# COMP4 Coursework

Joel Butcher

December 4, 2014

# Contents

# Chapter 1

# Analysis

## 1.1 Introduction

### 1.1.1 Client Identification

My client is Josh Campbell, he is 24 years old. He uses computers regularly for deisgn work, so has experience of computer systems. He uses his computer to design flyers, handouts, banners and visual graphics for projection, as well as surfing the web, email and various social media networks.He rarely uses hard copies other than to preview hes work before sending it off to print. Josh uses a 2012 Mac Pro with the latest version of Apple's operating system, OS X (10.9).

Josh is the head of the media department for Cambridge Community Church. This involves being responcible for the large amount of Audio and Visual equipment used on the churches Sunday services. This currently invloves spreadsheet with limited info on each item.

Josh would like to have a database management system to be able to hold information about each item and their various attributes. He would likke this database to be lovated on the churches central server so that it can be accessed by all staff if it it deemed necessary. He would use this database to store location, value and insurance details incase of damage or theft.he would like all of the information kept as a virtual copy as well as a hard copy to kept as a visual backup in case of harddrive failure or corruption.

He would also like to keep the location of each item as up to date as possible and if the location changes, he would like to be notified by email when it is entered/updated in the system.

### 1.1.2 Define the current system

The current system consists of multiple excel spreed sheets. There is one spread sheet for each of three locations; main office, main church building, and storage. Each spreedsheet consists of items located there as well as information on the value of each item, the quantity and the total value for the items with multiple entries. Each spreedsheet is divided up into equipment type (i.e Cableing, lighting, audio, visual/camera's)

### 1.1.3 Describe the problems

There are a number of problems with the current system. One of the problems is that there is no notification system to tell you when information is getting outdated or something is changed. For example, if an item is bought or sold, the total costings for that item will be updated and no-one will be notified. Another problem is that the current system doesn't show the PAT testings for all the items, these tests go out of date every 6 months and there is no way of being notified when a new PAT test is needed on an item.

### 1.1.4 Section appendix

Below are the questions that I asked my client at the interview and the answers he gave to me. I have typed up the questions and answers in markdown format then imported it as a pdf document so that it is easier to read.

Figure 1.1: Interview Questions (pg 1)

# Interview Questions

1. What does the current system do?
    - Multiple excel spreadsheets that list all the AV equipment
2. What are the problems or drawbacks of the current system?
    - There is no notification system
    - Data is easily out of date.
3. How much data is currently recorded?
    - Current data stored is the item name, its location, the quantity and it's value.
4. What extra data will need to be included?
    - PAT testing's
    - Current location
    - The item's usable state (working, in need of repair, being repaired etc)
5. How frequently will the data need to be updated?
    - The data will need to be updated a few times a month or so
    - Whenever the location changes.
6. Will new records need to be added or deleted? If so, how often?
    - New records will need to be entered, or some deleted every couple of months.
    - Whenever new equipment is bought or if an item is sold
7. How important is the data or information that is to be recorded?
    - Data is of high importance as it will be kept as a record for insurance in case of theft or damage
8. Are there any algorithms that are going to need to be implemented?
    - The number of a single item there is at a particular location
    - The total number of that item altogether
    - The sum of the values those individual items (value per unit * quantity)
9. When are the algorithms going to be run?
    - These will need to be run when there are new items added/removed to a group of the same item
    - If the value of an item changes

Figure 1.2: Interview Questions (pg 2)

10. What inputs are required for the proposed system?
    - Inputs are likely to be text, numbers and currency
11. What outputs are required for the proposed system?
    - Outputs are likely to be the same as the inputs
    - Notifications of when PAT tests are in need or reissue
    - Notifications when an items location or quantity is changed
    - A print function would be necessary
12. Are hard copies required?
    - Yes, hard copies would be required a visual backup.
13. Are back dated records required?
    - Yes, for insurance purposes
14. How long are these records going to be kept?
    - We will keep back dated records for a year
15. How are these records going to be stored?
    - We will store them electronically on the file server
16. How often will outputs be required?
    - Outputs will be required whenever possible
17. What computing resources do you currently possess to aid the new system's operation?
    - We currently have a Mac Pro that we use as a file server. This is where the database system will be placed.
18. Is security an issue?
    - No, security is not an issue, although the data would need to be backed up.
19. Should there be restricted access to certain areas?
    - No, restricted access is not needed.
20. What errors and exceptions will need to be reported in the new system?
    - I'm not 100% until we start testing the system.
21. How should these errors and exceptions be reported?
    - Errors should be reported to you either via email or another notification

Figure 1.3: Interview Questions (pg 3)

method.

22. Are there any constraints on hardware, software, data, cost or time?
    - No budget, time deadline is flexible and we'll adapt to whatever software/hardware resource available.

### 1.1.5 The current system

#### Data sources and destinations

In the current system, there are multiple data sources. The client and his colleagues as well as members of the AV crew for the church can enter data into the spreadsheet by using a computer in the office and accessing the on the server.

#### Algorithms

In the current system, there are only a few algorithms in place.

**Algorithm 1** Algorithm 1, When new item is bought:

```
1: IF Item = NewItem THEN
2:     SET Action TO EnterNewItem
3: ELSE IF Item = ItemMatch THEN
4:     SET Action TO UpdateItem
5: END IF
```

**Algorithm 2** Algorithm 2, When an item is sold or replaced:

```
1: IF Item = Sold THEN
2:     SET Action TO UpdateQuantity
3: ELSE IF Item = Damaged THEN
4:     SET Action TO UpdateQuantity
5:     SET Action TO FileInsuranceClaim
6: ELSE IF Item = Stolen THEN
7:     SET Action TO FileInsuranceClaim
8: END IF
```

### Data flow diagrams


Figure 1.4: Flow Diagram Key.

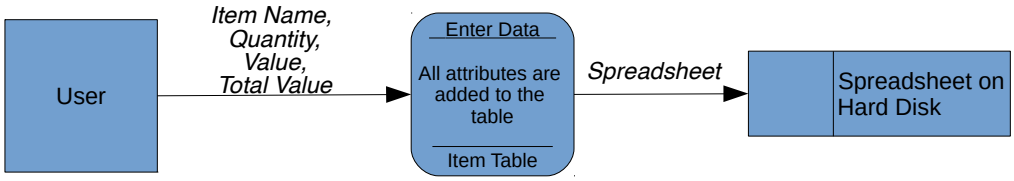



Figure 1.5: Entering a new item.

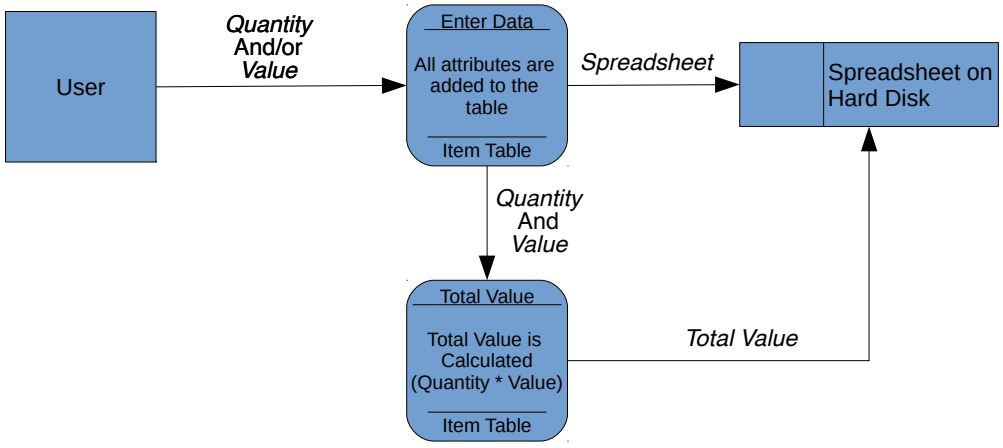



Figure 1.6: Updating an item that already exists in the table.

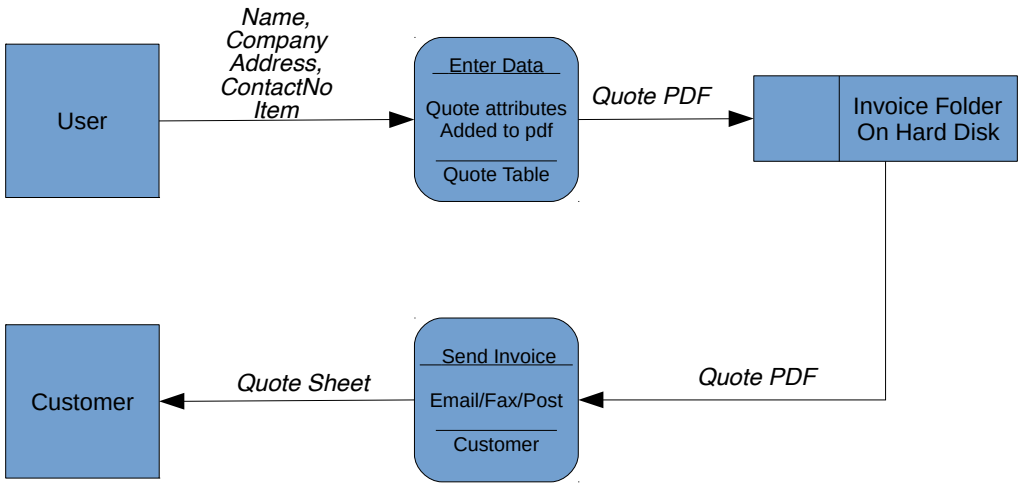



Figure 1.7: Creating and sending the initial quote for a loan.

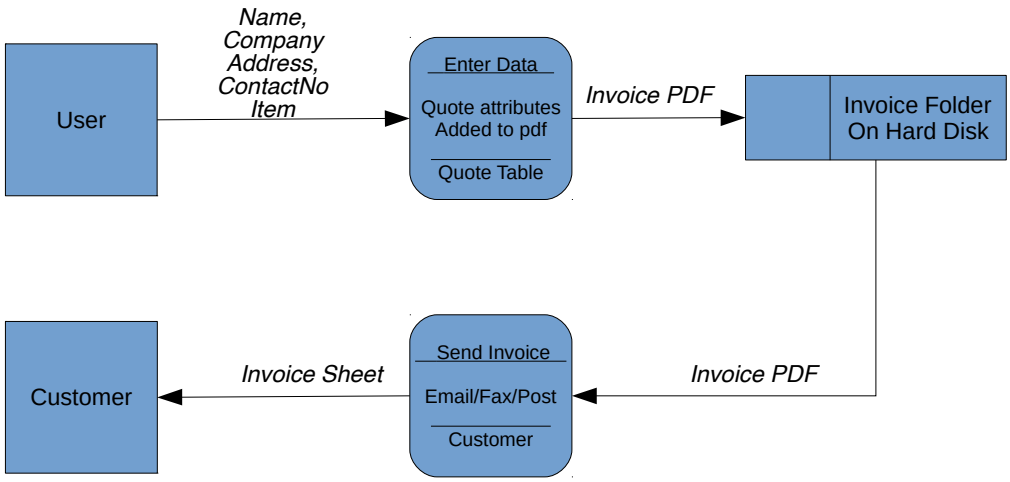



Figure 1.8: Creating and sending the final invoice for a loan.

### Input Forms, Output Forms, Report Formats

Josh has provided me with a screenshot of him entering some data into his current system. I have boxed out confidential information such as item values and their respective sub-total values:

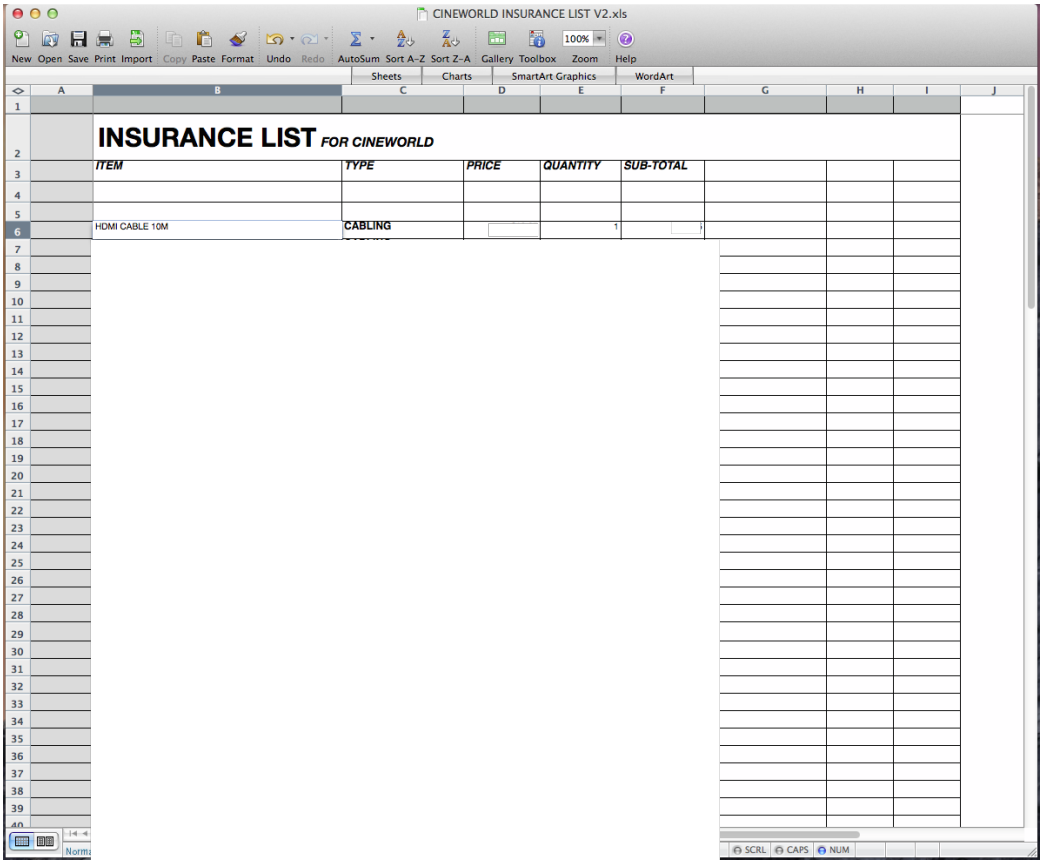



Figure 1.9: Josh Entering Item Name.

Here is an screen shot showing the calculation used to get the Sub-Total Value:

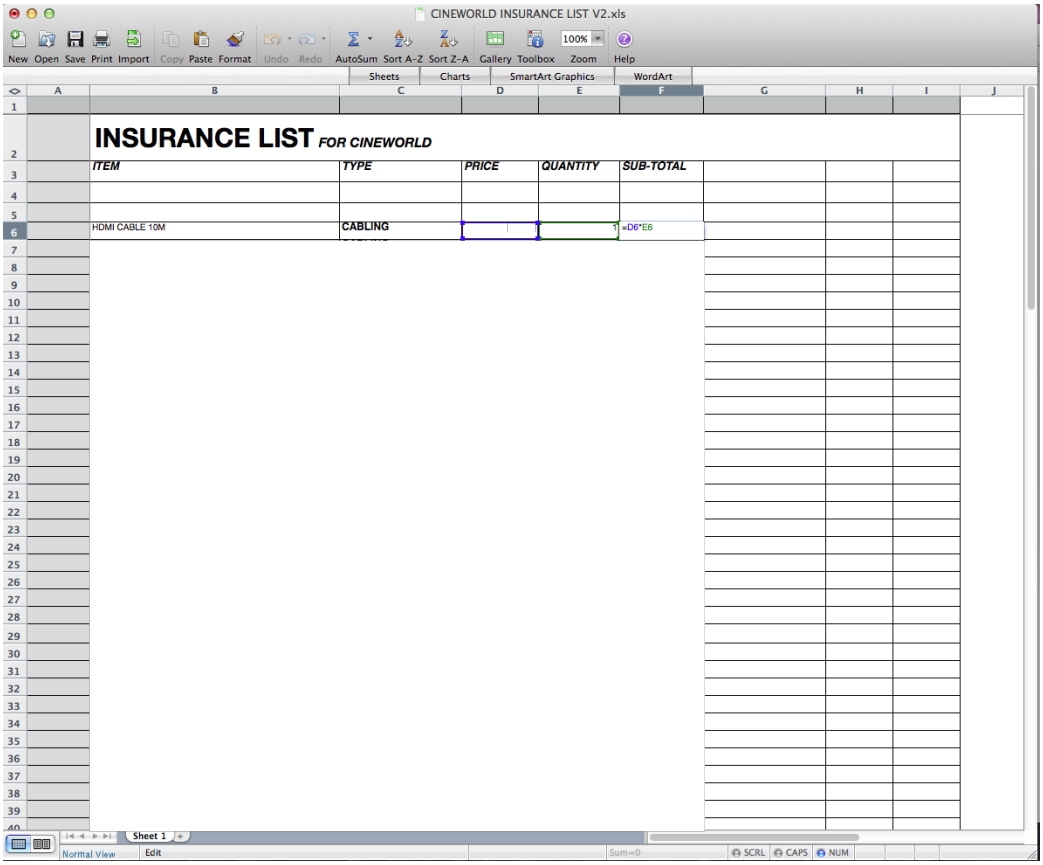



Figure 1.10: Sub-Total Calculation.

