 ./

**Report**

**MEDICAL STORE DATABASE MANAGEMENT**

Amaljith S

SF ID : 104447



Version Number:

Team Members :

Team No:

Module: Model Based System Engineering

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver.Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **Approved By** | **Remarks/Revision Details** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Document History**

# 

Contents

[1. INTRODUCTION 4](#_Toc53065984)

[1.1 Problem Statement 4](#_Toc53065985)

[1.2 Project Description 4](#_Toc53065986)

[2. REQUIREMENTS 5](#_Toc53065987)

[2.1 Software Used 5](#_Toc53065988)

[2.2 Header Files Required 5](#_Toc53065989)

[2.3 Functions Required 6](#_Toc53065990)

[3. DESIGN 7](#_Toc53065991)

[3.1 Architecture 7](#_Toc53065992)

[3.2 Explanation 8](#_Toc53065993)

[3.3 Use Case Diagram 9](#_Toc53065994)

[3.4 Activity Diagram 10](#_Toc53065995)

[4. TEST PLAN 11](#_Toc53065996)

[4.1 Introduction 11](#_Toc53065997)

[4.2 Test Strategy 11](#_Toc53065998)

[4.2.1 Scope of Testing 11](#_Toc53065999)

[4.2.2 Test Type 11](#_Toc53066000)

[4.2 Test Objective 12](#_Toc53066001)

[4.3 Test Criteria 12](#_Toc53066002)

[4.3.1 Suspension Criteria 12](#_Toc53066003)

[4.3.2 Exit Criteria 12](#_Toc53066004)

[4.4 Test Environment 12](#_Toc53066005)

[5. TEST CASES 13](#_Toc53066006)

[6. EXPECTED RESULTS AND OUTCOME 18](#_Toc53066007)

[7. CONCLUSION 20](#_Toc53066008)

# INTRODUCTION

## 1.1 Problem Statement

To manage the database in a medical store.

## 1.2 Project Description

The details of the medicines that are present in the medical store can be managed in a well organized way with the help of this project. The project helps in collecting the details of the medicines stored in a medical store and accesses those details when required. The user can keep track of the details of the medicines bought and sold. The user can also search for the details of the customer and supplier by using an id or a name. The Medical Shop Management System helps to maintain and keep the medicines in the medical store in the proper place. When you need a particular medicine, you can get to know the exact location of that particular medicine that you are searching for.

In this project file handling should be effectively used for almost all major functions. The whole project is based on file handling as all records are stored in a file. Data structure has also been used to store and organize records. The user can input many information like medicine Id, rack no., cabinet no., supplier’s name, unit cost etc while adding a medicine into the store which will be stored in files and can be accessed when required by the user.

# REQUIREMENTS

## 2.1 Software Used

* IDE : Code::Blocks version 20.03
* Compiler : MinGW compiler

## 2.2 Header Files Required

|  |  |  |
| --- | --- | --- |
| **Requirement ID** | **Header File** | **Description** |
| RQ\_H\_01 | <stdio.h>  (Standard input-output header) | Used to perform input and output operations in C. |
| RQ\_H\_02 | <conio.h>  (Console input-output header) | Perform console input and console output operations like clrscr() to clear the screen and getch() to get the character from the keyboard. |
| RQ\_H\_03 | <stdlib.h>  (Standard library header) | Perform standard utility functions. |
| RQ\_H\_04 | <time.h>  (Time header) | Perform functions related to date and time like setdate() and getdate(). To modify the system date and get the CPU time respectively. |
| RQ\_H\_05 | <string.h>  (String header) | Perform string manipulation operations like strlen and strcpy. |

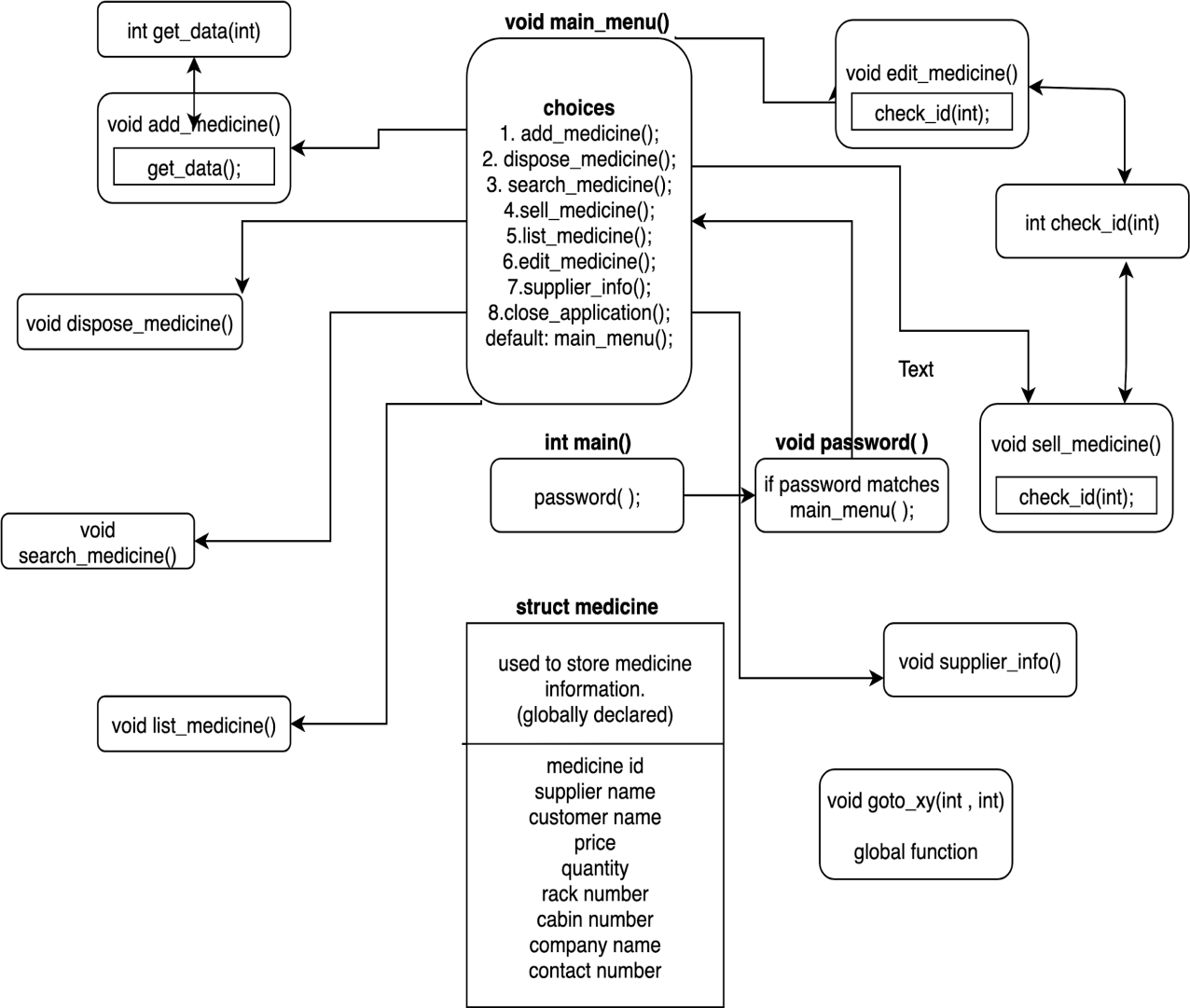
### 

## 2.3 Functions Required

|  |  |  |
| --- | --- | --- |
| **Requirement ID** | **Function** | **Description** |
| RQ\_F\_01 | goto\_xy | This function allows us to go to any coordinates on the screen. |
| RQ\_F\_02 | password | This function is used to check if the password entered by the user is correct. |
| RQ\_F\_03 | main\_menu | This function contains various function choices, which the user can choose from. |
| RQ\_F\_04 | add\_medicine | This function is called when a new medicine is needed to be added in the store. The details of the medicine and supplier are recorded. |
| RQ\_F\_05 | dispose\_medicine | This function is called when a medicine expires and we need to remove it from the store. |
| RQ\_F\_06 | search\_medicine | This function is called when we need to find the details of a medicine. The search can be done using medicine name or ID. |
| RQ\_F\_07 | sell\_medicine | This function is called when a medicine is sold to a customer. The details of the medicine and customer are recorded. |
| RQ\_F\_08 | list\_medicine | This function can be called if we need to know the details of all available medicine in the store. |
| RQ\_F\_09 | edit\_medicine | This function can be called if we need to edit the records of the medicines stored. |
| RQ\_F\_10 | supplier\_info | This function can be called if we need to find the details of the suppliers. |
| RQ\_F\_11 | check\_id | This function is used to check if an id entered by the user already exists. |

# DESIGN

## 3.1 Architecture



## 3.2 Explanation

#### **get\_data ( )**

* This function is used to collect the input entered by the user.
* If the input is successfully collected the function returns ‘1’.

#### **add\_medicine ( )**

* This function is used to store the details of a new medicine added to the store.
* The details are inputted by the user.
* If the value returned by the function get\_data ( ) is ‘1’ then the entered details are stored on to a file.

#### **dispose\_medicine ( )**

* This function is used to delete the records of a medicine required by the user.
* To delete the record, a temporary file is created and all the data except the required one is written into the temporary file.
* The file is then removed.
* Data from the temporary file is copied into the original file.

#### **search\_medicine ( )**

* This function is used to obtain details of a medicine required by the user.
* The user can search the medicine either by ID or name of the medicine.
* If the required medicine ID or name is available in the structure, its records are printed.

#### **check\_id ( )**

* This function checks if a medicine with a certain ID is available in the store.
* If the ID inputted is matched with a stored medicine ID the function returns value ‘1’.

#### **supplier\_info ( )**

* This function is used to obtain the details of suppliers.
* The function has two options, to list the details of all the suppliers and to search the details of a given customer.

#### **sell\_medicine ( )**

* This function has options to sell a medicine, list details of sold medicines, search for a sold medicine and delete details of the sold medicine.
* If the value returned by the function check\_id ( ) is ‘1’ then the medicine is sold.
* The details of the medicine sold and details of the customer will be stored in another file.

#### 

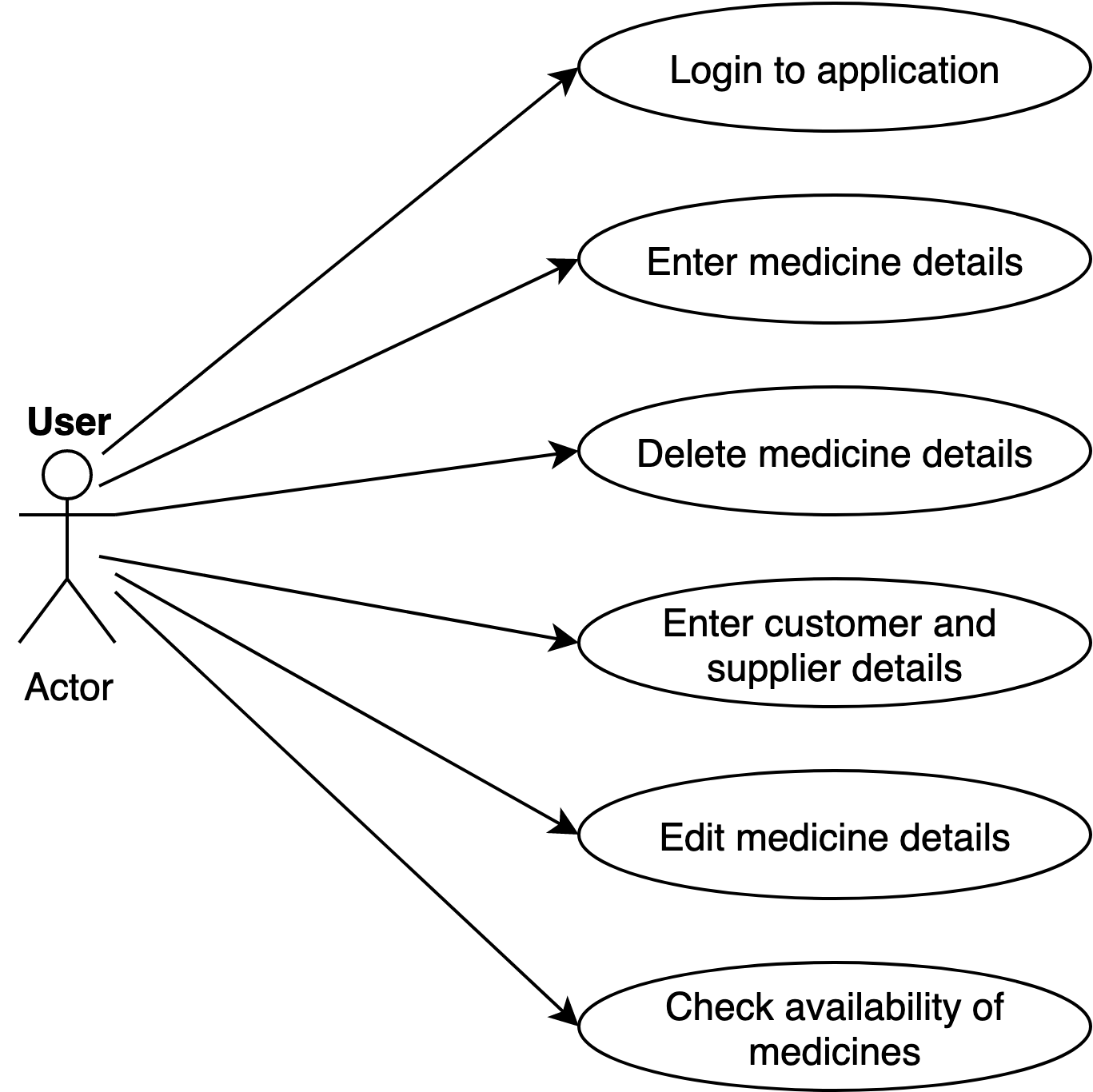
#### **list\_medicine ( )**

* This function is used to obtain the details of all the medicine available in the store.
* If the function is called, read the file in which the records of the medicines are stored.

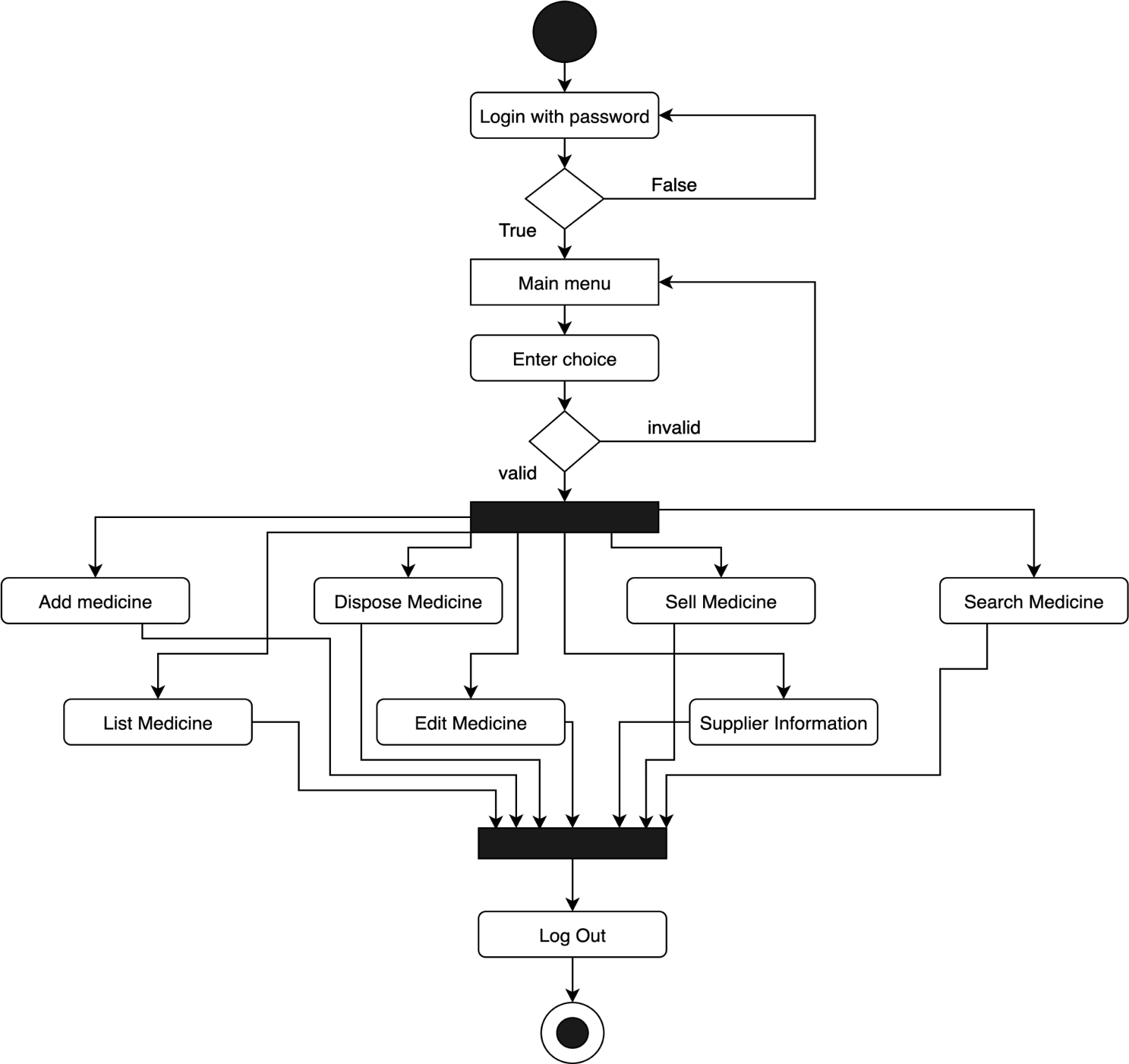
#### **edit\_medicine ( )**

* This function is used to edit the recorded details of medicine required by the user.
* To check if the required medicine is available function check\_id( ) is called.
* If the returned value is ‘1’ then the file is rewritten with new data.

## 3.3 Use Case Diagram



## 3.4 Activity Diagram



# TEST PLAN

## 4.1 Introduction

Test plan is designed to advise the scope, approach and strategy of testing an application, to identify risks and issues, to define test criteria and test environment, and to identify the type of testing to be performed on the Application Under Test (AUT).

## 4.2 Test Strategy

### 4.2.1 Scope of Testing

#### **Features to be tested**

|  |  |
| --- | --- |
| **Module** | **Description** |
| Login | Only the owner can have access to the database. |
| Input data | The user can input details regarding the medicine. |
| Store data | The user can save the data entered into the application. |
| Delete data | The user can delete stored records when needed. |
| View data | The user can access the desired data when needed. |
| Edit data | The user can edit the saved data when required. |
| Search data | The user can search for specific data. |

#### **Features to be tested**

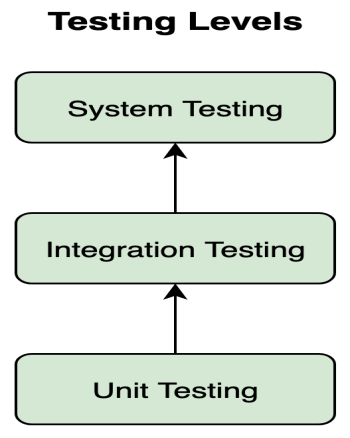
These features are not to be tested because they are not included in the software requirement specifications.

* User Interfaces
* Hardware Interfaces

### 4.2.2 Test Type

In this project there are three types of testing to be conducted:

* **Unit Testing:** It is a type of software testing where individual units or components of a software are tested.
* **Integration Testing**: It is a type of software technique where individual software modules are integrated and tested as a group.
* **System Testing:** It is a level of testing that validates the complete and fully integrated software product. It is done to check whether the software meets specified requirements or not.



Unit tests help to fix bugs early in the development cycle. Unit testing is of two types :

* Manual
* Automated

Under automated approach the developer writes a section of code in the application just to test the function. It is later commented out and finally removes the test code when the application is deployed.

## 4.2 Test Objective

The test objectives are to verify the different functionalities of the Application Under Test (AUT). The project should focus on testing operations such as entering, editing, deleting, storing and searching data to make it work without errors on an actual environment.

## 4.3 Test Criteria

### 4.3.1 Suspension Criteria

If more than 40% of test cases are failed then the testing is suspended until all failed cases are fixed.

### 4.3.2 Exit Criteria

This denotes a successful completion of a test phase.

* It is mandatory that the run rate is 100%.
* Pass rate is 95%.

## 4.4 Test Environment

All test cases are carried out in Code Blocks IDE version 20.03.

