An Ordering System For Prima
Continental Bakery
Roman Licznerski
Computing And Management Studies
Session (2004/2005)

The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

(Signature of student) _____

TABLE OF CONTENTS

CHAPTER 1 – INTRODUCTION

This chapter provides a brief look at the history of the stakeholder, background to the problem and the problem statement.

1.1 Introduction to Stakeholder

The stakeholder in question is Bradford based Prima Continental Bakery. The company was founded 1952 by John Paciej and has been passed down through the generations, with the current owners being Henry Szepler and a silent partner. The business expanded rapidly distributing its goods to major food retailers in the West Yorkshire region, which included major supermarket chain Wm Morrison. But recently the business has seen a down turn in revenue due to aggressive competition from major supermarkets including Wm Morrison, who ended their relationship with Prima Continental Bakery and started to produce goods in store.

Due to the increase in competition Prima Continental Bakery reviewed its business strategy and decided to focus more on its core business which is sandwich shops in the West Yorkshire, with emphasis being placed on producing hand made continental bread and baps. Recent financial data shows the business is making a turnover of around half million pounds, which shows its in a strong financial position to cope with competition from the major supermarkets.

1.2 Background to the problem statement

Prima Continental Bakery currently produces a range of products which can be categorised under the terms confectionary and bread. There is a total of 21 products categorised under the confectionary term, these include products like scones, curd slices, flapjack and rhubarb cake. There are also a total of 21 products categorised under the bread term, these include products like baps, baguettes, bagalinos, scufflers, ciabatta and panini. All these products are printed on an order form which is used to record data about what products each customer has ordered for the next working day. Each single order form has attached to it a carbonless duplicate form which leaves an impression on the duplicate order form (Appendix E). This duplicate is produced for legal reasons and also used as reference when customers have queries about previous orders.

Everyday an order form is filled out by the receptionist for each customer, and the duplicate is then stored in filing cabinet. The demand on physical storage space continues to escalate as the company has to store invoices for the past three years for legal reasons. Also the manual filing system hampers the retrieval of documents, as there is no set structure in filing and they can be easily lost. At the end of each day, the manager uses the original order forms from all customers to calculate manually the total amount of each product needed to be produced for the next working day.

Prima Continental Bakery has realised that if they are going to expand the business, which is one of their long term goals, they need a more efficient and effective ordering system to cope with an increase in incoming and outgoing data and also to improve data retention.

1.3 The Problem Statement

The problem to be solved in this project is to design and implement an ordering system which meets the end user requirements and future expectations. The first step will be to review the current paper based system, identify what the user wants from the new system and examine whether their expectations are realistic.

CHAPTER 2– PROJECT MANAGEMENT

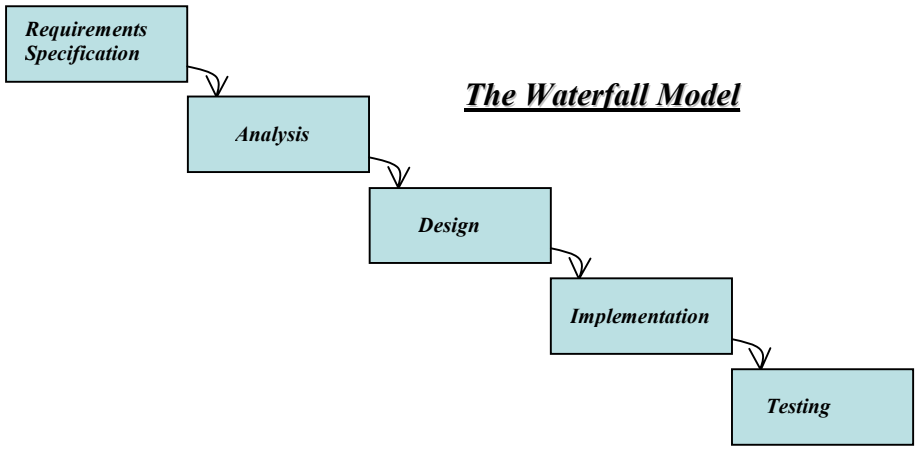
This chapter looks at the schedule of designing and implementing the deliverables of the project to the customer. The following sections describe the project methodology that is going to be followed and also examine the initial project plan and any revisions made to it.

2.1 Project Methodology

Due to the large scale of this project we must initially define which methodology we are going to follow throughout the development of the project. The foundation blocks of which methodology we are going to follow, are focused around the system development life cycle (SLDC) (Avison and Shah, 1997) which is a conceptual model used in project management. The SLDC model shows the process of developing a system through a series of stages, which are described briefly below:

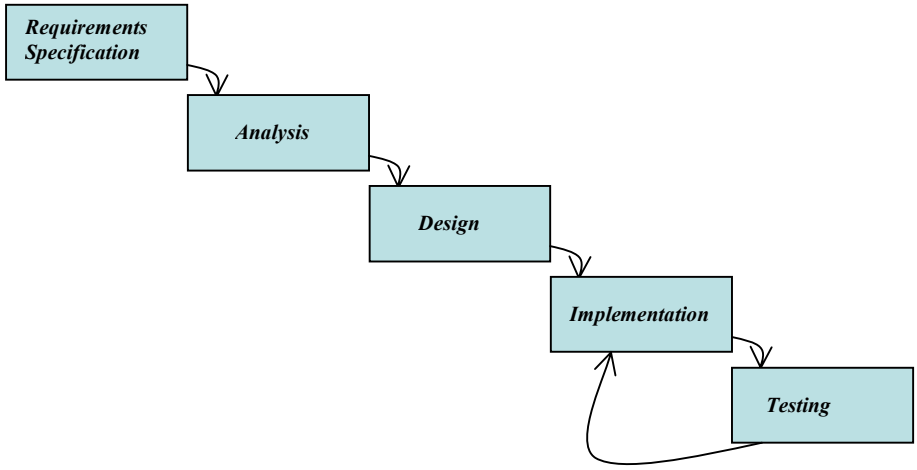
1. The existing system is evaluated. Deficiencies are identified by interviewing users of the system.
2. The new system requirements are defined. In particular, the deficiencies in the existing system must be addressed with specific proposals for improvement.
3. The proposed system is designed.
4. The new system is developed. The new components and programs must be obtained and installed. Users of the system must be trained in its use, and all aspects of performance must be tested.
5. The system is put into use. This can be done in various ways. The new system can phased in, according to application or location, and the old system gradually replaced. In some cases, it may be more cost-effective to shut down the old system and implement the new system all at once.
6. Once the new system is up and running for a while, it should be evaluated.

The stages described above, are a series of defined steps which can be used to develop a system. But not all systems have the same requirements and that is why a number of system development life cycle models have been developed. The most common model is the waterfall, which has distinct goals for each phase of development. Once a phase of development is completed, the development progresses to the next phase and there is no turning back. Below is a diagram of the waterfall model:



The advantage of the waterfall model is that it allows deadlines to be set for each stage of development. This enables the project manager to have clearly defined goals when certain stages have to be completed and this means that the solution will be delivered on time. The disadvantage of using the waterfall model is that is doesn't allow for revisions, so once the system is in the testing stage it can't go back to change aspects of the system which were not designed well.

Other system development life cycle models reviewed include Rapid application development (RAD), Joint application development (JAD), Spiral Model and the Fountain model. All these models mentioned above have clearly defined advantages and disadvantages and they have all be considered when deciding which methodology would be right for the development of the project. But in the end I decided that the methodology which I will be following throughout this project would be the waterfall model. But I will be making one modification to the model and this will be at the testing stage. In between the implementation and testing stages I will add an iteration, which will allow for modifications to be made to the system as a result of user feedback. Below is a diagram of the modification made to the waterfall model:



Overall I have chosen the waterfall model due to its simplicity and clear defined stages. The model is easy to understand and follow and with the clearly defined stages it will mean that I can keep to my schedule and produce the desired deliverables on time.