### 1.1.6 The proposed system

#### Data sources and destinations

The Following table shows the proposed data and their respective sources and destinations.

| **Source** | **Data** | **Data Type** | **Destination** |
|---|---|---|---|
| Generated | **ItemTypeID** | Integer | Database - ItemType Table |
| User | ItemType | Text | Database - ItemType Table |
| - | - | - | - |
| Generated | **LocationID** | Integer | Database - Location Table |
| User | Location | Text | Database - Location Table |
| - | - | - | - |
| Generated | **ItemID** | Integer | Database - Item Records |
| Database - ItemType Table | *ItemTypeID* | Integer | Database - Item Table |
| Database - Location Table | *LocationID* | Integer | Database - Item Table |
| User | ItemName | Text | Database - Item Table |
| User | Value | Real | Database - Item Table |
| User | ItemQuantity | Integer | Database - Item Table |
| User | SubTotal | Real | Database - Item Table |
| User | OnLoan | Boolean | Database - Item Table |

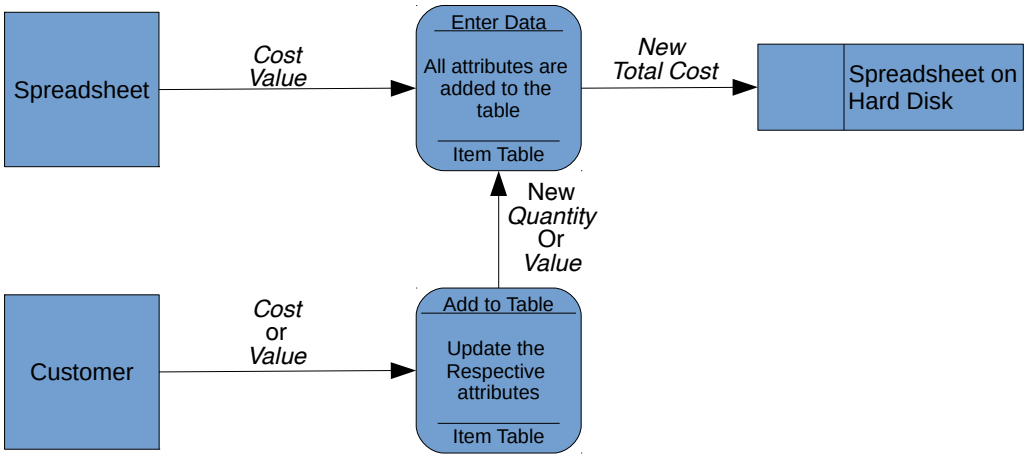
| **Source** | **Data** | **Data Type** | **Destination** |
|---|---|---|---|
| Generated | **LoanListingID** | Integer | Database - LoanListing Table |
| Database - Item Table | *ItemID* | Integer | Database - LoanListing Table |
| User | LoanQuantity | Integer | Database - LoanListing Table |
| - | - | - | - |
| Generated | **CustomerLoanID** | Integer | Database - Loan Table |
| Database - Customer Table | *CustomerID* | Integer | Database - Loan Table |
| User | LoanRate | Real | Database - Loan Table |
| User | LoanLength(Days) | Integer | Database - Loan Table |
| Calculated | LoanCost | Real | Database - Loan Table |
| - | - | - | - |
| Generated | **CustomerID** | Integer | Database - Customer Table |
| User | Forename | Text | Database - Customer Table |
| User | Lastname | Text | Database - Customer Table |
| User | Company | Text | Database - Customer Table |
| User | Street | Text | Database - Customer Table |
| User | Town | Text | Database - Customer Table |
| User | County | Text | Database - Customer Table |
| User | PostCode | Text | Database - Customer Table |
| User | MobileNumber | Text | Database - Customer Table |
| User | LandLine | Text | Database - Customer Table |
| User | Email | Text | Database - Customer Table |

| **Source** | **Data** | **Data Type** | **Destination** |
|---|---|---|---|
| Generated | **ItemTestID** | Integer | Database - ItemTest Table |
| Database - PATtest Records | *PATtestID* | Integer | Database - ItemTest Table |
| User | ItemDescription | Text | Database - ItemTest Table |
| User | ItemClass | Integer | Database - ItemTest Table |
| User | FuseRating | Text | Database - ItemTest Table |
| User | TestUsed | Text | Database - ItemTest Table |
| User | ProtectiveCondTest | Integer | Database - ItemTest Table |
| User | InsulationTest | Text | Database - ItemTest Table |
| User | Leakage | Float | Database - ItemTest Table |
| User | TestResult | Boolean | Database - ItemTest Table |
| - | - | - | - |
| Generated | **PATtestID** | Integer | Database - PATtest Table |
| User | TestDate | Date | Database - PATtest Table |

### Data flow diagram


Figure 1.11: Flow Diagram Key.

Figure 1.12: Enter New Item.



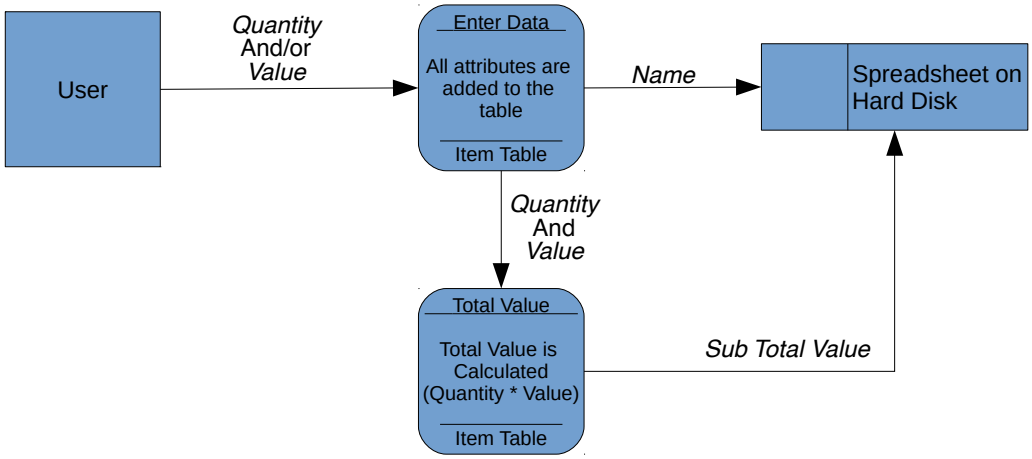



Figure 1.13: Enter New Item.

### Data dictionary

**Data dictionary**

| **Name** | **Data Type** | **Length** | **Validation** | **Example Data** | **Comment** |
|---|---|---|---|---|---|
| ItemTypeID | Integer | 1-435 | Range | 253 | This is the **Primary Key** for the ItemType class, and foreign key for the Item class |
| ItemType | Text | 5-40 Characters | Length | Arkaos Server | This holds the description of each type of Item. |
| LocationID | Integer | 1-3 Figures | Range | 1,300 | This is the **Primary Key** for the Location class and a *Foreign Key* for the Item class |
| Location | Text | 1-30 Characters | Length | Main Offices | This holds the name of the locations |

| **Name** | **Data Type** | **Length** | **Validation** | **Example Data** | **Comment** |
|---|---|---|---|---|---|
| ItemID | Integer | 1-435 | Range | 253 | This is the **Primary Key** for the Item class, and foreign key for the Loan and PATtest classes |
| ItemName | Text | 5-40 Characters | Length | Arkaos Server | This gives the name of each item entered |
| Value | Real | 2-5 Figures | Range | 1,300 | This holds the data for the monetary value for each item |
| ItemQuantity | Integer | 0-100 | Range | 35 | This holds the data for the number of each item owned |
| SubTotal | Real | 2-8 Figures | Range | 250 | This is calculated for each item by multiplying the value by the quantity |
| OnLoan | Boolean | True/False | Status Check | True | This holds the data of whether an item is on loan or not. Will be displayed as "Yes" or "No" |

| **Name** | **Data Type** | **Length** | **Validation** | **Example Data** | **Comment** |
|---|---|---|---|---|---|
| LoanListingID | Integer | 1-435 | Range | 56 | This is the **Primary Key** for the LoanListing class |
| ListingQuantity | Integer | 1-35 | Range | 4 | This holds the data for how many of an item has been loaned out |
| CustomerLoanID | Integer | 1-435 | Range | 21 | This is the **Primary Key** for the Loan class |
| LoanRate | Real | 1-5 Figures | Range | 75 | Holds data for how much is charged per day for the loan of an item |
| LoanLength | Integer | 1-3 Figures | Range | 7 | Holds the data for the length of the loan |
| LoanCost | Real | 1-4 Integers | Range | 250 | Holds the data for the amount to charge before the loan |

| **Name** | **Data Type** | **Length** | **Validation** | **Example Data** | **Comment** |
|---|---|---|---|---|---|
| CustomerID | Integer | 1-255 | Range | 52 | This is the **Primary Key** for the Customer class |
| Forename | Text | 3-20 Characters | Length | John | A field for the customers forename |
| Lastname | Text | 3-20 Characters | Length | Smith | A field for the customers surname |
| Company | Text | 3-20 Characters | Length | Digital Lighting Cambs | A field for the company's name |
| Street | Text | 3-30 Characters | Length | 129 Cedar Crescent | A field for the company's Street address |
| Town | Text | 3-30 Characters | Length | Sawston | A field for the company's Town |
| County | Text | 3-20 Characters | Length | Cambs | A field for the company's County |
| PostCode | Text | 6-7 Characters | Format | CB22 7RX | A field for the company's Postcode |
| MobileNumber | Text | 11 Characters | Format | 07891234567 | A field for the customers mobile number |
| LandLine | Text | 11 Characters | Format | 01234567890 | A field for the customers landline phone |
| Email | Text | 7 - 30 Characters | Length | john.smith@example.com | A field for the customers email address |

| **Name** | **Data Type** | **Length** | **Validation** | **Example Data** | **Comment** |
|---|---|---|---|---|---|
| ItemTestID | Integer | 1-255 | Range | 52 | This is the **Primary Key** for the ItemTest class |
| ItemDescription | Text | 3-400 Characters | Length | Waltham portable TV | A field that describes the item to be tested |
| ItemClass | Integer | 1 Character | Length | 2 | A field to show what class of electrical equipment the item is |
| FuseRating | Text | 1-3 Characters | Length | 5A | A field which displays the fuse rating |
| TestUsed | Text | 1-10 Characters | Length | II | A field to show what test was used on the item |
| ProtectiveCondTest | Float | 4 Characters | Length | - | A field displaying the resistance of an item, in Ohms, to a 200mA current |
| InsulationTest | Text | 3 Characters | Length | ¿20 | A field displaying the Insulation of an item, in Ohms, to a 250V or 500V Potential Difference |
| Leakage | Float | 4 Characters | Format | 0.03 | A field that shows the current not obtained by the item, in milliamperes |
| TestResult | Boolean | - | Presence Check | True | A field to show if an item Passed or not |

| **Name** | **Data Type** | **Length** | **Validation** | **Example Data** | **Comment** |
|---|---|---|---|---|---|
| PATtestID | Integer | 1-255 | Range | 52 | This is the **Primary Key** for the PATtest class |
| TestDate | Date | 10 Characters | Format | 01/12/2014 | A field that displays the date of the PAT test |

**Volumetrics**

I have chosen to start off with only 20 Item Records along with 20 Loan Records and 20 PAT Test Records. In total there will be 60 Records. I have chosen this number of records as my Client and I had previously agreed that this would be a suitable number of records to start with in order for him to get used to the system and train up other colleagues to know how to use it also. This can be increased as time goes by.

The Item Records Database, Loan Records Database and the PAT Test Records Database will store 18 fields of combined data. Each field should take up 1KB of hard disk space. With this the required initial storage space will be:

18KB * 60 = 1080KB

1080KB / 1024 = 1.05MB

If the rest of database managemnent system took up 28MB, the client would need 19.05MB of space for 60 records, with 18 fields of data

## 1.2 Objectives

### 1.2.1 General Objectives

- Easily understandable layout and structure for records.
- Data is easy to enter and edit
- Viewing of records is structured and well presented

### 1.2.2 Specific Objectives

Record viewing:

- Clear labels for data attributes.
- Next and Previous record buttons.
- Edit button so data cannot be changed accidentally.
- Submit button to save data changes (if any) to the current record.

- First and Last record buttons to jump to respective record.

Data input:

- Data fields become editable
- Drop down selection for location selection
- Changes saved immediately after editing has finished (i.e. submit button pressed)

Data output:

- Print button and functionality
- Export records to PDF
- Print/Export a batch of records to PDF
- Email notifications when new item is entered into database or an item is updated, the details and who entered/updated.

### 1.2.3 Core Objectives

- Viewing of Item/Loan/PAT-test Records
- Item/Loan/PAT-test data input
- Item/Loan/PAT-test data editing
- Sending of Loan Invoices

### 1.2.4 Other Objectives

- Generating and exporting of quote sheets to PDF
- Generating and exporting of invoices to PDF
- Printing and Exporting records to PDF
- Enable Full screen application on OS X

## 1.3 ER Diagrams and Descriptions

### 1.3.1 ER Diagram

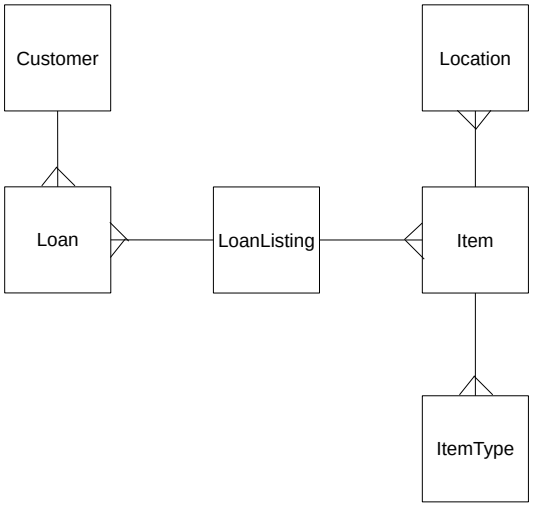



Figure 1.14: Loan Item ER Diagrams.

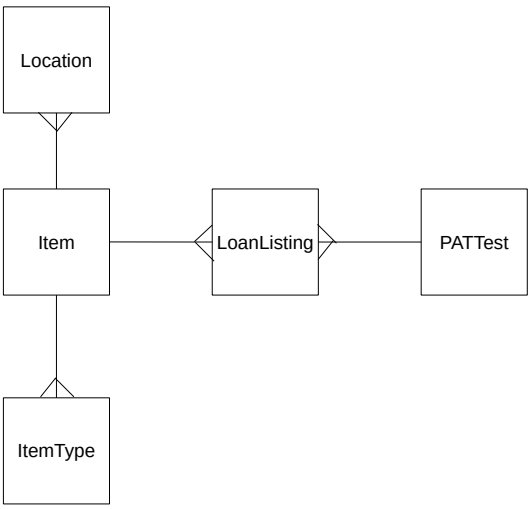



Figure 1.15: PAT Test ER Diagrams.