# TEST CASES

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Test Case Description** | **Test Steps** | **Test Data** |
| TU01 | Check Login with a valid password. | 1. Enter password  2. Click Enter. | password = ‘admin’ |
| TU02 | Check Login with an invalid password | 1. Enter password  2. Click Enter. | password = ‘user’ |
| TU03 | Check whether the add\_medicine option in main menu is working | 1. Enter choice  2. Click Enter | choice = 1 |
| TU03\_01 | Check whether category options in add\_medicine is working | 1. Enter choice  2. Click Enter | choice = 1 |
| TU03\_02 | Check whether category options in add\_medicine is working with a different choice | 1. Enter choice  2. Click Enter | choice = 3 |
| TU03\_03 | Check whether category options in add\_medicine is working with an invalid choice | 1. Enter choice  2. Click Enter | choice = 9 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Test Case Description** | **Test Steps** | **Test Data** |
| TU03\_01\_01 | Check if medicine details can be successfully added. | 1. Enter the details  2. Click ‘Y’ to add  more or ‘N’ to exit. | medicine\_id = 401  medicine\_name =  “clindamycin”  supplier\_name =  “supplier one”  contact = 3249542  quantity = 5  price = 100  rack\_no = 4  cabin\_no = 3  Add more medicine:Y  medicine\_id = 402  medicine\_name =  “azithromycin”  supplier\_name =  “supplier two”  contact=3255412  quantity = 6  price = 100  rack\_no = 4  cabin\_no = 7  Add more medicine:N |
| TU04 | Check whether the search medicine option in the main menu is working. | 1. Enter choice. | choice = 3 |
| TU04\_01 | Check if the medicine record can be deleted successfully with medicine ID. | 1. Enter medicine ID | medicine\_id =401 |
| TU04\_02 | Check if record can be deleted with invalid medicine ID | 1. Enter medicine ID | Medicine\_id = 999 |
| TU04\_03 | Check if the medicine record can be found by search by name option | 1. Enter choice  2. Enter medicine  name | choice = 2  medicine\_name =  ”clindamycin” |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Test Case Description** | **Test Steps** | **Test Data** |
| TU04\_04 | Check if a medicine record can be found by an invalid medicine name. | 1. Enter choice  2. Enter medicine  name | choice = 2  medicine\_name =  “abcd” |
| TU05 | Check if the list medicine option in main menu is working | 1. Enter choice | choice = 5 |
| TU06 | Check if the edit medicine option in main menu is working | 1. Enter choice  3. Enter medicine  ID to be edited  5. Enter new details  6. Enter ‘Y’ to modify another record or ‘N’ to exit | choice = 6  medicine\_id = 401  new name =  “clindamycin”  new supplier =  “supplier one”  new contact =  2426242  new quantity = 10  new price = 120  new rack\_no = 3  new cabin\_no = 4  Modify another = N |
| TU07\_01 | Check if sell option in sell medicine is working | 1. Enter choice  3. Enter medicine ID  to be sold  5. Enter name of  Customer  7. Enter contact no of  customer | choice = 1  medicine\_id = 401  customer\_name =  “customer one”  contact\_no =  9012345678 |
| TU07\_02 | Check if list medicine option inside sell medicine is working | 1. Enter choice | choice = 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Test Case Description** | **Test Steps** | **Test Data** |
| TU07\_03\_01 | Check if search medicine option inside sell medicine is working with valid ID | 1. Enter choice.  3. Enter medicine ID to be found | choice = 3  medicine\_id = 401 |
| TU07\_03\_02 | Check if search medicine option inside sell medicine is working with an invalid ID | 1. Enter choice  3. Enter medicine ID to be found | choice = 3  medicine\_id = 999 |
| TU07\_04 | Check if delete sold medicine option inside sell medicine is working | 1. Enter choice.  3. Enter medicine ID to be deleted | choice = 4  medicne\_id = 401 |
| TU08 | Check if supplier info option in main menu is working | 1.Enter choice | choice = 7 |
| TU08\_01 | Check if List of suppliers option inside supplier info is working | 1.Enter choicer | choice = 1 |
| TU08\_02 | Check if search supplier option inside supplier info is working | 1.Enter choice  3.Enter name | choice = 2  name = supplier one |
| TU09 | Check if an invalid entry will work on  main menu | 1. Enter choice | choice = 99 |
| TU10 | Check if the dispose medicine option in the main menu is working with a valid ID. | 1. Enter choice  3. Enter medicine ID | choice = 2  medicine\_id = 402 |
| TU10 | Check if the dispose medicine option in the main menu is working with a valid ID. | 1. Enter choice  3. Enter medicine ID | choice = 2  medicine\_id = 402 |
| **Test Case ID** | **Test Case Description** | **Test Steps** | **Test Data** |
| TU11 | Check if the dispose medicine option in the main menu is working with an invalid ID. | 1. Enter choice  3. Enter medicine ID | choice = 2  medicine\_id = 999 |
| TU12 | Check if close application option in  main menu is working | 1. Enter choice | choice = 9 |

