



OPSC7312 POE

Student Name: Karl Dicks
Student Number: 17667327
Course: BCAD3
Subject: OPSC7312
Lecturer: Nirasha Ramckurran
Assignment: POE
Due Date: 31/08/2020

Contents

Change Log	3
Introduction	3
Task 1 Improvements	4
Task 2 Improvements	5
Conclusion	6
Part A - Research	7
Introduction	7
Application Research	8
HERE WeGo – City Navigation	8
MAPS.ME – Offline maps, travel guides & navigation	14
Waze - GPS, Maps, Traffic Alerts & Live Navigation	20
Comparison	26
Feature List	29
Conclusion	31
Part B – Design	32
Application Overview	32
List of Functions	34
User Interface Design	36
Data Listing	54
Project Plan	56
Documentation	60
Introduction	60
Help File	62
Readme	66
Screenshots	69
Data Listing	86
Use Case Diagram	88
Preparation for Publishing the Application	89
Conclusion	91
References	92

Change Log

Introduction

As part of our OPSC7312 project, we were tasked with researching three existing mapping applications. The following information was provided in our Task 1 research document:

- Introduction
- Research of each mapping application, including:
 - Overview of the app.
 - Strengths and weaknesses of the app.
 - Understanding of how the application was implemented in Android Studio.
 - Screenshots of the app.
- A comparison of all three apps, including an infographic.
- A list of the best features of all the apps that you want to use in your final app.
- Conclusion.
- References.

The three applications which I assessed in my research document were:

- HERE WeGo – City Navigation
- MAPS.ME – Offline maps, travel guides & navigation
- Waze – GPS, Maps, Traffic Alerts & Live Navigation

Once the research document was completed, a design document was generated for our own application. This included the following information:

- Brief overview of your app including the app's innovative features.
- A list of functions, described in detail, that your app will perform.
- User interface design for each screen as well as navigation information between each screen.
- Detailed listing of the data that the app needs to capture from the user and store as well as how you will store this data.
- Project plan detailing deadlines and milestones for your project.
- References.

This information was used when developing our own mapping application.

For our POE submission, we were tasked with improving our Task 1 research document, and updating our design documentation. We were also tasked with improving our mapping application, and including additional features. Details of these two tasks are included in the following section.

Task 1 Improvements

The following feedback was received for my Task 1 submission:

Student	Dicks, Karl	
Student #	17667327	
Task 1	Max	Student
Intro & conclusion	10	8
App 1 research	10	9
App 2 research	10	9
App 3 research	10	9
Comparison	10	10
List of features	10	9
Innovation overview	10	8
List of functions	10	9
Data listing	10	9
Project plan	10	10
Total	100	90

Comments

Well researched and presented. Good choice of apps. Design done well.

I have made the following changes to my Task 1 assignment:

- Introduction and conclusion have been updated and improved (expanded on).
- Application 1 research improved and expanded upon.
- Application 2 research improved and expanded upon.
- Application 3 research improved and expanded upon.
- Grammar corrected.
- Spelling mistakes corrected.
- List of features expanded on to include new features in POE.
- Innovation overview expanded to include new features in POE.
- Design document updated, including the following changes:
 - List of features updated.
 - Screenshots updated.
 - Screenshot descriptions updated.
 - Data listing updated – inclusion of new fields implemented in POE.
 - Inclusion of additional features which have been implemented.
- References updated.
- All documents saved to PDF format, to ensure formatting is correct.

The above changes have been made to my Task 1 assignment, in order to update its content and improve it based on lecturer feedback.

Task 2 Improvements

The following feedback was received for my Task 2 submission:

Student Name	Dicks, Karl		
Student Number	17667327		
	Max	Student	Comments
App runs on emulator	10	10	
<u>Features</u>			
Login & reg	10	10	
Metric and imperial	10	10	
Storage / settings	10	10	
Landmarks / markers	10	10	
Route / mapping	10	10	
Time & distance / saving	10	10	
Interface	20	18	
Demo	10	10	
	100	98	login & reg / profile fully functional. Types of trans included. Map loads and markers work. Verifications added extras. Map styles added. Sms activity with location handled well. Demo link provided to youtube video. Demo done well with voice over. All reqs met. Docs and diagrams included. Pdf all docs

The following changes have been made to my Task 2 assignment (practical), in order to improve on the original design, and include required features for the POE submission:

- The application design has been improved, including:
 - Video renders have been used for login and registration pages – made in blender.
 - Button backgrounds have been changed (and transparency added).
 - Modal popup colours have been changed, and transparency has been added to some areas, where it does not impede clarity of text and functionality.
 - Text color has been changed to make it clearer, and look more professional.
 - Text spacing and font size was changed on some activities, in order to improve consistency.
 - Edit texts have been improved, by adding rounded corners, and a white background, which is semi-transparent, and icons have been added for the input fields (such as a lock for the password, etc). This makes it clearer for the users to understand inputs, and makes it look more professional.
 - Route history included (as it was in Task 2).
- The application has been built to an APK, and is ready for publishing on the Google Play Store – the process has been described in the documentation.
- Thorough testing was completed, in order to provide a high level of stability for the application on mobile devices running Android.
- The demo video has been updated to include additional functionality and improvements.
- Documentation has been updated, and converted to PDF to retain formatting.

The above changes have been made to my Task 2 assignment, in order to update its design and functionality based on lecturer feedback.

Conclusion

In conclusion, the above-mentioned changes have been implemented in my Task 1 and Task 2 assignments, in order to improve their quality. Additional functionality has been included in the practical submission, in order to complete all required functionality in the POE question paper, and the design of the application has also been improved.

Debugging has been carried out, in order to address application bugs and issues which were not obvious in Task 2. These issues have since been resolved, and changes have been made for the final POE submission.

The POE contains documentation for publishing the application to the play store, and the application has been built to an APK, so it can easily be installed on any Android device with an SDK of 23 or above.

The application is therefore ready to be published to the Google Play Store.

Part A - Research

Introduction

For our OPSC7312 module we have been tasked with developing a navigational application, for the Android platform, which provides the fastest route between the user's current location, and destination, and including a number of different transport methods, map themes, and unit conversion. In order to develop this application, research and planning must be completed, which will be included in this task.

Firstly, three applications were selected and researched, being the following:

- HERE WeGo – City Navigation (by HERE Apps LLC)
- MAPS.ME – Offline maps, travel guides & navigation (by My.com B.V)
- Waze - GPS, Maps, Traffic Alerts & Live Navigation (Hindy, 2019)

Research was conducted in order to better understand the market requirements and use case for these applications. My application will have to meet certain criteria, such as fulfilling all requirements in the POE document, and additional features that would make the application stand out from the competition – it will have to be innovative.

Research of the above three applications will include the following:

- Overview of the application.
 - Strengths and Weaknesses of the application.
 - How I think the application was implemented based on my knowledge of Android Studio App development.
 - Screenshots of the application, including detailed descriptions for each page.
- (The Independent Institute of Education, 2020)

In addition to this research, a comparison of features will be provided, in order to identify distinct differences between the three applications. After a comparison has been made, a list of features will be developed, which will include the features implemented in our application, during the course of our Task 2 and POE assignments. (The Independent Institute of Education, 2020)

My application will have to stand out in order to be competitive, and therefore a thorough analysis of existing applications has been completed, which details all innovative ideas, and how one might improve on these ideas, methods, and techniques in order to provide the most efficient and effective mapping application for the Android platform.

The following section contains research of the three existing applications, after this a design specification document will be provided, which details how I will implement my own mapping application in the Android Studio IDE. This document will detail how the application is to be developed, and provides the developer with a "roadmap" or specification details for how the application is to be developed.

Application Research

The following section contains research on the three chosen applications.

HERE WeGo – City Navigation

Overview

Here WeGo was originally designed by Nokia, and launched exclusively for Windows Phone smartphones. It has since been sold to a consortium of German car manufacturers, including BMW, Audi, and Mercedes, and has now been made available on mobile platforms including Android. (Bouckley, 2019)

It shares many characteristics with other map applications, such as Google Maps, however it has one unique feature which sets it apart from its competition. Here WeGo can completely replace your sat-nav device by downloading all map information locally, and use the GPS in your phone to locate your current position. (Bouckley, 2019)

While Google Maps allows users to download small amounts of map information locally, Here WeGo allows its users to download *all* maps for offline use. This means that users can completely disable their mobile connection, yet still have map navigation available if they have previously downloaded the map area to their phone. (Bouckley, 2019)

When the user travels to different countries this can be useful, as they do not have to have a constant connection to the internet, by using “collections” of previously downloaded areas before the trip. (One Trip at a Time, n.d.)

The application currently supports turn-by-turn navigation for over 100 countries.

Strengths and Weaknesses

Below are some of the strengths and weaknesses of the Here WeGo application.

Strengths

- The main benefit of this map application over its competitors, is its ability to store all maps locally, whether that is for a single area, city, state, province, or country. This means that the device does not have to have a cellular data connection at all times during navigation. Maps can be downloaded for 150 countries currently.
- Due to the maps being available for download, mobile data usage can be decreased. The user may wish to download maps using Wi-Fi at home, or use another cost-efficient method of internet connection, and then travel the route using only GPS connectivity (which is free).
- Drive mode provides information such as current speed, speed alerts and live traffic updates. This is very unique, as few competitors have implemented this system. This app turns your phone into a full Sat-Nav device.
- The application provides three map views, which can be used in combination, including:
 - Satellite: Offers a photo view of the area.
 - Transport: Offers metro rail, bus, and other forms of transport information.
 - Traffic: Offers real-time traffic information for 40 countries (which may be expanded in the future). This aspect does require a constant internet connection during navigation, and cannot be downloaded prior to departure.

- The application also provides night and day map views, which makes it more comfortable to view when driving at night.
- Here WeGo is free to use, and does not contain any advertisements – much the same as Google Maps and some other competitors. (Bouckley, 2019)
- Map routing is processed prior to departure, in order to provide the best route possible for different modes of transport (driving, walking, and transit). It includes other information, such as the estimated time of arrival and alternate routes – also similar to competitors. (Bouckley, 2019)

Weaknesses

- Offline maps, if used, can consume large amounts of local storage and bandwidth. For example, the United Kingdom map is 600MB, which could be costly when using mobile data. It may also consume too much storage on your mobile device.
- This application is not integrated with other internet sources. For example, Google Maps allows its users to search for a place of interest, view reviews, contact information, open hours, and more. Here WeGo provides basic information, but no reviews section, open hours, etc.
- Here WeGo provides a Sat-Nav experience, and therefore does not provide Street-View images like Google Maps does. It is more concerned about getting you from point A to point B as efficiently as possible.
- Maps are out of date – testing in Durban shows buildings which have been re-developed in the past few years. In contrast, Google maps is up to date. (Bouckley, 2019)

Implementation

The below is my understanding of how Here WeGo was implemented, based on my knowledge of the Android Studio IDE.

Back End

The back end was custom designed by the company HERE, which provides their SDK and API for developers, much like Google Maps and MapBox does. The Here WeGo application has based their mapping engine on their own SDK and API, from the same company.

Google Maps provides a similar solution, where they provide a “Google Maps” application, and then the API and SDK for external developers and charges them a subscription or usage fee.

Here WeGo developers most likely included these implementations and repositories within the application Gradle, but use an internal access key – the users of this application do not have any limit as to how many times they can call the HERE API. They have unlimited access from within the Here WeGo application, whereas external developers will have an account with HERE, and pay per API call. (Bouckley, 2019)

Front End

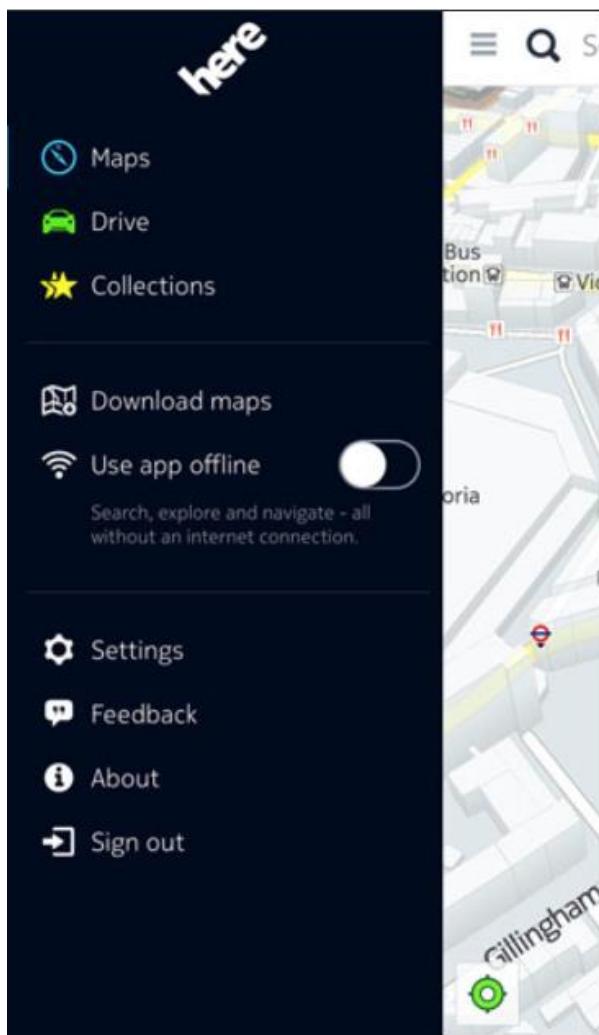
The Here WeGo application uses a nav-drawer design, with features and options available on the slide-out navigational pane from the left of the phone. This also includes sign in / out and settings buttons. These are included on a default app nav-drawer project, which can easily be generated within the Android Studio IDE.

The application uses fragments, and not separate activities. Transition effects between activities indicate that fragments have been used. (Bouckley, 2019)

Screenshots

Once loaded, the application will display your current location on the map. The user may proceed to click the menu button (hamburger icon), which will display the following interface.

Nav-Drawer



“Maps” is the home screen (the map view with the current user location).

“Drive” navigates the user back to the map, but in Sat-Nav view, where travelling speed, heading, and distance travelled is shown.

“Collections” allows the user to pin certain points of interest on the map.

“Download maps” allows the user to download maps for offline navigation.

Currently downloaded maps will be listed in this page, and can be deleted if needed.

“Use app offline” toggle switch enables or disables data usage. Essentially a flight mode for the Here WeGo app. This prevents users from running up large data bills.

“Settings” provides application settings, such as displayed units, voice options, day/night switch, and speed alerts.

“Feedback”, “About”, and “Sign out” are self-explanatory application features and information. (Bouckley, 2019)

(Bouckley, 2019)

Maps View



This is the home screen, which the application defaults to, when loaded. It shows the current location of the user, along with heading (compass), and the direction in which the user is heading.

“Search” provides an input field in which the user can enter search terms for places of interest. For example, they may wish to enter a street address, which will then be displayed on the map.

The route icon (to the right of the search field) allows the user to enter a place of interest and determine the best route to the destination.

Collections (the button to the right of the top nav bar) shows the saved “pins” for the user. The locations which are frequently visited by the user.

The menu button (hamburger icon) is displayed on all screens, as the developers have used fragments when developing the application. (Bouckley, 2019)

(Bouckley, 2019)

Sat-Nav View



(Bouckley, 2019)

The Here WeGo Sat-Nav view is shown in this image. It shows current location, user heading, and points of interest along the route.

The blue line indicates the intended travel route.

Distance to the destination is displayed on the lower left portion of the screen, and additional information is also provided when a route is entered, such as speed of travel, estimated time of arrival, and traffic (not shown). (Bouckley, 2019)

Saved Maps

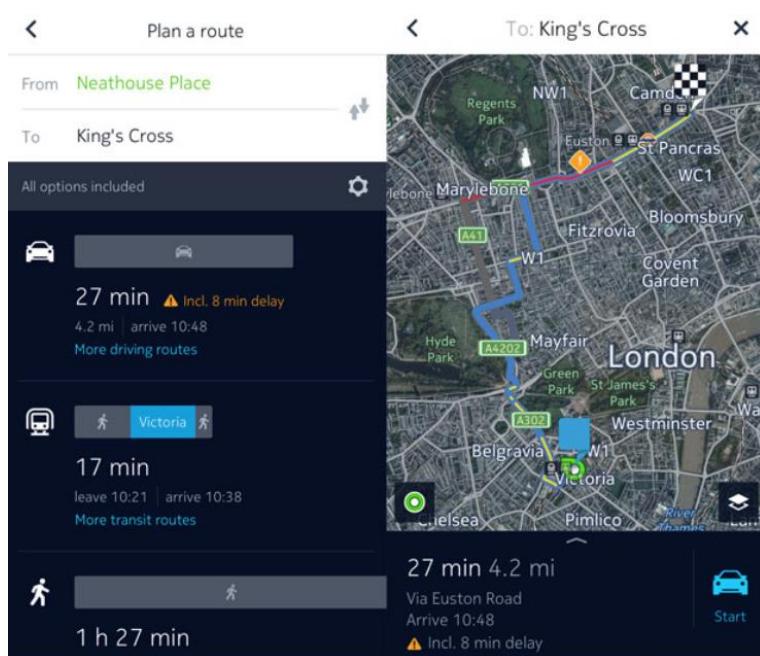
<	Download maps	<	France	
Africa	1,662.4 MB	>	France Whole area, 1,053.5 MB	
Asia	8,086.1 MB	>	Alsace 128.9 MB	
Australia/Oceania	785.7 MB	>	Aquitaine 170.6 MB	
Europe	9,315.6 MB	>	Auvergne 143.3 MB	
North and Central America	6,590.7 MB	>	Brittany 149.8 MB	
South America	1,743.5 MB	>	Burgundy 136.1 MB	
			Center 159.7 MB	
			Champagne-Ardenne 135 MB	

(Bouckley, 2019)

The view on the left displays saved maps, which are stored on the device.

The view on the right displays maps which can be downloaded. You can either download an entire country map or individual provinces, states, or oblasts. (Bouckley, 2019)

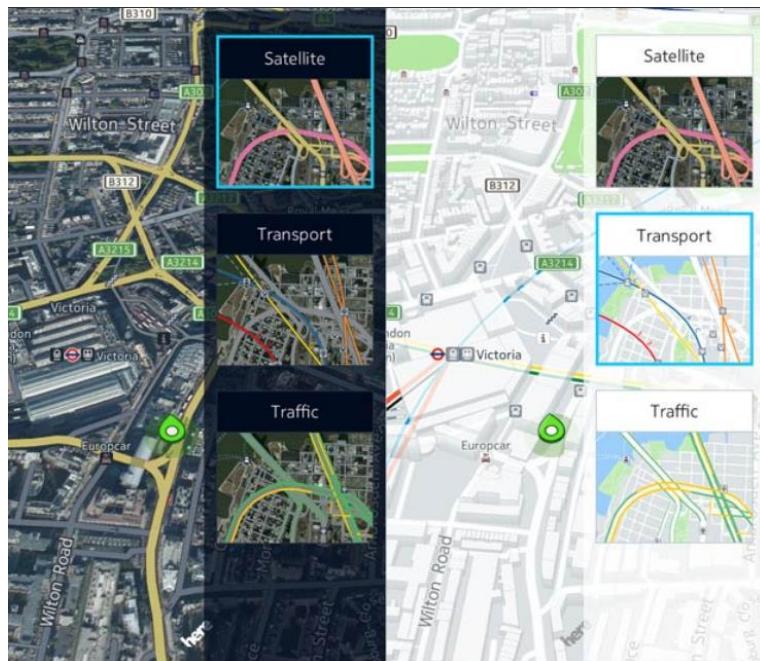
Routes



(Bouckley, 2019)

The routes view shows route information. This includes different modes of transportation, time of arrival, delays, and distance. (Bouckley, 2019)

Map Style



(Bouckley, 2019)

The map style can be changed within the application.

The light / dark theme will automatically change when the sun sets or rises. (Bouckley, 2019)

App Summary

In summary, Here WeGo application provides an excellent Sat-Nav replacement, however it does lack some extra features of the widely respected Google Maps. In addition to this, Here API is used by many car manufacturers, so it is likely your own car uses Here services, if your car has a Sat-Nav system. It is therefore highly respected as one of the best Sat-Nav services available on the market. (Bouckley, 2019)

MAPS.ME – Offline maps, travel guides & navigation

Overview

Maps.Me is a simple navigational application, based on the OpenStreetMap API. This API is open source, so anyone can edit and update map information. It is essentially the Wikipedia for map information. (Million Mile Secrets, 2016)

It only allows access to offline maps, unlike traditional map applications which allow for both online and offline use. Once the application loads, it will find your location and download the surrounding area information for the first time. For example, when starting the application in Durban, KZN it downloads the latest OpenStreetMap information for the province of KZN.

If you then move to another part of the map, it will ask if you would like to download more detailed information for that area. (Million Mile Secrets, 2016)

The application has a minimalist user-interface, which is user friendly and well-optimized. Maps.Me is free to use, however it does contain banner advertisements. In order to remove advertisements, a subscription fee is charged yearly. (Million Mile Secrets, 2016)

The application is innovative in that users can search offline maps. (Hoch, n.d.)

Strengths and Weaknesses

Below are some of the strengths and weaknesses of the Maps.Me application.

Strengths

- This application is based around the idea of it being completely offline. Map information is downloaded before it is used for navigational purposes, and therefore a constant connection is not required.
- Mobile data usage is reduced, as only GPS is used during navigation (if the area has been downloaded prior to departure). This will reduce cellular data costs, and can be used in areas that do not have cellular coverage.
- Maps.Me provides an open source platform on which users can add points of interest, such as restaurants, shopping malls, and other areas.
- It provides bookmarking of locations. The user can add a location to their bookmarks, which can be easily found again.
- Users can share their location with family and friends even if they do not have the Maps.Me application installed. This open-source mindset sets it apart from its competitors, where they would like everyone to use one particular map platform.
- The application provides search functionality and highly ranked places of interest in your surrounding area. For example, restaurants and malls are added to the highly ranked places of interest.
- Provides simple turn-by-turn navigation, including travel speed, estimated time duration, and distance travelled. The application will find the best route based on mode of transport, which includes walking, driving, and many more.
- Maps.Me is very easy to use, due to its simplistic user interface design. It does not provide extra functionality, which Google Maps does, but does provide the essentials, with the added benefit of reducing mobile data usage. (Million Mile Secrets, 2016)

Weaknesses

- Due to the application being open source, points of interest and other such information is not validated before being added to the map. This has increased the amount of inaccurate or incomplete information being added to the platform.
- The maps are not frequently updated, and do not include nearly as much information as its competitors (such as Waze and Google Maps). Each business owner is responsible for adding their own information to the map.
- No street view is available, unlike Google and Apple Maps – a feature used by many users.
- Unlike Google Maps or Here WeGo, the user must download an entire province or state at a time. For example, when the application first loaded it required the user to download the entire KZN province, which was 23MB.
- Advertisements are seen throughout the application, which are distracting. Maps.Me is however free to use.
The subscription fee to remove the advertisements is high (over R700).
- The user interface is easy to use; however, it appears that it has not been developed with fragments, but rather with separate activities. When switching between navigational elements, the animations could be improved by implementing a nav-drawer fragment design. (Million Mile Secrets, 2016)

Implementation

The below is my understanding of how Maps.Me was implemented, based on my knowledge of the Android Studio IDE.

Back End

The application makes use of the OpenStreetMap API for its maps and navigation system. Maps.Me has not developed their own maps API, so they have little control over how this is implemented in their application.

OpenStreetMap has been integrated within the Maps.Me Gradle files, and an external key or authentication method has been used. OpenStreetMap is a free alternative to Google Maps, and allows for commercial use of their mapping service.

Maps.Me does not have *any* form of registration or login from their side. They simply reroute users to OpenStreetMap, which allows users to sign up and edit map information on their freely available maps service.

In other words, Maps.Me is simply a front end to OpenStreetMap. (Million Mile Secrets, 2016)

Front End

The user interface has been designed using separate activities, causing animations between application components to be of lower quality than those of fragments.