### 1.3.2 Entity Descriptions

**ItemType**(ItemTypeID, ItemType)

**Location**(LocationID, Location)

**Item**(ItemID, *ItemTypeID*, *LocationID*, Name, Location, Value, ItemQuantity, SubTotal, OnLoan,)

**LoanListing**(LoanListingID, *ItemID*, ListingQuantity)

**Loan**(LoanID, *CustomerID*, LoanRate, LoanLength, LoanCost)

**Customer**(CustomerID, Forename, Lastname, Company, Street, Town, County, PostCode, MobileNumber, LandLine, Email)

**PATtest**(PATtestID, TestDate)

**ItemTest**(ItemTestID, *PATTestID*, ItemDescription, ItemClass, FuseRating, TestUsed, ProtectiveCondTest, InsulationTest, Leakage, TestResult)

## 1.4 Object Analysis

### 1.4.1 Object Listing

- Client
- Item
- Location

### 1.4.2 Relationship diagrams

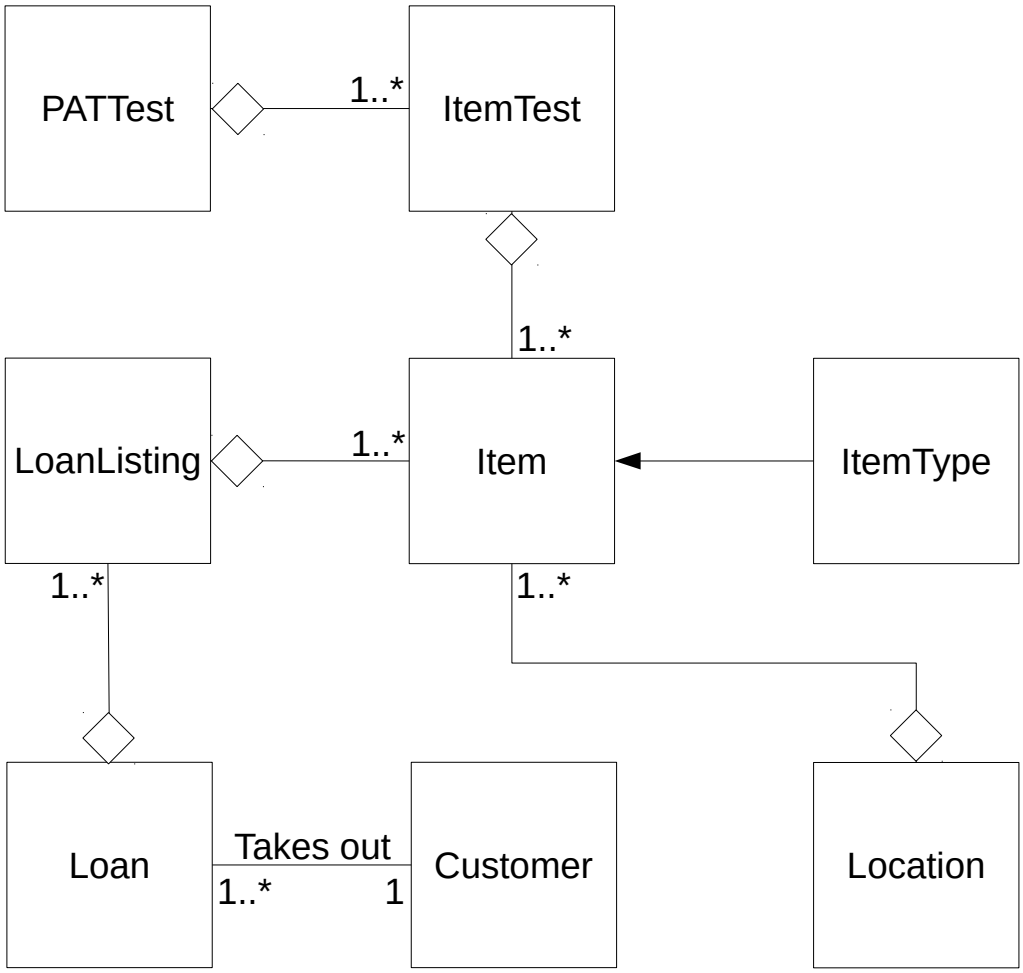



Figure 1.16: Relatioship Diagram.

### 1.4.3 Class definitions

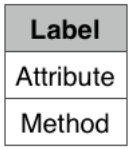



Figure 1.17: Class Diagram Key.

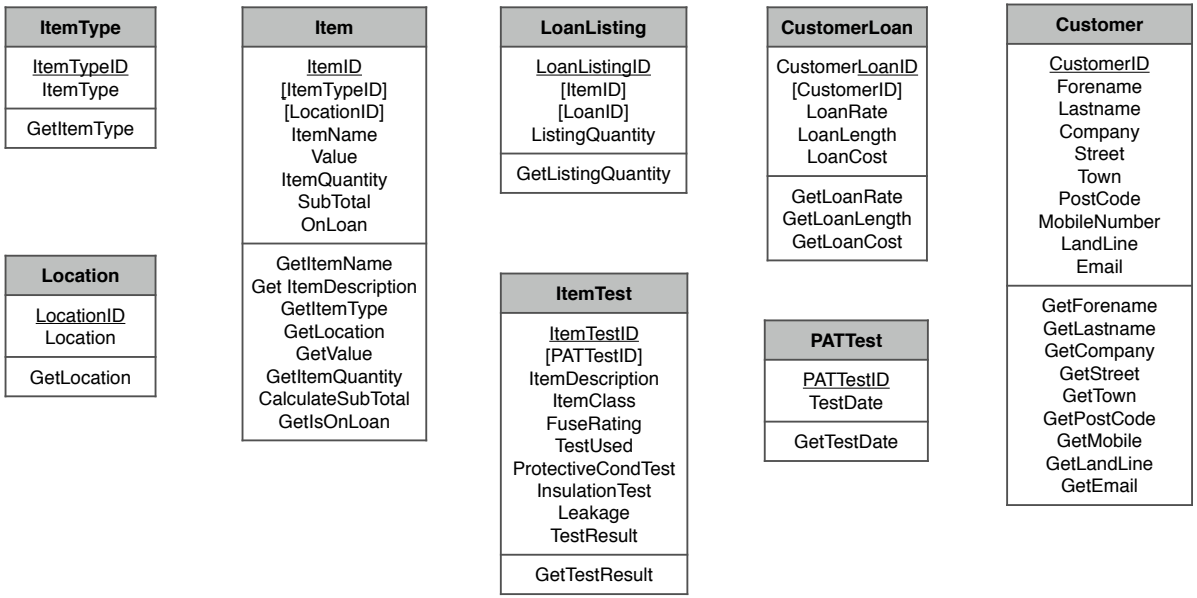



Figure 1.18: Class Diagrams.

## 1.5 Other Abstractions and Graphs

## 1.6 Constraints

### 1.6.1 Hardware

Presently, Josh uses a custom built, 2008 MacPro Desktop Computer. This is primarily used as a file server for images, audio and video files a well as a backup for his current work desktop. My system will need to be compatible with this system.

Computer Specifications:

- 2x 2.8 GHz Quad-Core Intel®Xeon™Processor
- ATI Radeon HD 2600 XT 256MB Graphics Card
- 661-4449 Apple Mac Pro A1186 Motherboard
- 16.00GB DDR3 RAM
- 1TB SATA Disk-Drive
- 6TB RAID Storage
- Apple SuperDrive
- 15" LG E1942 LCD Display. 1280 x 720 pixels

The proposed system should have little to no impact on this machine as the processing power and memory that can be dissipated by the computer, greatly exceeds the requirements for the proposed system.

One other constraint of the computer to be used is that it is a desktop computer. This means that the system is only accessible where Josh chooses to have the computer based in his place of work, as the computer is not portable. In addition to this, the computer requires a constant supply of power in order to operate as there is not internal battery.

One other constraint of the computer to be used is that it is a desktop computer. This means that the system is only accessible where Josh chooses to have the computer based in his place of work, as the computer is not

portable. In addition to this, the computer requires a constant supply of power in order to opperate as there is not internal battery.

### 1.6.2 Software

Josh has told me that he is able to adapt to the software that is required to run the system. The current operating system in place is Apples OSX 10.8 (Mountain Lion). Josh wishes to update the software sometime in the near future to OSX 10.9 (Mavericks) and possibly update to OSX 10.10 (Yosemite). This could prove to be constraint because OSX 10.10 (Yosemite) isn't yet fully supported by some applications.

### 1.6.3 Time

Josh has said that there is no deadline requirement for the proposed system to be in place and doesn't need it until I have finished implementing it. The only deadline I need to meet is the project deadline set by my Computing course leader. This is Friday 13 $^{th}$ February 2014.

### 1.6.4 User Knowledge

Josh posses a qualification in A level Media studies as well as 2 years use of computers during his degree. He has substantial understanding of how to use computers as his job requires he uses one most of the time. Josh also has required knowledge of how to use many varieties of applications. He uses Adobe Creative Suite for most of his job as he designs various forms of media. He also has knowledge of Apple's Final Cut Pro application as well as many others.

When designing and implementing the proposed system, Josh's experience with computers will have to be considered. Josh tends to use the internet browser Google Chrome for all his web-browsing and research as well as a third party mail application called. By designing the system similarly to these applications, it should make it easier to understand how the system works and get used to using it a lot faster than it would if the system had a

primitive design.

There will also be a full manual included to aid Josh with learning and understanding the familiar interface, the functionality of the new system and how to use certain features.

### 1.6.5 Access restrictions

The proposed system is primarily to be accessed by Josh himself. However, he can see it being an advantage if other people had access to the system.

For this reason, we have agreed that having the database password protected is the best way for Josh to control who can access the data. He will be able to distribute the passwords to other colleagues who he feels should have access to the database management system. This reduces the risk of records being changed or deleted by people who shouldn't need to use the system.

## 1.7 Limitations

### 1.7.1 Areas which will not be included in computerisation

Initial buying of new items will not be included in the computerisation as this is still done either in person or over the world wide web. Similarly, initial sales of items will not be included in the computerisation, it will only be once the item has been bought/sold that the data will be updated to coincide with the quantity changes and/or addition to or deduction or equipment.

### 1.7.2 Areas considered for future computerisation

When a customer loans out equipment, Josh sends out an initial quote, either as an email format or on paper. This could be included in the system by selecting the items the customer wants to high out, and draft a quote form for Josh. Similarly, Josh sends out an emailed invoice to the client, he does this manually by hand. It would be advantageous to include this into the system, by generating an invoice based on the attributes in Loan Records

and export it as a PDF for email or printing. These could be implemented in addition to the current database design at the end, if I have enough time to learn and understand how to enter this functionality it into the system

## 1.8 Solutions

### 1.8.1 Alternative solutions

| **Alternative solution** | **Advantages** | **Disadvantages** |
|---|---|---|
| Custom made database | • No need to install additional software, only a simple database management system such as "Microsoft Access" or "Filemaker". | • Database management systems often cost a substantial amount of money for a license. |
| Web based application | • Easily accessible by other users. Doesn't rely on one machine.<br>• Can have 'Cloud based' storage of files.<br>• More than one user can be logged on at a time. | • Website or server hosting can be expensive.<br>• More advanced security methods will be required due to the system being constantly online and therefore vulnerable to attack.<br>• Better networking knowledge required to compensate for the security implications and risks. |

| **Alternative solution** | **Advantages** | **Disadvantages** |
|---|---|---|
| Terminal or Command based application | • More power efficient as it isn't graphics heavy, much easier to design as the interface is just text.<br>• Fast efficient operation provided the client has knowledge of terminal and shell commands. | • Careful error handling needed as the user could enter any known/valid command.<br>• Training is required so that the client knows what commands to use when.<br>• There are often commands that the client don't know about that could potentially corrupt his computer. |
| Python desktop application with a GUI | • Designed and layout can be client specific.<br>• Minimal error with radio buttons and other widgets.<br>• Easy to understand layout as data can be formatted to fit the clients requirements.<br>• Easy to visualise what is happening with graphs and tables. | • More time needed to build the interface and sql database compared to a command based application.<br>• More resources needed from the computer for graphical visualisation and database storage<br>• Programming the graphical interface could prove a difficult task. |

### 1.8.2 Justification of chosen solution

I have chosen to use the *'Python Desktop Application with a GUI'* solution.

These are my reasons:

- The application takes up no physical space apart from the computer it is installed on.
- I already have the required language knowledge needed to program a

database and a GUI in Python

- Using a custom made desktop application is faster for Josh to manage his inventory than the current spreadsheet based system.
- Backup can be made and data can be restored easily in the event of corruption or unresolvable data loss

# Chapter 2

# Design

## 2.1 Overall System Design

### 2.1.1 Short description of the main parts of the system

- Media Inventory Database
  - General Interface
  - Adding Records
  - Displaying Records
  - Searching Records
  - Editing Records
  - Deleting Records

General Interface

- The user will be presented with a box whereby he/she will enter a password. This password will be the same for all users who have access to the system.
- Once logged in, the user will be confronted with an interface consisting of a series of menu options. These options will be "Add Record", "Display Records", "Search Records", "Edit Record", "Delete Record" and "Change Password".

- When the "Change Password" button has been clicked, the user will be taking to a box where they will be required to enter the previous password, then enter a new password twice.
- Clicking on the "Add Record" button will take the user to an interface where they will be required to select the type of record they wish to enter.
  - Clicking the "Add Loan" button will present an interface to the user where they will have a choice of selecting an existing customer specific loan or creating a new customer specific loan.
  - Selecting the "Add PAT Test" button will present the user with an interface to choose a PAT test date or to create a new PAT test date.
- Clicking on the "Display Records" button will send the user to an interface where they will have to select the table from which table they wish to see the records.
- Clicking on the "Edit Records" button will send the user to an interface where they will have to select the table from which they want to edit a record.
- Clicking on the "Delete Records" button will send the user to an interface where they will have to select the table from which they wish to delete a record.

Adding Information

- The system will present the user with a drop down menu from which the user will have to choose an option for which to enter information. After selecting the option, the user will then be presented with a group of data to add to the new record. If any of these options require the user to enter data relating to another table within the database, they will be presented with a drop down menu and will be required to select an option before they record can be created.
- Once all the required data fields have been complete, the system will add a unique identifier to the record of information and save in to the database

Displaying Records

- The system will present the user with an interface with a drop down menu, where they will have to select the database table from which they wan to view the data.
- Once the table has been selected, the user will then be presented with a view table that will display all the records within that database table. They can then choose to sort this information into ascending or descending order by selecting any row for which to sort it by.

Editing Records

- The system will bring up a user interface that will present a drop down menu where the user will have to select a database table from which they wish to edit a record.
- Once a table has been selected, the user will then be confronted with a user interface which will display all the records within that table and then prompt the user to select the record they would like to edit, by enter the unique identifier of this record.
- When the record has been selected, the user will be presented with an interface similar to the one where the user enters a new record, but the fields already contain the information. The user will then have to update which field of information to update.
- Once data has been updated and a "Done" button has been clicked, the user will then be asked to confirm the updates.
- When the updates have been confirmed, the system replace the old record with the new updated record.

Deleting Records

- The system will present the user with an interface containing a drop down menu where they will have to select a database table from which they wish to delete a record.
- After the database table has been selected, the user will be presented with a view table showing all the records within the database table. Underneath the view table will be a prompt, asking the user for the unique identifier of the record they wish to delete.

- When the user has selected the record they wish to delete, they will have to confirm this by entering the system password.
- The system will then remove the record from the database permanently.

### 2.1.2 System flowcharts showing an overview of the complete system

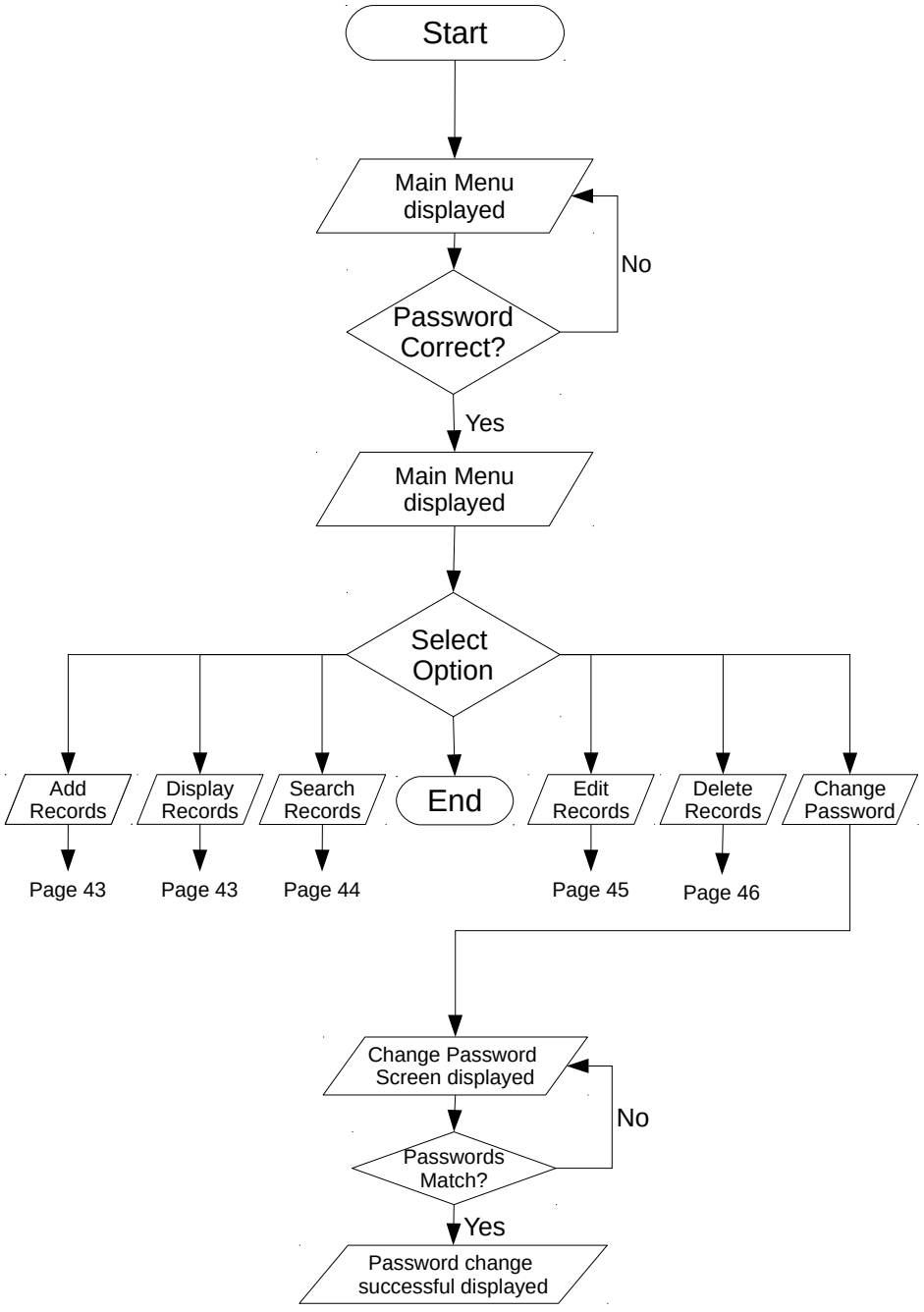



Figure 2.1: Main System Flowchart.

