

IT314

SOFTWARE ENGINEERING

Project: Health Center Management

Group: 12

Graphical User Interface Testing

White Box Testing

Tool : Android Studio

Language : Kotlin

Framework : Espresso

Reason : Android Espresso is a popular UI testing framework that can be written in Java and Kotlin. Espresso allows us to test a larger portion of the UI with much less code. It automatically verifies the expected behavior of GUI elements. It is a quick and efficient tool for UI testing especially in android. Also, it has a large community.

Tests:

Screen 1: Login and signup

Tests: UI automation tests are written for checking valid and invalid logins, and also checking whether a user is able to redirect to the sign up page or not when chosen.

Number of tests : 3

Results:

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The project is named "IT314_Health_Center - LoginTest". It includes an "app" module with "manifests", "java", and "test" sub-directories.
- Code Editor:** The main editor window displays the code for `LoginTest.java`. The code is annotated with `@RunWith(AndroidJUnit4.class)` and `@LargeTest`. It contains a test method `check_valid_Login()` which launches the application and performs UI interactions on the login screen.
- Run Tab:** The "Run" tab shows the test results for `LoginTest`, indicating 3 passed tests out of 3.
- Test Results:** The "Test Results" section shows the build was successful in 42s, with 66 actionable tasks: 2 executed, 64 up-to-date.
- Bottom Status Bar:** Shows the build time (43 s 52 ms), Gradle version (master), and system status (62:69, CRLF, UTF-8, ENG US, 03:56, 27-04-2023).

Status: All passed

Screen 2: Patient Homepage

Tests: UI automation tests are written for checking whether all elements in HomePage of Patient are responding or not, and whether correct screens are displayed or not.

Number of tests : 3

Results:

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The left sidebar shows the project structure under the `:app` module. It includes `AndroidManifest.xml`, `java` (containing `com.mypackage.it314_health_center`), `res`, and `src` (containing `androidTest` and `test`).
- Code Editor:** The main editor window displays the `HomePageTest.kt` file, which contains three test methods: `canNavigate()`, `isNotificationOpen()`, and `checkAllPages()`.
- Run Tab:** The bottom-left tab bar shows the `Run` tab is selected, with the `HomePageTest` configuration highlighted.
- Test Results:** The bottom-right panel shows the test results for the `HomePageTest`. It lists three tests: `canNavigate()`, `checkAllPages()`, and `isNotificationOpen()`, all of which passed (duration: 26s, 19s, 4s respectively). The overall build status is **BUILD SUCCESSFUL in 51s**.
- Notifications:** A floating notification from Android Studio indicates **Tests Passed 3 passed**.

Status: All passed

Screen 3: Appointment Booking

Tests: UI automation tests are written for checking whether date and time pickers are working for booking appointments and whether the interface for canceling and booking appointments is working or not.

Number of tests : 2

Results:

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** Shows the project tree with modules like `:app`, `java`, and `com.mypackage.it314_health_center`.
- Code Editor:** Displays the `AppointmentBookingTest.kt` file containing Java code for UI automation tests.
- Run Tab:** Shows the test results for `AppointmentBookingTest`. It indicates 2 passed tests, 2 tests total, and a duration of 46 ms.
- Log Tab:** Shows the build log: "BUILD SUCCESSFUL in 46s" and "66 actionable tasks: 1 executed, 65 up-to-date".
- Status Bar:** Shows "Tests Passed 2 passed" and other system information like "74.1 CRLF UTF-8 4 spaces ENG US 27-04-2023".

Status: All passed

Screen 4: Settings screen

Tests: UI automation tests are written for checking night mode and notification switches are working, and whether help and faq page is displayed.

Number of tests : 1

Results:

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** Shows the project tree for "IT314_Health_Center".
- Code Editor:** Displays the Java code for "SettingsTest.kt" under "com.mypackage.it314_health_center". The code includes imports, annotations (@RunWith, @LargeTest), and a test method "check_Settings" that launches an activity and performs UI checks.
- Run Tab:** Shows a green status bar indicating "1 passed" for 1 test in 30s.
- Test Results:** Shows a summary of the build: "BUILD SUCCESSFUL in 30s", "66 actionable tasks: 1 executed, 65 up-to-date", and "Build Analyzer results available".
- Bottom Bar:** Includes icons for Git, Run, Profiler, Logcat, App Quality Insights, Build, TODO, Problems, Terminal, Services, App Inspection, and a search bar.

Status: All passed

Black Box Testing

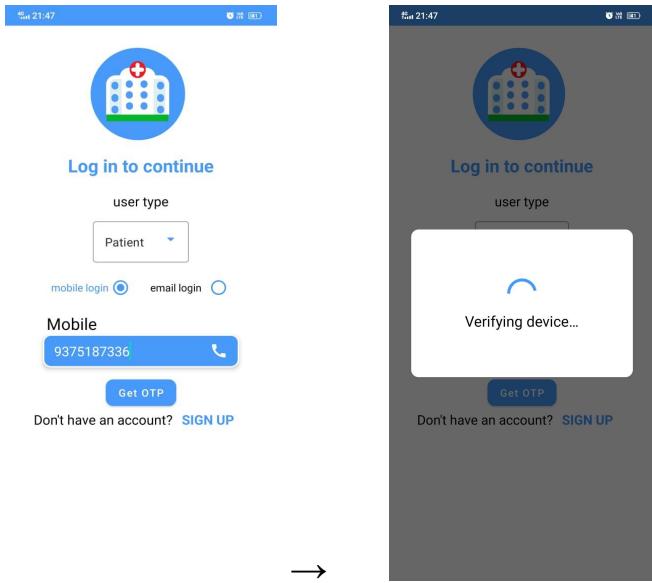
(1) Phone number validation:

Description: This test case tests whether the phone number entered by the user is valid or not and corresponding to that output is shown or not.

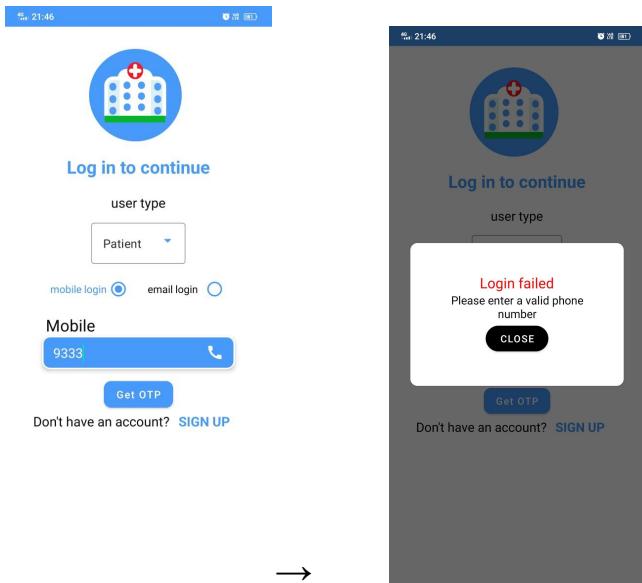
Expected Output: When the user enters a valid mobile number then the verifying process will be shown and if entered incorrectly then an error message should pop up.

Actual output:

If entered phone number is valid then,



If it is invalid then,



Status: Passed

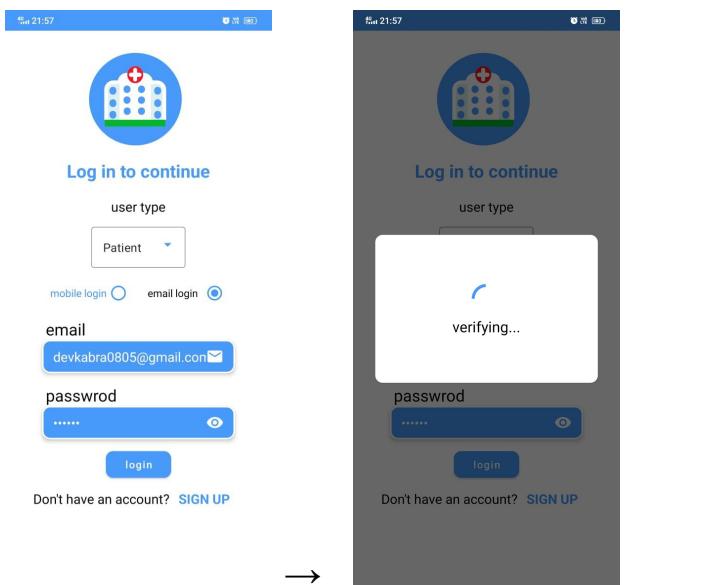
(2) Email validation:

Description: This test case tests whether the email entered by the user is valid or not and corresponding to that output is shown or not.

