

Plastering Management Application

Kyle Kirkby

October 2, 2014

Contents

1	Analysis	5
1.1	Introduction	5
1.1.1	Client Identification	5
1.1.2	Define the current system	5
1.1.3	Describe the problems	6
1.1.4	Section appendix	6
1.2	Investigation	8
1.2.1	The current system	8
1.2.2	The proposed system	24
1.3	Objectives	24
1.3.1	General Objectives	24
1.3.2	Specific Objectives	25
1.3.3	Core Objectives	25
1.3.4	Other Objectives	26
1.4	ER Diagrams and Descriptions	28
1.4.1	ER Diagram	28
1.4.2	Entity Descriptions	29
1.5	Object Analysis	30
1.5.1	Object Listing	30
1.5.2	Relationship diagrams	30
1.5.3	Class definitions	30
1.6	Other Abstractions and Graphs	30
1.7	Constraints	30
1.7.1	Hardware	30
1.7.2	Software	30
1.7.3	Time	31
1.7.4	User Knowledge	31
1.7.5	Access restrictions	31
1.8	Limitations	31
1.8.1	Areas which will not be included in computerisation	31
1.8.2	Areas considered for future computerisation	31
1.9	Solutions	32
1.9.1	Alternative solutions	32

1.9.2	Justification of chosen solution	34
2	Design	36
2.1	Overall System Design	37
2.1.1	Short description of the main parts of the system	37
2.1.2	System flowcharts showing an overview of the complete system	37
2.2	User Interface Designs	37
2.3	Program Structure	37
2.3.1	Top-down design structure charts	37
2.3.2	Algorithms in pseudo-code for each data transformation process	37
2.3.3	Object Diagrams	37
2.3.4	Class Definitions	37
2.4	Prototyping	37
2.5	Definition of Data Requirements	37
2.5.1	Identification of all data input items	37
2.5.2	Identification of all data output items	37
2.5.3	Explanation of how data output items are generated	37
2.5.4	Data Dictionary	37
2.5.5	Identification of appropriate storage media	37
2.6	Database Design	37
2.6.1	Normalisation	37
2.7	Security and Integrity of the System and Data	37
2.7.1	Security and Integrity of Data	37
2.7.2	System Security	37
2.8	Validation	37
2.9	Testing	37
2.9.1	Outline Plan	38
2.9.2	Detailed Plan	38
3	Testing	39
3.1	Test Plan	39
3.1.1	Original Outline Plan	40
3.1.2	Changes to Outline Plan	40
3.1.3	Original Detailed Plan	40
3.1.4	Changes to Detailed Plan	40
3.2	Test Data	41
3.2.1	Original Test Data	41
3.2.2	Changes to Test Data	41
3.3	Annotated Samples	41
3.3.1	Actual Results	41
3.3.2	Evidence	41
3.4	Evaluation	42
3.4.1	Approach to Testing	42
3.4.2	Problems Encountered	42

3.4.3	Strengths of Testing	42
3.4.4	Weaknesses of Testing	42
3.4.5	Reliability of Application	42
3.4.6	Robustness of Application	42
4	System Maintenance	43
4.1	Environment	44
4.1.1	Software	44
4.1.2	Usage Explanation	44
4.1.3	Features Used	44
4.2	System Overview	44
4.2.1	System Component	44
4.3	Code Structure	44
4.3.1	Particular Code Section	44
4.4	Variable Listing	44
4.5	System Evidence	44
4.5.1	User Interface	44
4.5.2	ER Diagram	44
4.5.3	Database Table Views	44
4.5.4	Database SQL	44
4.5.5	SQL Queries	44
4.6	Testing	44
4.6.1	Summary of Results	44
4.6.2	Known Issues	44
4.7	Code Explanations	44
4.7.1	Difficult Sections	44
4.7.2	Self-created Algorithms	44
4.8	Settings	44
4.9	Acknowledgements	44
4.10	Code Listing	44
4.10.1	Module 1	45
5	User Manual	46
5.1	Introduction	47
5.2	Installation	47
5.2.1	Prerequisite Installation	47
5.2.2	System Installation	47
5.2.3	Running the System	47
5.3	Tutorial	47
5.3.1	Introduction	47
5.3.2	Assumptions	47
5.3.3	Tutorial Questions	47
5.3.4	Saving	47
5.3.5	Limitations	47
5.4	Error Recovery	47
5.4.1	Error 1	47

5.4.2	Error 2	47
5.5	System Recovery	47
5.5.1	Backing-up Data	47
5.5.2	Restoring Data	47
6	Evaluation	48
6.1	Customer Requirements	49
6.1.1	Objective Evaluation	49
6.2	Effectiveness	49
6.2.1	Objective Evaluation	49
6.3	Learnability	49
6.4	Usability	49
6.5	Maintainability	49
6.6	Suggestions for Improvement	49
6.7	End User Evidence	49
6.7.1	Questionnaires	49
6.7.2	Graphs	49
6.7.3	Written Statements	49

Chapter 1

Analysis

1.1 Introduction

1.1.1 Client Identification

My client is 30 year old plasterer Dan Austin who runs his own plastering business known as DnA Plastering. Dan mainly uses his Toshiba laptop (Dual Core Intel with 6 GB Ram and running Windows 8 64 bit) to do basic tasks such as social networking and receiving/sending emails.

The current system is a paper based method where he records the prices and measurements of the plastering/screening/rendering jobs he undertakes. Dan works in an around the Suffolk/Essex area but occasionally takes on larger jobs further afield in places such as London or Epping. All the recording and calculations are done by Dan himself and does not require additional assistance in completing these tasks but is looking for a digital solution to the organisation problems faced with the current manual paper method.

Dan is looking to introduce a computer based system to replace the current one in order to make keeping track of jobs and pricing up new jobs easier and more efficient. Alongside this he would like to be able to keep information on all of his customers so he can simply search for clients' details and contact information all in one location. He will also be able to look up the jobs that he has done for them to make sending invoices easier and manageable.

