RECOGNISING ACHIEVEMENT

Exemplar Candidate Work

GCE in Applied ICT

OCR Advanced GCE in Applied ICT: H715

Unit G056: Program design, production & testing

Contents

Introduction

This exemplar material serves as a general guide. It provides the following benefits to a teacher:

- Gives teachers an appreciation of the variety of work that can be produced for this unit
- Shows how the mark scheme has been applied by a senior assessor

It is important to make the point that the teacher support materials play a secondary role to the Specification itself. The Specification is the document on which assessment is based and specifies what content and skills need to be covered in delivering the course. At all times, therefore, this teacher support should be read in conjunction with the Specification. If clarification on a particular point is sought then that clarification should be found in the Specification itself.

Moderator's Commentary: G056 Program design, production and testing

Total mark for portfolio: 45 (Max. 50)

This portfolio illustrates the work of a candidate who has designed a program specification to computerise a charity's gift aid system. This will work like a database system. They have looked at the needs of the user and made careful consideration of these needs whilst designing the system. The candidate has created a modular program based on their design notes. There is an analysis of the design methods used as well as evidence of all the necessary documentation, test plan and testing and a program review with an evaluation of the candidate's performance.

This appears to be a brief for a real end user as verified by the witness statement. The candidate has included a contents page and a signed and dated witness statement for the running/testing of the program. The portfolio is numbered electronically throughout with the contents page outlining where each task starts.

Task a (i)

Task a(i) and a(ii) are included within the section labelled task A. This covers pages 5 – 31. The candidate has spent some time finding out exactly what the user currently does and how they see the computerised version working. This includes collecting resources from the current system and collecting answers to pre-written questions about the system. The candidate then goes on to create data flow diagrams (DFDs) of the current system and analyses the problems with the current system. Whilst this may not always be necessary, or possible, it shows good practice for candidates to fully understand the problem they need to solve.

The user requirements for the program and the actual program specification are then started. The candidate details how the user requirements will be met, including some indication of the methods that will be used to input data and then considers any constraints or limitations. These are both detailed in nature whilst also being concise and easy to follow. The candidate then goes on to list inputs, processes, storage requirements and outputs in a table. This includes an explanation of the methods of output to be used. An entity relationship diagram is also created to show how the entities interact/relate to each other. Overall there is sufficient evidence for a mark at the top of mark band 3.

Mark Band 3 **Mark Awarded: 6 (Max. 6)**

Task a (ii)

As indicated above, the candidate has included task a(ii) within the section labelled task A work. Task a (ii) can be found mainly in pages 29 -31. The candidate has clearly shown how the user requirements will be met by the program by explaining how the appropriate inputs, outputs or processing will be implemented to meet each part of the requirements. The planned program appears to be complex in nature.

A mark in mark band 3 is appropriate overall. However, greater depth is required in the explanations of how all the implementations will cover the user needs for a mark at the top of the mark band.

Mark Band 3 **Mark Awarded: 5 (Max. 6)**

Task b (i)

The candidate has created an easy to follow systems flowchart and an entity relationship diagram to show how the proposed tables all relate to each other. A data dictionary has also been created to explain how the data types are to be set up, including any validation required. Different input forms and output reports have also been designed and a flowchart to show how the forms link together with the menu screens. The candidate has given a clear set of event actions explaining in detail exactly what steps will be executed from each form. A complete user interface has been designed to include the different input forms and reports. Some pseudo-code has also been given to explain the validation rules to be used. An action plan has also been included on pages 53 and 54 .

The information provided is clear and easy to follow. A mark in mark band 3 is appropriate. However, there is insufficient evidence of file structure and organisation for a mark at the top of the mark band.

Mark Band 3 **Mark Awarded: 7 (Max. 8)**

Task b (ii)

The candidate has included the evidence for task b(ii) in task B on pages 51-52. The candidate has systematically gone through each of their design methods and explained how appropriate they were, they have then given strengths and weaknesses of each and suggested some changes that could be made covering the requirements of mark band 3. A little more depth in criticising the methods used and their weaknesses would be more appropriate for the top of mark band 3, so a mark of 5 was awarded.

Mark Band 3 **Mark Awarded: 5 (Max. 6)**

Task c

The candidate has printed code listings with headings to indicate what each part does. Comments are also included within the program code. The program is extensive and sufficiently commented although a little more explanation could be included. A table outlining the main sub-routines used, their location, purpose and how they have been utilised is good practice and shows a good understanding of the program, as is the table of variables with their validation explained. Finally, the candidate has given an explanation of what security has been set up within the system and the VB functions that have been used. The pages of tables help readers to better understand the code provided. As a result, there is sufficient evidence to award the maximum mark.

Mark Band 3 **Mark Awarded: 9 (Max. 9)**

Task d

Testing has been planned and carried out efficiently. Test tables have been created for each separate form. Different data has been entered and thorough testing has been carried out. Both screenshots and printouts have been used as evidence of testing having taken place as well as entries on the witness statement. Where errors have occurred, evidence provided shows that corrections were made and retesting shows the process working successfully. This is an example of good, well planned, thorough testing and a maximum mark is justified.

Mark Band 3 **Mark Awarded: 6 (Max. 6)**

Task e

A full analysis of the system has been carried out giving strengths and weaknesses of each part. User feedback is also included within this. The candidate's own performance has been evaluated in a similar way. Whilst the evaluations are in some ways critical, they lack the depth required for a full critical analysis. Often parts are glossed over instead of going into more depth as to why problems occurred and how they could be improved upon. There is some evidence of mark band 3 but without greater depth, a mark at the bottom of the mark band is most appropriate.

Mark Band 3 **Mark Awarded: 7 (Max. 9)**

Total mark for portfolio: 45 (Max. 50)

G056 Assessment Evidence Grid

Unit G056: Program design, production and testing

What candidates need to do:

Candidates will produce: a working program with complete documentation to meet a given user requirement.
Evidence needs to include:

a: [AO1/3] a program specification to meet the given requirement with a description of how their specification meets the program requirements and how they have considered the user's needs [**12**];

b: [AO2/3] a program design arising from their specification and an analysis of the design methods they have used [**14**];

c: [AO1] an annotated modular program to realise the design, which must include at least **one** data structure, all data types, all control structures and all appropriate operators listed in the programming section [**9**];

d: [AO4] test documentation including a test plan with valid, invalid and boundary data, expected results, actual results and changes identified as a result of testing [**6**];

e: [AO4] a program review and evaluation report including an evaluation of their own performance [**9**].

How the candidate will be assessed:

Task	Assessment Objective	Mark Band 1	Mark Band 2	Mark Band 3	Mark Awarded
a(i)	**AO1**	The candidate shows that they have developed their skills by developing a specification which identifies some inputs, outputs and processing requirements for the given program requirement; **[0 1 2]**	The candidate shows that they have extended their skills by developing a specification which identifies most inputs, outputs and processing requirements for the given program requirement; **[3 4]**	The candidate shows that they have used their initiative to extend and enhance their skills by developing a clear and full specification which identifies all inputs, outputs and processing requirements for the given program requirement. **[5 6]**	**6/6**
a(ii)	**AO3**	The candidate applies their knowledge and skills to briefly describe how their specification meets the requirements of a straightforward problem, considering the user's needs; **[0 1 2]**	The candidate applies their knowledge and skills to describe how their specification meets the requirements of a complex problem and how they have identified the users' needs; **[3 4]**	The candidate applies their knowledge and skills to fully and clearly describe how their specification meets the requirements of a complex problem and fully considers how the user's needs will be met. **[5 6]**	**5/6**
b(i)	**AO2**	The candidate demonstrates knowledge of design techniques by using some appropriate techniques, such as pseudocode, flowcharts, event-action charts, to design processes; The candidate designs input screens, output formats, validation and verification, data structures and at least one file structure; **[0 1 2 3]**	The candidate demonstrates knowledge of different design techniques by using a range of appropriate techniques such as pseudocode, flowcharts, event-action charts, to design processes; The candidate's designs are accurate and cover the whole program (input, output, processes, data structures, all file structures and file organisation); **[4 5 6]**	The candidate demonstrates thorough, detailed knowledge of formal and informal design techniques by using a structured design method and a wide range of appropriate techniques, such as pseudo code, flowcharts, event-action charts; The candidate's designs are accurate, clear and complete and cover the whole program (input, output, processes, data structures, all file structures and file organisation). **[7 8]**	**7/8**

Unit G056: Program design, production and testing (continued)					
Task	**Assessment Objective**	**Mark Band 1**	**Mark Band 2**	**Mark Band 3**	**Mark Awarded**
b(ii)	**AO3**	The candidate applies their knowledge and skills to comment on the appropriateness of the design methods they used and identifies areas for improvement; **[0 1 2]**	The candidate applies their knowledge and skills to analyse the appropriateness of the design methods they used by describing strengths and weaknesses and suggesting improvements; **[3 4]**	The candidate applies their knowledge and skills to analyse the appropriateness and effectiveness of the design methods they used by describing strengths and weaknesses and showing how they have modified their design methods to address the identified weaknesses. **[5 6]**	**5/6**
c	**AO1**	The candidate shows that they have developed their skills by producing a program from their specification and design; The candidate's program meets most of the original requirements; **[0 1 2 3]**	The candidate shows that they have extended their skills by producing a fully working program from their specification and design; The candidate's program is modular, meets most of the original requirements and is easy to use; **[4 5 6]**	The candidate shows that they have used their initiative to extend and enhance their skills by producing a fully working program with clear and fluent annotation; The candidate's program is modular, meets all original requirements, is easy to use and makes full use of all appropriate data structures, data types, control structures and operators. **[7 8 9]**	**9/9**
d	**AO4**	The candidate produces a test plan and documents test results that cover all data validation; **[0 1 2]**	The candidate produces a test plan with valid, invalid and boundary data and documents test results to cover all eventualities; **[3 4]**	The candidate produces a test plan that covers all paths and user operations as well as all valid, invalid and boundary data, documenting test results to cover all eventualities and using the results to refine the solution. **[5 6]**	**6/6**
e	**AO4**	The candidate comments on the effectiveness of their solution and identifies at least one improvement that they could make; The candidate comments on their actions and role in solving the problem and identifies areas for improvement; The candidate's report may contain errors in spelling, punctuation and grammar; **[0 1 2 3]**	The candidate comments on the effectiveness of their solution by identifying strengths and weaknesses and by considering the problems found during testing; The candidate comments on how they could have reduced testing errors by changes to their design; The candidate includes an analysis of their own performance by identifying strengths and weaknesses, with some suggestions for improvement to the overall process; The candidate's report contains few spelling, punctuation and grammar errors; **[4 5 6]**	The candidate provides a critical analysis of their solution, taking account of user feedback, to identify the strengths and weaknesses; The candidate explains refinements that could be made to the solution as a result of their analysis; The candidate includes an analysis on their own performance by identifying strengths and weaknesses and uses this analysis to show how they will address these issues to be more effective in the future; The candidate's report is consistently well-structured and there will be few, if any, spelling, punctuation and grammar errors. **[7 8 9]**	**7/9**

Candidate’s work

Applied ICT – Witness Statement

Unit 17 – Program Design, Production and Testing

Candidate Name: [ ] **Candidate Number:** [ ]

This is to certify that the candidate has designed, written and tested a computer program which meets the given requirements specification for an end-user.

I confirm that I have used the program, that it runs successfully, and demonstrates the following features, as described in the candidate's documentation.

☑ The program starts up successfully and provides a main menu, or login screen

Login screen provides for different levels of access, (manager/cashier/etc.)

☑ Allows for successful navigation through the program

well-designed transitions between forms to complete processes required.

☑ Consistent layout of forms

Use of B Logo throughout, Simple colour scheme, font, etc.

☑ Data records are saved to file

files for 'Cashiers', 'Transactions', 'Customers', 'Tickets',

☑ Data records are loaded into the program

All records available when required, eg. Cashier – check login/change password.

☑ The program includes the validation of data input

Comprehensive data validation – eg. postcode/house no./Name/etc.

☑ Error messages are generated where appropriate

Many error msg's (eg. for invalid data entry, 'No matching records', etc.)

☑ Data items are updated by the program correctly

eg. Login passwords, Transaction details, Report of cashier performance. etc

An excellent program for a real end-user.

Assessor Name:

Assessor Signature **Date: 05/05/2010**

Unit 17

Unit GO56- Program Design, Production and Testing

CONTENTS

TASK A

ANALYSIS

Limited is a tourist attraction situated on the in the National Park. It holds the National Museum, and Gardens. The whole estate is owned by and his family.

As visitors enter the complex they are asked if they would become a Gift Aid visitor. This means that the organisation, can claim back the tax on the entrance fee. This is possible as the Museum is a charity run organisation. To be able to participate in the Gift Aid the visitor must be a UK tax payer, therefore they must record their details to prove this to HM Revenue and customs.

The current system involves each customer filling out a large form firstly with their details to reclaim the tax, and then they must sign a separate form to gain the free pass for 12 months which must be offered to Gift Aid visitors by law. These forms then go through lengthy processes of validation and verification until they are finally entered manually into a spreadsheet. Often forms are unreadable due to a wide range in handwriting, because of this they have to be discarded and the museum is then unable to reclaim the tax on those transactions. With an average of 1000 customers a day there are a large number of forms to process.

I propose to set up a computerised system that will allow the cashier to enter the customers name, postcode and house number. This will then be used to search for the full address which will then be checked by the customer and saved in a large data file. I then propose that the names and addresses of customers are stored in the system so that they can return for free within the next twelve months by giving their surname and postcode on return, this will hopefully reduce the need for extensive amounts of verification and data duplication and also reduce the need for storing large amounts of paper based documents.

problem clearly explained

The main contact with the organisation will be the main supervisor in charge of visitor admissions.

Investigation

The methods that I am going to use to investigate the problem will be by using questionnaires, observation and interviews. I will also look at the current system used.

I will conduct the interview with the supervisor in charge of the visitor reception which deals with the Gift Aid collection and validation. The questions that I will use in the interview will be;

1. *What does the current Gift Aid system do?*
   This will help me to realise the main purpose of the current system, which will help when creating the objectives of the new system.

2. *Do you feel that the current system is effective in both cost and time?*
   This will help to identify whether the staff feel that there is a need to update or adapt the Gift Aid system. I expect to find that the employees find the system ineffective, laborious and time consuming.

3. *On average how long do you think it takes to process each Gift Aid form from the customer filling in their details to it being stamped on entry to the database?*
   This will show how long it takes using the current system and will later help me when calculating a cost benefit analysis. I expect to find that the time taken will be quite high as there is a lot of data duplication.

4. *How many employees work on Gift Aid each day and what are their roles?*
   This will show how many employees the new system will affect and will also help to show how the new system could be structured to suit the many different roles. I expect to find that a number of employees are working on the Gift Aid at the same time in a number of different ways.

5. *Currently what security measures do you have when storing customer's details?*
   This will help when determining the more specific requirements of the system, as customer's details are being held for tax purposes there will probably be very specific security guidelines.

6. *In what ways, if any, do you think that the Gift Aid system could be improved?*
   This will identify the ways in which the staff involved with the system feels that it could be improved to help their work. I believe that this will return a variety of responses depending on the tasks that different employees carry out.

7. *What percentage of customers currently Gift Aid?*
   This will identify the number of times that the system will be used each day. I expect to find that about half the customers currently gift aid.

8. *On average how much does Gift Aiding currently cost the company each month?*
   This will show the amount of money that the company currently spends and will help when trying to reduce current costs. I expect that this value will be very high as they are currently employing several members of staff to input/ validate the same data.

9. *If a computerised system was implemented do you think that the current employees would be able to cope with a computerised system?*
   This will help to identify how much, if any training will be needed when the new system is implemented this will help when calculating overall cost. I expect to find that some staff will need a large amount of training whilst others will feel confident using a computer already so will only need the training with the new system.

10. *Would you personally feel confident enough to be in charge of a computerised system?*
    This again will help with calculating the cost. I expect to find that the supervisors will require some training to build their confidence.

✓

Comprehensive investigation

Interview Transcript

Below is the interview response from the main Visitor Reception supervisor.

1. What does the current Gift Aid system do?

The current system allows the National Museum Trust Ltd to claim back 25p for every £1.00 the customer gives in admission from the H.M Revenue and Customs. In return they receive a free annual pass to the Museum.

2. Do you feel that the current system is effective in both cost and time?

I believe our current system is not cost effective in the sense too many people are involved in each transaction. This means the time spent on each Gift Aid form is far too long and is costing more.

3. On average how long do you think it takes to process each Gift Aid form from the customer filling in their details to it being stamped on entry to the database?

The Gift Aid form is worked on for about 15 minutes from the customer filling it out and it going into the vault.

4. How many employees work on Gift Aid each day and what are their roles?

1 Gift Aider asking customers to fill out the form.
1 Cashier checking forms and attaching receipts.
1 Cashier at the end of the day checking amounts and working out %.
1 Person in the cash office working out daily amounts.
1 Person putting information into the computer.
1 Person putting the Gift Aid forms into the vault.

5. Currently what security measures do you have when storing customer's details?

We have passwords on our computers so only authorised people can access data. All forms are stored in our vault for 5 years and then destroyed.

6. In what ways, if any, do you think that the Gift Aid system could be improved?

By having an up to date till/computer system. Then instead of large paper

passes we could have a credit card type pass with bar code/ signature.

7. What percentage of customers currently Gift Aid?

During this month (August) we have had between 65% and 82% of customers Gift Aiding per day on average it is about 66%.

8. On average how much does Gift Aiding currently cost the company each month?

The company has to pay for;
Posters advertising Gift Aid.
Gift Aid pads
Gift Aid question and answer sheets
Donation Pads
Rubber Stamps x7 (regularly broken).
Wages for staff working on the Gift Aid system.

9. If a computerised system was implemented do you think that the current employees would be able to cope with a computerised system?

With training yes, however I would expect some teething problems.

10. Would you personally feel confident enough to be in charge of a computerised system?

With training yes, however again I would expect some teething problems.

✓ Excellent analysis of problem

Interview Analysis

1. Question one allows me to see what the current Gift Aid system does and has shown me that I will need to include a calculation to work out the amount that can be reclaimed from the admission prices.

2. This response shows that the supervisor feels that there is a problem with the number of people that work on the Gift Aid, therefore I will need to address this in my design so that both money and time can be saved.

3. Question three also highlights the same issue as is in question two as the Gift Aid form is worked on for a long time.

4. The response given in four highlights the several job roles needed within the Gift Aid system. This shows the large amounts of labour needed to complete the system and again this is very costly.

5. Whilst there are some security measures in place, these could be improved greatly.

6. The supervisor has suggested the use of a computerised system this will help when deciding the type of system that I will implement.

7. With a current average Gift Aid percentage of 66% this could be used after the implementation of the system to see how many more customers will Gift Aid with a faster system.

8. This question identifies the main costs of the Gift Aid system, when designing the system I will be able to look at these costs and will be able to create a system to try to reduce the number of these needed.

9. Question 9 will help when planning the implementation of the system as it identifies the training that staff will need.

10. Question 10 will help when planning the implementation of the system as it identifies the training that staff will need.

Questions for Questionnaire ✓ 2nd method of investigation used.

I will ask the employees who validate and collect the forms from the public to fill out a questionnaire, the questions that I will use in the questionnaire will be;

	Question	Reason for Asking	Type of Question	Answer Options
1	How often do you use the Gift Aid system?	This should help to identify the amount of experience that they have with the system. I expect to find a mix as I am going to give the questionnaire to a range of part and full time staff.	Tick Boxes	- Everyday - Once/twice a week - A few times a month
2	On a scale of one to ten how effective do you find the current system?	This should show how the employees feel about the current system and will help to identify if they feel the need for a new one.	Scale	1 to 10 (1 lowest- 10 highest)
3	What are the three main reasons why customers don't currently Gift Aid?	This will help to identify whether the customers feel the need for a faster system. I expect that the long form is the main reason for not Gift Aiding.	Line Spaces	3 statements
4	On average how long do you think it takes you to process each Gift Aid form?	This will show how much time the Gift Aid system uses	Tick Boxes	- Less than 1 min - 1 - 2 mins - 3 - 4 mins - More than 5 mins
5	Do you often find errors on the Gift Aid forms when validating them? (E.g. No Town/ Postcode)	This will help when identifying the ways in which the current system can be improved. I expect to find that the employees often find errors.	Tick Boxes	- All of the time - Sometimes - Never
6	How proficient are you using a computer?	This will show the level of training that staff will need once a new system has been implemented.	Tick Boxes	- Very - Enough for basic tasks - Barely - Not at all
7	If a computerised Gift Aid system was implemented would you feel confident using it, with appropriate training?	This will help when identifying the amount of training needed when the system is implemented. I expect to find that most of the employees will feel confident as many of them are students so will already use computers on a daily basis.	Tick Boxes	- Yes - No - Unsure
8	Do you have any other comments about the way the current Gift Aid system could be improved?	This will help when planning the new system and will show what the employees who are using the system feel should be happening.	Line Spaces	Line spaces for statements.

Questionnaire

1. How often do you use the Gift Aid system?

   □ Everyday
   □ Once/ twice a week
   □ A few times a month

2. On a scale of 1 to 10 how effective do you find the current system?
   (1 lowest - 10 highest)

1	2	3	4	5	6	7	8	9	10

3. What are the three main reasons why customers don't currently Gift Aid?

   1. _____
   2. _____
   3. _____

4. On average how long do you think it takes you to process each Gift Aid form?

Less than 1 min	
1 - 2 mins	
3 - 4 mins	
More than 5 mins	

5. Do you often find errors on the Gift Aid forms when validating them? (E.g. No Town/ Postcode)

   □ All of the time
   □ Sometimes
   □ Never

6. How proficient are you using a computer?

Very	
Enough for basic tasks	
Barely	
Not at all	

7. If a computerised Gift Aid system was implemented would you feel confident using it, with appropriate training?

   □ Yes
   □ No
   □ Unsure

8. Do you have any other comments about the way the current Gift Aid system could be improved?

   _____
   _____
   _____
   _____

Questionnaire Analysis

The questionnaire was successfully completed by 5 employees who collect the Gift Aid forms from the customers. The results were as follows;

- *Question 1* - How often do you use the Gift Aid system?

Every day	2
Once/twice a week	2
A few times a month	1

This shows that the employees questioned where generally experienced using the system, it also shows that the new system will be operational every day.

- *Question 2* - On a scale of 1 to 10 how effective do you find the current system?

1	2	3	4	5	6	7	8	9	10
	II	II	I						

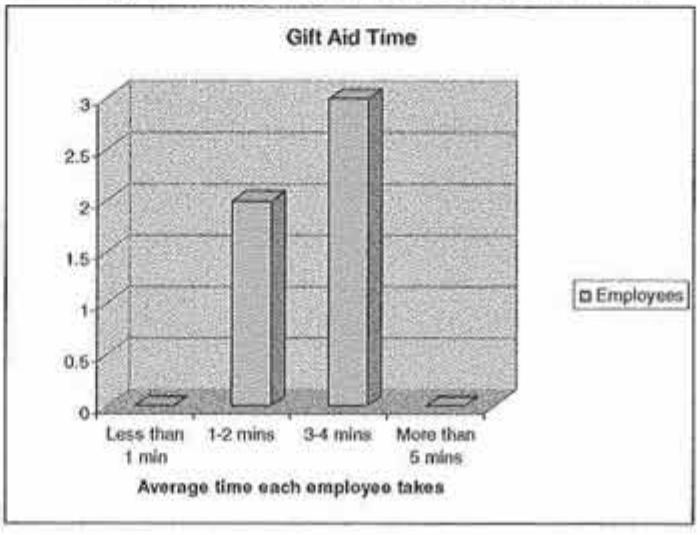
This shows that the employees aren't satisfied with the current system and feel that it is in effective and that there could be room for improvement.

- *Question 3* - What are the three main reasons why customers don't currently Gift Aid?

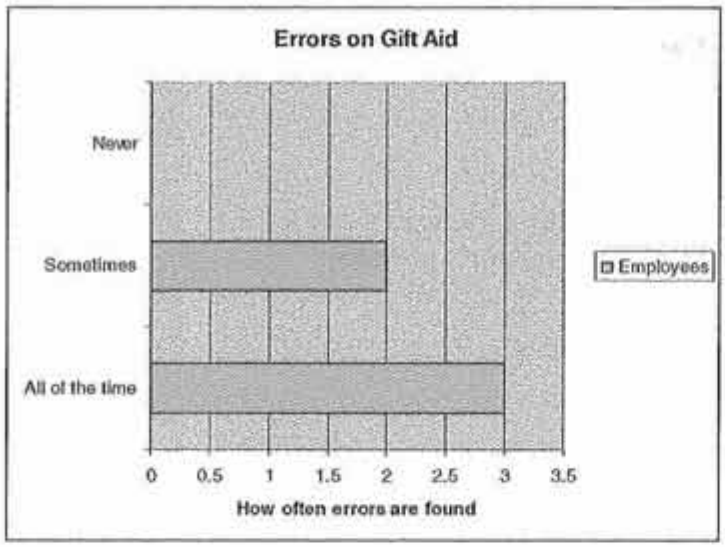
The main three reasons were that the customer;
has the wrong ticket,
can't be bothered,
form is too long.
This shows that more customers might Gift Aid if a new system were implemented which made it faster and also if they didn't have to fill out a long form.

- *Question 4* - On average how long does it take you to process each Gift Aid form?

This shows that on average the employees take about 3 minutes to complete each Gift Aid form when dealing with the customers.



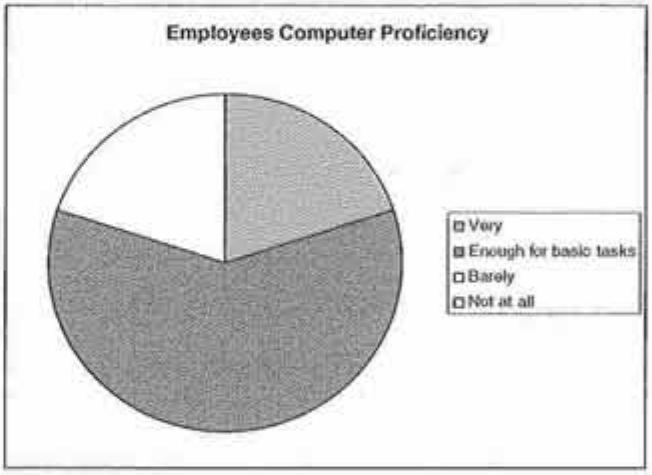
- *Question 5* - Do you often find errors on the Gift Aid forms when validating them?



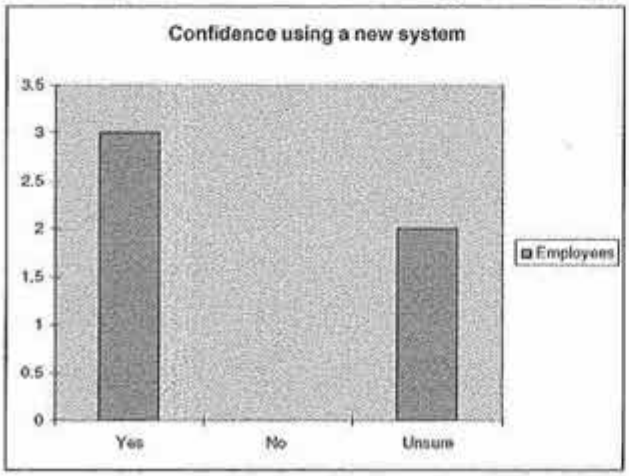
This shows that as expected the employees often find errors on the Gift Aid forms which could potentially have been avoided with a more efficient system.

- *Question 6* - How proficient are you using a computer?

This shows that most of the employees already have at least some basic skills on a computer which would indicate that a minimal training course will be required if a computerised system were to be implemented.



- *Question 7* - If a computerised Gift Aid system was implemented would you feel confident using it, with appropriate training?



This shows that most of the employees felt they would feel confident using a computer, those that said they would feel unsure may require some extra training and guidance to build their confidence.

- *Question 8* - Do you have any other comments about the way the current Gift Aid system could be improved?

  The three most popular responses were;
  It takes too long,
  Get rid of the long forms,
  Make it faster.

  This shows that the employees feel that there is a need for a new system that would speed up the admissions process.

Observations

✓ 3rd method of investigation

As I already have involvement with as part of the observation I have been able to carry out most of the tasks in the Gift Aid system.

Whilst working on the data collection part I found that some members of staff took a long time to validate the forms, they also seemed to get bored after validating often over 100 forms on a busy day. I also found that the customers forget or miss out important parts of the forms and unless the cashiers spotted it the forms were being taken with out vital data such as a postcode or signature. As well as checking the forms the cashiers also have to validate the free pass which again takes a lot of time.

When watching the tasks carried out in the cash office, several errors were identified on many of the forms. The tasks were only carried out for validation purposes ready for when the data is entered into the computer. Each price on the forms are entered into one long receipt, this is then subtotalled and totalled at the end of each day.

The data entry to the computer task was very laborious and it was hard to keep concentration when doing it, this inevitably increased the number of human errors. Here the forms are entered onto the computer. Firstly the price has to be checked to ensure it only includes motor museum tickets and then the details can be entered into the spreadsheet.

The name and postcode are entered and then the rest of the address is calculated from the postcode. If the address can't be found the postcode needs to be checked or found. This was either done by the employee trying to re-read the postcode on the form and trying other possible characters, if this isn't possible a web site is used to find the postcode by entering the whole address. However the web site used only allows 15 searches a day so if there is several inaccurate forms they can't be processed until the following day.

Then the price is entered into the spreadsheet and also checked off of the long receipt produced by the cash office. When a sub-total is reached on the receipt the prices for that day on the spreadsheet must be added up to ensure that they are the same, if not the employee must trace back until they find the incorrect price.

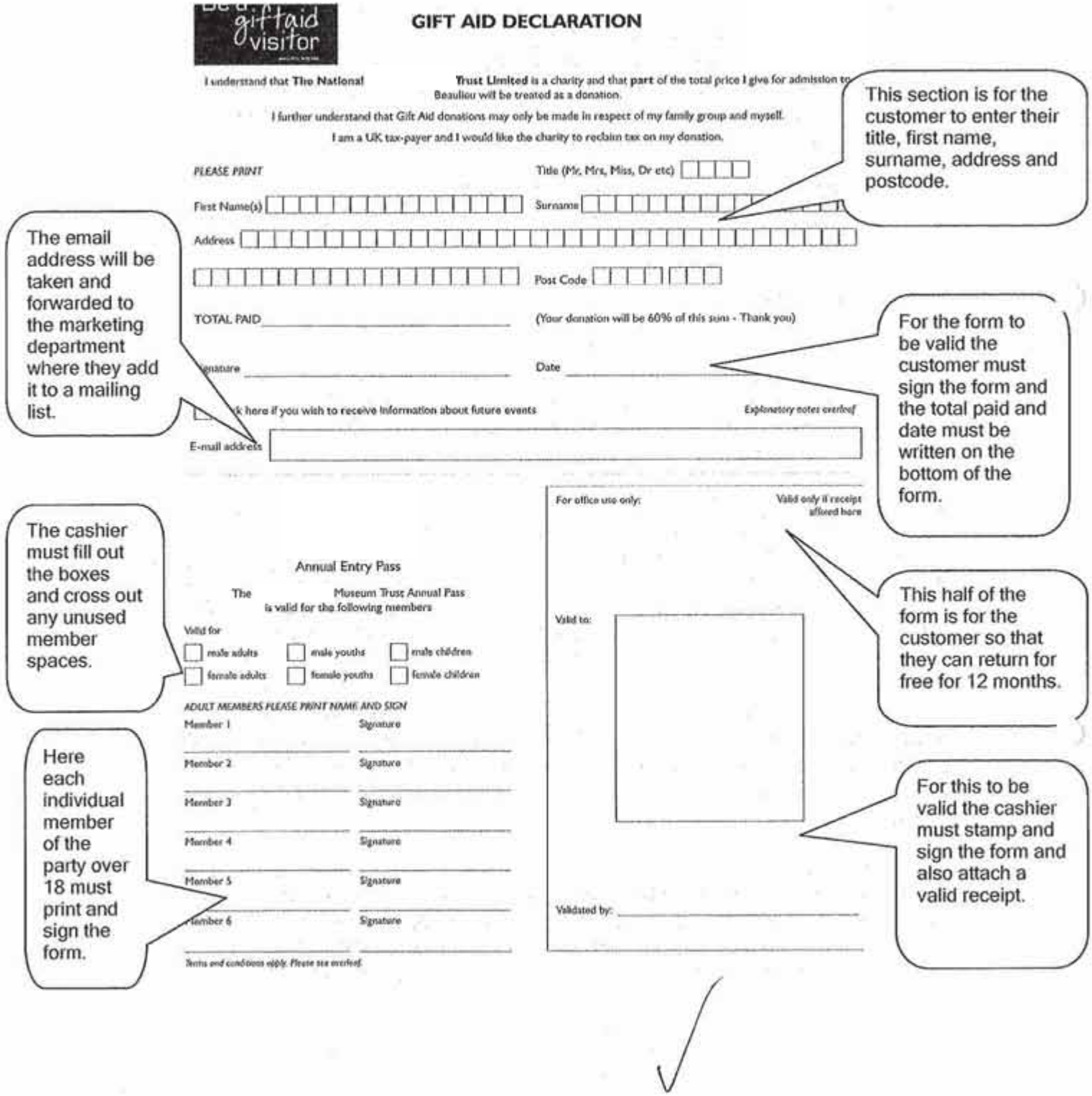
The form is then stamped with a unique number which corresponds with the key field used in the spreadsheet. Also with this process if a customer has added their e-mail address on the form then the address must be entered in another spreadsheet with their first name and surname so that they can be sent to the marketing department to add them to the mailing list.

Overall the processes that I observed seemed time consuming and often unnecessary as there was a lot of data duplication. The employees didn't seem motivated by the job and therefore more errors occurred. Several customers were discouraged to Gift Aid because of the size of the form.

✓

Forms Currently Used

Below is the front page of the first form, this is the main form filled out by both the customer and cashier. This shows some of the inputs that are currently put into the system.



This section is for the customer to enter their title, first name, surname, address and postcode.

The email address will be taken and forwarded to the marketing department where they add it to a mailing list.

For the form to be valid the customer must sign the form and the total paid and date must be written on the bottom of the form.

The cashier must fill out the boxes and cross out any unused member spaces.

This half of the form is for the customer so that they can return for free for 12 months.

Here each individual member of the party over 18 must print and sign the form.

For this to be valid the cashier must stamp and sign the form and also attach a valid receipt.

This is the reverse side of the previous form, all that is detailed here is the terms and conditions of both Gift Aiding and the Gift Aid pass. This is given to the customer and is an output of the current system.

Gift Aid - notes for donors

- You must expect in the current tax year to pay an amount of Income Tax or Capital Gains Tax at least equal to the tax the Charity reclaims on your donations
- If you pay tax at the higher rate, you can claim further tax relief on your Self Assessment tax return. Simply retain the Gift Aid Donations leaflet and your receipt as proof of your donation.
- 60% of the amount you hand over for admission to B will be accepted as a donation under Gift Aid. You need do nothing more – we will reclaim the tax you have already paid. There is no extra cost to you.
- By making a donation equivalent to our admission price you (and your family group) will not only receive free admission to the National Motor Museum for 364 days (excluding Beaulieu Special Events), but will be making a contribution to the work of the charity via the Gift Aid scheme.
- The information you give will remain confidential and will not be passed to any organisation other than HM Revenue and Customs.
- **If you would like to be kept informed of events at B please tick the box below your signature overleaf.**

Thank you for your help.

Document Analysis. **The National Museum Trust Limited**
Registered Charity No.

TERMS AND CONDITIONS

As you have purchased an all-inclusive ticket, the Museum Trust would like to offer you free re-entry to the National for one year following your initial visit. You can come back as many times as you like within this period but please be aware that the following rules and regulations apply.

Free re-entry to the non transferable and valid for one year following the original visit during normal opening hours (excluding certain key dates).

Your free return pass will not be valid for admission to any events held at Beaulieu.

Free re-entry will not cover entry to events or to the remainder of the complex, other than the Brabazon Restaurant. If you wish to visit the remainder of the complex or a major event when revisiting then a supplementary charge will be paid at the point of entry. Details of the current supplementary charges and dates for non event days can be found at www.beaulieu.co.uk/yearlygiftaidvisits

When re-visiting you must provide signed valid proof of your identity. Such valid proof will be deemed to be a valid credit or debit card, drivers licence (provisional or full), passport or other means as notified from time to time. You will also be requested to counter-sign your name as appropriate.

reserve the right to refuse admission if the signature on this pass is deemed not to match that of the person signing in on the return visit.

You and your party may on occasion be required to be hand stamped before entry to validate your returning status.

If your pass is lost a duplicate can be provided at a cost of £20. Please contact us for further information. We require 7 days notice for production of duplicate passes

THE NATIONAL

TRUST

Annual Membership

Be a gift aid visitor

Registered Charity No.

The following two images show the final form issued with the Gift Aid, this is an information sheet containing again the terms and conditions along with frequently asked questions.

giftaid it

Gift Aid Donations

How you can help us
- at no extra cost to you

Museum is one of the attractions of - and is a registered charity. Because your payment will give you and your family admission to the Museum for a year , we can take advantage of the "Gift Aid Admission" rules.

If you are a UK taxpayer, we can claim back the tax that you have paid on your donation from HM Revenue and Customs - at no extra cost to you.

All you need do is complete a Gift Aid declaration with your name and home address. We can then reclaim the tax on 60% of the total admission amount that you pay , for you and your family to visit Beaulieu for one year.

To us, that's worth an extra **25p for every £1** that you give in this way.

You can imagine the difference this will make to the finances of the National Motor Museum Trust - with your help we can preserve the historic vehicles and unique archives in our care for future generations to enjoy.

If you are a higher rate taxpayer, you can claim additional tax relief when you complete your Self Assessment tax return – you should retain your receipt as proof of your donation.

By making a donation equivalent to our admission price you (and your family group) will not only receive free admission today but also free re-entry to the Museum for a year following your initial visit. You can come back as many times as you like within this period to the Museum (excludes entrance to the rest of the complex and admission to Special Events).

That's all! You will not be contacted further on this matter. If you wish to receive news of future events at Beaulieu, just tick the box below your signature on the declaration.

We hope you will enjoy your visit to the National Motor Museum and that you will appreciate the work that is being done to preserve such a valuable part of our heritage.

Thank you for your interest in our work.

n Trust Limited

Registered Charity No. 1107656

Q **What is Gift Aid?**

A *Gift Aid is a scheme that allows charities to claim back the tax paid on donations and eligible payments from individuals who are UK taxpayers and who agree to make a Gift Aid declaration.*

Q **Why can the Museum ask for a donation instead of an admission charge?**

A *This is because we are a charity and your payment will give you and your family entry to the Museum for a year.*

Q **Who can donate using the Gift Aid scheme?**

A *Any individual who pays for him/her self and his family to have entry to the Museum for a year and also pays UK Income and/or Capital Gains Tax.*

Q **What do I have to do?**

A *Simply complete a declaration with your name and home address -- which will remain confidential. You do not need to tell the taxman you have made a Gift Aid declaration.*

If you pay Income Tax at the higher rate you should simply retain your receipt as proof of your donation, and then include your donation on your Self-Assessment Tax return.

Q **If I give you my name and home address, will I receive yet more junk mail?**

A *No! We promise that we shall not pass your details on to anyone other than HM Revenue and Customs -- and we only give them your name (but we have to keep your home address on file). We will only write to you if you ask for news of future events at Beaulieu*

Q **Can I donate my entrance fee even if I cannot or do not wish to take part in the Gift Aid scheme?**

A *Yes you can. If you make a donation equivalent to our entrance price you (and your family group) will still be given admission to the National Motor Museum for a year from today (conditions apply).*

Having looked at the documents currently used there is again a lot of data duplication as the Gift Aid is explained several times, when creating a new system the amount of paper work currently used will need to be addressed and if possible greatly reduced, although by law each Gift Aid visitor must be given a copy of the terms and conditions so although this form would still be required there would no longer be the need for the first form if a computerised system were to be implemented.

The Current Spreadsheet

Below is a list of the current fields used with in the spreadsheet.

Field	Description
Reference Number	Unique to each form this is equivalent to the number that is stamped on the form when it has been processed.
Date	This is the date that the transaction occurred; this is needed so that the Gift Aid totals can be added up for each day.
Title	This is just how the customer wishes to be addressed.
First Name	This is the name or initial that the customer gives on the form.
Surname	This is the name or initial that the customer gives on the form, with out this the Gift Aid can not be processed correctly.
House	This is either the customer's house name and or number.
Postcode	This is used to calculate the rest of the customers address. With out this the Gift Aid form cannot be processed.
Street	This is automatically calculated from the postcode.
Locality	This is automatically calculated from the postcode.
Town	This is automatically calculated from the postcode.
County	This is automatically calculated from the postcode.
Mailing List	This is a Boolean field using a '1' when a customer wishes to be added to the mailing list and a '0' when the customer doesn't want to go on the list.
Total Paid	This is the total of the transaction minus Palace House tickets and guide books.
Amount of Donation	This is calculated from the total paid.
Cumulative Total Paid	This is the sum of the total paid.
Cumulative Total Donated	This is sum of the total donations.

This is all displayed on a single worksheet within a spreadsheet and can appear confusing to user. When typing in data the employee has access to all the data, although some of the fields such as Cumulative total paid is calculated by the computer automatically it is still available for the user to edit this value, this could cause data inaccuracy. A new spreadsheet is started at the start of each month. However this means that large amounts of customer details can be seen on one screen this again can make it confusing for the

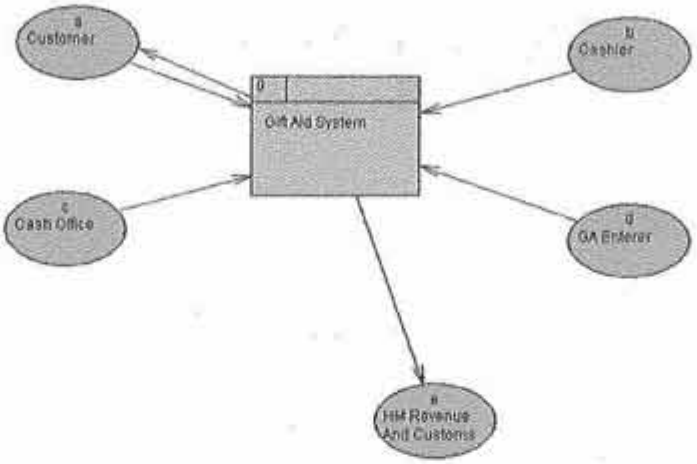
user and can also lead to data inaccuracy as the user may edit previously input information.

The whole of this spreadsheet is then passed on to HM Revenue and customs so that they can check that the people that have Gift Aided are UK tax payers and that they are paying enough tax to be able to make the Gift Aid donation. In addition to this the signed paper copies of the Gift Aid forms need to be kept as the tax office reserve the right to check that the customer has signed to agree to be part of the Gift Aid scheme.

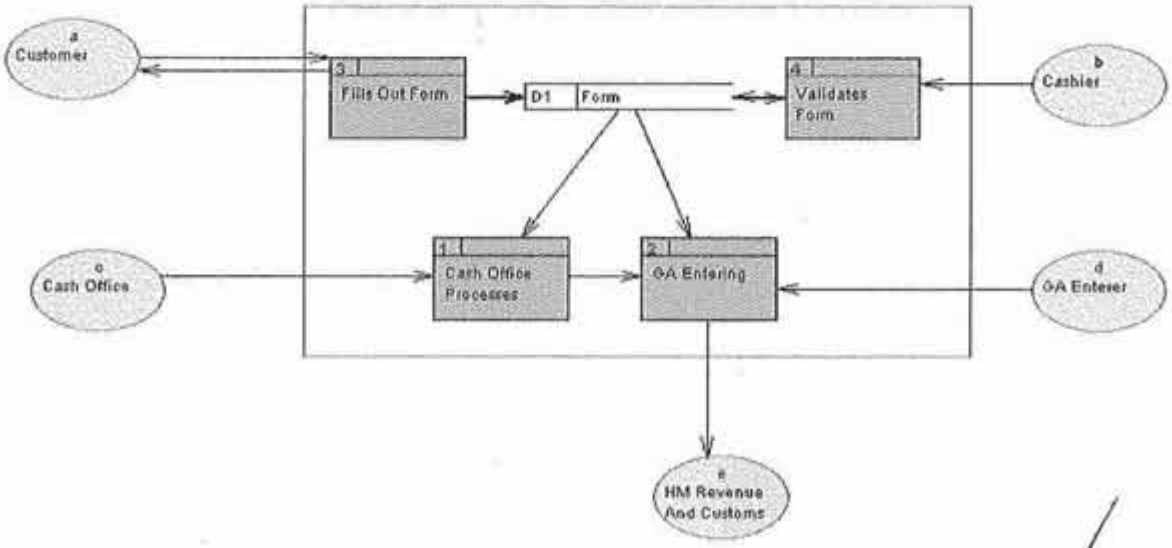
Data Flow Diagrams (DFD's)

The following diagrams show how the data moves around the current system.

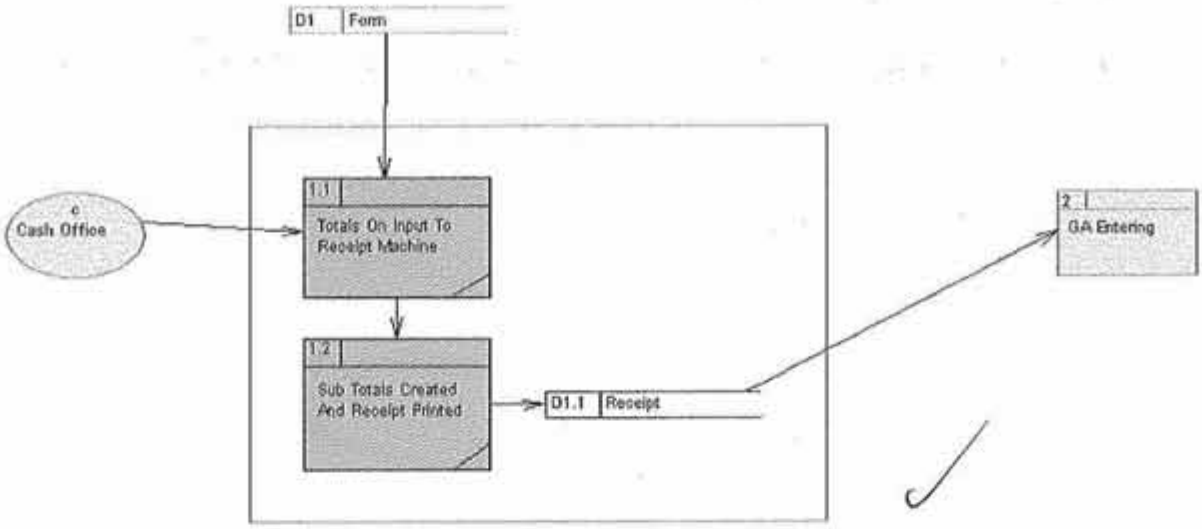
Current Gift Aid System



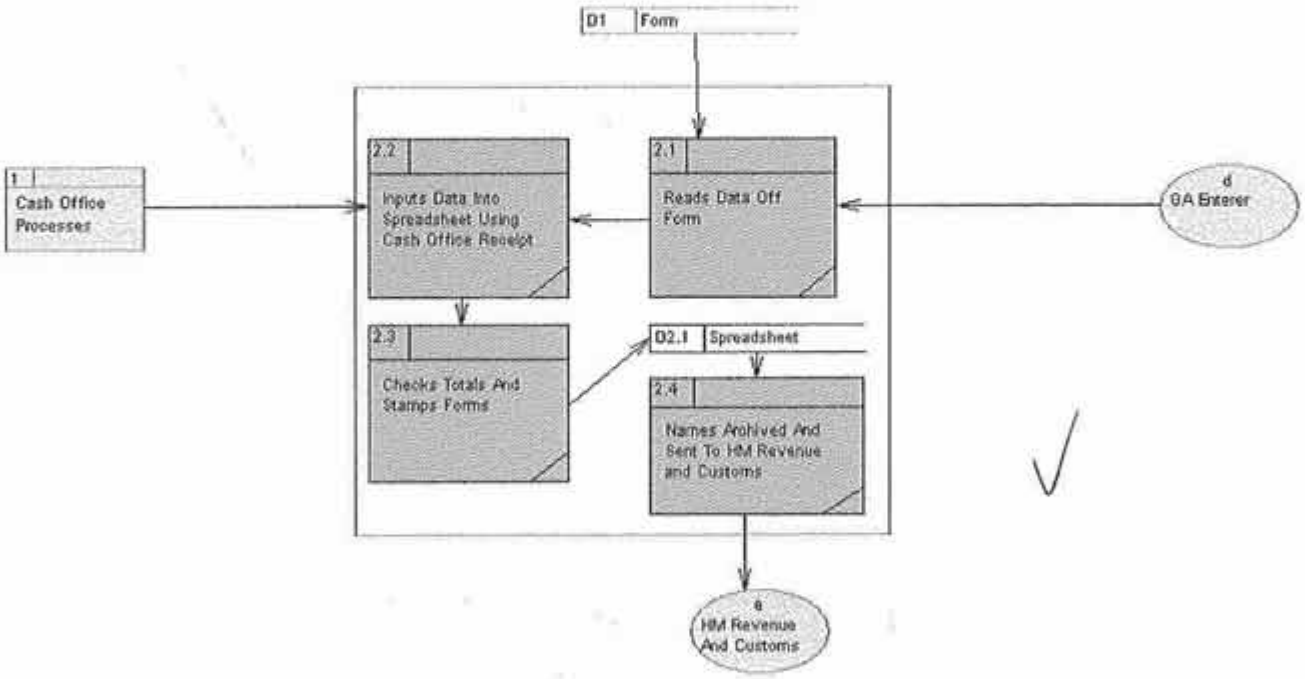
Gift Aid System (Level 1 Diagram)



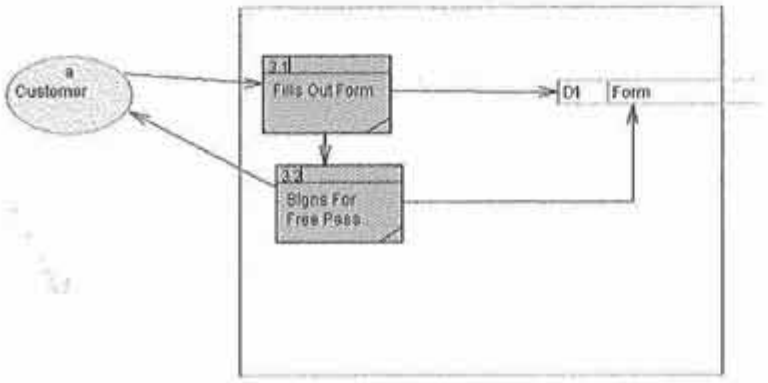
Cash Office Processes (Level 2 Diagram)



GA Entering (Level 2 Diagram)



Fills Out Form (Level 2 Diagram)



Problems with the Current System

After carrying out the investigation the following problems have been identified with the current system;

- A lot of time is wasted by data duplication
- Customers turn down Gift Aiding because it takes them to long to fill out the form
- Errors are easily made and due to this the company is unable to process some of the forms and therefore lose some of the tax
- Employees find the system time consuming
- Some customers, especially the more mature ones, seem confused with the form and often need assistance filling it out
- Those customers that are hard of hearing or didn't speak English aren't explained the Gift Aid scheme
- Several employees are paid to work on the Gift Aid and often two employees are doing the same task but both of them are being paid

Specialist Terminology

Whilst doesn't use a lot of specialist terminology they do use a lot of abbreviations. Some of these abbreviations can be seen below.

GA= Gift Aid
PH= P H
NMM= N M Museum
GA Upgrade= Gift Aid Upgrade
GB= Guidebook

Some of these abbreviations will be necessary in my program, all the staff fully understand what these abbreviations stand for as they are regularly used across the whole site.

User Requirements

Generally a system will be required to allow the staff member to enter the customer's details and store them along with the transaction price to be able to reclaim the tax. To do this the new system must be able to carry out the following tasks.

The program must allow till operators to;

- Log in with an individual ID
- Enter and store customer details including; first name, surname, address and email address
- Enter and store the price of each transaction
- Check customers in who have previously Gift Aided, it must also give the customer the option to upgrade their Gift Aid pass to give them access to the whole complex, therefore the system must store surnames, first names and postcodes of each customer with the number of tickets that they bought
- It must provide a check screen to ensure that the customer is declaring that they are a UK tax payer before the transaction is complete

The manager or supervisor must be able to;

- Log in using an ID that gives them higher access levels than normal till operators
- View all customer's details
- View the total Gift Aid value
- Edit the price of the tickets
- View a total Gift Aid tickets sold for comparison of employee sales
- View the total guidebook sales and percentages for each till operator

At the end of the day the system must produce;

- A completed list of customers and their details. This should be available in a hard copy so that all records can be kept
- It must also calculate a total for the Gift Aid reclaimed on the entry
- Produce an output of customer email address and names in a different document to the rest of the customer's details so that it can be sent to the marketing department
- Produce a total for the number of people that have visited in the day and produce an overall percentage of the amount of revenue that has been Gift Aided

Ideally the system should do be able to;

- Give each transaction a unique identifier so that it can be identified for the Gift Aid
- Allow event tickets to be sold through the system and allow the supervisors and managers to view all of the customers details that have bought event tickets
- Allow supervisors and managers to carry out refunds to void out the transactions that have been carried out
- At the end of the day a report should be generated which calculates a percentage for the number of people that have bought a guidebook or upgraded their Gift Aid return ticket

Program Specification

Inputs, Outputs and Processing

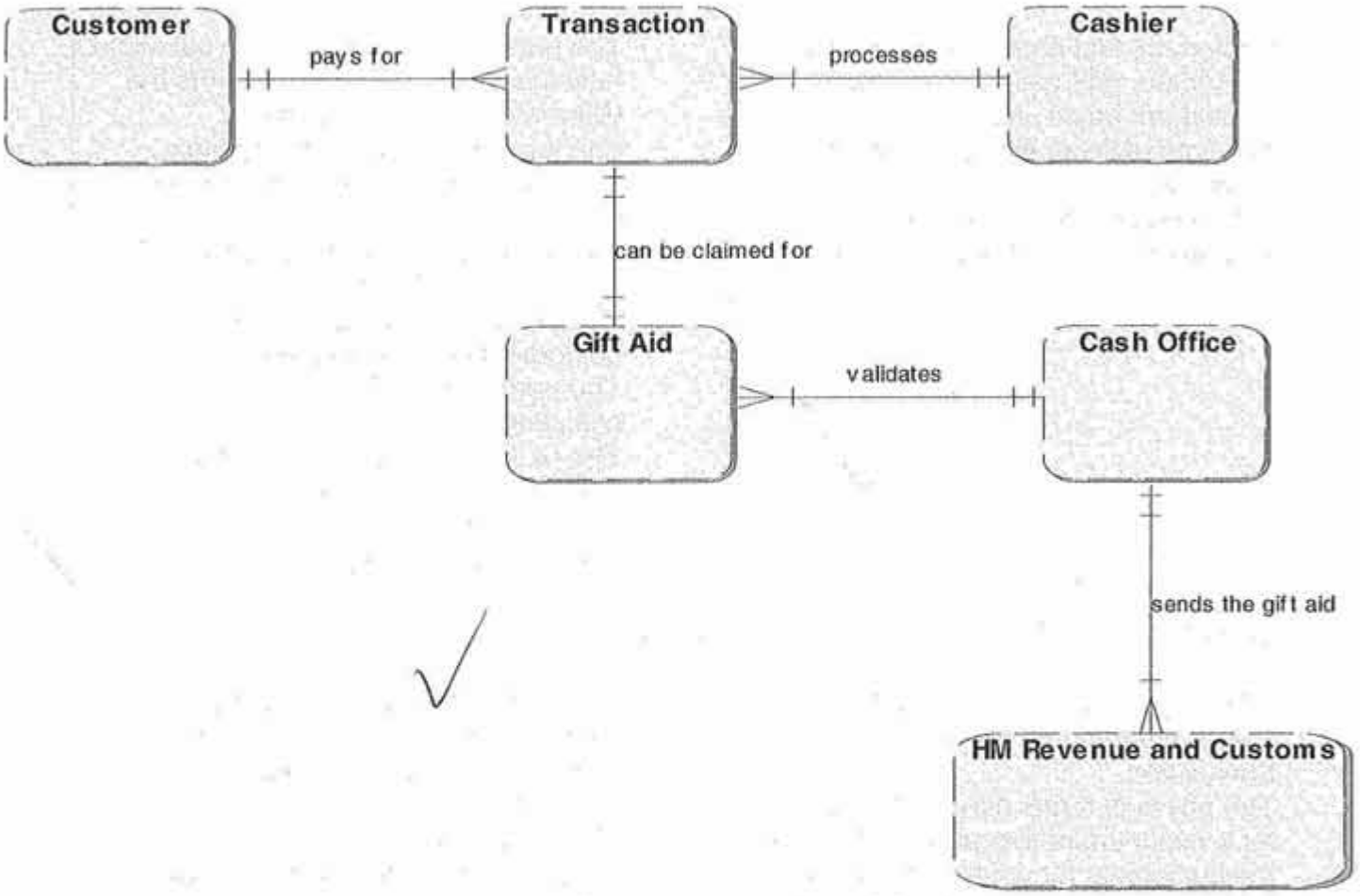
Various pieces of data need to be collected from the different entities. The inputs for each of the different entities can be seen in the table below.

INPUTS	PROCESSES
- Customers first name and surname - Customers address and postcode - Customers email address - The type of ticket that the customer is purchasing - The method of the customers payment - The amount of cash a customer has given	- The price needs to be used to calculate how much money will go towards the Gift Aid from each transaction - The total amount of reclaimed tax needs to be totalled for each month and then for year - The total amount of money received needs to be calculated - The amount of change to give a customer needs to be calculated - The percentage of Gift Aid needs to be calculated - The percentage of Guidebook sales also needs to be calculated - Z reports need to be generated at the end of each working session
STORAGE REQUIREMENTS	**OUTPUTS**
- All customer details needs to be kept, along with the date and price of the transaction - The physical forms need to be stored for a minimum of five years. A list of all the customers for each month needs to be stored and kept for 5 years and then archived after that - A separate spreadsheet needs to be set up containing the customers email address so that it can be given to the marketing department	- At the end of the month a list containing all customers' details needs to be sent to the tax office. This is done electronically - The total money donated and total money taken needs to be displayed in the database - All email addresses need to be recorded so that they can be added to a mailing list and sent to the marketing department - When a returning customer visits the cashier must be able to search for the customers details and they must be output to the screen - All of the calculated information must be output to the screen - The Z reports generated at the end of each working session should be output to print

Entity Relationship Diagram

The entity relationship diagram below shows how the entities within the current system interact with each other.

Gift Aid (Current System)



The diagram above shows that there are several entities interacting with the system. Although it doesn't show this, all of the information that the different entities are working on is the same information. This causes a lot of data duplication and in some cases data inconsistency.

How the User Requirements Will Be Met

I propose to set up a computerised system which will allow the user to set up their details and

Below is the list of user requirements that have been given for the new system and also how I propose to meet these requirements.

The program must allow till operators to;

- Log in with an individual ID
  To do this I will set up a login screen that will need the user to login before they can gain access to the main system
- Enter and store customer details including; first name, surname, address and email address
  To do this I will set up a screen that allows the cashier to enter all of the customer details, the screen will consist of a number of labels and text boxes to make the data entry as easy and straight forward as possible
- Enter and store the price of each transaction
  The transaction file will store all of the information about the file such as the price, each transaction can be uniquely identified with the Transaction ID field
- Check customers in who have previously Gift Aided, it must also give the customer the option to upgrade their Gift Aid pass to give them access to the whole complex, therefore the system must store surnames, first names and postcodes of each customer with the number of tickets that they bought
  I will create a screen that allows the cashier to search for existing customers, on the same screen I will also then give the cashier the option of booking in the customer
- It must provide a check screen to ensure that the customer is declaring that they are a UK tax payer before the transaction is complete
  When a Gift Aid transaction is processed I will set up a message box to alert the cashier to ensure that they ask a customer if they wish to Gift Aid

The manager or supervisor must be able to;

- Log in using an ID that gives them higher access levels than normal till operators
  I will assign each cashier an access level, when they login this access level will be used to determine which screens the cashier is able to view
- View all customer's details
  I will set up a screen that will allow the supervisor to search through all of the existing customer details
- View the total Gift Aid value
  I screen will be provided that will allow the user to view the total Gift Aid value as well as the individual Gift Aid scores for each customer
- Edit the price of the tickets
  I will set up a further form which will allow the supervisors to change the prices of the tickets

- View the total guidebook sales and percentages for each till operator
  I will also set up a form that will show a list of all of the till operators along with the percentage of guidebooks they have sold over all

At the end of the day the system must produce;

- It must also calculate a total for the Gift Aid reclaimed on the entry
  Within the report the Gift Aid percentage for the total for each day will be shown
- Produce a total for the number of people that have visited in the day and produce an overall percentage of the amount of customers that have purchased a Guide Book or upgraded their Gift Aid return pass
  This is also something that will be shown up in the Z report

Ideally the system should do be able to;

- Give each transaction a unique identifier so that it can be identified for the Gift Aid
  Each transaction will be assigned a unique transactionID
- Allow event tickets to be sold through the system and the customer details stored
  I will set up a form which will allow the supervisors and managers to be able to sell event tickets
- Allow supervisors and managers to carry out refunds to void out the transactions that have been carried out
  I will allow the supervisors and managers to change the quantity of the tickets to a minus value, this will only be allowed to be carried out by users of the correct access level
- Produce an output of customer email address and names in a different document to the rest of the customer's details so that it can be sent to the marketing department
  Within the report that shows all of the customer details I will produce a report which allows the supervisor to view only the email address' and names of the customers which will also be able to be printed
- A printed list of Gift Aid customers and their details. This should be available in a hard copy so that all records can be kept
  The supervisor will be able to view the customer details based upon the month that they entered, at the end of the month this information can be printed and stored in the Cash Office for safe keeping until the point where they can be destroyed

Constraints and Limitations

One of the limitations of the project is that the current Gift Aid system is used at the same time as the tills. Ideally the new system would be linked with a new till system which would be connected to the computer network that the company already have in place. This would then be able to deal with many aspects such as processing a group, generating an invoice or selling items in the gift shop. However for me to be able to complete all of this within my time set would be unrealistic and therefore I will only concentrate on a new till system that will be used to directly process the customer Gift Aid and process the entry tickets.

Whilst some employees have said that they are confident using a computer they may still require some training, to make this easier I will implement a logical system to make it as user friendly as is possible.

I had decided that I would implement a system that allowed the customer to enter their details on entry to the building which could be then picked up by the receptionist when they get to the desk. However this would have been very expensive as it would have involved a network being set up to link the computers and it would also involve several more computers.

The company already has a network in place as all the computers on site are already networked to allow staff to log in at several locations. To implement this system at each till the company will have to purchase more computers ideally with touch screens to make it easier for the cashier to use. Ideally these computers will be networked as well as the customer details will need to be sent to the cash office and the email addresses will need to be sent to the marketing department.

The whole project needs to be completed before the 30th April 2010.

The company already have one computer in the main Visitor Reception area. However if the company wanted to be able to use the system on all of the tills they would need to replace the existing tills with computers. This would involve 7 new computers being purchased. None of these would need any specialist software to be able to run the new system as the system that I propose to make will be an executable file.

The software that will be used to create the program will be Microsoft Visual Basic 6.0, which I have access to both at home and at college. A DAT file will be used to store the customer's details.

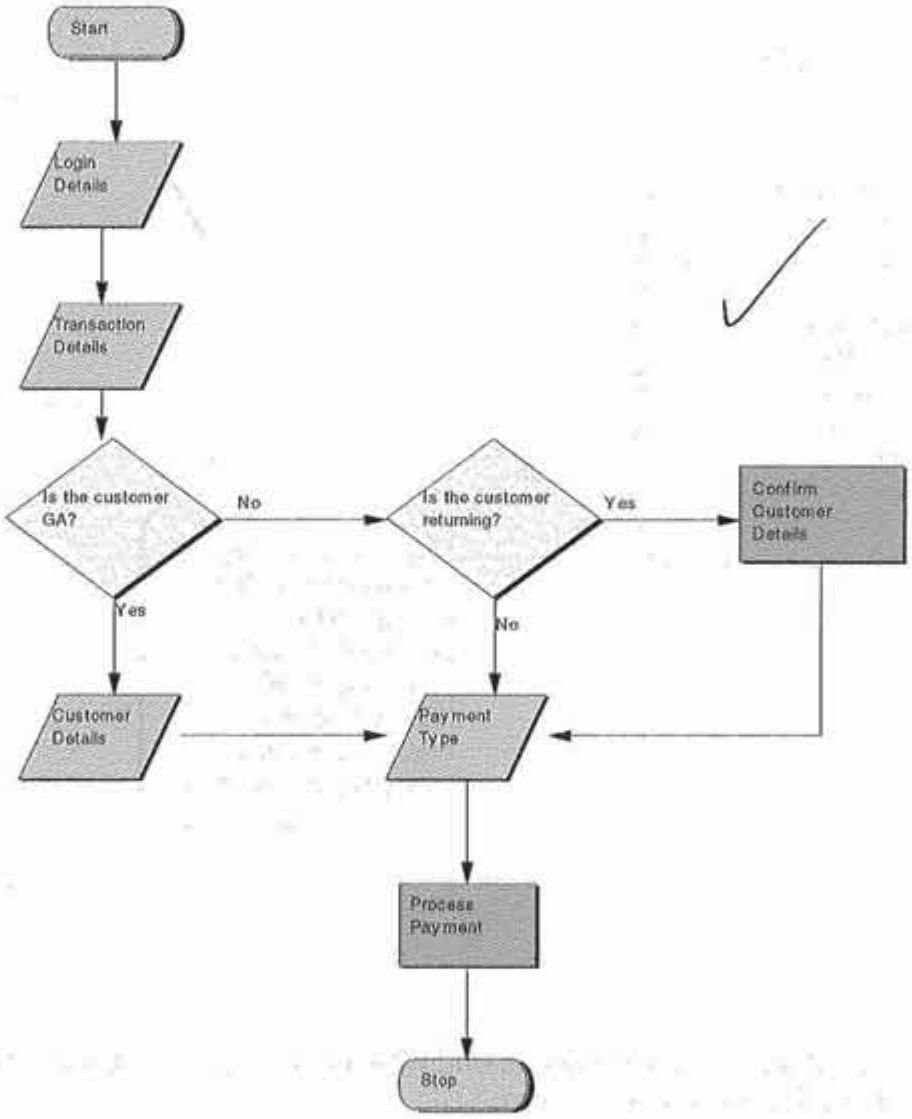
TASK B

DESIGN

Systems Flowchart

The chart below shows the overall process that a normal cashier would undergo to process the different types of tickets available.

Processing a Normal Transaction



The process starts when the cashier logs in to the system. The cashier then enters the type and amount of the different ticket types that the customer requires.

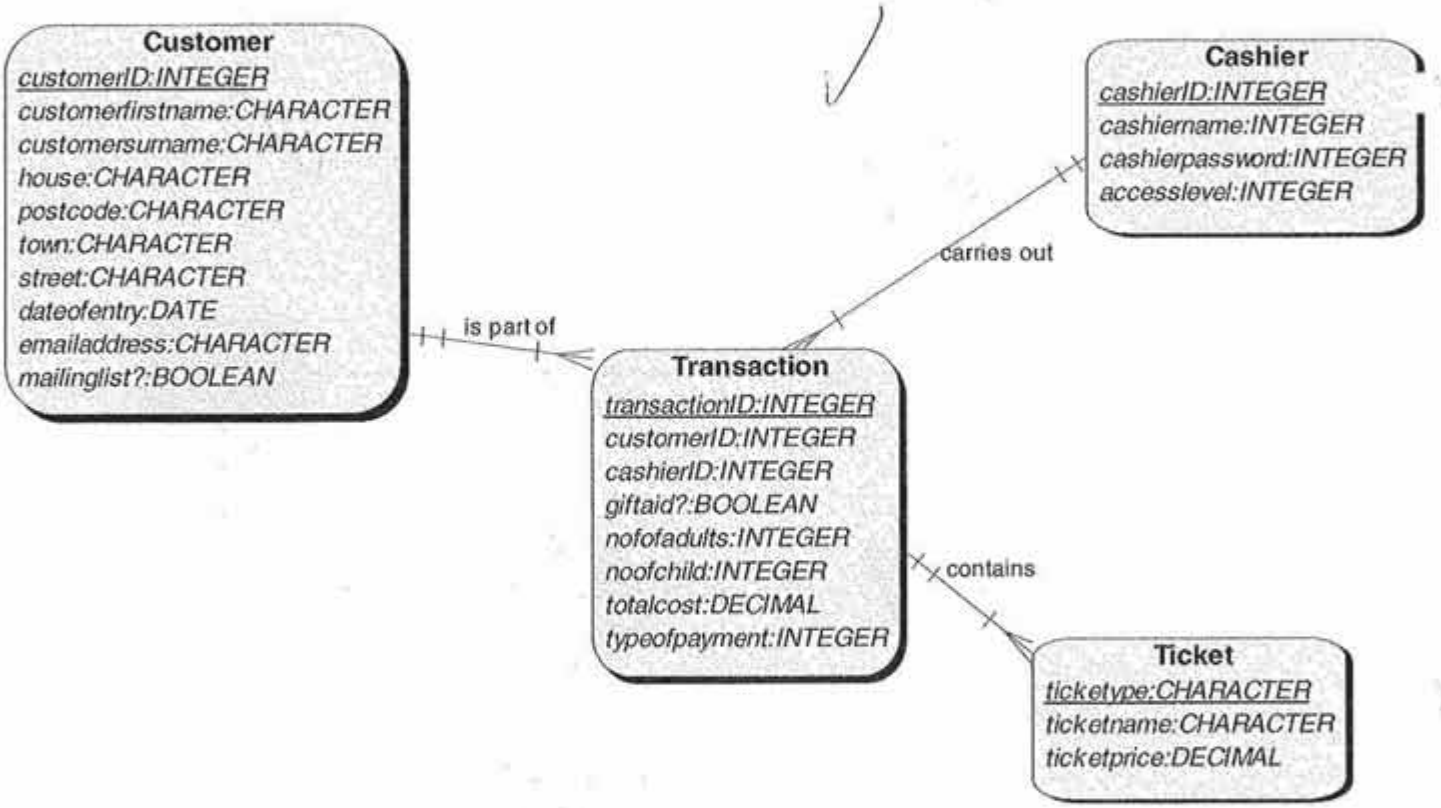
Entity Relationship Diagram

The entities within the system will be;

- Customer
- Cashier
- Transaction
- Ticket

The relationships for these can be seen in the entity relationship diagram below.

- Gift Aid System



This shows all of the entities and what information that is going to be stored about them in a new system.

Data Dictionary

The tables below give the details of all of the information that is going to be stored about each of the entities. This is shown in a data dictionary.

Customer Details

Data Item	Data Type	Length	Description/Validation/Format
customer_ID	Long Integer	4 bytes	This is unique to each customer and is generated automatically. This is also the primary key for the customer entity.
firstname	String	Max 20 characters	The customer's first name, field can't be left blank.
surname	String	Max 20 characters	The customer's surname, field can't be left blank.
property	String	Max 20 characters	The name or number of the customer's property. This field is required.
postcode	String	Max 8 characters	The customer's postcode can be no longer than 8 characters including a space. This field is also required.
street	String	Max 20 characters	The street that the customers address is in, field can't be left blank.
town	String	Max 20 characters	The town that the customers address is in, field can't be left blank.
county	String	Max 20 characters	The county that the customers address is in, field can't be left blank.
dateofentry	Date	8 bytes	Automatically saved to the customer file based upon the current date. Format dd/mm/yyyy.
emailaddress	String	20 characters	The customers email address taken from the text box on the details form.
mailings	Boolean	2 bytes	The customer will either be on the mailing list, true, or not on the mailing list displayed as false.

Cashier Details

Data Item	Data Type	Length	Description/Validation/Format
cashier_ID	Long Integer	4 bytes	This is unique to each cashier and is generated automatically. This is also the primary key for the customer entity.
cashiername	String	8 characters	The cashier name taken from the text box when they log in.
password	String	8 characters	The cashiers password again taken from the text box when they log in.
accesslevel	Integer	2 bytes	This will be set when the new cashier is added. It will vary depending on whether the operator is a cashier or a supervisor. This can not be left blank.

Transaction Details

Data Item	Data Type	Length	Description/Validation/Format
transaction_ID	Long Integer	4 bytes	This is unique to each transaction and is generated automatically. This is also the primary key for the customer entity.
customer_ID	Long Integer	4 bytes	This is automatically taken from the customer file.
cashier_ID	Long Integer	4 bytes	This is automatically taken from the cashier file.
amountpaid	Currency	8 bytes	The price of the transaction. With a validation check of between £14.50 and £300.
giftaid?	Boolean	2 bytes	This will save whether or not the customer has allowed the transaction to be signed up to Gift Aid or not.
noofadults	Integer	2 bytes	Must be between 0 and 100.
noofchildren	Integer	2 bytes	Must be between 0 and 100.
typeofpayment	Boolean	2 bytes	Will either be C or V for cash or Visa, this can not be left blank.

Ticket Details

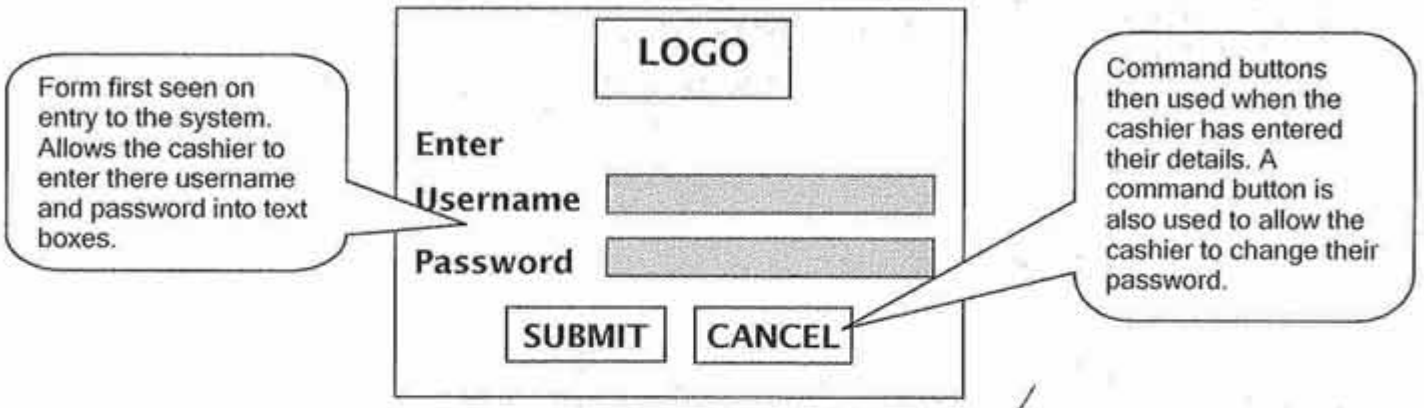
Data Item	Data Type	Length	Description/Validation/Format
tickettype	Text	14 characters	Gives the type of ticket e.g. gaadult. Can't be left blank. This must be unique.
ticketname	Text	16 characters	Gives the name of each ticket e.g. Adult GA Return.
ticketprice	Currency	8 bytes	This must be between £0.00 and £65.00.

Form Designs

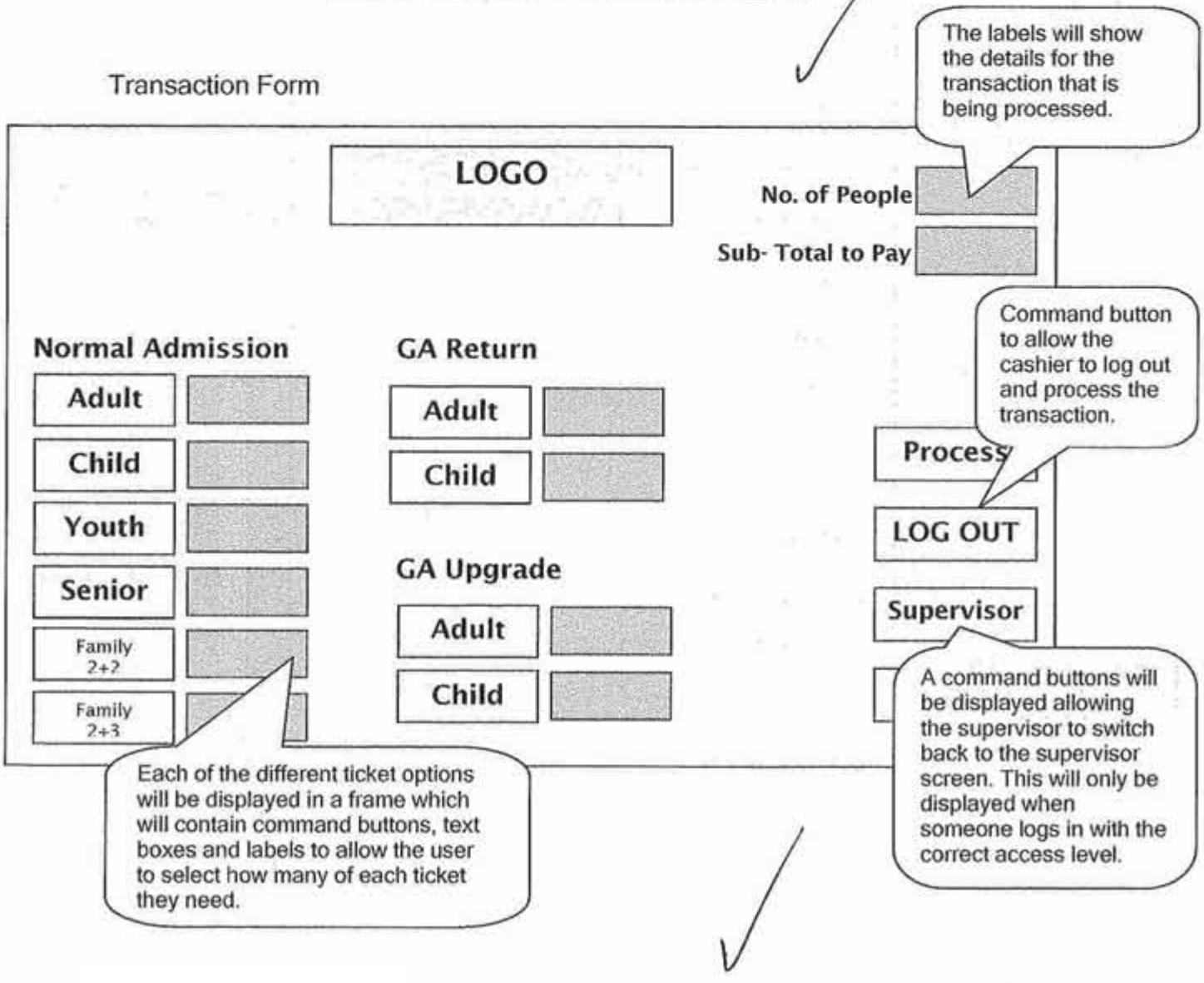
The images below show designs for all of the forms that I am going to include in my program.

