INDY-1 Fitness App

Software Design Specification

CS 4850 Section 2

Spring Semester 2024

Professor Perry

02/13/2024

Luke Zeches

Adam Borowski

Michael Bolnik

Table of Contents

1. Introduction

1.1 Document Overview

This document is being formulated in order to provide a streamline overview of the design of our fitness app. We are looking to provide information about components and constraints of our system and showing how the systems will interact in order to allow all the functionalities to perform in the fashion that they are designed to. We are also looking to specify the major functionalities of our system and what each system is going to be responsible for in terms of its own role in the function of the app.

The scope of our project entails an initial version in which the functionality will include a system for creating and storing a new account made through the use of personal data as well as an interface that will allow the user to see what they need to include and how to navigate. It will also have a system responsible for allowing the start or continuation of a trip as well as a navigational tool to see the statistics of the users' exercise time. There will be another system that is responsible for the interface of the actual trip which will involve getting user data and displaying that data in a way where it is mapped onto a real-time setting of some place that the user would like to go.

The end goal will be to make sure that the user can choose a preallocated trip that will give the user a visual of some kind of place that they might want to travel to and use real time user data from a fitness tracking app in order to map the user's path in a real world setting. This app is intended for any user's that want to exercise while also getting the experience of being in a different setting while exercising, setting goals of where they want to go and providing the ability to be visually in some different place around the world. This can also be used as a regular fitness app as it will have many of the major functionalities of a fitness app with lots of data that can be used to track one's own fitness levels. This document will contain mainly methodologies and system overviews for how our app is designed and show most of the system components and their functionalities in running the entire system.

1.2 System Overview

Our app will use several systems to ensure correct functionality. The first system is connectivity to the Android Health app. This application automatically tracks the distance traveled, so our program will only need to fetch the distance information. This functionality will only work when the user is currently on a trip. These distance metrics will be used to update the current trip while users have the app open. To ensure no data leaks occur, we will connect our app to a security system that encrypts data. The security system will be in effect while a user logs into their account or creates a new one.

The next system included will be a database that holds data about each user. This includes their profile information (username, password, email address) and stats that update as

they use the app such as total distance traveled and elapsed time. The last system included is Google Street View. Once selected, the app will open Google Street View so users can see where they are on their current trip. It will allow participants to look around freely but will only update once they are moving. Once these systems are in place, the app will have full functionality.

2. Design Considerations

2.1 Assumptions and Dependencies

The operating system that our app will run on is Android. For this reason, any APIs we use will need to be compatible with this OS. This includes the distance tracking system, the database we use, and the security system. Our group considered developing for Android and IOS together, but IOS was deemed too difficult as it's not open-source.

2.2 General Constraints

Since our app is being developed for mobile devices, there are certain constraints we must follow. The first of these involves the GUI. All interface design should be compatible with the 9:16 aspect ratio of a phone and be readable on small screens. Mobile devices impart other restrictions not seen in desktop applications. The app is built to be used outside, so we need to ensure that it doesn't consume too much battery life while running. Storage space is also a factor since phones don't have as much space as a desktop computer. If the app uses more space than necessary, it could be difficult for users to download. Performance is another concern. While running the app, it shouldn't be slow or stutter. Distance will be updated every few seconds, but our goal is to make movement on maps and Google Street View as smooth as possible. Connection to a network must also be fast and secure. Experiencing connection issues while using the app would ruin the user experience.

2.3 Goals and Guidelines

Our goals for the design of the app is to keep it simple yet functional. This means that there is no learning curve for using the app and that the functions of each design element are immediately recognizable. We also want to focus on making sure that our app does not suffer from any possible overuse of memory that might have brought down the performance of our app, so we looked to deploy memory management practices for any system that required storage of data. The last goal was to make sure that the design of our app was simple with a sleek look that had only the necessary information that the user needed to maneuver through the app. We want the app to have the feel of a regular fitness app with the addition of a couple of custom features which will display the app's originality. We hope to develop this app without necessarily innovating in the main features of fitness apps but rather innovate in the possibility of extending outside of the usual realm of data by providing the user with a different type of experience and interfacing method.