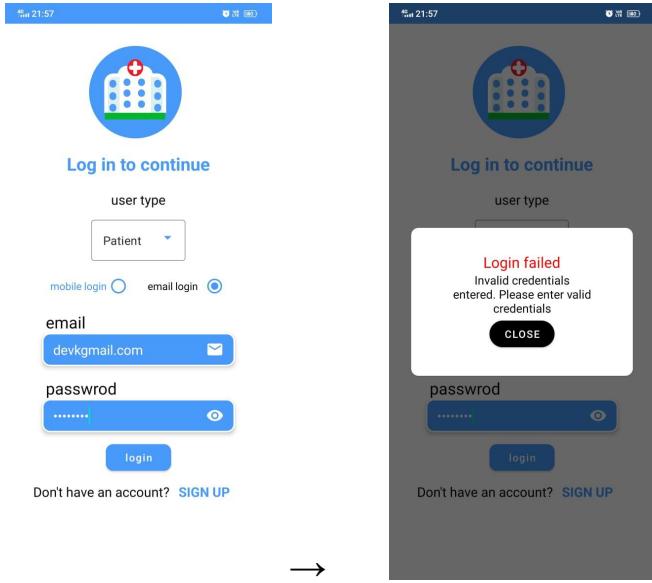
Expected Output: When the user enters a valid email then verifying process will be shown and if the email is entered incorrectly then an error message should pop up.

Actual output:

If entered email is valid then,



If it is invalid then,



Status: Passed

(3) Checking the password hide/show button:

Description: When the user clicks the password hide/show button then does it show the output corresponding to its current state.

Expected output: When the user clicks on the button, if its initially in the hide state then after clicking on it the password entered should be visible to the user and if it's initially in the show state then after clicking on it the password should get hidden.

Actual output:

If it's in hidden state and after clicking on it,



Status: Passed

(4) Checking the user type drop down menu:

Description: This tests whether the user can select the user types as per his/her preference.

Expected Output: When the user clicks on the drop down menu then it should show all the 3 user types: Patient, Doctor and Manager and when the user selects one of them then the drop down menu should get closed and the selected option should be visible in the drop down menu.

Actual Output:

When the drop down menu is clicked and an option is selected,

The figure consists of three side-by-side screenshots of a mobile application's login interface. Each screenshot shows a blue circular logo with a white hospital building icon and the text "Log in to continue". Below the logo is a dropdown menu labeled "user type" with "Patient" selected. At the bottom of each screen is a "Mobile" section containing a text input field "Enter mobile number" with a phone icon, a "Get OTP" button, and a "SIGN UP" link.

- Screenshot 1:** Shows the initial state where "Patient" is selected in the dropdown.
- Screenshot 2:** Shows the dropdown expanded to show "Patient", "Doctor", and "Manager" options. "Patient" is still selected.
- Screenshot 3:** Shows the dropdown expanded again, with "Doctor" now selected.

Below the dropdown in all three screenshots are two radio buttons: "mobile login" (selected) and "email login".

Status: Passed

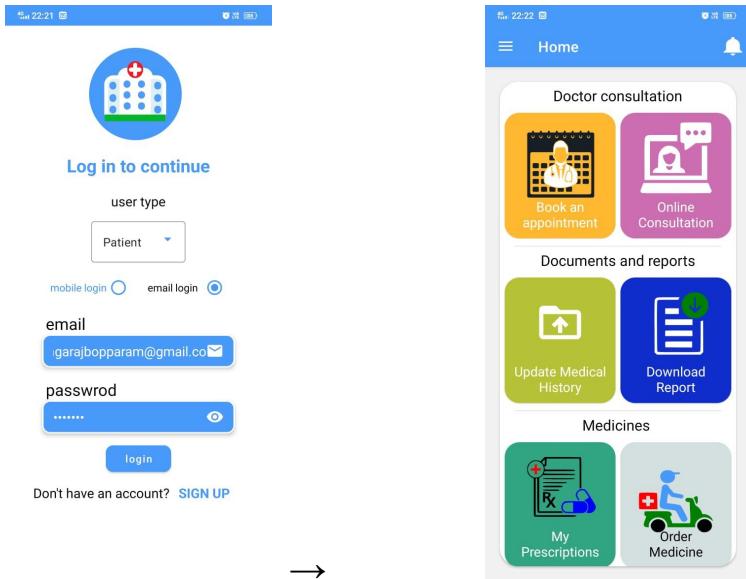
(5) Password validation:

Description: This tests whether the password entered by the user is valid or not.

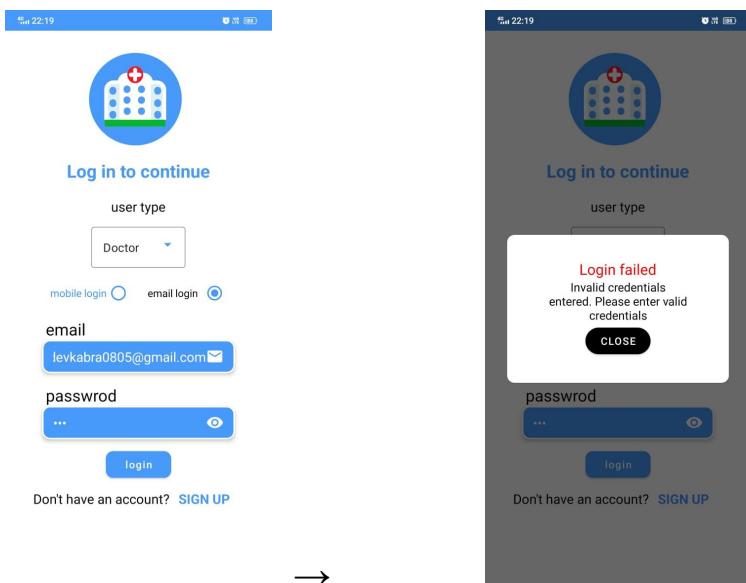
Expected Output: The password entered by the user should be of length greater than equal to 5 or else it will show an error message to enter a valid password.

Actual Output:

When password entered is valid,



When it is invalid,



Status: Passed

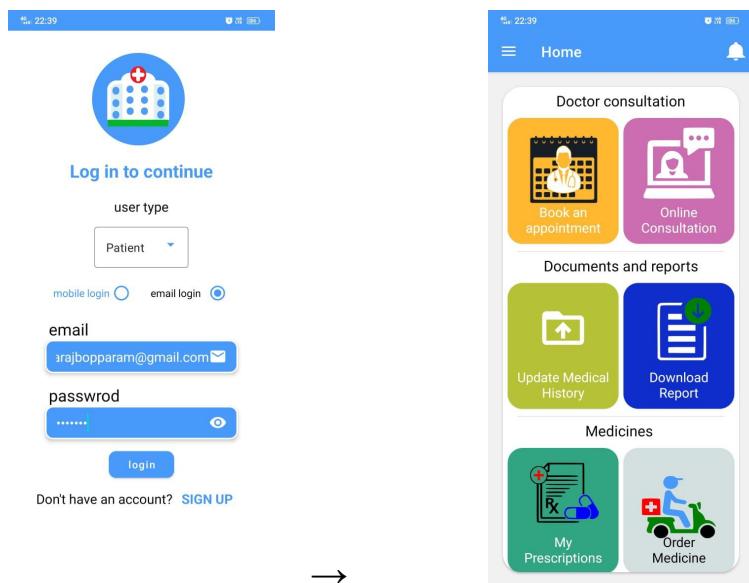
(6) Testing login button:

Description: It tests whether when clicking on the login button a valid user should be able to enter and an invalid user should not be able to proceed.

