# RoboMed

# Software Test Plan

## CSCI-P465/565 (Software Engineering I)

## Project Team 7

### Aravind Ramalingam

### Daksha Nagre

### Eugene Wang

## 1. Overview

### 1.1 Test Objectives

The objectives for testing are: verify that specifications of the software have been meet, discover potential software bugs, and assess overall user experience

### 1.2 Test Environment

We’re creating our software product with mainly with MongoDB, Express.js, React.js, and Node.js. During the first few sprints, we’re testing our code in the web testing module.

### 1.3 Test Personnel

Aravind and Eugene will write test functions.

Daksha will check if these functions are designed correctly and run them.

Level of involvement should be roughly equal for everyone.

### 1.4 Acceptance Criteria

The minimum acceptable/deliverable software must have these basic features:

- Login and signup

- Set up and sell insurance: insurance provider users

- Buy insurance and schedule doctor appointments: patient users

- Post schedule and accept appointment requests: doctor users

- Instant messaging system between all user types

- Two factor authentication login

- Email and identity confirmation

- Oauth Login

- Captcha

- Search

- Profile page and review system

### 1.5 Noted Omissions

Features not included above are:

- Review system for doctors and insurance providers

- Map visual for doctor’s office in appointment scheduler

- Customizable insurance quote calculator/formula for insurance providers

- Recommendation algorithm designed with machine learning

- Billing and payment feature

## 2. Test Cases

**Number:** 1

**Name:** Login, Sign up, and Verifications

**Description:** All the flows in the activity of login and sign up should be in place. All types of user should be able to sign up, verify email, verify identity, login with email and password, and login with two-factor authentication

**Initial Conditions:** Before performing the tests, we assume we already have a working page with a prompt for user to login or sign up.

**Input Data:**  User full name, email, password, account type selection, personal info, identity documents

**Specifications:**

· Login feature should redirect user to homepage if he entered the correct email and password and completion of second factor authentication(optional).

· An email-confirmation email should be sent to the user after he fills out and submits the sign up form. User should be redirected to homepage if he submits back the correct email-confirmation code.

· If not already identity-verified, a user will be prompted to fill out a detailed personal information page and upload identity proving documents; after he submits, the user’s account status will be set to “verified” if those information checks out.

· An optional TFA login can be set up and enabled by the user. If enabled the user will have to perform TFA check every time he login in the future.

· User can choose to login with OAUTH.

· Captcha during login after 3 failed attempts

**Procedure:**  https://robomed.herokuapp.com

**Number:** 2

**Name:** Dashboard view

**Description:** A dashboard/homepage view for each user type. Relevant information to each user type should be included in the dashboard. Links to use the relevant functionalities should be displayed for each user.

**Initial Conditions:** There are no preconditions for this test.

**Input Data:**  No input data, user can only click around.

**Specifications:**

· For patient users, display of current insurance, scheduled appointments, buy/change insurance provider function, doctor lookup/scheduler function, and instant messaging function.

· For doctor users, display of scheduled appointments, scheduler editor function, and instant messaging function.

· For insurance providers, display financials from currently active insurances, insurance offers editor, insurance quote formula customizer, and instant messaging function.

**Procedure:**  <https://robomed.herokuapp.com/dashboard> (login required)

**Number:** 3

**Name:** Doctor’s appointment scheduler

**Description:** This feature facilitates communication between doctor users and patient users. Each user type will have access to different functionalities of the scheduler.

**Initial Conditions:** This test assumes we have a working dashboard.

**Input Data:**  Patient user input: doctor’s name, location info.

Doctor user input: available time slots

**Specifications:**

· For patient users: They should be able to enter a doctor’s name to search the doctor or they can enter a location information and get a list of doctors info in the area. Then they can select a doctor, view his available times and send an appointment request to the doctor.

· For doctor users: They should be able to edit the schedule shown to patient users, and accept or deny appointment requests.