There is a bottom navigation bar, which provides links to these separate activities. This presents a fractured design, which could be improved with the use of fragments.

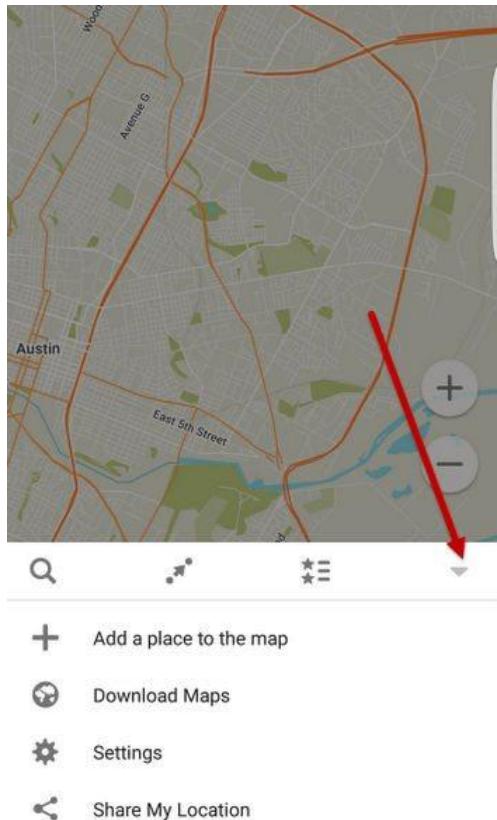
Due to this design method, the menu button (hamburger icon) cannot be seen in all activities, making navigation more difficult (the user must navigate back to the main menu activity each time they wish to switch to different activities). (Million Mile Secrets, 2016)

The back end Gradle would have been synced with OpenStreetMap implementations, and then the front end would have been designed on separate activities. The bottom navigation bar has then been set to launch these activities (screens). The “Back” button would have been enabled (by setting the “Parent Activity” within the manifest) on each activity, which brings the user back to the previously used activity. (Million Mile Secrets, 2016)

Screenshots

Once loaded, the application will display your current location on the map. The user may proceed to click the menu button (hamburger icon), which will display the following interface.

Nav-Drawer



This menu provides access to a number of key functions within the application.

Users may “Add a place to the map”, which allows them to pin places of interest to the map. This will be added for all users of Maps.Me and other OpenStreetMap users.

“Download Maps” allows users to download maps of other areas in the world. They can download country, state, province, oblast, or city maps.

“Settings” allows users to change measurement units, among other preferences.

“Share My Location” allows users to share their current location via social media, messages, or even email. (Million Mile Secrets, 2016)

(Million Mile Secrets, 2016)

Map View



Once the application loads, the map view will be displayed.

The user can search for points of interest by clicking the magnifying glass button. This includes restaurants, hotels, and other such locations.

The directions button provides a page where the user can input their current location, and where they would like to travel. The application will provide the best route possible, factoring in the mode of transport.

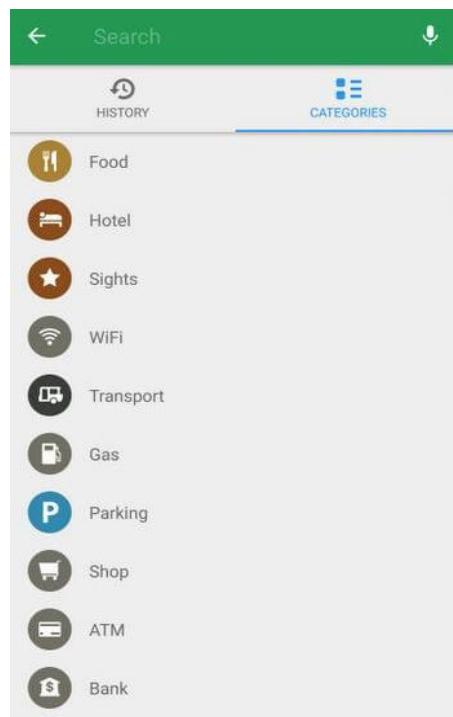
Bookmarks can be saved, much like in Google Maps and Here WeGo. The user can add a point of interest to their bookmarks page.

The user's current location will be found using GPS (triangulation). (Trimble, n.d.)

The map zoom level can be changed using buttons on the right side of the application user interface. (Million Mile Secrets, 2016)

(Million Mile Secrets, 2016)

Search View



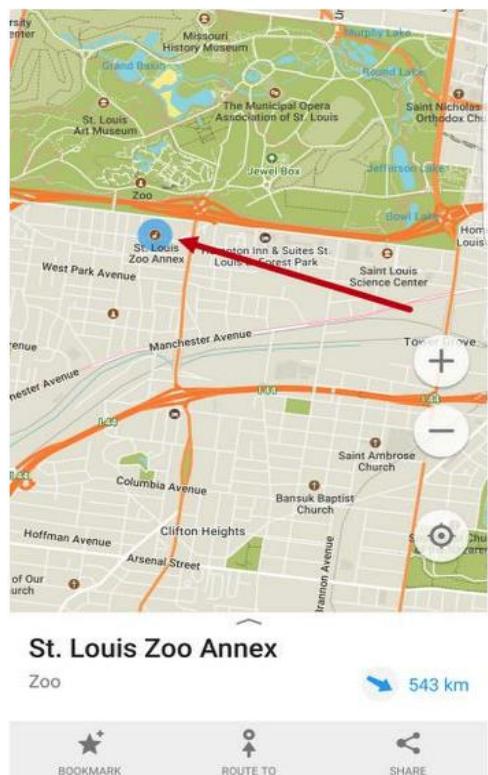
The user may search for points of interest, such as hotels, restaurants, shopping malls, and other such locations. They can search by category or simply enter a name or address in the search bar at the top of the page.

The user may also use speech to text to enter a search term.

The "History" tab provides previously searched for locations. (Million Mile Secrets, 2016)

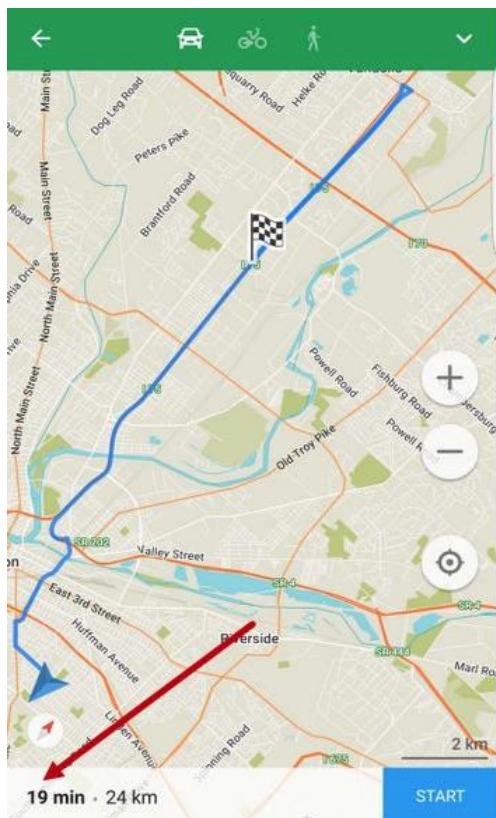
(Million Mile Secrets, 2016)

Location and Bookmarks



(Million Mile Secrets, 2016)

Directions



(Million Mile Secrets, 2016)

The user can click on a point of interest on the map, which will present a navigational bar at the bottom of the user interface.

The name of the location will be displayed, along with category, and distance.

The user can click the “Bookmark” button to add this location to their bookmarks.

The “Route To” button allows the user to obtain directions to the selected location. Once clicked, the user can choose their transport method, and preferred route.

The “Share” button allows the user to share the location of the selected point of interest via social media, messages, and email. (Million Mile Secrets, 2016)

Once a destination location has been selected, a route will be generated, based on the preferred method of transport. The estimated time duration and distance is calculated and displayed.

A blue line indicates the intended route.

The user can select “Start” which will provide turn-by-turn directions until the user reaches the destination address. (Million Mile Secrets, 2016)

App Summary

In summary, the Maps.Me application provides an easy to use user interface, however it lacks many features of the widely respected Google Maps. In addition to this, Maps.Me does not use its own map API, and rather relies on OpenStreetMap, which is an open source map service.

Users of Maps.Me can add points of interest to OpenStreetMap, which has resulted in inaccurate or outdated information in some cases.

The application is ideal for users who wish to download maps for offline use. Maps.Me does not provide online services, but rather downloads all requested map locations before use. This can result in lower mobile data usage, as it only requires GPS during navigation, which is free.

Waze - GPS, Maps, Traffic Alerts & Live Navigation

Overview

Waze application is focused on providing the most accurate source of traffic information for drivers, and navigating them using the most efficient routes possible. It was recently acquired by Google, which has been slowly integrating its system into Google Maps, as to increase its accuracy of real-time traffic information. (Sterling, 2019)

Waze is community-powered, which means that its users provide traffic alerts and other such information, which is shared with other Waze users for free.

This map application is not however a fully-fledged alternative to Google Maps. It only provides directions and traffic information for drivers, however this has been expanding in the last while and will soon allow for other modes of transport. (Abigail, 2020)

It does not provide additional information, such as 3D buildings and topographic information, which one would find on Google or Apple maps.

The application is free to use, and does not require the user to sign up in order to use it. Only if the user would like to share traffic information will they have to sign up to a Waze account.

What makes the Waze application unique, is that its data store of traffic information is constantly updating. This provides very accurate real-time traffic updates. (Cabebe & Goodwin, 2014)

Strengths and Weaknesses

Below are some of the strengths and weaknesses of the Waze application.

Strengths

- The Waze user base is large, resulting in support in many countries and increased accuracy of traffic information. The application is constantly collecting traffic information from its users, and therefore it can quickly adapt to ensure the fastest route possible is provided.
- The application is free to use, and does not require any form of login. However, in order to add traffic events or information, the user will have to sign up.
- Input fields can be entered via text or speech, so drivers can concentrate on driving, instead of inputting text in input fields while driving, which is dangerous.
- Turn-by-turn navigation includes text-to-speech street names. For example, it will say “Turn left on St Thomas Road”, instead of simply saying “Turn left at the next intersection”, which competitors often do.
- It provides social media integration (such as Facebook), so that you can share traffic information with friends and family more easily.
- Notifications are provided when users have reported slow traffic, incidents or accidents, and even police presence and speed cameras, along the route the user is taking, or in surrounding areas.
- Displays nearby businesses when the user is stopped, such as at a red light.
- Waze users can add favourite places to a list, which they can access from within the main menu. This is essentially a bookmark feature, much like Maps.Me and Here WeGo provide.
- It shows the current speed of the vehicle, and other Sat-Nav information.
- In addition to traffic information, road conditions (such as potholes), is presented on the map. (Beech, n.d.)

Weaknesses

- The user friendliness of Waze could be improved. Icons are small and sometimes hidden, which makes navigating the application difficult, especially when the user is driving.
Some users have described the layout as being cluttered and ugly. (Page, 2018)
- In areas with large populations, the number of icons on the screen can be overwhelming, which makes navigation more difficult. (Page, 2018)
- Users have noted that Waze can be distracting, as advertisements pop up, routes are re-routed, and social media integration cause friends to pop up on the screen. This could lead to accidents, and result in dangerous driving. (Page, 2018)
- High data and battery usage. Battery usage is similar to that of Google Maps when open, however background activity drains battery much more rapidly when using Waze, as it is continually reporting data to Waze. (Page, 2018)
- This application has decreased reporting accuracy when travelling at higher speeds. Therefore, city driving benefits more from this application. (Beech, n.d.)
- Traffic alerts can only be reported at the incident location, which is when you should be focusing on driving. Once you have a moment to enter the report, you are likely too far away from the traffic incident. (Beech, n.d.)

Implementation

The below is my understanding of how Waze was implemented, based on my knowledge of the Android Studio IDE.

Back End

Waze has created their application from the ground up, with its own map and traffic SDK. They have also created an API, which developers can use to integrate Waze traffic and maps into their own applications. They do however tightly control who has access to their maps and traffic information. A company which requests access must be assessed before access is granted. The official Waze app must have had this SDK integrated in the Gradle files.

Users of Waze can sign up using their phone number or Facebook account, unlike competitors who usually use email verification. Waze use three backend servers to store these details in different regions for improved performance and security. (Waze, n.d.)

Waze also provides integration with social media (Facebook), in order to determine where friends are located.

The map engine uses GPS to determine speed and location, and an internet source to retrieve map and traffic information. (Page, 2018)

Front End

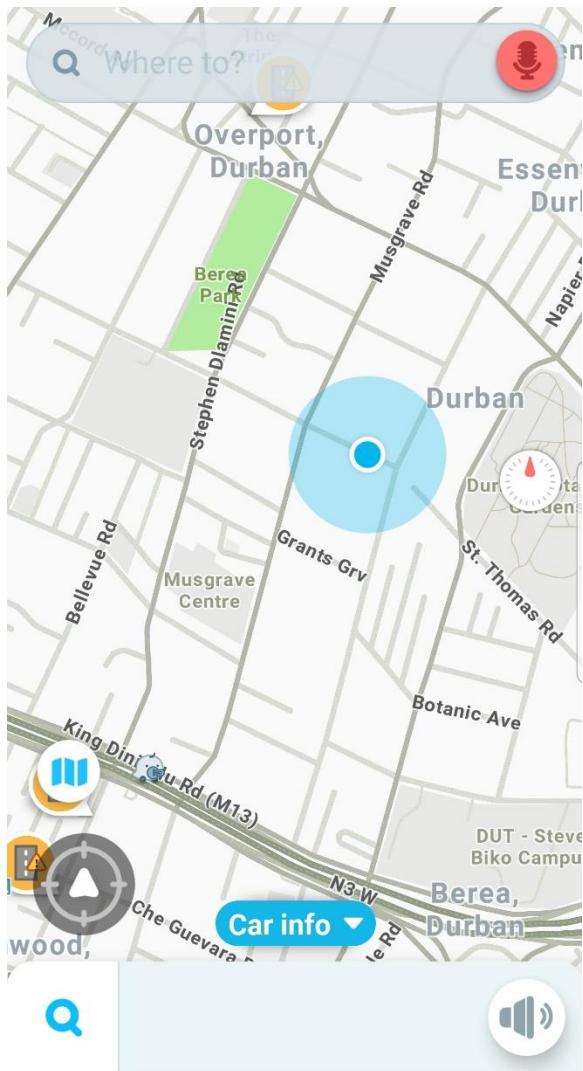
The Waze application uses a Nav-Drawer user interface design. However, it appears to have been custom designed, unlike Maps.Me which uses a standard theme on the Android Studio IDE. Waze appear to have put a lot of effort into designing the application user interface.

The front end of the system can be controlled using speech-to-text input or by using regular text input. Waze has likely used an already available open source speech-to-text API for user input. Waze also appears to have used multiple activities to display information, instead of fragments. The way in which one screen moves to another indicates this. (Page, 2018)

Screenshots

Once loaded, the application will display your current location on the map, and will provide the below functionality.

Map View



(Waze, n.d.)

When the application first loads, a map will be presented, which shows the current location of the user.

The home page is minimalistic, with only a few buttons available.

The user can enter a destination in the “Where to?” text box, or click the red microphone icon to use speech-to-text.

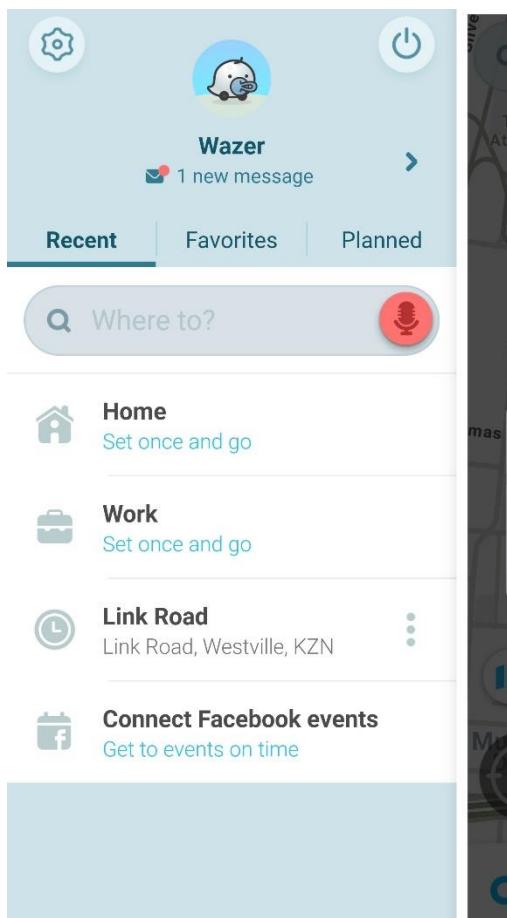
The map button (bottom left of the screen) allows the user to change display options, such as 2D or 3D maps, and color selection.

The arrow button returns the user back to their current location on the map.

Users can change their preferred mode of transport in the “Car info” dropdown list, which will affect how the system routes users.

The main menu can be accessed by pressing the lower left button (magnifying glass), and the volume/ voice can be changed by using the lower right button (speaker). (Waze, n.d.)

Menu



(Waze, n.d.)

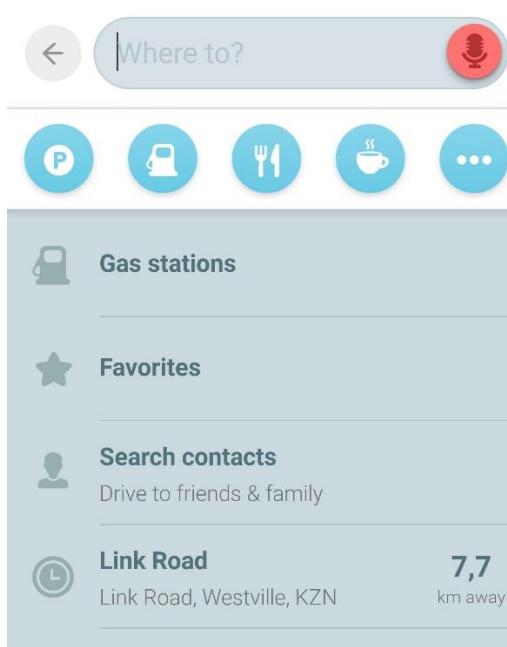
When the user presses the search button (magnifying glass), the following menu will be displayed.

The user can view recent location searches and set their home and work address.

Favorites allow users to bookmark certain locations.

Planned trips can be entered in the "Planned" menu. This will provide the user with notifications when they should leave, based on traffic and road conditions. (Waze, n.d.)

Search



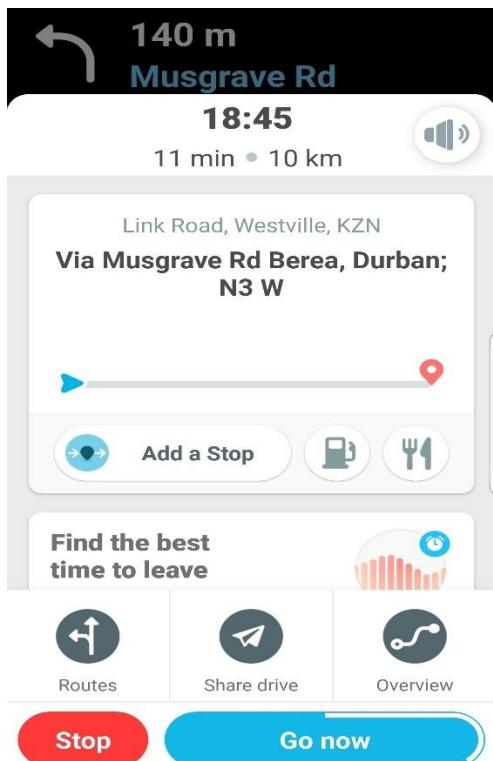
(Waze, n.d.)

When the user clicks on the "Where to?" search box, the following screen is presented.

The user can search for a certain address or point of interest, or they can navigate categories of locations in their area.

The user may type the address or location name, or use the speech-to-text function. (Waze, n.d.)

Navigation



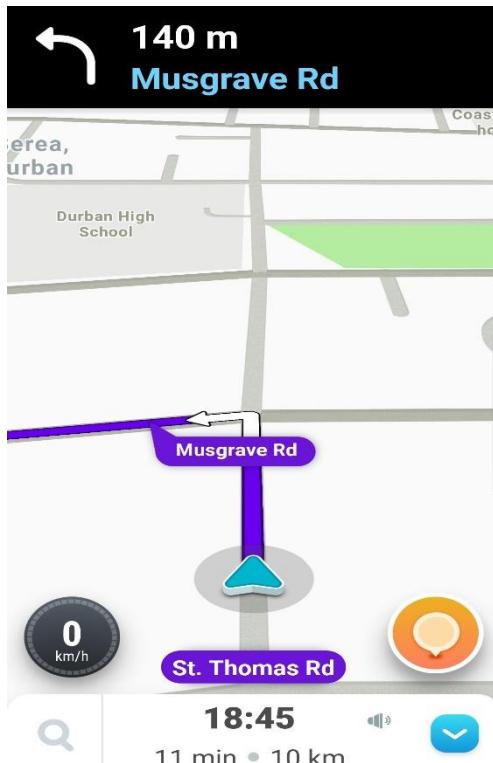
(Waze, n.d.)

Once the user enters a destination address, a trip overview will be presented.

The user can view the estimated time of arrival, travel time duration, distance, and other information on this page.

The user can also view different routes, share the directions, or view the overview for the route.
(Waze, n.d.)

Navigation Route



(Waze, n.d.)

Once the "Go now" button is pressed, or a timer has been reached, the navigation will begin.

The following information is provided:

- Next turn – directions.
- The current road you are travelling on.
- Current speed.
- Estimated time of arrival.
- Travel duration.
- Distance from destination.
- Report (Orange icon)

The report button allows users to report traffic incidents, such as accidents, police presence, and more.

Turn-by-turn navigation is provided with a voice-over. (Waze, n.d.)

App Summary

In summary, the Waze application provides an excellent source of traffic-related information, which is constantly updated by an ever-increasing user base. In addition to this, turn-by-turn navigation has been implemented in a way that makes use of this powerful user-base to re-route the user in order to make driving as efficient as possible.

Users of Waze can add traffic incidents, accidents, speed cameras, and police presence on the Waze app.

There is however a pitfall to the current system, in that the application only allows users to submit reports at the scene.

The application is ideal for users who frequently travel in areas where there are closed roads, high traffic, and accidents, such as in large cities. Waze does not provide large amounts of surrounding information, such as shops, petrol stations, and other such locations.

This application has been designed to get people from point A to B as efficiently as possible.
(Waze, n.d.)

Comparison

Summary

	Here WeGo	Maps.Me	Waze
GPS support	✓	✓	✓
Offline map support	✓	✓	✗
Supports multiple modes of transport	✓	✓	✗
Can bookmark locations	✓	✓	✓
Supports text-to-speech road names.	✗	✗	✓
Free to use	✓	✓	✓
Advertisements Present	✗	✓	✓
Community-based traffic reporting	✗	✗	✓
Speech-to-text input – built in.	✗	✓	✓
Uses in-house maps SDK	✓	✗	✓
Calculates current travelling speed	✓	✓	✓
Full public transit support (train times, etc)	✓	✗	✗
Street view support	✗	✗	✗
Automatic map downloads	✓	✗	✓
Location reviews and information	✓	✓	✓

Comparison - Continued

Each of the three map applications provide basic navigation functionality, however there are some distinct differences between them. These differences lead to users preferring certain applications over others.

One of the major differences between the three applications is that Waze only supports car, taxi, or motorcycle navigation. When compared to Maps.Me and Here WeGo, it lacks support for those users who use public transport, walk, or cycle to their destination. Waze was developed solely for drivers, and its main feature is that it has access to large amounts of traffic information, which is collected from its large user base. (Beech, n.d.)

