SP-6 Blue – Time Mileage Location Tracker – RouteLink Tracker

Design Document

CS 4850 – Section 02 – Spring 2024 – January 19, 2024
Professor Sharon Perry

Armando Orti	Jacob Carrington	David Huu Nguyen
Andrew Millsap	Ethan Strickland	KSU Spirit

Table of Contents

1. Introduction

This document will outline the overall design of the mobile app and each of its main subsystems including the relationship between the app and database to the design of the mobile app. Some important terms that will be used throughout this document are API (Application Programming Interface) and Database (organized collection of information or data that is normally stored electronically in a computer system). By the end of the document, you will not only be able to understand what we are trying to accomplish but how we will go about achieving this end goal.

2. Design Considerations

2.1. Assumptions

Our app will be very basic to not only use but also easy to understand if we assume the following:

- All the devices that use the app will have location services enabled.
- All the devices that use the app will have the necessary GPS hardware for location tracking.
- All the devices that use the app will have IOS or android operating systems.
- All the devices that use the app will have good internet connection or service.

2.2. General Constraints

Some constraints that could possibly impact us during the development of this app are:

- Limited Time - developing the application and implementing monetization can take time it could lead to other challenges.
- No budget - Having no budget could potentially lead to limitations when it comes to APIs used, as many require payment.
- Limited libraries - Certain libraries require permission from the creator and could impede us from using it.

2.3. Goals and Guidelines

Our goal is to have users use our app as their new navigation app. This app will provide the user with trip information like how many miles they drove and how long they were out on the road. Our app will also be beneficial for people who work in delivery or driving services as they can track their miles for their taxes.

Some other goals and guidelines we are aiming for are

- The app must be accurate and provide the correct mileage for users to have reliable data for their taxes
- The app must be user friendly and reliable
- The app must be accessible to users with disabilities
- The app must have scalability to implement new features/updates

3. Architectural Strategies

React Native for Mobile Application Programming Framework
React Native will help create the user interface and functionalities of the application. React Native gives us the advantage of code reusability across Android and IOS devices. It will also give us access to a plethora of libraries and components.

Alternative Mobile Application Programming Framework
Flutter was a programming framework in consideration for the application's development. Flutter uses Google's Dart programming language, and the group is not familiar with the language. Also, some of the groups have already worked with React Native. Flutter is a great programming framework, but it has limited libraries and a small number of packages. On the other hand, react native has various libraries and packages.

Firebase for database management
Firebase will handle the data from the user's trip information. Firebase provides various backend services like real time database, authentication, and cloud storage. Firebase will aid us in having good synchronization of mileage and trip information data. Firebase will help us with our authentication because it implements phone number/email verification. We can send a code to the user to create an account, or to recover their account.

Alternative Database Management
MySQL was a database management that we were considering because it allows manages data and uses SQL, but MySQL is mostly used for application that require a vast majority of data like for example a banking app since it would need to store transactions, deposits, withdrawals, transfers, etc. Are app doesn't have a lot of information to be stored. In the end we chose Firebase because it was the ideal database for cross platform mobile app development.

Google Maps API
The Google Maps API will provide mapping and location services such as route planning and traffic updates. Integrating this API will allow the user to get directions to their destination. It will also help us retrieve the trip information and then report that info back to the user. Another reason for using Google Maps API is that it is the most used and recognized by users.

Alternative Map API
Mapbox was a map API we were considering. It is somewhat like google maps and has similar features. Mapbox isn't really known by users and having to learn how to work with it could lead to users not wanting to use our app.

4. System Architecture

When designing the Mileage Tracker app, the system architecture should be designed to manage the tracking and recording of the data efficiently. The system architecture will be made up of various key components responsible for specific functionalities.

User Interface

- The user interface will allow users to interact with the app.
- They will be able to login into the app.
- Be able to change the settings of their app like light or dark mode.
- Choose to and from where they want to go.

GPS Mileage Tracking

- This component should allow the user to use the app as a navigation system.
- Must provide the time it will take to reach the destination.
- Generate the path and provide directions for users to reach their destination.

Data Management

- Our data management component should be able to handle storing the miles that the user has driven.
- Must be able to save and store the users overall trip information.