Input Screen Designs

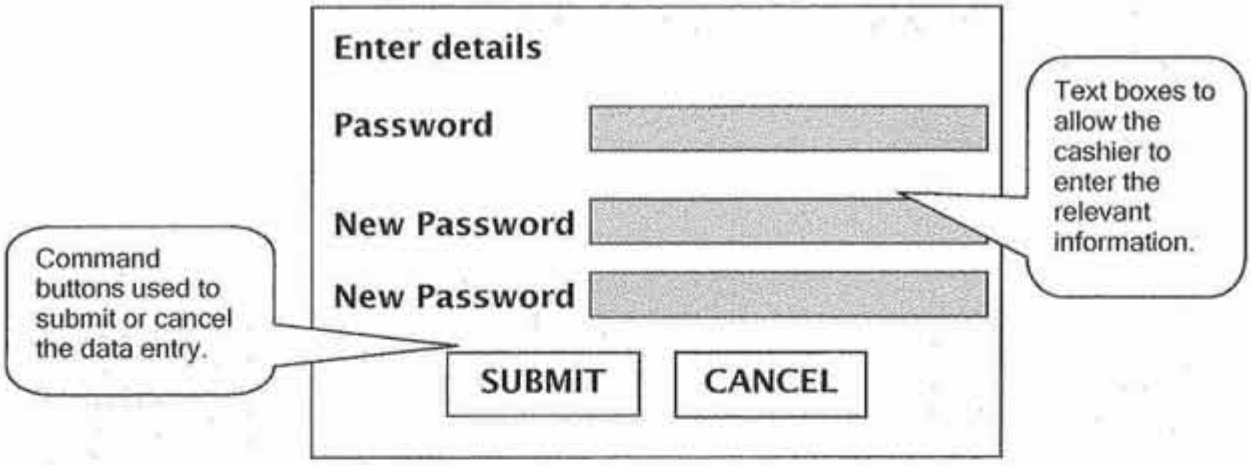
Login Form



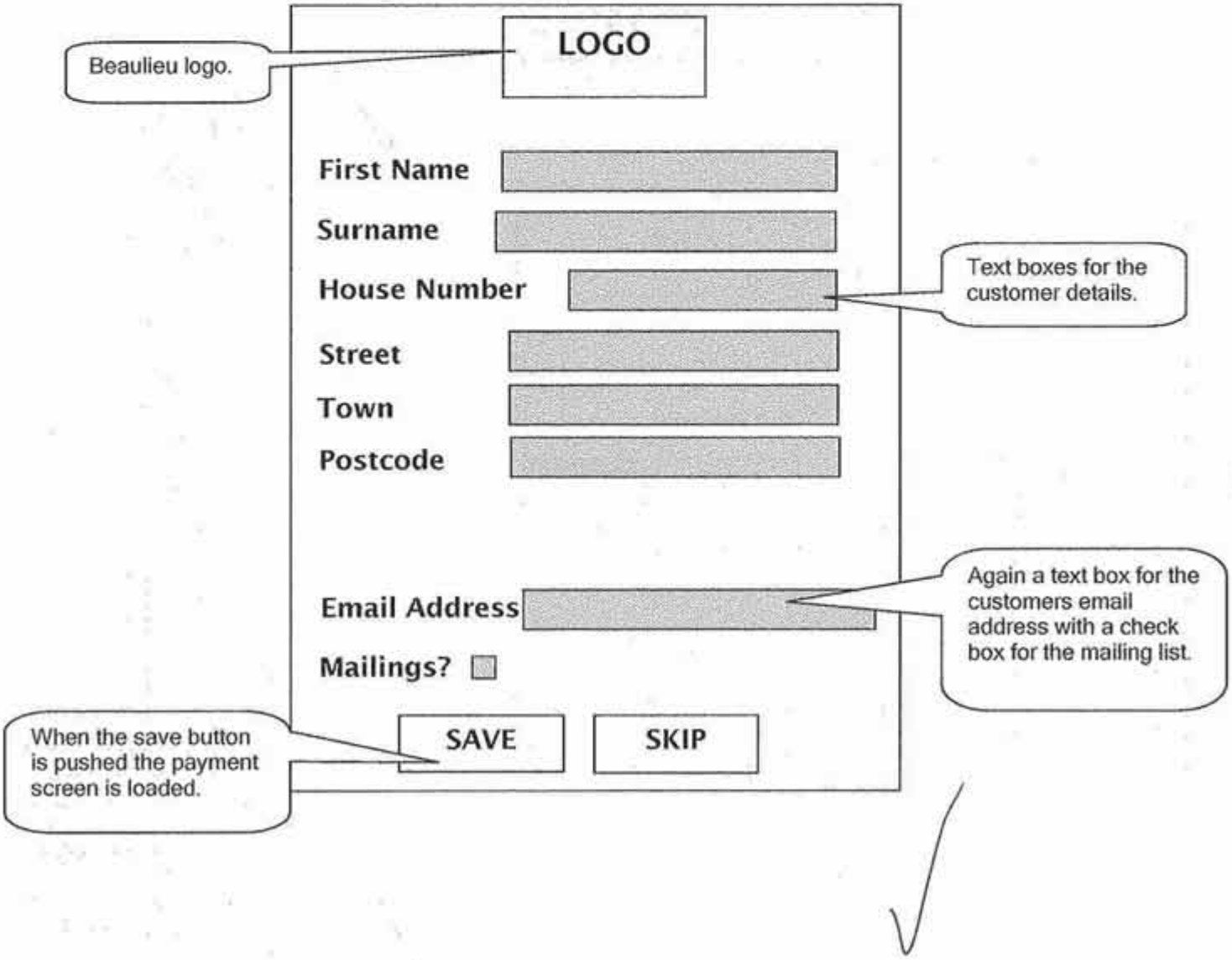
Transaction Form



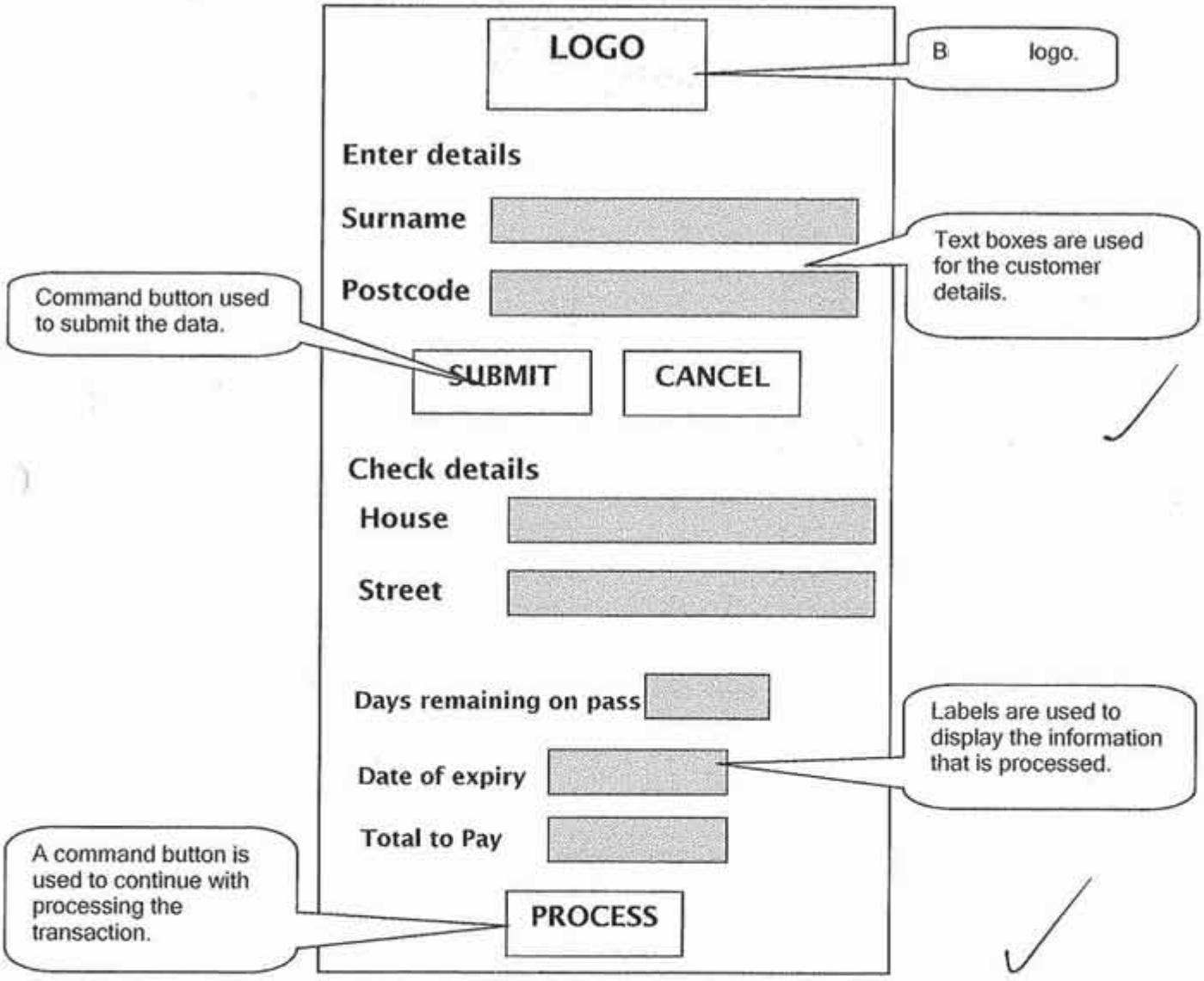
Change Password



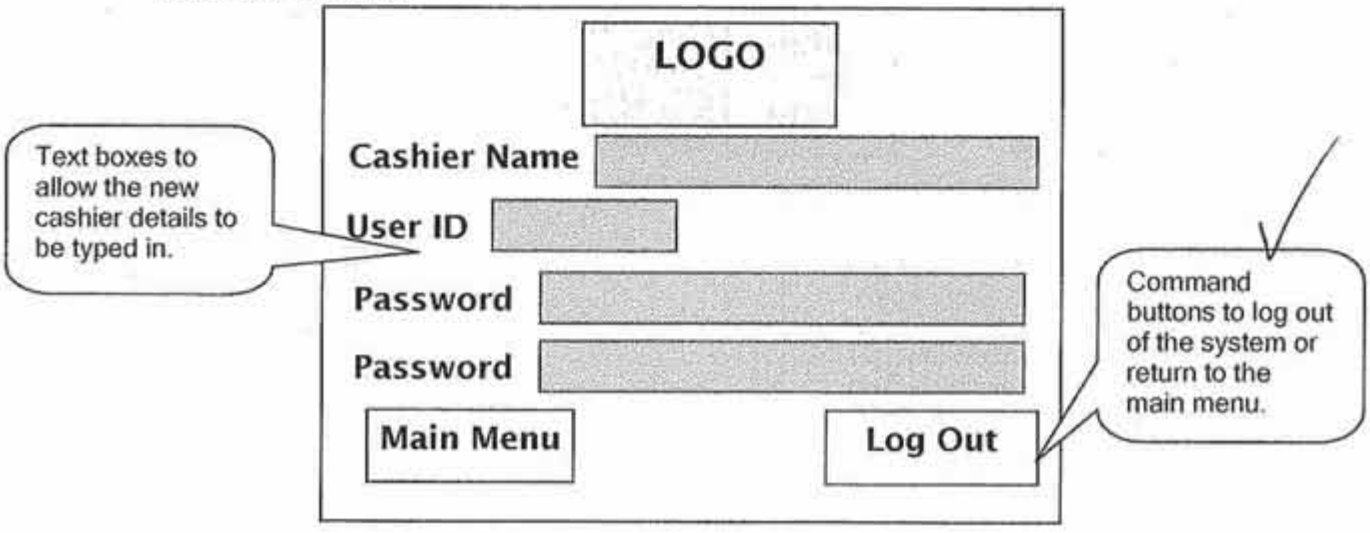
Customer Details Form



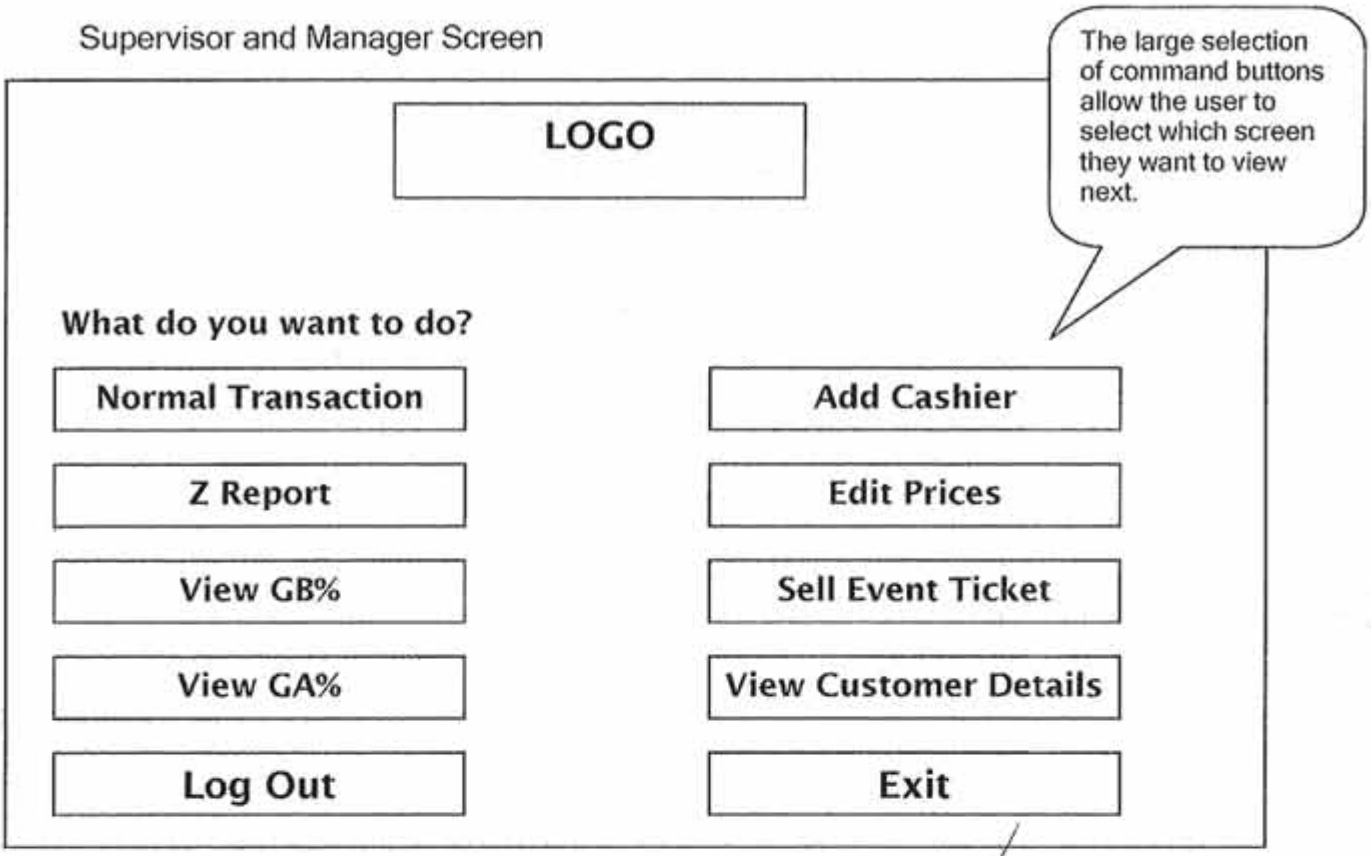
Returning Customer Form



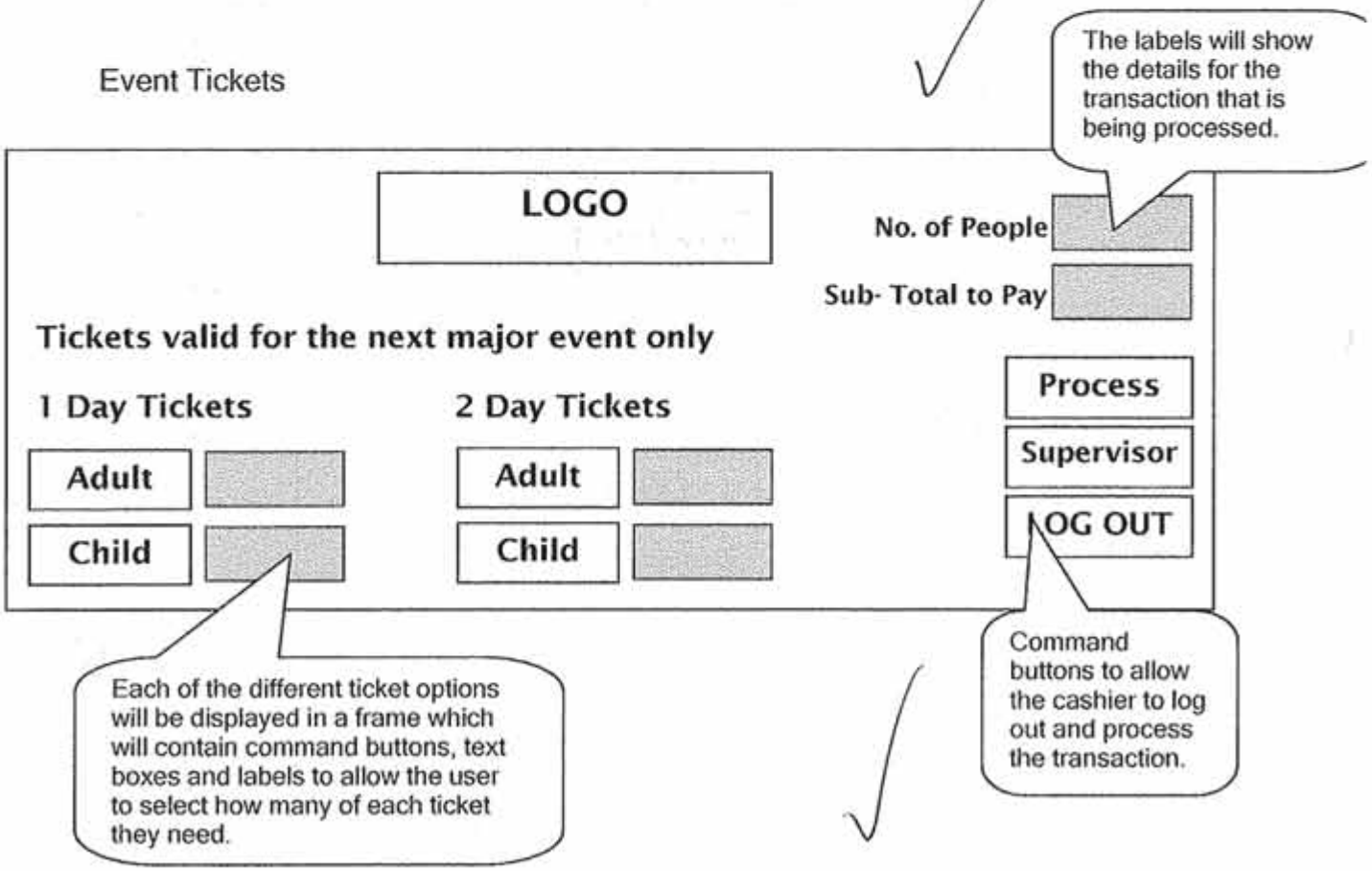
Add New Cashier



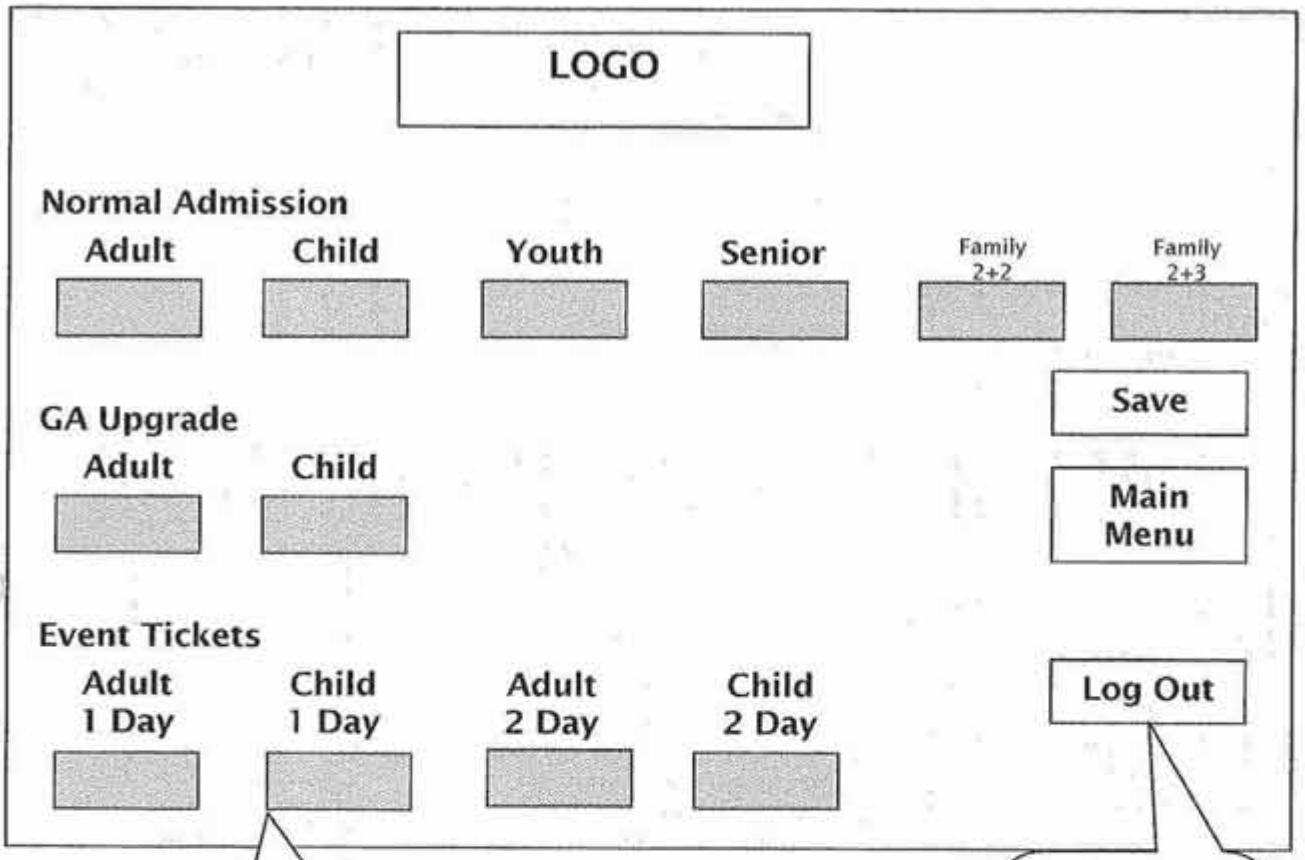
Supervisor and Manager Screen



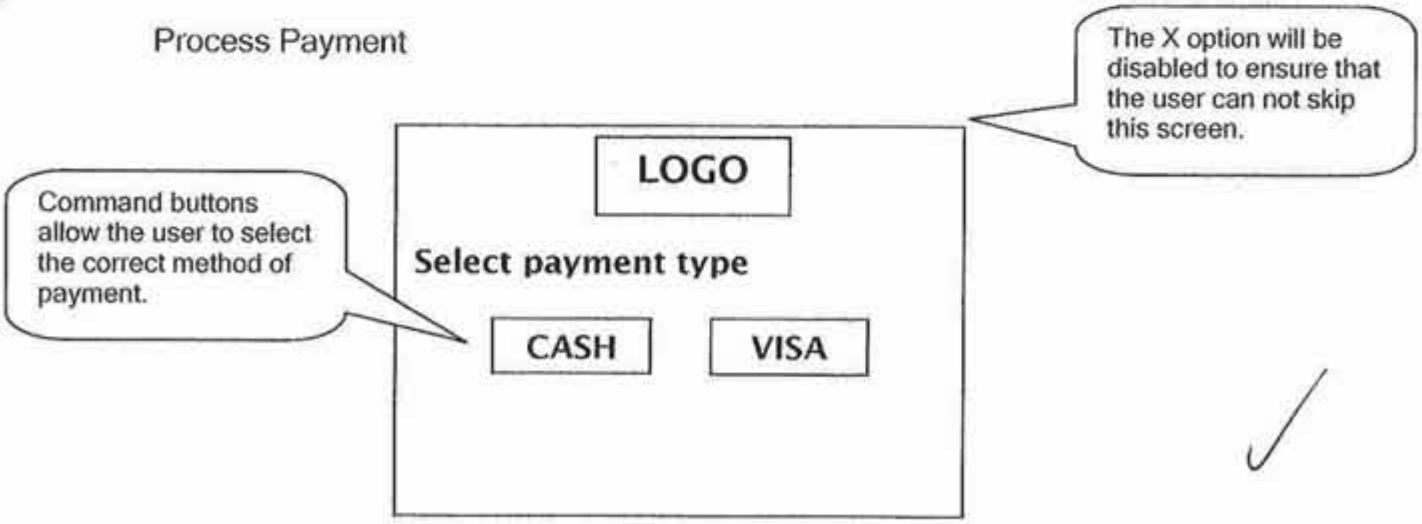
Event Tickets



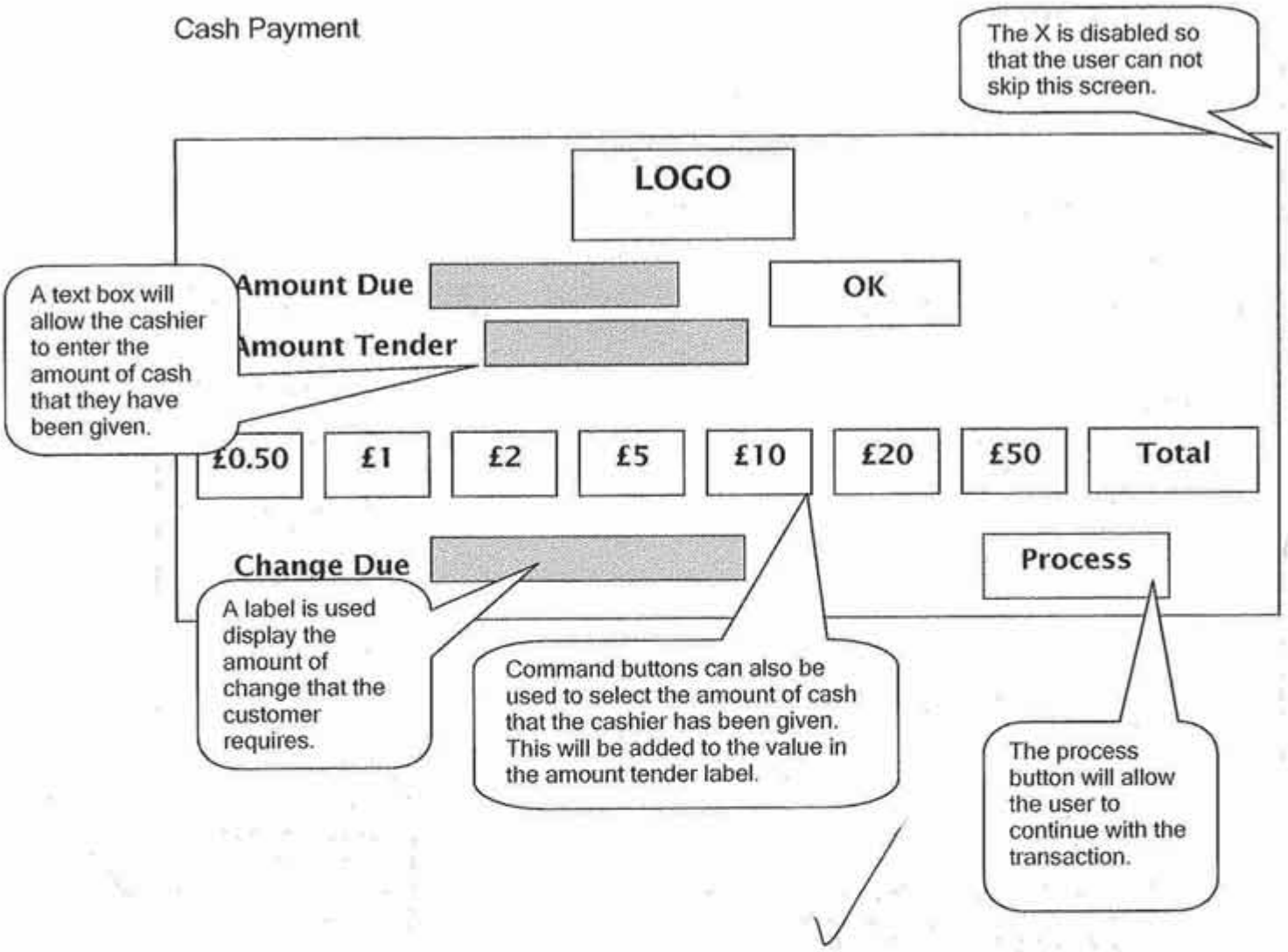
Edit Prices



Process Payment

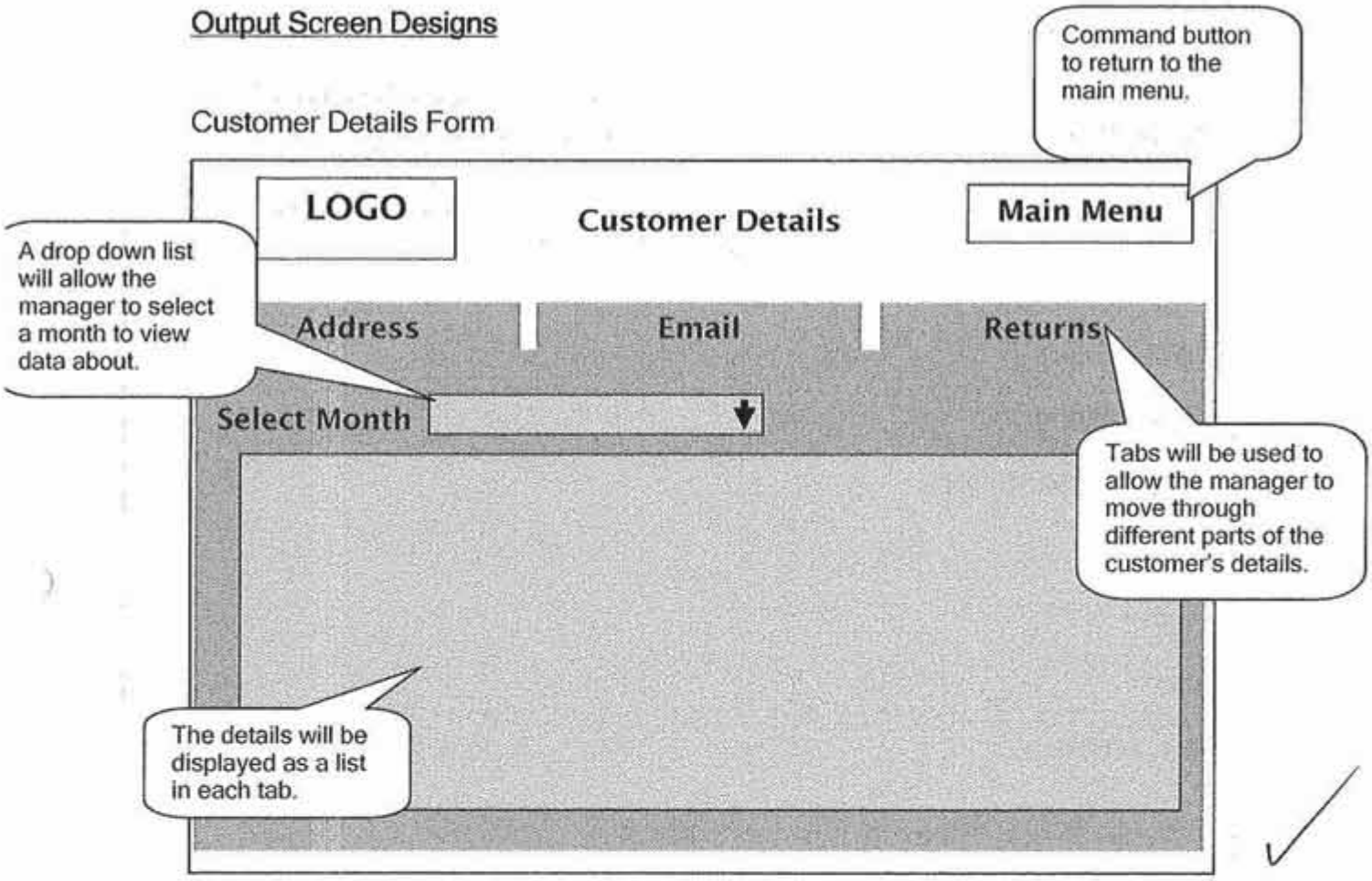


Cash Payment

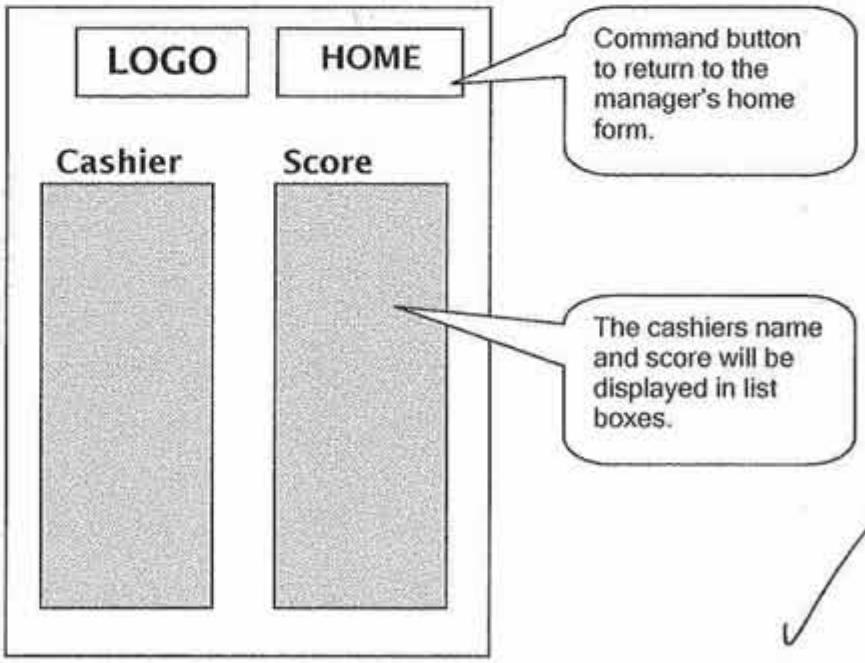


Output Screen Designs

Customer Details Form



Cashier GA and Guidebook Performance Form



I will create two forms that look like this; one will show the GA% the other will show the GB%.

Z Report Design

I will also be generating a report sent to a printer. This report will contain the information for the daily takings, the layout of the report can be seen below.

Z Report- dd/mm/yyyy

No of Adults =

No of Children =

TOTAL ADMISSION =

Cash =

Visa =

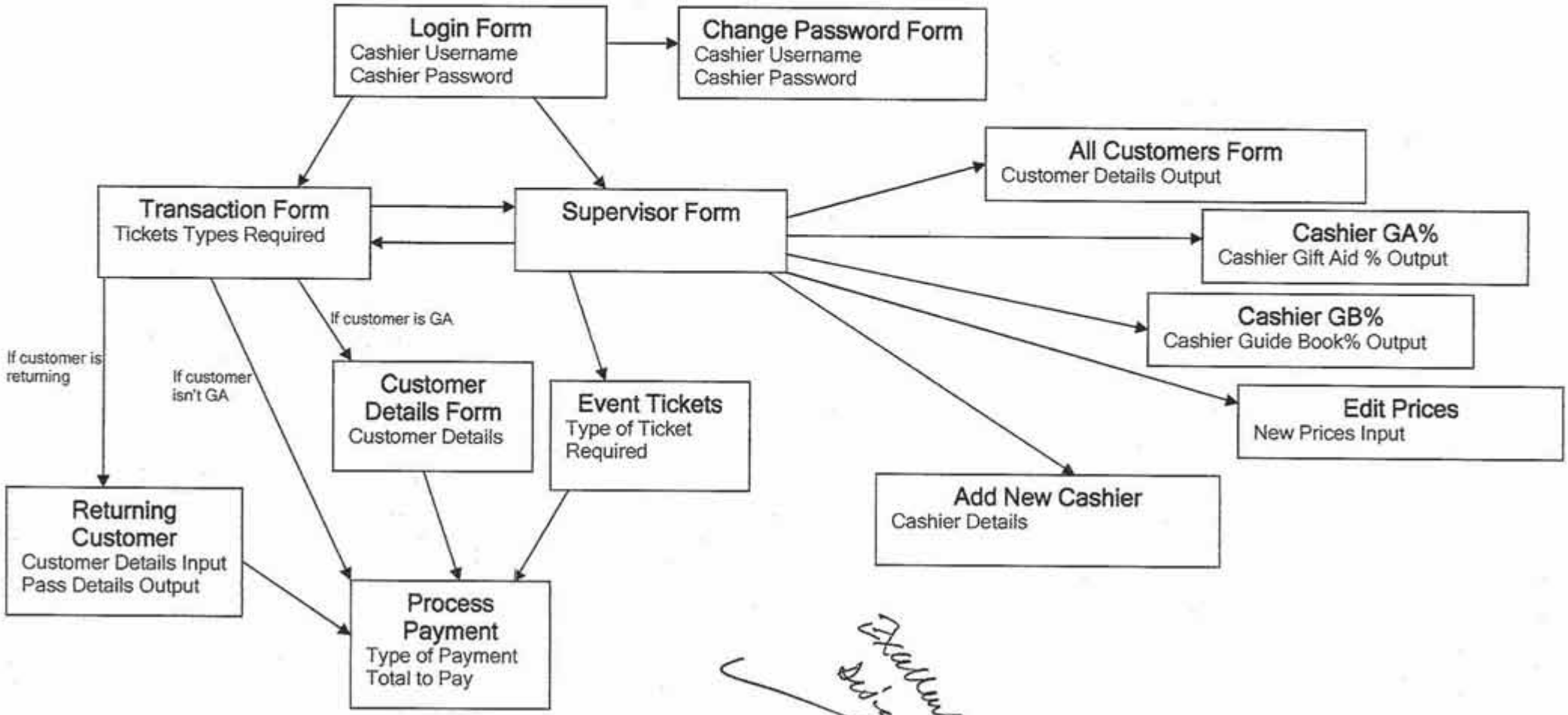
TILL TOTAL =

Gift Aid (%) =

Guide Book (%) =

Flow Chart

The chart below shows how the forms all link together. It also shows the details that will be input and output.



Event Actions

The steps below show the actions that will be executed from each form.

Login Form

- When the form loads the data file containing the user's password will be uploaded for use
- When the user clicks login the user name will be found in the data file
- If found the password will be compared with the one in the data file
- If correct the next form will load. The form to be loaded will depend on who has logged in
- If a cashier has logged in then the transaction form will be loaded
- If a manager or supervisor logs in then the main menu form will be loaded
- If the username can't be found an error message will be displayed
- If the password is incorrect an error message will be displayed
- If the user selects change password then the change password form will be loaded

Transaction Form

- When the cashier has entered the types of ticket that they require the total in the number of people label will be updated and the sub total to pay will be updated
- When the cashier selects Process the payment type form is loaded. If a supervisor is logged in the command button providing the link to the supervisor menu will be available to click, if they do click it the main menu will load
- The data for the transaction will be sent to the variables ready to be saved to the file when the transaction has been completed

Change Password

- Cashier enters their current user ID and password and new password twice
- When they click submit the passwords data file will be called and checked to ensure that the password that has been given is correct
- If they push cancel they will be returned to the login form

Customer Details

- The cashier will enter the customers address into the relevant text boxes, if the cashier enters an invalid piece of data a message box will be displayed informing them of an error
- The cashier must then check that the customers details are correct with the customer

- If the details are correct a message will be displayed to say that they have been accepted and the transaction form will then be loaded again

Returning Customers

- The cashier will enter the customers name and address and select submit. A search will then be carried out in the data files to check that the customer is returning
- If there is a match then the customers details will be displayed in the labels at the bottom of the form, if the customer is upgrading their ticket then the total that they have left to pay will also be displayed
- If there is no match a message will be displayed stating that no match could be found and will then set the focus to the surname text box so that the postcode can be checked and entered again
- If the cashier pushes cancel then the transaction form will be loaded again
- If the cashier selects Process then the type of payment form will be loaded

Add New Cashier

- The supervisor will enter the new cashier details
- If they push main menu they will be returned to the supervisors screen
- If they push submit then the information that they have entered will saved in the cashier data file

Supervisor and Manager Screen

- The various command buttons on the main menu will allow the user to navigate to different forms depending on what task they want to carry out

Event Tickets

- When the cashier has entered the types of ticket that they require the total in the number of people label will be updated and the sub total to pay will be updated
- When the cashier selects Process the payment type form is loaded. If a supervisor is logged in the command button providing the link to the supervisor menu will be available to click, if they do click it the main menu will load
- The data for the transaction will be sent to the variables ready to be saved to the file when the transaction has been completed

Edit Prices

- When the form is loaded the current prices will be loaded into the relevant text boxes
- The user can then change the values in the text boxes
- If they click save then the information in the ticket data file will be updated
- They can also click the main menu button to return to the main menu form, or they can click on log out to return to the login form

Process Payment

- This form will give the user the chance to click on one of two command buttons
- If they click on Cash then the cash payment form will be loaded
- If they click on Card then the card payment will be accepted and the transaction will be saved, the cashier will be returned to the transaction form

Cash Payment

- The amount due will be displayed in the label when the form is loaded
- The cashier can then enter the amount of cash that they have been given in the Amount Tender text box
- They can also push the command buttons to increase the value in the text box without having to type the information in
- The amount of change due will be calculated when the user selects OK, it will then be displayed in the change due label
- When the user selects the Process the transaction will be saved and the cashier will be returned to the transaction form

Customer Details Output Forms

- When a manager or supervisor logs in using the log in forms the data from the files necessary will be uploaded into the output forms for the manager to view as required

GA and Guidebook Percentage Forms

- When the form loads the cashier details will be loaded into the list boxes and the percentage will be calculated
- The cashier can also push the main menu button to return to the main menu form

Validation

To try and aid the accuracy of the data that the cashiers are going to enter I will set some validation rules on the fields where data is entered.

