***TEST SPECIFICATION***

Sai Shanthan Baltha - 2793395

Sai Sree Cheekatimarla - 2793560

Sathvik Reddy Kandhi - 2825505

Prabhath Chandra Ausipuram - 2825686

**1.0 Introduction**

This section provides an overview of the entire test document. This document describes both the test plan and the test procedure.

**1.1 Goals and objectives**

The goal of this document is to clear all the provided test cases in section 3.

**1.2 Statement of scope**

N/A

**2.0 Test Plan**

This section describes the overall testing strategy and the project management issues that are required to properly execute effective tests.

**2.1 Software to be tested**

Android application should be tested in android studio using virtual machine.

**2.3 Testing tools and environment**

Android Studio emulator/Mobile device

**2.4 Test schedule**

For this application, we follow waterfall model. So, the testing will be done at the end after the application is developed. However, unit testing will be done by individual developer during development.

**3.0 Test Cases**

This section enumerates a complete list of test cases for the software. A template for test cases is as follows.

|  |  |
| --- | --- |
| ID | **1** |
| Test Input | **User name should not exceed 60 characters.** |
| Expected Output | **An error will be thrown saying “User Name is not valid”** |
| Description | **When the user signs up from the signup page, they need to fill the username field. If the user enters more than 60 characters in the field, it should throw an error and should not let user to go ahead.** |

|  |  |
| --- | --- |
| ID | **2** |
| Test Input | **User name should not contain special characters or number.** |
| Expected Output | **An error will be thrown saying “User Name is not valid”** |
| Description | **When the user signs up from the signup page, they need to fill the username field. If the user enters special characters or numbers in the field, it should throw an error and should not let user to go ahead.** |

|  |  |
| --- | --- |
| ID | **3** |
| Test Input | **User name should not be blank.** |
| Expected Output | **An error will be thrown saying “User Name is required”** |
| Description | **When the user signs up from the signup page, they need to fill the username field. If the user leaves the field blank, it should throw an error and should not let user to go ahead.** |

|  |  |
| --- | --- |
| ID | **4** |
| Test Input | **First name should not be blank.** |
| Expected Output | **An error will be thrown saying “First Name is required”** |
| Description | **When the user signs up from the signup page, they need to fill the first name field. If the user leaves the field blank, it should throw an error and should not let user to go ahead.** |

|  |  |
| --- | --- |
| ID | **5** |
| Test Input | **First name should not contain special characters or numbers.** |
| Expected Output | **An error will be thrown saying “First Name is not valid”** |
| Description | **When the user signs up from the signup page, they need to fill the first name field. If the user enters special characters or numbers in the field, it should throw an error and should not let user to go ahead.** |

|  |  |
| --- | --- |
| ID | **6** |
| Test Input | **First name should not exceed 60 characters.** |
| Expected Output | **An error will be thrown saying “First Name is not valid”** |
| Description | **When the user signs up from the signup page, they need to fill the first name field. If the user enters more than 60 characters in the field, it should throw an error and should not let user to go ahead.** |

|  |  |
| --- | --- |
| ID | **7** |
| Test Input | **Last name should not exceed 60 characters.** |
| Expected Output | **An error will be thrown saying “Last Name is not valid”** |
| Description | **When the user signs up from the signup page, they need to fill the last name field. If the user enters more than 60 characters in the field, it should throw an error and should not let user to go ahead.** |

|  |  |
| --- | --- |
| ID | **8** |
| Test Input | **Last name should not contain special characters or numbers.** |
| Expected Output | **An error will be thrown saying “Last Name is not valid”** |
| Description | **When the user signs up from the signup page, they need to fill the last name field. If the user enters special characters or numbers in the field, it should throw an error and should not let user to go ahead.** |

|  |  |
| --- | --- |
| ID | **9** |
| Test Input | **Last name cannot be blank.** |
| Expected Output | **An error will be thrown saying “Last Name is required”** |
| Description | **When the user signs up from the signup page, they need to fill the last name field. If the user leaves the field blank, it should throw an error and should not let user to go ahead.** |