Reporting

- This component will give the user mileage reports such as their trip history, total mileage and total time, and will provide them with information for their taxes.
- Should provide the user with weekly trip history.
- If a user is a premium user, they should be allowed to retrieve trip information from which time periods they want.

Monetization

- This component will let us make money from the app.
- Banner ads will be added to the app, but they will not bother the user like the pop-up video ads on other applications.
- There will also be a subscription feature users would need to pay to get other functions from the app.

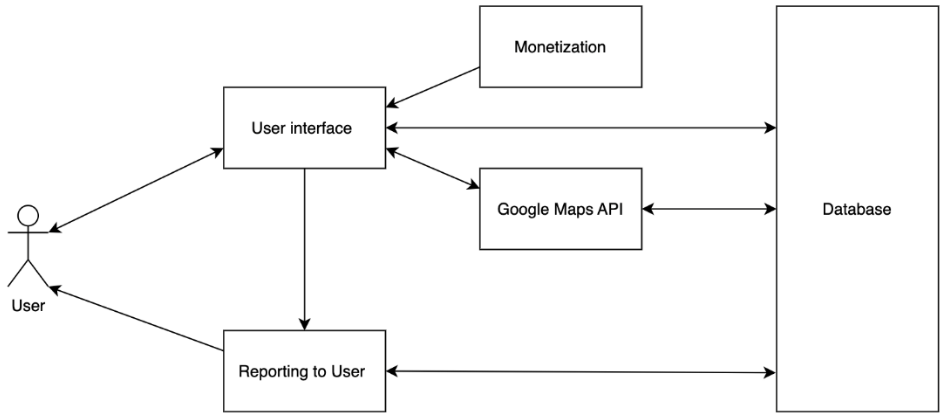
5. Policies and Tactics

Policy #1 (Code Language)

All code will be written and developed in the language "react native" so the app can be published on both Apple and Android stores with little to no extra work.

6. Detailed System Design

This diagram is a detailed design on the mobile app and how each subsection below is connected.



6.1. Database System Design

6.1.1. Classification

Subsystem

6.1.2. Definition

- The database is the subsystem used for storing and providing data to the mobile application.
- Any data that must be stored by the system long-term will be held in the database.

6.1.3. Responsibilities

- The database is responsible for housing and maintaining the data used throughout the system for both accounts and trip tracking.

6.1.4. Composition

The database will be a relational database with two primary tables, USERS and TRIPS.

The USERS table contains all information regarding the user and their account. This includes AccountID, Name, Username, Email, Password, Use2FA, and AccountStatus,

- AccountID is an arbitrary number associated with each account for table relation purposes.
- Name is the concatenation of the inputs for FirstName and LastName.
- Username is the chosen username associated with an account and can be used to sign in.
- Email is the email that the user creates their account with and can be used to sign in.
- Password is the hashed version of the password associated with the user's account.
- Use2FA is a boolean indicating whether a 2FA code should be sent to the user's email.
- AccountStatus indicates the status of an account (Standard, Premium, Admin).

The TRIPS table contains all information regarding the trips taken by an account. This includes AccountID, StartTime, EndTime, Distance, and Route.

- AccountID contains the ID of the account that started the trip, used as the key for the relation to the USERS table.
- StartTime is the time when the trip was started in UTC.
- EndTime is the time when the trip ended in UTC.
- Distance is the total distance covered in miles (can be converted in app when displaying).
- Route is the set of roads and turns taken throughout the trip to see the path.

6.1.5. Uses/Interactions

- This component will store all the data that will be needed during the use of the app.
- The accounts feature will require access to the database to allow a user to create an account and log in.
- The trip tracking feature will require the app to send data to the database to store that data.
- The trip reporting feature will require the database to send trip data for the corresponding account to the app to display.