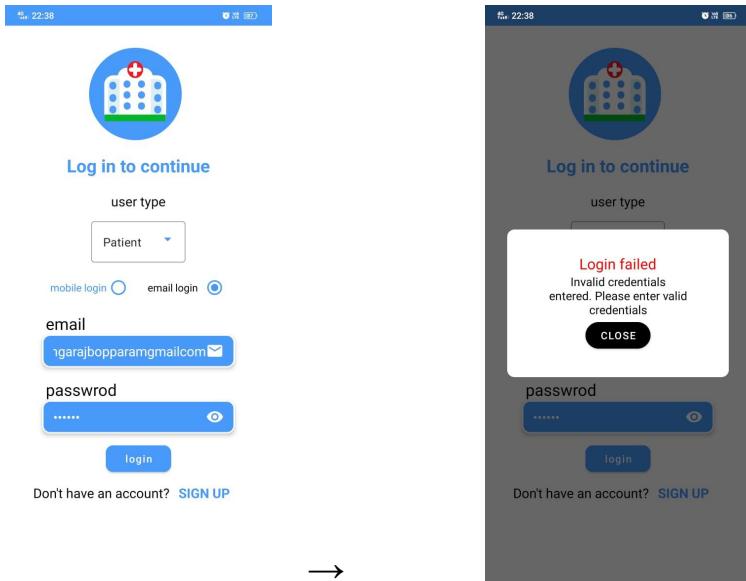
Expected Output: When the user clicks on the login button then it should show the main home page if and only if all the entries by the user are valid and if any of them are invalid then it should show an error message corresponding to that field.

Actual Output:

When all entries are valid and then Login button is clicked,



When any of the entry is invalid and then Login button is clicked,



Status: Passed

(7) Password and confirm password validation:

Description: This tests whether the password entered in the Password and Confirm Password field is the same or not.

Expected Output: When the user enters the same password in both of them then after clicking on it should ask the user to verify the mail and if not equal then it should display an error message.

Actual Output:

When both are equal,

22:45

22:45



sign up to continue

mobile signup email signup

email

helloworld@gmail.com

password

123456

confirm password

123456

Verify Your E-mail

Verify the email and Log in with your credentials. If you can't find the email in your mailbox, kindly check in spam folder



[signup](#)



When both are different,

22:44

22:44



sign up to continue

mobile signup email signup

email

levkabra0805@gmail.com

password

123456

Select Select all Input method ►

12345y7

[signup](#)

sign up to continue

Signup Failed

password and confirm password are not matching. Please enter correct details

[CLOSE](#)

confirm password

12345y7

[signup](#)



Status: Passed

Module 2: Patient Home page

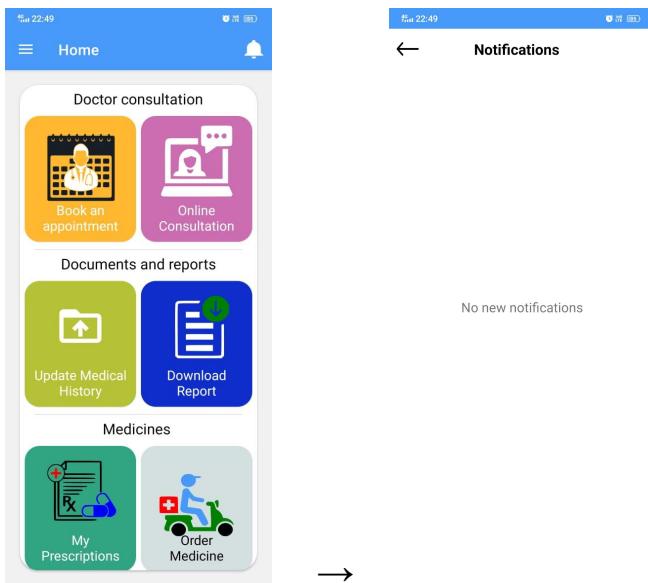
(1) Testing notification button:

Description: This tests whether the notification button is working properly or not.

Expected Output: When the user clicks on the notification button, then it should open a new screen where all the recent notifications should be displayed if there are any.

Actual Output:

When notification button is clicked,



Status: Passed

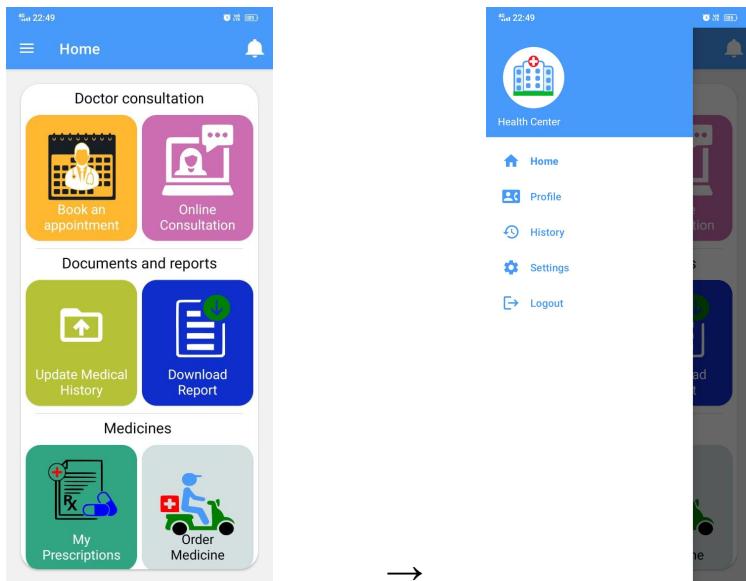
(2) Testing navigation button:

Description: This tests whether the navigation button is working properly or not.

Expected Output: When the user clicks on the navigation button on the top left side, then it should display all the options available such as Home, Profile, History, etc.

Actual Output:

When navigation button is clicked,



Status: Passed

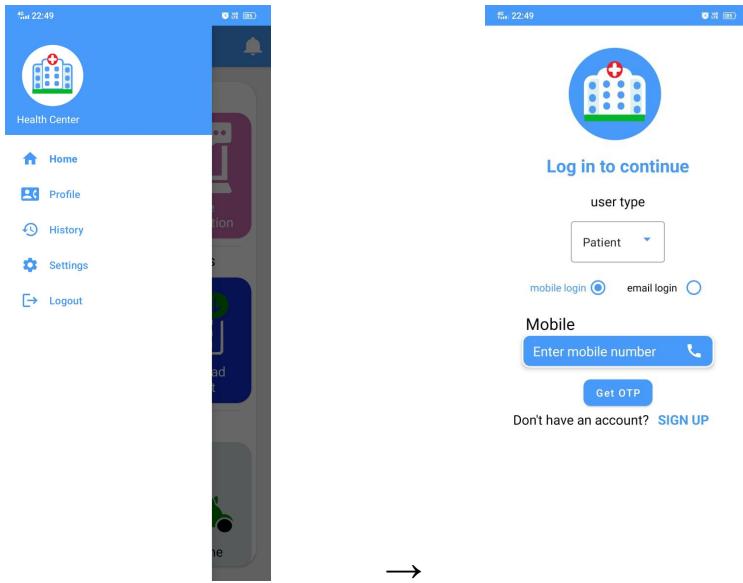
(3) Testing Logout option:

Description: This tests whether the Logout option logs out the user of the application or not.

Expected Output: When the user clicks on the logout button(which is in the sidebar which opens up after clicking on the navigation button) then it should take the user to the Login page.

