GitHub Username: dstaflund

Saskatchewan Geo-Memorial Application

Description

The Saskatchewan Geo-Memorial Project is a Government of Saskatchewan initiative that names landmarks after residents who have died in service of Saskatchewan or Canada. Although the program originally concerned itself with casualties from World War II (1938 - 1945), it has been extended to include casualties from:

- The Boer Wars (1880 - 1881, 1895 - 1896, 1899 - 1902)
- World War I (1914 - 1918)
- Merchant Conflicts
- The Korean War (1950 - 1953)
- United Nations missions (such as Golan Heights, Bosnia, Rwanda, etc.)
- The War in Afghanistan (2001 - 2014)
- Incidents involving Police Officers and Emergency Responders

Geo-memorials are also used to honour those who have made important contributions to the Province of Saskatchewan. The initiative honours 3,922 residents at the present time.

Saskatchewan Geo-Memorial will be an Android application that people can use to learn more about the Saskatchewan Geo-Memorial Project and about its residents and geo-memorials in particular. Users will be able to search for residents by name, hometown, birth date, rank, obit date, etc. and will be shown on a map where their geo-memorials are located. Users will also be able to explore these geo-memorials on the map by zooming in and out as they pan across the Province.

Intended User

- Family, friends, and relatives of the fallen:
  - Who want to know where their loved ones grew up and when they died
  - Who want to know who they can contact to have plaques laid
  - (FUTURE) Who want to view or upload comments and photos
  - (FUTURE) Who want to learn where their ancestors are buried

- People with an interest in Saskatchewan:
  - Who want to learn how many of the Provinces lakes and rivers came to be named
  - Who want to learn who to contact to get more information on the Project

    - Who want to obtain links to related books and websites

- People with an interest in the military:
    - Who want to learn more about the fallen
    - (FUTURE) Who want to learn more about regiments that the fallen served with
    - (FUTURE) Who want to learn more about conflicts that the fallen died in

- People with an interest in history:
    - Who want to learn more about the fallen themselves
    - Who want to learn how many of Saskatchewan’s landmarks were named
    - (FUTURE) Who want to learn of the towns that lost citizens in battle.

- People with an interest in travel:
    - Who want to know of the Geo-Memorials in their surrounding area
    - Who want to view the NTS names and numbers of these Geo-Memorials
    - Who want to view the latitudes and longitudes of these Geo-Memorials
    - (FUTURE) Who want to be notified when as they approach Geo-Memorials of interest
    - (FUTURE) Who want to view and upload comments and photos

Features

At its heart, this is a search and mapping application:

- Search
    - For Residents:
        - By Name
        - By Birth Date
        - By Hometown
        - By Rank
        - By Geo-Memorial Name
        - By Obit. Date
    - For Geo-Memorial
        - By Geo-Memorial Name
        - By Resident Name
        - By Geographic Location
            - Latitude / Longitude
            - National Topographic System
                - NTS Series Name or Number
                - NTS Area Name or Letter
                - NTS Sheet Name of Number
    - Search History

        - View Past Searches
        - Run Past Searches
        - Delete Search History

- Map
    - Built-In Features:
        - Zoom In / Zoom Out
        - Terrain / Satellite / Hybrid Modes
        - Runtime Geo-Location
        - Off-Line Map Capabilities
        - Photo Upload
        - Commenting
    - Customized Markers
        - Displayed As Poppies
        - With Snippets Designed To Display:
            - Name of Hometown (If Marker Is For Hometown)
            - Name of Geo-Memorial and the Resident it was Named After (If Marker is for Geo-Memorial)
    - Polylines
        - To display bounded areas such as:
            - NTS Series
            - NTS Areas
            - NTS Sheets

In addition to these core features, users will also be able to do the following:

- Bookmark interesting geo-memorials
- Share geo-memorials with others
- Leave comments for the developer
- Store their map display preferences

As noted above, there are a number of features that I can't implement right now but hope to deliver in a future release:

- The ability to upload comments and photos
- The ability to sign up for and receive push notifications when comments and photos are uploaded
- The ability to view the following additional information on honoured residents:
    - Photos
    - Biographies
    - Regiment descriptions
    - Hometown descriptions
    - Conflict descriptions

