

Computing Coursework

Henry Mason

December 5, 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	5
1.1.4 Section appendix	6
1.2 Investigation	7
1.2.1 The current system	7
1.2.2 The proposed system	9
1.3 Objectives	11
1.3.1 General Objectives	11
1.3.2 Specific Objectives	11
1.3.3 Core Objectives	11
1.3.4 Other Objectives	12
1.4 ER Diagrams and Descriptions	12
1.4.1 ER Diagram	12
1.4.2 Entity Descriptions	13
1.5 Object Analysis	13
1.5.1 Object Listing	13
1.5.2 Relationship diagrams	14
1.5.3 Class definitions	15
1.6 Constraints	15
1.6.1 Hardware	15
1.6.2 Software	16
1.6.3 Time	16
1.6.4 User Knowledge	16
1.6.5 Access restrictions	16
1.7 Limitations	17
1.7.1 Areas which will not be included in computerisation	17
1.7.2 Areas considered for future computerisation	17
1.8 Solutions	18
1.8.1 Alternative solutions	18
1.8.2 Justification of chosen solution	18

2 Design	19
2.1 Overall System Design	19
2.1.1 Short description of the main parts of the system	19
2.1.2 System flowcharts showing an overview of the complete system	21
2.2 User Interface Designs	21
2.3 hardware specification	21
2.4 Program Structure	21
2.4.1 Top-down design structure charts	21
2.4.2 Algorithms in pseudo-code for each data transformation process	21
2.4.3 Object Diagrams	21
2.4.4 Class Definitions	21
2.5 Prototyping	21
2.6 Definition of Data Requirements	21
2.6.1 Identification of all data input items	21
2.6.2 Identification of all data output items	21
2.6.3 Explanation of how data output items are generated	21
2.6.4 Data Dictionary	21
2.6.5 Identification of appropriate storage media	22
2.7 Database Design	22
2.7.1 Normalisation	22
2.8 Security and Integrity of the System and Data	27
2.8.1 Security and Integrity of Data	27
2.8.2 System Security	27
2.9 Validation	27
2.10 Testing	27
2.10.1 Outline Plan	28
2.10.2 Detailed Plan	28
3 Testing	29
3.1 Test Plan	29
3.1.1 Original Outline Plan	30
3.1.2 Changes to Outline Plan	30
3.1.3 Original Detailed Plan	30
3.1.4 Changes to Detailed Plan	30
3.2 Test Data	31
3.2.1 Original Test Data	31
3.2.2 Changes to Test Data	31
3.3 Annotated Samples	31
3.3.1 Actual Results	31
3.3.2 Evidence	31
3.4 Evaluation	32
3.4.1 Approach to Testing	32
3.4.2 Problems Encountered	32
3.4.3 Strengths of Testing	32

3.4.4	Weaknesses of Testing	32
3.4.5	Reliability of Application	32
3.4.6	Robustness of Application	32
4	System Maintenance	33
4.1	Environment	34
4.1.1	Software	34
4.1.2	Usage Explanation	34
4.1.3	Features Used	34
4.2	System Overview	34
4.2.1	System Component	34
4.3	Code Structure	34
4.3.1	Particular Code Section	34
4.4	Variable Listing	34
4.5	System Evidence	34
4.5.1	User Interface	34
4.5.2	ER Diagram	34
4.5.3	Database Table Views	34
4.5.4	Database SQL	34
4.5.5	SQL Queries	34
4.6	Testing	34
4.6.1	Summary of Results	34
4.6.2	Known Issues	34
4.7	Code Explanations	34
4.7.1	Difficult Sections	34
4.7.2	Self-created Algorithms	34
4.8	Settings	34
4.9	Acknowledgements	34
4.10	Code Listing	34
4.10.1	Module 1	35
5	User Manual	36
5.1	Introduction	37
5.2	Installation	37
5.2.1	Prerequisite Installation	37
5.2.2	System Installation	37
5.2.3	Running the System	37
5.3	Tutorial	37
5.3.1	Introduction	37
5.3.2	Assumptions	37
5.3.3	Tutorial Questions	37
5.3.4	Saving	37
5.3.5	Limitations	37
5.4	Error Recovery	37
5.4.1	Error 1	37
5.4.2	Error 2	37

5.5	System Recovery	37
5.5.1	Backing-up Data	37
5.5.2	Restoring Data	37
6	Evaluation	38
6.1	Customer Requirements	39
6.1.1	Objective Evaluation	39
6.2	Effectiveness	39
6.2.1	Objective Evaluation	39
6.3	Learnability	39
6.4	Usability	39
6.5	Maintainability	39
6.6	Suggestions for Improvement	39
6.7	End User Evidence	39
6.7.1	Questionnaires	39
6.7.2	Graphs	39
6.7.3	Written Statements	39

Chapter 1

Analysis

1.1 Introduction

1.1.1 Client Identification

My client is Susannah Mason, she is 50 years old and has little usage of computers, except when having to order new stock for the pharmacy. currently the pharmacy uses computerised methods to submit orders to the warehouse.