The Waze application does not provide any form of offline support, where the other two applications do support this feature. Waze therefore uses much more mobile data during navigation, which can become costly.

Due to the constant background activity of this application, it uses more battery power than its competitors (such as Google Maps). (Beech, n.d.)

All three applications use GPS to determine the user's location, however the map and traffic overlay is implemented differently in each of the three applications. Waze downloads map and traffic information as the user moves from one location to the next, much the same as Here WeGo. (Beech, n.d.) Maps.Me differs from this model, as it only allows users to obtain directions once they have downloaded areas that they will be travelling through. (Million Mile Secrets, 2016) Traffic information is then streamed once navigation is started (only available in 38 countries). (Photo, 2016)

Here WeGo supports offline maps, as mentioned above, but traffic information can be streamed from their servers during navigation if requested. (Bouckley, 2019)

All three applications can calculate the current speed of travel by using the phone's GPS.

One of the key features of Waze, is that it supports text-to-speech, so road names are spoken during navigation. Their competitors do not support this functionality. For example, Waze will say "Turn left on St Thomas Road", instead of saying "Turn left in 600 meters". This makes navigation much easier for travellers. Maps.Me and Here WeGo do not support this feature. (Beech, n.d.)

All three applications are free to use, and do not require the user to sign up an account. Here WeGo does not include advertisements, unlike Maps.Me and Waze. (Bouckley, 2019)

Waze shows advertisements when the car is at rest (such as at a stop street). It will show advertisements of local stores, hotels, and other such points of interest. (Beech, n.d.)

Maps.Me shows banner advertisements in its main menu. (Million Mile Secrets, 2016)

Waze allows its users to sign up and report traffic incidents, unlike the other two applications. (Cabebe & Goodwin, 2014)

Maps.Me allows its users to sign up to an OpenStreetMap account, and add points of interest to the open source map. (Million Mile Secrets, 2016)

Here WeGo provides a bookmark section, which is stored on a user account. Its users can sign up an account and view their bookmarks within the application, which has been implemented in all three of the researched applications. (Bouckley, 2019)

Maps.Me and Waze support speech-to-text natively, whereas Here WeGo does not. Here WeGo can still however use the google assistant speech-to-text feature. This allows users to use speech to enter text fields, instead of typing in search terms using the phone keyboard. (Bouckley, 2019)

Here WeGo has full metro transit support, including departure times and navigation. For example, train departure times are found when selecting a metro/ train station. Both Maps.Me and Waze do not support this feature. (Bouckley, 2019)

All three applications do not support street-view, unlike competitors such as Google and Apple Maps.

These applications are mainly used for navigation, much like a Sat-Nav device would provide. They display current travelling speed, and other such information which Google and Apple Maps do not provide.

Another difference between the applications, is the use of in-house map and traffic SDK's. Waze and Here WeGo have developed their own SDKs, which are very powerful, and used by developers through an API. Maps.Me on the other hand has not developed their own SDK, and rather depends on an open source maps SDK (OpenStreetMap). This means that Maps.Me has little control over mapping information and how it is presented to its users. Maps.Me is essentially a front end to OpenStreetMap, and provides a minimalistic user interface. (Million Mile Secrets, 2016)

Infographic

Infographic link: <https://create.piktochart.com/output/44887928-opsc7312>

The Infographic has also been included in the OPSC7312 – Karl Dicks 17667327 folder.

Feature List

My final application will include the following features (The Independent Institute of Education, 2020):

- **Signup / Login:**

The user will be able to sign up an account and log in to the application in order to access the mapping application.

They can edit personal information, set the map theme, system preferences such as unit conversion (metric or imperial), and mode of transport. This information will be stored in an online hosted database – Firebase Auth for login credentials and Firebase Firestore for preferences.

This is also a great security measure, as nobody but the rightful user can view or edit their preferences, trip logs, or personal information.

- **Set Preferences**

The user will be able to change displayed units from the metric to imperial system, as well as their preferred mode of transport (car, walking, or public transport).

By changing the mode of transport, the application will find the most efficient route possible for the chosen transport mode. For example, the route will not include highways or roads that do not allow pedestrians, if the mode of transport is set to walking.

In addition to this, the user can set their preferred map theme, from a number of different available themes (dark mode, traffic, etc), and their full name.

- **Set Map Theme**

The user may set their preferred map theme to show traffic on a dark map theme, or any other available map theme from the MapBox SDK. This allows users to personalize their view of the map, which will be stored in their online account, and can be accessed on multiple different devices.

- **Share Location**

The user will be able to send their location, encoded in a google maps link, to any cell phone number. This allows users to easily share their current location with others, from within the application.

- **Set Destination Waypoint**

The user will be able to set a waypoint on the map, by dropping a pin on the location in the map view – by pressing on the phone screen or using the search feature.

Once this is done, a route will be generated by the integrated MapBox service, and retrieved and displayed on the phone screen.

- **Search for Location**

The user will be able to search for a location by address, text phrase, or other input. This will allow the user to search for landmarks, instead of clicking on a location on the map interface.

- **Calculate Best Route**

The application will be able to determine the best (most efficient) route between the user's current location and the destination address. Preferred modes of transport and traffic incidents will determine how the application routes the user.

- **Navigation**

The application will be able to display the route visually, and with audio prompts. It will also display the estimated time duration, and distance to the destination address, along with other navigational information.

- **Map**

The application will be able to display a map, which will include the route overlay, if a pin is dropped on the destination address and a route returned from MapBox. The map will include street names, points of interest, and all additional information from the integrated MapBox API.

- **Record Routes**

The application will be able to record all the trips that the user takes, and store them in the database. These trips will be stored as JSON objects.

- **Route History**

The application will be able to retrieve stored trips and present them on the map interface. This is essentially a trip log for all trips that the user has taken. The user will be able to scroll through the trip log, and select any trip that they wish to view.

- **Clear Map**

The user will be able to clear all waypoints and navigation routes when they press a "Clear Map" button. This essentially refreshes the map.

Conclusion

The research phase of development has provided insights into three well respected and used mapping applications which are currently available on the Google Play store. All key features of the applications were assessed, and compared with each other, which provided a great way of generating a list of features which I would like to implement in my task two and three submissions.

The three applications were similar in that they were all navigation applications, however they each had unique features which set them apart from the competition. These unique features, along with user feedback was assessed in order to understand if these features should be implemented in my own mapping application, and if so, how they should be implemented.

This research assisted the developer in understanding the current market for mapping applications on the Android platform, and how the developer could potentially leverage innovative features to compete with well-established and known mapping applications, such as the applications which were researched.

With this research completed, a design document will be generated which will use key features identified during the research phase. By completing research into these applications, it has also helped to identify strengths and weaknesses of the applications. It was beneficial to understand these aspects before creating my own navigation application.

Weaknesses can be addressed, and a solution could be made to resolve these flaws or inefficiencies.

I would also be able to replicate or draw inspiration from some of the well-liked features (strengths) in my own submission – therefore this research phase was necessary to understand which features are currently out there, and how they could be improved upon.

Research has also provided a way for us to compare user interface designs, and to determine how we would best implement our application user interface, in order to promote efficiency, and also attract users with a professional yet unique design.

The screenshots section, which provided images of the user interfaces of the applications, and descriptions of their contents was beneficial as we gained insight into how these three application were developed, and how we could improve or replicate their designs.

A comparison section further highlighted the differences between these three navigation applications, which assisted us in understanding which features we may want to implement in our own solution.

The feature list section included all features which will have to be developed for our Task 2 and POE submissions, including additional innovative features which will also be implemented to make the application stand out from the competition.

Part B – Design

Application Overview

I have developed a map and navigation application called One Direction, which provides similar functionality to Maps.Me, and is also implemented in a similar fashion.

One Direction uses the free MapBox service, which has been integrated in much the same way as Maps.Me integrated the OpenStreetMap SDK. (Million Mile Secrets, 2016)

One Direction provides a map and route service for free, which allows its users to set a waypoint, and obtain visual and audio navigational feedback during their trip.

When the user first loads the application, the map service will triangulate the current user location, using the phone's GPS function. (Trimble, n.d.) After the user location has been triangulated, the user can set a waypoint by clicking a map location, which will then start the navigation process.

The map service will then determine the best route, based on the preferred mode of transport and traffic, and display it to the user.

If the user clicks the “Start Navigation” button, then the route overlay will be provided, along with visual and audio navigation.

The user has the choice of different modes of transport, including driving, walking, and cycling. The map service will determine the best (most efficient) route based on the transport method, and traffic.

For example, if the user sets their preferred method of transport to walking, it will avoid highways and roads that do not allow pedestrians. It will also increase the estimated time of arrival, due to it being a slower method of transport.

The user may also set their preferred units of measure (imperial or metric).

Visual feedback will be presented in the form of a map, with the following information provided (The Independent Institute of Education, 2020):

- Current user location (including street name)
- Current user heading
- Estimated travel time
- Estimated time of arrival
- Distance remaining (in metric or imperial units)
- Visual directions (turns / offramps)
- Mute / Unmute switch for audio navigation
- Report – the user can report road closures, etc. This information will be stored by MapBox to improve routing – similar to OpenStreetMap.
- Route overview, including original location, and destination location. The current user location is also provided.
- Detailed routing view, including all turns and offramps (and road names).

In addition to the above visual feedback, the application will provide voice navigation of upcoming turns, offramps, and other such useful information.

Trip logs will be provided for all trips that the user has made. These will be accessible from a “Route History” page. The user will only be able to view their own trips, stored on their user profile, securing the user’s personal information.

The user can view all their trips, which are stored with a timestamp and transport method on the “Route History” page. They will be able to scroll down and view more logs, if they have multiple entries.

Once, or if, a route entry is pressed, the route will be displayed on the map and the modal popup will disappear, providing an easy way for the user to view their past navigation logs.

List of Functions

One Direction will provide numerous navigational functions, which will assist its users in getting from point A to point B in the most time efficient manner possible. The below list of functions was developed with the question paper requirements in mind. (The Independent Institute of Education, 2020)

- **Sign Up/ Login:**

The user will have to sign up an account and log in to the application in order to save preferences and edit personal information. This information will be stored in an online database (Firebase). It will use both Firebase Authentication, and Firestore in order store authentication and user preferences.

The registration form will request that the user enter their full name, email address, a password, and set their preferred units of measure (imperial or metric).

The login process will authenticate users based on their email address and password – stored in Firebase Authentication.

- **Set Preferences**

The user will be able to change their preferred units of measure from metric to imperial, as well as preferred mode of transport – which includes walking, cycling, and driving.

When the user initially registers an account with the application, their preferred mode of transport will automatically be set to driving. However, they can change this on the user preferences page.

Users may edit their personal information, preferred method of transport, and units (metric / imperial) on this preferences page.

In addition to this, the user can set their preferred map theme, from a number of different available themes (dark mode, traffic, etc), and their full name.

- **Set Map Theme**

The user may set their preferred map theme to show traffic on a dark map theme, or any other available map theme from the MapBox SDK. This allows users to personalize their view of the map, which will be stored in their online account, and can be accessed on multiple different devices.

- **Set Destination Waypoint**

The user will be able to set a waypoint on the map, by dropping a pin on the map – tapping on the screen. This action will trigger the MapBox routing SDK to find the best possible route from the current user location to the destination address, factoring in the user's preferred method of transport and traffic conditions.

- **Search for Location**

The user will be able to search for a location by address, text phrase, or other input. This will allow the user to search for landmarks, instead of clicking on a location on the map interface.

- **Share Location**

The user will be able to send their location, encoded in a google maps link, to any cell phone number. This allows users to easily share their current location with others, from within the application.

- **Calculate Best Route**

The MapBox SDK will determine the best (fastest) route between the user's current location and the destination address, once a pin has been dropped on the map.

User preferences will determine how the navigation engine works. For example, the user's preferred mode of transport will affect which roads will be used, or avoided, when routing occurs.

Once the routing process has started, and the MapBox SDK has returned a valid route, a button ("Start Navigation") will be enabled and change colour. The user can then view the route overview on the map, when or if they press this button.

- **Route Information**

One Direction will be able to visually display navigation information, which will include the following:

- Current user location (including the street name)
- Current user heading
- Estimated travel time
- Estimated time of arrival
- Distance remaining (in metric or imperial units)
- Visual directions (turns / offramps)
- Mute / Unmute switch for audio navigation
- Report – the user can report road closures, etc. This information will be stored by MapBox to improve routing – similar to OpenStreetMap.
- Route overview, including original location, and destination location. The current user location is also provided.
- Detailed route view, including all turns and offramps (and road names).

This Information will be available visually, and in addition to this, audio navigation prompts will provide assistance when travelling.

- **Map**

One Direction will be able to display routes visually, on the phone screen. This map information will be pulled from the MapBox SDK, on which the One Direction application has been based.

The map includes points of interest, such as public parks, tourist attractions, and universities.

The map also includes all road names and walking paths, which will be used when generating routes.

- **Record Routes**

One Direction will be able to record all the trips that the user takes, and store them in the database. These trips will be stored as JSON objects.

- **Route History**

One Direction will be able to retrieve stored trips and present them on the map interface. This is essentially a trip log for all trips that the user has taken.

The user will be able to scroll through the trip log, and select any trip that they wish to view.

- **Clear Map**

One Direction will allow users to clear existing routes, by pressing a “Refresh Map” button from the main menu.

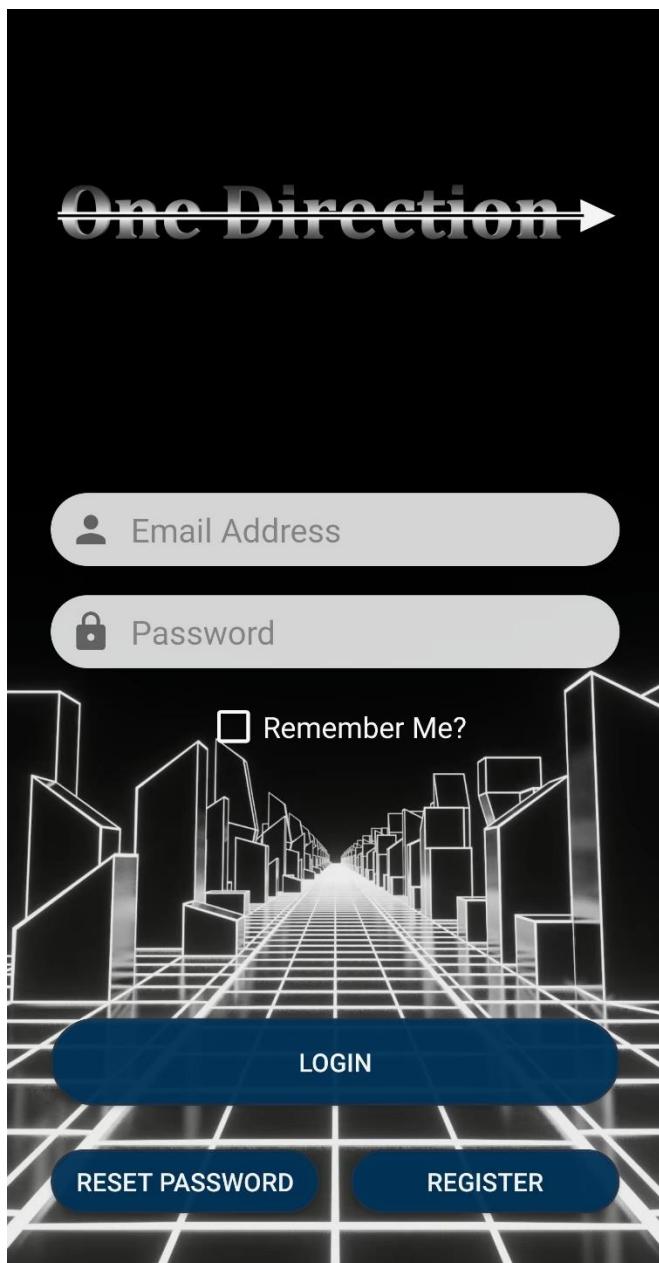
This will reset the map, clear all waypoints, and move the map location back to the current user location.

The above features are to be implemented in our Task 2 and POE submissions.

User Interface Design

The user interface for One Direction has been designed, and all functionality has been implemented. Below is the interface for my proposed map and route application:

Login



Once the application has loaded, the user will be presented with the login page.

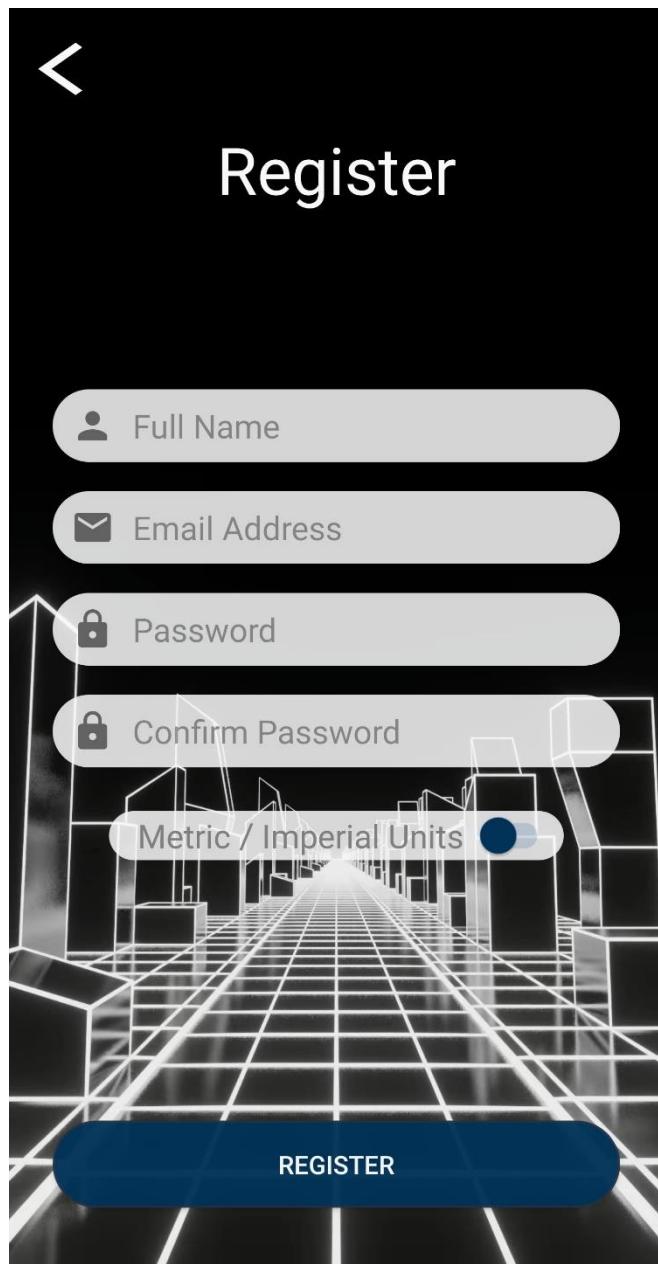
User authentication will occur once the user enters their unique email and password, and presses the “LOGIN” button.

The login process is asynchronous, which means that there will be a delay when contacting the Firebase data store. During this time interval, a pink spinner will be displayed to let the user know that this process is currently occurring in the background.

The user may register a One Direction account by pressing the “REGISTER” button, which will bring them to the register page (shown on the following page), or reset their password by clicking the “RESET PASSWORD” button, which will send the user a link via email to reset their password.

The user can also select the “Remember Me?” checkbox, which will remember the user credentials, and automatically log the user in each time they open the application.

Register



When the user presses the “REGISTER” button on the login page, they will be presented with a register form (shown on the provided image).

They will need to enter the following information:

- Full Name
- Email Address
- Password
- Confirm Password
- Preferred Units of Measure

Input validation has been implemented in the following ways:

- Passwords Must Match
- Minimum Password Length is 6 Characters.
- No Input Fields Can be Null
- Email Address Must Conform to Regular Expression – Must be Valid.
- Email Address Cannot be Currently in Use by Another Account.

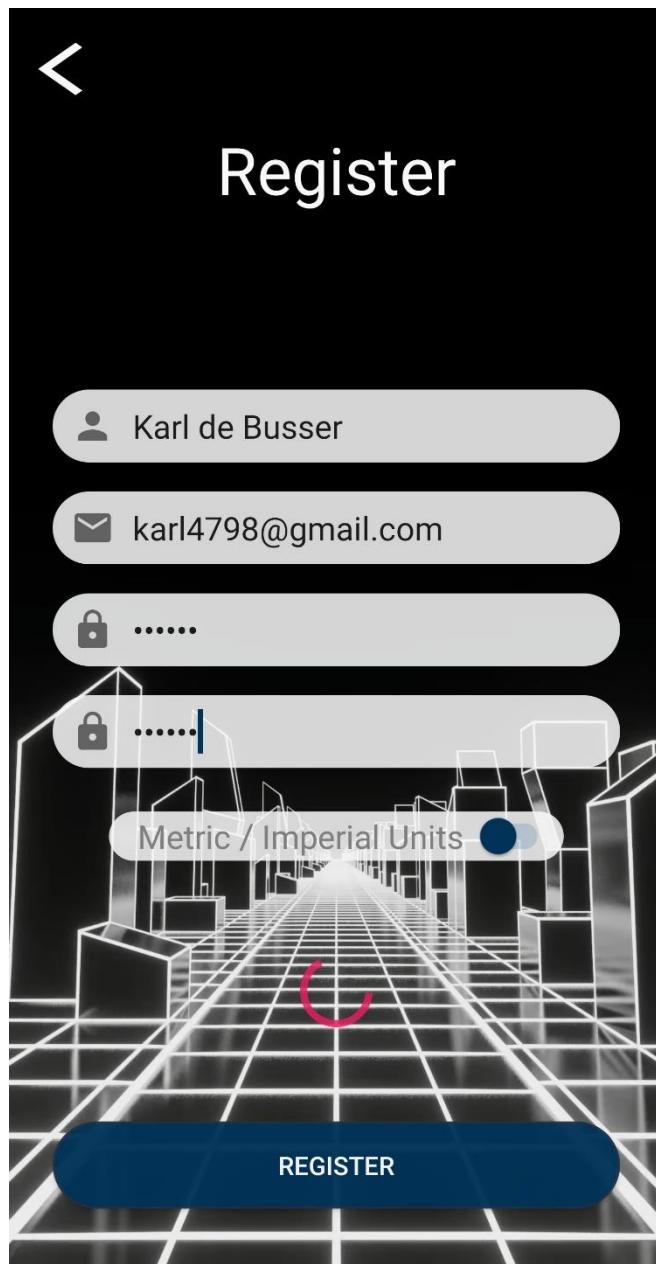
This validation is completed to ensure that garbage, duplicate, or malicious values are not persisted in the data store.

Once the user presses the “REGISTER” button, validation will be processed, and if this is passed, the user account will be created.

After the user account has been created, the user will be sent a verification link, via the email address they provided in the registration form. Once verified, they can sign in, and will be presented with the home screen (the map view).

If the user wishes to return to the login page, they can press the left arrow “<” button, which will navigate the user back to the login page.

Account Created



Once the user has pressed the “REGISTER” button, their account will be created (as shown).