2.4 Development Methods

Since we are developing our app to be simple and functional, we aim to make the UI easy for anyone to understand. This guideline is true for the design of our systems too. The app is meant for tracking exercise, so this should be immediately clear for users. We don't want to waste any of the user's time navigating menus or setting up accounts, so starting and viewing a trip will be quick and simple. We also plan to make our app feel unique but still have a familiar foundation as other fitness apps. There are countless other apps which track your steps, so we want ours to stand out. The integration of Google Street View and the ability to virtually travel to new locations will help us achieve this goal.

3. Architectural Strategies

One of the first architectural strategies that we employed started with the framework that we chose to work with. We decided to go with React Native due to its purpose of developing apps, more specifically Android apps, so that we had all the tools and methods we needed to develop the app in the scope that we wanted to. We are also constrained to using Javascript as this is the language that is used when working with the React framework. In terms of databases, we had some options of creating our own database through SQL server and using an API to link our system with the database or to do it directly through React using the built in functionalities for creating a database to store data locally. These options were chosen in order to maintain a relatively simple approach to developing the app and making sure that all the components were compatible.

In terms of reuse of features, our design was relatively simple and straightforward in the first place but there were different areas where information such as personal information or statistics could be reused for different output results on the interfaces that were including relative information. This software will have much room for improvements and extension as the baseline model will simply look like an ordinary fitness app with some custom features and the live view while further extensions could involve systems such as new map possibilities where the user could find themselves in virtual spaces rather than real life locations or we could look to incorporate further functional capabilities such as a system where the app will allow you the option to block certain sites or apps such as Youtube or Instagram until a your preset exercise goal is complete.

In terms of any user interface paradigms, the main input our systems take entails customer information and distance information that will be gained from the health app that we draw our data from. There will also be a section of our system that is responsible for recording fitness input data that will be important for calculating any statistics that we wish to include as well as to link the input data to the mapped out interface that we will show the path on. Error detection will be mainly be used to ensure that the user only follows the possible actions that will be displayed on our interfaces with certain functions to make sure that issues such as validity of enter information and possible issues with the input data to map translations occur without any problems that could crash or severely harm the functioning of our systems. We will make sure to

efficiently use memory by reusing any objects that can be used in multiple areas as well as optimizing our user interfaces through the use of simple layouts that don't lead to any complicated view hierarchies.

4. System Architecture

Our app will be broken down into several systems and subsystems. The most important system that will run through our app will be the distance tracker. By taking the distance information collected by the Android Health app, we can track the user's steps without needing specific location data. This system will only be in effect when the user taps a button to 'start a workout'. Another important system will be the use of Google Street View. This can be used by participants when on a trip. It will use the virtual location of the user (their location on the current trip) to pull from the Google Street View API. With this, users can see the surroundings of their trip. When the user travels a certain distance, this view will update as if they're physically at the location. This system will be started by the user and won't run in the background. The last major system will be a database which stores user data. Each user will be able to access their account information through the app and all of the data will be stored in a single secure database. This database will also be updated as users accumulate more miles while using the app. When logging in, the user's input will be compared to the information in the database to ensure they're entering the correct information.

5. Policies and Tactics

In the beginning stages we discussed how our trip system would work. There were two possible ways to implement this feature; either a predetermined trip would be set or users would begin a trip from their current location. We decided that creating predetermined trips would work better since it would be much less complicated to implement. Another reason is that a leaderboard system would be impossible if every user was on a unique trip.

We made the decision to code our app using Android Studio and use React Native as the framework. With these tools, we plan to write our code so it's easy to read and debug. As mentioned above, we will use a database to store user information. In this database, usernames, passwords, and emails will be used as primary keys which will identify users from each other. Data such as distance and elapsed time will be attributes that users can retrieve from their account settings.

