# Computing Coursework

Henry Mason

October 23, 2014

# Contents

# Chapter 1

# Analysis

## 1.1 Introduction

### 1.1.1 Client Identification

my client is Susannah Mason, she is 50 years old and has very 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.

Suesannah is a pharmacutical manager at spire healthcare in impington

by creating this program it would speed up the process making leeping track of and ordering of new equipment and stock alot 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 conted by hand

### 1.1.4 Section appendix

he

| Data Source | Travels via | destination |
|---|---|---|
| doctor | gives prescription | patient |
| patient | requests medicine | pharmacist |

Table 1.1:

## 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

#### Algorithms

i will be using quite a few algorithums for this assignment

IF (item = lowest minimum amount) THEN order more item

ELSE check next item

END IF

this other algorithm will be used to calculate the exact price of all of the order

if order submitted = True THEN calculate order

ELSE restart program

END IF

using the information in the list Items the exact price is calulated

IF item in Items add price item

ELSE add price 0

END IF

| data Source | travels via | destination |
|---|---|---|
| stock information | request for stock information | pharmacy computer |
| stock price | request stock price | pharmacy computer |
| show stock info | display info | pharmacist |

| Data | Uses | Name |
|---|---|---|

**Data flow diagram**

**Input Forms, Output Forms, Report Formats**

### 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 contol point before being sent off

**Data sources and destinations**

**Data flow diagram**

**Data dictionary**

**Volumetrics**

## 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 acordingly

### 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

### 1.3.4 Other Objectives

## 1.4 ER Diagrams and Descriptions

### 1.4.1 ER Diagram

### 1.4.2 Entity Descriptions

- Client(clientID, PharmacyNum,)
- Pharmacist(PharamacistID,*PharmacyNum*, surname, firstname, PhoneNumber, address,email)
- Pharmacy(PharmacyNum)
- Warehouse(WareHouseNum, )
- Order(OrderNum, *WareHouseNum*, OrderDate,)

## 1.5 Object Analysis

### 1.5.1 Object Listing

- Client
- Pharmacist
- Pharmacy
- Warehouse
- Order

| solution | advantages | disadvantages |
|---|---|---|
| problem | solved | hopwfully |

Table 1.2:

### 1.5.2 Relationship diagrams

### 1.5.3 Class definitions

## 1.6 Other Abstractions and Graphs

## 1.7 Constraints

### 1.7.1 Hardware

### 1.7.2 Software

### 1.7.3 Time

### 1.7.4 User Knowledge

### 1.7.5 Access restrictions

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

## 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

### 1.9.2 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

### 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



### 4.5.5 SQL Queries

## 4.6 Testing

### 4.6.1 Summary 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

# 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