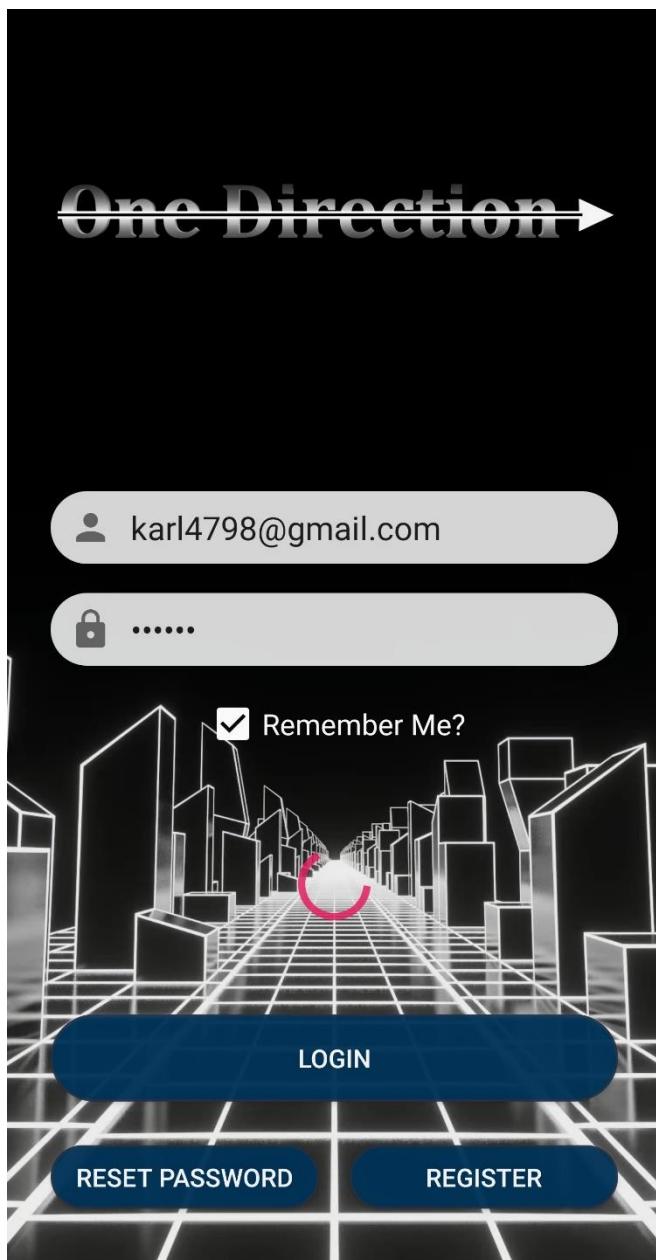
The image on this page provides an example of how account creation will appear.

The spinner will be presented (and turn) during this asynchronous event.

If the phone cannot reach Firebase due to network or other transient errors, a toast will be displayed. This message will provide error details.

If validation fails, then a toast message will describe how the user can fix the incorrect values.

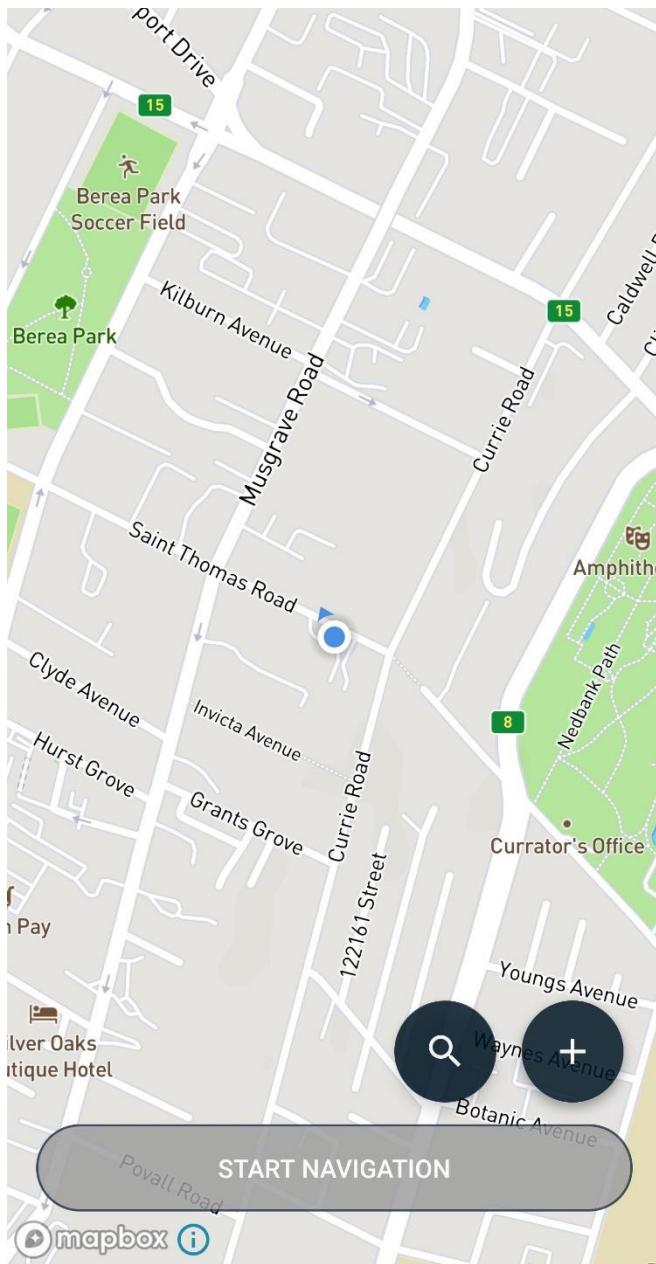
Successful Login



Once the account has been verified, by clicking the link in the email provided, the user can log into their new account on One Direction.

This login process is asynchronous, and therefore a pink progress bar is provided, indicating that the authentication process is occurring in the background.

Home



Once the user has logged in, they will be presented with the home screen.

This contains the following features:

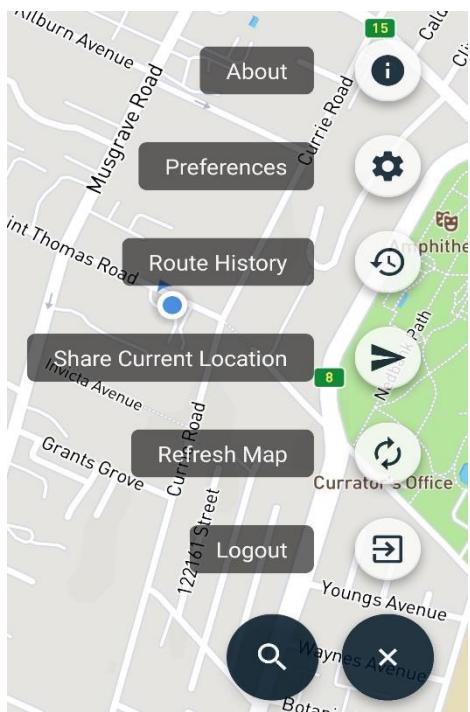
- A Detailed Map
- Current Location on the Map
- Start Navigation Button, which is Disabled by Default.
- Search Button.
- Menu Button (Floating Action Button).

The “START NAVIGATION” button is disabled by default, and only once the user drops a pin on the map, will it become enabled.

The user can drop a pin on the destination address by moving the map to the destination, and pressing the screen of the device.

The user can access the main menu of One Direction, by pressing the Floating Action Button (Pink + Symbol). The menu is displayed on the following page.

Menu



Once the user has pressed the menu (Floating Action Button), additional buttons will appear.

The following actions can be triggered from this menu:

- About
- Preferences
- Route History
- Share Current Location
- Refresh Map
- Logout

A detailed description of these actions is provided below.

About

This action simply brings the user to the about page, which shows application information, such as the version number, functions, and a link to a YouTube video.

Preferences

This action results in a modal popup being displayed (shown on the next page), which allows the user to set their preferences, including personal details and system preferences, such as preferred units of measure, map theme, and mode of transport.

Route History

This action results in a modal popup being displayed, which allows the user to view all previous trips. Route history logs can be selected, which will pull the JSON object from the database, and show the route taken on the map interface.

Share Current Location

This action results in an edit text being displayed, where a user can enter their phone number and send their current location to the recipient phone number.

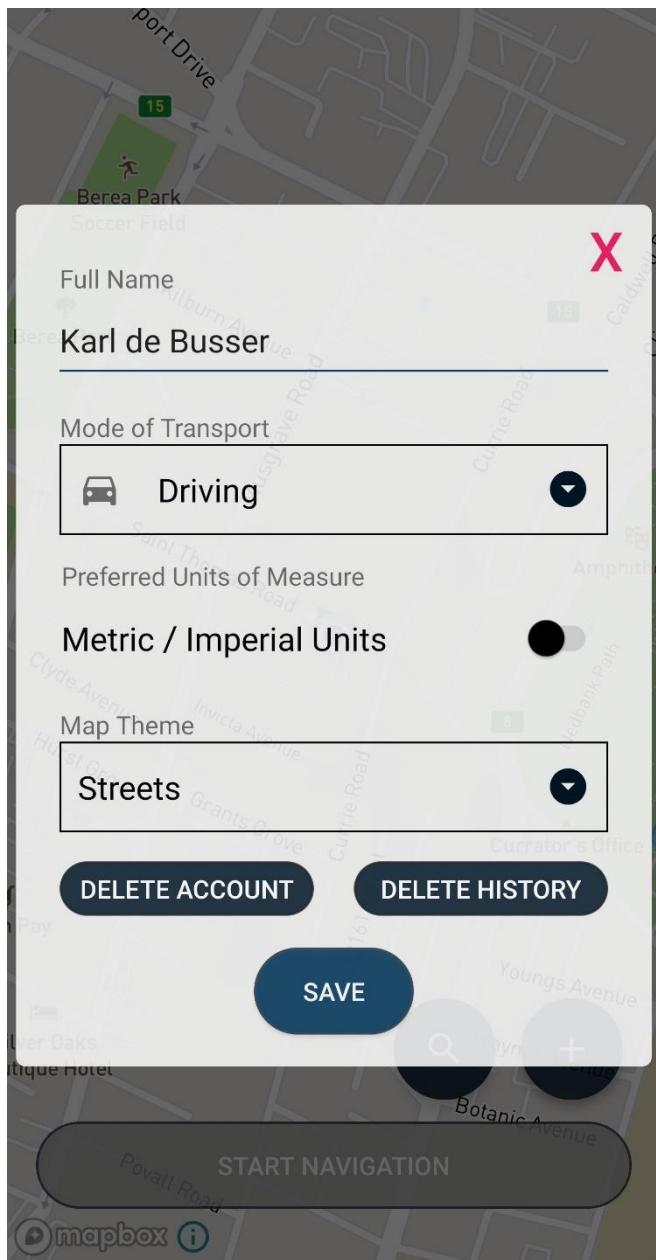
Refresh Map

This action results in the set destination address, and navigation route, being cleared from the map. It refreshes the map and re-triangulates the current user location, along with their set preferences.

Logout

The user may log out of their account by pressing this button. This action will end the session, so even if they press the return button on the phone, they will not be able to log in again without providing their unique login credentials.

Preferences



Once the “Preferences” button is pressed, a modal popup will be displayed, which includes the following information:

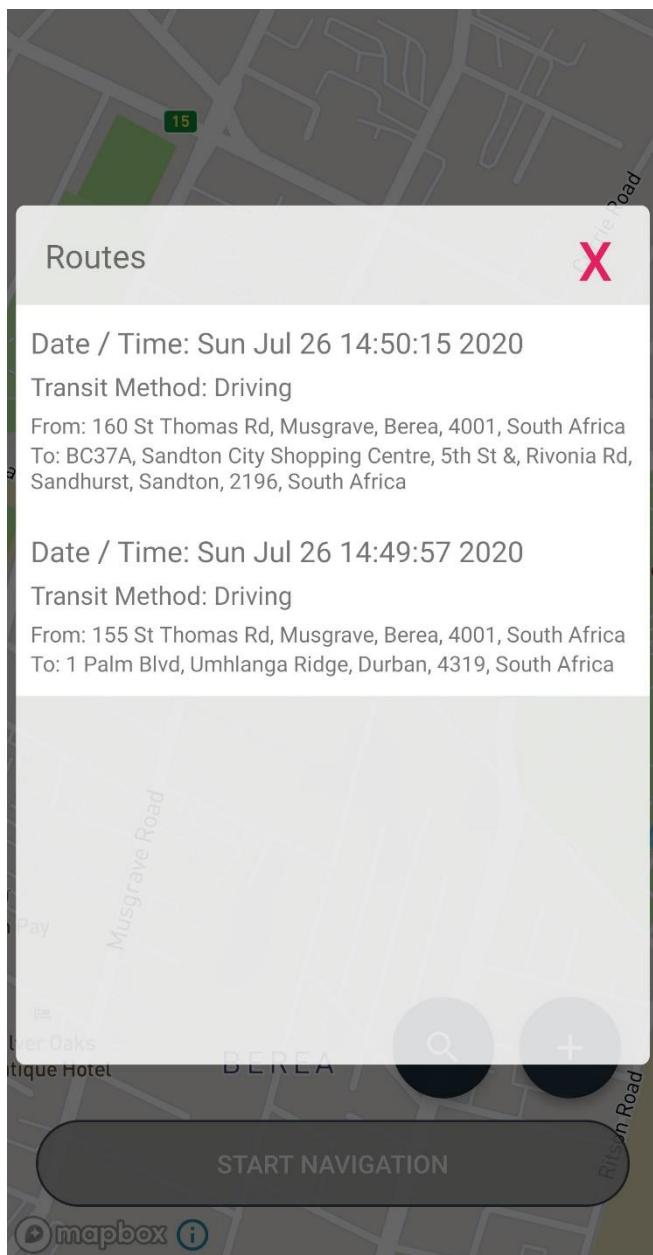
- Full Name of the Logged in User
- Preferred Method of Transport
- Preferred Units of Measure
- Preferred Map Theme

The user may change these values by pressing on the respective edit text, spinner, or switch button.

For example, the user may change their preferred mode of transport to Walking, Cycling, or Driving. They may also set their preferred units of measure from Metric to Imperial, or vice versa.

Once changed, the user can press the “SAVE” button to save the preferences to their Firestore profile.

Route History

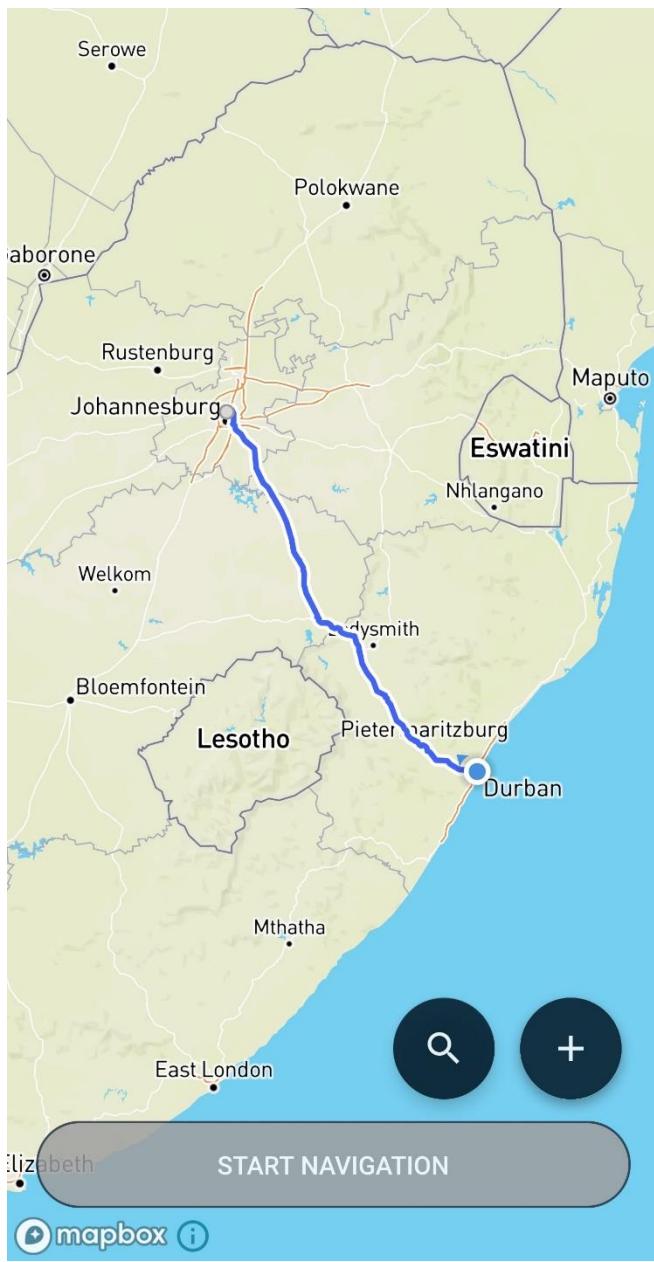


Once the “Route History” button is pressed, a modal popup will be displayed, which includes the following information:

- Method of Transport
- Date and time of the trip

The user can click any of the available items on the list, which will draw the route on the map, and close the modal popup window (shown on the next page).

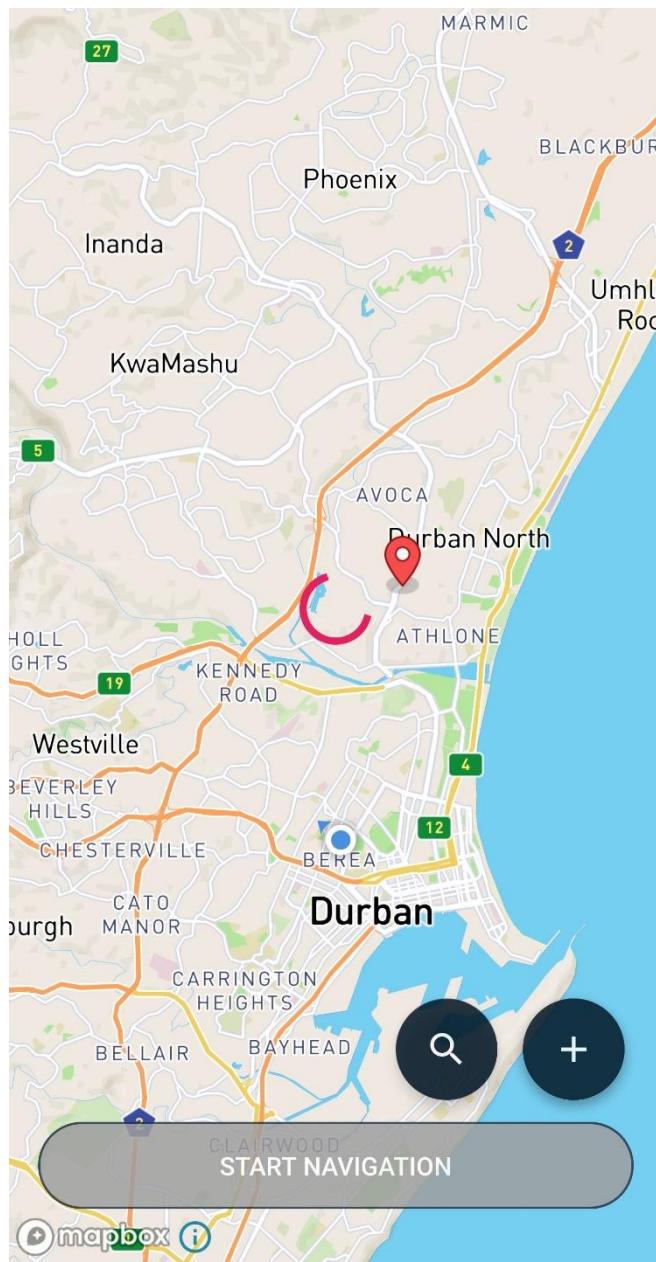
Saved Route



Once the user presses an available route log item, the route will be drawn on the map.

This map will include the beginning and end points of the route, and all navigation turns that were required for the chosen route.

Finding Route



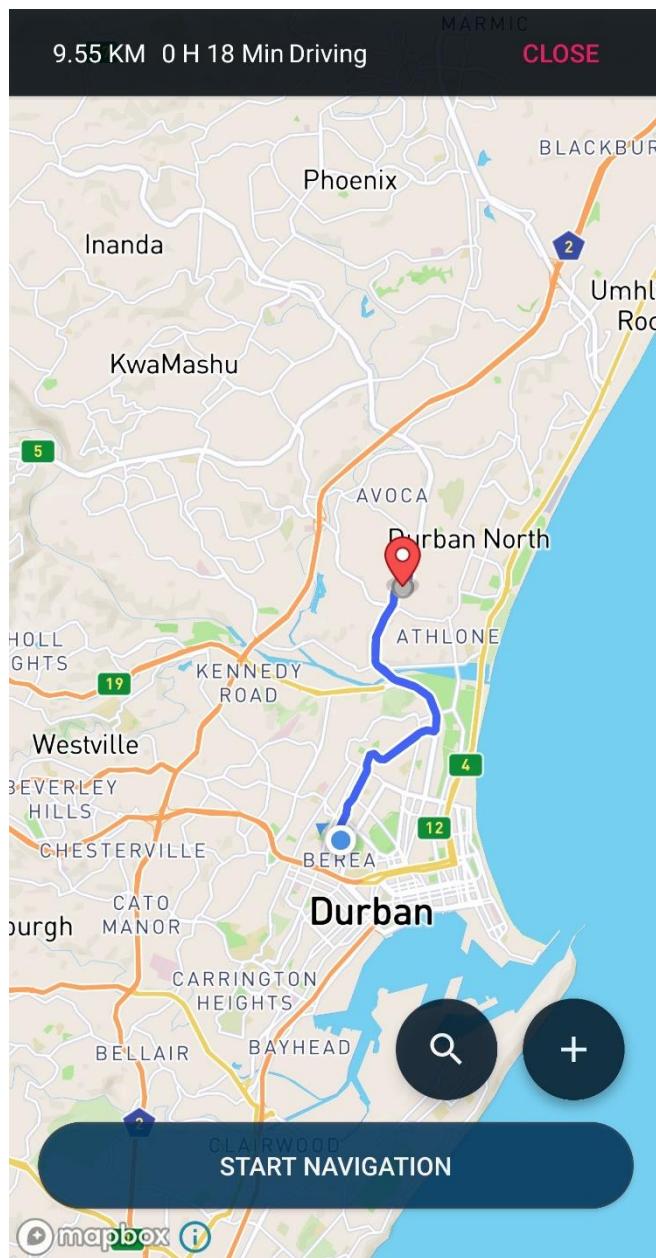
When the user drops a pin on the map (by pressing on a location on the map), a route will be generated.

The MapBox SDK will take into consideration the user's preferred mode of transport. For example, if the preferred mode of transport is walking, then the routing will avoid highways.

The picture provided on this page displays the processing of the route, which is an asynchronous process and is completed by MapBox integration.

The next page provides an overview of what a route would look like, once generated and retrieved from MapBox.

Generated Route

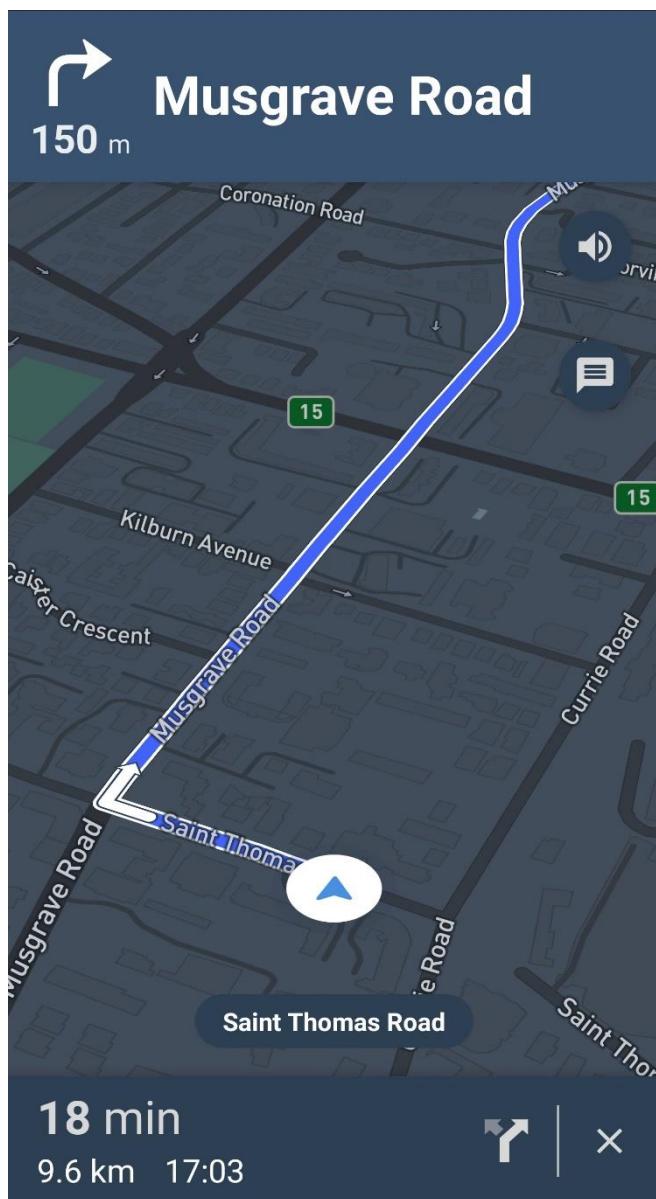


Once the route has been generated, the “START NAVIGATION” button will be enabled, and change color to green – signalling that the route has been generated and that they can start the navigation process.

