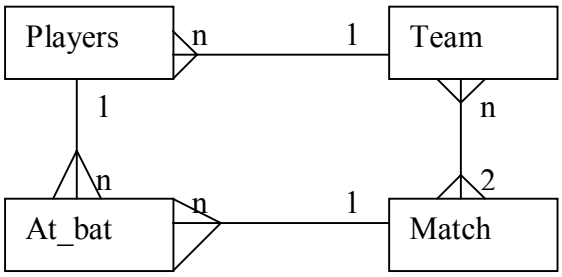
PLAYERS (player_id, forename, surname, pos, E)

TEAM (team_id, name, location, colour1, colour2, contact, Contact_email, Contact_tel, league, position)

MATCH (home_team, away_team, winning_pitcher, losing_pitcher, save, date)

AT_BAT (inning_num, player_id, batter_num, match_num, 1B, 2B, 3B, 4B, Out, HitTo, replacement_id, replacement_place, count, pitcher_id)

This schema is now in BCNF and does not need to be normalised more as there is no attribute that is dependant any other non-key attribute and only dependant on the PK(s) of the table in question. The inning table has been removed due to its irrelevance and the possibility of including its attributes in other tables and to derive them from other tables as well.
This now needs to be represented as an entity relationship diagram in order to give a good idea of how these tables interact with each other and how each table is linked with others.
This ER diagram is shown below:



This E-R diagram may seem pretty simple, which is a good thing since it shall be easy to maintain. Although this is the best E-R diagram for representing this database, it could be broken down further as there is a many to many relationship (many to 2); since this is always two, it is not necessary to break it down as it will result in more storage being needed and the complexity increasing.

We now must discuss the retrieval time in order to decide whether the stats should be stored within the PLAYERS table or whether it is sufficient to leave it within the AB table and perform a calculation every time a query is executed.

If the statistics were stored in the PLAYERS table, there would need to be a huge increase in storage space, as there would need to be a lot of extra fields in this table. There is also the issue of updating these stats when they need to be. This could mean data redundancy due to the deadlock scenario if two people are uploading at the same time. The time taken to upload the results of a game would be significantly increased as there would need to be a number of calculations done for each player on each side and accumulated stats calculated taking into account the previous totals:

13 batting stat categories * at least 18 batters * at least 9 inning * at least 3 calc/queries (^)

+

12 pitching stat categories * at least 2 pitchers * at least 9 inning * at least 3 calc/queries (^)

=

More than 6966 calculations or queries per game

(^) – 3 calculations or queries is estimated as the first to retrieve the original figure, the second to query the AB table to get the adjustment and the third to calculate what the new figure should be. This will be more for some stat categories such as AVG and OBP.

This is a lot of calculations per game and it is estimated that the actual figure would be closer to 10,000 per game when all the additional figures are included on top to include substitutions and the more complex stat categories.

The alternative to this is to eliminate these calculations and to do the calculations every time a query is executed on the website. This would result in more queries and calculations being produced:
First returning all the AB that the player has had will give a new reduced temporary table, which can then be queried to get stats.
Secondly, for each stat do a calculation on one or more columns to get a resulting figure.
This will return an online profile of the player that users can look at.
Clearly this is an improvement on the earlier method even though it is querying a lot of data (a player can have up to 300 AB per season) it is very efficient and even if it was to look at one row at a time and one stat at a time (300 * 13 * 2 calculations) it would be very similar to that above per game. However this is not the case, it is more efficient and so can dramatically improve the efficiency and effectiveness of this query.

Due to these results it seems the sensible thing to keep statistics stored in the AB table and calculated when needed. This is because of the efficiency it provides and the space that it saves. It could be beneficial to store stats in the PLAYERS table if they were to be used elsewhere apart from the website such as in the input program but this is not the case.

2.12.2 Screen Designs

INPUT PROGRAM

The designs referred to in this chapter can be found in appendix 10.

The first thing that the user is presented with when the input program starts is the main page (a); this allows the user to choose the teams and also switch between the different tabs of the system. It is a fairly basic screen in order to give the user a clear idea of what to do. Combo boxes are used to make input easier and eliminate errors. The colour is consistent throughout and is dependant on the user's Windows settings; this means that the colours will suit the user and it will seem like a windows program is running.

The user can then go to input the details of the game, by clicking on the relevant tab that appears when the teams are entered. This screen (b) looks fairly complex with a lot of options on it, but it is quite simple to use and the buttons are lined up in order of frequency used to help the user when inputting the scores. It has also been decided that this screen should show where each runner is around the bases because the scorer would like an idea of what is happening and whether any errors need correcting. Abbreviations are used on this screen to de-clutter the screen and to make it more condensed; it is a lot shorter to have a button saying "KS" than "Strike swinging". This will not cause confusion at all because the user will know these abbreviations like second nature, as they are standard baseball notation.

When a ball is put in play there is a screen displayed (not showed in appendix) that is simply a picture of a baseball pitch for the user to choose where the ball was hit. This is done by a single mouse click for simplicity and speed. After the position has been picked the AB screen is displayed (c), which is separated into SAFE and OUT for the user to quickly identify which section of the screen the correct play is used. The more common ways of putting a ball in play are displayed, but not all of them due to the fact that this would look too cluttered; these are included in another screen that the "OTH" buttons lead to.

The next screen that could be produced for the user is how the batter was putout. This is a very small screen and it only requires numbers to be input and a button pressed. Again, this is standard baseball notation that when a player is out the number of the player that got him out is input. If the player was safe or a sacrifice has been made then the user is presented with an "advance runners" page (d). This is as simple as possible, although still quite complex due to

the nature of the operation. The combo boxes only allow valid input and are defaulted to what would usually happen given the play defined in the previous screen to help the user in terms of speed. Usually all that will happen on this screen is to press OK, but it may differ if an unusual play has happened or there is a particularly fast/slow player on base.

The final major screen on the input program is the Upload / Download page (e). This is a very simple page to make it easier for the user. The two buttons contained on this page are very straightforward.

Throughout the input program, new screens are popped up, rather than take the place of the old one. This make navigation easier, a user can simply click on the X button to go back a stage. It also allows the user to see their current position within the system and the path they have used to get there, this eliminates the need for a status bar or a "you are here" display.

WEBSITE

The website shall also be simple and as easy to use as possible. There shall not be many pages in order to preserve this simplicity and there will be no scroll bars so that everything fits on the one page. The colours used on the website shall be consistent with the website of BSUK to give the impression that it is the same site and to promote confidence in security.

When the user enters the URL or clicks on the link to the website, they will be presented with the home page (f). This is a very simple page with two main links on it, which lead to the search pages. The first search page (g) is to search for players, this is split into to combo boxes, teams and players to make it much easier to use; it would be very difficult for a user to select a player from 1000 and much easier to select the appropriate team and then select the player from 25. The combo boxes shall be sorted alphabetically, to further enhance the ease of use. The second search page (h) is very similar to the first one; this is for consistency and to stop the user getting confused. By separating into weeks and then games, it reduces confusion of which game they are looking at because teams may play each other more than once. If it is possible, then these two search screens will become one separated by a vertical line. This will eliminate the confusion of which search page the user is looking at and cut down on the number of links. The results of the player search shall be displayed on a page (i), which will popup from the current window. It pops up to allow easy navigation and so that the user can keep it open whilst searching for another player/match, which allows the user to make comparisons.

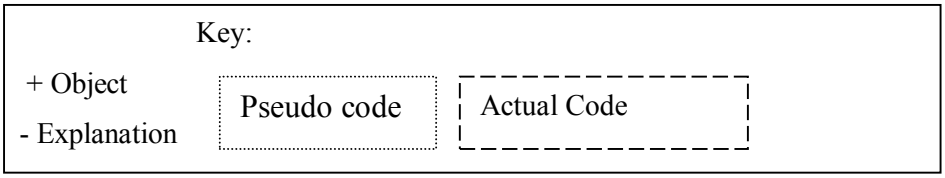
This page shall be formed in such a way that it is easy to see the main stats such as AVG and HR but other stats will not be as prominent. The presence of a printer friendly button allows the user to print the stats on half a sheet of A4.

The other results page (j) shall also pop up but is more complex. The combo boxes on this page allow users to easily and quickly select what they want to see. On links on this page shall link to some pages that are not covered in this project as well as to more detailed information on the game in question.

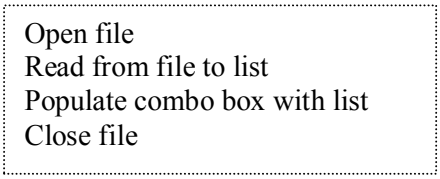
The final page of the website is the admin page. This is access by a username and password via a small link on the main page. This is to stop users paying much attention to it and so reduce the likelihood of a security breach. The main admin page shall be very small and shall only perform the main duties that need to be done.

2.12.3 Functionality Design

This chapter explores the input program in detail and designs the actual coding of the input program. Below is a hierarchical diagram showing the main items within the system and the workings behind them.



+ Main Screen
  + Game Overview tab
    + Home team combo box
      - Populated from a file on the user's computed acquired from upload screen



    + Away team combo box
      - As home team
    + OK button
      - Checks for errors and saves two teams
      - Creates lists of players
      - Initialises game

```
IF combobox IS NULL
        Display error
ELSE IF home_combo_box = away_combo_box
        Display error
ELSE IF game already started
        Display error
ELSE
        Home_team = home_combo_box
        Away_team = away_combo_box
        Open Players file
        WHILE NOT EOF
                IF player_team = home_team
                        Home_list = Home_list + player
                ELSE IF player_team = away_team
                        Away_list = Away_list + player
        END WHILE
        Close Players file
        Displays Label 1
        Display Game Details screen
END ELSE
```

+ Label 1

  - Asks the user to start inputting details

+ Game Details tab

  + Labels

    - A label for each base to show the player at this base, is null if no player on base

    - A label saying who is pitching

    - A label for the person who is batting, this is in bold to stand out

    - Labels for headings for balls, strikes and outs.

  + “KS” button

    - Adds a strike to count and displays and calls procedure drawCount (1)

  + “KL” button

    - As “KS”

  + “BB” button

    - Adds a ball to count then if balls != 4 calls procedure drawCount (1)

    - If ball is 4 then moves player to 1st base and advances any runners

  + “FL” Button

    - If strikes != 2 then adds a strike to count and calls procedure drawCount (1)

    - If strikes = 2 then does nothing

+ "OTH" button
  + Low frequency plays screen
    + Out buttons
      - A number of these are given, different ways that a batter can be out. These will not be used very often. They set the current batters OUT attribute to the text on the button and calls procedure Write_to_file (2), increases the number of outs and calls procedures get_next (3) and drawCount (1)
    + Free base buttons
      - These buttons are way that a batter can be given a free base; this calls the procedure that deals with walks as it does the same thing.
    + Substitution buttons
      - PH and EJC means that the player has been taken out of the game and therefore needs to be substituted
      + Player combo box
        - A drop down list of players not in the top 9 of the list of players
        - Only appears when substitution option is chosen
      + OK button
        - Sets current player's replacement attribute to the value of player combo box.
        - Set current player's replacement place attribute to the text on the button pressed.
        - Calls procedure write_to_file (2)
        - Replaces player AB with the player selected.
+ In play button
  - Sets current batter's count attribute to current count.
  - Resets count
  - Calls procedure draw_count (1)
  + Positional Screen
    - Records the position of where ball is hit by saving the co-ordinates of the mouse when it is clicked. Saves this value to the current batter's HitTo attribute
    + Select play screen
      + Out buttons
        - Either link to catch frame or putout frame

- Sets the current batters out attribute to the text on button pressed

+ Catch frame

+ Text box

- User enters number of player who caught batter

- On Exit checks if it is a single number

+ OK Button

- Adds catcher of ball to current players out attribute:

```
Out = out + textbox_value
```

- Calls procedure Write_to_file

+ Putout frame

+ Text box

- User enters combination of batters out

+ OK Button

- As catch frame

+ Safe buttons

- Sets the default values of the combo boxes of the advance runners screen according to which button were pressed.

+Advance runners screen

+ Combo boxes

- These are populated by the possible advancements of players and defaulted to the values defined previously.

- Cannot be erroneous data typed into these

+ OK button

- Writes any runs that have been scored to the file and advances the runners according to what has been selected in the combo boxes (4)

+ Upload / Download tab

+ Upload button

- Calls procedure upload (5)

- Uploads the saved game to the server in order to be saved to the DB

+ Download button

- Downloads the up to date tables fro players and teams in order to replace the ones locally.

(1) Draw_count

```
All bitmaps.visible = FALSE
IF outs > 0
        Out_bitmap_1.visible = TRUE
        IF Outs = 2
                Out_bitmap_2.visible = TRUE
        ENDIF
ENDIF
IF strikes = 3
        Balls = 0
        Strikes = 0
        Player.out = struck out
        Get_next()
        IF outs < 2
                Increase outs
                Out_bitmap_1.visible = TRUE
                IF Outs = 2
                        Out_bitmap_2.visible = TRUE
                ENDIF
        ELSE
                Increase inning
IF strikes >0
        Strike_bitmap_1.visible = TRUE
        IF strikes =2
                Strike_bitmap_2.visible = TRUE
        ENDIF
ENDIF       ## Continued
```

```
IF balls > 0
        Ball_bitmap_1.visible = TRUE
        IF balls > 1
                Ball_bitmap_2.vsisble = TRUE
                IF balls = 3
                        Ball_bitmap_3.visible = TRUE
                ENDIF
        ENDIF
ENDIF
```

(2) Write_to_file

```
# first make tuples into a string
(b,c) = batter.gethitto()
hit = b + “, ” + c “, ”
(d,e) = a.getcount()
count = d + “, ” + e “, ”

# then make one long string with all info in
thestring =  “(”+inning + “, ” a.getname() + “, ”+a.getmatch() + “, ”+ a.getout()+
##  This carries on for each attribute
+ a.getpitcher() + “) \n”

# Then write it to file
thefile = open (‘data’, ‘a’)
thefile.write(thestring)
thefile.close()
```

(3) Get_next

```
IF home_team_AB
        Player_at_bat = home_list.pop(0)
        IF last_play = OUT
                Home_list.push(Player_out)
        IF player_at_4 NOT NULL
                Home_list.push(Player_at_4)
                Player_at_4 = NULL
ELSE
        Player_at_bat = away_list.pop(0)
        IF last_play = OUT
                Away_list.push(Player_out)
        IF player_at_4 NOT NULL
                Away_list.push(Player_at_4)
                Player_at_4 = NULL
ENDELSE
```

(4) Advance runners OK button procedure

```
IF two players on same base ## see appendix 11 for code
        Display error message
ELSE
        Save batters at current positions
        IF player_at_4 IS NOT NULL
                Write_to_file()

        ## See appendix 12 for full code
```

(5) Upload

```
Display Confirmation Box
IF confirm
        Establish connection with DB
        Confirm User name and Password
        OPEN data_file
        Thedata = "INSERT INTO database.AB VALUES"
        Thedata = Thedata + data_file.READ
        SEND Thedata AS REQUEST
        RECEIVE confirmation
        CLOSE data_file
        CLOSE connection
ENDIF
```

2.12.4 UML diagram of proposed system

This is contained within appendix 13 and its shows a big improvement on the system that is currently in place and has been illustrated in appendix 2. The activity diagram is a lot more streamlined with not as much choices and parallel activity going on. The use case diagram is also clearer and there is a clearer idea of who can do what and how. There are some things that have changed very slightly such as the component diagram, which now includes the new system and website. The class diagram also now represents the database schema to an extent and is of a more organised structure to the uncertain diagram in appendix 2.

2.13 Project schedule

Below is the schedule that has been adhered to in this project, this has been adjusted and tweaked at various times throughout the project and this is the final version. It has changed dramatically since the original schedule was drawn up in the mid project report due to unforeseen circumstances. Some earlier drafts of this schedule can be found in appendix 14.

W/C	**Work done / to be done**
20/09/04	None
27/09/04	Decide on project idea and communication with company
04/10/04	Start looking at aims and requirements
11/10/04	Meet with company

18/10/04	Establish aims and requirements
25/10/04	Analyse environment / meet with company
01/11/04	Analyse environment
08/11/04	Look at current solution
15/11/04	Analysis current solution / Collect relevant background material
22/11/04	Start input system development
29/11/04	Start reading and analysing literature
06/12/04	Write up literature review / Write mid project report
13/12/04	Write up stakeholder needs
20/12/04	Extend stakeholder needs
	CHRISTMAS AND EXAM PERIOD
24/01/05	Look at current alternatives / analysis limitations
31/01/05	Write up Method justification
07/02/05	Establish requirements
14/02/05	Establish requirements
21/02/05	Design System
28/02/05	Design System / Meet with company
07/03/05	Design System / Set up database
14/03/05	None – holiday
21/03/05	Establish evaluation criteria
28/03/05	Start website construction
04/04/05	Complete Input program / complete website construction
11/04/05	Evaluation
18/04/05	Evaluation
25/04/05	Personal reflection / Proof read / Submit
09/05/05	Meet with company
16/05/05	Meet with company
01/08/05	Meet with company

3 Evaluation

3.1 Assessment criteria

In order to decide whether this project has been successful, it is necessary to determine some criteria that quantifies the success and gives a good idea of what could be improved upon. This also gives a good idea of what to aim for when designing and implementing the products.

The first test of success is whether the users and stakeholders are happy with the product. They are the most important people within this project and it is imperative that these are pleased as much as possible. This will be measured by the needs specified and weighted in section 1.7. This will give an overall score, which will tell us how well the product meets these needs.

Another test, which is separate to the needs of users, is to judge the functionality and how well the product can function given a list of requirements that it needs to fulfil This is important because it is imperative that the system can work fully without errors and without user intervention when it is not needed. Whilst these requirements can sometimes overlap with user requirements, it is important that they are given a separate section. These requirements have been set out in section 2.8. The way that these shall be measured is a simple yes or no. Since each of these requirements is roughly of the same importance there is no need to assign different weights to each of these. This project will be considered a success in this area 85% of these are met and a failure if less than 60% are met. These requirements are more important than any other criteria and so must be looked at first when deciding whether this product was successful or not.

Another measure that is not as rigidly quantifiable is the perceived success of the product. This will be achieved by asking different users and non-users to rate the system out of a possible 10 for the following categories:

Aesthetics of input system – It is slightly important that this is considered because a user will not want to use the system if it is hard on the eyes or difficult to see fonts/text. Weighting of overall score: 15%.

Aesthetics of website – more important than the previous because it will be used by more people and it is used by choice and so the system will not be used if this is not met. Weighting of overall score 20%.

Ease of use of input system – It is important the input system must be easy to use and an improvement on the previous system otherwise the user will just resort back to the previous system. If it is easy to use, there is also a larger possibility that new scorers will be recruited to the league. Weighting of overall score 25%

Ease of use of website – This is also important for the same reasons as the aesthetics. It is important that fans can do what they came to the website to do, otherwise they will not use the system again. Weighting of overall score 20%

Speed of all systems – This is the speed at which the systems can be navigated, to reduce waiting time and frustrated users stopping using the system. It is not a huge factor due to the fact that it is very difficult to cut this down and it will not vary very much. Weighting of overall score 5%

Ease of navigation - Mainly on the website, but also includes the input system. It is important that users can navigate easily and get to the correct part of the system; otherwise they will become confused and lose faith in the system. It is no use the system working perfectly if the user cannot navigate to the correct section of such a system. Weighting of overall score 15%

This gives a good idea of how well the system is presented and how likely a user is to want to use the system again. This will give a score out of 100 when weighted appropriately. It then needs to be adjusted to how important the person testing it is. The people that are to be asked on this and the importance of each are:

A non-interested party – This will give a general overview of how well the system looks and feels for someone that does not know a great deal about the sport or system. It is not important that this person is pleased with the system but still should still be considered. This person's score will count 10% towards the overall score.

A general baseball fan – This will be the general users of the website and it is important that these are pleased with the way the system is presented. This also a representative of the entire baseball fan base and could be used to see whether this will increase the fan base. This person's score will count 40% of the overall score but the questions on the input system will be excluded and the website questions weighting doubled.

A scorer of baseball in the UK – The main user of the input system, it is important that this person is pleased with the resulting system. If they are not, then they will probably not use the system and this project shall be wasted. This person's score shall count 50% of the overall score, but questions on the input system excluded and the weighting of the input system questions doubled.

This score and the two other scores mentioned above would then give a strong idea of how well this product has been developed.

Finally, there will be an explanation and evaluation of how well this project has progressed in the eyes of the author. This shall be expressed in terms of a qualitative description in the rest

of this evaluation section of the report excluding section 3.4, which will concentrate on the results of the quantitative measures mentioned in this chapter.

3.2 Evaluation against minimum requirements

When the minimum requirements were set out in this project, it was important that these were the first objective of this project. These requirements have been met and this is explained below.

- An input interface to allow users to input scores remotely
  - This has been accomplished with the development of the input system and the facility to do it remotely is upheld by creating it as an EXE file.
- An online system to allow uploading of scores by authorised personnel
  - This has been incorporated in the input system and a single button on this system calls the uploading procedure. It is restricted to authorised personnel by including a user name and password dialogue.
- An online system to allow users to view scores and auto-generated stats
  - This has been accomplished by the development of the website and the ability to query the database and retrieve the stats and information requested by the user
- A maintained database that keeps information about scores and statistics
  - This has been achieved by creating the database that is centrally stored and updated every time a score is uploaded or the admin page is used.
- A detailed report which explains this development
  - This document explains the development of the system in detail and gives a good grounding onto how it was designed and evaluates the result.

All of these minimum requirements have been met and so the project has succeeded in terms of passing these.

3.3 Evaluation against aims and requirements

The expectations of this project have been met in terms of what was the minimum that needed to be achieved. It is, however more success than that as it has also met the requirements set out in section 2.2. The numbers used below reference the requirement number given in section 2.2.

1) This has been achieved in sections 2.1 and 2.2.

2) This has been achieved in the first main sections of this report, "Understanding the problem" and "Producing a solution". Evidence of object-oriented methods can be found in appendices 2 and 13 and section 2.12.3.
3) This has been achieved by the development of the input system. It can be used on any computer that is running Windows.
4) This has been achieved. The input system is point and click and the majority (90%) of actions are performed by mouse clicks.
5) This has not been achieved. It was discovered that this was not as important as some other functionality. The input system is built in such a way that this requirement can be achieved using the given input system.
6) This has been accomplished by using a single button for the uploading procedure
7) This has been achieved with development of the database, evidence of the design and normalisation can be found in section 2.12.1
8) This has been achieved to a degree. There is no need to comply with the data protection act as highlighted in section 2.6. The data is constantly updated with each upload and the data stored locally on the user's machines is updated with each download.
9) This has been achieved with development of the website. Users cannot write to system due to the restrictions put on the database and the requirement of a password. Users can read from the database and query it using the search facilities on the website.
10) Input validation is not in place on every input field but the absence of text fields and the use of combo-boxes does this job just as well.
11) The website shows the results of queries in a good format, it is not merely a table but is returned in a similar manner to American systems as shown in appendix 10, i) and j) and is explained in section 2.12.2
12) This has been done as much as possible and no clear bugs are visible through the testing that has been undertaken. This is also shown in sections 2.7 and 3.4.
13) This has been achieved by section 3 of this report, "Evaluation" and specifically in sections 3.1 and 3.4
14) This has been achieved, although not to an extent that could have been achieved. This is shown in section 2.13.

This chapter has shown that the project has been a success in terms of meeting the requirements and this report is evidence of this along with the development of the website, input program and database.

3.4 Evaluation against assessment criteria

Firstly it is necessary to see whether the needs of the users and stakeholders have been met with the development of this product. This is done by looking at the list of stakeholder needs expressed in section 1.7 and filling in the ones that have been met to come up with a final overall score for this section. This is done in appendix 15 and the results are shown in the summary below.

Input system – Total needs met: 20.3 out of 30

Input system – Total weighted score: 299.1 out of 403.6

Database – Total needs met: 5.9 out of 8

Database – Total weighted score: 109.6 out of 141.5

Website – Total needs met: 14.95 out of 21

Website – Total weighted score: 309.91 out of 396.7

General – Total weighted score: 33.9 out of 52.5

Users total percentage met

1) 77% 2) 68% 3) 72% 4) 75% 5) 66% 6) 94% 7) 73% 8.1) 69% 8.2) 75% 9) 64%

Total number of needs met: 43.75 out of 63 (69%)

Overall score out of 994.3: 752.619 (76%)

These figures are encouraging and it sows that no group users or stakeholders will be disappointed with the resulting product. It is good that user 6 (British government) has such a high percentage as this indicates that no laws will be broken in the using of this program. The lowest score of 64% is not a great contributor to this project and is still not dissatisfied and will be pleased with the result of this program. Almost 70% of needs have been met, which is a large number since it does not mean that 31 out of 100 have not been met, but the needs have been met to a quality of 69%. In fact there were only 5 needs that were not met at all (7%) and the rest were passed to some degree. It is significant that the overall score is greater than the number of needs met because this shows that the effort that has been made was made on the more important of needs.

It is then necessary to look at the functionality requirements set out in section 2.8. The results of this can be found in appendix 16 and shows us that the input system has a total of 13 out of 16 functional requirements met. This is 81%, which is slightly under the success rate established in section 3.1, although it is no way a failure, as it is still a very high percentage and is a good starting point, from which future development can go forward from. It also shows us that the website has a score of 7.5 out of 8, which is 93%. Therefore, this part of the

system can be considered a success in terms of functionality and the number of errors that occur. These two averaged out is 85%, just on the success rate set out previously; this does not make the input program a success, but it means that the overall software development part of this project can be considered a success in this area.

Next, it was necessary to ask the judgement of certain people as explained in section 3.1, this is shown in the table below, along with the relevant weightings.

Question	**3rd Party**	**Fan**	**Scorer**	**Weighting**
Aesthetics of input system	5/10	-----	7/10	15%
Aesthetics of website	6/10	8/10	-----	20%
Ease of use - input system	7/10	-----	9/10	25%
Ease of use – website	7/10	8/10	-----	20%
Speed of systems	9/10	9/10	9/10	5%
Ease of navigation	7/10	7/10	5/10	15%
Total	41/60	32/40	30/40	100%
Weighted Total (Q)	66% / 100%	79% / 100%	78%/100%	---
Weighted total (User)	6.6%/10%	31.6%/ 40%	39%/50%	**77.2 %**

This is a good result overall and is representative of how well the system has performed and what it is like to use. It is clear to see that this can be considered a success and the users will probably not mind using the system again. The lowest score was that of an un-interested 3rd party and this may be due to the fact that this person does not know the basics of baseball and so shall be biased in a negative way. The high scores of the fan and the scorer means that the system will be accepted when it goes live and shall not need too much extra development in terms of the non-functional aspects of the system.