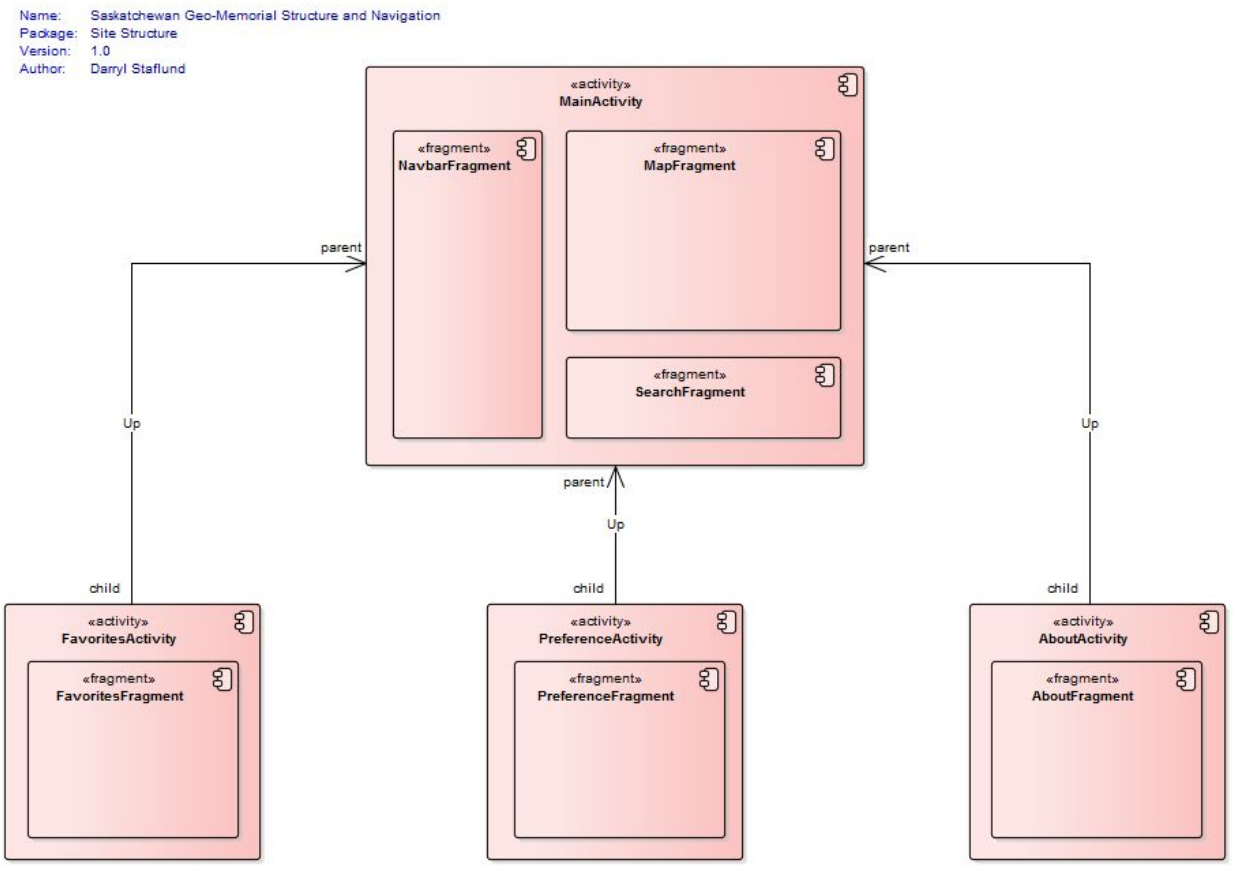
- ○ Cemetery descriptions

I am not able to implement these features right now because of the sheer magnitude of the work involved in trying to track down photos, biographies, and so on.

User Interface Mocks

1. Saskatchewan Geo-Memorial Site / Navigation Map

The following is a combination site / navigation map of the Saskatchewan Geo-Memorial application It shows the app's activities, the fragments they coordinate, and the navigation relationships between activities.

