

# **SQTA Mini-Project 1**

## **Title:**

Manual Testing of Application

## **Problem Definition:**

Create a Medical Healthcare System by selecting 14- Sept -2020 relevant system environment / platform and programming languages. Narrate concise Test Plan consisting features to be tested and bug taxonomy. Prepare Test Cases inclusive of Test Procedures for identified Test Scenarios. Perform selective Black-box and White-box testing covering Unit and Integration test by using suitable Testing tools. Prepare Test Reports based on Test Pass/Fail Criteria and judge the acceptance of application developed.

## **Software Requirements:**

Eclipse java, Java, JavaFx, MYSQL

## **Hardware Requirement:**

8GB Ram

## **Learning Objectives:**

1. To understand manually testing and preparation of test cases.
2. To understand different types of testing.

## **Learning Outcomes:**

Students will be able to:

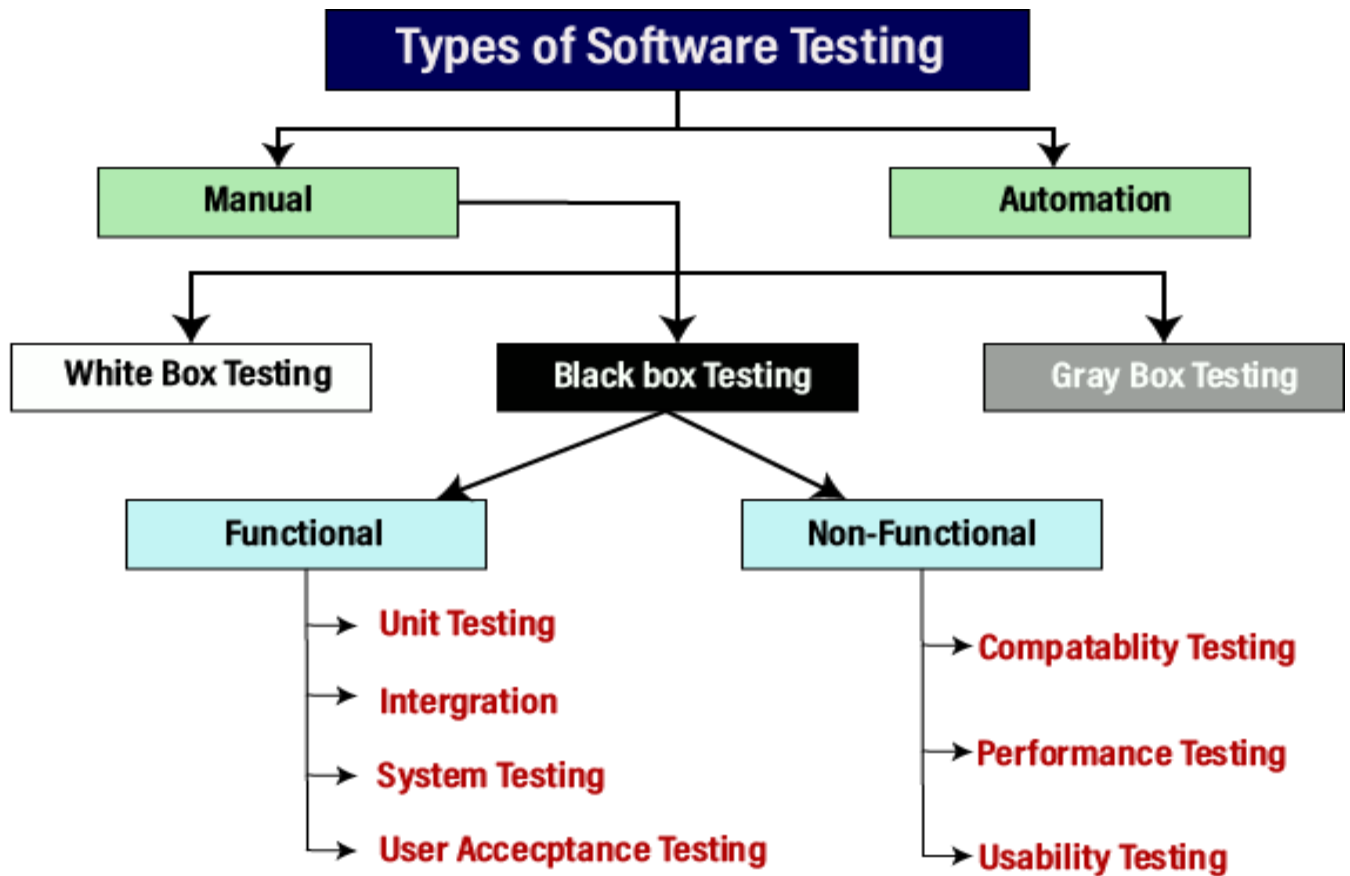
1. Prepare test cases and manually test applications.
2. Understand different types of testing paradigms.

