

COMP4 Coursework

Joel Butcher

November 5, 2014

Contents

1	Analysis	5
1.1	Introduction	5
1.1.1	Client Identification	5
1.1.2	Define the current system	6
1.1.3	Describe the problems	6
1.1.4	Section appendix	6
1.1.5	The current system	10
1.1.6	The proposed system	14
1.2	Objectives	26
1.2.1	General Objectives	26
1.2.2	Specific Objectives	26
1.2.3	Core Objectives	27
1.2.4	Other Objectives	27
1.3	ER Diagrams and Descriptions	28
1.3.1	ER Diagram	28
1.3.2	Entity Descriptions	29
1.4	Object Analysis	29
1.4.1	Object Listing	29
1.4.2	Relationship diagrams	30
1.4.3	Class definitions	31
1.5	Other Abstractions and Graphs	33
1.6	Constraints	33
1.6.1	Hardware	33
1.6.2	Software	34
1.6.3	Time	34
1.6.4	User Knowledge	34
1.6.5	Access restrictions	35
1.7	Limitations	35
1.7.1	Areas which will not be included in computerisation	35
1.7.2	Areas considered for future computerisation	35

1.8	Solutions	36
1.8.1	Alternative solutions	36
1.8.2	Justification of chosen solution	37
2	Design	39
2.1	Overall System Design	39
2.1.1	Short description of the main parts of the system . . .	39
2.1.2	System flowcharts showing an overview of the complete system	39
2.2	User Interface Designs	39
2.3	Program Structure	39
2.3.1	Top-down design structure charts	39
2.3.2	Algorithms in pseudo-code for each data transforma- tion process	39
2.3.3	Object Diagrams	39
2.3.4	Class Definitions	39
2.4	Prototyping	40
2.5	Definition of Data Requirements	40
2.5.1	Identification of all data input items	40
2.5.2	Identification of all data output items	40
2.5.3	Explanation of how data output items are generated .	40
2.5.4	Data Dictionary	40
2.5.5	Identification of appropriate storage media	40
2.6	Database Design	41
2.6.1	Normalisation	41
2.7	Security and Integrity of the System and Data	44
2.7.1	Security and Integrity of Data	44
2.7.2	System Security	44
2.8	Validation	44
2.9	Testing	44
2.9.1	Outline Plan	45
2.9.2	Detailed Plan	45
3	Testing	46
3.1	Test Plan	46
3.1.1	Original Outline Plan	47
3.1.2	Changes to Outline Plan	47
3.1.3	Original Detailed Plan	47
3.1.4	Changes to Detailed Plan	48
3.2	Test Data	49
3.2.1	Original Test Data	49

3.2.2	Changes to Test Data	49
3.3	Annotated Samples	49
3.3.1	Actual Results	49
3.3.2	Evidence	49
3.4	Evaluation	50
3.4.1	Approach to Testing	50
3.4.2	Problems Encountered	50
3.4.3	Strengths of Testing	50
3.4.4	Weaknesses of Testing	50
3.4.5	Reliability of Application	50
3.4.6	Robustness of Application	50
4	System Maintenance	51
4.1	Environment	52
4.1.1	Software	52
4.1.2	Usage Explanation	52
4.1.3	Features Used	52
4.2	System Overview	52
4.2.1	System Component	52
4.3	Code Structure	52
4.3.1	Particular Code Section	52
4.4	Variable Listing	52
4.5	System Evidence	52
4.5.1	User Interface	52
4.5.2	ER Diagram	52
4.5.3	Database Table Views	52
4.5.4	Database SQL	52
4.5.5	SQL Queries	52
4.6	Testing	52
4.6.1	Summary of Results	52
4.6.2	Known Issues	52
4.7	Code Explanations	52
4.7.1	Difficult Sections	52
4.7.2	Self-created Algorithms	52
4.8	Settings	52
4.9	Acknowledgements	52
4.10	Code Listing	52
4.10.1	Module 1	53
5	User Manual	54
5.1	Introduction	55

5.2	Installation	55
5.2.1	Prerequisite Installation	55
5.2.2	System Installation	55
5.2.3	Running the System	55
5.3	Tutorial	55
5.3.1	Introduction	55
5.3.2	Assumptions	55
5.3.3	Tutorial Questions	55
5.3.4	Saving	55
5.3.5	Limitations	55
5.4	Error Recovery	55
5.4.1	Error 1	55
5.4.2	Error 2	55
5.5	System Recovery	55
5.5.1	Backing-up Data	55
5.5.2	Restoring Data	55
6	Evaluation	56
6.1	Customer Requirements	57
6.1.1	Objective Evaluation	57
6.2	Effectiveness	57
6.2.1	Objective Evaluation	57
6.3	Learnability	57
6.4	Usability	57
6.5	Maintainability	57
6.6	Suggestions for Improvement	57
6.7	End User Evidence	57
6.7.1	Questionnaires	57
6.7.2	Graphs	57
6.7.3	Written Statements	57