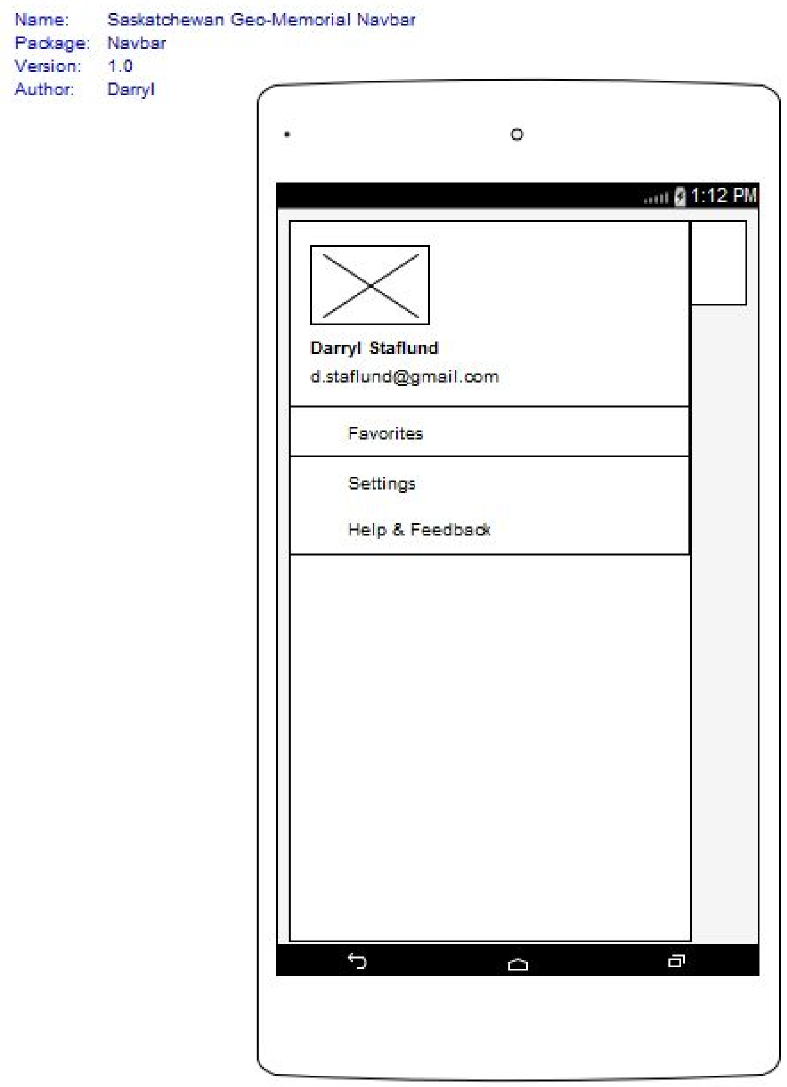

Name: Saskatchewan Geo-Memorial Structure and Navigation
Package: Site Structure
Version: 1.0
Author: Darryl Staflund
«activity» MainActivity
«fragment» NavbarFragment
«fragment» MapFragment
«fragment» SearchFragment
parent
Up
child
parent
Up
child
parent
Up
child
«activity» FavoritesActivity
«fragment» FavoritesFragment
«activity» PreferenceActivity
«fragment» PreferenceFragment
«activity» AboutActivity
«fragment» AboutFragment

2. Navbar

Although the navbar is part of the MainActivity, I discuss it here apart from the activity since it serves as the central point from which users can do the following:

- Log onto the application
- View and manage their favorite geo-locations
- View and manage their preferences
- Get build and help information on the Saskatchewan Geo-Memorial application

The navbar will be structured as follows:



Name:	Saskatchewan Geo-Memorial Navbar
Package:	Navbar
Version:	1.0
Author:	Darryl

The navbar will work as follows:

- The upper part of the navbar will show the name of the application if the user hasn't logged onto Google Sign-In, or it will show, as in this case, an icon assigned with the user's account along with their name and email address. Users will be able to log on or off from their account in this part of the navbar.

- When the Favorites item is pressed, users will be brought to the Favorites activity where they will be able to view their favorite geo-memorials, or remove them from the list.

- When the Settings item is pressed, users will be brought to the Preferences activity where they will be able to view and modify their application settings.

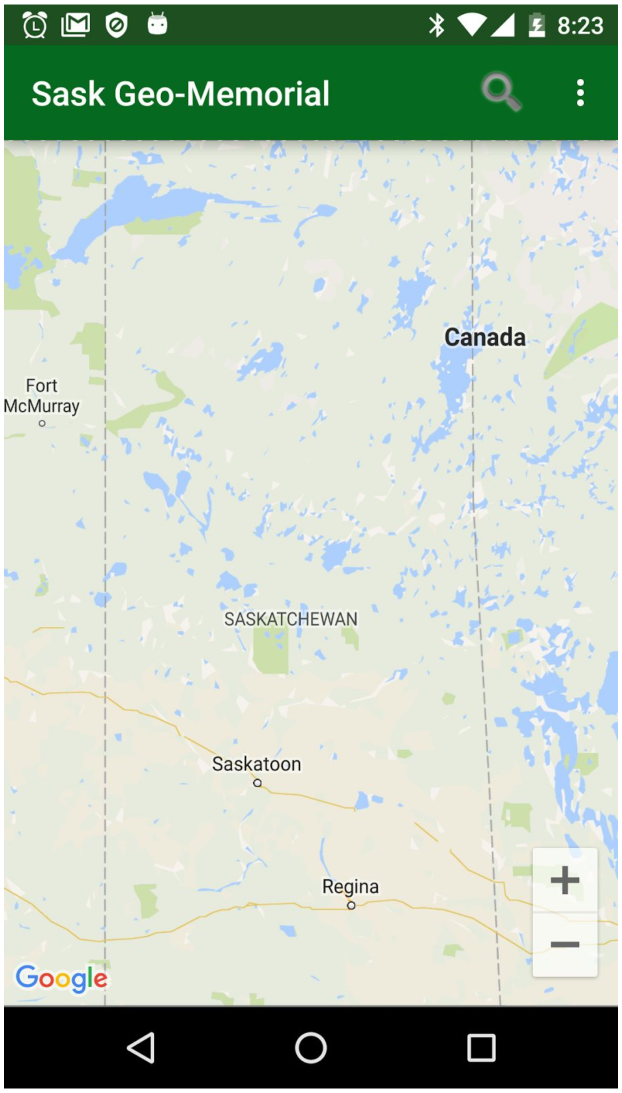
- When the Help & Feedback item is pressed, users will be brought to the About activity where they can see build and contact information about the app as well as acknowledgements, links, and help information.

