Software Requirements Specification

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 requirement specification document is to outline the requirements and specifications of the mental health app. This document will specify the expected behavior of the mental health app and include sample inputs and outputs. The program should at its core be able to store information to a database and retain it to present in a graphical format while also using the data to generate reminders to help the user.

1.2 Purpose

The purpose of the mental health app is to assist college age students in managing their mental health. Due to the recent pandemic, students have been isolated at home which has reduced social interaction and the ability to go outside. This has been negatively impacting the mental health of society as a whole. The goal of this app is to encourage college students to improve their mental health through self monitoring. This app does not serve as a replacement for a therapist or professional advice but rather as a compilation of features that other mental health apps include as well as a motivator to use these features. These features will be expanded on in **Section 2.2.**

1.3 Scope

The target audience for this app is college age students. To account for the busy lives of college students, the goal of the app is to be as unobtrusive as possible while maintaining functionality. The app also includes a short game (expanded on in **Section 2.3**) that acts as a motivator for users to log their information into the app daily. Though there exist other apps that rigorously keep track of information, this app is specifically designed to be still useful to those who may not have the motivation or knowledge to thoroughly log and keep track of the specifics (portions, exact types of food, etc) of their diet or exactly what kind of exercises they do.

1.4 Stakeholders

College aged adults are the primary stakeholder for this project. The app is directed to them for their priorities and generational preferences to help motivate them to monitor their mental health and keep them more aware of their moods over an extended period of time.

Florida Tech is also a stakeholder in that this app may contain somewhat sensitive information that, although not bound by specific laws such as medical information, would likely cause discomfort and reflect poorly on Florida Tech if it was exposed.

1.5 Definitions and Acronyms

1.4.1 GUI: Graphical User Interface

2. Product Description

2.1 Product

The product is an app. The app will be available only for Android devices and will be developed to support smart phones specifically. It will not connect to the internet and should take up relatively little space to remain as unobtrusive as possible.

2.2 Features

This section will outline the key features of the app that will be the main focus. These features are functional and make up the core functionality of the app.

2.2.1 Input Daily Information Into the App

The user will be able to input their daily information into the app when they start it. This information includes their mood, daily eaten food groups, and whether or not they exercised that day. The mood will be on a scale from 1-10 using a slider. The app will allow for the selection of daily eaten food groups. The food groups that can be selected are high sugar foods, fruits, and vegetables currently. The user will input the exercise information into a text box as a time amount in minutes. The user has the option to skip any of those information requests, and the app will not count it as they did not exercise or did not eat any of those food groups and instead simply not count that day when keeping track of the skipped questions. This information can be recalled with a window accessible by the main UI, which presents the user with a graph. The user can change the time frame being the x axis of the graph to view data on the date, over a week, a month and a year. The user can also change what data is presented to them: a line graph for their mood, a multi bar graph with three bars representing the three tracked food groups, and a line graph of exercise time. The buttons that make these graphs appear can all be pressed at once, allowing the user to directly view their trends and bring to their attention any possible correlation.

Ideal scenario:

1. The user starts the app.
2. The user selects the “log all-in-one” button from the home screen.
3. The user logs their daily activities (mood level, food intake, exercise).
4. The user navigates to graphs.
5. The user selects to see mood, diet and exercise graphs.
6. The user expands the graphs to view trends over the past month.
7. The user closes the graph pop up.
8. The user examines the new game scenario.
9. The user closes the app.

Sample input for feature (for Step 3 above):

1. Mood level: User uses the slider to select 7

2. Food intake: User selects fruits and high sugar foods
3. Exercise: User types 30 minutes

Sample output for feature (for Step 4 above):
A line graph is created and displayed for the 1-10 mood scale based on the user's inputs over a certain period of time. A multibar graph is created and displayed for both the inputted food intake and exercise (one graph for each) over a certain period of time. The periods of time are today, the current week, the current month, and the past year, meaning there are 4 graphs for each of the 3 inputs.

Sample incorrect behavior (for Step 4 above):
No graph is displayed or an incorrect graph is shown.

2.2.2 Database

The logged activities will be stored in a SQLite database. The database will hold all the daily information up to a year and is accessible to the user through graphs. This database should correctly save and recall this information while also pruning the data to ensure the database does not get large enough to influence the user into uninstalling to save phone space. See the above **Section 2.2.1** for more information about the expected behavior and sample inputs.

2.2.3 View Logged Information From Previous Days

The user will be able to view information that was inputted from up to a year ago. This information includes their mood, daily eaten food groups, and whether or not they exercised that day.

Ideal scenario:
1. The user starts the app.
2. The user selects a previous day from the calendar.
3. The user is able to view the inputted information from that day in a pop up.
4. The user closes the pop up.
5. The user closes the app.

Sample output for feature (for Step 3 above):
In the pop up a full overview of the inputted information for that day will be displayed. This information includes the mood, food groups, and amount of exercise and their respective times of input.

Sample incorrect behavior (for Step 3 above):
No information is displayed in the pop up or incorrect information is displayed.