Susannah is a pharmaceutical manager at Spire Healthcare in Impington
by creating this program it would speed up the process making keeping track of
and ordering of new equipment and stock a lot easier for her

1.1.2 Define the current system

The current system uses mostly computer based order submission and price checks but the orders have to be put through the computer manually

1.1.3 Describe the problems

The orders for the stock take too long to submit and all stock has to be counted
by hand

1.1.4 Section appendix

Henry Mason Candidate Number: 3634 centre number: 22151

1.4. Section Appendix

Questionnaire

1. please give a brief description of your job Title?
Pharmacy manager of the village Pharmacy.
I am responsible for dispensing and stock control.
2. please give an explanation on what computer programs you currently use?
Computer system is used to maintain patient records
and produce labels for medication.
3. with the current system in use at the moment,(if any), have you found any problems?
Current stock control system is via a card base system which is labour intensive and makes accurate stock levels difficult.
4. what would you like the program to do?, what are the main function of your current computer program?
I need a programme which will maintain information with regards to the amount of stock held. It would also produce an order at the end of each day.
5. how much computer usage do you have both inside and outside of your worklife?
1 computer at home and 1 at work.

The client confirms that this information is accurate:

date: 24/10/2014

Henry Mason

Figure 1.1: questionnaire

1.2 Investigation

1.2.1 The current system

the current system at the pharmacy is a data base that holds the information of over 500 items. the data base holds the price the mass the desription and how much is in the pharmacy at that point in time. when an item is taken out of stock the pharmacist has a card to say that an item has been removed from the storage cupboard. sometimes the system deosn't update even when the card is swiped to say a product has been removed

Data sources and destinations

Data Source	Travels via	destination
doctor	gives prescription	patient
patient	requests medicine	pharmacist
pharmacist	checks stock	stock system
stock system	gives information	pharmasict
pharmacist	collects medication	medicine cupboard
pharmacist	gives medicine	patient

Table 1.1:

Algorithms

i will be using quite a few algorithms for this assignment

Algorithm 1 if statement

```

1: FOR EACH item to check = 0 to 50 IN
2:   IF THENDOitem = lowest minimum amount
3:     "you don't need any more tablets"
4:   ELSE
5:     "you need more tablets"
6:   END IF
7: END FOR

```

this other algorithm will be used to calculate the exact price of all of the order using the information in the list Items the exact price is calulated

Data flow diagram

```

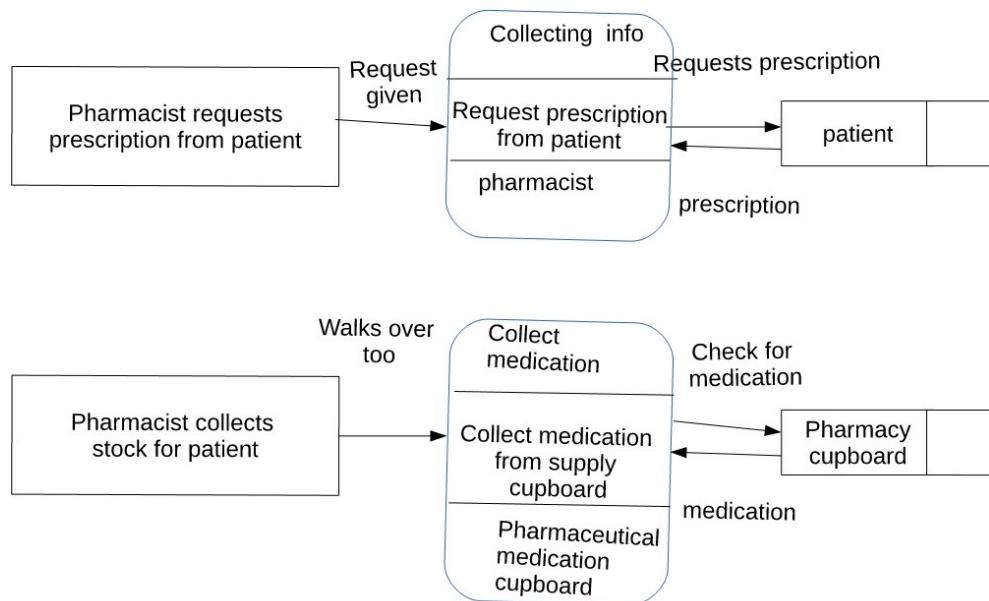
1: IF order Submitted = True THEN
2:   calculate Order
3: ELSE
4:   Restart stock check
5: END IF