3. Main Activity

This is the heart of the Saskatchewan Geo-Memorial application as it is from the MainActivity that users will be able to view and interact with the map as well as search for geo-memorials. The following are screenshots of the actual MainActivity as it looks in development so far, highlighting some of its features:

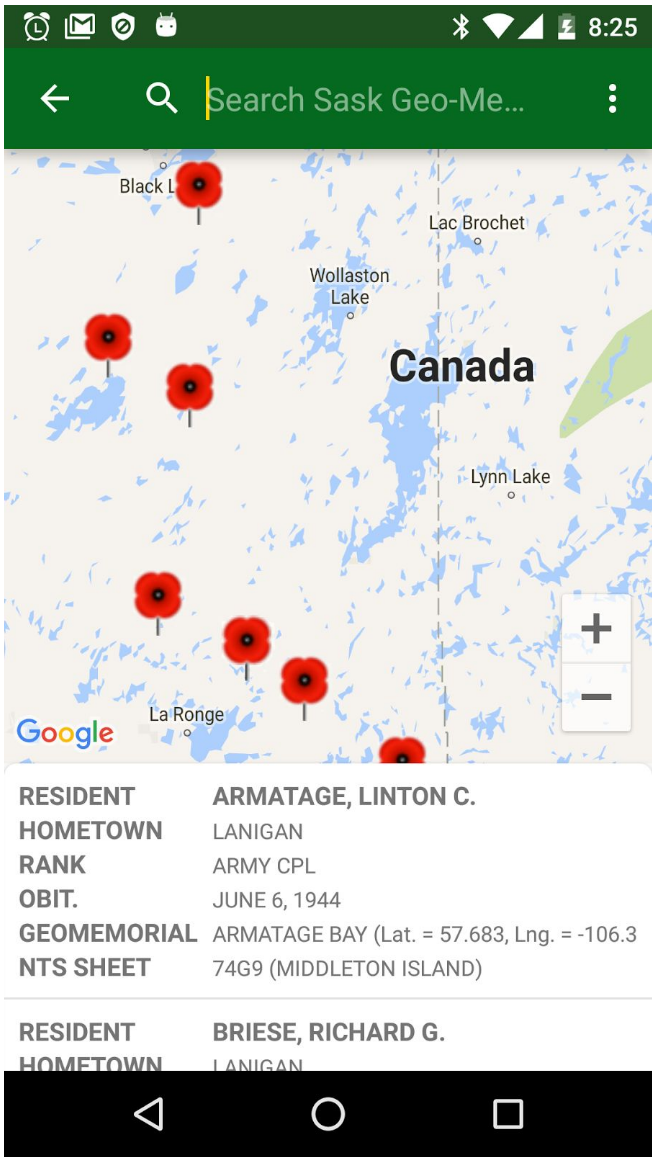
Initial View

This is how the MainActivity will look after the user starts the application. The entire Province of Saskatchewan is shown, but it doesn't contain markers or search results since searches haven't been performed yet.



Subsequent View

This is how the MainActivity will look after the user has performed some searches. Poppies will mark the geo-memorials of the last search, and information on the residents honoured by these memorials will be visible.