1.1.2 Define the current system

The current system in place is a paper/notebook based system where details of clients are stored along with prices of jobs and cost of materials needed etc.

The details of the clients include their address, phone number, email, first name and surname. The information about the job usually includes the measurements of what needs to be plastered along with how long it will take to complete and if he is taking any labourers to too. Calculations are often also made to work out how much to charge depending on the price he is charging per square meter. This rate often changes depending on the current economy.

Once all the calculations are made, he works out how much the materials are going to cost and also how long it will take him to complete the job. Once all these calculations and prices have been evaluated he notifies the client of the price; when the price is confirmed the job is undertaken.

Finally, Dan writes out an invoice using a standard invoice book purchased from a stationary store to inform the client of the costs and charges of the job. The current folder containing the invoices for his clients is not organised and offers another problem whereby finding information for jobs is difficult due to the inability to search quickly for any given customer.

1.1.3 Describe the problems

Problems are plentiful in the current system. One of the main problems is keeping valuable client data from being lost or damaged as there is only one hard copy made in a notebook. Another problem with the notebook is not being able to easily search through the details of all the clients to find specific phone numbers or contact details. Using a computer based system would allow Dan to search through his clients efficiently and allow him to make backups of the valuable client and job data.

1.1.4 Section appendix

Interview with client

1. What system are you currently using?

I've got a Toshiba laptop with Windows 8, 6GB RAM and a dual core processor

2. Are there any issues with the system currently being employed?

Just organisational issues when it comes to finding clients information etc. Would be good to have a database which I could search for info with.

3. What data do you record at the moment?

The clients name, address, phone number, job information like measurements of the rooms etc.

4. **How often do you record data in the current system and how much data is stored each time?**

A few times a week and not much, a few lines in the workbook I use.

5. **What happens to older client information?**

Normally gets lost as and when I replace any of my books I write the details down in.

6. **Are there any storage issues when it comes to storing data manually?**

No, nothing really as I only need room for a book or two.

7. **Are changes often made to existing records of client data or job info?**

Occasionally when a client gives me a new contact number but nothing major.

8. **What is the typical routine when it comes to gaining a new client or job?**

Normally I get a call from a colleague who gives me the details about a job which I can take on or not depending on the time I have available. Sometimes clients will find my number through my website dnaplastering.co.uk or through word of mouth and will ask me to give them a quote on a job. Then I will go to the job and work out any costs involved and the time it would take to complete. Then I can arrange a time to do the job and after the job is completed I will usually write out an invoice for the client telling them about the cost for materials,labouring and profit etc.

9. **What does the client recieve in terms of invoices?**

I write out an invoice and give it to them in person after the job is completed.

10. **What format should the outputs be in within the new system?**

It would be good to be able to print the clients invoices out with minimal effort involved and possibly email the clients the invoice.

11. **Will you need to print reports and invoices or will it be sent entirely digitally to the client via email?**

I definitely need to print out some of the invoices as most of my work is done face to face so it is easier to give it to them in person. It might be

useful to email it to them as well.

12. **Is there a security issue in regards to the data you store in the current system?**

Not much of an issues as it is just client information but added security would be a bonus.

13. **Are there any foreseeable constraints required in the proposed system?**

None that I am aware of.

14. **Do you have a picture in mind of what the new system could look like?**

Not really bothered about the look of it too much anything that works as expected would be brilliant.

Signed:

A handwritten signature in blue ink, appearing to be 'Dan', written on a piece of lined paper.

1.2 Investigation

1.2.1 The current system

Data sources and destinations

There are four main data sources within the current system - The plasterer, the client, the builders merchant and visiting the clients job. A client contacts Dan through a phone call placed to Dan's mobile. Sometimes a client may leave Dan a voicemail message if he is too busy to answer the call at that given moment. If this is the case then Dan will get back to the client as soon as possible. Most of the data in the current system will come from the client or the clients job - this data will be the job measurements and the clients contact information. The main data destinations are the forms given to the client i.e the quote and the invoice document.

Source	Data	Example Data	Destination
Client	Client Contact information First-name - Lastname - phoneNumber - AddrLine1 - AddrLine2 - AddrLine3 - AddrLine4 - PostCode - Email - JobType	John - Smith - 07809726812 - 15 - Crowley Road - Haverhill - Suffolk - CB90DJ - john@gmail.com - Plastering Bedroom	Appointment and Client Book.
Plasterer	Appointment Time and Place	16:00 at 15 Crowley Road, Haverhill	Client Calendar or Diary
Visiting Job Site	Measurements of Job Size and Materials that need to be purchased	4m x 5m x 3m = $60m^2$ 10 Bags of Plaster	Work Notebook
Plasterers Calculations	Quote for the work that needs doing and agree a date it can be done.	£600, 1 Day, 15th October	Quote written out on paper or agree in person.
Plasterers calculations for the materials needed for the job	Quantity of materials needed for the job	25 bags of plaster and 12m of angle beading	Builders Merchant
Builders Merchant	A price for the materials needed	£350 for the bags of plaster and angle beading	Plasterer
Plasterer	Total cost of the job broken down - cost of parts,labouring and vat. Date of Job	£600 - £350 materials - £50 VAT - 14/08/14	Client.

Algorithms

There are three main algorithms utilised in the current system. The first is an algorithm to agree the price of the job with the client.

Algorithm 1 Agreeing a price Algorithm

```
1: SET agreed TO false
2: WHILE agreed = False DO
3:   IF "Client does not agree with quoted price" THEN
4:     Discuss price and change quote if new price is agreed upon.
5:   ELSE
6:     SET agreed TO true
7:   Arrange a date for the work to be started on.
8:   END IF
9: END WHILE
```

The second algorithm currently being used in the system is an algorithm used to see whether the work is completed.

Algorithm 2 Checking whether work is complete or not.

```
1: SET Complete TO False
2: WHILE Complete = False DO
3:   IF "Issue/problem not fixed." THEN
4:     Check the current problem and fix issue.
5:   ELSE
6:     SET Complete TO True
7:   END IF
8: END WHILE
9: Create and send invoice
```

The third algorithm being used in the system is an algorithm used to see whether the work has been paid for completely.