```

```

1: IF item in items = True THEN
2:   total ← total + itemPrice
3: ELSE
4:   total ← total
5: END IF

```



Input Forms, Output Forms, Report Formats

Patient details

Hospital prescription: 92078000
Name: Henry Mason
Age: 20
Address: 123 Smith St.
City: Sydney
State: NSW
Postcode: 2000

Prescription

Prescription number: 1234567890
Prescription date: 01/01/2023
Prescription type: New
Prescription status: Pending

Stock levels

Drug name and code	Strength	Stock level and quantity	Quantity	Unit	Stock	Min. level	Max. level
Aspirin 500mg	500mg	500 tablets	50	tablets	50	40	60
Paracetamol 500mg	500mg	500 tablets	50	tablets	50	40	60
Ibuprofen 200mg	200mg	500 tablets	50	tablets	50	40	60
Sleeping tablets	1 tablet	50 tablets	50	tablets	50	40	60
Antibiotic tablets	1 tablet	50 tablets	50	tablets	50	40	60
Contraceptive tablets	1 tablet	50 tablets	50	tablets	50	40	60
Pain relief	1 tablet	50 tablets	50	tablets	50	40	60

Treatment history

Prescriber: Dr A Pratikoski Prescriber number: 123456
Requester: A Practitioner Ref ID: 0004-01
Page number: 1 of 1
Please see notes for previous visits

Notes (optional) for which form will be used for future medication requests:
Prescription of Aspirin 500mg with a history of reduction in pain while on aspirin.

Prescription of Paracetamol 500mg with a history of reduction in pain while on paracetamol.

Prescription of Ibuprofen 200mg with a history of reduction in pain while on ibuprofen.

Prescription of sleeping tablets with a history of reduction in pain while on sleeping tablets.

Prescription of antibiotic tablets with a history of reduction in pain while on antibiotic tablets.

Prescription of contraceptive tablets with a history of reduction in pain while on contraceptive tablets.

Prescription of pain relief with a history of reduction in pain while on pain relief.

pres.JPG

1.2.2 The proposed system

the proposed system will be used to order, check stock and be informed as soon as anything leaves the pharmacy the data base will be updated of the removal, as well as if the product falls below a certain point it will be program to replace the stock by ordering new stock form the wearhouse automatically but the order will go through a master control point before being sent off

Data sources and destinations

Data Source	Travels via	destination
doctor	sends email	pharmacy
pharmacist	checks stock	stock system
stock system	gives information	pharmacist
pharmacist	collects medication	medicine cupboard
patient	collects from	pharmacy