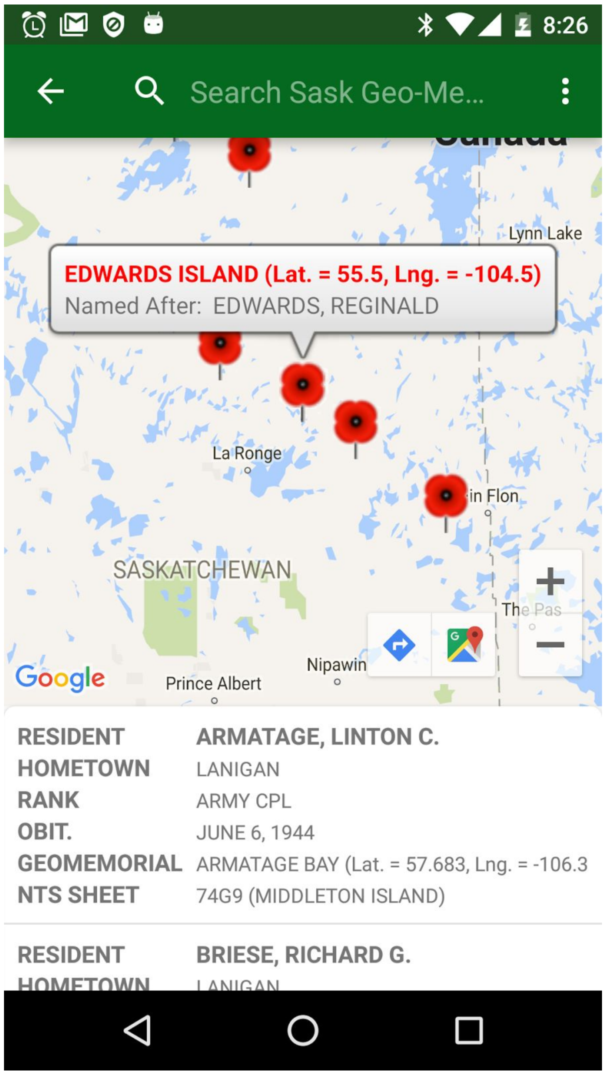


RESIDENT	ARMATAGE, LINTON C.
HOMETOWN	LANIGAN
RANK	ARMY CPL
OBIT.	JUNE 6, 1944
GEOMEMORIAL	ARMATAGE BAY (Lat. = 57.683, Lng. = -106.3
NTS SHEET	74G9 (MIDDLETON ISLAND)
RESIDENT	BRIESE, RICHARD G.
HOMETOWN	LANIGAN

Although I don’t depict it here, each of the search results will have Favorite and Share buttons that users can press to mark a resident / geo-memorial as a favorite, or share the resident / geo-memorial with someone else.

Viewing Information about a Geo-Memorial Marker

When a user touches one of the geo-memorial markers, an information balloon will be opened displaying the name of the geo-memorial, its latitude and longitude, as well as the name of the resident it was named after.

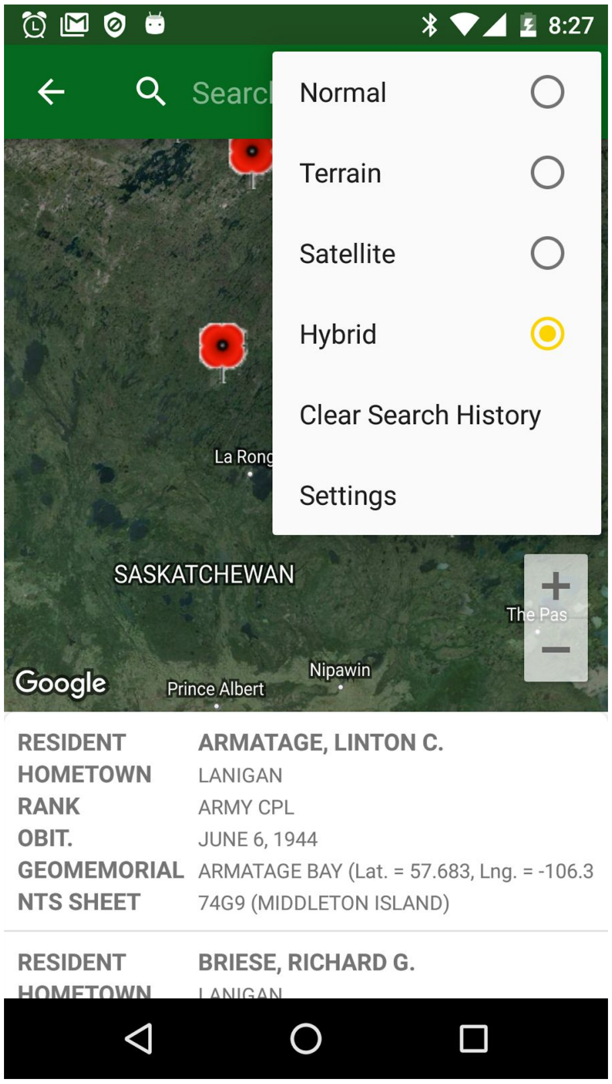


EDWARDS ISLAND (Lat. = 55.5, Lng. = -104.5)
Named After: EDWARDS, REGINALD

RESIDENT	ARMATAGE, LINTON C.
HOMETOWN	LANIGAN
RANK	ARMY CPL
OBIT.	JUNE 6, 1944
GEOMEMORIAL	ARMATAGE BAY (Lat. = 57.683, Lng. = -106.3
NTS SHEET	74G9 (MIDDLETON ISLAND)
RESIDENT	BRIESE, RICHARD G.
HOMETOWN	LANIGAN