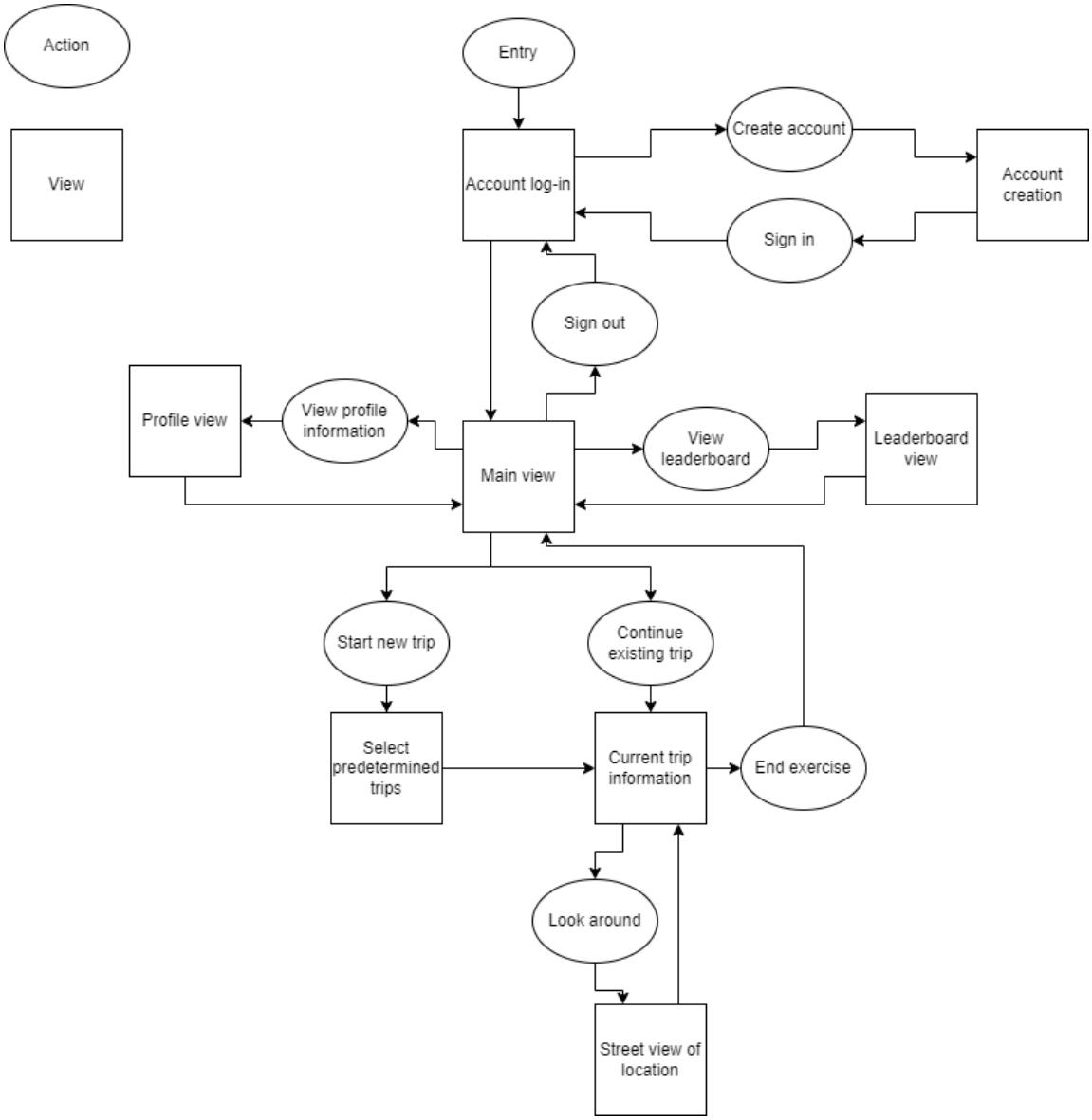
To secure this database, all passwords will be encrypted using an API. Passwords won't be encrypted in the database, but the token which sends the password from the user to the database will be. This is to stop anyone who tries to steal passwords while they're transmitted from the app.