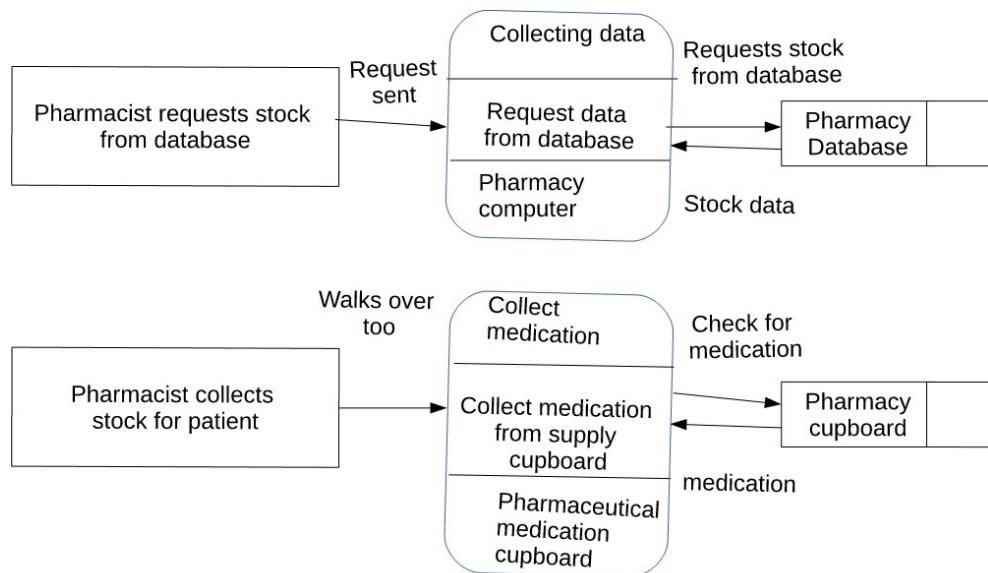
Data flow diagram

Figure 1.2: data flow diagram

Data	Uses	Name
stock detail	stock number	stock check
prescription information	mediction needed	prescription
enough item in stock False	order more item	update stock

Data dictionary

Volumetrics

this system should only be used by pharmaceutical staff in hospitals my client uses this system normally after every patient has gone through to update the stock. so my predicted amount of memory used by the system should calculate up to around 64 bytes to 256 bytes.

1.3 Objectives

1.3.1 General Objectives

- to make a stable system that checks, updates, restocks and sends payment for the ordered items
- to give the system to auto restock items when they fall below a certain number of items
- to graph which items are being bought or used faster and updates the resocking system accordingly

1.3.2 Specific Objectives

- to design a program that will make sorting through the items at the pharmacy as well as store the price and item location in the pharmacy as well as the amount.

1.3.3 Core Objectives

- self updating stock system
- easy accessability
- order more items to refill stock

1.3.4 Other Objectives

- the stock keeping on the program should be accurate. E.G. showing how much one tablet of paracetamol costs
- the system must have automatic communication between the wholesale (warehouse) and the pharmacy

1.4 ER Diagrams and Descriptions

1.4.1 ER Diagram

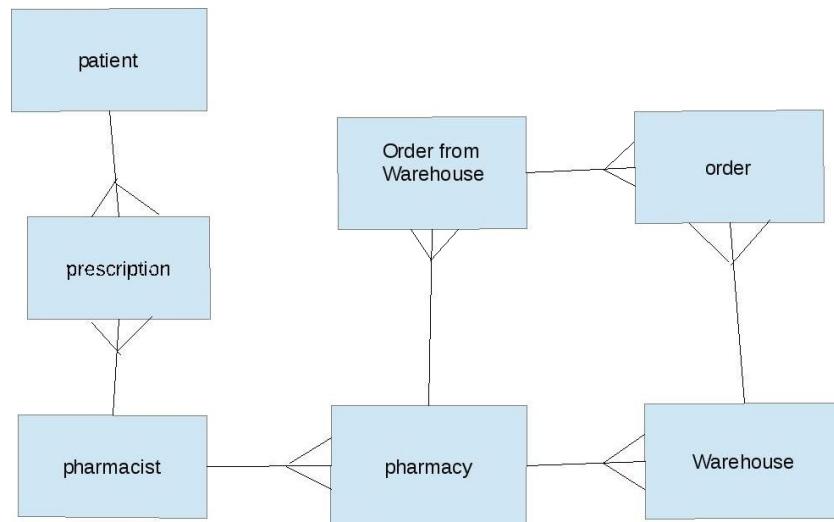


Figure 1.3: entity relationship diagram

1.4.2 Entity Descriptions

- Client(clientID, PharmacyNum, surname, FirstName, PhoneNumber, Address, Postcode)
- Pharmacist(PharamacistID, *PharmacyNum*, Surname, FirstName, PhoneNumber, Address, Email)
- Pharmacy(PharmacyNum, PharmacyAddress, PharmacyPhoneNumber)
- Warehouse(WareHouseNum, PharmacyAddress, WareHouseAddress)
- Order(OrderNum, WareHouseNum, PharmacyLocation, OrderDate, size)