2.2.4 Notifications

The app will notify the user to log in their daily information. The notifications will encourage the user to stay healthy. If the user doesn't stay active, the app will notify the user to try and get some exercise. If the user doesn't stay on a balanced diet in the basic level of the app, the notification can send recommendations on if they should eat more fruits and vegetables or if they should eat less high-sugar food. These recommendations are different for every user so the settings can be adjusted through the settings menu on the home screen. The settings include the ability to disable notifications for diet and exercise separately and the frequency of reminders in days. There are no mood reminders as such reminders would probably feel invasive and judgemental to the user, decreasing the usefulness of the app.

Ideal scenario:

1. The user starts the app.
2. The user navigates to the settings menu from the home screen.
3. The user selects the desired reminder statuses and frequency.
4. The user closes the settings pop-up.
5. The user closes the app.
6. X days later the user receives reminders later where X is the number of days set by the user.

Sample input for feature (for Step 3 above):

1. Reminders: User enables exercise reminders and disables diet reminders
2. Frequency: User sets the frequency of exercise reminders to every 7 days

Sample outputs for feature (for Step 6 above):
Precondition: The user has logged no exercise for the past 7 days.
The user receives a notification that encourages the user to exercise more as they had not logged any exercise for the past 7 days.

Precondition: The user has logged some exercise for the past 7 days.
The user receives a notification that encourages the user, telling them that they have been doing well in exercising and to keep up the good work.

Sample incorrect behavior (for Step 6 above):
The user does not receive any notification regarding exercise, or they receive an incorrect notification.

2.3 Additional Features

This section will outline additional features of the app that are not the main focus. These features are not part of the core system and will have less development time put into them.

2.3.1 Daily Game

On the main menu of the app is a window that displays an environment with a player character and a randomly generated object. Under the window is a text box describing a generated scenario that occurred that day for the player character. When the user logs their daily information for the first time in a day, that window and description generates again to create a new situation and art, saving the scenario in text form in a log available to the user. Initially the user will have a default character pixel art assigned to them, although depending on the future of the project multiple different avatars could be selected. On first activation of the app, the user will write a name for the avatar, which will be saved and used in the scenario writing.

Ideal scenario (On first activation):

1. The user starts the app for the first time.
2. The user is presented with a pop up asking for a name, which they fill in.
3. The user presses “ok” closing the pop up.
4. The user selects the “log all-in-one” button from the home screen and logs their information.
5. The user closes the pop-up.
6. The user is able to view an updated game scenario from the home screen with their name saved.
7. The user closes the pop-up.

Ideal scenario (Not first activation):

1. The user starts the app.
2. The user selects the “log all-in-one” button from the home screen and logs their information.
3. The user closes the pop-up.
4. The user is able to view an updated game scenario from the home screen.
5. The user clicks on the view log button and a window pops up with a text list of all scenarios that user has experienced.
6. The user closes the pop-up.

The game should accurately record its scenario for an extended period of time in a formatted way.

Sample Incorrect Input:
The user attempts to write a name that is too long for the app to store or tries to write an empty name.

Sample Incorrect Behavior:

The scenario generated is the same as the last one or the scenario is improperly saved to the log and shows a different scenario to what was experienced, replacing a previous scenario, or not being appended to the last line in the file. The app does not correctly save the player's name and does not replace the default name for the scenarios properly.

The daily game uses a random generator to randomly select a text scenario, and randomly pick an "object" or entity related to that scenario and pick it from the stored graphical files. This should produce a new scenario the user has not seen before due to chance, as enough scenarios should be written so that the exact same scenario is highly unlikely to occur again in the exact same way unless the user uses the app for an extremely long period of time. The game also should pick a scenario and object that thematically matches the current theme of the app.

Sample Incorrect Behavior of Generator:
The scenario generated is the exact same in every way (text, object) to a recent scenario. The generator selects an unrelated object to the scenario where it does not make sense, such as the user's avatar having a conversation with a bonfire. The generator does not properly obtain a scenario or object and prints nothing or an error message. The generator selects an object or scenario unrelated to the current game theme, such as encountering a giant fish in the forest.

2.3.2 App Theme

The app will, upon starting, check for the date from the Android system. If the app detects a new month, then it will change the overall theme of the app. This will replace almost all graphical components of the app, having a different background, designs and edges of pop ups, although it will not change the UI format, as in where the buttons are. The themes are designed after fantasy inspired environments, and will be made to be visually very different from each other. They also will be somewhat related to the month they occupy, with a winter themed environment during December. This is meant to motivate the user to use the app by giving a sense of progression. The aim of the themes will be to be visually appealing without inspiring any negative emotions or feeling exhausting to look at, such as a desert. The theme will also change the daily game, making the generator pick thematic scenarios and objects to align with the theme. The theme's graphical elements will be stored in the app so it doesn't have to connect to the internet to download new data, which makes the app more secure. Since the elements will be fairly small and there will only be twelve months of them, they should not take up a large amount of space on the device.

Sample Incorrect Behavior of App Theme Selector:

The app selects the incorrect theme for the month. The app does not check for the new theme and so does not implement it. The app improperly applies the theme resulting in a blank background or an error and crash from not properly fetching the images to use. The app does not properly

notify the game generator of the correct theme and results in the generator creating scenarios not fitting the theme. The app changes themes in the middle of the month.

2.3.3 Potential Future Requirements

There are some future additional requirements that could be added to the project if time allows, listed in order by priority.

1. Multiple game avatars for the user to select.
2. More elaborate game mechanics.