## **Theory:**

### **1. What is Testing?**

- Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free.
- Testing is done to find bugs and errors that can be resolved to make software reliable and durable.

## 2. Testing types



## 3. Unit Testing:

- It focuses on the smallest unit of software design.
- In this we test an individual unit or group of interrelated units.
- It is often done by programmers by using sample input and observing its corresponding outputs.
- Examples: In a program we are checking if loop, method or function is working fine, Misunderstood or incorrect, arithmetic precedence.

## 4. Integration Testing:

- It is defined as a type of testing where software modules are integrated logically and tested as a group.
- A typical software project consists of multiple software modules, coded by different programmers.

### 5. **White Box Testing:**

- It is used for verification.
- In this we focus on internal mechanisms i.e. **how the output is achieved?**
- It is a software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security.
- In white box testing, code is visible to testers so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing and Glass box testing.

### 6. **Black Box Testing:**

- It is used for validation.
- In this we ignore internal working mechanisms and focus on **what is the output?**

### **Application Developed:**

An application that can be used to maintain digital records of the patients admitted at an covid ward. This will help to have the records at one place and neatly maintained.

### **Technology Stack: Java, JavaFx**

**Testing:** Manually tested

### **Team:**

Srushti S. Raybhoge (41258)

Sahil S. Patil (41250)

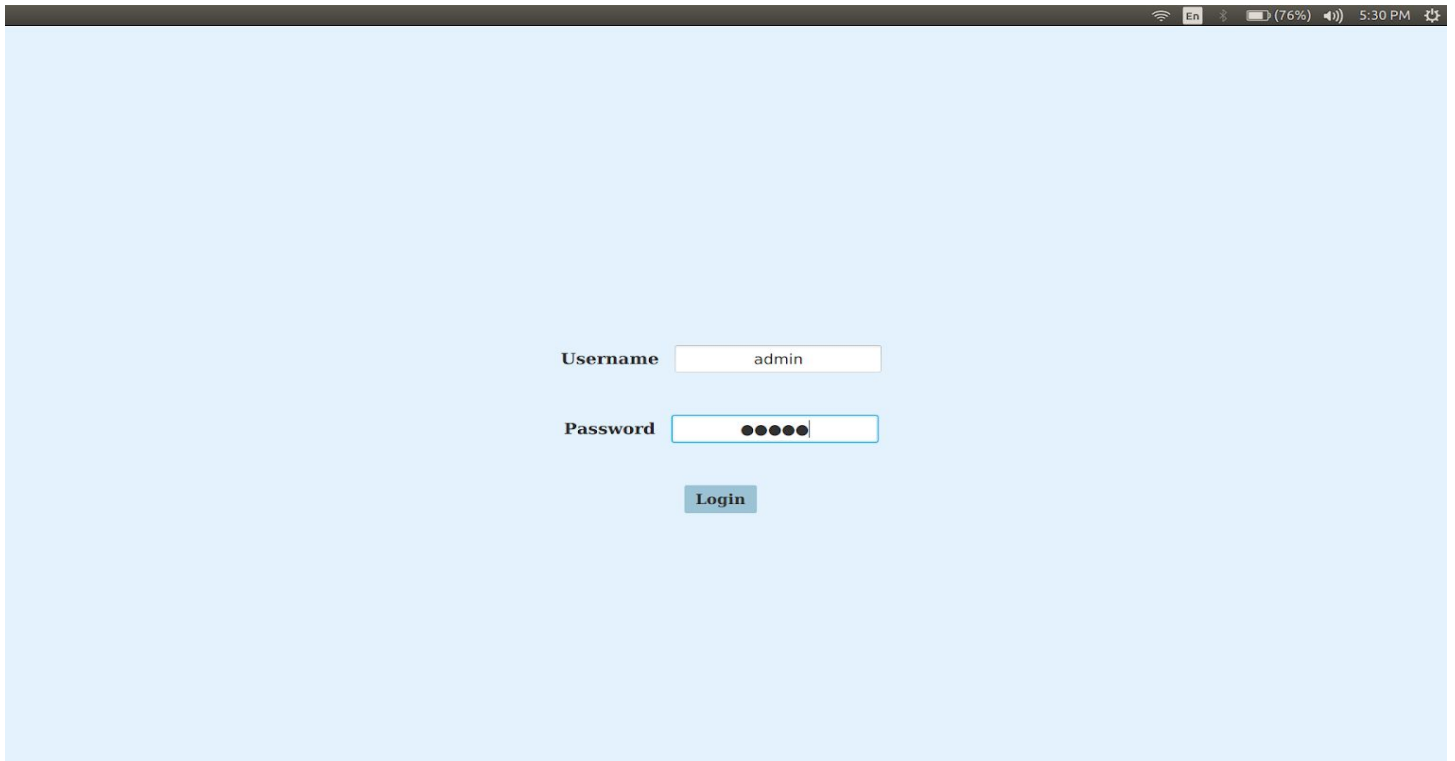
Pralhad A. Kulkarni (41254)