2.2 Project Milestones

Throughout any major project there are key milestones and deliverables which need to be completed. Four key millstones have been identified for this project and have been stated below:

Milestone 1: Mid Project Report
This milestone does not contribute to the overall solution of the problem, but indicates the stage where the Aims, Objectives and Minimum requirements are agreed with the end users.

Milestone 2: Design, Implement
This milestone indicates that the ordering system has been build.

Milestone 3: Testing
This milestone involves testing the system with user involvement and making modifications to the system with respect to the user's feedback.

Milestone 4: Final Year Project
This milestone does not contribute to the overall solution of the problem, but indicates that the ordering system has been completed and that all the processes involved in the development have been documented.

2.3 Project Schedule
With the methodology decided we can now draw up a schedule, which sets out the timescale for when each stage in the model and when the deliverables have to be completed by. Revisions had to be made to the initial plan due to some unforeseen circumstances that needed to be taken into account. These included:

(1) The first delay occurred at the testing section when we had to get the end users feedback on the system. Due to the users work commitments we had to reschedule the appointment on 03/02/2005 to the following week.
(2) The second delay occurred during the schedule to write up the project report and to submit the report. Due to a family bereavement the schedule for these tasks had to be reviewed. The schedule was extended by a week to take into account these circumstances.

Both the actual and revised project schedule can see in Appendix G

CHAPETER 3 – BACKGROUND READING

3.1 Introduction

To be able to understand the nature of the problem being solved, I need to do some background research related to the design and implementation of a database system. Topics going to be researched that are linked with this project include why projects fail, user involvement, gathering user requirements and system deployment.

3.2 Why Projects Fail

Many businesses, whether they a small or large, are trying to improve their systems to keep with competition and to increase efficiency and data retention. But figures from the Standish Group (1995) show that project failures are high, with 31.1% of all projects being cancelled before completion. These figures are 10 years old now, but current research being carried out by the same group show that there is a slight decrease in the percentage of projects that are cancelled before completion.

This information would surprise many people due to the increase in technology available to modern businesses today, but the fundamental reason why projects fail is due to the fact that end user input into the development of system is ignored. This view is backed up by Yeates (1986, pg 262) who suggests implementations can fail due to '*the lack of user acceptance*'. Lack of user acceptance is due to the fact that the end users do not fully understand the need for the new system. But many users do not understand the need for the new system because they have not been included in the discussions about the new system and its aims.

Also sometimes user's judgement on developing a new system can be clouded because they may feel that when the system is introduced they could lose their jobs. This research clearly indicates that when developing the new system for Prima Continental Bakery we must listen to the needs of the end users and management and define each stage in development clearly to both parties.

3.3 User Involvement

As the previous topic shows, user involvement is an important aspect that project managers need to take into account when developing a new system. Involving users in stages of analysis and design allows the user an opportunity to have an input into the development process. This user involvement makes the users feel important as they are having an input into how the system is run, and this in turn makes it more likely that they will accept the introduction of the new system. This view is accepted by Yeates (1986, pg 21-23), when he discuses the fact that an increase in participation from users leads them to gain '*more confidence and control*'.

3.4 Gathering User Requirements

With any major project, the developer first need to get a clear understanding of the users needs. As stated in the introduction, the paper based system currently in use at Prima Continental Bakery has problems and the mail goals are to have a more efficient and effective ordering system to cope with an increase in incoming and outgoing data and to improve data retention. To investigate the processes involved in the current paper based system we need to use some type of fact finding technique which will allow us to record all procedures that take place from start to finish.

Using notes from IS21 Object Oriented Analysis and Design module (Johnson, 2003) and also (Bennett, McRobb and Farmer, 1999, pg107) the most common technique used in industry is SQIRO. SQIRO stands for Sampling,

Questionnaire, Interviewing, Reading and Observation and these techniques can be used to help gather the user's requirements of the new system and scrutinise the current paper based system. Overall I feel that the most useful techniques from SQIRO are:

- *Sampling*; Sampling involves gathering documents that are currently being used as part of the paper based system. The most obvious document in this case would be the order form, which is used to store customers orders.
- *Interviewing*; Interviewing will enable me to gather user's views about the current paper based system and also the requirements about the new system.
- *Observation*; Observing the users performing real time tasks will provide me with firsthand experience of the way that the system works and allow me to verify the information gathered from the end users about the current paper bases system.

3.5 System Deployment

After the system has been completed to the standard that meets the needs of the end users and the organisation, the next step is to introduce the system to the organisation. Bennet, McRobb and Farmer (1999, pg 427) state that there are four methods that an organisation can use to introduce a new system, these are:

(1) *Direct Changeover*
In direct changeover, the old system is stopped at the end on one working day and the following day the new system is used. The major problem with this deployment is if there are errors in the new system developed which have not been identified.

(2) *Parallel Running*
In parallel running, the old system and new system are run together for a certain period agreed by the organisation. The old system is stopped when the new system is running effectively to meet the needs of the organisation in real time. The advantages of this deployment is that their can be no disasters if there are errors in the new system. But the disadvantage is that there is a lot of duplicate work being done as both systems are being run.

(3) *Phased Changeover*
In phased changeover, part of the new system is used and when all the problems have been solved and it meets the end users needs, another part is implemented.

(4) *Pilot Project*
In pilot project, the new system will be tried out by a department in the organisation. This option is not available to us as there is only one place where all the processes take place in the organisation.

Methods 3 and 4 are not available to Prima Continental Bakery due to it being a small organisation. The best option which I would recommend for the organisation is that it chooses the parallel running deployment plan. Although this would mean an increase in work load for the receptionist as work would have to be duplicated, the benefits of using this method is that no disasters can occur if errors are found in the new system.

CHAPTER 4 – ANALYSIS OF CURRENT SYSTEM

4.1 Introduction

The aim of this chapter is to explain the system that this project is concerned with, providing the base for the chapters that follows. We will be focusing on the two systems available to the company, the paper based system and a new computer based system. I have gathered the following information using a couple of SQIRO techniques described in the background research section (Chapter 3).

4.2 Procedures of the paper based system

I have gathered this information about the current system by observing the end user performing the tasks in real time and also by conducting interviews with Halina Brown (Receptionist, End User).

The current ordering system at Prima Continental Bakery is paper based and has been since it was established in 1952. There is a set of procedures that are carried out by the receptionist and manager every day to document daily orders by customers. The step-by-step procedure was observed on the 11/11/2004 and is described below:

- Orders are collected by the receptionist between the times of 12:30 – 3:00 weekdays, and 10:30- 12:00 on Saturday.

Procedure carried out by receptionist:
(1) The receptionist rings the customer
(2) Fill out order form (Appendix E), recoding products ordered by customer
(3) Tare out one copy of the order form for delivery, and keep duplicate copy in book
(4) Repeat steps 1-3 for all customers

Procedure carried out by the manager after all the orders are collected by receptionist:
(5) Using the order forms, the manager calculates the amount of each product ordered that needs to be produced for the following working day. This procedure is done manually using a calculator, and calculations are repeated three times to avoid errors.

The steps described above are done daily, but another task that is carried out by the manager every week is calculating the invoice for each customer. This task involves calculating the total number of products ordered by each customer over the previous week and then calculating the total amount of the bill. Again this procedure is done manually using a calculator, and calculations are repeated twice to avoid errors. This task is very time consuming and takes around 7 hours to be completed.

4.3 The case for the paper based system

The aim of developing any system is to provide a solution to a need that not only meets user requirements, but one that is also accepted and used by the users. One major reason why the system may not be accepted is that users fear change. Whilst the users may admit the system has problems, they learn to accept them and changes to their procedure could result in discomfort. The strengths of the current system need to be determined so that developers can work with them in order to help users overcome their fears and thus reduce the dependency on the paper-based system.
The strengths of the paper-based system revolve around the fact that users are familiar with current working practices and a new system would inevitably require changes to work methods and users may reject the need to retrain, and so I need to

ensure that the impact of change is minimised so that the new system is successfully introduced. Time needs to be allocated for training users in using new system, and also a user manual would help users with any problems or uncertainty they have regards performing tasks on the new system.

4.4 The case for automating the paper based system

The main goal of producing the new automated system is to minimise the impact on the users and provide visible benefits that will be recognised by all end users. With the introduction of a automated system there is the potential for more efficient use of staff time through reducing the number of paper handling activities as all the information is stored electronically not on paper. The required information should be available at the touch of a button rather than spend hours searching through filing cabinets, and the chance of order forms being mislaid are reduced greatly. Centralised electronic storage results in no further need for physical filing cabinet space, storage needs become smaller in size for example utilising optical or tape drives. Improved security results from reducing the access to the information, rather than using filing cabinets.

CHAPTER 5– REQUIREMENTS SPECIFICATION

5.1 Introduction

Documenting requirements of the users is an important part of ordering systems development, as the information discovered at this stage forms the basis for the system design. In this chapter we will be looking at various types of topics relevant to the development of the system, including the benefits the stakeholder and users can expect from the system. I have gathered the following information by observing and interviewing the end users (receptionist and manager).

5.2 Business Opportunity

The automation of the current paper based ordering system will provide the business with the opportunity to improve the efficiency of documenting daily orders and other tasks relevant for the day-to-day running of the business.

5.3 Problem Statement

The problem presented in the current ordering system is the time consuming process of documenting daily orders, calculating weekly invoices and the ineffective way of retrieving order forms. The problem affects receptionist and managers. The impact of the problem is that firstly, reception staff spend to much time documenting and searching for order forms. Secondly the demand on physical storage space continues to escalate as the company has to store invoices for the past three years for legal reasons. And finally the tasks performed by the manager are time consuming and error prone due to the fact that he calculates invoices manually using a calculator.

5.4 Core Problem

One of the major problems of the current paper based system is that each week around 240 order forms are generated. Due to legal reasons these order forms have to be stored and presently they occupy 7 filing cabinets. Currently there is no set structure of where order forms are placed and so if someone wanted to locate a specific form, this would require a staff member to search though all the cabinets manually. The problem of missing documents is not easily handled too and the fact that the staff have to so much time searching cabinets and handling paper has resulted in the suggestion that Prima Continental Bakery would benefit from the automating of the process.

5.5 Stakeholder Summary

The following table identifies the people involved and how they will be affected by the new automated ordering system.

Name	***Represents***	***Role***
Customers	Recipient of order/invoice forms	Testing/Viewing the invoice layout

Management	Purpose of the system development, reporting facilities and will be an end user	Involved in testing the reports produced and also the final system
Receptionist	Purpose of the system development and will be an end user	Involved in testing the final system

5.6 User Summary

The following table identifies the categories of users that will be using the system being developed:

Name	***Description***	***Stakeholder***
Novice User	User unfamiliar with the operating system. Knows little about the company products and processes	New Staff
Intermediate User	User unfamiliar with the operating system. But knows in detail about the company products and processes	Current staff & management

5.7 User Environment

The environment where the system will be installed in compromises of 3 people, and these will be the users of the new system. The two main roles are performed by the receptionist and the managers, with each of them performing different tasks throughout the week. This indicates that the new system must meet the needs of the users for the tasks that they perform throughout the procedure of processing new daily orders and calculating weekly invoices.

5.8 Summary Of Capabilities

There are a number of benefits to be gained from the deployment of an automated solution. The following have been identified:

Benefit	***Supporting Features***
Reception staff quickly save time in searching and retrieving documents	Centralised storage of data on database will mean that data can be retrieved faster than the existing filing cabinet solution. Also this will result in a reduced chance of documents being lost or misplaced
Speed of access to information and the results format is improved	Ability to query stored data and return order forms that match the query request

Speed of calculating weekly invoices for customers	Ability to use formulas to automatically calculate total of bills
Prima Continental Bakery	Producing reports for management improved greatly

5.9 Training Requirements

There are currently only two users of the current paper based ordering system, and these users will be the main users of the new system. As stated in the background reading chapter lack of user acceptance can arise if users do not understand the business need for a new system. This is why I would recommend that the end users are trained to familiarise themselves with the system and have an opportunity to practice using the system in a calm environment. The purpose of user training is to ensure:

(1) The user understands the purpose of the new system
(2) The user understand how to carry out their previous tasks on the new system
(3) The user understands the new tasks involved in the new system
(4) The user has the opportunity to ask questions about issues and problems about the new system
(5) The user gains practical experience of carrying out tasks in the new system.

The above points demonstrate that training should assist the deployment of the system to help the users build confidence and trust in the new system. To support the new system a user manual is provided. It is necessary to provide instruction and guidance as there are come procedures that need to be detailed step by step. Weiss (1985) discussed three main uses of user documentation as being to 'help the user get started', to 'help productivity/satisfaction', and for 'help when things go wrong' (Weiss, 1985, pg 5). The last point is in particular reason why we are going to produce the user manual that accompanies the system.

5.10 Constraints

Prima Continental Bakery has expressed their concerns with respect to security in context of the internet, and therefore in this project we will not be implementing an internet solution As a possible extension to the minimum requirements, we could document relevant information regarding internet security and also the advantages and disadvantages of implementing an internet based ordering system in the future, which the company could consider when or if they ever consider expanding their system and going on-line.

5.11 System Requirements

With the current system being paper based we have to consider which platform we are going to use when we transfer from the paper based system to the automated ordering system. The system requirements chosen are shown below:

- Platform – The platform will be Microsoft based and the software requested is Microsoft Word and Microsoft Access.

The price of Microsoft Word 2003 is £169.00 and the price of Microsoft Access 2003 = £167.00. So the total cost **£336.00**

After considering this price we decided that the best option was to buy Microsoft Office which included all the major Microsoft platforms Microsoft Word, Microsoft Access, Microsoft Outlook, Microsoft Excel and Microsoft PowerPoint. The price of this package is **£319.00**, so it was £17 cheaper and it also included other platforms which the company could use in the future to improve the service that they are providing.

5.12 Objectives

The set of objectives for the project:

1) *Perform analysis of current system and gather user requirements*
2) *Design and implement a database to add, hold and view data*
3) *Provide a payment function which archives customers payments*

5.13 Minimum Requirements

The minimum requirements set for this project:

1) *A specification of the bakery requirements*
2) *A database management system*
3) *Produce agreed reports after discusses with users*
4) *A payment function*
5) *Warning letters Function (Mail Merge With Microsoft Word)*

5.14 Possible Extensions

Possible extensions for this project:

6) *User manual – For new workers*
7) *Vendor Ordering System*
8) *Stock System*
9) *Produce document relating to security aspects and advantages and disadvantages of having web based system.*

CHAPTER 6– DESIGN

Currently we are at the third stage of the waterfall model, and in this chapter we will be focusing on the aspects of designing the new system. The previous chapters have allowed us to build an impression of the type of system that the end users requires. Topics discussed in the following sections include elements involved in database design and ER-Models.

6.1 Design Of Database

6.1.1 Data Modelling

Before we start the design process, we first need to know what information needs to be stored in the database. Previous analysis of the current system leads us to focus on the order form which represents the hub of all the processes involved in the system. Using the template of the current order form (Appendix E) we can begin to build an Entity-Relationship Model (ER Model). The ER model is a '*high level conceptual data model*' (Elmasri and Navathe, 2000) which is used to represent the data requirements of the system. The model consists of entities and relationships.

