**Ctrl-Alt-Elite**

**Mindlift**

**TEST PLAN**

Date: 10/18/2023

Table of Contents

[Introduction 3](#_Toc109387551)

[1.1 Objectives 3](#_Toc109387552)

[1.2 Team Members 3](#_Toc109387553)

[2 Scope 3](#_Toc109387554)

[3 Assumptions / Risks 4](#_Toc109387555)

[3.1 Assumptions 4](#_Toc109387556)

[3.2 Risks 4](#_Toc109387557)

[4 Test Approach 4](#_Toc109387558)

[4.1 Test Automation 4](#_Toc109387559)

[5 Test Environment 5](#_Toc109387560)

[6 Milestones / Deliverables 5](#_Toc109387561)

[6.1 Test Schedule 5](#_Toc109387562)

[6.2 Deliverables 5](#_Toc109387563)

# Introduction

The Test Plan has been created to communicate the test approach to team members. It includes the objectives, scope, schedule, risks and approach. This document will clearly identify what the test deliverables will be and what is deemed in and out of scope.

## Objectives

The purpose of the application is to allow a user struggling with depression, or any form of mental disorder, to communicate with an AI. The application, on top of the chat feature, also stores conversations for later viewing, seeks correlations between how a user is feeling and the events throughout their day, allows the user to store their mood throughout the day, and features a crisis detection and prevention ability.

## Team Members

|  |  |
| --- | --- |
| **Resource Name** | **Role *(examples are given below)*** |
| Austin Hoffman | Developer |
| Akash Patel | Developer |
| Hadia Bilal | Developer |
| Shawana Tahseen | Developer |
| Timmy Akindunni | Developer |

# Scope

The initial sprint will include ‘must have’ requirements. These and any other requirements that get included must all be tested.

The following sections indicate what is tested during each sprint. The scope of testing is determined at the beginning of the current sprint.

At the end of Sprint 1, a user must be able to:

1. Navigate through the app to each screen. R10
2. See what kind of content each screen will contain. R10

At the end of Sprint 1, the application must:

1. Be usable on both IOS and Android.

At the end of Sprint 2, a user must be able to:

1. Decide at the start of sprint 2

At the end of Sprint 3, a user must be able to:

1. Decide at the start of sprint 3

**Assumptions / Risks**

## Assumptions

1. The user has a mobile device that is listed as compatible and has used it long enough to be proficient in navigating apps and the Appstore.
2. The user speaks, reads, and writes English.
3. The user has some level of awareness regarding the performance of local AI.
4. The user has hands, or some way of interacting with onscreen UI.

## Risks

The following risks have been identified and the appropriate action identified to mitigate their impact on the project. The impact (or severity) of the risk is based on how the project would be affected if the risk was triggered. The trigger is what milestone or event would cause the risk to become an issue to be dealt with.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Risk | Impact | Trigger | Mitigation Plan |
| 1 | The local AI does not respond meaningfully | Medium | Feelings of disconnect between user and application. | AI’s response creativity will be increased, and more training data will be given. |
| 2 | The local AI responds with harmful responses | High | A response that provides harmful advice, abuse, or triggering statements. | AI creativity will be reduced until training data has been confirmed safe, and model response quality is above the acceptable threshold. |
| 3 | AI performance is too low | Medium | Response time is greater than five seconds. | AI parameter amount will be reduced. Training set will be trimmed. |
| 4 | Application drains too much battery while not actively in use | Medium | Battery drain on device is significant, draining 2-3 times more battery than other sleeping applications. | Optimize the crisis detection feature. Lessen the number of sounds to look for to initiate crisis prevention. |
| 5 | Training set required is too large | Medium | The current training set does not produce responses in the AI that meet the quality threshold. | Gather another five thousand examples. |
| 6 | AI model becomes too large to use with GitHub | Low | Github does not allow upload of model. | Store the model in Git LFS. |
| 7 | Training a model from scratch becomes too demanding on time | High | Estimates of model training exceed allotted project time. | Curate and specialize a pretrained model. |

# Test Approach

The project is using an agile approach, with 3-week sprints. *Mention how you will conduct testing during the sprint in terms of the techniques you plan to do and when. Add a new subsection for each sprint.*

Sprint 1

The primary focus of Sprint 1 is on creating the foundation for MindLift, specifically the screen displays and the primary interactivity. Our testing strategy for this sprint is as follows:

* Testing Methodologies: The focus of the first sprint will be functional testing. We will make sure that the screens and interactions with them follow the screen layout in the design document and they operate as intended.
* We will also carry out User Interface (UI) testing to confirm that the screens are user-friendly, and they look good. Screens and interactions will be evaluated along with the creation of each screen. This way we can recognize and address any issues that may arise.
* Regression testing will be done as new features and screens are added throughout Sprint 1 to make sure that existing functionality is still good.
* Test cases will be developed to confirm that each screen and interaction is functional. Different user scenarios will be simulated using appropriate test data.