Login Form

txt_userID
- can only be numeric
- max length = 4 digits

txt_password
- maximum of 5 characters

Transaction Form

All the Text Boxes
- the maximum value that can be entered into all of the text boxes will be 250 as this is the largest group that would expect to take in, in a single transaction
- the data entered must be numeric, (only if the cashier is not a supervisor)

Change Password

txt_userID
- can only be numeric
- max length = 4 digits

txt_password, txt_new1, txt_new2
- maximum of 5 characters

Customer Details Form

txt_firstname, txt_surname
- maximum 20 characters
- can't be left blank

txt_house, txt_street, txt_town, txt_county
- maximum 20 characters
- can't be left blank

txt_postcode
- maximum 8 characters
- can't be left blank

txt_email
- maximum 20 characters

Returning Customer Form

txt_surname
- maximum 20 characters
- can't be left blank

txt_postcode
- maximum 8 characters
- can't be left blank

Add New Cashier

txt_cashiername
- must be text
- maximum 8 characters

txt_userID
- can only be numeric
- max length = 4 digits

txt_password, txt_password2
- maximum of 5 characters

Event Tickets

All the Text Boxes
- the maximum value that can be entered into all of the text boxes will be 250 as this is the largest group that would expect to take in, in a single transaction
- the data entered must be numeric, (only if the cashier is not a supervisor)

Edit Prices

All the Text Boxes
- the value entered can not exceed £50
- all the values must be numeric

Cash Payment

txt_amounttender
- must be numeric, (only if the cashier is not a supervisor)

Analysis of Design Methods

Design Methods Used

I used several different design methods to create a design for the system. One of the methods I used was flow charts. I created two different types of flow chart. The first one I created using a program called Superlite. This flow chart showed the process of carrying out a transaction and the different steps that are taken. The second flow chart I made shows how the user will move between the different forms.

Both of these will help me when implementing the system as they will be able to ensure that the implemented forms follow the correct stages to allow the user to work through all of the steps correctly.

After I had identified the entities that will be needed in the system I created an Entity Relationship Diagram (ERD). This diagram allows me to see clearly how the different entities will be interacting within the system and the different attributes that will be stored about each entity.

From this I was able to create a data dictionary this again shows the information that is stored about each entity but it also shows the data type, length and validation. This will help when implementing the system as I will know what validation is needed on each data entry point. It will also help as I can see the size of each record that is going to be stored.

Another design method I used was designing the forms. By designing the forms I was able to think about how the data is going to be entered and displayed as well as ensuring that it is in a user friendly format. Having designed the forms first I will be able to implement the forms much more easily.

I then went on to think about the different events that will need to be carried out. I have generally stated what processes will be carried out when the user triggers a certain action. This will help when I am programming the system as I will know what actions need to be set up on the different objects.

The final design method I carried out was by creating an additional list of validation steps. From the validation specified in the data dictionary I thought about where the data will be entered into the different forms and also recorded the individual validation rules that will need to be implemented on each of the objects on the forms.

Strengths and Weaknesses

Having completed the design section I believe that there are both strengths and weaknesses with the designs.

I think that the two flow charts are very strong as they show in a clear pictorial form how the different forms are linked together. This would enable someone else to use the design work to implement the form structure.

Another strength is the form designs themselves as I have thought carefully about which objects would be most suited for the required task. For example I am using list boxes to view the customer details. Also again someone else would be able to pick up my form designs and create the forms.

A weakness of my design is that I didn't use pseudocode to show the event actions of validation steps. Whilst I did create bullet pointed lists of what needs to be carried out it isn't in the format of pseudocode. If I had done this then it would have made it easier for someone else to use the design to program the system.

Changes Made

When creating the forms in Visual Basic I did add some additional objects that I hadn't specified in my designs. For example in some cases I had missed out a command button that would be required to allow the user to successfully move through the system. I was able to use the systems flow charts to help me when find out which buttons I needed to add.

I also thought about the colours when implementing the forms. I thought about using different colour backgrounds and tried different colours when I was implementing the forms. I eventually settled on using white and black as I felt that this made it clear for the user to be able to see the forms and would effect any users who may be colour blind.

Action Plan

The table below shows the action plan needed to complete the project.

Week	Date	Task	Home or College	Comments (What Next?)
Introduction and Program Specification				
1	22/02/10	Write introduction to the problem; identify the inputs and outputs required for the system. Generate user requirements for the system.	College and Home	Continue thinking about the inputs and outputs.
2	1/03/10	Finish identifying the inputs, processes and outputs.	College	Move on to the design section
Program Design				
2	1/03/10	Began the design section and create a systems flow chart.	College	Create the ERD.
2	5/03/10	Complete the entity relationship diagram.	College	Make a data dictionary.
2	07/03/10	Begin the data dictionary.	Home	Continue with data dictionary.
3	08/03/10	Finish the data dictionary.	College	Create the form designs.
3	12/03/10	Began the form designs.	College	Finish form designs and Z report designs.
4	15/03/10	Complete the form designs and created the Z report designs.	Home	Create the flow chart.
4	15/03/10	Complete the flow chart.	College	Make a list of all the event actions.
4	19/03/10	Start to create a list of the event actions for each form.	College	Continue with event actions.
4	21/03/10	Complete event action list.	Home	Create list of validation steps.
5	22/03/10	Create a list of the validation steps.	College	Create a detailed test plan.
5	23/03/10	Create a detailed test plan.	Home	Analyse the design methods used.
5	24/03/10	Analyse the design methods used.	College	Start implementation.
Implementation of Program				
5	26/03/10	Implement the form designs.	Home	Set up the module and declare variables.
6	29/03/10	Set up the module and declare the variables and each of the arrays.	College	Continue working through the forms and implement the code.
6	2/04/10	Start coding the forms and work through each form. Code the transaction form.	College	Code the customer details form.
6	3/04/10	Code the customer details form.	Home	Code the returning customer form.
6	4/04/10	Code the returning details form.	Home	Code the login and change password forms.
7	5/04/10	Code the login and change password forms.	College	Code the supervisor and manager forms.
7	5/04/10	Code the supervisor and manager form. Also code the add new cashier form.	College	Code the event ticket form.
8	12/4/10	Code the event ticket form.	Home	Code the edit prices form.

8	13/04/10	Code the edit prices form.	Home	Code the form to process the payments.
8	16/04/10	Code the form to process the payment and cash payment.	College	Code the output forms and Z report.
8	17/04/10	Code the output forms and Z report.	Home and College	Begin the testing.
Program Testing				
9	23/04/10	Using the test plan created in the design section work through the first few tests.	Home	Continue with the testing.
10	24/04/10	Continued with the testing.	Home	Complete the testing.
10	25/04/10	Finish the testing and carried out the necessary changes after testing.	College	Complete the evaluation.
Evaluation and review report				
10	28/04/10	Began the evaluation section.	Home	Continue with the evaluation.
10	30/04/10	Complete the evaluation section.	Home	
Deadline for handing in: April 30th 2010				

TASK C

IMPLEMENTATION AND TECHNICAL DOCUMENTATION

Code Listings

The pages below show the code for the module and each of the forms used in the program.

Module

```
'Declares the variables used for each cashier
Public Type CashierType
    cashiername As String * 15
    cashierID As Integer
    cashierpassword As String * 5
    supervisor As Boolean
End Type

'Declares the variables used for each customer
Public Type CustomerType
    customerID As Integer
    firstname As String * 15
    surname As String * 15
    house As String * 15
    postcode As String * 8
    street As String * 20
    town As String * 15
    county As String * 15
    emailaddress As String * 25
    mailings As Boolean
    dateofentry As Date
End Type

'Declares the variables used for each transaction
Public Type TransactionType
    transactionID As Integer
    cashierID As Integer
    customerID As Integer
    amountpaid As Currency
    giftaid As Boolean
    noofadults As Integer
    noofchildren As Integer
    typeofpayment As String * 1
    dateoftransaction As Date
    guidebook As Integer
End Type

'Declares the variables used for each ticket
Public Type TicketType
    typeofticket As String * 7
    ticketprice As Currency
End Type

'Declares the arrays that are used to save the information about the
entities
Public cashier(1 To 50) As CashierType
Public customer(1 To 1000) As CustomerType
Public transaction(1 To 1000) As TransactionType
```

```
Public ticket(1 To 12) As TicketType
Public transactioncount As Integer
Public customercount As Integer
Public cashiercount As Integer
Public ticketcount As Integer
Public cashiernum As Integer
Public subtotal As Currency
Public eventticket As Boolean
```

✓ comments.

```
'code for disabling the X so that the user can not close the window,
taken from VB Forums

Public Declare Function GetMenuItemCount Lib "user32" (ByVal hMenu As
Long) As Long
Public Declare Function GetSystemMenu Lib "user32" (ByVal hwnd As
Long, ByVal bRevert As Long) As Long
Public Declare Function RemoveMenu Lib "user32" (ByVal hMenu As Long,
ByVal nPosition As Long, ByVal wFlags As Long) As Long
Public Declare Function DrawMenuBar Lib "user32" (ByVal hwnd As Long)
As Long
Public Const MF_REMOVE = &H1000&
Public Const MF_INSERT = &H0&
Public Const MF_ENABLED = &H0&
Public Const MF_BYPOSITION = &H400&

Public Sub loadcashiers()
Dim filenumber As Integer
Dim onecashier As CashierType
Dim i As Integer

'Finds the file and opens the data one cashier at a time
filenumber = FreeFile
Open App.Path & "\Cashiers.dat" For Random As filenumber Len =
Len(onecashier)
   cashiercount = LOF(filenumber) / Len(onecashier)
    If cashiercount <> 0 Then
        For i = 1 To cashiercount
            Get #filenumber, i, cashier(i)
        Next i
    End If
Close #filenumber

End Sub
```

✓ file handling.

```
Public Sub savenewcashier(p)
```

✓ subroutines with parameters.

```
Dim filenumber As Integer
Dim onecashier As CashierType

'Sets the file path and data to be saved
filenumber = FreeFile
Open App.Path & "\Cashiers.dat" For Random As filenumber Len =
Len(onecashier)
      Put #filenumber, p, cashier(p)
Close #filenumber

End Sub


Public Sub loadcustomers()
```

```
Dim filenumber As Integer
Dim onecustomer As CustomerType
Dim i As Integer

'Finds the file and opens the data one customer at a time
filenumber = FreeFile
Open App.Path & "\Customers.dat" For Random As filenumber Len =
Len(onecustomer)
   customercount = LOF(filenumber) / Len(onecustomer)
    If customercount <> 0 Then
        For i = 1 To customercount
            Get #filenumber, i, customer(i)
        Next i
    End If
Close #filenumber

End Sub

Public Sub savenewcustomer(i)
Dim onecustomer As CustomerType

'Sets the file path and data to be saved
filenumber = FreeFile
Open App.Path & "\Customers.dat" For Random As filenumber Len =
Len(onecustomer)
      Put #filenumber, i, customer(i)
Close #filenumber

End Sub


Public Sub loadtransactions()

Dim filenumber As Integer
Dim onetransaction As TransactionType
Dim i As Integer

'Finds the file and opens the data one transaction at a time
filenumber = FreeFile
Open App.Path & "\Transactions.dat" For Random As filenumber Len =
Len(onetransaction)
   transactioncount = LOF(filenumber) / Len(onetransaction)
    If transactioncount <> 0 Then
        For i = 1 To transactioncount
            Get #filenumber, i, transaction(i)
        Next i
    End If
Close #filenumber

End Sub

Public Sub savenewtransaction(p)

Dim onetransaction As TransactionType

'Sets the file path and data to be saved
filenumber = FreeFile
Open App.Path & "\Transactions.dat" For Random As filenumber Len =
Len(onetransaction)
      Put #filenumber, p, transaction(p)
Close #filenumber
```

```
End Sub


Public Sub loadtickets()

Dim filenumber As Integer
Dim oneticket As TicketType
Dim i As Integer

'Finds the file and opens the data one transaction at a time
filenumber = FreeFile
Open App.Path & "\Tickets.dat" For Input As filenumber
   ticketcount = 12
    'If ticketcount <> 0 Then
        For i = 1 To ticketcount
            Input #filenumber, ticket(i).typeofticket,
ticket(i).ticketprice

        Next i
    'End If
Close #filenumber

End Sub

Public Sub savenewticket(h)

Dim oneticket As TicketType

'Sets the file path and data to be saved
filenumber = FreeFile
Open App.Path & "\Tickets.dat" For Append As filenumber
      Write #filenumber, ticket(h).typeofticket,
ticket(h).ticketprice
Close #filenumber

End Sub


'sub routine to prevent the user from closing the form taken from VB
Forums

Public Sub DisableX(frm As Form, blnDisabled As Boolean)
    Dim hMenu As Long
    Dim nCount As Long

    If blnDisabled = True Then
        hMenu = GetSystemMenu(frm.hwnd, 0)
        nCount = GetMenuItemCount(hMenu)
        Call RemoveMenu(hMenu, nCount - 1, MF_REMOVE Or
MF_BYPOSITION)
        Call RemoveMenu(hMenu, nCount - 2, MF_REMOVE Or
MF_BYPOSITION)
        DrawMenuBar frm.hwnd
    Else
        hMenu = GetSystemMenu(frm.hwnd, True)
        DrawMenuBar frm.hwnd
    End If

End Sub
```

✓ User-defined subroutines.

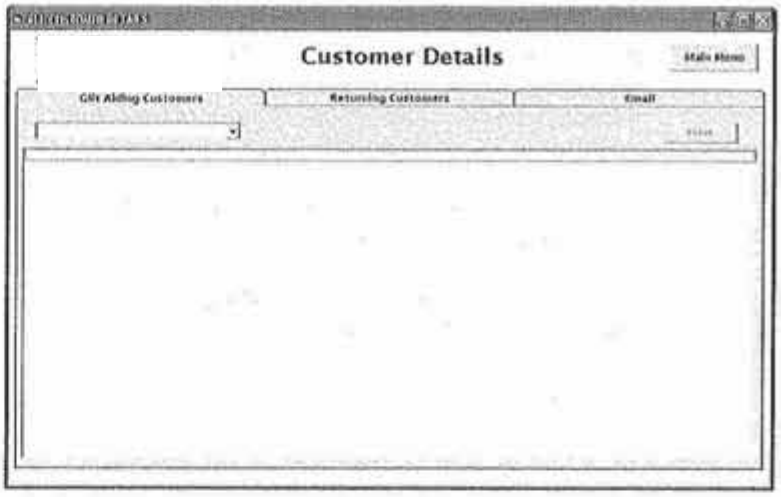
Add New Cashier


```
Private Sub cmd_logout_Click()

'calls the next form
Load frm_login
frm_login.Show
Unload Me

End Sub

Private Sub cmd_mainmenu_Click()

'calls the next form
Load frm_mainmenu
frm_mainmenu.Show
Unload Me

End Sub

Private Sub cmd_submit_Click()

'Ensures that the cashier name doesn't include any numbers
If IsNumeric(txt_cashiername)' Then
    MsgBox (" Invalid Cashier Name!")
    txt_cashiername = ""
    txt_password = ""
    txt_password2 = ""
    txt_userID = ""
    chk_supervisor = False

Else

    'Prevents the cashier from entering a UserID of less than 4
integers
    If Val(txt_userID) < 1 Then
        MsgBox ("Please enter a valid UserID")
    Else


        If txt_cashiername.Text = "" Then
            MsgBox (" Invalid Cashier Name!")
            txt_cashiername = ""
            txt_password = ""
            txt_password2 = ""
            txt_userID = ""
            chk_supervisor = False
        Else

                'Compares the two passwords to make sure that they
are the same
                If txt_password.Text = txt_password2.Text And
Len(txt_password.Text) = 5 Then

                    namefound = False
                    'Makes sure that only one cashier has one ID at
any time
                    For c = 1 To cashiercount
                        With cashier(c)
                            If txt_userID.Text = .cashierID Then
```

```
                                namefound = True
                            End If
                        End With
                    Next c
                    If namefound = True Then
                        MsgBox ("Please choose another UserID")
                    Else

                                cashiercount = cashiercount + 1
                                'Takes the cashier details from the
text boxes to send to the data file
                                With cashier(cashiercount)
                                    .cashierID = txt_userID.Text
                                    .cashiername =
txt_cashiername.Text
                                    .cashierpassword =
txt_password.Text
                                    .supervisor = Val(chk_supervisor)

                                End With

                            'Calls the sub routine that will save the
data
                            Call savenewcashier(cashiercount)

                            'Displays a message box to say that the
details have been saved
                            MsgBox ("Cashier Details Accepted!")

                    End If

                Else

                        'Else a message box is displayed saying the
password is incorrect
                        MsgBox ("Invalid username or password, please
rekey")
                        txt_username = ""
                        txt_password = ""
                        txt_password2 = ""
                        chk_supervisor = False
            End If
        End If
    End If
    'Clears the text box for the next user
    txt_cashiername = ""
    txt_userID = ""
    txt_password = ""
    txt_password2 = ""
    chk_supervisor = False

    End If
End Sub

Private Sub txt_cashiername_KeyPress(KeyAscii As Integer)

'Prevents the cashier from entering a number or invalid character in
the customers name
```

```
If (KeyAscii < 65 Or KeyAscii > 90) And (KeyAscii < 97 Or KeyAscii >
122) And KeyAscii <> 32 And KeyAscii <> 8 Then
    MsgBox "Only Letters Allowed"
    KeyAscii = 0
End If
End Sub


Private Sub Form_Load()

'Calls the subroutine to load the cashiers from the file
Call loadcashiers

End Sub


Private Sub txt_userID_KeyPress(KeyAscii As Integer)

'Checks to ensure that the user can only enter a number, space or
decimal point
If (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 8 Then
    MsgBox ("Please enter a valid User ID")
    txt_userID = ""
    txt_userID.SetFocus
End If


End Sub

Private Sub txt_userID_LostFocus()

'ensures that the user must enter a user ID of 4 integers
If Len(txt_userID) <> 4 Then
    MsgBox ("Please enter a valid User ID")
    txt_userID = ""
    txt_userID.SetFocus
End If

End Sub
```

All Customers

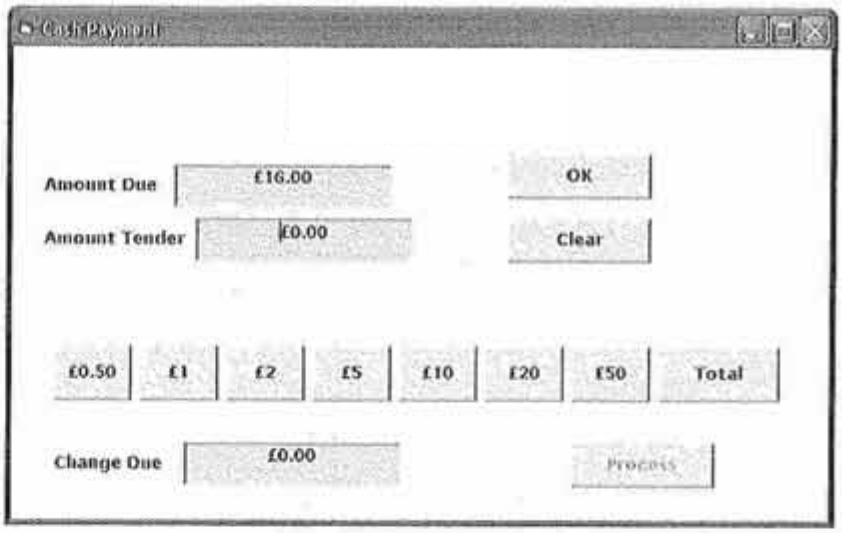



```
Dim monthnum As Integer
Dim monthnum2 As Integer
Dim returnmonthnum As
Integer
Dim camountpaid As Currency

Private Sub cmd_home_Click()

    'calls the next form
    Load frm_mainmenu
    frm_mainmenu.Show
    Unload Me

End Sub


Public Sub displaycustomers()

'Calls all the list boxes
lst_return.Clear
lst_giftaiding.Clear
lst_email.Clear

'For each customer display their information on the correct lists
For i = 1 To customercount
    With customer(i)

        'Loads the gift aiding customers address into the list box
        If DatePart("m", .dateofentry) = monthnum Then
            'Call findamountpaid(i)
            addresstoshow = .firstname & .surname & .house & .street
& .town & .county & .postcode & "  " & .dateofentry
            lst_giftaiding.AddItem addresstoshow
        End If

        'Trims the email address
        email = RTrim(.emailaddress)

        'If the trimed value is not 0 then display the other email
address'
        If Len(email) <> 0 Then
            emailtoshow = .firstname & .surname & .emailaddress &
.mailings
            lst_email.AddItem emailtoshow
        End If

        'Loads the returning customers into the list box
        'If .returning = True And DatePart("m", .joindate) =
returnmonthnum Then
            'returnstoshow = .customerID & "  " & .Title & "  " &
.firstname & "  " & .surname & "  " & .joindate & "  " & .numreturns
            'lst_return.AddItem returnstoshow
        'End If

    End With
Next i


End Sub
```

use of many built-in functions eg. 'DatePart' 'Val' 'Len' etc

```
Private Sub cbo_month_Click()

'Adds one to the list index when the month is selected
monthnum = cbo_month.ListIndex + 1

'Calls the sub routine to display the customer details
Call displaycustomers

'Enables the printing button
cmd_print1.Enabled = True

End Sub
```

Cash Payment



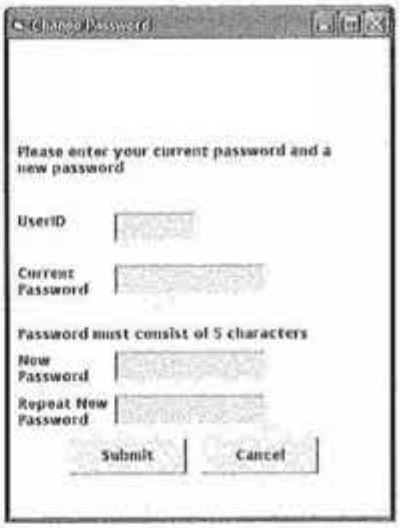

```
Dim r As Currency

Private Sub
cmd_1_Click()

'calls the subroutine
to add £1 to the amount
tender
r = 1
Call
calculateamounttender

End Sub

Private Sub cmd_10_Click()

'calls the subroutine to add £10 to the amount tender
r = 10
Call calculateamounttender

End Sub

Private Sub cmd_2_Click()

'calls the subroutine to add £2 to the amount tender
r = 2
Call calculateamounttender

End Sub

Private Sub cmd_20_Click()

'calls the subroutine to add £20 to the amount tender
r = 20
Call calculateamounttender

End Sub

Private Sub cmd_5_Click()

'calls the subroutine to add £5 to the amount tender
r = 5
Call calculateamounttender

End Sub

Private Sub cmd_50_Click()

'calls the subroutine to add £50 to the amount tender
r = 50
Call calculateamounttender

End Sub

Private Sub cmd_50p_Click()

'calls the subroutine to add £0.50 to the amount tender
r = 0.5
Call calculateamounttender
```

```
End Sub

Private Sub cmd_clear_Click()

'sets the values in the labels back to 0
txt_amounttender = FormatCurrency(0)
lbl_change = FormatCurrency(0)


End Sub

Private Sub cmd_ok_Click()

'gets the values from the labels
i = lbl_amountdue
j = txt_amounttender


'calculates the amount of change required from the values in the
amount due and the amount tender labels
If i > j Then
    MsgBox ("Please enter a valid amount of tender!")
    txt_amounttender = FormatCurrency(0)
Else
    lbl_change.Caption = FormatCurrency(j - i)
    cmd_process.Enabled = True

End If


End Sub

Private Sub cmd_process_Click()

If eventticket = True Then

    'accpets the payment
    MsgBox ("Payment Accepted!")

    'calls the next form
    Load frm_event
    frm_event.Show
    Unload Me

Else

    'accpets the payment
    MsgBox ("Payment Accepted!")

    'calls the next form
    Load frm_transaction
    frm_transaction.Show
    Unload Me
End If

End Sub

Private Sub cmd_total_Click()

'adds the total of the amountdue to the amounttender
```

```
If Val(lbl_amountdue) > Val(txt_amounttender) Then
    MsgBox ("Please enter a valid amount of tender!")
Else
    txt_amounttender = lbl_amountdue
End If

End Sub

Private Sub Form_Load()

'calls the sub routine from the module to disable the X so that the
user can not close the form
Call DisableX(Me, True)

'sets the labels to 0 when the form loads
txt_amounttender = FormatCurrency(0)
lbl_change = FormatCurrency(0)
lbl_amountdue = FormatCurrency(subtotal)

End Sub

Sub calculateamounttender()

'calculates the amount given based on the comman buttons that have
been selected
p = txt_amounttender
p = p + r

'sends the calculated value back to the label
txt_amounttender = FormatCurrency(p)
r = 0
p = 0


End Sub

Private Sub txt_amounttender_KeyPress(KeyAscii As Integer)

'Checks to ensure that the user can only enter a number, space or
decimal point
If (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 8 And KeyAscii <>
46 Then
    MsgBox ("Please enter a valid amount of tender")
    txt_amounttender = FormatCurrency(0)
End If

End Sub
```

✓ Userdefined Sub.

✓ Validation of key presses.



Change Password

```
Private Sub cmd_cancel_Click()

'calls the next form
Load frm_login
frm_login.Show
Unload Me

End Sub

Private Sub cmd_submit_Click()

'Checks to see that the user has not left
the form blank
If txt_userID = "" Or txt_current = "" Or
txt_new1 = "" Or txt_new2 = "" Then
    MsgBox ("Please enter a valid user Id or password")
Else

    'Declares the variables used within this subroutine
    Dim filenumber As Integer
    Dim onecashier As CashierType
    Dim i As Integer
    Dim foundmatch As Boolean
    foundmatch = False
    'Closes any open files
    Close #0

    filenumber = FreeFile
    'Opens the file to get the data from
    Open App.Path & "\Cashiers.dat" For Random As filenumber Len =
Len(onecashier)
    'Calculates the number of cashiers in the file
       cashiercount = LOF(filenumber) / Len(onecashier)
       'If the cashier count is greater then 0 move through the
records to find a match
        If cashiercount <> 0 Then
          For i = 1 To cashiercount
            Get #filenumber, i, cashier(i)
            With cashier(i)

                'Checks to make sure that the password and username
are correct
                  If .cashierID = Val(txt_userID.Text) And
.cashierpassword = txt_current.Text Then
                       foundmatch = True

                       'Checks to make sure that both copies of the
new password are the same
                       If txt_new1.Text = txt_new2.Text Then
                           .cashierpassword = txt_new1.Text

                           'Calls the subroutine to save the
password
                           Call savenewcashier(i)
                           MsgBox ("Cashier Details Accepted!")
                           txt_userID = ""
                           txt_current = ""
                           txt_new2 = ""
                           txt_new1 = ""
```

```
                        Else

                            'If the passwords are different a message
box is displayed
                            MsgBox ("Invalid Password!")
                            txt_userID = ""
                            txt_current = ""
                            txt_new2 = ""
                            txt_new1 = ""

                        End If
                    End If
            End With
          Next i

      Close #filenumber

        If foundmatch = False Then
                'Else gives a message box to say that the username or
password is invalid
                MsgBox ("Invalid username or password!")
                txt_userID = ""
                txt_new1 = ""
                txt_new2 = ""
                txt_current = ""
        End If
    End If
End If
End Sub
```

Customer Details

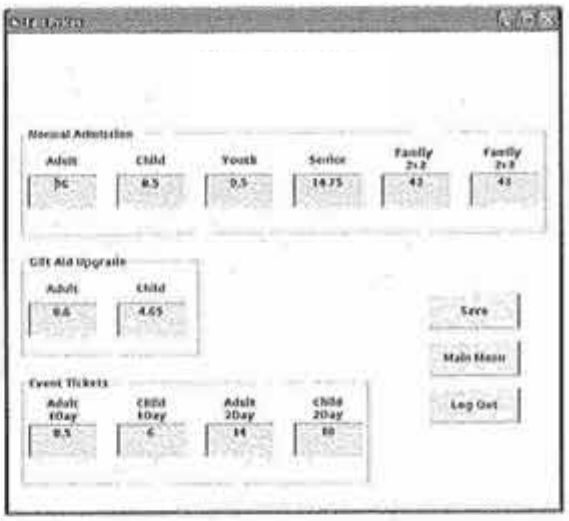
```
Private Sub cmd_save_Click()

'gives a message box to check whether or not
the customer is a UK tax payer or not
Style = vbYesNo + vbQuestion
message = "Is the customer a UK tax payer?"
Title = "Tax Payer?"
response = MsgBox(message, Style, Title)

If response = vbYes Then

        'Makes sure that a valid first name
had beeen entered
        If txt_firstname.Text = "" Then
            MsgBox ("Please enter a valid
firstname")
            txt_firstname.Text = ""
        Else

            'Makes sure that a valid surname has been entered
            If txt_surname.Text = "" Then
                MsgBox ("Please enter a valid surname")
                txt_surname.Text = ""
            Else

                'Makes sure that a valid house number or name has
been entered
                If txt_house.Text = "" Then
                    MsgBox ("Please enter a valid house number or
name")
                    txt_house.Text = ""
                Else

                        'Makes sure that the street can't be numeric
                        If IsNumeric(txt_street) Then
                            MsgBox ("Please enter a valid street
name")
                            txt_street.Text = ""
                        Else

                        'Makes sure that a postcode has been entered
                        If txt_postcode.Text = "" Then
                            MsgBox ("Please enter a valid postcode")
                            txt_postcode.Text = ""
                        Else

                        customercount = customercount + 1

                         'Formats the customer details
                            txt_postcode = Format(txt_postcode, ">")

                            'Takes the customer details from the text
boxes to send to the data file
                            With customer(customercount)
                                .customerID = customercount
                                .firstname = txt_firstname.Text
                                .surname = txt_surname.Text
                                .house = txt_house.Text
                                .street = txt_street.Text
```

```
                                    .town = txt_town.Text
                                    .county = txt_county.Text
                                    .postcode = txt_postcode.Text
                                    .emailaddress = txt_email


                                    'Sets the boolean value to save for
the mailings list
                                    If chk_mailings = 1 Then
                                        .mailings = True
                                    Else
                                        .mailings = False
                                    End If

                                    .dateofentry = Date



                                End With

                                With transaction(transactioncount)
                                    .customerID = customercount
                                    .giftaid = True
                                End With

                                'Calls the sub routine that will save the
data
                                Call savenewcustomer(customercount)

                                'Displays a message box to say that the
details have been saved
                                MsgBox ("Customer Details Accepted!")

                                'calls the next form
                                Load frm_processpayment
                                frm_processpayment.Show
                                Unload Me

                            End If
                        End If
                    End If
                End If
            End If


Else

   With transaction(transactioncount)
        .customerID = 0
        .giftaid = False
    End With
    'calls the next form
Load frm_processpayment
frm_processpayment.Show
Unload Me
End If
```

```
End Sub

Private Sub cmd_skip_Click()

'calls the next form
Load frm_processpayment
frm_processpayment.Show
Unload Me

    With transaction(transactioncount)
        .customerID = 0
        .giftaid = False
    End With

End Sub

Private Sub Form_Load()


'calls the sub routine from the module to disable the X so that the
user can not close the form
Call DisableX(Me, True)

Call loadcustomers

End Sub

Private Sub txt_firstname_KeyPress(KeyAscii As Integer)

'Prevents the cashier from entering a number or invalid character in
the customers name
If (KeyAscii < 65 Or KeyAscii > 90) And (KeyAscii < 97 Or KeyAscii >
122) And KeyAscii <> 32 And KeyAscii <> 8 Then
    MsgBox "Only Letters Allowed"
    KeyAscii = 0
End If

End Sub

Private Sub txt_surname_KeyPress(KeyAscii As Integer)

'Prevents the cashier from entering a number or invalid character in
the customers name
If (KeyAscii < 65 Or KeyAscii > 90) And (KeyAscii < 97 Or KeyAscii >
122) And KeyAscii <> 32 And KeyAscii <> 8 Then
    MsgBox "Only Letters Allowed"
    KeyAscii = 0
End If

End Sub

Private Sub txt_street_KeyPress(KeyAscii As Integer)

'Prevents the cashier from entering a number or invalid character in
the customers details
If (KeyAscii < 65 Or KeyAscii > 90) And (KeyAscii < 97 Or KeyAscii >
122) And KeyAscii <> 32 And KeyAscii <> 8 Then
    MsgBox "Only Letters Allowed"
    KeyAscii = 0
End If
```

```
End Sub

Private Sub txt_town_KeyPress(KeyAscii As Integer)

'Prevents the cashier from entering a number or invalid character in
the customers details
If (KeyAscii < 65 Or KeyAscii > 90) And (KeyAscii < 97 Or KeyAscii >
122) And KeyAscii <> 32 And KeyAscii <> 8 Then
    MsgBox "Only Letters Allowed"
    KeyAscii = 0
End If

End Sub

Private Sub txt_county_KeyPress(KeyAscii As Integer)

'Prevents the cashier from entering a number or invalid character in
the customers details
If (KeyAscii < 65 Or KeyAscii > 90) And (KeyAscii < 97 Or KeyAscii >
122) And KeyAscii <> 32 And KeyAscii <> 8 Then
    MsgBox "Only Letters Allowed"
    KeyAscii = 0
End If

End Sub
```

Edit Prices



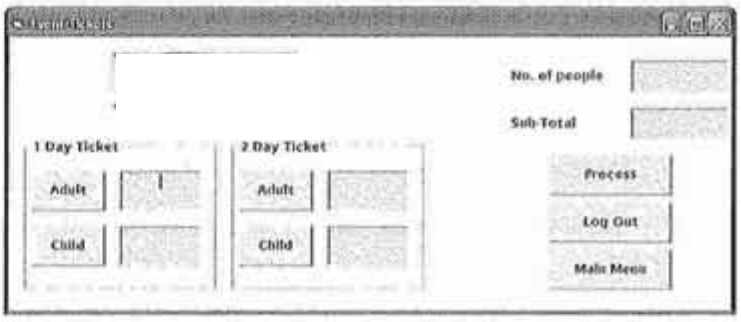

```
Private Sub cmd_logout_Click()

'calls the next form
Load frm_login
frm_login.Show
Unload Me

End Sub

Private Sub cmd_mainmenu_Click()

'calls the next form
Load frm_mainmenu
frm_mainmenu.Show
Unload Me

End Sub

Private Sub cmd_save_Click()

'deletes the existing ticket price file
Kill App.Path & "\Tickets.dat"
```

File deletion

```
'works through each of the text boxes in the array and collects the
amount from the text box
For t = 1 To 12

    With ticket(t)
        .ticketprice = txt_ticketprice(t - 1)
    End With

    'calls the sub routine to save the new ticket
    Call savenewticket(t)
    MsgBox ("New Prices have been saved!")

Next t


End Sub

Private Sub Form_Load()

'calls the sub routine the load the tickets
Call loadtickets

'works through the text boxes in the array and adds the value of each
ticket to the correct text box
For t = 1 To 12

   With ticket(t)
            txt_ticketprice(t - 1) = .ticketprice

  End With
Next t

End Sub
```

```
Private Sub txt_ticketprice_KeyPress(Index As Integer, KeyAscii As
Integer)

'Checks to ensure that the user can only enter a number, space or
decimal point
If (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 8 And KeyAscii <>
46 Then
    MsgBox ("Please enter a valid amount of tender")
End If

End Sub
```

Event Tickets

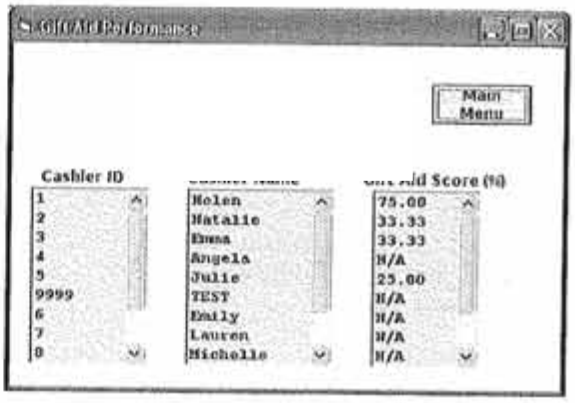



```
Private Sub
cmd_adult1_Click()

    'Adds one to the
adult quantity
    i = Val(txt_adult1)
    i = i + 1
    txt_adult1 = i

End Sub

Private Sub cmd_logout_Click()

'calls the next form
Load frm_login
frm_login.Show
Unload Me

End Sub

Private Sub cmd_process_Click()

eventticket = True

'calls the next form
Load frm_processpayment
frm_processpayment.Show
Unload Me

End Sub

Private Sub cmd_supervisor_Click()

'calls the next form
Load frm_mainmenu
frm_mainmenu.Show
Unload Me

End Sub


Private Sub Form_Load()

'calls the sub routine to load the tickets out of the ticket file
Call loadtickets

End Sub

Private Sub txt_adult1_Change()

'calls the subroutine to recalculate the total number of people
Call calcnoofpeople
Call calceventsubtotal

End Sub

Sub calcnoofpeople()
```

```
'adds the values in all of the text boxes to calculate the total
number of people
lbl_noofpeople = Val(txt_adult1) + Val(txt_child1) + Val(txt_adult2)
+ Val(txt_child2)

End Sub

Private Sub cmd_child1_Click()

    'Adds one to the adult quantity
    i = Val(txt_child1)
    i = i + 1
    txt_child1 = i

End Sub

Private Sub txt_child1_Change()

'calls the subroutine to recalculate the total number of people
Call calcnoofpeople
Call calceventsubtotal

End Sub


Private Sub cmd_child2_Click()

    'Adds one to the adult quantity
    i = Val(txt_child2)
    i = i + 1
    txt_child2 = i

End Sub

Private Sub txt_child2_Change()

'calls the subroutine to recalculate the total number of people
Call calcnoofpeople
Call calceventsubtotal

End Sub

Private Sub cmd_adult2_Click()

    'Adds one to the adult quantity
    i = Val(txt_adult2)
    i = i + 1
    txt_adult2 = i

End Sub

Private Sub txt_adult2_Change()

'calls the subroutine to recalculate the total number of people
Call calcnoofpeople
Call calceventsubtotal

End Sub

Sub calceventsubtotal()
```

```
subtotal = 0

'loops through the ticket types to find the ticket called adult
For e = 1 To 12

    With ticket(e)

        'calcualtes the subtotal and adds it to the label
        If .typeofticket = "adult1 " Then
            subtotal = subtotal + (Val(txt_adult1) * .ticketprice)
        Else
            If .typeofticket = "child1 " Then
                subtotal = subtotal + (Val(txt_child1) *
.ticketprice)
            Else
                If .typeofticket = "adult2 " Then
                    subtotal = subtotal + (Val(txt_adult2) *
.ticketprice)
                Else
                    If .typeofticket = "child2 " Then
                        subtotal = subtotal + (Val(txt_child2) *
.ticketprice)
                    End If
                End If
            End If
        End If

    End With
Next e


'sends the value of the subtotal to the label
lbl_subtotal = FormatCurrency(subtotal)

End Sub
```

Gift Aid Performance

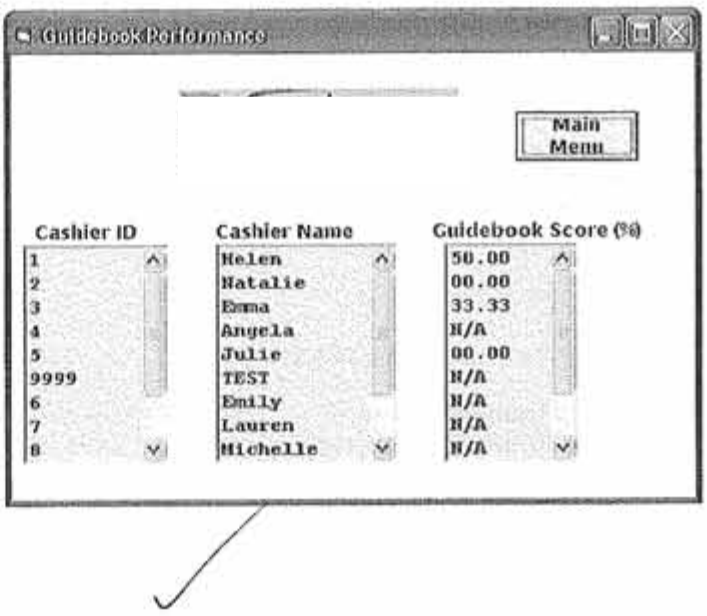



```
Private Sub cmd_home_Click()

'calls the next form
Load frm_mainmenu
frm_mainmenu.Show
Unload Me

End Sub

Private Sub Form_Load()

Call loadcashiers
Call loadtransactions

Dim ga As Integer
Dim tr As Integer


    'Loops through the cashiers to collect each name and user ID
    For i = 1 To cashiercount
        ga = 0
        tr = 0

        With cashier(i)
                lst_cashier.AddItem .cashiername
                lst_cashierID.AddItem .cashierID

                For j = 1 To transactioncount
                    With transaction(j)
                        If .cashierID = cashier(i).cashierID Then
                            tr = tr + 1
                            If .giftaid = True Then
                                ga = ga + 1
                            End If
                        End If
                    End With
                Next j


            If ga = 0 And tr = 0 Then
                performance = "N/A"
            Else
                'Calculates the cashiers performance
                performance = Format((ga / tr) * 100, "00.00")
            End If
            lst_score.AddItem performance
        End With
    Next i

End Sub
```

✓ deals with null data.

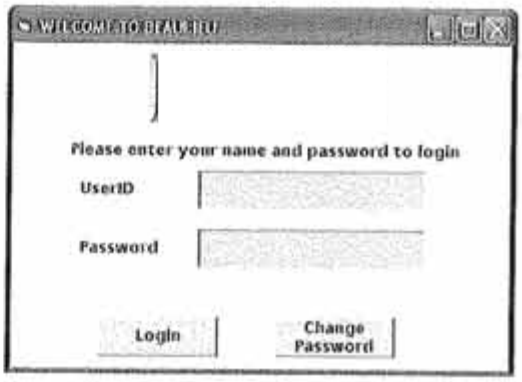
✓

Guidebook Performance




```
Private Sub cmd_home_Click()

'calls the next form
Load frm_mainmenu
frm_mainmenu.Show
Unload Me

End Sub

Private Sub Form_Load()

Call loadcashiers
Call loadtransactions

Dim gb As Integer
Dim tr As Integer

    'Loops through the cashiers to collect each name and user ID
    For i = 1 To cashiercount
        gb = 0
        tr = 0

        With cashier(i)
                lst_cashier.AddItem .cashiername
                lst_cashierID.AddItem .cashierID

                For j = 1 To transactioncount
                    With transaction(j)
                        If .cashierID = cashier(i).cashierID Then
                            tr = tr + 1
                            gb = gb + .guidebook
                        End If
                    End With
                Next j


            If ga = 0 And tr = 0 Then
                performance = "N/A"
            Else
                'Calculates the cashiers performance
                performance = Format((gb / tr) * 100, "00.00")
            End If
            lst_score.AddItem performance
        End With
    Next i

End Sub
```

Login

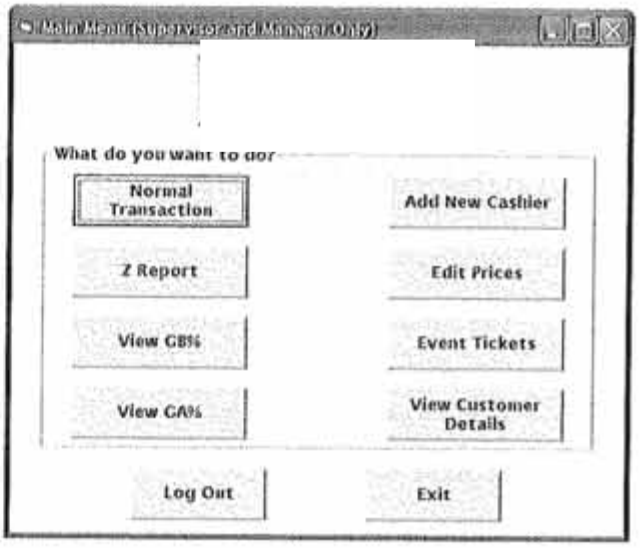



```
Private Sub cmd_change_Click()

'Loads the change password form
Load frm_changepassword
frm_changepassword.Show
Unload Me

End Sub

Private Sub cmd_login_Click()

'Calls the login subroutine
Call login

End Sub

Private Sub login()

Dim filenumber As Integer
Dim onecashier As CashierType
Dim i As Integer
Dim foundmatch As Boolean
foundmatch = False
'Closes any open files
Close #0

filenumber = FreeFile
'Opens the file to get the data from
Open App.Path & "\Cashiers.dat" For Random As filenumber Len =
Len(onecashier)
'Calculates the number of cashiers in the file
   cashiercount = LOF(filenumber) / Len(onecashier)
   'If the cashier count is greater then 0 move through the records
to find a match
    If cashiercount <> 0 Then
      For i = 1 To cashiercount
        Get #filenumber, i, cashier(i)
        With cashier(i)

            'Checks to see if a manger is loging in and then displays
the relevant form
            If .supervisor = True Then
                If .cashierID = Val(txt_userID.Text) And
.cashierpassword = txt_password.Text Then
                        foundmatch = True
                        cashiernum = i
                        Load frm_mainmenu
                        frm_mainmenu.Show
                        Unload Me
                End If
            Else
                    'Checks to make sure that the password and
username are correct
                    If .cashierID = Val(txt_userID.Text) And
.cashierpassword = txt_password.Text Then
                        foundmatch = True
                        cashiernum = i
                        Load frm_transaction
```

```
                        frm_transaction.Show
                        Unload Me
                    End If

            End If
        End With
    Next i

  Close #filenumber

    If foundmatch = False Then
            'Else gives a message box to say that the username or
password is invalid
            MsgBox ("Invalid username or password!")
            txt_userID = ""
            txt_password = ""
    End If
End If

End Sub


Private Sub txt_password_KeyPress(KeyAscii As Integer)

'Allows the cashier to push enter to then push login
If KeyAscii = 13 Then
    cmd_login.SetFocus
End If
End Sub

Private Sub txt_userID_KeyPress(KeyAscii As Integer)

'Allows the cashier to push enter to then enter the password
If KeyAscii = 13 Then
    txt_password.SetFocus
End If

End Sub
```

Main Menu

```
Private Sub
cmd_allcustomers_Click()

'calls the next form
Load frm_allcustomers
frm_allcustomers.Show
Unload Me

End Sub

Private Sub cmd_cashier_Click()

'calls the next form
Load frm_addnewcashier
frm_addnewcashier.Show
Unload Me

End Sub

Private Sub cmd_event_Click()

'calls the next form
Load frm_event
frm_event.Show
Unload Me

End Sub

Private Sub cmd_exit_Click()
End
End Sub

Private Sub cmd_GA_Click()

'calls the next form
Load frm_gaperformance
frm_gaperformance.Show
Unload Me

End Sub

Private Sub cmd_GB_Click()

'calls the next form
Load frm_gbperformance
frm_gbperformance.Show
Unload Me

End Sub

Private Sub cmd_logout_Click()

'calls the next form
Load frm_login
frm_login.Show
Unload Me

End Sub
```



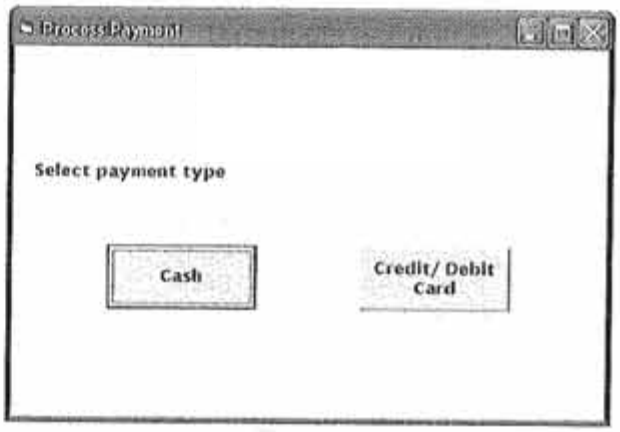
```
Private Sub cmd_normaltransaction_Click()

'calls the next form
Load frm_transaction
frm_transaction.Show
Unload Me

End Sub

Private Sub cmd_prices_Click()

'calls the next form
Load frm_editprices
frm_editprices.Show
Unload Me

End Sub

Private Sub cmd_zreport_Click()

Call loadtransactions

Dim a As Integer
Dim c As Integer
Dim m As Currency
Dim v As Currency
Dim g As Currency
Dim b As Integer
Dim t As Currency
Dim p As Integer
Dim ga As Single
Dim gb As Single

a = 0
c = 0
m = 0
v = 0
g = 0
b = 0
t = 0
p = 0
ga = 0
gb = 0


If transactioncount <> 0 Then

    'searches through the transaction data to find the information
for the days transactions
    For i = 1 To transactioncount

        With transaction(i)

            If .dateoftransaction = Date Then

                match = True

                a = a + .noofadults
                c = c + .noofchildren

                    If .typeofpayment = "C" Then
```

```
                m = m + .amountpaid
            Else
                v = v + .amountpaid
            End If

            If .giftaid = True Then
                g = g + .amountpaid
            End If

            b = b + .guidebook
            t = t + .amountpaid

        End If

    End With

Next i

If match = True Then
    p = a + c
    ga = (g / t) * 100
    gb = (b / p) * 100


    'Sets the page layout to portrait
    Printer.Orientation = 1

    'Sets the font type
    Printer.FontName = "Courier New"

    'Defines the text properties for the heading
    Printer.FontSize = 24
    Printer.FontUnderline = True
    Printer.FontBold = True
    Printer.Print "Z Report"
    Printer.FontUnderline = False
    Printer.Print "  "
    Printer.FontSize = 18
    Printer.Print "Date = "; Date


    'Defines the text values for the list
    Printer.FontUnderline = False
    Printer.FontSize = 14
    Printer.Print "  "
    Printer.Print "  "
    Printer.Print "  "
    Printer.Print "ADMISSION"
    Printer.Print "  "
    Printer.FontBold = False
    Printer.Print "Number of Adults ="; Spc(20); a
    Printer.Print "  "
    Printer.Print "Number of Children ="; Spc(18); c
    Printer.Print "  "
    Printer.Print "Total Admission ="; Spc(21); p
    Printer.Print "  "
    Printer.Print "  "
    Printer.FontBold = True

    Printer.Print "REVENUE"
```

```
        Printer.Print "   "
        Printer.FontBold = False
        Printer.Print "Cash Total ="; Spc(26); FormatCurrency(m)
        Printer.Print "   "
        Printer.Print "Visa Total ="; Spc(26); FormatCurrency(v)
        Printer.Print "   "
        Printer.Print "Total ="; Spc(31); FormatCurrency(t)
        Printer.Print "   "
        Printer.Print "   "
        Printer.FontBold = True

        Printer.Print "GIFT AID"
        Printer.Print "   "
        Printer.FontBold = False
        Printer.Print "Gift Aid Total ="; Spc(22); FormatCurrency(g)
        Printer.Print "   "
        Printer.Print "Gift Aid % ="; Spc(26); Format(ga, "00.00");
"%"
        Printer.Print "   "
        Printer.Print "   "
        Printer.FontBold = True

        Printer.Print "GUIDEBOOKS"
        Printer.Print "   "
        Printer.FontBold = False
        Printer.Print "Guidebook Total ="; Spc(20); b
        Printer.Print "   "
        Printer.Print "Guidebook % ="; Spc(25); Format(gb, "00.00");
"%"


        Printer.EndDoc

        'Gives a message box to say that the details have been
successfully printed
        MsgBox ("The Z Report Details Have Been Sent To The Printer")
    Else

        'Gives a message box to say that there is no transactions to
print
        MsgBox ("No Transactions to Print!")
   End If
Else
    'Gives a message box to say that there is no transactions to
print
    MsgBox ("No Transactions to Print!")
End If


End Sub
```

printing - deals with no data to print.

Process Payment




```
Private Sub cmd_card_Click()

If eventticket = False Then

    With
transaction(transactioncount)
        .typeofpayment = "V"
    End With

    Call
savenewtransaction(transactioncount)

    'accpets the payment
    MsgBox ("Payment Accepted!")

    'calls the next form
    Load frm_transaction
    frm_transaction.Show
    Unload Me

Else

    'accepts the payment
    MsgBox ("Payment Accepted!")

    'calls the next form
    Load frm_event
    frm_event.Show
    Unload Me

End If

End Sub

Private Sub cmd_cash_Click()

If eventticket = False Then
    With transaction(transactioncount)

        .typeofpayment = "C"
    End With

    Call savenewtransaction(transactioncount)
End If

Load frm_cashpayment
frm_cashpayment.Show
Unload Me
End Sub

Private Sub Form_Load()

'calls the sub routine from the module to disable the X so that the
user can not close the form
Call DisableX(Me, True)

End Sub
```

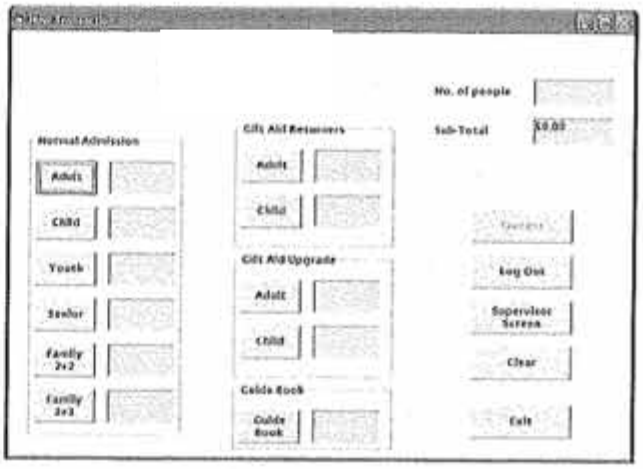
Returning Customer


```
Dim returnid As Integer

Private Sub cmd_cancel_Click()

'loads the next form
Load frm_transaction
frm_transaction.Show
Unload Me

End Sub

Private Sub cmd_process_Click()


Call savenewcustomer(returnid)


'loads the next form
Load frm_processpayment
frm_processpayment.Show
Unload Me

End Sub

Private Sub cmd_submit_Click()

'makes sure that a surname has been entered
and calls the subroutine to load the customers
If txt_surname = "" Then
    MsgBox ("Please enter a valid surname!")
Else
    Call loadreturncustomers
End If

End Sub

Sub loadreturncustomers()

Dim filenumber As Integer
Dim onecustomer As CustomerType
Dim i As Integer
Dim m As Boolean

'close any open files
Close #0

'sets the match for the boolean value to false
m = False

filenumber = FreeFile

'opens the file to get the data from
Open App.Path & "\Customers.dat" For Random As filenumber Len =
Len(onecustomer)

'calculates the number of customers in the file
customercount = LOF(filenumber) / Len(onecustomer)
```

```
'if the customer count is greater than 0 move through the records to
find a match
If customercount <> 0 Then
    For i = 1 To customercount

            With customer(i)
                l = Len(txt_surname)
                p = Len(txt_postcode)

                'searches for the customers details and displays them
                If UCase(txt_postcode.Text) = UCase(Left$(.postcode,
p)) And UCase(txt_surname.Text) = UCase(Left$(.surname, 1)) Then

                    'sets the boolean value to true
                    m = True

                    'displays the customer details in the correct
labels
                    lbl_house.Caption = .house
                    lbl_street.Caption = .street
                    txt_surname = .surname
                    txt_postcode = .postcode
                     returnid = i
                    .returning = True

                    'work out remaining days on the pass
                    daysleft = DateDiff("d", .dateofentry, Date)
                    lbl_remainingdays = Val(365 - daysleft)
                    lbl_expirydate = DateAdd("yyyy", 1, .dateofentry)

                    'loops through the transaction file to find the
transaction that is associated to the customer
                    For n = 1 To transactioncount

                        With transaction(n)

                            'the number of adults and children first
admitted are sent to the label
                            If .customerID = customer(i).customerID
Then

                                lbl_adults = .noofadults
                                lbl_child = .noofchildren
                            End If

                        End With
                    Next n
                End If
            End With
    Next i

    'if a match is found the amount to pay is displayed and the
process command button is enabled
    If m = True Then
        lbl_totaltopay = FormatCurrency(subtotal)
        cmd_process.Enabled = True
    End If



    'if m is false then no matches have been found so displays a
message box
```

```
    If m = False Then
        MsgBox ("No matching records found")
    End If

End If

Close #filenumber

End Sub

Private Sub Form_Load()

'calls the subroutines to load in the necessary files
Call loadtransactions
Call loadcustomers

End Sub
```

New Transaction




```
Private Sub cmd_adult_Click()

    'Adds one to the adult quantity
    i = Val(txt_adult)
    i = i + 1
    txt_adult = i

End Sub

Private Sub cmd_adultr_Click()

    'Adds one to the adult quantity
    i = Val(txt_adultr)
    i = i + 1
    txt_adultr = i

End Sub

Private Sub cmd_adultu_Click()

    'Adds one to the adult quantity
    i = Val(txt_adultu)
    i = i + 1
    txt_adultu = i

End Sub

Private Sub cmd_child_Click()

    'Adds one to the child quantity
    i = Val(txt_child)
    i = i + 1
    txt_child = i

End Sub

Private Sub cmd_childr_Click()

    'Adds one to the adult quantity
    i = Val(txt_childr)
    i = i + 1
    txt_childr = i

End Sub

Private Sub cmd_childu_Click()

    'Adds one to the adult quantity
    i = Val(txt_childu)
    i = i + 1
    txt_childu = i

End Sub

Private Sub cmd_clear_Click()

'clears the text boxes on the form
txt_adult = ""
```

```
txt_child = ""
txt_youth = ""
txt_senior = ""
txt_family2 = ""
txt_family3 = ""
txt_childr = ""
txt_adultr = ""
txt_adultu = ""
txt_childu = ""
txt_gb = ""

'enables all of the frames and command buttons
fra_return.Enabled = True
fra_upgrade.Enabled = True
fra_normal.Enabled = True
cmd_adult.Enabled = True
cmd_child.Enabled = True
cmd_youth.Enabled = True
cmd_senior.Enabled = True
cmd_fam2.Enabled = True
cmd_fam3.Enabled = True
cmd_adultr.Enabled = True
cmd_childr.Enabled = True
cmd_adultu.Enabled = True
cmd_childu.Enabled = True

End Sub

Private Sub cmd_exit_Click()

    'ends the program
    End

End Sub

Private Sub cmd_fam2_Click()

    'Adds one to the family quantity
    i = Val(txt_family2)
    i = i + 1
    txt_family2 = i

End Sub

Private Sub cmd_fam3_Click()

    'Adds one to the family quantity
    i = Val(txt_family3)
    i = i + 1
    txt_family3 = i

End Sub

Private Sub cmd_GB_Click()

    'Adds one to the guidebook quantity
    i = Val(txt_gb)
    i = i + 1
    txt_gb = i

End Sub
```

```
Private Sub cmd_logout_Click()

    'calls the next form
    Load frm_login
    frm_login.Show
    Unload Me

End Sub

Private Sub cmd_process_Click()
'checks to see which type of ticket is being purchased and then loads
the appropriate form
If fra_normal.Enabled = True Then

        transactioncount = transactioncount + 1

        'gathers the information to save the transaction information
        With transaction(transactioncount)
            .noofadults = Val(txt_adult) + Val(txt_senior) +
(Val(txt_family2) * 2) + (Val(txt_family3) * 2) + Val(txt_adultr) +
Val(txt_adultu)
            .noofchildren = Val(txt_child) + Val(txt_youth) +
(Val(txt_family2) * 2) + (Val(txt_family3) * 3) + Val(txt_childr) +
Val(txt_childu)
            .dateoftransaction = Date
            .transactionID = transactioncount
            .cashierID = cashier(cashiernum).cashierID
            .guidebook = Val(txt_gb)
            .amountpaid = subtotal
        End With


        eventticket = False

        Style = vbYesNo + vbQuestion
        message = "Does the customer wish to Gift Aid?"
        Title = "Gift Aid?"
        response = MsgBox(message, Style, Title)

        If response = vbYes Then

            'calls the next form
            Load frm_customerdetails
            frm_customerdetails.Show
            Unload Me

        Else

            'calls the next form
            Load frm_processpayment
            frm_processpayment.Show
            Unload Me

        End If
Else

        'checks to see which type of ticket is being purchased and
then loads the appropriate form
        If fra_return.Enabled = True Then
```

```
            'calls the next form
            Load frm_returningcustomer
            frm_returningcustomer.Show
            Unload Me
        Else

                'checks to see which type of ticket is being
purchased and then loads the appropriate form
                If fra_upgrade.Enabled = True Then

                    'calls the next form
                    Load frm_returningcustomer
                    frm_returningcustomer.Show
                    Unload Me

                End If
        End If
End If


End Sub

Private Sub cmd_senior_Click()

    'Adds one to the senior quantity
    i = Val(txt_senior)
    i = i + 1
    txt_senior = i

End Sub

Private Sub cmd_supervisor_Click()

    'calls the next form
    Load frm_mainmenu
    frm_mainmenu.Show
    Unload Me

End Sub

Private Sub cmd_youth_Click()

    'Adds one to the youth quantity
    i = Val(txt_youth)
    i = i + 1
    txt_youth = i

End Sub

Private Sub Form_Load()

'sets the subtotal to zero ready for the new transaction
subtotal = 0
lbl_subtotal = FormatCurrency(subtotal)

'shows the command button to load the supervisor form if the user is
a supervisor or manager, else the form is hidden
For i = 1 To cashiercount
    With cashier(i)
        If cashiernum = .cashierID Then
            If .supervisor = True Then
```

```
                cmd_supervisor.Visible = True
            Else
                cmd_supervisor.Visible = False
            End If
        End If
    End With
Next i


Call loadtickets
Call loadtransactions


End Sub

Private Sub fra_normal_Click()

fra_return.Enabled = False
fra_upgrade.Enabled = False

End Sub

Private Sub txt_adult_Change()

'calls the subroutine to recalculate the total number of people
Call calcnoofpeople
Call calcsubtotal
Call disableupgrade
Call disablereturn

End Sub

Public Sub calcnoofpeople()

'adds the values in all of the text boxes to calculate the total
number of people
lbl_noofpeople = Val(txt_adult) + Val(txt_senior) + Val(txt_child) +
Val(txt_youth) + (Val(txt_family2) * 4) + (Val(txt_family3) * 5) +
Val(txt_adultr) + Val(txt_childr) + Val(txt_adultu) + Val(txt_childu)

End Sub

Private Sub txt_adultr_Change()

'calls the subroutine to recalculate the total number of people
Call calcnoofpeople
Call disablenormal
Call disableupgrade

End Sub

Private Sub txt_adultu_Change()

'calls the subroutine to recalculate the total number of people
Call calcnoofpeople
Call calcsubtotal
Call disablenormal
Call disablereturn

End Sub
```

```
Private Sub txt_child_Change()

'calls the subroutine to recalculate the total number of people
Call calcnoofpeople
Call calcsubtotal
Call disableupgrade
Call disablereturn

End Sub

Private Sub txt_childr_Change()

'calls the subroutine to recalculate the total number of people
Call calcnoofpeople
Call disablenormal
Call disableupgrade

End Sub

Private Sub txt_childu_Change()

'calls the subroutine to recalculate the total number of people
Call calcnoofpeople
Call calcsubtotal
Call disablenormal
Call disablereturn

End Sub

Private Sub txt_family2_Change()

'calls the subroutine to recalculate the total number of people
Call calcnoofpeople
Call calcsubtotal
Call disableupgrade
Call disablereturn

End Sub

Private Sub txt_family3_Change()

'calls the subroutine to recalculate the total number of people
Call calcnoofpeople
Call calcsubtotal
Call disableupgrade
Call disablereturn

End Sub

Private Sub txt_gb_Change()

'calls the sub routine to calculate the subtotal
Call calcsubtotal

End Sub

Private Sub txt_senior_Change()

'calls the subroutine to recalculate the total number of people
Call calcnoofpeople
Call calcsubtotal
```

```
Call disableupgrade
Call disablereturn

End Sub

Private Sub txt_youth_Change()

'calls the subroutine to recalculate the total number of people
Call calcnoofpeople
Call calcsubtotal
Call disableupgrade
Call disablereturn

End Sub

Sub calcsubtotal()

subtotal = 0

'loops through the ticket types to find the ticket called adult
For q = 1 To 12

    With ticket(q)

        'calcualtes the subtotal and adds it to the label
        If .typeofticket = "adult   " Then
            subtotal = subtotal + (Val(txt_adult) * .ticketprice)
        Else
            If .typeofticket = "child   " Then
                subtotal = subtotal + (Val(txt_child) * .ticketprice)
            Else
                If .typeofticket = "youth   " Then
                    subtotal = subtotal + (Val(txt_youth) *
.ticketprice)
                Else
                    If .typeofticket = "senior " Then
                        subtotal = subtotal + (Val(txt_senior) *
.ticketprice)
                    Else
                        If .typeofticket = "family2" Then
                            subtotal = subtotal + (Val(txt_family2) *
.ticketprice)
                        Else
                            If .typeofticket = "family3" Then
                                subtotal = subtotal +
(Val(txt_family3) * .ticketprice)
                            Else
                                If .typeofticket = "adultu " Then
                                    subtotal = subtotal +
(Val(txt_adultu) * .ticketprice)
                                Else
                                    If .typeofticket = "childu " Then
                                        subtotal = subtotal +
(Val(txt_childu) * .ticketprice)
                                    End If
                                End If
                            End If
                        End If
                    End If
                End If
            End If
```

```
        End If

    End With
Next q

'adds the value of the guidebooks to the subtotal
subtotal = subtotal + (Val(txt_gb) * 3)

'sends the value of the subtotal to the label
lbl_subtotal = FormatCurrency(subtotal)

End Sub

Sub disableupgrade()

'disables the frame and command buttons if the user selects a
different type of ticket
fra_upgrade.Enabled = False
cmd_adultu.Enabled = False
cmd_childu.Enabled = False

End Sub

Sub disablereturn()

'disables the frame and command buttons if the user selects a
different type of ticket
fra_return.Enabled = False
cmd_adultr.Enabled = False
cmd_childr.Enabled = False

End Sub

Sub disablenormal()

'disables the frame and command buttons if the user selects a
different type of ticket
fra_normal.Enabled = False
cmd_adult.Enabled = False
cmd_child.Enabled = False
cmd_youth.Enabled = False
cmd_senior.Enabled = False
cmd_fam2.Enabled = False
cmd_fam3.Enabled = False

End Sub
```

Main Sub Routines Used

The tables below show the main sub routines used throughout the program.

Subroutine Name	Location	Purpose	How Used?
loadcashiers	Module- Public Sub	To load the cashiers records from the file cashiers.dat	Called at various points throughout the program to load the cashiers details into the array
savenewcashier	Module- Public Sub	To save the cashiers records to the file cashiers.dat	Called at various points throughout the program to save the cashiers details
loadcustomers	Module- Public Sub	To load the customers records from the file customers.dat	Called at various points throughout the program to load the customers details into the array
savenewcustomer	Module- Public Sub	To save the customers records to the file customers.dat	Called at various points throughout the program to save the customers details
loadtransactions	Module- Public Sub	To load the transactions records from the file transactions.dat	Called at various points throughout the program to load the transaction details into the array
savenewtransaction	Module- Public Sub	To save the transaction details to the file transaction.dat	Called at various points throughout the program to save the transaction details
loadtickets	Module- Public Sub	To load the ticket records from the file tickets.dat	Called at various points throughout the program to load the ticket details into the array
savenewticket	Module- Public Sub	To save the ticket records to the file ticekts.dat	Called at various points throughout the program to save the ticket details

DisableX	Module- Public Sub	This disables the X on the form to prevent the user from closing the program	Called from the form load event of the forms used during a transaction, for example frm_processpayment
cmd_submit_click	Add New Cashier- Private Sub	Carries out several validation checks on the input data, then calls the subroutine savenewcashier	Executed when the user click on the submit command button
displaycustomers	All Customers- Public Sub	Takes the necessary data from the array and sets how and where to display it	Called when the user selects a month to view the data about
cmd_ok_click	Cash Payment- Private Sub	Carries out validation checks on the amount of tender entered, then calculates the amount of change	Executed when the user click on the ok command button
cmd_total_click	Cash Payment- Private Sub	Carries out validation checks and displays the amount tender	Executed when the user click on the total command button
calculateamounttender	Cash Payment- Public Sub	Calculates the amount given based on the command buttons that have been selected	Called when the user either types a value into a text box or click on one of the command buttons
cmd_submit_click	Change Password- Private Sub	Carries out validation checks on the data that has been input. Calls the subroutine, savenewcashier	Called when the user selects submit to save the new password
cmd_save_click	Customer Details- Private Sub	Carries out several validation checks on the data input. Calls the subroutine, savenewcustomer. Also adds one to the	Called when the user selects the Save command button
Form_Load	Edit Prices- Private Sub	Calls the subroutine, loadtickets, displays the ticket prices in the text boxes	Executed when the user loads the form
cmd_save_click	Edit Prices- Private Sub	Calls the subroutine, savetickets	Executed when the user selects the save command button

cmd_process_click	Event Tickets- Private Sub	Calls the next form, frm_processpayment	Executed when the user selects the process command button
calceventsubtotal	Event Tickets- Public Sub	Calculates the total to pay based upon the quantity the user has entered	Executed from each text box and command button when the user changes the quantity
calcnoofpeople	Event Tickets- Public Sub	Calculates the number of tickets based upon the quantity the user has entered	Executed from each text box and command button when the user changes the quantity
Form_Load	Gift Aid Performance- Private Sub	Calls the subroutine, loadcashiers and loadtransactions. Sends the records to the list boxes	Executed when the user loads the form
Form_Load	Guidebook Performance- Private Sub	Calls the subroutine, loadcashiers and loadtransactions. Sends the records to the list boxes	Executed when the user loads the form
login	Login- Private Sub	Checks through the records of cashiers to ensure that the login details are valid	Called when the user selects the login command button
cmd_zreport_click	Main Menu- Private Sub	Calculates the information for the days transactions and sets the layout to use when printing them	Called when the user selects the Z report command button
cmd_card_click	Process Payment- Private Sub	Calls the subroutine to save the transaction details	Called when the user selects the card command button
cmd_cash_click	Process Payment- Private Sub	Calls the subroutine to save the transaction details and loads the next form	Called when the user selects the cash command button
cmd_submit_click	Returning Customer- Private Sub	Calls the subroutine, loadreturncustomers	Called when the user selects the submit command button

loadreturncustomers	Returning Customer- Public Sub	Searches through the customer details to find a match with the details entered	Called when the user selects the submit command button
cmd_process_click	New Transaction- Private Sub	Saves the transaction details to the transaction array	Called when the user selects the process command button
calcnoofpeople	New Transaction- Public Sub	Calculates the number of tickets based upon the quantity the user has entered	Executed from each text box and command button when the user changes the quantity
calcsubtotal	New Transaction- Public Sub	Calculates the total to pay based upon the quantity the user has entered	Executed from each text box and command button when the user changes the quantity

Some of these subroutines are used at several different times during the program this makes the program more efficient.

Main Variables Used with Validation

The table below shows a data dictionary for the program. It gives details of the main data stored about each entity aswell as the validation that is carried out on each variable.

Variable Name	Data Type	Purpose	Validation/ Explanation
CUSTOMER			
customerID	Integer	2 bytes	This is unique to each customer and is generated automatically. This is also the primary key for the customer entity.
firstname	String	Max 15 characters	The customer's first name, field can't be left blank.
surname	String	Max 15 characters	The customer's surname, field can't be left blank.
house	String	Max 15 characters	The name or number of the customer's property. This field is required.
postcode	String	Max 8 characters	The customer's postcode can be no longer than 8 characters including a space. This field is also required.
street	String	Max 20 characters	The street that the customers address is in, field can't be left blank.
town	String	Max 15 characters	The town that the customers address is in, field can't be left blank.
county	String	Max 15 characters	The county that the customers address is in, field can't be left blank.
dateofentry	Date	8 bytes	Automatically saved to the customer file based upon the current date. Format dd/mm/yyyy.

emailaddress	String	25 characters	The customers email address taken from the text box on the details form.
mailings	Boolean	2 bytes	The customer will either be on the mailing list, true, or not on the mailing list displayed as false.
TRANSACTION			
transactionID	Integer	2 bytes	This is unique to each transaction and is generated automatically. This is also the primary key for the customer entity.
customerID	Integer	2 bytes	This is automatically taken from the customer file.
cashierID	Integer	2 bytes	This is automatically taken from the cashier file.
amountpaid	Currency	8 bytes	The price of the transaction. With a validation check of between £14.50 and £300.
giftaid	Boolean	2 bytes	This will save whether or not the customer has allowed the transaction to be signed up to Gift Aid or not.
noofadults	Integer	2 bytes	Must be between 0 and 100.
noofchildren	Integer	2 bytes	Must be between 0 and 100.
typeofpayment	String	1 character	Will either be C or V for cash or Visa, this can not be left blank.
dateoftransaction	Date	8 bytes	Automatically saved to the customer file based upon the current date. Format dd/mm/yyyy.
guidebook	Integer	2 bytes	Must be between 0 and 10.

CASHIER			
cashierID	Integer	2 bytes	This is unique to each cashier and is generated automatically. This is also the primary key for the customer entity.
cashiername	String	15 characters	The cashier name taken from the text box when they log in.
cashierpassword	String	5 characters	The cashiers password again taken from the text box when they log in.
supervisor	Boolean	2 bytes	This is set when the new cashier is added. It will vary depending on whether the operator is a cashier or a supervisor.
TICKET			
typeofticket	String	7 characters	This can not be edited within the program.
ticketprice	Currency	8 bytes	This must be numeric and between £0.00 and £25.00.

Built in VB Functions Used

The VB functions used during the program can be seen below.

Function	Example	Purpose
FormatCurrency	`lbl_subtotal = FormatCurrency(subtotal)`	Displays the variable subtotal as a currency in £s and pence.
Val	`i = Val(txt_adultr)`	Converts the text entered into a numeric value.
Len	`l = Len(txt_surname)`	Takes the length of the surname and does not count any spaces left on the end.
UCase	`UCase(txt_postcode.Text)`	Converts the value in the text box into capital letters.
Left$	`(Left$(.postcode, p))`	Uses the left string function to take the left hand part of the variable, in this case the postcode.

Security

The security of the system involves each cashier having a unique ID and password. To be able to access the system the cashier must log in. Normal cashiers do not have the option to search through the customers details. Only the supervisors and managers are able to do this. This will help to keep the customers details more secure.

Physical security of the computer on which the data is stored on will also need to be taken into account. For example it should be kept in a locked room and be password protected.

TASK D

TESTING

As specified in the design section I am going to use Black Box testing to see if the program produces the correct outputs from specific inputs. The table below shows the tests that I have carried out with the outcome they gave. The following table shows the evidence for all of these tests.

Login Form

Test Number	Reason	Test Data	Expected Result	Actual Result
1	To make sure that the text boxes can't be left blank.	" "	Error MsgBox	Expected
2	To check that user ID only contains integers.	emm1	Error MsgBox	Expected
3	To check that it doesn't accept invalid user ID's.	1123	Error MsgBox	Expected
4	To check that it doesn't accept invalid passwords.	With user ID=0001 Password =00002	Error MsgBox	Expected
5	To check that it accepts valid data for user ID and passwords.	With user ID=0001 Password =00001	Transaction Form should load	Expected
6	To check that the change password command button loads the change password form.	Click	Change Password Form should be loaded.	Expected

Change Password

Test Number	Reason	Test Data	Expected Result	Actual Result
7	To ensure that only 5 characters can be put in the current password text box.	123456	Max length of text box reached at 5.	Expected
8	To ensure that only 5 characters can be put in the password 1 text box.	123456	Max length of text box reached at 5.	Expected
9	To ensure that only 5 characters can be put in the current password 2 text box.	123456	Max length of text box reached at 5.	Expected
10	To ensure that only 4 characters can be put in the	12345	Max length of text box	Expected

	user ID text box.		reached at 4.	
11	To ensure that user ID is numeric.	123e	Error MsgBox.	Expected
12	To check that the user ID can't be left blank.	" "	Error MsgBox.	Expected
13	To ensure that password 1 can't be left blank.	" "	Error MsgBox.	The details were accepted.
14	To ensure that password 2 can't be left blank.	" "	Error MsgBox.	Expected
15	To ensure that the passwords in the two boxes must be the same.	Password 1 = 12345 Password 2 = 54321	Error MsgBox.	Expected
16	To ensure that valid data is accepted.	User ID = 0002 Current Password = 00002 New Password 1= 22222 New Password 2= 22222	MsgBox to say that the data has been accepted.	Expected
17	To ensure that an invalid password is not accepted.	With User ID= 0003 Password = 00002	Error MsgBox.	Expected

Customer Details

Test Number	Reason	Test Data	Expected Result	Actual Result
18	Check that the first name can't be numeric.	emma1	Error MsgBox.	Expected
19	Check that surname can't be numeric.	varty1	Error MsgBox.	Expected
20	Check that the first name can't be left blank.	" "	Error MsgBox.	Expected
21	Check that the surname can't be left blank.	" "	Error MsgBox.	Expected
22	Check that the house name/ number can't be left blank.	" "	Error MsgBox.	Expected
23	Check that the postcode can't be	" "	Error MsgBox.	Expected

	left blank.			
24	Check that the customer details are saved.	Emma Varty 28 SO42 7ZN	MsgBox to say that details have been saved and a new form is loaded.	Expected
25	Check that the Boolean value of Gift Aid Y/N is set correctly.	Yes	Gift Aiding customers will be displayed in one list box on the mangers form whilst donating customers will be displayed on the other.	Expected
26	Check that when skip is clicked the payment form loads.	Click	Payment form is loaded.	Expected
27	Check that street can't be numeric.	Rock2ry	Error MsgBox.	Expected
28	Check that town can't be numeric.	Dibd3n	Error MsgBox.	Expected
29	Check that county can't be numeric.	H4mpshir3	Error MsgBox.	Expected

Main Menu

Test Number	Reason	Test Data	Expected Result	Actual Result
30	Check that the Normal Transaction command button loads the correct form.	Click	Transaction Form Loaded	Expected
31	Check that the Z Report command button prints the Z report.	Click	Z Report Printed	Expected
32	Check that the View GB% command button loads the correct form.	Click	Guidebook Form Loaded	Expected
33	Check that the GA% command button loads the correct form.	Click	Gift Aid Form Loaded	Expected
34	Check that the Add New Cashier command button loads the correct form.	Click	Add New Cashier Form Loaded	Expected
35	Check that the Edit Prices command button loads the correct form.	Click	Edit Prices Form Loaded	Expected
36	Check that the Event Ticket command button loads the correct form.	Click	Event Ticket Form Loaded	Expected
37	Check that the View Customer Details command button loads the correct form.	Click	Customer Details Form Loaded	Expected
38	Check that the Log Out command button loads the correct form.	Click	Log In Form Loaded	Expected
39	Check that the Exit command button ends the program.	Click	Program Ended	Expected

New Transaction

Test Number	Reason	Test Data	Expected Result	Actual Result
40	Check that when one ticket type is selected the other types are disabled.	Click	The frame not used is disabled.	Expected
41	Check that by clicking on a command button the ticket text box is updated.	Click	The number of tickets is updated in the text box.	Expected
42	Check that as the ticket numbers are changed the no. of people label is updated.	Click	The number of people is correctly calculated and displayed.	Expected
43	Check that as the ticket numbers are changed the sub total is updated.	4	The subtotal is correctly calculated.	Expected
44	Check that the Process command button loads the correct message box.	Click	The message box is given correctly.	Expected
45	Check that if the user says Yes to Gift Aid the correct form is loaded.	Click	The customer details form is loaded.	Expected
46	Check that if the user says No to Gift Aid the correct form is loaded.	Click	The payment form is loaded.	Expected
47	Check that if the user is not a supervisor the supervisor screen command button is not shown.	Form Load with user 0005	The button is not shown.	Expected
48	Check that if the user is a supervisor the supervisor screen command button is shown.	Form Load with user 0001	The button is shown.	Expected
49	Check that the Log Out command button loads the correct form.	Click	The correct form is loaded.	Expected
50	Check that the Clear command button clears the form.	Click	The form is cleared.	Expected
51	Check that the Exit command button ends the program.	Click	The program ends.	Expected

Returning Customer

Test Number	Reason	Test Data	Expected Result	Actual Result
52	Check that surname can't be numeric.	V4rty	Error MsgBox.	The details are accepted as far as, the records are still searched, although as this

			✓	validation has been carried out on first entry no matching records were found.
53	Check that invalid searches give the correct response.	Varty SO42 7ZN	MsgBox saying that the data could not be found.	Expected
54	Check that part of valid data is searched for and displayed correctly.	Ne So	Record of customer Varty is found and should be displayed correctly.	Expected
55	Check that the cancel button loads the form and takes one away from the transaction count.	Click	Form will load and the transaction count should be changed.	Expected
56	Check that the days remaining on the pass are calculated and displayed correctly.	Test with a customer who entered yesterday, 364 days.	364 displayed in the correct text box.	Expected
57	Check that the date of expiry is calculated and displayed correctly.	Test with a customer who entered yesterday.	A year should be added to the first date of entry.	Expected
58	Check that the Process command loads the next form and saves the returning customer details.	Click	Payment form should be loaded.	Expected
59	Check that the total to pay is displayed correctly.	£8.80	The correct amount is displayed in the label.	Expected
60	Check that the number of adults is displayed correctly.	1	The correct amount is displayed in the label.	Expected
61	Check that the number of children is displayed correctly.	0	The correct amount is displayed in the label.	Expected

Process Payment

Test Number	Reason	Test Data	Expected Result	Actual Result
62	Check that the cash command button loads the cash payment form.	Click	The correct form will be loaded.	Expected
63	Check that the credit/ debit card button loads a message box.	Click	Message box displayed saying the payment has been accepted.	Expected
64	Check that the user can not push the X in the top right hand corner.	Click	The program will not end.	X is not disabled.