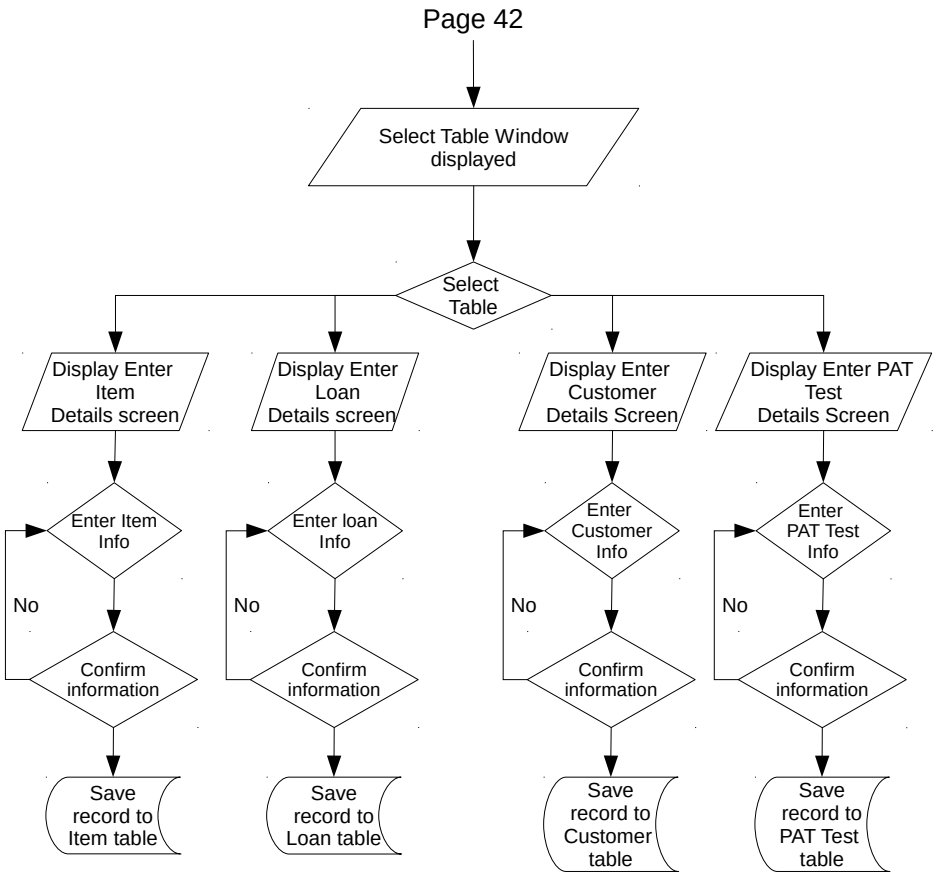



Figure 2.2: Add Records Flowchart.

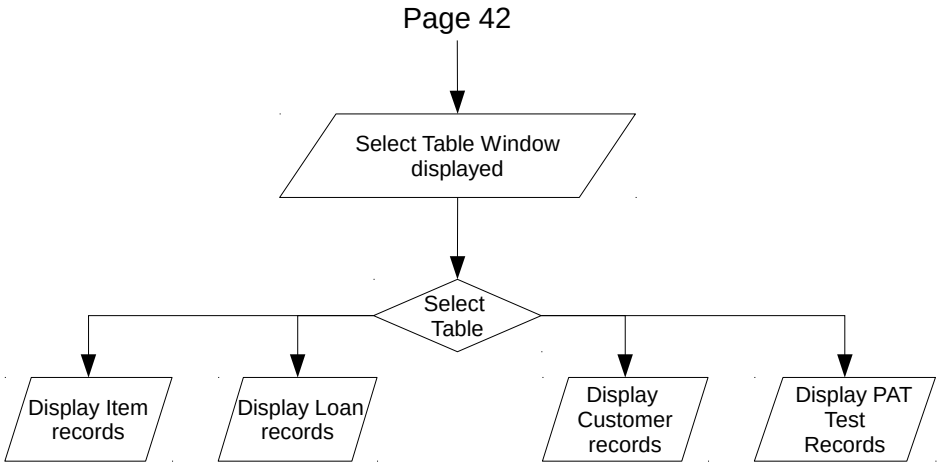



Figure 2.3: Display Records Flowchart.

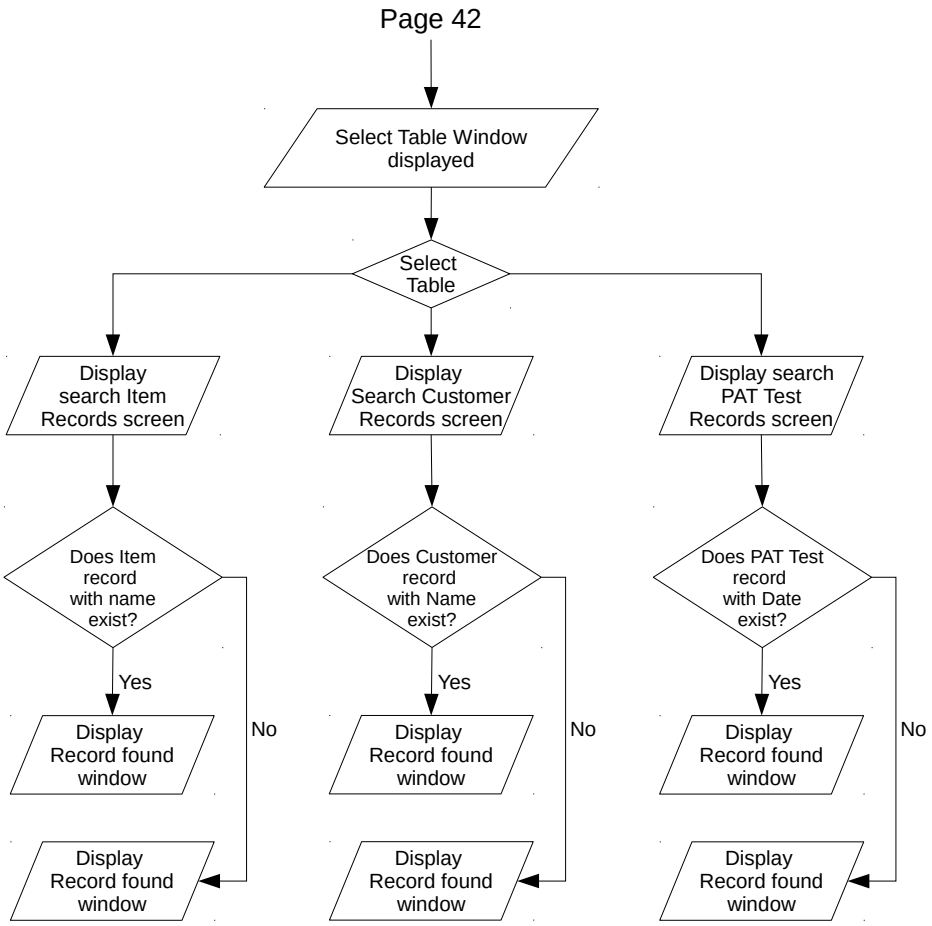



Figure 2.4: Search Records Flowchart.

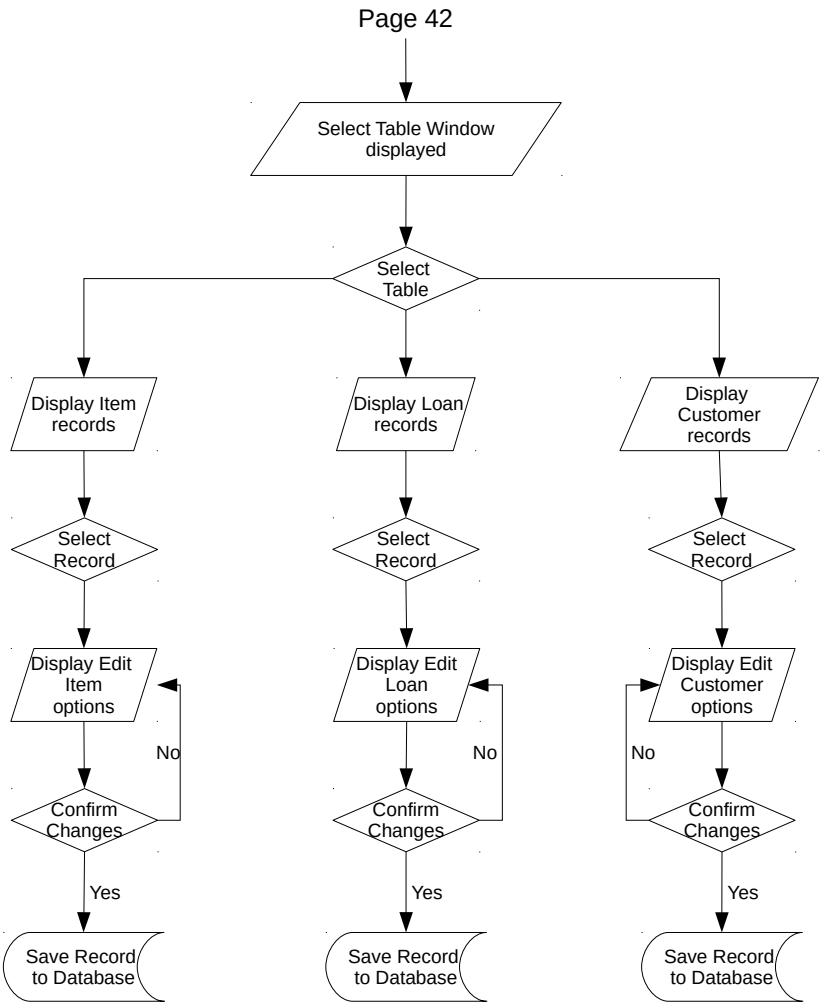



Figure 2.5: Edit Records Flowchart.

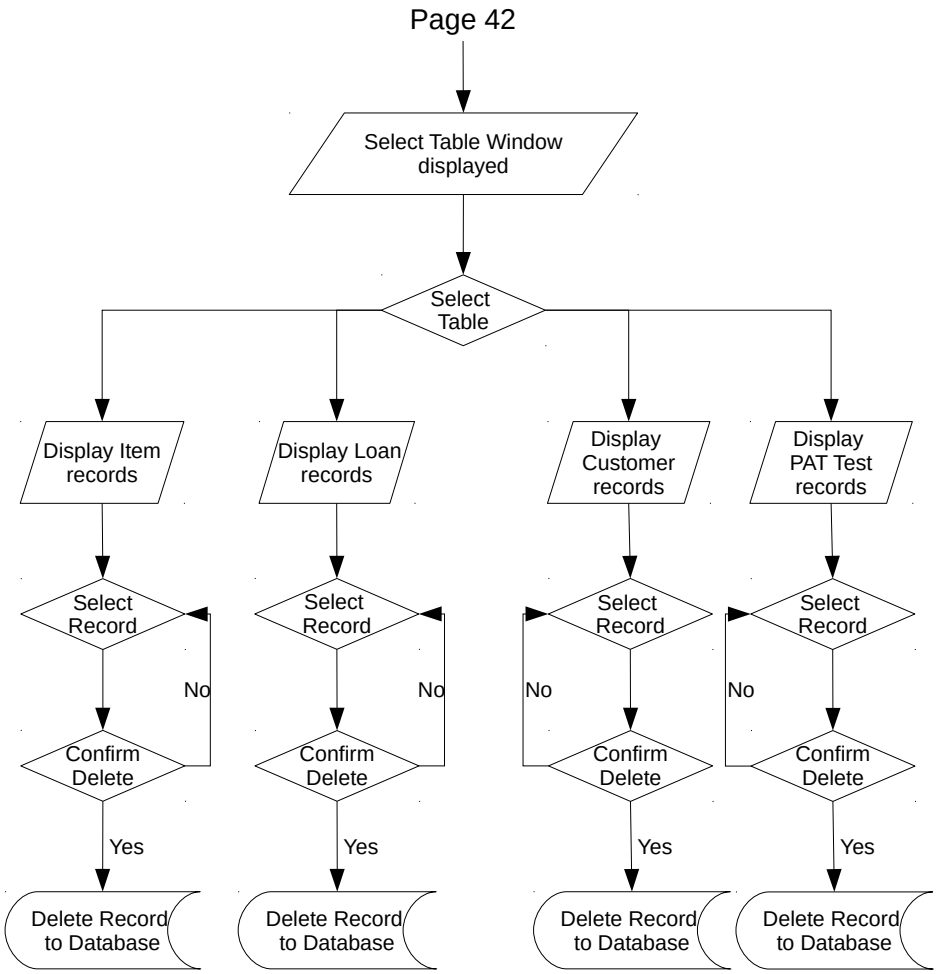



Figure 2.6: Delete Records Flowchart.

## 2.2 User Interface Designs

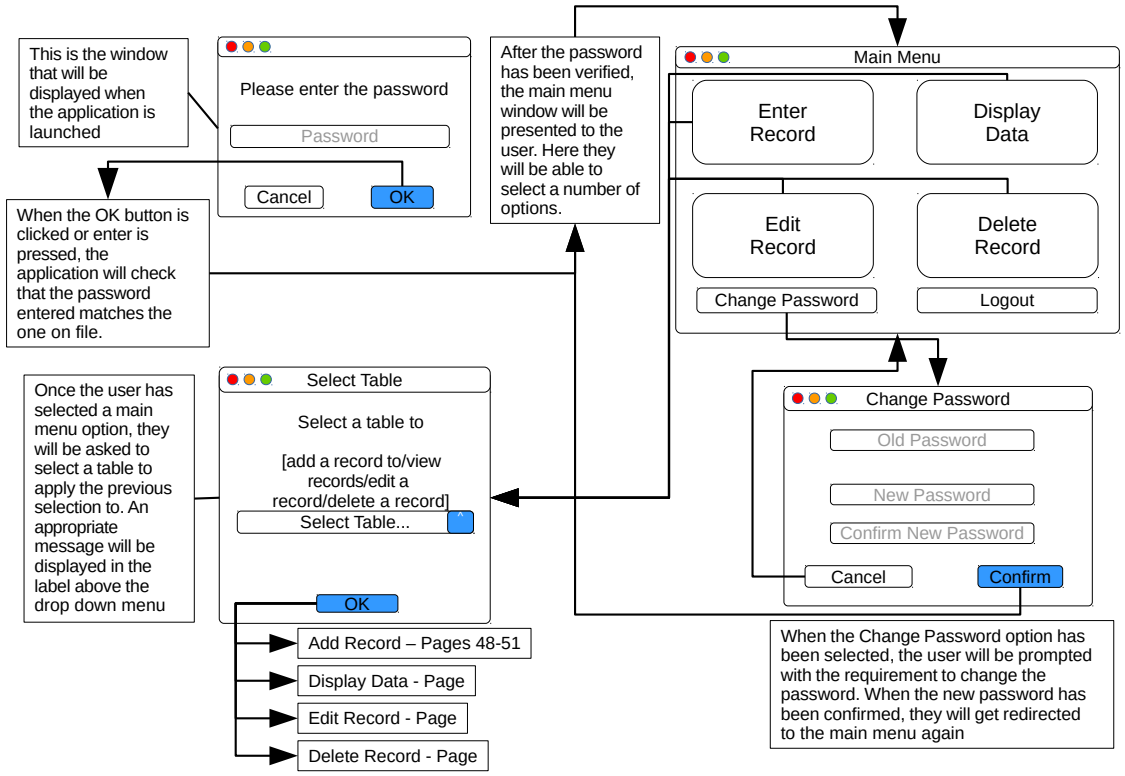



Figure 2.7: Login and Main Menu windows.

Clicking the ”Logout” button will return you to the login screen.