1.5 Object Analysis

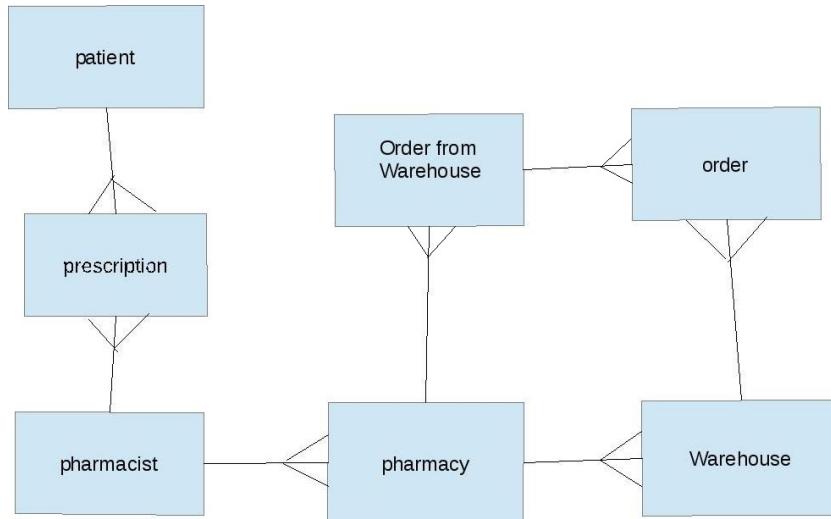
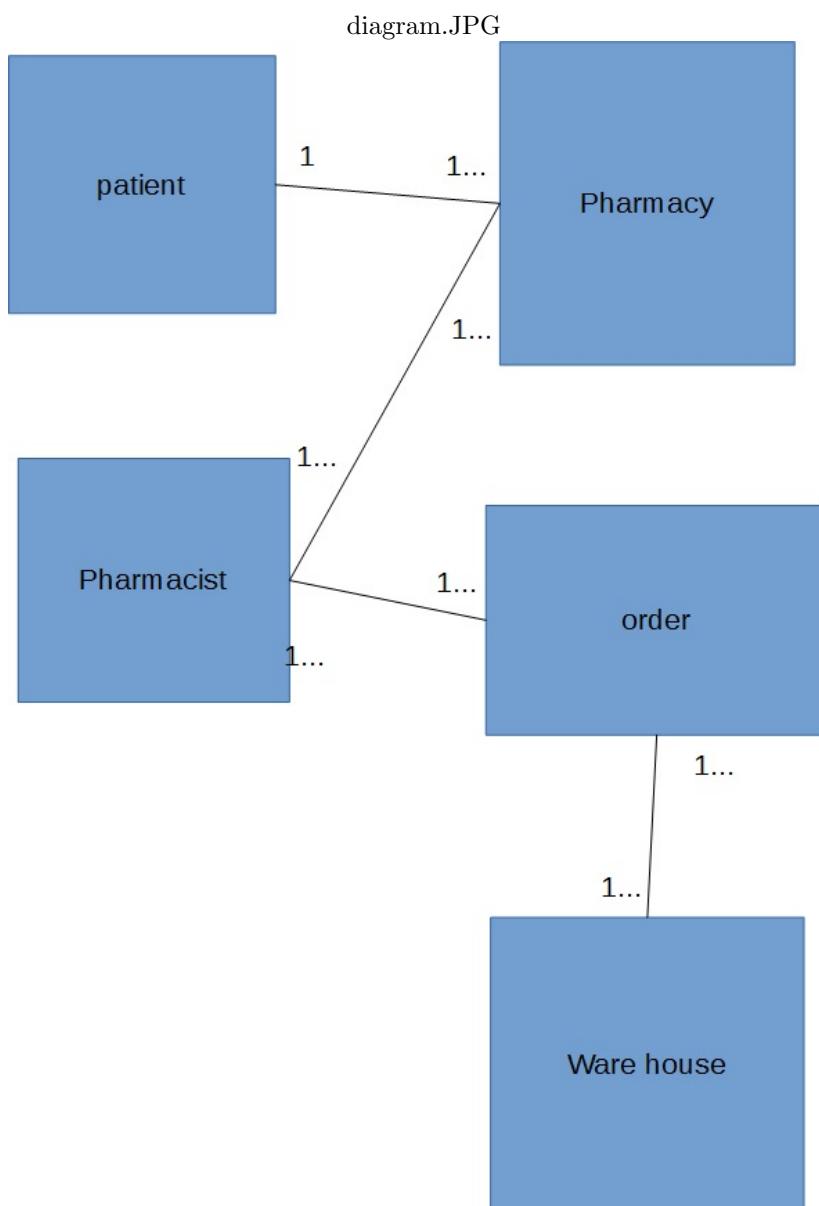


