**Ryan Carson**

**Dyson Fraser**

**Tyrel Froese**

**COMP 4910: Computing Science Project**

**Project Report**

Table of Contents

[Abstract 4](#_Toc38445941)

[Requirements 6](#_Toc38445942)

[Initial Requirements 6](#_Toc38445943)

[Week 3 6](#_Toc38445944)

[Week 4 6](#_Toc38445945)

[Week 5 7](#_Toc38445946)

[Week 6 7](#_Toc38445947)

[Week 10 7](#_Toc38445948)

[Analysis 8](#_Toc38445949)

[Initial Analysis 8](#_Toc38445950)

[Week 3 8](#_Toc38445951)

[Week 4 9](#_Toc38445952)

[Week 5 9](#_Toc38445953)

[Week 6 9](#_Toc38445954)

[Week 7 10](#_Toc38445955)

[Week 10 10](#_Toc38445956)

[UI Design 11](#_Toc38445957)

[Initial UI Design 11](#_Toc38445958)

[Second Iteration 12](#_Toc38445959)

[Final Iteration 13](#_Toc38445960)

[System Design / Architecture 16](#_Toc38445961)

[Implementation 17](#_Toc38445962)

[Code and Project Deliverables 17](#_Toc38445963)

[Building the project and running the app 17](#_Toc38445964)

[Android 17](#_Toc38445965)

[Running the Debug Version 17](#_Toc38445966)

[Building the Release Version 17](#_Toc38445967)

[iOS build 18](#_Toc38445968)

[Relevant Project Files & Folders 18](#_Toc38445969)

[App.js 18](#_Toc38445970)

[package.json & node\_modules 18](#_Toc38445971)

[ios & android 19](#_Toc38445972)

[DeepMapSite 19](#_Toc38445973)

[Testing 19](#_Toc38445974)

[Project Presentation 20](#_Toc38445975)

[Appendix 23](#_Toc38445976)

[Client Documentation 23](#_Toc38445977)

[Installing the Deep Map app on Android 23](#_Toc38445978)

[Adding a marker to the Deep Map app 29](#_Toc38445979)

[Weekly Progress Reports 33](#_Toc38445980)

[Week 1 & 2 33](#_Toc38445981)

[Week 3 34](#_Toc38445982)

[Week 4 35](#_Toc38445983)

[Week 5 36](#_Toc38445984)

[Week 6 38](#_Toc38445985)

[Week 7 39](#_Toc38445986)

[Week 8 40](#_Toc38445987)

[Week 9 41](#_Toc38445988)

[Week 10 43](#_Toc38445989)

[Week 11 44](#_Toc38445990)

[Week 12 45](#_Toc38445991)

Abstract

The idea of a cross-platform mobile application is a very appealing product idea for many different organizations and can seem quite simple for a non-technical client on the surface. However, when starting to look into the problem and analyzing what it would take to create a mobile app for multiple different platforms with the correct systems in place for a client to manage the content that would be on the app, it is apparent that the scope of developing such a system can become much larger than expected. This is the experience we found when developing the Deep Map Mobile App.

Our project for the Computing Science Project course this semester was the Deep Map Mobile App. Proposed by the Accessibility Experiential Learning Coordinator Jennifer Mei at TRU, this project was to create a mobile app that would replicate the functionality currently present on the [Deep Map Website](https://access.trubox.ca/). Our motivation for working on this project was that our team was already familiar and comfortable with Mobile App Development from some courses taken at TRU for Android development, and were interested in what it would be like to develop a cross-platform app for an actual business client , as cross-platform development is a highly sought-after skill in the industry.

Two main challenges to solve this problem arose with this project. The first problem is how we could develop a cross-platform app that would be up to professional standards on both iOS and Android in the limited development time of this project. We were all familiar with developing for Android, and if we had simply stuck with the Android side this would be a simple problem to solve. Knowing that we had to create a product up to our client’s expectations not only for Android, but also for iOS meant that we could either split our team into two halves and develop this application natively on both platforms, effectively doubling our implementation time, or look to an alternative solution. This led us to look into cross-platform frameworks, and subsequently to the React Native framework. Developed by Facebook, this platform is still being constantly worked on but seemed like the industry standard when it came to cross-platform frameworks. While this platform was foreign to us at the beginning of the project, leading to a steep learning curve, we felt that this would be the best approach for us to develop our skillset in software development and save time developing different products at the same time.

The next challenge we had to solve was the challenge of how our client would actually manage the content of the app she was wanting to be made. While we considered creating a custom version of the app for our client to manage the content, we felt the most user-friendly way to achieve this was to create a new webpage solely for the purpose of managing this app. We believed that it would be a better user experience to use a simple user interface rather than manipulate a database directly. To achieve this, we developed a WordPress site specifically made for managing the database content for our app.

This Project Report aims to provide depth into the technical and non-technical requirements we were faced with in this project and the analysis and decisions we made to meet the needs of these requirements. It also serves to present a chronological view of the UI design of the app and the system architecture of the entire software system, from the app to the database and to the webpage created for our client to manage the database. Finally, it serves to provide references to video demonstrations to validate the testing of the app for all features in its feature set.

Requirements

For this project, the requirements were rather simple from a client deliverable perspective. She simply wanted an app to do all of the functionality available currently on the DeepMap website. While the deliverables seemed easy, the requirement to have the app present on both Android and iOS presented some unique challenges in terms of technical requirements. In addition, we also had to consider how our client would manage the content on the app itself.

# Initial Requirements

* Develop a mobile app with the same features as the [Deep Map Website](https://access.trubox.ca/)
  + Integrated map of locations
  + Filter for different resources to appear on the map

# Week 3

* Possibility of Community Feature system, using Database and User Authentication & Management features
  + Added after the base prototype of the app looks polished and functional to the client’s expectations
  + Considered as a "nice to have" feature, but not a necessary requirement
* Distributing through both iOS and Android
  + Android is simple and easy to distribute, Ryan has gone through the process of publishing on the Google Play Store before
  + iOS has an overhead cost of $99 USD per year, client will follow up with her contacts at TRU about the costs and process of distributing the app through TRU's channels
* Develop for Android and iOS using React Native
  + Developing using a cross-platform tool would be a great learning experience
  + Open to alternatives depending on how the development of the first iteration goes

# Week 4

* App needs to scale for tablets and larger phones
* Need to make sure the app will run on iOS early in development

# Week 5

* Cloud-based database required for client to add, edit and manage the points on the map
  + Decided to try Google Firebase, as we have some experience with it already and integrates nicely with mobile and web platforms
* App should have scrolling tabs, rather than having buttons on the bottom of the screen to go to separate pages
* App should use a native map within the app, rather than using a WebView displaying content already on the DeepMap Site

# Week 6

* Can distribute app on app stores with our own accounts, doesn't have to go through official TRU channels
* App should have a tab for "Virtual Communities", online resources without a physical location (ie. a Facebook group)
* We should be providing an easy-to-use webpage hosted on a website that the client can manage the data from the database

# Week 10

* Map markers should have a pop-out page, where a marker can be tapped on again to show a page that consists of all relevant information for that marker

Analysis

The analysis of the problem gave us many different approaches we could take. Since this product was already available on a webpage, we could have easily taken that webpage and just made it display within a mobile app. We decided against this approach, opting to create our own mobile application not only for the flexibility and iteration that we could do with the final product, but also so that we could gain experience as developers to solve a common problem in the industry right now which is how we can create a cross-platform mobile app that maintains a consistent look and feel. We also had a few options to approach the solution to content management. We opted to create a webpage for our client, as that would allow her to manage the database on any device. While this did expand the scope of the project as a whole, it gave us a better glimpse into what true system design looks like in practice.