Cash Payment

Test Number	Reason	Test Data	Expected Result	Actual Result
65	Check that the amount due is correctly displayed.	£35.00	£35.00 displayed in the label.	Expected
66	Check that the tender amount is added up correctly.	£40.00	£40.00 displayed in the text box.	Expected
67	Ensure that invalid values can not be typed into the amount tender.	45.th	A message box should be displayed asking for a suitable tender to be input.	The program bugged.
68	Check that the change due is calculated and displayed correctly.	£5.00	£5.00 displayed in the label.	Expected
69	Check that the Process command button is disabled on form load.	Form Load	Process button disabled.	Expected
70	Check that when the Ok buttons has been selected the Process command button is then enabled.	Click	Process button enabled.	Expected
71	Check that when the Clear button is selected the form is cleared.	Click	The amount tender and change due labels are cleared.	Expected

Guidebook Performance

Test Number	Reason	Test Data	Expected Result	Actual Result
72	Check that all cashiers are displayed in the list	Cashier names and User ID's	All cashiers should be displayed in the	Expected

	box.		list boxes.	
73	Check that when one list box is scrolled all the other list boxes scroll to keep the data in line.	Scroll CashierID	All list boxes will scroll inline with the box being used.	Expected
74	Check that the cashier performance is calculated correctly.	Use a new cashier who has sold one guidebook and two adults.	Guidebook score should be 50%.	Expected
75	Check that if the cashier hasn't seen any customers their score is N/A.	0 transactions	Cashier performance should be N/A.	Expected
76	Check that Main Menu command button loads the next form.	Click	The supervisor form is loaded.	Expected

Gift Aid Performance

Test Number	Reason	Test Data	Expected Result	Actual Result
77	Check that all cashiers are displayed in the list box.	Cashier names and User ID's	All cashiers should be displayed in the list boxes.	Expected
78	Check that when one list box is scrolled all the other list boxes scroll to keep the data in line.	Scroll CashierID	All list boxes will scroll inline with the box being used.	Expected
79	Check that the cashier performance is calculated correctly.	Use a new cashier who has made four transactions and Gift Aided three of them.	Gift Aid score should be 75%.	Expected
80	Check that if the cashier hasn't seen any customers their score is N/A.	0 transactions	Cashier performance should be N/A.	Expected
81	Check that Main Menu command button loads the next form.	Click	The supervisor form is loaded.	Expected

Add New Cashier

Test Number	Reason	Test Data	Expected Result	Actual Result
82	Check that cashier name can't be left blank.	" "	Error MsgBox.	Expected
83	Check that the cashier name can't be numeric.	564kjhk	Error MsgBox.	Expected

84	Check that the User ID must be four integers.	34	Error MsgBox. ✓	Details were saved.
85	Check that the User ID must be four integers.	564878	Maximum length of the text box has been reached.	Expected
86	Check that the User ID can't be alphanumeric.	12em	Error MsgBox. ✓	Program bugged.
87	Check that the password 1 can't be left blank.	" "	Error MsgBox.	Expected
88	Check that the password 2 can't be left blank.	" "	Error MsgBox.	Expected
89	To ensure that the passwords in the two boxes must be the same.	Password 1 = 12345 Password 2 = 54321	Error MsgBox.	Expected
90	To check that the passwords must be 5 characters long.	Password= 123	Error MsgBox.	Expected
91	To check that the passwords must be 5 characters long.	Password= 123456	Maximum length of the text box is reached.	Expected
92	To check that valid data is accepted and saved.	CashierName= Alison UserID = 0066 Password1 = 00066 Password2 = 00066	Data should be saved and MsgBox given to say that the details have been accepted.	Expected

Edit Prices

Test Number	Reason	Test Data	Expected Result	Actual Result
93	Check that only numeric values can be entered into the text boxes.	adu12	A message box will be displayed asking for a valid price to be entered.	Program bugs.
94	Check that the Save button allows valid data to be saved.	16, Click	A message box will be displayed stating that the data has been correctly displayed.	Data saved but no message box given.
95	Check that the Main Menu command button loads the correct form.	Click	The main menu form is loaded.	Expected
96	Check that the Log Out command button loads the correct form.	Click	The log in form is loaded.	Expected
97	Check that the correct prices are displayed in the text boxes when the form loads.	Form Load	The correct prices are displayed.	Expected

Event Tickets

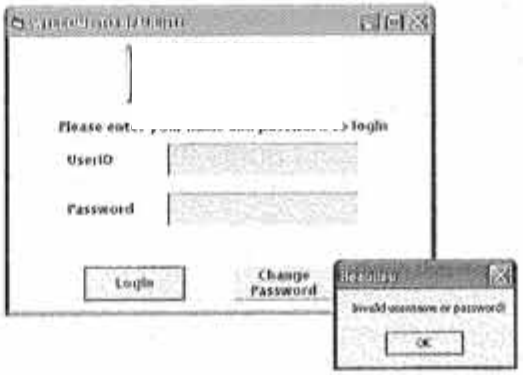
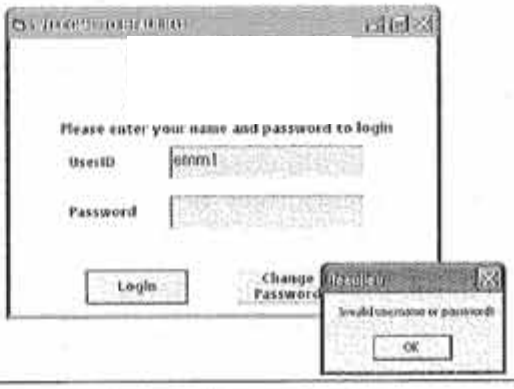
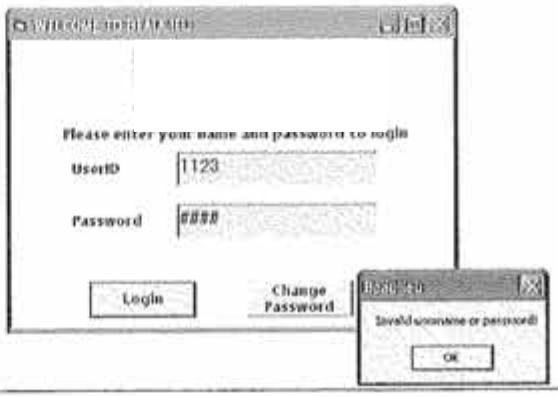
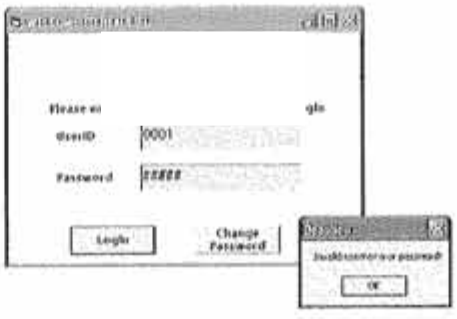
Test Number	Reason	Test Data	Expected Result	Actual Result
98	Check that by clicking on a command button the ticket text box is updated.	Click	The number of tickets is updated in the text box.	Expected
99	Check that as the ticket numbers are changed the no. of people label is updated.	2	The number of people is correctly calculated and displayed.	Expected
100	Check that as the ticket numbers are changed the sub total is updated.	£17.00	The subtotal is correctly calculated.	Expected
101	Check that the Process command button loads the correct sheet.	Click	The correct form is loaded.	Expected
102	Check that the Log Out command button loads the correct form.	Click	The correct form is loaded.	Expected

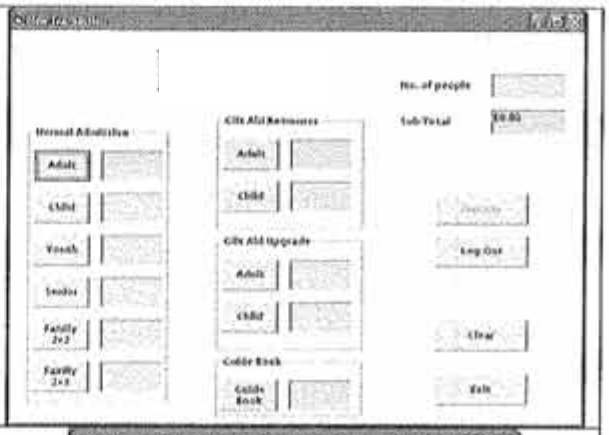
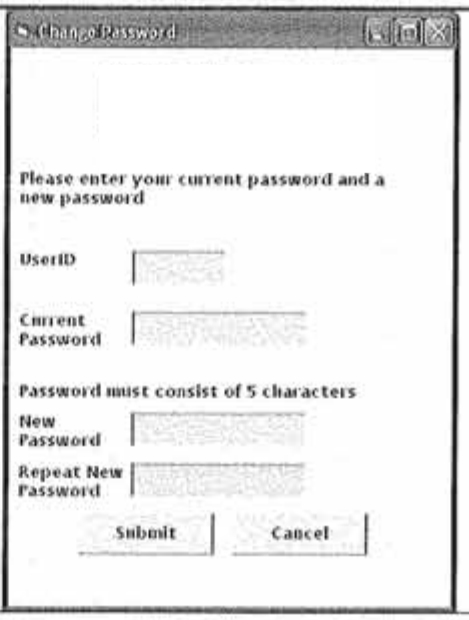
All Customers Details

Test Number	Reason	Test Data	Expected Result	Actual Result
103	Check that the gift aiding customers are displayed correctly and filtered by the month.	Gift aiding customers in March.	Correct customers displayed in the list box.	Expected
104	Check that the gift aiding print command prints the correct customers also filtering by month.	Gift aiding customers in March.	Prints the Gift Aiding customers in March.	Expected
105	Check that the returning customers are displayed correctly and filtered by the month.	Gift aiding customers in March.	Correct customers displayed in the list box.	Expected
106	Check that the returning print command prints the correct customers also filtering by month.	Gift aiding customers in March.	Prints the returning customers in March.	Expected
107	Check that the customers with email address' are displayed in the list box.	Customers with email address'.	Only displays those customers with email addresses in the list box.	Expected
108	Check that the Main Menu button loads the correct form.	Click	The main menu should be loaded.	Expected

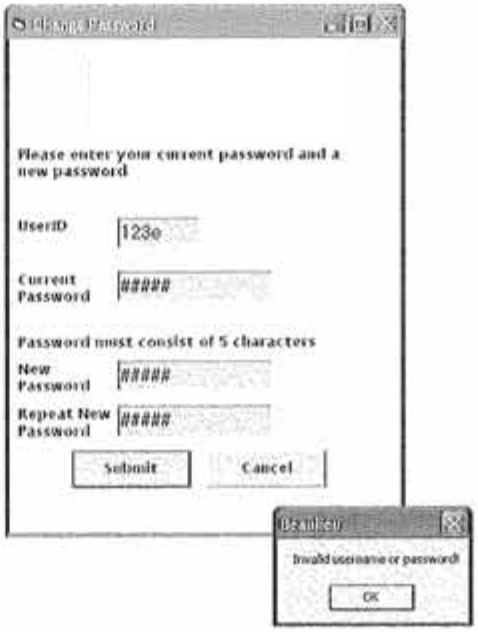
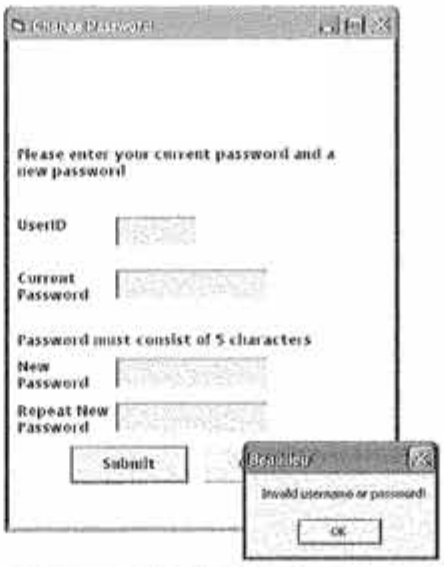
Testing Evidence

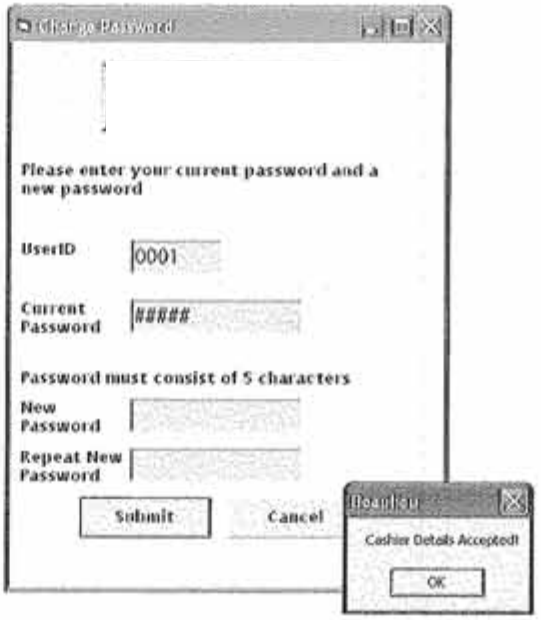
Login Form

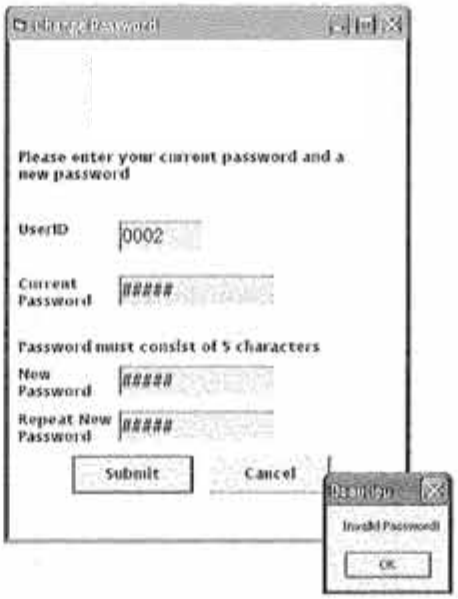
Test Number	Evidence Shows	Test Evidence
1	Text boxes can not be left blank.	
2	The user ID can only contain integers.	
3	Invalid user ID's are not accepted.	
4	Invalid passwords are not accepted.	

5	Correct user ID and password are accepted and loads the correct form.	
6	The change password form is correctly loaded.	

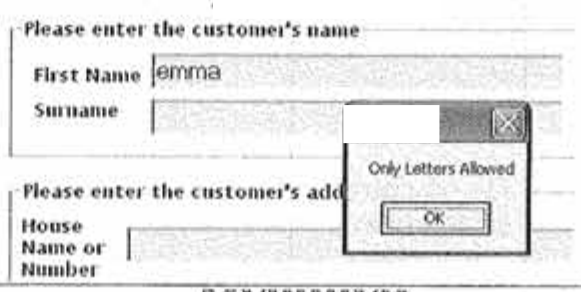
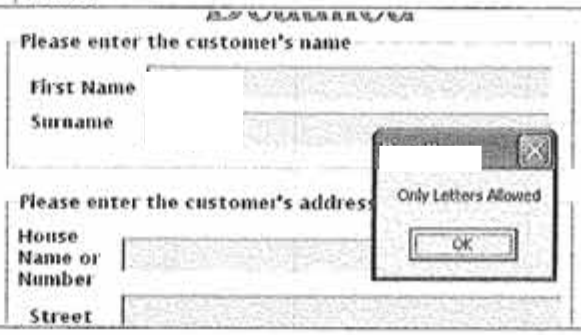
Change Password

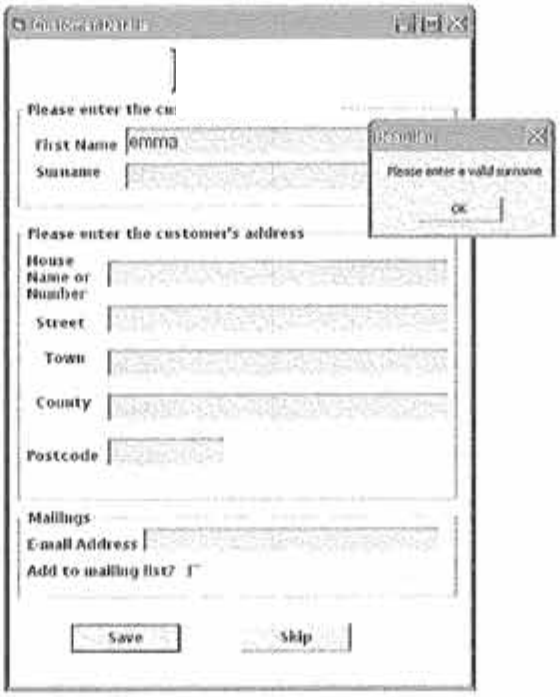
Test Number	Evidence Shows	Test Evidence
7	Maximum length reached.	Current Password #####|
8	Maximum length reached.	New Password #####|
9	Maximum length reached.	Repeat New Password #####|
10	Maximum length reached.	UserID 0001|
11	An alphanumeric user ID is not accepted.	
12	User ID can not be left blank.	

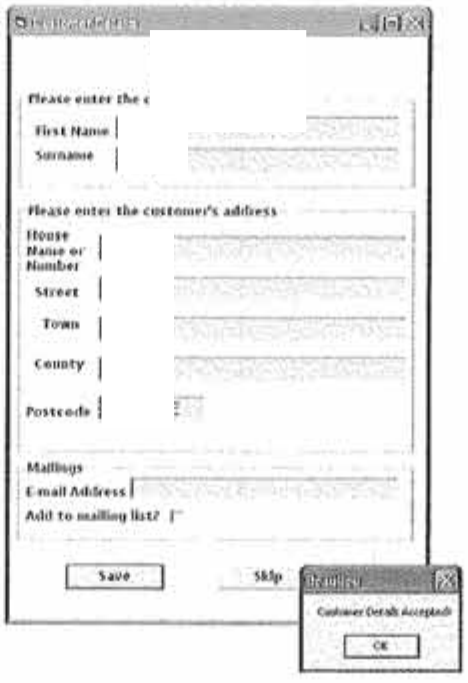
13	The cashier details were accepted.	
14	Must have both passwords.	

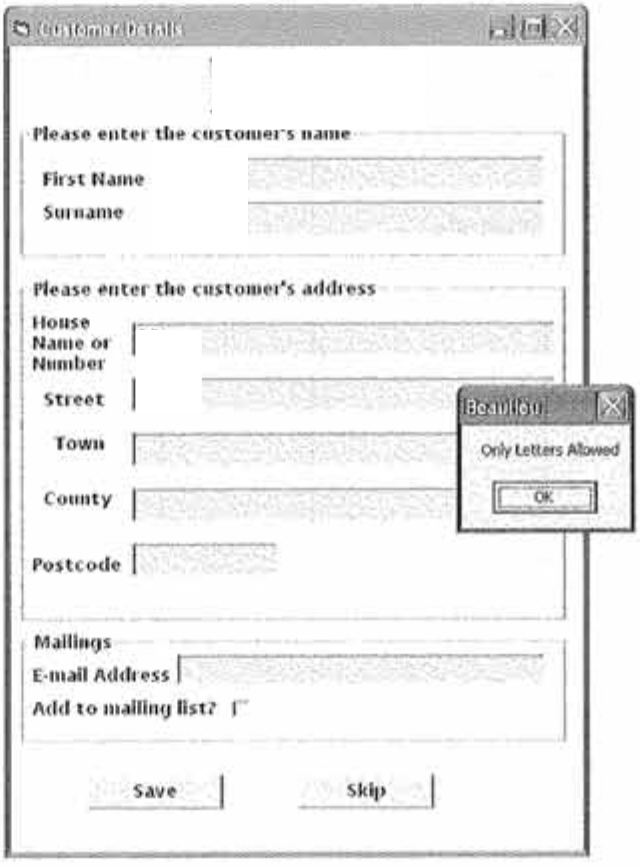
15	Passwords must be the same.	
16	Correct details are accepted.	
17	Invalid passwords are not accepted.	

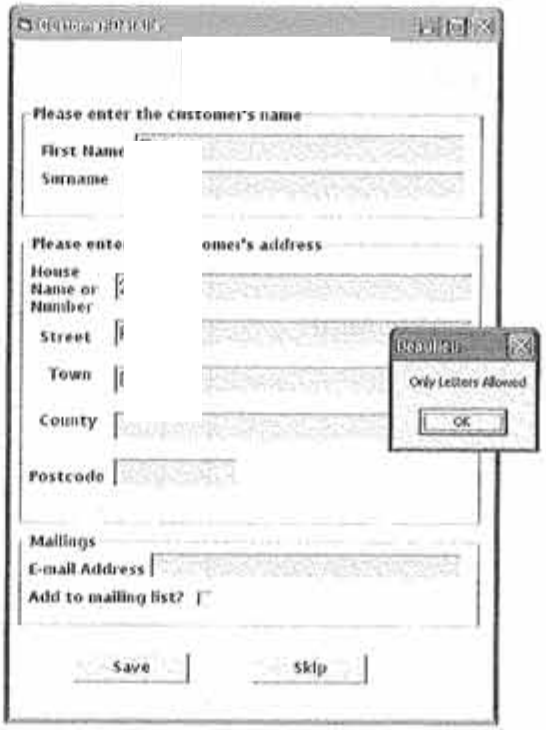
Customer Details

Test Number	Evidence Shows	Test Evidence
18	Only letters are allowed.	
19	Only letters are allowed.	
20	Name can't be left blank.	

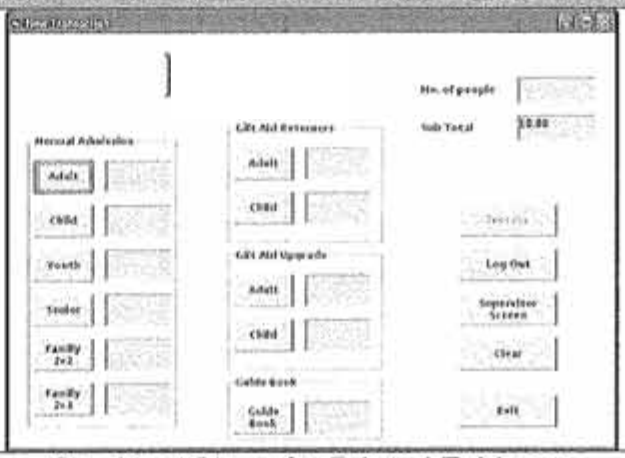
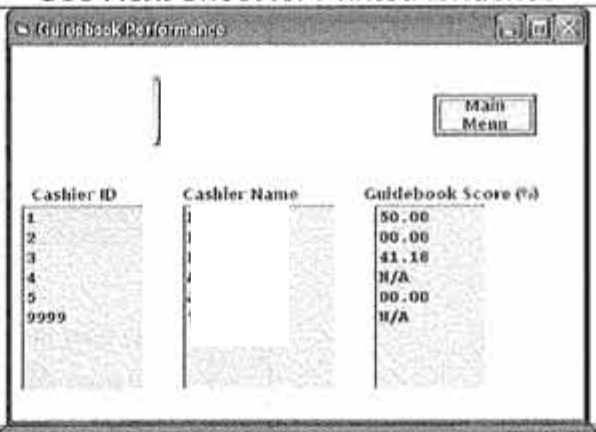
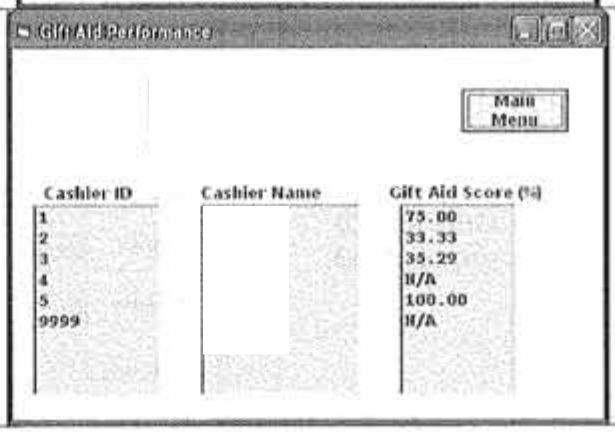
21	Name can't be left blank.	
22	House field can not be left blank.	

23	Post code can not be left blank.	
24	Details accepted.	

25	Gift Aided test customer is displayed in the GA list.	
26	Payment form is loaded.	
27	Only letters are allowed.	

28	Only letters are allowed.	
29	Only letters are allowed.	

Main Menu

Test Number	Evidence Shows	Test Evidence
30	Correct form loaded.	
31	Z report printed.	See Next Sheet for Printed Evidence
32	Correct form loaded.	
33	Correct form loaded.	

Date = 06/05/2010

ADMISSION

Number of Adults =	4
Number of Children =	1
Total Admission =	5

REVENUE

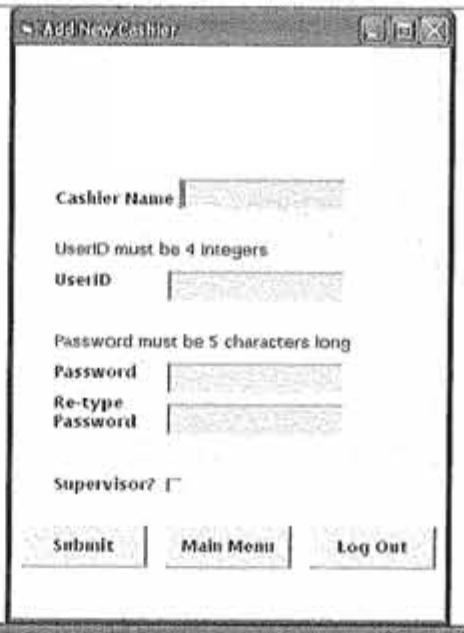
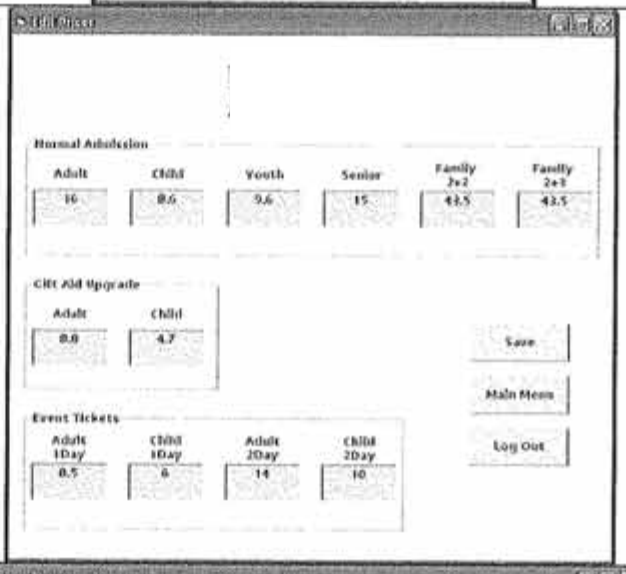
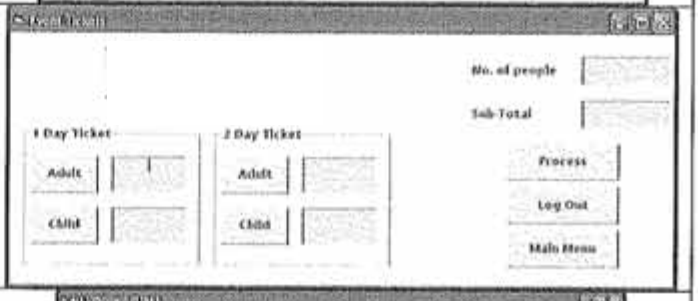
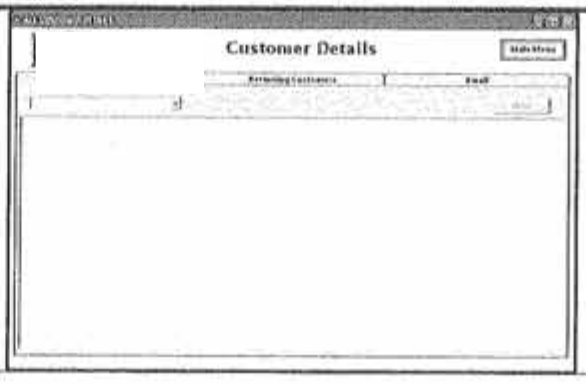
Ca.) Total =	£31.50
Visa Total =	£43.00
Total =	£74.50

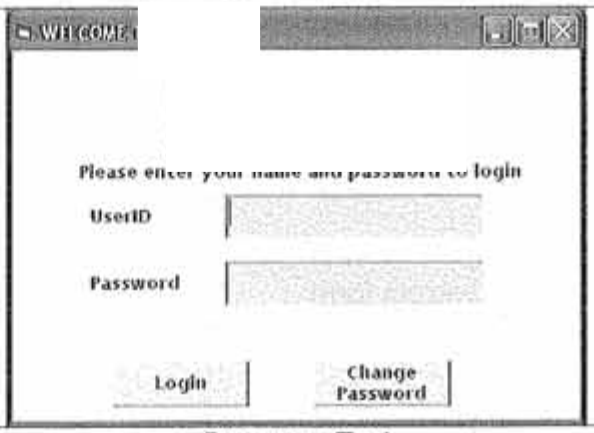
GIFT AID

Gift Aid Total =	£58.75
Gift Aid % =	78.86%

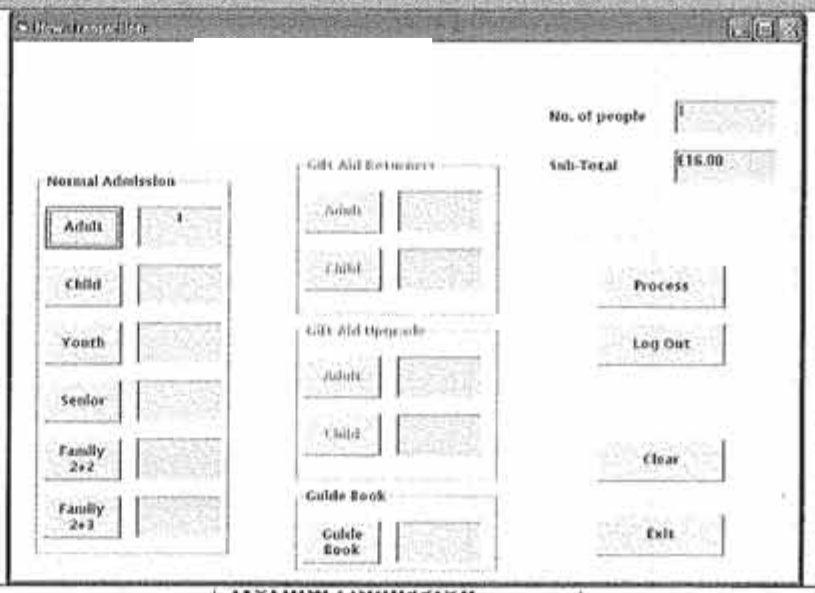
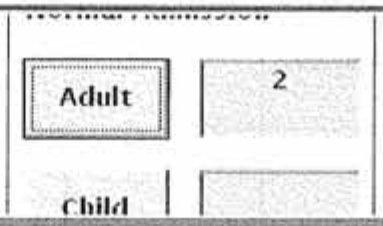
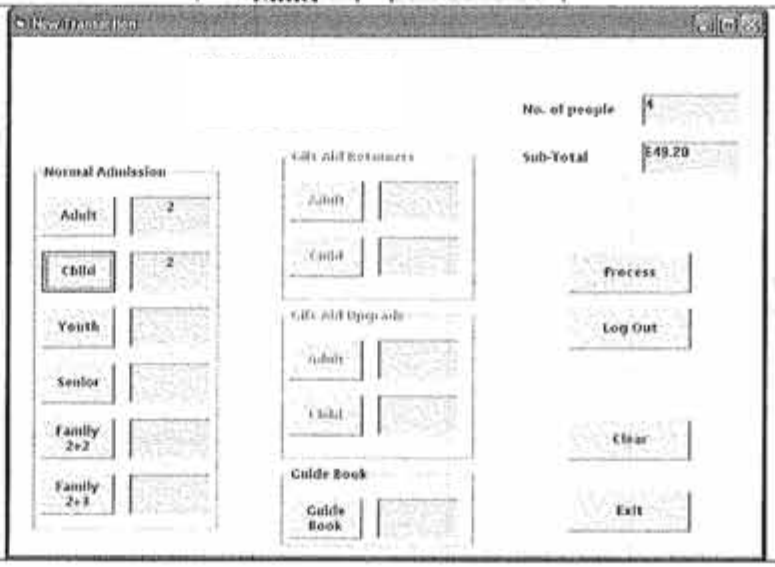
GUIDEBOOKS

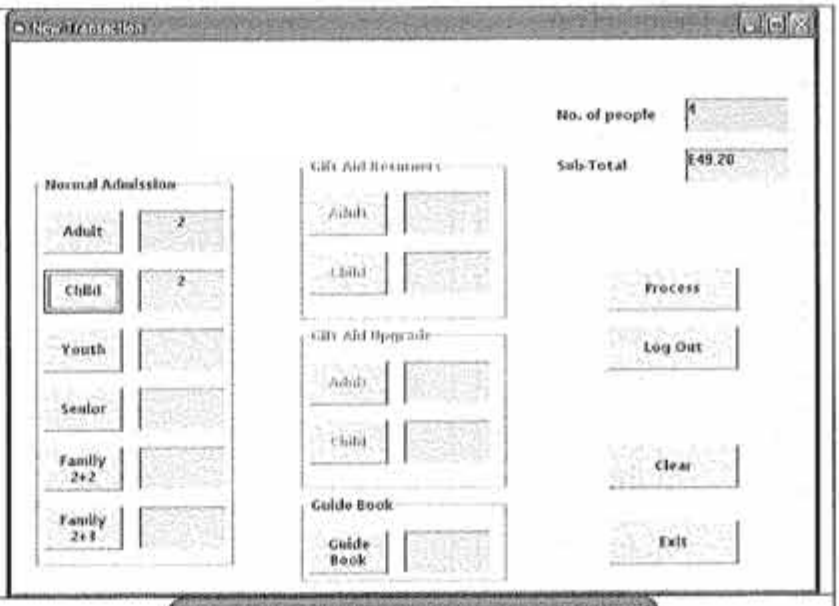
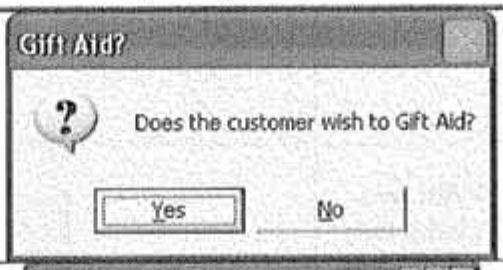
Guidebook Total =	1
Guidebook % =	20.00%

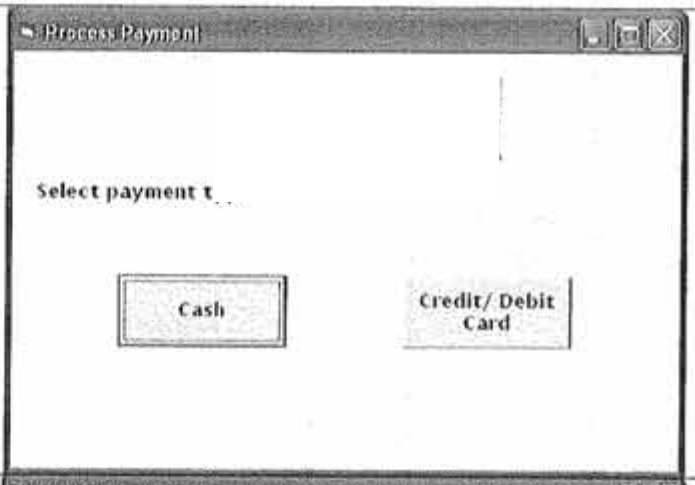
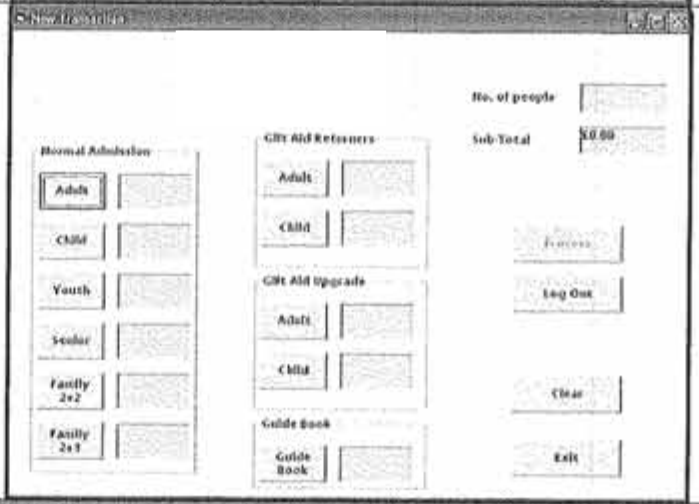
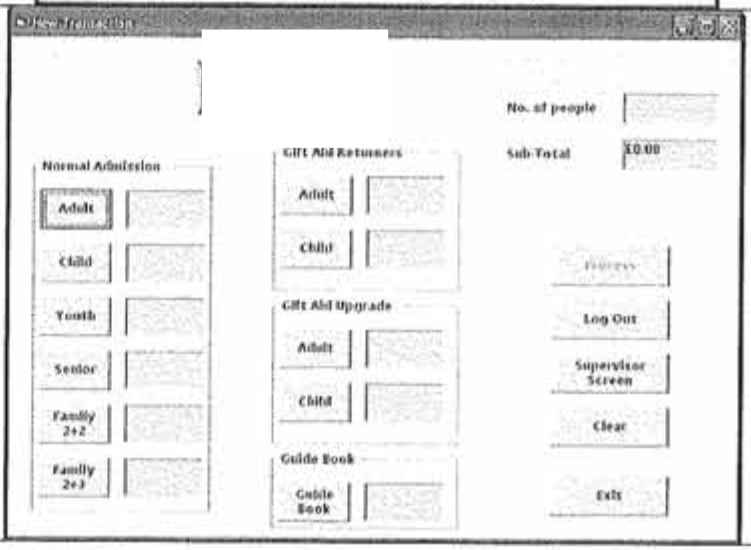
34	Correct form loaded.	
35	Correct form loaded.	
36	Correct form loaded.	
37	Correct form loaded.	