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 design 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 his 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 responsible for the large amount of Audio and Visual equipment used on the church's Sunday services. This currently involves a 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 like this database to be located on the church's central server so that it can be accessed by all staff if it is deemed necessary. He would use this database to store location, value and insurance details in case of damage or theft. He would like all of the information kept as a virtual copy as well as a hard copy to be kept as a visual backup in case of hard drive 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 spread sheets. There is one spread sheet for each of three locations; main office, main church building, and storage. Each spreadsheet 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 spreadsheet 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.

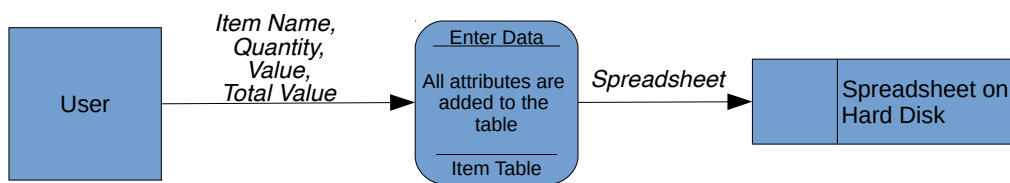


Figure 1.5: Entering a new item.



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

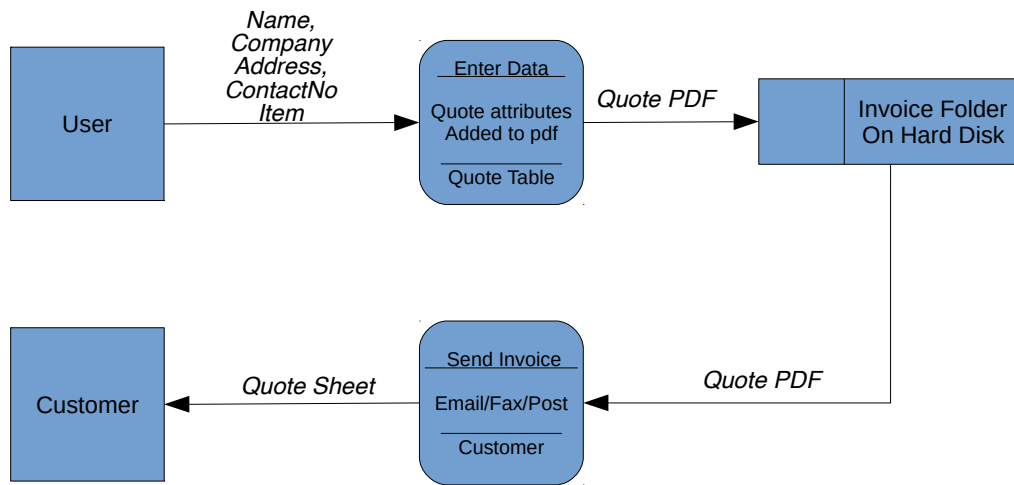


Figure 1.7: Creating and sending the initial quote for a loan.



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:

	A	B	C	D	E	F	G	H	I	J
1										
2		INSURANCE LIST FOR CINEWORLD								
3		ITEM	TYPE	PRICE	QUANTITY	SUB-TOTAL				
4										
5										
6		HDMI CABLE 10M	CABLING		1	1				
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22										
23										
24										
25										
26										
27										
28										
29										
30										
31										
32										
33										
34										
35										
36										
37										
38										
39										
40										

Figure 1.9: Josh Entering Item Name.

