

# TEST PLAN

19CS10026 D SASI KUMAR

19CS30010 R ATHITHYA PRAKASH

19CS10006 ADITYA ANANTWAR

## *ChangeLog*

Version	Change Date	By	Description
1.0	27/03/2021	Team	Initial Test Plan

# 1.TEST PLAN IDENTIFIER

MSA-TPV1.0

## 2.REFERENCES

IEEE -829 IEEE Recommended Practice for Test plan document. IEEE Computer Society.

## 3.INTRODUCTION

The Test Plan is designed to prescribe the scope, approach, resources, and schedule of all testing activities of the project Medical Shop Automation.

The plan identifies the items to be tested, the features to be tested, the types of testing to be performed, the personnel responsible for testing, the resources and schedule required to complete testing, and the risks associated with the plan.

### 3.1 Objective

The test objectives are to **verify** the Functionality of Medical Shop website, the project should focus on testing the **selling and managing medicines** such as adding stock , medicines, vendors to database, and selling and purchasing from vendors...etc. to **guarantee** all these operation can work **normally** when the Employee or the Admin uses it.

## 4.TEST ITEMS

All the features of medical shop automation which were defined in software requirement specification are needed to be tested.

- ☐ Login
- ☐ new user
- ☐ Adding new medicines
- ☐ Adding new vendors
- ☐ Adding Stocks
- ☐ Sell medicine
- ☐ Listing the expired medicines
- ☐ Medicines List(all the medicines )
- ☐ Purchase Medicine list (list of medicines that are less than a threshold value)
- ☐ Vendor list
- ☐ View users
- ☐ Sales list
- ☐ Revenue and Profit

## 5. SOFTWARE RISK ISSUE

There are several parts of the project that are not within the control of the development team application but have direct impacts on the process and must be checked as well.

- The software we prepared may not have the ability to change its process. So, hence any software that will have updates at the time deployment may cause malfunctions.
- We use third party softwares for both the front and backend of an application, this may cause risks when they get updated.
- Specific functions have particular data type as inputs any irregular input by the user may cause some software issues.

## 6.FEATURES TO BE TESTED

The users who use this software are:

## **Admin:**

All features including the ability to check revenue and profits over a duration will be available to the admin. The admin also has the ability to create a new user.

## **Employees:**

All features except the ability to check the revenue and profits and the ability to create a new user will be available to the employees.

Hence the features which are to be tested is

### **6.1 Login**

#### **6.1.1 Description**

Takes username and password from the user and checks whether the credentials are valid. This is a High Priority feature as it controls who can use the software.

Admin and employee should first login to use the software.

### **6.2 Create New User**

#### **6.2.1 Description**

Used to add a new user for the system. Has a form which takes as input first name, middle name, last name, gender, date of birth, e-mail, address and username as CharFields(string). It also takes landline number and mobile number as long integer fields. Role is a dropdown field to choose whether the added person is an admin or an employee. Password and confirm password are password fields.

After submitting password and confirm password should match otherwise a popup warning should appear.

Only the admin has access to create any user and any number of users..

This is a High Priority feature as it is necessary to add new employees for the system.

## **6.3 Add New Medicine**

### **6.3.1 Description**

Used to add a new medicine to the stores list of medicines. It has a form which takes as input trade name and generic name as CharFields(string) and unit sell price and unit purchase price as integer values.

Can add multiple medicines.

This is a High Priority feature as the store needs medicines in its inventory to sell them. The unique id is generated after adding. Admin and employee can be able to use this function.

## **6.4 Add New Vendor**

### **6.4.1 Description and Priority**

Used to add a new vendor (supplier) to the store for medicines. It has a form which takes as input the vendor name, vendor email and address as CharFields(string) and the phone as an integer field. The medicine ids sold are comma separated medicine ids for medicines already in the database which are sold by the vendor.

Can add many vendors.

The unique id is generated after adding to avoid repetitions. Admin and employee can be able to use this function.

## **6.5 Add Stock**

### **6.5.1 Description**

Used to add stock of existing medicines after they have been delivered. It takes as input the medicine id, batch id, quantity and expiry date as input. It has provisions to add multiple types of medicines at the same time. It calculates the amount per batch using the purchasing price of the corresponding medicine and totals up the amount to be paid to the vendor. It can then be used to draw a cheque in the name of the vendor.

Can add multiple medicines. Warning as a popup should appear if vendor or medicine id is not found

Admin and employee can be able to use this. This is a high priority feature.

## **6.6 View Expired Medicine**

### **6.6.1 Description**

At the end of each day the user can see what medicines have expired and need replacement as a table on this page. The table gives the medicine id, the available stock (other than the expired batch), the batch id that has expired, the vendor id and their email and address for getting the replacement

Both admin and employee can access this page and can also edit.