Algorithm 3 Checking whether work has been paid for yet.

```
1: SET Paid TO False
2: WHILE Paid = False DO
3:   IF "Money has not been given." THEN
4:     Send invoice and contact client
5:   ELSE
6:     SET Paid TO True
7:   END IF
8: END WHILE
9: Update job to paid for in book.
```

Data flow diagrams

Key

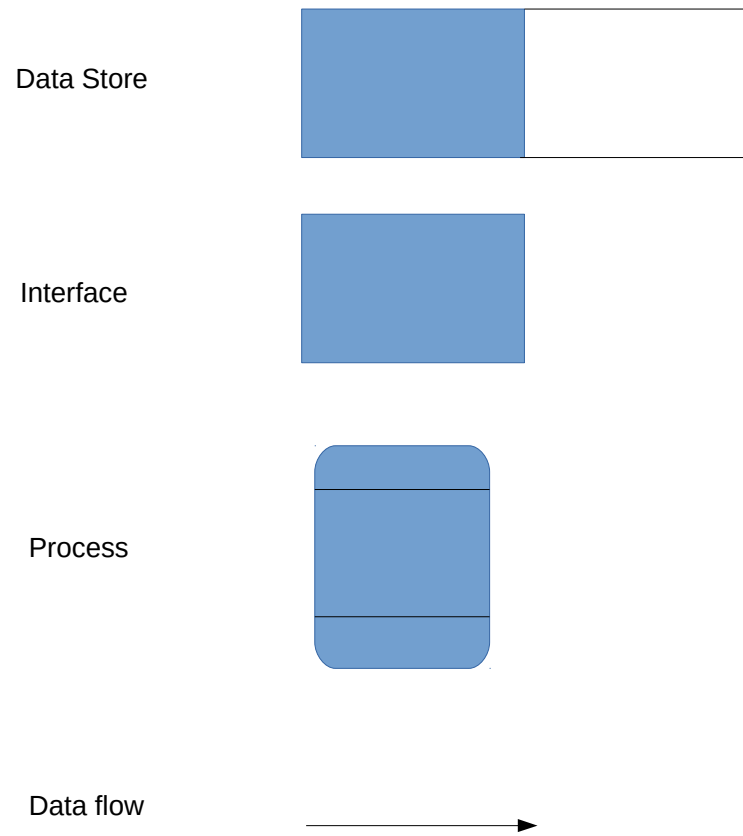


Figure 1.1: This is the Key to be used for the following data flow diagrams.

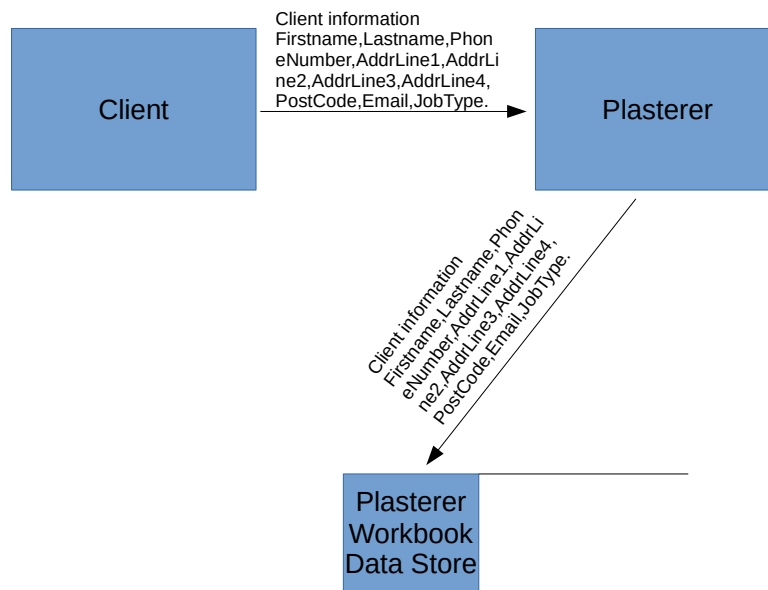


Figure 1.2: This diagram shows the flow of data when gaining a new clients information.

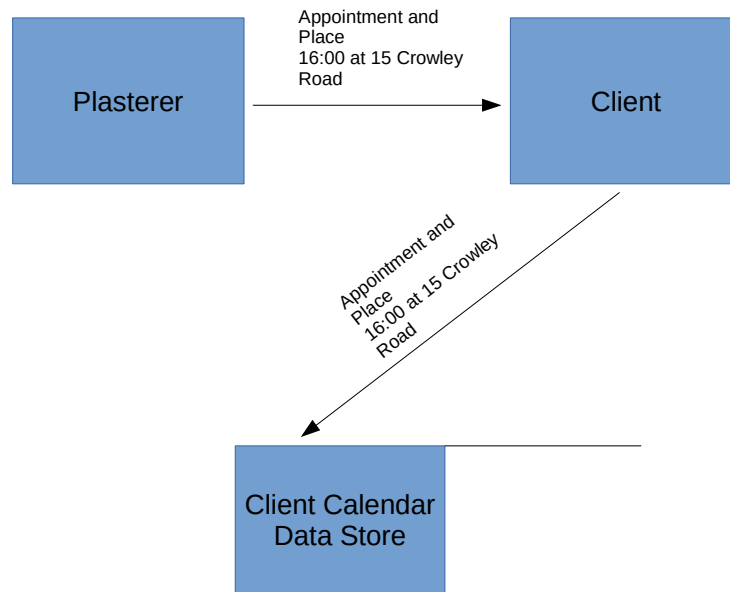


Figure 1.3: The signifies the flow of data when making an appoint for a job.