# Initial Analysis

* Best mobile development platform for us would be React Native
  + Cross-platform support
  + Exports to native projects, where native development can take place if necessary
  + An alternative to this would be to develop both apps simultaneously for both platforms, but the overhead work to do so might take more time that simply learning a cross-platform environment such as React Native

# Week 3

* Community features would require some additional requirements
  + Would require a user authentication system and database support
* How to deal with content management from the client
  + Creating a separate version of the app with additional access to add / update content
  + Creating a simple webpage that can access a Database
* Distributing the app to users
  + Easy for Google Play Store
    - only $25 USD one-time fee
    - Less restrictive
  + Difficult for iOS
    - $100 USD yearly fee
    - more restrictive with content

# Week 4

* App should scale on multiple different devices and screen sizes
  + Decided to start working on the iOS app early in development and continuously test both versions of the app whenever changes were made
  + This decision ultimately lead to our decision to divide the work of the three of us into 3 different platforms, one on Android, one on iOS and one focused on the client webpage using Javascript

# Week 5

* Our client presented us with a contact from a recent trip with some app development experience that we could contact. After discussing as a group, we decided against incorporating him into our project, instead leveraging him as a resource if we needed. We had already determined our technical requirements and didn't want to include someone else that might have a different vision and scope of the final project than our own, especially considering our limited development time for this project.
* Decided to use the Cloud-based Database Google Firebase for content management, as we have some experience with it already and it integrates nicely with mobile and web platforms

# Week 6

* Decided it would be best to change the user interface of the app from buttons at the bottom to take you to a new screen to scrolling tabs
* Switched our map screen implementation from an iFrame that just displays the current DeepMap site to a MapView in React Native that takes advantage of the Google Maps API. Google Maps Services had to be enabled on both platforms for this to work, but it was a necessary improvement that allows us to integrate our database with the map for easy content management for our client

# Week 7

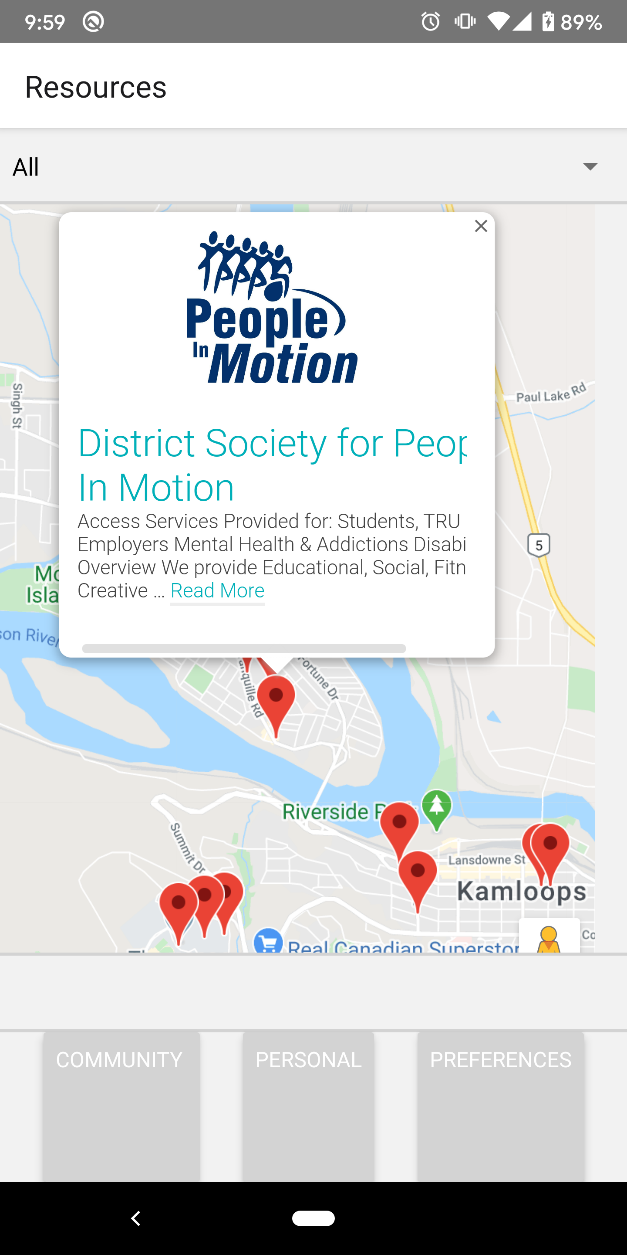
* Added a section on the map for Virtual locations, as some community resources might not have a physical location (eg. a Facebook group, website)
* Created a webpage on Wordpress using our Firebase database and javascript to allow our client to easily edit
  + An alternative to this that we considered was to create a separate version of the app made specifically for our client that would allow her to edit the map, but felt it would be more user-friendly and take less development time to simply create this as a webpage

# Week 10

* To account for the filter we were planning to add to the app, we modified the database and the client webpage to allow for location categories. A location in the database now has a subtree of categories with True or False values to indicate whether it is part of a community or not.

UI Design

# Initial UI Design



For our first prototype, we decided to revolve our UI elements around the main focus of the app, which is the Deep Map itself. In order to save on development time and get a first iteration to our client as soon as possible, we opted to simply display the map of the DeepMap webpage as a WebView, simply displaying the Deep Map as it currently is on the DeepMap Webpage. While not entirely functional at this point in the project, we also included tabs at the bottom of the page for future pages to be added, such as a Community and Preferences page. We also included a filter at the top of the map, as one of our requirements is to allow for filtering by different communities.

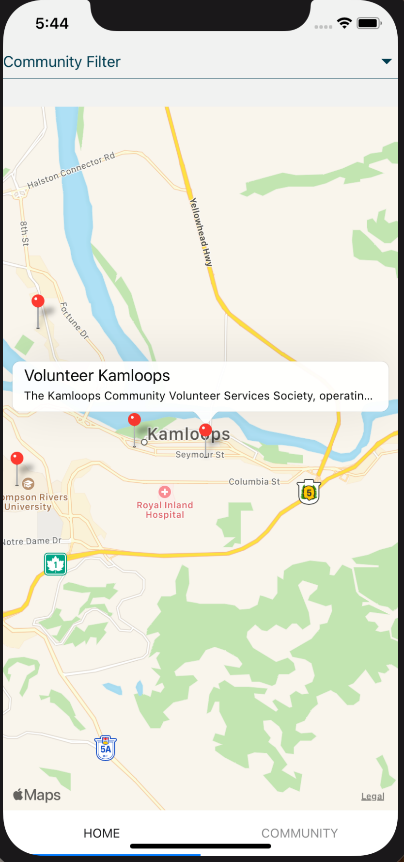
# Second Iteration

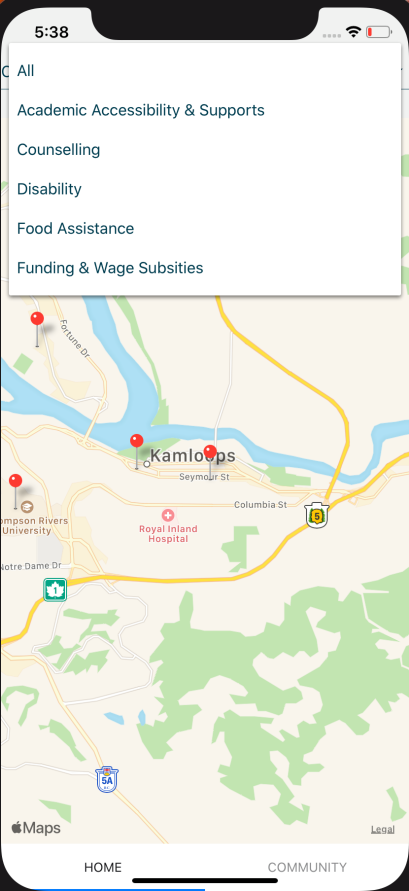
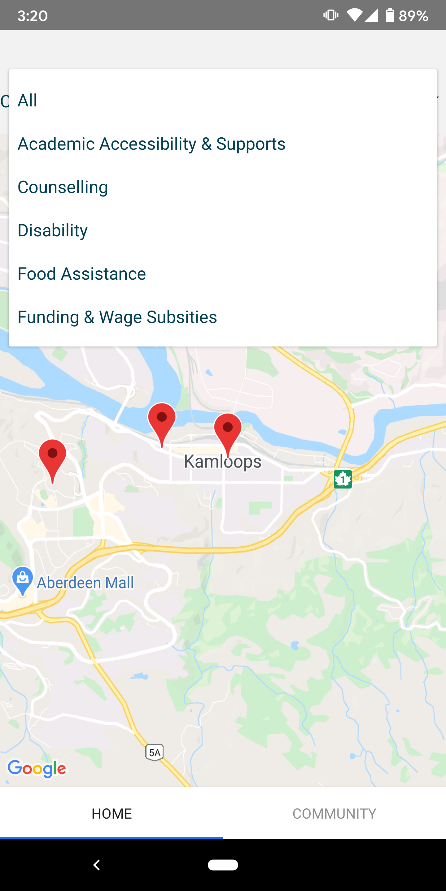


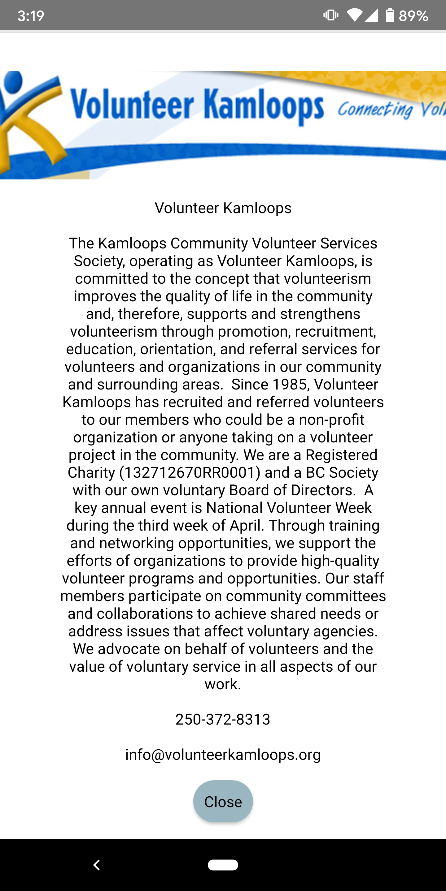
In our second iteration, we implemented the MapView within React Native. This view takes advantage of the Google Maps API. We chose to create our own MapView rather than continue using a WebView to the DeepMap webpage, as this would allow us to manipulate the markers on the map using our own database and do any further edits to the map outside of using the webpage.