Entities are data objects which show which information is to be collected, examples of these include ORDER FORM, CUSTOMER, PRODUCTS. Below is a table showing the key entities and associated attributes for the new system:

CUSTOMER - 'CustomerID', 'Title', 'Forename', 'Surname', 'Home Address', 'City', 'Postcode', 'Home Telephone Number', 'Mobile Number', 'Shop Name', 'Shop Address','Shop Postcode', 'Shop Telephone Number', 'PastDue', 'CurrentDue'

ORDER - **'**OrderID', 'CustomerID', 'OrderDate'

ORDER_LINE - **'**OrderID', 'ProductID' 'Quantity'

PRODUCT - 'ProductID' , 'ProduceName', 'Price'

USES - **'**ProductID', 'MaterialID'

RAW_MATERIAL - **'**MaterialID', 'Name', 'Amount_In_Stock', 'Unit_Price'

SUPPLIERS - **'**VendorID', 'MaterialID'

VENDOR - **'**VendorID', 'VendorName', 'Vendor Address', 'City', 'Postcode', 'TelephoneNumber', 'AmountDueToVendor'

ORDERSTO - **'**OrderID', 'VendorID', 'DateOrdered', 'ExpectedDelivery'

ORDER_LINESTO - **'**OrderID', 'MaterialID', 'Quanity'

The next stage in design involves defining the relationships between the entities. There are three common types of relationships and they are:

(1) *One-to-One* – This relationship occurs when one instance of an entity is associated with another one instance of entity.

(2) *One-to-Many* – This relationship occurs when one instance of an entity is associated with another entity which has more then one instance (2 or more instances).

(3) *Many-to-Many* - This relationship occurs when both entities which are associated have more then one instance (2 or more instances).

In the new system being developed for Prima Continental Bakery there are only 1-to-N relationships present. An example of one of the relationship is between the customer and the order. Figure 1 shows the One-to-Many relationship were a 'Customer submits many Orders'

Figure.1 = One-To-Many Relationship

A full E-R Model for the minimum requirement and extended system is available in appendix B.

6.1.2 Database Schema

The schema of the database is a group of related objects in the database. Below is an example of the schema of the Customer and Orders table:

	Field Name	Data Type	Description
🔑	CustomerID	AutoNumber	Primary Key For Customer Table
	Title	Text	Title Of Customer
	Forename	Text	Forename Of Customer
	Surname	Text	Surname Of Customer
	Home Address	Text	First Line Of Customer Address
	City	Text	City Where Customer lives
	Postcode	Text	Postcode Of The Customer
	Home Telephone Number	Text	Home Telephone Number Of Customer
	Mobile Number	Text	Mobile Number Of Customer
	Shop Name	Text	Shop Name Of Customer
	Shop Address 1	Text	First Line Of Shop Address
	Shop Postcode	Text	Postcode Of The Shop
	Shop Telephone Number	Text	Telephone Number Of Shop
	PastDue	Currency	Past Amount Due
	Current Due	Currency	This Weeks Amount Due

Customer Table

	Field Name	Data Type	Description
🔑	Order_ID	AutoNumber	Primary Key For Order Table
	CustomerID	Number	Foreign Key For The Order Table
	OrderDate	Date/Time	Date Value (Input Mask, 00/00/0000;0;)

Order Table

The database schema shows the attributes in each of the table and also their data type. There is also a brief description about each attribute on the far right of the schema. The schema above shows that in the customer table the primary key is the CustomerID attribute, with the other 12 attributes being text data types and 2 attributes being currency data types. For the Order table there is one primary key which is OrderID and a foreign key which is CustomerID. The foreign key is a reference to a primary key in a different table, which in this example is the customer table. The last attribute in the order table is OrderDate which is a Date/Time data type and this field stores date value.

The Database schema for all the tables is included in Appendix C**.**

6.1.3 Integrity Constraints

When designing a database you have to try and maintain the accuracy and integrity of the data in the database. This can be achieved by using constraint rules, and the two main rules are known as the entity integrity constraint and the referential integrity constraint. The rules are described below:

- '*The entity integrity rule is designed to assure that every relation has a primary key and that the data values for the primary key are all valid*' (MrFadden F.R, Hoffer J.A, Presscott M.B. ,1999, pg 213). So the main role of this rule is to make sure that every primary key attribute is non-null.

- *'The referential integrity constraint rule states that either each foreign key value must match a primary key value in the other relation or else the foreign key value must be null'*(MrFadden F.R, Hoffer J.A, Presscott M.B. ,1999, pg 214). So the main role of this rule is to make sure that the database must not have any unmatched foreign key values.

These integrity constraints mentioned above have to be implemented in the new system to ensure that the data in the database is accurate and maintain its integrity.

6.2 Form Design

The design of the form depends on the information which needs to be visible to the end user. After discussions with the end users of the current paper based system, I noted three priorities that they would like to be taken into account when designing the forms. The first query raised was with regards to the layout of the order form on the new system. Then end users requested that the layout of the order should match as closely as possible the paper based version, their reasoning for this request was to maintain consistency. The second query raised was about the layout of the other forms in the system. The end users requested that the layout of the forms should be consistent, with the function buttons being located at a specific area. This will enable the user to quickly become familiar about where buttons are located in the system. The final query raised was with regard to human error in data entry. Both users know the basic about computers but they have reservations about data entry, that's where validation rules for data entry must be implemented. Validation rules in enable me to set bounders about the type of data entered in specific fields e.g. you shouldn't be able

to add letters in a section which requires you to fill in your telephone number. Validation rules are described in more detail in section 6.2.1

In Appendix F, there are some sketches of how the main forms for the system might look like after taking the end users opinions into account.

Their will be various forms created in the system, with each one designed for one of the following three options: (a) Add data (b) Search data (c) Delete data. The add data option will allow the user to open a specified form and add data to it e.g. new order form. The search data option will allow the user to search for a specific form and then be allowed to just view that data or make amendments to it. The final option is the delete option which allows the user to delete data present in the database.

Below are descriptions of the main formed created for the new system:

Form1 – This form plays an integral part in the new system. This is the order form which the user adds customers new orders.
Form2- The Subform for the order form
Form3 -The Customer form, which can be edited
Form4 – The Product form, which can be edited
Form5 – The Raw Material form, which can be edited
Form6 – The Vendor form, which can be edited
Form7 – The Order form for the stock
Form8 – The Subform for the order form (stock)

6.2.1 Validation Rules

Validation rules allow the database designer to specify the data entry format that fields have to adhere to. The validation rule is implemented when creating the tables in design view. The validation rules are mainly used for data entry forms and that is why we have used them in the order forms, which involves the end users inputting data into the system.

6.3 Query design

In a database system it is vital that specific data can be viewed at a click of a button, and this is where queries play a vital role in achieving this aim. There are two main types of queries that are available to us and they are the select and action queries. The select query gathers, collates and presents information that the user has requested. The action query makes changes in specified records, or creates new tables with related data inserted. Both of these queries will be used in the new system, and descriptions of the main queries designed are listed below:

QryListVan - This query shows the Vendor_ID and Vendor Name, of all the vendors present in the vendor table.
QappPayment & QappPaymentVendor - This query transfers the payment details of all customers and vendors into an archive table (backup of all payments)
QdelPayment & QdelPaymentVendor – This query deletes the payments details of all the customers and vendors in the original payment table.
QdelStockCh – This query deletes all data in the StockCh table
QdelStockDel - This query deletes all data in the StockDel table
QdelStockOrder - This query deletes all data in the StockOrder table

QmakDueThisWeek – This query calculates the total bill for the each customer for the past week and stores the data into a new table called DueThisWeek
QmakStockDue – This query shows if any stock delivery is expected today.
QryProductList - This query shows the Product_ID and Product Name, of all the products present in the product table.
QryRevDel – This query adds the stock expected to be delivered today to the raw material table.
QryTotalVen - This query calculates the total bill for the delivery by the vendor for the raw materials ordered and stores the data in the AmountDueToVendor in the Vendor table.
QupdCurrentDue,QupdCurrentToZero,QupdPostPayment,QupdPostPaymentVen – These queries are related to the payments made by customers.
WeeklyInvoice - This query gathers data about the stock ordered by each customer from the past week.

6.4 Report design

One of the minimum requirements is that the manager can view a report which shows relevant data that he requires. After consultation with the manager their will be two reports that will be produced to adhere to the minimum requirements. The design of the report is not vital, just as long as the data is clear and easy to read/understand.

Displaying the weekly invoices and daily order forms will also be designed using reports. The layout of these reports should be similar to the current order form, so that it remains consistent to the customers. In Appendix F, there is a sketch of how the invoice form might look like after taking the end users opinions into account.