**Procedure:**  <https://robomed.herokuapp.com/dashboard> (login required)

**Number:** 4

**Name:** Buy and sell insurance

**Description:** Setting up insurance offers and buying insurances.

**Initial Conditions:** This test assumes we have a working dashboard.

**Input Data:**  For patient user: application information, For insurance providers: formula to calculate insurance offers

**Specifications:**

· For patient users, they should be able to search and filter insurance offers, get a custom insurance quote, and submission of an application

· For insurance providers, they can edit their insurance offers, edit the formula that gives patients custom quotes, view patient application, accept/deny patient application

**Procedure:**  N/A #Provide a link to the Test Case Procedure for this Test Case

**Number:** 5

**Name:** Instant Messaging

**Description:** Real time chat system for patient/doctor, and patient/insurance providers.

**Initial Conditions:** This test assumes we have a working dashboard.

**Input Data:**  Custom messages.

**Specifications:**

· For patient users, they should be able to send a message to any doctor or insurance providers.

· For doctor users, they should be able to send a message to former patients, future patients, or a patient user who has sent him a message first.

· For insurance providers, they can send messages to patient users who have sent them a message first.

**Procedure:**  N/A #Provide a link to the Test Case Procedure for this Test Case

**Number:** 6

**Name:** Profile Page and Review System

**Description:**

Allow all users to adjust profile information. Allows patient users to view doctor and insurance providers’ profile page and give a review. Allows insurance providers to set customizable weights that adjusts the formula that calculates their insurance price quote unique to each customer.

**Initial Conditions:** This test assumes we have a working dashboard.

**Input Data:**  Name of doctors and weights, insurance providers, rating score, review comments, and other profile information.

**Specifications:**

· For patient users, they should be able to view any doctor and insurance providers profile page and leave a rating and comment for them.

· For doctor and insurance provider users, they should be able to edit the information displayed on their profile page

**Procedure:**  <http://robomed.herokuapp.com/dashboard/profile>

**Number:** 7

**Name:** Purchase & Recommendation

**Description:** Recommends and sells insurance to patients

**Initial Conditions:** This test assumes we have a working dashboard.

**Input Data:**  The patient will customize (input) his desired coverages for each category of things covered by the insurance.

**Specifications:**

· For patient users, they purchase the insurance by first filling out a form that lists the max coverage for each insured category of things. Base of these coverage dollars amount, we will cross check the offer price for the desired plan against all the insurance providers on the website. At last we recommend the cheapest options to the patient

**Procedure:**  N/A #Provide a link to the Test Case Procedure for this Test Case

**Number:** 8

**Name:** Billing and Payment

**Description:** A system that tracks and facilitates payment between different user types

**Initial Conditions:** This test assumes we have a working dashboard.

**Input Data:**  Doctor’s appointment cost, insurance plan cost, insurance deductible cost

**Specifications:**

· When a doctor finishes an appointment report with the expenses, a bill with that expense gets sent from the doctor to the insurance provider for the full amount of the doctor’s charge and a bill gets sent from the insurance provider to the patient for the amount of the deductible.

When a patient buys an insurance plan, a bill get sent from the insurance provider to patient for the monthly insurance fee.

**Procedure:**  N/A #Provide a link to the Test Case Procedure for this Test Case

**Number:** 9

**Name:** Chat

**Description:** Let users communicate with each other

**Initial Conditions:** This test assumes we have a working dashboard.

**Input Data:**  Patient user’s text input

**Specifications:** A group chat for all doctors, and private chat between patients and doctors/insurance-providers. Patients can contact any doctors and insurance providers, but not other patients. Doctors can only chat with patients that initiate a chat or had an appointment with him. Insurance providers can only chat with patients that initiated a chat with them. A chatroom get created when a new chat is initialized.

**Procedure:**  N/A #Provide a link to the Test Case Procedure for this Test Case

**Number:** 9

**Name:** Statistics

**Description:** Show relevant metrics to users

**Initial Conditions:** This test assumes we have a working dashboard.

**Input Data:**  No user input needed. Data used are all from the database

**Specifications:** For insurance provider: number of active plan subscribers, average monthly price of plan, monthly revenue chart, monthly insurance payout, month-over-month revenue growth

For doctor: chart for monthly income and number of appointments

For patient: Amount saved with insurance plan (insurance payout)

**Procedure:**  N/A #Provide a link to the Test Case Procedure for this Test Case

## Revision History

|  |  |  |
| --- | --- | --- |
| Revision | Date | Change Description |
| Added oauth, captcha, profile, and review as necessary features. Billing changed to optional feature | 10/19/2019 | none |
| Links added. | 11/2/2019 |  |
| Purchase and recommendation test case added | 11/18/2019 |  |
| Statistics, Chat, and Billing added | 12/7/2019 |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |