

1 - Executive Summary	2
2 – Implementation Plan and Status	5
Implementation Plan	5
Implementation Status	6
3 – Updated System Architecture	8
Updated Use Case Diagram	8
Updated Class Diagram	8
Test Cases	8
4 – Full Security and Survivability Plan	23
5 – Drafts of Technical and User Manuals	24
User Manual	24
Technical Manual	25
6 – Results of User Testing	29
7 – Project History and Lessons Learned	30
8 - Project Management	33
Time Accounting	33
Feasibility Update	35
QA Manager’s Report	36
Project Metrics	37
Risk Management	42
Problem Analysis	47
9 – Appendix	48

1 - Executive Summary

I. Project Development

Phase 4 is the continuation of implementing our system as well as preparing for the final presentation. We have “golded” all entities and implemented all A level and most B level use cases. Another focus of this phase was the user testing results, which gave us a better idea of how the potential users would feel about using our system. Using the information from user testing, we have made significant improvements on user-friendliness and the design of site template. All the progress we have made in this phase has better prepared us for the completion the final product.

a. System Updates

- i. Architecture and Design - The architecture and design of Athena system have been solidified and the site has been developed according to the architecture and design. Therefore, there are no updates.
- ii. Test Cases - There are no updates.
- iii. Interface - For the first draft of the design, we decided to design the site to be user-centered with easy navigation. We started by using tables in a two-column by three-row design in templates. We had three basic templates---faculty based, student based and log-in. The colors were in shades of red and white, with black text. In our second draft, we changed the colors to different shades of blue. The design altered to have a header and footer, with a two-table page design. The simple design starts with a log-in page, and then based on whether the user is a student or faculty the website shows a side-menu with the user-oriented links. Depending on the user, the navigation stays consistent throughout each page the user goes through the site. Through user testing, it was concluded users liked the navigation and header of the first design and the page layout in the second design. In conclusion, we will be combining these aspects into the final design.

b. Entities – A table of entities golded and silvered in the phase

Entities	Status
PPS	Gold
UCPE	Gold

c. Use cases – A table of use cases golded and silvered in this phase

Use Cases	Status
Create Account	Gold
Create Project	Gold
Create Timelog entry	Gold
Delete Project	Gold
Modify Account	Gold
Modify Project	Gold
Modify PPS	Gold
Modify Timelog Entry	Gold
Select Project	Gold
View Alerts	Gold
View PPS	Gold
View UCPE	Gold
Modify UCPE	Silver
Get Help	Silver

d. User Testing

During this phase, we initialized our first series of user-testing with a number of different users ranging from Information Systems professors to Information Systems undergraduates. This portion of our phase is extremely important because we will utilize the users' feedback to better Project Athena. Because Project Athena is not completely finished, there were portions that were not presentable for our users. However, we wanted to begin testing as soon as possible in order to make the necessary changes while completing the system. With these preliminary comments, we will be able to reference them, improve what is needed, and hopefully test the same users again to see if they are satisfied with the changes. We would like as much feedback and comments as possible, therefore we felt it was necessary to begin user-testing regardless of Project Athena's completion.

e. Lessons Learned

The attached document (section 7) is a compilation of the lessons we have learned about people, process, and technology. Specific examples from the semester are given. By doing so, we can clearly identify the origin of the lesson and how it has helped the team grow during the semester. The record of lessons learned show that much of our mistakes were found during the beginning of the semester due. However, we were able to learn from our mistakes and increase our productivity and strengthen our relationships.

II. Project Management

Task Analysis – For the final phase, we are expected to work for 252 hours (3 weeks, 7 members, 12 hours/week) in average. The actual work hours according to the implementation plan will be 197 hours, which does not account for the time lost during carnival – 42 hours (half efficiency). During phase 5, we will spend 31% of the time on documentation, 23% on quality assurance, 30% on implementation, 6% on the 2nd round of user testing, and 10% on the final presentation preparation. We have allocated a lot more time on technical quality assurance (30%) since we need to further refine our site before the final evaluation. In addition, hours for the final presentation and user testing have been included into our tasks.

a. *Quality Assurance* –

- i. *Cost of Quality* – Cost of Quality required about 24.3% of the 239 total hours placed into this phase. In order to assure quality in our tasks, we utilized different methodologies for defect prevention, detection, and correction. To implement defect prevention, a number of tasks were divided amongst team members. For detection, we integrated individual parts and ran through the use cases to see if appropriate results displayed. Corrections required the team to submit their work to the TQA who had to review and verify the correctness of the work. Flaws that arose from combining different parts were fixed by code review sessions and conferring with Professor Heimann. Constantly checking our work with Professor Heimann, the original coder and the TQA, we were able to correct our work until we reached near production status on some use cases.
- ii. *Time Accounting* - In terms of time accounting, our time and effort for this phase was on par with our estimates. As a team, we put in close to 79 hours per week or approximately 239 hours for the entire phase. Much of our work this time around focused primarily finishing up the remaining use cases, refining previous “ungolded” use cases, and most importantly, integration and design refinement. According to weekly meeting and implementation status reports, 88% of our documentation tasks met both personal and course deadlines. Though our punctuality with documentation has decreased, we were able to significantly raise our implementation percentage from previous phases. In our subsequent phases, we will work to

implement measures that allow us to bring this metric, for both set of tasks, as close to 100 percent as possible.

- iii. *Technical Quality* – In order to assure a high level of technical quality, we reviewed the system as a whole and how individual use cases functioned inside of the system. We utilized the system to generate test data inside of our phpMyAdmin database. By doing so, we compared the expected outputs with the database data. We also verified error messages and expected redirected. If we encountered any inconsistencies, we would repeat testing and repair the code on the spot. Specifically, we would check the code that pertained to the functions that gave incorrect results and use a series of error messages to pinpoint the mistake
 - b. *Risk Analysis* – “Integration difficulty” has become the most frequently signed off risk because the focus of this phase was to integrate all completed use cases and finish building the system as much as possible. In fact, it has become the most time consuming task during this phase. Despite the frequency of integration problems, only abatement strategies were needed for the majority of the occurrences. “Lack of programming skills” and “Conflicts of schedule” were signed off frequently as well – they both have been risks that we experience and have been abated most of the time. New risks that were signed off were “Feature Creep” and “Difficult Adaptation by User.” One level C use case was actually dropped (mitigation) due to the increasing hours spent on integration. “Getting criticisms from the results of user testing” was added as a warning sign in “Difficult Adaptation.” It would trigger the abatement strategies and possible lead to the use of mitigation strategies, such as adding more help functionalities in the system.
 - c. *Problem Status Update* – “Emergency and sickness” problem was reopened during phase 4 because team members were missing meetings because of them. “Schedule Conflict” has remained a problem even though the team has learned to cope with it better by this phase. The biggest problem we had during phase 4 was “integration difficulty,” which took away a lot of our time from completing our milestone. “Limited data collection” was closed because we were set with our data collection and put all the effort into implementing accordingly. “Lack of programming skills” will still be a potential problem since not many team members can help out with site integration. Finally, “Outside Distraction” is open because of the incoming Carnival week.

2 – Implementation Plan and Status

It is estimated that a student works for an average of 12 hours a week. Since there are 7 team members and 3 weeks to complete phase 5, our team should ideally work for approximately **252 hours** by the phase 4 deadline. The total estimated phase 5 work hours for is **197 hours**; this concludes **62 hours (31%) of documentation, 46 hours (23%) of total quality assurance hours, 59 hours (30%) of implementation, 20 hours (10%) to prepare for the final presentation, and 10 hours (6%) of round 2 user testing**. However, this total hours does not include the time we lose during the week of carnival, which will be **42 hours lost (half efficiency)**. The overall hours with hours lost will be $197 + 42 = 239$ hours. This means that we expect to work for **11.4 hours per week**.

The 11.4 hours work hours per week seems like a more relaxed schedule than that in phase 4 (11.76 hours/week/person). But since this phase is the very last phase to complete our system, we need to account enough time for any last moment problems – final project presentation, integration, code freeze, data loss, database crash...etc.