Modifying the Map Type

Users will be able to select the type of map shown by the application. In this particular screenshot, the user is shown a hybrid map immediately after selecting Hybrid in the drop-down menu.

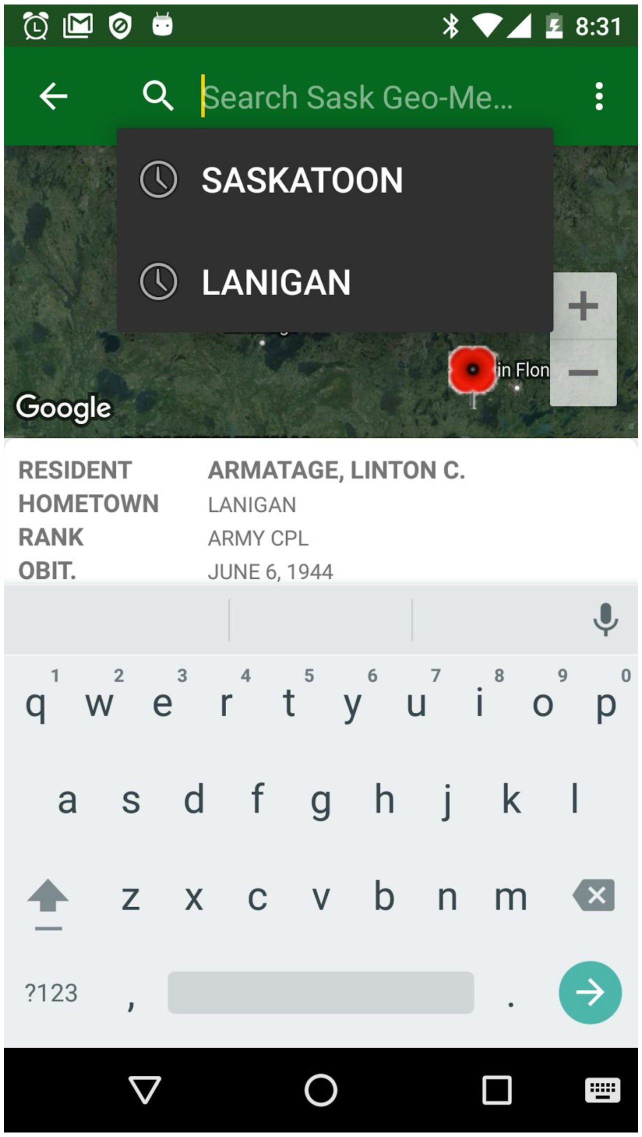


Normal ○
Terrain ○
Satellite ○
[✓ Hybrid] ●

RESIDENT	ARMATAGE, LINTON C.
HOMETOWN	LANIGAN
RANK	ARMY CPL
OBIT.	JUNE 6, 1944
GEOMEMORIAL	ARMATAGE BAY (Lat. = 57.683, Lng. = -106.3
NTS SHEET	74G9 (MIDDLETON ISLAND)
RESIDENT	BRIESE, RICHARD G.
HOMETOWN	LANIGAN

