Clemson University

Android Application Development for Navigation of Inductive Charging Roads

Team 8

Dylan McCoy, Andrew Gresham, Nathaniel Link, Dar Parsadoust

CPSC 4910

Dr. Herzog

30 April 2017

Executive Summary

This report provides an analysis and outline of the methodology being used to implement an android application. The app is used to collect feedback from the users of electric vehicles equipped with in-motion wireless charging technology. The app will have two main functions: The first, similar to google maps navigation app, is to provide the shortest route from origin to destination while following as many wireless charging lanes as needed. In order to test satisfiability of the route provided, the second function will be to provide a survey at the end of the trip.

The intended outcome of this project is a mobile app for Android that connects to the Google Directions API and allows users to provide feedback at the end of the trip. Android Studio, the Google Directions API, and Clemson phpmyadmin SQL servers are utilized in the development of this application. In order to achieve this goal, the team met with the client once a week to discuss the current progress of the project. On top of meeting with the client, the group met biweekly to work on the project design and implementation.

Project Description

1. *Client Background*
   Dr. Ilya Safro's graduate student, Hayato Ushijima-Mwesigwa, has developed an algorithm that calculates which roads are most optimal for wireless charging lanes anywhere in the world. Wireless charging roads are a huge initiative towards an energy friendly future, promoting the use of electric and hybrid cars. With Hayato's project complete, he has given us our project.
2. *Problem Statement*
   In order to test his algorithm, Dr. Ilya Safro needs users to be able to drive along the theoretical routes and provide feedback on whether or not they enjoyed them (not too far out of the way, reached destination without issues).
3. *Project Goals and Benefits*
   We have been given the problem to make sure that drivers of smart cars will be able to make use of all the possible charging roads while they are driving. To solve this problem, we plan to develop an app that will work similarly to google maps best route calculator. The add-on will perform a data table look-up to incorporate the electric roads as an option for drivers when picking their route. We are using Hayato's database as theoretical locations for the electric roads for the purpose of testing the app. We will also implement a post-trip survey to gauge the satisfaction of the electric drivers with the route that was calculated for them. With an increased use in these roads, there will be more incentive for others to buy electric vehicles, in hopes of not paying as much for gas. This could also lead to a decrease to global fuel emission.
4. *Development Methodology*
   To complete this task the group will meet twice a week to work on the implementation of the application. One of the group meetings is spent spent with Hayato to give updates and discuss the next steps. The group will have a pre--designated note taker to document our discussions and planning for the implementation, making sure we all have access to the information. We will end each meeting by planning what we will accomplish in the next meeting, and what should be completed before the meeting.
5. *Challenges*
   The primary challenges of this project include familiarizing ourselves with the Google Maps and Directions API, along with establishing an ODBC connection between the android application

and the Clemson phpmyadmin SQL server. Understanding the expected data output based on user input (such as the inductive charging roads the user took) will require implementing data tables with unique identifiers based on each trip. There will also be a data table for the charging lane locations and a survey asking whether that route was preferable to the initial Google Maps choice. The survey is one question at the end of the trip that allows Hayato to make alterations to his routing algorithm as need be.

Resources Needed

A large majority of the resources that we are using are accessible from our personal computers. We have the Google API, Android app software, the database for the roads, responses, and metadata. We will use a git repository to keep everyone up to date on code changes.

Requirements

1. The product needs to be an Android app
    a. Our clients, Dr. Safro and Hayato Ushijima-Mwesigwa, requested an Android application on the basis that Hayato uses an Android phone and so would be able to demo the application to his necessary counterparts in the office.
2. Must make use of the Google API
    a. Google has a set of APIs that are already used to implement Google Maps. Such examples would be Google Directions API and Google Directions API. Since these are already used to implement the official Google Maps application, they will be repurposed for this project.
3. Must have a post-trip survey to give feedback on route satisfaction
    a. The purpose of this data is to determine whether or not the algorithm Hayato and Dr. Safro created is actually viable from a traveler's point of view. This data will allow Hayato and Dr. Safro to determine if said algorithm actually needs to be remodeled in order to alleviate traveling time while maximizing battery charging capabilities.
4. Code and processes must be well documented
    a. There is a Github account with the code and documentation available so that Hayato, Dr. Safro, and future students working upon this program will be able to understand the material. This will enable future work to be completed with greater ease and allow the project to move forward at a greater rate.

Design

The design for this project consists of a navigational app, and a database. The database contains multiple tables, there are three tables.

1. The first table contains coordinates for all of the roads that are used for testing the application throughout this process.

id	Start_Lat	Start_Long	End_Lat	End_Long
1	40.7554	-73.9617	40.7552	-73.9619
2	40.7593	-73.9582	40.7581	-73.9592
3	40.7544	-73.9627	40.7529	-73.964
4	40.7552	-73.9619	40.7544	-73.9627
5	40.7517	-73.9652	40.7479	-73.9682
6	40.7529	-73.964	40.7517	-73.9652

2. The second table contains information about the driver's
    a. Location
    b. Destination
    c. Starting and ending battery charge
    d. The difference in time from the charging route from the original Google Maps preferred route.

id	vehicle_make	vehicle_model	vehicle_year	Start_Lat	Start_Long	End_Lat	End_Long	Start_Charge	End_Charge	Added_Time	Response
1	Chevrolet	Volt	2017	34.6834	82.8374	34.0522	118.244	89.11	65.89	27.43	1

3. The third table consists of survey responses allowing for an aggregation of data for future updates to the charging lane routing algorithm.

id	Trip_id	Start_Lat	Start_Long	End_Lat	End_Long
1	1	40.7554	-73.9617	40.7552	-73.9619
2	1	40.7593	-73.9582	40.7581	-73.9592

The navigational app is implemented in Android Studio with further specifications listed in the "Implementation" section. The first part of the design is to recreate the google directions API, addresses entered, making directions based on the user's current location, and directing the user towards it destination. This will be done using online tutorials to provide the jumping point into the main point of our mission. After that task is completed, the next phase will be to implement a survey that allows for a user to answer a question posed by, Hayato. The last phase of the app design is having the application enact two-way communication with the database. It needs to be able to send trip statistics and survey results and also allow for updating coordinates where the charging lanes are present (the roadways exist, but the charging lanes currently only exist hypothetically).

Implementation

The application was developed in two parts

1. Maps and Directions were implemented using Android Studio
2. User survey and establishing a connection to the database were implemented using Eclipse.

Maps and Directions were developed using Google Directions API, and everything was written in android studio, which is based in java. The Google Directions API will be the primary coding libraries where we will be referencing code for this application. We are making use of user tracking, global maps, and the large database of roads around the world that come with Google's libraries. We are using an SQL database to contain the coordinates of the electric roads, which will then have an algorithm that determines the length and locations of the charging roads. The app is built, run, and tested on our personal android devices, which run the latest OS Marshmallow. The locations of the charging roads are pulled from the database, and show relevant routes for the driver to take that will include the roads.

The survey requirement is fulfilled using the JOptionPane object, which prompts the user for a yes/no response to a question and returns an integer. This integer is then cast to a string that is appended to a text file. Connection to the MySQL database was attempted with jdbc (Java Database Connectivity), but permission issues prevented us from accessing it, so we could not store the user's response into the database.

Testing, Planning, and Results

Each feature of the application was tested before implementing any dependencies. The survey was changed from a JFrame object to a JOptionPane, which simplified testing to checking four buttons and their mappings, which were confirmed to be as listed in the following table.

Button	Integer
Yes	0
No	1
Cancel	2
X	-1

Issues occurred when attempting to test connection to the MySQL database, as permissions were denied from multiple access points (the McAdams lab computers, personal laptops on campus, and personal laptops connected to the Clemson VPN). Multiple tickets were opened with CCIT, but due to having dynamic IP's and no known accepted IP range, attempts to connect were unsuccessful. To circumvent the inability to connect the the database, a class was made that handled the JOptionPane responses and wrote them to a text file, translating the integer to its appropriate string response. This feature will be removed when permissions issues are cleared.

The application was more challenging than expected. We went through multiple tutorials that were tested on emulated android devices, along with actual devices. Our group ended up creating two map applications through the tutorials, each with different defining features. The first app contains address searching, with trip time and distance calculations. The other application implements user tracking, along with a simple touch screen use of selecting the coordinates. They were not able to access the database containing the coordinates, given our problems with accessing the databases. We were also not able to recreate all the features that are generally seen in a directions application, such as setting the user on the route, displaying the current direction, among other things.