# Final Iteration

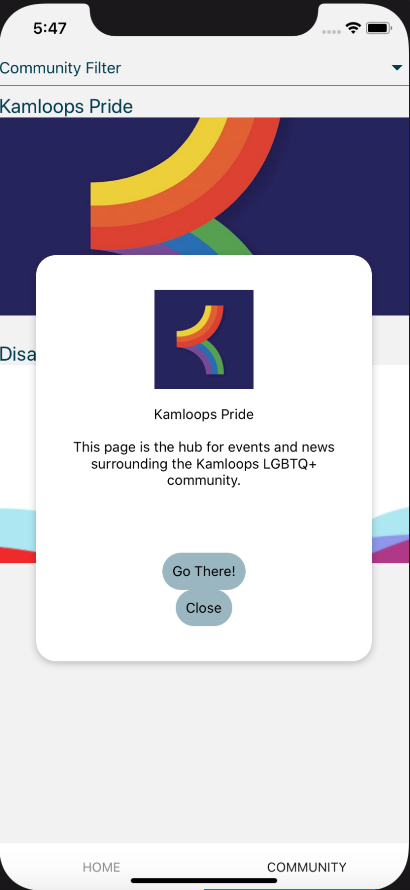
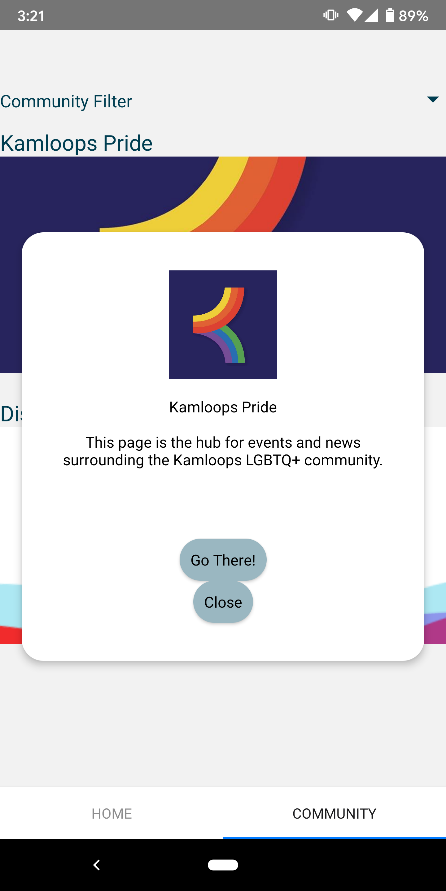












Our Final Iteration of the app further polishes the look and feel of the app, taking advantage of scrolling tabs at the bottom of the screen. It also uses a Community tab to show Virtual Locations, community resources that don't have a specific location such as Facebook groups and websites. In addition, we created a modal pop-out page that displays a larger, more detailed view of each individual marker on the map. This modal is available on the map as well as the Virtual Communities page.

# System Design / Architecture

Our system design / architecture is composed of three main parts. The first, and most obvious part of this architecture is the app itself. This part of the system simply displays the content of the system to the end users, and is the only user-facing part of the system.

The second part of our architecture is our Firebase database itself. This NoSQL database is a simple database that has a unique child for each marker on the map. Under each marker, the database stores all relevant information for that marker, including the title, description, email, phone number and website of the location. In addition, it also includes a subtree of booleans, indicating which communities each marker is a part of.

Finally, the last piece of this architecture is the client webpage. This webpage allows our client to directly manipulate the database in a very user-friendly interface. This hides the backend technical database from our client and ensures the integrity of the database, as the client webpage is very well structured to not "break" the underlying structure of the data in the database.

Implementation

# Code and Project Deliverables

All implementation code and deliverables created for this project can be located at our public GitHub Repository at <https://github.com/djryancarson/DeepMap>.

# Building the project and running the app

In order to run the DeepMap project, you first have to setup React Native on your local machine. The instructions on how to do so are located on the [React Native Website](https://reactnative.dev/docs/environment-setup). For this project, we built the project using the React Native CLI Quickstart instructions, so the project has not been tested with the Expo environment.

## Android

### Running the Debug Version

1. Open a command line and navigate to the project folder
2. run npm i
3. adb uninstall com.deepmap (only necessary if a previous version is already installed)
4. npx react-native run-android

Note: this will often fail when trying to build the debug app. In our experience, you sometimes have to run npx react-native run-android up to 5 times for it to successfully build. An alternative to this would be to simply run the android folder within Android Studio.

### Building the Release Version

1. Move the **gradle.properties** file with the keystore info to DeepMap\android
2. Move the **my-upload-key.keystore** to DeepMap\android\app
3. Open a command line and navigate to the project folder
4. Run npm i
5. Open a command line and navigate to the android folder of the project (DeepMap\android)
6. Run gradlew bundleRelease
   * Note: the [Official Documentation](https://reactnative.dev/docs/signed-apk-android) advises to use ./gradlew bundleRelease. In our experience, this does not work on a Windows Environment
7. Navigate back to DeepMap and run npx react-native run-android --variant=release
8. The release apk will be located at DeepMap\android\app\build\outputs\apk\release

## iOS build

You have have some issues using the command npx react-native run-ios

If you get errors when building

1. Make sure to have Cocoapods installed
2. Navigate to the IOS Folder in your Terminal
3. Use the command pod install
4. Try building the app again

# Relevant Project Files & Folders

Many of the files in the project directory are generated by React Native, but there are some important files that must be worked on in order to modify the project.

## App.js

This file is the main file that loads everything in the project upon opening. It is similar to the Driver class or Index file of a typical program or webpage. This file includes all of the import statements relevant to the project, the HomeScreen class including all the UI elements that appear when the app loads for the first time, the integration for the Firebase database and the Style Sheet that includes all the styling necessary for the UI elements.

## package.json & node\_modules

The **package.json** file includes a list of the necessary dependencies needed for the React Native project. When the command npm i is run, it looks to this file to create everything necessary in the **node\_modules** folder.

## ios & android

The **ios** and **android** folders hold the project files for the iOS and Android projects respectively. They were automatically generated when the React Native project was created and change content based on and change as the React Native project is edited and rebuilt. These folders can be directly imported into the xCode and Android Studio IDEs, allowing for easy testing if already familiar with these IDEs or further implementation specific to either platform.