Over these three quantitative tests, it is consistent that the result is above 75% in terms of success, which means that this project has been completed well, although there is room for improvement (see section 3.8). After looking at all these results, it is fair to say that the overall success factor as a percentage would fall somewhere between 75 and 80 per cent. Success of the project in terms of the impression that it gives to the developer has been discussed in sections 3.2, 3.3, 3.5, 3.6 and 3.7.

3.5 Success of product

The product that has been developed for this project has been so quite well. The exact functionality that has been achieved in accordance with the needs of the users and other quantifiable success factors have been discussed in other chapters of this report. These results shows that the product developed is of a good standard and meets a lot of the wants if the users. The simplicity of all of the products in terms of what the user is presented with is very good and means that training is minimised and the products are usable with not a lot of experience. The simplicity of the product is the most beneficial feature to BSUK and it will prove very beneficial to the company in the future. The complexity of the code could prove problematic in the future if the system needs to be upgraded or adjusted. This could have been avoided with more careful planning and design and the encapsulation of more functions to stop long, complicated procedures with multiple tasks. Although these procedures are complex, they do the correct function and the product does work. It is more important that the product works and fulfils user needs than it is efficient and tidy. The user does not mind about how the system works, as long as it does. Both the website and the input system have the appearance of working well and are pleasing to use and so should be considered a success, regardless of the workings behind these products.

The database is much more successful and was designed and implemented very well, in accordance with literature such as Codd's relational model [17]. Access to the database has been restricted as stated in the requirements and the integrity of the data stored here has been upheld. Although the user does not "see" the database, it forms the centre of the Information System and can hugely affect the rest of the system if an error occurs. Because the database is structured in a very efficient and logical manner, it means that the retrieval time is significantly reduced and the queries on the database can be completed quickly.

3.6 Success of project

The development of the project and the way that it has been developed has been completed with as much efficiency as possible. All deadlines were met and the project completed within the timescale. However, during the development of this project, there needed to be a review

of the schedule a number of times. This was due to the fact that the original schedule was too ambitious and the development of certain aspects of the system taking longer than was initially thought. There was a lot of time taken near the beginning of the process learning Python and wxPython that extended the completion of the input program. There was also a miscalculation with the amount of time that would be needed to research the topic and reflect on these findings.

The methodologies used in this project are consistent with those initially set out and there has been constant evidence of object-oriented methods across the design of the system and by basing the design on this, it was easier to convert this into an actual program. It was a good idea to use this method as it meant that the skills learnt previously can be put into action and there was a clear understanding of where the project was heading.

Another aspect of this project was the structure of it. Rather than having the traditional waterfall method, it was tweaked a bit to come up with the current structure. This was successful in the fact that it still managed to cover all the aspects of the development and allowed the needs and requirements to be discovered along the way.

The integration of the systems was done in such a way that used previous knowledge and this was a good decision as there was no need to do extra background reading on middleware or similar technologies, this speeded up the process and meant that more time and effort could be spend on other aspects of the system.

The success of the project in terms of personal development and lessons learnt are discussed in appendix A.

3.7 Comparison with existing solution / alternatives

The system that has been developed is a huge improvement of the system that is being used at present. The elimination of effort is one good thing to come out of the new system; there is no need to record the scores twice as was previously the case. The new system has the same capabilities of the previous system, in that it can still record the scores but there are additional features that add to the value of the product. These features do not take anything away that the old system used to be able to do, but merely expands upon the functionality, keeping the main features that were useful in the previous system such as the facility to record score and input details about the game. The additional functionality includes the automation of such features and the ability to check input to reduce errors. This is clearly an advantage and cannot be considered a hindrance to the company in any way. Since the system works and does the same job as the previous system with more automation and records more details for

storage in the database, it can be considered a significant improvement and therefore a success in replacing the previous system.

The alternative products that were discussed in section 1.8 are still valid alternatives to this system and it is possible that these alternatives do have some advantages over the product that has been developed. These products do similar functions to the new system and could be used for different parts of the system but not as one integrated system that does the whole job of what has been developed. The fact that the new system does the combined functions of input, storage and output and the consistency of style and functionality throughout means that this system is much preferred to separate systems for each part of the business and struggling to connect these. There is also the advantage of this product needing no or very little monetary outlay to implement as apposed to some of the other systems that could be bought it and used.

This system seems to be on a par with the previously built bespoke systems and does a the same function, but it is much simpler to use and does not require much knowledge of baseball or training in the product as other systems seem to do. There does not seem to be any advantages that other systems have over this new system that would warrant the extra costs involved, as well as not having the support that I would give to the company when installing and maintaining the new system. There is also nothing that these systems can do that the developed system cannot do. This product is therefore a success as a viable alternative to these existing solutions and is more beneficial to BSUK when implemented.

The website that has been developed is not as good as the website currently used on American sites such as mlb.com [14] and CBS [13]. This is probably because these systems have evolved over many years and have a continuous supply of money and resources and user feedback and other evaluation has meant that they have improved significantly over a number of seasons. There is not as much functionality in this system as the American systems and it does not hold as many stats or is not as aesthetically pleasing. There is a huge drawback about the American systems; it is unobtainable without breaking certain laws and therefore this system cannot be used to replace the current one employed by BSUK. The developed website is a good starting point fro the future and this site can be developed further over the seasons to become better and incorporate more and more functionality so that it can rival the American systems. Although it seems that this system has been a failure in competing with the US methods, it has significantly improved the current UK baseball site and now there is much more information available to fans, with a more in-depth look at all games. It has therefore increased the popularity of the website and consequently the support and awareness of baseball in the UK.

3.8 Possible future developments

There are a number of possible developments that could be done in the future. Most of these developments are for aesthetic use and also the things that stood out in the assessment criteria. Firstly the actual layout of the website, the information and queries available could be increased. This could include a more detailed search allowing the return of 2 or more players for comparison and some more detailed stats. This was not completed in this project due to more important things that needed to be done and the complexity of SQL it would require. There could also be the expansion of the website and input system to include pictures and personal details of players. This will improve the enjoyment of the fans and players; however there is the issues that stopped this being implemented in this project, which are memory restrictions stopping the storage of pictures on the database server and the increase in legal responsibilities. This includes making sure that data is kept up to date and correct at all times, this was not a viable thing to do in this project because of personnel restrictions and the absence of a single person that could be responsible fro ensuring the following of the DPA. The input system could be improved and this will happen as I continue to work with the company after this project has finished. The main issues that shall be addressed is the improvement of error detection and the flexibility of the program to include such things as more effective correction methods and free-text input to describe certain plays.
The goal for the future is to make the input program able to record every single little details of the game and an explanation so that if the results of this were to be picked up, the complete game could be re-enacted as a duplicate. This would have been ideal for this project but the restrictions in place, such as personnel, memory and other hardware restrictions meat that this could not happen.

References

[1] Vision document template – RUP http://www.comp.leeds.ac.uk/is21/RUP-VISION-DOC.html [11th November 2004]

[2] JoAnn Hackos, Janice Redish - User and task analysis for interface design (Wiley Computer publishing – 1998)

[3] Ken Barclay, John Savage – Object-oriented design with c++ (Prentice Hall –1997)

[4] Stuart Roberts – Guidelines for Database topics http://www.comp.leeds.ac.uk/tsinfo/projects/database_g.html [20th November 2004]

[5] Principles of database design (Prentice Hall – 1985) Various contributors. Relevant contributors – B.K. Kahn, G.U Hubbard, Y.E. Lien, P.P.S Chen

[6] Philip J Pratt, Joseph J Adamski – Database systems management and design 3rd edition (Boyd & Fraser 1994)

[7] Suzanne W Dietrich – Understanding relational database Query languages (P.Hall 2001)

[8] P. Mott – Internet systems technologies notes (University of Leeds 2004)

[9] J. Thatcher, P. Bohman, M Burks, S. L. Henry, B. Reagan, S Swierenga, M.D. Urban, C.D. Waddell – Constructing accessible web sites (glasshaus – 2002)

[10] K. Mullet, D. Sano – Designing visual interfaces (Prentice Hall – 1995)

[11] D. Lea, D. de Champeaux, P. Faure – Object Oriented System development (Addsion-Wesley – 1993)

[12] P. Jimmack, N. Efford, M Clark, S. Fores – Guidelines for software engineering projects http://www.comp.leeds.ac.uk/tsinfo/projects/software_g.html [December 19 2004]

[13] CBS sportsline online scoreboard www.cbssportsline.com/mlb/scoreboard [07 Dec 2004]

[14] MLB official stats page - http://mlb.mlb.com/NASApp/mlb/mlb/stats/index.jsp [2004]

[15] Jim Sterne – Web Metrics: Proven methods for measuring web site success (Wiley & Sons – 2002)

[16] D. Kahneman, P. Slovic, A. Tversky – Judgement under uncertainty: Heuristics and biases (Cambridge University press – 1982)

[17] E.F. Codd – A relational model for large shared databanks

[18] Fox sports online – http://msn.foxsports.com/mlb [27 January 2005]

[19] Score it website - http://www.scoreit.com/ [29 January 2005]

[20] scorePad PAK online site - http://www.palmgear.com/ and search for "ScorePad" [29 January 2005]

[21] Data protection act online -http://www.hmso.gov.uk/acts/acts1998/19980029.htm [29 January 2005]

[22] Computer misuse act online - http://www.hmso.gov.uk/acts/acts1990/Ukpga_19900018_en_1.htm [15 February 2005]

Appendix A – Personal reflection

Over the time developing this system, there have been many improvements in terms of my personal development. One of the main things that have come out of this project is the improvement of time management that has become apparent. At the start of this project, the schedule was too ambitious and didn't include as much allotted time that was necessary to research the relevant procedures for this project. Another problem that was encountered in terms of time management was the amount of work done in the first few months of the project was not consistent with what was then required fro the remainder. This was due to the complacency that was in place and the idea that "there is plenty of time". This was not the case and resulted in more effort having to be completed in the remaining months. Another reason for this is that there are no deadlines that need to be met and so the urgency was not as great.

If this project was to be started again this could be easily rectified and the benefit of hindsight will mean that more work will be completed in the earlier months and more time is assigned to the preparation of the project.

Another thing to be learnt from this project is the amount of planning needed for such a large development. At the beginning of this development, when designing it was not clear to me, the amount of work needed to learn the relevant techniques to complete this project. This meant that technologies such as PHP and ASP had to be learned quickly, which resulted in the not as efficient coding as it could be.

What seemed to develop well in this project is the evaluation. I seemed to be able to understand this process and by having pre-defined goals, it was easy to judge success and to justify it with quantitative results.

It was surprising that the hardest part of this project was the actual development of the programs, whereas this would be considered one of my strengths before this project began, it is now clear to see that my strength lies more in analysing and evaluating certain situations.

If this project were to be completed again, I would go about it fairly differently, with more time for preparation and making the design as full as possible so that it does not need much effort for development.

Appendix 1

The following is an extraction from MLB.com [14] and has been shortened to include as little irrelevant information as possible

Official Rules:
1.00 Objectives of the Game

1.01
Baseball is a game between two teams of nine players each, under direction of a manager, played on an enclosed field in accordance with these rules, under jurisdiction of one or more umpires.

1.02
The objective of each team is to win by scoring more runs than the opponent.

1.03
The winner of the game shall be that team which shall have scored, in accordance with these rules, the greater number of runs at the conclusion of a regulation game.

1.04
THE PLAYING FIELD. The field shall be laid out according to the instructions below, supplemented by Diagrams No. 1, No. 2 and No. 3. The infield shall be a 90 foot square. The outfield shall be the area between two foul lines formed by extending two sides of the square, as in Diagram 1. The distance from home base to the nearest fence, stand or other obstruction on fair territory shall be 250 feet or more. A distance of 320 feet or more along the foul lines, and 400 feet or more to center field is preferable. The infield shall be graded so that the base lines and home plate are level. The pitcher's plate shall be 10 inches above the level of home plate. The degree of slope from a point 6 inches in front of the pitcher's plate to a point 6 feet toward home plate shall be 1 inch to 1 foot, and such degree of slope shall be uniform. The infield and outfield, including the boundary lines, are fair territory and all other area is foul territory. It is desirable that the line from home base through the pitchers plate to second base shall run East Northeast. It is recommended that the distance from home base to the backstop, and from the base lines to the nearest fence, stand or other obstruction on foul territory shall be 60 feet or more. See Diagram 1. When location of home base is determined, with a steel tape measure 127 feet, 3 3/8 inches in desired direction to establish second base. From home base, measure 90 feet toward first base; from second base, measure 90 feet toward first

DIAGRAM 1 -

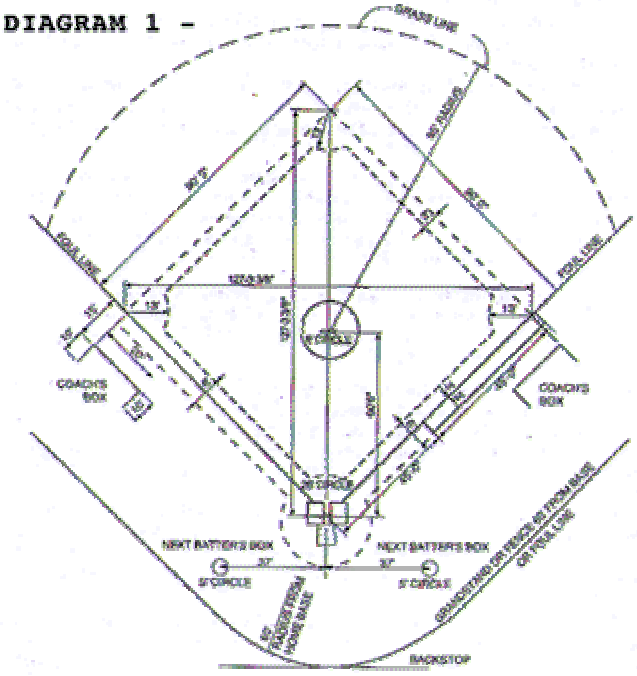
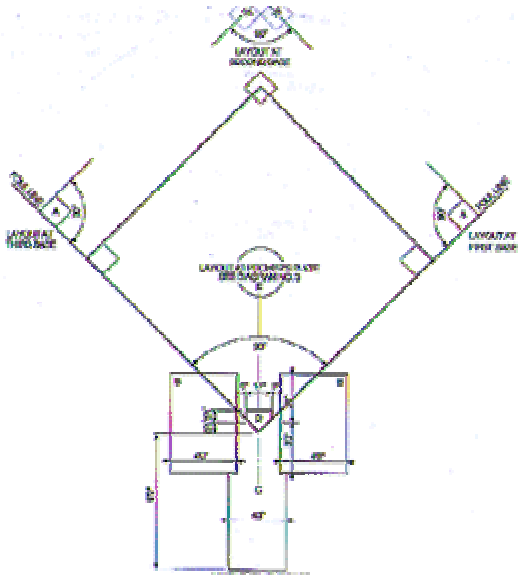
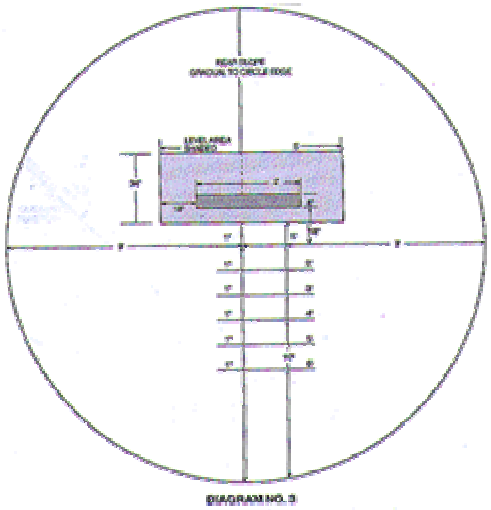


DIAGRAM 2 -



base; the intersection of these lines establishes first base. From home base, measure 90 feet toward third base; from second base, measure 90 feet toward third base; the intersection of these lines establishes third base. The distance between first base and third base is 127 feet, 3 3/8 inches. All measurements from home base shall be taken from the point where the first and third base lines intersect. The catcher's box, the batters' boxes, the coaches' boxes, the three foot first base lines and the next batter's boxes shall be laid out as shown in Diagrams 1 and 2. The foul lines and all other playing lines indicated in the diagrams by solid black lines shall be marked with wet, unslaked lime, chalk or other white material. The grass lines and dimensions shown on the diagrams are those used in many fields, but they are not mandatory and each club shall determine the size and shape of the grassed and bare areas of its playing field. NOTE (a) Any Playing Field constructed by a professional club after June 1, 1958, shall provide a minimum distance of 325 feet from home base to the nearest fence, stand or other obstruction on the right and left field foul lines, and a minimum distance of 400 feet to the center field fence. (b) No existing playing field shall be remodeled after June 1, 1958, in such manner as to reduce the distance from home base to the foul poles and to the center field fence below the minimum specified in paragraph (a) above.

DIAGRAM 3 -



1.05
Home base shall be marked by a five sided slab of whitened rubber. It shall be a 17 inch square with two of the corners removed so that one edge is 17 inches long, two adjacent sides are 8 1/2 inches and the remaining two sides are 12 inches and set at an angle to make a point. It shall be set in the ground with the point at the intersection of the lines extending from home base to first base and to third base; with the 17 inch edge facing the pitcher's plate, and the two 12 inch edges coinciding with the first and third base lines. The top edges of home base shall be beveled and the base shall be fixed in the ground level with the ground surface. Diagram 2:

1.06
First, second and third bases shall be marked by white canvas bags, securely attached to the ground as indicated in Diagram 2. The first and third base bags shall be entirely within the infield. The second base bag shall be centered on second base. The bags shall be 15 inches square, not less than three nor more than five inches thick, and filled with soft material.

1.07
The pitcher's plate shall be a rectangular slab of whitened rubber, 24 inches by 6 inches. It shall be set in the ground as shown in Diagrams 1 and 2, so that the distance between the pitcher's plate and home base (the rear point of home plate) shall be 60 feet, 6 inches.

1.08
The home club shall furnish players' benches, one each for the home and visiting teams. Such benches shall not be less than twenty five feet from the base lines. They shall be roofed and shall be enclosed at the back and ends.

1.09
The ball shall be a sphere formed by yarn wound around a small core of cork, rubber or similar material, covered with two stripes of white horsehide or cowhide, tightly stitched together. It shall weigh not less than five nor more than 5 1/4 ounces avoirdupois and measure not less than nine nor more than 9 1/4 inches in

circumference.

1.10
(a) The bat shall be a smooth, round stick not more than 2 3/4 inches in diameter at the thickest part and not more than 42 inches in length. The bat shall be one piece of solid wood. NOTE: No laminated or experimental bats shall be used in a professional game (either championship season or exhibition games) until the manufacturer has secured approval from the Rules Committee of his design and methods of manufacture. (b) Cupped Bats. An indentation in the end of the bat up to one inch in depth is permitted and may be no wider than two inches and no less than one inch in diameter. The indentation must be curved with no foreign substance added. (c) The bat handle, for not more than 18 inches from its end, may be covered or treated with any material or substance to improve the grip. Any such material or substance, which extends past the 18 inch limitation, shall cause the bat to be removed from the game. NOTE: If the umpire discovers that the bat does not conform to (c) above until a time during or after which the bat has been used in play, it shall not be grounds for declaring the batter out, or ejected from the game. (d) No colored bat may be used in a professional game unless approved by the Rules Committee.

1.11
(a) (1) All players on a team shall wear uniforms identical in color, trim and style, and all players uniforms shall include minimal six inch numbers on their backs. (2) Any part of an undershirt exposed to view shall be of a uniform solid color for all players on a team. Any player other than the pitcher may have numbers, letters, insignia attached to the sleeve of the undershirt. (3) No player whose uniform does not conform to that of his teammates shall be permitted to participate in a game. (b) A league may provide that (1) each team shall wear a distinctive uniform at all times, or (2) that each team shall have two sets of uniforms, white for home games and a different color for road games. (c) (1) Sleeve lengths may vary for individual players, but the sleeves of each individual player shall be approximately the same length. (2) No player shall wear ragged, frayed or slit sleeves. (d) No player shall attach to his uniform tape or other material of a different color from his uniform. (e) No part of the uniform shall include a pattern that imitates or suggests the shape of a baseball. (f) Glass buttons and polished metal shall not be used on a uniform. (g) No player shall attach anything to the heel or toe of his shoe other than the ordinary shoe plate or toe plate. Shoes with pointed spikes similar to golf or track shoes shall not be worn. (h) No part of the uniform shall include patches or designs relating to commercial advertisements. (i) A league may provide that the uniforms of its member teams include the names of its players on their backs. Any name other than the last name of the player must be approved by the League President. If adopted, all uniforms for a team must have the names of its players.

1.12
The catcher may wear a leather mitt not more than thirty eight inches in circumference, nor more than fifteen and one half inches from top to bottom. Such limits shall include all lacing and any leather band or facing attached to the outer edge of the mitt. The space between the thumb section and the finger section of the mitt shall not exceed six inches at the top of the mitt and four inches at the base of the thumb crotch. The web shall measure not more than seven inches across the top or more than six inches from its top to the base of the thumb crotch. The web may be either a lacing or lacing through leather tunnels, or a center piece of leather which may be an extension of the palm, connected to the mitt with lacing and constructed so that it will not exceed any of the above mentioned measurements.

1.13
The first baseman may wear a leather glove or mitt not more than twelve inches long from top to bottom and not more than eight inches wide across the palm, measured from the base of the thumb crotch to the outer edge of the mitt. The space between the thumb section and the finger section of the mitt shall not exceed four inches at the top of the mitt and three and one half inches at the base of the thumb crotch. The mitt shall be constructed so that this space is permanently fixed and cannot be enlarged, extended, widened, or deepened by the use of any materials or process whatever. The web of the mitt shall measure not more than five inches from its top to the base of the thumb crotch. The web may be either a lacing, lacing through leather tunnels, or a center piece of leather which may be an extension of the palm connected to the mitt with lacing and constructed so that it will not exceed the above mentioned measurements. The webbing shall not be constructed of wound or wrapped lacing or deepened to make a net type of trap. The glove may be of any weight.

1.14
Each fielder, other than the first baseman or catcher, may use or wear a leather glove. The measurements covering size of glove shall be made by measuring front side or ball receiving side of glove. The tool or measuring tape shall be placed to contact the surface or feature of item being measured and follow all contours in the process. The glove shall not measure more than 12" from the tip of any one of the 4 fingers, through the ball pocket to the bottom edge or heel of glove. The glove shall not measure more than 7 3/4"

wide, measured from the inside seam at base of first finger, along base of other fingers, to the outside edge of little finger edge of glove. The space or area between the thumb and first finger, called crotch, may be filled with leather webbing or back stop. The webbing may be constructed of two plies of standard leather to close the crotch area entirely, or it may be constructed of a series of tunnels made of leather, or a series of panels of leather, or of lacing leather thongs. The webbing may not be constructed of wound or wrapped lacing to make a net type of trap. When webbing is made to cover entire crotch area, the webbing can be constructed so as to be flexible. When constructed of a series of sections, they must be joined together. These sections may not be so constructed to allow depression to be developed by curvatures in the section sides. The webbing shall be made to control the size of the crotch opening. The crotch opening shall measure not more than 4 1/2" at the top, not more than 5 3/4" deep, and shall be 3 1/2" wide at its bottom. The opening of crotch shall not be more than 4 1/2" at any point below its top. The webbing shall be secured at each side, and at top and bottom of crotch. The attachment to be made with leather lacing, these connections to be secured. If they stretch or become loose, they shall be adjusted to their proper condition. The glove can be of any weight.

1.15
(a) The pitcher's glove shall be uniform in color, including all stitching, lacing and webbing. The pitcher's glove may not be white or gray.
(b) No pitcher shall attach to his glove any foreign material of a color different from the glove.

1.16
A Professional League shall adopt the following rule pertaining to the use of helmets:
(a) All players shall use some type of protective helmet while at bat. (b) All players in National Association Leagues shall wear a double ear flap helmet while at bat. (c) All players entering the Major Leagues commencing with the 1983 championship season and every succeeding season thereafter must wear a single ear flap helmet (or at the player's option, a double ear flap helmet), except those players who were in the Major League during the 1982 season, and who, as recorded in that season, objected to wearing a single ear flap helmet. (d) All catchers shall wear a catcher's protective helmet, while fielding their position. (e) All bat/ball boys or girls shall wear a protective helmet while performing their duties. If the umpire observes any violation of these rules, he shall direct the violation to be corrected. If the violation is not corrected within a reasonable time, in the umpire's judgment, the umpire shall eject the offender from the game, and disciplinary action, as appropriate, will be recommended.

1.17
Playing equipment including but not limited to the bases, pitcher's plate, baseball, bats, uniforms, catcher's mitts, first baseman's gloves, infielders and outfielders gloves and protective helmets, as detailed in the provisions of this rule, shall not contain any undue commercialization of the product. Designations by the manufacturer on any such equipment must be in good taste as to the size and content of the manufacturer's logo or the brand name of the item. The provisions of this Section 1.17 shall apply to professional leagues only. NOTE: Manufacturers who plan innovative changes in baseball equipment for professional baseball leagues should submit same to the Official Playing Rules Committee prior to production.

Official Rules: 5.00 Putting the ball in play. Live ball.

5.01
At the time set for beginning the game the umpire shall call "Play."

5.02
After the umpire calls "Play" the ball is alive and in play and remains alive and in play until for legal cause, or at the umpire's call of "Time" suspending play, the ball becomes dead. While the ball is dead no player may be put out, no bases may be run and no runs may be scored, except that runners may advance one or more bases as the result of acts which occurred while the ball was alive (such as, but not limited to a balk, an overthrow, interference, or a home run or other fair ball hit out of the playing field). Should a ball come partially apart in a game, it is in play until the play is completed.

5.03
The pitcher shall deliver the pitch to the batter who may elect to strike the ball, or who may not offer at it, as he chooses.

5.04
The offensive team's objective is to have its batter become a runner, and its runners advance.

5.05
The defensive team's objective is to prevent offensive players from becoming runners, and to prevent their advance around the bases.

5.06
When a batter becomes a runner and touches all bases legally he shall score one run for his team. A run legally scored cannot be nullified by subsequent action of the runner, such as but not limited to an effort to return to third base in the belief that he had left the base before a caught fly ball.

5.07
When three offensive players are legally put out, that team takes the field and the opposing team becomes the offensive team.