Actual Output:

When Logout option is clicked,



Status: Passed

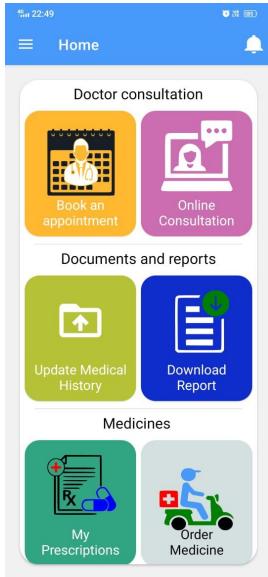
(4) Testing the visibility of the elements/options on the home page:

Description: This tests whether all the options on the home page are clearly visible to the user.

Expected Output: When the patient home page opens up then all the options such as Book Appointment, Download reports, etc. should be clearly visible to the user such that the user can select any one of them.

Actual Output:

Visibility of Patient home page,



Status: Passed

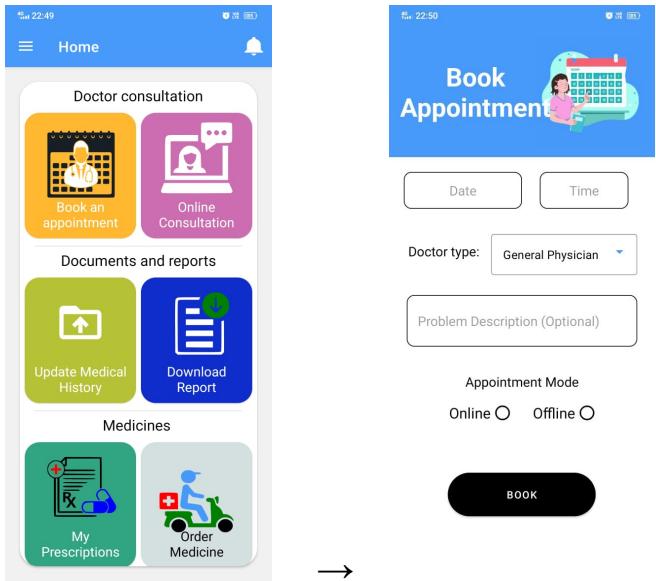
(5) Testing each element/option on the home page:

Description: This tests whether on clicking any option on the home page it works properly or not.

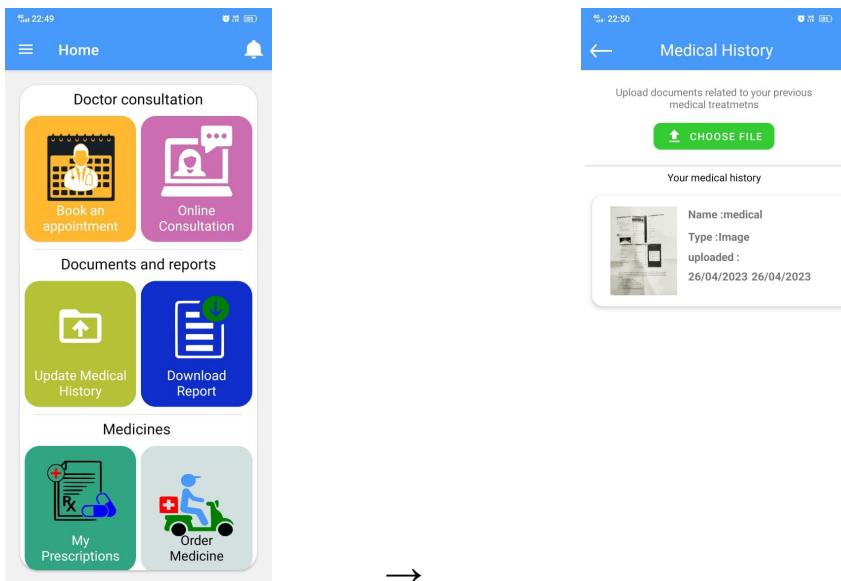
Expected Output: When the user clicks on any option then it should open a new screen as per the selected option, for example if the user clicks on Book Appointment option then it should display the screen to book an appointment.

Actual Output:

When Book Appointment is clicked,



When Update Medical History is clicked,



Status: Passed

Module 3: Book Appointment

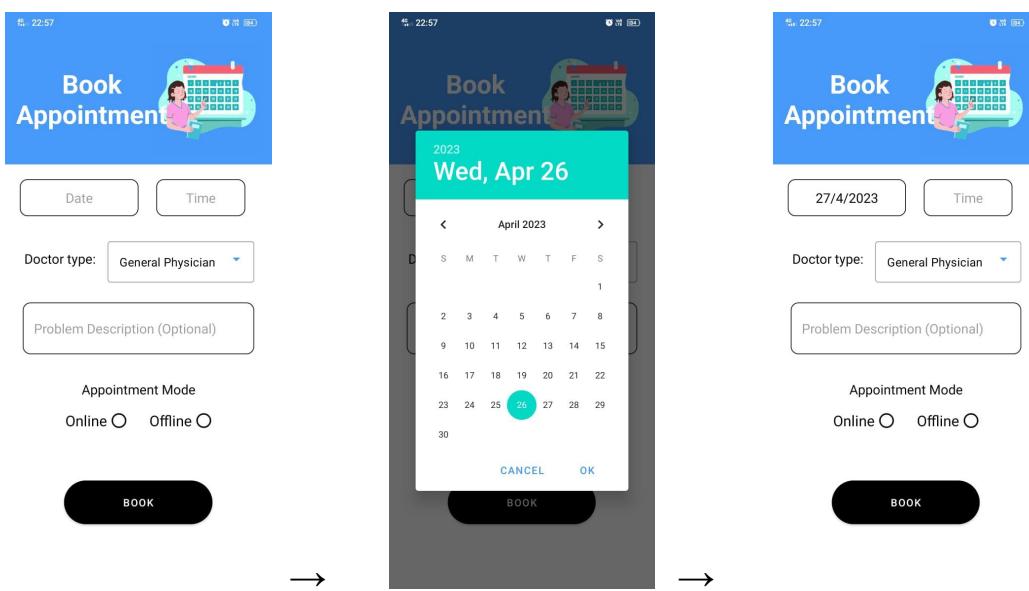
(1) Testing date field:

Description: This tests whether the date field has a valid value or not and does it work properly or not.

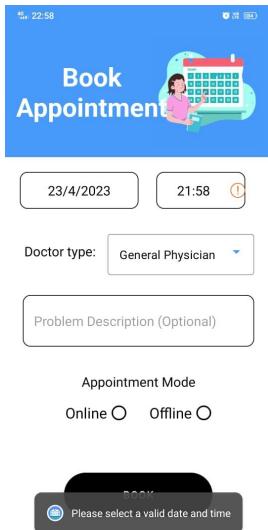
Expected Output: When the user clicks on the date field then a date selector should open up and then the user should select a date which can be today or later and if the user selects a previous date then when clicking on the Book button it should show an error message. When a date is selected then that date should be displayed in the field.

Actual Output:

When the date field is clicked and a date is selected,



When user selects a previous date,



Status: Passed

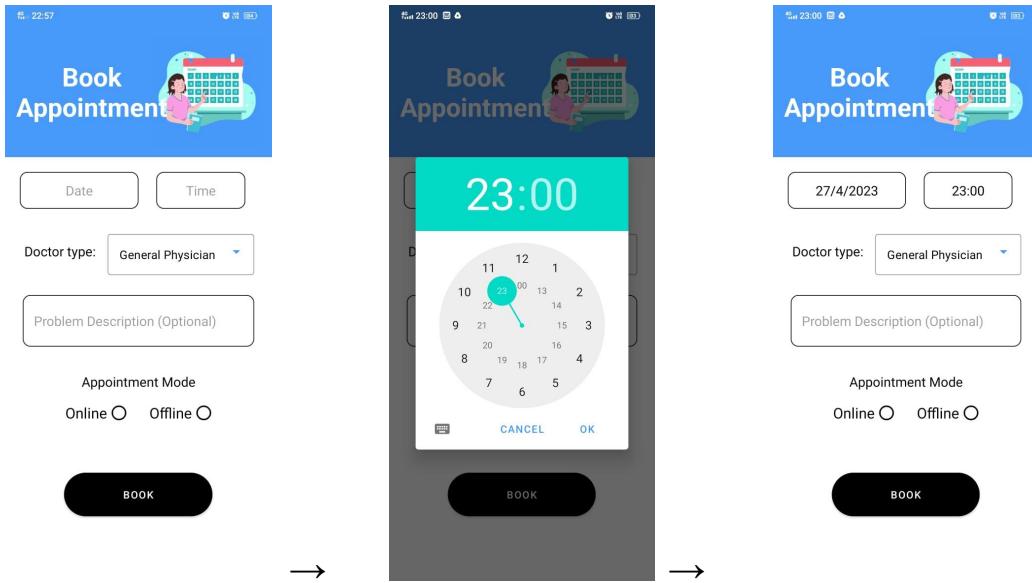
(2) Testing time field:

Description: This tests whether the time field has a valid value or not and does it work properly or not.