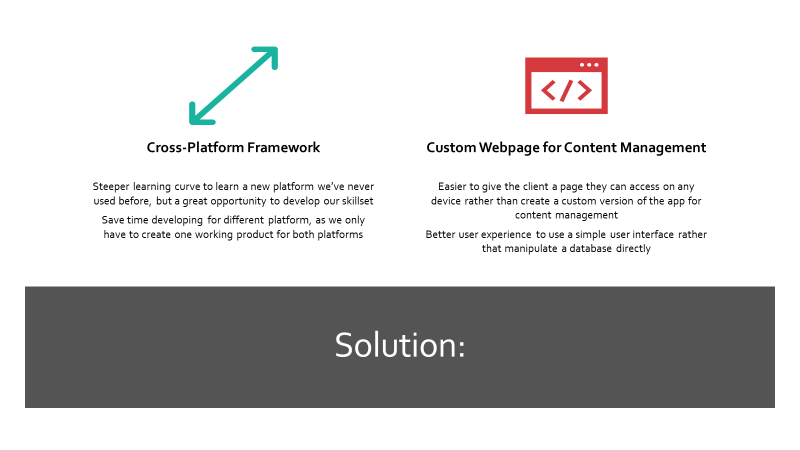
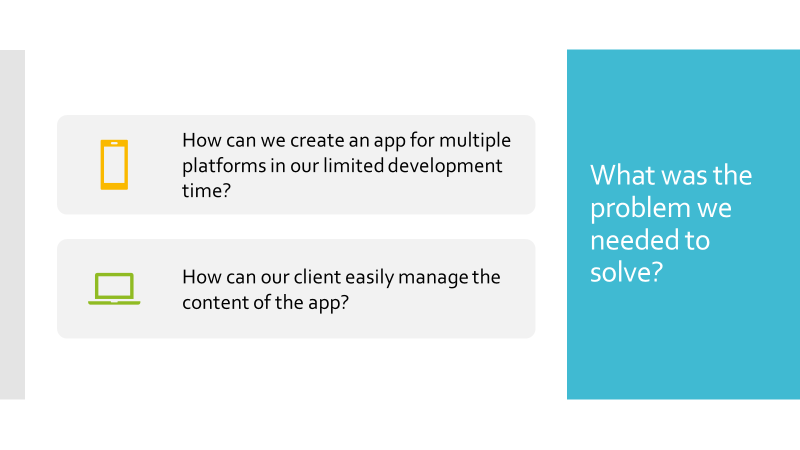
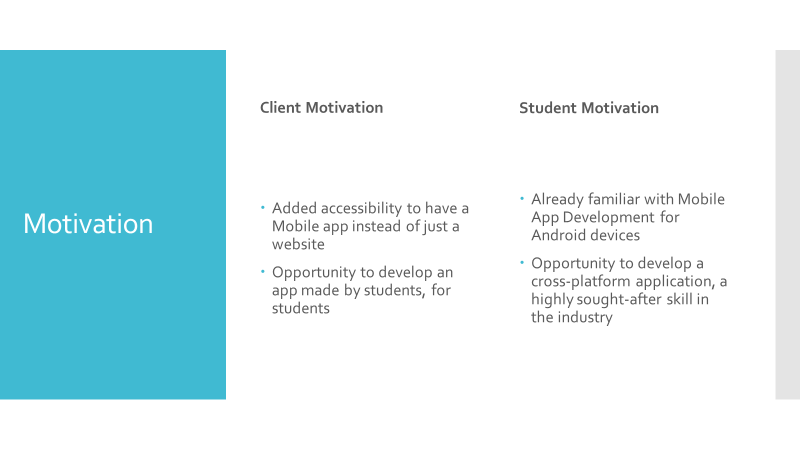
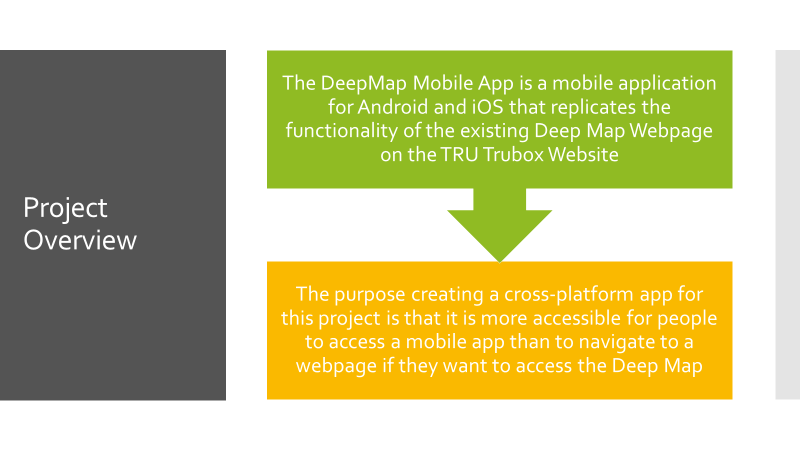
## DeepMapSite

This folder contains the PHP files for the site which is used to input data to the database to be displayed by the app. deepmap-login.php, logout.php and session.php are all for simply handling sessions and specifically login.php is providing a simple login page. The entirety of the functionality of site lies within home.php. The site consists of an input form that is invisible by default. It becomes visible when the user clicks the 'Add New' button or the 'Edit' button on any of the resources and in that case the list of current resources will then become invisible. The list of current resources is built from the database dynamically and certain elements for each are shown or hidden depending on whether it is virtual or not. Additionally, when editing a resource, the form fields will be filled with its current values. Certain form fields will be hidden or shown depending or whether the resource is a virtual one or not.

Testing

For this project, our testing plan was quite simple. Unlike a complex product such as a machine learning product, our final product was rather simple, so we simply focused on making sure our app and the client website remained stable and reliable under different use cases. To illustrate the different use cases and feature set working under different conditions, we have included two demonstration videos to our Project Binder on our GitHub page. These demonstration videos are located at <https://github.com/djryancarson/DeepMap/tree/master/Project%20Binder/5.%20Testing>. While there is a bit of lag with the demonstration video for the app, this is caused by the software used to capture the footage and is not reflective of the final product.

Project Presentation

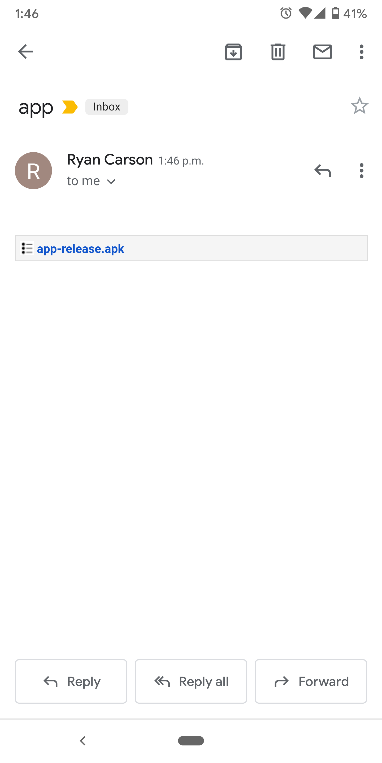


Appendix

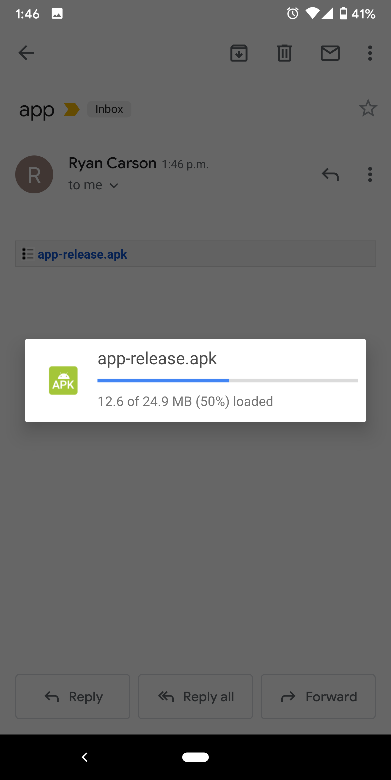
# Client Documentation

## Installing the Deep Map app on Android

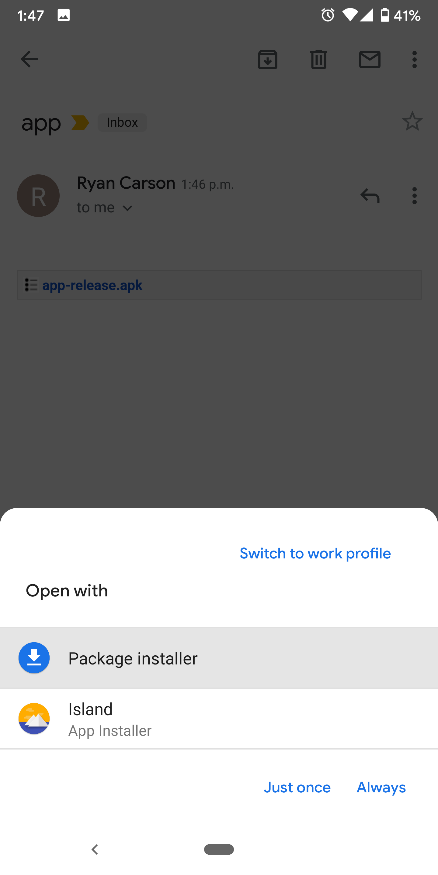
**Step 1.** Open the app-release.apk file from the email on your phone.



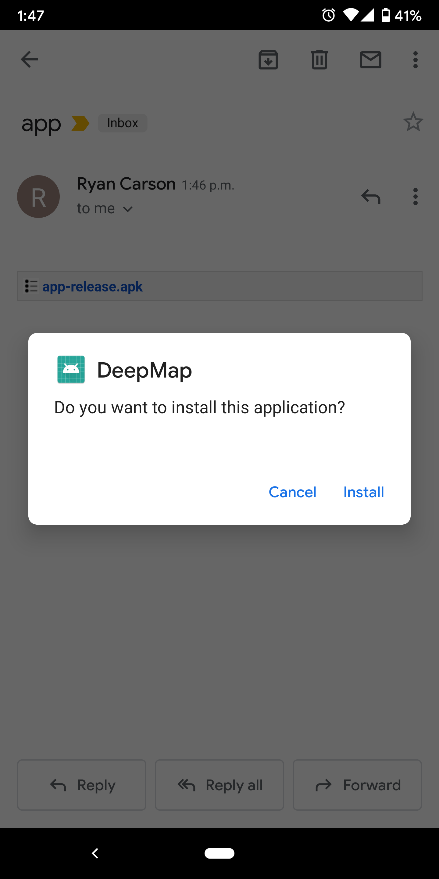
**Step 2.** Allow the app to load.



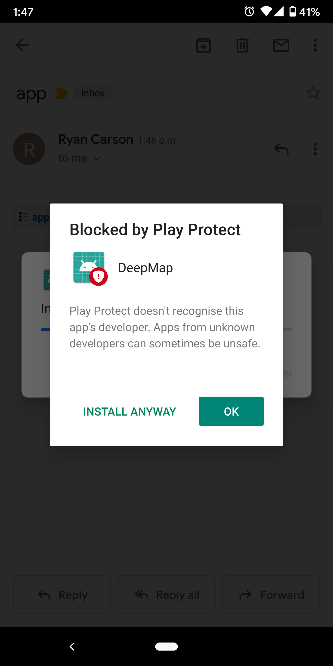
**Step 3 (if prompted).** Choose the Package Installer.