|  |  |
| --- | --- |
| ID | **10** |
| Test Input | **Email Address should follow email conventions (@,.).** |
| Expected Output | **An error will be thrown saying “Email Id not valid”** |
| Description | **When the user signs up from the signup page, they need to fill the email address field. If the user does not enter the email address in correct format, it should throw an error and should not let user to go ahead.** |

|  |  |
| --- | --- |
| ID | **11** |
| Test Input | **Re-entered Email Address should follow email conventions (@,.).** |
| Expected Output | **An error will be thrown saying “Both the email address should be same”** |
| Description | **When the user signs up from the signup page, they need to fill the re-entered email address field. If the user does not enter the re-entered email address in correct format, it should throw an error and should not let user to go ahead.** |

|  |  |
| --- | --- |
| ID | **12** |
| Test Input | **Both the email address fields should match.** |
| Expected Output | **An error will be thrown saying “Both the email address should be same”** |
| Description | **When the user signs up from the signup page, they need to fill the email address field and re-enter email address field. If both the emails does not match, it should throw an error and should not let user to go ahead.** |

|  |  |
| --- | --- |
| ID | **13** |
| Test Input | **Email Address cannot be blank.** |
| Expected Output | **An error will be thrown saying “Email Address cannot be blank”** |
| Description | **When the user signs up from the signup page, they need to fill the email address field. If they leave the field blank, it should throw an error and should not let user to go ahead.** |

|  |  |
| --- | --- |
| ID | **14** |
| Test Input | **Re-enter Email Address cannot be blank.** |
| Expected Output | **An error will be thrown saying “Both the email address should be same”** |
| Description | **When the user signs up from the signup page, they need to fill the re-enter email address field. If they leave the field blank, it should throw an error and should not let user to go ahead.** |

|  |  |
| --- | --- |
| ID | **15** |
| Test Input | **Password field cannot be blank.** |
| Expected Output | **An error will be thrown saying “Password field is not valid”** |
| Description | **When the user signs up from the signup page, they need to fill the password field. If they leave the field blank, it should throw an error and should not let user to go ahead.** |

|  |  |
| --- | --- |
| ID | **16** |
| Test Input | **Re-enter password field cannot be blank.** |
| Expected Output | **An error will be thrown saying “Passwords do not match”** |
| Description | **When the user signs up from the signup page, they need to fill the re-enter password field. If they leave the field blank, it should throw an error and should not let user to go ahead.** |

|  |  |
| --- | --- |
| ID | **17** |
| Test Input | **Password need to match given constraints. (It should have 8-20 letters, at least 1 capital letter, at least one lower case letter, at least 1 number and at least 1 special character)** |
| Expected Output | **An error will be thrown saying “Password is invalid”** |
| Description | **When the user signs up from the signup page, they need to fill the password field. If the given value does not match password constraints, it should throw an error and should not let user to go ahead.** |

|  |  |
| --- | --- |
| ID | **18** |
| Test Input | **Re-enter password need to match given constraints. (It should have 8-20 letters, at least 1 capital letter, at least one lower case letter, at least 1 number and at least 1 special character)** |
| Expected Output | **An error will be thrown saying “passwords do not match”** |
| Description | **When the user signs up from the signup page, they need to fill the re-enter password field. If the given value does not match password constraints, it should throw an error and should not let user to go ahead.** |

|  |  |
| --- | --- |
| ID | **19** |
| Test Input | **Password and Re-enter Password field should match** |
| Expected Output | **An error will be thrown saying “Passwords do not match”** |
| Description | **When the user signs up from the signup page, they need to fill the password and re-enter password fields. If they do not match, it should throw an error and should not let user to go ahead.** |

|  |  |
| --- | --- |
| ID | **20** |
| Test Input | **User type field should be selected.** |
| Expected Output | **An error will be thrown saying “User type not selected”** |
| Description | **When the user signs up from the signup page, they need to select the user type field from the given 3 options. If they do not select anything, it should throw an error and should not let user to go ahead.** |

