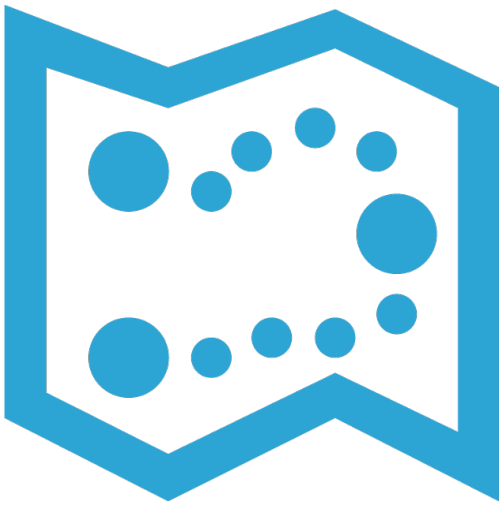


Find Your Path

Team 25 - Design Document

Revanth Nagurla, Aldio Ngudjiharto, Simron Patel, Willy Tanggono, Jackie Wu

Table Of Contents

Purpose

People often want to walk, run, or bike to new places or would like to enjoy the scenery while they are exercising. Our project aims to provide these features as a mobile application so people can enjoy working out even if they are not familiar with the exciting places around them. Whether they want to run by a trail, park, or the woods, we will provide our users with a route they want to experience.

Our functional requirements will include:

1. Customizable Routes
    1. Users can generate their own customizable routes based on their personal preferences
    2. Users can view and pick popular routes
    3. Users can choose past routes
    4. Users can generate routes based on scenery around their area

2. Personal Statistics
    1. Users can view time elapsed during their workout
    2. Users can view distance traveled during their workout
    3. Users can view how many calories they have burned for each workout
    4. Users can view average miles and pace per workout

5. Users can view how many times they ran a specific route
6. Users can view their own personal workout records
7. Users can sort and view their workout statistics based on walking, running, and biking

3. Notifications
    1. Users can receive notifications about daily reminders to workout
    2. Users can receive notifications to remind them to workout if they have not worked out for a while

4. Personal Accounts
    1. Users can register for an account
    2. Users can stay logged in their account
    3. Users can reset their password if they have forgotten it

5. Route conditions
    1. Users can view the weather before going on their customized route

6. Editing Routes
    1. Users can pause during their workout
    2. Users can modify the route in the middle of their workout

