Winter Progress Report Capstone Project

Individual Report By: Charles Henninger
Sponsor: Nancy Hildebrandt

Instructor: D. Kevin McGrath
CS 462, Winter 2017, Oregon State University

Abstract

The Santiam wagon trail is historic trail located in the Willamette National Forest. The local ranger stations wish to have a mobile app that is capable of taking users on a tour of the wagon trail without needing a connection to the internet. The mobile application come in two forms: one developed for the Android mobile platform, and the other developed for the iOS mobile platform. While these two forms of the mobile application will be developed separately, they will be using the same methods of providing a tour to the user. The mobile application will render a map using a pre-downloaded map tile file and place waypoints onto the map that will be related to relevant information, in the form of videos and text files, about that area of the map. In order to achieve this functionality without internet access, we will rely on pre-downloaded content packs that will contain the map tiles, videos, text files, and waypoint information. These content packages will be created by staff of the local ranger stations, and uploaded via a website that will be developed along with the mobile app. Our team will divide these three larger sections of this project (the Android application, iOS application, and the Web Control panel/backend engineering) between the team members as follows: Android application development: Charles Henninger, Web Control panel/backend engineering: Duncan Millard, iOS application development: Jiawei Liu. This document is the individual report of progress for Winter term 2017 by group member Charles Henninger.

March 23, 2017

Current Progress

Group 43 has made decent progress in the completion of our solution for the Santiam Wagon road application. Currently we have functioning versions on both the Android and iOS platforms, and have a functioning server that can be queried by both versions of the application. Maps can be loaded dynamically based on the body of the response from server to application, and both UI developers (Jiawei Liu and myself) have a working grasp of the Mapbox library. Development of the web based control panel has yet to begin, due to the server only recently coming online.

While development has progressed a fair amount during the winter term, the main thing that held back the development progress was the lack of time some of the developers had to work on the project. Both Duncan Millard and myself have found very little time to work on the project due to school and non-academic responsibilities. Despite this, I believe our group has made acceptable, if only just, progress in the development of our solution.

Other more personal challenges that I found during the winter term development concerned getting our mapping service, Mapbox, working smoothly with the other facets of our application. One major problem was finding a way for Mapbox to run inside of an Android activity while also allowing that activity to perform other functions such as inter-application navigation. This was challenging mainly due to Mapbox limited ability to be supported on fragments. Mapbox is only compatible with the fragment class Support Fragment, which is mainly used as a compatibility class to use some features of fragments on older Android API's. I eventually created a solution by making our main Navigation activity, a Navigation Drawer class, be the only class that sets the layout views for the application. Any new activity simply passes the name of it's specific XML layout file to the Navigation Drawer class, and continues running it's intended purpose. The Navigation Drawer class then inflates that activity's XML layout inside a frameview located in the Drawer's XML intended for that specific purpose. Relevant code is shown below.

(The overridden setContentView method inside the Drawer Activity)

```
public class Drawer extends AppCompatActivity implements NavigationView.OnNavigationItemSelectedListener {


   protected DrawerLayout fullLayout;
   protected FrameLayout frameLayout;
   @Override
   public void setContentView(int layoutResID) {

       //this appends the desired activity layout to the Drawer activity layout

       fullLayout = (DrawerLayout) getLayoutInflater().inflate(R.layout.activity_drawer, null);
       frameLayout = (FrameLayout) fullLayout.findViewById(R.id.drawer_frame);
```

```java
        getLayoutInflater().inflate(layoutResID, frameLayout, true);
        super.setContentView(fullLayout);

        //The navigation bar everything from here on

        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
                this, drawer, toolbar, R.string.navigation_drawer_open, R.string.navigation_drawer_close);
        drawer.setDrawerListener(toggle);
        toggle.syncState();

        NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
        navigationView.setNavigationItemSelectedListener(this);
    }
```

(The XML file for the Drawer Activity. The layouts for other Activities will be inflated inside the FrameLayout located here)

```xml
<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.DrawerLayout xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="fill_parent"
   android:fitsSystemWindows="true"

   android:id="@+id/drawer_layout"
   tools:openDrawer="start">

   <include
       layout="@layout/app_bar_drawer"
       android:layout_width="match_parent"
       android:layout_height="match_parent" />

   <FrameLayout
       android:id="@+id/drawer_frame"
       android:layout_width="match_parent"
       android:layout_height="match_parent" />

   <android.support.design.widget.NavigationView
       android:id="@+id/nav_view"
       android:layout_width="wrap_content"
       android:layout_height="match_parent"
       android:layout_gravity="start"
       android:fitsSystemWindows="true"
       app:headerLayout="@layout/nav_header_drawer"
       app:menu="@menu/activity_drawer_drawer" />

</android.support.v4.widget.DrawerLayout>
```

Retrospective

The first three weeks of the winter term were spent mainly with each group member becoming situated in their individual school schedule, and meeting weekly to look at what we should focus on, both in the application development and for the documents due for the capstone class, in the weeks to come. From that time until the midterm progress report, and the alpha version of our application, the UI developers (Jiawei Liu and myself) worked on creating an alpha version of the application that could be navigated through, and would display a random intractable map through Mapbox. During this time, Duncan Millard was working with representatives through Oregon State to connect a NUC server into the Oregon State server and get a hole in Oregon State's firewall. After the midterm progress report was completed, we took two weeks to focus on our own classes while meeting once a week to remain on the same page about what we needed to focus on next. The last few weeks of the term were spent with all of us making sure that the communication between our applications and our server was working and stable.