5.08
If a thrown ball accidentally touches a base coach, or a pitched or thrown ball touches an umpire, the ball is alive and in play. However, if the coach interferes with a thrown ball, the runner is out.

5.09
The ball becomes dead and runners advance one base, or return to their bases, without liability to be put out, when_ (a) A pitched ball touches a batter, or his clothing, while in his legal batting position; runners, if forced, advance; (b) The plate umpire interferes with the catcher's throw; runners may not advance. NOTE: The interference shall be disregarded if the catcher's throw retires the runner. (c) A balk is committed; runners advance; (See Penalty 8.05). (d) A ball is illegally batted; runners return; (e) A foul ball is not caught; runners return. The umpire shall not put the ball in play until all runners have retouched their bases; (f) A fair ball touches a runner or an umpire on fair territory before it touches an infielder including the pitcher, or touches an umpire before it has passed an infielder other than the pitcher; If a fair ball touches an umpire working in the infield after it has bounded past, or over, the pitcher, it is a dead ball. If a batted ball is deflected by a fielder in fair territory and hits a runner or an umpire while still in flight and then caught by an infielder it shall not be a catch, but the ball shall remain in play. If a fair ball goes through, or by, an infielder, and touches a runner immediately back of him, or touches a runner after being deflected by an infielder, the ball is in play and the umpire shall not declare the runner out. In making such decision the umpire must be convinced that the ball passed through, or by, the infielder and that no other infielder had the chance to make a play on the ball; runners advance if forced; (g) A pitched ball lodges in the umpire's or catcher's mask or paraphernalia, and remains out of play, runners advance one base; If a foul tip hits the umpire and is caught by a fielder on the rebound, the ball is "dead" and the batsman cannot be called out. The same shall apply where such foul tip lodges in the umpire's mask or other paraphernalia. If a third strike (not a foul tip) passes the catcher and hits an umpire, the ball is in play. If such ball rebounds and is caught by a fielder before it touches the ground, the batsman is not out on such a catch, but the ball remains in play and the batsman may be retired at first base, or touched with the ball for the out. If a pitched ball lodges in the umpire's or catcher's mask or paraphernalia, and remains out of play, on the third strike or fourth ball, then the batter is entitled to first base and all runners advance one base. If the count on the batter is less than three balls, runners advance one base. (h) Any legal pitch touches a runner trying to score; runners advance.

5.10
The ball becomes dead when an umpire calls "Time." The umpire in chief shall call "Time"_ (a) When in his judgment weather, darkness or similar conditions make immediate further play impossible; (b) When light failure makes it difficult or impossible for the umpires to follow the play; NOTE: A league may adopt its own regulations governing games interrupted by light failure. (c) When an accident incapacitates a player or an umpire; (1) If an accident to a runner is such as to prevent him from proceeding to a base to which he is entitled, as on a home run hit out of the playing field, or an award of one or more bases, a substitute runner shall be permitted to complete the play. (d) When a manager requests "Time" for a substitution, or for a conference with one of his players. (e) When the umpire wishes to examine the ball, to consult with either manager, or for any similar cause. (f) When a fielder, after catching a fly ball, falls into a bench or stand, or falls across ropes into a crowd when spectators are on the field. As pertains to runners, the provisions of 7.04 (c) shall prevail. If a fielder after making a catch steps into a bench, but does not fall, the ball is in play and runners may advance at their own peril. (g) When an umpire orders a player or any other person removed from the playing field. (h) Except in the cases stated in paragraphs (b) and (c) (1) of this rule, no umpire shall call "Time" while a play is in progress.

5.11
After the ball is dead, play shall be resumed when the pitcher takes his place on the pitcher's plate with

a new ball or the same ball in his possession and the plate umpire calls "Play." The plate umpire shall call "Play" as soon as the pitcher takes his place on his plate with the ball in his possession.

Official Rules: 6.00 The Batter

6.01
(a) Each player of the offensive team shall bat in the order that his name appears in his team's batting order. (b) The first batter in each inning after the first inning shall be the player whose name follows that of the last player who legally completed his time at bat in the preceding inning.

6.02
(a) The batter shall take his position in the batter's box promptly when it is his time at bat. (b) The batter shall not leave his position in the batter's box after the pitcher comes to Set Position, or starts his windup. PENALTY: If the pitcher pitches, the umpire shall call "Ball" or "Strike," as the case may be. The batter leaves the batter's box at the risk of having a strike delivered and called, unless he requests the umpire to call "Time." The batter is not at liberty to step in and out of the batter's box at will. Once a batter has taken his position in the batter's box, he shall not be permitted to step out of the batter's box in order to use the resin or the pine tar rag, unless there is a delay in the game action or, in the judgment of the umpires, weather conditions warrant an exception. Umpires will not call "Time" at the request of the batter or any member of his team once the pitcher has started his windup or has come to a set position even though the batter claims "dust in his eyes," "steamed glasses," "didn't get the sign" or for any other cause. Umpires may grant a hitter's request for "Time" once he is in the batter's box, but the umpire should eliminate hitters walking out of the batter's box without reason. If umpires are not lenient, batters will understand that they are in the batter's box and they must remain there until the ball is pitched. If pitcher delays once the batter is in his box and the umpire feels that the delay is not justified he may allow the batter to step out of the box momentarily. If after the pitcher starts his windup or comes to a "set position" with a runner on, he does not go through with his pitch because the batter has stepped out of the box, it shall not be called a balk. Both the pitcher and batter have violated a rule and the umpire shall call time and both the batter and pitcher start over from "scratch." (c) If the batter refuses to take his position in the batter's box during his time at bat, the umpire shall order the pitcher to pitch, and shall call "Strike" on each such pitch. The batter may take his proper position after any such pitch, and the regular ball and strike count shall continue, but if he does not take his proper position before three strikes are called, he shall be declared out.

6.03
The batter's legal position shall be with both feet within the batter's box. APPROVED RULING: The lines defining the box are within the batter's box.

6.04
A batter has legally completed his time at bat when he is put out or becomes a runner.

6.05
A batter is out when_ (a) His fair or foul fly ball (other than a foul tip) is legally caught by a fielder; (b) A third strike is legally caught by the catcher; "Legally caught" means in the catcher's glove before the ball touches the ground. It is not legal if the ball lodges in his clothing or paraphernalia; or if it touches the umpire and is caught by the catcher on the rebound. If a foul tip first strikes the catcher's glove and then goes on through and is caught by both hands against his body or protector, before the ball touches the ground, it is a strike, and if third strike, batter is out. If smothered against his body or protector, it is a catch provided the ball struck the catcher's glove or hand first. (c) A third strike is not caught by the catcher when first base is occupied before two are out; (d) He bunts foul on third strike; (e) An Infield Fly is declared; (f) He attempts to hit a third strike and the ball touches him; (g) His fair ball touches him before touching a fielder; (h) After hitting or bunting a fair ball, his bat hits the ball a second time in fair territory. The ball is dead and no runners may advance. If the batter runner drops his bat and the ball rolls against the bat in fair territory and, in the umpire's judgment, there was no intention to interfere with the course of the ball, the ball is alive and in play; If a bat breaks and part of it is in fair territory and is hit by a batted ball or part of it hits a runner or fielder, play shall continue and no interference called. If batted ball hits part of broken bat in foul territory, it is a foul ball. If a whole bat is thrown into fair territory and interferes with a defensive player attempting to make a play, interference shall be called, whether intentional or not. In cases where the batting helmet is accidentally hit with a batted or thrown ball, the ball remains in play the same as if it has not hit the helmet. If a batted ball strikes a batting helmet or any other object foreign to the natural ground while on foul territory, it is a foul ball and the ball is dead. If, in the umpire's judgment, there is intent on the part of a baserunner to interfere with a batted or thrown ball by dropping the helmet or throwing it at the ball, then the runner would be out, the ball dead and runners

would return to last base legally touched. (i) After hitting or bunting a foul ball, he intentionally deflects the course of the ball in any manner while running to first base. The ball is dead and no runners may advance; (j) After a third strike or after he hits a fair ball, he or first base is tagged before he touches first base; (k) In running the last half of the distance from home base to first base, while the ball is being fielded to first base, he runs outside (to the right of) the three foot line, or inside (to the left of) the foul line, and in the umpire's judgment in so doing interferes with the fielder taking the throw at first base; except that he may run outside (to the right of) the three foot line or inside (to the left of) the foul line to avoid a fielder attempting to field a batted ball; (l) An infielder intentionally drops a fair fly ball or line drive, with first, first and second, first and third, or first, second and third base occupied before two are out. The ball is dead and runner or runners shall return to their original base or bases; APPROVED RULING: In this situation, the batter is not out if the infielder permits the ball to drop untouched to the ground, except when the Infield Fly rule applies. (m)A preceding runner shall, in the umpire's judgment, intentionally interfere with a fielder who is attempting to catch a thrown ball or to throw a ball in an attempt to complete any play: The objective of this rule is to penalize the offensive team for deliberate, unwarranted, unsportsmanlike action by the runner in leaving the baseline for the obvious purpose of crashing the pivot man on a double play, rather than trying to reach the base. Obviously this is an umpire's judgment play. (n) With two out, a runner on third base, and two strikes on the batter, the runner attempts to steal home base on a legal pitch and the ball touches the runner in the batter's strike zone. The umpire shall call "Strike Three," the batter is out and the run shall not count; before two are out, the umpire shall call "Strike Three," the ball is dead, and the run counts.

6.06
A batter is out for illegal action when_ (a) He hits a ball with one or both feet on the ground entirely outside the batter's box. If a batter hits a ball fair or foul while out of the batter's box, he shall be called out. Umpires should pay particular attention to the position of the batter's feet if he attempts to hit the ball while he is being intentionally passed. A batter cannot jump or step out of the batter's box and hit the ball. (b) He steps from one batter's box to the other while the pitcher is in position ready to pitch; (c) He interferes with the catcher's fielding or throwing by stepping out of the batter's box or making any other movement that hinders the catcher's play at home base. EXCEPTION: Batter is not out if any runner attempting to advance is put out, or if runner trying to score is called out for batter's interference. If the batter interferes with the catcher, the plate umpire shall call "interference." The batter is out and the ball dead. No player may advance on such interference (offensive interference) and all runners must return to the last base that was, in the judgment of the umpire, legally touched at the time of the interference. If, however, the catcher makes a play and the runner attempting to advance is put out, it is to be assumed there was no actual interference and that runner is out not the batter. Any other runners on the base at the time may advance as the ruling is that there is no actual interference if a runner is retired. In that case play proceeds just as if no violation had been called. If a batter strikes at a ball and misses and swings so hard he carries the bat all the way around and, in the umpire's judgment, unintentionally hits the catcher or the ball in back of him on the backswing before the catcher has securely held the ball, it shall be called a strike only (not interference). The ball will be dead, however, and no runner shall advance on the play. (d) He uses or attempts to use a bat that, in the umpire's judgment, has been altered or tampered with in such a way to improve the distance factor or cause an unusual reaction on the baseball. This includes, bats that are filled, flat surfaced, nailed, hollowed, grooved or covered with a substance such as paraffin, wax, etc. No advancement on the bases will be allowed and any out or outs made during a play shall stand. In addition to being called out, the player shall be ejected from the game and may be subject to additional penalties as determined by his League President.

6.07
BATTING OUT OF TURN. (a) A batter shall be called out, on appeal, when he fails to bat in his proper turn, and another batter completes a time at bat in his place. (1) The proper batter may take his place in the batter's box at any time before the improper batter becomes a runner or is put out, and any balls and strikes shall be counted in the proper batter's time at bat. (b) When an improper batter becomes a runner or is put out, and the defensive team appeals to the umpire before the first pitch to the next batter of either team, or before any play or attempted play, the umpire shall (1) declare the proper batter out; and (2) nullify any advance or score made because of a ball batted by the improper batter or because of the improper batter's advance to first base on a hit, an error, a base on balls, a hit batter or otherwise. NOTE: If a runner advances, while the improper batter is at bat, on a stolen base, balk, wild pitch or passed ball, such advance is legal. (c) When an improper batter becomes a runner or is put out, and a pitch is made to the next batter of either team before an appeal is made, the improper batter thereby becomes the proper batter, and the results of his time at bat become legal. (d) (1) When the proper batter is called out because he has failed to bat in turn, the next batter shall be the batter whose name follows that of the proper batter thus called out; (2) When an improper batter becomes a proper batter because no appeal is made before the next pitch, the next batter shall be the batter whose name follows that of such legalized improper batter. The instant an improper batter's actions are legalized, the batting

order picks up with the name following that of the legalized improper batter. The umpire shall not direct the attention of any person to the presence in the batter's box of an improper batter. This rule is designed to require constant vigilance by the players and managers of both teams. There are two fundamentals to keep in mind: When a player bats out of turn, the proper batter is the player called out. If an improper batter bats and reaches base or is out and no appeal is made before a pitch to the next batter, or before any play or attempted play, that improper batter is considered to have batted in proper turn and establishes the order that is to follow. APPROVED RULING To illustrate various situations arising from batting out of turn, assume a first inning batting order as follows: Abel Baker Charles Daniel Edward Frank George Hooker Irwin. PLAY (1). Baker bats. With the count 2 balls and 1 strike, (a) the offensive team discovers the error or (b) the defensive team appeals. RULING: In either case, Abel replaces Baker, with the count on him 2 balls and 1 strike. PLAY (2). Baker bats and doubles. The defensive team appeals (a) immediately or (b) after a pitch to Charles. RULING: (a) Abel is called out and Baker is the proper batter; (b) Baker stays on second and Charles is the proper batter. PLAY (3). Abel walks. Baker walks. Charles forces Baker. Edward bats in Daniel's turn. While Edward is at bat, Abel scores and Charles goes to second on a wild pitch. Edward grounds out, sending Charles to third. The defensive team appeals (a) immediately or (b) after a pitch to Daniel. RULING: (a) Abel's run counts and Charles is entitled to second base since these advances were not made because of the improper batter batting a ball or advancing to first base. Charles must return to second base because his advance to third resulted from the improper batter batting a ball. Daniel is called out, and Edward is the proper batter; (b) Abel's run counts and Charles stays on third. The proper batter is Frank. PLAY (4). With the bases full and two out. Hooker bats in Frank's turn, and triples, scoring three runs. The defensive team appeals (a) immediately, or (b) after a pitch to George. RULING: (a) Frank is called out and no runs score. George is the proper batter to lead off the second inning; (b) Hooker stays on third and three runs score. Irwin is the proper batter. PLAY (5). After Play (4) (b) above, George continues at bat. (a) Hooker is picked off third base for the third out, or (b) George flies out, and no appeal is made. Who is the proper leadoff batter in the second inning? RULING: (a) Irwin. He became the proper batter as soon as the first pitch to George legalized Hooker's triple; (b) Hooker. When no appeal was made, the first pitch to the leadoff batter of the opposing team legalized George's time at bat. PLAY (6). Daniel walks and Abel comes to bat. Daniel was an improper batter, and if an appeal is made before the first pitch to Abel, Abel is out, Daniel is removed from base, and Baker is the proper batter. There is no appeal, and a pitch is made to Abel. Daniel's walk is now legalized, and Edward thereby becomes the proper batter. Edward can replace Abel at any time before Abel is put out or becomes a runner. He does not do so. Abel flies out, and Baker comes to bat. Abel was an improper batter, and if an appeal is made before the first pitch to Baker, Edward is out, and the proper batter is Frank. There is no appeal, and a pitch is made to Baker. Abel's out is now legalized, and the proper batter is Baker. Baker walks. Charles is the proper batter. Charles flies out. Now Daniel is the proper batter, but he is on second base. Who is the proper batter? RULING: The proper batter is Edward. When the proper batter is on base, he is passed over, and the following batter becomes the proper batter

6.08

The batter becomes a runner and is entitled to first base without liability to be put out (provided he advances to and touches first base) when_ (a) Four "balls" have been called by the umpire; A batter who is entitled to first base because of a base on balls must go to first base and touch the base before other base runners are forced to advance. This applies when bases are full and applies when a substitute runner is put into the game. If, in advancing, the base runner thinks there is a play and he slides past the base before or after touching it he may be put out by the fielder tagging him. If he fails to touch the base to which he is entitled and attempts to advance beyond that base he may be put out by tagging him or the base he missed. (b) He is touched by a pitched ball which he is not attempting to hit unless (1) The ball is in the strike zone when it touches the batter, or (2) The batter makes no attempt to avoid being touched by the ball; If the ball is in the strike zone when it touches the batter, it shall be called a strike, whether or not the batter tries to avoid the ball. If the ball is outside the strike zone when it touches the batter, it shall be called a ball if he makes no attempt to avoid being touched. APPROVED RULING: When the batter is touched by a pitched ball which does not entitle him to first base, the ball is dead and no runner may advance. (c) The catcher or any fielder interferes with him. If a play follows the interference, the manager of the offense may advise the plate umpire that he elects to decline the interference penalty and accept the play. Such election shall be made immediately at the end of the play. However, if the batter reaches first base on a hit, an error, a base on balls, a hit batsman, or otherwise, and all other runners advance at least one base, the play proceeds without reference to the interference. If catcher's interference is called with a play in progress the umpire will allow the play to continue because the manager may elect to take the play. If the batter runner missed first base, or a runner misses his next base, he shall be considered as having reached the base, as stated in Note of Rule 7.04 (d). Examples of plays the manager might elect to take: 1. Runner on third, one out, batter hits fly ball to the outfield on which the runner scores but catcher's interference was called. The offensive manager may elect to take the run and have batter called out or have runner remain at third and batter awarded first base. 2. Runner on second base. Catcher interferes with batter as he bunts ball fairly

sending runner to third base. The manager may rather have runner on third base with an out on the play than have runners on second and first. In situations where the manager wants the "interference" penalty to apply, the following interpretation shall be made of 6.08 (c): If the catcher (or any fielder) interferes with the batter, the batter is awarded first base. If, on such interference a runner is trying to score by a steal or squeeze from third base, the ball is dead and the runner on third scores and batter is awarded first base. If the catcher interferes with the batter with no runners trying to score from third on a squeeze or steal, then the ball is dead, batter is awarded first base and runners who are forced to advance, do advance. Runners not attempting to steal or not forced to advance remain on the base they occupied at the time of the interference. If the catcher interferes with the batter before the pitcher delivers the ball, it shall not be considered interference on the batter under Rule 6.08 (c). In such cases, the umpire shall call "Time" and the pitcher and batter start over from "scratch." (d) A fair ball touches an umpire or a runner on fair territory before touching a fielder. If a fair ball touches an umpire after having passed a fielder other than the pitcher, or having touched a fielder, including the pitcher, the ball is in play.

6.09
The batter becomes a runner when_ (a) He hits a fair ball; (b) The third strike called by the umpire is not caught, providing (1) first base is unoccupied, or (2) first base is occupied with two out; When a batter becomes a base runner on a third strike not caught by the catcher and starts for the dugout, or his position, and then realizes his situation and attempts then to reach first base, he is not out unless he or first base is tagged before he reaches first base. If, however, he actually reaches the dugout or dugout steps, he may not then attempt to go to first base and shall be out. (c) A fair ball, after having passed a fielder other than the pitcher, or after having been touched by a fielder, including the pitcher, shall touch an umpire or runner on fair territory; (d) A fair ball passes over a fence or into the stands at a distance from home base of 250 feet or more. Such hit entitles the batter to a home run when he shall have touched all bases legally. A fair fly ball that passes out of the playing field at a point less than 250 feet from home base shall entitle the batter to advance to second base only; (e) A fair ball, after touching the ground, bounds into the stands, or passes through, over or under a fence, or through or under a scoreboard, or through or under shrubbery, or vines on the fence, in which case the batter and the runners shall be entitled to advance two bases; (f) Any fair ball which, either before or after touching the ground, passes through or under a fence, or through or under a scoreboard, or through any opening in the fence or scoreboard, or through or under shrubbery, or vines on the fence, or which sticks in a fence or scoreboard, in which case the batter and the runners shall be entitled to two bases; (g) Any bounding fair ball is deflected by the fielder into the stands, or over or under a fence on fair or foul territory, in which case the batter and all runners shall be entitled to advance two bases; (h) Any fair fly ball is deflected by the fielder into the stands, or over the fence into foul territory, in which case the batter shall be entitled to advance to second base; but if deflected into the stands or over the fence in fair territory, the batter shall be entitled to a home run. However, should such a fair fly be deflected at a point less than 250 feet from home plate, the batter shall be entitled to two bases only.

6.10
Any League may elect to use the Designated Hitter Rule. (a) In the event of inter league competition between clubs of Leagues using the Designated Hitter Rule and clubs of Leagues not using the Designated Hitter Rule, the rule will be used as follows: 1. In World Series or exhibition games, the rule will be used or not used as is the practice of the home team. 2. In All Star games, the rule will only be used if both teams and both Leagues so agree. (b) The Rule provides as follows: A hitter may be designated to bat for the starting pitcher and all subsequent pitchers in any game without otherwise affecting the status of the pitcher(s) in the game. A Designated Hitter for the pitcher must be selected prior to the game and must be included in the lineup cards presented to the Umpire in Chief. The designated hitter named in the starting lineup must come to bat at least one time, unless the opposing club changes pitchers. It is not mandatory that a club designate a hitter for the pitcher, but failure to do so prior to the game precludes the use of a Designated Hitter for that game. Pinch hitters for a Designated Hitter may be used. Any substitute hitter for a Designated Hitter becomes the Designated Hitter. A replaced Designated Hitter shall not re enter the game in any capacity. The Designated Hitter may be used defensively, continuing to bat in the same position in the batting order, but the pitcher must then bat in the place of the substituted defensive player, unless more than one substitution is made, and the manager then must designate their spots in the batting order. A runner may be substituted for the Designated Hitter and the runner assumes the role of Designated Hitter. A Designated Hitter may not pinch run. A Designated Hitter is "locked" into the batting order. No multiple substitutions may be made that will alter the batting rotation of the Designated Hitter. Once the game pitcher is switched from the mound to a defensive position this move shall terminate the Designated Hitter role for the remainder of the game. Once a pinch hitter bats for any player in the batting order and then enters the game to pitch, this move shall terminate the Designated Hitter role for the remainder of the game. Once the game pitcher bats for the Designated Hitter this move shall terminate the Designated Hitter role for the remainder of the game. (The game pitcher may only pinch hit for the Designated Hitter). Once a Designated Hitter assumes a defensive position this move shall terminate the Designated Hitter role for

the remainder of the game. A substitute for the Designated Hitter need not be announced until it is the Designated Hitter's turn to bat.

Official Rules: 7.00 The Runner

7.01
A runner acquires the right to an unoccupied base when he touches it before he is out. He is then entitled to it until he is put out, or forced to vacate it for another runner legally entitled to that base. If a runner legally acquires title to a base, and the pitcher assumes his pitching position, the runner may not return to a previously occupied base.

7.02
In advancing, a runner shall touch first, second, third and home base in order. If forced to return, he shall retouch all bases in reverse order, unless the ball is dead under any provision of Rule 5.09. In such cases, the runner may go directly to his original base.

7.03
Two runners may not occupy a base, but if, while the ball is alive, two runners are touching a base, the following runner shall be out when tagged. The preceding runner is entitled to the base.

7.04
Each runner, other than the batter, may without liability to be put out, advance one base when_ (a) There is a balk; (b) The batter's advance without liability to be put out forces the runner to vacate his base, or when the batter hits a fair ball that touches another runner or the umpire before such ball has been touched by, or has passed a fielder, if the runner is forced to advance; A runner forced to advance without liability to be put out may advance past the base to which he is entitled only at his peril. If such a runner, forced to advance, is put out for the third out before a preceding runner, also forced to advance, touches home plate, the run shall score. Play. Two out, bases full, batter walks but runner from second is overzealous and runs past third base toward home and is tagged out on a throw by the catcher. Even though two are out, the run would score on the theory that the run was forced home by the base on balls and that all the runners needed to do was proceed and touch the next base. (c) A fielder, after catching a fly ball, falls into a bench or stand, or falls across ropes into a crowd when spectators are on the field; A fielder or catcher may reach or step into, or go into the dugout with one or both feet to make a catch, and if he holds the ball, the catch shall be allowed. Ball is in play. If the fielder or catcher, after having made a legal catch, should fall into a stand or among spectators or into the dugout after making a legal catch, or fall while in the dugout after making a legal catch, the ball is dead and runners advance one base without liability to be put out. (d) While he is attempting to steal a base, the batter is interfered with by the catcher or any other fielder. NOTE: When a runner is entitled to a base without liability to be put out, while the ball is in play, or under any rule in which the ball is in play after the runner reaches the base to which he is entitled, and the runner fails to touch the base to which he is entitled before attempting to advance to the next base, the runner shall forfeit his exemption from liability to be put out, and he may be put out by tagging the base or by tagging the runner before he returns to the missed base.