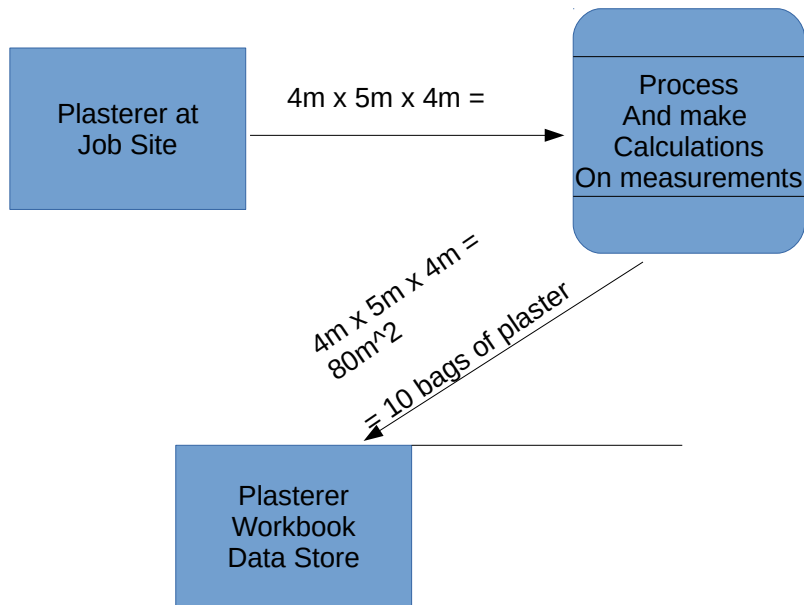


Figure 1.4: This diagram shows the flow of data when collecting the measurements for a job.

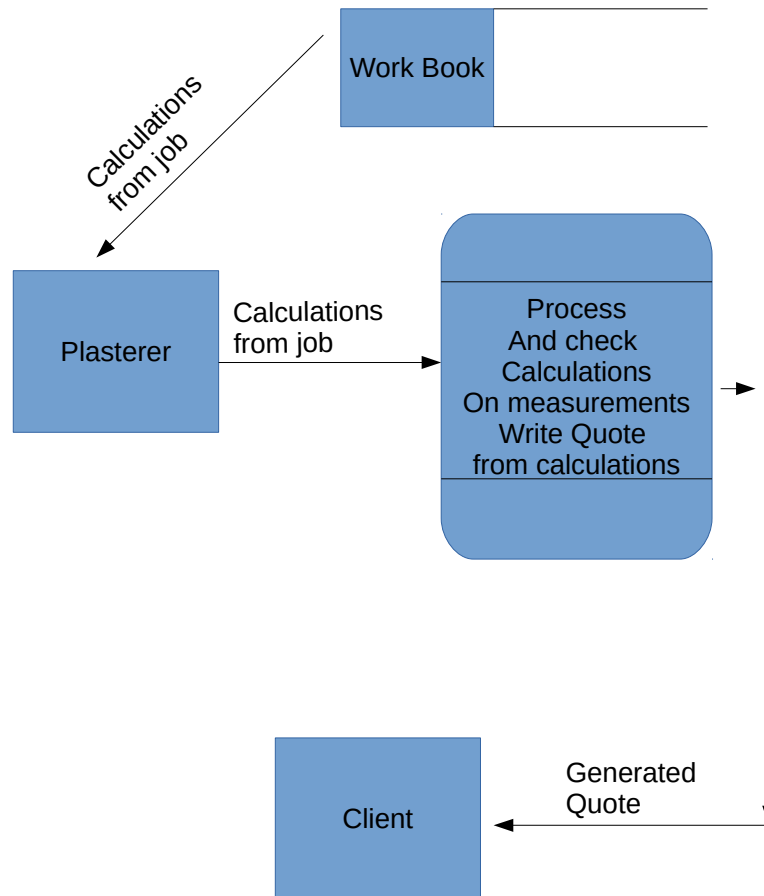


Figure 1.5: The flow of data when generating a quote for the client.

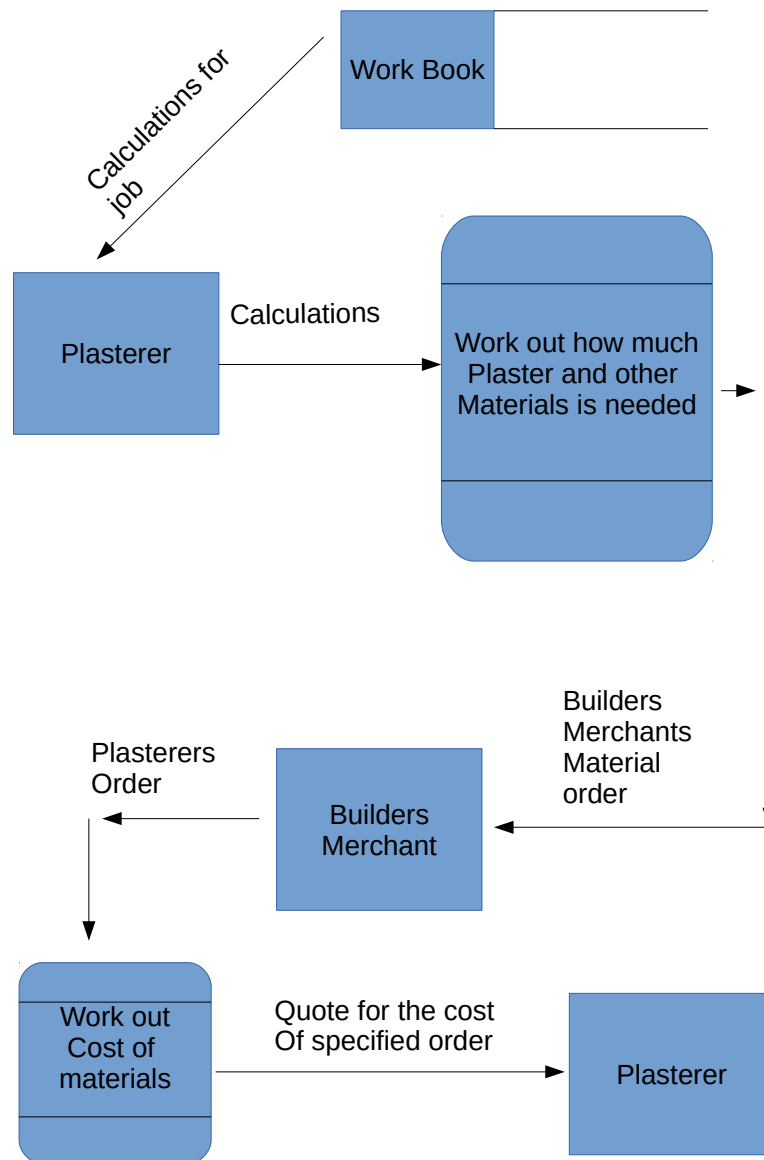


Figure 1.6: This shows the data flow when getting a quote for the materials.