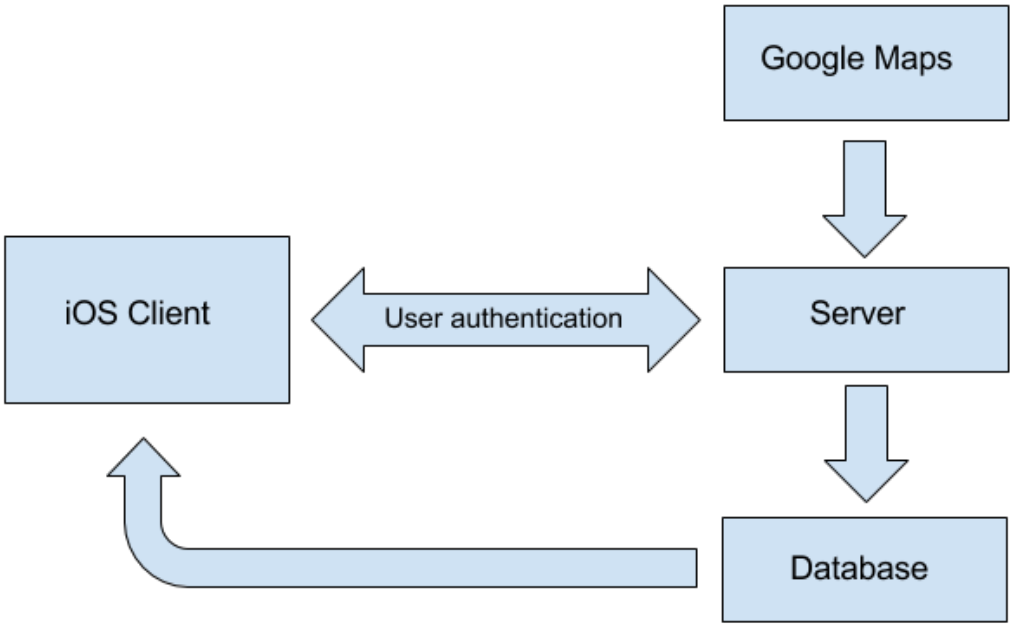
3. Users can stop their workout route

7. Route Directions
    1. Users can see an outline of their route on a map
    2. Users can receive audio directions of their route
    3. Users can see the directions of their route

Design Outline

1. High Level Overview

Figure 1: Overview Flowchart



Client server model - We decided to adopt a client-server model because we will have many clients submit their route preferences to a server in addition to Google Maps. The server will then combine and adjust the information then return it back to a client.

1. User authentication
    1. In order for user to look at their preferences across all devices
    2. To hide personal fitness data
2. IOS client
    1. Creates preferences for the route (ie. change in elevation/ scenery)
    2. Sends information to the server
    3. Receives preferences combined with Google Maps data
    4. Display map, route, and directions
    5. Sends user statistics to server once done with route
3. Google Maps
    1. Service that helps create the route
4. Server
    1. Receives instructions and routes from Google Maps and the client
    2. Uses the instructions to make adjustments to the route
    3. Sends the route to the client and tells Google Maps service to start
    4. Receives user's statistics from the client and sends it to the database
5. Database
    1. Receives statistics about each of the users

Design Issues

1. What type of architecture should we use?

- **Option 1: Client - Server architecture**
- Option 2: Peer-Peer architecture
- Option 3: Unified architecture

We decided to adopt a client-server model because we will have many clients submit their route preferences for a serve in addition to Google Maps. The server will then combine and adjust the information then return it back to a client. We considered using a peer-peer model where each client would also act as a server but we need some sort of middleware to modify and handle each route. We also chose not to have a unified architecture because we would not be able to work on the front end and back end simultaneously.Therefore we decided to go with the client-server model.

2. What database software should we use?

- **Option 1: MySQL**
- Option 2: SQLite

- Option 3: Oracle PL/SQL

Our group decided to use MySQL as our database software for our Find Your Path. One of the reason behind this decision is MySQL is one of the most widely used database in the world therefore there will be a lot of resources for our group members who are not familiar with MySQL. The cost is also one of the factor for our decision. MySQL is free for us to use but for Oracle PL we need to pay a certain amount of money and none of the group members have experience using Oracle PL. Oracle PL is also rated as a complex database that may cause more trouble for our group in the future. Furthermore SQLite is not suitable for our app because our app required the use of server but SQLite is not the best database to do server activities.

3. What front end framework should we use?

Option 1: React Native

Option 2: Swift/Java & XML

Option 3: Phonegap with Javascript

We decided to go with React Native because it allows us to do mobile application development through JavaScript, React, and HTML rather

than Java or Swift. We didn't choose option two since all of us are much more comfortable with JavaScript rather than Swift. We didn't go with Phonegap either because they charge some monthly fees.

4. What back end framework should we use?

Option 1: Node.js

Option 2: Python

Option 3: Java

We went with Node.js as our back end framework because we are familiar with it and also are using modules that we know are configurable with Node.js like Passport.js. It is a very popular back end framework so it has a lot of support.

5. What type of design scheme should we use?

Option 1: Responsive

Option 2: Adaptive

We decided to go with an adaptive design scheme because we will only be making this application for mobiles only. As a result, we wouldn't need a responsive scheme because we don't have to worry about how it looks on different platforms.

6. How should user navigate between personal data and the routes?

Option 1: Tab Bar

Option 2: Segmented Control Navigation Bar

Option 3: Side Bar

We decided to with option 1. Since we will only have two high level views, we can just have two icons in the tab bar making the UI accessible everywhere in the app. We do not have a lot high level views so using a sidebar is waste of space.