Viewing Search History

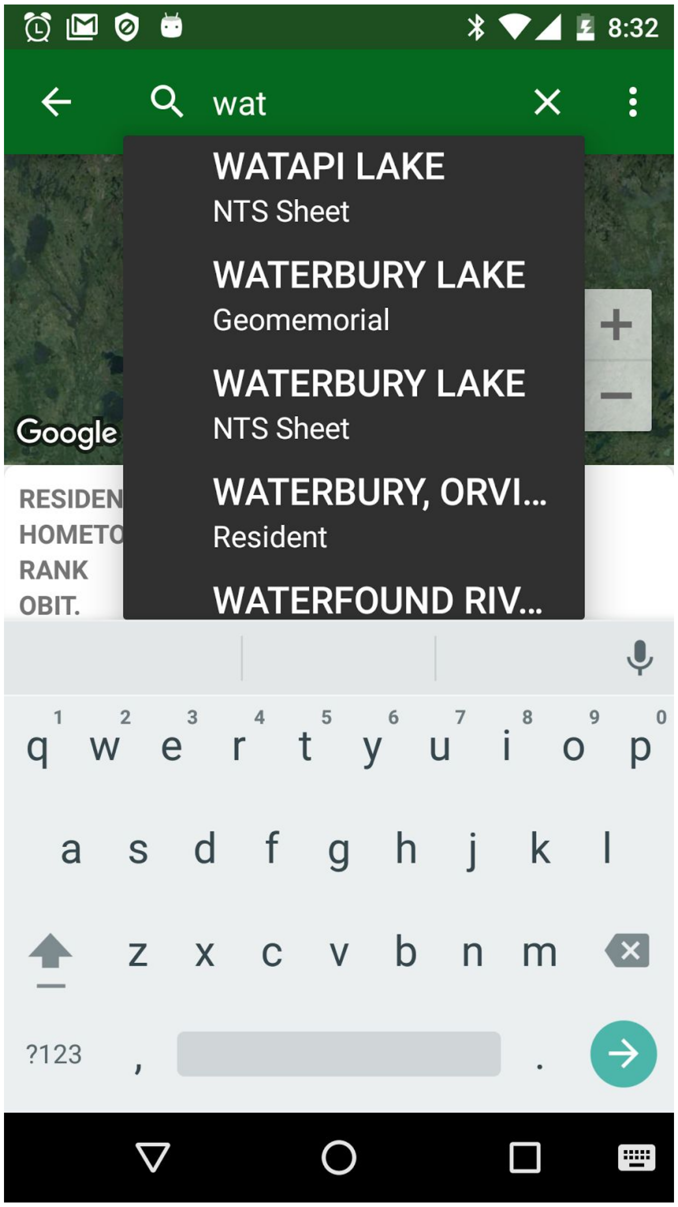
When a user starts a new search, a list of their past searches will be shown to them. In this screenshot, the user has performed two previous searches -- one on the city of Saskatoon, and one on the town of Lanigan.



RESIDENT	ARMATAGE, LINTON C.
HOMETOWN	LANIGAN
RANK	ARMY CPL
OBIT.	JUNE 6, 1944

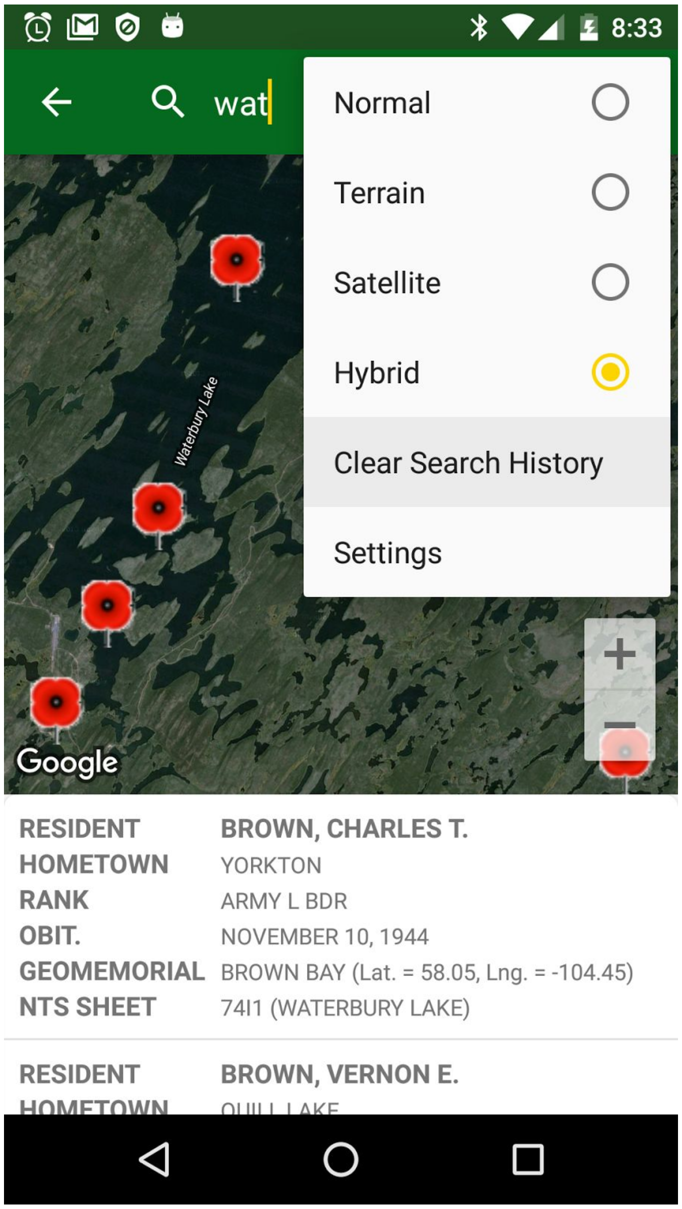
Viewing Search Suggestions

While a user is entering their search criteria, real-time categorized search suggestions are displayed. In this screenshot, the user has typed in 'wat' as part of their search criteria, and two NTS sheets, a Geomemorial, and a Resident are shown as search suggestions.