6.5 Macro Design

Macros perform particular operations e.g. Open form, print order form. Macros will help automate the tasks in the system and will be used throughout the design of the implemented of the system.

6.6 User Manual Design

The purpose of the user manual is to provide the current users and novice user's information about how to perform tasks that are vital for the day to day running of the business. The format of the user manual will be as follows:

Front Page
- *Task 1* - How to add new customer
- *Task 2* - How to search for a customer and edit/delete their details
- *Task 3* - How to add new product to table
- *Task 4* - How to add new order
- *Task 5* - How to print out daily order forms
- *Task 6* - How to print out weekly invoices
- *Task 7* - How to print warning letters

CHAPTER 7– IMPLEMENTATION OF DESIGN

This chapter looks at the implementation phase in the methodology which covers the building of the system. The next sections go through the steps of implementing the system from start to finish.

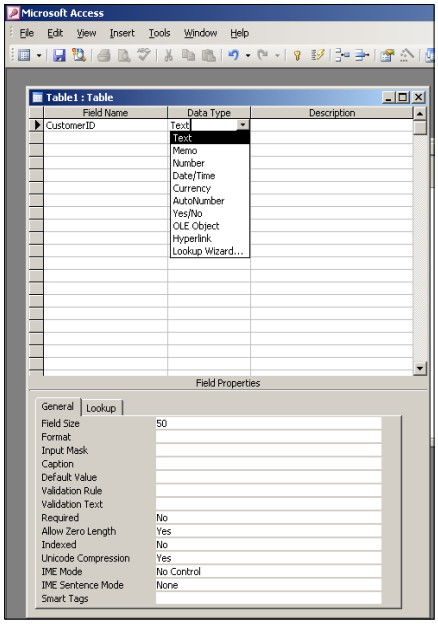
7.1 Implementation of the database

The database will be implemented in Microsoft Access 2003, as this software is readily available to the organisation. Also with many organisation in the industry operating Microsoft applications and platforms their will be no compatibility problems in the future if the business decides to link its system with other organisations e.g. suppliers.

7.2 Creating Tables And Relationships

The first task in implementing the system is to create the tables for the data to be stored. To create the tables you have to select the table menu and then double click on 'Create table in design view' icon. This will open the form (Fig.2) which is where you fill in the attributes for the table.

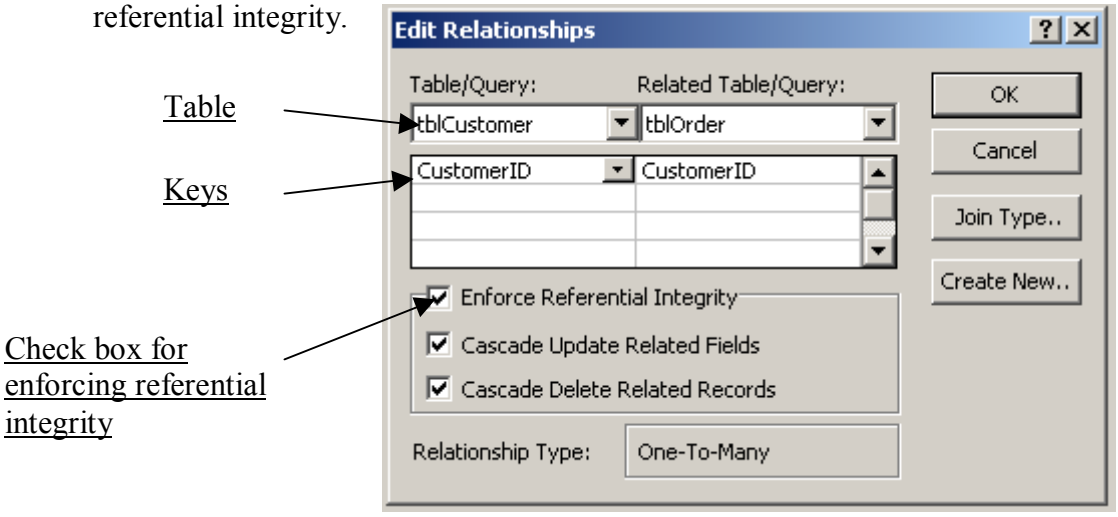
You must also select what data type each attribute is to have and also give a brief description about what data the attribute stores. The table schema for this system is located in appendix C. At this stage you can also set field properties, which includes adding an input mask to the field. This option allows you to set the type of data entered into the field. Examples of this option being used is in the customer form when we have input masks for the:



(Fig 2)

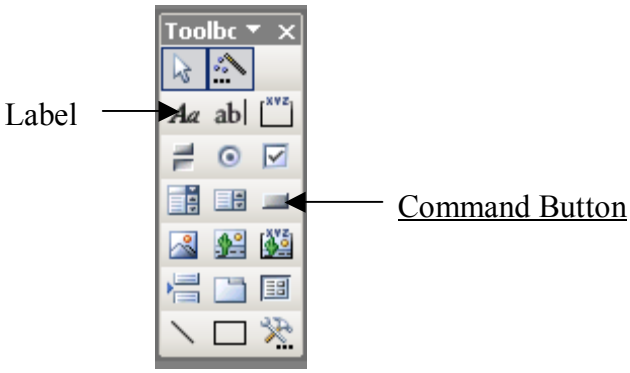
Postcode: Input Mask '**>LL0?\ 0LL;;_**'
Telephone Number: Input Mask '**\(99999\)999999;;_**'

The primary key is also defined when creating tables which is used when creating relationships. To create relationships between tables based on the ER model (Appendix B), you must select Relationship option from the Tools menu. A box appears where you select the tables you want to create relationships for, and then you select the 'Relationship tool' and check the box for referential integrity.



7.3 Implementation of the system main menu interface

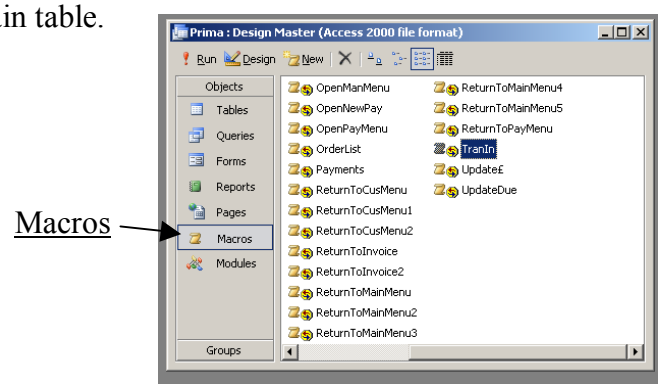
The main menu interface will appear straight away when the user opens Access 2003. There are two options available to us for implementing the main menu interface, and this is by either using the tool called switchboard manager or an access Form. We have decided to create the main menu using the access form, as it is simple to create and amend. All you have to do is select the forms option, and double click on 'Create form in Design View'. A blank form appears with a toolbox (Fig 3), and you can use the tools available to create buttons for navigation around the system.



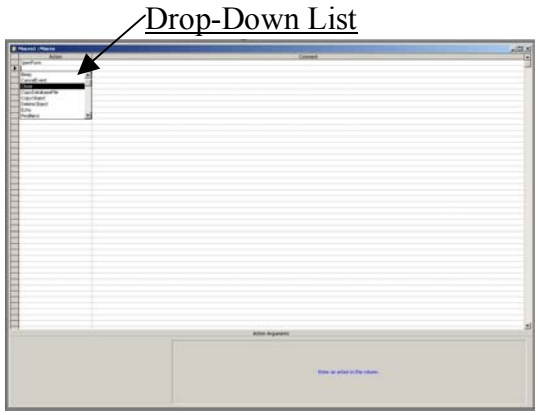
(Fig 3)

7.3.1 Creating Macros to Link Menus

When you have created the forms for each menu, you need to create macros that enable you to open and close forms. To create macros you have to select the macro menu from the main table.