7. Do users need to have an account and login to use our services?

Option 1: No account needed to use our services

Option 2: Can log in with google or facebook account

Option 3: Create a unique username and password for our application

We decided to go with option 2. In order to keep track of the user's statistics, we need to have a record of the user in our database so they must need some sort of account. We didn't choose option three because it would be more of a hassle for the user's to remember their username

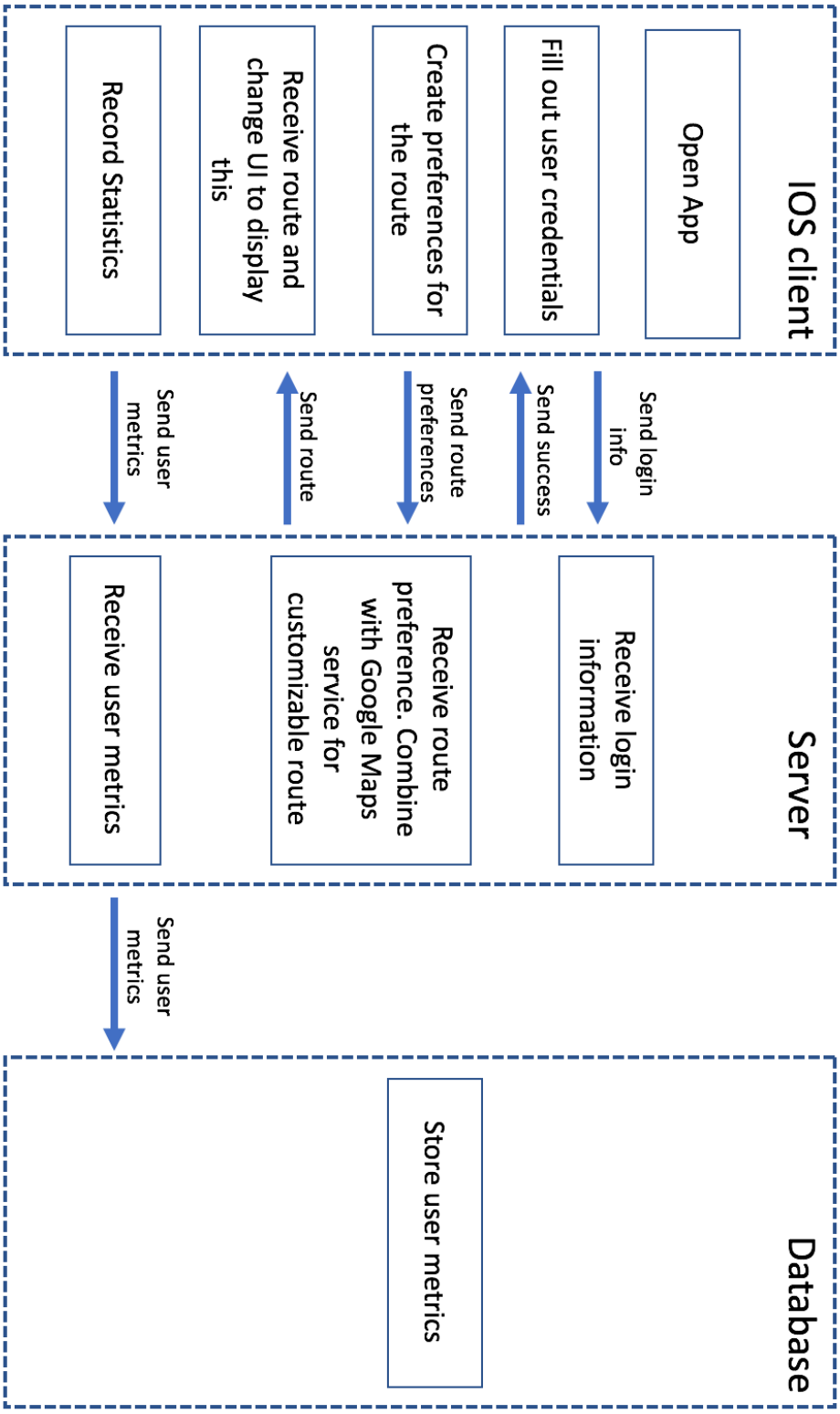
and password and we also have to worry about the security aspect of storing that information. We went with option 2 because it will be easy for the user to login to their account and it would be easier for us to store their information and statistics in our database.