Figure 1.4: entity relationship diagram

1.5.1 Object Listing

- Client
- Pharmacist
- Pharmacy
- Warehouse
- Order

1.5.2 Relationship diagrams



<p>hospital member</p> <p>Surname</p> <p>Firstname</p> <p>Phone Number</p> <p>address</p>	<p>pharmacist: hospital member</p> <p>not inherited:</p> <p><u>PharmacistID</u></p> <p><u>PharmacyNum</u></p> <p>email</p> <p>inherited:</p> <p>Surname</p> <p>Firstname</p> <p>Phone Number</p> <p>address</p>										
<p>patient: hospital member</p> <p>not inherited:</p> <p><u>ClientID</u></p> <p><u>PharmacyNum</u></p> <p>Postcode</p> <p>inherited:</p> <p>Surname</p> <p>Firstname</p> <p>Phone Number</p> <p>address</p>	<table border="1"> <tr> <td data-bbox="690 741 943 777">Warehouse</td> <td data-bbox="951 741 1264 777">Pharmacy</td> </tr> <tr> <td data-bbox="690 779 943 815">not inherited:</td> <td data-bbox="951 779 1264 815">not inherited:</td> </tr> <tr> <td data-bbox="690 817 943 853"><u>WareHouseNum</u></td> <td data-bbox="951 817 1264 853"><u>PharmacyNum</u></td> </tr> <tr> <td data-bbox="690 855 943 891"><u>PharmacyAddress</u></td> <td data-bbox="951 855 1264 891"><u>PharmacyAddress</u></td> </tr> <tr> <td data-bbox="690 893 943 925">WareHouseAddress</td> <td data-bbox="951 893 1264 925">PharmacyPhoneNumber</td> </tr> </table>	Warehouse	Pharmacy	not inherited:	not inherited:	<u>WareHouseNum</u>	<u>PharmacyNum</u>	<u>PharmacyAddress</u>	<u>PharmacyAddress</u>	WareHouseAddress	PharmacyPhoneNumber
Warehouse	Pharmacy										
not inherited:	not inherited:										
<u>WareHouseNum</u>	<u>PharmacyNum</u>										
<u>PharmacyAddress</u>	<u>PharmacyAddress</u>										
WareHouseAddress	PharmacyPhoneNumber										
<p>Order</p> <p>not inherited:</p> <p><u>OrderNum</u></p> <p>OrderDate</p> <p>SizeOfOrder</p> <p>inherited:</p> <p><u>WareHouseNum</u></p> <p><u>PharmacyAddress</u></p>											

1.5.3 Class definitions

1.6 Constraints

1.6.1 Hardware

Susannah uses a small laptop computer. the components of that laptop are listed below:

- 10.1" display
- intel core N270 atom 1.6 ghz
- 1.00GB DDR3 RAM

- 160GB HDD

The proposed system should work on this laptop because of the laptops fast processor it will run through the calculations fast enough.

If using the laptop doesn't work i will switch back to my desktop computer which has:

- 17" display 1024 x 768 pixels
- 34" display 1360 x 768 pixels
- amd A6-3500 APU 3.0 ghz
- 4.00GB DDR3 RAM
- 0.5TB HDD

There should be no problems running the program on this computer.

1.6.2 Software

The operating system used on the laptop is windows xp. Whereas the operating system used on the desktop is running windows 7. The programs that i will be using will be python 3.2.

1.6.3 Time

The final submission of the implimentation of the program must be in by the 13th of February

1.6.4 User Knowledge

The system i will be building will require at least a little background into the pharmaceutical area. This is due to some of the medicines in the system do not have abbreviated names. The installation should take about 5 to 10 minutes.

1.6.5 Access restrictions

the proposed system should only be accessable and privileges to the people in pharmacy, as well as the system should be password protected to ensure no body outside the system can access the stock information.

1.7 Limitations

1.7.1 Areas which will not be included in computerisation

the prescriptions are not given in electronic format so when the patients come to the pharmacy to collect there medicine it has to be collected by the pahrmacists once the patient has got to the pharmacy

1.7.2 Areas considered for future computerisation

the prescriptions should be sent by the doctors to the pharmacy before the patient leaves the doctor so the pharmacy have time to prepare for the patient so they can pick up there prescription and pay and leave all within the space of one minute.