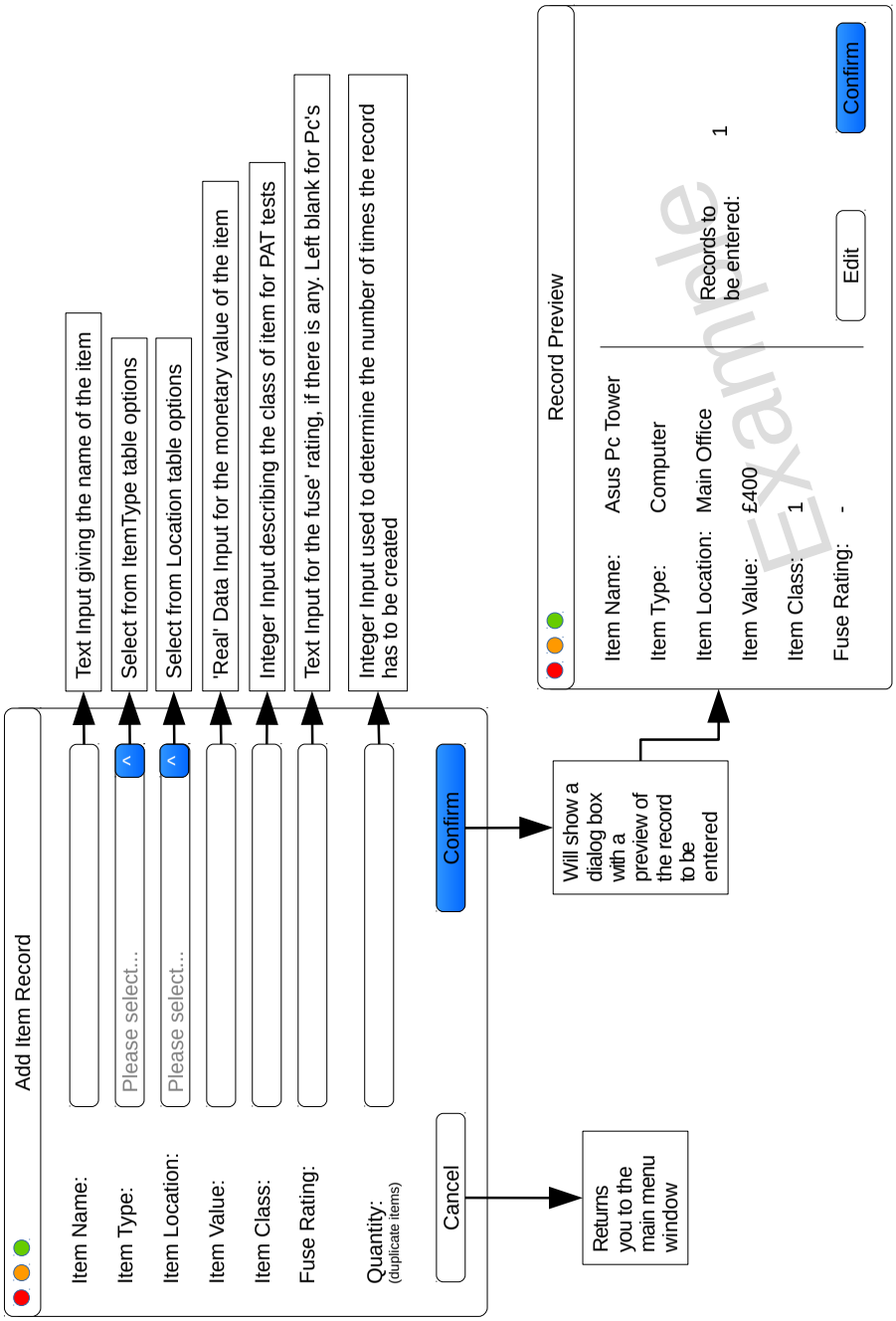



Figure 2.8: Login and Main Menu windows.

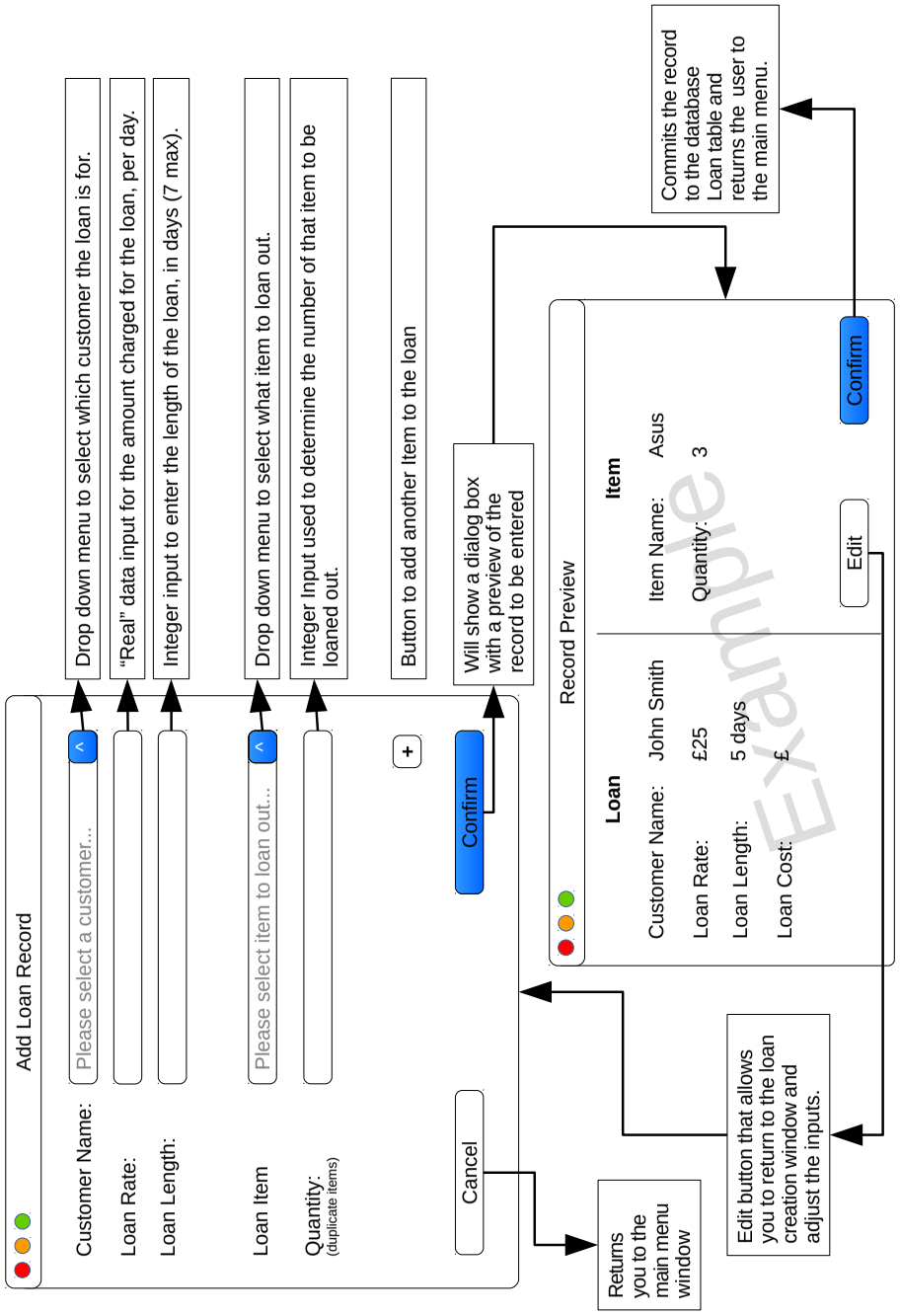



Figure 2.9: Login and Main Menu windows.

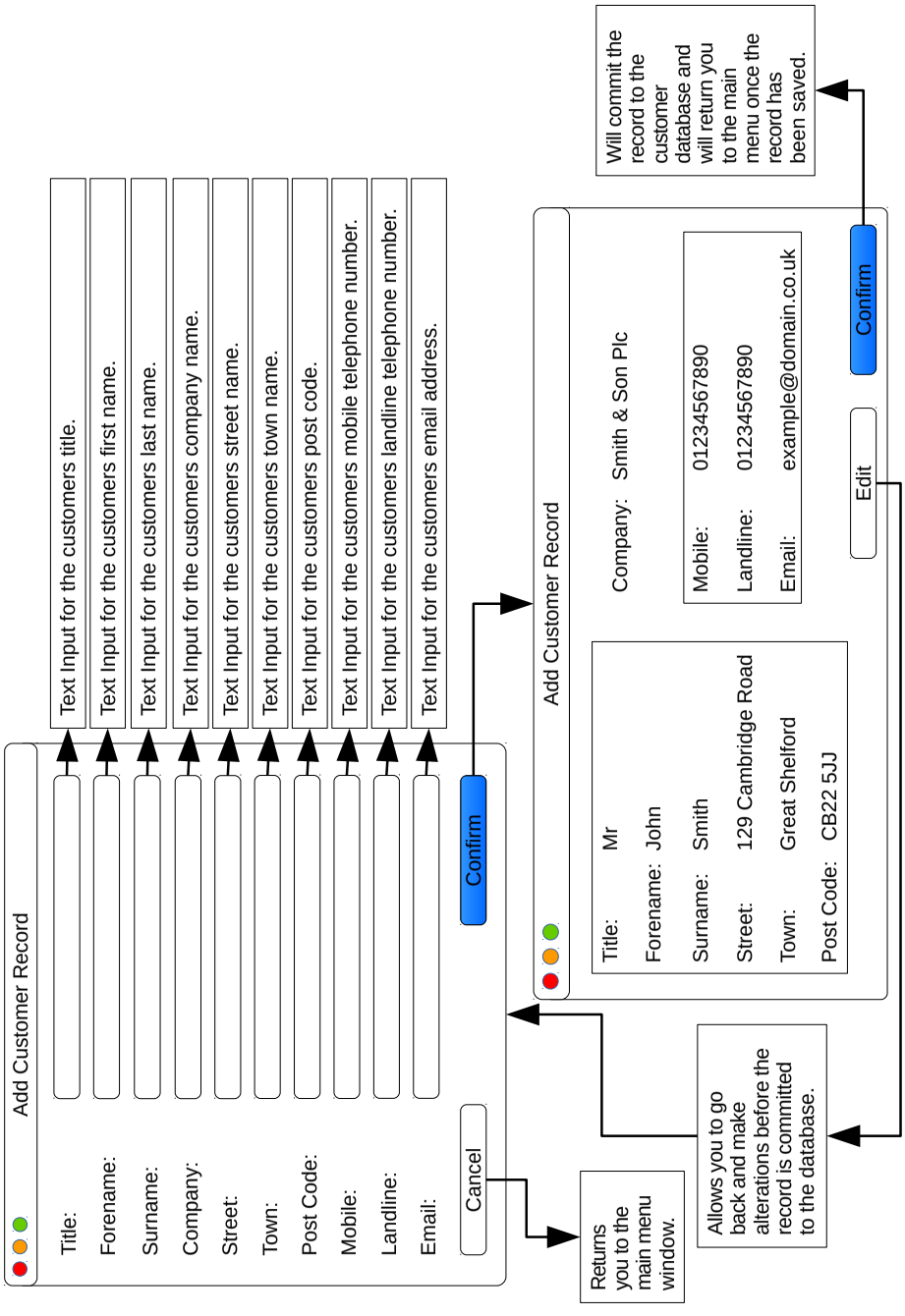



Figure 2.10: Login and Main Menu windows.

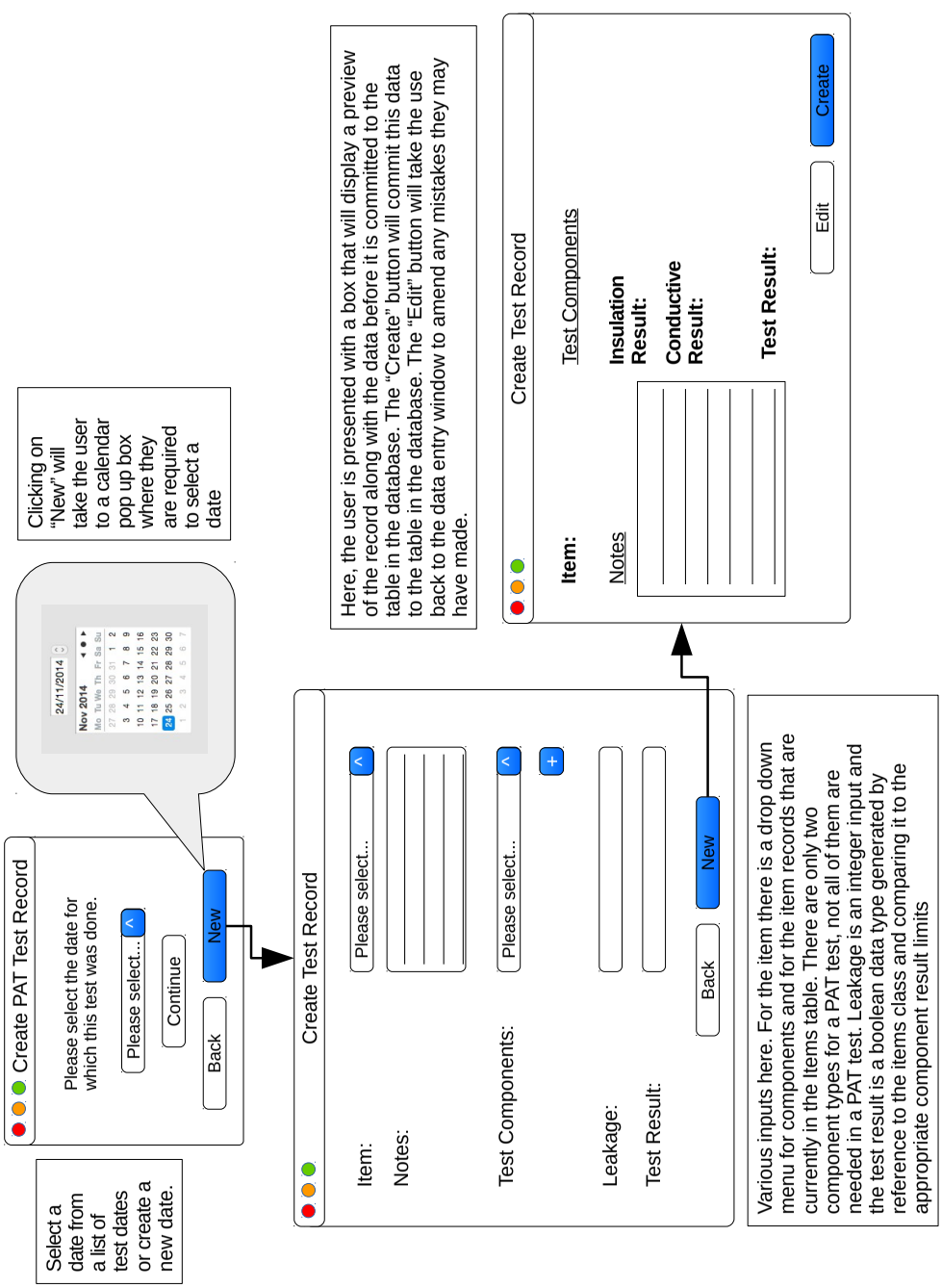



Figure 2.11: Login and Main Menu windows.

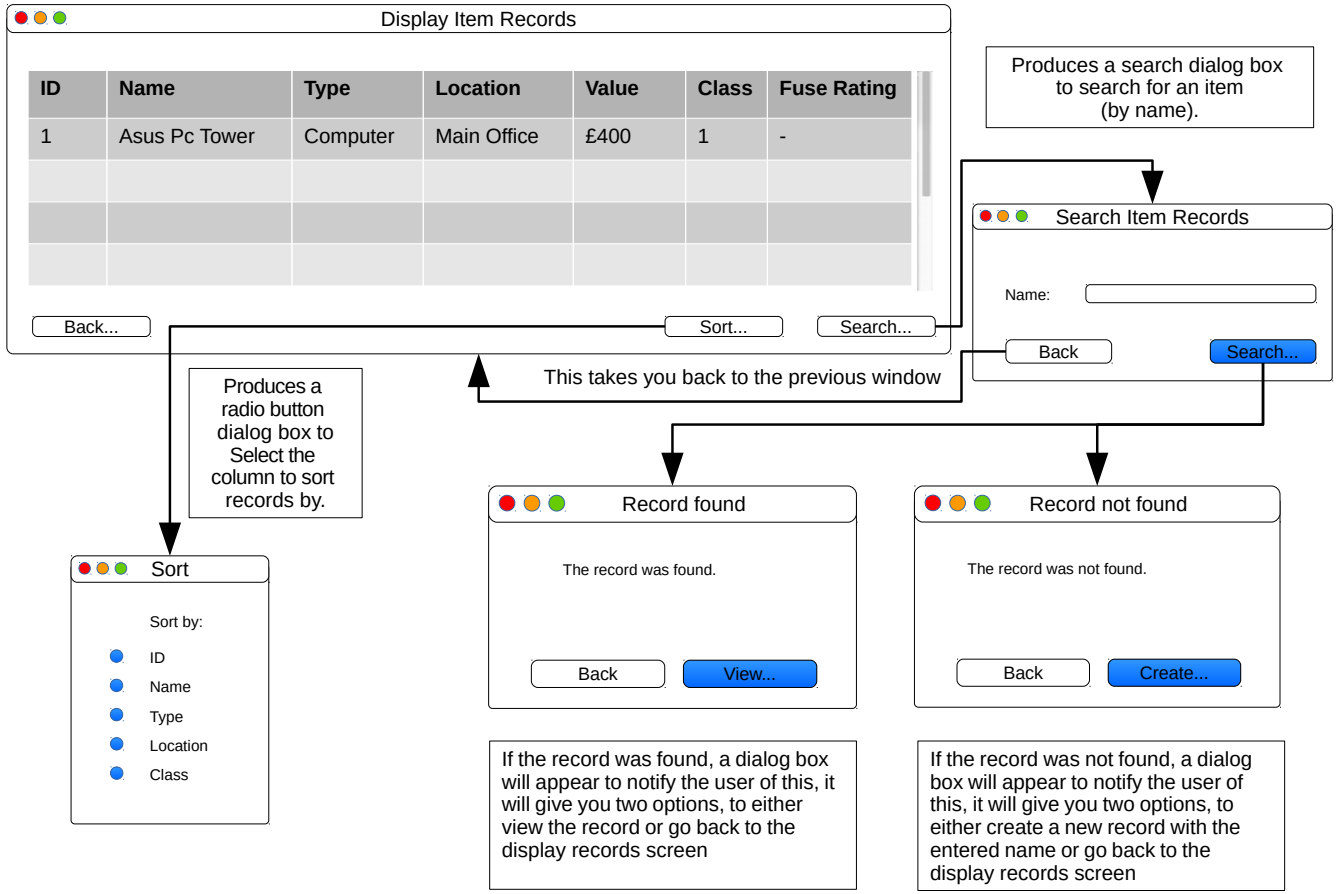



Figure 2.12: Login and Main Menu windows.

## 2.3 Hardware Specification

The hardware I am going to use are for a custom built Early 2008 Mac Pro. The specifications are as follows:

- 2x 2.8 GHz Quad-Core Intel®Xeon™Processor
- ATI Radeon HD 2600 XT 256MB Graphics Card
- 661-4449 Apple Mac Pro A1186 Motherboard
- 16.00GB DDR3 RAM
- 1TB SATA Disk-Drive
- 6TB RAID Storage
- Apple SuperDrive

I have chosen to build my system for this specification as this is the computer my client is going to run the application on, it is also a low cost choice of system spec to run on as the hardware has already been bought and is therefore ready and available to use.

## 2.4 Program Structure

### 2.4.1 Top-down design structure charts

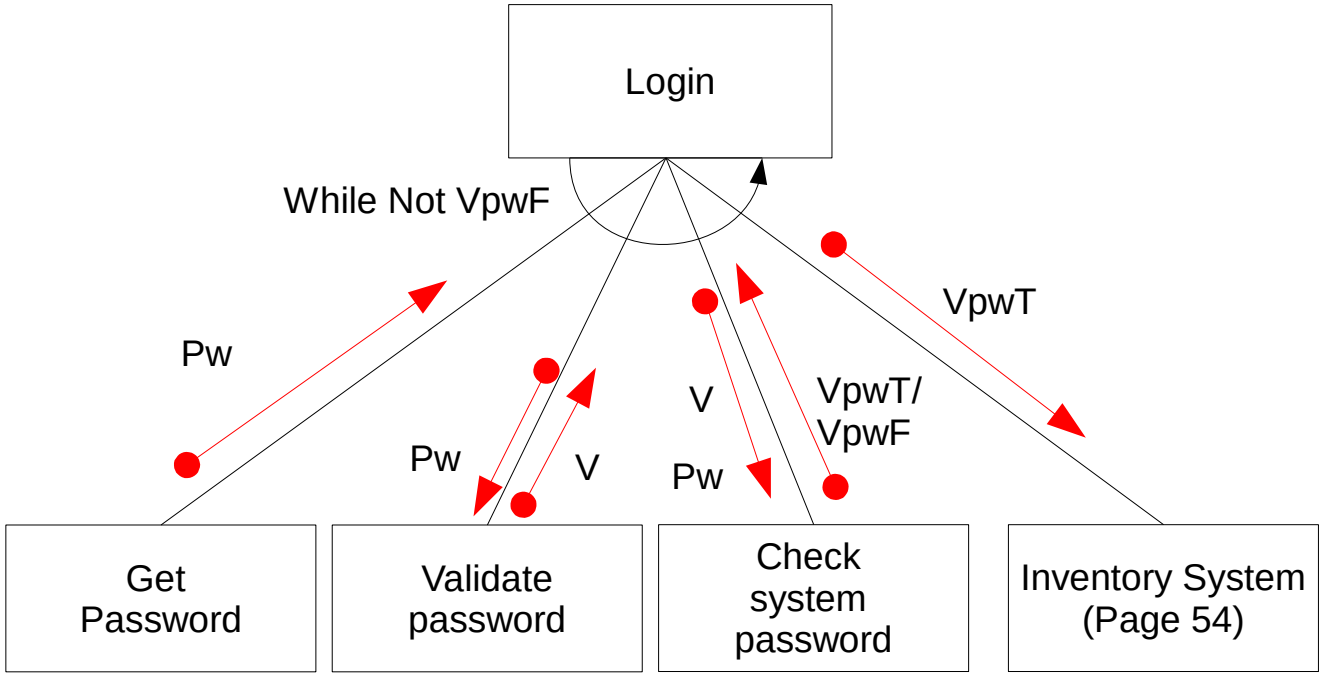



Figure 2.13: Object Diagram.

### 2.4.2 Algorithms in pseudo-code for each data transformation process

### 2.4.3 Object Diagrams

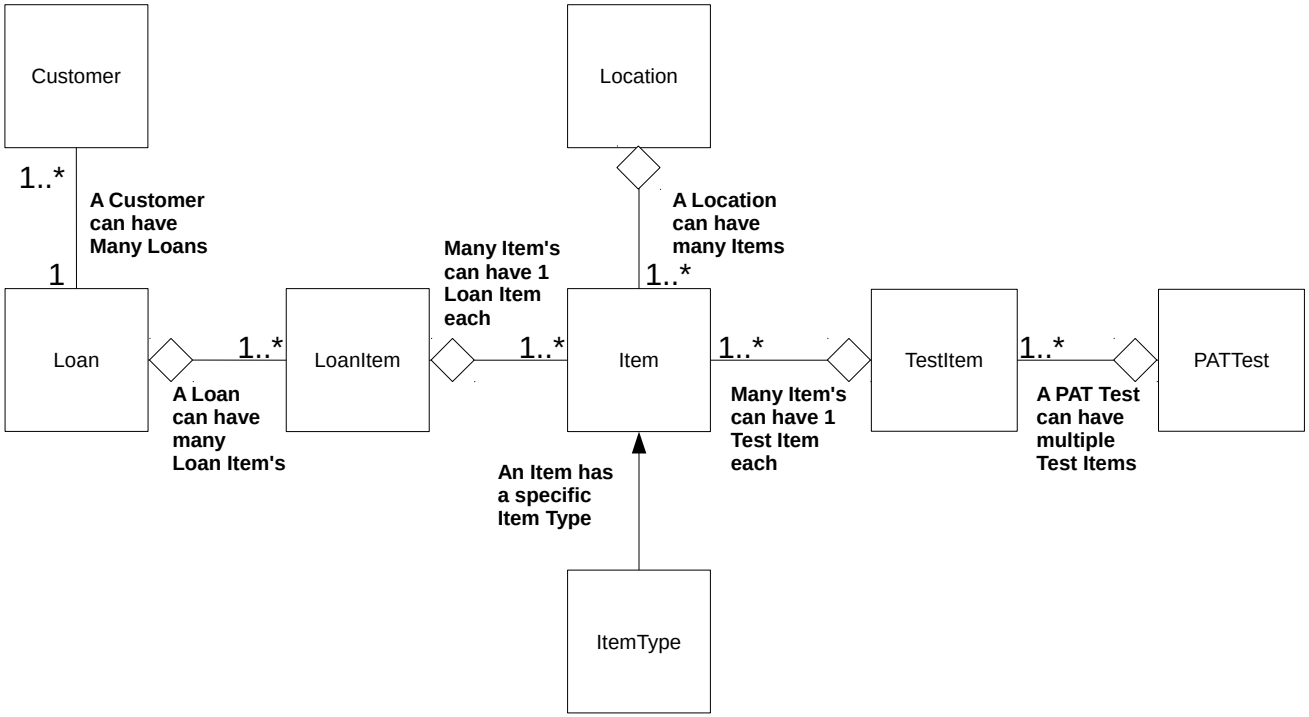



Figure 2.14: Object Diagram.

### 2.4.4 Class Definitions

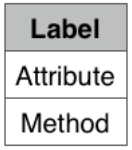



Figure 2.15: Class Diagram Key.

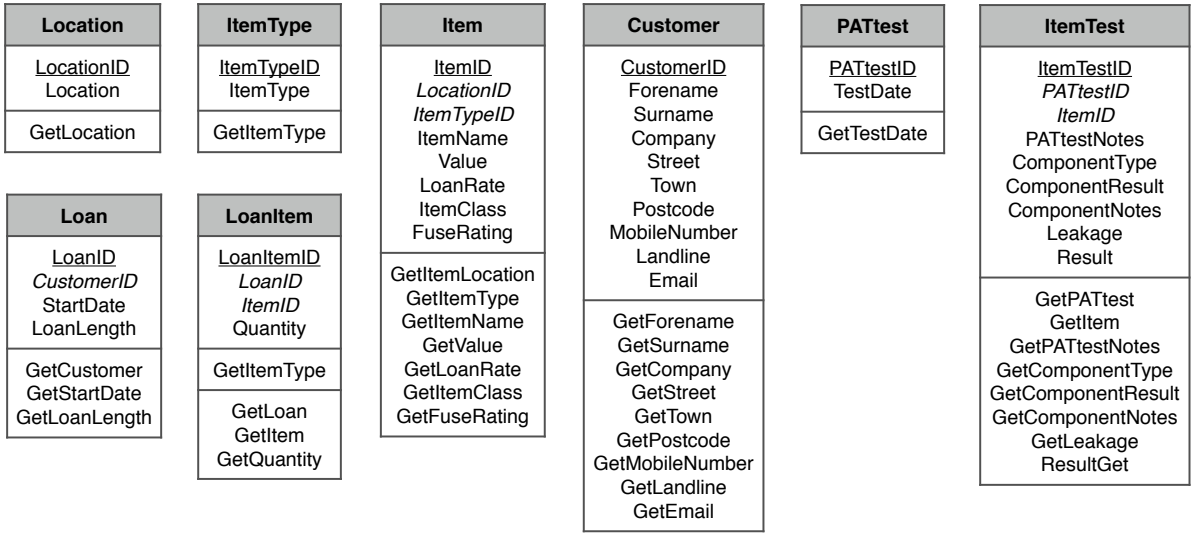



Figure 2.16: Class Diagrams.

## 2.5 Prototyping

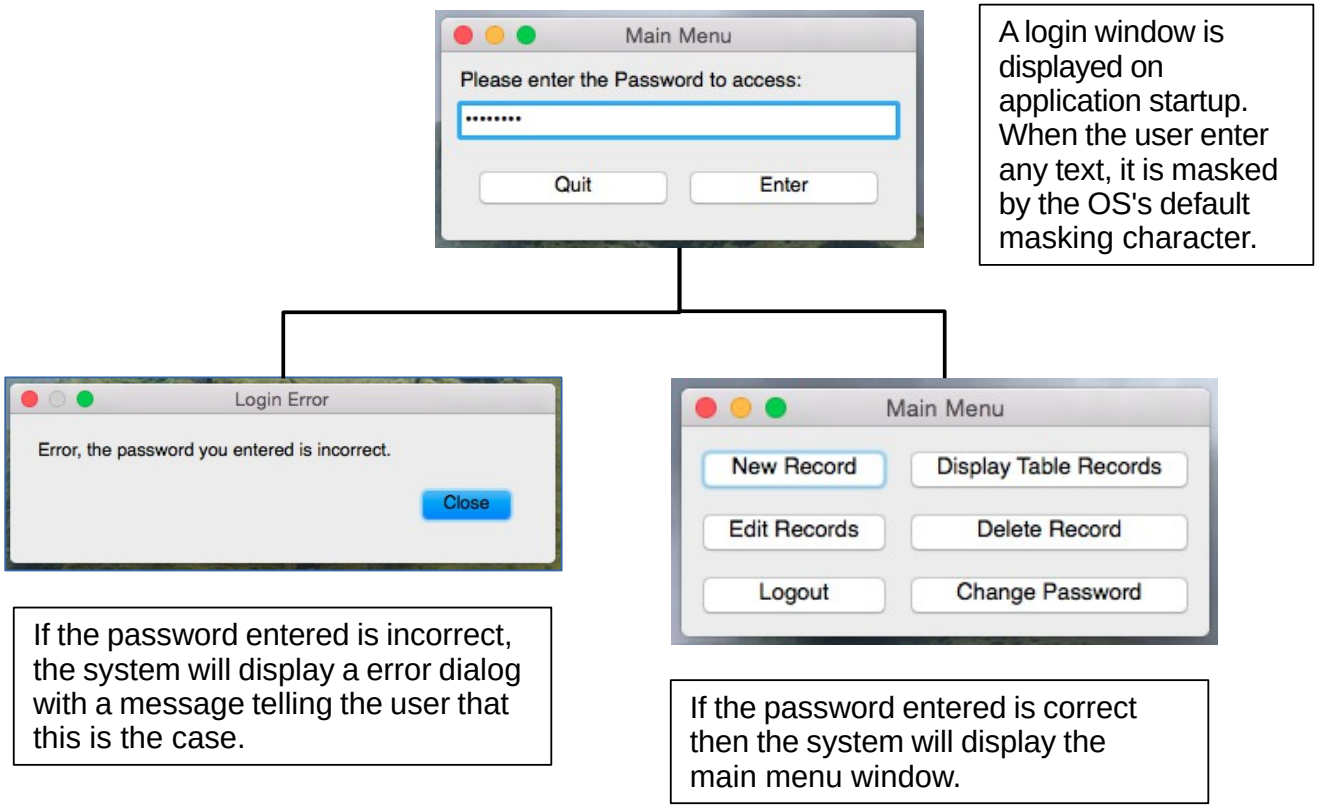



Figure 2.17: Login Prototype.

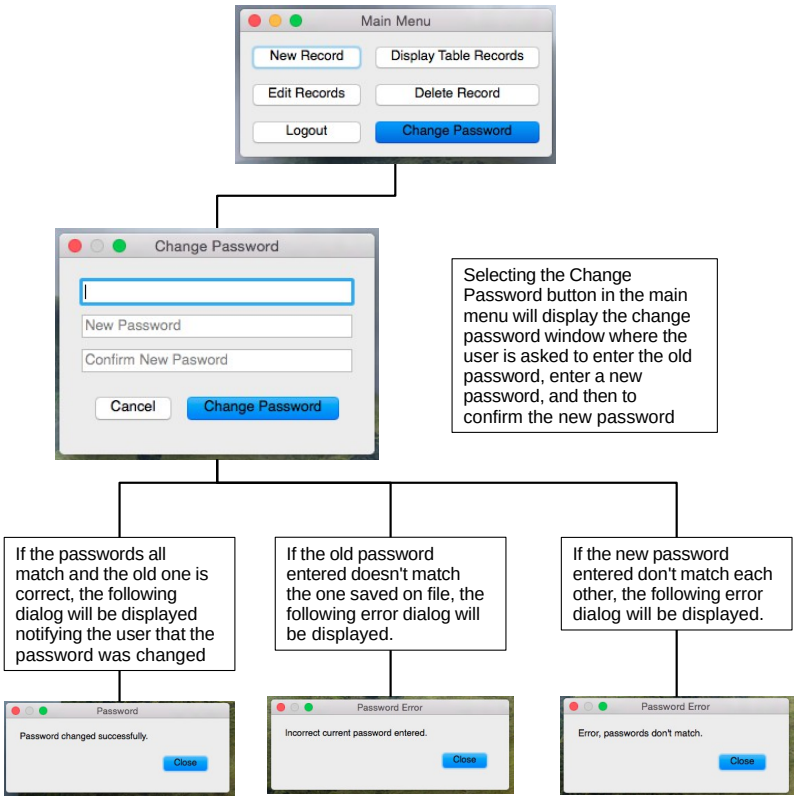



Figure 2.18: Change Password Prototype.