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

	A	B	C	D	E	F	G	H	I	J
1										
2		INSURANCE LIST FOR CINEWORLD								
3		ITEM	TYPE	PRICE	QUANTITY	SUB-TOTAL				
4										
5										
6		HDMI CABLE 10M	CABLING			=D6*E6				
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22										
23										
24										
25										
26										
27										
28										
29										
30										
31										
32										
33										
34										
35										
36										
37										
38										
39										
40										

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 - Item-Type Table
User	ItemType	Text	Database - Item-Type Table
-	-	-	-
Generated	LocationID	Integer	Database - Location Table
User	Location	Text	Database - Location Table
-	-	-	-
Generated	ItemID	Integer	Database - Item Records
Database - ItemType Table	<i>ItemTypeID</i>	Integer	Database - Item Table
Database - Location Table	<i>LocationID</i>	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	<i>ItemID</i>	Integer	Database - LoanListing Table
User	LoanQuantity	Integer	Database - LoanListing Table
-	-	-	-
Generated	CustomerLoanID	Integer	Database - Loan Table
Database - Customer Table	<i>CustomerID</i>	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 - Cus- tomer Table
User	Forename	Text	Database - Cus- tomer Table
User	Lastname	Text	Database - Cus- tomer Table
User	Company	Text	Database - Cus- tomer Table
User	Street	Text	Database - Cus- tomer Table
User	Town	Text	Database - Cus- tomer Table
User	County	Text	Database - Cus- tomer Table
User	PostCode	Text	Database - Cus- tomer Table
User	MobileNumber	Text	Database - Cus- tomer Table
User	LandLine	Text 16	Database - Cus- tomer Table
User	Email	Text	Database - Cus- tomer Table

Source	Data	Data Type	Destination
Generated	ItemTestID	Integer	Database - ItemTest Table
Database - PATtest Records	<i>PATtestID</i>	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 - PAT- test Table
User	TestDate	Date	Database - PAT- test Table

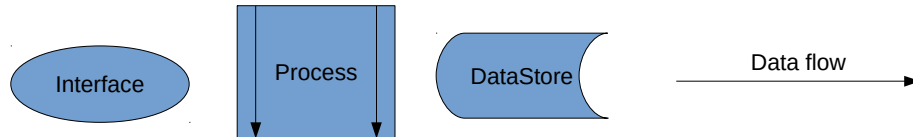
Data flow diagram

Figure 1.11: Flow Diagram Key.

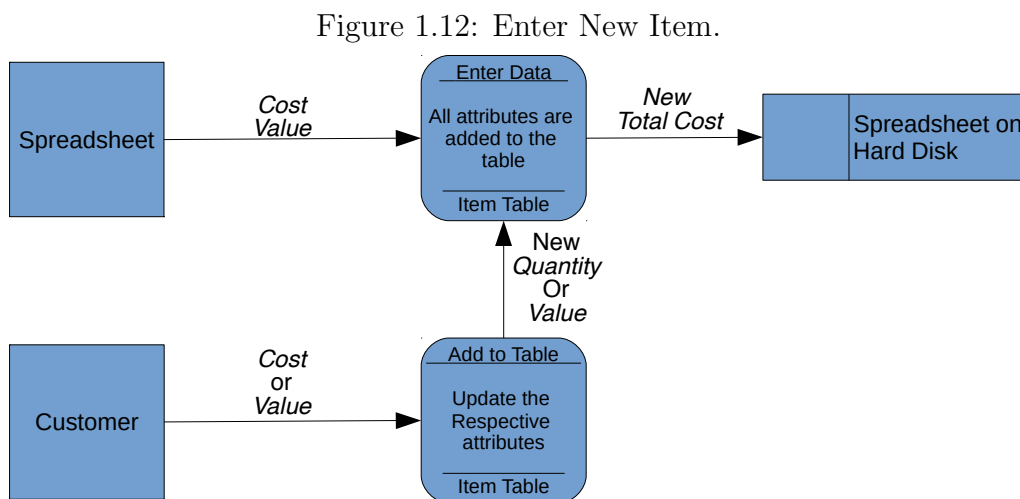


Figure 1.12: Enter New Item.



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 <i>Foreign Key</i> 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:

$$18\text{KB} * 60 = 1080\text{KB}$$

$$1080\text{KB} / 1024 = 1.05\text{MB}$$

If the rest of database management 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: Relationship Diagram.

1.4.3 Class definitions

Label
Attribute
Method

Figure 1.17: Class Diagram Key.

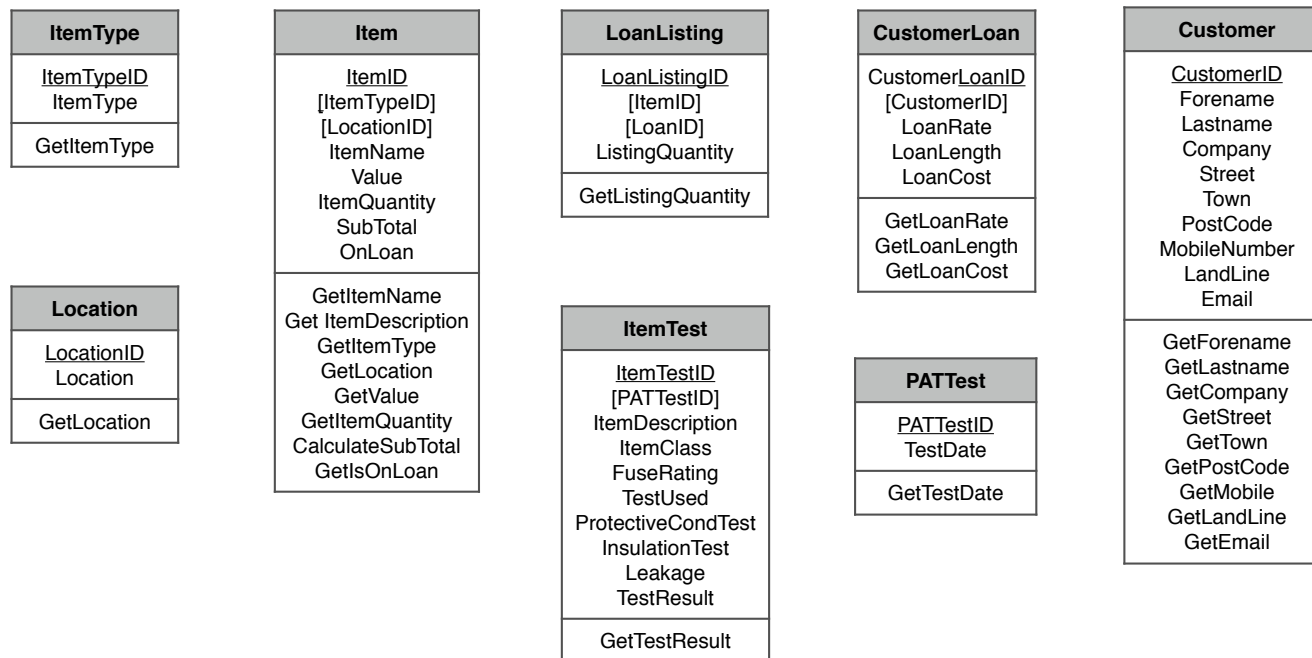


Figure 1.18: Class Diagrams.

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 as 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 operate 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 13th 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 of 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	<ul style="list-style-type: none">• No need to install additional software, only a simple database management system such as "Microsoft Access" or "Filemaker".	<ul style="list-style-type: none">• Database management systems often cost a substantial amount of money for a license.
Web based application	<ul style="list-style-type: none">• Easily accessible by other users. Doesn't rely on one machine.• Can have 'Cloud based' storage of files.• More than one user can be logged on at a time.	<ul style="list-style-type: none">• Website or server hosting can be expensive.• More advanced security methods will be required due to the system being constantly online and therefore vulnerable to attack.• Better networking knowledge required to compensate for the security implications and risks.

Alternative solution	Advantages	Disadvantages
Terminal or Command based application	<ul style="list-style-type: none"> • More power efficient as it isn't graphics heavy, much easier to design as the interface is just text. • Fast efficient operation provided the client has knowledge of terminal and shell commands. 	<ul style="list-style-type: none"> • Careful error handling needed as the user could enter any known/valid command. • Training is required so that the client knows what commands to use when. • There are often commands that the client don't know about that could potentially corrupt his computer.
Python desktop application with a GUI	<ul style="list-style-type: none"> • Designed and layout can be client specific. • Minimal error with radio buttons and other widgets. • Easy to understand layout as data can be formatted to fit the clients requirements. • Easy to visualise what is happening with graphs and tables. 	<ul style="list-style-type: none"> • More time needed to build the interface and sql database compared to a command based application. • More resources needed from the computer for graphical visualisation and database storage • 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

