

CS 4720 – S18 - Final Project Documentation

Device Name: Nexus 7 Platform: Android

Name: Dylan Nguyen Computing ID: dcn3en

Name: Sam Prescott Computing ID: sbp9as

App Name: Food Truck Finder

Project Description:

One of the difficulties with Food Trucks is that it is very hard to find/remember the location of a particular Food Truck. The Food Truck Finder app allows users to add markers onto a Google Map to designate a Food Truck location and be able to add the Food truck to the Map. The users will also be able to submit reviews with ratings and descriptions that will be able to be seen by other users.

Features:

- Users will be able to select a spot on the map to add a Food Truck
- Users will be able to add a Food Truck to the map and a Firebase database
- There are two options for adding a marker to the map. A long click will bring up a dialog to add a Food Truck at a specified location. Pressing the FloatingActionButton will also add a Food Truck, but at the location of the device.
- The Firebase database will store the Latitude and Longitude of the Food Truck as well as a unique Id key which ties the Food Truck to all of its other data members.
- The Food Truck data will also be stored in a SQLite database which reduces the need for multiple reads from the web service.

- The user can access the camera to take a picture of the food truck and store the picture for later.
- The user can also submit a review with a name, title, rating, and description
- The user can also send an email to a friend, sharing the Food Truck's longitude and latitude

Features Implemented

- 15pts GPS/Location Awareness
- 15pts Camera
- 15pts Build and consume your own web service using a third-party platform
- 20pts Data storage using Core Data (iOS) or SQLite (Android)
- 5pts Open shared activity / features

Wireframe

The design of the actual app has changed significantly from the wireframe based on time constraints and feasibility for the application. The main activity is a Map with markers that can be resized to fit more or less markers. Pressing the floating action button will allow a user to add. Selecting a marker will open an info menu with the name that when clicked will direct to a FoodDetail activity. The activity is separated into two tabs, no longer three tabs, which show an Overview and a Review Fragment. It also contains a back button to navigate back to the main map. The Review Fragment contains a list of reviews that can navigate to another screen that allows a user to see the rating and description of each review. Another button navigates to the AddReview activity which allows the user to add a review to the review list.

Testing Methodologies

- To test the GPS, we tried both methods of adding a Food Truck to make sure there would be no conflicts.
- We used the Firebase console to make sure that all of the Food Trucks were being added correctly
- We tested the Review component by generating several reviews with different ratings, descriptions, and Titles, to test how many can be stored in the database at a time.
- We tested the SQLite database by pulling data from the database several times to test if it was faster than Firebase

- We tested the Camera by taking pictures and selecting different stored pictures to make sure that the layout can successfully show the ImageView correctly

Usage

No special instructions needed.

Lessons Learned

- How to successfully integrate the Google Maps API with an Android App.
- How to store complex data structures locally and through a web services
- How to design custom layouts and manipulate xml files
- How to create a Tab Layout and create Fragments within a view
- How to add navigation buttons to a layout
- How to use a Scroll View with embedded components
- How to change the icons/styles/colors for an app
- How to use a Rating Bar and store that data.
- The difficulties of MVC app architecture.
- How convenient web services are for creating usable app activities.
- How to typecast data for intents/databases
- How to send intents through several activities
- The difficulties of adding accessibility/usability standards to an app design
- The importance of permissions in the context of privacy standards and app usability
- How picture data and camera location are stored in a phone and accessed through a mobile application
- How to send data to another shared activity and populate values in that activity (Gmail)
- How to use adapters and dynamically change views based on addition of reviews.