An overview of the route will be displayed on the map, which will also provide the original and destination locations.

The user can now press the “START NAVIGATION” button and retrieve visual and audio navigational information (shown on the next page).

Navigation



When the user presses the “START NAVIGATION” button, visual and audio navigational prompts will provide turn-by-turn navigation until the user reaches their destination.

The following information will be provided on this page:

- Next Road Turn
- Distance Until Turn
- Turn Direction
- Current User Location
- Current Road
- Travel Time Duration
- Travel Distance
- Estimated Time of Arrival

In addition to this visual feedback, audio navigation will also be provided, much like Google Maps and their competitors.

This includes distance to offramps, turns, and other such navigational information.

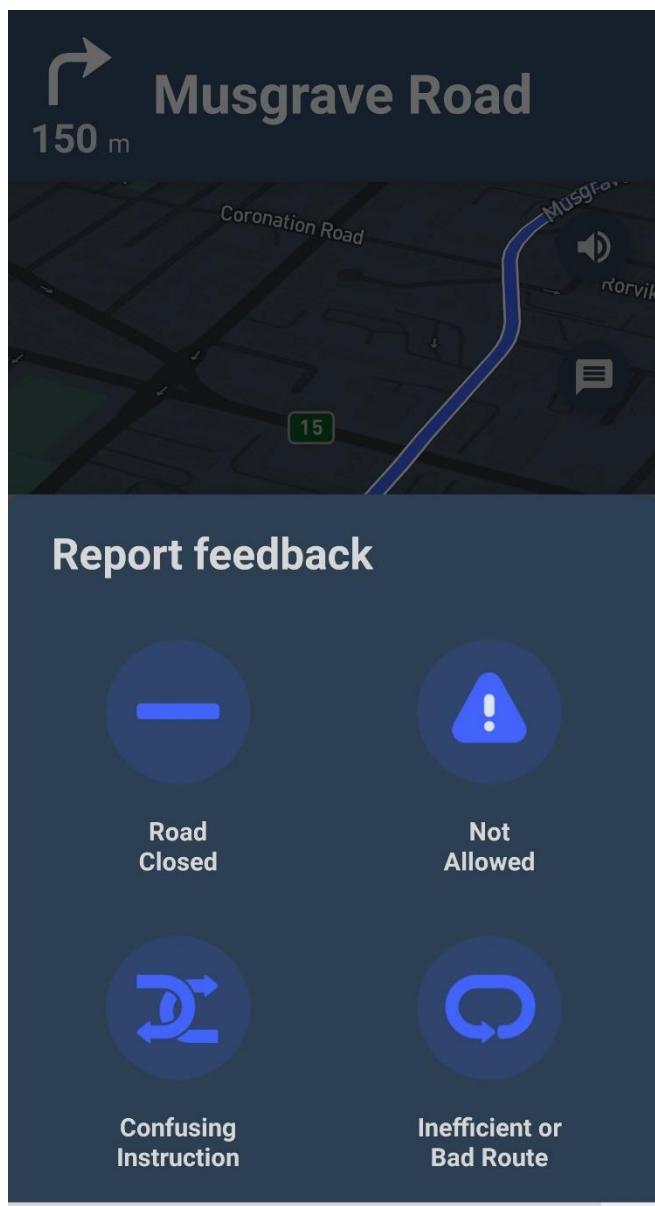
This audio can be muted by pressing the mute button (top right of the screen).

Some additional features present on this page include:

- Reports: Allows users to report road closures, traffic incidents, map issues, and other such information. This is sent to MapBox, and will be used to better navigate its users.
- Detailed Navigation: A detailed listing of turns, offramps, and the distance from the current user location is available.
- Route Overview: The user can view the route overview.
- Stop Navigation: The user can stop navigation by pressing the X button at the bottom right of the screen.

Details of the above information is provided in the following pages.

Reports



During navigation, the user can press a “reports” button (below the mute / unmute button).

When the user selects this button, they are able to select any of the displayed report options.

For example, the user can report a road closure, which will help other MapBox users, as it will re-route them around the road closure or incident.

This “reports” page automatically hides after a few seconds, as shown in the very bottom of the image – the progress bar.

Detailed Navigation

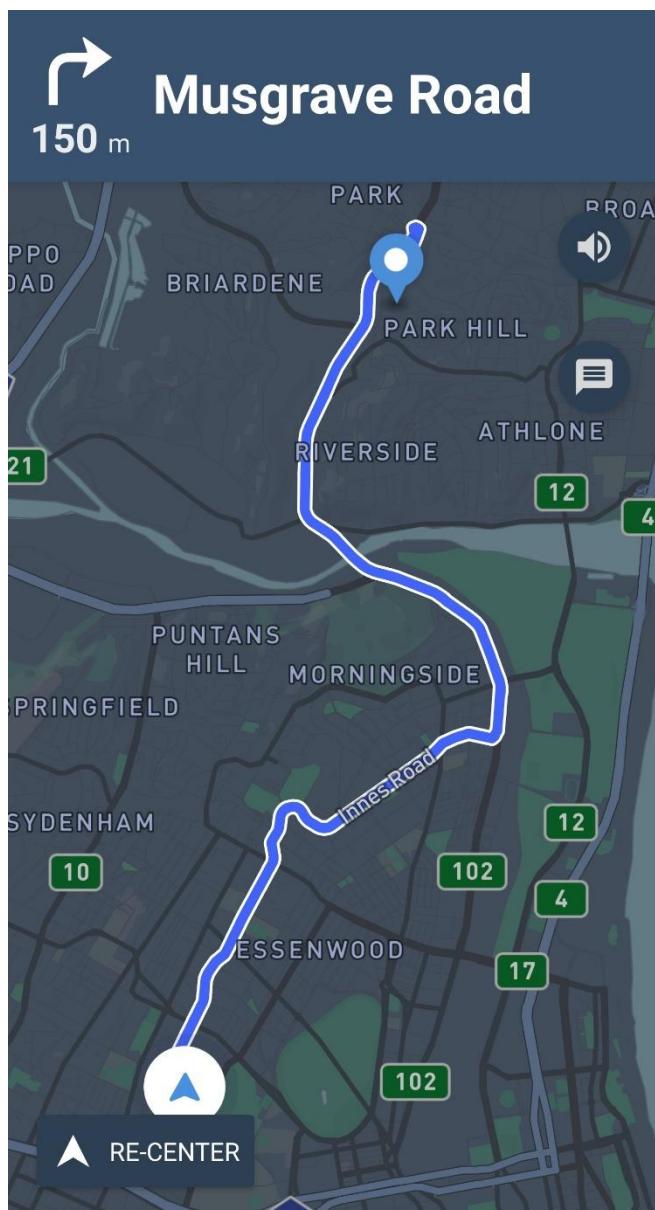


During navigation, the user can press the upcoming turn (topmost banner).

When the user presses this banner button, a list of all turns, offramps onramps, and distance between them is provided.

In addition to this, turn information is provided. For example, in this case the user should take a right turn onto Musgrave Road, towards Essenwood.

Route Overview

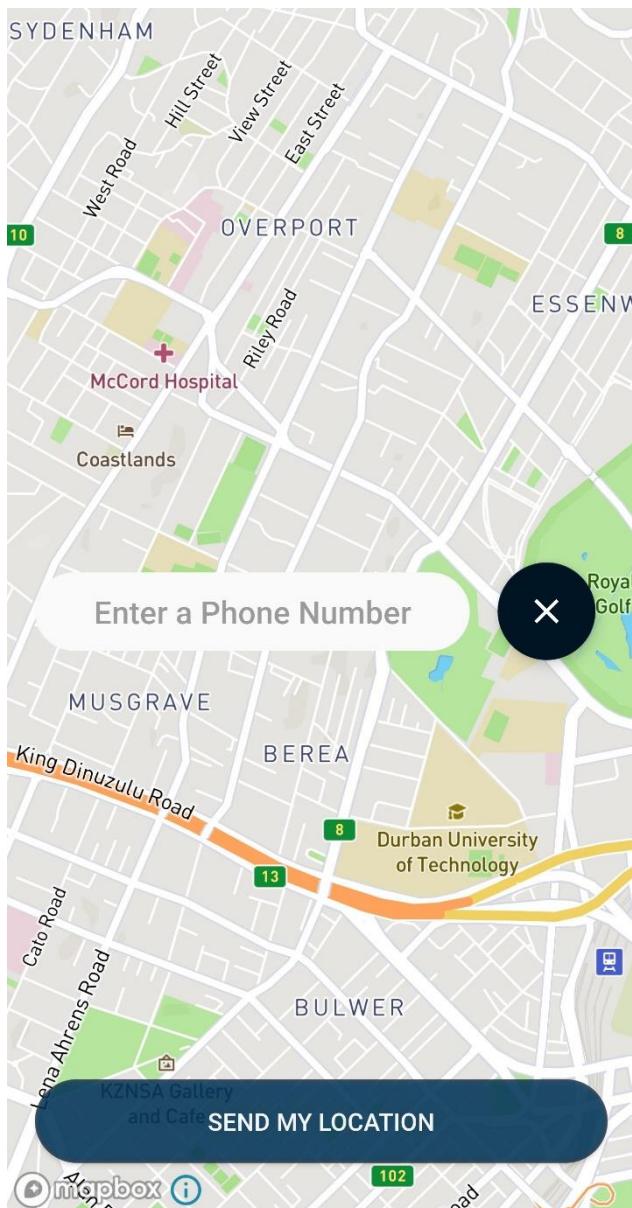


During navigation, the user can press a “route overview” button (to the left of the stop route button “X”).

When the user selects this button, the route overview is displayed.

The current user location and heading is included, along with the destination and calculated route.

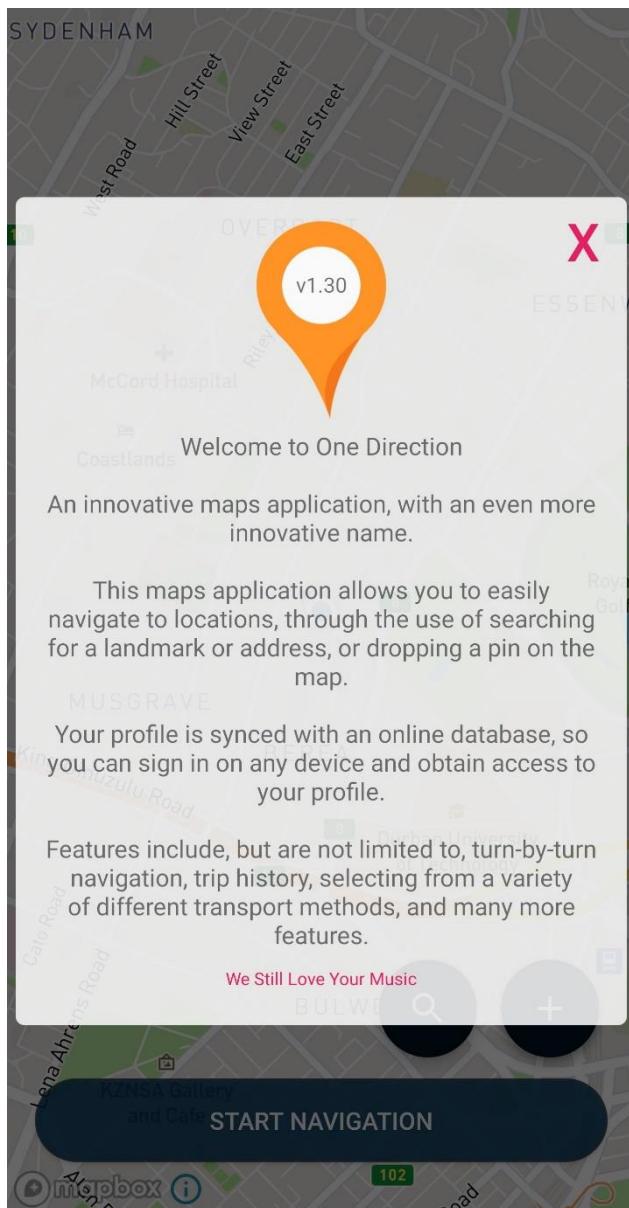
Share Current Location



The user may share their current location by navigating to the “Share Current Location” tab on the main menu, entering the recipient phone number, and clicking the “SEND MY LOCATION”, which will send an encoded google maps link to the recipient phone number as an SMS.

The user can cancel this operation by clicking the “X” button next to the edit text, or they can send the location by clicking the “SEND MY LOCATION”.

About



The user may view the “About” page within the application, by pressing the “About” action link in the main menu floating action button.

The About page provides the current version number of the application (currently v1.30), along with functionality, and a YouTube link to a video.

Data Listing

One Direction uses Firebase Firestore, Authentication, and SQL Lite services to store user profiles (authentication and preferences), and SQL Lite for trip logs – which can be played back at a later stage. The following information will be stored in the database.

Data Name	Type (Java)	Type (Database)	Nullable	Length
User UID	String	Text	False	Max
Email Address	String	Text	False	Max
Password	String	Text (Hashed)	False	Max
Full Name	String	Text	False	Max
Units of Measure	String	Text	False	Max
Transport Method	String	Text	False	Max
Map Theme	String	Text	False	Max
Trip Log	JSON	Text	True	Max

Some data will be calculated within the application, including:

Data Name	Type (Java)	Length
Distance to Destination	Double	Max
Estimated Travel Time	Double	Max
Estimated Time of Arrival	Double	Max

Some considerations have been taken during the implementation of the database, including:

- All information stored in the database (Firebase) will only be accessible by authorised users.
Each user will have their own account, and will only be able to view trip logs and preferences for their account, and not others – securing information.
- Email addresses which have been used to sign up must be unique, otherwise Firebase will throw an exception, which will be caught in a Toast message. This error message will inform the user that there is already an account created with the provided email address.
The account email address must also be verified before use.
- Route information will be cleared from the asynchronous MapBox session when, or if, the user presses the “Refresh Map” button.
- The user may delete their trip history or entire account within the preferences page if they wish to do so.

- Logging out will end the Firebase Authentication session, which secures the application and user data, as the user cannot use the “back” buttons to restore the session – they must be re-authenticated using their unique email and password combination.

How is the data stored?

One Direction uses Firebase for authentication and data storage (Authentication and Firestore), as well as SQL Lite.

The application will make use of Firebase Authentication for registering and logging in users, with email address and password combinations.

In addition to storing login credentials, One Direction also needs to store user preferences and trip logs. In order to accomplish this, Firestore is used to store user preferences for each registered user in the system. Trip logs are stored in an SQL Lite database file on the phone, for one specific reason: Firebase Firestore does not allow for documents larger than 1mb – I am storing the entire trip JSON (the DirectionsRoute object) in the database, which is then placed on the map when a user clicks on a certain trip log.

Default values will be used for some user preferences, which they may change at a later stage. For example, when the user signs up and enters their details and preferences, their preferred mode of transport is automatically set to driving. They may change this within their profile settings once they have signed up, and logged in.

Project Plan

WBS Number	Task Description	Dependencies	Start Date	End Date
1	Research		19/01/2020	30/01/2020
1.1	Identify three currently available Android mapping applications from the Play Store, and determine if they are suitable for this scenario.		19/01/2020	19/01/2020
1.2	Summarize functionality of the applications, and determine innovative features.		20/01/2020	21/01/2020
1.3	Evaluate strengths and weaknesses of the application.		22/01/2020	23/01/2020
1.4	Create a master list of all functionality from the three applications, for further analysis.		24/01/2020	25/01/2020
1.5	Compare applications, including functionality, design, and ease of use.		26/01/2020	27/01/2020
1.6	Create a list of features that the developer would like to implement in subsequent submissions (Task 2 and POE).		28/01/2020	28/01/2020
1.7	Perform risk and feasibility analysis before commencing design and planning of the application.		28/01/2020	30/01/2020
2	Plan		31/01/2020	04/02/2020
2.1	Continue feasibility analysis, to determine resource requirements, technological challenges, and risks.	Dependent on 1.7	31/01/2020	31/01/2020
2.2	Determine major components that the new application will require.	Dependent on 1.6	01/02/2020	02/02/2020
2.3	Schedule the work, and ensure tasks are completed sequentially or are able to be completed concurrently.	Dependent on 2.2	03/02/2020	03/02/2020
2.4	Set up development environment, and prepare for the design phase.		04/02/2020	04/02/2020
2.5	Prepare estimated timeline of when milestones will be completed. This will ensure that there is enough time to develop the application.	Dependent on 2.3	04/02/2020	04/02/2020

3	Design		05/02/2020	11/02/2020
3.1	Design the user interfaces of the application, using Android Studio. Determine how the user will navigate through the application, and how they will interact with its functionality.	Dependent on 2.2	05/02/2020	07/02/2020
3.2	Design the database schema. The developer can use the data listing from the previous section. Determine how the Firebase Authentication data store will link up with Firestore, which stores user preferences and trip logs.	Dependent on 2.2	08/02/2020	08/02/2020
3.3	Design methods and classes that the application will require in order to support all functionality.	Dependent on 2.2	09/02/2020	09/02/2020
3.4	Design integrity controls, such as maximum input lengths, and security controls that prevent unauthorised access. A password policy will be developed.	Dependent on 2.2	10/02/2020	10/02/2020
3.5	Develop test plan for the application.	Dependent on 2.2	11/02/2020	11/02/2020
4	Build		12/02/2020	13/03/2020
4.1	Build all application components for the system, including the user interface, and features. This includes all text fields, buttons, and other user interface elements.	Dependent on 3.1	12/02/2020	24/02/2020
4.2	Develop the database for the application.	Dependent on 3.2	25/02/2020	25/02/2020
4.3	Perform testing to ensure that the application does not have any bugs. The developer should also assess quality – which ensures maintainability, and a high level of accuracy is achieved.	Dependent on 3.5	25/02/2020	02/03/2020
4.4	Develop application documentation, for users and maintenance. This will assist users of the application if they require help.		03/03/2020	10/03/2020
4.5	Perform final analysis, before the application is published and deployed on the Android Play Store. A Google developer account will be required to complete this step.		11/03/2020	13/03/2020

5	Evaluate		14/03/2020	19/03/2020
5.1	Determine development inefficiencies, and find ways to overcome these delays.		14/03/2020	16/03/2020
5.2	Evaluate the final application, and compare it with competitors. Does it complete all functionality that is required?	Dependent on 4.5	17/03/2020	17/03/2020
5.3	Deploy the solution.	Dependent on 4.3, 4.5	18/03/2020	18/03/2020
5.4	Maintain the application, and provide support for users.	Dependent on 4.4	19/03/2020	19/03/2020

(The Independent Institute of Education, 2020)

Project Plan (continued)

Problem and solution domain:

As part of our OPSC7312 module, we have been tasked with developing a map (and routing) application for the Android mobile platform. This application aims to improve navigational routing, by finding the fastest route possible between the user's current location and the destination, set by the user. The application will provide this support by using MapBox's free to use SDK. All functionality will be implemented, including all functions outlined in the assignment question paper, and also stated in the list of functions previously.

In order to best fulfil these requirements, an android application will be developed, which will contain the required functionality. An easy to use user interface will be developed, so that the user can easily navigate to these features. (The Independent Institute of Education, 2020)

Milestones and deliverables:

Major milestones are provided below, with estimated time duration required:

- Complete research of existing applications (12 days)
- Plan project including the following (4 days):
 - Develop detailed list of major components
 - Determine feasibility
 - Development time estimation
 - Set up environment
- Design all system components (5 days):
 - Database
 - User interfaces
 - Functionality
 - Integrity controls
- Build all system components (30 days):
 - Database
 - User interfaces
 - Functionality
 - Integrity controls
 - Complete testing
 - Documentation
- Evaluate development process (6 days).

With this research and design documentation completed, the developer will be in a position to develop the prototype Android application.

After this is completed, the prototype build will be refined and submitted as the POE for this module.

Documentation

Introduction

As part of our OPSC7312 module, we were tasked with developing a mapping application for the Android platform. I chose to develop the application in Java, using the Android Studio IDE, as we were familiar with this IDE from OPSC7311 module.

We also required a maps SDK, and therefore MapBox SDK was chosen, due to it being open-source and allowing for integration within an Android application. (The Independent Institute of Education (Pty) Ltd, 2020)

The mapping application allows users to perform multiple actions, including:

- **Register and login**

The application allows the user to register an account with their own preferences, and log in. User profiles are stored in an online server, so they can log in on any device, which will set all their preferences automatically.

- **Select a destination**

The application allows users to either search for a location, by typing in an address, landmark name, or other phrase, and selecting the item from the dropdown list, or by dropping a pin on the map – by pressing a location on the phone screen.

- **Select preferred method of transport**

The user can select their preferred method of transport, including driving, cycling, and walking. The route calculator will take into consideration this preference when generating a route – for example, the application will not suggest that pedestrians travel on highways, etc.

- **Select preferred units of measurement**

The user may select either metric or imperial units, providing different localization support.

- **Find a route**

The application allows the user to find the fastest route from the origin to destination address, with the use of MapBox SDK integration.

- **Display ETA and distance**

The application will provide the user with trip statistics, such as travel time, estimated time of arrival, total distance travelled, distance remaining, and other such information.

- **Turn-by-turn navigation**

The maps application will prompt the user to take the correct turns and offramps in order to follow the predetermined route – the fastest route which MapBox has determined for the user. These navigation prompts will be visually displayed on the map, and also via an audio overlay.

- **Trip history**

All trips are recorded and viewable by only the logged in user – which keeps user information secure – they have to log in before they can access their profile and trip history.

The following sections will detail and explain all functionality of the application, including a help file, screenshots with descriptions, a use case diagram, and storage information. It aims to describe why, and how, certain features were developed in order to meet the requirements set out in our question paper.

Help File

The application provides numerous functions, which will be described in depth in the following section. The help file has been broken up into multiple sections, describing each page of the program.

Login Page

When the user first starts the application, a login page is presented. This provides the following functionality:

- Email Address Edit Text
- Password Edit Text
- Login Button
- Register Button
- Reset Password Button
- Remember Me Checkbox

The email address input allows users to enter their unique email address, and below this they can enter their account password in the password textbox. They can select the “Remember Me?” checkbox, which will remember the login details, and automatically log the user in when they open the application – making for more efficient use of the application.

Below this, three buttons are displayed:

- Login: Brings the user to the main screen (map view), where they can view their current location on the map – first-time users will be requested to allow the application to use their phone’s GPS functionality.
- Register: Brings the user to a registration page, which allows the user to register a new account on the system.
- Reset Password: Displays a popup dialog which allows the user to input their email address, which will send an email with a reset-password link to the email address provided, if an account exists for the user.

Reset Password

When a user selects this button, they can enter their email address, and click “Confirm”, which will send them a password reset link.

If they do not wish to reset their password, they can click “Cancel”.

Register Page

When the user presses the “Register” button on the login page, a registration page is presented. This allows users to register a new account with the system, and provides the following functionality:

Account information must be entered, including the following:

- Full Name Edit Text
- Email Address Edit Text
- Password Edit Text
- Confirm Password Edit Text
- Preferred Units of Measure Switch

After they enter their user account details, and press the “Register” button, they will be sent an email verification link, which they will have to use to register a new profile with the application.