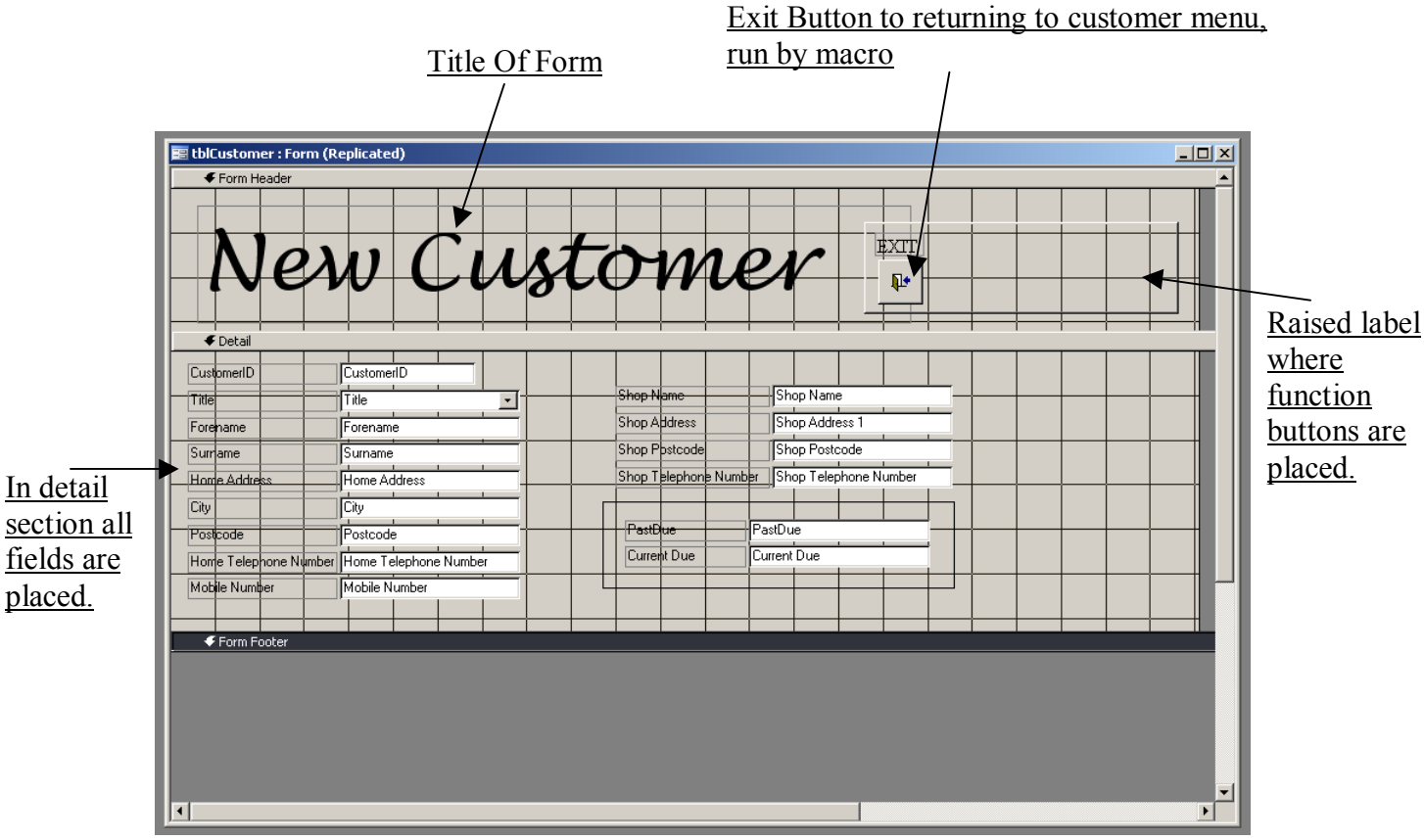
Once you have selected the Macros menu you have to click on 'New' and a Macro form appear as shown in figure 4. The next step is to select the right action from the drop-down list. To be able to move between forms you need to select the actions 'OpenForm' and 'Close' ,when you selected these actions you have to specify which form you would like to close and open. This can be done in the action arguments section which appear at the bottom of the macro form. You should do these macros for all forms operations and other tasks like printing out order forms and invoices.



(Figure 4)

7.4 Implementation of Forms

As mentioned in the design section the design of the form depends on the information which needs to be visible to the end user. But all forms will have the same generic layout which is shown in figure 5.



(Figure 5)

7.5 Implementation of queries

The implementation of the queries are all related to the needs of the end users. There is a range of queries that have been implemented in this system and the main queries are the Payment system, Warning System and the Vendor and Stock Order System. The implementation of these queries is below:

(1) Payment System

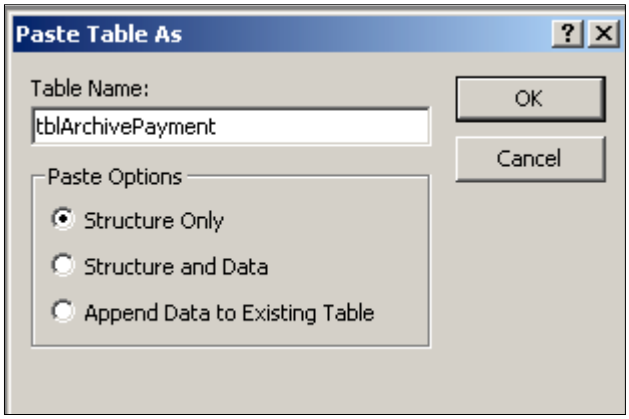
This query allows the receptionist to record the payments made by the customers. The query is designed in the following steps:

a) First you have to create a table with the following attributes:

Field Name	***Data Type***
CustomerID	Number(Long Integer)
PaymentAmount	Currency
PaymentDate	Date/Time

You will have to define the primary key fields, which in this case will be **CustomerID** and **PaymentDate**. This query assumes that no customer will make more than one payment on the same date. You need to save this table as tblPayment.

b) The next step is to make a copy of the tblPayment. From the menu select Edit, Paste. And a box appears (Figure 6):



(Figure 6)

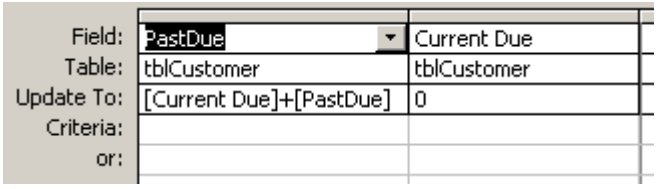
As the table name type tblArchivePayment and select Structure only and then click on OK.

c) Now that we have created the table which the queries are going to run on. The next step is to build the queries. Four queries are needed to perform the payment tasks and they are:

(1) **QupdCurrentToZero**; This query adds CurrentDue to PastDue and sets CurrentDue back to zero on the customer table

Create this query by:

- Start a new query in Design View and add **tblCustomer** to the query window.
- Select Query, Update Query from the menu bar.
- Add the two fields **CurrentDue** and **PastDue** to the query grid.
- Make entries in the **Update To** row as shown in (Figure 7), inserting 0 in the **CurrentDue** column and [CurrentDue]+[PastDue] in the **PastDue** column



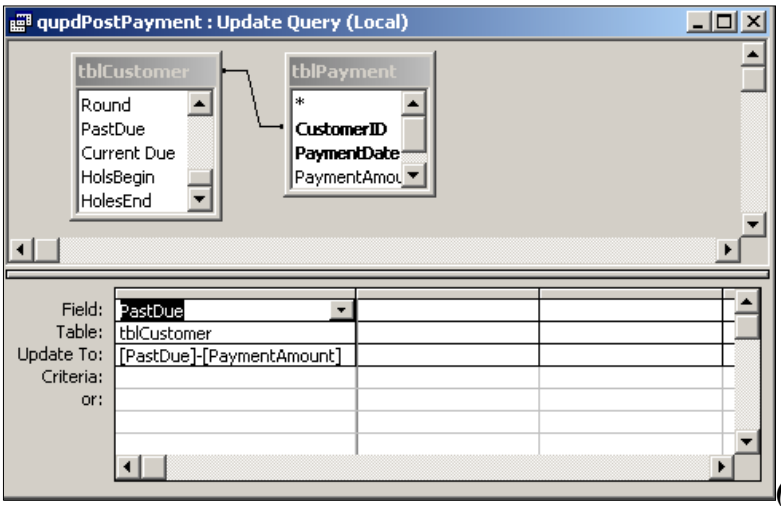
(Figure 7)

- Save the query as **QupdCurrentToZero**

(2) **QupdPostPayment**; This query subtracts each customers payments from their PastDue amount.

