Software Test Plan

Version 1.0

Mental Health App

By:

Daniel Bornemann
Joshua Breininger
Phi Duong

Table of Contents

1. Introduction

1.1 Objective

The objective of this test plan is to outline the test cases that will be used to determine whether or not the features listed in the Software Requirements Specification are following the correct behavior for the mental health app. The test cases are focused on the interactions between the user and the app. Since the app does not require an internet connection, the tests are written considering the user and the app to be a closed system.

1.2 Purpose

The purpose of this test plan is to have test cases that can be used to find all possible bugs in the mental health app. The test cases are written to include both correct and incorrect inputs to account for all possible scenarios where a bug could occur. Correct inputs are inputs that are expected from the user in ideal scenarios. Incorrect inputs are inputs that are not expected from the user such as entering a letter where a number is expected. Using these inputs, most if not all bugs in the logic of the app can be found.

1.3 Approach

The approach of the tests is to run through them multiple times with slight variations in the test cases. Since the app has a very limited number of inputs, it is important to test a few variations per input field.

2. Test Cases

2.1 Test Case 1 - Input Daily Information Into the App

Brief Description: The user inputs their daily information into the app when they start it.

Goal: The user is able to input their daily information.

Precondition(s): The app is installed.

Scenario 1: Ideal Scenario

Steps	Expected Result
1. The user starts the app.	1. The app opens with the home screen being displayed.
2. The user selects the “log all-in-one” button from the home screen.	2. A pop-up appears with the mood slider showing.
3. The user logs their daily mood as 7 and selects the “ok” button.	3. The information is stored in the database and the food group selection is showing.

4. The user selects “fruits” and “high sugar foods” and selects the “ok” button.	4. The information is stored in the database and the exercise screen is showing.
5. The user inputs “30” as the amount of time they exercised and selects the “ok” button.	5. The information is stored in the database and the pop-up closes. The daily game scenario also changes.

Scenario 2: Skip Logging Information

Steps	Expected Result
1. The user starts the app.	1. The app opens with the home screen being displayed.
2. The user selects the “log all-in-one” button from the home screen.	2. A pop-up appears with the mood slider showing.
3. The user selects the “skip” button.	3. The food group selection is showing.
4. The user selects the “skip” button.	4. The exercise screen is showing.
5. The user selects the “skip” button.	5. The pop-up closes. The daily game scenario does not change.

Scenario 3: Unfinished Logging

Steps	Expected Result
1. The user starts the app.	1. The app opens with the home screen being displayed.
2. The user selects the “log all-in-one” button from the home screen.	2. A pop-up appears with the mood slider showing.
3. The user logs their daily mood as 7 and selects the “ok” button.	3. The information is stored in the database and the food group selection is showing.
4. The user selects the “skip” button.	4. The exercise screen is showing.
5. The user inputs “30” as the amount of time they exercised and selects the “ok” button.	5. The information is stored and the pop-up closes. The daily game scenario does not change.
6. The user selects the “log all-in-one” button from the home screen.	6. A pop-up appears with the food group selection showing.
7. The user selects “fruits” and “high sugar foods” and selects the “ok” button.	7. The information is stored in the database and the pop-up closes. The daily game

	scenario also changes.

Note: Pressing the “ok” button with no input will cause the app to move on to the next screen in the same way that the “skip” button works. The input can be retried by simply selecting the “log all-in-one” button again as seen in this scenario.

Scenario 4: Already Logged Information

Steps	Expected Result
1. The user starts the app.	1. The app opens with the home screen being displayed.
2. The user selects the “log all-in-one” button from the home screen.	2. A pop-up appears with the mood slider showing.
3. The user logs their daily mood as 7 and selects the “ok” button.	3. The information is stored in the database and the food group selection is showing.
4. The user selects “fruits” and “high sugar foods” and selects the “ok” button.	4. The information is stored in the database and the exercise screen is showing.
5. The user inputs “30” as the amount of time they exercised and selects the “ok” button.	5. The information is stored in the database and the pop-up closes. The daily game scenario also changes.
6. The user selects the “log all-in-one” button from the home screen.	6. The app displays a pop-up with the message “You have already logged your daily information.”
7. The user selects the “ok” button on the pop-up.	7. The pop-up closes.