1.8 Solutions

1.8.1 Alternative solutions

solution	advantages	disadvantages
created program that checks the stock after an item is removed	this will keep a continually accurate stock check	this will take up more space to program in

Table 1.2:

1.8.2 Justification of chosen solution

I have chosen to the Python 3.2 desktop application with a GUI and SQL' solution. My reason for using this method is:

- the application will be specific for pharmacy which will be updated at the start of every week and will continuously keep track of the database where the old system.
- the database used will take up less space required to store the data.
- due to the databases size making back ups is very easy so if the system.

Chapter 2

Design

2.1 Overall System Design

2.1.1 Short description of the main parts of the system

my system will consist of 5 main parts, using this information i have given a short description of what the system is meant to do:

- Log in window
 - this will display a login window and the central widget has input boxes for the user to input there user name and password
 - checks in the pharmacy's database to find if the user is registered to the system and has priority to enter the database.
 - the system then checks the users name and finds the password registered to that user. if the password entered is not the same the user will be asked to reenter the password after the input box has been cleared.
- Database display
 - once the user has been accepted into the database the stock will come up on screen for the user
 - the system will then check the stock to see if the amount of items is up to the minimum level and if the stock isn't to the minimum level then the system begins to request for more stock.
 - if the system finds an error e.g. missing stock or calculation problem it will give an error report to the user.
- stock order form

- if the system finds a product low the system will bring up a order request form which will be sent to the warehouse.

–

2.1.2 System flowcharts showing an overview of the complete system

2.2 User Interface Designs

2.3 hardware specification

2.4 Program Structure

2.4.1 Top-down design structure charts

2.4.2 Algorithms in pseudo-code for each data transformation process

2.4.3 Object Diagrams

2.4.4 Class Definitions

2.5 Prototyping

2.6 Definition of Data Requirements

2.6.1 Identification of all data input items

2.6.2 Identification of all data output items

2.6.3 Explanation of how data output items are generated

2.6.4 Data Dictionary

Data	Uses	Name
stock detail	stock number	stock check
prescription information	mediction needed	prescription
enough item in stock False	order more item	update stock

2.6.5 Identification of appropriate storage media

2.7 Database Design

2.7.1 Normalisation

ER Diagrams

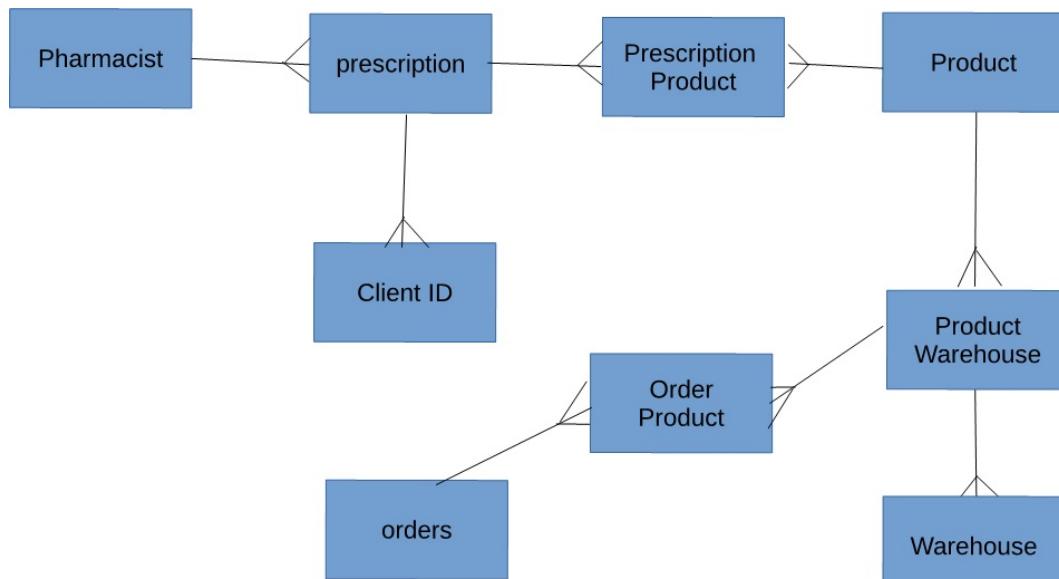


Figure 2.1: ERdiagram

Entity Descriptions

- Client(clientID, FirstName, surname, ClientPhoneNumber, Town, street-Name, HouseNumber/name, Postcode)
- Product(ProductID, ProductName, ProductWeight, ProductCode, Manufacturer, Price)
- Pharmacist(PharamacistID, PharmacistNum, Pharmacistname, PharmacistEmail, PharmacistTown, PharmacistStreet, PharmacistPostcode)
- Warehouse(WareHouseNum, WarehouseTown, WarehouseStreet, Warehouse-Postcode)

- PrescriptionCode(PrescriptionCode, *PharmacistID*, *ClientID*, *Quantity-OfMed*)
- Order(OrderNum, *WareHouseNumber*, *ProductID*, *OrderDate*, *size*)
- Product on prescription(PrescriptionProduct, *ProductID*, *PrescriptionCode*)
- product delivered from warehouse(ProductWareHouse, *ProductID*, *Ware-HouseNumber*)
- Product ordered (OrderProduct, *ProductWareHouse*, *Order*)

UNF

<u>Un-Normalised</u>
ClientID
PharmacyNum
Surname
Firstname
ClientPhoneNumber
ClientAddress
Postcode
PrescriptionCode
PharmacistID
PharmacistTown
PharmacistStreet
PharmacistPostcode
PharmacistEmail
PharmacyAddress
PharmacyPhoneNumber
OrderNum
OrderDate
size
ProductCode
QuantityOfMed
Weight
ProductName
Manufacturer
Price
Town
Postcode
StreetName
HouseNumber/Name
WareHouseNumber
WareHouseStreet
WareHouseTown
WarehousePostcode

1NF to 3NF1NF

non-repeating	repeating
PharmacyNum	<u>ClientID</u>
OrderDate	<u>PharmacyNum</u>
OrderNum	FirstName
Pharmacystreet	Surname
PharmacyTown	ClientPhoneNumber
PharmacyPostcode	Town
	Postcode
	HouseNumber/Name
	StreetName
	ProductCode
	QuantityOfMed
	ProductWeight
	ProductName
	PrescriptionCode
	Manufacturer
	Price
	size
	PharmacistName
	PharmacistNumber
	PharmacistTown
	PharmacistStreet
	PharmacistPostcode
	WarehouseNumber
	WarehouseTown
	WarehouseStreet
	WarehousePostcode

2NF

repeating	Non repeating
<u>ClientID</u>	PharmacyNum
<u>PharmacyNum</u>	OrderDate
PrescriptionCode	Size
	OrderNum
<u>ClientID</u>	PharmacyStreet
FirstName	PharmacyTown
Surname	PharmacyPostcode
ClientPhoneNumber	PharmacyPhoneNumber
HouseNumber/Name	
Town	
Postcode	
StreetName	
<u>PharmacyNum</u>	
PharmacistNumber	
PharmacistName	
PharmacistEmail	
PharmacistTown	
PharmacistStreet	
PharmacistPostcode	
QuantityOfMed	
ProductName	
ProductWeight	
ProductCode	
Manufacturer	
Price	
WarehouseTown	
WarehouseStreet	
WarehousePostcode	

3NF

<u>ClientID</u> FirstName Surname ClientPhoneNumber HouseNumber/Name Town Postcode StreetName	<u>ProductID</u> ProductName ProductWeight ProductCode Manufacturer Price	<u>OrderNumber</u> <i>WareHouseNumber</i> <i>ProductID</i> OrderDate Size	<u>PharmacistID</u> PharmacistNumber PharmacistName PharmacistEmail PharmacistTown PharmacistStreet PharmacistPostcode
<u>PrescriptionCode</u> <i>PharmacistID</i> <i>clientID</i> QuantityOfMed	<u>WareHouseNumber</u> WarehouseTown WarehouseStreet WarehousePostcode	<u>PrescriptionProduct</u> <i>ProductID</i> <i>PrescriptionCode</i>	<u>ProductWareHouse</u> <i>ProductID</i> <i>WareHouseNumber</i>
<u>OrderProduct</u> <i>ProductWareHouse</i> <i>Order</i>			

2.8 Security and Integrity of the System and Data

2.8.1 Security and Integrity of Data

2.8.2 System Security

2.9 Validation

2.10 Testing

2.10.1 Outline Plan

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

2.10.2 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

30

Test Se- ries	Purpose of Test	Test Descrip- tion	Test Data	Test Data Type (Nor- mal/ Er- roneous/ Boundary)	Expected Result	Actual Re- sult	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

31

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

34

4.5.5 SQL Queries

4.6 Testing

4.6.1 Summary of Results

4.10.1 Module 1

Henry Mason

Candidate No. 2634

Centre No. 22151

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

37

5.3.4 Saving

5.3.5 Limitations

5.4 Error Recovery

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