7.05
Each runner including the batter runner may, without liability to be put out, advance_ (a) To home base, scoring a run, if a fair ball goes out of the playing field in flight and he touched all bases legally; or if a fair ball which, in the umpire's judgment, would have gone out of the playing field in flight, is deflected by the act of a fielder in throwing his glove, cap, or any article of his apparel; (b) Three bases, if a fielder deliberately touches a fair ball with his cap, mask or any part of his uniform detached from its proper place on his person. The ball is in play and the batter may advance to home base at his peril; (c) Three bases, if a fielder deliberately throws his glove at and touches a fair ball. The ball is in play and the batter may advance to home base at his peril. (d) Two bases, if a fielder deliberately touches a thrown ball with his cap, mask or any part of his uniform detached from its proper place on his person. The ball is in play; (e) Two bases, if a fielder deliberately throws his glove at and touches a thrown ball. The ball is in play; In applying (b c d e) the umpire must rule that the thrown glove or detached cap or mask has touched the ball. There is no penalty if the ball is not touched. Under (c e) this penalty shall not be invoked against a fielder whose glove is carried off his hand by the force of a batted or thrown ball, or when his glove flies off his hand as he makes an obvious effort to make a legitimate catch. (f) Two bases, if a fair ball bounces or is deflected into the stands outside the first or third base foul lines; or if it goes through or under a field fence, or through or under a scoreboard, or through or under shrubbery or vines on the fence; or if it sticks in such fence, scoreboard, shrubbery or vines; (g) Two bases when, with no spectators on the playing field, a thrown ball goes into the stands, or into a bench (whether or

not the ball rebounds into the field), or over or under or through a field fence, or on a slanting part of the screen above the backstop, or remains in the meshes of a wire screen protecting spectators. The ball is dead. When such wild throw is the first play by an infielder, the umpire, in awarding such bases, shall be governed by the position of the runners at the time the ball was pitched; in all other cases the umpire shall be governed by the position of the runners at the time the wild throw was made; APPROVED RULING: If all runners, including the batter runner, have advanced at least one base when an infielder makes a wild throw on the first play after the pitch, the award shall be governed by the position of the runners when the wild throw was made. In certain circumstances it is impossible to award a runner two bases. Example: Runner on first. Batter hits fly to short right. Runner holds up between first and second and batter comes around first and pulls up behind him. Ball falls safely. Outfielder, in throwing to first, throws ball into stand. APPROVED RULING: Since no runner, when the ball is dead, may advance beyond the base to which he is entitled, the runner originally on first base goes to third base and the batter is held at second base. The term "when the wild throw was made" means when the throw actually left the player's hand and not when the thrown ball hit the ground, passes a receiving fielder or goes out of play into the stands. The position of the batter runner at the time the wild throw left the thrower's hand is the key in deciding the award of bases. If the batter runner has not reached first base, the award is two bases at the time the pitch was made for all runners. The decision as to whether the batter runner has reached first base before the throw is a judgment call. If an unusual play arises where a first throw by an infielder goes into stands or dugout but the batter did not become a runner (such as catcher throwing ball into stands in attempt to get runner from third trying to score on passed ball or wild pitch) award of two bases shall be from the position of the runners at the time of the throw. (For the purpose of Rule 7.05 (g) a catcher is considered an infielder.) PLAY. Runner on first base, batter hits a ball to the shortstop, who throws to second base too late to get runner at second, and second baseman throws toward first base after batter has crossed first base. Ruling Runner at second scores. (On this play, only if batter runner is past first base when throw is made is he awarded third base.) (h) One base, if a ball, pitched to the batter, or thrown by the pitcher from his position on the pitcher's plate to a base to catch a runner, goes into a stand or a bench, or over or through a field fence or backstop. The ball is dead; APPROVED RULING: When a wild pitch or passed ball goes through or by the catcher, or deflects off the catcher, and goes directly into the dugout, stands, above the break, or any area where the ball is dead, the awarding of bases shall be one base. One base shall also be awarded if the pitcher while in contact with the rubber, throws to a base, and the throw goes directly into the stands or into any area where the ball is dead. If, however, the pitched or thrown ball goes through or by the catcher or through the fielder, and remains on the playing field, and is subsequently kicked or deflected into the dugout, stands or other area where the ball is dead, the awarding of bases shall be two bases from position of runners at the time of the pitch or throw. (i) One base, if the batter becomes a runner on Ball Four or Strike Three, when the pitch passes the catcher and lodges in the umpire's mask or paraphernalia. If the batter becomes a runner on a wild pitch which entitles the runners to advance one base, the batter runner shall be entitled to first base only. The fact a runner is awarded a base or bases without liability to be put out does not relieve him of the responsibility to touch the base he is awarded and all intervening bases. For example: batter hits a ground ball which an infielder throws into the stands but the batter runner missed first base. He may be called out on appeal for missing first base after the ball is put in play even though he was "awarded" second base. If a runner is forced to return to a base after a catch, he must retouch his original base even though, because of some ground rule or other rule, he is awarded additional bases. He may retouch while the ball is dead and the award is then made from his original base.

7.06

When obstruction occurs, the umpire shall call or signal "Obstruction." (a) If a play is being made on the obstructed runner, or if the batter runner is obstructed before he touches first base, the ball is dead and all runners shall advance, without liability to be put out, to the bases they would have reached, in the umpire's judgment, if there had been no obstruction. The obstructed runner shall be awarded at least one base beyond the base he had last legally touched before the obstruction. Any preceding runners, forced to advance by the award of bases as the penalty for obstruction, shall advance without liability to be put out. When a play is being made on an obstructed runner, the umpire shall signal obstruction in the same manner that he calls "Time," with both hands overhead. The ball is immediately dead when this signal is given; however, should a thrown ball be in flight before the obstruction is called by the umpire, the runners are to be awarded such bases on wild throws as they would have been awarded had not obstruction occurred. On a play where a runner was trapped between second and third and obstructed by the third baseman going into third base while the throw is in flight from the shortstop, if such throw goes into the dugout the obstructed runner is to be awarded home base. Any other runners on base in this situation would also be awarded two bases from the base they last legally touched before obstruction was called. (b) If no play is being made on the obstructed runner, the play shall proceed until no further action is possible. The umpire shall then call "Time" and impose such penalties, if any, as in his judgment will nullify the act of obstruction. Under 7.06 (b) when the ball is not dead on obstruction and an obstructed runner advances beyond the base which, in the umpire's judgment, he

would have been awarded because of being obstructed, he does so at his own peril and may be tagged out. This is a judgment call. NOTE: The catcher, without the ball in his possession, has no right to block the pathway of the runner attempting to score. The base line belongs to the runner and the catcher should be there only when he is fielding a ball or when he already has the ball in his hand.

7.07
If, with a runner on third base and trying to score by means of a squeeze play or a steal, the catcher or any other fielder steps on, or in front of home base without possession of the ball, or touches the batter or his bat, the pitcher shall be charged with a balk, the batter shall be awarded first base on the interference and the ball is dead.

7.08
Any runner is out when_ (a) (1) He runs more than three feet away from a direct line between bases to avoid being tagged unless his action is to avoid interference with a fielder fielding a batted ball; or (2) after touching first base, he leaves the baseline, obviously abandoning his effort to touch the next base; Any runner after reaching first base who leaves the baseline heading for his dugout or his position believing that there is no further play, may be declared out if the umpire judges the act of the runner to be considered abandoning his efforts to run the bases. Even though an out is called, the ball remains in play in regard to any other runner. This rule also covers the following and similar plays: Less than two out, score tied last of ninth inning, runner on first, batter hits a ball out of park for winning run, the runner on first passes second and thinking the home run automatically wins the game, cuts across diamond toward his bench as batter runner circles bases. In this case, the base runner would be called out "for abandoning his effort to touch the next base" and batter runner permitted to continue around bases to make his home run valid. If there are two out, home run would not count (see Rule 7.12). This is not an appeal play. PLAY. Runner believing he is called out on a tag at first or third base starts for the dugout and progresses a reasonable distance still indicating by his actions that he is out, shall be declared out for abandoning the bases. In the above two plays the runners are considered actually abandoning their base paths and are treated differently than the batter who struck out as described. APPROVED RULING OF 7.08 (a). APPROVED RULING: When a batter becomes a runner on third strike not caught, and starts for his bench or position, he may advance to first base at any time before he enters the bench. To put him out, the defense must tag him or first base before he touches first base. (b) He intentionally interferes with a thrown ball; or hinders a fielder attempting to make a play on a batted ball; A runner who is adjudged to have hindered a fielder who is attempting to make a play on a batted ball is out whether it was intentional or not. If, however, the runner has contact with a legally occupied base when he hinders the fielder, he shall not be called out unless, in the umpire's judgment, such hindrance, whether it occurs on fair or foul territory, is intentional. If the umpire declares the hindrance intentional, the following penalty shall apply: With less than two out, the umpire shall declare both the runner and batter out. With two out, the umpire shall declare the batter out. If, in a run down between third base and home plate, the succeeding runner has advanced and is standing on third base when the runner in a run down is called out for offensive interference, the umpire shall send the runner standing on third base back to second base. This same principle applies if there is a run down between second and third base and succeeding runner has reached second (the reasoning is that no runner shall advance on an interference play and a runner is considered to occupy a base until he legally has reached the next succeeding base). (c) He is tagged, when the ball is alive, while off his base. EXCEPTION: A batter runner cannot be tagged out after overrunning or oversliding first base if he returns immediately to the base; APPROVED RULING: (1) If the impact of a runner breaks a base loose from its position, no play can be made on that runner at that base if he had reached the base safely. APPROVED RULING: (2) If a base is dislodged from its position during a play, any following runner on the same play shall be considered as touching or occupying the base if, in the umpire's judgment, he touches or occupies the point marked by the dislodged bag. (d) He fails to retouch his base after a fair or foul ball is legally caught before he, or his base, is tagged by a fielder. He shall not be called out for failure to retouch his base after the first following pitch, or any play or attempted play. This is an appeal play; Runners need not "tag up" on a foul tip. They may steal on a foul tip. If a so called tip is not caught, it becomes an ordinary foul. Runners then return to their bases. (e) He fails to reach the next base before a fielder tags him or the base, after he has been forced to advance by reason of the batter becoming a runner. However, if a following runner is put out on a force play, the force is removed and the runner must be tagged to be put out. The force is removed as soon as the runner touches the base to which he is forced to advance, and if he overslides or overruns the base, the runner must be tagged to be put out. However, if the forced runner, after touching the next base, retreats for any reason towards the base he had last occupied, the force play is reinstated, and he can again be put out if the defense tags the base to which he is forced; PLAY. Runner on first and three balls on batter: Runner steals on the next pitch, which is fourth ball, but after having touched second he overslides or overruns that base. Catcher's throw catches him before he can return. Ruling is that runner is out. (Force out is removed.) Oversliding and overrunning situations arise at bases other than first base. For instance, before two are out, and runners on first and second, or first, second and third, the ball is hit to an infielder who tries for the

double play. The runner on first beats the throw to second base but overslides the base. The relay is made to first base and the batter runner is out. The first baseman, seeing the runner at second base off the bag, makes the return throw to second and the runner is tagged off the base. Meanwhile runners have crossed the plate. The question is: Is this a force play? Was the force removed when the batter runner was out at first base? Do the runs that crossed the plate during this play and before the third out was made when the runner was tagged at second, count? Answer: The runs score. It is not a force play. It is a tag play. (f) He is touched by a fair ball in fair territory before the ball has touched or passed an infielder. The ball is dead and no runner may score, nor runners advance, except runners forced to advance. EXCEPTION: If a runner is touching his base when touched by an Infield Fly, he is not out, although the batter is out; If two runners are touched by the same fair ball, only the first one is out because the ball is instantly dead. If runner is touched by an Infield Fly when he is not touching his base, both runner and batter are out. (g) He attempts to score on a play in which the batter interferes with the play at home base before two are out. With two out, the interference puts the batter out and no score counts; (h) He passes a preceding runner before such runner is out; (i) After he has acquired legal possession of a base, he runs the bases in reverse order for the purpose of confusing the defense or making a travesty of the game. The umpire shall immediately call "Time" and declare the runner out; If a runner touches an unoccupied base and then thinks the ball was caught or is decoyed into returning to the base he last touched, he may be put out running back to that base, but if he reaches the previously occupied base safely he cannot be put out while in contact with that base. (j) He fails to return at once to first base after overrunning or oversliding that base. If he attempts to run to second he is out when tagged. If, after overrunning or oversliding first base he starts toward the dugout, or toward his position, and fails to return to first base at once, he is out, on appeal, when he or the base is tagged; Runner who touches first base in overrunning and is declared safe by the umpire has, within the intent of Rule 4.09 (a) "reached first base" and any run which scores on such a play counts, even though the runner subsequently becomes the third out for failure to return "at once," as covered in Rule 7.08 (j). (k) In running or sliding for home base, he fails to touch home base and makes no attempt to return to the base, when a fielder holds the ball in his hand, while touching home base, and appeals to the umpire for the decision. This rule applies only where runner is on his way to the bench and the catcher would be required to chase him. It does not apply to the ordinary play where the runner misses the plate and then immediately makes an effort to touch the plate before being tagged. In that case, runner must be tagged.

7.09
It is interference by a batter or a runner when: (a) After a third strike he hinders the catcher in his attempt to field the ball; (b) After hitting or bunting a fair ball, his bat hits the ball a second time in fair territory. The ball is dead and no runners may advance. If the batter runner drops his bat and the ball rolls against the bat in fair territory and, in the umpire's judgment, there was no intention to interfere with the course of the ball, the ball is alive and in play; (c) He intentionally deflects the course of a foul ball in any manner; (d) Before two are out and a runner on third base, the batter hinders a fielder in making a play at home base; the runner is out; (e) Any member or members of the offensive team stand or gather around any base to which a runner is advancing, to confuse, hinder or add to the difficulty of the fielders. Such runner shall be declared out for the interference of his teammate or teammates; (f) Any batter or runner who has just been put out hinders or impedes any following play being made on a runner. Such runner shall be declared out for the interference of his teammate; If the batter or a runner continues to advance after he has been put out, he shall not by that act alone be considered as confusing, hindering or impeding the fielders. (g) If, in the judgment of the umpire, a base runner willfully and deliberately interferes with a batted ball or a fielder in the act of fielding a batted ball with the obvious intent to break up a double play, the ball is dead. The umpire shall call the runner out for interference and also call out the batter runner because of the action of his teammate. In no event may bases be run or runs scored because of such action by a runner. (h) If, in the judgment of the umpire, a batter runner willfully and deliberately interferes with a batted ball or a fielder in the act of fielding a batted ball, with the obvious intent to break up a double play, the ball is dead; the umpire shall call the batter runner out for interference and shall also call out the runner who had advanced closest to the home plate regardless where the double play might have been possible. In no event shall bases be run because of such interference. (i) In the judgment of the umpire, the base coach at third base, or first base, by touching or holding the runner, physically assists him in returning to or leaving third base or first base. (j) With a runner on third base, the base coach leaves his box and acts in any manner to draw a throw by a fielder; (k) In running the last half of the distance from home base to first base while the ball is being fielded to first base, he runs outside (to the right of) the three foot line, or inside (to the left of) the foul line and, in the umpire's judgment, interferes with the fielder taking the throw at first base, or attempting to field a batted ball; The lines marking the three foot lane are a part of that "lane" but the interpretation to be made is that a runner is required to have both feet within the three foot "lane" or on the lines marking the "lane." (l) He fails to avoid a fielder who is attempting to field a batted ball, or intentionally interferes with a thrown ball, provided that if two or more fielders attempt to field a batted ball, and the runner comes in contact with one or more of them, the umpire shall determine which fielder is entitled to the benefit of

this rule, and shall not declare the runner out for coming in contact with a fielder other than the one the umpire determines to be entitled to field such a ball; When a catcher and batter runner going to first base have contact when the catcher is fielding the ball, there is generally no violation and nothing should be called. "Obstruction" by a fielder attempting to field a ball should be called only in very flagrant and violent cases because the rules give him the right of way, but of course such "right of way" is not a license to, for example, intentionally trip a runner even though fielding the ball. If the catcher is fielding the ball and the first baseman or pitcher obstructs a runner going to first base "obstruction" shall be called and the base runner awarded first base. (m) A fair ball touches him on fair territory before touching a fielder. If a fair ball goes through, or by, an infielder, and touches a runner immediately back of him, or touches the runner after having been deflected by a fielder, the umpire shall not declare the runner out for being touched by a batted ball. In making such decision the umpire must be convinced that the ball passed through, or by, the fielder, and that no other infielder had the chance to make a play on the ball. If, in the judgment of the umpire, the runner deliberately and intentionally kicks such a batted ball on which the infielder has missed a play, then the runner shall be called out for interference. PENALTY FOR INTERFERENCE: The runner is out and the ball is dead.

7.10
Any runner shall be called out, on appeal, when_ (a) After a fly ball is caught, he fails to retouch his original base before he or his original base is tagged; "Retouch," in this rule, means to tag up and start from a contact with the base after the ball is caught. A runner is not permitted to take a flying start from a position in back of his base. (b) With the ball in play, while advancing or returning to a base, he fails to touch each base in order before he, or a missed base, is tagged. APPROVED RULING: (1) No runner may return to touch a missed base after a following runner has scored. (2) When the ball is dead, no runner may return to touch a missed base or one he has left after he has advanced to and touched a base beyond the missed base. PLAY. (a) Batter hits ball out of park or ground rule double and misses first base (ball is dead)_he may return to first base to correct his mistake before he touches second but if he touches second he may not return to first and if defensive team appeals he is declared out at first. PLAY. (b) Batter hits ball to shortstop who throws wild into stand (ball is dead)_batter runner misses first base but is awarded second base on the overthrow. Even though the umpire has awarded the runner second base on the overthrow, the runner must touch first base before he proceeds to second base. These are appeal plays. (c) He overruns or overslides first base and fails to return to the base immediately, and he or the base is tagged; (d) He fails to touch home base and makes no attempt to return to that base, and home base is tagged. Any appeal under this rule must be made before the next pitch, or any play or attempted play. If the violation occurs during a play which ends a half inning, the appeal must be made before the defensive team leaves the field. An appeal is not to be interpreted as a play or an attempted play. Successive appeals may not be made on a runner at the same base. If the defensive team on its first appeal errs, a request for a second appeal on the same runner at the same base shall not be allowed by the umpire. (Intended meaning of the word "err" is that the defensive team in making an appeal threw the ball out of play. For example, if the pitcher threw to first base to appeal and threw the ball into the stands, no second appeal would be allowed.) Appeal plays may require an umpire to recognize an apparent "fourth out." If the third out is made during a play in which an appeal play is sustained on another runner, the appeal play decision takes precedence in determining the out. If there is more than one appeal during a play that ends a half inning, the defense may elect to take the out that gives it the advantage. For the purpose of this rule, the defensive team has "left the field" when the pitcher and all infielders have left fair territory on their way to the bench or clubhouse. If two runners arrive at home base about the same time and the first runner misses home plate but a second runner legally touches the plate, the runner is tagged out on his attempt to come back and touch the base or is called out, on appeal, then he shall be considered as having been put out before the second runner scored and being the third out. Second runner's run shall not count, as provided in Rule 7.12. If a pitcher balks when making an appeal, such act shall be a play. An appeal should be clearly intended as an appeal, either by a verbal request by the player or an act that unmistakably indicates an appeal to the umpire. A player, inadvertently stepping on the base with a ball in his hand, would not constitute an appeal. Time is not out when an appeal is being made.

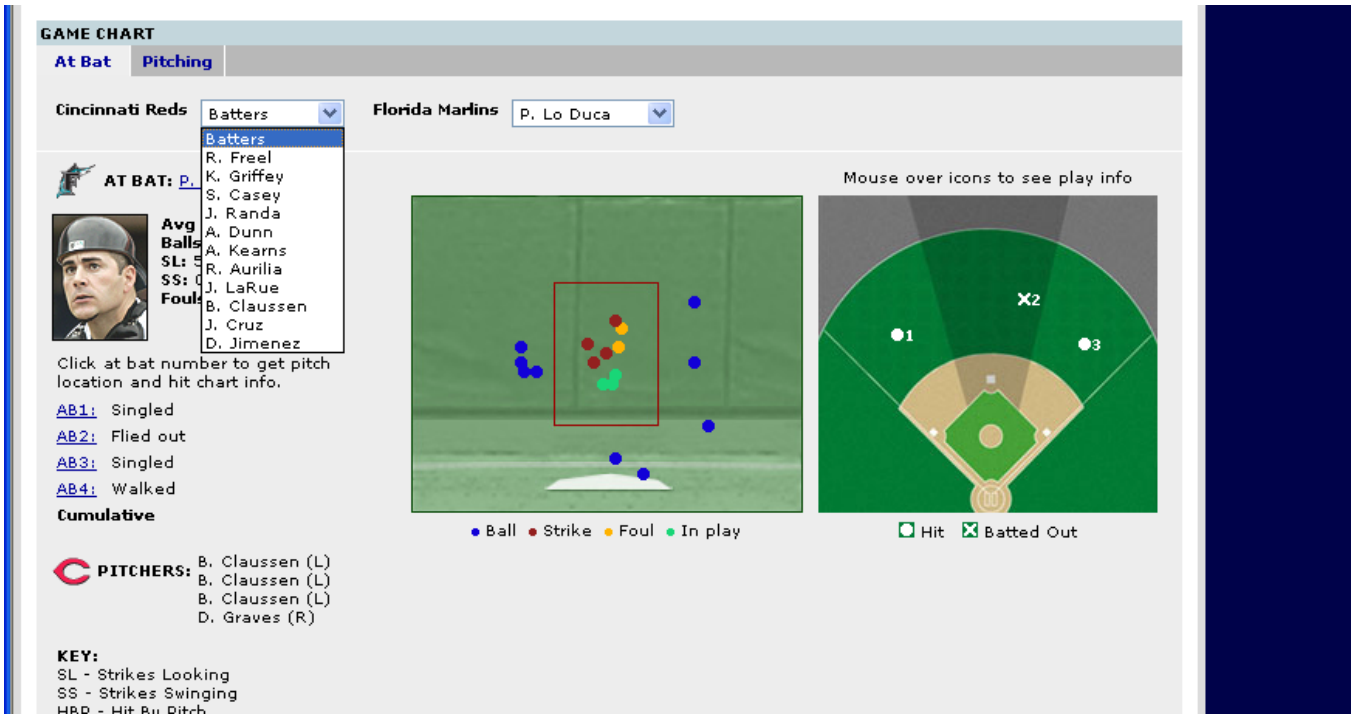
7.11
The players, coaches or any member of an offensive team shall vacate any space (including both dugouts) needed by a fielder who is attempting to field a batted or thrown ball. PENALTY: Interference shall be called and the batter or runner on whom the play is being made shall be declared out.

7.12
Unless two are out, the status of a following runner is not affected by a preceding runner's failure to touch or retouch a base. If, upon appeal, the preceding runner is the third out, no runners following him shall score. If such third out is the result of a force play, neither preceding nor following runners shall score.

Appendix 3 – Box score

SCOREBOARD												
Final	1	2	3	4	5	6	7	8	9	**R**	**H**	**E**
Cincinnati «	1	0	1	0	0	0	0	0	0	**2**	**9**	**0**
Florida	0	0	0	0	0	0	0	0	1	**1**	**4**	**1**

Appendix 5


GAME CHART
At Bat
Pitching
Cincinnati Reds
Batters
Florida Marlins
P. Lo Duca
Batters
R. Freel
K. Griffey
S. Casey
J. Randa
A. Dunn
A. Kearns
R. Aurilia
J. LaRue
B. Claussen
J. Cruz
D. Jimenez
AT BAT:
Mouse over icons to see play info
Click at bat number to get pitch location and hit chart info.
AB1: Singled
AB2: Flied out
AB3: Singled
AB4: Walked
Cumulative
Ball
Strike
Foul
In play
Hit
Batted Out
PITCHERS:
B. Claussen (L)
B. Claussen (L)
B. Claussen (L)
D. Graves (R)
KEY:
SL - Strikes Looking
SS - Strikes Swinging

Cincinnati	AB	R	H	RBI	BB	SO	LOB	AVG
Freel, 2B	5	2	2	0	0	2	1	.382
Griffey, CF	4	0	1	0	0	1	1	.203
Casey, 1B	4	0	3	1	0	0	0	.288
Randa, 3B	4	0	0	0	0	0	4	.262
Dunn, LF	2	0	0	0	2	0	2	.288
Kearns, RF	4	0	2	0	0	1	4	.222
Aurilia, SS	4	0	1	0	0	1	4	.220
LaRue, C	3	0	0	0	0	2	1	.222
Claussen, P	2	0	0	0	0	0	0	.000
Valentine, P	0	0	0	0	0	0	0	.000
a-Cruz, PH	1	0	0	0	0	0	0	.200
Wagner, P	0	0	0	0	0	0	0	.000
b-Jimenez, PH	1	0	0	0	0	0	1	.200
Graves, P	0	0	0	0	0	0	0	.000
Totals	**34**	**2**	**9**	**1**	**2**	**7**	**18**	

a-Grounded out for Valentine in the 7th. b-Lined out for Wagner in the 9th.

BATTING
2B: Casey (3, Perisho), Aurilia (4, Jones).
TB: Freel 2; Griffey; Casey 4; Kearns 2; Aurilia 2.
RBI: Casey (4).
Runners left in scoring position, 2 out: Dunn; Aurilia; Kearns; Freel.
S: LaRue.
GIDP: Aurilia.
Team LOB: 8.

BASERUNNING
SB: Freel (2, 2nd base off Moehler/Lo Duca).

FIELDING
DP: (Freel-Aurilia-Casey).

Florida	AB	R	H	RBI	BB	SO	LOB	AVG
Pierre, CF	5	0	0	0	0	1	2	.295
Easley, 2B	3	0	0	0	1	0	0	.160
Delgado, 1B	3	0	0	0	0	1	1	.319
Encarnacion, RF	3	0	0	0	1	0	1	.277
Lo Duca, C	3	1	2	0	1	0	1	.368
Lowell, 3B	3	0	1	0	1	1	3	.183
Conine, LF	4	0	0	0	0	2	7	.222
Gonzalez, SS	2	0	0	0	1	0	2	.211
b-Cabrera, PH	1	0	1	1	0	0	0	.310
1-Aguila, PR	0	0	0	0	0	0	0	.200
Moehler, P	2	0	0	0	0	0	3	.000
a-Castillo, PH	1	0	0	0	0	0	0	.288
Perisho, P	0	0	0	0	0	0	0	.000
Jones, P	0	0	0	0	0	0	0	.000
c-Harris, PH	0	0	0	0	1	0	0	.286
Totals	**30**	**1**	**4**	**1**	**6**	**5**	**20**	

a-Grounded out for Moehler in the 7th. b-Singled for Gonzalez in the 9th. c-Walked for Jones in the 9th.
1-Ran for Cabrera in the 9th.

BATTING
2B: Lowell (3, Claussen).
TB: Lo Duca 2; Lowell 2; Cabrera.
RBI: Cabrera (16).
2-out RBI: Cabrera.
Runners left in scoring position, 2 out: Moehler 2; Gonzalez; Conine; Pierre.
GIDP: Delgado.
Team LOB: 9.

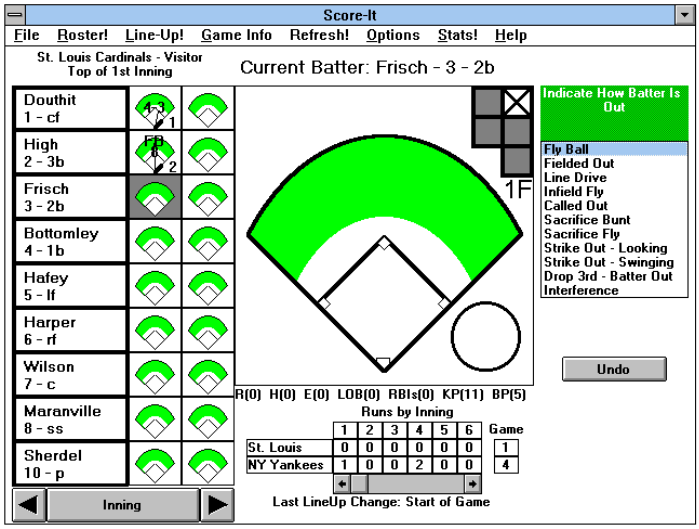
FIELDING
E: Easley (1, throw).
DP: (Easley-Gonzalez-Delgado).