**Step 4.** Select **Install**.

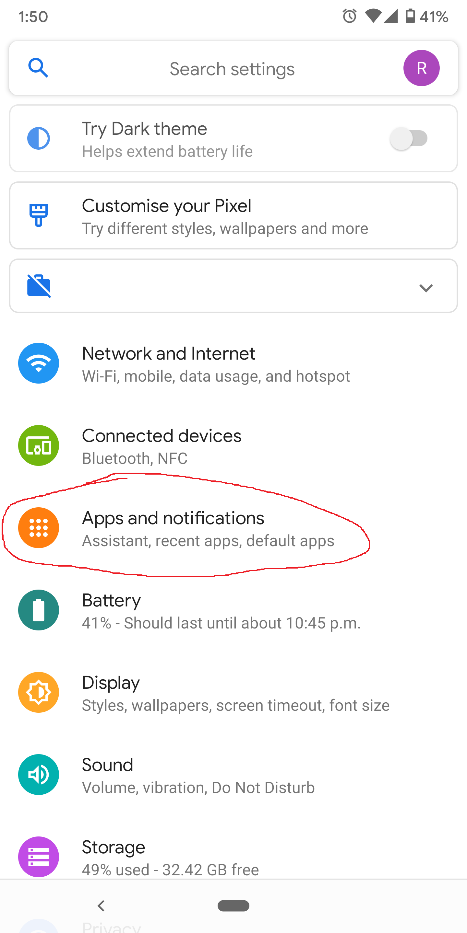


**Step 5.** Select **INSTALL ANYWAY**.

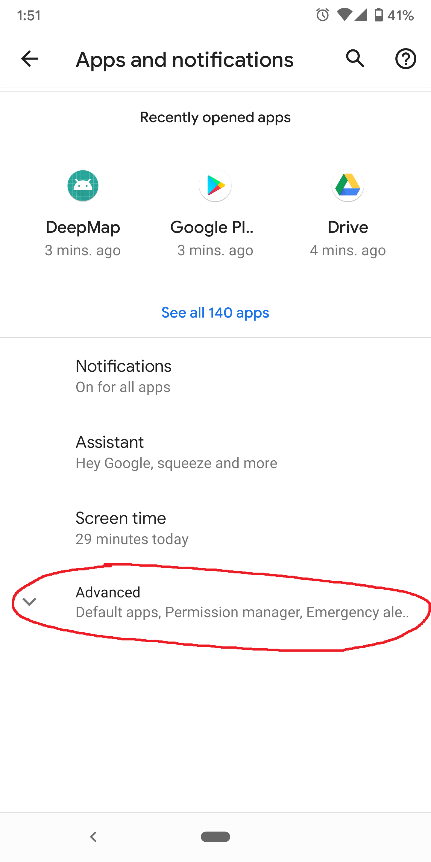


**Note**: You may be asked to enable your email program to allow you to install APK files. If so, follow these steps:

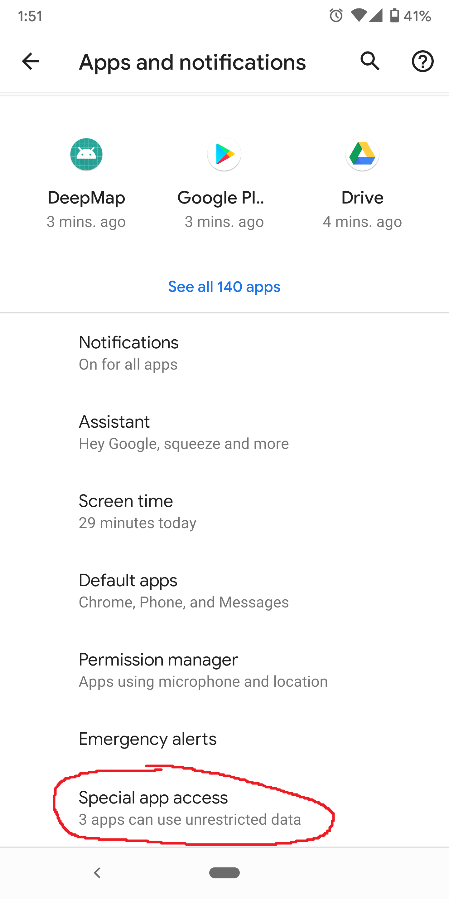
**Step 1.** Select **Apps and notifications.**

****

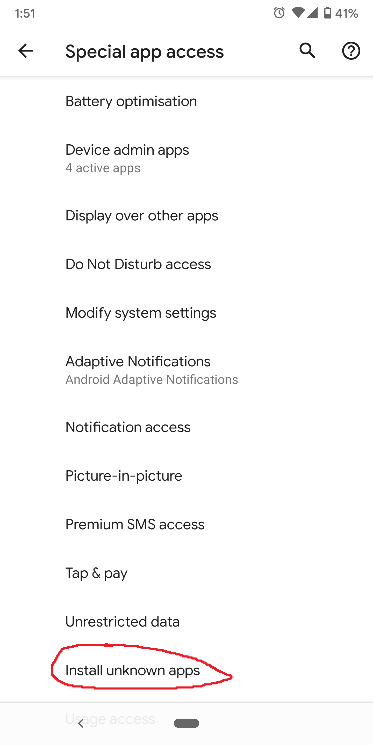
**Step 2.** Select **Advanced**.



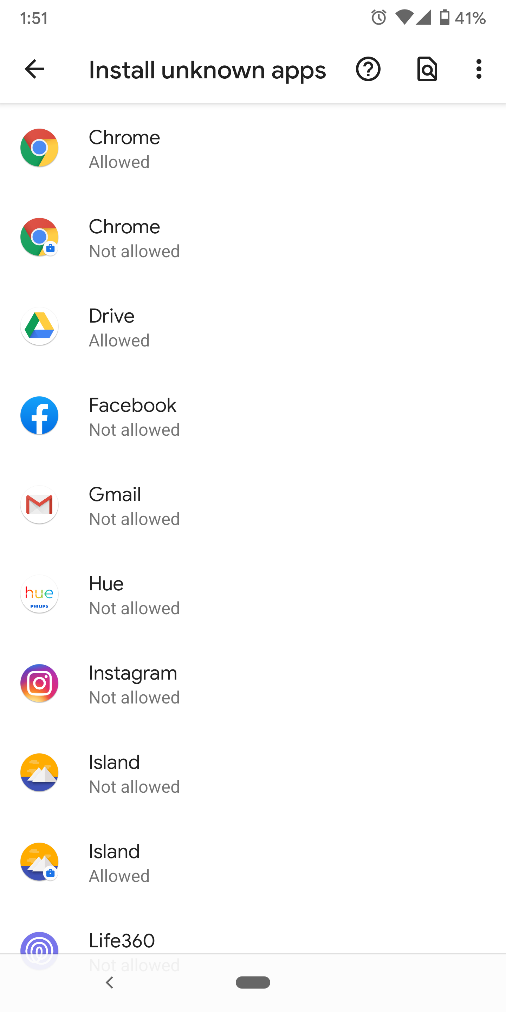
**Step 3.** Select **Special app access**.



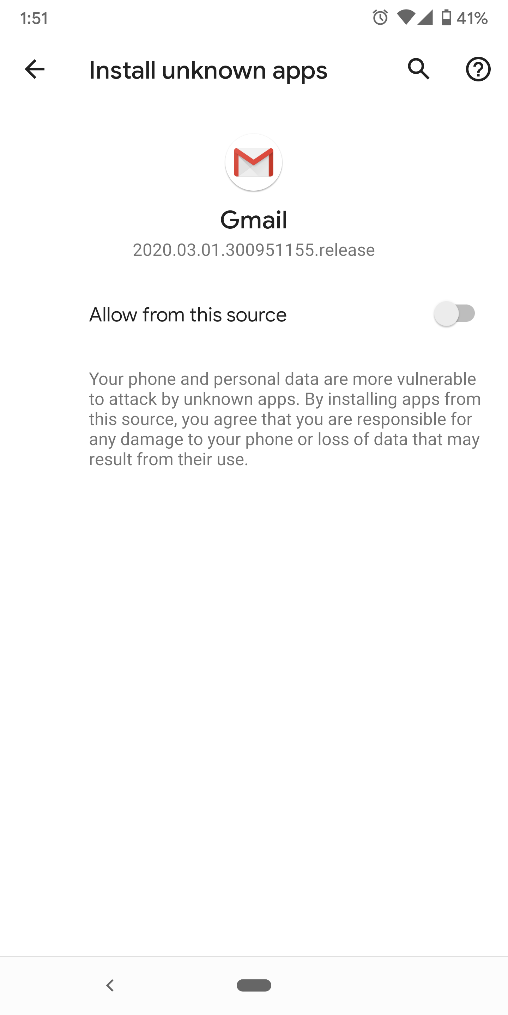
**Step 4.** Select **Install unknown apps.**

****

**Step 5.** Select your email app you are trying to open the file on (eg. Gmail).



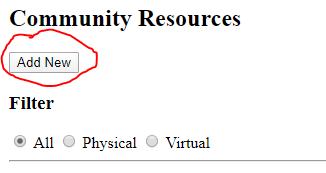
**Step 6.** Select the option **Allow from this source**.