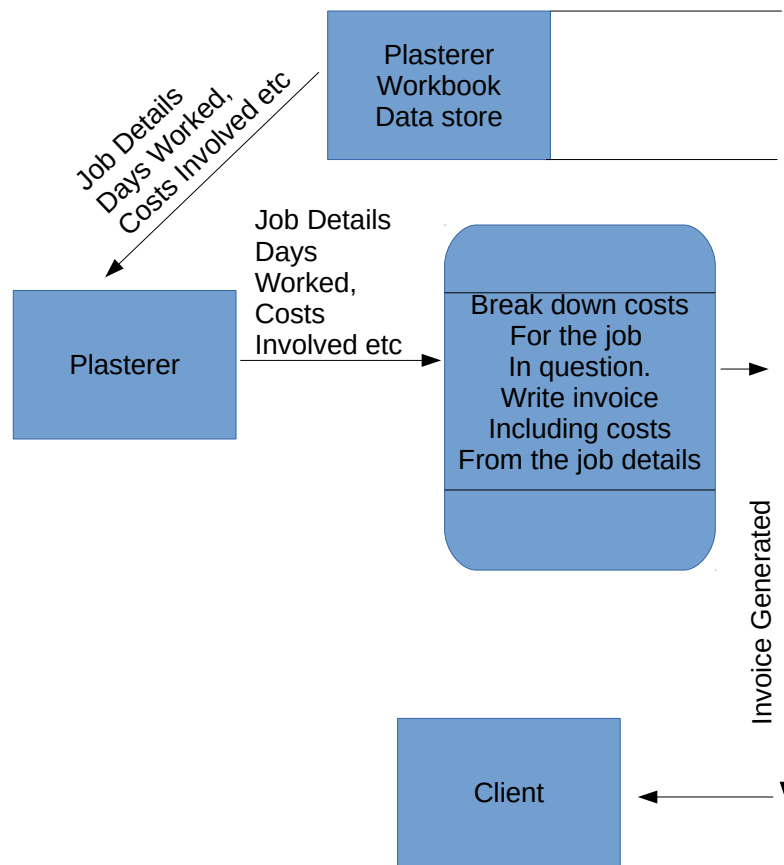


Figure 1.7: The Data flow when an invoice is given to the client.

Input Forms, Output Forms, Report Formats

day	1st	0715	houlston 11.45	—	—	—
4th	Dell	School	£180	cash		
5th	Day	off				
6th	Dell	floor halstead	£180	cash		
7th	kelvin		£100	cash		
8th	Day	off				
11th	Day	off				
12th	Paul	trinder				
13th	kelvin		£100	cash		
14th	Dell	floor little Brook	£180	cash		
15th	Justin	Out side work Riggwell	£200	cash		
Monday	Justin	"	£200	cash		
Tuesday 19th	Justin	out side work	£100	cash		
20th	Day	off				
21st	Sue	osden	Cash £200	"		"
22nd	Sue	osden	Cash £200	not Tax paid		
25th	Bank	holiday Monday				
26th	Day	off				
27th						
Wednesday 28th	Dell	halstead	1/2 day	£90		
29	Day off	Dell	halstead	£180		

Figure 1.8: This is a page from a work book used to store details about jobs.

Figure 1.9: This is an example of an invoice output form which is given to clients.