### **Conclusion:**

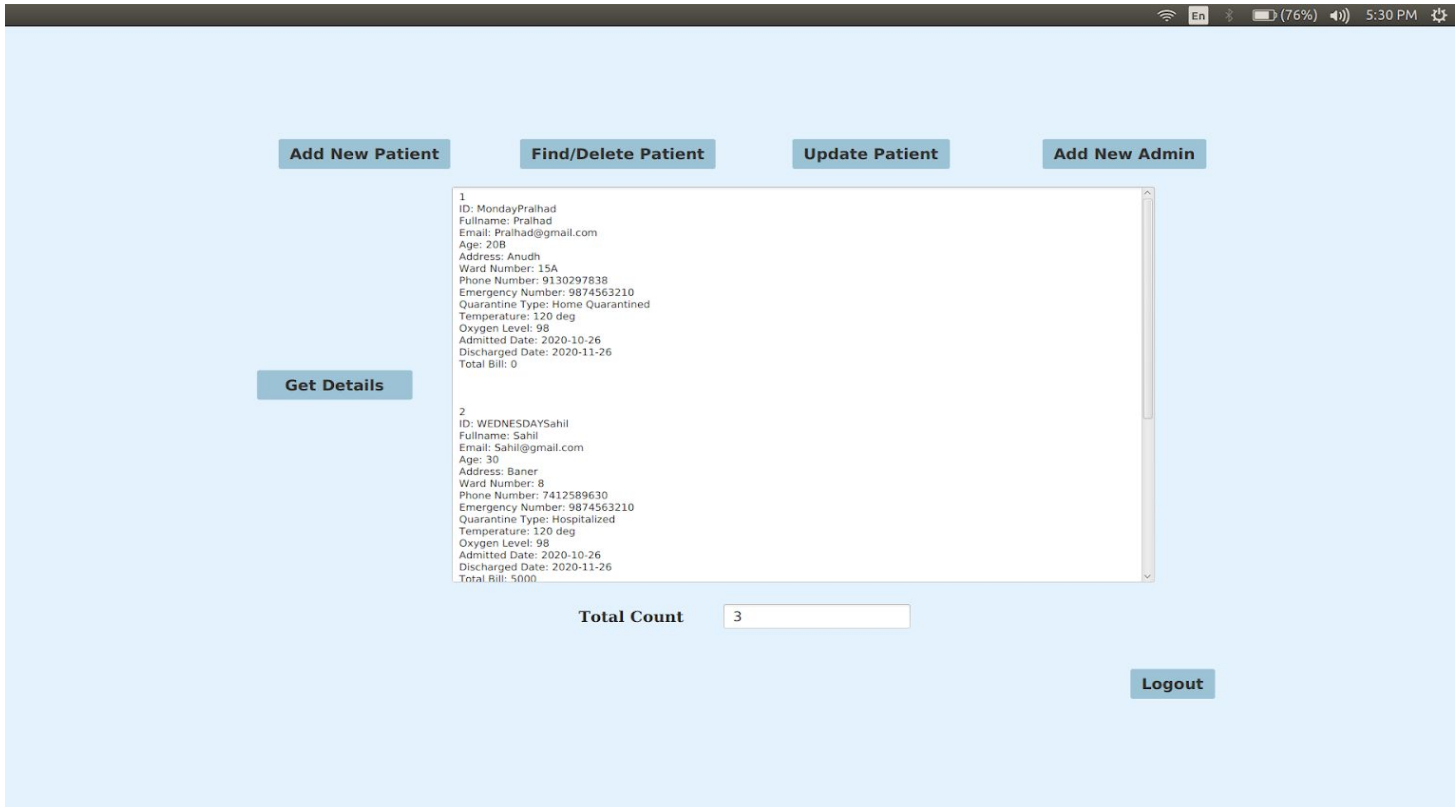
The application can be used in small wards and later scaled to cloud based on the requirements.