Implementation Plan

Phase 5: (Due 5/5)					
Total Estimated Phase 5 Hours	197				
Total Estimated Documentation Hours	62	31%			
Total Estimated QA Hours	46	23%			
Documentation QA	6				
Technical QA/Testing	40				
Total Implementation Hours	59	30%			
Total Hours For Final Presentation	20	10%			
Total Hours For Round 2 User Testing	10	6%			
Executive Summary	2	Steph, Eric	5/2	5/3	5/4
Implementation Status	2	Will	5/2	5/3	5/4
Test Cases and Defect Report	4	Will	4/24	4/25	4/26
Final Usability Report	4	Karin	4/24	4/25	4/26
Project Management					
Time Accounting	2	Jamil	5/2	5/3	5/4
Feasibility Update	1	Jamil	4/25	4/26	4/27
QA Manager's Report	3	Rob	5/2	5/3	5/4
Project Metrics	2	Jamil	5/2	5/3	5/4
Risk Management	1	Eric	5/2	5/3	5/4
Problem Status	1	Eric	5/2	5/3	5/4
Project History / Lessons Learned					
Project Description	4	Steph, Eric	4/24	4/25	4/26
Overview of Implementation	3	Eric	4/27	4/28	4/29
Lessons Learned Analysis	4	Steph	4/24	4/25	4/26
Final Assessment	4	Karin	4/27	4/28	4/29
Individual Statement of Accomplishment	7	Everyone	4/24	4/25	4/26
Archival materials					

Copies of all files to preserve system	4	Rob	5/1	5/2	5/3
A Read Me document	4	Karin	5/1	5/2	5/3
Final Versions of User Documentation	3	Steph	4/15	4/16	4/17
Final Versions of Technical Documentation	3	Jamil	4/15	4/16	4/17
Electronic copies of project History/Lessons Learned	4	Karin	4/30	5/1	5/2
Implementation Stage 3					
Export UCPE	12	Rob	4/17	4/18	4/19
View Projected Time	12	Karin	4/24	4/25	4/27
View Student Submissions	12	Karin	4/17	4/18	4/19
View Statistics	15	Rob	4/24	4/25	4/27
View Statistics – Learning PHP SWF classes	8	Rob, Jamil	4/24	4/25	4/27

Implementation Status

Entities	Implementation Status
Account	Gold
Project	Gold
Timelog Entry	Gold
Student	Gold
PPS	Gold
UCPE	Gold

Use Case	Implementation Status
Login	Gold
Logout	Gold
View Accounts	Gold
View Projects	Gold
View Timelogs	Gold
Create Timelog Entry	Gold
View PPS	Gold
View UCPE	Gold
Modify Timelog Entry	Gold
Create Account	Gold
Create Project	Gold
Delete Project	Gold
Modify PPS	Gold
Modify Project	Gold
Select Project	Gold
Modify Account	Gold
View Alerts	Gold
Change Password	Silver
Get Help	Silver
Modify Current UCPE	Silver
Resend Password	Silver
View Projected Time	Not Started

View Statistics	Not Started
View Student Submissions	Not Started
Export UCPE	Not Started

3 – Updated System Architecture

Updated Use Case Diagram

The overall Use Case Diagram remains unchanged from last phase.

Updated Class Diagram

The overall Class Diagram remains unchanged from last phase.

Test Cases

Change Password

Test Case	Category	Test Data	Expected Result
Blank password	Null Input	<<empty field>>	Error Message
Verify matching passwords	Proper Processing	New Password: larry Re-enter new password: james	Error Message
Verify that old password matches the account	Security Violations	Old Password: qwerty	Error Message
Access page without being logged in	Security Violations	Type the URL of specific pages into the browser	Error Message stating must be logged in first.
Password is not 4 to 15 characters long	Proper Processing	A Cat AlanJamilEricWill	Error Message stating password is too short or too long
Proper Cancellation	Proper Processing	Click on the "Cancel" button	Return to previous page
Changing Password	Proper Processing	<<all data correct>>	Confirmation Page; password changed

Create Account

Test Case	Category	Test Data	Expected Result
Invalid Input Email	Invalid Input	mjordan@,cmu,edu larry larry@bird larry@.com	Error Message
Invalid First Name	Invalid input	Fred is #2 James & Larry Beth.	Error Message
Invalid Last Name	Invalid Input	Mary82 Kate-()&	Error Message
Blank First Name	Null Input	<<empty field>>	Error Message

Blank Last Name	Null Input	<<empty field>>	Error Message
Blank Email	Null Input	<<empty field>>	Error Message
Blank Password	Null Input	<<empty field>>	Error Message
Duplicate Email Address	Proper Processing	Create account with email: fred@cmu.edu . Create another account with email: fred@cmu.edu	Error Message stating email already belongs to someone else.
Access page without being logged in	Security Violations	Type the URL of specific pages into the browser	Error Message stating must be logged in first.
Blank Andrew ID	Null Input	<<empty field>>	Error Message
Invalid Andrew ID	Proper Processing	&^*^&*&@ Jame()2*	Error Message
Creating Account	Proper Processing	<<all data correct>>	Confirmation Page; account created
Access whiled logged in as student	Security Violations	Log in as student, type URL into browser	Error

Create Project

Test Case	Category	Test Data	Expected Result
Invalid Time Period	Proper Processing	End time: 10:00AM Date: 10/23/2005 Start time: 11:00AM Date: 10/23/2005	Error Message
Blank Start Time	Null Input	<<empty field>>	Error Message
Blank End Time	Null Input	<<empty field>>	Error Message
Blank Project Name	Null Input	<<empty field>>	Error Message
Blank Start Date Entry	Null Input	<<empty field>>	Error Message
Blank End Date Entry	Null Input	<<empty field>>	Error Message
Invalid Project Name	Invalid Input	~`!@#\$%^*()+= "/<>	Error Message
Invalid Beginning Date Entry	Invalid Input	Larry Two/01/2006 Feb 26, 2005	Error Message
Invalid End Date Entry	Invalid Input	fred 01/two/2006 january 26 2006	Error Message
Access page without being logged in	Security Violations	Type the URL of specific pages into the browser	Error Message stating must be logged in first.
Faculty Only Functionalities	Security Violations	Log in as student and test to see if student can add or edit a project.	Students should not be able to see the functions to add or edit project.
Proper Cancellation	Proper Processing	Click on the "Cancel" button	Return to previous page
Creating Project	Proper Processing	<<all data correct>>	Confirmation Page; project created

Create Time Log Entry

Test Case	Category	Test Data	Expected Result
Blank Activity Name	Null Input	<<empty field>>	Error Message
Blank Category Name	Null Input	<<empty field>>	Error Message
Blank Start Date	Null Input	<<empty field>>	Error Message
Blank End Date	Null Input	<<empty field>>	Error Message
Blank Interruption Time	Null input	<<empty field>>	Error Message
Blank Start Time	Null Input	<<empty field>>	Error Message
Blank End Time	Null Input	<<empty field>>	Error Message
Dates Valid Invalid Time	Proper Processing	Start time: 11:00AM Start Date: 10/25/2005 End time: 10:00AM Start Date: 10/25/2005	Error Message
Time Valid Invalid Dates	Proper Processing	Start Date: 10/25/2006 End Date: 10/24/2006	Error Message
Invalid Date Entry	Invalid Input	Larry Two/01/2006 Feb 26, 2005	Error Message
Access page without being logged in	Security Violations	Type the URL of specific pages into the browser	Error Message stating must be logged in first.
Proper Cancellation	Proper Processing	Click on the "Cancel" button	Return to previous page
Time Log Entry end time past project deadline	Proper Processing	Project End Date: 01/05/2006 Time Log End Date: 01/07/2006	Error Message
Time Log Submission past project deadline	Proper Processing	Project End Date: 01/05/2006 Time Log Submission Date: 01/07/2006	Error Message
Creating Time Log Entry	Proper Processing	<<all data correct>>	Confirmation Page; Time Log created
Interruption time is smaller than the difference of the times	Proper Processing	Start time: 10:00:00 Date: 10/25/2005 End time: 11:00:00 Date: 10/25/2005 Interruption Time : 100	Error Message

Delete Project

Test Case	Category	Test Data	Expected Result
Access page without being logged in	Security Violations	Type the URL of specific pages into the browser	Error Message stating must be logged in first.
Invalid Project Deletion	Proper Processing	Delete a project when students' projects exist.	Error Message stating that you cannot delete if students' projects already exist.
No project selected	Proper Processing	Do not choose a project to delete	Error Message
Proper Cancellation	Proper Processing	Click on the "Cancel" button	Return to previous page
Proper Deletion	Proper Processing	<<click delete button>>	Confirmation Page; project deleted
Access while logged in as student	Security Violations	Log in as student, type URL into browser	Error

Export UCPE

Test Case	Category	Test Data	Expected Result
Access page without being logged in	Security Violations	Type the URL of specific pages into the browser	Error Message stating must be logged in first.
No project selected	Proper Processing	Type the URL for UCPE export page	Error Message

Get Help

Test Case	Category	Test Data	Expected Result
Access page without being logged in	Security Violations	Type the URL of specific pages into the browser	Error Message stating must be logged in first.
Correct Help popup	Proper Processing	Click on the help link	Correct help message

Login

Test Case	Category	Test Data	Expected Result
Invalid Input Email	Invalid Input	mjordan@andrew.cmu.edu larry larry@bird larry@.com	Error Message
Blank Email	Null Input	<<empty field>>	Error Message
Blank Password	Null Input	<<empty field>>	Error Message
Account Does Not Exist	Proper Processing	Email: fred@cmu.edu Pw: fred <<valid email and valid password>>	Error Message
Verify Student Account	Proper Processing	Login with a student account.	Student accounts should have student options
Verify Administrator Account	Proper Processing	Login in with an admin account.	Admin accounts should have administrative options.