The application will prompt the user to check their email inbox for this email, and navigate the user back to the login page.

If the user does not wish to create an account, they can select the “Back to Login” button, which will navigate the user back to the login page.

Home Page

Once the user has successfully logged in to their account, they will be brought to the home page, which provides the following functionality:

- Map view of surrounding location
- Current location (blue dot)
- User heading (blue arrow)
- Start Navigation Button – disabled
- Search Button
- Menu Button (“+” symbol)

The user will be able to view a map, which they can navigate using swipe and pinch to zoom gestures. Their current location and heading will also be provided, which will be displayed as a blue dot with an arrow pointing in the user’s direction.

Menu and search buttons are shown just above the Start Navigation button.

The search button opens up a page where the user can input their desired destination, such as an address, landmark, or other phrase. A dropdown list of options is then presented, and the user can select an item. Once selected, the user will be brought back to the map view, and the route will be drawn on the map – also enabling the Start Navigation button in the process.

The user may select a location on the map by dropping a pin on a location (pressing the screen), which will draw the route and enable the Start Navigation button – another way in which the user can select a destination address.

Menu

When the user selects the “+” menu floating action button (FAB), a menu will be expanded, which will include the following functions and information:

- About – a page describing the application.
- Preferences – allows the user to edit their user preferences.
- Route History – displays all previous trips (ordered by date added)
- Share Current Location – allows the user to send their location via SMS
- Refresh Map – refreshes the map and finds the user using GPS
- Logout – logs the user out of the application

These functions will be explained in more depth in the following pages.

About

Once the user selects this button, a page with application information such it's functionality and version number is presented, along with a “[We Still Love Your Music](#)” link.

This page also displays the application name and icon.

Preferences

Once clicked, a modal popup will be displayed with the following functionality:

- Full Name Edit Text
- Mode of Transport Spinner
- Preferred Units of Measure
- Delete Account Button
- Delete History Button
- Save Button

The user may update their account preferences, including selecting their preferred mode of transport and units of measure.

In addition to this, the user can delete their profile (which includes trip history, preferences, and login details), or just delete their history by using the Delete Account or Delete History buttons respectively. Confirmation popups are presented with both actions, so the user cannot accidentally delete their account or history.

Route History

Once clicked, the trip history will be presented, which will include the following information:

- Date / Time of the start of the trip
- Transit method – Driving, Cycling, or Walking
- “From” address line 1.
- “To” address line 1.

These trip records are ordered by date added, so the most recent trips will be displayed first.

If a user clicks a route, the route will be displayed on the map, with the use of Directions Route JSON objects.

Share Current Location

Once clicked, the Start Navigation button text will change to “Send my Location”, and an input field will appear in the middle of the screen requesting the user to “Enter a Phone Number”.

The user can cancel the action by pressing the pink X button, or send the message by pressing the “Send my Location” button (at the bottom of the page).

The user’s location is sent as a link to google maps, where it adds a pin with the latitude and longitude of the user’s current location – overlaying on a map.

Refresh Map

This action refreshes the map, by completing the following actions:

- Removes any current navigation routes
- Moves the map location back to the current user’s location
- Disables the “Start Navigation” button
- Prepares the application for the next navigation or other user action

Logout

Logs the user out, and clears shared preferences, so they will not be automatically signed in again if they open the application again – they will be brought to the login page.

Search

There is a search button on the home screen, which looks like a magnifying glass. Once clicked, a page will appear where the user can enter a phrase, address, or landmark name. Once this is input, results will appear, which the user can select.

When a location is selected from the dropdown list, the fastest route will be calculated from the origin position (the user’s current location), and the destination location.

Readme

Project Title: One Direction

Welcome to One Direction. An innovative maps application, with an even more innovative name. This maps application allows you to easily navigate to locations, through the use of searching for a landmark or address, or dropping a pin on the map.

Your profile is synced with an online database, so you can sign in on any device and obtain access to your profile.

Features include, but are not limited to, turn-by-turn navigation, trip history, selecting from a variety of different transport methods, and many more features.

Getting Started

The following steps are required to get One Direction application running in the development environment:

- Open the application source code in Android Studio
- Build the application
- Run the application to a phone or emulator on API level 23 or greater

Prerequisites

There are a few prerequisites required to run the application, including:

- Install the *latest Android Studio
- Set physical device to development mode (USB) or set up emulator environment.
- Ensure device is using Android API 23 or above

*latest Android Studio as of when the application was developed is: Android Studio 4.0

More detailed specifications are included below

Build #AI-193.6911.18.40.6514223, built on May 20, 2020
Runtime version: 1.8.0_242-release-1644-b01 amd64
VM: OpenJDK 64-Bit Server VM by JetBrains s.r.o
Windows 10 10.0
GC: ParNew, ConcurrentMarkSweep
Memory: 32716M
Cores: 12
Registry: ide.new.welcome.screen.force=true

Installing

- Open the application source code in Android Studio
- Build the application
- Run the application to a phone or emulator on API level 23 or above

*The development test system has been detailed on the following page.

Test System
Development PC

OS Name	Microsoft Windows 10 Pro
Version	10.0.18363 Build 18363
Other OS Description	Not Available
OS Manufacturer	Microsoft Corporation
System Name	KARL
System Manufacturer	System manufacturer
System Model	System Product Name
System Type	x64-based PC
System SKU	SKU
Processor	Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz, 3696 Mhz, 6 Core(s), 12 Logical Processor(s)
BIOS Version/Date	American Megatrends Inc. 2301, 2020/02/25
SMBIOS Version	3.0
Embedded Controller Version	255.255
BIOS Mode	Legacy
BaseBoard Manufacturer	ASUSTeK COMPUTER INC.
BaseBoard Product	ROG MAXIMUS X HERO
BaseBoard Version	Rev 1.xx
Platform Role	Desktop
Secure Boot State	Unsupported
PCR7 Configuration	Binding Not Possible
Windows Directory	C:\WINDOWS
System Directory	C:\WINDOWS\system32
Boot Device	\Device\HarddiskVolume5
Locale	South Africa
Hardware Abstraction Layer	Version = "10.0.18362.752"
User Name	KARL\Karl
Time Zone	South Africa Standard Time
Installed Physical Memory (RAM)	48,0 GB
Total Physical Memory	47,9 GB
Available Physical Memory	32,7 GB
Total Virtual Memory	54,9 GB
Available Virtual Memory	35,4 GB
Page File Space	7,00 GB
Page File	C:\pagefile.sys
Kernel DMA Protection	Off
Virtualization-based security	Not enabled
Hyper-V - VM Monitor Mode Extensions	Yes
Hyper-V - Second Level Address Translation Extensions	Yes
Hyper-V - Virtualization Enabled in Firmware	Yes
Hyper-V - Data Execution Protection	Yes

Development Phone: Samsung Galaxy S8

Android Version	9
API Level	28

Built With

Android Studio – The IDE used to develop the application

Maven – Dependency management

Gradle – Automation and management of the build process

MapBox – Maps integration (SDK)

Versioning

Version Code 30

Version Name 1.30

Authors

Karl Dicks – 17667327

Acknowledgments

Inspiration: One Direction

Website: <http://www.onedirectionmusic.com/>

Deployment

Documentation will be included in the final POE, as marks have been allocated to this section for the POE, and not Task 2.

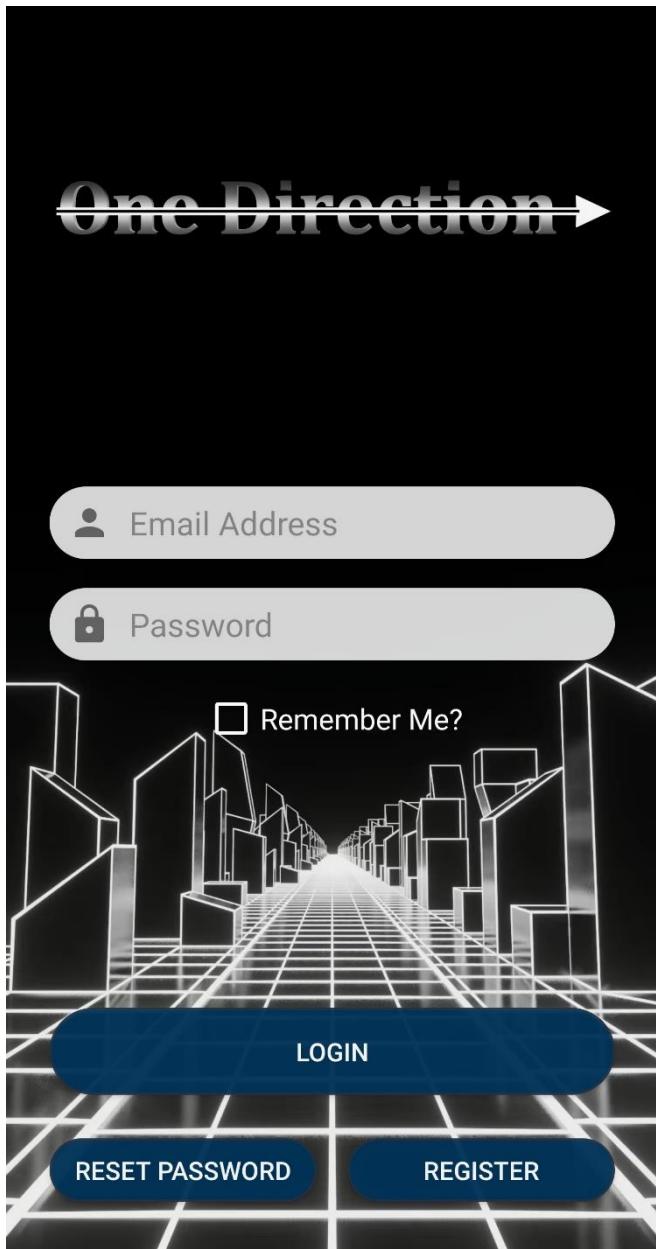
Demo

Video link: <https://youtu.be/1bB2kAicRQo>

Screenshots

The user interface for One Direction has been designed, and all functionality has been implemented. Below is the interface for my maps application:

Login



Once the application has loaded, the user will be presented with the login page.

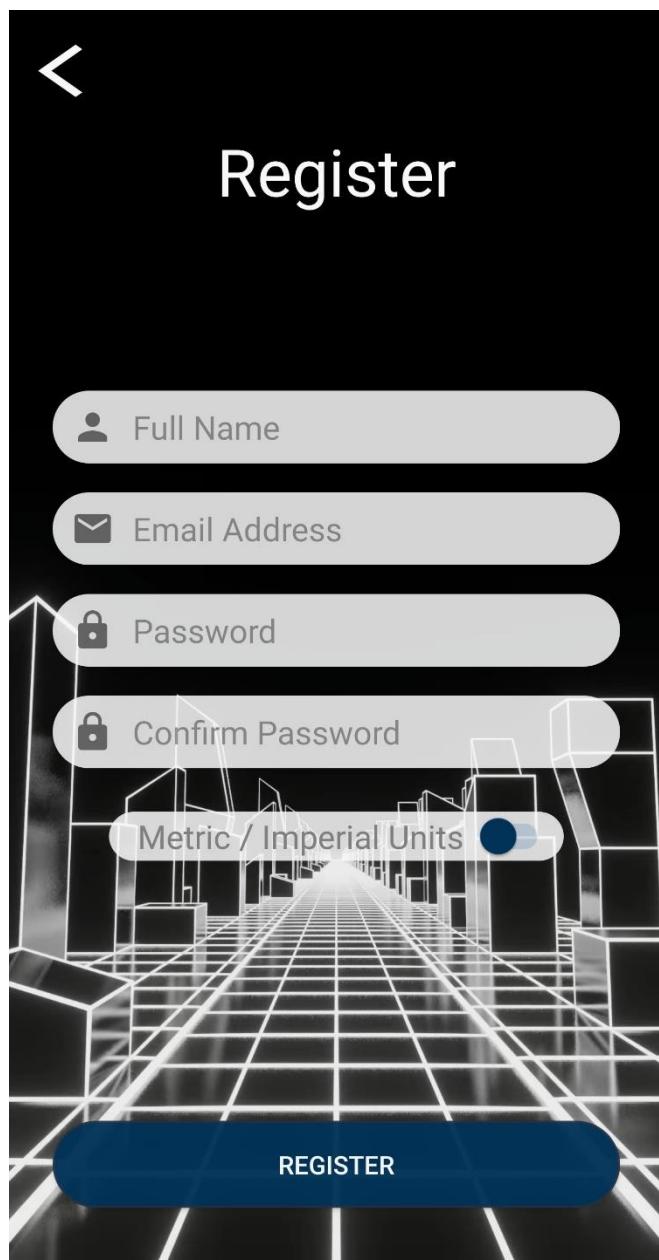
User authentication will occur once the user enters their unique email and password, and presses the “LOGIN” button.

The “Remember Me?” checkbox will remember the user account, and not request user login each time they open the application.

The login process is asynchronous, which means that there will be a delay when contacting the Firebase data store. During this time interval, a pink spinner will be displayed to let the user know that this process is currently occurring.

The user may register a One Direction account by pressing the “REGISTER” button, which will bring them to the register page (shown on the following page), or reset their account password by pressing the “RESET PASSWORD” button.

Register



When the user presses the “REGISTER” button on the login page, they will be presented with a register form (shown on this page).

They will need to enter the following information:

- Full Name
- Email Address
- Password
- Confirm Password
- Preferred Units of Measure

Input validation has been implemented in the following ways:

- Passwords Must Match
- Minimum Password Length is 6 Characters.
- No Input Fields Can be Null
- Email Address Must Conform to Regular Expression – Must be Valid.
- Email Address Cannot be Currently in Use by Another Account.

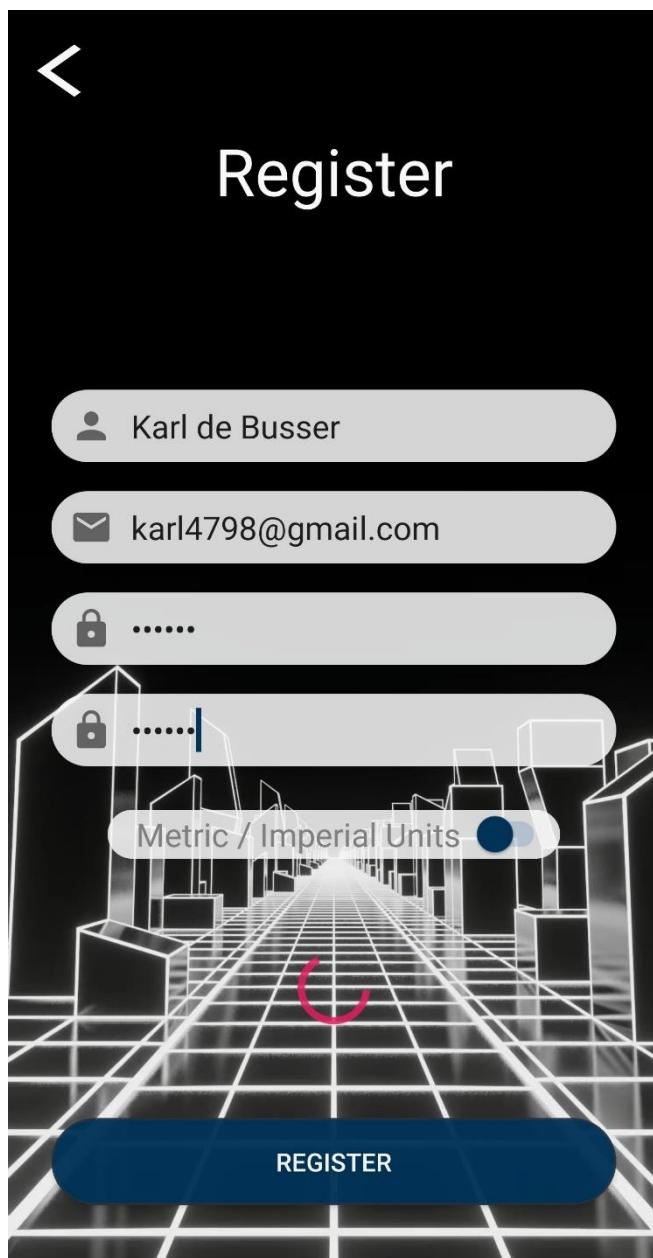
This validation is completed to ensure that garbage, duplicate, or malicious values are not persisted in the data store.

Once the user presses the “REGISTER” button, validation will be processed, and if this is passed, the user account will be created.

After the user account has been created, the user will be requested to verify their email address, by clicking a link which would have been sent to their email address automatically.

They will be navigated back to the login page after registration, and prompted to validate their account.

Account Created



Once the user has pressed the “REGISTER” button, their account will be created (as shown).

The image on this page provides an example of how account creation will appear.

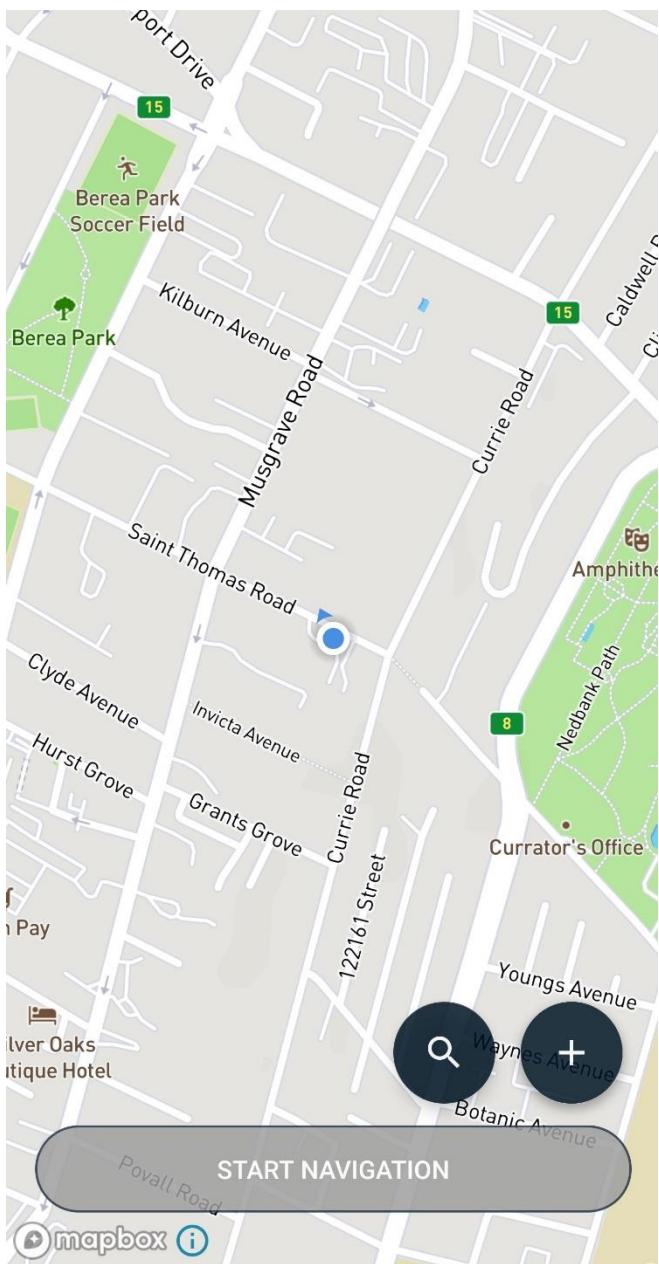
The spinner will be presented (and turn) during this asynchronous event.

Once finished, the user will be prompted to check their email for a verification link, which they will have to click in order to use their account.

If the phone cannot reach Firebase due to network or other transient errors, a toast will be displayed. This message will provide error details.

If validation fails, then a toast message will describe how the user can fix the incorrect values.

Home



Once the user has logged in, they will be presented with the home screen.

This contains the following features:

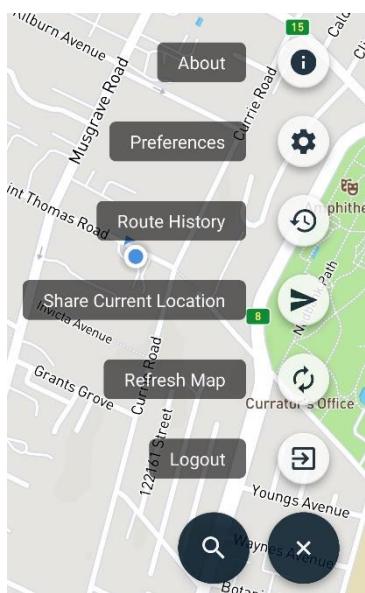
- A Detailed Map
- Current Location on the Map
- Start Navigation Button, which is Disabled by Default.
- Search Button.
- Menu Button (Floating Action Button).

The “START NAVIGATION” button is disabled by default, and only once the user drops a pin on the map, will it become enabled.

The user can drop a pin on the destination address by moving the map to the destination, and pressing the screen of the device, or they can press the search button (Pink magnifying glass symbol) and enter a search phrase.

The user can access the main menu of One Direction, by pressing the Floating Action Button (Pink + Symbol). The menu is displayed on the following page.

Menu



Once the user has pressed the menu (Floating Action Button), additional buttons will appear.

The following actions can be triggered from this menu:

- About
- Preferences
- Route History
- Share Current Location
- Refresh Map
- Logout

A detailed description of these actions is provided below.

Menu Functions

About

This action results in a modal popup being displayed (shown on the next page), which allows the user to view application information (version number, etc).

Preferences

This action results in a modal popup being displayed, which allows the user to set their preferences.

Route History

This action results in a modal popup being displayed, which allows the user to view all previous trips.

Share Current Location

This action results in an edit text being displayed, and the “START NAVIGATION” button changing to “SEND MY LOCATION”. Once pressed, the user’s current location will be sent to the phone number which has been input in the edit text.

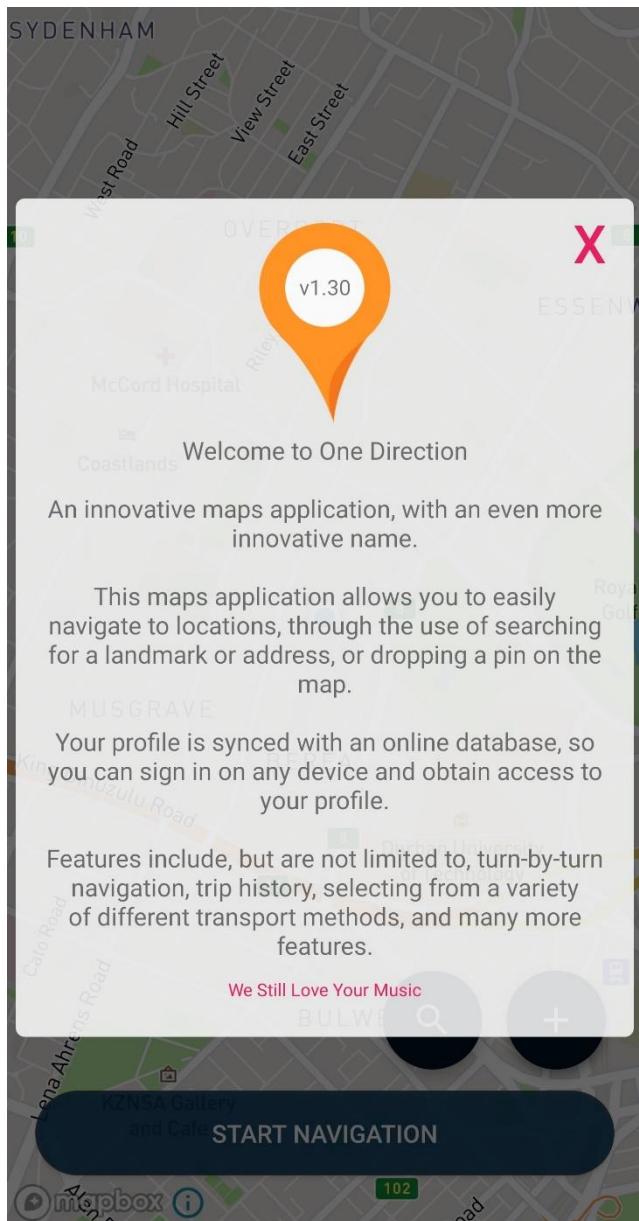
Refresh Map

This action results in the destination address, and navigation route, being cleared from the map. It refreshes the map and re-triangulates the current user location, along with their set preferences.