## Adding a marker to the Deep Map app

**Step 1.** Go to the [Deep Map Database Webpage](http://www.djryan.ca/DeepMap/home.php)

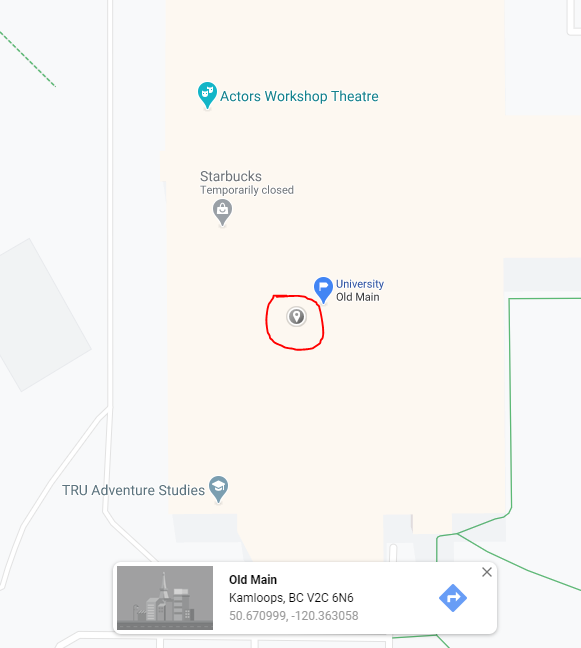
**Step 2.** Select Add New



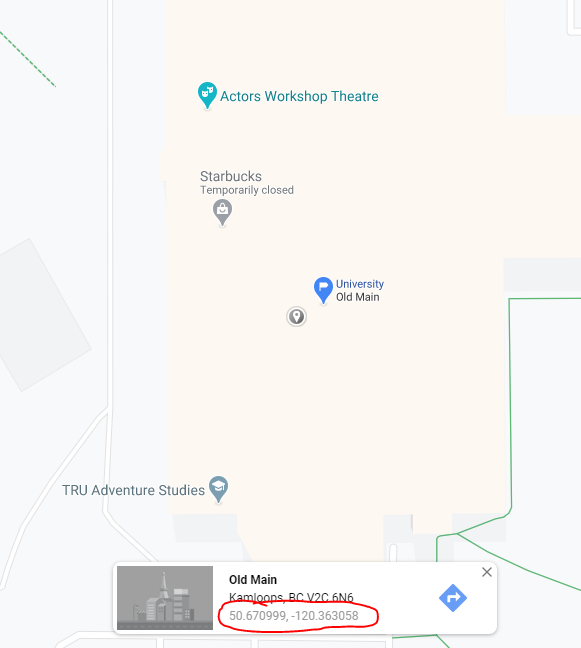
**Step 3.** Go to [Google Maps](https://www.google.ca/maps/)

**Step 4.** Click on the point you would like to add

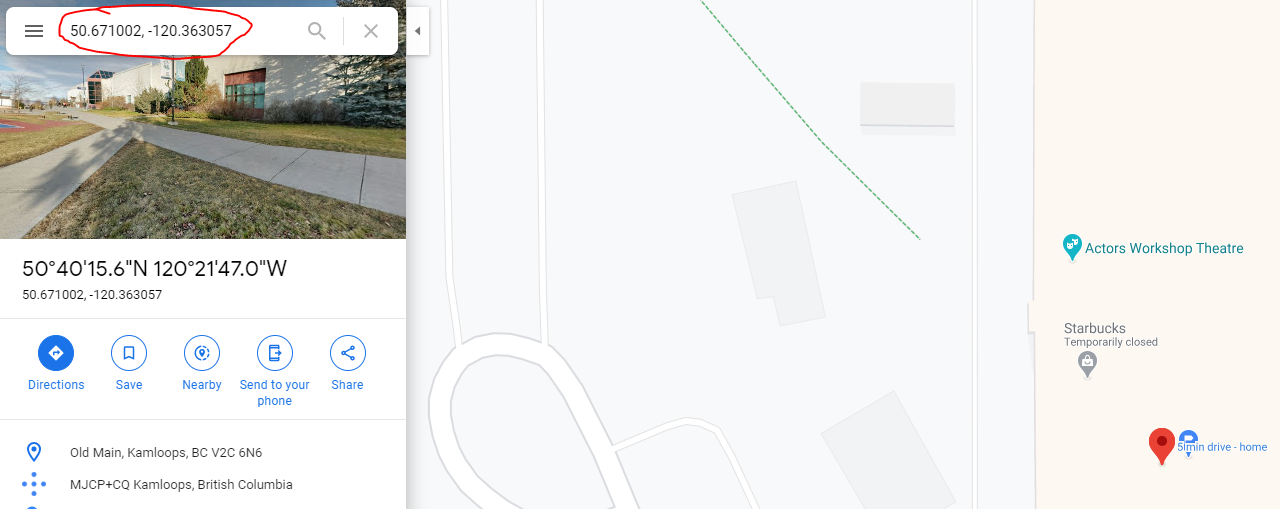
**NOTE:** Do not click on an actual marker on Google Maps, click on an empty space instead



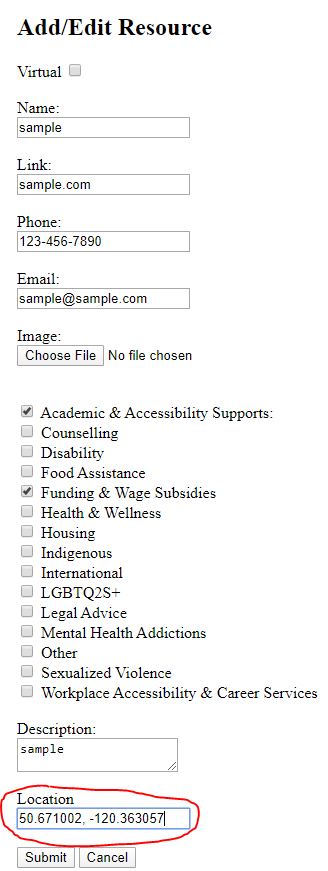
**Step 5.** Click on the **Coordinates**



**Step 6.** Copy the Coordinates at the top of the screen



**Step 7.** Paste these coordinates into the Location field along with any other relevant information you’d like to add and click **Submit**



# Weekly Progress Reports

## Week 1 & 2

**Week 1 & 2: Weekly Report for COMP 4910 Project Course**

**Deep Map**

**Jennifer Mei**

**Ryan Carson, Dyson Fraser, Tyrel Froese**  
  
**Summary:**

In the first two weeks of the semester, our team has formed and decided on our project of the Deep Map. We also had our first meeting with our client and determined some key requirements of our project. We have identified that the minimum viable product of our project is to simply develop an app with the same features as the website currently has. Our client also advised that she will reach out to the individual that developed the current website and will have him reach out to us in the case that we can convert some of the work he has done to our Mobile App. Our client would also like to know who will maintain this app post-project. We have determined that we would like to develop this app mainly using the React Native framework for its multiplatform functionality, but are willing to consider alternatives if they better suit our needs. Finally, we booked a follow-up meeting with our client for February 6th to discuss the work we will have completed at that point.

**Task Completion from Last Week**

* Formed team and determined project
* Had first meeting with client
* Determined Minimum Viable Product
* Determined our preferred development framework of React Native

**Task for This Week**

* Research the feasibility of React Native as a framework
* Determine additional feature set that we could implement into this mobile app to expand the overall scope of this project
* Determine the technical requirements of the project and become familiar with the development environment in preparation for development
* Determine post-project support for this application

**Additional Information**

None this week.

**Known issues / things blocking progress**

None this week.

## Week 3

**Week 3: Weekly Report for COMP 4910 Project Course**

**Deep Map**

**Jennifer Mei**

**Ryan Carson, Dyson Fraser, Tyrel Froese**  
  
**Summary:**

In the third week of the project, our team has spent most of our time solidifying our requirements, both from a technical perspective but also from a client perspective. After our weekly meeting with Kevin, we determined that we had to follow up with our client more frequently than we had planned, and immediate organized a meeting with her to clarify our requirements with her. We have also scheduled concurrent meetings for the next month and have agreed with our client that we will be having weekly meetings moving forward.

In this week’s meeting, we discussed the feasibility and challenges that come with a community feature system, including the necessity for a database system and user authentication & management system. We agreed that this feature would remain as a “nice to have” feature, but will be left out until after we create the base prototype of the app and make sure that it is polished and functional to the client’s expectations.

We also circled back with the client on the post-project support of this app, advising her that while ourselves or the Computer Science department cannot actively continue with this project after the course is complete, we can do our best to help her find a third-party that could continue the groundwork that we will be creating with this project.