Logout

Test Case	Category	Test Data	Expected Result
Verify session is destroyed	Proper Processing	Logout	User cannot access members web pages
User is not logged into the system	Proper Processing	Type URL of "Logout" page	Error Message

Modify Account

Test Case	Category	Test Data	Expected Result
Invalid Input Email	Invalid Input	mjordan@andrew.cmu.edu larry larry@bird larry@.com	Error Message
Blank Email	Null Input	<<empty field>>	Error Message
Blank Password	Null Input	<<empty field>>	Error Message
Access page without being logged in	Security Violations	Type the URL of specific pages into the browser	Error Message stating must be logged in first.
Proper Cancellation	Proper Processing	Click on the "Cancel" button	Return to previous page
Blank Andrew ID	Null Input	<<empty field>>	Error Message
Invalid Andrew ID	Proper Processing	&^*^&*&@ Jame()2*	Error Message
Modifying Account	Proper Processing	<<all data correct>>	Confirmation Page; account modified
Access while logged in as student	Security Violations	Log in as student, type URL into browser	Error Message
Access with an non-existing account	Security Violations	Type Url in the browser without being logged in	Error Message

Modify Current UCPE

Test Case	Category	Test Data	Expected Result
Blank UCPE Form Fields	Null Input	<<Leave some UCPE form fields empty>>	Error Message
Invalid Quantity for Actors	Invalid Input	1.5 A -1	Error Message
Invalid Quantity for Actors Use Cases	Invalid Input	22.2 * -29	Error Message
Blank Actor Fields	Null Input	Leave at least 1 actor field blank	Error Message
Blank Use Cases Fields	Null Input	Leave at least 1 use cases field blank	Error Message
Access page without being logged in	Security Violations	Type the URL of specific pages into the browser	Error Message stating must be logged in first.
Unable to modify UCPE in past projects	Proper Processing	Go to a past project and try to edit the UCPE	Student should be unable to edit past items.
Invalid Environmental and Technical Values	Proper Processing	1a -38 One 6 4.9999999	Error Message
Proper Cancellation	Proper Processing	Click on the "Cancel" button	Return to previous page
Modifying UCPE	Proper Processing	<<all data correct>>	Confirmation Page; UCPE modified
Access invalid UCPE via URL	Proper Processing	Type in URL of UCPE with different ID	Error Message

Modify PPS

Test Case	Category	Test Data	Expected Result
Blank Total Minutes	Null Input	<<empty field>>	Error Message
Unable to modify PPS in past projects	Proper Processing	Go to a past project and try to edit PPS.	Student should be unable to edit past items.
Invalid "Lines Of Codes"	Invalid Input	A -2 3j	Error Message
Invalid "Time in Phase"	Invalid Input	Asd -234 *(-=)2-	Error Message

Invalid input for PPS form fields	Proper processing	Aek #\$(#)	Error Message
Time entered in "Max" or "Estimate" field is less than time entered in "Min" field. Time "Max" is less than "Estimate"	Proper processing	Min: 200 Estimate: 175 Max: 150 Max: 150 Estimate: 125	Error Message
Blank "Lines of Codes"	Null Input	Leave at least one field blank	Error Message
Blank "Program Complexity"	Null Input	Leave at least one field blank	Error Message
Blank "Time in Phase"	Null Input	Leave at least one field blank	Error Message
Access page without being logged in	Security Violations	Type the URL of specific pages into the browser	Error Message stating must be logged in first.
Proper Cancellation	Proper Processing	Click on the "Cancel" button	Return to previous page
Modifying PPS	Proper Processing	<<all data correct>>	Confirmation Page; PPS modified
Access invalid PPS via URL	Proper Processing	Type in URL of PPS with different ID	Error Message

Modify Project

Test Case	Category	Test Data	Expected Result
Invalid Time Entry	Proper Processing	Start time: 11:00AM End time: 10:00AM	Error Message
Blank Time Entry	Null Input	<<empty field>>	Error Message
Invalid Date Entry	Proper Processing	Start Date: 03/02/2006 End Date: 03/01/2006	Error Message
Blank Date Entry	Null Input	<<empty field>>	Error Message
Access page without being logged in	Security Violations	Type the URL of specific pages into the browser	Error Message stating must be logged in first.
Faculty Only Functionalities	Security Violations	Log in as student and test to see if student can edit a project.	Students should not be able to see the functions edit project.
Cannot Modify Project	Proper Processing	Try to edit a field in modify page	Faculty should not be able to modify once project has started.

Invalid Name Entry	Proper Processing	# % % ^ & % ^ # \$ %	Error Message
Blank Name Entry	Proper Processing	< < empty field > >	Error Message
Proper Cancellation	Proper Processing	Click on the "Cancel" button	Return to previous page
Modifying Project	Proper Processing	< < all data correct > >	Confirmation Page; project modified

Modify Time Log Entry

Test Case	Category	Test Data	Expected Result
Invalid Time Period	Proper Processing	End time: 10:00AM Start time: 11:00AM	Error Message
Duplicate times for Time Log entries	Proper Processing	Create a Time Log entry with: Start time: 10:00AM End time 12:00PM Second Time Log entry with: Start time: 10:00AM End Time: 11:00PM	Error Message
Dates Valid Invalid Time	Proper Processing	Start time: 11:00AM Start Date: 10/25/2005 End time: 10:00AM Start Date: 10/25/2005	Error Message
Time Valid Invalid Dates	Proper Processing	Start Date: 10/25/2006 End Date: 10/24/2006	Error Message
Time Log Entry end time past project deadline	Proper Processing	Project End Date: 01/05/2006 Time Log End Date: 01/07/2006	Error Message
Time Log Submission past project deadline	Proper Processing	Project End Date: 01/05/2006 Time Log Submission Date: 01/07/2006	Error Message
Invalid "Interruption" Time	Invalid Input	Any character that isn't a number.	Error Message
Blank "Interruption Time"	Null Input	<<empty field>>	Error Message
Blank Date Entry	Null Input	<<empty field>>	Error Message
Access Page Without Being Logged In	Security Violations	Type the URL of specific pages into the browser	Error Message stating must be logged in first.
Unable to Modify Items in Past Projects	Proper Processing	Go to a past project and try to edit the Time Log entries.	Student should be unable to edit past items.
Invalid Date Entry	Invalid Input	March March 10, -06/02/2005	Error Message
Proper Cancellation	Proper Processing	Click on the "Cancel" button	Return to previous page
Interruption time is smaller than the difference of the times	Proper Processing	Start time: 10:00:00 Date: 10/25/2005 End time: 11:00:00 Date: 10/25/2005 Interruption Time: 100	Error Message

Resend Password

Test Case	Category	Test Data	Expected Result
Invalid Email	Proper Processing	james@.cmu.edu larry@andrew3	Error Message
Email does not exist in database	Proper Processing	fred@cmu.edu	Error Message stating that password cannot be sent
Proper Cancellation	Proper Processing	Click on the "Cancel" button	Return to previous page
Proper Sending	Proper Processing	< < all data correct > >	Password Sent

Select Project

Test Case	Category	Test Data	Expected Result
Access Page Without Being Logged In	Security Violations	Type the URL of specific pages into the browser	Error Message stating must be logged in first.
Correct List of Projects	Proper Processing	Go to Select Project view and make sure all students projects are there	All the projects should be there.