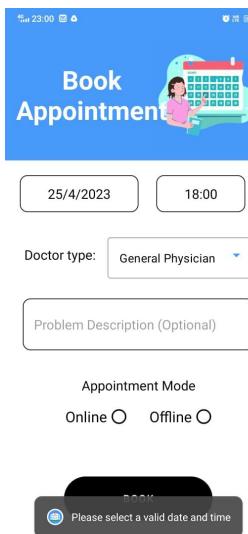
Expected Output: When the user clicks on the time field then a 24 hour time selector should open up and then the user should select a valid time and if it is invalid then clicking on the Book button it should show an error message. When the time is selected then that time should be displayed in the field.

Actual Output:

When the time field is clicked and a time is selected,



When user selects invalid time,



Status: Passed

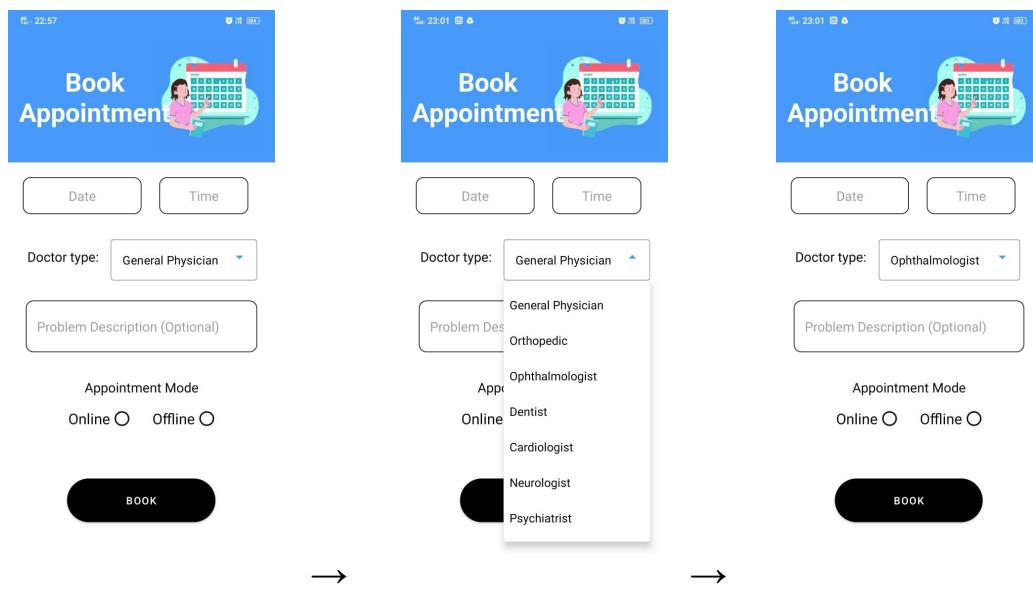
(3) Checking the doctor type drop down menu:

Description: This tests whether the user can select the doctor types as per his/her preference.

Expected Output: When the user clicks on the drop down menu then it should show all the types of doctors that are available for the appointment and when the user selects one of them then the drop down menu should get closed and the selected option should be visible in the drop down menu.

Actual Output:

When the drop down menu is clicked and an option is selected,



Status: Passed

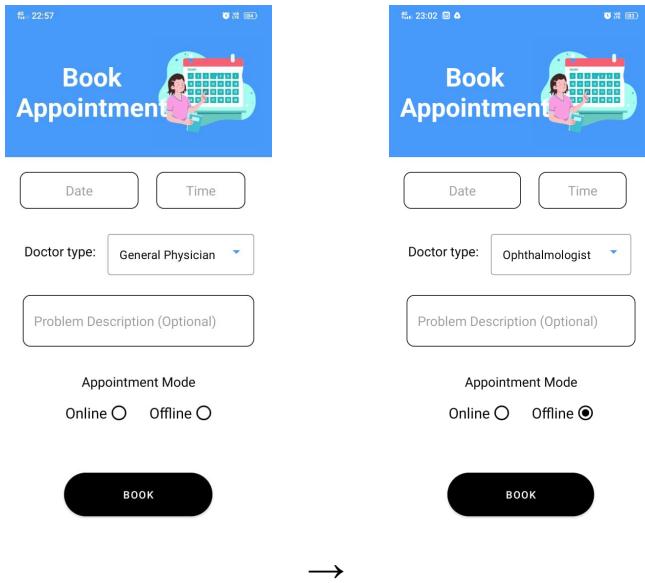
(4) Testing the radio button:

Description: This tests whether the radio button works properly or not, when the user selects the appointment mode.

Expected Output: When the user selects an option then that circle should get darkened.

Actual Output:

When any option is selected,



Status: Passed

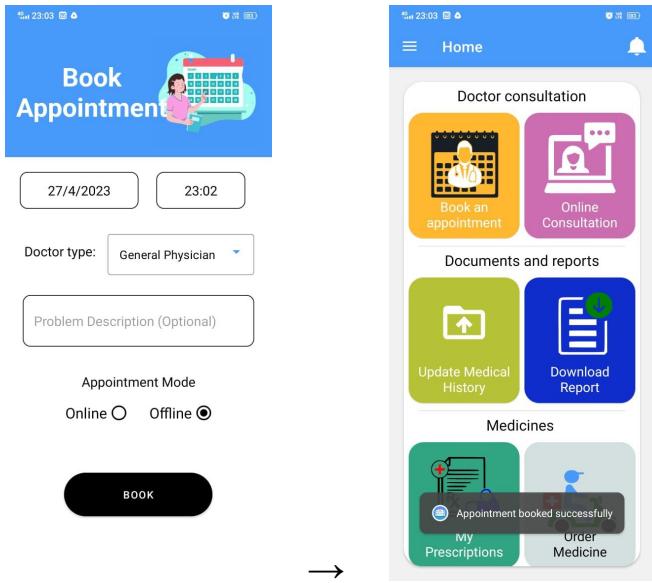
(5) Testing book button:

Description: It tests whether when clicking on the book button a valid user should be able to enter and an invalid user should not be able to proceed.

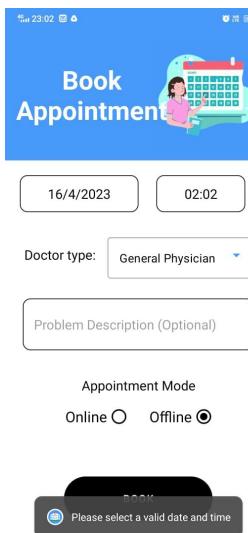
Expected Output: When the user clicks on the book button then it should show the confirmation message that the appointment is booked if and only if all the entries by the user are valid and if any of them are invalid then it should show an error message corresponding to that field.

Actual Output:

When all entries are valid and then Book button is clicked,



When any of the entry is invalid and then Book button is clicked,



Status: Passed

Module 4: Order Medicine

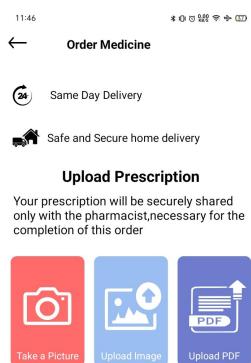
(1) Testing the visibility of the elements/options on the home page:

Description: This tests whether all the options on the order medicine page are clearly visible to the user.

Expected Output: When the order medicine page opens up then all the options such as select pdf, image , etc. should be clearly visible to the user such that the user can select any one of them.

Actual Output:

Visibility of order medicine page,



Status: Passed

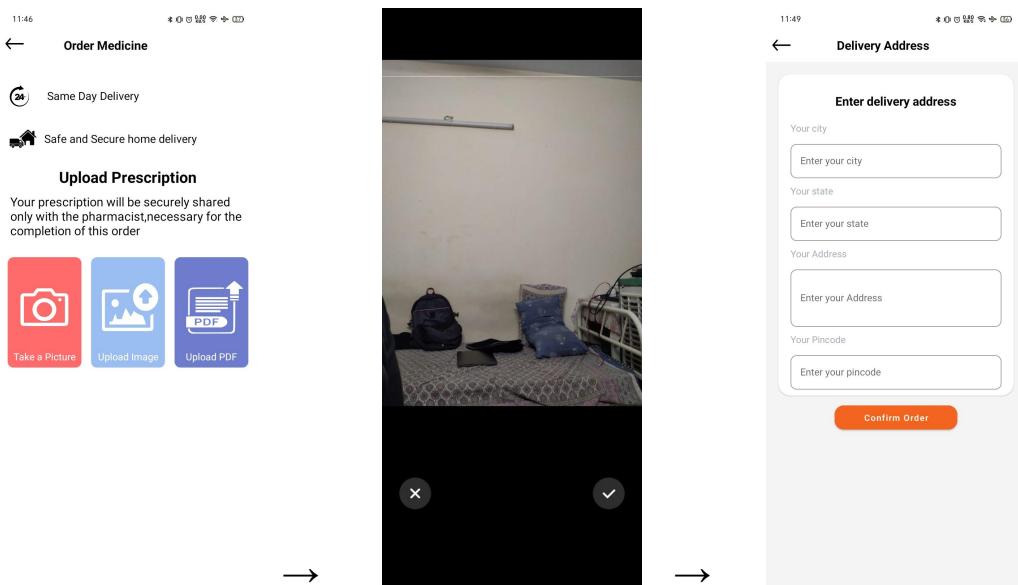
(2) Testing camera:

Description: This tests whether the camera is working properly or not.

Expected Output: When the user clicks on the take a picture option then the camera should open up and when the user takes a picture then after selecting that picture the delivery information page should be displayed.

Actual Output:

When the take a picture option is clicked and when the picture is selected,



Status: Passed

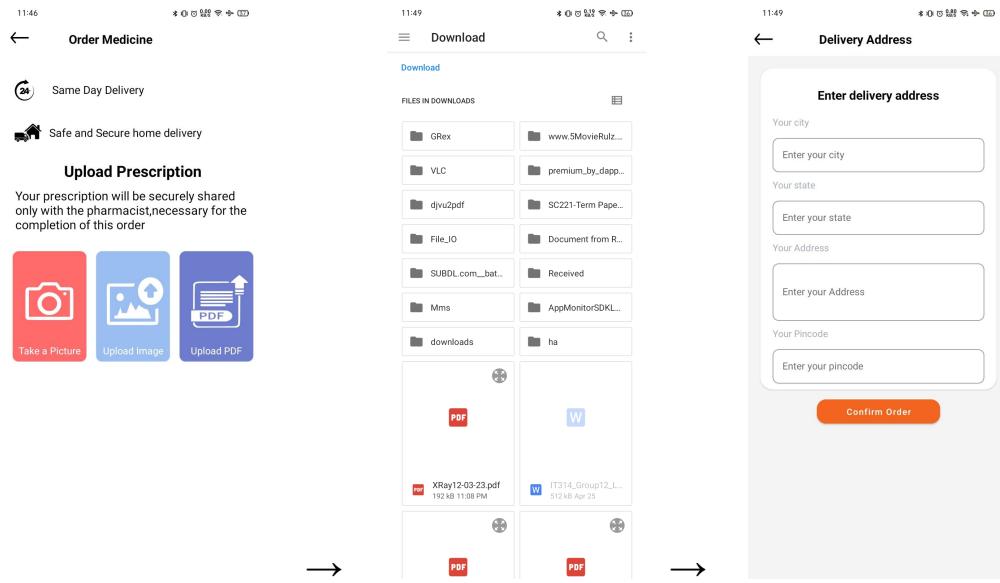
(3) Testing upload image and pdf options:

Description: This tests whether upload image and pdf options are working properly or not when a file is selected.

Expected Output: When the user clicks on the upload an image or pdf option then the file manager should open up and when the user selects a file, the delivery information page should be displayed.

Actual Output:

When the upload a image or pdf option is clicked and when the file is selected,



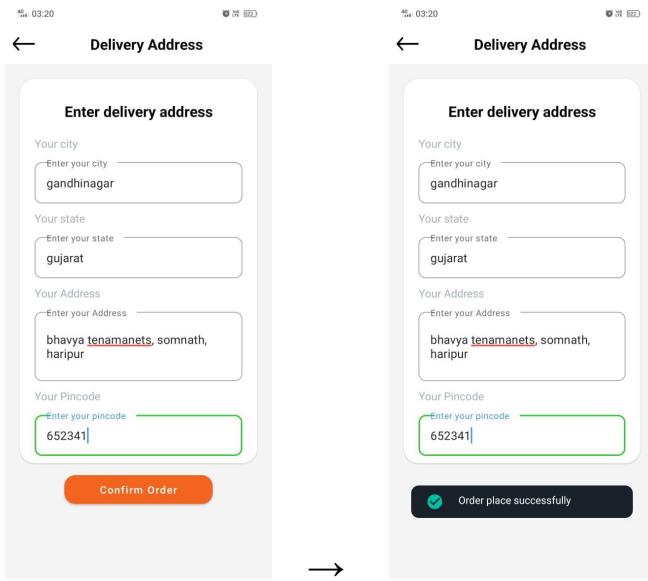
(4) City, Address and State validation:

Description: This test case tests whether the city, address and state entered by the user is valid or not and corresponding to that output is shown or not.

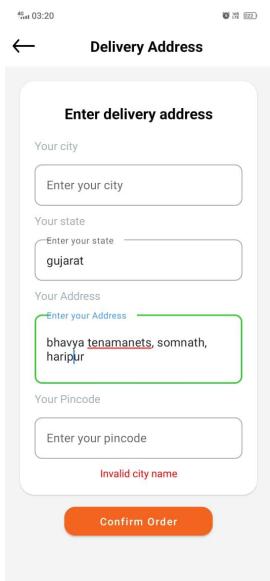
Expected Output: When the user enters a valid city, address and state (that is not empty) then delivery should get confirmed and a message should be sent to the user's mobile and if entered incorrectly then an error message should pop up.

Actual output:

If entered all the fields are valid then,



If they are invalid then,



Status: Passed

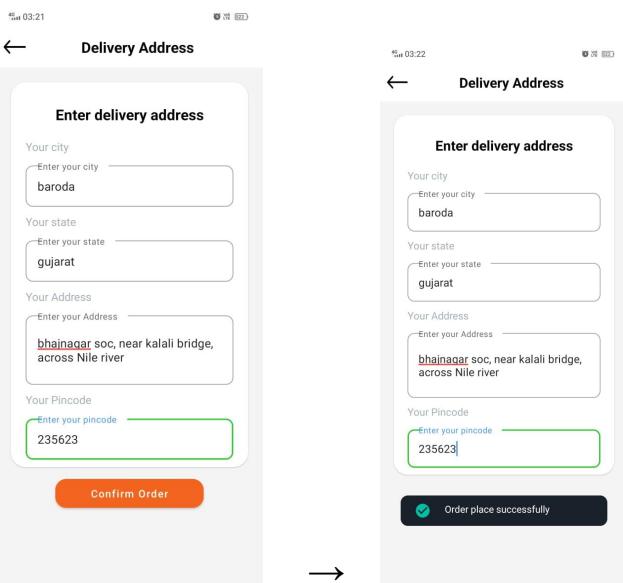
(5) Pin code validation:

Description: This test case tests whether the pin code entered by the user is valid or not and corresponding to that output is shown or not.