Scenario 5: Unusual Inputs

Steps	Expected Result
1. The user starts the app.	1. The app opens with the home screen being displayed.
2. The user selects the “log all-in-one” button from the home screen.	2. A pop-up appears with the mood slider showing.
3. The user logs their daily mood as 3 and selects the “ok” button.	3. The information is stored in the database and the food group selection is showing.
4. The user selects “vegetables” and selects	4. The information is stored in the database

the “ok” button.	and the exercise screen is showing.
5. The user inputs “test” as the amount of time they exercised.	5. An error message displays “Invalid characters detected, please try again.”
6. The user inputs “-30” as the amount of time they exercised.	6. An error message displays “Invalid characters detected, please try again.”
7. The user inputs “1441” as the amount of time they exercised.	7. An error message displays “Invalid time detected, please try again.”
8. The user inputs “60” as the amount of time they exercised and selects the “ok” button.	8. The information is stored in the database and the pop-up closes. The daily game scenario also changes.

2.2 Test Case 2 - View Previously Logged Information

Brief Description: The user can view information that was logged previously.

Goal: The user is able to view previously logged information from up to a year ago.

Precondition(s): The app is installed and the user has previously logged information from a week ago.

Scenario 1: Ideal Scenario

Steps	Expected Result
1. The user starts the app.	1. The app opens with the home screen being displayed.
2. The user selects the left arrow button that is under the weekly calendar section of the app.	2. The week moves back by one and the previous week is displayed.
3. The user clicks on Tuesday.	3. A pop-up appears with a summary of the information inputted on that day.
4. The user selects the “ok” button.	4. The pop-up closes.

Scenario 2: No Data

Steps	Expected Result
1. The user starts the app.	1. The app opens with the home screen being

	displayed.
2. The user selects the right arrow button that is under the weekly calendar section of the app.	2. The week moves forward by one and the next week is displayed.
3. The user clicks on Friday.	3. A pop-up appears with the text “No data” under the summary of the day.
4. The user selects the “ok” button.	4. The pop-up closes.
5. The user selects the left arrow button twice.	5. The week moves back by two and the previous week (from the current) is displayed.
6. The user clicks on Monday.	6. A pop-up appears with the text “No data” under the summary of the day.
7. The user selects the “ok” button.	7. The pop-up closes.

2.3 Test Case 3 - View Graphs Created From Data

Brief Description: The user can view the graphs that are created after daily and previous information is logged.

Goal: The user is able to view the resulting graphs from inputting daily and previous information.

Precondition(s): The app is installed and information has been logged for the past year.

Scenario 1: View Mood Graphs

Steps	Expected Result
1. The user starts the app.	1. The app opens with the home screen being displayed.
2. The user selects the “Graphs” button.	2. The graph pop-up opens with no graphs currently displayed.
3. The user selects the “Mood” and “Today” buttons.	3. A line graph with one point is displayed at the current date.
4. The user selects the “Week” button.	4. A line graph with the current week’s mood information is displayed.

4. The user selects the “Month” button.	4. A line graph with the current month’s mood information is displayed
4. The user selects the “Year” button.	4. A line graph with the current year’s mood information is displayed.
7. The user selects the “ok” button.	7. The pop-up closes.