Cincinnati	IP	H	R	ER	BB	SO	HR	ERA
Claussen (W, 1-1)	5.1	3	0	0	3	3	0	2.61
Valentine (H, 2)	0.2	0	0	0	0	1	0	3.12
Wagner (H, 6)	2.0	0	0	0	1	0	0	0.96
Graves (S, 7)	1.0	1	1	1	2	1	0	4.32

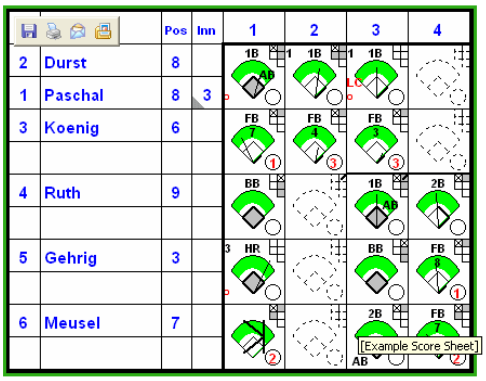
Florida	IP	H	R	ER	BB	SO	HR	ERA
Moehler (L, 1-1)	7.0	7	2	1	1	5	0	1.83
Perisho	0.1	1	0	0	0	1	0	1.80
Jones	1.2	1	0	0	1	1	0	2.00

IBB: Gonzalez (by Claussen), Dunn (by Jones).
HBP: Delgado (by Claussen).
Pitches-strikes: Claussen 97-55, Valentine 9-6, Wagner 20-13, Graves 30-16, Moehler 100-65, Perisho 10-6, Jones 27-16.
Ground outs-fly outs: Claussen 6-7, Valentine 0-1, Wagner 6-0, Graves 0-2, Moehler 8-8, Perisho 0-0, Jones 2-2.
Batters faced: Claussen 23, Valentine 2, Wagner 6, Graves 6, Moehler 28, Perisho 2, Jones 7.
Inherited runners-scored: Valentine 2-0, Jones 1-0.
Umpires: HP: Mike DiMuro. 1B: Alfonso Marquez. 2B: Mark Carlson. 3B: Joe West.
Weather: 77 degrees, sunny.
Wind: 17 mph, L to R.
T: 2:42.
Att: 28,219.

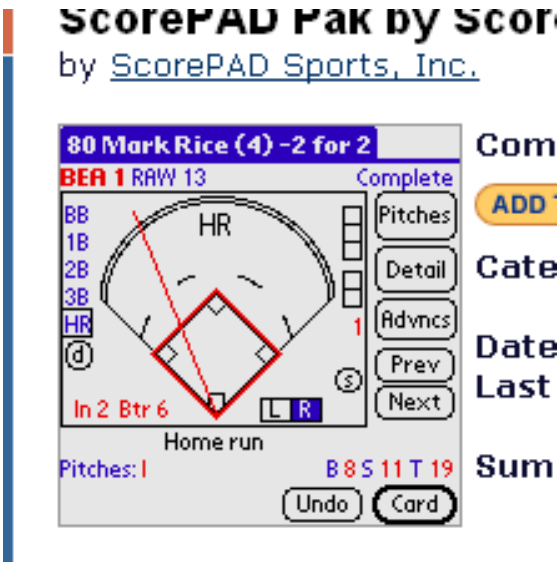
Appendix 6

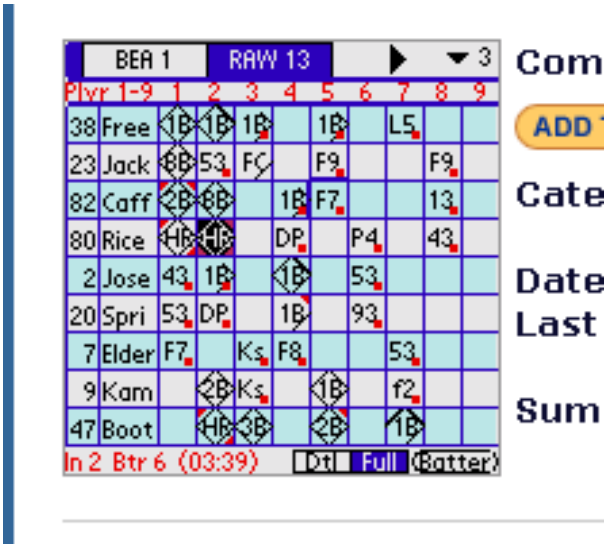


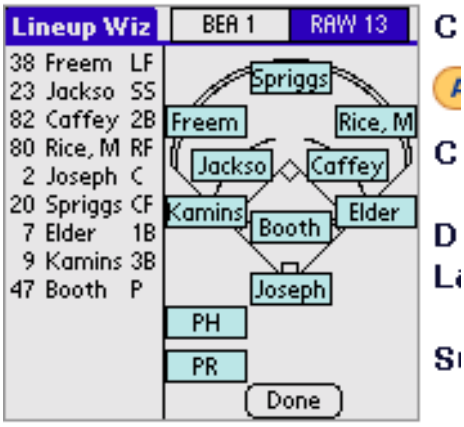
Above is the screen you see when indicating a player is out. Other screens for batter on base,



Appendix 7


ScorePAD Pak by Scor
by ScorePAD Sports, Inc.
80 Mark Rice (4) -2 for 2
BEA 1 RAW 13
Complete
BB
1B
2B
3B
HR
HR
Pitches
Detail
Advncs
Prev
Next
In 2 Btr 6
L R
Home run
Pitches: I
B 8 S 11 T 19
Undo
Card
Com
ADD
Cate
Date
Last
Sum


BEA 1
RAW 13
3
Plyr 1-9 1 2 3 4 5 6 7 8 9
38 Free
23 Jack
82 Caff
80 Rice
2 Jose
20 Spri
7 Elder
9 Kam
47 Boot
In 2 Btr 6 (03:39)
Dtl
Full
Batter
Com
ADD
Cate
Date
Last
Sum

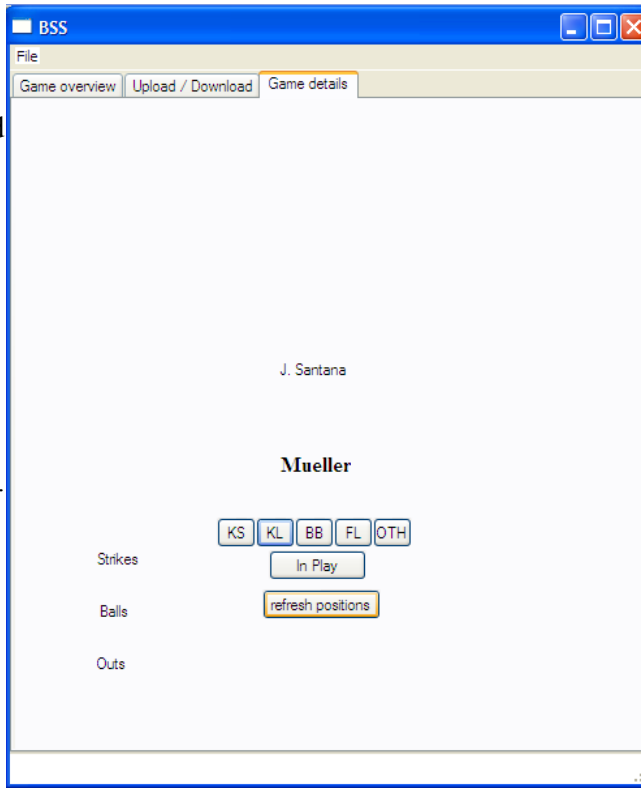

Lineup Wiz
BEA 1
RAW 13
38 Freem LF
23 Jackso SS
82 Caffey 2B
80 Rice, M RF
2 Joseph C
20 Spriggs CF
7 Elder 1B
9 Kamins 3B
47 Booth P
Spriggs
Freem
Rice, M
Jackso
Caffey
Kamins
Booth
Elder
Joseph
PH
PR
Done

Appendix 9

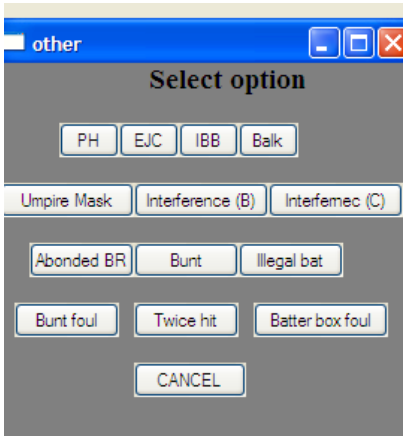
User manual

This is a walkthrough guide to the input system and helps you, the user, get acquainted with it. To load the system up, you must first locate the file that is stored on your disk space and double-click it. This file is called BSS.exe and can usually found on the desktop of the system in question. This will load a "command line shell", which can be ignored and the actual scoring program on top of this.

The first screen that is displayed allows you to enter the teams that are competing in this particular match. Selecting the teams using the combo boxes provided does this. When the teams have been selected click on the OK button. This will prompt you to click on the newly formed tab at the top of the program. This will take you to the main screen, which will look similar to:



This is where the bulk of the scoring takes place. Each ball can be recorded by the buttons underneath the batter's name. If the batter puts the ball in play or they are putout in a way other than a strikeout or the ways listed on the screen below (which is accessible via the "oth" button) then the "in play" button should be pressed.



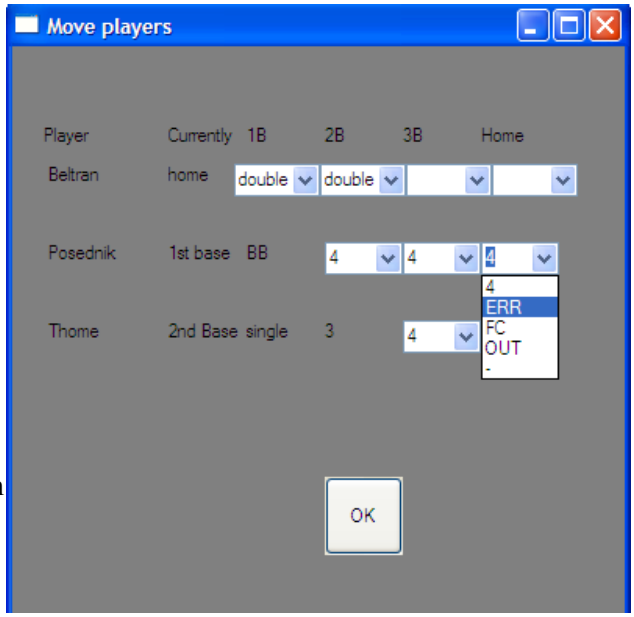
When the ball is put in play, the next screen asks for the position that it was hit to, this only requires clicking the mouse once and then you are taken to the next screen.

The next screen is where the play is chosen. When the relevant play is chosen, then the appropriate screen is displayed, which allows you to enter more details on that particular play, such as who caught the ball, how the groundout was achieved or how the current base-runners have advanced around the bases.

The numbers that must be entered in the "OUT" screens are the standard numbers used in baseball for fielders.

To advance the runners around the bases the following screen is used.

The drop down boxes have default values assigned to them and usually all that is required is to click OK. If it is not correct then the correct play can be selected from the drop down list as shown (showing Posednik advancing to home plate on an error.



The above process should be repeated until the game is finished and the system will automatically detect this.

When the game has been completed, it is then possible to upload the results to the web site. The first stage in this is to connect to the Internet. This shall be done, either before the program is started or when the program is running.

IT IS IMPORTANT THAT NO MORE SCORES WILL BE INPUT BEFORE THE PREVIOUS GAME HAS BEEN UPLOADED.

It is simple to upload the results and this is done by clicking on the relevant tab and clicking "upload" and entering your user name and password.

It is recommended that whenever you upload, you also download and update the files on your own machine. This is done by clicking the download button.

To exit the program click the X in the top right hand corner.

FAQs

What if I can't find the file?

Use windows search facility by pressing the windows key on your keyboard and the "F" key together, click "search all files and folders" if available and enter the file name – "BSS.exe" in the search criteria

What if I choose the wrong position of the ball?

Simply cancel the next step by pressing the  *button and clicking "in play" again to reselect the position*

What are the standard numbers for fielders?

These are standard for all baseball games –

1 – Pitcher

2 – Catcher

3 – 1st Baseman

4 – 2nd Baseman

5 – 3rd Baseman

6 – Shortstop

7 – Left fielder

8 – Centre fielder

9 – Right fielder

What if I make an error?

This can usually be rectified by pressing the  *button on the relevant screen, except when it has been written to file. This can then be rectified via the website admin page.*

What if I forget my user name / password

This cannot be retrieved from the system. If this happens then you must contact BSUK, or use the admin page of the website.

Appendix 11

```
if ManAt1.getname() <> "":
                    if AB.getb1() <> "OUT":
                        if (ManAt1.getb2() == "-") or (ManAt1.getb2() == ""):
                            self.error()


if ManAt2.getname() <> "":
                    if (ManAt2.getb3() == "-") or (ManAt2.getb3() == ""):
                        if ((AB.getb2() <> "-") and (AB.getb2() <>"")) or (((ManAt1.getb2()
                        <> "-") and (ManAt1.getb2() <> "")) and (ManAt1.getname() <>"")):
                                    self.error() # two men on 2nd
if ManAt1.getname() <> "":
                    if (ManAt1.getb3() == "-") or (ManAt1.getb3() ==""):
                        if (AB.getb2() <> "-") and (AB.getb2() <> ""):
                                self.error() # two men on 2nd
if ManAt3.getname() <> "":
                    if (ManAt3.getb4 =="-") or (ManAt3.getb4() ==""):
                        if ((ManAt1.getb3() <> "") and (ManAt1.getb3() <> "-")) or
                        ((AB.getb3() <> "") and (AB.getb3() <> "-")) or ((ManAt2.getb3()
                        <> "") and (ManAt2.getb3() <> "-")):
                                    self.error() # two men on 3rd
if (ManAt2.getname()<>"") and ((ManAt2.getb4()=="-") or (ManAt2.getb4() == "")):
                        if ((ManAt1.getb3()<>"-") and (ManAt1.getb3()<>"")) or
                        ((AB.getb3()<>"-") and (AB.getb3()<>"")):
                                    self.error() # two men on 3rd
if (ManAt1.getname()<>"") and ((ManAt1.getb4()=="-") or (ManAt1.getb4() == "")):
                        if ((AB.getb3()<>"-") and (AB.getb3()<>"")):
                                    self.error() # two men on 3rd
```

Appendix 12

```
def Dostuff(self, event):
        #  First set values of advancements
        AB.setb1(self.abc1.GetValue())
        AB.setb2(self.abc2.GetValue())
        AB.setb3(self.abc3.GetValue())
        AB.setb4(self.abc4.GetValue())
        if ManAt1.getname() <> "":
                ManAt1.setb2(self.comb1.GetValue())
                ManAt1.setb3(self.comb2.GetValue())
                ManAt1.setb4(self.comb3.GetValue())
                if ManAt2.getname() <> "":
                        ManAt2.setb3(self.comb4.GetValue())
                        ManAt2.setb4(self.comb5.GetValue())
                        if ManAt3.getname() <> "":
                                ManAt3.setb4(self.comb6.GetValue())
                elif ManAt3.getname() <> "":
                        ManAt3.setb4(self.comb7.GetValue())
        elif ManAt2.getname() <> "":
                ManAt2.setb3(self.comb8.GetValue())
                ManAt2.setb4(self.comb9.GetValue())
                if ManAt3.getname() <> "":
                        ManAt3.setb4(self.comb10.GetValue())
        elif ManAt3.getname() <> "":
                ManAt3.setb4(self.comb11.GetValue())
        # Then move the player onwards
        if (ManAt3.getb4() <> "-") and (ManAt3.getb4() <> ""):
              if ManAt3.getb4() <> "OUT":
                Write_to_file(ManAt3)
                clearstats(ManAt3)
                ListOfBatters.append(ManAt3)
                Swap(ManAt3,Blank)
           else:
                Write_to_file(ManAt3)
                clearstats(ManAt3)
                ListOfBatters.append(ManAt3)
```

```
                Swap(ManAt3,Blank)
            if (ManAt2.getb3() <> "-") and (ManAt2.getb3() <> ""):
              if (ManAt2.getb4() <> "-") and (ManAt2.getb4() <> ""):
                    if ManAt2.getb4() <> "OUT":
                       Write_to_file(ManAt2)
                       clearstats(ManAt2)
                       ListOfBatters.append(ManAt2)
                       Swap(ManAt2,Blank)
                 else:
                       Write_to_file(ManAt2)
                       clearstats(ManAt2)
                       ListOfBatters.append(ManAt2)
                       Swap(ManAt2,Blank)
            else:
                    if ManAt2.getb3() <> "OUT":
                          Swap(ManAt3,ManAt2)
                          Swap(ManAt2,Blank)

            if (ManAt1.getb2() <> "-") and (ManAt1.getb2() <> ""):
                if (ManAt1.getb3() <> "-") and (ManAt1.getb3() <> ""):
                    if (ManAt1.getb4() <> "-") and (ManAt1.getb4() <> ""):
                          if ManAt1.getb4() <> "OUT":
                             Write_to_file(ManAt1)
                             clearstats(ManAt1)
                             ListOfBatters.append(ManAt1)
                             Swap(ManAt1,Blank)
                        else:
                             Write_to_file(ManAt1)
                             clearstats(ManAt1)
                             ListOfBatters.append(ManAt1)
                             Swap(ManAt1,Blank)
                    else:
                          if ManAt1.getb3() <> "OUT":
                                Swap(ManAt3,ManAt1)
                                Swap(ManAt1,Blank)
                else:
```

```
                    if ManAt1.getb2() <> "OUT":
                        Swap(ManAt2,ManAt1)
                        Swap(ManAt1,Blank)


    if (AB.getb1() <> "-") and (AB.getb1() <> ""):
        if (AB.getb2() <> "-") and (AB.getb2() <> ""):
            if (AB.getb3() <> "-") and (AB.getb3() <> ""):
                if (AB.getb4() <> "-") and (AB.getb4() <> ""):


                    if AB.getb4() <> "OUT":
                        Write_to_file(AB)
                        clearstats(AB)
                        ListOfBatters.append(AB)
                        Swap(AB,ListOfBatters.pop(0))
                    else:
                        Write_to_file(AB)
                        clearstats(AB)
                        ListOfBatters.append(AB)
                        Swap(AB,ListOfBatters.pop(0))
                else:
                    if AB.getb3() <> "OUT":
                        Swap(ManAt3,AB)
                        Swap(AB,ListOfBatters.pop(0))
            else:
                if AB.getb2() <> "OUT":
                    Swap(ManAt2,AB)
                    Swap(AB,ListOfBatters.pop(0))
        else:
            if AB.getb1() <> "OUT":
                Swap(ManAt1,AB)
                Swap(AB,ListOfBatters.pop(0))
```

Appendix 15

num	Need	How test	Weighting	Score out of 100	Weighted Score	Comments
	INPUT SYSTEM					
1.1	System shall be easy to use	3rd Party	13.6	80	10.9	Obtained by assessment criteria
1.2	System shall not allow invalid input	Test Data	21.5	90.0	19.4	No "invalid" input allowed but still errors can occur
1.3	System shall be attractive	3rd Party	12.5	60	7.5	Obtained by assessment criteria
1.4	System shall use appropriate colours	3rd Party	12.5	95	11.9	Uses the colours defined by user for MS Windows
1.5	System shall minimize text input	Observation	8.3	98	8.1	Only 2 text input - both validated
1.6	System shall be able to record each ball	Observation	11.5	100	11.5	Every option covered
1.7	System shall record where ball was hit	Observation	12.5	80	10.0	Erroroneous value if window is manually moved
1.8	System shall record all events that happen	Observation	11.5	90	10.4	Most events included
1.9	System shall have no "lag"	Observation	19.3	80	15.4	No lag except on start-up, upload
1.10	System shall not allow contradicting events	Test Data	24.7	95	23.5	Procedures used to stop this; example in appendix 11
1.11	System shall update events when input	Observation	9.3	10	0.9	Button needed to update baserunner posistions
1.12	System shall have appropriate error messages	Observation	18.1	85	15.4	Messages - clear and concise; Windows errors still meaningless
1.13	System shall automatically enter inevitable events	Test Data	13.6	95	12.9	See functional requirements
1.14	System shall recognise possible input errors	Test Data	8.4	0	0.0	Only spots errors, not possible mistypes etc.
1.15	System should allow correction of errors	Observation	10.5	50	5.3	Can correct to an extent. But once written, need to use website
1.16	System shall be easy to install	3rd Party	16.3	95	15.5	Very simple, just copy and paste

1.17	System shall be a .EXE file	Observation	12.7	100	12.7	It is
1.18	System shall save data to an external file	Test Data	17.8	100	17.8	It does, a number of fiels are used
1.19	Coding shall be well commented	Observation	13.6	80	10.9	Mostly commented with headers for each function
1.20	Coding shall be split into appropriate functions	Observation	13.6	15	2.0	Not enough encapsulation
1.21	Coding shall be efficient	Test Data	13.6	25	3.4	Not efficient, but does the job it si supposed to
1.22	There shall be a user manual	Observation	22.5	100	22.5	Yes - Appendix 9
1.23	There shall be a one click interface to upload data	Observation	13.9	90	12.5	Almost - oen click then user name and password entering
1.24	There shall be a troubleshooting section	Observation	20.4	20	4.1	Small section in user manual - appendix 9
1.25	Uploading shall be password protected	Test Data	9.7	100	9.7	User name and password required
1.26	The system shall include pictures of players	Observation	4.1	0	0.0	No pictures due to memory and speed restrictions
1.27	The system shall allow notes to be added to plays	Test Data	7.4	0	0.0	No notes allowed
1.28	The system shall automatically rotate through players	Observation	11.4	100	11.4	Automatically goes through each team when batting
1.29	The system shall allow substitutions of batters	Test Data	13.6	100	13.6	Yes - via "oth" button then "PH"
1.30	The system shall allow substitutions of fielders	Test Data	5.2	0	0.0	Not allowed due to complex function required
	DATABASE					
2.1	Data shall be stored in a relational database	Observation	21.3	95	20.2	Fully normalised database used - see section 2.12.1
2.2	Data shall not be stored more than once	Observation	21.2	90	19.1	Very small amount of data repeated - unavoidable. See 2.12.1
2.3	Data shall be updated when it changes	Test Data	45.1	75	33.8	Done as soon as possible but not at real-time
2.4	Data shall not be directly manipulated	Test Data	7.3	95	6.9	Can not be manipulated except via given

						methods
2.5	Data shall have clear field names	Observation	14.8	95	14.1	As clear as possible - some are lengthy
2.6	Data shall be accessible via Internet	Test Data	12.6	85	10.7	Accessible via Internet, but restricted to certain queries
2.7	Data shall be compatible with current system	Test Data	9.6	0	0.0	Replaces current system
2.8	There shall be no installation needed	Observation	9.6	50	4.8	Installation shall be performed by developer
	WEBSITE					
3.1	Website shall be attractive	3rd Party	16	70	11.2	Obtained from assessment criteria
3.2	Website shall be easy to use	3rd Party	16.7	75	12.5	Obtained from assessment criteria
3.3	There shall be a FAQ section	Observation	19.3	25	4.8	No FAQ but help on most pages
3.4	Website shall fit it with current site	3rd Party	15.3	90	13.8	Uses exact same layout/colours
3.5	Website shall allow queries to database	Test Data	25.4	90	22.9	Allows queries, but only those given
3.6	Website shall allow generation of statistics	Test Data	28	90	25.2	Most possible statistics can be calculated
3.7	Website shall not contain any sensitive information	Observation	36.9	100	36.9	No sensitive information on database at all
3.8	Website shall allow users to see full details of all games	Observation	15.2	90	13.7	Can see most details - excluding little details such as temperature
3.9	Website shall allow users to see stats on one player	Observation	9.9	95	9.4	This can be done via a query. Some stats not shown. Eg SLG
3.10	Website shall allow users to see stats on all players	Observation	12.7	10	1.3	Can see all players but need lots of queries run seperately
3.11	Website shall allow production of comparisons	Test Data	9.3	25	2.3	Not really but popup window allows manual comparisons
3.12	Website shall contain fixtures	Observation	15.4	20	3.1	Kept from original site as no clear improvements can be made
3.13	Website shall contain league tables	Observation	15.4	20	3.1	Kept from original site as no clear improvements can be made

3.14	Website shall not include copyrighted information	Observation	36.9	100	36.9	No copyrighted information/data used except logo of BSUK
3.15	Website shall increase in hits/day	Counter	12.3	50	6.2	not known
3.16	Website shall have no errors or missing pages	Observation	31.6	100	31.6	Limited number of pages facilitates this
3.17	Website shall only allow queries on games in past	Test Data	15.5	100	15.5	Combobox ensures this
3.18	Website shall not allow changing of data	Test Data	10.6	75	8.0	Allowed with authorisation
3.19	Website shall include password protected admin page	Observation	9.8	100	9.8	Yes it does
3.20	Admin page shall not be authorised by general public	Test Data	35.8	100	35.8	User name / password required
3.21	Admin page shall allow manipulation of data in database	Test Data	8.7	70	6.1	Allowed to some extent
	GENERAL					
4.1	Number of fans and players shall go up	Observation	13.3	50	6.7	Not known
4.2	Systems shall have similar aesthetics	Observation	12.2	80	9.8	Very similar colours used
4.3	Systems shall be developed with a recognised manner	Observation	13.8	70	9.7	Developed using OO techniques; structured report used
4.4	Systems shall be consistant in style and functionality	Observation	13.2	60	7.9	Varies depending on function being undertaken

TOTAL WEIGHTING:	994.3	4373		752.619
Total interested in:	63	6300		
Average Weighting per need:	15.78254	0.694127		11.94633