|  |  |
| --- | --- |
| ID | **21** |
| Test Input | **All the fields are entered correctly.** |
| Expected Output | **A message will be populated saying “Everything looks good” and user will be directed to the home page based on the user type.** |
| Description | **When the user signs up from the signup page, they need to enter all the fields. If they enter everything correctly, it should go ahead to the home page.** |

|  |  |
| --- | --- |
| ID | **22** |
| Test Input | **User name should be new.** |
| Expected Output | **An error will be thrown saying “User name already exists”** |
| Description | **When the user signs up from the signup page, they need to fill the username field. If the username is already taken by another user, it should throw an error and should not let user to go ahead.** |

|  |  |
| --- | --- |
| ID | **23** |
| Test Input | **User should already be signed in.** |
| Expected Output | **An error will be thrown saying “User name not found”** |
| Description | **When the user logs in from the login page, they need to fill the username field. If the username is not present in the data base, it should throw and error and should not go forward.** |

|  |  |
| --- | --- |
| ID | **24** |
| Test Input | **Username and password should match database values.** |
| Expected Output | **An error will be thrown saying “Invalid details entered”** |
| Description | **When the user logs in from the login page, they need to fill the username and password field. If the username is present in the data base and entered password does not match the database value, it should throw and error and should not go forward.** |

|  |  |
| --- | --- |
| ID | **25** |
| Test Input | **Username and password are correct.** |
| Expected Output | **An error will be thrown saying “Login successful” and logins into the home page based on the type of user.** |
| Description | **When the user logs in from the login page, they need to fill the username and password field. If the username and password fields match as present in the database, it should validate and take the user to the home page based on the user type.** |

|  |  |
| --- | --- |
| ID | **26** |
| Test Input | **User clicks on forgot password page from login.** |
| Expected Output | **Forgot password page should be displayed and the user should enter username and email address in the respective fields and an email will be sent to respective user with password information.** |
| Description | **When the user forgets password, the user can get the details by entering the proper details in forgot password page.** |

|  |  |
| --- | --- |
| ID | **27** |
| Test Input | **Entered username in forgot password page does not exist in database.** |
| Expected Output | **An error should be thrown saying “entered username does not exist”** |
| Description | **When the user forgets password, the user can get the details by entering the proper details in forgot password page.** |

|  |  |
| --- | --- |
| ID | **28** |
| Test Input | **Entered email address does not match the database record with the username.** |
| Expected Output | **An error should be thrown saying “entered email address is wrong”** |
| Description | **When the user forgets password, the user can get the details by entering the proper details in forgot password page.** |

|  |  |
| --- | --- |
| ID | **29** |
| Test Input | **For Student – When a user logs in as student, the home page should contain 2 fragments – one for scheduling and other for profile.** |
| Expected Output | **Home page should contain a drop down menu which should have the list of professors. User should be able to select one professor out of the given list** |
| Description | **When the user logs in from login page with correct credentials as student, it should take the user to home page and the user should be able to see the list of professors.** |

|  |  |
| --- | --- |
| ID | **30** |
| Test Input | **For Student – When a user logs in as student, the home page should contain 2 fragments – one for scheduling and other for profile.** |
| Expected Output | **Home page should contain a drop down menu which should have the list of professors. User should be able to select one professor out of the given list. When the user selects a professor and clicks ok, the timings of that particular professor should be shown.** |
| Description | **When the user logs in from login page with correct credentials as student, it should take the user to home page and the user should be able to see the list of professors and the selected professor’s available timings.** |

|  |  |
| --- | --- |
| ID | **31** |
| Test Input | **For Student – When a user logs in as student, the home page should contain 2 fragments – one for scheduling and other for profile.** |
| Expected Output | **Home page should contain a drop-down menu which should have the list of professors. User should be able to select one professor out of the given list. When the user selects a professor and clicks ok, the timings of that professor should be shown. When the student clicks on a time slot, a pop up should be shown for confirmation.** |
| Description | **When the user logs in from login page with correct credentials as student, it should take the user to home page and the user should be able to see the list of professors and the selected professor’s available timings.** |