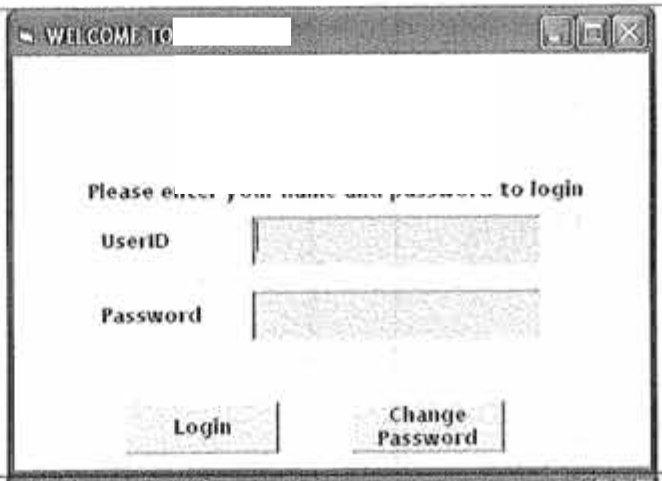
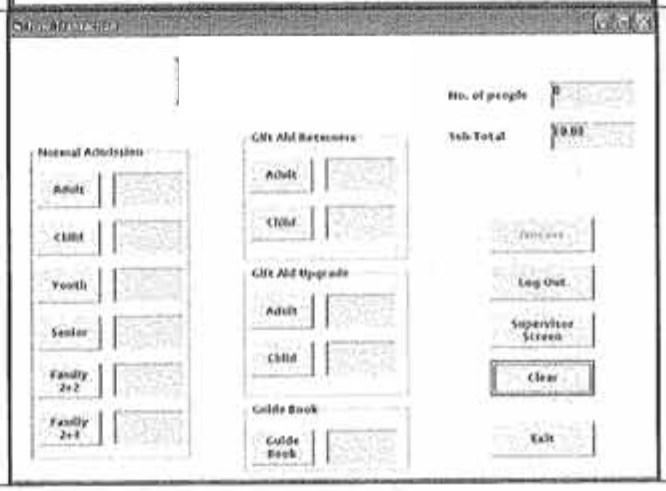
38	Correct form loaded.		
39	Program Ended	Program End	

New Transaction

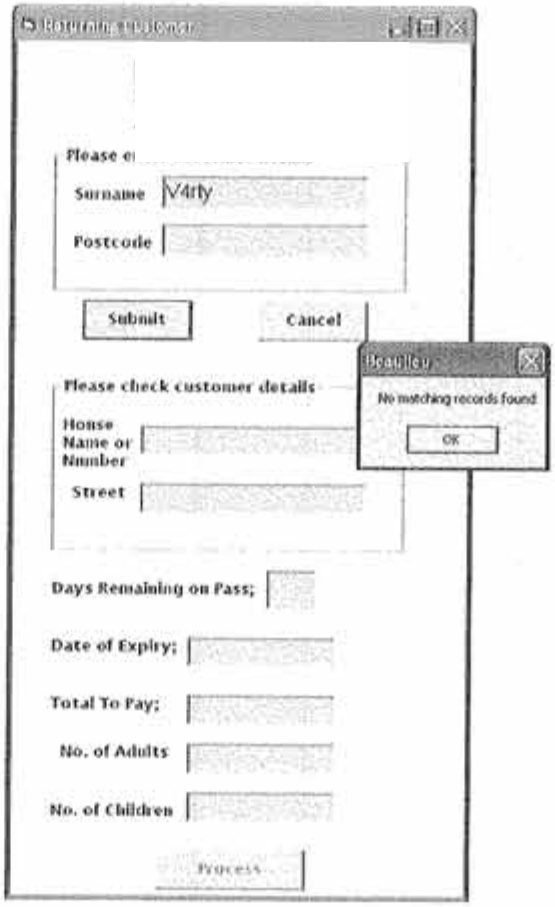
Test Number	Evidence Shows	Test Evidence
40	When one type of admission is selected the others are disabled.	
41	With two clicks two adults are added.	
42	Number of people correctly added together.	

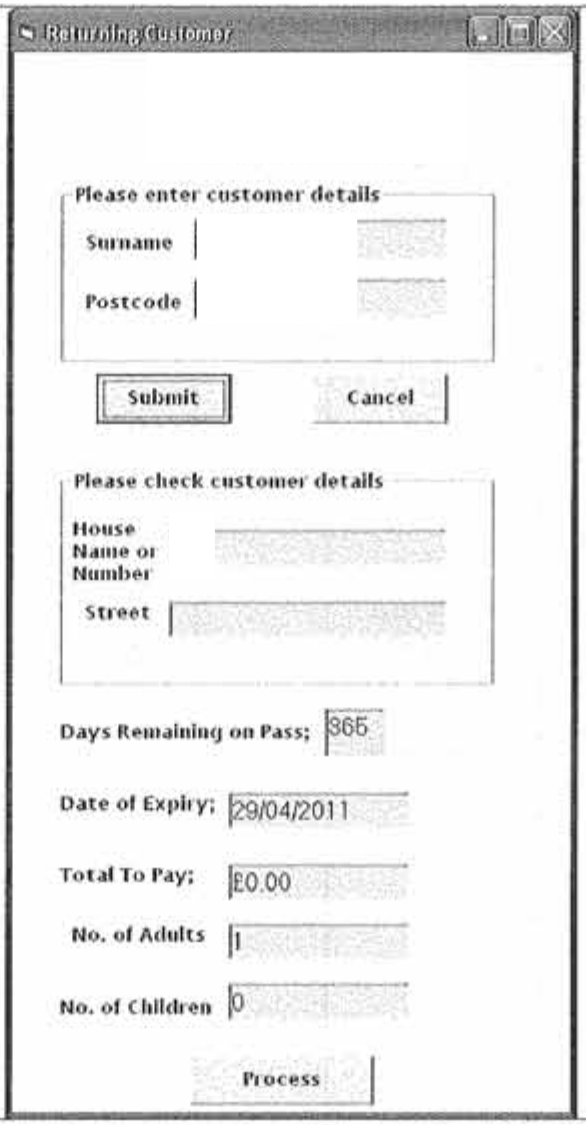
43	Sub total is correctly calculated and displayed.	
44	Correct message box is displayed.	
45	Correct form is loaded.	

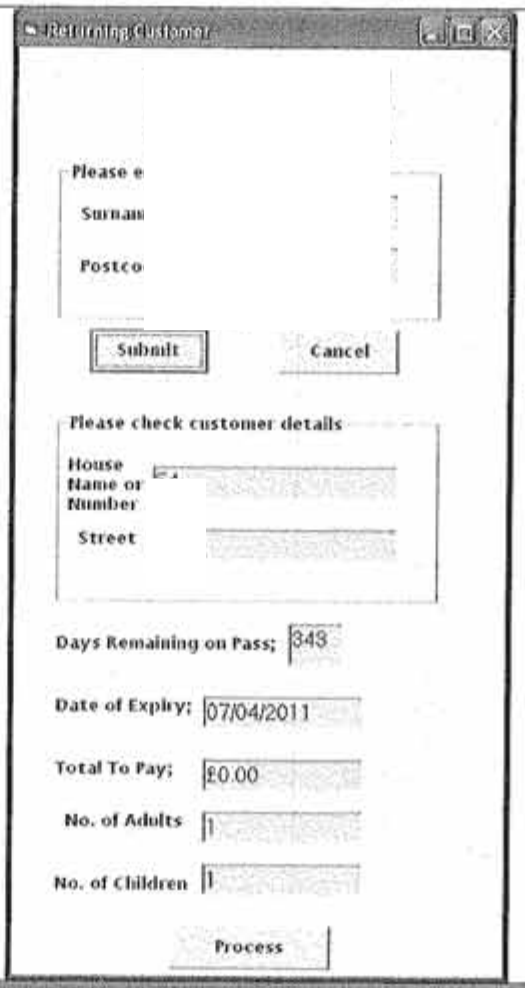
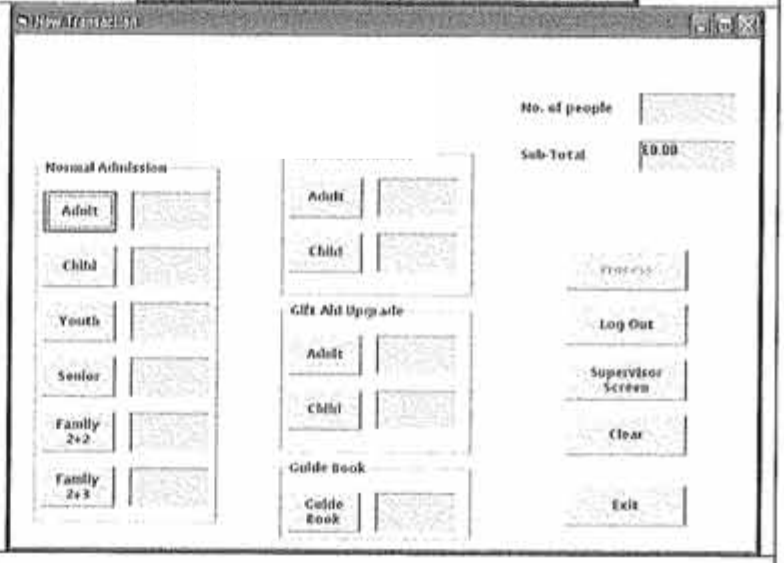
46	Correct form is loaded.		
47	Supervisor command button not shown.		
48	Supervisor command button is shown.		

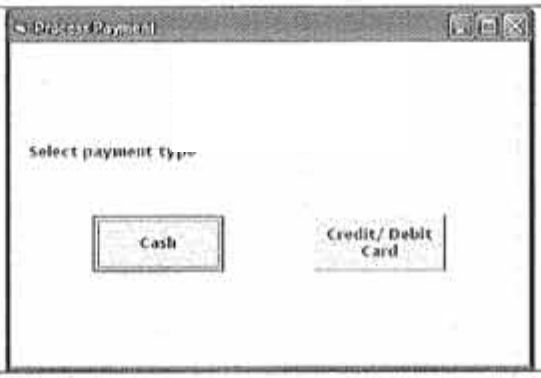
49	Correct form loaded.		
50	The form is cleared.		
51	The program ends.	Program Ends	

Returning Customer

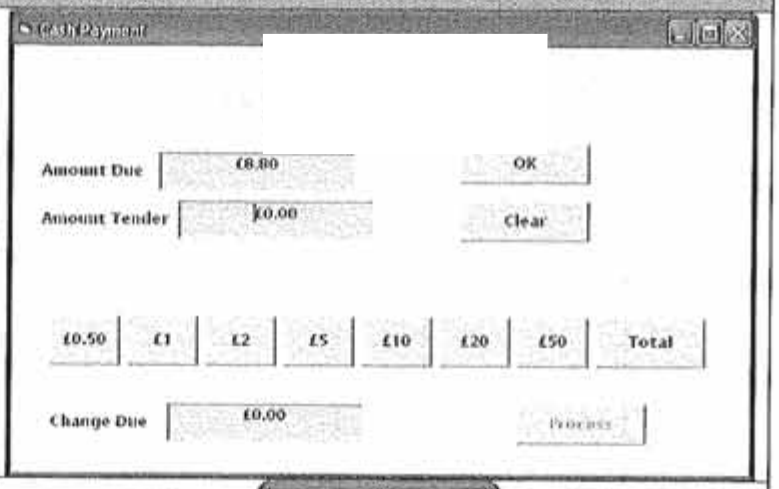
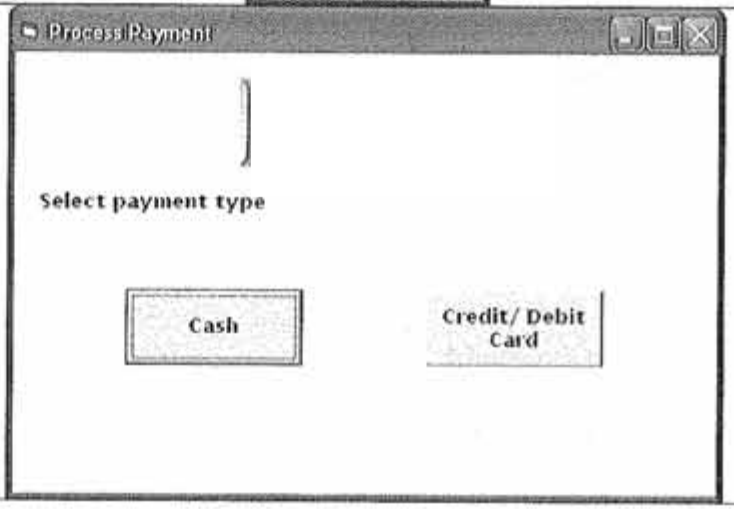
Test Number	Evidence Shows	Test Evidence
52	Details are accepted, still searches the records.	

53	Matching record found.		

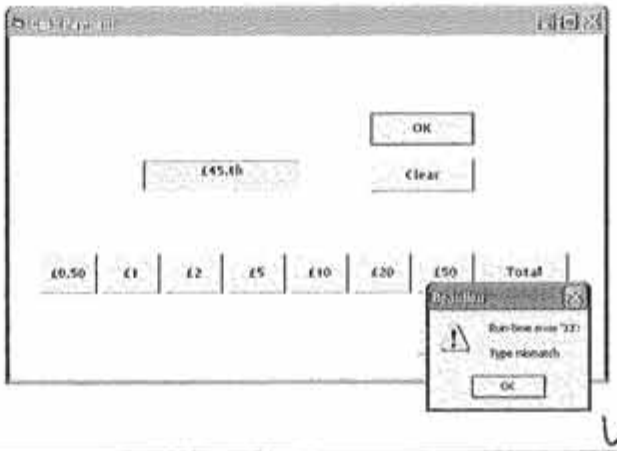
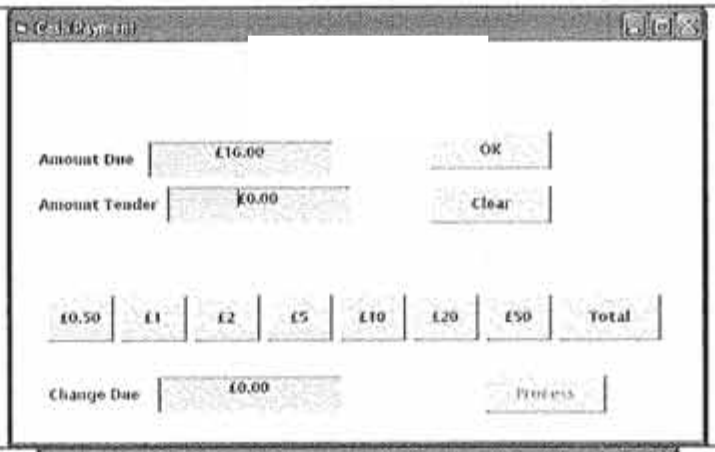
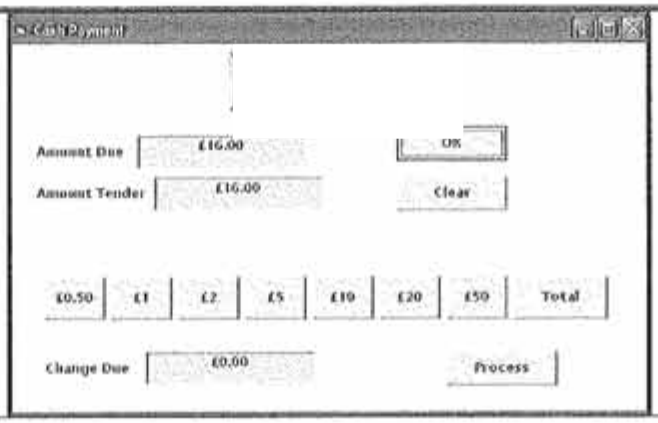
54	Matching record found.	
55	Correct form is loaded.	
56	Today's date is the 30/04/2010 so days remaining are correctly calculated.	Days Remaining on Pass; 364

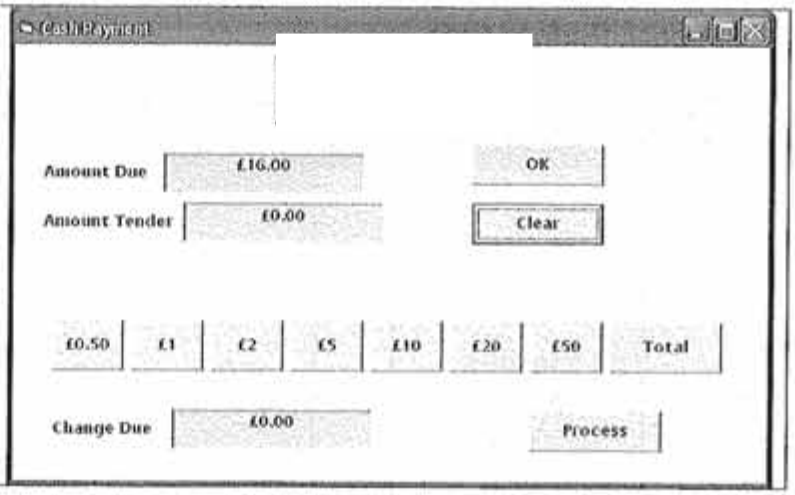
57	Date is correctly calculated.	**Date of Expiry;** 29/04/2011
58	Correct form loaded.	
59	Total to pay correctly displayed.	**Total To Pay;** £8.80
60	Number of adults correctly displayed.	**No. of Adults** 1
61	Number of children correctly displayed.	**No. of Children** 0

Process Payment

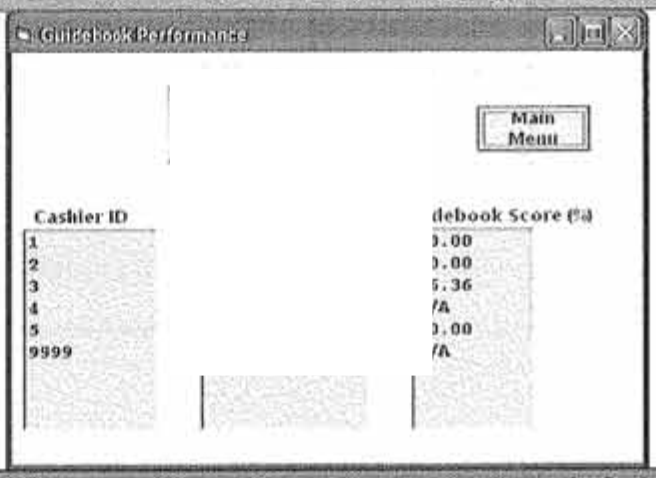
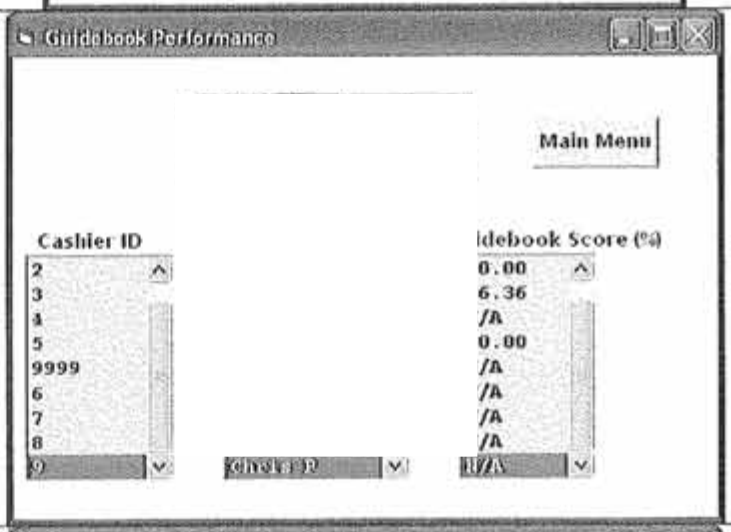
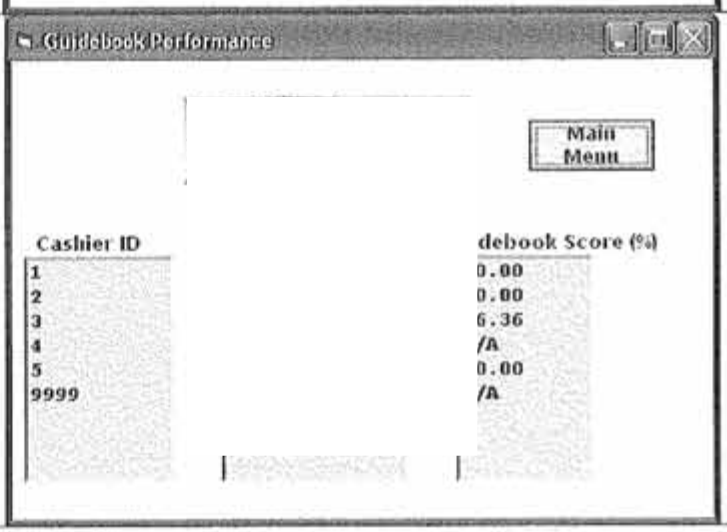
Test Number	Evidence Shows	Test Evidence
62	Correct form loaded.	
63	Correct message given.	
64	X is not disabled. ✓	

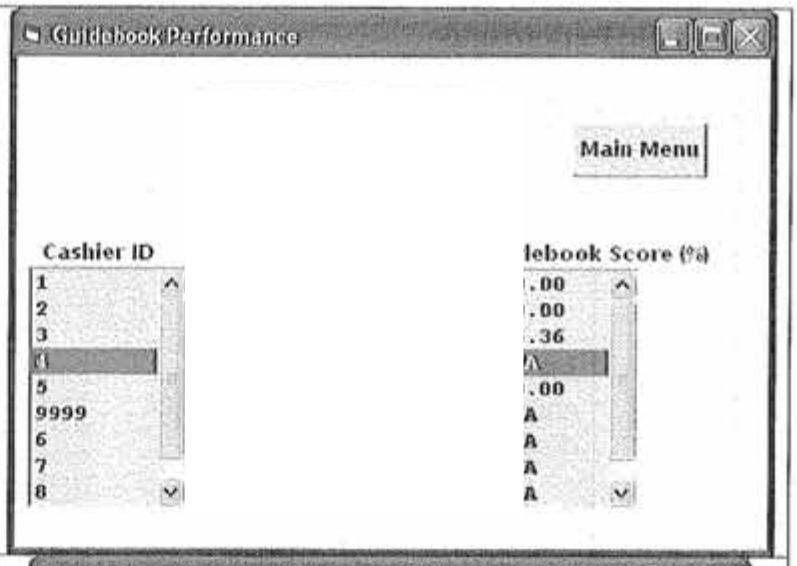
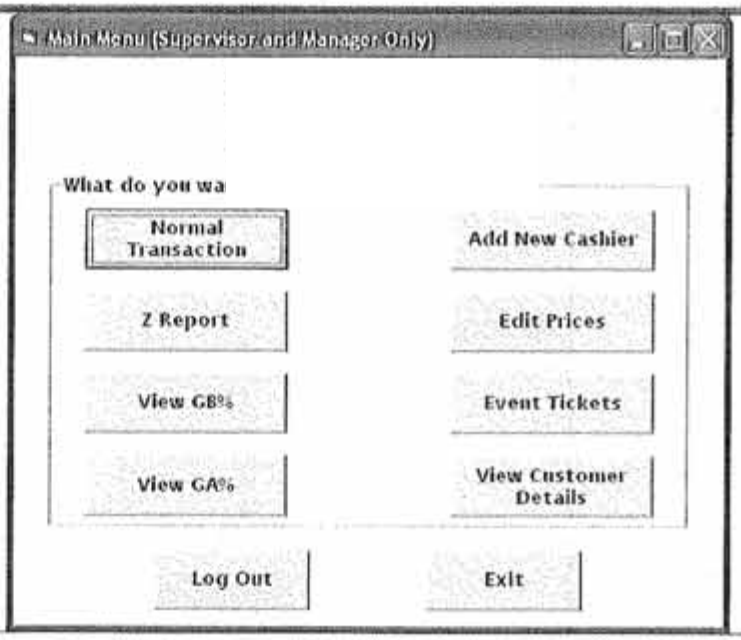
Cash Payment

Test Number	Evidence Shows	Test Evidence
65	Correct amount is displayed.	Amount Due £35.00
66	Correct amount is displayed.	Amount Tender £40.00
67	The program bugged.	
68	Correct amount is displayed.	Change Due £5.00
69	Command button is disabled.	
70	Command button enabled.	

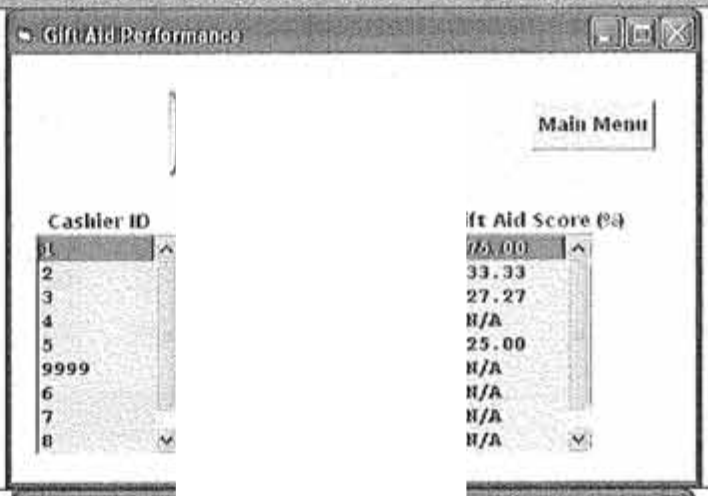
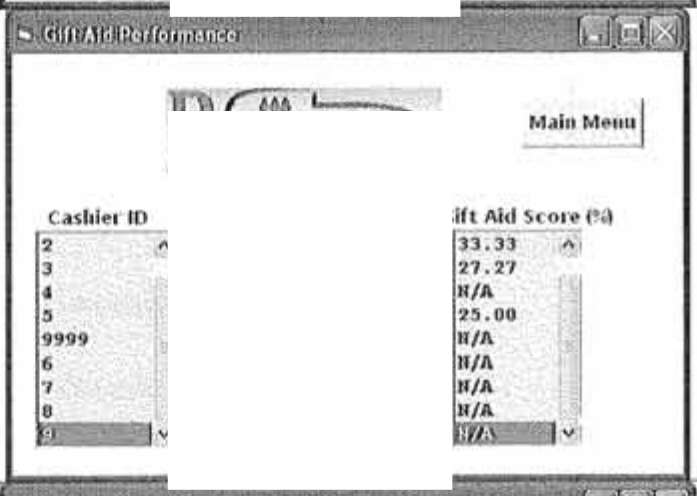
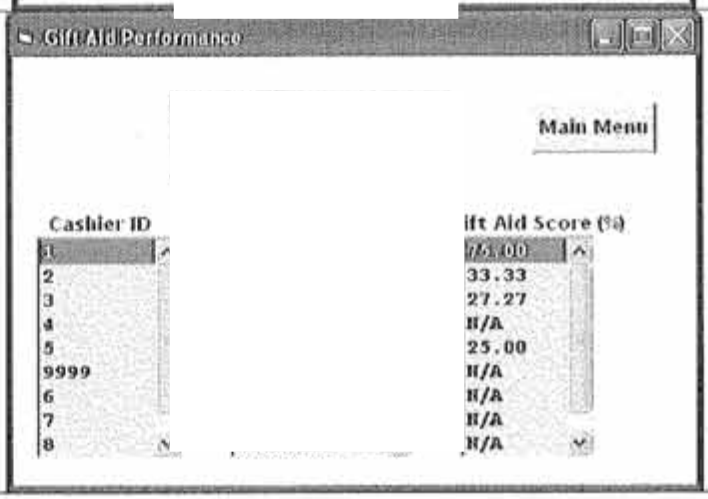
71	Form is cleared.	

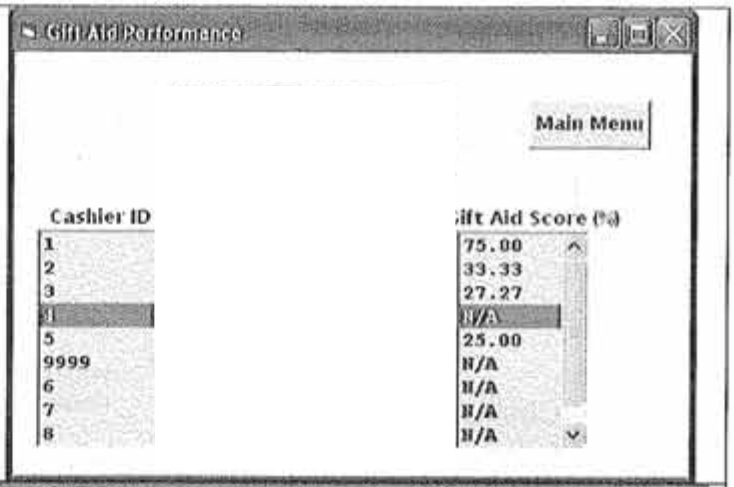
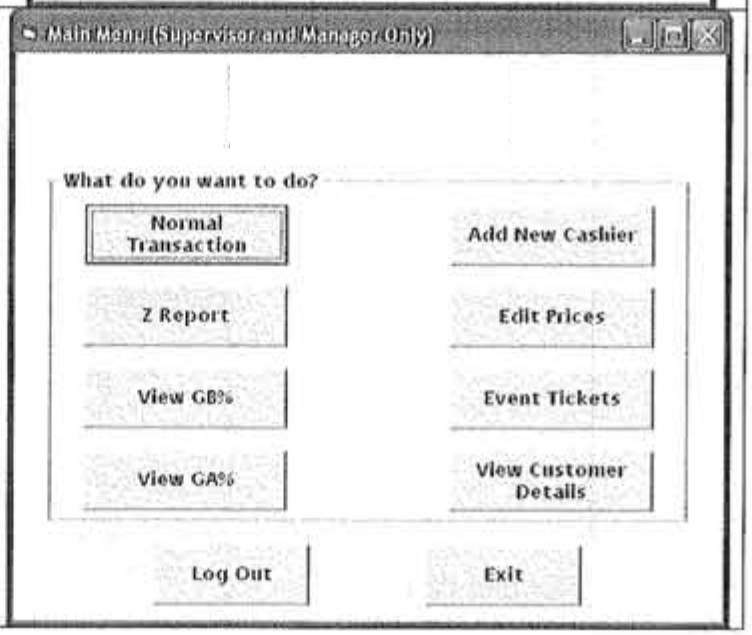
Guidebook Performance

Test Number	Evidence Shows	Test Evidence
72	All cashiers correctly displayed.	
73	All scroll simultaneously.	
74	The cashier called Helen has sold one guidebook to 2 people.	

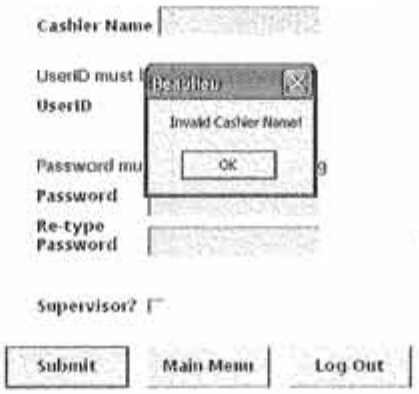
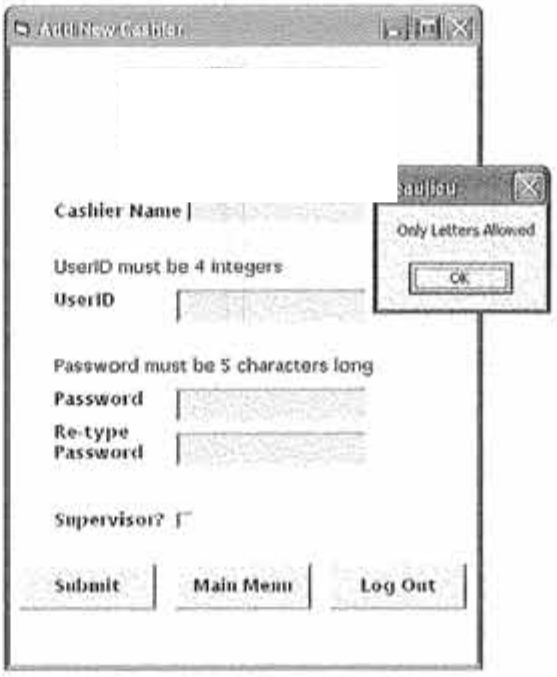
75	The cashier called Angela has not made any transactions.	
76	Correct form is loaded.	

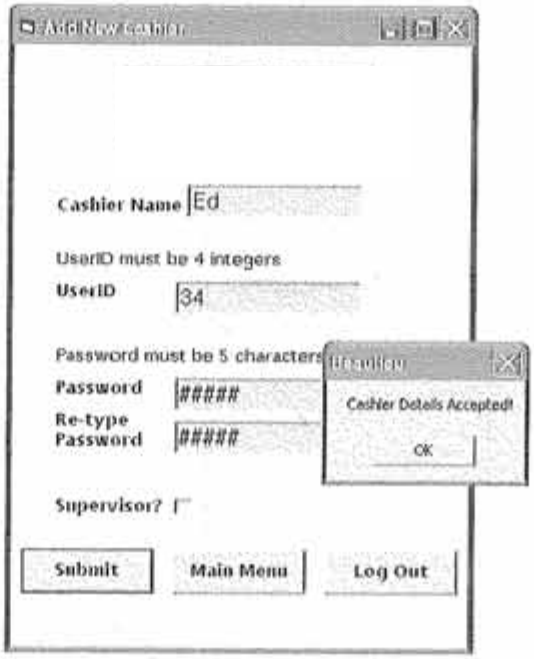
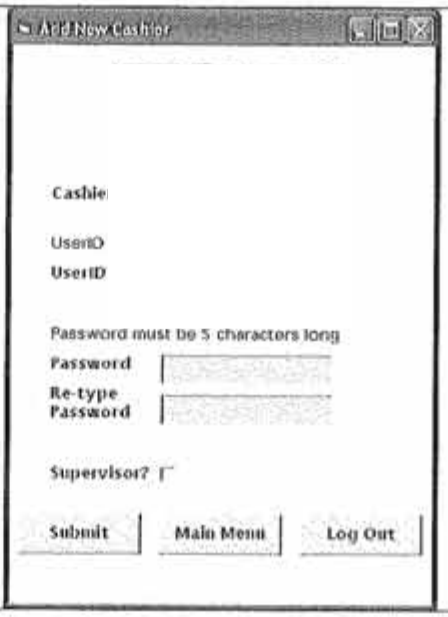
Gift Aid Performance

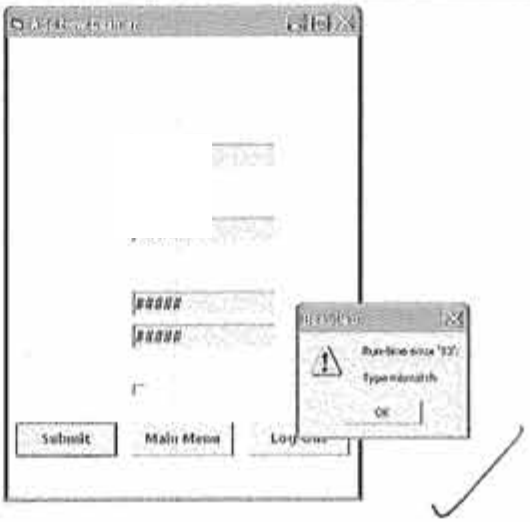
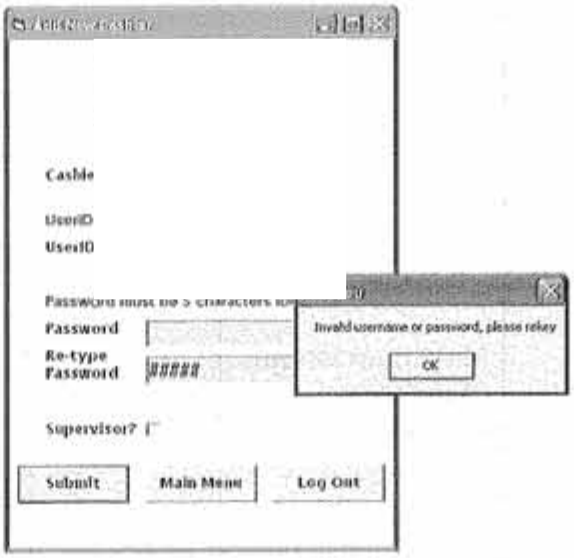
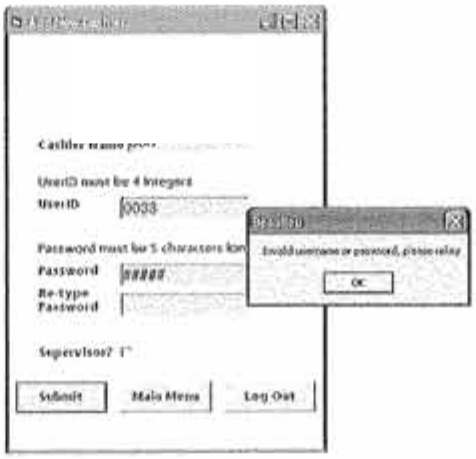
Test Number	Evidence Shows	Test Evidence
77	All cashiers are correctly displayed.	
78	All scroll simultaneously.	
79	The cashier called Helen shows the correct % for the example given.	

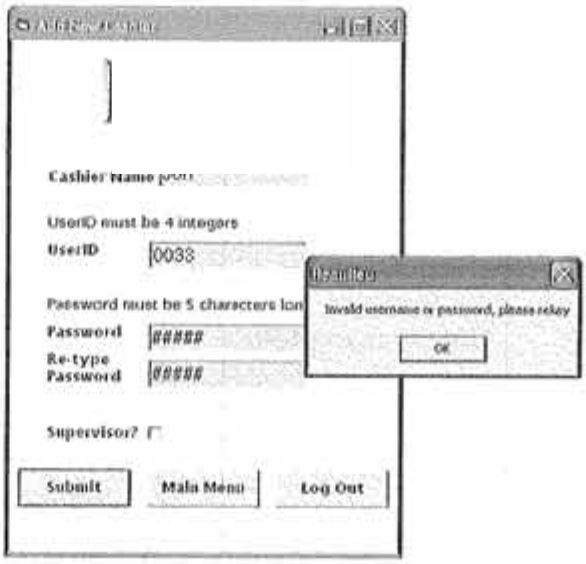
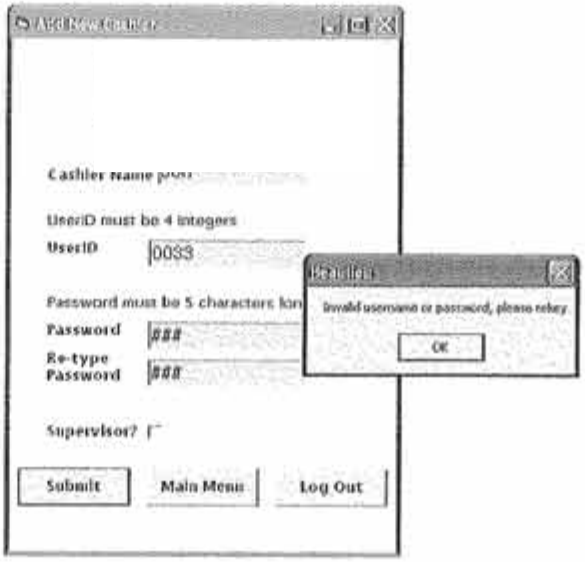
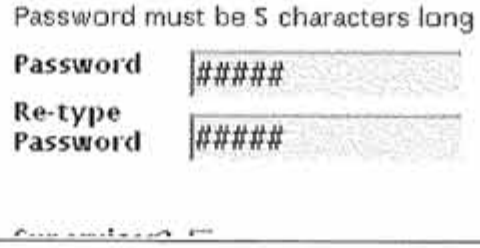
80	The cashier called Angela has not made any transactions.	
81	Correct form is loaded.	

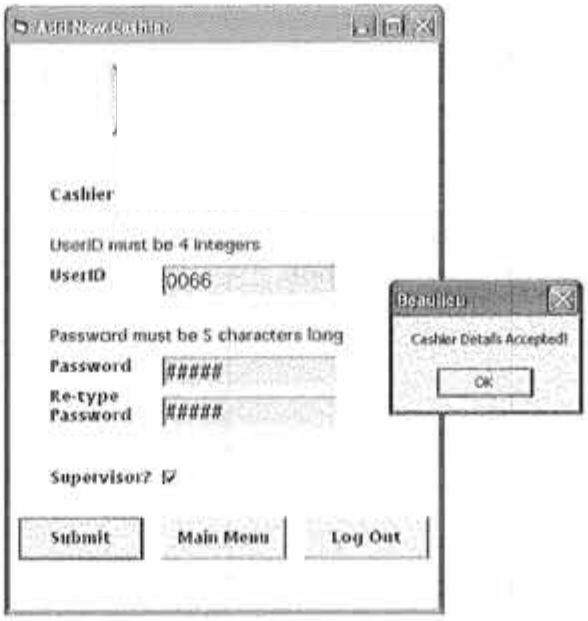
Add New Cashier

Test Number	Evidence Shows	Test Evidence
82	Cashier name can not be left blank.	
83	Cashier name can not contain letters.	

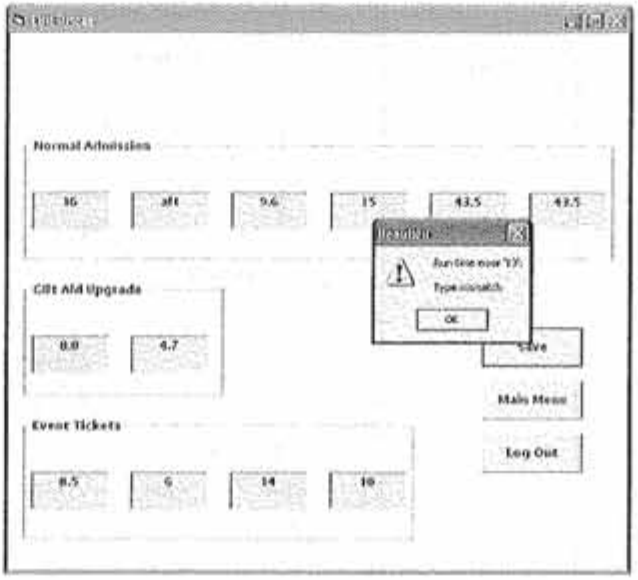
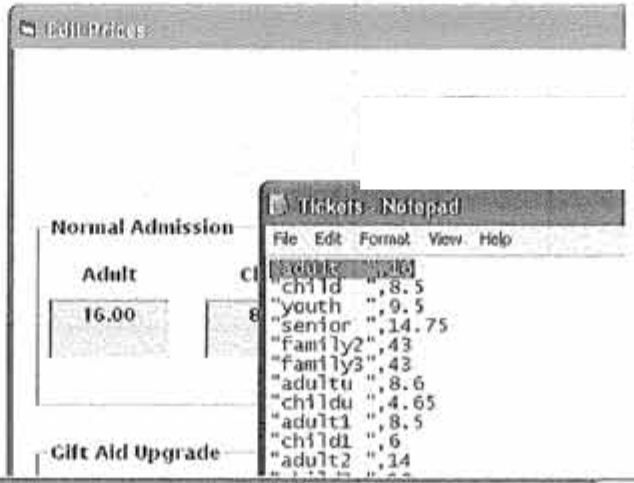
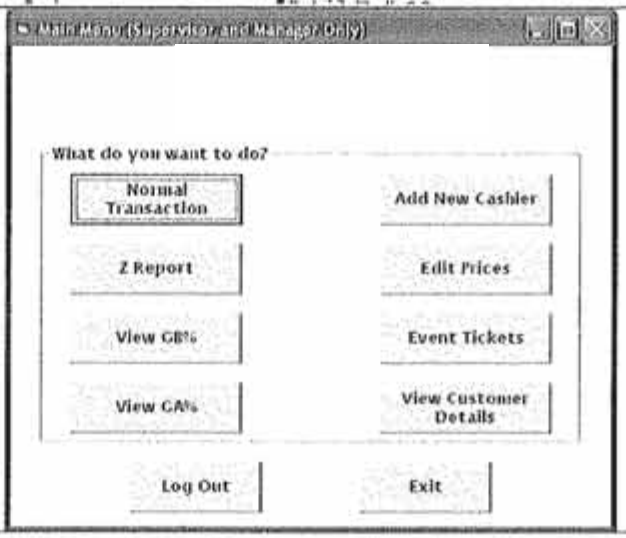
84	User ID was accepted.	
85	Maximum length of text box reached.	

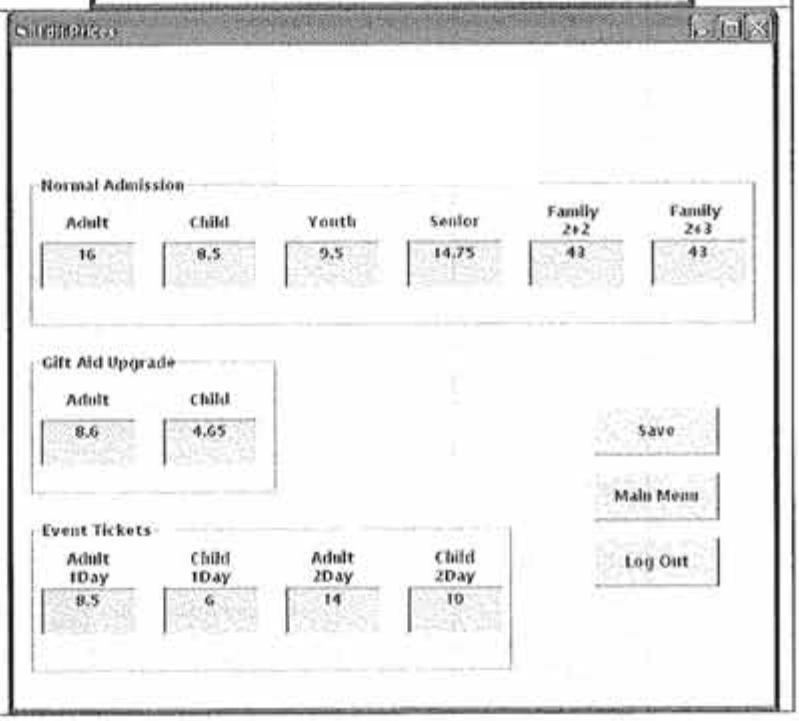
86	Program bugged.	
87	Password must be entered.	
88	Password must be entered.	

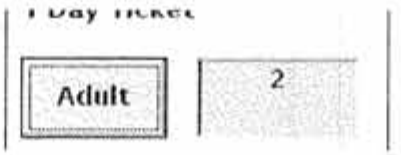
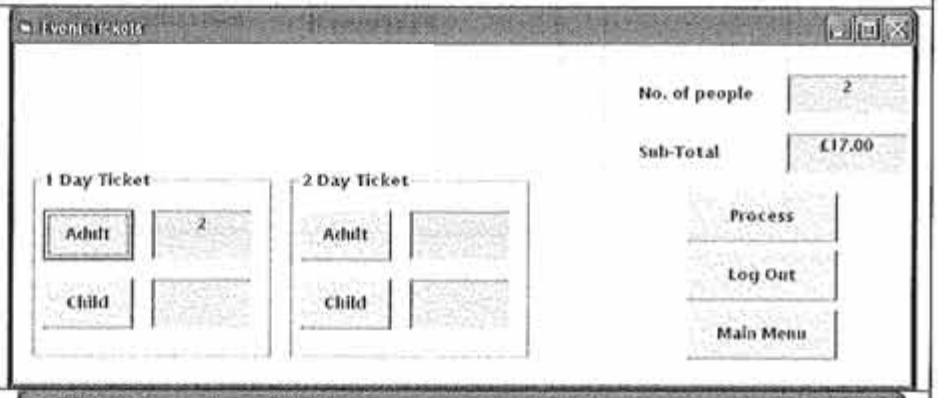
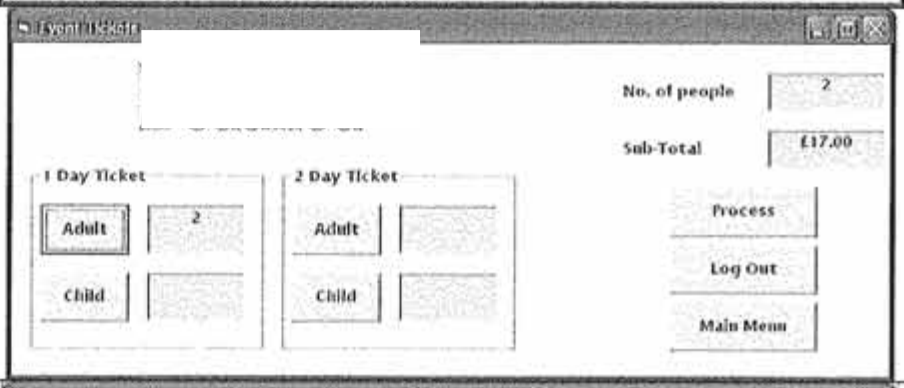
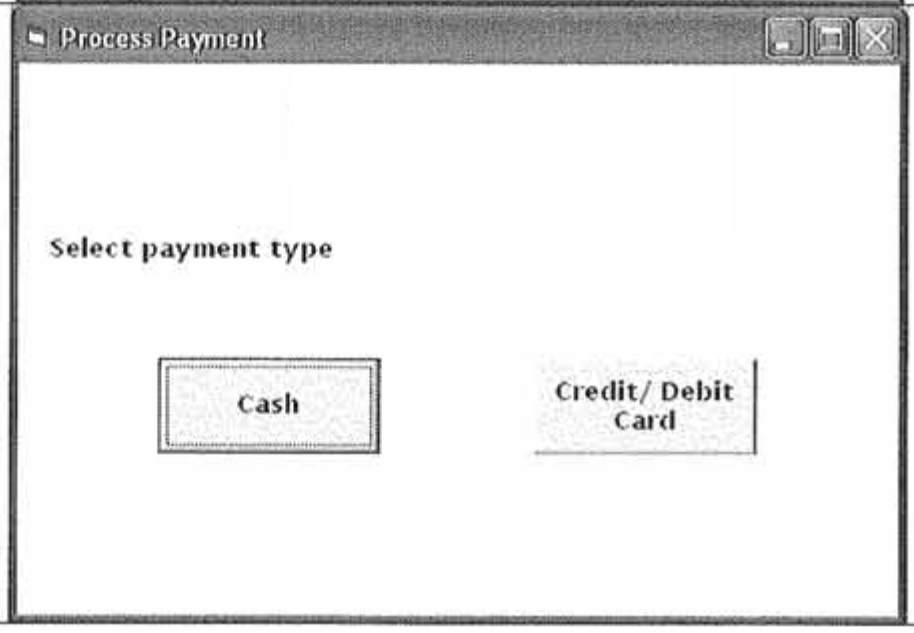
89	Incorrect passwords not accepted.	
90	Password must be at least 5 characters.	
91	Maximum length of the text box is reached.	

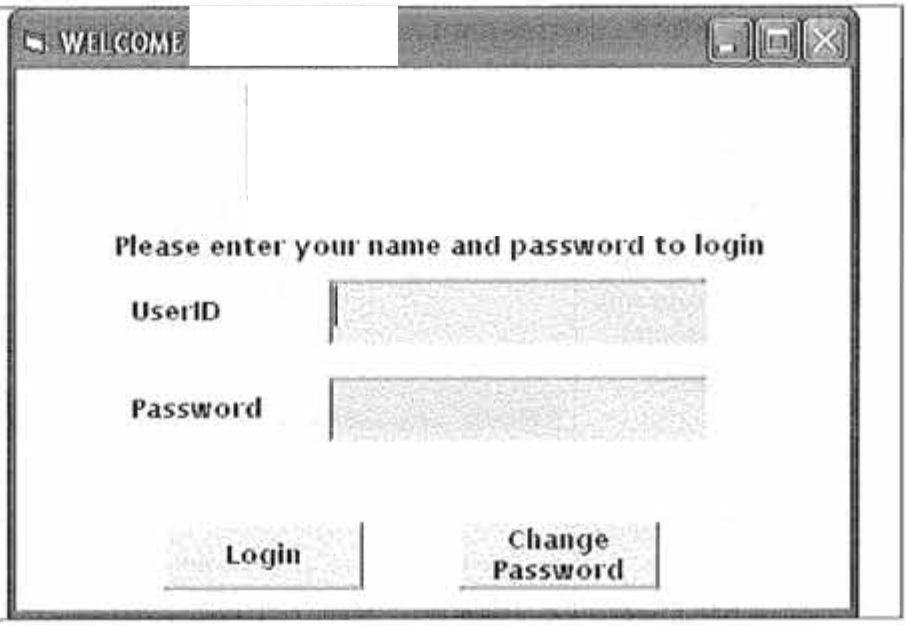
92	Details accepted.	

Edit Prices

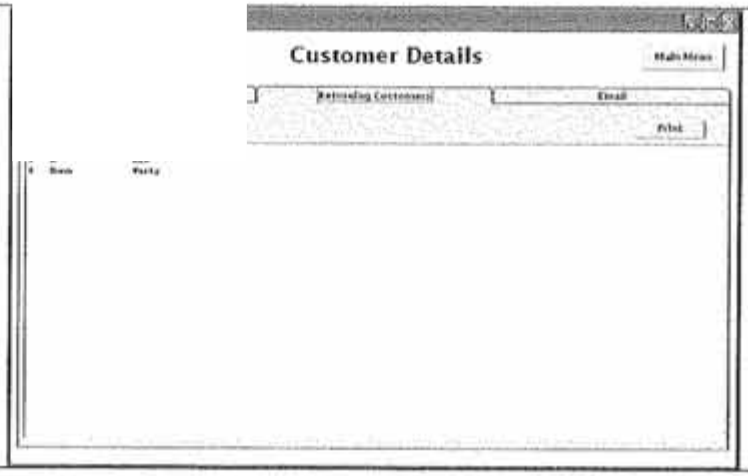
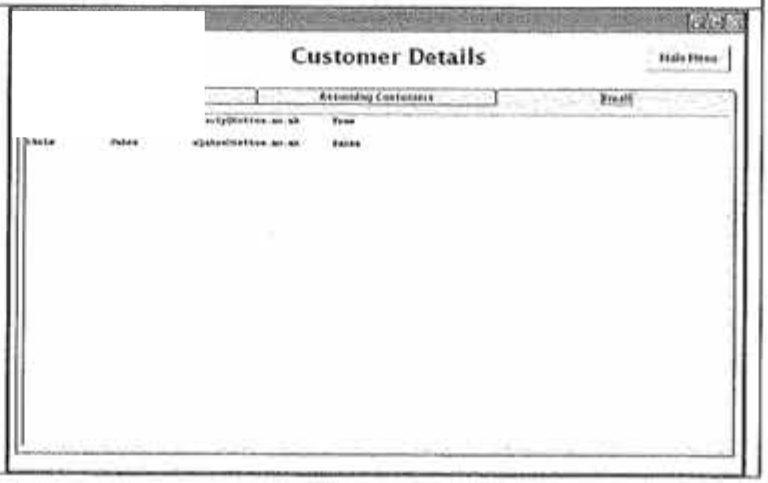
Test Number	Evidence Shows	Test Evidence
93	Program bugs.	
94	Data is saved although no message box is given.	
95	Correct form is loaded.	

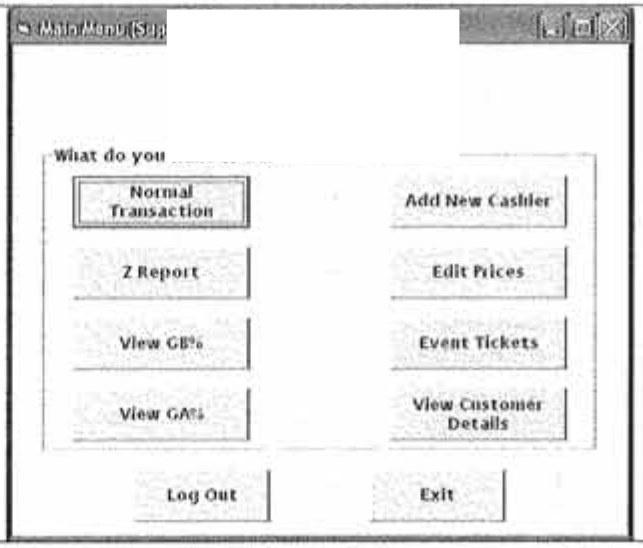
96	Correct form is loaded.	
97	Correct prices are displayed.	

Test Number	Evidence Shows	Test Evidence
98	With two clicks, two tickets are correctly added.	
99	Number of people correctly calculated.	
100	Sub total correctly calculated.	
101	Correct form is loaded.	

102	Correct form is loaded.	

All Customer Details

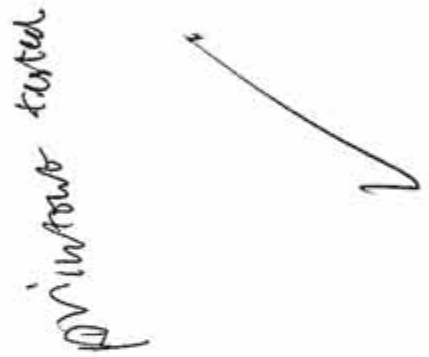
Test Number	Evidence Shows	Test Evidence
103	Correct customers displayed correctly.	
104	Details correctly printed.	See Next Sheet for Printed Evidence
105	Correct customers displayed correctly.	
106	Details correctly printed.	See Next Sheet for Printed Evidence
107	Correct details displayed.	

108	Correct form loaded.		

Aid Customer Details

omer Details for the Month April

	Address			Postcode
new	54	hjky	jlf	fjkc
Varty	28			



rning Customer Details

omer ID Name

h new

)

Re Testing

Having completed the testing I have found some errors within the program. I am going to go through these errors and make alterations in the code to try and fix the problems. When this is complete I am going to re-test each function to ensure it is working as expected.

Using tests for improvements to program

Test 13

Change Password Form

13	To ensure that password 1 can't be left blank.	" "	Error MsgBox.	The details were accepted.

This test showed that I had not correctly set the validation up on the text box. I therefore went back to the code and added the necessary validation step. The changes I made to the code can be seen below.

```
'Checks to see that the user has not left the form blank
If txt_userID = "" Or txt_current = "" Or txt_new1 = "" Or txt_new2 = "" Then
    MsgBox ("Please enter a valid user Id or password")
Else
```

This code is executed when the user has clicked on submit.

Test 52

Returning Customer Form

52	Check that surname can't be numeric.	V4rty	Error MsgBox.	The details are accepted as far as; the records are still searched, although as this validation has been carried out on first entry no matching records were found.

This test showed that again I had not set the validation up correctly. Whilst the details were not accepted as no matching records could be found I was expecting the form to display a message when the user typed the number into the text box. To ensure that this happens I added the relevant piece of coding which can be seen below.

```
Private Sub txt_surname_KeyPress(KeyAscii As Integer)

'Prevents the cashier from entering a number or invalid character in the customers name
If (KeyAscii < 65 Or KeyAscii > 90) And (KeyAscii < 97 Or KeyAscii > 122) And KeyAscii <> 32 And KeyAscii <> 8 Then
    MsgBox "Only Letters Allowed"
    KeyAscii = 0
End If


End Sub
```

This additional sub routine checks to ensure that the character the cashier is using when entering data into the surname text box is a letter.

Test 64

Process Payment Form

64	Check that the user can not push the X in the top right hand corner.	Click	The program will not end.	X is not disabled.

This test result has shown that there is an error with the coding, I have set up a public sub routine in the module which I have then been able to call from different forms to ensure that the X is disabled.

Having looked at the code I had missed out the code to call this sub routine on this particular form. The additional code I have added can be seen below.

```
Private Sub Form_Load()

'calls the sub routine from the module to disable the X so that the user can not close the form
Call DisableX(Me, True)

End Sub
```

Test 67

Cash Payment Form

67	Ensure that invalid values can not be typed into the amount tender.	45.th	A message box should be displayed asking for a suitable tender to be input.	The program bugged.

To ensure that this error does not occur I needed to carry out an additional validation check. The code needed for this validation check can be seen below.

```
Private Sub txt_amounttender_KeyPress(KeyAscii As Integer)

'Checks to ensure that the user can only enter a number, space or decimal point
If (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 8 And KeyAscii <> 46 Then
    MsgBox ("Please enter a valid amount of tender")
End If

End Sub
```

This code ensures that the user only types; numbers, spaces or decimal points into the text box. If they type another value then a message box is given asking them to enter a valid amount.

Test 84

Add New Cashier Form

84	Check that the User ID must be four integers.	34	Error MsgBox.	Details were saved.

This test shows that the user Id has been accepted when it is smaller than 4 digits. Again another validation check will be needed to overcome this. The additional code required can be seen below.

```
Private Sub txt_userID_LostFocus()

'ensures that the user must enter a user ID of 4 integers
If Len(txt_userID) <> 4 Then
    MsgBox ("Please enter a valid User ID")
    txt_userID = ""
    txt_userID.SetFocus
End If

End Sub
```

This checks that the user ID is 4 characters long and provides a message if it is not.

Test 86

Add New Cashier Form

86	Check that the User ID can't be alphanumeric.	12em	Error MsgBox.	Program bugged.

Again this test has shown that there is no validation in place to stop the user from using letters in the user ID. The additional validation I have included to stop this can be seen below.

```
Private Sub txt_userID_KeyPress(KeyAscii As Integer)

'Checks to ensure that the user can only enter a number, space or decimal point
If (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 8 Then
    MsgBox ("Please enter a valid amount of tender")
End If
```

Test 93

Edit Prices Form

93	Check that only numeric values can be entered into the text boxes.	adu12	A message box will be displayed asking for a valid price to be entered.	Program bugs.

This test has shown that the user can enter invalid characters into the text boxes. Again I have set up an additional validation check to ensure that this is not allowed. The additional code can be seen below.

```
Private Sub txt_ticketprice_KeyPress(Index As Integer, KeyAscii As Integer)

'Checks to ensure that the user can only enter a number, space or decimal point
If (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 8 And KeyAscii <> 46 Then
    MsgBox ("Please enter a valid amount of tender")
End If

End Sub
```

As the text boxes on this form are set up as an array I only needed to assign the coding to the name given to all of the text boxes to ensure that all of them carry out the same validation check.

Test 94

Edit Prices Form

94	Check that the Save button allows valid data to be saved.	16, Click	A message box will be displayed stating that the data has been correctly displayed.	Data saved but no message box given.

This test has shown that whilst the data is saved correctly the message box is not displayed. To ensure the message box is displayed I have added an additional piece of code.

```
    'calls the sub routine to save the new ticket
    Call savenewticket(t)
    MsgBox ("New Prices have been saved!")

Next t
```

After the sub routine has been called to save the data the message box is given to say that it has been correctly saved.

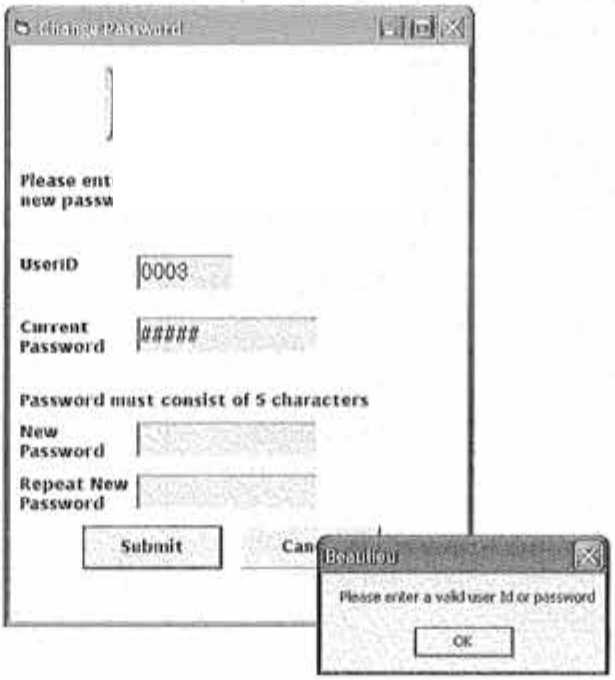
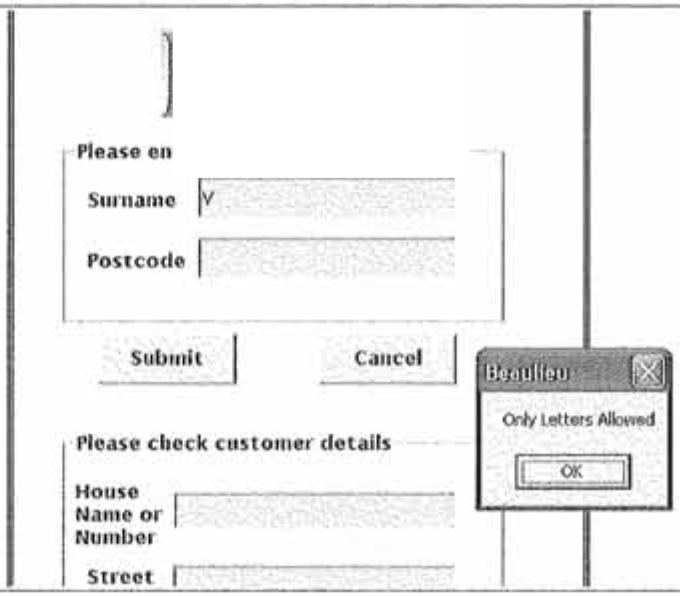
Having corrected the errors that I had identified I then went back and carried out the tests again to ensure that the corrections had worked.

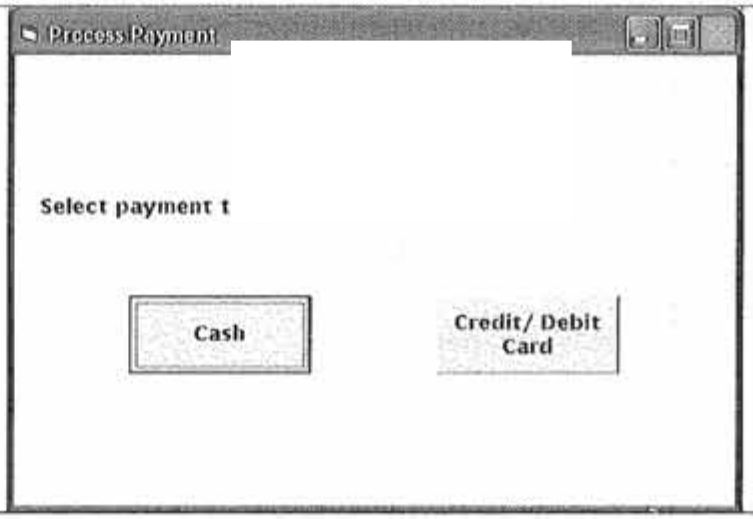
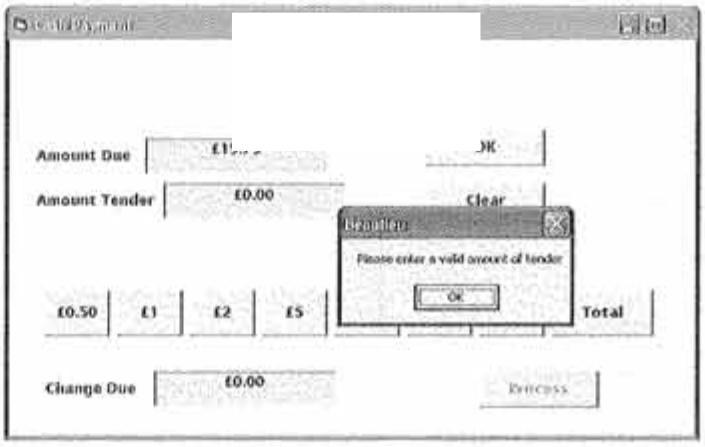
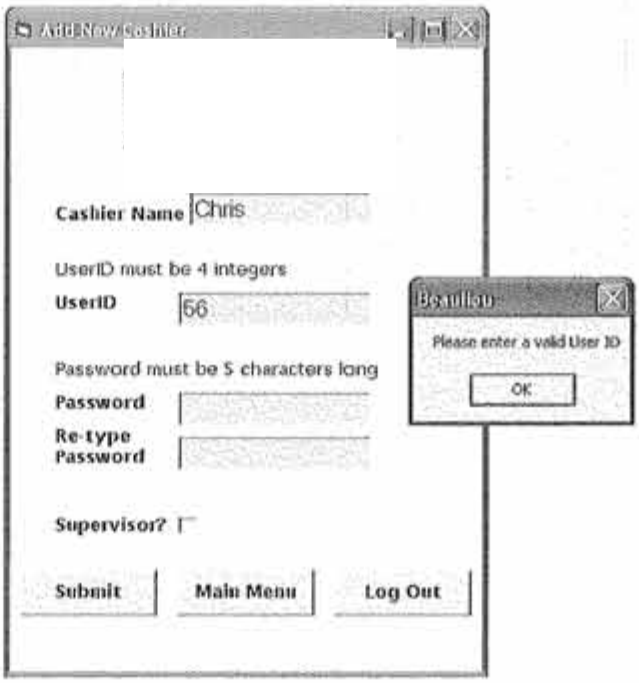
Re- Test Table

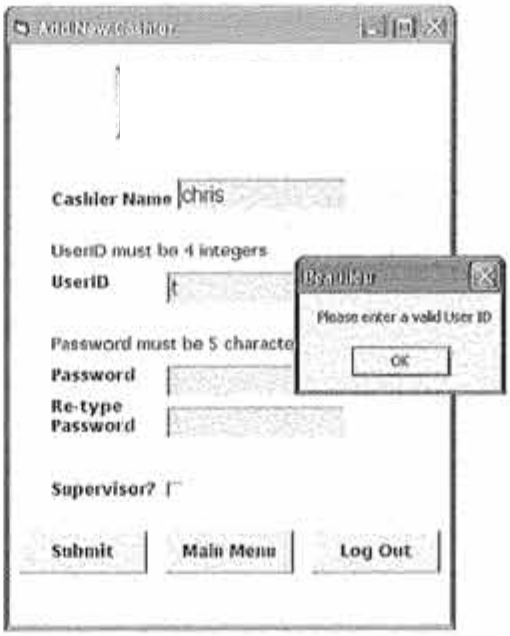
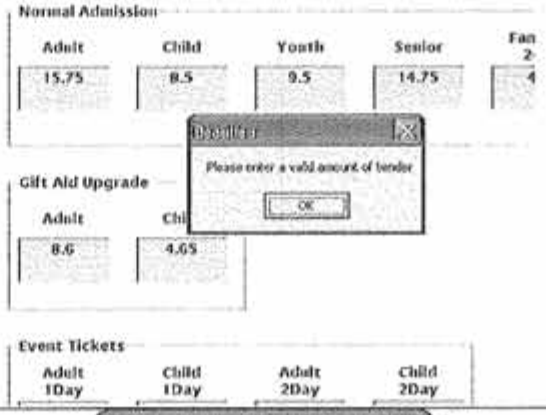
Test Number	Reason	Test Data	Expected Result	Actual Result
13b	To ensure that password 1 can't be left blank.	" "	Error MsgBox.	Expected
52b	Check that surname can't be numeric.	V4rty	Error MsgBox.	Expected
64b	Check that the user can not push the X in the top right hand corner.	Click	The program will not end.	Expected
67b	Ensure that invalid values can not be typed into the amount tender.	45.th	A message box should be displayed asking for a suitable tender to be input.	Expected
84b	Check that the User ID must be four integers.	34	Error MsgBox.	Expected
86b	Check that the User ID can't be alphanumeric.	12em	Error MsgBox.	Expected
93b	Check that only numeric values can be entered into the text boxes.	adu12	A message box will be displayed asking for a valid price to be entered.	Expected
94b	Check that the Save button allows valid data to be saved.	16, Click	A message box will be displayed stating that the data has been correctly displayed.	Expected

✓ Excellent!

Re- Test Evidence

Test Number	Evidence Shows	Test Evidence
13b	Password 1 can not be left blank.	
52b	The surname will not accept a numeric value.	

64b	The X has been disabled.	
67b	Letters are not accepted in the amount tender.	
84b	User ID must be 4 characters.	

86b	User ID can not contain letters.	
93b	The text boxes will only accepted numeric values.	
94b	A message box is now displayed to say that the prices have been saved.	

TASK E

EVALUATION

Having completed and fully tested the system I asked the end user to give feedback on the completed system. I also went on to analyse the system and my own performance throughout the project.

1. Program Solution

User Requirements

At the start of the project in the analysis section I set out several user requirements, each of these requirements is listed below with how I have met them.

The program must allow the till operators to;

- *Log in with an individual ID*
  To be able to use the system the user must login to the system using their personal user ID and password.

- *Enter and store customer details including; first name, surname, address and email address*
  When processing a transaction if a customer is Gift Aiding then the customer details form is loaded allowing the cashier to enter an save the customers details.

- *Enter and store the price of each transaction*
  The price of the transaction is automatically generated by the user entering the number of tickets that they require. The price is then stored in the transaction file.

- *Check customers in who have previously Gift Aided, it must also give the customer the option to upgrade their Gift Aid pass to give them access to the whole complex, therefore the system must store surnames, first names and postcodes of each customer with the number of tickets that they bought*
  The 'new transaction' form allows the cashier to select an upgrade ticket. The returning customer form searches for a returning customer as well as showing the number and type of tickets that the user has bought.

- *It must provide a check screen to ensure that the customer is declaring that they are a UK tax payer before the transaction is complete*
  When selecting 'Save' on the customer details form a message box is displayed to confirm that the customer is a UK tax payer.

The manager or supervisor must be able to;

- Log in using an ID that gives them higher access levels than normal till operators
  When a manager or supervisor log in the main menu is loaded giving them access to the extra forms

- *View all customer's details*
  A form 'All Customers' allows the user to view details about all of the customers.

- *View the total Gift Aid value*
  The Z report shows the Gift Aid % and total for the days transactions.

- *Edit the price of the tickets*
  The edit price form allows the user to edit the ticket prices.

- *View a total Gift Aid tickets sold for comparison of employee sales*
  The Gift Aid performance form shows the Gift Aid percentage for each cashier.

- *View the total guidebook sales and percentages for each till operator*
  The guidebook performance shows the guidebook percentage for each cashier.

At the end of the day the system must produce;

- *A completed list of customers and their details. This should be available in a hard copy so that all records can be kept*
  Whilst this report isn't printed all of the customers details can be seen on the 'all customers' form. The details are sorted by date so each days customers can be easily selected.

- *It must also calculate a total for the Gift Aid reclaimed on the entry*
  This total can be seen on the Z report printed at the end of each day.

- *Produce an output of customer email address and names in a different document to the rest of the customer's details so that it can be sent to the marketing department*
  The customers email address is not produced in a separate document but they are sent to a different list box to the rest of the details.

- *Produce a total for the number of people that have visited in the day and produce an overall percentage of the amount of revenue that has been Gift Aided*
  The Z report gives the number of people admitted as well as the revenue and Gift Aid totals.

Ideally the system should do be able to;

- *Give each transaction a unique identifier so that it can be identified for the Gift Aid*
  Each transaction can be identified by a unique transaction ID.

- *Allow event tickets to be sold through the system and allow the supervisors and managers to view all of the customers details that have bought event tickets*
  The system allows supervisors and managers to sell event tickets on the event ticket form. It doesn't however save any details about the customers who purchase event tickets.

- *Allow supervisors and managers to carry out refunds to void out the transactions that have been carried out*
  I did not meet this requirement as it was one of the requirements which was not essential the system still functions without this. I will however add this requirement to my list as a suggested improvement.

- *At the end of the day a report should be generated which calculates a percentage for the number of people that have bought a guidebook or upgraded their Gift Aid return ticket*
  The Z report contains all of this information.

Ease of Use

I believe that the completed system is very user friendly as it allows the user to move through the system step by step.

I have also tried to include helpful user messages to assist the user when they are entering data. Another way that I have tried to make the data entry easy is by the use of clear fonts and colour schemes as well as using many validation checks.

User Feedback

Having asked the user to try out the completed solution I asked them to fill out a form to help gather their feedback. This form can be seen in the appendix.

The points that the user identified were;

- The layout of the program makes it simple and direct for the user to process both the transaction and the Gift Aid.

- The returning customer form is well laid out and provides a lot of information about the returning customer.

- The buttons on the main transaction form are all the same colour. This may make the form confusing for the user.

- At present there is not a way of viewing a users ID and password so if they forget them they can not be changed.

- There is no option to be able to process vouchers or discounts. This would be an essential requirement if the user is to be able to process all types of transactions.

Having analysed the user feedback I then created a list of advantages and disadvantages of the implemented system.

Advantages

The system is very user friendly with the clear colour scheme and layout. The returning customer form allows the user to search for an existing customer. They are also given a lot of information about the customer such as; date of pass expiry, the number of adults and the number of children that originally visited.

Another advantage of the system is the wide range of information that the supervisors and managers can see. For example they can quickly and easily see the Gift Aid and guidebook percentage for all of the cashiers. They can also see all of the Gift Aiding customer's details as well as a mailing list. This is a major advantage compared to how they used to do this with the old system.

Disadvantages

One of the disadvantages of the system would be that the user can not process through vouchers or discounts. Whilst this was not one of the user requirements, if the system is going to be able to process all the different types of ticket this would be an essential requirement.

Another disadvantage would be that I was unable to get the program to be able to automatically find the customers address based upon their postcode. I had at first thought that this would be a simple task however I was not able to do this without purchasing a licence to do so.

Another disadvantage would be that the system only covers a very limited part of the admissions system. I briefly mentioned in the constraints and limitations that I would only be able to concentrate on one area however it would be better if the system could handle other types of ticket such as trust membership and Needs Ore permits.

Suggested Improvements

There are several ways in which I feel that the system could be improved;

- The manager could be able to change the DAT file at the end of each month so that the data is kept completely separate. This would involve me setting up an additional sub routine to create a new file. It would also involve additional programming on the all customers form to ensure that the data is gathered from the right DAT file for the right month.

- Another improvement would be to use the postcode lookup; this would make processing the customers through the tills much faster. To do this I would have to purchase a user licence with a piece of software that searches for the address.

- One of the improvements that the user suggested was the use of coloured command buttons. This would mean that the user would be able to more easily identify the differences between the types of tickets.

- Another improvement suggested by the end user was to ensure that the system can process vouchers and discounts. To be able to do this I could set up a combo box with a list of all the possible vouchers to allow the user to select the appropriate one.

- One of the possible but not essential user requirements was to allow managers and supervisors to void out items. This was something that I did not manage to complete. If I were to complete this I would need to change the validation rules to ensure that a supervisor or manager would be able to create a minus value in the text boxes, whilst the normal cashiers would still not be allowed to do this.

- As mentioned in one of the disadvantages the system could be improved so that it can be used in more departments of the business.

2. Own Performance

Arriving at Solution

To arrive at this solution I followed the stages of the systems life cycle. This involved me carrying out a thorough investigation to find out how the system is currently used.

I then went on to create a list of user requirements which I was then able to create a design for a programmed solution.

Having completed both a detailed analysis and design I was then able to move on to the implementation. Once this was completed I moved onto the final stages of testing and technical documentation.

Changes Made

As the systems life cycle is a cycle of iterative processes I had to go back to some stages and make changes before moving onto the next stage.

When implementing the designs of the forms I realised I had missed out some of the important objects such as command buttons. I therefore had to make some changes to my designs and add these objects in.

During the testing process I identified several errors. This meant that I had to go back to the implementation section and make the necessary changes to correct these errors. I then went on to the testing section again and re-tested the areas where I had to make changes.

Another change I made when implementing the system were to use some additional validation checks. I needed this as when implementing I thought of extra checks that I hadn't thought about in the design section.

What Went Well

Having previously completed unit 6 I felt that the analysis and design sections went very well. I used the techniques that I had previously learnt about, such as an interview and questionnaire, to find out as much about the current system as possible.

Generally I felt that most of the implementation went well. As I have previous experience of using Visual Basic I found setting up some of the simple coding fairly straight forward.

What Didn't Go Well

I struggled when implementing some of the more advanced pieces of code. For example when setting up some of the validation to check the keys that are being pressed, I struggled with trying to use the KeyAscii values to be able to implement this I had to ask for help.

Another feature that I struggled with was hiding the X on some of the forms. To be able to do this I had to use a piece of pre- written coding. I found this code on the VB Forums web site.

Another feature I struggled with was being able to look the customers address up automatically from the postcode. I had thought that this would be a simple process however I found that I would need to purchase a licence to be able to

do this. As this was something I could not do I was not able to implement this feature.

Mistakes Made

The main mistake that I made during the implementation was missing out validation rules. This meant that in some cases the invalid data was accepted or the program bugged.

Another frequent mistake I made during the implementation was typing errors. This meant that when I tried to run the program I would frequently encounter bugs.

Both of these mistakes were identified during the testing section.

Tools Used

During the analysis and design sections I used several different tools. This included data capture techniques such as an interview, questionnaire and document analysis. I then used other tools to represent the design of both the current and the new system. These tools included system flow charts, data flow diagrams and entity relationship diagrams.

To create the program I used Visual Basic code and Visual Basic 6.0 to do this.

Conclusion

In conclusion I am very pleased with the overall completed project. During the task I have further developed my analytical skills and gained confidence in using some of the design tools. I have also broadened my knowledge of using Visual Basic.

If I were to complete a similar task in the future I would be confident in doing this. I would do a few things differently, for example I would check the form designs with the end user. This could have meant that the user could have suggested the different coloured buttons in the design stage so I could then have implemented them the first time.

APPENDIX

User Feedback Form

Strengths- What works well?

- Layout simple and direct
- Upgrade well laid out ✓

Weaknesses- What doesn't work well?

- Buttons could be coloured to divide GA & upgrades with normal admission.
- Cashier name not displayed on Z report. ✓

Suggested Improvements

- Can't see user ID or password - if their lost then what? could an over write be made
- Could add option to process vouchers and discounts. ✓

Any Other Comments

- Well thought out - very impressed. ✓

Completed By;

—

Date;

17th April 2010

GA. ✓
Authenticity Confirmed.