

# **SYSTEM ANALYSIS & DESIGN**

## **OPEN SOURCE STUDY VIRTUAL CLINIC**

### **Software Test Plan (STP)**

#### **Test Plan**

The Virtual Clinic is an integrated care system that manages all stakeholders in the medical domain efficiently. Each feature is made up of a number of use cases, and each use case can be tested in a number of ways.

#### **Software for testing**

The testing of Virtual Clinic is done using **Travis CI**. It is a distributed continuous integration service used to build and test software projects.

#### **Testing Approach**

We have used the SRS and the SDD to prepare the design test cases and their procedures. These tests are designed to verify the documentation previously listed.

To test Virtual Clinic, we have accounted the following testing approaches:

- **Interface Testing**

Virtual Clinic has many modules and components, including, but not limited to, appointments, medical tests, prescriptions, and profile view. Information from these components must be passed between each other and to other components throughout the application. Interface testing will be used to evaluate whether these components and modules pass data and control correctly to one another.

- **Regression Testing**

Virtual Clinic is being developed in Waterfall with the Backflow process model. After each code/build change, our development environment is set up to run all test cases against the current code base with each push to the main repository.

- **Coverage Testing**

As per the SRS, we will be writing our own unit tests on each new code push. Because of this requirement, in addition to other testing methods, we aim to have near 100% code coverage. This means that 100% (or extremely close to 100%) of the source code written passes through a test at some point in the testing suite.

- **Automated Test & Deployment**

Virtual Clinic is being developed and improvised when some bug is found by the developers while using it, as a result, it becomes very important for the team to have automated tests after every code push and automatically deploy it to the server where the application is hosted in order to reduce a lot of manual work and get the new application code running in few minutes itself.

- **Functional Testing**

Virtual Clinic is a frontward-facing web application. Due to the nature of this application, we have conducted manual testing on the web page to ensure that the application reacts and responds in the way that it is expected to.

- **Unit Testing**

Virtual Clinic is utilizing unit and integration testing. All of the unit and integration test cases are constructed with detailed knowledge of the codebase and have been automated to run with each codebase change.

- **System Testing**

Virtual Clinic is a web application and hence will be needed to set up for different hospital management systems and in a different way and according to their needs. Hence we will be testing the application in different environments and verifying its successful working.

## Functional Testing

| Test No. | Function             | Pre Conditions   | Test Description (steps)  | Expected Outcome  | Outcome |
|----------|----------------------|--|---|---|---------|
| 1        | Patient Registration | The system is set up and functional  | 1. User clicks on the register button<br>2. User enter email as username<br>3. User enters the password, first name, last name<br>4. User reenters password<br>5. User clicks register button | The page displays profile page for user and an option to logout                                   | PASS    |
| 2        | System Setup         | The system is visited for the first time   | 1. First time a user enters the email as username<br>2. User enters the password 3. User reenters the password, first name, last name<br>4. User clicks on Register account button            | Page is redirected to admin profile and displays message saying Successfully setup admin account. | PASS    |
| 3        | User Login           | The system is set up and functional  | 1. User enters username and password<br>2. Click the login button   | The page displays the profile of the corresponding user   | PASS    |
| 4        | Add specialty        | The system is set up and functional and is on specialty page Admin has logged in | 1. Admin clicks on add specialty button<br>2. Adds specialty name and description<br>3. Clicks on add specialty button  | Specialty page displays the added specialty   | PASS    |
| 5        | Delete specialty     | The system is set up and functional and is on specialty page Admin has logged in | 1. Admin clicks on delete specialty button<br>2. Cancel confirmation box pops up.<br>3. Admin clicks on deleting the speciality   | Specialty page displays the other specialties that were not deleted                               | PASS    |
| 6        | Add symptom          | System is setup and functional and is on speciality page Admin has logged in     | 1. Admin clicks on add symptom button<br>2. Adds symptom name and description<br>3. Clicks on add symptom button  | Speciality page does not get changed  | PASS    |
| 7        | Delete symptom       | System is setup and functional and is on speciality page Admin has logged in     | 1. Admin clicks on delete speciality button<br>2. Cancel confirmation box pops up.<br>3. Admin clicks on delete the speciality  | Speciality page does not get changed  | PASS    |

|    |                       |   |   |   |      |
|----|-----------------------|---|---|---|------|
| 8  | Employee Registration | System is setup and functional. Admin is logged in  | 1. Admin clicks create new user option<br>2. Admin enters email, password, and fullname<br>3. Admin reenters password<br>4. Admin chooses the role of the new user<br>5. Admin clicks register button | Page displays that account for the user has been successfully created | PASS |
| 9  | Create an appointment | System is setup and functional user is logged in    | 1. User clicks on appointment button<br>2. User clicks on new appointment.<br>3. Fills the appointment form correctly<br>4. user clicks on create appointment   | Appointment gets created  | PASS |
| 10 | Provide consultation  | System is setup and functional doctor is logged in  | 1. Doctor clicks on prescription button<br>2. Fills the prescription form correctly<br>3. Doctor clicks on add prescription   | Prescription gets added   | PASS |
| 11 | Deliver Medicines     | System is setup and functional chemist is logged in | 1. Chemist clicks on prescription button<br>2. Updates its delivery status<br>3. clicks on update prescription  | Delivery status gets updated  | PASS |
| 12 | Perform Lab tests     | System is setup and functional lab is logged in     | 1. Lab clicks on prescription button<br>2. Uploads the test results<br>3. Updates the prescription  | Prescription gets updated   | PASS |
| 13 | Patient display       | System is setup and functional patient is logged in | 1. Patient clicks on medical info button  | Page displays the medical history of patient                          | PASS |
| 14 | Update Profile        | System is setup and functional patient is logged in | 1. User clicks on Update Profile<br>2. User updates the fields<br>3. User clicks on Update Profile button   | Page displays message that the profile is successfully updated        | PASS |
| 15 | Change Password       | System is setup and functional patient is logged in | 1. User is on Change Password page<br>2. User enters current password<br>3. User enters the new password and then reenters the new password.<br>4. User clicks on Change Password button              | Page displays that the password is successfully updated               | PASS |