APPENDIX 16

Num	Requirment	Met	Comment if not met
	INPUT PROGRAM		
1.1	Can not put details in before selected teams	1	
1.2	Can not have two players on same base	1	
1.3	System should cycle through players automitically	1	
1.4	System should remember the place of batting order	1	
1.5	System should automatically increase inning	1	
1.6	System should automatically increase scores	1	
1.7	Should not be able to manually increase scores	1	
1.8	Should not be able to insert "non-real" players	1	
1.9	System should not take more than 5 seconds to load	0	Due to memory restrictions, this is longer
1.10	Data should not take more than 10 seconds to upload	1	
1.11	Should not be able to activate inactive window	0	Unable to create restriction
1.12	Should be cancel button on all pages	0.5	There is a standards Windows close button with same effect
1.13	Should confirm before writing to file	0.5	does, but not to extent that is required
1.14	Should automatically record out when 3rd strike	1	
1.15	Should automatically advance batters when 4th ball	1	
1.16	Foul should increase strikes except on 2 strikes	1	
	WEBSITE		
2.1	Should be no text input	1	
2.2	Automatically populate player combo box when team selcted	0.5	Need to press a button
2.3	Should give all stats on one particular player	1	
2.4	Should have page for each match taken place	1	
2.5	Can not write to database	1	
2.6	Should be two pages with search facilities - player + match	1	
2.7	Can not select player without first selecting a team	1	
2.8	Should be a button to produce printer friendly sheet.		Whole site is printer friendly

Appendix 17 – table 1

Attribute	Type	Notes
player_id	Integer	unique identifier for player
forename	string	player's first name
surname	string	player's surname
team_id	Integer	unique identifier for team
team_name	string	teams name
team_location	string	where the team is located
colour1	string	the main colour of uniforms
colour2	string	the 2nd colour of uniforms
contact	string	contact name at club
contact_email	string	the email of the above person
contact_tel	string	telephone num of above person
league	string	the league that the team compete in
posistion	Integer	the position in this league
team_wins	Integer	Amount of times team has won
team_losses	Integer	Amount of times team has lost
for	Integer	total number of runs gained
against	Integer	total number of runs conceeded
player_pos	string[2]	posistion this player plays
hits	integer	number of times player has hit ball
RBI	integer	See appendix 18
HR	integer	See appendix 18
2B	integer	total number of doubles acchieved
3B	integer	total number of tripples acchieved
BB	integer	See appendix 18
IBB	integer	See appendix 18
E	integer	See appendix 18
SB	integer	See appendix 18
CS	integer	See appendix 18
FREE1B	integer	total times a player has been given 1B
Avg	real	See appendix 18
Slugging	real	Slugging percentage of player
OBP	real	See appendix 18
AB	Integer	See appendix 18
ERA	real	See appendix 18
Player_wins	Integer	Number of wins accredited to this player
Player_losses	Integer	Number of loses accredited to this player
saves	Integer	Number of saves accredited to this player
IP	Integer	Innings pitched - total
CG	Integer	Total number of complete games
Shutouts	Integer	Total number of shutouts acchieved
Pitch_BB	Integer	Total number of walks given up
K	Integer	Total number of strike outs
R	Integer	Total number of runs given up by pitcher
ER	Integer	Total number of earned runs given up
HR_allowed	Integer	Total number of HR given up by pitcher

Opponent	Integer	Team_id of opposing team
home_score	Integer	total number of runs in this match by home team
away_score	Integer	total number of runs in this match by away team
winning_pitcher	Integer	player_id of pitcher that won
losing_picther	Integer	player_id of pitcher that lost
save	Integer	player_id of pitcher accredited with save
datea	Date/Time	Date of match
inning	Integer	inning number
inning_home_runs	Integer	total runs by home team in inning
inning_away_runs	Integer	total runs by away team in inning
1stBase	string	how this player got to 1st base
2ndBase	string	how this player got to 2nd base
3rdBase	string	how this player got to 3rd base
Home	string	how this player got to home plate
out	string	how this player was putout
hitToX	Integer	x co-ord of where ball was hit
hitToY	Integer	y co-ord of where ball was hit
RBI_atbat	Integer	how many RBIs acchieved in this AB
replacment_id	Integer	player_id of player replacing this batter
replacment_place	string	where this player was replaced
CountS	Integer	number of strikes when this player was safe/out
CountB	Integer	number of balls when this player was safe/out
pitcher_id	Integer	player_id of pitcher facing batter

Appendix 17 – table 2

Num	**Attribute**	**How got**	**Notes**
1	team_wins	COUNT (SELECT * FROM match WHERE ((home_team = @team_id) AND (@home_score>@away_score) OR ((away_team = @team_id) AND (@home_score<@away_score)	@team_id is team in question's id. @home_score see 31, @away_score see 32
2	team_losses	COUNT (SELECT * FROM match WHERE ((home_team = @team_id) AND (@home_score<@away_score) OR ((away_team = @team_id) AND (@home_score>@away_score)	@team_id is team in question's id. @home_score see 31, @away_score see 32
3	for	No longer needed	no league table to be developed
4	against	No Longer needed	no league table to be developed
5	hits	(COUNT (SELECT * FROM at_bat WHERE (b1<>"") and (player_id=@player_id)) - Free1B	@player_id is player in question's id. Free1b - see 14
6	RBI	COUNT (SELECT * FROM at_bat WHERE b4=@player_id)	@player_id is player in question's id.
7	HR	COUNT (SELECT * FROM at_bat WHERE b4="HR" and player_id=@player_id)	@player_id is player in question's id.
8	2B	COUNT (SELECT * FROM at_bat WHERE b2="double" and player_id=@player_id)	@player_id is player in question's id.
9	3B	COUNT (SELECT * FROM at_bat WHERE b3="triple" and player_id=@player_id)	@player_id is player in question's id.
10	BB	COUNT (SELECT * FROM at_bat WHERE b1="BB" and player_id=@player_id)	@player_id is player in question's id.
11	IBB	COUNT (SELECT * FROM at_bat WHERE b1="IBB" and player_id=@player_id)	@player_id is player in question's id.
12	SB	COUNT (SELECT * FROM at_bat WHERE (b2="SB" or b3="SB" or b4 = "SB") and player_id = @player_id	@player_id is player in question's id.
13	CS	COUNT (SELECT * FROM at_bat WHERE out = "CS" and player_id=@player_id	@player_id is player in question's id.

14	FREE1B	COUNT (SELECT * FROM at_bat WHERE b1 like "OTH%" or b1="FC") + @BB	@player_id is player in question's id. @BB - see 10
15	Avg	@hits/@AB	@hits - see 5. @AB - see 18
16	Slugging	@hits + @2B + @3B + @3B + (3*@HR) / @AB	@hits - see 5, @2B - see 8. @3B - see 9. @HR - see 7. @AB - see 18
17	OBP	@Hits + @Free1B / @AB + @Free1B	@hits - see 5. @Free1b - see 14 @AB - see 18
18	AB	COUNT (SELECT * FROM at_bat WHERE player_id = @player_id) - @Free1B	@player_id is player in question's id. @Free1b - see 14
19	ERA	(@ER / @IP)*9	@ER - see 29 @IP - see 23
20	Player_wins	COUNT (SELECT * FROM matcha WHERE winning_pitcher = @player_id)	@player_id is player in question's id.
21	Player_losses	COUNT (SELECT * FROM matcha WHERE losing_pitcher = @player_id)	@player_id is player in question's id.
22	saves	COUNT (SELECT * FROM matcha WHERE savea = @player_id)	@player_id is player in question's id.
23	IP	COUNT (SELECT * from at_bat where pitcherid = @Player_id and out <>"")/3	@player_id is player in question's id.
24	CG	No longer needed	Stat not needed - Change in spec
25	Shutouts	No longer needed	Stat not needed - Change in spec
26	Pitch_BB	COUNT (SELECT * FROM at_bat where pitcherid= @Player_id and b1="BB")	@player_id is player in question's id.
27	K	COUNT (SELECT * FROM at_bat where pitcherid= @Player_id and out like "K%")	@player_id is player in question's id.
28	R	(COUNT (SELECT * FROM at_bat WHERE b4<>"" and pitcherId = @player_id))	@player_id is player in question's id.
29	ER	(COUNT (SELECT * FROM at_bat WHERE b4<>"" and pitcherId = @player_id))	@player_id is player in question's id.

30	HR_allowed	(COUNT (SELECT * FROM at_bat WHERE b4="HR" and pitcherId = @player_id))	@player_id is player in question's id.
31	home_score	(COUNT (SELECT * FROM at_bat WHERE b4<>"" and inning_num like "%.1" and match_id = @match_id)	@match_id is id of match in question
32	away_score	(COUNT (SELECT * FROM at_bat WHERE b4<>"" and inning_num like "%.2" and match_id = @match_id)	@match_id is id of match in question

Appendix 18 _- Abbreviations and acronyms used

The following is a list of abbreviations and acronyms that are used within this project

AB – at bat: a player's appearance as a batter

AVG – The percentage of times a player hits the ball when they have an AB

BB – base on balls: when a batter advances to 1st base on 4 "balls"

BOC – British Olympic committee

BSUK – baseball softball UK: the company this software is being developed for

DB – Database

DPA – Data protection Act

EJC – Ejected: when the player is ejected from the game for bad behaviour

ER – Entity Relationship: Used in database design to show the relation between tables

ERA – Earned run average: How many runs a pitcher gives up on average in 9 innings.

ERR – Error: An error committed by a fielder that allows a runner to advance

FC – Fielders Choice: Where a fielder chooses to put out a different player, which allows this runner to advance

FL – Foul: A ball hit in foul territory

GUI – Graphical User Interface

HR – Home Run: When a player makes it round all the bases in one go

IBB – Intentional base on balls: Where a batter is intentionally walked

IF – Importance factor: Used in measuring success in some chapters.

IFR – Infield Fly Rule: A rule in baseball, where the ball is not caught but is counted as being caught because of the inevitability of this event.

K - Strike

KS – Strike swinging: Batter swings at a pitch and therefore is classed a strike

KL – Strike looking: A pitch that is in the strike zone but the batter does not swing at

MLB – Major League Baseball: The organising committee of baseball in America.

OBP – On base %: The percentage of times a player gets on base from a being a hitter.

OO – Object oriented.

OTH – Other: Leading to other screens with less frequent play on.

PH – Pinch Hit: Where a player is substituted whilst batting

RBI – Run(s) batted in: Where a player hits a ball, which allows another runner to score on the same play

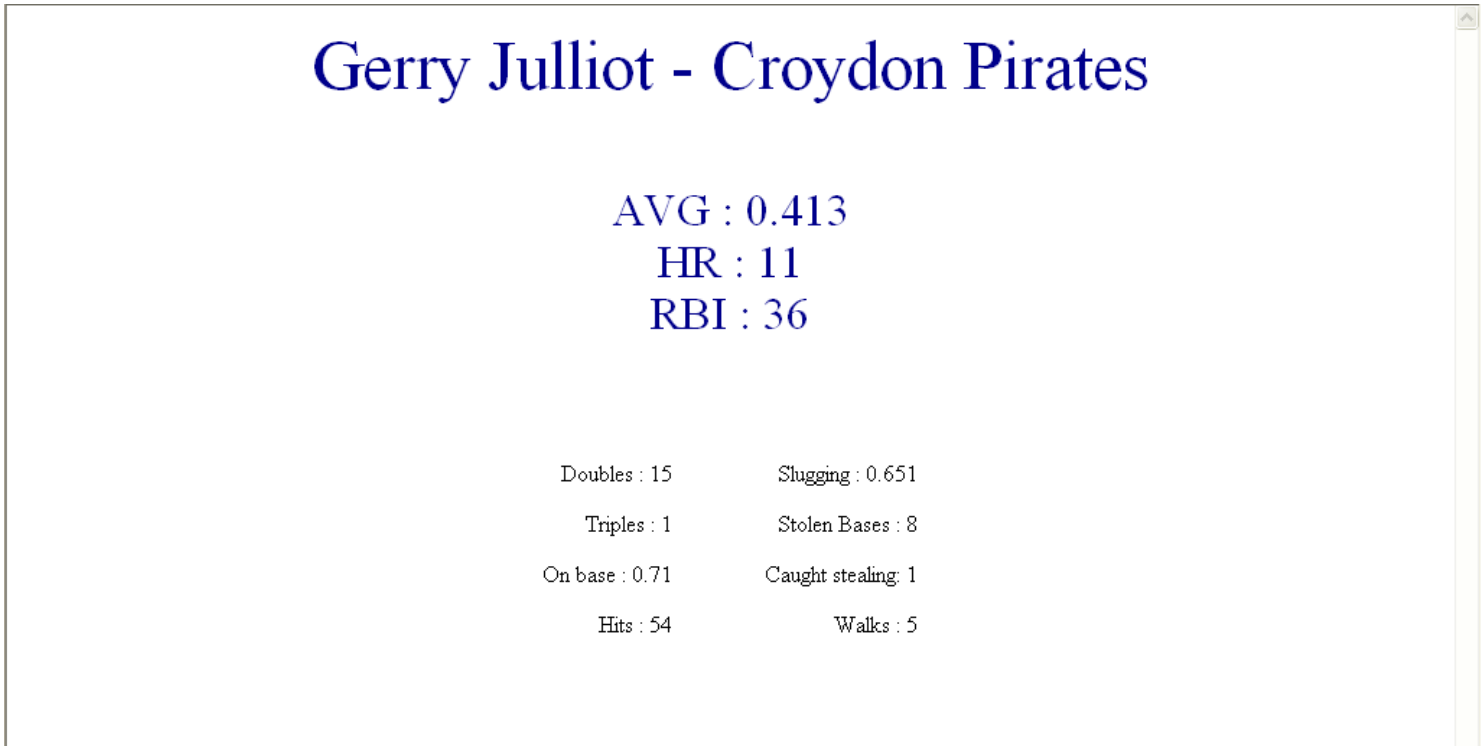
Sac – Sacrifice: where a batter purposefully gets out in order to advance another runner

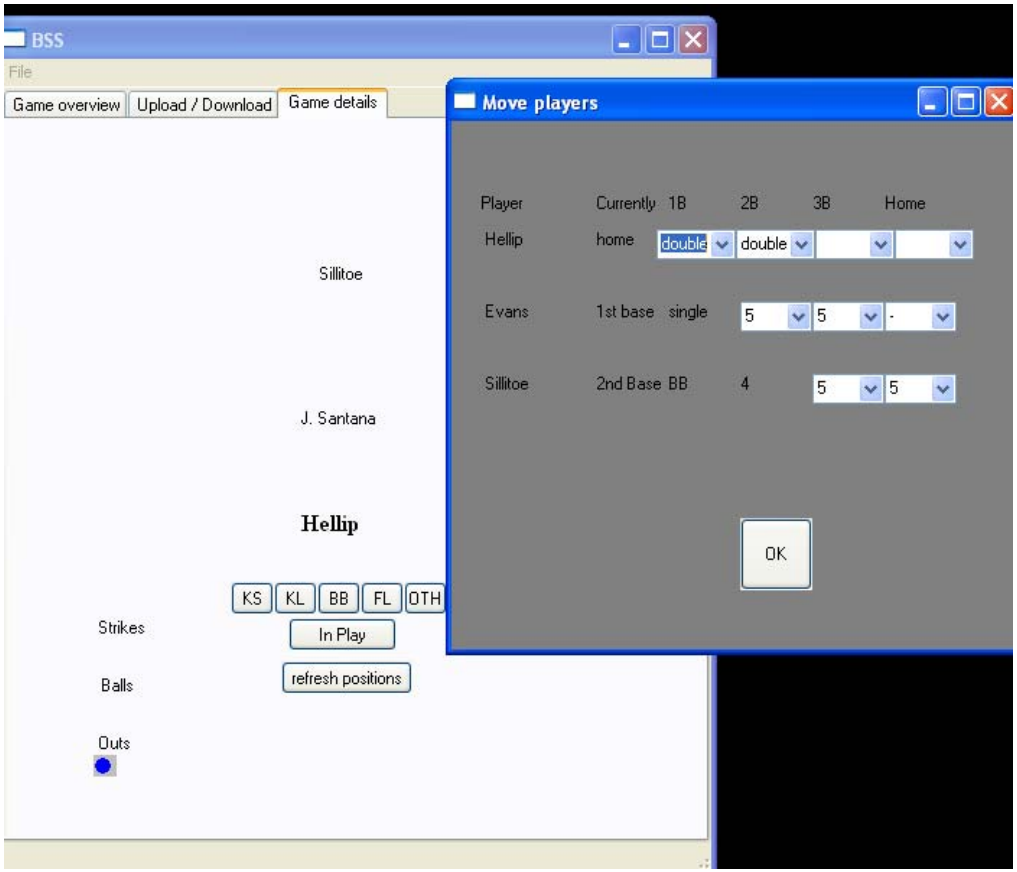
SB – Stolen base: Where a runner advances a base by "stealing" between pitches.

SQL – Structured Query Language: A language that allows the querying of databases.

UML – Unified modelling language: An object oriented design technique.

Appendix 20 – sample screen shots


Gerry Julliot - Croydon Pirates
AVG : 0.413
HR : 11
RBI : 36
Doubles : 15
Slugging : 0.651
Triples : 1
Stolen Bases : 8
On base : 0.71
Caught stealing: 1
Hits : 54
Walks : 5


BSS
File
Game overview
Upload / Download
Game details
Sillitoe
J. Santana
Hellip
KS
KL
BB
FL
OTH
Strikes
In Play
Balls
refresh positions
Outs
Move players
Player
Currently
1B
2B
3B
Home
Hellip
home
double
double
Evans
1st base
single
5
5
-
Sillitoe
2nd Base
BB
4
5
5
OK

Appendix 21 – code for input program

```
#!usr/bin/python
from wxPython.wx import *
from batter import *
import os
import sys


ID_ABOUT=101
ID_BUTTON1=110
ID_EXIT=200
(x,y)=(20,200)
inning = 1
outs = 0
homescore = 0
awayscore = 0
home = 1
match = 22
homenum = 1
awaynum = 1
homeplayerfile = open("playersh","r")
awayplayerfile = open("playersa","r")
battera = 0

def clearstats(a):
        # clears the current batter
        a.setb1("")
        a.setb2("")
        a.setb3("")
        a.setb4("")
        a.setout("")
        a.sethitto((0,0))
        a.setcount((0,0))
        a.setpitcher(0)

def Get_next1(): # function to get the next player to bat
                x = Batter(0, 0, 0, 0, "", "", "", "", "", "", (0,0), 0, "", (0,0),0)
                return x
def Get_next(): # function to get the next player to bat
                global inning
                global match
                global home
                global homenum
                global awaynum
                global homeplayerfile
                global awayplayerfile
                global battera
                print homenum
                print awaynum
                battera=battera+1
                if home == 0:
                        homeplayerfile.seek(0)
                        for x in range(0,homenum):
                                num = homeplayerfile.readline(5) #first 5 chars
                                players = homeplayerfile.readline()
                        if homenum == 9:
                                homenum = 1
                                homeplayerfile.seek(0)
```

```
                else:
                        homenum = homenum + 1
                batnum = homenum

        else:
                awayplayerfile.seek(0)
                for x in range (0,awaynum):
                        num = awayplayerfile.readline(5) #first 5 chars
                        players = awayplayerfile.readline()
                if awaynum == 9:
                        awaynum = 1
                        awayplayerfile.seek(0)
                else:
                        awaynum = awaynum + 1
                batnum = awaynum


        theReturn = Batter(inning,num,batnum, match,players,"","","","","","",(0,0), 0,
"",(0,0),0)
        return theReturn


def Write_to_file(a): # writes to file
        global inning

        (b,c) = a.gethitto()
        hit = str(b) + ", " + str(c) + ", "
        (d,e) = a.getcount()
        count = str(d) + ", " + str(e) + ", "


        thestring = "("
        thestring = thestring + str(inning) + "," + str(a.getplayer()) +", "
        thestring = thestring + str(a.getbatter()) + ", " + str(a.getmatch())
        thestring = thestring + ", " + a.getb1()+ ", " + a.getb2() + ", " + a.getb3() + ", "
        thestring = thestring + a.getb4() + ", " + a.getout() + ", " + hit
        thestring = thestring + str(a.getreplacement()) + ", " + a.getreplaceplace()
        thestring = thestring + ", " + count + str(a.getpitcher())+ ") \n"

        thefile = open('data','a')
        thefile.write(thestring)
        thefile.close()
        print ("written")


def Addrun(): # adds a run
        global homescore
        global awayscore
        global home
        if home == 0:
                homescore = homescore+1
        else:
                awayscore = awayscore+1

        print("inning :", inning)
        print("outs :", outs)
        print("homescore :", homescore)
        print("away score :", awayscore)
```

```
def Addout(): # adds an out
	global outs
	if outs == 2:
		outs = 0
		Addinning()
	else:
		outs = outs + 1

	print("inning :", inning)
	print("outs :", outs)
	print("homescore :", homescore)
	print("away score :", awayscore)

def endofgame(): # function for end of game
	print("end of game ")


def Addinning(): # adds an inning
	global home
	global inning
	global AB
	global OnDeck
	global Bat1
	global Bat2
	global Bat3
	global Bat4
	global Bat5
	global Bat6
	global Bat7
	global ListOfBatters
	global ManAt1
	global ManAt2
	global ManAt3

	ManAt1 = Get_next1()
	ManAt2 = Get_next1()
	ManAt3 = Get_next1()

	if (inning == 9) and (home == 1) :
		endofgame()
	else:
		if home == 1:
			home = 0
		else:
			home = 1
			inning = inning + 1
		print("inning :", inning)
		print("outs :", outs)
		print("homescore :", homescore)
		print("away score :", awayscore)
		AB = Get_next()
	    OnDeck = Get_next()
		Bat1 = Get_next()
		Bat2 = Get_next()
		Bat3 = Get_next()
		Bat4 = Get_next()
		Bat5 = Get_next()
		Bat6 = Get_next()
		Bat7 = Get_next()
```

```
                ListOfBatters = [OnDeck, Bat1, Bat2, Bat3, Bat4, Bat5, Bat6, Bat7]

def Swap(a,b): # swaps position of two batters
        global ListOfBatters
        a.settinning(b.getinning())
        a.setplayer(b.getplayer())
        a.setbatter(b.getbatter())
        a.setmatch(b.getmatch())
        a.setname(b.getname())
        a.setb1(b.getb1())
        a.setb2(b.getb2())
        a.setb3(b.getb3())
        a.setb4(b.getb4())
        a.setout(b.getout())
        a.sethitto(b.gethitto())
        a.setreplacement(b.getreplacement())
        a.setreplaceplace(b.getreplaceplace())
        a.setcount(b.getcount())
        a.setpitcher(b.getpitcher())
        print ListOfBatters


class MainFrame(wxFrame): # MAIN FRAME
   def __init__(self,parent,id,title):
      self.dirname=''
      wxFrame.__init__(self,parent,wxID_ANY, title, size=(500,600),
style=wxDEFAULT_FRAME_STYLE|
                           wxNO_FULL_REPAINT_ON_RESIZE)
      self.CreateStatusBar() # A Statusbar in the bottom of the window

      # Setting up the menu.
      filemenu= wxMenu()
      filemenu.Append(ID_ABOUT, "About"," Information about this program")
      filemenu.AppendSeparator()
      filemenu.Append(ID_EXIT,"Exit"," Terminate the program")
      # Creating the menubar.
      menuBar = wxMenuBar()
      menuBar.Append(filemenu,"File") # Adding the "filemenu" to the MenuBar
      self.SetMenuBar(menuBar)  # Adding the MenuBar to the Frame content.
      EVT_MENU(self, ID_ABOUT, self.OnAbout)
      EVT_MENU(self, ID_EXIT, self.OnExit)
          AB = Batter()

   def OnAbout(self,e):
      d= wxMessageDialog( self, " A system for scoring baseball \n"
                              " Author: P Brennan","BSS", wxOK)
                  # Create a message dialog box
      d.ShowModal() # Shows it
      d.Destroy() # finally destroy it when finished.

   def OnExit(self,e):
      self.Close(true)  # Close the frame.




class MainFrame2(wxFrame): # frame to select AB details
   def __init__(self,parent,id,title):
      wxFrame.__init__(self,parent,wxID_ANY, title,wxPoint(111,111), size=(400,400),
```

```
        style=wxDEFAULT_FRAME_STYLE|wxNO_FULL_REPAINT_ON_RESIZE)
        AB.sethitto((R,D))



        self.label1 = wxStaticText(self, -1, "SAFE", wxPoint(63,10))
        self.label2 = wxStaticText(self, -1, "OUT", wxPoint(280,10))


        #    Safe buttons
        self.single = wxButton(self, 1, "single", wxPoint(10,30), wxSize(60,-1))
        self.double = wxButton(self, 2, "double", wxPoint(10,60), wxSize(60,-1))
        self.triple = wxButton(self, 3, "triple", wxPoint(10,90), wxSize(60,-1))
        self.homrun = wxButton(self, 4, "  HR  ", wxPoint(10,120), wxSize(60,-1))
        self.insHR  = wxButton(self, 5, "Inside HR", wxPoint(10,150), wxSize(60,-1))
        self.error  = wxButton(self, 6, "error", wxPoint(10,180), wxSize(60,-1))
        self.fchit  = wxButton(self, 7, "FC", wxPoint(10,210), wxSize(60,-1))

        # out buttons
        self.grd    = wxButton(self, 8, "Ground out", wxPoint(270,30), wxSize(60,-1))
        self.fly    = wxButton(self, 9, "Fly out", wxPoint(270,60), wxSize(60,-1))
        self.pop    = wxButton(self, 10, "Pop out", wxPoint(270,90), wxSize(60,-1))
        self.foul   = wxButton(self, 11, "Foul out", wxPoint(270,120), wxSize(60,-1))
        self.sacf   = wxButton(self, 12, "Sac fly", wxPoint(270,150), wxSize(60,-1))
        self.sacb   = wxButton(self, 13, "Sac bunt", wxPoint(270,180), wxSize(60,-1))
        self.ifr    = wxButton(self, 14, " IFR ", wxPoint(270,210), wxSize(60,-1))
        self.outo   = wxButton(self, 15, "Other", wxPoint(270,240), wxSize(60,-1))

        EVT_BUTTON(self,1,self.singlep)
        EVT_BUTTON(self,2,self.doublep)
        EVT_BUTTON(self,3,self.triplep)
        EVT_BUTTON(self,4,self.homerunp)
        EVT_BUTTON(self,5,self.insidehr)
        EVT_BUTTON(self,6,self.errorp)
        EVT_BUTTON(self,7,self.fcp)
        EVT_BUTTON(self,8,self.groundout)
        EVT_BUTTON(self,9,self.flyout)
        EVT_BUTTON(self,10,self.popout)
        EVT_BUTTON(self,11,self.foulout)
        EVT_BUTTON(self,12,self.sacfly)
        EVT_BUTTON(self,13,self.sacbunt)
        EVT_BUTTON(self,14,self.infieldfly)
        EVT_BUTTON(self,15,self.otherp)


    def singlep(self,event): # single procedure

        AB.setb1("single")
        frame3 = MovePlayerFrame(None,-1,"Move players")
        frame3.Show(1)
        self.Close(True)

    def doublep(self,event): # double procedure
        AB.setb1("double")
        AB.setb2("double")
        frame3 = MovePlayerFrame(None,-1,"Move players")
        frame3.Show(1)
```

```
        self.Close(True)

    def triplep(self,event): # triple procedure
        AB.setb1("triple")
        AB.setb2("triple")
        AB.setb3("triple")
        frame3 = MovePlayerFrame(None,-1,"Move players")
        frame3.Show(1)
        self.Close(True)

    def homerunp(self,event): # hr procedure
        AB.setb1("HR")
        AB.setb2("HR")
        AB.setb3("HR")
        AB.setb4("HR")
        frame3 = MovePlayerFrame(None,-1,"Move players")
        frame3.Show(1)
        self.Close(True)
    def insidehr(self,event): # ihr procedure
        AB.setb1("IHR")
        AB.setb2("IHR")
        AB.setb3("IHR")
        AB.setb4("IHR")
        frame3 = MovePlayerFrame(None,-1,"Move players")
        frame3.Show(1)
        self.Close(True)
    def errorp(self,event): # procedure for error
        AB.setb1("ERR")
        frame3 = MovePlayerFrame(None,-1,"Move players")
        frame3.Show(1)
        self.Close(True)
    def fcp(self,event): # procedure for fielders choice
        AB.setb1("FC")
        frame3 = MovePlayerFrame(None,-1,"Move players")
        frame3.Show(1)
        self.Close(True)

    def groundout(self,event): # ground out initialisation
        frame5 = putoutframe(None,-1,"sacrifice")
        frame5.Show(1)
        self.Close()
    def flyout(self,event): # flyout initialisation
        AB.setout("fly out")
        frame4 = catchframe(None, -1,"fly-out")
        frame4.Show(1)
        self.Close(True)
    def popout(self,event): # popout initialisation
        AB.setout("fly out")
        frame4 = catchframe(None, -1,"pop-out")
        frame4.Show(1)
        self.Close(True)
    def foulout(self,event): # foul out initialisation
        AB.setout("fly out")
        frame4 = catchframe(None, -1,"foul-out")
        frame4.Show(1)
        self.Close(True)
    def sacfly(self,event): # sac fly initialisation
        AB.setout("sac-fly")
        AB.setb1("OUT")
```

```
            frame3 = MovePlayerFrame(None,-1,"Move players")
            frame3.Show(1)
            frame4 = catchframe(None, -1,"sacrifice")
            frame4.Show(1)
            self.Close(True)

    def sacbunt(self,event): # sac bunt initialisation
            AB.setb1("OUT")
            Ab.setout("sac-bunt")
            frame3 = MovePlayerFrame(None,-1,"Move players")
            frame3.Show(1)
            frame5 = putoutframe(None,-1,"sacrifice")
            frame5.Show(1)
            self.Close(True)
    def infieldfly(self,event): # IFR initialisations
            AB.setb1("IFR")
            self.Close(True)

class catchframe(wxFrame): # frame if batter is caught
    def __init__(self,parent,id,title):
        wxFrame.__init__(self,parent,wxID_ANY, title, wxPoint(200,200), size=(180,100),
style=wxDEFAULT_FRAME_STYLE|
                                wxNO_FULL_REPAINT_ON_RESIZE)
            self.label1 = wxStaticText(self, -1, "Enter catcher of ball and press OK", wxPoint(0,0))
            self.entry  = wxTextCtrl(self,1,"",wxPoint(80,15),wxSize(20,-1))
            self.okbutt = wxButton(self,2,"OK",wxPoint(70,40),wxSize(40,-1))
            EVT_BUTTON(self,2,self.OKI)

    def OKI(self,event): # writes to fiel
            global ListOfBatters
            AB.setout(AB.getout()+" "+self.entry.GetValue())
            Addout()
            clearstats(AB)
            ListOfBatters.append(AB)
            Swap(AB,ListOfBatters.pop(0))
            self.Close(True)


class putoutframe(wxFrame): # frame if player is out but not caught
    def __init__(self,parent,id,title):
        wxFrame.__init__(self,parent,wxID_ANY, title, wxPoint(200,200), size=(220,100),
style=wxDEFAULT_FRAME_STYLE|
                                wxNO_FULL_REPAINT_ON_RESIZE)
            self.label1 = wxStaticText(self, -1, "Enter combination of putout and press OK",
wxPoint(0,0))
            self.entry  = wxTextCtrl(self,1,"",wxPoint(80,15),wxSize(70,-1))
            self.okbutt = wxButton(self,2,"OK",wxPoint(95,40),wxSize(40,-1))
            EVT_BUTTON(self,2,self.OKI)

    def OKI(self,event):
            AB.setout(AB.getout()+" "+self.entry.GetValue())
            global ListOfBatter
            Addout()
            clearstats(AB)
            ListOfBatters.append(AB)
            Swap(AB,ListOfBatters.pop(0))
            self.Close(True)

class otherframe(wxFrame): # other play frame – for unusual plays
```

```
        def __init__(self,parent,id,title):
        wxFrame.__init__(self,parent,wxID_ANY, title, wxPoint(111,111), size=(300,300),
style=wxDEFAULT_FRAME_STYLE|
                        wxNO_FULL_REPAINT_ON_RESIZE)
#       text2 = wxStaticText(self, -1, AB.getname(), wxPoint(200,300), wxSize(75,-1),
wxALIGN_CENTRE)


                    self.label122 = wxStaticText(self, -1, "Select option", wxPoint(100,0), wxSize(200,-
1), wxALIGN_CENTRE)
                    self.label122.SetFont(wxFont(14, wxROMAN, wxNORMAL, wxBOLD))
                    self.a = wxButton(self, 1, "PH", wxPoint(40,40), wxSize(40,-1))
                    self.b = wxButton(self, 1, "EJC", wxPoint(80,40), wxSize(40,-1))
                    self.c = wxButton(self, 1, "IBB", wxPoint(120,40), wxSize(40,-1))
                    self.d = wxButton(self, 1, "Balk", wxPoint(160,40), wxSize(40,-1))
                    self.e = wxButton(self, 1, "Umpire Mask", wxPoint(0,80), wxSize(90,-1))
                    self.f = wxButton(self, 1, "Interference (B)", wxPoint(90,80), wxSize(90,-1))
                    self.g = wxButton(self, 1, "Interfernec (C)", wxPoint(180,80), wxSize(90,-1))
                    self.h = wxButton(self, 1, "Abonded BR", wxPoint(20,120), wxSize(70,-1))
                    self.i = wxButton(self, 1, "Bunt", wxPoint(90,120), wxSize(70,-1))
                    self.i = wxButton(self, 1, "Illegal bat", wxPoint(160,120), wxSize(70,-1))
                    self.j = wxButton(self, 1, "Bunt foul", wxPoint(10,160), wxSize(70,-1))
                    self.k = wxButton(self, 1, "Twice hit", wxPoint(90,160), wxSize(70,-1))
                    self.l = wxButton(self, 1, "Batter box foul", wxPoint(170,160), wxSize(90,-1))
                    self.m = wxButton(self, 1, "CANCEL", wxPoint(90,200))

                    EVT_BUTTON(self,1,self.stuff)
    def stuff(self,event):
            self.Close(True)

class MainFrame1(wxFrame): # frame to choose position where ball was hit
  def __init__(self,parent,id,title):
    wxFrame.__init__(self,parent,wxID_ANY, title, wxPoint(111,111), size=(400,400),
style=wxDEFAULT_FRAME_STYLE|
                        wxNO_FULL_REPAINT_ON_RESIZE)
        wxInitAllImageHandlers() # Make images available
        img = wxImage('pitch.bmp', wxBITMAP_TYPE_BMP) # load baseball pitch image
        self.bmp1 = img.ConvertToBitmap()    # loaded into
        self.pic = wxBitmapButton(self, 13, self.bmp1, wxPoint(60,440))
        EVT_BUTTON(self,13,self.GetPos)

  def GetPos(self,event): # gets position of mouse and saves it
        global R
        global D
        (R,D) = wxGetMousePosition()
        print('x co-ord: ', R)
        print('y co-ord: ', D)
        frame2 = MainFrame2(None,-1,"Action Details")
        frame2.Show(1)
        self.Close(True)
        self.count2 = 0


class Form2(wxPanel): #player details screen
  def __init__(self, parent, id):
    wxPanel.__init__(self, parent, id)
    # First create all labels and place in correct positions
        # TO DO - Make for loop to traverse a list getting text, position and size for all labels
        global homeplayerfile
```

```
        global awayplayerfile
        homeplayerfile.seek(0)
        awayplayerfile.seek(0)
        global AB
        global ManAt1
        global ManAt2
        global ManAt3
        global Blank
    global OnDeck
        global Bat1
        global Bat2
        global Bat3
        global Bat4
        global Bat5
        global Bat6
        global Bat7
        global ListOfBatters
        global Out
        global battera
        self.Strike=0
        self.Ball=0
        outs=0
        self.inning=0
        battera=0
        AB = Get_next()
    OnDeck = Get_next()
        Bat1 = Get_next()
        Bat2 = Get_next()
        Bat3 = Get_next()
        Bat4 = Get_next()
        Bat5 = Get_next()
        Bat6 = Get_next()
        Bat7 = Get_next()
        ListOfBatters = [OnDeck, Bat1, Bat2, Bat3, Bat4, Bat5, Bat6, Bat7]
        ManAt1 = Get_next1()
        ManAt2 = Get_next1()
        ManAt3 = Get_next1()
        Blank =  Get_next1()


        Pitcher = "J. Santana"
        self.label1 = wxStaticText(self, -1, "Strikes", wxPoint(63,345))
        self.label2 = wxStaticText(self, -1, "Balls",   wxPoint(65,385))
        self.label3 = wxStaticText(self, -1, "Outs", wxPoint(63,425))
        self.label4 = wxStaticText(self, -1, Pitcher, wxPoint(200,200), wxSize(60,-1),
wxALIGN_CENTRE)
        self.text2 = wxStaticText(self, -1, "", wxPoint(200,270), wxSize(75,-1),
wxALIGN_CENTRE)
        self.text2.SetFont(wxFont(12, wxROMAN, wxNORMAL, wxBOLD))
        self.Draw() # call function to draw necessaray players and buttons


        # Pictures for balls and strikes
        wxInitAllImageHandlers() # Make images available
        red = wxImage('red.bmp', wxBITMAP_TYPE_BMP) # load red image
        green = wxImage('green.bmp', wxBITMAP_TYPE_BMP) # load green image
        blue = wxImage('blue.bmp', wxBITMAP_TYPE_BMP) # load blue image
        self.bmp1 = red.ConvertToBitmap()    # loaded into
        self.bmp2 = red.ConvertToBitmap()    # seperate variables
```

```
        self.bmp3 = green.ConvertToBitmap()  # so that they can
    self.bmp4 = green.ConvertToBitmap()  # be manipulated
    self.bmp5 = green.ConvertToBitmap()  # seprately
        self.bmp7 = blue.ConvertToBitmap()
        self.bmp8 = blue.ConvertToBitmap()




        # Option buttons for each AB
        self.kS = wxButton(self,1,"KS",  wxPoint(155,320),  wxSize(30,-1))
        self.kL = wxButton(self,2,"KL",  wxPoint(185,320),  wxSize(30,-1))
        self.bb = wxButton(self,3,"BB",  wxPoint(215,320),  wxSize(30,-1))
        self.ff = wxButton(self,4,"FL",  wxPoint(245,320), wxSize(30,-1))
        self.ot = wxButton(self,5,"OTH", wxPoint(275,320), wxSize(30,-1))
        self.ip = wxButton(self,6,"In Play", wxPoint(195,345))
        self.refr = wxButton(self,99,"refresh positions", wxPoint(190,375))



        #change pitcher button
#       self.pit =wxButton(self,10,"Act", wxPoint(215,215),wxSize(30,-1))

        # Call functions when button is pressed

        EVT_BUTTON(self, 1, self.Ks)
        EVT_BUTTON(self, 2, self.Kl)
        EVT_BUTTON(self, 3, self.Bb)
        EVT_BUTTON(self, 4, self.Ff)
        EVT_BUTTON(self, 5, self.Ot)
        EVT_BUTTON(self, 6, self.Ip)
        EVT_BUTTON(self, 99, self.Draw1)

  def Draw1(self,event):
            self.Draw()
            self.Draw_Count()

  def Draw(self): # function to draw the buttons and labels as appropriate
        self.label5 = wxStaticText(self, -1, ManAt1.getname(), wxPoint(320,200), wxSize(60,-1),
wxALIGN_CENTRE)
        self.label6 = wxStaticText(self, -1, ManAt2.getname(), wxPoint(200,100), wxSize(60,-1),
wxALIGN_CENTRE)
        self.label7 = wxStaticText(self, -1, ManAt3.getname(), wxPoint(50,200), wxSize(60,-1),
wxALIGN_CENTRE)
        # label for AB player

        self.text2.SetLabel(AB.getname())

  def   Bb(self, event): # base on balls – free first base
          global ListOfBatters
          print("Ball")
          if self.Ball<3:
                  self.Ball=self.Ball+1
          else:
                  print("base on balls")
                  AB.setcount((self.Strike,self.Ball))
                  self.Strike=0
                  self.Ball=0
                  if ManAt1.getname() <>"":
```

```
                        if ManAt2.getname()<>"":
                            if ManAt3.getname()<>"":
                                ManAt3.setb4(str(AB.getbatter()))
                                Write_to_file(ManAt3)
                                clearstats(ManAt3)
                                ListOfBatters.append(ManAt3)
                                Swap(ManAt3,ManAt2)
                                ManAt3.setb3(str(AB.getbatter()))
                                Swap(ManAt2,ManAt1)
                                ManAt2.setb2(str(AB.getbatter()))
                                Swap(ManAt1,AB)
                                ManAt1.setb1("BB")
                            else:
                                Swap(ManAt3,ManAt2)
                                ManAt3.setb3(str(AB.getbatter()))
                                Swap(ManAt2,ManAt1)
                                ManAt2.setb2(str(AB.getbatter()))
                                Swap(ManAt1,AB)
                                ManAt1.setb1("BB")
                        else:
                                Swap(ManAt2,ManAt1)
                                ManAt2.setb2(str(AB.getbatter()))
                                Swap(ManAt1,AB)
                                ManAt1.setb1("BB")
                    else:
                                Swap(ManAt1,AB)
                                ManAt1.setb1("BB")

                    Swap(AB,ListOfBatters.pop(0))

                    self.Draw()
                    self.Clear_things()

            self.Draw_Count()

    def Ks(self, event): # strike out
            global outs
            global ListOfBatters
            print("Strike Swinging")
            if self.Strike<2:
                    self.Strike=self.Strike+1
            else:
                    print("out")
                    self.Clear_things()
                    AB.setcount((self.Strike,self.Ball))
                    AB.setout("KS")
                    Write_to_file(AB)
                    clearstats(AB)
                    ListOfBatters.append(AB)
                    Swap(AB,ListOfBatters.pop(0))
                    if outs == 2:
                            self.Clear_out()
                    else:
                            self.Draw()
                    Addout()

            self.Draw_Count()
```

```
    def Kl(self, event): # strike out
        global outs
        global ListOfBatters
        print("Strike Looking")
        if self.Strike<2:
                self.Strike=self.Strike+1
        else:
                print("out")
                self.Clear_things()
                AB.setcount((self.Strike,self.Ball))
                AB.setout("KL")
                Write_to_file(AB)
                clearstats(AB)
                ListOfBatters.append(AB)
                Swap(AB,ListOfBatters.pop(0))
                self.Draw()
                if outs == 2:
                        self.Clear_out()
                else:
                        self.Draw()


                Addout()

        self.Draw_Count()


    def Ff(self, event): # foul button
        print("Foul")
        if self.Strike<2:
                self.Strike=self.Strike+1
        self.Draw_Count()

    def Ot(self, event): # other frame
        print("Other")
        frame6 = otherframe(None, -1, "other")
        frame6.Show(1)
    def Ip(self, event): # frame to open when put in play
        print("In Play")
        self.Clear_things()

        frame1 = MainFrame1(None,-1,"Click Position")
        frame1.Show(1)

# next functions clear current things
    def Clear_out(self):
            self.bmp6 = wxNullBitmap
            self.out1 = wxStaticBitmap(self, -1, self.bmp6, wxPoint(60,440))
            self.out2 = wxStaticBitmap(self, -1, self.bmp6, wxPoint(80,440))


    def Clear_things(self):
            self.bmp6 = wxNullBitmap
        self.Strike=0
        self.Ball=0
            self.ball1 = wxStaticBitmap(self, -1, self.bmp6, wxPoint(50,400))
            self.ball2 = wxStaticBitmap(self, -1, self.bmp6, wxPoint(70,400))
            self.ball3 = wxStaticBitmap(self, -1, self.bmp6, wxPoint(90,400))
```

```
        self.strike1 = wxStaticBitmap(self, -1, self.bmp6, wxPoint(60,360))
        self.strike2 = wxStaticBitmap(self, -1, self.bmp6, wxPoint(80,360))


    def Draw_Count(self):
        global outs
        self.bmp3.SetWidth(0)
        self.bmp4.SetWidth(0)
        self.bmp5.SetWidth(0)
        self.bmp1.SetWidth(0)
        self.bmp2.SetWidth(0)
        self.bmp7.SetWidth(0)
        self.bmp8.SetWidth(0)


        print("strike: ",self.Strike)
        print("ball: ",self.Ball)
        print("outs: ",outs)

        if self.Strike>1:
            self.bmp2.SetWidth(14)
            self.bmp1.SetWidth(14)
      elif self.Strike>0:
            self.bmp1.SetWidth(14)
            self.bmp2.SetWidth(0)


        if self.Ball>2:
            self.bmp3.SetWidth(14)
            self.bmp4.SetWidth(14)
            self.bmp5.SetWidth(14)
        elif self.Ball>1:
            self.bmp3.SetWidth(14)
            self.bmp4.SetWidth(14)
            self.bmp5.SetWidth(0)
        elif self.Ball>0:
            self.bmp3.SetWidth(14)
            self.bmp4.SetWidth(0)
            self.bmp5.SetWidth(0)


        if outs>1:
            self.bmp7.SetWidth(14)
            self.bmp8.SetWidth(14)
        elif outs>0:
            self.bmp7.SetWidth(14)
            self.bmp8.SetWidth(0)

        self.ball1 = wxStaticBitmap(self, -1, self.bmp3, wxPoint(50,400))
        self.ball2 = wxStaticBitmap(self, -1, self.bmp4, wxPoint(70,400))
        self.ball3 = wxStaticBitmap(self, -1, self.bmp5, wxPoint(90,400))
        self.strike1 = wxStaticBitmap(self, -1, self.bmp1, wxPoint(60,360))
        self.strike2 = wxStaticBitmap(self, -1, self.bmp2, wxPoint(80,360))
        self.out1 = wxStaticBitmap(self, -1, self.bmp7, wxPoint(60,440))
        self.out2 = wxStaticBitmap(self, -1, self.bmp8, wxPoint(80,440))


class MovePlayerFrame(wxFrame): # Frame that moves the players round the bases
    def __init__(self,parent,id,title):
```

```
    wxFrame.__init__(self,parent,wxID_ANY, title,wxPoint(111,111), size=(400,400),

        style=wxDEFAULT_FRAME_STYLE|wxNO_FULL_REPAINT_ON_RESIZE)

        self.alabel1 = wxStaticText(self, -1, "Player", wxPoint(20,50))
        self.alabel1a= wxStaticText(self, -1, "Currently", wxPoint(100,50))
        self.alabel2 = wxStaticText(self, -1, "1B",wxPoint(150,50))
        self.alabel3 = wxStaticText(self, -1, "2B",wxPoint(200,50))
        self.alabel4 = wxStaticText(self, -1, "3B",wxPoint(250,50))
        self.alabel5 = wxStaticText(self, -1, "Home",wxPoint(300,50))
        self.alabel6 = wxStaticText(self, -1, AB.getname(), wxPoint(20,75))
        self.alabel7 = wxStaticText(self, -1, "", wxPoint(20,125))
        self.alabel8 = wxStaticText(self, -1, "", wxPoint(20,175))
        self.alabel9 = wxStaticText(self, -1, "", wxPoint(20,225))
        self.alabel10 = wxStaticText(self, -1, "home", wxPoint(100,75))
        self.alabel11 = wxStaticText(self, -1, "", wxPoint(100,125))
        self.alabel12 = wxStaticText(self, -1, "", wxPoint(100,175))
        self.alabel13 = wxStaticText(self, -1, "", wxPoint(100,225))
        self.ways = ["single", "double", "triple", "HR", "IHR", "ERR","FC","-","OUT"]
        self.abc1 = wxComboBox(self, -1, AB.getb1(), wxPoint(142,75), wxSize(55,-1), self.ways,
wxCB_DROPDOWN)
        self.abc2 = wxComboBox(self, -1, AB.getb2(), wxPoint(197,75), wxSize(55,-1), self.ways,
wxCB_DROPDOWN)
        self.abc3 = wxComboBox(self, -1, AB.getb3(), wxPoint(252,75), wxSize(55,-1), self.ways,
wxCB_DROPDOWN)
        self.abc4 = wxComboBox(self, -1, AB.getb4(), wxPoint(307,75), wxSize(55,-1), self.ways,
wxCB_DROPDOWN)
        self.alabel14 = wxStaticText(self, -1, "", wxPoint(150,125))
        self.alabel15 = wxStaticText(self, -1, "", wxPoint(150,175))
        self.alabel16 = wxStaticText(self, -1, "", wxPoint(200,175))
        self.alabel17 = wxStaticText(self, -1, "", wxPoint(150,225))
        self.alabel18 = wxStaticText(self, -1, "", wxPoint(200,225))
        self.alabel19 = wxStaticText(self, -1, "", wxPoint(250,225))
        self.alabel20 = wxStaticText(self, -1, "", wxPoint(250,175))
        self.alabel21 = wxStaticText(self, -1, "", wxPoint(150,125))
        self.alabel22 = wxStaticText(self, -1, "", wxPoint(200,125))
        self.alabel23 = wxStaticText(self, -1, "", wxPoint(150,175))
        self.alabel24 = wxStaticText(self, -1, "", wxPoint(200,175))
        self.alabel25 = wxStaticText(self, -1, "", wxPoint(250,175))
        self.alabel26 = wxStaticText(self, -1, "", wxPoint(250,125))
        self.default1 = str(AB.getbatter())
        self.default2 = "-"
        self.default3 = "-"




        self.okbutt  = wxButton(self,1,"OK",wxPoint(200,275), wxSize(50,50))
        EVT_BUTTON(self,1,self.Dostuff)

        self.thelist = [str(AB.getbatter()), "ERR", "FC","OUT","-"]

        if AB.getb1() == "ERR":
                self.default1 = "ERR"
        elif AB.getb1() == "OUT":
                self.default1 = str(AB.getbatter()-1)
        elif AB.getb1() <> "single":
                if AB.getb1() <> "FC":
                                self.default2 = str(AB.getbatter())
```

```
                                self.default3 = str(AB.getbatter())
                        else:
                                self.default2 = "FC"
                                self.default3 = "-"
                else:
                        self.default2 = "-"
                        self.default3 = "-"

                if ManAt1.getname() <> "":
                        self.alabel7.SetLabel(ManAt1.getname())
                        self.alabel11.SetLabel("1st base")
                        self.alabel14.SetLabel(ManAt1.getb1())
                        self.comb1   = wxComboBox(self, -1, self.default1, wxPoint(200, 125), wxSize(50,-
1),
                                self.thelist, wxCB_DROPDOWN)
                        self.comb2   = wxComboBox(self, -1, self.default2, wxPoint(250, 125), wxSize(50,-
1),
                                self.thelist, wxCB_DROPDOWN)
                        self.comb3   = wxComboBox(self, -1, self.default3, wxPoint(300, 125), wxSize(50,-
1),
                                self.thelist, wxCB_DROPDOWN)

                        if ManAt2.getname() <> "":
                                self.alabel8.SetLabel(ManAt2.getname())
                                self.alabel12.SetLabel("2nd Base")
                                self.alabel15.SetLabel(ManAt2.getb1())
                                self.alabel16.SetLabel(ManAt2.getb2())
                                self.comb4   = wxComboBox(self, -1, self.default1, wxPoint(250, 175),
                                        wxSize(50,-1), self.thelist, wxCB_DROPDOWN)
                                self.comb5   = wxComboBox(self, -1, self.default2, wxPoint(300, 175),
                                        wxSize(50,-1), self.thelist, wxCB_DROPDOWN)

                                if ManAt3.getname() <> "":
                                        self.alabel9.SetLabel(ManAt3.getname())
                                        self.alabel13.SetLabel("3rd Base")
                                        self.alabel17.SetLabel(ManAt3.getb1())
                                        self.alabel18.SetLabel(ManAt3.getb2())
                                        self.alabel19.SetLabel(ManAt3.getb3())
                                        self.comb6   = wxComboBox(self, -1, self.default1, wxPoint(300,
225),
                                                wxSize(50,-1), self.thelist,
wxCB_DROPDOWN)

                        elif ManAt3.getname() <> "":
                                        self.alabel8.SetLabel(ManAt3.getname())
                                        self.alabel12.SetLabel("3rd Base")
                                        self.alabel15.SetLabel(ManAt3.getb1())
                                        self.alabel16.SetLabel(ManAt3.getb2())
                                        self.alabel20.SetLabel(ManAt3.getb3())
                                        self.comb7   = wxComboBox(self, -1, self.default1, wxPoint(300,
175),
                                                wxSize(50,-1), self.thelist,
wxCB_DROPDOWN)

                elif ManAt2.getname() <> "":
                        self.alabel7.SetLabel(ManAt2.getname())
                        self.alabel11.SetLabel("2nd Base")
                        self.alabel21.SetLabel(ManAt2.getb1())
                        self.alabel22.SetLabel(ManAt2.getb2())
```

```
                self.comb8   = wxComboBox(self, -1, self.default1, wxPoint(250, 125), wxSize(50,-
1),
                                        self.thelist, wxCB_DROPDOWN)
                self.comb9   = wxComboBox(self, -1, self.default2, wxPoint(300, 125), wxSize(50,-
1),
                                        self.thelist, wxCB_DROPDOWN)
                if ManAt3.getname() <> "":
                        self.alabel8.SetLabel(ManAt3.getname())
                        self.alabel12.SetLabel("3rd Base")
                        self.alabel23.SetLabel(ManAt3.getb1())
                        self.alabel24.SetLabel(ManAt3.getb2())
                        self.alabel25.SetLabel(ManAt3.getb3())
                        self.comb10  = wxComboBox(self, -1, self.default1, wxPoint(300, 175),
                                        wxSize(50,-1), self.thelist, wxCB_DROPDOWN)

        elif ManAt3.getname() <> "":
                                self.alabel7.SetLabel(ManAt3.getname())
                                self.alabel11.SetLabel("3rd Base")
                                self.alabel21.SetLabel(ManAt3.getb1())
                                self.alabel22.SetLabel(ManAt3.getb2())
                                self.alabel26.SetLabel(ManAt3.getb3())
                                self.comb11  = wxComboBox(self, -1, self.default1, wxPoint(300,
125),
                                                wxSize(50,-1), self.thelist,
wxCB_DROPDOWN)
   def error (self):
        d = wxMessageDialog( self, "error - two men on the same base, please reselect","ERROR",
wxOK)
        d.ShowModal()
      d.Destroy()
        self.cont = 0

 def Dostuff(self, event): # moves playesr around the bases
          #  First set values of advancements

          global ListOfBatters
          AB.setb1(self.abc1.GetValue())
          AB.setb2(self.abc2.GetValue())
          AB.setb3(self.abc3.GetValue())
          AB.setb4(self.abc4.GetValue())
          if ManAt1.getname() <> "":
                ManAt1.setb2(self.comb1.GetValue())
                ManAt1.setb3(self.comb2.GetValue())
                ManAt1.setb4(self.comb3.GetValue())
                if ManAt2.getname() <> "":
                        ManAt2.setb3(self.comb4.GetValue())
                        ManAt2.setb4(self.comb5.GetValue())
                        if ManAt3.getname() <> "":
                                ManAt3.setb4(self.comb6.GetValue())
                elif ManAt3.getname() <> "":
                        ManAt3.setb4(self.comb7.GetValue())
          elif ManAt2.getname() <> "":
                ManAt2.setb3(self.comb8.GetValue())
                ManAt2.setb4(self.comb9.GetValue())
                if ManAt3.getname() <> "":
                        ManAt3.setb4(self.comb10.GetValue())
          elif ManAt3.getname() <> "":
                ManAt3.setb4(self.comb11.GetValue())
          self.cont = 1
```

```
            # Then check for double base runners
            if ManAt1.getname() <> "":
                    if AB.getb1() <> "OUT":
                            if (ManAt1.getb2() == "-") or (ManAt1.getb2() == ""):
                                    self.error()

        if ManAt2.getname() <> "":
                    if (ManAt2.getb3() == "-") or (ManAt2.getb3() == ""):
                            if ((AB.getb2() <> "-") and (AB.getb2() <>"")) or (((ManAt1.getb2() <>
"-") and (ManAt1.getb2() <> "")) and (ManAt1.getname() <>"")):
                                    self.error() # two men on 2nd
            if ManAt1.getname() <> "":
                    if (ManAt1.getb3() == "-") or (ManAt1.getb3() ==""):
                            if (AB.getb2() <> "-") and (AB.getb2() <> ""):
                                    self.error() # two men on 2nd
            if ManAt3.getname() <> "":
                    if (ManAt3.getb4 =="-") or (ManAt3.getb4() ==""):
                            if ((ManAt1.getb3() <> "") and (ManAt1.getb3() <> "-")) or ((AB.getb3()
<> "") and (AB.getb3() <> "-")) or ((ManAt2.getb3() <> "") and (ManAt2.getb3() <> "-")):
                                    self.error() # two men on 3rd
            if (ManAt2.getname()<>"") and ((ManAt2.getb4()=="-") or (ManAt2.getb4() == "")):
                            if ((ManAt1.getb3()<>"-") and (ManAt1.getb3()<>"")) or
((AB.getb3()<>"-") and (AB.getb3()<>"")):
                                    self.error() # two men on 3rd
            if (ManAt1.getname()<>"") and ((ManAt1.getb4()=="-") or (ManAt1.getb4() == "")):
                            if ((AB.getb3()<>"-") and (AB.getb3()<>"")):
                                    self.error() # two men on 3rd

            if self.cont == 0:
                    self.cont == 1
            else:
            # Then move the player onwards
                    if (ManAt3.getb4() <> "-") and (ManAt3.getb4() <> ""):
                            if ManAt3.getb4() <> "OUT":
                               Write_to_file(ManAt3)  # Add run to total
                               Addrun()
                               clearstats(ManAt3)
                               ListOfBatters.append(ManAt3)
                               Swap(ManAt3,Blank)
                       else:
                               Write_to_file(ManAt3)
                               clearstats(ManAt3)
                               ListOfBatters.append(ManAt3)
                               Swap(ManAt3,Blank)
                    if (ManAt2.getb3() <> "-") and (ManAt2.getb3() <> ""):
                       if (ManAt2.getb4() <> "-") and (ManAt2.getb4() <> ""):
                                    if ManAt2.getb4() <> "OUT":
                                       Write_to_file(ManAt2) # add run to total
                                       Addrun()
                                       clearstats(ManAt2)
                                       ListOfBatters.append(ManAt2)
                                       Swap(ManAt2,Blank)
                                else:
                                       Write_to_file(ManAt2)
                                       clearstats(ManAt2)
                                       ListOfBatters.append(ManAt2)
                                       Swap(ManAt2,Blank)
                            else:
                                    if ManAt2.getb3() <> "OUT":
```

```
                                    Swap(ManAt3,ManAt2)
                                    Swap(ManAt2,Blank)

            if (ManAt1.getb2() <> "-") and (ManAt1.getb2() <> ""):
                if (ManAt1.getb3() <> "-") and (ManAt1.getb3() <> ""):
                    if (ManAt1.getb4() <> "-") and (ManAt1.getb4() <> ""):
                        if ManAt1.getb4() <> "OUT":
                            Write_to_file(ManAt1) # add run to total
                            Addrun()
                            clearstats(ManAt1)
                            ListOfBatters.append(ManAt1)
                            Swap(ManAt1,Blank)
                        else:
                            Write_to_file(ManAt1)
                            clearstats(ManAt1)
                            ListOfBatters.append(ManAt1)
                            Swap(ManAt1,Blank)
                    else:
                        if ManAt1.getb3() <> "OUT":
                            Swap(ManAt3,ManAt1)
                            Swap(ManAt1,Blank)
                else:
                    if ManAt1.getb2() <> "OUT":
                        Swap(ManAt2,ManAt1)
                        Swap(ManAt1,Blank)

            if (AB.getb1() <> "-") and (AB.getb1() <> ""):
                if (AB.getb2() <> "-") and (AB.getb2() <> ""):
                    if (AB.getb3() <> "-") and (AB.getb3() <> ""):
                        if (AB.getb4() <> "-") and (AB.getb4() <> ""):

                            if AB.getb4() <> "OUT":
                                Write_to_file(AB) # add run to total
                                Addrun()
                                clearstats(AB)
                                ListOfBatters.append(AB)
                                Swap(AB,ListOfBatters.pop(0))
                            else:
                                Write_to_file(AB)
                                clearstats(AB)
                                ListOfBatters.append(AB)
                                Swap(AB,ListOfBatters.pop(0))
                        else:
                            if AB.getb3() <> "OUT":
                                Swap(ManAt3,AB)
                                Swap(AB,ListOfBatters.pop(0))
                    else:
                        if AB.getb2() <> "OUT":
                            Swap(ManAt2,AB)
                            Swap(AB,ListOfBatters.pop(0))
                else:
                    if AB.getb1() <> "OUT":
                        Swap(ManAt1,AB)
                        Swap(AB,ListOfBatters.pop(0))


            self.Close(True)
            self.alabel14.SetLabel("")
            self.alabel15.SetLabel("")
```

```
                self.alabel16.SetLabel("")
                self.alabel17.SetLabel("")
                self.alabel18.SetLabel("")
                self.alabel19.SetLabel("")
                self.alabel20.SetLabel("")
                self.alabel21.SetLabel("")
                self.alabel22.SetLabel("")
                self.alabel23.SetLabel("")
                self.alabel24.SetLabel("")
                self.alabel25.SetLabel("")
                self.alabel26.SetLabel("")
                self.abc1.SetValue("-")
                self.abc2.SetValue("-")
                self.abc3.SetValue("-")
                self.abc4.SetValue("-")

class FormUploadDownload(wxPanel): # upload and download screen
   def __init__(self, parent, id):
      wxPanel.__init__(self, parent, id)
         self.uploadbutton = wxButton(self,1,"UPLOAD", wxPoint(100,100))
         self.downloadbutton = wxButton(self,2,"DOWNLOAD", wxPoint(200,100))
         EVT_BUTTON(self,1,self.uploadstuff)
         EVT_BUTTON(self,2,self.downloadstuff)

   def uploadstuff(self,event):
            self.upload()
            print ("uploaded")


   def downloadstuff(self,event):
            self.download()
            print ("downloaded")

class Form1(wxPanel): # first user screen
   def __init__(self, parent, id):
      wxPanel.__init__(self, parent, id)
      self.title = wxStaticText(self, -1, "GAME OVERVIEW",wxPoint(150, 1), wxSize(300,-1))
         self.teamfile = open("teams","r")
         # Get list of teams
      self.teamList = self.teamfile.readlines()
         ### Need to remove newline char from end of each tuple.
         self.teamfile.close()
         self.marker = 0
      # Choosing the Home team
      self.label4home = wxStaticText(self,-1,"Home Team",wxPoint(45, 20))
      self.edithome = wxComboBox(self, -1, "", wxPoint(20, 40), wxSize(100,-1), self.teamList,
wxCB_DROPDOWN)

         # Choosing the Away team
      self.label4away = wxStaticText(self,-1,"Away Team",wxPoint(205, 20))
      self.editaway = wxComboBox(self, -1, "", wxPoint(180, 40), wxSize(100,-1), self.teamList,
wxCB_DROPDOWN)

         self.newgame = wxButton(self,2109,"Start game",wxPoint(120,90))

         EVT_BUTTON(self, 2109, self.newgameproc)

   def newgameproc(self,event):
         if self.edithome.GetValue() == "":
```

```
                d = wxMessageDialog( self, " Please choose a home team","ERROR", wxOK)
            d.ShowModal()
        d.Destroy()
        elif self.editaway.GetValue() == "":
                d = wxMessageDialog( self, " Please choose an away team","ERROR", wxOK)
            d.ShowModal()
        d.Destroy()
        elif self.editaway.GetValue() == self.edithome.GetValue() :
                d = wxMessageDialog( self, " Please choose two different teams","ERROR", wxOK)
            d.ShowModal()
        d.Destroy()

        elif self.marker == 0:
                print("new game")
                form2=Form2(nb, -1)
                nb.AddPage(form2,"Game details")
                self.marker = 1
                self.labelthis = wxStaticText(self,-1,"Game started, please input details on game
details tab", wxPoint(100,140))

        else:
                d = wxMessageDialog( self, "Game already started - Select from tabs","ERROR",
wxOK)
            d.ShowModal()
        d.Destroy()

   def EvtComboBox1(self, event):
      self.labelboxscore1.SetLabel('%s' % event.GetString())
   def EvtComboBox2(self, event):
      self.labelboxscore2.SetLabel('%s' % event.GetString())


  ## main function
app = wxPySimpleApp()
frame = MainFrame(None,-1,"BSS")
nb = wxNotebook(frame,-1)
form1=Form1(nb, -1)
form3=FormUploadDownload(nb, -1)
nb.AddPage(form1, "Game overview")
nb.AddPage(form3, "Upload / Download")
frame.Show(1)
app.MainLoop()
```