This test plans in line with Sprint 1's goals, i.e., laying basic framework for the application. This will make sure functional testing is prioritized, app is appealing to look at and is interactive.

Sprint 2

Sprint 3

## Test Automation

*Role of automated testing:*

*-Improves efficiency, helps in early detection of issues, can be performed as many times as we want, helps with regression testing by making sure that new changes are not affecting existing functionality.*

*Plan to conduct any (including tools):*

*-If we plan to conduct automated testing in the future, we will use “Flutter Driver” tool.*

## Test Cases (Black Box)

### Feature 1: Login Screen

*Have a table for the test cases needed to test the User Story*

| **Test Case ID** | **Description** | **Requirements Trace** | **Directions** | **Expected Output** |
| --- | --- | --- | --- | --- |
| TC-001 | Valid credentials for login | List the user story/stories this is linked to | 1. Launch app “MindLift” 2. Put in a valid email and password. 3. Click button that says “Login”. | User should login and see homepage. |
| TC-002 | Valid credentials while logging in |  | 1. Launch app “MindLift” 2. Put in an invalid email and password.   Click button that says “Login”. | User should be denied access and screen should give error. |
| TC-003 | Login without entering data |  | 1. Launch app “MindLift” 2. Put in nothing.   Click button that says “Login”. | User should be denied access and screen should give error. |

### Feature 2: Signup Screen

*Have a table for the test cases needed to test the User Story*

| **Test Case ID** | **Description** | **Requirements Trace** | **Directions** | **Expected Output** |
| --- | --- | --- | --- | --- |
| TC-101 | Valid email, password and unique username entered to signup | List the user story/stories this is linked to | 1. Launch app “MindLift” 2. Click on “Signup” 3. Put in a valid email and password and unique username. 4. Click button that says “Signup”. | User should be able to sign up and can login be using credentials. |
| TC-102 | Invalid email, invalid password or a username that is not unique is entered to signup |  | 1. Launch app “MindLift” 2. Click on “Signup” 3. Put in an invalid email or password that does not meet requirements or a username that has been taken.   Click button that says “Signup”. | Users should not be able to sign up and will receive errors. |
| TC-103 | Either one or all boxes are left empty |  | 1. Launch app “MindLift” 2. Click on “Signup” 3. Put in nothing.   Click button that says “Signup”. | User should not be able to sign up and will receive error. |

### Feature 3: Home Screen

*Have a table for the test cases needed to test the User Story*

| **Test Case ID** | **Description** | **Requirements Trace** | **Directions** | **Expected Output** |
| --- | --- | --- | --- | --- |
| TC-201 | Verify user lands on home screen once he/she logins | List the user story/stories this is linked to | 1. Login with valid email and password 2. Notice that you are landing on homepage | User should land on home screen after logging in. |
| TC-202 | Verify home screen has all the contents that need to be displayed. |  | 1. Access Home screen 2. Verify you can see all contents including mood tracking, goals, check in, etc. | User should be able to see all required content on homescreen. |
| TC-203 | Verify Mood tracking functionality |  | 1. Access home Screen 2. Click “Mood Tracking” 3. Observe going to Mood tracking screen | User should be able to access mood tracking screen from homescreen. |
| TC-204 | Verify goals tracking functionality |  | 1. Access home Screen 2. Click “Goals” 3. Observe going to goals screen | User should be able to access goals screen from homescreen. |
| TC-205 | Verify Check In functionality |  | 1. Access home Screen 2. Click “Check In” 3. Observe going to check in screen | User should be able to access check in screen from homescreen. |

### Feature 4: Menu Screen

*Have a table for the test cases needed to test the User Story*

| **Test Case ID** | **Description** | **Requirements Trace** | **Directions** | **Expected Output** |
| --- | --- | --- | --- | --- |
| TC-301 | Accessing menu | List the user story/stories this is linked to | 1. Go to menu icon on top right corner of home screen. 2. Click on three lines representing menu. 3. Observe menu | User should see all the list of screens on the menu. |
| TC-302 | Accessing Chat screen from menu |  | 1. Click on the menu icon (three parallel lines) on top right corner of home screen. 2. Click on Chat from list 3. Observe going to chat screen | User should land on chat screen. |
| TC-303 | Accessing Goals screen from menu |  | 1. Click on the menu icon (three parallel lines) on top right corner of home screen. 2. Click on Goals from list 3. Observe going to goals screen | User should land on goals screen. |
| TC-304 | Accessing Profile screen from menu |  | 1. Click on the menu icon (three parallel lines) on top right corner of home screen. 2. Click on Profile from list 3. Observe going to Profile screen | User should land on profile screen. |
| TC-305 | Accessing Conversation History screen from menu |  | 1. Click on the menu icon (three parallel lines) on top right corner of home screen. 2. Click on Conversation History from list 3. Observe going to Conversation History screen | User should land on conversation history screen. |
| TC-306 | Accessing Emotion History screen from menu |  | 1. Click on the menu icon (three parallel lines) on top right corner of home screen. 2. Click on emotion history from list 3. Observe going to emotion history screen | User should land on emotion history screen. |

| **Test Case ID** | **Description** | **Requirements Trace** | **Directions** | **Expected Output** |
| --- | --- | --- | --- | --- |
| TC-307 | Accessing Password Reset screen from menu | List the user story/stories this is linked to | 1. Click on the menu icon (three parallel lines) on top right corner of home screen. 2. Click on Password reset from list 3. Observe going to Password reset screen | User should land on password reset screen. |
| TC-308 | Accessing Settings screen from menu in |  | 1. Click on the menu icon (three parallel lines) on the top right corner of home screen. 2. Click on settings from list 3. Observe going to settings screen | User should land on settings screen. |

### Feature 4: Chat Screen

*Have a table for the test cases needed to test the User Story*

| **Test Case ID** | **Description** | **Requirements Trace** | **Directions** | **Expected Output** |
| --- | --- | --- | --- | --- |
| TC-401 | New chat from chat screen | List the user story/stories this is linked to | 1. Click chat screen on menu 2. Click new chat 3. Chat with bot | User should be able to open new chat and can talk with chat bot. User gets response with 80 % accuracy. |

### Feature 5: Goals Screen

*Have a table for the test cases needed to test the User Story*

| **Test Case ID** | **Description** | **Requirements Trace** | **Directions** | **Expected Output** |
| --- | --- | --- | --- | --- |
| TC-501 | Can create new goals | List the user story/stories this is linked to | 1. Click goals screen on menu 2. Click create new goal 3. Write goal | Goal can be written and gets saved |
| TC-502 | Can access previous goals |  | 1. Click chat screen on menu 2. You can see past goals | User should be able to see past goals. |
|  |  |  |  |  |

### Feature 6: Profile Screen

*Have a table for the test cases needed to test the User Story*

| **Test Case ID** | **Description** | **Requirements Trace** | **Directions** | **Expected Output** |
| --- | --- | --- | --- | --- |
| TC-601 | See/Edit profile photo | List the user story/stories this is linked to | 1. Click profile screen on menu 2. You can see profile picture 3. You can edit profile picture by clicking edit button | User should be able to see profile picture and edit it. |
| TC-602 | See/Edit profile info |  | 1. Click profile screen on menu 2. You can see profile info. 3. You can edit profile info by clicking edit button | Users should be able to see profile info including name etc. and they should be able to edit it. |
| TC-603 |  |  |  |  |

### Feature 7: Conversation History Screen

*Have a table for the test cases needed to test the User Story*

| **Test Case ID** | **Description** | **Requirements Trace** | **Directions** | **Expected Output** |
| --- | --- | --- | --- | --- |
| TC-701 | User can see and access past conversations. | List the user story/stories this is linked to | 1. Click conversation history screen on menu 2. You can see conversation history. 3. You can select conversation based on date and be able to view it. | User should be able to view and access past conversations by date. |
|  |  |  |  |  |
|  |  |  |  |  |

### Feature 8: Emotion History Screen

*Have a table for the test cases needed to test the User Story*

| **Test Case ID** | **Description** | **Requirements Trace** | **Directions** | **Expected Output** |
| --- | --- | --- | --- | --- |
| TC-801 | User can see past emotions. | List the user story/stories this is linked to | 1. Click emotion history screen on menu 2. You can see emotion history by date. | User should be able to view emotion history by date and should see emojis representing emotions. |

### Feature 9: Password Reset Screen

*Have a table for the test cases needed to test the User Story*

| **Test Case ID** | **Description** | **Requirements Trace** | **Directions** | **Expected Output** |
| --- | --- | --- | --- | --- |
| TC-901 | User can reset password. | List the user story/stories this is linked to | 1. Click emotion history screen on menu 2. You can reset the password. | User should be able to change password successfully. |