Our group did a fantastic job of staying in communication over the course of the term. Even if we didn't meet in person over the course of a week, we still made sure to exchange messages to insure we were all on the same page.

Our group struggled with finding time to meet beyond a few hours per week, which meant that most development was done individually and was sometimes put off for later.

Our group has notice this issue, and has collectively decided to make sure to have at least one developing meeting per week next term where we help each other code and develop the next version of the application.

Team Assessment

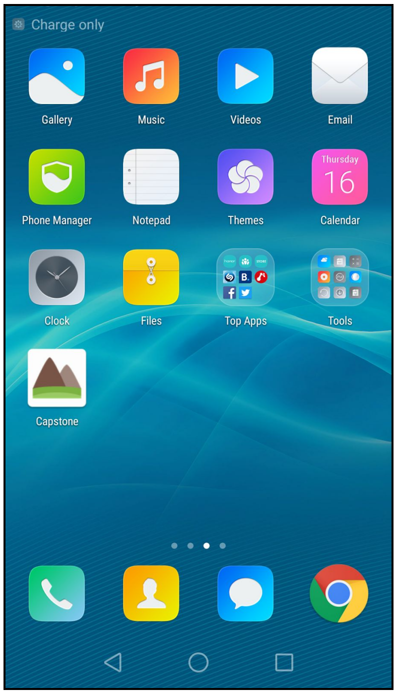
I have been more than impressed with how fast this team was able to begin work in a united and organized manner. We have yet to have any conflict on a personal level, and I believe that our individual differences actually work together in a way that strengthens the group as a whole.

Duncan Millard is an experienced engineer with a substantial amount of experience working with backend systems. This is extremely fortunate, as neither Jiawei or I have anything near to the experience that he has with servers, and would be much farther behind without him. Duncan’s experience in the industry, being an employee for Tektronix for the last five years, means that he understands some of the pitfalls that can occur during team development and helps us avoid them. Duncan tends to prepare for the worst case scenario almost on reflex, which is a nice pairing to my general attitude and keeps the group grounded in a space between cynicism and excess enthusiasm.

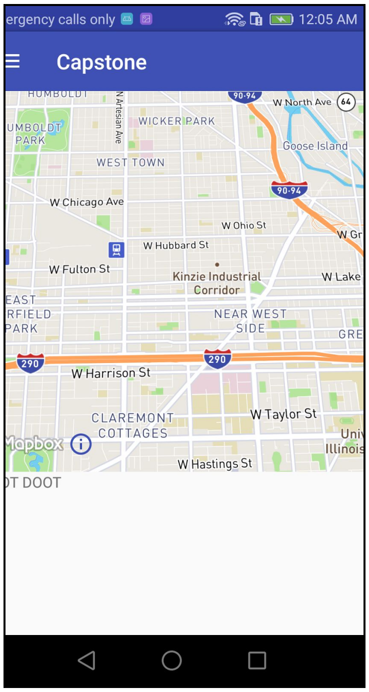
Jiawei Liu is an extremely driven engineer that has experience in front-end development. Despite most of his previous development experience being on the Android platform, Jiawei has managed to pick up the iOS platform with little to no noticeable lag in his development time. Quite the opposite in fact, Jiawei puts consistently more time on his development that either Duncan or I can manage. Jiawei’s exceptional work ethic keep our group moving forward and ensure that we never become stagnant in our development process.

I believe my experience with UI development, as well as my experience working in a team based development environment, has made me a good fit for this team. My skills are a good pairing to Jiawei and Duncan’s, and my attitude keeps the atmosphere in the group casual yet professional.

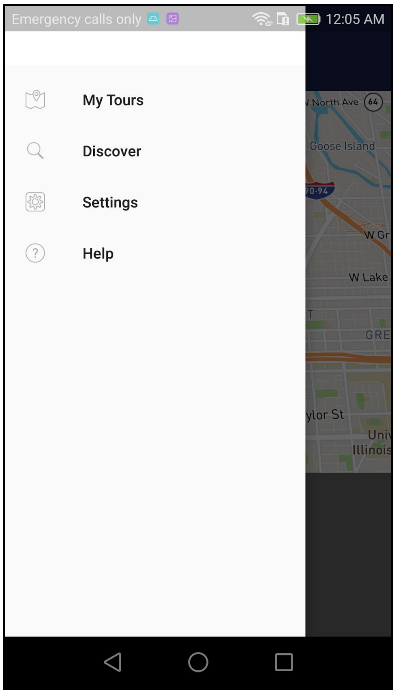
Images of the Android Application



(Application launch icon)



(Interactive Mapbox Map in app)



(Functioning Navigation Drawer)