Scenario 2: View Exercise Graphs

Steps	Expected Result
1. The user starts the app.	1. The app opens with the home screen being displayed.
2. The user selects the “Graphs” button.	2. The graph pop-up opens with no graphs currently displayed.
3. The user selects the “Exercise” and “Today” buttons.	3. A bar graph with one bar is displayed at the current date.
4. The user selects the “Week” button.	4. A bar graph with the current week’s exercise information is displayed.
4. The user selects the “Month” button.	4. A bar graph with the current month’s exercise information is displayed
4. The user selects the “Year” button.	4. A bar graph with the current year’s exercise information is displayed.
7. The user selects the “ok” button.	7. The pop-up closes.

Scenario 3: View Diet Graphs

Steps	Expected Result
1. The user starts the app.	1. The app opens with the home screen being displayed.
2. The user selects the “Graphs” button.	2. The graph pop-up opens with no graphs currently displayed.
3. The user selects the “Diet” and “Today” buttons.	3. A bar graph with one bar is displayed at the current date.
4. The user selects the “Week” button.	4. A bar graph with the current week’s diet information is displayed.

4. The user selects the “Month” button.	4. A bar graph with the current month’s diet information is displayed
4. The user selects the “Year” button.	4. A bar graph with the current year’s diet information is displayed.
7. The user selects the “ok” button.	7. The pop-up closes.

Scenario 4: All In One Graphs

Steps	Expected Result
1. The user starts the app.	1. The app opens with the home screen being displayed.
2. The user selects the “Graphs” button.	2. The graph pop-up opens with no graphs currently displayed.
3. The user selects the “Mood,” “Exercise,” and “Diet” buttons.	3. Three different colored graphs are created at the current date.
4. The user selects the “Week” button.	4. Three different colored graphs are created with the week’s information displayed.
5. The user selects the “Month” button.	5. Three different colored graphs are created with the month’s information displayed.
6. The user selects the “Year” button.	6. Three different colored graphs are created with the year’s information displayed.
7. The user selects the “ok” button.	7. The pop-up closes.

2.4 Test Case 4 - Notifications and Settings

Brief Description: The user can adjust reminder settings and view reminders.

Goal: The user is able to receive reminders and adjust the settings.

Precondition(s): The app is installed and data from the previous week has been entered.

Scenario 1: Enable Both Reminders (Ideal Diet and Exercise)

Steps	Expected Result
1. The user starts the app.	1. The app opens with the home screen being

	displayed.
2. The user selects the “Settings” button.	2. A pop-up appears with the reminder settings showing.
3. The user enables diet and exercise reminders for every “7” days. The user selects the “ok” button.	3. The reminders are saved and the pop-up closes.
4. The user waits 7 days, each day the user inputs fruits and vegetables into diet and 30 minutes of exercise.	4. Two reminders will appear on the user’s device encouraging the user to keep up the good work for both diet and exercise.

Scenario 2: Enable Both Reminders (Unideal Diet and Exercise)

Steps	Expected Result
1. The user starts the app.	1. The app opens with the home screen being displayed.
2. The user selects the “Settings” button.	2. A pop-up appears with the reminder settings showing.
3. The user enables diet and exercise reminders for every “7” days. The user selects the “ok” button.	3. The reminders are saved and the pop-up closes.
4. The user waits 7 days, each day the user inputs high sugar foods into diet and 0 minutes of exercise.	4. Two reminders will appear on the user’s device encouraging the user to try and stay active and to try and eat less high sugar foods.

Scenario 3: Disable Both Reminders

Steps	Expected Result
1. The user starts the app.	1. The app opens with the home screen being displayed.
2. The user selects the “Settings” button.	2. A pop-up appears with the reminder settings showing.
3. The user disables diet and exercise reminders.	3. The reminders are saved and the pop-up closes.
4. The user waits 7 days.	4. No reminders appear on the user’s device.