View Accounts

Test Case	Category	Test Data	Expected Result
Access Page Without Being Logged In	Security Violations	Type the URL of specific pages into the browser	Error Message stating must be logged in first.
Display correct Accounts	Proper Processing	Go to the Accounts view Page	Accounts views are displayed correctly

View Alerts

Test Case	Category	Test Data	Expected Result
Access Page Without Being Logged In	Security Violations	Type the URL of specific pages into the browser	Error Message stating must be logged in first.
Display correct alerts	Proper Processing	Go to the Alerts view Page	Alerts views are displayed correctly

View PPS

Test Case	Category	Test Data	Expected Result
Access Page Without Being Logged In	Security Violations	Type the URL of specific pages into the browser	Error Message stating must be logged in first.
Display correct PPS	Proper Processing	Go to the PPS view Page	PPS views are displayed correctly
Access page without selecting project	Proper Processing	Type URL of "View PPS" page	Redirects to "Select Project" page
Access Page Without Complete UCPE	Proper Processing	User selects link without completing UCPEs	Redirect to "Modify UCPE" page

View Projected Time

Test Case	Category	Test Data	Expected Result
Access Page Without Being Logged In	Security Violations	Type the URL of specific pages into the browser	Error Message stating must be logged in first.
Display correct projected time	Proper Processing	Go to the project time view Page	Projected time views are displayed correctly

View Projects

Test Case	Category	Test Data	Expected Result
Access Page Without Being Logged In	Security Violations	Type the URL of specific pages into the browser	Error Message stating must be logged in first.
Display correct projects	Proper Processing	Go to the Projects view Page	Projects views are displayed correctly

View Statistics

Test Case	Category	Test Data	Expected Result
Access Page Without Being Logged In	Security Violations	Type the URL of specific pages into the browser	Error Message stating must be logged in first.
Display correct Statistics	Proper Processing	Go to the Statistics view Page	Statistics views are displayed correctly
Faculty Only Functionalities	Security Violations	Log in as student and test to see if student can have access to other students statistics	Students should not be able to see the statistics of other students
No x-axis parameters specified	Proper Processing	Click the "Show Stats" button	Error Message
No y-axis parameters specified	Proper Processing	Click the "Show Stats" button	Error Message

View Student Submissions

Test Case	Category	Test Data	Expected Result
Access Page Without Being Logged In	Security Violations	Type the URL of specific pages into the browser	Error Message stating must be logged in first.
Display correct student submissions	Proper Processing	Go to the student submissions view page	Student submissions views are displayed correctly
Faculty Only Functionalities	Security Violations	Log in as student and test to see if the student can view other students submissions	Students should not be able to see other students' submissions.
No student selected (faculty)	Null Input	Click on the "Display" button	Error Message

View Time Logs

Test Case	Category	Test Data	Expected Result
Access Page Without Being Logged In	Security Violations	Type the URL of specific pages into the browser	Error Message stating must be logged in first.
Display correct Time Logs	Proper Processing	Go to the "Time Logs View" Page	Time Logs views are displayed correctly
Access Page Without Complete UCPE	Proper Processing	User selects link without completing UCPEs	Redirect to "Modify UCPE" page
Access Page Without Complete PPS	Proper Processing	User selects link without completing PPSs	Redirect to "Modify PPS" page

View UCPE

Test Case	Category	Test Data	Expected Result
Display correct UCPE	Proper Processing	Go to the UCPE View Page	UCPE Views are displayed correctly
Access Page Without Being Logged In	Security Violations	Type the URL of specific pages into the browser	Error Message stating must be logged in first.
Access page without selecting project	Proper Processing	Type URL of "View PPS" page	Redirects to "Select Project" page

4 – Full Security and Survivability Plan

Security is an important feature in the survivability of a system. Below is a table of possible scenarios that could lead to system failures and its preventions. Major concerns will be addressed by the team in the coming Phase. While our system does not contain particularly sensitive data, site-wide and consistent validation will be a critical component of a complete system.

Security and Privacy Risks	Stakeholder	Risk-reducing practices (programming)	Risk-reducing practices (non-programming)	Cost
URL attacks	Student	Check admin status on all web pages	Computing ethics education to the students	Time
Database crash	Administrator	Validation to prevent bad data entries	Back up database data daily or possibly even more frequently	Disk Space
SQL string injection	Student	Avoid dynamically generated SQL	Tutorial/help pages that could help an inexperienced database administrator manage the system database	Time
System Administrator Errors/accidents	Administrator	Show confirmation page for major data modification or deletion	Tutorial/help pages that could help an inexperienced database administrator manage the system database	Time

5 – Drafts of Technical and User Manuals

A hardcopy of the manuals can be found in the Appendix.

User Manual

(Incomplete due to emergency)

Introduction:

Welcome to Project Athena! The following help pages are designed to guide you. Project Athena is the system that will allow you to submit your Project Plan Summaries and Use Case Point Estimates from the comfort of your own room. Also, you are able to submit time logs more conveniently! With all of this information, the faculty and staff members of the Information Systems major will be able to view how you are doing and you will not be inconvenienced with any paperwork!

The purpose of the Project Plan Summaries, Use Case Point Estimates, and the timelogs is to give you, the faculty, and staff members a better idea of how you're doing in comparison to your classmates and to keep track of your own progress.

Steps to Using Athena:

1. Need to create a UCPE
2. Need to create a PPS
3. Then capable of creating time logs

Definitions and Terms:

UCPE (Use Case Point Estimates): An aggregation of the number of actors and use cases and the environmental and technical factors that are calculated to give you an estimate for how many hours the project is predicted to take.

PPS (Project Plan Summary): A chart of the predicted number of lines of code and the number of minutes the project will take along with the actual number of lines of code and the number of minutes for the project. By keeping track, we hope that you will gain a better sense of your own programming skills and can time yourself better.

Time Log: A record of all the time you placed into your project. We also keep track of your "interruption time" to see how much time was actually allocated towards your project. The purpose of this is, again, to give you a better perspective of how much time you take for each project throughout the semester.

Statistics:

"Easy to install": Managing an installation in which users' technical skills are not alike; irrelevant to web-based applications (hint: put down a 1)

Technical Manual

Table of Contents:

1 Overview of System

- 1.1 Purpose
- 1.2 Functionality
- 1.3 Unimplemented but Suggested Functionality

2 Use Case Descriptions

3 Test Case Descriptions

4 ER Diagram

5 Data Models

6 Data Dictionary

- 6.1 Account Table
- 6.2 AcctProj Table
- 6.3 Category Table
- 6.4 Entry Table
- 6.5 PPS Table
- 6.6 Project Table
- 6.7 UCPE Table

7 Site Map

8 Key Design Decisions

9 Known Defects

1 - Introduction

1.1 - We have designed and implemented an information system for the analysis and integration of PPS and Use Case Point Estimate Worksheets. The stakeholders are the Information Systems faculty and students of the IS course 67-272.

Through interviews of current 272 faculty, we learned that the current process of processing every 272 project requires hours of manual ordering, data entry, and excel manipulation. Instructors are pressed for time and can only perform the most basic data analyses. Our system fundamentally addresses the way PPS worksheets, time logs, and Use Case Point Estimates are handled.

1.2 - The system is:

- *Educational* – 272 students will be able to complete an online web form for their PPS worksheets, time logs, and Use Case Point estimates. A dedicated and comprehensive help system will help students understand the individual components of good project planning and they will be able to see projections of future project times, so that they can better allocate their time for future projects.
- *Integrated* – 272 students, faculty and administration will be able to go to one site and be able to login and out, access grades, paste worksheets, and view descriptive statistics related to a given project through one interface. To ensure this, our project will be built around the current 272 website.
- *Time-saving* – 272 faculty will be able to electronically and graphically view detailed and specific statistics of submitted forms as well as create their own Use Case Point estimates. 272 students will also no longer need to make the physical walk to public printing and then to Porter Hall.

1.3 - Below is a list of suggested use cases we had initially planned but have yet to be implemented as they were not critical.

Use Case	Description
Export UCPE	This use case allows a student to export their Use Case Point Estimates (UCPE) as a CSV or Comma Delimited File which can be opened by Microsoft Excel format.
View Projected Time	This use case allows a student or faculty member to view the projected time required for the current project based on a calculation of past results. "Projected Time": A mathematical calculation of actual time/Use Case Point based on past timelogs and applied to a UCPE for the current project.

2 - Use Case Descriptions

Please refer to previous phases.

3 - Test Cases

Please refer to 'Test Cases' section of phase 3.

4 - Class Diagram

Please refer to the Class Diagram in Phase 3.

5 - Data Models

Please refer to the Data Model in Phase 3.

6 - Data Dictionary

Please refer to previous phases.

7 - Site Map

Student	Faculty
Login	Login
UCPE	Account
<i>Create UCPE</i>	<i>Create Account</i>
<i>Modify UCPE</i>	<i>Modify Account</i>
PPS (only available after UCPE is created)	<i>Disable Account</i>
<i>Create PPS</i>	Project
<i>Modify PPS</i>	<i>Create Project</i>
Time Log (only available after UCPE and PPS are created)	<i>Modify Project</i>
<i>Create Time Log Entry</i>	<i>Cancel Project</i>
<i>Delete Time Log Entry</i>	<i>Delete Project</i>
<i>Modify Time Log Entry</i>	View Statistics
View Statistics	Student Submissions
View Projected Time	<i>View PPS Submissions</i>
Get Help	<i>View UCPE Submissions</i>
Change Password	<i>View Time log Submissions</i>
Resend Password	View Alerts
Logout	Logout

8 - Key Design Decisions

When designing our system, we chose to adopt the highly recommended Model View Control or MVC design pattern. MVC is a model that enforces the separation between the input, processing, and output of an application. To this end, our application is divided into three core components; the model, the view, and the controller, each of which handles a discreet set of tasks.

The view is the interface or HTML the user sees and interacts with. The model on the other hand, represents our data and internal logic. It's where most of the processing takes place. Our database falls under the model. The data returned by the model is display neutral, meaning that the model applies no formatting. This way, a single model can provide data for any number of display interfaces. This reduces code duplication, because model code is only written once and is then reused by all of the views. Lastly, the controller interprets any requests from the user and calls portions of the model and view when necessary to fulfill the request. Hence, when the user clicks a link or submits an HTML form, the controller itself doesn't output anything or perform any real processing. It takes the request and determines which model components to invoke and which formatting to apply to the resulting data

There are many benefits from using MVC, many of which prompted our design choice.

- First, and possibly most important, is the ability to have multiple views that rely upon a single model. Because the model returns data without applying any formatting, the same components can be used and called for use with any interface. Also, code duplication is limited because we've separated the data and business logic from the display.
- As the model is self contained and separate from the controller and the view, it's much less painful to change our data layer or logic. For example, if we switched databases, say from MySQL to Oracle, we need only alter our model. If written correctly, the view won't care at all whether that list of users came from a database or an LDAP server. The three parts of an MVC application are black boxes whose inner workings are hidden from the other portions. It makes you build well-defined interfaces and self-contained components.
- The concept of the controller is also a benefit. The controller is essentially used to stitch together different pieces of the model and the view to fulfill a request. This places significant power in our hands. Presented with a number of reusable building blocks in the model and the view, the controller picks and chooses which blocks are needed to handle specific processing and display requirements.

When coding in languages such as PHP, as we have opted, the initial temptation is to mix data layer code like database queries with display code such as HTML. The MVC design pattern enforces the separation of these portions of an application. While it surely required extra effort up front on our part to implement, the payoffs proved significant and useful.

9 - Known Defects

Though we have certain defects currently open, we will provide a final list of any unresolved defects once the semester and project nears completion.

6 – Results of User Testing

In the course of creating a system, the developers need to analyze and test the system for any flaws in user interface and in the system itself. By having potential users run through all the use cases and experience the system, developers gain the knowledge to fix flaws that can develop into greater problems in the future. By implementing the best of the tester suggestions we will enhance usability and overall quality of the system.

A copy of the first round of user test results will be included in the appendix for reference. Below is a compilation and summary of user tests.

Strengths

- Alternating colors in views for rows increases readability
- Fluid process of PPS->timelog->PPS
- UCPE validation working when values outside 0-5
- Website look and feel is good. Navigation is easy and understandable.

Suggestions

General

- Switch “Delete” and “Modify” icons
- Put error messages in more appropriate places (next to fields)
- The header should have the course name. Also the index should describe what the project is about.
- Project header- the Project Drop down list should be next to the label.
- Time what format? Minutes? Hours? Time should be in am/pm format. Boxes for time (drop down lists)

Timelog

- Move categories for timelog entry
- Validation for timelog entries to check that they occur during the project
- Units are unclear in form entry
- Activity name-use JS to auto select category based on regex code-words (code, study...)
- Timelog should use datagrids. UCPE and PPS shouldn't use the datagrid unless you partition it right (technical and environmental stuff).
- View timelog shrink to eliminate horizontal scroll bar

UCPE

- UCPE-show how it is multiplying for total
- Modify UCPE-some fields not being auto filled. So when trying submit blank fields can't submit
- For create UCPE, show a range of values for input and an explanation of what it is.
- UCPE breaks down to 3 groups

PPS

- PPS actual LOC do when? You don't have this until end of project
- PPS LOC should be on bottom of page
- PPS layout confusing, format not matching other documents (horizontal vs vertical)

Overall, our testers had assigned us an average score of 4.4. They were confused over the formatting for time entry (am/pm). They also had some notable concerns about general formatting (spacing, colors). They felt that the site would be much more usable if it included better instructions and guidelines on how to proceed. Descriptions of the functions would aid in usability as well. By moving around different elements of different forms usability would be enhanced.

7 – Project History and Lessons Learned

- I. Project History
 - a. Description of system
 - i. Purpose: The purpose of Athena is to provide undergraduate students in the Information Systems major with a system that will allow them to conveniently submit worksheets necessary for their 67-272 course. Rather than manually submitting worksheets to Ms. Blazeovich, Athena will permit students to submit the worksheets online.
 - ii. Major Stakeholders: The major stakeholders are the Information System students currently registered for 67-272 and the professors and staff members involved in the Information Systems major.
 - iii. Intended Users: The intended users are the 67-272 students, professors, and staff members.
 - iv. Benefits: The major benefit of this system is convenience. The system will conveniently allow students to submit worksheets from the comfort of their own rooms. Also, the system will provide comfort for the staff members and faculty by neatly storing and organizing the information in a database. All users will be able to recall the information and be able to get a clear view of how they are performing in regards to other students.
 - v. Summary of capabilities the system provides to meet the needs of the users: The system provides a database for the worksheets to be stored in and it also returns statistics in which students, faculty, and staff members can utilize.
- II. Project Data and Metrics
 - a. Overview of the implementation -
 - b. Process and project data -
 - c. Details of the design, implementation and process for students to understand the tools that were used
 - d. Project Metrics summary:

The tables below show the percent of documentation and project coding tasks completed on time. This phase, 88% of documentation tasks were completed on time, down from 95% in last phase. However, the percentage of use cases completed on time rose by 33%, a significant improvement from last phase.

The team has become more efficient at documentation production, but the difference in this phase was the presence of use cases that were also due, that took precedence over documentation concerns. In order to meet project implementation deadlines, some documentation tasks were placed at a lower priority. The difference can be seen in the following two tables.

Lessons learned in project implementation are largely due to the team becoming more accustomed to coding in a group environment.

Project Documentation

Phases	Total Number of Tasks	Tasks Completed on Time	Percentage completed on time
Phase 2	34	30	88%
Phase 3	19	18	95%
Phase 4	17	15	88%

Project Implementation:

Phases	Total Number of Tasks	Tasks Completed on Time	Percentage completed on time
Phase 3	15	8	53%
Phase 4	14	12	86%

III. Lessons Learned Analysis

- a. The lessons that were learned (needs to be realistic, appropriate, thoughtful, and consequential)
 - i. Teamwork: This experience has taught the entire team the benefits and difficulties in working with a group made up of seven different members for an entire semester. We learned how to adjust to different personalities and work schedules in order to maximize the productivity of the group. Most of the group was unfamiliar with the different members of the group, so this was a new experience in which we had to break barriers and be confident enough to bring our best to the team.
 - ii. Communication: One of the most important lessons we learned during this semester is communication. Because the group is made up of seven different members with various lifestyles, we had to quickly establish ways in which to communicate with each other. Not only was it important to find out different means of communication but we also had to learn ways in which we could easily communicate personal feelings with each other. It is difficult to confront members that were unfamiliar with each other, but it was an obstacle and important lesson that had to be learned in order to benefit the progress and relationship of the team members.
 - iii. Planning: In order to guarantee the most success for this project, we had to make sure that the project was correctly and well planned. Planning and organization of the system and project were the most important strategies to prevent any mistakes from occurring in the future. We tried to be very particular with details in the system and predict any other functionalities and changes that need to be made. In the end, we understood the importance of planning everything ahead.
 - iv. User friendliness: Throughout this experience, we always understood that we had to keep the users in mind throughout the entire process. In the end, the system will be used by the public- our users. With our user testing, surveys, and storyboards, we were able to make the best adjustments that would cater to our users. It was important to learn that the public will not completely understand what we want from the system. We need to learn what the users want from the system. What functionalities are the most important to them? What design will best guide the users through the website? All these questions needed to be answered because the system will be utilized by the users and not ourselves.