### Feature 10: Settings Screen

*Have a table for the test cases needed to test the User Story. A sample is below*

| **Test Case ID** | **Description** | **Requirements Trace** | **Directions** | **Expected Output** |
| --- | --- | --- | --- | --- |
| TC-111 | User can change themes | List the user story/stories this is linked to | 1. Click settings screen on menu 2. Click change them to dark or light | App theme changes to dark or light. |
| TC-121 | User can see past emotions. | List the user story/stories this is linked to | 1. Click settings screen on menu 2. Click | User should be able to view emotion history by date and should see emojis representing emotions. |

## Test Cases (White Box)

3.3.1..n Name by Feature (like above)

*Organize the test cases using the table below. Be sure to have a directory in your repository for your test suite that follows a naming scheme that matches the items in the table below.*

*Like section 3.2, you will add to the table as you progress through each sprint.*

| **Test Case ID** | **Description** | **Directions/Goals** | **Expected Output** |
| --- | --- | --- | --- |
| TC-1001 | Selecting a restaurant to visit. | Follow TCB-001 when there are no restaurants available. Does the system handle not having a restaurant to select? | Should load the home page without errors. However, there are no database checks for null pointers so if the DB fails, there will be an error. |
| TC-1002 | Attempt to go to a restaurant page without selecting a restaurant. | Follow TCB-001 only do not select a restaurant. Does the system detect that no restaurant is selected. | The system should force the user to select a restaurant. Should fail however since there is no null pointer check in the code. |
| TC-1003 | Check site links. | Browse all available pages clicking on each link available to make sure they point to pages. Also, check graphics as well. | Each page should link properly to intended pages. |
| TC-1004 | Ensure data validation is working properly. | Follow TCB-006 only with improper input by leaving fields blank. Also attempt to use improperly formatted email and phone numbers, i.e. The last name for both fields. | System should check to make sure that fields are filled in and warned about otherwise. If there is an improper format for the email and phone number, that should be marked as well. |
| TC-1005 | Attempt to use the system when the database is down. | Access any page without the database running. Check to make sure the error seems reasonable. | Although not a requirement, if the database connection cannot be made, a reasonable error message would be nice. |
| TC-1006 | Attempt to use the system when JavaScript is disabled in the user’s browser. | Can you still make a reservation with blanks in the field if you disable JavaScript? See how verbose the error handling is. | Since JavaScript does the error handling, not the code, disabling JavaScript should allow bad input. |
| TC-1007 | Attempt to use apostrophes in text fields whose values get used in SQL statements. | Follow TCB-006 but use O’Conner as the last name.  Checking to make sure data checking includes escaping characters. | SQL uses the apostrophe character as a special character, so does the system escape the apostrophe character if it appears in a name to prevent a bad SQL call. |
| TC-1008 | Bounds checking on the Erlang implementation. | Use the Erlang page and try using a blank field or negative numbers. | Should fail since there is no error handling to round or check the input to the Erlang implementation. |
| TC-1009 | Add an incentive directly to the database and ensure it is displayed. | 1. Open a web browser and go to <http://rrs.se.rit.edu/rrs/sql.jsp>. 2. Enter the SQL statement “INSERT INTO incentives (restaurant\_id, start\_time, end\_time, description) VALUES (0, sysdate, TO\_DATE(‘MM/DD/YYYY’, ‘12/31/2002’), ‘<B>TC-1009 Incentive</B><BR><I>This is a new incentive!</I>’)” 3. Execute TCB-001. | The incentive just entered should be displayed on the page. |

# Test Environment

*For example, A new server is required for the web server, the application and the database.*

# Test Schedule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task Name *(sample is below, focus on spring 1 to start)*** | **Start** | **Finish** | **Effort** | **Comments** |
| *Test Planning* |  |  |  |  |
| *Review Requirements documents* |  |  |  |  |
| *Create initial test estimates* |  |  |  |  |
| *Learn new test resources* |  |  |  |  |
| *First deploy to QA test environment* |  |  |  |  |
| *Functional testing – Sprint 1* |  |  |  |  |
| *Iteration 2 deploy to QA test environment* |  |  |  |  |
| *Functional testing – Sprint 2* |  |  |  |  |
| *System testing* |  |  |  |  |
| *Regression testing* |  |  |  |  |
| *Usability Testing* |  |  |  |  |
| *Resolution of final defects and final build testing* |  |  |  |  |
| *Deploy to Staging environment* |  |  |  |  |
| *Performance testing* |  |  |  |  |
| *Release to Production* |  |  |  |  |