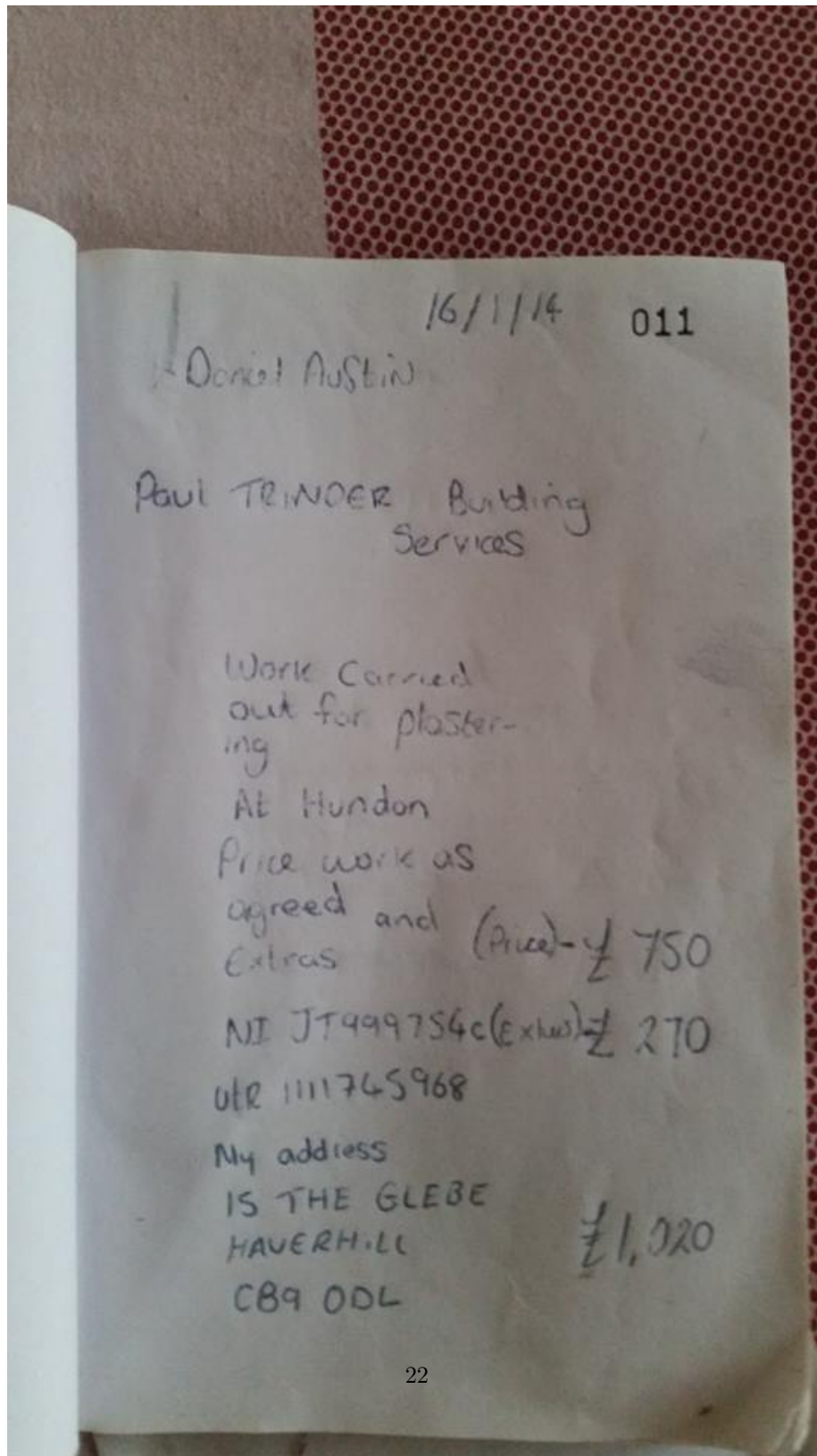


Figure 1.10: Here is an illustration of the type of information which goes on an invoice to the client.

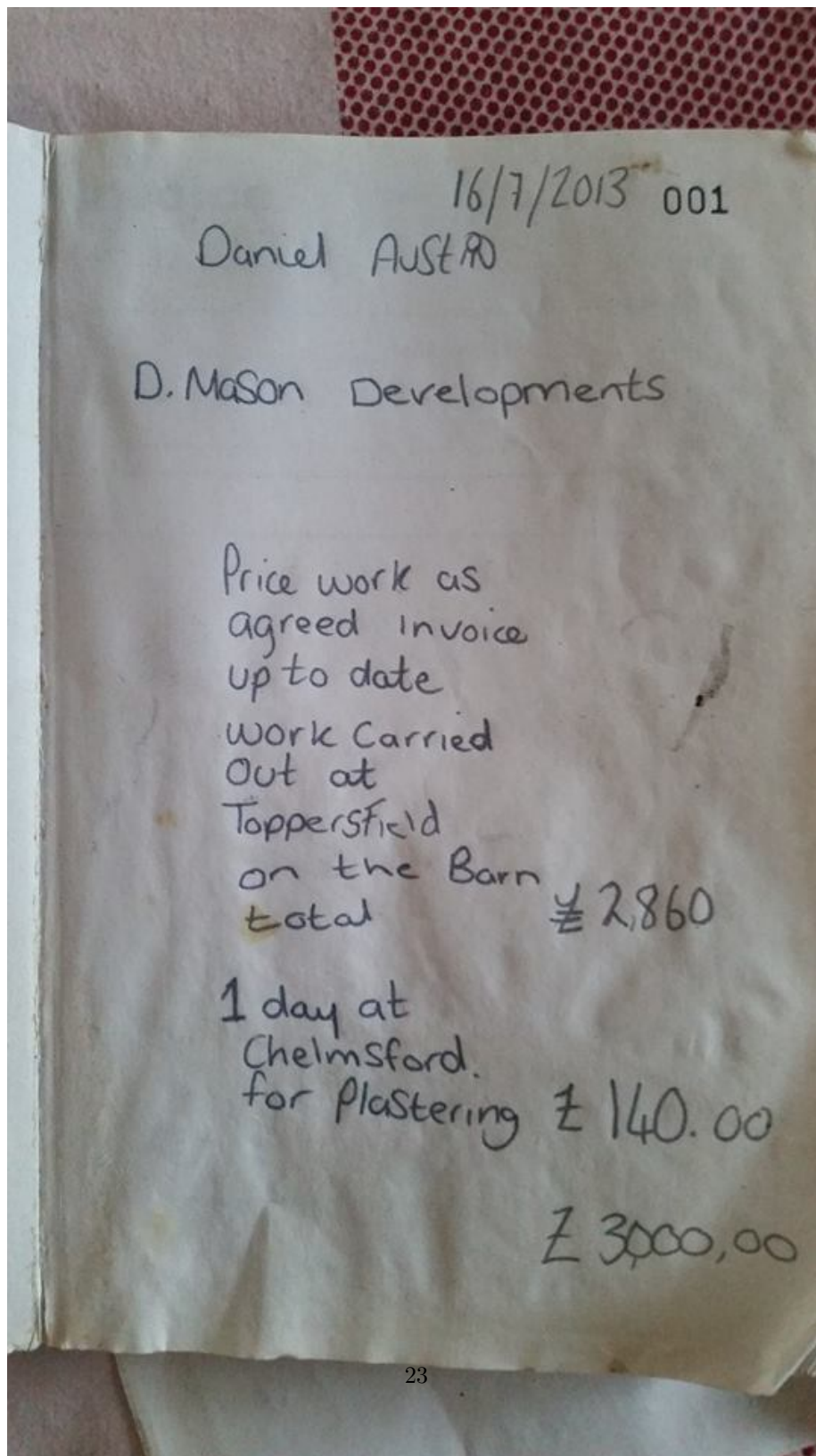


Figure 1.11: This is also another example of an invoice which is given to a client.