Logout

The user may log out of their account by pressing this button. This action will end the session, so even if they press the return button on the phone, they will not be able to log in again without providing their unique login credentials.

About



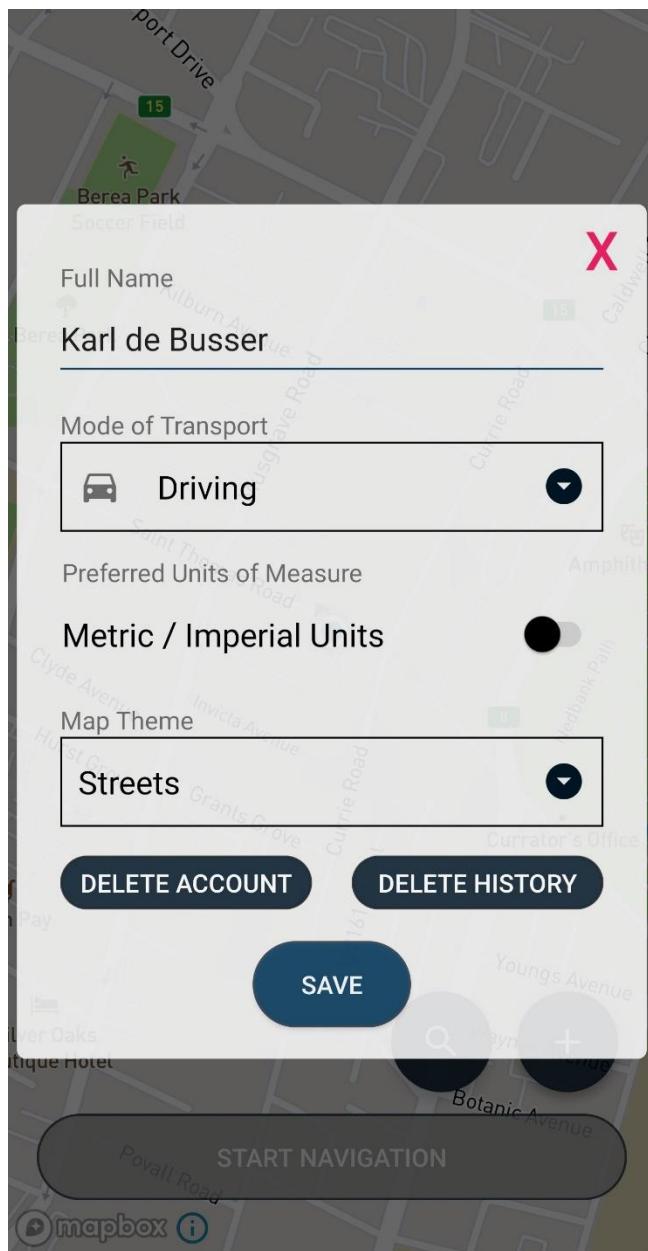
Once the “About” button is pressed, a modal popup will be displayed, which includes the following information:

- Application Name
- Application Version Number
- Features
- A “[We Still Love Your Music](#)” Link.

This information can be used to determine which version has been downloaded onto the phone, and also contains feature information.

When pressed, the “[We Still Love Your Music](#)” link will navigate the user to a YouTube video.

Preferences



Once the “Preferences” button is pressed, a modal popup will be displayed, which includes the following information:

- Full Name of the Logged in User
- Preferred Method of Transport
- Preferred Units of Measure
- Map Theme
- Delete Account Button
- Delete History Button

The user may change their preferences by pressing on their respective edit text, spinner, or switch button.

For example, the user may change their preferred mode of transport to Walking, Cycling, or Driving.

They may also set their preferred units of measure from Metric to Imperial, or vice versa.

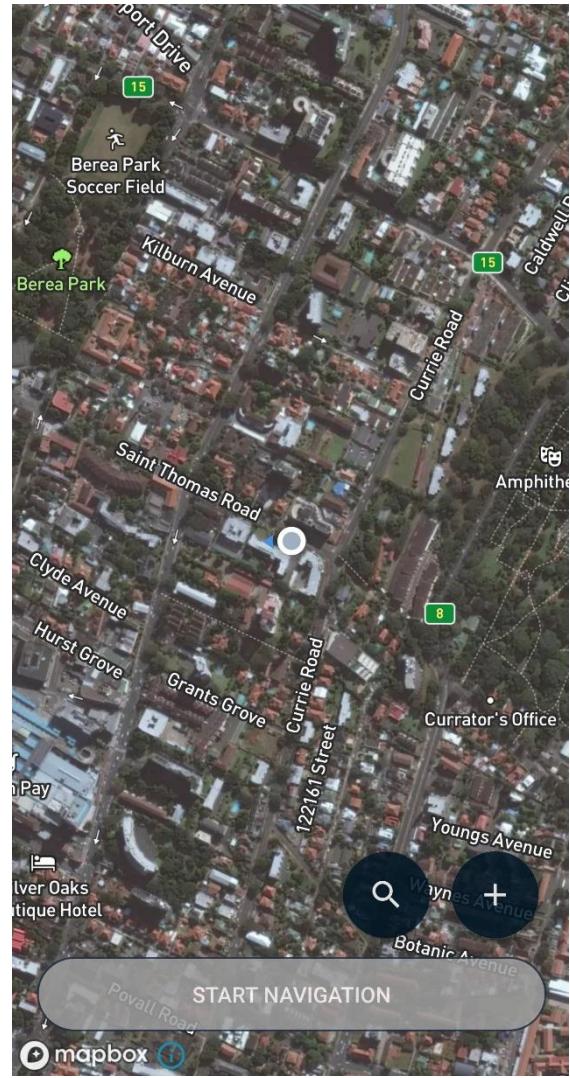
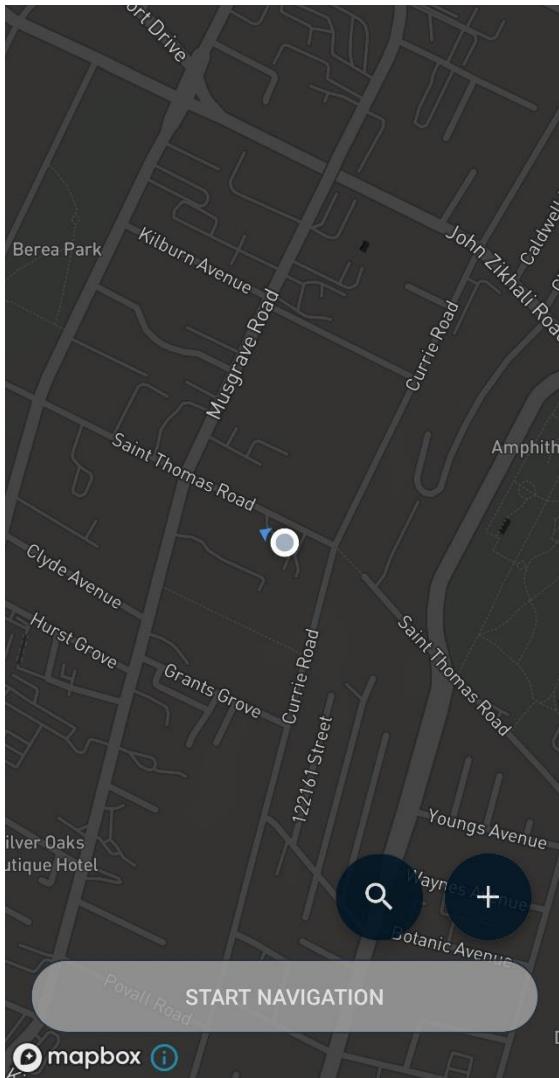
The map theme can be changed from “Streets” to a number of other themes, which are shown on the next page.

The user can delete their trip history or entire account by pressing the “DELETE HISTORY”, or “DELETE ACCOUNT” buttons, respectively.

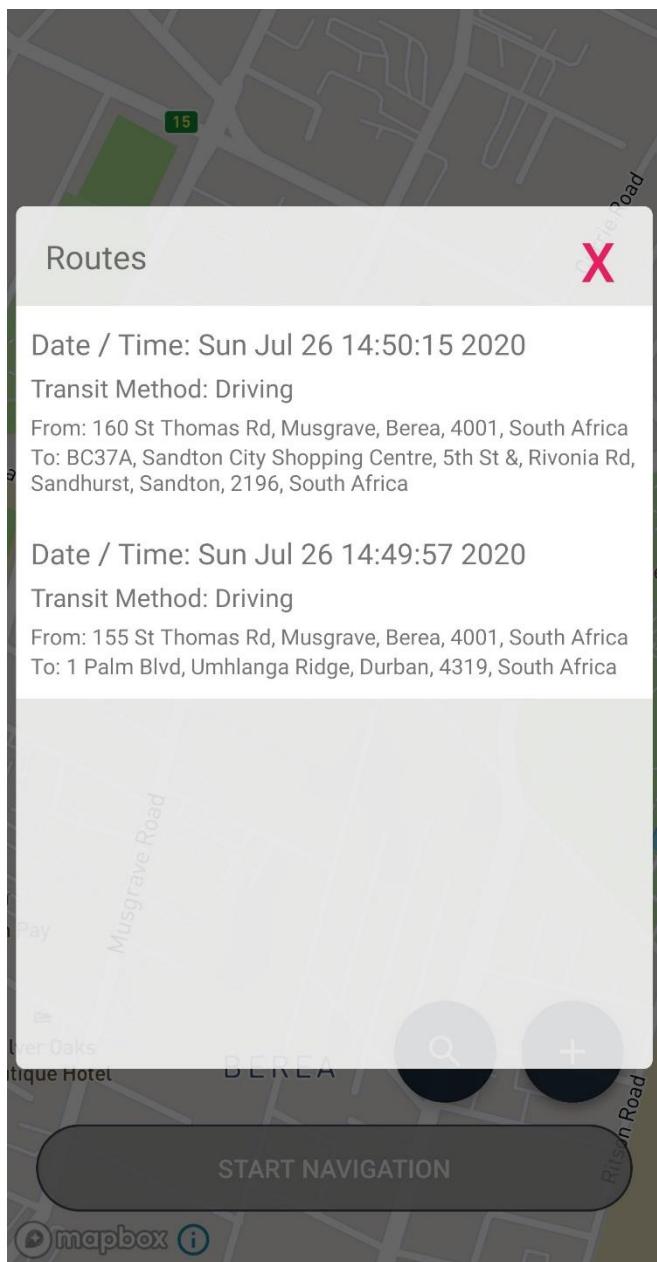
Once changed, the user can press the “SAVE” button to save the preferences to their Firestore profile.

Map Themes

The user can select different map themes from within the user preferences page, and some examples are shown below.



Route History

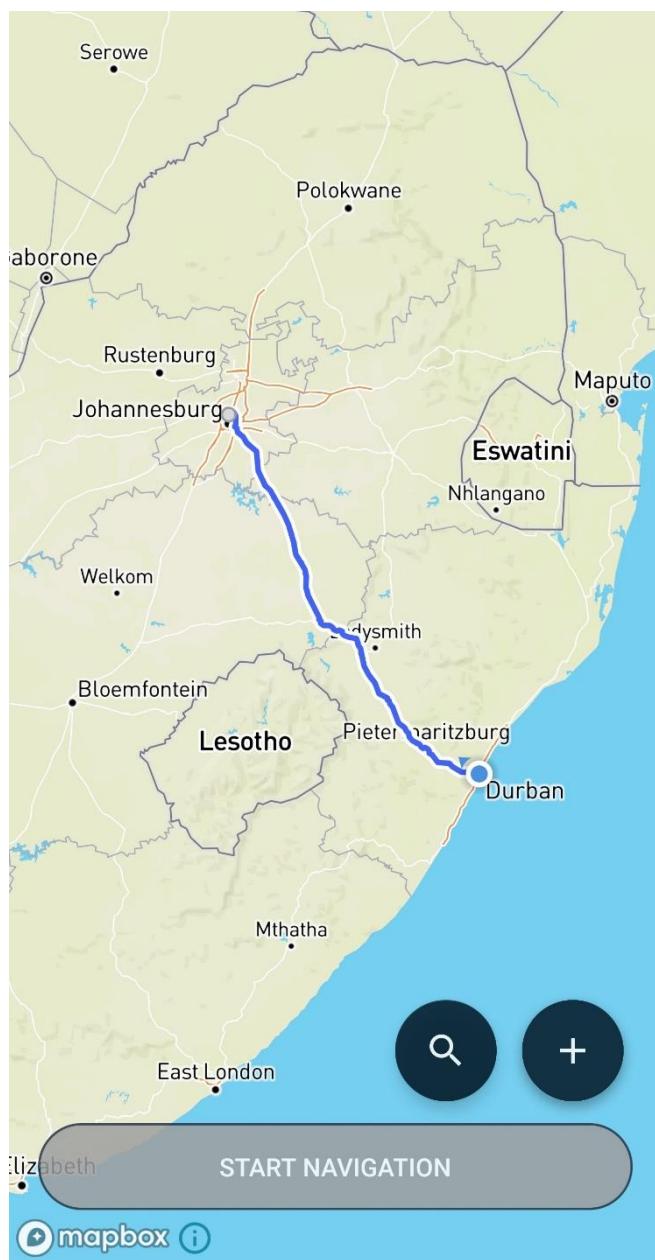


Once the “Route History” button is pressed, a modal popup will be displayed, which includes the following information:

- Date and time of the trip
- Method of Transport
- From Address
- To Address

The user can click any of the available items on the list, which will draw the route on the map, and close the modal popup window (shown on the next page).

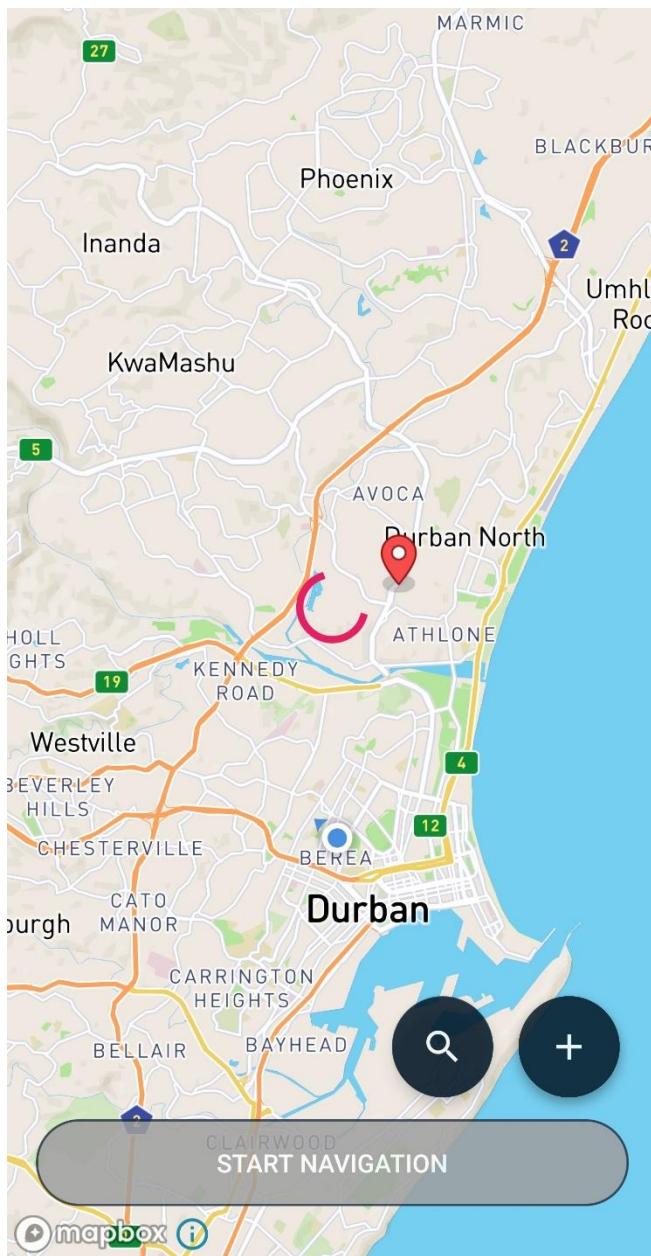
Saved Route



Once the user presses an available route log item, the route will be drawn on the map.

This map will include the beginning and end points of the route, and all navigation turns that were required for the chosen route.

Finding Route



When the user drops a pin on the map (by pressing on a location on the map), a route will be generated.

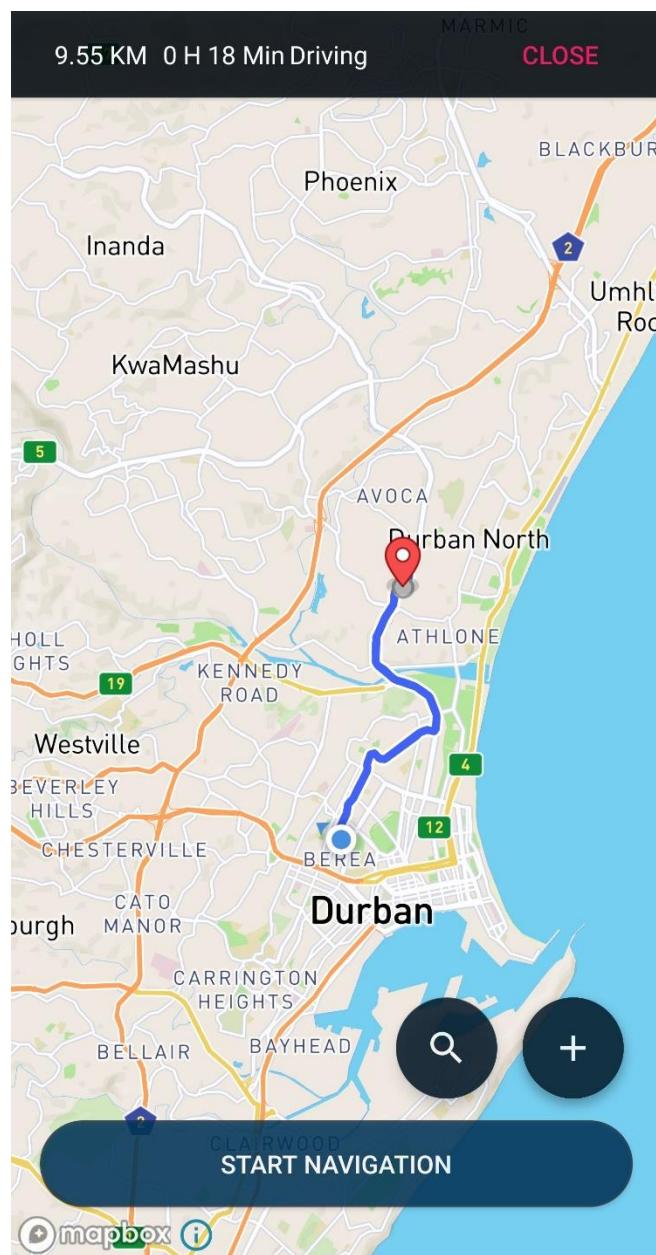
They can also search for a location using a text phrase, entered on the search page (shown on the following page).

The MapBox SDK will take into consideration the user's preferred mode of transport. For example, if the preferred mode of transport is walking, then the routing will avoid highways.

The picture provided on this page displays the processing of the route, which is an asynchronous process and is completed by MapBox integration.

The next page provides an overview of what a route would look like, once generated and retrieved from MapBox.

Generated Route

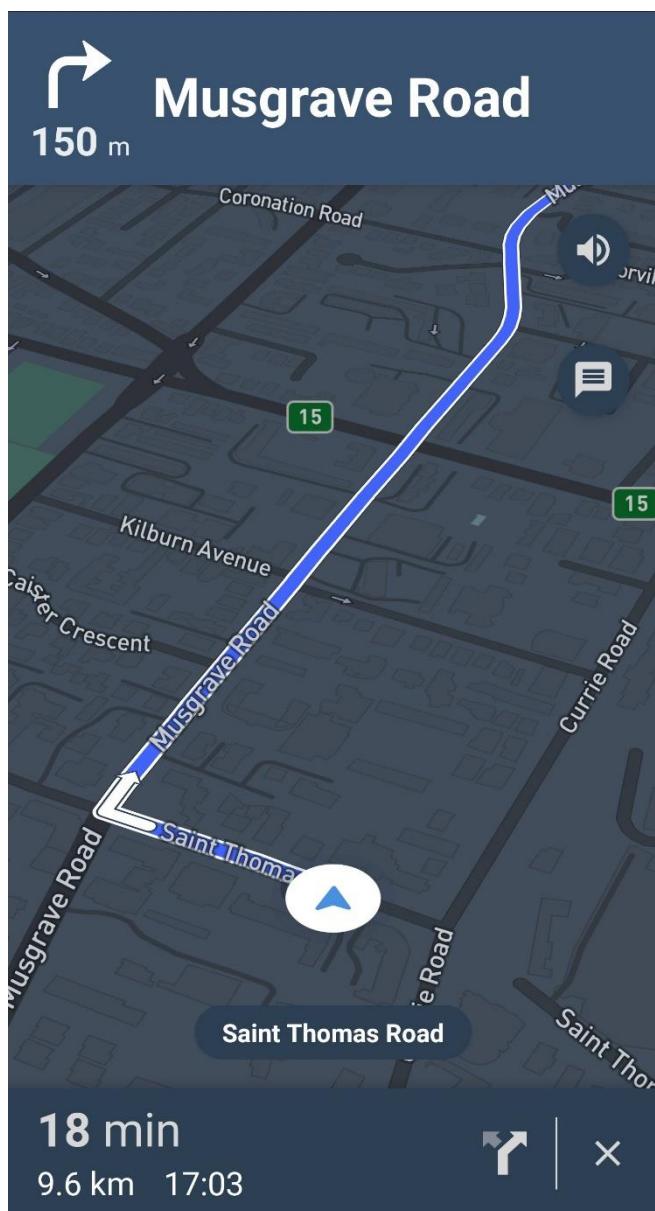


Once the route has been generated, the “START NAVIGATION” button will be enabled, and change color to dark blue – signalling that the route has been generated and that they can start the navigation process.

An overview of the route will be displayed on the map, which will also provide the original and destination locations.

The user can now press the “START NAVIGATION” button and retrieve visual and audio navigational information (shown on the next page).

Navigation



When the user presses the “START NAVIGATION” button, visual and audio navigational prompts will provide turn-by-turn navigation until the user reaches their destination.

The following information will be provided on this page:

- Next Road Turn
- Distance Until Turn
- Turn Direction
- Current User Location
- Current Road
- Travel Time Duration
- Travel Distance
- Estimated Time of Arrival

In addition to this visual feedback, audio navigation will also be provided, much like Google Maps and their competitors.

This includes distance to offramps, turns, and other such navigational information.