Create this query by:

- Start a new query in Design View and add **tblCustomer** and **tblPayment** to the query window.
- Select Query, Update Query from the menu bar.
- Add the **PastDue** field to the query grid and enter [PastDue]-[PaymentAmount] in the **Update To** row as shown below in Figure 8)



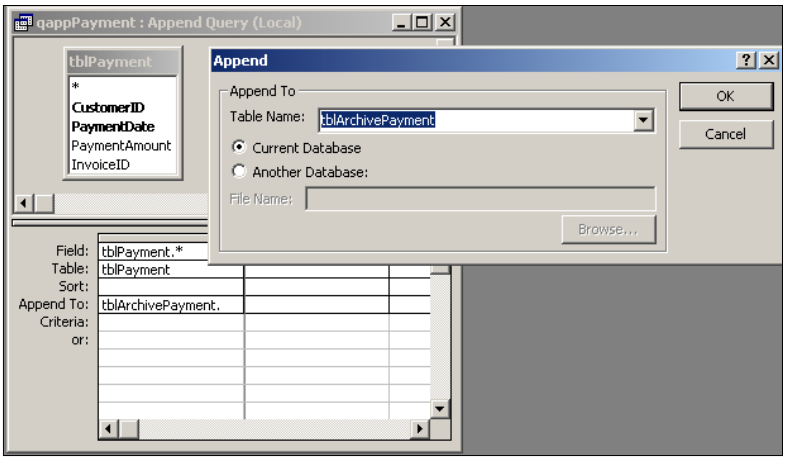
(Figure 8)

- Save the query as **QupdPostPayment**

(3) **QappPayment**; This query append the records from the payment table (tblPayment) to the archive file (tblArchivePayment)

Create this query by:

- Start a new query in Design View and add **tblPayment** to the query window.
- Double click the asterisk (*) that appear at the top of the field list in **tblPayment**. This adds all the fields in the table to the query.
- From the main menu select Query, Append Query. A box will appear asking you to enter the name of the table you table you want to append to. Type tblArchivePayment (as shown in Figure 9) and click OK.



(Figure 9)

- Save this query as **QappPayment**

(4) **QdelPayment;** This query deletes all the records from the payments table.

Create this query by:

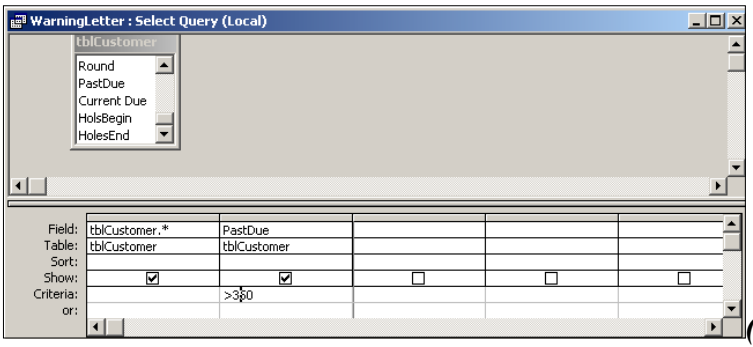
- Start a new query in Design View and add **tblPayment** to the query window.
- From the main menu select Query, Delete Query.
- Save the query as **QdelPayment**

(2) Warning System

This system allows the receptionist to print out warning letters you customer who have PastDue amount over the 350 pounds limit. The query is designed in the following steps:

(a) First you have to create a query which stores all customer who have PastDue amount above 350 pounds. Create this query by:

- Start a new query in Design View and add **tblCustomer** to the query window.
- Double click the asterisk (*) that appear at the top of the field list in **tblCustomer**. Also add the **PastDue** field to the query grid.
- Enter >350 in the **Criteria** row as shown below in (Figure 10)



(Figure 10)

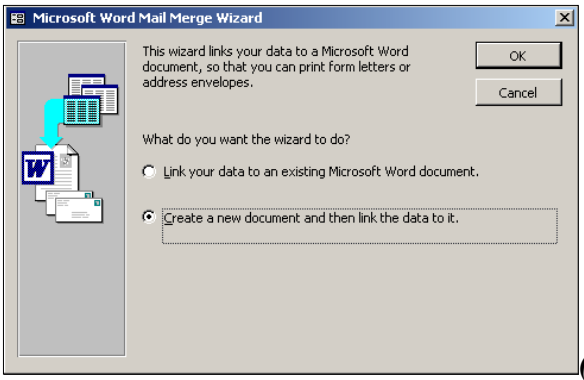
- Save the query as **WarningLet**

(b) The next step is to create a mail merge letter in Microsoft Word to send to the selected customer who have over PastDue amount. This is done by:

- In the Database window select the Queries tab and select **WarningLet**.
- Click the arrow on the Office Links button to display the list of options, and select Merge It (Figure 11)

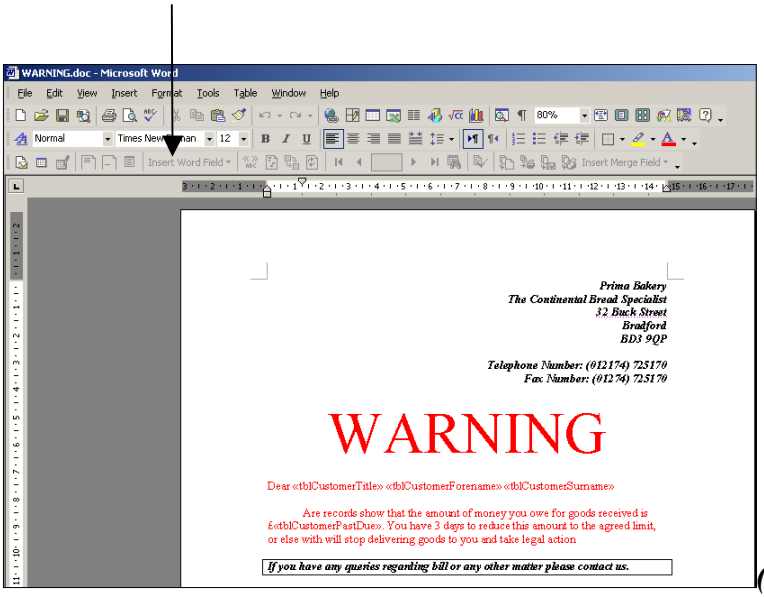
(Figure 11)

- The Microsoft Word Mail Merge Wizard Opens (Figure 12), and select the option **Create a new document and then link the data to it**. Click Ok



(Figure 12)

- Word open in mail merge mode. Now you create a standard warning letter as shown below (Figure 13), inserting the relevant field by clicking on the **Insert Merge Field** Button.



(Figure 12)

- Then save the document in the same folder as Prima database application. And close word.

CHAPTER 8– TESTING

This chapter looks at the testing phase in the methodology, the purpose of this chapter is to determine whether the system works and if the end users are happy with the new system.

8.1 Introduction

Their will be three types of testing carried out on the system and these will be unit testing, integration testing and user acceptance testing. Unit testing involves testing out every component of the system to look for the existence or errors. The purpose of this test is to insure that once the system is deployment in real life then no errors occur causing the system to fail. Integration testing verifies that the components of the system work as described in the design specification ensuring the system is ready for the final stage of testing which is user acceptance. User acceptance testing involves participation of the users in operating the system to check that all the requirements have been met. At this stage any iteration will be performed to the new system with regards to user feedback as stated in the project management section.

