TripTracker Final Report

Karina Rodriguez / Advised by Ruzica Piskac / May 2nd, 2019

ABSTRACT

For my CPSC 490 final project, I wanted to expand on all of the skills I’ve learned through my Computer Science coursework and internships and apply them to iOS development. I’d previously had very minimal exposure to the world of mobile development so I was hoping to learn new coding languages, new design paradigms and generally become more familiar with the iOS application development cycle. Apps have become an integral part of many of our daily lives and therefore I think mobile application development is a very important skill for any aspiring software engineer to have. In order to tangibly assess my progress as I developed this new skill set, my final goal was to create a fully functioning mobile app. I ultimately decided to build my app using Swift, and not Objective-C, because it is a language that was developed by Apple specifically for the purpose of iOS development. It has thorough documentation, a vibrant online community and many online tutorials available, which were all very useful as I taught myself this new skill. The application I built, TripTracker, does exactly what it sounds like it does, it tracks a user’s trips. I implemented a mobile variation of a travel journal. As a frequent traveller myself, I am always looking for ways to document my trips, whether that’s through photographs, physical journals, social media or just jotting down notes on my phone. The goal of this application was to get all of this information in one place, allowing a user to track all of the places they’ve been, browse through their photos, reminisce about their favorite sites and restaurants, jot down notes and more all within the application.
METHOD

The first step in learning iOS development was to begin familiarizing myself with Swift and becoming more comfortable using the XCode environment. I read through the Swift Language Guide docs in order to familiarize myself with the syntax and played around with XCode playgrounds in order to get more comfortable in the environment. Before beginning to work on smaller projects I wanted to make sure that I was comfortable reading and writing the fundamentals of Swift (like optionals, guard statements, loops, class definitions, variables, structs, switch statements etc).

I continued by going through the Stanford course CS193P: iPhone Application Development. This was an incredibly useful starting point and I can’t recommend this course enough. While the Swift docs were useful for learning basic syntax, this course really taught me the fundamentals of iOS development starting with the structure of file directories and the basics of the Model-View-Controller paradigm. The course also taught me the ins and out of XCode. Using XCode was originally a bit overwhelming as there were so many toggles and sub-views. It was information overload and navigating it originally wasn’t super intuitive. The course was very helpful in breaking down the features of XCode and helped me better understand things like the Interface Builder, objects and their attributes, constraints and responsive layouting, and so much more. Another great aspect of the course was that it provided semi-guided tutorials for building relatively simple applications, which made the process of building my first app much less daunting. My first starter app was a Calculator – it was very useful for learning how to use auto layouting and constraints in the UI and for helping me understand how exactly the UI and code work together.
Figures 1a and 1b: My first iOS app! With the help of the Stanford iPhone development course I built a fully functioning Calculator app that responded to device orientation.

Next I went through the Apple Start Developing iOS Apps in Swift guide which really built on the knowledge I had acquired in the previous tutorial. With this guide I programmatically built a much more complex meal app. Through this guide I learned how to create my own custom data models and custom controls, how to use UI Views and Table Views, how to use segues in order to navigate through different scenes, how to allow users to add new data or edit existing data, how to persist data between app sessions, and how to use the XCode unit testing framework. The guide was extremely thorough and served as a great jumping off point for beginning development on my travel app.

By the end of my introductory research and self guided study I felt very comfortable using Swift and navigating Xcode. I was excited to build on my newfound knowledge and start work on my trip tracking app.
APP OVERVIEW

Specifications

In my application I implemented the following functionality:

- Signing Up/ Logging In
- Adding Trips
- Deleting Trips
- Viewing Trips in a table view
- Viewing Trips as an interactive map
- Searching Trips list
- Adding corresponding Trip Details (journal entries, photo uploads, location reviews)
- Editing Trip Details
- Deleting Trip Details

Data Models

- **Trip**
  - Location
  - Latitude
  - Longitude
  - Start Date
  - End Date
  - Trip Description
  - Trip ID
- **Trip Details** - Each trip has an associated array of trip details which can take the following forms.
  - **Journal Detail**
    - Title
    - Date
    - Entry
  - **Photo Upload**
    - Image
    - Date
    - Caption
Frameworks and SDKs

For my app I leveraged various different frameworks and SDK’s in order to improve app performance.

Firebase Authentication SDK

- The Firebase Authentication SDK was very useful for implementing the Sign Up/Log In flow, and for providing unique user IDs.

Google Maps SDK

- I used the Google Maps SDK in order to present the map view portion of my app and the corresponding pins.

Google Places SDK

- I used the Google Places SDK in conjunction with Google Maps, in order to allow a user to search for a location and then use the corresponding information to populate the map.

Chameleon Framework

- The Chameleon Framework was a very powerful tool for programmatically styling my app without having to go through and update every individual scene.

APP WALKTHROUGH

Opening the app for the first time leads users to the Sign Up/Log In Flow (Figure 2a). For this I used the Firebase Authentication SDK in order to securely store user emails and passwords and to help with performing the sign up and log in actions. When a user logs in it persists across app usage so the user won’t see these screens again until they log out, instead they get taken directly to the next view.
The first screen after logging in is the Trip Map View (Figure 3a). This uses the Google Maps SDK and there is a pin at each latitude and longitude location of a trip. When you click on a pin it displays an information box with the Trip location name displayed (Figure 3b), and clicking on that info box will take you to a screen with that Trip's details.

The navigation bar on this screen contains the Log Out button that returns users to the the Sign Up/ Log In flow above and it also contains a List View button which toggles the view and loads the next scene.
Figures 3a and 3b: The entry point to the app. A Google Maps view with a pin at each Trip Location, clicking on pins displays a info bar, and click that info bar navigates to Trip details.

After clicking the List View button the user gets taken to a screen that loads the same trips in a table format. For each trip you can see the location name, the corresponding date range and a brief trip description (Figure 4a). Upon pulling down on the table there is a search bar that allows users to search their trips based on location name (Figure 4b). The navigation bar on this page has a Map View button that toggles back to the previous screen (Figure 3a) and a plus button that allows a user to add another trip.
Figures 4a and 4b: Table view format for trips with relevant information. The page has a search bar that allows for search on trip name.

Clicking on the plus button on the List View screen (Figure 4a) navigates to the New Trip screen (Figure 5a). From here the user can add a new Trip object, this includes the trip location, start and end dates and a description. Upon saving, this new trip object will get added to the list view screen and a pin will be added to the map view.

When clicking the location text box the app uses the Google Places SDK to allow the user to search for a specific City or Country without having to type in the entire string manually (Figure 5b). This is useful as well because Google Places provides relevant information for each place, so when a user selects a location the
TripViewController stores the relevant latitude and longitude coordinates, which allows a pin to be added at the proper location on a map.

When clicking the date text fields the app loads a date picker and toolbar instead of a traditional keyboard.

From the New Trip screen the user can either cancel or save the their current addition. The save button is unclickable until all fields are filled in as they're all required. Clicking either button returns the user back to the List View screen.

![Figures 5a, 5b, and 5c: The add new trip screen with different interactions for different text inputs.](image)

Clicking through on any Trip will navigate the user to its corresponding trip detail view. This occurs when clicking the info box on the map view in 3b or clicking any trip cell in the list view in Figure 4a. Figure 6a shows the trip details for my New Haven trip. There are currently 3 types of details a user can engage with: a journal entry, a photo upload or a location review. For any trip, a user can add various of these detail entries in order to document their experiences.
The navigation bar on the screen allows a user to return back to their trips (either the list view or map view depending on how they navigated here) and it also allows a user to add a new trip detail.

![Image](image1.png)

Figures 6a and 6b: Trips details for my New Haven trip displaying a journal entry, a photo upload and a restaurant review. As well as the menu for adding a new detail.

Each trip detail type (journal entry, photo upload and review) has its own view as they each require different things. From 6b you can add a new detail of each type. If you click any existing detail from 6a you can expand to view more information and you can also edit. Figures 7a and 7b show the photo and upload views for adding a new detail, figure 7c shows the view when you’re editing a pre existing detail.
On top of editing trip details, the app allows users to delete existing trips or trip details by swiping on the relevant table cell. Upon attempted to delete the user receives an alert and must confirm the action, as once something is deleted it can no longer be retrieved.
Discussion

Completing this project was extremely fulfilling as I feel like I’ve gained a useful skill set and I’ve produced an app that solves an existing problem that I had. It wasn’t always an easy process, there were many days spent tracking down Google Maps bugs and figuring out how to persist custom classes was a nightmare, but overall I am extremely happy with my progress and results.
That said, there is definitely more that I can do to continue iterating and on improving on this application, and I will definitely continue to do so in my free time. A feature that I wanted to add but ultimately decided was outside of the scope of this project was having a social component within the app, allowing you to view and share trip itineraries and recommendations with friends. Other additions that I can foresee are adding more options for trip details, like video or audio.

Another aspect of my app that could use improvement is the unit test coverage. I added unit tests at the end, instead of practicing test driven development, and therefore there isn’t as much coverage as I’d like. When working on future applications I will definitely test as I go along in order to ensure functionality and track down any bugs or edge cases.

ACKNOWLEDGEMENTS

I’d like to thank my advisor, Ruzica Piskac, for helping guide me throughout this process.

I’d like to thank my suitemate, Julia, for her neverending support and encouragement.

I’d like to thank everyone who has ever written an iOS tutorial or answered a Stack Overflow question. I wouldn’t be here without you.

And lastly, I'd like to thank House of Naan for keeping me fed in the late hours of the night.