|  |  |
| --- | --- |
| ID | **32** |
| Test Input | **For Student – When a user logs in as student, the home page should contain 2 fragments – one for scheduling and other for profile.** |
| Expected Output | **Home page should contain a drop-down menu which should have the list of professors. User should be able to select one professor out of the given list. When the user selects a professor and clicks ok, the timings of that professor should be shown. When the student clicks on a time slot, a pop up should be shown for confirmation. If the user clicks yes, an email should be sent to student and professor with the appointment details.** |
| Description | **When the user logs in from login page with correct credentials as student, it should take the user to home page and the user should be able to see the list of professors and the selected professor’s available timings.** |

|  |  |
| --- | --- |
| ID | **33** |
| Test Input | **For Student – When a user logs in as student, the home page should contain 2 fragments – one for scheduling and other for profile.** |
| Expected Output | **Profile page should contain the details of the student like student name, email address and password in encrypted format.** |
| Description | **When the user logs in from login page with correct credentials as student, it should take the user to home page and the user should be able to see the list of professors and the selected professor’s available timings and a profile button** |

|  |  |
| --- | --- |
| ID | **34** |
| Test Input | **For Student – When the user schedules and appointment and the Gmail server is low** |
| Expected Output | **An error should be shown saying "server not connected”** |
| Description | **When the user logs in from login page with correct credentials as student, it should take the user to home page and the user should be able to see the list of professors and the selected professor’s available timings and a profile button.** |

|  |  |
| --- | --- |
| ID | **35** |
| Test Input | **For professor – When a user logs in as professor, the home page should contain 2 fragments – one for appointments and other for profile.** |
| Expected Output | **Home page should have a list of appointments with the time and a delete button beside that.** |
| Description | **When the user logs in from login page with correct credentials as professor, it should take the user to home page and the user should be able to see the list of appointments along with the time.** |

|  |  |
| --- | --- |
| ID | **36** |
| Test Input | **For professor – When a user logs in as professor, the home page should contain 2 fragments – one for appointments and other for profile.** |
| Expected Output | **Home page should have a list of appointments with the time and a delete button beside that.** |
| Description | **When the user logs in from login page with correct credentials as professor, it should take the user to home page and the user should be able to see the list of appointments along with the time.** |

|  |  |
| --- | --- |
| ID | **37** |
| Test Input | **For professor – When a user logs in as professor, the home page should contain 2 fragments – one for appointments and other for profile and wants to delete an appointment.** |
| Expected Output | **Home page should have a list of appointments with the time and a delete button beside that. User should be able to delete an appointment using the delete button** |
| Description | **When the user logs in from login page with correct credentials as professor, it should take the user to home page and the user should be able to see the list of appointments along with the time.** |

|  |  |
| --- | --- |
| ID | **38** |
| Test Input | **For professor – When a user logs in as professor, the home page should contain 2 fragments – one for appointments and other for profile and wants to delete an appointment but the Gmail server is down** |
| Expected Output | **An error should be thrown saying server is not connected.** |
| Description | **When the user logs in from login page with correct credentials as professor, it should take the user to home page and the user should be able to see the list of appointments along with the time.** |

|  |  |
| --- | --- |
| ID | **39** |
| Test Input | **For professor – When a user logs in as professor, the home page should contain 2 fragments – one for appointments and other for profile.** |
| Expected Output | **When the user clicks on the profile fragment, it should show the user details like name, email address and password in encrypted format.** |
| Description | **When the user logs in from login page with correct credentials as professor, it should take the user to home page and the user should be able to see the list of appointments along with the time.** |

|  |  |
| --- | --- |
| ID | **40** |
| Test Input | **Application server is dowm** |
| Expected Output | **Should not let the user to next steps from the main page** |
| Description | **N/A** |