IV. Quality Assessment

- a. Final assessment of overall quality
 - i. Teamwork and process: The quality of our teamwork continuously increased throughout the semester. The teamwork was not as strong in the beginning of the semester due to the unfamiliarity amongst team members. However, as the semester progressed, the members became more comfortable and the quality of our teamwork increased. Teamwork and process came hand in hand. As our teamwork increased, the process became more successful, boosting the quality in

- our process. There was not as much delay in how long every task took and the quality of each task improved throughout the semester.
- ii. Technology: The quality in our technology improved throughout the semester. In the beginning of the semester, our skills assessments established the notion that the entire team did not have a strong foundation in technical and programming skills.
- b. Quality of team's process, communications, management, attention to quality and ultimate deliverables
 - i. As mentioned earlier, the quality of the team's process, communications, management, and such improved throughout the semester. We were able to learn valuable lessons from each phase that was due. With those lessons, we were able to act upon reoccurring situations, which improved our productivity and efficiency.
 - c. What have you really learned this term?
 - i. We learned the challenges of working with different types of people. Every team member brought a different trait to the table. Each member had different obligations, schedules, priorities, and skills. We had to figure ways in which we could integrate this course into our different life styles.
 - d. What challenges did you face?
 - i. The hardest challenge we faced in our system was integrating the use cases. Many use cases were added, modified, or deleted because we gained more information about the scope of our project and received more feedback from our intended users.
 - e. How did you solve these challenges?
 - i. We thoroughly planned out our system with several diagrams and consulting with our faculty advisor. We also conducted user testing to gain a better sense of what our users were comfortable and uncomfortable with.
 - f. What lessons would you pass on to future IS project teams?
 - i. The lessons that we would pass on to future IS project teams are to quickly establish relationships with your other members and to securely integrate use cases. With both lessons learned, every IS team will be able to be more proficient with their skills and time. These obstacles are all from how team members interact from the beginning. If they are open with each other and willing to learn from one another, they can accelerate their progress and improve their relationships.

8 - Project Management

Time Accounting

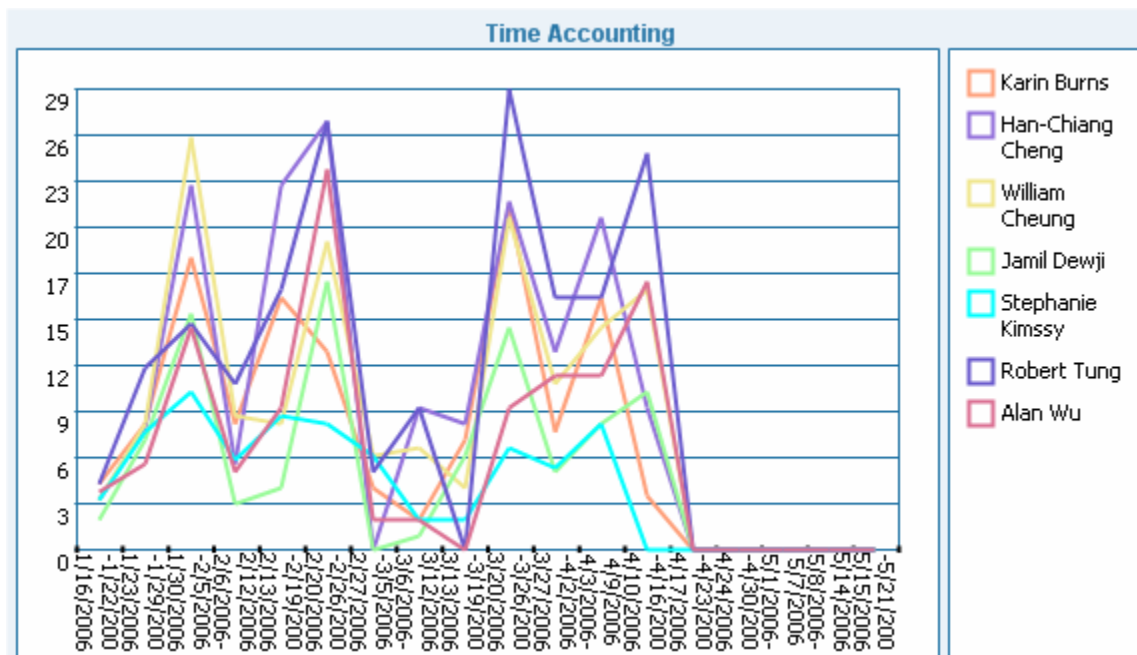
As in the last phase, each section was assigned to certain people in the team. The tasks assigned to each person were worked on both independently and or with other group members. For Phase 4, the project was broken down and assigned as follows.

Documentation Tasks	Team Member
Executive Summary	Steph, Eric
Implementation Status	Eric, Jamil
Updated Architecture and Design Documents	Alan, Karin, Rob, Will
<i>Final set of Test Cases</i>	Alan, Will
Security and Survivability Plan	Will
Hardcopy of User Manual	Stephanie
Hardcopy of Technical Manual	Jamil
Result of User Tests	Rob
Draft of Project History / Lessons Learned	Eric, Jamil, Rob, Stephanie
<i>Introduction</i>	Eric, Stephanie
<i>Project Data and Metrics</i>	Eric, Jamil
<i>Lesson Learned Analysis</i>	Stephanie
Quality Assessment	Rob, Stephanie
Project Management	Eric, Jamil, Stephanie
<i>Time Accounting</i>	Jamil
<i>Feasibility Update</i>	Jamil
<i>QA Manager's Report</i>	Stephanie
<i>Project Metrics</i>	Jamil
<i>Risk Management</i>	Eric
<i>Problem Status</i>	Eric

Implementation Tasks	Team Member
Create Account	Eric
Create Project	Alan
Create Time log Entry	Karin
Delete Project	Alan
Modify Account	Eric
Modify Current UCPE	Stephanie
Modify PPS	Will
Modify Project	Alan
Modify Time log Entry	Karin
Select Project	Jamil
Get Help	Will
View Alerts	Karin
View Statistics	Rob
View Projected Time	Jamil

Much of the phase was completed during its initial week so that we would have first drafts which could be reviewed and corrected accordingly before compilation. As a result, much of our time was taken having our work looked over and revised. In terms of tasks, a majority of our effort was focused on implementing and testing our uses cases rather than documentation as it held a greater priority. A breakdown of individual effort for all completed phases thus far can be seen below:

Date	Karin Burns	Han-Chiang Cheng	William Cheung	Jamil Dewji	Stephanie Kimssy	Robert Tung	Alan Wu	Total
1.16 - 1.22	4.25	3.25	3.25	2	3.25	4.25	3.75	24
1.23 - 1.29	8	7.75	8	7	7.5	11.5	5.5	55.25
1.30 - 2.05	18.5	23	26	15	10	14.25	14	120.75
2.06 - 2.12	8	5.5	8.5	3	5.75	10.5	5	46.25
2.13 - 2.19	16	23	8	4	8.5	16.5	9	85
2.20 - 2.26	12.5	27	19.5	17	8	27	24	135
2.27 - 3.05	4	0	6	0	6	5	2	23
3.06 - 3.12	2	9	6.5	1	2	9	2	31.5
3.13 - 3.19	7	8	4	6	2	0	0	27
3.20 - 3.26	20	22	21	14	6.5	22	9	114.5
3/27/2006-4/2/2006	7.5	12.5	10.5	5	5.25	16	11	67.75
4/3/2006-4/9/2006	16	21	14	8	8	16	11	94
4/10/2006-4/16/2006	3.5	9	16.5	10	0	25	17	81
Total	129	171	151.75	92	72.75	184	113.25	913.75



Similar to Phase 2, each task for this phase was allocated to certain sets of individuals who then collaborated on its completion. Along with documentation responsibilities however, each member was also assigned programming tasks as Phase 3 marked the beginning of our implementation. The team as a whole worked more as a single unit, choosing many times to hold group work sessions where each member could work side by side.

Feasibility Update

Technical Feasibility

The overall technical feasibility of our system remains unchanged from past phases.

Economic Feasibility

The overall economic feasibility of our system remains unchanged from past phases.