Testing our app is another important step in development. We must ensure that all systems, such as the distance tracker, work properly. In order to test this, we will walk a

predetermined distance and watch the tracker to see if it records the correct distance. Creating an account is another important system that will require testing. To accomplish this, we plan on creating several temporary accounts with unique login information. If the profile information shows up correctly in the database, we know it works as intended. Testing security is another important step for protecting our users. When we can make sure the encryption algorithm is working correctly, our app will have proper security measures. Further tests will be run throughout the process to check if smaller features such as buttons and screen transitions are running smoothly.

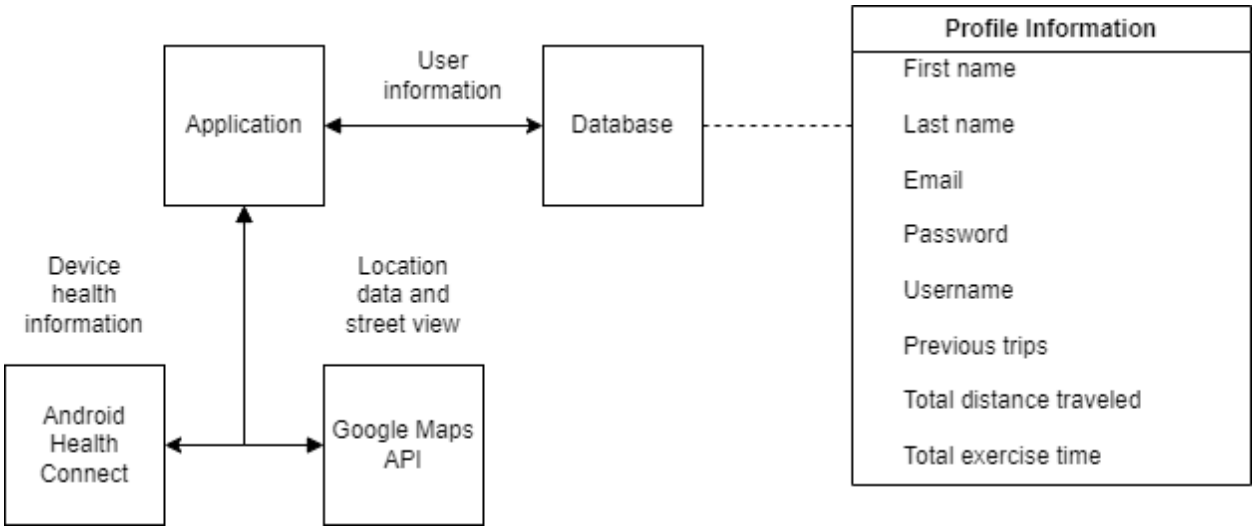
6. Detailed System Design

6.1 User experience system



This diagram shows the various views available to the user and the connections between them. The user is first greeted with a log-in screen, with the option to move to an account creation screen. After logging in, they are taken to the main view, where they can start a new trip from a set of pre-generated trips or, if one already exists, they can choose to continue exercising on the existing trip. While exercising, they can look around the street view of where they are on the trip. They can also exit exercising to return to the main view.
From the main view the user can view their profile information, and view the leaderboard of other users' exercise activity. These views provide a return to the main view.

6.2 System design



This diagram shows the various components of the system and their connections. User information is passed between the application and database, such as name, username, and exercise metrics. The application communicates with the Android Health Connect service to get health information recorded by the device while the user is exercising. The application communicates with the Google Maps API to display maps and routes, and to access the Google Street View data.