6.1.6. Processing

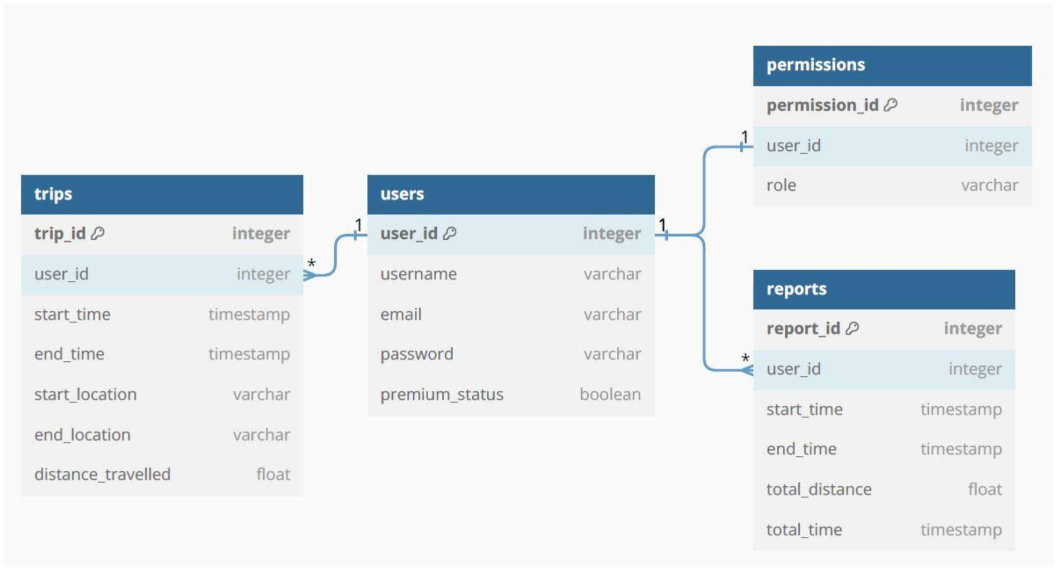
- The database will receive a query from the app and generate a report with the corresponding information.

6.1.7. Software Interface

- This component will export data in response to queries from the mobile application.

6.1.8. Design Diagrams

Below is the database diagram explaining the relations between.



trips	
trip_id	**integer**
user_id	integer
start_time	timestamp
end_time	timestamp
start_location	varchar
end_location	varchar
distance_travelled	float

users	
user_id	**integer**
username	varchar
email	varchar
password	varchar
premium_status	boolean

permissions	
permission_id	**integer**
user_id	integer
role	varchar

reports	
report_id	**integer**
user_id	integer
start_time	timestamp
end_time	timestamp
total_distance	float
total_time	timestamp

6.2. Mobile Application

6.2.1. Classification

- Subsystem

6.2.2. Definition

- The Mobile Application subsystem provides users an interface through which they can interact with all the features available.

6.2.3. Responsibilities

- The app will allow users to access all data and features they have permission for.
- The app will be responsible for allowing new users to fully utilize the app (meaning no external tutorials).

6.2.4. Composition

- The application will be composed of several different screens to utilize each feature.
- Each screen will have its functionality as well as buttons to navigate to different screens.

6.2.5. Uses/Interactions

- The Mobile Application subsystem uses data from the database to display reports and trip history.
- The subsystem will send login requests which will retrieve the hashes user password to check.
- The subsystem will send trip information to the database to be stored.

6.2.6. Processing

- The app will need to do basic data processing regarding trip information such as computing average speed.
- The app will employ a simple password hashing function to protect user information.

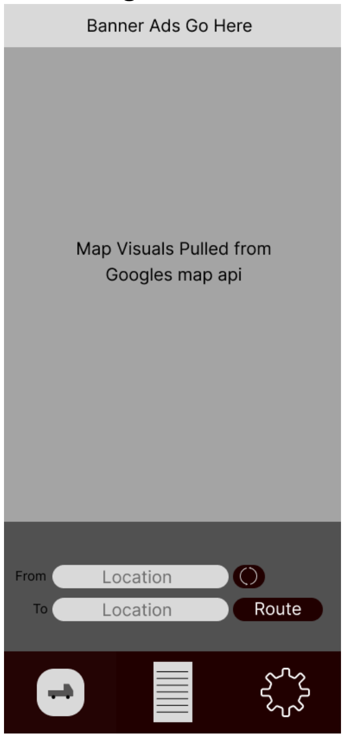
6.2.7. Software Interface

- The mobile application will interact with Google APIs to implement certain features.
- Google AdSense will be used to implement the advertising monetization aspect of the application.
- Google Maps API will be used to generate routes, save paths, and embed a dynamic map to display to the user while in the app.

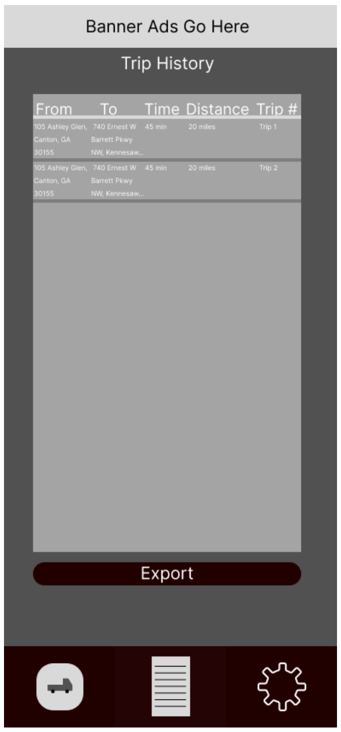
6.2.8. Design Diagrams

The pictures that are pulled from Figma are very early stage and are not the final design of the mobile app but hopefully they give you an idea on the direction that we are taking with the UI of this mobile app. The pictures are going to show you the base tier of the app with ads and then how it might look with the paid tier that contains no ads.

Home Page w/ ads

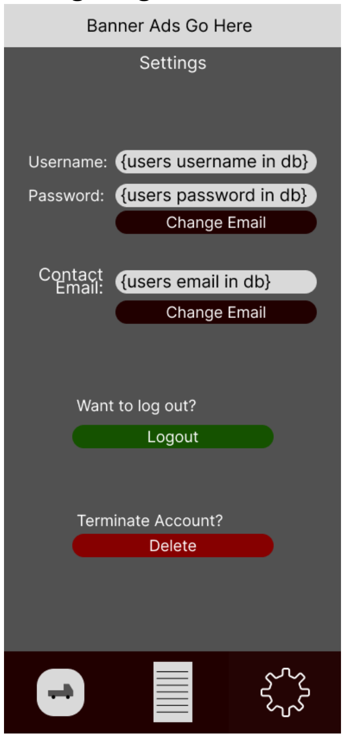


History Page w/ ads

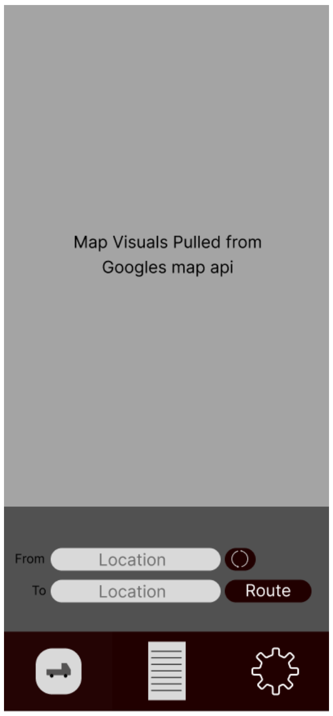


From	To	Time	Distance	Trip #
105 Ashley Glen, Canton, GA 30155	740 Ernest W Barrett Pkwy NW, Kennesaw...	45 min	20 miles	Trip 1
105 Ashley Glen, Canton, GA 30155	740 Ernest W Barrett Pkwy NW, Kennesaw...	45 min	20 miles	Trip 2

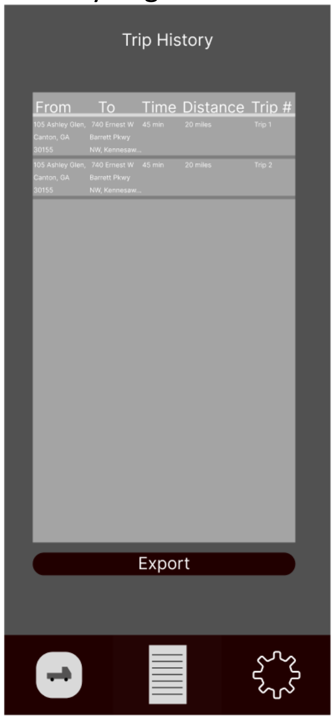
Setings Page w/


Banner Ads Go Here
Settings
Username: {users username in db}
Password: {users password in db}
Change Email
Contact Email: {users email in db}
Change Email
Want to log out?
Logout
Terminate Account?
Delete

Home Page no ads

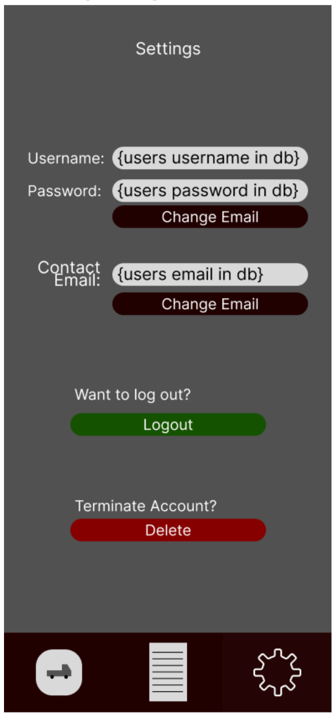


History Page no ads

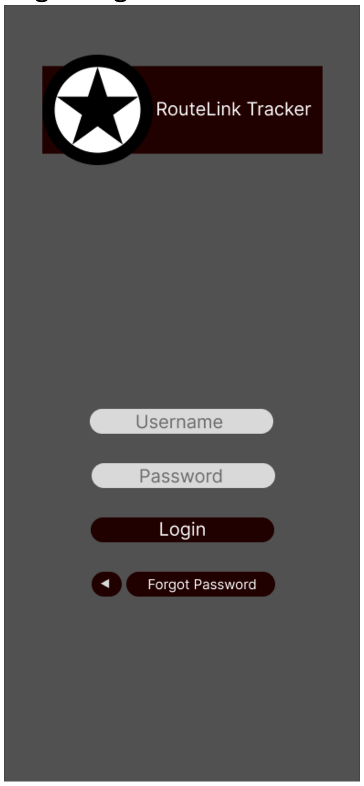


From	To	Time	Distance	Trip #
105 Ashley Glen, Canton, GA 30155	740 Ernest W Barrett Pkwy NW, Kennesaw...	45 min	20 miles	Trip 1
105 Ashley Glen, Canton, GA 30155	740 Ernest W Barrett Pkwy NW, Kennesaw...	45 min	20 miles	Trip 2

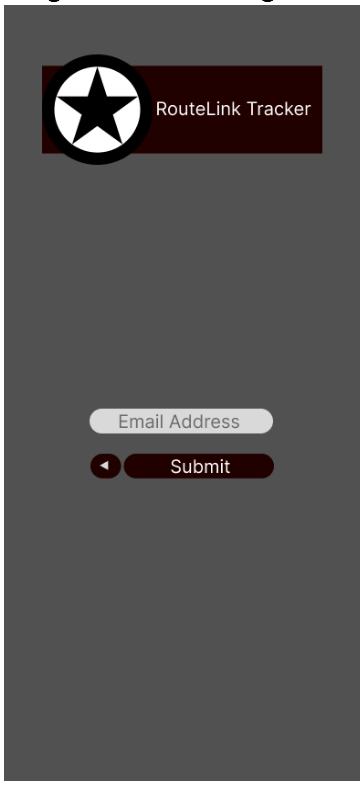
Settings Page no ads


Settings
Username: {users username in db}
Password: {users password in db}
Change Email
Contact Email: {users email in db}
Change Email
Want to log out?
Logout
Terminate Account?
Delete

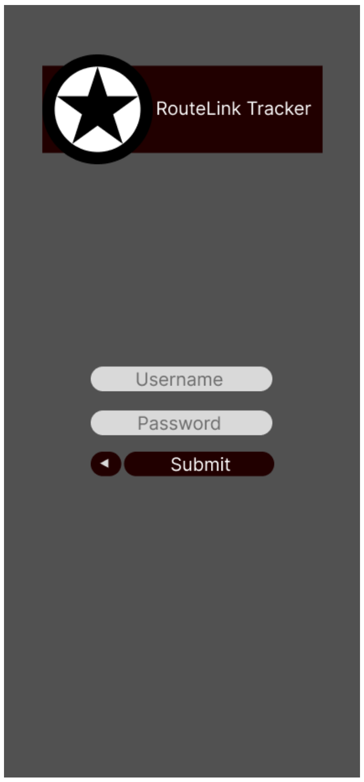
Login Page



Forgot Password Page



Create Account Page



7. Appendix

7.1. Appendix A – Figma App Skeleton

Figma Mockup for App Skeleton