Organizational Feasibility

The overall organization feasibility of our system remains unchanged from past phases.

System Difficulty

The project still has a system difficulty rating of 1.

Factors contributing to this rating include Scope, Project Size, Coding difficulty, Team Composition/Skills, and Technology integration, all of which remain unchanged.

**To see our overall system feasibility, please refer to Phase 1.*

While there haven't been any significant changes or deviations from our project vision or project plan, there was some work slippage as we attempted to work around Spring Break. However, this did not prevent us from completing the phase on time as tasks again were only overdue by an average of approximately two days.

A summary of our use case estimations from Phase 2 can be seen below.

Items	All Use Cases	
	Previous	Current
Actors	2	2
Use case simple	12	18
Use case average	9	6
Use case complex	0	1
UUCP	156	171
TCF	1.07	1.07
EF	0.64	0.64
UCP	105.99	116.19

Scenario (All Use Cases)	Hours / UCP	Person-hours
Best case	10	1161.90
Average case	15	1742.79
Worst case	20	2323.72

QA Manager's Report

Defect Prevention:

The method we took for defect prevention was careful examination of code by the coders themselves as well as constant consultation with Professor Heimann to verify code quality. As a result, the following chart displays the number of hours that the coders took in order to prevent any mistakes from occurring the initial stage of the task.

	Hours	Percentage (Hrs/Total Hrs of Phase 2)
View Files	3	1.3%
Total	3	1.3 %

Defect Detection:

Our method for defect detection was carrying the quality level of tasks from bronze to silver and so on. The detection took up some of our hours because this activity only included the TQA reviewing the code line by line and understanding the execution. As errors were found they were noted and examined to see if a quick fix was available. If not, the errors were explained in detail to the original code writer. The following chart exemplifies the number of hours that were placed into the brief review of the initial coding files.

	Hours	Percentage (Hrs/Total Hrs of Phase 2)
View Files	2	.8%
Integration	30	12.6%
Total	32	13.4 %

Defect Correction:

This portion of quality assurance required the most hours because this method required team members to thoroughly review and edit the mistakes and changes that were made along the processes. These bug correction sessions were very long and involved participation by the original coder, the TQA as well as Professor Heimann. As a result, most of the "cost of quality" hours were allocated towards defect correction. The following chart represents the hours that the team placed into correcting any defects that were found in each task.

	Hours	Percentage (Hrs/Total Hrs of Phase 2)
View Files	3	1.3%
Integration	20	8.4%
Total	23	9.7 %

Total Hours for Quality Assurance: 58 Hours
Total Hours for Phase 4: 239 Hours

Cost of Quality Assurance: $58 \div 239 = 24.3\%$

To conclude this report, the methodologies of quality included:

- Pairing team members
- Review and editing sessions
- An established metrics system.

Project Metrics

Time and Effort

Please see the Time Accounting Section above.

Defects

Because defects have a direct relationship to software quality, tracking of defects is critical. In order to ensure quality, all defects were logged and their closure tracked. For this procedure, information including the manifestation of the defect, its location, the names of the team member(s) who found it, and its status were recorded through the QuickBug utility for Ensemble. Our collected defect data can be seen below:

Name	Description	Priority	Status	Created By	Assigned To	Date Created	Date Resolved
Modify UCPE	Button does not exist	High	Open	Robert Tung	N / A	3.22	N / A
Incorrect PPS Validation	"Minimum" and "Maximum" fields switched	High	Open	Robert Tung	William Cheung	N / A	N / A
Incorrect "Modify Timelog" validation	not being calculated in the right places (set and gets).	High	Open	Robert Tung	Robert Tung	N / A	N / A
Incorrect "Modify PPS" view	\$query4 = "INSERT INTO tblUCPE VALUES (NULL, '\$ACT_avera...	High	Open	Robert Tung	Robert Tung	N / A	N / A
Bad SQL Query in "Modify Timelog"	public function deleteProject() { // Get project ID to de...	High	Open	Robert Tung	Robert Tung	N / A	N / A
Select Project not working	Select Project does not reload page Select Project does not set ap id session id not set correctly	Critical	Closed	Robert Tung	Robert Tung	4/10	4/12
Menu not dynamically changing	Student Array not being set Account class inaccessible because of private variables	Critical	Closed	Robert Tung	Robert Tung	4/10	4/15
Session data incorrect	All session variables were not being updated	High	Closed	Robert Tung	Robert Tung	4/8	4/10
Modify Timelog Validation	Validation is not implemented with modify form	Medium	Closed	Robert Tung	Robert Tung	4/7	4/10

Activities and Issues

As mentioned in our previous phases, meeting our established project goals continues to be our basic motive and thus all aspects of project execution that can affect the attainment of our goals was monitored. To monitor individual performance throughout Phase 3, incremental status reports were prepared and collected during our meetings. The reports took into account all personal checkpoints and deadlines made or missed. A summary of the collected data can be seen below.

Documentation Task	Assigned To	Priority	Completed On Time
Executive Summary	Stephanie, Eric	3	Yes
Agenda	Eric, Alan	3	Yes
Project Review		1	Yes
System Vision	Jamil	1	Yes
Project Plan	Jamil	1	Yes
GUI Prototype	Jamil, Karin	1	Yes
Review Report	Stephanie, Karin, Alan	1	No
Implementation Status	Rob, Will	3	Yes
Updated Task List	Eric, Karin	2	Yes
Updated Architecture	Rob	2	Yes
Update Design Document	Stephanie, Karin	2	Yes
Detailed outline for Final Evaluation		1	Yes
User Manual	Stephanie	1	Yes
Technical Manual	Jamil	1	Yes
User Testing	Rob	1	Yes
Project Management		2	Yes
Time Accounting	Will	2	Yes
Feasibility Update	Jamil	2	Yes
QA Manager's Report	Stephanie, Alan	2	Yes
Project Metrics	Jamil	2	Yes
Risk Management	Eric	2	Yes
Problem Status	Eric	2	Yes
Total Number of Tasks		19	Tasks Completed on Time
			18

Implementation Tasks (A/B level)	Assigned To	Priority	Completed On Time
<i>Entities</i>		1	
Account	Eric	1	Yes
PPS	Will	1	Yes
Project	Alan	1	Yes
Time log Entry	Karin	1	Yes
UCPE	Rob	1	Yes
<i>Use Cases</i>		1	
Change Password	Stephanie	1	No
Login	Stephanie	1	No
Logout	Jamil	1	No
Resend Password	Jamil	1	No
View Accounts	Eric	1	Yes
View Time logs	Karin	1	Yes
View PPS	Will	1	No
View Projects	Alan	1	Yes
View UCPE	Rob	1	No
View Student Submissions	Rob	1	No
Total Number of Tasks	15	Tasks Completed on Time	8

Looking at the information above, our team completed 95 percent of its documentation tasks on time. This is a significant improvement from last phase especially since we were able to accomplish it given the added deviation of Spring Break between. On the other hand, we were only able to complete approximately 53 percent of our implementation tasks, a definite problem. Individual statistics in terms of percentage of tasks completed on time can be seen in the table below.

While Spring Break and finals played a part in this, there seems to be larger issues. As each task has been done on an individual basis, we found it very difficult to fully integrate various components of our system. Also, team members do not fully understand one another's code while changes are regularly made without informing others. As a result, entities and uses cases we initially believed "complete" in fact had to be changed, ultimately affecting our progress. To alleviate our problem, we have taken steps to get us back on track. Working closely with Professor Heimann and with each other will prove beneficial. In addition, periodic code freezes will allow for careful analysis and testing by the entire team simultaneously.

* For a more detailed explanation of our integration issue , please refer to the Risk Management section.

The tasks in this table are broken down by priority and contain both documentation and implementation tasks. Priority 1 tasks are tasks that were the most important and were essential above all to the success of this phase. Priority 2 tasks were tasks that were necessary but could be completed with a little less urgency. Priority 3 tasks were tasks that were only completed after all others had been done.

