Hodophilia Inc

Software Test Plan

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

## Akshay Venkatesh Murthy (akmurthy@iu.edu) Indu Bhanu Venkata Pothineni (vpothine@iu.edu) Stephen Michael Smith (sms10@iu.edu)

**Griffin Wall (griwall@iu.edu)**

# Overview

This section provides an overview of the testing approach used to verify the software product.

## Test Objectives

Ensure our code works as expected without any issues; verify that the user can interact with all provided functionality seamlessly. We will thus conduct performance testing, feature testing, acceptance testing, unit testing, API testing, and integration testing.

**Sprint 2:**

**Sprint 1:**

Feature Testing: Login/Sign up feature along with Google Sign in Component Testing: Login, Sign up. Database, Security components API Testing: Login, Signup

Integration Testing: Testing by integrating react components with REST API endpoints

White box testing: Database, API Testing

Black box testing: Testing login, sign up features with Google sign in.

Feature Testing: Search for location, display location details, sign up with 2FA, account recovery with security questions, account recovery with email Component Testing: Search bar component, search route, location display route, account recovery route, reset password route, QR code route

API Testing: Get city endpoint, get city details endpoint, integer code 2FA endpoint, forgot password (send recovery email) w/ email endpoint, forgot password w/ security questions endpoint, reset password endpoint Integration Testing: Ensure frontend communicates properly with backend through REST API endpoints

White box testing: Database, API Testing

Black box testing: Testing signup with/without 2FA, testing password recovery systems

## Test Environment

Our test environment is Google Chrome and Firefox as browser clients. We are using Heroku as our hosting service. A machine with at least 4 GB of ram with processor at least Pentium 4 is required for this testing. We are also using POSTMAN to do API testing.

## Test Personnel

|  |  |
| --- | --- |
| Akshay Venkatesh Murthy | API Testing, White box testing, database testing |
| Griffin Wall | Integration testing, Front end testing |
| Stephen Smith | Front end Testing, API Testing |
| Indu Bhanu Venkata Pothineni | API Testing, White box testing, database testing, Component Testing. |

* 1. **Acceptance Criteria**

Sprint 1:

* The user should be able to click a button to log in, either logging in using their email or google, and their username, email, password, first name, last name, and provider should be stored.
* The user should also be able to sign up for an account, using their first name, last name, email, username, and password.
* The user should not be able to enter an invalid email, and the user should not be able to log in before signing up.
* The user should be directed back to the home page after logging in.
* The user should be able to see appropriate error messages during conditions of wrong entry of credentials in input fields.

Sprint 2:

* A user should be able to successfully store signup details in backend including two security questions and 2 factor authentication opt-in.
* A user should be able to successfully login without being prompted for 2FA *unless* opted in on signup
* A user who has opted in to 2FA on signup should receive a QR code through a route that will allow them to receive integer codes on login request
* A user who has opted in to 2FA on login should receive a 6-digit code through their mobile device that will allow the user to login if and only if the correct code is provided
* A user who requests to change their password:
  + User can request email reset: Will be sent link that, on click, will lead to a route with a unique token to allow only that user to change their password
  + User can request security question reset: Will be prompted to answer two security questions that, if and only if *all* inputs match security questions on signup, will allow only that user to change their password
* Once a user is asked to change password:
  + Password and password repeat must be the same input
  + New password must successfully overwrite old password
  + New password must work when attempting to login

## Noted Omissions

Currently no omissions

# Test Cases

The test cases are the partitioning of the verification of the software into manageable sections. Often these sections correspond to the set of active use case scenarios but can be organized as the test developer sees fit. Test cases should be in place to cover all the software verification methods. Even non-execution-based testing methods (i.e., inspection/analysis) may be detailed here. The intent is for the test case procedures to provide a repeatable verification of the software S.

For each test case describe the following:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Name | Description | Initial conditions | Input Data | Specificati ons | Procedure |
| T-S01-01 | Test home page | To test the successful load of the home page on starting of the application | Successful running front end and backend component s | None | P565T31-5 | * Launch hodophilia app. * Test if the app launches successfully and has a login link on the home page |
| T-S01-02 | Test user signup and valid entries for user sign up | To Test if the user is able to sign up with valid entries | The user must be present in the sign up page , which is linked from the login page. | Valid email and password of minimum 6-max 40  characters, First Name Last Name | P565T31-7 | * Launch Hodophilia app * Test if the app successfully launches. * Click on login * Click on Sign up * Enter the input data mentioned. * Check if the user signed up successfully. |
| T-S01-03 T-S01-04 | Test user login | Test if the user is able to login through his signed up credentials | The user must be present on the login page | Valid signed up credentials such as, email and password | P565T31-7 | * Launch Hodophilia app * Test if the app successfully launches. * Click on login * Click on Sign up * Enter the input data mentioned. * Check if the user signed up successfully. * Route to sign in link at the bottom * Enter input data * Check if the user is |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | able to sign in successfully |
| T-S01-05 | Test unsuccessf ul user login | Test if the user is unable to login through wrong credentials | The user must be present on the login page | Invalid email, wrong password, Non-signe d up email and password | P565T31-7 P565T31-1 0 | * Launch hodophilia app * Click on login * Enter a invalid email and password combination * Check for unsuccessful login * Enter valid email but a wrong password and check for unsuccessful login * Enter valid email and valid password which is not signed up. Check for message that , user not found. |
| T-S01-06 | Test login through google sign in | Test if the user is able to login through google | The user must be present on the login page | Valid google account | P565T31-1 5 | * Launch hodophilia app * Click on login * Click on Continue with google * Enter valid google credentials * Check if the user is successfully logged in |
| T-S01-07 | Test if the password is encrypted and stored in the DB | Test if the user’s password is stored securely using encryption | The user must be present on the login page and login with valid credentials Postgres server set up | Valid signed up email and password |  | * Launch hodophilia app * Click on login * Enter valid credentials * Launch postgres server and enter into the DB to check for user entry and encrypted password |
| T-S02-01 | Test proper | Ensure | The user | User has |  | * Launch hodophilia |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | data submission on signup | that when the user goes to create a new account, the backend properly receives 2FA  checkbox, security question 1, and security question 2 inputs | must be present on the signup page, fill in all the forms, provide a valid email (<name>@  <domain>.  <TLD>),  and submit the form. | navigated to Signup page |  | app   * Click on Signup * Insert data into name field * Insert UNUSED email into email field * Insert password into password field * Insert string values for both security questions * Check multifactor authentication * Click submit button * Verify from database that submitted email has an entry with mfa = t, security\_question1 and security\_question2 have inputs as submitted |
| T-S02-02 | Test 2FA incorrect code on login | Test if user is UNABLE  to login on an account with 2FA enabled by verifying with incorrect mobile device code | User has created a new account with 2FA enabled | Correct email, password |  | * Sign up with 2FA enabled * Install and run Microsoft Authenticator mobile app to authenticate with QR code * Insert proper email, password * Await code * Insert code into verify form that does not match received code * Test if user is prevented from logging in and/or receives an error message in UI |
| T-S02-03 | Test 2FA correct code on | Test if user is able to login on an | User has created a new | Correct email, password |  | * Sign up with 2FA enabled * Install and run |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | login | account with 2FA enabled by verifying with mobile device code | account with 2FA enabled |  |  | Microsoft Authenticator mobile app to authenticate with QR code   * Insert proper email, password * Await code * Insert matching code into verify form * Test if user is able to log in and/or receives a success message in UI |
| T-S02-04 | Search and explore | Test if the user can explore a given search entry in a different page | No requireme nt for login | Hodophilia must be up |  | * Launch the app * Enter a valid text entry in a search field and view the available places options. * On clicking each of them, the user must see the page with ‘Explore   <place\_name>’ and all the buttons to click on for different functions on the page. |

You may represent this information as a table, or simply list it.

**NOTE:** If you are using the rigorous process, the Specifications described above are already captured in the RVM and may be omitted from this document

# Revision History

|  |  |  |
| --- | --- | --- |
| **Revision** | **Date** | **Change Description** |
| v1 | 10/04/2022 | Sprint 1 test plan update with Login/Sign up feature |

|  |  |  |
| --- | --- | --- |
| V2 | 10/18/2022 | Sprint 2 test plan update with search & account recovery features |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |