Winter Progress Report
Capstone Project

Author: Jiawei Liu
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 outlines the possible technologies that will be used to address problems in each of these three major development sections, written by the team member heading each of the sections.

March 23, 2017

Contents

1 Jiawei Liu's Winter Progress Report

1.1 Team number and project name

I am Jiawei Liu, a member of group #43, and project name is Santiam Wagon Trail Mobile App.

1.2 iOS App Development

This project is going to develop iOS and Android map, then offer a web control panel to manage waypoints. In detail, this project will include a mobile application (iOS and Android), the Trail Companion, that will work with custom content packages downloaded in the app. A website will be included for administrators to upload content packages to a server where the application then request and download the packages. The mobile application will provide the user with an interactive map of an area, the user's location, and waypoints on within the area containing information about the park. This application will require no Internet access after downloading the content packages and application itself.

Currently, I am working on the iOS app for our project. I finished the basic UI and include all setting options in the page such as silent mode, app version, credits and help page. As screenshots show, the home page will be "My Tours" and display the map, waypoints and user's location. Then, there are four subpages when the user click menu button, which are My Tours, Discover, Setting and Help. Now the setting page and help page are ready for use. In addition, I added a button called "download", the user can click it and download content packages from our server. Overall, the app achieved beta level functionality.

During the iOS app development, the major problem is nobody familiar with iOS development, and OSU doesn't teach iOS development. And then, iOS using Swift to develop apps, so I need to learn a new language to develop the iOS app. Our group decides to use Android style slider menu as our app's UI, but this is not the default iOS UI, I will include extra style file to achieve it. In addition, Xcode doesn't like Android Studio, it only works on Mac OS system. For the map, since most map providers charge for using offline map, we also did some research for the map provider to avoid charging.

For solving above problems, I start to learn how to develop iOS app on the Internet. There are some good tutorials on YouTube and Apple website to help me begin the project. Since I can find some useful tutorials on the Internet, so it solved the problem that nobody familiar with iOS development. For learning swift, I found a website called "runoob.com", this website provides a good tutorial for swift, I learn the basic knowledge of swift programming during the winter break. For the UI, since Xcode didn't include this UI template, so I google how to design slide bar in iOS. Fortunately, I found a great tutorial about how to create slide menu on iOS, which is "ashishkakkad.com". then I use this tutorial as the template to develop our project. Since I am the only person that own a Mac device in our group, I decide to work on iOS app development. For the map provide, Since Mapbox can provide free offline map and customized map. We choose Mapbox as the map provider and default map framework. In addition, Mapbox has a detailed tutorial to teach me how to include the map in the iOS app and display waypoints.

Recalling the past ten weeks, our group decided to meet every Thursday for doing the capstone project in the first week. Proceeding from the second week, our group met with our TA, Xinze Guan on every Tuesday. After that, our Android UI is coming together on paper, and our iOS UI will be following the same design very closely to provide a unified experience. The initial database planning steps have also been taken during our first team meeting. On week 3, Duncan received an Intel NUC as a temp server to host our web control panel and mobile content API, and he spends a sizable amount of time configuring the server to our needs and requesting ports and a hostname for our group. I did a lot of research and finish the rough iOS app UI (Android style slider menu). On week 4, 5 and 6, I mainly worked on adding more functions to the iOS app, such as setting page, mapbox, storyboard and help page. On week 7, 8 and 9, since I have couples of midterm and final exam, therefore, I didn't walk a lot on iOS development, and Duncan had a family issue, so our group didn't move forward a lot. On week 10, I worked on how to interact with our server, so I added a button on iOS app to apply download content package function. That is my work for the past ten weeks.

In next term, I will work on optimizing the functionality and UI of iOS app. For example, the app can indicate the

direction of users facing, so users will get clear information from the map. In addition, since the API didn't finish yet, so read available regions function haven't finished yet, but that is a minor function, I can finish it in the next term.

According to thefreedictionary.com, the definition of experimental design is "research design that eliminates all factors that influence outcome except for the cause being studied (independent variable). All other factors are controlled by randomization, investigator-controlled manipulation of the independent variable, and control of the study situation by the investigator, including the use of control groups. [1]" In my opinion, experimental design is useful for our project, it can improve our design and provide a more comfortable user experience. For example, I can design an experiment to test user map preference in many ways. I can download couples of maps app such as Google Map, Here Map, and Open Street Map. We can observe users' habit and summary their preference. Then we can optimize our app UI and provide a better user experience.

For the interface design, we discussed the UI design as a group and our group member Charles did a great app UI paper prototypes, then uploaded to our group GitHub. We use this paper prototype as the reference to develop our project. I hope users are going to click the waypoint and understand how to visit the waypoint when the first time they use this app. In addition, Since I used the Android style slider menu, it is popular in Android but not in iOS, I hope users can click the menu button to check more features.

For the code and screenshots, I have seven screenshots, which are home page, slider menu, discover page, setting page, help page, Xcode and Storyboard. In addition, I also provide two code segments, which are slider menu and download content package. These code segments and screenshots are located at the bottom of this report.

1.3 Brief Evaluation

Jiawei Liu: The role as a UI designer which work on iOS UI and Functionality, Web Control Panel UI, and Remote API Interactions.
Charles Henninger: Android UI Design and Functionality, Map Rendering, Android Remote API Interactions.
Duncan Millard: Web Control Panel functionality, Inter-App framework.

Our team is interesting because nobody is the leader, but every group member will remind others the next important event. It seems that we manage this group together. Our team has a great communication and relationship. We can support each other if there are any problems. Therefore, there is no manager and technical expert in our group. Every person is completely equal.

For the contribution, I believe that each group member contributes equally. Because Charles is working on Android development, Duncan is working on server development and I am working on iOS development. Our project is separated equally by three. In addition, everyone can finish their development task on time, thus I see no reason to say anyone contributes lower than me.

Our group is using Facebook Messenger to connect with each other. Therefore, we can easily get the current development progress of group members. During the weekly group meeting, we will work together and show their code and app to other group members. Thus we can make sure every function works. I am comfortable with developing as a group because my group members are friendly and enthusiastic. We can help each other to solve the problem, and reduce the workload for everyone. Since our capstone can be divided into three equal parts, so each member can work on his own part. Overall, our group did a great job on the capstone project, it can effectively speed up the development of the progress, and I really enjoy to working as a group.

1.4 Retrospective Table

Positives	Deltas	Actions
-Team interactions are extremely easy to deal with and teamwork comes naturally to the group. -Team skill sets are varied and fit perfectly into the required platforms for this project	-Scheduling needs to be handled further in advance to prevent last minute completions of documents.	-One round of scheduling should be done early in the development term to create an awareness of what we need to complete and when.

References

[1] "thefreedictionary," *thefreedictionary*. [Online].
Available: `http://medical-dictionary.thefreedictionary.com/experimental+design` [Accessed: 16-Feb-2017].

1.5 Swift Code for Slider Menu

```
import UIKit

protocol SlideMenuDelegate {
    func slideMenuItemSelectedAtIndex(_ index : Int32)
}

class MenuViewController: UIViewController, UITableViewDataSource, UITableViewDelegate {

    @IBOutlet var tblMenuOptions : UITableView!

    @IBOutlet var btnCloseMenuOverlay : UIButton!

    var arrayMenuOptions = [Dictionary<String,String>]()
    var btnMenu : UIButton!
    var delegate : SlideMenuDelegate?

    override func viewDidLoad() {
        super.viewDidLoad()
        tblMenuOptions.tableFooterView = UIView()
        // Do any additional setup after loading the view.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    override func viewWillAppear(_ animated: Bool) {
        super.viewWillAppear(animated)
        updateArrayMenuOptions()
    }

    func updateArrayMenuOptions(){
        arrayMenuOptions.append(["title":"My Tours", "icon":"HomeIcon"])
        arrayMenuOptions.append(["title":"Discover", "icon":"PlayIcon"])
        arrayMenuOptions.append(["title":"Setting", "icon":"SettingIcon"])
        arrayMenuOptions.append(["title":"Help", "icon":"HelpIcon"])

        tblMenuOptions.reloadData()
    }

    @IBAction func onCloseMenuClick(_ button:UIButton!){
        btnMenu.tag = 0

        if (self.delegate != nil) {
            var index = Int32(button.tag)
            if(button == self.btnCloseMenuOverlay){
                index = -1
            }
            delegate?.slideMenuItemSelectedAtIndex(index)
        }

        UIView.animate(withDuration: 0.3, animations: { () -> Void in
            self.view.frame = CGRect(x: -UIScreen.main.bounds.size.width, y: 0, width: UIScreen.main.bounds
            self.view.layoutIfNeeded()
```

```
            self.view.backgroundColor = UIColor.clear
            }, completion: { (finished) -> Void in
                self.view.removeFromSuperview()
                self.removeFromParentViewController()
        })
    }

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
        let cell : UITableViewCell = tableView.dequeueReusableCell(withIdentifier: "cellMenu")!

        cell.selectionStyle = UITableViewCellSelectionStyle.none
        cell.layoutMargins = UIEdgeInsets.zero
        cell.preservesSuperviewLayoutMargins = false
        cell.backgroundColor = UIColor.clear

        let lblTitle : UILabel = cell.contentView.viewWithTag(101) as! UILabel
        let imgIcon : UIImageView = cell.contentView.viewWithTag(100) as! UIImageView

        imgIcon.image = UIImage(named: arrayMenuOptions[indexPath.row]["icon"]!)
        lblTitle.text = arrayMenuOptions[indexPath.row]["title"]!

        return cell
    }

    func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
        let btn = UIButton(type: UIButtonType.custom)
        btn.tag = indexPath.row
        self.onCloseMenuClick(btn)
    }

    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
        return arrayMenuOptions.count
    }

    func numberOfSections(in tableView: UITableView) -> Int {
        return 1;
    }
}
```

1.6 Swift Code for Download Content Package

```
import UIKit

class PlayVC: BaseViewController, URLSessionDownloadDelegate, UIDocumentInteractionControllerDelegate {

    var downloadTask: URLSessionDownloadTask!
    var backgroundSession: URLSession!

    @IBOutlet weak var MY_download: UIButton!

    @IBAction func MY_download_btn(_ sender: AnyObject) {
        let url = URL(string: "http://fw-santiam.eecs.oregonstate.edu:3000/testJson")!
        downloadTask = backgroundSession.downloadTask(with: url)
        downloadTask.resume()
```

```
}
@IBAction func pause(_ sender: AnyObject) {
    if downloadTask != nil{
        downloadTask.suspend()
    }
}
@IBAction func resume(_ sender: AnyObject) {
    if downloadTask != nil{
        downloadTask.resume()
    }
}
@IBAction func cancel(_ sender: AnyObject) {
    if downloadTask != nil{
        downloadTask.cancel()
    }
}
@IBOutlet var progressView: UIProgressView!

override func viewDidLoad() {
    super.viewDidLoad()
    addSlideMenuButton()
    let backgroundSessionConfiguration = URLSessionConfiguration.background(withIdentifier: "background
    backgroundSession = Foundation.URLSession(configuration: backgroundSessionConfiguration, delegate:
    progressView.setProgress(0.0, animated: false)
}
override func didReceiveMemoryWarning() {
    super.didReceiveMemoryWarning()
    // Dispose of any resources that can be recreated.
}


func showFileWithPath(path: String){
    let isFileFound:Bool? = FileManager.default.fileExists(atPath: path)
    if isFileFound == true{
        let viewer = UIDocumentInteractionController(url: URL(fileURLWithPath: path))
        viewer.delegate = self
        viewer.presentPreview(animated: true)
    }
}

//MARK: URLSessionDownloadDelegate
// 1
func urlSession(_ session: URLSession,
                downloadTask: URLSessionDownloadTask,
                didFinishDownloadingTo location: URL){

    let path = NSSearchPathForDirectoriesInDomains(FileManager.SearchPathDirectory.documentDirectory, F
    let documentDirectoryPath:String = path[0]
    let fileManager = FileManager()
    let destinationURLForFile = URL(fileURLWithPath: documentDirectoryPath.appendingFormat("/Content.Js

    if fileManager.fileExists(atPath: destinationURLForFile.path){
        showFileWithPath(path: destinationURLForFile.path)
    }
    else{
        do {
            try fileManager.moveItem(at: location, to: destinationURLForFile)
```

```
                // show file
                showFileWithPath(path: destinationURLForFile.path)
            }catch{
                print("An error occurred while moving file to destination url")
            }
        }
    }
    // 2
    func urlSession(_ session: URLSession,
                    downloadTask: URLSessionDownloadTask,
                    didWriteData bytesWritten: Int64,
                    totalBytesWritten: Int64,
                    totalBytesExpectedToWrite: Int64){
        progressView.setProgress(Float(totalBytesWritten)/Float(totalBytesExpectedToWrite), animated: true)
    }

    //MARK: URLSessionTaskDelegate
    func urlSession(_ session: URLSession,
                    task: URLSessionTask,
                    didCompleteWithError error: Error?){
        downloadTask = nil
        //progressView.setProgress(0.0, animated: true)
        if (error != nil) {
            print(error!.localizedDescription)
        }else{
            print("The task finished transferring data successfully")
        }
    }

    //MARK: UIDocumentInteractionControllerDelegate
    func documentInteractionControllerViewControllerForPreview(_ controller: UIDocumentInteractionControlle
    {
        return self
    }
}
```

1.7 Screenshots

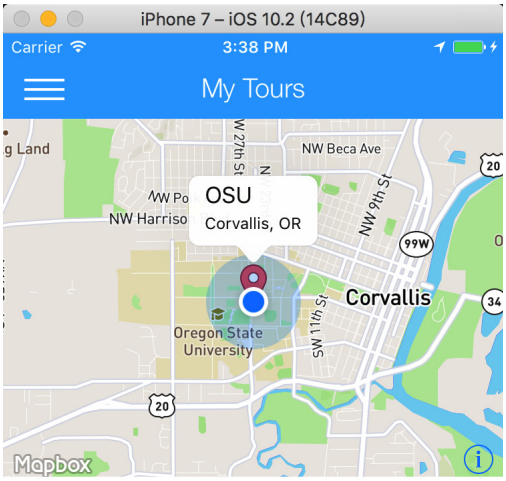

Figure 1: My Tours

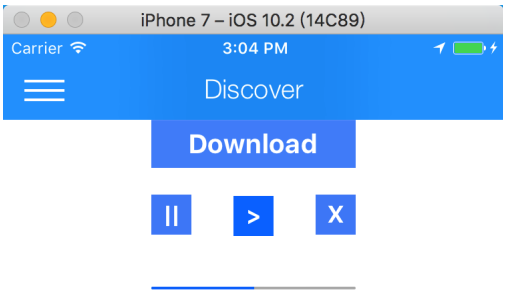

Figure 2: Discover Page

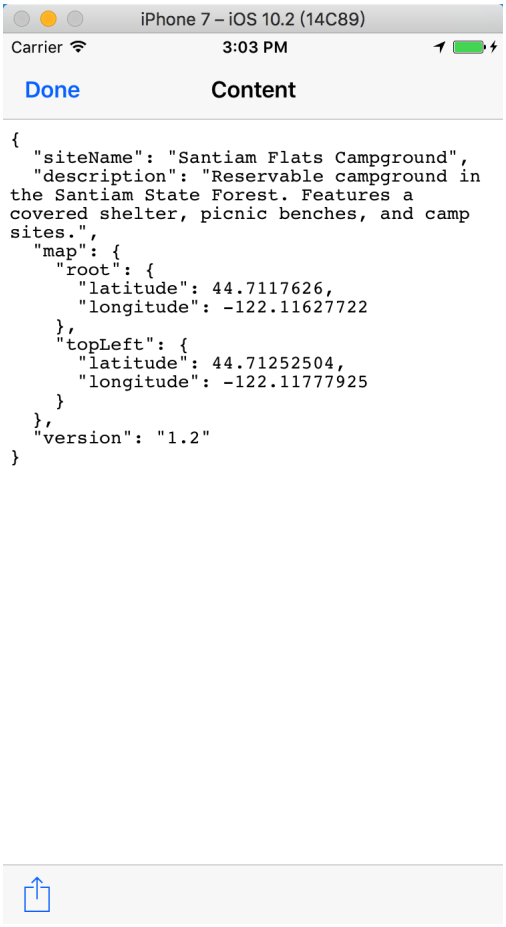

Figure 3: Open the Example File from Server

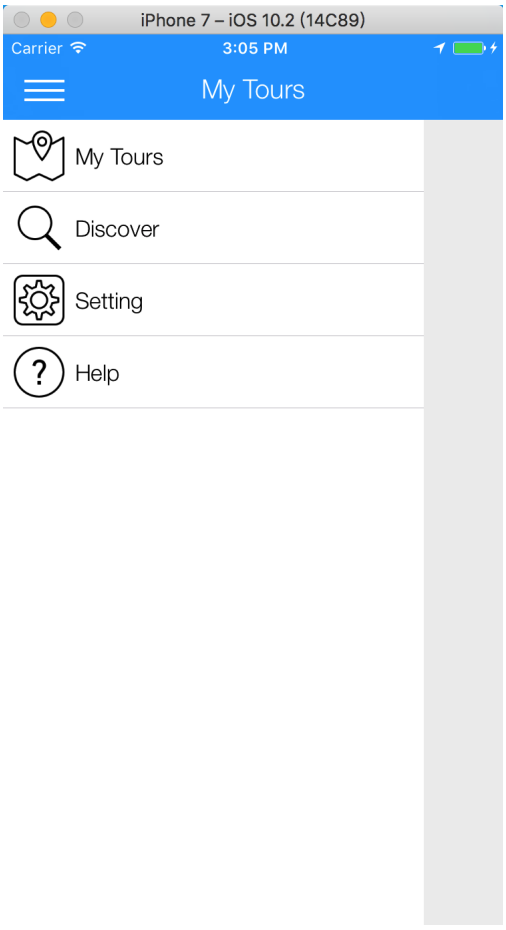

Figure 4: Slider Menu

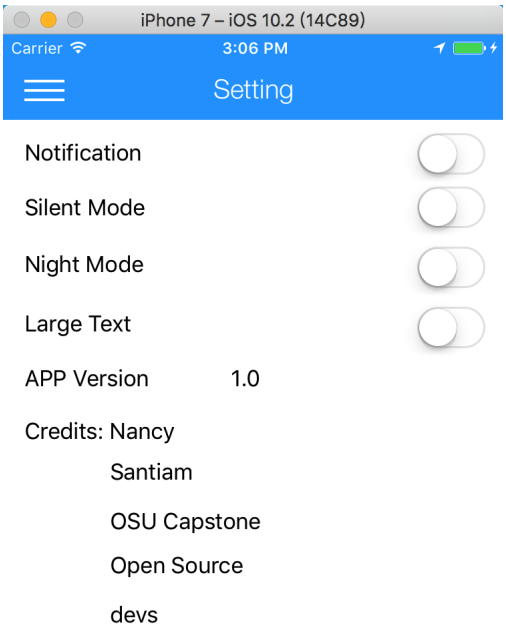

Figure 5: Setting Page

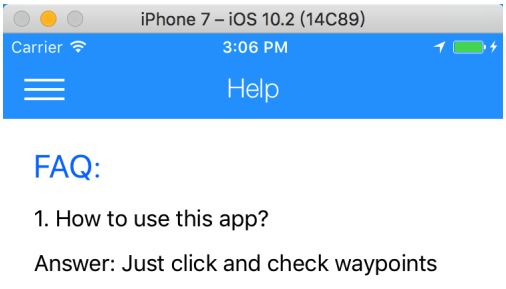

Figure 6: Help Page

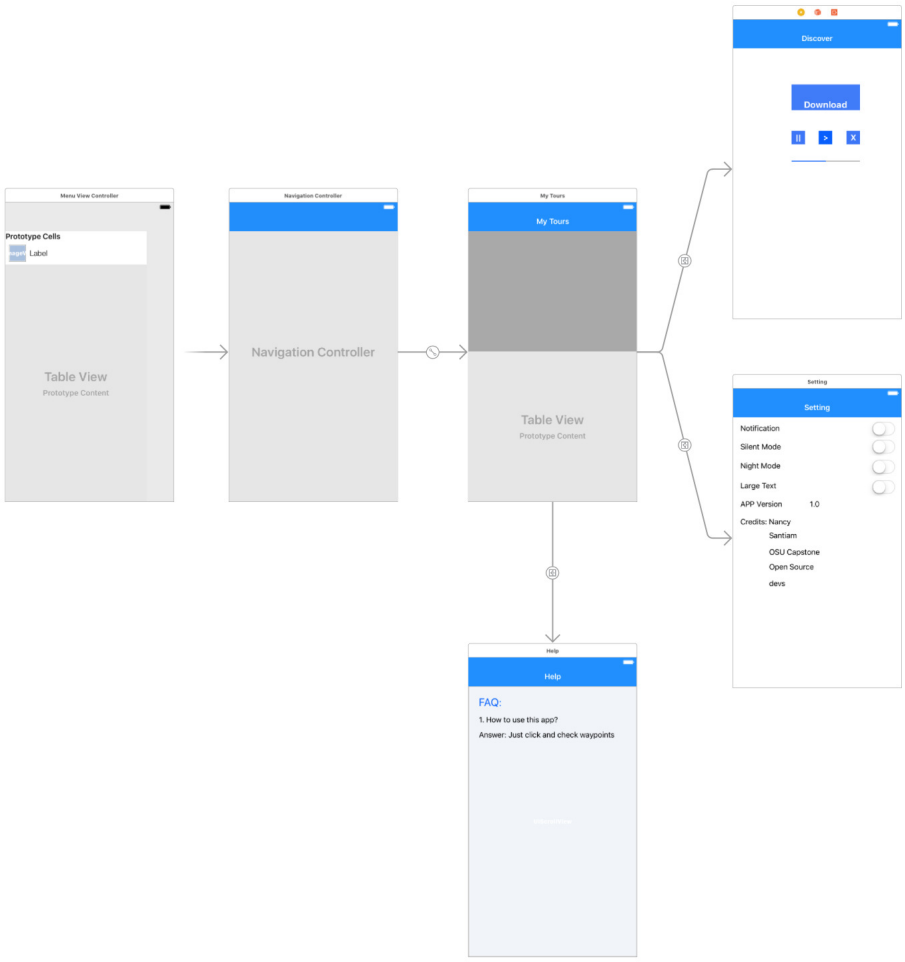

Figure 7: iOS App Storyboard