Unit and integration testing is know as program level view of testing, and user acceptance testing is known as system level view of testing. The summary of results for each type of test is shown below with the test plans and results located in appendix H.

8.2 Unit Testing

Unit testing is performed on individual parts of the system in order to test whether it functions properly are specified. We will be testing the software for errors, validating the correctness of data input and checking integrity rules are adhered to. Many of the components would have been tested in the implementation stage so we expect minimal problems to occur at this stage. The test plan looks at all aspects of the system and tries to replicate tasks done in real life to see if any errors occur.

8.2.1 Summary Of Results

Overall the results of the unit testing were successful with all forms being tested and no major problems have been discovered. These results were expected as most of the components of the system had been thoroughly tested in the implementation stage. The test plan and results can be seen in appendix H.

8.3 Integration Testing

Integration testing is performed to discover any errors with the main interface of the system and links. The testing involves testing each one of the command buttons and whether the expected outcome is achieved.

8.3.1 Summary Of Results

Overall the results of the integration testing were successful with a few links not working to the specified tasks. The first fault occurred when testing the link between the main user interface and the payment menu. When clicking on the '*Payment Menu*' command button this resulted in the user being transferred to the '*Customer menu*'. This problem was easily rectified by changing the macro design to open payment menu instead on customer menu. The second fault occurred when testing the function button '*Add new customer*'. On the form appears an exit function button with should close the form and open the customer menu form, but the flaw with this button was that it did not do anything when clicked on. The problem occurred because the button

was not linked to any macro and this was quickly solved by producing a new macro which saved and closed the '*Add new customer*' form and open the '*Customer menu*' form.

8.4 User Acceptance Testing

User acceptance testing involves an end user testing the system and seeing it performs to satisfy the needs of the users. The testing plan was conducted using a set of pre-determined functions that the users currently performs on the paper based system. Also some other functions were added to this list with respect to new features added to the system e.g. payment function. Halina Brown was selected to carry out the user acceptance testing and the testing results and any feedback is shown below:

8.4.1 Summary Of Results

The end user was satisfied with the way that the system performed with the features meeting her needs, but found the process of performing some tasks hard due to her being unfamiliar with the system. The user was impressed by the automated features which drastically reduced the time involved in completing tasks and also was happy with the layout of the forms. But one of the iteration that the user requested was the specific design of the error message in the customer form for each field. Currently just standard error messages appear when invalid data inputted, and so the user would like specially design error message which state what the problem is and the type of data needed to be inputted into the field.

TEST	***FEEDBACK***
Can you open the database, and does the main menu interface appear	Yes, the user can access the system and the main menu interfaces appears when opened
Filing in the order form	Impressed by the resemblance to the current paper based order form and happy with location of function box.
Is the order form easy to navigate round and simple to understand?	The user agreed that the order form is simple to navigate round, with its clearly defined sections
Can you navigate yourself to the customer menu	Yes, the user found the main menu interface easy to locate

When adding data regarding customer, did invalid data formats result in errors messages appearing	Yes errors message did appear but only the standard format. The user requested that the messages should be designed specifically for each field, defining what data has to be entered into the field
Can the user easily select drop down menu lists e.g. Titles , in the customer form	Yes, and the user was impressed at the drop down lists as it reduces space needed on form.
Can you edit customer details?	Yes, easy to search for customer details using the CustomerID search function.
Can you edit vendor details?	Yes, again the user was impressed with the ease to search for vendor details using the vendor name search function.
Can you edit order details?	Yes, again the user was impressed with the ease to search for a specific order as currently in the paper based system this would require her to search through the whole book.
Can you easily print of daily order forms?	User found this function easily, as she only has to press one command button and all order forms are printed.

CHAPTER 9 – EVALUATION

This chapter evaluates the new system build for the purpose of replacing the current paper based ordering system used at Prima Continental Bakery. I will be evaluating the system using the criteria set out in the requirements specification which include the objectives and requirements. We will be using this criteria as it is the best way to evaluate whether the system meets the end users needs and expectations. The criteria will be marked using a scale of 0-2 where:

0 = Requirement/Objective not Met
1 = Evidence of an attempt to meet the requirement/objective
2 = Requirement/Objective met

Using this scale it can be clearly defined whether the requirement/objectives have been met. Below I have specified the objectives and requirements of the system and whether I have met the criteria specified by the users:

9.1 Objectives

The objectives of the project are:
(1) Perform analysis of current system and gather user requirements
Score = 2
(2) Design and implement a database to add, hold and view data
Score = 2
(3) Provide a payment function which archives customer's payments
Score = 2

9.2 Minimum Requirements

The minimum requirements set for this project are:
(1) A specification of the bakery requirements
Score = 2
(2) A database management system
Score = 2
(3) Produce agreed reports after discusses with users (Appendix I)
Score = 2
(4) A payment function
Score = 2
(5) Warning letters Function (Mail Merge With Microsoft Word) (Appendix I)
Score = 2

9.2.1 Possible Extensions

Possible extensions for this project:
(6) User manual – For new workers (Appendix J)
Score = 2
(7) Vendor Ordering System
Score = 2
(8) Stock System
Score = 2
(9) Produce document relating to security aspects and advantages and disadvantages of having web based system.
Score = 0

Overall the results of the evaluation show that all the objectives and most of the requirements have been completed in this project.

9.3 Evaluation against the paper-based system

The overall aim of this project was '*To provide a system that is more efficient and effective then the current paper based system*'. And I feel that the new system produced does succeed in accomplishing the main aim of the project. Using this aim we can now try to compare two systems and back up my claim that that the new comprised system is more efficient and effective then the paper based system. Two points raised below clearly show that the new system is more efficient and effective:

- The time is takes for searching for a specific Order form has been greatly reduced due to all the forms being stored in the order table. Search tools have enabled the users to quickly search an order form using CustomerID or Shop Name.
- The weekly invoices are produced in minutes due to automated forms and macros. In the previous paper based system this process took around 7 hours for the manger to do.

9.4 Improvements To The System

The possible extensions to the system that could be made are limited due to the fact that the organisation is a small company with limited resources. But if the organisation grew over the next couple of years and expand its network and the following improvements could be made to the system:

- Create links with vendors ordering system, which enables you to automatically make order for raw materials through the internet.
- Create an online ordering form, which allows customer to place orders online.

References

Avison, D.E. and Shah, H.U. (1997). **The information Systems Development Life Cycle: A First Course in Information System**. Berkshire: McGraw – Hill Book Company Europe.

The Standish Group (1995) Chaos: **The Standish Group Report**, available as http://www.scs.carleton.ca

Yeates, D. (ed) (1986) **Systems Project Management**. London: Pitman Publishing Ltd

Johnson, O. (2003) IS21 **Object Oriented Analysis and Design Lecture Notes**. School of Computing, University Of Leeds

Bennett, S., McRobb, S. and Farmer R. (1999) **Object-Oriented Systems Analysis and Design using UML**. Berkshire, England : McGraw-Hill Publishing Company

Weiss, E. H. (1985). **How to Write a Usable Manual**. Philadelphia: ISI Press

Elmasri, R. and Navathe, S. (2000). **Fundamentals of Database Systems**. Third Edition. Reading, Massachusetts : Addison-Wesley

MrFadden F.R, Hoffer J.A, Presscott M.B. (1999). **Modern Database Management.** 5th Edition. England: Addision-Wesley

Stevens, p. with Pooley, R. (2000), **Using UML Software engineering with objects and components**. Updated Edition. Harlow, England : Addision-Wesley, Pearson Education Limited.

Johnson, O. (2003) IS21 **Object Oriented Analysis and Design Lecture Notes**. School of Computing, University Of Leeds

Roberts, S. (2004) DB21 **Object Oriented Analysis and Design Lecture Notes**. School of Computing, University Of Leeds

Jenkins, AMJ. (2002) DB11 **Intro to Databases Notes**. School of Computing, University Of Leeds

Curtis .G., Cobham .D. (2002) **Business information systems.** Fourth Edition, Harlow : Financial Times/Prentice Hall