Name	Overall Tasks	Priority 1 Tasks	Priority 2 Tasks	Priority 3 Task
Karin Burns	83 %	75 %	100 %	100 %
Eric Cheng	100 %	100 %	100 %	100 %
William Cheung	75 %	50 %	100 %	100 %
Jamil Dewji	75 %	67 %	100 %	-
Stephanie Kimmsy	57 %	25 %	100 %	100 %
Robert Tung	67 %	50 %	100 %	100 %
Alan Wu	80 %	67 %	100 %	100 %

Risk Management

Risk Name:	Integration Difficulty				
Triage Level :	Moderate Impact, High Likelihood				
Description :	Our project will require the integration of the use cases that our team produces and the original 67-272 website. During phase 4, the integration problem has become a serious risk due to the accumulation of new use cases and different versions of templates. In order to organize our site better, we had to move the files around, which resulted in many broken links and dysfunctional buttons.				
Warning Signs :	• Team does not understand Professor Heimann's code • Team does not understand Professor Heimann's new graphing classes • Team does not understand each other's code during integration • Team members change each other's code without letting everyone know • Our use cases depend on each other's work. One change might lead to errors in other places • Team understands the given code, but does not know how to implement • Team knows how to implement use the classes, but does not know what it produces				
Abatement :	• Work closely with Professor Heimann to make sure that we understand his code • Work with Professor Heimann to go over the specific functionalities of his classes • Make sure that when we are implementing, it is in sync from the beginning to the end with Professor Heimann's code • Simultaneous code review				
Mitigation :	• Institute periodic "Code Freezes" where all production of new code stops. The following procedure will be followed: - o A set time deadline for the "Code Freeze" when all code is moved to a different server - o Round-robin testing of all entity classes by all team members (minimum 6 persons present) - o Subject all entities to standard and specific sets of tests - o Fix all errors on the spot - o Re-integrate site from the ground up: All entities will be added one by one to a starting "barebones" site, and tested as each entity gets included - o All "Gold" entities moved to predetermined "Gold" directory, only accessible by technical lead. Approval by two team members is required for updates or changes to items in the "Gold" directory				
Signed Off On	Signed Off By	Warning Signs	Abatement	Mitigation	
3/20/2006	Robbie Tung	X	X		
3/21/2006	Robbie Tung	X	X		
3/22/2006	Robert Tung	X	X	X	
4/7/2006	Robert Tung	X	X		

4/8/2006	Robert Tung	X	X	
4/9/2006	Robert Tung	X	X	

Risk Name:	Feature Creep			
Triage Level :	Moderate Impact, Moderate Likelihood			
Description :	As we analyze, design, and actually start implementing the system for the IS department, it is very possible that there will be requirement changes, hence the feature creep.			
Warning Signs :	• New attempts to add in requirements or features that do not apply to the system • New discovery in future interviews – more complicated details and different requirements • One team member is working completely alone on a particular assignment and no one else has seen his or her work			
Abatement :	• Review the requirements gathered all together meticulously so that there is minimum requirement changing when the actual implementation starts • The project manager should make sure that no one works on additional use case until all entities and A level use cases are implemented • Should have consensus of everyone before changing the implementation • Have a fixed implementation plan approved by majority vote			
Mitigation :	• Assign more people to help gather requirement and catch up with the implementation on the case • Implement a more “down and dirty” version to catch up • Drop the less important B level and C level use cases			
Signed Off On	Signed Off By	Warning Signs	Abatement	Mitigation
4/11/2006	Eric Cheng	X	X	X

Risk Name:	Schedule Conflict			
Triage Level :	Low Impact, Moderate Likelihood			
Description :	Schedule Conflict can be a serious threat to our project because we have the most people among all the IS groups and it can be quite hard to find time to meet. This can slow down our progress and give the project manager more workload as she tries to work out everyone's schedule			
Warning Signs :	• Team members cannot meet on time • Team members are late or missing meetings • Project management has no clear idea what everyone's schedule is			
Abatement :	• Let team know any change of schedule or other commitments ahead of the time • Make good use of ensemble to communicate • Share our progresses with each other – making sure that everyone is on the same page.			
Mitigation :	• Divide the team into subgroups and assign project managers for each subgroup • Long meetings on weekends • Short, frequent meetings during weekdays to keep everyone in check			
Signed Off On	Signed Off By	Warning Signs	Abatement	Mitigation
3/5/2006	Eric Cheng	X	X	X
3/7/2006	Eric Cheng	X	X	X
3/9/2006	Eric Cheng	X	X	
4/8/2006	Eric Cheng	X	X	
4/10/2006	Eric Cheng	X	X	

Risk Name :	Difficult adaptation by user			
Triage Level :	Low Impact, Low Likelihood			
Description :	Although the occurrence likelihood and the potential loss are low, users (i.e. Sharon, the IS faculty, and potential 272 students) might not adapt to the system well. Sharon has been collecting and organizing IS homework manually, and the way it is now is probably pretty efficient for Sharon. Professors and TA's might still like to review the PSP worksheets in paper if this system does not make collecting assignments or grading more convenient.			
Warning Signs :	• Team members cannot agree on interface design • Sharon cannot understand the sample throw-away prototype – most likely a simple HTML page or wire frame • Professor Heimann's warning on the use case specifications • Criticism from the results of user testing			
Abatement :	• Stay in close communication with Sharon and fellow IS students regarding the design and outlook of the system. • Creating a more specific survey for former 272 students regarding the general features and design of the			
Mitigation :	• Use the think aloud method to test the user-friendliness and find out what changes will result in a more efficient system • Implement help or assistance functionalities to help initial users start off			
Signed Off On	Signed Off By	Warning Signs	Abatement	Mitigation
4/10/06	Robert Tung	X		

Problem Analysis

Note: More occurrences of emergencies and sickness happened during phase 4. Team members often had to go out of town on weekends and miss some meetings during the week. Therefore, the problem has been “reopened” for this phase. In addition to emergencies, team members are still experiencing some “schedule conflict problems” as the semester approaches the final stretch. “Integration difficulty” has become the most critical issue during phase 4 as all the use cases were finished but were still dysfunctional due to the integration process. The status stays “open.” Two potential problems – “lack of programming skills” and “outside distraction” – still remain open as we finish up the more complicated use cases during the month of Carnival. “Lack of database skills” has been closed because no one seemed to have a hard time writing query or managing the phpMyAdmin database.

Problems Encountered during Phase 3

<i>Problem</i>	<i>Description</i>	<i>Solution</i>	<i>Status</i>
Emergency and Sickness	Our group members have been getting sick or need medical treatments. There were also instances that not all members can meet due to bad weather conditions. This can potentially hinder our progress as a team.	The project manager has to delegate assignments to distribute the work while someone is missing and leaves a lot of time in the end for mistakes.	Reopened
Integration difficulty	Team members have problem understanding each other's code because we all worked alone for BreadExpress in 272. Also, team members change each other's code, causing multiple errors in different use cases and entity classes.	Our team should talk about what needs to be changed with the consensus from everyone so that the risk of causing errors in other parts will be minimized.	Open
Schedule Conflict	Team members have a lot of classes and extracurricular commitments and everyone seems to only have little time to meet between classes.	Even though meetings are short sometimes, the team should still meet frequently to keep everything in check.	Open
Lack of Communication	Team members or the project manager often set up meetings and cancel meetings without letting the others know. Team members sometimes do not let each other know if he/she will be late or miss certain meetings.	Change of project manager might help. The new project manager should have a better idea of keeping everyone well informed of what is going on. Also, we should use more phone calls rather than just emails.	Closed
Lack of Team Activity Participation	A lot of activities besides work have been set up – breakfasts, lunches, dinners, and milestone celebrations. However, only few people participate and actually benefit from the morale boost and	Team can organize a more detailed and planned out calendar for each week so that team members can inform the team in advance and confirm if the event	Closed

	fun. This can hurt teamwork in the long run.	is happening at all.	
Limited Data Collection	We have only had interviews with Sharon, Professor Heimann, and Professor Stolarick and created one on-line survey for our classmates to fill out.	Meet with Professor Heimann immediately after problem occurs and develop new set of interview questions for Sharon to gather necessary requirements	Closed

Potential Problems in the future

Problem	Description	Solution	Status
Lack of programming skills	Accordingly to our skill assessment sheets, not many members put down more than a 5 (average) on the familiarity with PHP	When one spends more time than the average coding time for specific entities or use case, he/she should start scheduling meetings with Professor Heimann to get help instead of wasting more time.	Open
Outside distraction	The occurrence of Carnival week	Make most tasks due before Spring Break, specially all documentation and difficult entity classes. Also, when people are not as efficient, team should still make sure there are a lot of communication and reminders of how far the team has progressed.	Open
Lack of database skills	Accordingly to our skill assessment sheets, not many members put down more than a 5 (average) on the familiarity with database and SQL. It has actually been reflected in phase 2 in which only one person was able to provide insight on database design.	Meet with Professor Heimann to guide us through analyzing and designing our database	Closed

9 -- Appendix