2.1.2 System flowcharts showing an overview of the complete system

2.2 User Interface Designs

2.3 Program Structure

2.3.1 Top-down design structure charts

2.3.2 Algorithms in pseudo-code for each data transformation process

2.3.3 Object Diagrams

2.3.4 Class Definitions

2.4 Prototyping

2.5 Definition of Data Requirements

2.5.1 Identification of all data input items

2.5.2 Identification of all data output items

2.5.3 Explanation of how data output items are generated

2.5.4 Data Dictionary

2.5.5 Identification of appropriate storage media

2.6 Database Design

2.6.1 Normalisation

ER Diagrams

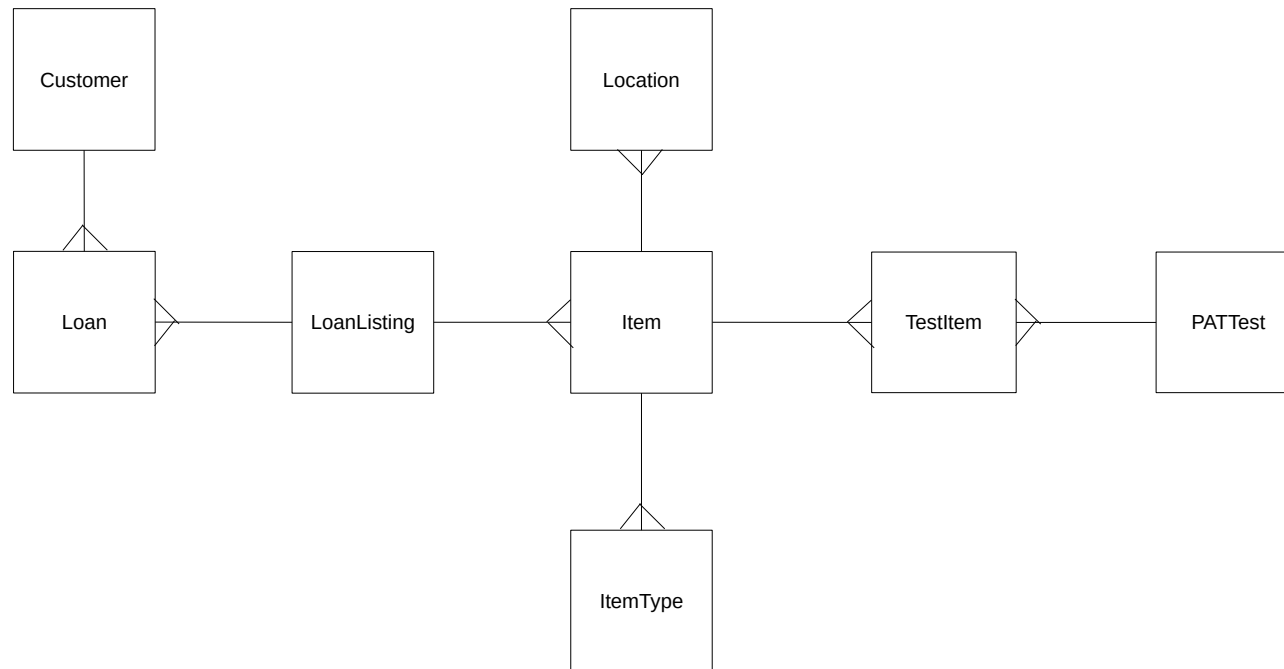


Figure 2.1: ER Diagrams.

Entity Descriptions

Un-Normalised Form(UNF)
ItemID
ItemName
ItemType
Location
Value
ItemQuantity
SubTotal
OnLoan
LoanID
LoanQuantity
LoanRate
LoanLength
LoanCost
CustomerID
Forename
Lastname
Company
Street
Town
PostCode
MobileNumber
LandLine
Email
PATtestID
TestResult
TestDate
ItemDescription
ItemClass
FuseRating
TestUsed
ProtectiveCondTest
InsulationTest
Leakage

1NF to 3NF

First-Normalised Form(1NF)	
Non-Repeating	Repeating
<u>ItemID</u>	<u>LoanID</u>
ItemName	<i>ItemID</i>
ItemType	LoanQuantity
ItemLocation	LoanRate
Value	LoanLength
ItemQuantity	LoanCost
SubTotalValue	CustomerID
OnLoan	Forename
	Lastname
	Company
	Street
	Town
	PostCode
	MobileNumber
	Landline
	Email
	PATtestID
	TestDate
	ItemDescription
	ItemClass
	FuseRating
	TestUsed
	ProtectiveCondTest
	InsulationTest
	Leakage
	TestResult

2.7 Security and Integrity of the System and Data

2.7.1 Security and Integrity of Data

2.7.2 System Security

2.8 Validation

2.9 Testing

2.9.1 Outline Plan

Test Series	Purpose of Test Series	Testing Strategy	Strategy Rationale
Example	Example	Example	Example

2.9.2 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

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 52

4.5.3 Database Table Views

4.5.4 Database SQL

4.5.5 SQL Query

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

55

Question 2

5.3.4 Saving

5.3.5 Finishing

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