## **6.7 View Order List**

### **6.7.1 Description and Priority**

The store should have an inventory of medicines higher than a certain threshold at all times. Any medicines whose stock in the inventory has gone below a pre-decided threshold

are shown in the table in the order list along with the vendors who can supply the medicines.

Both admin and employee can access this page and can also edit.

This is a Medium Priority Feature.

## **6.8 Revenue and Profit**

### **6.8.1 Description and Priority**

This page is available only to admins. The admins can enter any two dates and calculate the revenue generated between these two dates and the profits.

Only admins should be able to access the page. In case an employee tries to open the page, they should get a pop-up stating the page is only for admins.

This is a Low Priority Feature.

## **6.9 Medicine List**

### **6.9.1 Description and Priority**

A table showing the details of all the medicines available in the inventory of the store. It also has an option to edit an entry. In the future, it may get a search bar for quick access.

Both admin and employee can access this page and can also edit.

It is a Low Priority feature.

## **6.10 Vendor List**

### **6.10.1 Description**

A table showing the information regarding all the vendors who supply the store. It has an option to edit the details regarding every vendor. In the future, a feature to search the vendor by name may be added.

Both admin and employee can access this page and can also edit.

It is a Low Priority Feature.

## **6.11 List Users**

### **6.11.1 Description**

A table showing the information regarding all the users (admins and employees) of the system. It has the order to edit the details regarding all the users.

Only admins can access this page and can also edit.

It is a Low Priority Feature.

## **7.FEATURES NOT TO BE TESTED**

The feature are not be tested are

- Hardware Interfaces
- Database logical
- Communications Interfaces.---
- Deployment of web app will not be tested as it can be affected by various servers and web browsers.
- The validity of the front end may not last as it can be deprecated by the provider at any time.
- Website Security and Performance.

## **8. APPROACH**

### **8.1 Testing Levels:**



The testing for this project will consist of Unit Testing and manual testing. This testing will be done by the total team according to their assigned roles.

UNIT Testing will be done by the team. Proof of unit testing (test case list, sample output) must be provided by the team before unit testing will be accepted.

## **9. ITEM PASS/FAIL CRITERIA**

The test process will be completed once the final design and implementation of the project is done. When the Unit test output is satisfied to the expected output of the project for the same test cases mentioned in the document.

Test of the interface will be done manually since the input is given by the user and sometimes administrator. These outputs should be tallied with the documented outputs as well.

## **10. TEST DELIVERABLES**

Test deliverables are provided as below

### **Before testing phase**

- Test plans document.
- Test cases documents
- Test Design specifications.

### **During the testing**

- Test Data

### **After the testing cycles is over**

- Test Results/reports
- Installation/ Test procedures guidelines
- Release notes

## 11.RESPONSIBILITIES

RESPONSIBILITIES	TEAM	CLIENT
Unit test documentation and execution	✓	
Test plan and Test case	✓	
SRS Documentation and implementation	✓	
System Design Reviews	✓	✓

The team will be responsible for verifying and accepting all unit test plans and documentation and implementation.

The entire project team will participate in the review of the system and detail designs and review any change requests generated by the user or defects discovered during development and testing

## 12.TEST ENVIRONMENT

- A python environment with django version 3.1.7 should be created and dependencies that need to be installed via pip are mysqlclient, django-crispy-forms. A MySQL server should be hosted with a database named msa with a user with username 'dbadmin' and password 'admin123' and hosted on port 3306. Then this database should be hosted over localhost.
- Then cd to the project directory upto the folder with manage.py in it and run *python manage.py makemigrations* followed by *python manage.py migrate*. This should update the database and create required tables. This can then be followed by running *python manage.py test msa\_app* to test using the tests.py file attached in the test suit.

- For running the manual tests for interface based features, host the application using *python manage.py runserver*. This should host the django application over a port with the link in the terminal. Then the manual test cases can be checked on the web browser.

## **13.SCHEDULE**

Time has been allocated for viewing the project plan for the following reviewing activities. The specific dates and times for each activity are defined in the project plan timeline.

- Review and documentation of the Software Requirements Specifications (SRS) by the team and create the inventory classes, methods, and objectives.
- Development of the Master Plan and documentation of the test cases and class diagrams.
- Review of the implementation and the unit test and test cases.
- Presentation of the final output of the project.

## **14.PLANNING RISKS AND CONTINGENCIES**

- Delay in the training of application or tools :

We are making a web-based application that is very new to the team, and using python as middleware is also new to get trained for that software and tools takes time, which may cause a delay in submission of unit test code.

- Change in original requirements or design :

This test plan is being made without code and may contain some warnings when implementing a unit test or code implementation. Some of the mentioned conditions may change at the time of execution and testing according to the team's convenience and resources availability.

However, the team's time availability may balance the problem mentioned above and submit the test plan and implementation and unit test on time.