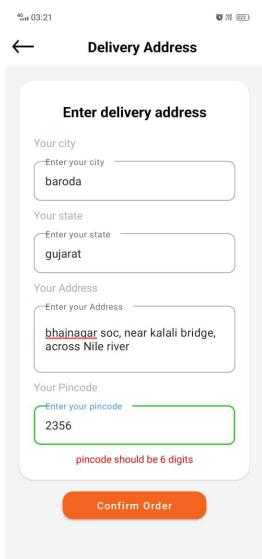
Expected Output: When the user enters a valid pin code then the delivery confirmation message should be sent to the user's mobile and if entered incorrectly then an error message should pop up.

Actual output:

If entered pin code is valid then,



If it is invalid then,



Status: Passed

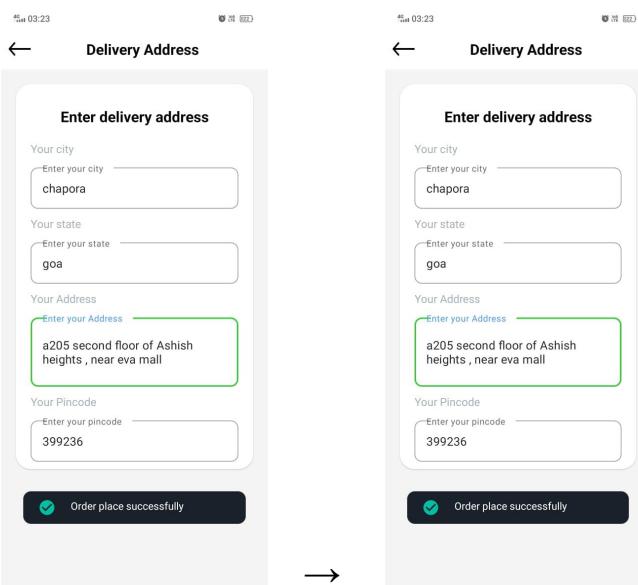
(6) Testing confirm button:

Description: It tests whether when clicking on the confirm button a valid user should be able to and an invalid user should not be able to proceed.

Expected Output: When the user clicks on the confirm button then it should show the confirmation message for delivery if and only if all the entries by the user are valid and if any of them are invalid then it should show an error message corresponding to that field.

Actual Output:

When all entries are valid and then confirm button is clicked,



When any of the entry is invalid and then confirm button is clicked,

The screenshot shows a mobile application interface titled "Delivery Address". At the top, there is a back arrow and the title "Delivery Address". Below the title, there is a section titled "Enter delivery address". This section contains four input fields: "Your city" with the value "chapora", "Your state" with the value "goa", "Your Address" (empty), and "Your Pincode" with the value "399236". A note below the pincode field states "Address should be atleast 20 characters". At the bottom of the screen is a large orange button labeled "Confirm Order".

Status: Passed

Module 5: Download reports and prescriptions

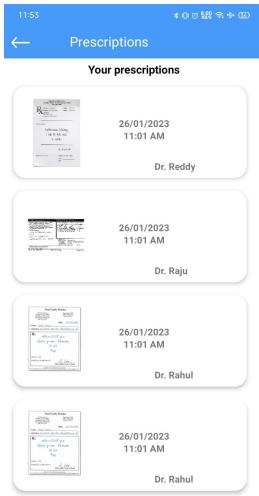
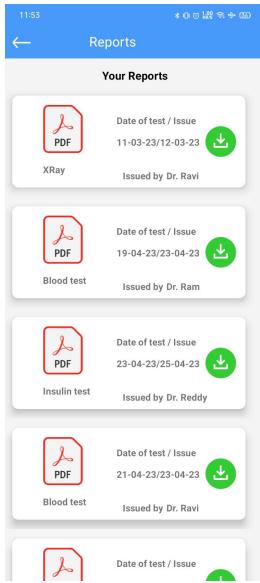
(1) Testing the visibility of reports and prescriptions screen:

Description: It tests whether all the reports and prescriptions are visible and can the user interact with them.

Expected Output: It should show all the past prescriptions and reports and when the user clicks on them then the file should open up.

Actual Output:

Visibility of reports and prescriptions screen,

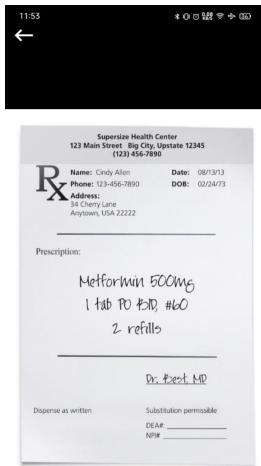
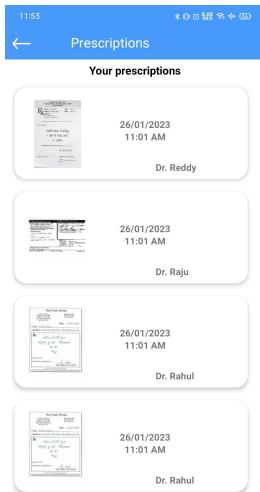


and

Reports

Prescriptions

When the user clicks on any file,



Status: Passed

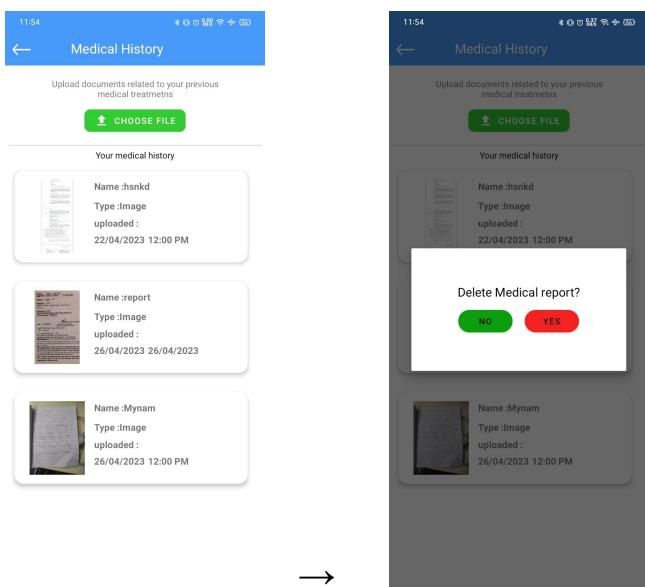
(2) Testing the delete option:

Description: It tests whether the user can delete any of the file as per his/her preferences.

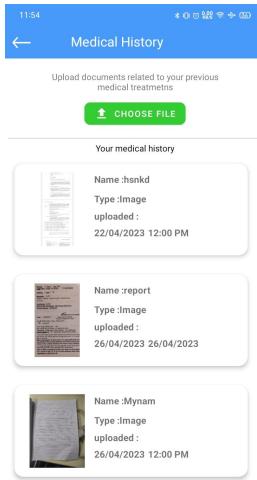
Expected Output: When the user long clicks on any of the report or prescription then it should open a pop up asking a message whether to delete that file or not and as per the option selected by the user it updates the reports. If user selects NO then no changes happen and if user selects YES then that particular file gets deleted.

Actual Output:

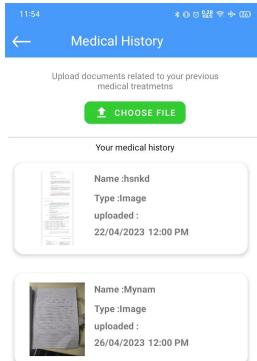
When user long clicks on any of the file,



When user selects NO in the delete option,



When user selects YES in the delete option,



Status: Passed

Module 6: Online Consultation

(1) Testing the Join button:

Description: It tests whether when clicking on the join button can join the call or not.

Expected Output: When the user clicks on the join button then if the doctor has joined the call, then it will show the UI of a video call where the user can see the doctor's video and if the doctor has not joined yet then it will show the message that appointment has not started.