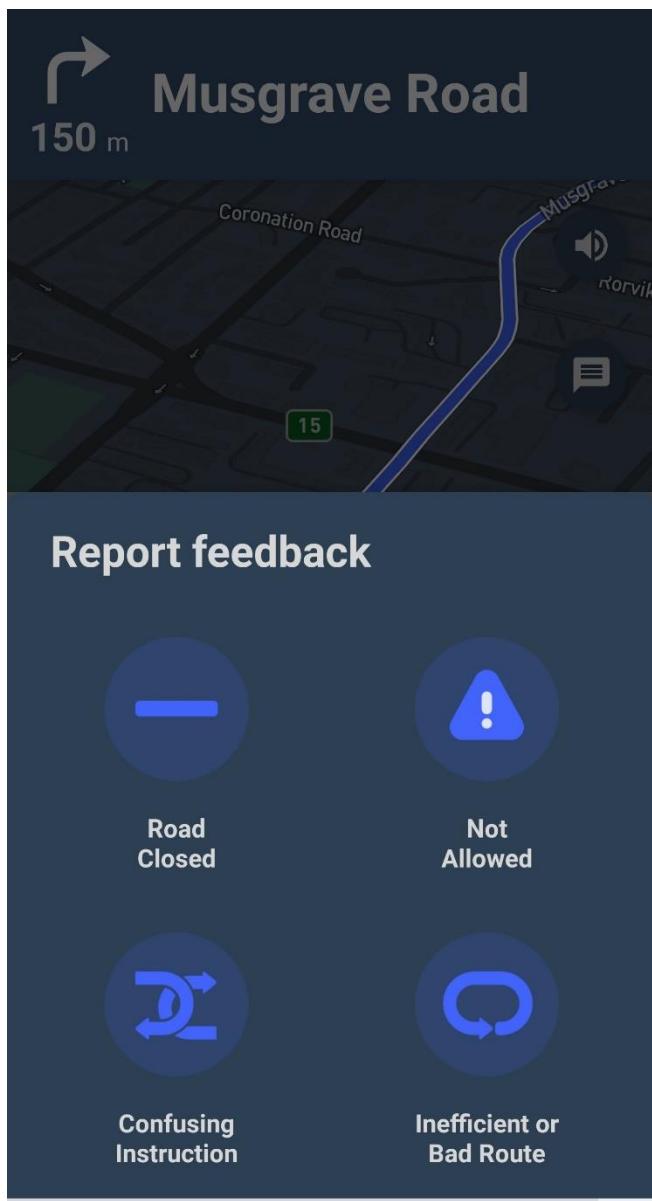
This audio can be muted by pressing the mute button (top right of the screen).

Some additional features present on this page include:

- Reports: Allows users to report road closures, traffic incidents, map issues, and other such information. This is sent to MapBox, and will be used to better navigate its users.
- Detailed Navigation: A detailed listing of turns, offramps, and the distance from the current user location is available.
- Route Overview: The user can view the route overview.
- Stop Navigation: The user can stop navigation by pressing the X button at the bottom right of the screen.

Details of the above information is provided in the following pages.

Reports



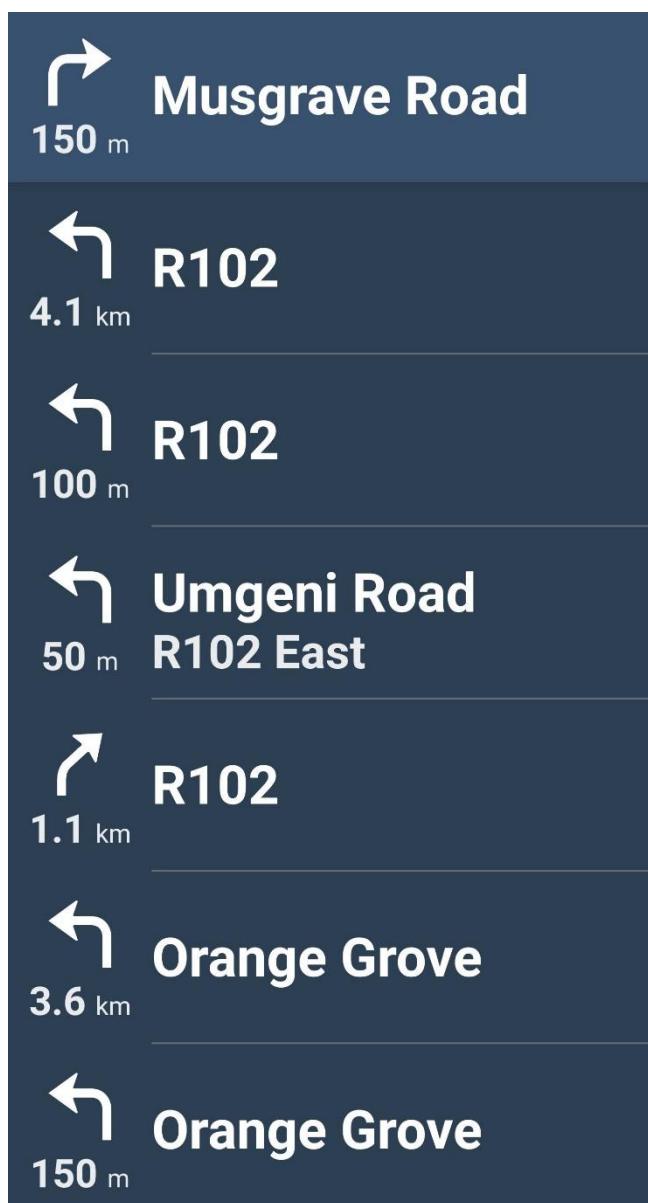
During navigation, the user can press a “reports” button (below the mute / unmute button).

When the user selects this button, they are able to select any of the displayed report options.

For example, the user can report a road closure, which will help other MapBox users, as it will re-route them around the road closure or incident.

This “reports” page automatically hides after a few seconds, as shown in the very bottom of the image – the progress bar.

Detailed Navigation

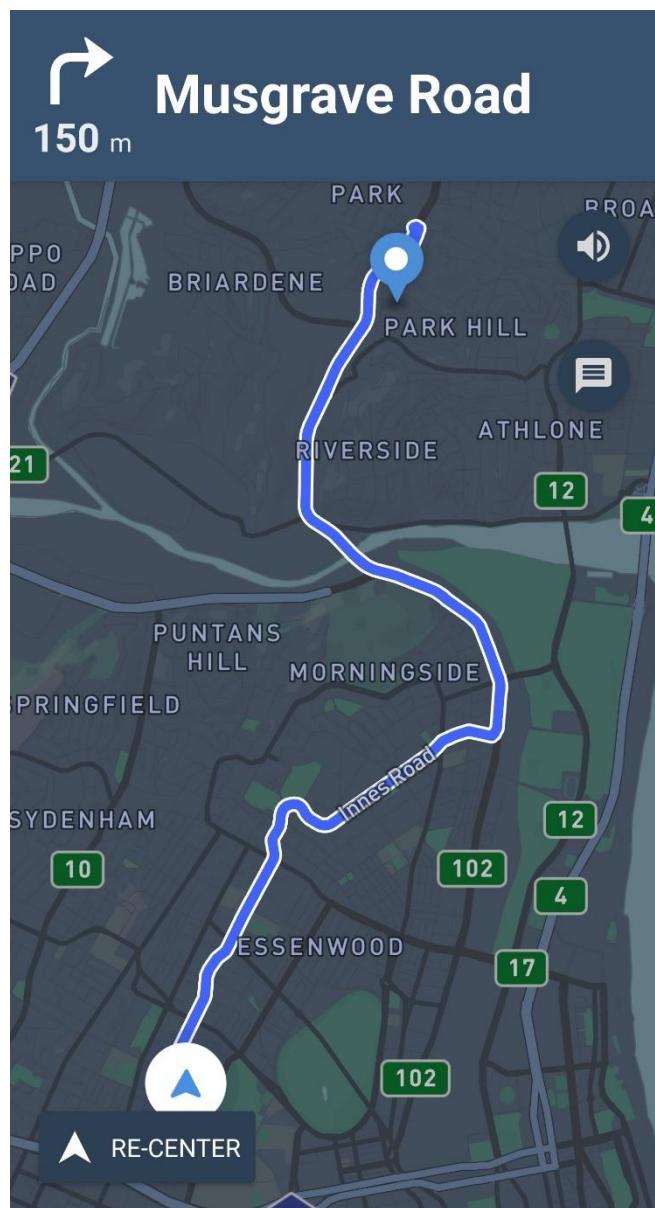


During navigation, the user can press the upcoming turn (topmost banner).

When the user presses this banner button, a list of all turns, offramps onramps, and distance between them is provided.

In addition to this, turn information is provided. For example, in this case the user should take the onramp to the N3, towards Pietermaritzburg.

Route Overview

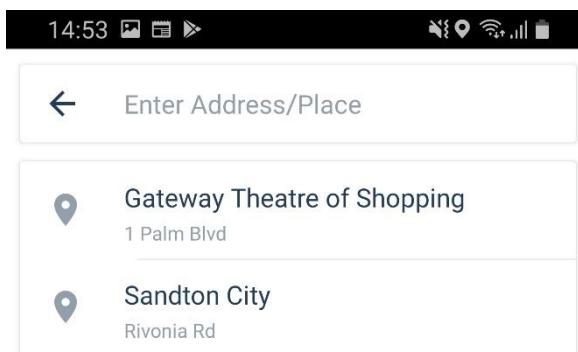


During navigation, the user can press a “route overview” button (to the left of the stop route button “X”).

When the user selects this button, the route overview is displayed.

The current user location and heading is included, along with the destination and calculated route.

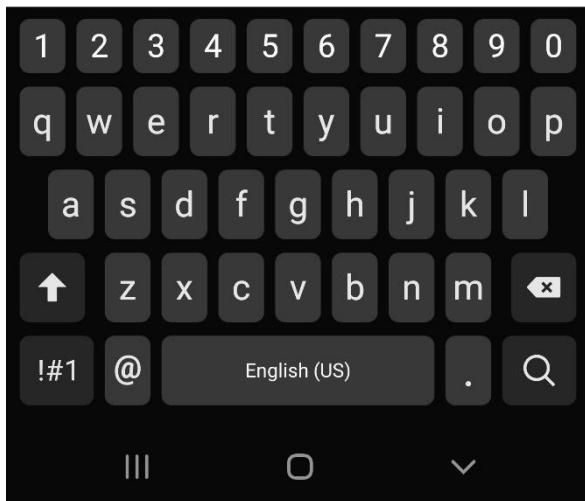
Search



The user can select a destination by clicking on a location on the map (by pressing on the map interface), or by clicking the “Search” button on the home screen (map view) – The magnifying glass symbol.

Once clicked, a new page will be presented, which will allow the user to enter a search term or phrase, which will populate a drop-down list of suggestions. The user may select a location, which will then find the route and navigate the user back to the map page (home screen).

Recently searched for terms will be stored on the page, for ease of use.



Data Listing

One Direction uses Firebase Firestore, Authentication, and SQL Lite services to store user profiles (authentication and preferences), and SQL Lite for trip logs – which can be played back at a later stage. The following information will be stored in the database.

Data Name	Type (Java)	Type (Database)	Nullable	Length
User UID	String	Text	False	Max
Email Address	String	Text	False	Max
Password	String	Text (Hashed)	False	Max
Full Name	String	Text	False	Max
Units of Measure	String	Text	False	Max
Transport Method	String	Text	False	Max
Map Theme	String	Text	False	Max
Trip Log	JSON	Text	True	Max

Some data will be calculated within the application, including:

Data Name	Type (Java)	Length
Distance to Destination	Double	Max
Estimated Travel Time	Double	Max
Estimated Time of Arrival	Double	Max

Some considerations have been taken during the implementation of the database, including:

- All information stored in the database (Firebase) will only be accessible by authorised users.
Each user will have their own account, and will only be able to view trip logs and preferences for their account, and not others – securing information.
- Email addresses which have been used to sign up must be unique, otherwise Firebase will throw an exception, which will be caught in a Toast message. This error message will inform the user that there is already an account created with the provided email address.
The account email address must also be verified before use.
- Route information will be cleared from the asynchronous MapBox session when, or if, the user presses the “Refresh Map” button.
- The user may delete their trip history or entire account within the preferences page if they wish to do so.

- Logging out will end the Firebase Authentication session, which secures the application and user data, as the user cannot use the “back” buttons to restore the session – they must be re-authenticated using their unique email and password combination.

How is the data stored?

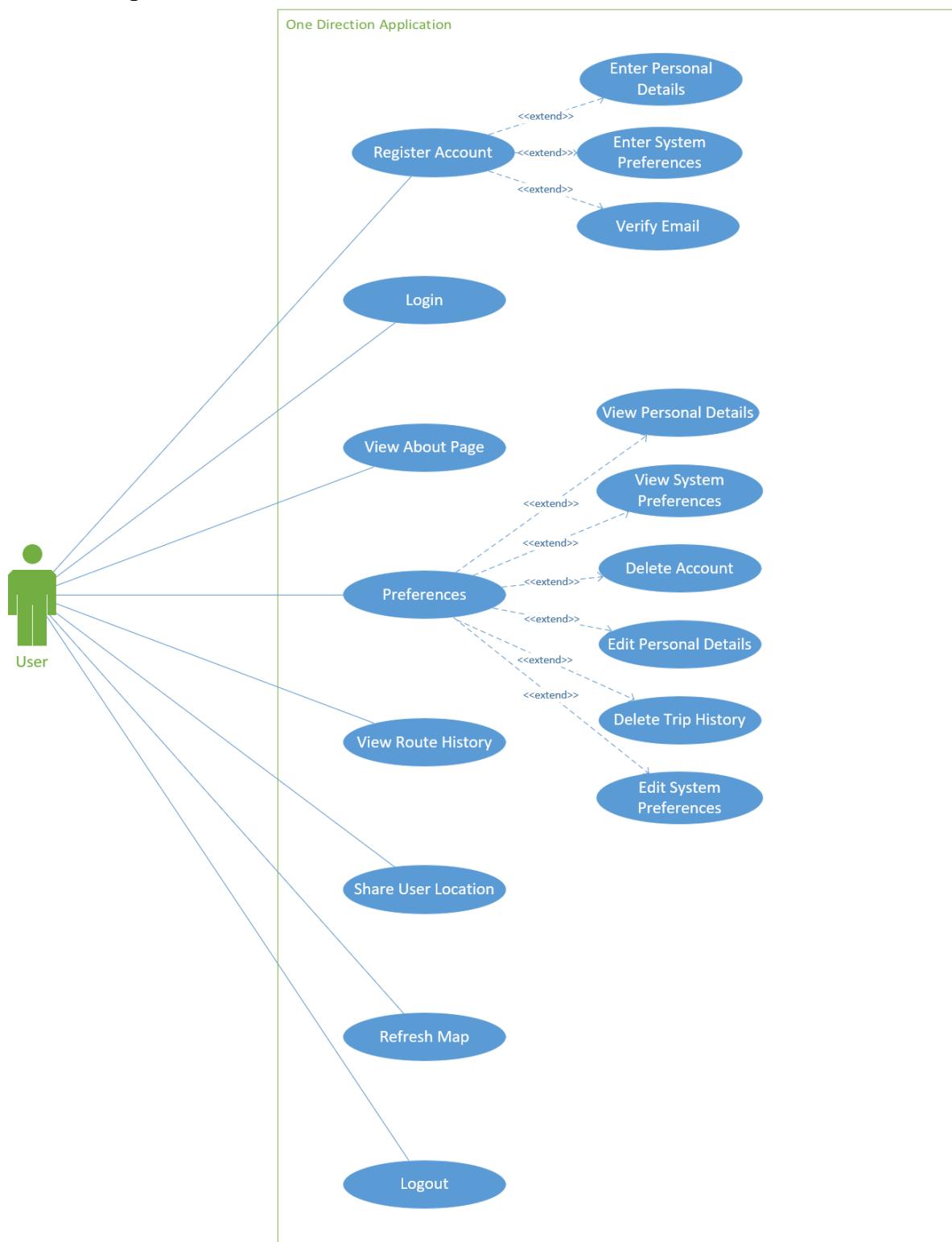
One Direction uses Firebase for authentication and data storage (Authentication and Firestore), as well as SQL Lite.

The application will make use of Firebase Authentication for registering and logging in users, with email address and password combinations.

In addition to storing login credentials, One Direction also needs to store user preferences and trip logs. In order to accomplish this, Firestore is used to store user preferences for each registered user in the system. Trip logs are stored in an SQL Lite database file on the phone, for one specific reason: Firebase Firestore does not allow for documents larger than 1mb – I am storing the entire trip JSON (the DirectionsRoute object) in the database, which is then placed on the map when a user clicks on a certain trip log.

Default values will be used for some user preferences, which they may change at a later stage. For example, when the user signs up and enters their details and preferences, their preferred mode of transport is automatically set to driving. They may change this within their profile settings once they have signed up, and logged in.

Use Case Diagram



Preparation for Publishing the Application

The following steps will be taken to publish my One Direction application to the Google Play Store (Oragui, 2018):

1. Firstly, a google developer account will have to be created, and the \$25 one-time registration fee will have to be processed. This allows the developer to publish the APK to the Play Store.
2. If the developer chooses to sell the application, a merchant account will have to be linked, within the google developer account. For the moment being, the application will be free to use, however advertisements will be included later on, in order to monetize the application.
3. The application will then have to be created – which has been done in our Task 2 assignment, and refined and completed in the POE. This application will have no errors, as it will have been thoroughly tested in every possible scenario.
After the application has been developed and tested, it can be added to the developer account by navigating to the “Create Application” tab on the developer console (within the Google developer account).
4. The store listing will have to be prepared, essentially advertising it on the Play Store. This includes the following information, which must be filled out before the application can be published:
 - a. Product Details – Application details must be filled out during this process, including an application title, and description. This will all be filled out for One Direction.
 - b. Graphic Assets – Screenshots, videos, promotional graphics, and icons can be added here, in order to showcase the application. This is essentially an advertisement for the application, so it must look very professional.
 - c. Languages & Translations – Translations for the application can be added for different regions. Localized images, screenshots, and languages can be used for different regions around the world.
For now, only English is supported by One Direction, however this may change in the future if requirements change.
 - d. Categorization – The application will need to be added to a certain category (in this case Navigational and Mapping applications). This category is used to filter results for users who search the Play Store.
 - e. Contact Details – This part allows customers to access support regarding the application. An email address is required, however other information can be included.
The email of the developer will be included here, which is “karl@karldicks.co.za”. (Oragui, 2018)

- f. Privacy Policy – A privacy policy for the application will have to be developed, and incorporated in the application (via a URL, or other method). Due to this application requiring personal information, a suitable privacy policy will be required.
5. Upload the Android Package Kit (APK), which has been generated for One Direction, to the Google developer console, and release the application either to a select number of “test” users, or for the greater public. The application will be published and accessible by only a limited number of application testers before it is available publicly, across selected regions that the developer has chosen.
I would publish the application globally, in all regions that support Google Play Store.
6. The application listing will require a content rating. This application is very similar to Google Maps or other such applications, and therefore will have a 3+ rating. The Google content rating will be assigned through the use of their questionnaire, which aims to determine the content of the application through a series of questions.
7. Set up pricing and distribution – The application will be set to “free” for the moment being, and advertisements may be added later on in order to monetize the application.
Making the application free to use will draw in users initially, and once they are using the application, advertisements can be introduced to monetize the application across a number of users, which is therefore much more effective.
8. Once the above processes have been completed, the application will be reviewed, in order to make sure all requirements have been met for the publishing of the application. Once the developer is satisfied with the release rollout, they can confirm the rollout, which in turn will publish the application to all users and regions specified during the process.
The One Direction application will now be accessible on the Google Play Store.
(Oragui, 2018)

The above list details how I would publish my One Direction Android application to the Google Play Store. I have provided the Android Package Kit (APK), which is the installer for the One Direction application within the solution files in a folder named “APK”.
(Oragui, 2018)

Conclusion

In conclusion, this document has provided extensive development information in order to detail how and why the application was developed in the way that it was.

It described the functions of each page within the “Help File” section, and provided design information within the “Screenshots” section.

A “Readme” section was also included to provide development environment information, instructions on how the application should be run, and other such information.

Additional information such as a data listing was provided, which detailed how, where, and why data was stored by the application.

Lastly, a use case diagram was included, which showed all functionality of the program, from the user’s perspective.

During the course of this project, we have learnt how to develop advanced Android applications in the Java programming language and XML. We also learnt how to use SQL Lite databases and Firebase Firestore and Authentication, which provides versatile relational, and non-relational databases.

Once we receive feedback for this task, we will be in a position to complete our POE task, which aims to improve our Task 1 and 2 projects. (The Independent Institute of Education (Pty) Ltd, 2020)

References

- Abigail, 2020. *Waze Review: All Features and Benefits*. [Online]
Available at: <https://robots.net/tech-reviews/waze-review-all-features-and-benefits/>
[Accessed 21 07 2020].
- Beech, H., n.d.. *The Waze App Reviewed: All You Need to Know about the Traffic App*. [Online]
Available at: <https://trendingtechnow.com/waze-app-review/>
[Accessed 22 02 2020].
- Bouckley, H., 2019. *Here WeGo: The free mapping app that could replace your sat nav*. [Online]
Available at: <https://home.bt.com/tech-gadgets/phones-tablets/here-the-free-mapping-app-that-could-replace-your-sat-nav-11363969427308>
[Accessed 19 01 2020].
- Cabebe, J. & Goodwin, A., 2014. *Waze review: Your trusted source for traffic data*. [Online]
Available at: <https://www.cnet.com/reviews/waze-social-gps-maps-traffic-review/>
[Accessed 22 02 2020].
- Hindy, J., 2019. *10 best GPS apps and navigation apps for Android!*. [Online]
Available at: <https://www.androidauthority.com/best-gps-app-and-navigation-app-for-android-357870/>
[Accessed 20 02 2020].
- Hoch, D., n.d.. *Maps Me Review – The Best Map App*. [Online]
Available at: [https://longhaultrekkers.com/maps-me-review/#:~:text=One%20of%20the%20main%20issues,driven%20and%20completely%20open%20so](https://longhaultrekkers.com/maps-me-review/#:~:text=One%20of%20the%20main%20issues,driven%20and%20completely%20open%20source.)
urce.
[Accessed 21 07 2020].
- Million Mile Secrets, 2016. *My Review of Maps.me: The Best Offline Map App!*. [Online]
Available at: <https://millionmilesecrets.com/reviews/maps-me/>
[Accessed 19 02 2020].
- One Trip at a Time, n.d.. *Ultimate Guide to Using Here WeGo*. [Online]
Available at: <https://www.onetripatatime.com/how-to-use-here-wego-for-easy-navigation-on-your-next-trip/>
[Accessed 21 07 2020].
- Oragui, D., 2018. *How to Publish an App on Google Play: A Step-by-Step Guide*. [Online]
Available at: <https://themanifest.com/mobile-apps/how-publish-app-google-play-step-step-guide>
[Accessed 24 07 2020].
- Page, V., 2018. *Waze: The Pros and Cons*. [Online]
Available at: <https://www.investopedia.com/articles/investing/060415/pros-cons-waze.asp>
[Accessed 22 02 2020].
- Photo, D. O., 2016. *Information on Traffic Is Now Available in MAPS.ME*. [Online]
Available at: <https://medium.com/@Dimitryophoto/information-on-traffic-is-now-available-in-maps-me-for-ios-and-android-bb52ac04d822>
[Accessed 23 02 2020].

- Sterling, G., 2019. *Google brings the Assistant to Waze*. [Online] Available at: <https://searchengineland.com/google-brings-the-assistant-into-waze-317986> [Accessed 22 02 2020].
- The Independent Institute of Education (Pty) Ltd, 2020. *Open Source Coding (Intermediate) 7312 Question Paper*. 2 ed. Sandton: The Independent Institute of Education (Pty) Ltd.
- The Independent Institute of Education, 2020. *Open Source Coding (Intermediate)*, s.l.: The Independent Institute of Education.
- Trimble, n.d.. *How GPS works?*. [Online] Available at: https://www.trimble.com/gps_tutorial/howgps.aspx [Accessed 11 03 2020].
- UI Here, n.d.. *UI Here*. [Online] Available at: <https://www.uihere.com/free-cliparts/map-computer-icons-location-logo-2081762> [Accessed 19 02 2020].
- Waze, n.d.. *Banks of Servers*. [Online] Available at: https://wiki.waze.com/wiki/Banks_of_Servers [Accessed 22 02 2020].
- Waze, n.d.. *Waze - GPS, Maps, Traffic Alerts & Live Navigation*. [Online] Available at: <https://play.google.com/store/apps/details?id=com.waze&gl=ZA> [Accessed 22 02 2020].