Finally, we discussed how the distribution of this app would look like. We mentioned that distributing the Android version would be rather painless, as Google is very flexible with publishing to the Google Play Store, and the app can even be easily distributed outside it. We also mentioned the challenges that come with publishing on the Apple App Store, including the overhead costs associated with it. Our client was relatively confident that she could find the budget from TRU to publish this, or even publish the app through the same avenues that TRU has published apps on the app store already. She will be following up with TRU regarding this.

Regardless of the platform limitations, we have still decided that we would like to try developing our app for both platforms using the React Native Framework, as we agreed that developing for one specific platform is limiting in the real world and it would be a valuable skill for us to learn to develop on a cross-platform framework. We have also spent some time individually to learn how to develop for the React Native Framework, and will be using the next week of development to try to create a first prototype of this app using this framework, or seek an alternative framework based on our experience this coming week.

The requirements that we have for our first prototype are simply that it must display the map as shown on the Deep Map website. This will either be done by simply displaying the Deep Map website on the app itself, or moving the Deep Map onto our own app. We have also reached out to the creator of the Deep Map, in the event that we could reuse or modify some of the work that he has done to suit our needs with this project.

**Task Completion from Last Week**

* Researched the feasibility of React Native as a framework
* Determined additional feature set that we could implement into this mobile app to expand the overall scope of this project
* Determine the technical requirements of the project and became familiar with the development environment in preparation for development
* Determined post-project support for this application
* Clarified technical and non-technical requirements with the client

**Task for This Week**

* Develop prototype of the Deep Map using React Native as a framework
* Determine if React Native will suit our needs of this project, or if we need to find an alternative

**Additional Information**

None this week.

**Known issues / things blocking progress**

None this week.

## Week 4

**Week 4: Weekly Report for COMP 4910 Project Course**

**Deep Map**

**Jennifer Mei**

**Ryan Carson, Dyson Fraser, Tyrel Froese**  
  
**Summary:**

In the fourth week of our project, we began development of a prototype of our app to show to our client using React Native. Our client is out of town for the week so we weren’t able to follow up with her in person, but that gives us more time to work on and polish up our prototype to show to her in this coming week. We also met with Kevin to discuss our progress and determined that we should be looking ahead to support different screen sizes, such as tablets, and that we should be frequently testing and making sure that our app not only works well for Android but also for iOS.

So far, our prototype consists of a WebView (the React Native version of an iframe), with bottom tabs to navigate to the other parts of the Deep Map Website. Right now we are just trying to solidify the User Interface to make sure our client is happy with how it will look. We are planning to implement a native Map view on the app itself and organize the UI of the map in a way that makes the usability better for the mobile platform, but are using a WebView for the time being to be able to create a Minimum Viable Product earlier in the development so we can begin the iteration process quicker.

In this coming week, we will be meeting with our client and presenting our prototype. We will be gathering feedback from her on the look, feel, and usability with the current prototype for consideration into our next iteration of the app. We will also be trying to determine how we will store the information of our app, as well as how our client will be able to manage the content of the app.

**Task Completion from Last Week**

* Developed prototype of the Deep Map using React Native as a framework
* Determined that React Native will suit our needs of this project.

**Task for This Week**

* Continue working on the UI of the prototype in anticipation of our client meeting
* Implement a Map View in React Native and start to develop the unique UI of the app
* Identify database and content management requirements of the app

**Additional Information**

None this week.

**Known issues / things blocking progress**

None this week.

## Week 5

**Week 5: Weekly Report for COMP 4910 Project Course**

**Deep Map**

**Jennifer Mei**

**Ryan Carson, Dyson Fraser, Tyrel Froese**  
  
**Summary:**

In the fifth week of our project, we finished our prototype and presented it to our client. Our client was very pleased with the prototype and had no negative feedback for us. She also mentioned that she had met an individual on her recent trip that has experience creating mobile apps such as ours and will be sending him our information so we can use him as a resource if we’d like.

We also discussed the issues with distributing the app via iOS. Jennifer had not yet followed up with TRU about this and wanted some more information into the specifics with pricing. We provided this to Jennifer for her to follow up with TRU and are awaiting her response.

This week, we are beginning the next iteration of our app. This next iteration will use a MapView within React Native, rather than a WebView that shows the Deep Map Website. We will also start integrating our app with the cloud-based database Firebase so that we can have our map update over time with new contacts that want to be on the Deep Map. Finally, we are working to implement scrolling pages into our app, rather than using the tabs that we currently have implemented.

**Task Completion from Last Week**

* Finished the prototype of the app and presented it to the client for feedback
* Identified database and content management requirements of the app
* Began implementing a Map View in React Native

**Task for This Week**

* Continue to work on implementing a MapView in React Native
* Integrate the Firebase Database into the React Native project to be integrated with the map locations
* Change the Tabs on the app into scrolling pages

**Additional Information**

We met with a friend who has three years of experience with React Native this week to try to get further understanding of React Native as a platform.

**Known issues / things blocking progress**

None this week.

## Week 6

**Week 6: Weekly Report for COMP 4910 Project Course**

**Deep Map**

**Jennifer Mei**

**Ryan Carson, Dyson Fraser, Tyrel Froese**  
  
**Summary:**

In the sixth week of the project, we began the development of our second iteration of the app. This iteration will be more representative of the final project, and as such, will be more involved than the last iteration. We have begun setting up the Google Maps API for our project, which is turning out to be more involved than originally anticipated, as there is a significant amount of setup and configuration that must be done natively for the Android project and the iOS project.

We have also begun setting up Firebase in our app. Once integrated, it will be used to store and display the map information as it is displayed on the Deep Map Site. We are also looking to integrate it with a web page for easy content management for our client. We have access to the current Deep Map website on the TRU website, so we will try to implement this on a separate page on that site.

Finally, we have worked to polish the User Interface further, implementing scrolling tabs rather than buttons that lead to different pages.

We also met with our client and discussed distribution of the app once more. We concluded that there is a lot of “red tape” involved with publishing this app through the official TRU channels and it would be best to simply create our own distribution on iOS and Android for this project. Our client advised that she is willing to expense the cost of an Apple Developer Account.

In the coming reading break and next week, we will continue developing our next iteration of the app with complete or near-complete functionality of the MapView and Firebase Integration in anticipation of the coming Midterm Review with our client.

**Task Completion from Last Week**

* Worked toward implementing a MapView in React Native
* Started to setup and integrate the Firebase Database into the React Native project to be integrated with the map locations
* Changed the Tabs on the app into scrolling pages

**Task for This Week**

* Complete the implementation of the MapView for iOS and Android
* Complete the Firebase Integration with the app
* Create a simple WordPress Form for the client to manage content on the app

**Additional Information**

None this week.

**Known issues / things blocking progress**

None this week.

## Week 7

**Week 7: Weekly Report for COMP 4910 Project Course**

**Deep Map**

**Jennifer Mei**

**Ryan Carson, Dyson Fraser, Tyrel Froese**  
  
**Summary:**

In the seventh week of the project, we were able to finalize the basic functionality of the MapView on Android. The application now shows a map with markers, that focus in and display the name and description of the location when selected.

We also cleaned up the User Interface more, moving the tabs from the top to the bottom of the screen and adding a tab for “Virtual Locations”, where we can display online resources for marginalized communities.

Finally, we have implemented our Firebase database within our app. While this is on a separate branch of the project, and only holds strings at the moment, this is a huge step toward being able to manage our locations in Firebase. We have also created the form on WordPress that will allow our client to update and manage the content on the app and have implemented the same Firebase database in that form.

In the coming week we will continue working on the Firebase implementation, setting up map coordinates in the database and reflecting these coordinates in the app. We will also implement functionality in iOS for the MapView and prepare the app to be published on the Google Play Store and the Apple App Store.

**Task Completion from Last Week**

* Completed the implementation of the MapView for Android
* Completed the Firebase Integration with the app (for simple strings)
* Created a simple WordPress Form for the client to manage content on the app
* Integrated Firebase with the WordPress Form (for simple strings)
* Implemented scrolling tabs on the bottom of the screen
* Implemented “Virtual locations” tab for external links on different tab

**Task for This Week**

* Integrate map coordinate values in Firebase and store map locations in the database
* Implement the MapView in iOS
* Prepare the app for the Google Play and Apple App Store

**Additional Information**

We have been working from a Public GitHub Repository. This repository includes our project file, as well as a folder called Project Binder that includes all of our documentation. The repository is located at <https://github.com/djryancarson/DeepMap>.

**Known issues / things blocking progress**

None this week.

## Week 8

**Week 8: Weekly Report for COMP 4910 Project Course**

**Deep Map**

**Jennifer Mei**

**Ryan Carson, Dyson Fraser, Tyrel Froese**  
  
**Summary:**

In the eighth week of the project, we began working on integrating some of the functionality we have been working on separately. First, in preparation of integrating the map markers with Firebase, we started to integrate Firebase into the Android version of the app. The fact that we are attempting this within the constrains of react native meant that it is becoming a larger task than anticipated and is still a work in progress.