Actual Output:

When doctor has started the appointment,



Status: Passed

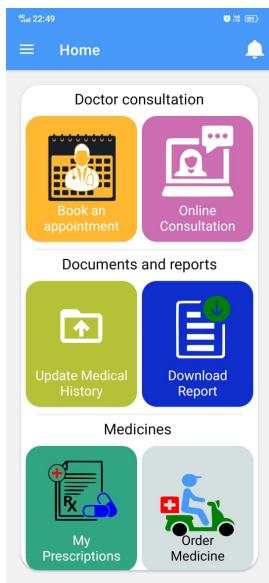
(2) Testing the Leave button:

Description: It tests whether when clicking on the leave button is the user able to leave or end the call.

Expected Output: The leave button should be disabled when the user has not yet joined the meeting and if already joined then after clicking on the leave button it should leave the call and should be redirected to the home page.

Actual Output:

When the user clicks the leave button,



Status: Passed

Module 7: Settings

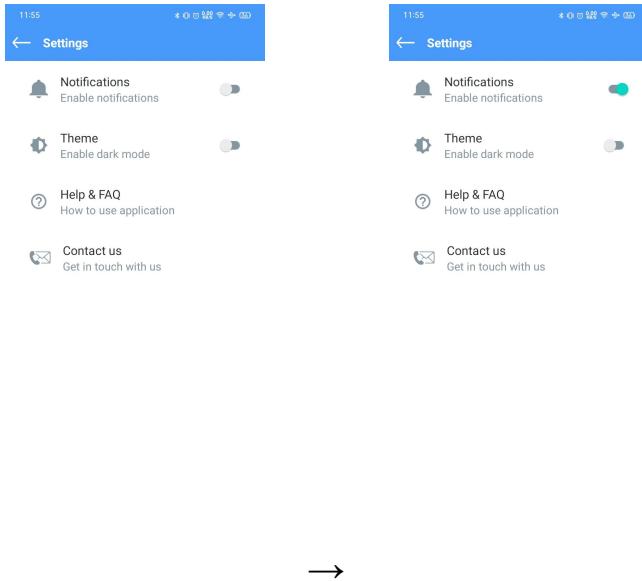
(1) Testing the switch:

Description: It tests whether a user can interact with the switch properly or not.

Expected Output: When the user clicks on the switch it changes its state, if it's off then it becomes on and vice versa.

Actual Output:

When the switch is off and is clicked,



Status: Passed

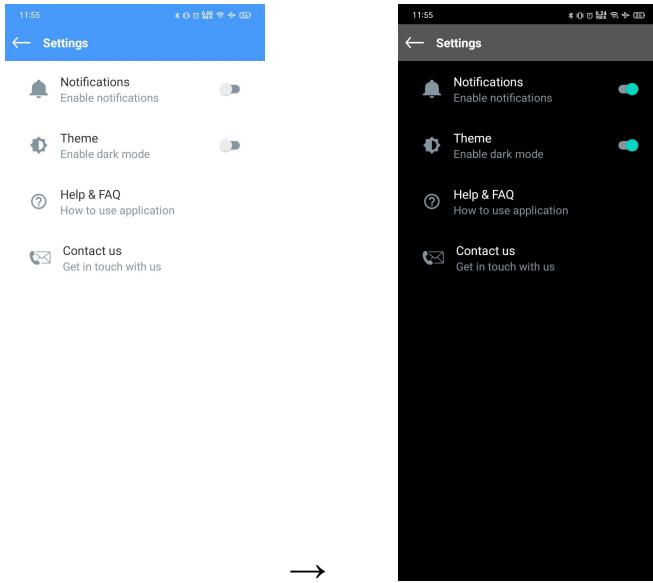
(2) Testing the theme switch:

Description: It tests whether when clicking on the switch, the UI changes as per the state of the switch

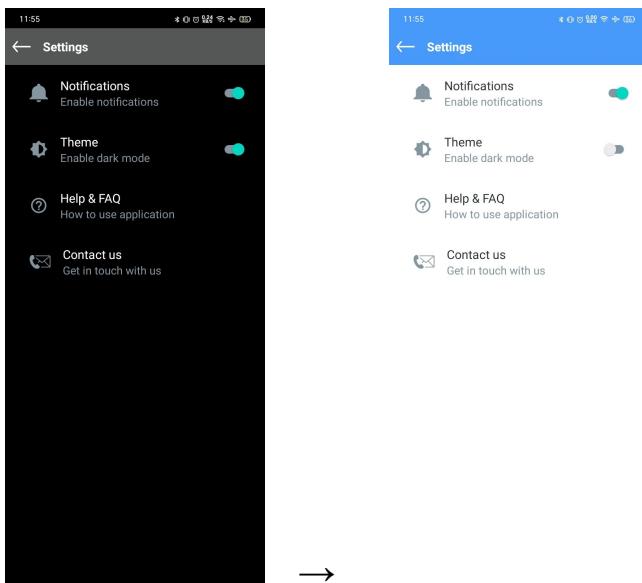
Expected Output: When the user clicks on the theme and is in ON state then the theme should change to light theme and when it is off and then it is clicked then the theme should change to black theme.

Actual Output:

When it is light theme and then switch is clicked,



When it is dark theme and then switch is clicked,



Status: Passed

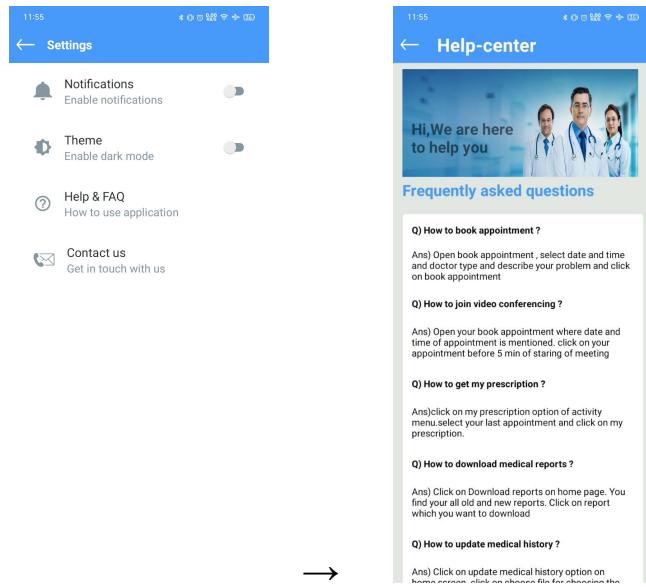
(3) Testing the 'Help & FAQ' and 'Contact us' options:

Description: It tests whether both of these options are working properly or not and shows output accordingly.

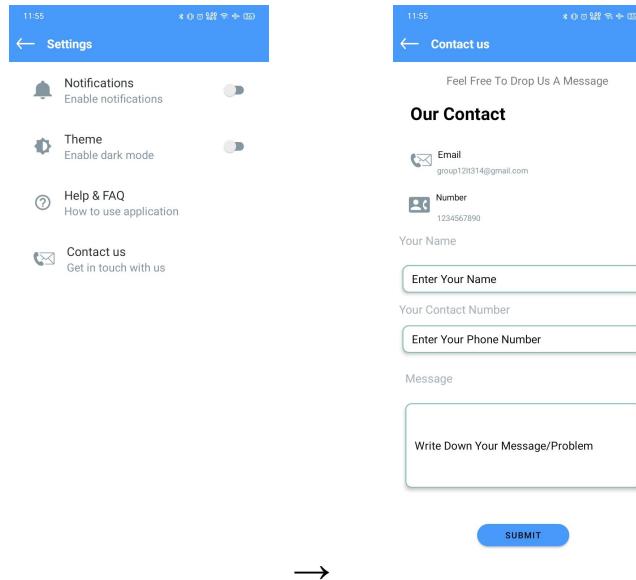
Expected Output: When the user clicks on any of these options then a new screen should appear with their respective elements in it.

Actual Output:

When the ‘Help & FAQ’ is clicked,



When the ‘Contact us’ is clicked,



Status: Passed