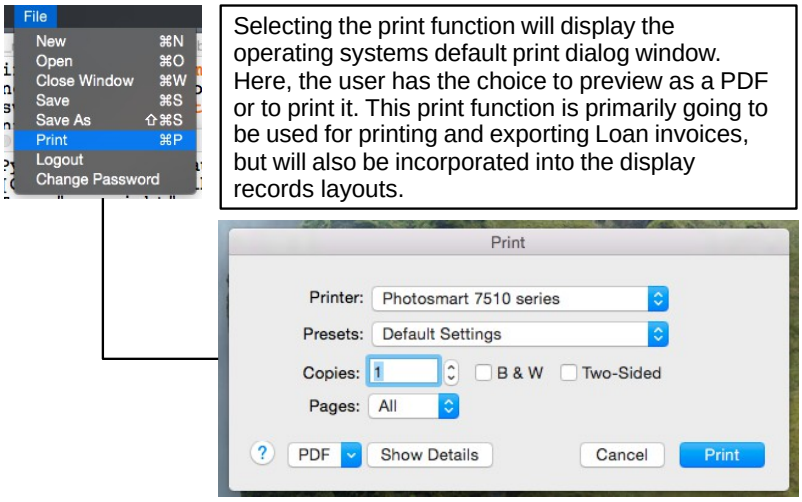



Figure 2.19: Change Password Prototype.

## 2.6 Definition of Data Requirements

### 2.6.1 Identification of all data input items

- Item Name
- Item Value
- Loan Rate (The amount charged, per day, for the loan of the item)
- Item Class (This is the class for electric items and determines the type of PAT test it receives)
- Fuse Rating

---

- Start Date (The exact date a loan started)
- Loan Length (The length of the loan in days)
- Quantity (The quantity of an item to be loan out, if there is more than one in stock)

---

- Forename
- Surname
- Company
- Street
- Town
- Post Code
- Mobile Number
- Email Address
- Landline Number

---

- Test date (The date on which the PAT tests took place)
- Test Description (Notes referring to why an item failed or other notes about an individual item)
- Leakage (The current not obtained by an electrical item)
- Test Result (The result of the PAT test either Pass or Fail)

### 2.6.2 Identification of all data output items

- Sub Total Cost (Loan Rate multiplied by the Quantity)
- Total Cost (The sum of all the Sub Total Costs in a single loan)

**Output to database**

- Item Name
- Item Value
- Loan Rate (The amount charged, per day, for the loan of the item)
- Item Class (This is the class for electric items and determines the type of PAT test it receives)
- Fuse Rating

---

- Start Date (The exact date a loan started)
- Loan Length (The length of the loan in days)
- Quantity (The quantity of an item to be loan out, if there is more than one in stock)

---

- Forename
- Surname
- Company
- Street
- Town
- Post Code
- Mobile Number
- Email Address
- Landline Number

---

- Test date (The date on which the PAT tests took place)
- Test Description (Notes referring to why an item failed or other notes about an individual item)

- Leakage (The current not obtained by an electrical item)
- Test Result (The result of the PAT test either Pass or Fail)

### 2.6.3 Explanation of how data output items are generated

### 2.6.4 Data dictionary

| **Name** | **Data Type** | **Length** | **Validation** | **Example Data** | **Comment** |
|---|---|---|---|---|---|
| ItemTypeID | Integer | 1-435 | Range | 253 | This is the **Primary Key** for the ItemType table, and *Foreign Key* for the Item table |
| ItemType | Text | 5-40 Characters | Length | Computer | This holds the description of each type of Item. |
| LocationID | Integer | 1-3 Figures | Range | 3 | This is the **Primary Key** for the Location table and a *Foreign Key* for the Item table |
| Location | Text | 1-30 Characters | Length | Main Offices | This holds the name of the locations |

| **Name** | **Data Type** | **Length** | **Validation** | **Example Data** | **Comment** |
|---|---|---|---|---|---|
| ItemID | Integer | 1-435 | Range | 253 | This is the **Primary Key** for the Item table, and *Foreign Key* for the LoanItem and ItemTest tables |
| ItemName | Text | 5-40 Characters | Length | Arkaos Server | This gives the name of each item entered |
| ItemValue | Real | 2-5 Figures | Range | 1,300 | This holds the data for the monetary value for each item |
| LoanRate | Real | 2-5 Figures | Range | 7 | This holds the data for the monetary loan rate for each item |
| ItemClass | Integer | 1 Character | Length | 2 | A field to show what class of electrical equipment the item is |
| FuseRating | Text | 1-3 Characters | Length | 5A | A field which displays the fuse rating |

| **Name** | **Data Type** | **Length** | **Validation** | **Example Data** | **Comment** |
|---|---|---|---|---|---|
| LoanID | Integer | 1-435 | Range | 56 | This is the **Primary Key** for the Loan table and is a *Foreign Key* in the Loan Item table |
| StartDate | Real | 1-5 Figures | Range | 75 | Holds data displaying when the loan started |
| LoanLength | Integer | 1-3 Figures | Range | 7 | Holds the data for the length of the loan |
| | | | | | |
| LoanItemID | Integer | 1-425 | Range | 26 | This is the **Primary Key** for the Loan Listings table |
| Quantity | Integer | 1-10 | Range | 3 | This hold data referring to the amount of one item has been loaned out |

| **Name** | **Data Type** | **Length** | **Validation** | **Example Data** | **Comment** |
|---|---|---|---|---|---|
| CustomerID | Integer | 1-255 | Range | 52 | This is the **Primary Key** for the Customer table |
| Forename | Text | 3-20 Characters | Length | John | A field for the customers forename |
| Lastname | Text | 3-20 Characters | Length | Smith | A field for the customers surname |
| Company | Text | 3-20 Characters | Length | Digital Lighting Cambs | A field for the company's name |
| Street | Text | 3-30 Characters | Length | 129 Cedar Crescent | A field for the company's Street address |
| Town | Text | 3-30 Characters | Length | Sawston | A field for the company's Town |
| County | Text | 3-20 Characters | Length | Cambs | A field for the company's County |
| PostCode | Text | 6-7 Characters | Format | CB22 7RX | A field for the company's Postcode |
| MobileNumber | Text | 11 Characters | Format | 07891234567 | A field for the customers mobile number |
| LandLine | Text | 11 Characters | Format | 01234567890 | A field for the customers landline phone |
| Email | Text | 7-30 Characters | Length | john.smith@example.com | A field for the customers email address |

| **Name** | **Data Type** | **Length** | **Validation** | **Example Data** | **Comment** |
|---|---|---|---|---|---|
| PATtestID | Integer | 1-255 | Range | 52 | This is the **Primary Key** for the PATtest table |
| TestDate | Date | 10 Characters | Format | 01/12/2014 | A field that displays the date of the PAT test |
| | | | | | |
| ItemTestID | Integer | 1-255 | Range | 52 | This is the **Primary Key** for the ItemTest table |
| ItemDescription | Text | 3-400 Characters | Length | Waltham portable TV | A field that describes the item to be tested |
| ProtectiveCondTest | Float | 4 Characters | Length | - | A field displaying the resistance of an item, in Ohms, to a 200mA current |
| InsulationTest | Text | 3 Characters | Length | ¿20 | A field displaying the Insulation of an item, in Ohms, to a 250V or 500V Potential Difference |
| Leakage | Float | 4 Characters | Format | 0.03 | A field that shows the current not obtained by the item, in milliamperes |
| TestResult | Boolean | - | Presence Check | True | A field to show if an item Passed or not |

### 2.6.5 Identification of appropriate storage media

My system will not need to be accessed by more than 5 people, storing the database file on the server won't be necessary as everyone will then have access to the database file. Therefore, I have chosen to store the database file and application on a single machine which can be accessed by the people who need to use it at any time. The computer is in a central location and easily accessible by those who need to use it and has multiple hard disk drives which I can make use of for storage and backup.

## 2.7 Database Design

### 2.7.1 ER Diagrams

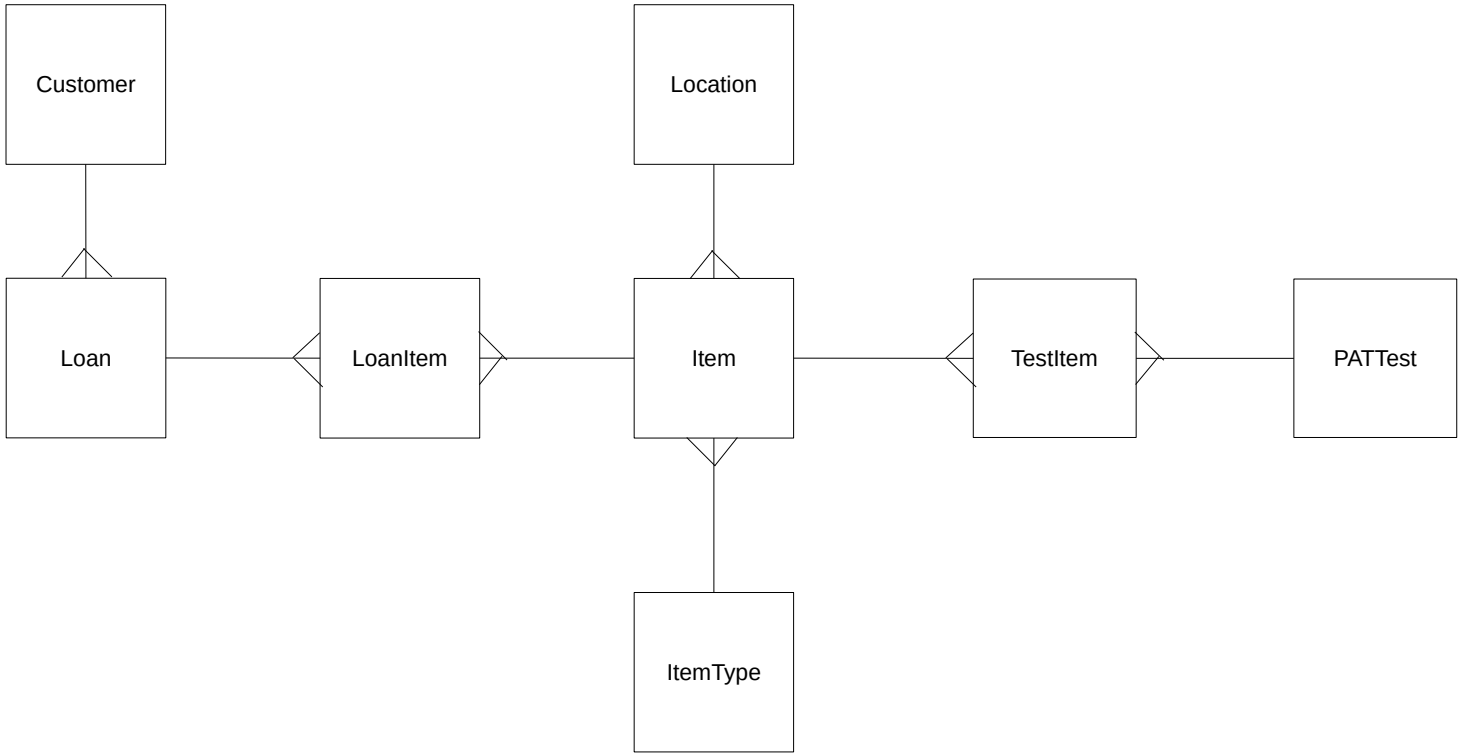



Figure 2.20: ER Diagrams.

### 2.7.2 Entity Descriptions

**Location**(LocationID, Location)

**ItemType**(ItemTypeID, ItemType)

**Item**(ItemID, ItemName, ItemValue, LoanRate, ItemClass, FuseRating, *ItemTypeID*, *LocationID* )

**Customer**(CustomerID, Forename, Surname, Company, Street, Town, PostCode, MobileNumber, Landline, Email)

**Loan**(LoanID, *CustomerID*, StartDate, LoanLength)

**LoanItem**(LoanItemID, *LoanID*, *ItemID*, Quantity)

**PATtest**(PATtestID, TestDate)

**ItemTest**(ItemTestID, *PATtestID*, *ItemID*, PATtestNotes, ComponentType, ComponentResult, ComponentNotes, Leakage, TestResult)

### 2.7.3 Normalisation

**UNF to 3NF**

| **Un-Normalised Form(UNF)** |
|---|
| ItemID |
| ItemName |
| ItemType |
| Location |
| ItemValue |
| LoanRate |
| LoanID |
| StartDate |
| CustomerID |
| Forename |
| Lastname |
| Company |
| Street |
| Town |
| PostCode |
| MobileNumber |
| LandLine |
| Email |
| PATtestID |
| TestResult |
| TestDate |
| ItemDescription |
| ItemClass |
| FuseRating |
| PATTestNotes |
| ComponentType |
| ComponentResult |
| ComponentNotes |
| Leakage |

| **First-Normalised Form(1NF)** | |
|---|---|
| **Non-Repeating** | **Repeating** |
| ItemID | LoanID |
| ItemName | *ItemID* |
| ItemValue | StartDate |
| LoanRate | CustomerID |
| ItemClass | Forename |
| FuseRating | Lastname |
| | Company |
| | Street |
| | Town |
| | PostCode |
| | MobileNumber |
| | Landline |
| | Email |
| | PATtestID |
| | TestDate |
| | PATTestNotes |
| | ComponentType |
| | ComponentResult |
| | ComponentNotes |
| | Leakage |
| | TestResult |

| **Second-Normalised Form(2NF)** | |
|---|---|
| **Non-Repeating** | **Repeating** |
| ItemID | LoanID |
| ItemName | *ItemID* |
| ItemValue | StartDate |
| LoanRate | |
| ItemClass | CustomerID |
| FuseRating | Forename |
| | Lastname |
| | Company |
| | Street |
| | Town |
| | PostCode |
| | MobileNumber |
| | Landline |
| | Email |
| | PATtestID |
| | TestDate |
| | PATTestNotes |
| | ComponentType |
| | ComponentResult |
| | ComponentNotes |
| | Leakage |
| | TestResult |
| | Location |
| | ItemType |

| **Third-Normalised Form(3NF)** | |
|---|---|
| **Non-Repeating** | **Repeating** |
| ItemID | LoanID |
| *LocationID* | *CustomerID* |
| *ItemTypeID* | LoanLength |
| ItemName | |
| ItemValue | |
| LoanRate | LoanItemID |
| ItemClass | *LoanID* |
| FuseRating | *ItemID* |
| | Quantity |
| | |
| | CustomerID |
| | Forename |
| | Lastname |
| | Company |
| | Street |
| | Town |
| | PostCode |
| | MobileNumber |
| | Landline |
| | Email |
| | |
| | PATtestID |
| | TestDate |
| | |
| | TestItem |
| | *PATtestID* |
| | *ItemID* |
| | PATTestNotes |
| | ComponentType |
| | ComponentResult |
| | ComponentNotes |
| | Leakage |
| | TestResult |
| | |
| | LocationID |
| | Location |
| | |
| | ItemTypeID |
| | ItemType |

## 2.8 SQL Queries

**SQL query getting all the items from the Item table in the database ready to be formatted and displayed on screen**

```
SELECT
Item.ItemID,
Item.ItemName,
Item.ItemValue,
Item.LoanRate,
Item.ItemClass,
Item.FuseRating,
ItemType.ItemType,
Location.Location
FROM Item, ItemType, Location
WHERE Item.LocationID = Location.LocationID
   AND Item.ItemTypeId = ItemType.ItemTypeID
```

**SQL query getting all the loans from the Loan table in the database ready to be formatted and displayed on screen**

```
SELECT
Loan.LoanID,
Loan.StartDate,
Loan.LoanLength,
Customer.CustomerID,
Customer.Company,
FROM Loan, Customer
WHERE Loan.CustomerID = Customer.CustomerID
```

**SQL query getting all the loan items from the LoanItem table in the database ready to be formatted and displayed on screen**

```
SELECT
LoanItem.LoanItemID,
LoanItem.LoanID,
LoanItem.Quantity,
Item.ItemName,
```

```
    Item.LoanRate,
    FROM LoanItem, Item
    WHERE LoanItem.ItemID = Item.ItemID
```

**SQL query getting all the item tests from the ItemTest table in the database ready to be formatted and displayed on screen**

```
      SELECT
      ItemTestID,
      ItemTest.PATtestNotes,
      ItemTest.Leakage,
      ItemTest.TestResult,
      Item.ItemName,
      Item.ItemClass,
      Item.FuseRating
      FROM ItemTest
      WHERE ItemTest.ItemID = Item.ItemID
```

**SQL Query searching the database for Items at a specific Location then orders them A-Z by ItemName**

Location and Item are tables in the database, Item ID, ItemName and ItemValue are attributes in the Item table. LocationID is an attribute in the Location table

```
    SELECT
    Item.ItemID,
    Item.ItemName,
    Item.ItemValue,
    Location.LocationID
    FROM Item, Location
    WHERE LocationID = ? AND
    Location.LocationID = Item.LocationID
    ORDER BY ItemName ASC
```

**SQL Query searching the database for certain ItemTypes then orders them by ItemName from A-Z**

Location and Item are tables in the database, Item ID, ItemName and ItemValue are attributes in the Item table. ItemTypeID is an attribute in the ItemType table

```
1     SELECT
2     Item.ItemID,
3     Item.ItemName,
4     Item.ItemValue,
5     ItemType.ItemTypeID
6     FROM Item, Location
7     WHERE ItemTypeID = ? AND
8     ItemType.ItemTypeID = Item. ItemTypeID
9     ORDER BY ItemName ASC
```

**SQL Query searching the database to display Loans taken out by a certain Company, ordered by date ascending**

```
1     SELECT
2     Loan.LoanID
3     Customer.Company,
4     Item.ItemName
5     Item.LoanRate
6     LoanItem.Quantity,
7     Loan.StartDate,
8     Loan.LoanLength,
9     FROM Loan, Customer, Item
10    WHERE Customer.CustomerID = Loan.CustomerID AND
         Loan.LoanItemID = LoanItem.LoanItemID AND
         LoanItem.ItemID = Item.ItemID
```

## 2.9 Security and Integrity of the System and Data

### 2.9.1 Security and Integrity of Data

The system will store personal data referring to an individual or a company. This data will fall under the data protection acts. This will mean that the data will need to be kept up to date and would therefore need a way to edit the data. All the information stored in the database should therefore be encrypted to keep this data secure and only accessible through my program which will be protected with a password. I will need to make sure the data stored is valid and correct, to do this I will need to use validation algorithms to make sure they are feasible. I will also need to use referential integrity in my database to ensure that when adding, updating and removing data to the database, key information isn't missing from records.