# Screenshots of the application:

## LOGIN:



## HOMEPAGE:



# ADD/UPDATE NEW ENTRY:

Full Name

Srushti

Age

20

Email

srushti@gmail.com

Address

Nigdi

Ward

5U

Phone Number

9130297838

Type of Quarantine

Home

Emergency Number

9130297838

Temperature

98

Oxygen Level

98

Date of admission

27/11/2020

Date of Discharge

4/12/2020

Bill Amount

0

Save

Clear

Back

# DELETE ENTRY:

Enter Id

WEDNESDAYSr

Get Details

Details

ID: WEDNESDAYSr  
Fullname: Srushti  
Email: Srushti@gmail.com  
Age: 15  
Address: Yamunanagar, Ni  
Ward Number: 15A  
Phone Number: 91302978  
Emergency Number: 9874  
Quarantine Type: Home Q  
Temperature: 120 deg  
Oxygen Level: 98  
Admitted Date: 2020-10-26  
Discharged Date: 2020-11-26  
Total Bill: 5000

Patient DELETED

Record is deleted.

OK

Delete

Cancel

# ADD NEW ADMIN:

⊞ En 🔊 🔋 (74%) 5:34 PM ⚙

Username

Password

Save

Cancel

	Type	Test Case	Expected Output	Actual Output	Status	Solution/Remark
<b>LOGIN MODULE</b>						
1	Component	Login	User is able to login with given username and password	User is able to login with given username and password	Passed	No Remark
2	Integration	Validations for username and password	Display error if not satisfied	Display error if not satisfied	Passed	No Remark
3	Integration	Validate password from database	Reject or approve login if password is wrong or correct	Rejects or approves login if password is wrong or correct	Passed	No Remark
4	Unit	Need not login again if application is closed	Login Details are saved	Login details are not saved and have to login again	Failed	Save details in local storage
<b>Homepage Module</b>						
5	Integration	Show all the records	Show records	Records are visible after button click	Partially Passed	Get data in constructor and display
6	Unit	Button Clicks	Button clicks redirect to the respective pages	Button clicks redirect to the respective pages	Passed	No Remark
7	Integration	Visibility of records and count	Records and clearly visible and readable. Count is right.	Records and clearly visible and readable. Count is right.	Passed	No Remark
8	Unit	Logout Button	Logs out of the application	Logs out of the application	Passed	No Remark
<b>Form Module</b>						
9	Unit	Visibility and readability of labels and fields	Labels and fields are arranged properly and are readable	Labels and fields are arranged properly and are readable	Passed	No Remark
10	Unit	Validations of fields	Display error message if data is wrong	Displays error message if data is wrong. Few cases are not satisfied	Partially Passed	Validate thoroughly
11	Integration	Save data in database	Data is added to database	Data is added to Database	Passed	No Remark
		Error caught	Catch errors and display	Errors are caught and displayed appropriately	Passed	No Remark
<b>Find/Delete Module</b>						
12	Integration	Find the data on id	Data is found and displayed or show message Data not found	Data is found and displayed or show message Data not found	Passed	No Remark
13	Integration/ Regression	Delete	Delete the data with given id	Deletes the data with the given id	Passed	No Remark
<b>Update Module</b>						
14	Component	Update Data	Update only the given fields	Have to create new entry	Failed	Create a separate Page for upadte
<b>Add Admin Module</b>						
15	Component	Add new admin	Details are checked and added	Details are checked and added	Passsed	No Remark
16	Component	Check if admin already present	Error message if details already present	Error message if details already present	Passsed	No Remark
<b>Other (UI,Database)</b>						
17	UI	Responsiveness	Should fit in different desktops and resize fields accordingly	Fits in different desktop size but does not resize fields accordingly	Partially Passed	Use CSS
18	Database	Accessibility	Can be accessed anywhere anytime	Cannot be accessed anywhere anytime since locally stored.	Failed	Store database on cloud
19	Database	Mysql	Does not need to be installed	Needs to be installed	Failed	Shift to cloud
20	Deployment	Tested on different desktop	Build a .exe and works on different desktop	Build a .exe and works on different desktop	Passed	No Remark