Design Details

1. Sequence of Events

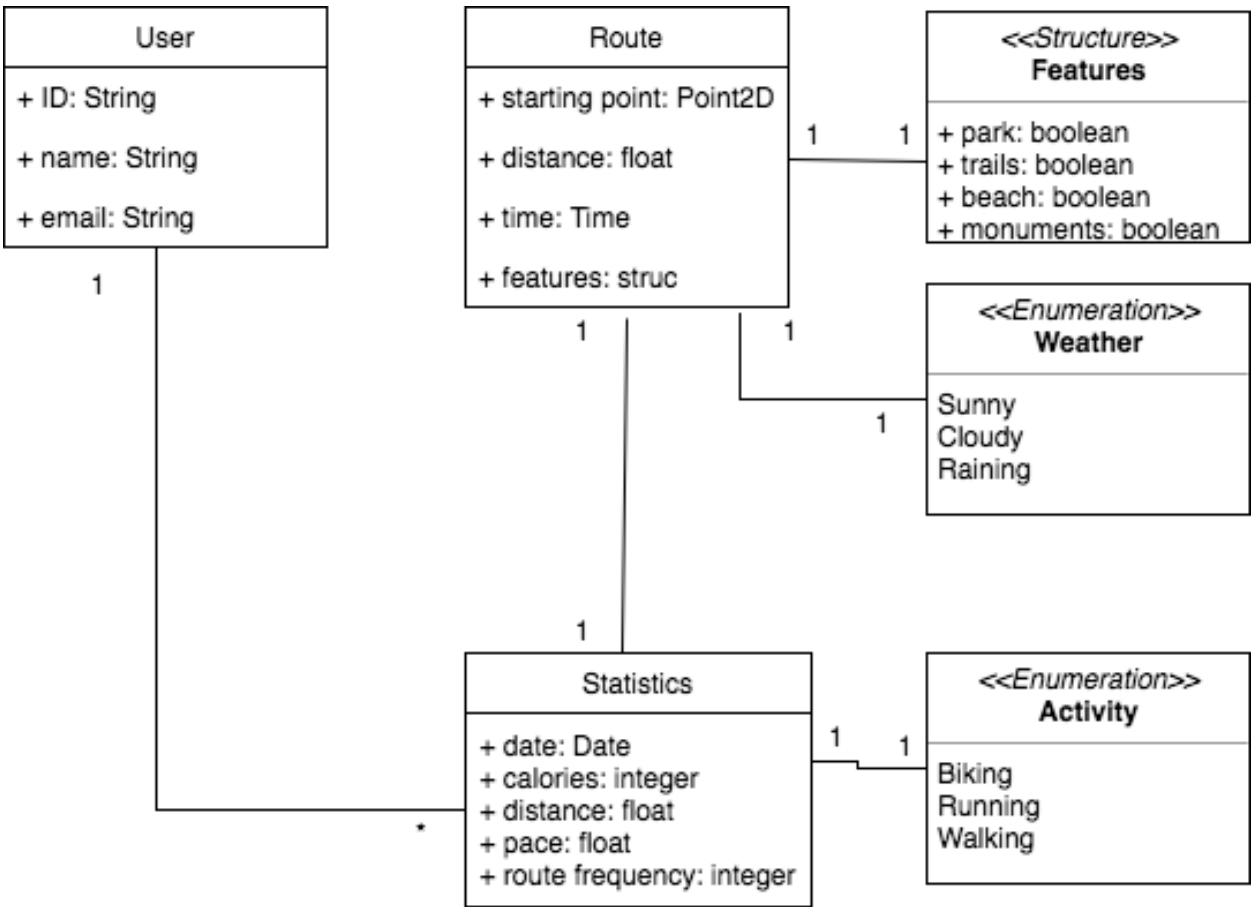
The UML diagram represents the order of events to occur upon opening the app. The user opens the app. Fills out the login credentials and submits as a HTTP request to the server for authentication. Server will send back to client a login success message. The client will create preferences for the route and send another request with this information. The server receives this and combines the data with Google Maps service and creates customizable route. The server sends back the route details to the client. Client displays map with directions. As the route finishes, the client records user metrics and sends it to the server. Server sends user metrics to the database for the statistics to be stored.

Figure 2: Sequence Diagram



2. Class design

Figure 3: Class Design Diagram



Description of Classes and their Interactions

Our classes design is separated into three main classes which are the User, Route, and Statistics. Within Route there is a struct and an enumeration and within statistics there is an enumeration

User:

- Represents the user that is using the mobile application
- This is the only hierarchy and the highest level the user can choose to be
- It is created when the user creates an account

Route:

- Represents the route generated by the user
- It is created when the user wants to create their workout
- It will contain information such as the workouts starting point, the total distance, and the time it took to complete the route
- It will contain the struct features as well

Statistics:

- Represents the information created from the previous routes the user has worked out with
- It is created when the user has completed their first workout
- Contains statistics such as date of workouts, calories burned, average pace, distance, and how often they ran that route.

Features (Structure):

- Represents the selective scenery options the users can choose from to customize their routes

- It will contain boolean variables such as whether or not they want to come across parks, trails, beaches, etc.
- Created when the user begins planning their route

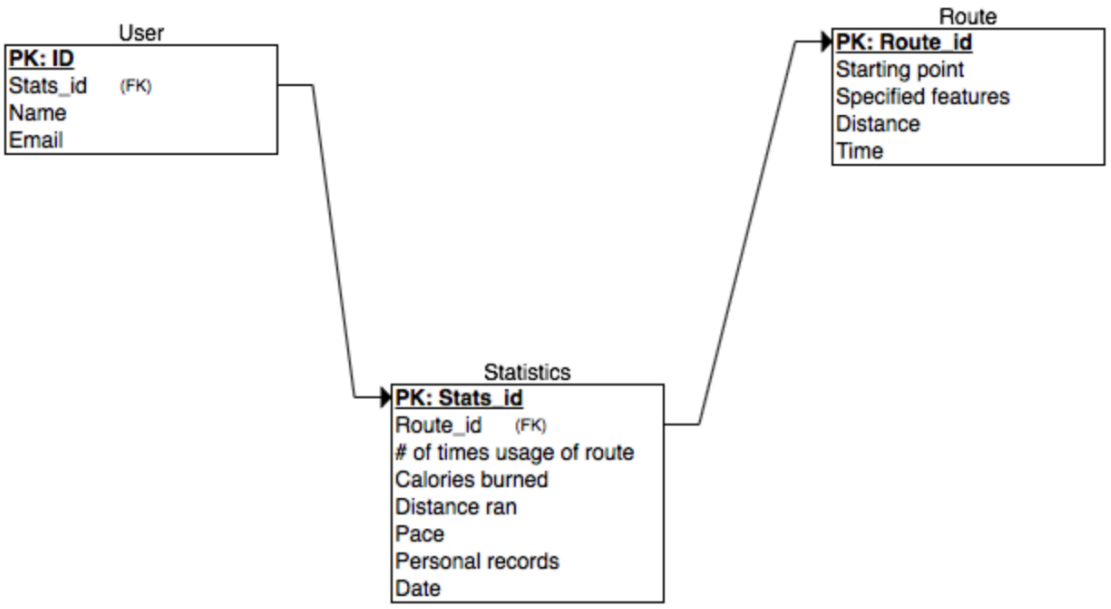
  Weather (Enumeration):
- Used to identify what the weather will be like during their workout
- Created when the user generates the customized route

Activity (Enumeration):

- Represents the different types of exercises the user has done
- It will contain the user's statistics and separate them based on the type of workout
- It is created when the user has completed their first workout

3. Database design

Figure 4: Database Design Diagram



Description of Database Table

1. User

    1. All users belong to this class since we only have one type of user which are people who use the app for workouts
    2. Contains information such as name, email, password

3. User ID, used to differentiate unique users in the database therefore our group picked user ID as the primary key

2. User's Statistics

1. This class contains all of the user's statistics based on the history of their previous workout routes
2. We can calculate the calories burned by using the formula. The variables for the formula can be obtain through the workout or the user's personal data
3. Our group decided to use the date of the workout as primary key

3. Routes

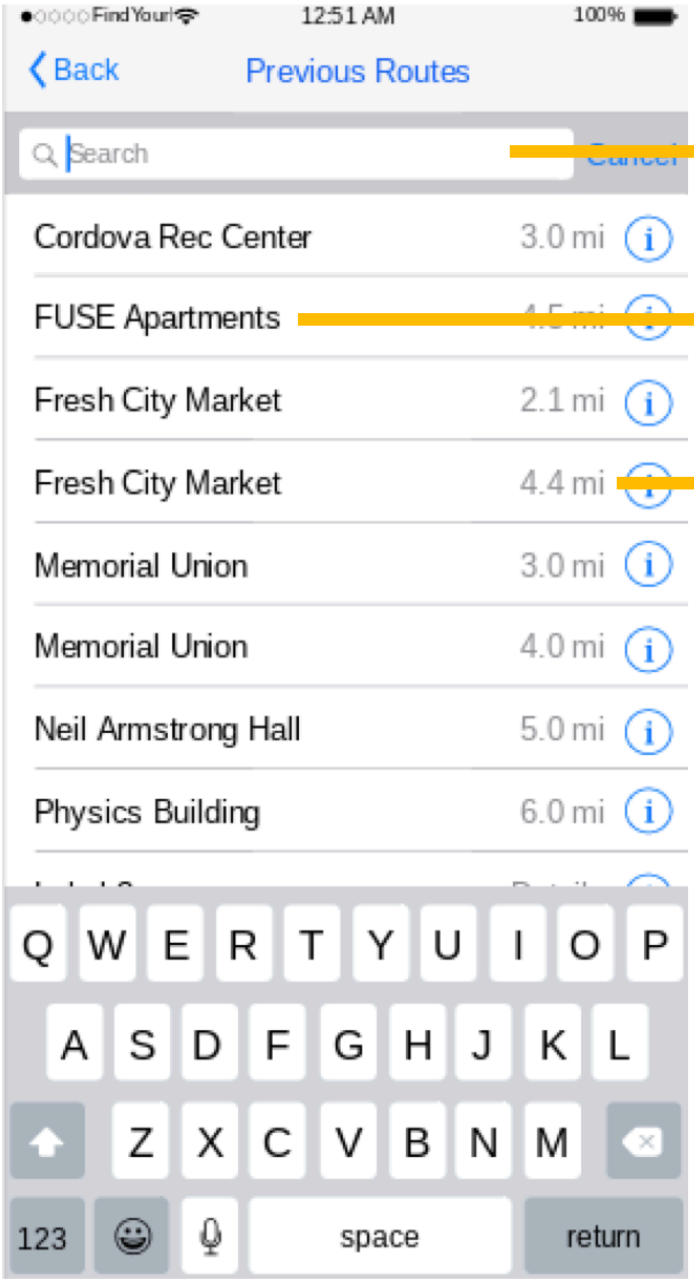
1. This class will contain the starting point, features, distance, and time
2. Our group picked the starting point and specified feature as the primary key of the route table

4. UI Mockups

Figure 5: Splash Screen

Figure 6: User Route Preferences

Figure 7: Previous Routes



search bar to search for previous routes

starting point of previous routes

total distance ran on a particular route

Figure 8: Route Display