Scenario 5: Unusual Inputs

Steps	Expected Result
1. The user starts the app.	1. The app opens with the home screen being displayed.
2. The user selects the “Settings” button.	2. A pop-up appears with the reminder settings showing.
3. The user enables diet and exercise reminders for every “TEST” days. The user selects the “ok” button.	3. An error message displays “Invalid characters detected, please try again.”
4. The user enables diet and exercise reminders for every “-1” days. The user selects the “ok” button.	4. An error message displays “Invalid characters detected, please try again.”
5. The user enables diet and exercise reminders for every “0” days. The user selects the “ok” button.	5. An error message displays “Invalid number detected, please try again.”
6. The user enables diet and exercise reminders for every “3” days. The user selects the “ok” button.	6. The reminders are saved and the pop-up closes.
7. The user waits 3 days, each day the user does not input anything	7. Two reminders will appear on the user’s device encouraging the user to input their exercise and diet.

2.5 Test Case 5 - View Daily Game Information

Brief Description: The user can view the daily game scenario and adventure log after inputting their daily information.

Goal: The user is able to view the daily game scenario and adventure log, with the game scenario changing every day.

Precondition(s): The app is installed and information has been logged for the past week.

Scenario 1: View Scenario After Logging Information

Steps	Expected Result
1. The user starts the app.	1. The app opens with the home screen being displayed, the previous day’s game scenario is

	viewable.
2. The user selects the “log all-in-one” button and inputs their information.	2. The game scenario changes.
3. The user selects “View Adventure Log” from the home screen.	3. The adventure log pop-up contains daily scenario text from all inputted days including the current day.
4. The user selects the “ok” button.	4. The adventure log pop-up closes.

Scenario 2: View Scenario After Skipping Logging Information

Steps	Expected Result
1. The user starts the app.	1. The app opens with the home screen being displayed, the previous day’s game scenario is viewable.
2. The user selects the “log all-in-one” button and skips inputting their information.	2. The game scenario does not change.
3. The user selects “View Adventure Log” from the home screen.	3. The adventure log pop-up contains daily scenario text from all inputted days excluding the current day.
4. The user selects the “ok” button.	4. The adventure log pop-up closes.

2.6 Test Case 6 - First Activation

Scenario 1: New Ideal User

Steps	Expected Result
1. The user starts the app for the first time	1. The app opens with the home screen being displayed. There is no game scenario displayed.
2. The user enters a name within a certain number of characters for the pop up requesting a name.	2. The app saves the name for usage in the scenarios.
3. The user selects the “log all-in-one” button and inputs their information.	3. The game scenario changes.
4. The user selects “View Adventure Log”	4. The adventure log pop-up contains daily

from the home screen.	scenario text from the current day with the correct name.
5. The user selects the “ok” button.	5. The adventure log pop-up closes.

Scenario 2: Too Long Name

Steps	Expected Result
1. The user starts the app for the first time	1. The app opens with the home screen being displayed. There is no game scenario displayed.
2. The user enters an extremely long name for the pop up requesting a name.	2. The app states the name is too long and requests a new one.
3. The user enters a name within a certain number of characters for the pop up requesting a name.	3. The app saves the name for usage in the scenarios.
4. The user selects the “log all-in-one” button and inputs their information.	4. The game scenario changes.
5. The user selects “View Adventure Log” from the home screen.	5. The adventure log pop-up contains daily scenario text from the current day with the correct name.
6. The user selects the “ok” button.	6. The adventure log pop-up closes.

Scenario 3: Empty Name

Steps	Expected Result
1. The user starts the app for the first time	1. The app opens with the home screen being displayed. There is no game scenario displayed.
2. The user enters nothing for the pop up requesting a name.	2. The app states the user needs to enter a name, and requests a new one.
3. The user enters a name within a certain number of characters for the pop up requesting a name.	3. The app saves the name for usage in the scenarios.
4. The user selects the “log all-in-one” button and inputs their information.	4. The game scenario changes.
5. The user selects “View Adventure Log”	5. The adventure log pop-up contains daily

from the home screen.	scenario text from the current day with the correct name.
6. The user selects the “ok” button.	6. The adventure log pop-up closes.