We have also worked on integrating the MapView that we developed in Android for our midterm review into iOS. Fortunately, this was a rather simple implementation that is stable on our iOS app.

We started looking into publishing the app on the Google Play store. The roadblock to this is that the Google Play Store requires a logo to be uploaded for the app to be published. We will discuss how we want to design our logo with our client this week.

Finally, we were able to upload images in the form of base64 image files into the Firebase Database. This will allow us to display images of the different physical and virtual locations within our app.

In the coming week, we will continue to work on the integration work on our app. We will also spend some time as a team to map out the remaining sprints of our project, create a handover process and determine the final scope of what we can accomplish before the project is handed over. We will review this with the client to confirm we are still meeting the expectations of the client. We will also look into integrating our Wordpress form onto the Deepmap site to be used by our client.

**Task Completion from Last Week**

* Worked on implementing Firebase in the Android version of the app
* Implemented the MapView in iOS
* Determined the requirements to publish the app on the Google Play store
* Implemented images into Firebase Database

**Task for This Week**

* Finish integrating Firebase into the Android version of the app
* Continue determining requirements of publishing the app on both app stores
* Create timeline of final sprints and wrap-up
* Work on integrating the Wordpress form on the DeepMap site

**Additional Information**

None this week.

**Known issues / things blocking progress**

* Integrating Firebase with the Android version of the app

## Week 9

**Week 9: Weekly Report for COMP 4910 Project Course**

**Deep Map**

**Jennifer Mei**

**Ryan Carson, Dyson Fraser, Tyrel Froese**  
  
**Summary:**

In the ninth week of the project, we continued working on integrating and developing the parts of our app that we need functional for the final product. After some trouble with the development environment, we were finally able to incorporate Firebase into the Android version of our app. We also continued working on the Wordpress site for our client, implementing functionality for our client to edit the markers on the map. Finally, we have continued to work on integrating Firebase into our map. We are having some trouble trying to merge the old branch where the Firebase implementation was developed into the master branch of our Git repository, but when this is done, we should be able to fully integrate Firebase into our map. The challenges we are having with integration has slowed our progress a bit, so we are still in the process of publishing the app to the app store, but we have accounted for overflow with our development in our timeline leading up to the final deliverable.

In the coming week, we will merge our Firebase implementation with the master branch of our Git directory and integrate it with our map. We will also setup the filter for our map to filter by the different categories currently on the DeepMap Website, including setting up our database to support the different categories. We will try to publish the app on the Google Play store and the iOS app store if time permits. Finally, we will move the WordPress page we created for our client from the test site we’ve been working from onto the Trubox site that our client has access to.

**Task Completion from Last Week**

* Finished integrating Firebase into the Android version of the app
* Continued determining requirements of publishing the app on both app stores
* Created timeline of final sprints and wrap-up
* Worked on integrating the Wordpress form on the DeepMap site

**Task for This Week**

* Setup filter for the map, including creating children in the database to act as filter categories for the markers.
* Post app to Google Play Store and possibly iOS app store
* Upload the client WordPress page to her Trubox site
* Merge the Firebase implementation with the working branch and integrate the database with the map

**Additional Information**

None this week.

**Known issues / things blocking progress**

* Merging the changes of an older branch that the Firebase implementation was developed on into the master branch of the repository

## Week 10

**Week 10: Weekly Report for COMP 4910 Project Course**

**Deep Map**

**Jennifer Mei**

**Ryan Carson, Dyson Fraser, Tyrel Froese**  
  
**Summary:**

In the tenth week of the project, we spent the majority of our time with integration. We had been working on implementing Firebase in one branch and the map on a different branch in Git. Now that we had both branches working on both platforms, we spent this week integrating both branches together into one main branch to continue developing on. This process was much more difficult than anticipated, as we not only had to incorporate the learning of our unfamiliar development platform, but also learning how to use Git to merge wildly different branches of development. After much bug testing and collaboration, we finally managed to get the functionality of our two large branches integrated and working in one main branch.

As this merge took longer than anticipated, we have yet to publish the app to the Google Play store. We have had the chance to research how the filter will work in React Native and will begin implementing this filter early next week, which will require us to implement the filter functionality in React Native, our Firebase implementation and our implementation of our map.

Finally, we completed the functional requirements of our client’s webpage. It now has full functionality, including the ability to add community resources, including adding an image, selecting which communities it is a part of, and entering specific coordinates to where it will be. In addition, it also allows her to edit any point on the map or delete it.

In the coming week, we will continue to work on our map filter using the newly integrated master branch of our project. We will also upload the WordPress page to the TruBox site now that it is functionally complete. We will again try to post our app to both app stores. Finally, we will work toward creating our pop-out page, where a user can select a point on the map and have it open a new page with all relevant information.

**Task Completion from Last Week**

* Merged the Firebase implementation branch with the master branch of the project
* Integrated and tested the Firebase implementation in the master branch of the project
* Finished developing the full functionality of the client Wordpress site

**Task for This Week**

* Setup filter for the map, including creating children in the database to act as filter categories for the markers.
* Post app to Google Play Store and possibly iOS app store
* Upload the client WordPress page to her Trubox site
* Work toward creating a Pop-out page for the markers

**Additional Information**

None this week.

**Known issues / things blocking progress**

None this week.

## Week 11

**Week 11: Weekly Report for COMP 4910 Project Course**

**Deep Map**

**Jennifer Mei**

**Ryan Carson, Dyson Fraser, Tyrel Froese**  
  
**Summary:**

In the eleventh week of the project, we created the release build of the app in progress. This build required the app to be digitally signed with a Java Keystore file. This file was generated and added to the app to create the signed release build. We then uploaded this build to the Google Play Store along with the store listing including screenshots and a description of the app. We also provided this build to our client, along with documentation (with images) on how to install this app to her Android phone, as she is not necessarily well versed in this type of technology.

We also continued to work on the client webpage that she will be managing the locations with. We added the functionality to edit the Virtual Locations, which are resources located online such as Facebook groups and online communities. We hit a roadblock with publishing the page on the client’s Trubox site due to the permissions of the Trubox environment. We have reached out to the Trubox administrators for assistance with this and are awaiting their reply.

Finally, we began researching how to create the pop-out page, where a selected location will open in a new page populated with relevant information. There is a lot of work to do with this functionality within React Native and our database, so this is still an ongoing task.

In the coming week, we will continue to work on the pop-out page. We will also polish the client webpage to make sure it is reliable and easy to use. We will continue to monitor the release of the app to the Google Play Store, making adjustments if the app release is rejected. Finally, we will start to finalize the Project Binder and any other necessary documentation for our client and any other project team that might continue with this app in the future.

**Task Completion from Last Week**

* Setup filter for the map, including creating children in the database to act as filter categories for the markers
* Created a signed release build of the Android app to submit to the Google Play Store
* Provided release build of the app to the client with documentation on how to install it to her Android phone
* Posted app to Google Play Store
* Tried to publish the webpage to the Trubox site and started to add functionality for the virtual communities
* Researched how to create the pop-out page

**Task for This Week**

* Continue to work on the Pop-out page
* Continue to “polish” the client webpage and upload it to her Trubox site if possible
* Monitor the release of the app to the Google Play Store, making changes if necessary based on the approval process
* Start to finalize the Project Binder and any further documentation in anticipation of the project closeout and hand-off.

**Additional Information**

None this week.

**Known issues / things blocking progress**

None this week.

## Week 12

**Week 12: Weekly Report for COMP 4910 Project Course**

**Deep Map**

**Jennifer Mei**

**Ryan Carson, Dyson Fraser, Tyrel Froese**  
  
**Summary:**

In the twelfth week of the project, we created the Pop-out page on the map. This page opens in a Modal window. We have yet to populate this page with any relevant information, but it does show up once selecting a marker’s description. We also started to integrate the virtual communities into the app, now that it is working on the client’s webpage. Finally, we have decided to try creating a brand-new Wordpress page for our client to host the client webpage on.

We made some progress on the Project Binder for this project, filling out all relevant information for the Requirements, Analysis and Design. We will continue to work on this given feedback from Kevin.

In our final week of the project, we will finish off all the pieces of the app we are busy working on, including the Pop-out page and the virtual community page. We will also continue preparing the project binder and hand-off documentation, as well as prepare for the final Client Evaluation.

**Task Completion from Last Week**

* Created the Pop-out page, still working on populating data within it
* Continued to “polish” the client webpage
* Worked on setting up the Virtual Communities tab and integrating it with the database
* Monitored the release of the app to the Google Play Store, making changes if necessary, based on the approval process
* Started to finalize the Project Binder and any further documentation in anticipation of the project closeout and hand-off.

**Task for This Week**

* Finish the Pop-out page, having it filled with all relevant data
* Try to publish the client webpage on a new Wordpress site
* Finish integrating the Virtual Communities with the database
* Finalize the Project Binder and project hand-off

**Additional Information**

None this week.

**Known issues / things blocking progress**

None this week.