Appendix 22 – match details code for website

```
<html>
<html>
	<head>
		<title>
			BaseballSoftballUK.com BSUK game/player details
		</title>
	</head>
	<body bgcolor = "white">
	<?php print "Hello world"; ?>
<font color = "darkblue" size=7>
	<center>
		<script> <!default values>
		hometeam   = ""
		hshort = ""
		awayteam   = ""
		ashort = ""
		homescore  = 0
		awayscore  = 0
		h1 = 0
		h2 = 0
		h3 = 0
		h4 = 0
		h5 = 0
		h6 = 0
		h7 = 0
		h8 = 0
		h9 = 0
		a1 = 0
		a2 = 0
		a3 = 0
		a4 = 0
		a5 = 0
		a6 = 0
		a7 = 0
		a8 = 0
		a9 = 0
		hh = 0
		ah = 0
		he = 0
		ae = 0
		inning = 0
		batter = ""
		var plays = []
		var batlist = []
		var homelist = []
		var awaylist = []
		</script>
```

```
</font>
<! establish connection with DB and get values>
$link = mysql_connect("BSUK.com");
mysql_select_db("bss",$link);
$query = "SELECT * FROM players WHERE team_id = $home_id";
$result=mysql_query($query,$link);
homelist = $result;
$query = "SELECT * FROM players WHERE team_id = $away_id";
$result=mysql_query($query,$link);
awaylist = $result;



$query="COUNT (SELECT * FROM at_bat WHERE
matchnum=$match_id and inning_num=1.1 and b4 != ")";
$result=mysql_query($query,$link);
$query="SELECT B1 FROM at_bat WHERE inning_num=1.1";
$sec=mysql_query($query,$link);
h1 = "<script> inning= 1.1 plays = $sec
document.writeln($result)</script>";

$query="COUNT (SELECT * FROM at_bat WHERE
matchnum=$match_id and inning_num=2.1 and b4 != ")";
$result=mysql_query($query,$link);
$query="SELECT B1 FROM at_bat WHERE inning_num=2.1";
$sec=mysql_query($query,$link);
h1 = "<script> inning= 2.1 plays = $sec
document.writeln($result)</script>";

$query="COUNT (SELECT * FROM at_bat WHERE
matchnum=$match_id and inning_num=3.1 and b4 != ")";
$result=mysql_query($query,$link);
$query="SELECT B1 FROM at_bat WHERE inning_num=3.1";
$sec=mysql_query($query,$link);
h1 = "<script> inning= 3.1 plays = $sec
document.writeln($result)</script>";

$query="COUNT (SELECT * FROM at_bat WHERE
matchnum=$match_id and inning_num=4.1 and b4 != ")";
$result=mysql_query($query,$link);
$query="SELECT B1 FROM at_bat WHERE inning_num=4.1";
$sec=mysql_query($query,$link);
h1 = "<script> inning= 4.1 plays = $sec
document.writeln($result)</script>";

$query="COUNT (SELECT * FROM at_bat WHERE
matchnum=$match_id and inning_num=5.1 and b4 != ")";
$result=mysql_query($query,$link);
```

```
		$query="SELECT B1 FROM at_bat WHERE inning_num=5.1";
		$sec=mysql_query($query,$link);
		h1 = "<script> inning= 5.1 plays = $sec
document.writeln($result)</script>";

		$query="COUNT (SELECT * FROM at_bat WHERE
matchnum=$match_id and inning_num=6.1 and b4 != '')";
		$result=mysql_query($query,$link);
		$query="SELECT B1 FROM at_bat WHERE inning_num=6.1";
		$sec=mysql_query($query,$link);
		h1 = "<script> inning= 6.1 plays = $sec
document.writeln($result)</script>";

		$query="COUNT (SELECT * FROM at_bat WHERE
matchnum=$match_id and inning_num=7.1 and b4 != '')";
		$result=mysql_query($query,$link);
		$query="SELECT B1 FROM at_bat WHERE inning_num=7.1";
		$sec=mysql_query($query,$link);
		h1 = "<script> inning= 7.1 plays = $sec
document.writeln($result)</script>";

		$query="COUNT (SELECT * FROM at_bat WHERE
matchnum=$match_id and inning_num=8.1 and b4 != '')";
		$result=mysql_query($query,$link);
		$query="SELECT B1 FROM at_bat WHERE inning_num=8.1";
		$sec=mysql_query($query,$link);
		h1 = "<script> inning= 8.1 plays = $sec
document.writeln($result)</script>";

		$query="COUNT (SELECT * FROM at_bat WHERE
matchnum=$match_id and inning_num=9.1 and b4 != '')";
		$result=mysql_query($query,$link);
		$query="SELECT B1 FROM at_bat WHERE inning_num=9.1";
		$sec=mysql_query($query,$link);
		h1 = "<script> inning= 9.1 plays = $sec
document.writeln($result)</script>";


		$query="COUNT (SELECT * FROM at_bat WHERE
matchnum=$match_id and inning_num=1.2 and b4 != '')";
		$result=mysql_query($query,$link);
		$query="SELECT B1 FROM at_bat WHERE inning_num=1.2";
		$sec=mysql_query($query,$link);
		h1 = "<script> inning= 1.2 plays = $sec
document.writeln($result)</script>";

		$query="COUNT (SELECT * FROM at_bat WHERE
matchnum=$match_id and inning_num=2.2 and b4 != '')";
		$result=mysql_query($query,$link);
```

```
		$query="SELECT B1 FROM at_bat WHERE inning_num=2.2";
		$sec=mysql_query($query,$link);
		h1 = "<script> inning= 2.2 plays = $sec
document.writeln($result)</script>";

		$query="COUNT (SELECT * FROM at_bat WHERE
matchnum=$match_id and inning_num=3.2 and b4 != '')";
		$result=mysql_query($query,$link);
		$query="SELECT B1 FROM at_bat WHERE inning_num=3.2";
		$sec=mysql_query($query,$link);
		h1 = "<script> inning= 3.2 plays = $sec
document.writeln($result)</script>";

		$query="COUNT (SELECT * FROM at_bat WHERE
matchnum=$match_id and inning_num=4.2 and b4 != '')";
		$result=mysql_query($query,$link);
		$query="SELECT B1 FROM at_bat WHERE inning_num=4.2";
		$sec=mysql_query($query,$link);
		h1 = "<script> inning= 4.2 plays = $sec
document.writeln($result)</script>";

		$query="COUNT (SELECT * FROM at_bat WHERE
matchnum=$match_id and inning_num=5.2 and b4 != '')";
		$result=mysql_query($query,$link);
		$query="SELECT B1 FROM at_bat WHERE inning_num=5.2"
		$sec=mysql_query($query,$link);
		h1 = "<script> inning= 5.2 plays = $sec
document.writeln($result)</script>";

		$query="COUNT (SELECT * FROM at_bat WHERE
matchnum=$match_id and inning_num=6.2 and b4 != '')";
		$result=mysql_query($query,$link);
		$query="SELECT B1 FROM at_bat WHERE inning_num=6.2";
		$sec=mysql_query($query,$link);
		h1 = "<script> inning= 6.2 plays = $sec
document.writeln($result)</script>";

		$query="COUNT (SELECT * FROM at_bat WHERE
matchnum=$match_id and inning_num=7.2 and b4 != '')";
		$result=mysql_query($query,$link);
		$query="SELECT B1 FROM at_bat WHERE inning_num=7.2";
		$sec=mysql_query($query,$link);
		h1 = "<script> inning= 7.2 plays = $sec
document.writeln($result)</script>";

		$query="COUNT (SELECT * FROM at_bat WHERE
matchnum=$match_id and inning_num=8.2 and b4 != '')";
		$result=mysql_query($query,$link);
		$query="SELECT B1 FROM at_bat WHERE inning_num=8.2";
```

```
		$sec=mysql_query($query,$link);
		h1 = "<script> inning= 8.2 plays = $sec
document.writeln($result)</script>";

		$query="COUNT (SELECT * FROM at_bat WHERE
matchnum=$match_id and inning_num=9.2 and b4 != ")";
		$result=mysql_query($query,$link);
		$query="SELECT B1 FROM at_bat WHERE inning_num=9.2";
		$sec=mysql_query($query,$link);
		h1 = "<script> inning= 9.2 plays = $sec
document.writeln($result)</script>";


		<font size = 6 color = "darkblue">
		<script>
		document.writeln (hometeam+'  '+homescore+' -
'+awayscore+'  '+awayteam)
		</script>
		<table border="1" cellpadding="4" cellspacing="1">
			<tr>

	<th></th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th>

	<th>6</th><th>7</th><th>8</th><th>9</th><th>R</th><th>H</th><th>E</t
h>
			</tr>
		<script>
		document.writeln
('<tr><td>'+hshort+'</td><td>'+h1+'</td><td>'+h2+'</td><td>'+h3+'</td><td>'+h4+'
</td><td>'+h5+'</td><td>'+h6+'</td><td>'+h7+'</td><td>'+h8+'</td><td>'+h9+'</td
><td>'+homescore+'</td><td>'+hh+'</td><td>'+he+'</td></tr>')
		document.writeln
('<TR><TD>'+ashort+'</td><td>'+a1+'</td><td>'+a2+'</td><td>'+a3+'</td><td>'+a
4+'</td><td>'+a5+'</td><td>'+a6+'</td><td>'+a7+'</td><td>'+a8+'</td><td>'+a9+'<
/td><td>'+awayscore+'</td><td>'+ah+'</td><td>'+ae+'</td></tr></table>')

		</script>
	</center>
	<br>
	<table border="1" width="98%" height="68%">
	<tr><td width = "20%" align = center>
		<script>
		document.writeln ('<font size =6>Inning
'+inning+'<br><hr><br><font size=4>')
		var count = 0
		while (plays[count] != undefined) {
			document.writeln (plays[count]+'<br><hr
width="30%"><br>');
			count ++;
```

```
		}
	</script>
	</td>
	<td align = center><br><br><br>
	<script>
	document.writeln('<form name="select player">')
	document.writeln('<select name ="home" size ="1">')
	document.writeln('<option value="">home players</option>')
	var count1 = 0
	while (homelist[count1] != undefined) {

		document.writeln ('<option
value="">'+homelist[count1]+'</option>');
		count1 ++;
		}
	document.writeln('</select>')
	document.writeln('<select name ="away" size ="1">')
	document.writeln('<option value="">away players</option>')
	var count2 = 0
	while (homelist[count2] != undefined) {

		document.writeln ('<option
value="">'+awaylist[count2]+'</option>');
		count2 ++;
		}
	document.writeln('</select><br><br>')
	document.writeln('<font size =6>'+batter+'<br>')
	document.writeln('<table border="1" width="98%"
height="60%"><tr><td align=center>')

	var count3 = 0
	while (batlist[count3] != undefined) {
		document.writeln (batlist[count3]+'<br><hr width ="30%">');
		count3 ++;
		}
	</script>
	</td>
	<td width="50%" align=center>
		<br>
		<img src ="pitch.bmp" height="80%" width="80%">
	</td></tr>
	</table>
	</td></tr>
	</table>


</body>
</html>
```

Appendix 23 – Code for getting player details on website

```
<html>
<html>
	<head>
		<title>
			BaseballSoftballUK.com BSUK game/player details
		</title>
	</head>
<body bgcolor = "white">
<font color = "darkblue" size=7>
	<center><!default values>
		<script>
			name = ""
			team = ""
			average = 0.0
			hits = 0
			rbi = 0
			hr = 0
			b2 = 0
			b3 = 0
			bb = 0
			ibb = 0
			sb = 0
			cs = 0
			slg = 0.0
			obp = 0.0
			ab = 0
			document.writeln (name+'  -  '+team+"<BR><BR>")
			</script>
			$link = mysql_connect("BSUK");
			mysql_select_db("bss",$link);
			$query="select forename,surname from players where
player_id=$player_id"
			$result=mysql_query($query,$link);
			name = $result;

			$query="select a.name, a.location from team a, players b where
b.player_id=$player_id and a.team_id=b.team_id"
			$result=mysql_query($query,$link);
			team = $result;

			$query="(count (select * from at_bat WHERE (b1 != '') and (player_id
= $player_id)))"
			$result=mysql_query($query,$link);
			hits = $result;

			$query="COUNT (SELECT * FROM at_bat WHERE b4 =
$player_id"
			$result=mysql_query($query,$link);
			rbi	= $result;

			$query="COUNT (SELECT * FROM at_bat WHERE b4 = 'HR' and
player_id = $player_id)"
			$result=mysql_query($query,$link);
			hr = $result;

			$query="COUNT (SELECT * FROM at_bat WHERE b2 = 'double'
and player_id = $player_id)"
```

```
                        $result=mysql_query($query,$link);
                        b2 = $result;

                        $query="COUNT (SELECT * FROM at_bat WHERE b3 = 'triple' and
player_id = $player_id)"
                        $result=mysql_query($query,$link);
                        b3 = $result;

                        $query="COUNT (SELECT * FROM at_bat WHERE b1 = 'BB' and
player_id = $player_id)"
                        $result=mysql_query($query,$link);
                        bb = $result;

                        $query="COUNT (SELECT * FROM at_bat WHERE b1 = 'IBB' and
player_id = $player_id)"
                        $result=mysql_query($query,$link);
                        ibb  = $result;

                        $query="COUNT (SELECT * FROM at_bat WHERE (b2 = 'SB' or
b3='SB' OR b4='SB') and (player_id = $player_id))"
                        $result=mysql_query($query,$link);
                        sb   = $result;

                        $query="COUNT (SELECT * FROM at_bat WHERE out = 'CS' and
(player_id = $player_id))"
                        $result=mysql_query($query,$link);
                        cs   = $result;


                        $query="COUNT (select * FROM at_bat WHERE player_id =
$player_id) - bb - ibb"
                        $result=mysql_query($query,$link);
                        ab   = $result;

                        obp = (hits + bb + ibb) / (ab + bb + ibb)
                        slugging = (hits + b2 + b3 + b3 + hr + hr + hr + hr) /ab
                        average = hits/AB


                </font>
                <font size = 6 color = "darkblue">
                <script>
                        document.writeln ('AVG : '+average+"<BR>")
                        document.writeln ('HR : '+hr+'<br>')
                        document.writeln ('RBI : '+rbi+'<BR><BR><BR>')
                </script>
                </font>
                <font size = 4 color = "darkblue">
                <table border="0" cellpadding="5" cellspacing="6" text="darkblue">
                <script>
                document.writeln('<tr><td align =right>Doubles : '+b2
+'</td><td></td><td></td><td></td><td align =right>Slugging : '+slg+'</td></tr>')
                document.writeln('<tr><td align =right>Triples : '+b3
+'</td><td></td><td></td><td></td><td align =right>Stolen Bases : '+sb +'</td></tr>')
                document.writeln('<tr><td align =right>On base : '+obp
+'</td><td></td><td></td><td></td><td align =right>Caught stealing: '+cs +'</td></tr>')
                document.writeln('<tr><td align =right>   Hits : '+hits
+'</td><td></td><td></td><td></td><td align =right>Walks : '+bb+'</td></tr></table>')
                </script>
```

```
</body>
</html>
```