1.2.2 The proposed system

Data sources and destinations

Data flow diagrams

Data dictionary

Volumetrics

In the current system Dan only has a few regular clients but it is possible to receive up to 4-5 additional clients each week (mentioned in interview question 8). Therefore using the guideline of around 4 clients a week and a trial period for the application of 3 months, there could be upto 48 new clients (4 clients a week x 4 weeks in month x 3 months = 48) added into the database. It would also be useful to add the existing clients from paper so another 50 clients could be added from what is already stored manually. So around 100 clients may be added to the new system. This size can be increased later if necessary.

In the job details table each client may have up to 10 or more jobs. Some clients will only have one and some will be recurring customers so will have a few more. In the calculations for minimum storage space I will say that each field takes up 1KB of hard disk space and all together there will be around 50 related fields for each client so:

- $100 \times 50\text{KB} = 5000 \text{ KB}$
- $5000 \text{ KB} / 1024 = 4.8\text{MB}$

s If the application took up around 5MB itself then the total space required for the proposed system would be around 9.8MB ($4.8\text{MB} + 5\text{MB} = 9.8\text{MB}$). Dans computer has plenty of hard disk space that could be used when installing the application. This proposed system would therefore have enough storage space for 100 clients to be added to the client database.

1.3 Objectives

1.3.1 General Objectives

- Clean and easy to use GUI.
- Use a database for storing data.
- Make it as easy as possible to find data.
- Add clients to database.
- Make it easy to calculate costs involved.

- Sort client data and add search functionality.
- Ability to add multiple jobs per client.

1.3.2 Specific Objectives

Client Data Store Objectives

- Ability for Dan to add a client to a database.
- Ability for Dan to be able to delete a client from the database if needed.
- Dan should be able to modify and append existing clients details with an easy to use system.
- A search feature that will let Dan filter the database of clients to find vital information.

Jobs Objectives

- Dan will be able to add a job which will relate to a client stored in the client entity.
- Multiple jobs can be added.
- The jobs will contain the job details (description) and address etc (see entity descriptions for more details).
- Each job will be able to generate invoices from the data.
- The invoice will be able to be sent to the client digitally (by email) or manually (print invoice).
- The user will have the ability to edit the details if needed for each job.

Management features objectives.

- Dan will be able to generate reports between x and y time periods in order to see the amount earned within that period.
- A feature will be implemented whereby the user can deduct tax and other specified costs from the amount earned within that period.
- Ability to print this pay report and generate a digital versatile copy of the report.

1.3.3 Core Objectives

- The application must store the client details in a database.
- The application must be able to add jobs for each client.
- The application must be able to modify client and job details.

- The application must be able to send an invoice to the client via email.
- The application must be able to generate a report for the amount earned within a time period.

1.3.4 Other Objectives

- The application may be able to print an invoice for the client.
- The application may be able to send quotes for jobs via email.
- The application may be able to print a report for the amount earned within a time period.

1.4 ER Diagrams and Descriptions

1.4.1 ER Diagram

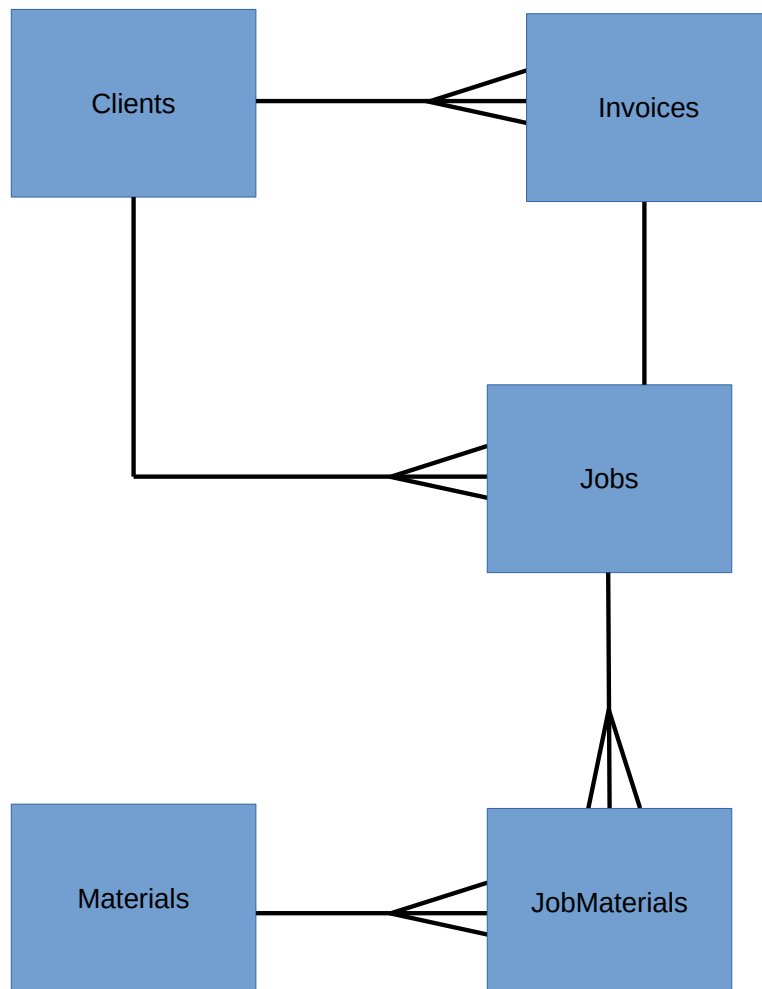


Figure 1.12: This is the entity relationship diagram for the sqlite3 database.

1.4.2 Entity Descriptions

Entities

- Client
- Jobs
- Materials
- JobsMaterials
- Invoices

Below are the entity descriptions for the various entites in the proposed system. An underlined attribute denotes a primary key and a *emphasised* attribute signifies a foreign key in the entity.