### 2.9.2 System Security

It is important that the information in my database is secure and free from theft, corruption and tampering. This will be prevented with the use of a password to access the system. If the password that was entered is incorrect, the user will not be able to gain access to the system and will be notified by a pop-up window. I will need to encrypt my data to avoid people from outside my system from being able to access the data. All of the data entered into the system will undergo validation to make sure that it is suitable and correct. Because some of the data fall under the data protection act, I will need to ensure that:

- The data will be destroyed after 11 years of collection
- Only data that is necessary will be collected and stored.
- The data will be updated when necessary so that the data is up to date and accurate
- The data that is stored will only be used by the Church and not passed on to anyone else
- The data will be secured securely, to ensure that it is only accessed by authorised people
- The data will not be transferred to other countries

## 2.10 Validation

## 2.11 Testing

### 2.11.1 Outline Plan

| **Test Series** | **Purpose of Test Series** | **Testing Strategy** | **Strategy Rationale** |
|---|---|---|---|
| 1 | Test the flow of control between interfaces | Top-down testing | |
| 2 | Test data input validation works | Botton-up testing | Test each component will commence when they have been developed |
| 3 | Test data input is stored in the correctly | Black box testing | |
| 4 | Test Algorithms and check output is correct | White box testing | |
| 5. | Test system meets requirements | Acceptance Testing | |

### 2.11.2 Detailed Plan

| Test Series | Purpose of Test | Test Description | Test Data | Test Data Type (Normal/ Erroneous/ Boundary) | Expected Result | Actual Result | Evidence |
|---|---|---|---|---|---|---|---|
| 1.01 | Test the "Login" button functions correctly | This should link to the main menu screen | Enter "password" and click the "Login" button | Normal | The main menu window should be displayed | | |
| 1.02 | Test the "Logout" button functions correctly | This should link back to the login screen | Click "Logout" button | Normal | The login screen should be displayed | | |
| 1.03 | Test the "Change Password" functions correctly | This should link to the Change password dialog window | Click "Change Password" button | Normal | The change password window should be displayed | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.04 | Test the and "Cancel" buttons functions correctly | This should link back to the main menu window | Click "Cancel" button | Normal | The main menu should be displayed | | |
| 1.05 | Test the "Confirm Password" button functions correctly | This should link to a message dialog confirming that the password has been changed, other messages will be displayed if the current password is incorrect or the new passwords don't match | Click the "Confirm Password" button | Normal | The message dialog window should be displayed | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.06 | Test the "OK" button on the message dialog functions correctly | This should link back to the main menu window | Click the "OK" button | Normal | The main menu window should be displayed | | |
| 1.07 | Test the enter record button functions correctly | This should link to a table selection window | Click "Enter Record" button | Normal | The table selection dialog box should be displayed | | |
| 1.08 | Test the table selection dialog function correctly on selecting the Item table | This should link to the enter item record window | Select "Item Table" and click "Enter Record" | Normal | The enter Item Record Window should be displayed | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.09 | Test the table selection dialog function correctly on selecting the Customer table | This should link to the enter customer record window | Select "Customer Table" and click "Enter Record" | Normal | The enter Customer Record Window should be displayed | | |
| 1.10 | Test the table selection dialog function correctly on selecting the Loan table | This should link to the enter loan record window | Select "Loan Table" and click "Enter Record" | Normal | The enter Loan Record Window should be displayed | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.10 | Test the table selection dialog function correctly on selecting the PAT test table | This should link to the enter PAT test record window | Select "PAT test Table" and click "Enter Record" | Normal | The enter PAT test Record Window should be displayed | | |
| 1.11 | Check that the "Confirm" button functions correctly on the enter Item record window | This should link back to the main menu | Select the "Confirm" button | Normal | The main window should be displayed | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.12 | Check that the "Confirm" button functions correctly on the enter Customer record window | This should link back to the main menu | Select the "Confirm" button | Normal | The main window should be displayed | | |
| 1.13 | Check that the "Confirm" button functions correctly on the enter Loan record window | This should link back to the main menu | Select the "Confirm" button | Normal | The main window should be displayed | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.14 | Check that the "Confirm" button functions correctly on the enter PAT test record window | This should link back to the main menu | Select the "Confirm" button | Normal | The main window should be displayed | | |
| 1.15 | Check that the "Cancel" button functions correctly on the enter Item record window | This should link back to the main menu | Select the "Cancel" button | Normal | The table selection window should be displayed | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.16 | Check that the "Cancel" button functions correctly on the enter Customer record window | This should link back to the main menu | Select the "Cancel" button | Normal | The table selection window should be displayed | | |
| 1.17 | Check that the "Cancel" button functions correctly on the enter Loan record window | This should link back to the main menu | Select the "Cancel" button | Normal | The table selection window should be displayed | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.18 | Check that the "Cancel" button functions correctly on the enter PAT test record window | This should link back to the main menu | Select the "Cancel" button | Normal | The table selection window should be displayed | | |
| 1.19 | Test the "Display Records" button functions correctly | This should link to a table selection window | Click "Display Records" button | Normal | The table selection dialog box should be displayed | | |
| 1.20 | Test the table selection dialog function correctly on selecting the Item table | This should link to the display item records window | Select "Item Table" and click "Display Record" | Normal | The enter Item Record Window should be displayed | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.21 | Test the display Item records "Search" button functions correctly | This should link to the search dialog | Click the "Search" button | Normal | The search dialog should be displayed | | |
| 1.22 | Test the search dialogs "Search" button functions correctly | This should link to a dialog window displaying the found record or a dialog window displaying a message that the record wasn't found | Click the "Search" button | Normal | The appropriate dialog should be displayed if a record was found or not | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.23 | Test the table selection dialog function correctly on selecting the Customer table | This should link to the display customer records window | Select "Customer Table" and click "Display Record" | Normal | The enter Customer Record Window should be displayed | | |
| 1.24 | Test the display Customer records "Search" button functions correctly | This should link to the search dialog | Click the "Search" button | Normal | The search dialog should be displayed | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.25 | Test the search dialogs "Search" button functions correctly | This should link to a dialog window displaying the found record or a dialog window displaying a message that the record wasn't found | Click the "Search" button | Normal | The appropriate dialog should be displayed if a record was found or not | | |
| 1.26 | Test the table selection dialog function correctly on selecting the Loan table | This should link to the display item loan records window | Select "Loan Table" and click "Display Record" | Normal | The enter Loan Record Window should be displayed | | |
| 1.27 | Test the display loan records "Search" button functions correctly | This should link to the search dialog | Click the "Search" button | Normal | The search dialog should be displayed | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.28 | Test that the display loan records "Print" button functions correctly | This should link to the print dialog box | Click the "Print" button | Normal | The print dialog box should be displayed | | |
| 1.28 | Test the search dialogs "Search" button functions correctly | This should link to a dialog window displaying the found record or a dialog window displaying a message that the record wasn't found | Click the "Search" button | Normal | The appropriate dialog should be displayed if a record was found or not | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.29 | Test the table selection dialog function correctly on selecting the PAT test table | This should link to the display PAT test records window | Select "PAT test Table" and click "Display Record" | Normal | The enter PAT test Record Window should be displayed | | |
| 1.30 | Test the display PAT test records "Search" button functions correctly | This should link to the search dialog | Click the "Search" button | Normal | The search dialog should be displayed | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.31 | Test the search dialogs "Search" button functions correctly | This should link to a dialog window displaying the found record or a dialog window displaying a message that the record wasn't found | Click the "Search" button | Normal | The appropriate dialog should be displayed if a record was found or not | | |
| 1.32 | Test the edit record button functions correctly | This should link to a table selection window | Click "Edit Record" button | Normal | The table selection dialog box should be displayed | | |
| 1.33 | Test the table selection dialog function correctly on selecting the Item table | This should link to the edit item record window | Select "Item Table" and click "Edit Record" | Normal | The edit Item Record Window should be displayed | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.34 | Test the table selection dialog function correctly on selecting the Customer table | This should link to the edit customer record window | Select "Customer Table" and click "Edit Record" | Normal | The edit Customer Record Window should be displayed | | |
| 1.35 | Test the table selection dialog function correctly on selecting the Loan table | This should link to the edit loan record window | Select "Loan Table" and click "Edit Record" | Normal | The edit Loan Record Window should be displayed | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.36 | Test the table selection dialog function correctly on selecting the PAT test table | This should link to the edit PAT test record window | Select "PAT test Table" and click "Edit Record" | Normal | The edit PAT test Record Window should be displayed | | |
| 1.37 | Check that the "Confirm" button functions correctly on the edit Item record window | This should link back to the main menu | Select the "Confirm" button | Normal | The main window should be displayed | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.38 | Check that the "Confirm" button functions correctly on the edit Customer record window | This should link back to the main menu | Select the "Confirm" button | Normal | The main window should be displayed | | |
| 1.39 | Check that the "Confirm" button functions correctly on the edit Loan record window | This should link back to the main menu | Select the "Confirm" button | Normal | The main window should be displayed | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.40 | Check that the "Confirm" button functions correctly on the edit PAT test record window | This should link back to the main menu | Select the "Confirm" button | Normal | The main window should be displayed | | |
| 1.41 | Check that the "Cancel" button functions correctly on the edit Item record window | This should link back to the main menu | Select the "Cancel" button | Normal | The table selection window should be displayed | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.42 | Check that the "Cancel" button functions correctly on the edit Customer record window | This should link back to the main menu | Select the "Cancel" button | Normal | The table selection window should be displayed | | |
| 1.43 | Check that the "Cancel" button functions correctly on the edit Loan record window | This should link back to the main menu | Select the "Cancel" button | Normal | The table selection window should be displayed | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.44 | Check that the "Cancel" button functions correctly on the edit PAT test record window | This should link back to the main menu | Select the "Cancel" button | Normal | The table selection window should be displayed | | |
| 1.45 | Test the delete record button functions correctly | This should link to a table selection window | Click "Delete Record" button | Normal | The table selection dialog box should be displayed | | |
| 1.46 | Test the table selection dialog function correctly on selecting the Item table | This should link to the delete item record window | Select "Item Table" and click "Delete Record" | Normal | The edit Item Record Window should be displayed | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.47 | Test the table selection dialog function correctly on selecting the Customer table | This should link to the delete customer record window | Select "Customer Table" and click "Delete Record" | Normal | The edit Customer Record Window should be displayed | | |
| 1.48 | Test the table selection dialog function correctly on selecting the Loan table | This should link to the delete loan record window | Select "Loan Table" and click "Delete Record" | Normal | The edit Loan Record Window should be displayed | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.49 | Test the table selection dialog function correctly on selecting the PAT test table | This should link to the delete PAT test record window | Select "PAT test Table" and click "Delete Record" | Normal | The edit PAT test Record Window should be displayed | | |
| 1.50 | Check that the "Confirm" button functions correctly on the delete Item record window | This should link back to the main menu | Select the record(s) for delete and click "Confirm" button | Normal | The main window should be displayed | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.51 | Check that the "Confirm" button functions correctly on the delete Customer record window | This should link back to the main menu | Select the record(s) for delete and click "Confirm" button | Normal | The main window should be displayed | | |
| 1.52 | Check that the "Confirm" button functions correctly on the delete Loan record window | This should link back to the main menu | Select the record(s) for delete and click "Confirm" button | Normal | The main window should be displayed | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.53 | Check that the "Confirm" button functions correctly on the delete PAT test record window | This should link back to the main menu | Select the record(s) for delete and click "Confirm" button | Normal | The main window should be displayed | | |
| 1.54 | Check that the "Cancel" button functions correctly on the delete Item record window | This should link back to the main menu | Click the "Cancel" button | Normal | The table selection window should be displayed | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.55 | Check that the "Cancel" button functions correctly on the delete Customer record window | This should link back to the main menu | Click the "Cancel" button | Normal | The table selection window should be displayed | | |
| 1.56 | Check that the "Cancel" button functions correctly on the delete Loan record window | This should link back to the main menu | Click the "Cancel" button | Normal | The table selection window should be displayed | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.57 | Check that the "Cancel" button functions correctly on the delete PAT test record window | This should link back to the main menu | Click the "Cancel" button | Normal | The table selection window should be displayed | | |
| | | | | | | | |
| 2.01 | Verify that a password was entered | An error dialog box should appear if no password is entered | "password"<br><br><br>nothing | Normal<br><br><br>erroneous | Accepted<br><br><br>Rejected | | |

# Chapter 3

# Testing

## 3.1 Test Plan

### 3.1.1 Original Outline Plan

| **Test Series** | **Purpose of Test Series** | **Testing Strategy** | **Strategy Rationale** |
|---|---|---|---|
| Example | Example | Example | Example |

### 3.1.2 Changes to Outline Plan

| **Test Series** | **Purpose of Test Series** | **Testing Strategy** | **Strategy Rationale** |
|---|---|---|---|
| Example | Example | Example | Example |

### 3.1.3 Original Detailed Plan

| **Test Series** | **Purpose of Test** | **Test Description** | **Test Data** | **Test Data Type (Normal/ Erroneous/ Boundary)** | **Expected Result** | **Actual Result** | **Evidence** |
|---|---|---|---|---|---|---|---|
| Example | Example | Example | Example | Example | Example | Example | Example |

### 3.1.4 Changes to Detailed Plan

| **Test Series** | **Purpose of Test** | **Test Description** | **Test Data** | **Test Data Type (Normal/ Erroneous/ Boundary)** | **Expected Result** | **Actual Result** | **Evidence** |
|---|---|---|---|---|---|---|---|
| Example | Example | Example | Example | Example | Example | Example | Example |

## 3.2 Test Data

### 3.2.1 Original Test Data

### 3.2.2 Changes to Test Data

## 3.3 Annotated Samples

### 3.3.1 Actual Results

### 3.3.2 Evidence

## 3.4 Evaluation

### 3.4.1 Approach to Testing

### 3.4.2 Problems Encountered

### 3.4.3 Strengths of Testing

### 3.4.4 Weaknesses of Testing

### 3.4.5 Reliability of Application

### 3.4.6 Robustness of Application

# Chapter 4

# System Maintenance

## 4.1 Environment

### 4.1.1 Software

### 4.1.2 Usage Explanation

### 4.1.3 Features Used

## 4.2 System Overview

### 4.2.1 System Component

## 4.3 Code Structure

### 4.3.1 Particular Code Section

## 4.4 Variable Listing

## 4.5 System Evidence

### 4.5.1 User Interface

### 4.5.2 ER Diagram



### 4.5.3 Database Table Views

### 4.5.4 Database SQL

### 4.10.1 Module 1

# Chapter 5

# User Manual

## 5.1 Introduction

## 5.2 Installation

### 5.2.1 Prerequisite Installation

#### Installing Python

#### Installing PyQt

#### Etc.

### 5.2.2 System Installation

### 5.2.3 Running the System

## 5.3 Tutorial

### 5.3.1 Introduction

### 5.3.2 Assumptions

### 5.3.3 Tutorial Questions

#### Question 1

#### Question 2

### 5.3.4 Saving

# Chapter 6

# Evaluation

## 6.1 Customer Requirements

### 6.1.1 Objective Evaluation

## 6.2 Effectiveness

### 6.2.1 Objective Evaluation

## 6.3 Learnability

## 6.4 Usability

## 6.5 Maintainability

## 6.6 Suggestions for Improvement

## 6.7 End User Evidence

### 6.7.1 Questionnaires

### 6.7.2 Graphs



### 6.7.3 Written Statements