# EXPECTED RESULTS AND OUTCOME

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Expected Output** | **Actual Output** |
| TU01 | Login should be successful and output window should show main menu | Login successful. Main Menu appeared. |
| TU02 | Login should be unsuccessful and password should be asked again | Login unsuccessful. Password was asked again. |
| TU03 | Entry should be accepted and a screen showing different categories of medicine should appear. | Entry accepted. Screen Showing different categories of medicine appeared. |
| TU03\_01 | Entry should be accepted and a list of queries should appear asking the user to input details of medicine. | Entry accepted and a list of queries appeared asking the user to input details of medicine. |
| TU03\_02 | Entry should be accepted and a list of queries should appear asking the user to input details of medicine. | Entry accepted and a list of queries appeared asking the user to input details of medicine. |
| TU03\_03 | Entry should not be accepted and should ask for reentry. | Entry not accepted and asked for reentry. |
| TU03\_01\_01 | Details should be successfully saved and a message saying record successfully saved should appear | Details successfully saved and a message saying record successfully appeared |
| TU04 | Entry should be accepted and a query should appear asking the user to choose either search by ID or search by name. | Entry accepted and a query appeared asking the user to choose either search by ID or search by name. |
| TU04\_01 | Medicine records should be collected successfully. | Medicine records collected successfully. |
| TU04\_02 | Medicine records could not be found, a message saying “record does not exist” should appear. | Medicine not found, a message saying “record does not exist” appeared |
| TU04\_03 | Medicine records should be collected successfully. | Medicine records should be collected successfully |
| TU04\_04 | Medicine records could not be found, a message saying “record does not exist” should appear. | Medicine records not found, a message saying “record does not exist” appeared. |

|  |  |  |
| --- | --- | --- |
| TU05 | Entry should be accepted and a screen showing details of all available medicines should appear. | Entry accepted and details of all available medicines appeared. |
| TU06 | Entries will be accepted and the information inputted will be saved. A message saying “records modified successfully” should appear. | Entries accepted and the information inputted was saved. A message saying “records modified successfully” appeared. |
| TU07 | Entry should be accepted and a query will appear asking the user to choose from the options sell medicine, list medicine, search medicine and delete medicine. | Entry should accepted and a query will appear asking the user to choose from the options sell medicine, list medicine, search medicine and delete medicine |
| TU07\_01 | Entry should be accepted, customer details should be saved and a message saying medicine sold successfully should appear. | Entry accepted, customer details were saved and a message saying medicine sold appeared. |
| TU07\_02 | Entry should be accepted and an output screen displaying details of all sold medicines should appear. | Entry accepted and an output screen displaying details of all sold medicines appeared. |
| TU07\_03\_01 | Entries should be accepted and records of the requested medicine will be collected. | Entries accepted and records of the requested medicine were collected |
| TU07\_03\_02 | The requested records of the medicine cannot be collected. A message saying “no such records exist” will appear. | The requested records of the medicine could not be collected. A message saying “no such records exist” will appear. |
| TU07\_04 | Entry should be accepted and the records of the specified ID will be deleted. | Entry accepted and the records of the specified were deleted. |
| TU08 | Entry should be accepted and a query should appear asking the user to choose from ‘’list of supplier’’ and’’ search supplier’’. | Entry accepted and a appeared asking the user to choose from ‘’list of supplier’’ and ’’ search supplier’’ |
| TU08\_01 | Entry should be accepted and the details of all suppliers should appear. | Entry accepted and the details of all suppliers appeared. |
| TU08\_02 | Entry should be accepted and a screen should appear showing the details of that customer. | Entry accepted and a screen appeared showing the details of that customer. |
| TU09 | Entry should not be accepted and should ask again for input. | Entry not accepted and asked again for input. |
| TU10 | Entry should be accepted and the records of specified medicine should be deleted. | Entry accepted and the records of specified medicine were deleted. |

# CONCLUSION

This software helps in the effective management of the medical store. The application provides an easy way for the user to interact with the database and to manipulate the data in the database. It works as per the requirement of the user and as options accordingly. The user is able to add, delete, search and update the records in the database with ease. It provides a good user interface and is easily understandable to the user.