Client entity

Client(ClientID, ClientTitle, ClientFirstName, ClientSurname,
ClientAddrLine1, ClientAddrLine2, ClientAddrLine3, ClientAddrLine4,
ClientEmail, ClientPhoneNumber)

Jobs entity

Job(JobID, *ClientID*, JobDescription, JobAddrLine1, JobAddrLine2,
JobAddrLine3, JobAddrLine4, JobComplete, JobPaid)

Materials entity

Materials(MaterialsID,MaterialName,MaterialPrice)

JobsMaterials entity

JobsMaterials(JobMaterialsID, *JobID*, *MaterialsID*, JobMaterialsQuantity)

Invoices entity

Client(InvoiceID, *ClientID*, *JobID*, InvoiceAmountPreTax,
InvoiceAmountAfterTax, InvoiceReceived, InvoiceDateSent)

1.5 Object Analysis

1.5.1 Object Listing

1.5.2 Relationship diagrams

1.5.3 Class definitions

1.6 Other Abstractions and Graphs

1.7 Constraints

1.7.1 Hardware

Dan currently has a Toshiba Laptop with the following hardware components:

- 6GB DDR3 RAM
- Intel 2.0ghz Dual Core Processor
- High Resolution 1920 x 1080 Display
- 1 TB HDD
- Intel Onboard Integrated Graphics

The specification of Dan's laptop is more than powerful enough and has enough RAM to run the proposed python application alongside multiple other programs that he regularly uses (such as a Web Browser and Media Player). The hard drive has enough storage space to install the application (which will only be around 10MB) so storage is not a problem. The only possible complication when it comes to hardware may be the screen resolution as the application will have to keep within the resolution and be optimized for this screen size and many more to keep the software versatile.

1.7.2 Software

Dan is running the Windows 8 operating system on his laptop. The proposed system will need to be able to run on this operating system which is not a problem as the proposed system can easily be packaged into a installation executable which he will be able to run and install the proposed system. This will remove the constraint of not having python natively installed on the system.

1.7.3 Time

Dan is very flexible when it comes to time as he is a freelancer and is in no rush for a replacement to his old system. He does not mind how long the new system takes to be implemented just that it works and functions as expected. Therefore the only time constraint for the proposed system would be the coursework deadline which is Friday 27th March 2015.

1.7.4 User Knowledge

Dan has no formal qualifications in IT and has never performed complex computerised tasks. Browsing the internet, watching films and playing music is the extent to which Dan currently uses his laptop. Therefore user knowledge is a possible constraint and the proposed system will need to be easy to navigate and use. This may be achieved by providing a simple non-complex GUI and detailed tutorials/documentation.

1.7.5 Access restrictions

The proposed system will not need additional security measures as no one else will be using the system and Dan is the only person that will use the application so different user accounts will not be needed.

1.8 Limitations

1.8.1 Areas which will not be included in computerisation

1.8.2 Areas considered for future computerisation

1.9 Solutions

1.9.1 Alternative solutions

Solution	Advantages	Disadvantages
Web Application	Can be accessed on multiple devices where there is an internet connection. Only a web browser required (already installed in most cases). Can be easier to make it look "pretty" as there are many graphical libraries for the web. Can use different server side languages to program the backend of the website. (Python (Django), Ruby (RoR), PHP (Zend, Magento)).	Need an internet connection to visit site if not hosted locally. A web host may need to be purchased. Need to optimize for different browsers (Internet Explorer, Chrome, Firefox, Safari). Problems may happen with host which means the client may not be able to access their application (site may be subject to DDOS (may need to purchase Cloudflare or similar service)).
Mobile Phone Application	Can mean the application is extremely mobile so the client will be able to take the app wherever he goes which may be useful to take to jobs. Client is familiar with using a mobile phone so may be easier for them to use.	Will take longer to complete. It is more complex and can be solved easier with a different method. Client may lose phone which would then result in loss of application.
Command-Line Application	No long waiting time for GUI to load. System resources are not used up as much.	Can be more complicated, so it seems to the client. Interface would be less effective and harder to use for the non-technical client. Errors may occur and look unfamiliar to client. If laptop is lost/stolen the program may be lost too so backups need to be kept online (not so much a problem with a web application).

Spreadsheet	Can be faster than a application with a GUI to load. Data can be exported easily. Queries are easy to run and execute fast.	No abstraction from the data. Client would have to familiarise themselves with the syntax for queries. You can add a lot more features in PyQt and Python Applications. No GUI to make it easier for the client to control the functions and run queries.
PyQt4 GUI and Python3 Application	Python is easy to write and can therefore get the project done quicker leaving more time for debugging and testing. The GUI will make it a lot easier for the client to use the application. With this form of application it can be packaged for Mac, Windows and Linux Systems. GUI gives a clear visual representation of the applications features.	Uses python which needs to be packaged with the program as windows does not have Python natively installed but Mac and Linux do. Writing a GUI is a lot more work than other solutions.

1.9.2 Justification of chosen solution

I have chosen to use Python 3 alongside the Qt Framework (PyQt4) to create a standalone application. This is due to the versatility which Python and PyQt4 offers. Using this method to program a solution to my clients problem I will be able to get it done efficiently due to my existing knowledge of Python and I will be able to generate different executables for different operating systems which may be beneficial to the client as he is looking to buy a Apple System. It will also offer a visual representation through the use of GUI as opposed to hard to use Console application approach; which would not be suitable for the client as he does not have enough technical knowledge with computers to understand and use it easily - therefore going for something with a GUI would be a good choice. A web application may have been good but

sometimes Dan takes his laptop where there is no internet connection and hosting the site locally would not be feasible. Also there are hidden costs involved with purchasing a domain and hosting if a web application solution was chosen. A mobile application may have also been good but would have taken longer to accomplish due to the time it takes to develop complex mobile phone applications. A spreadsheet would also not have been the perfect solution as it requires knowledge of writing queries which, due to Dan's lack of technical computer knowledge, he does not have. Therefore I believe that a Python and PyQt4 application would be the best solution to go for.

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

2.9.1 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

4.5.3 Database Table Views

4.5.4 Database SQL

44

4.5.5 SQL Queries

4.6 Testing

4.6.1 Screenshots of Results

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

5.3.5 Limitations

5.4 Error Recovery

5.4.1 Error 1

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