Clearing Search History

Users are able to clear their search history from the drop-down menu.



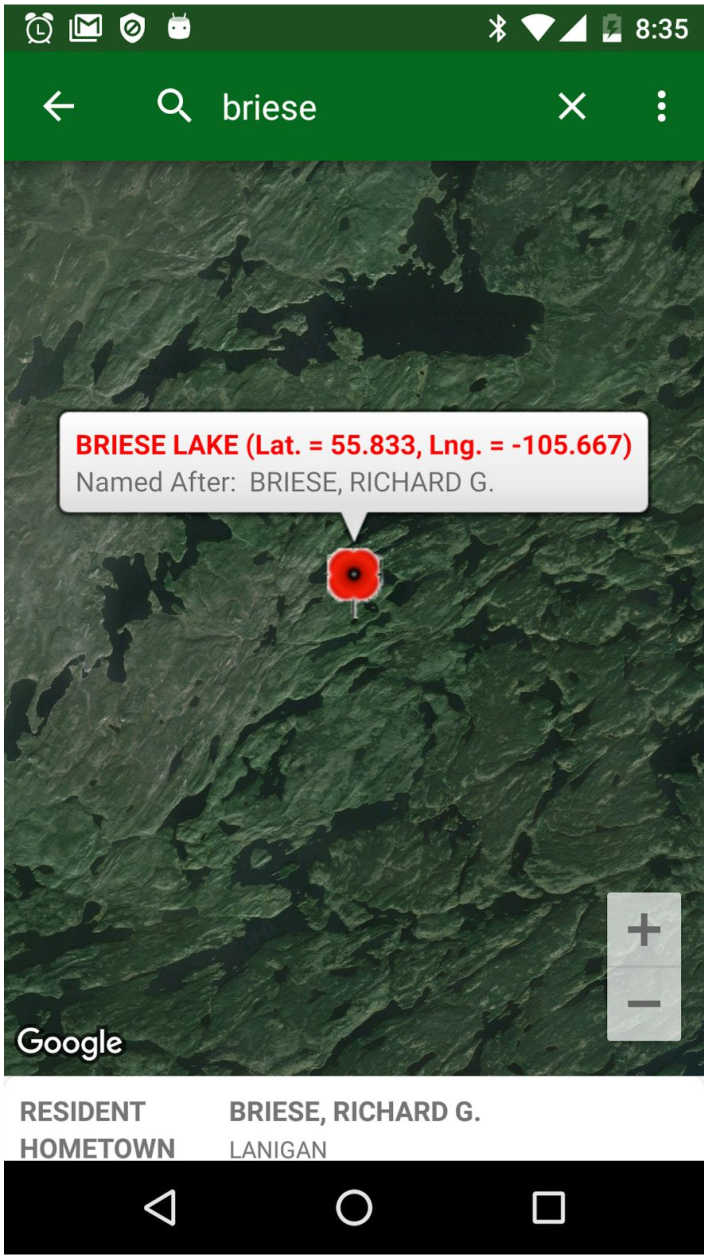
Normal ○
Terrain ○
Satellite ○
Hybrid ●
Clear Search History
Settings

RESIDENT	BROWN, CHARLES T.
HOMETOWN	YORKTON
RANK	ARMY L BDR
OBIT.	NOVEMBER 10, 1944
GEOMEMORIAL	BROWN BAY (Lat. = 58.05, Lng. = -104.45)
NTS SHEET	74I1 (WATERBURY LAKE)
RESIDENT	BROWN, VERNON E.
HOMETOWN	QUILL LAKE

NB: I will be removing ‘Settings’ from this menu once I code the navbar.

Displaying a Single Search Result

If a single geo-memorial is displayed as a search result, the map will move to the memorial's location, but it won't zoom to a level of 21, since nine times out of ten that will have you looking at a lake surface without shorelines on the side (NB: Saskatchewan has some VERY big lakes in the north.) Instead, the application will zoom to show a bounded area covering a ¼ degree latitude and ½ degree longitude (i.e. the size of an NTS Sheet -- more on this later in the document) with the geo-memorial centered within it. This strategy will show the location of the geo-memorial to the user as well as its position relative to other lakes and landmarks in the area.

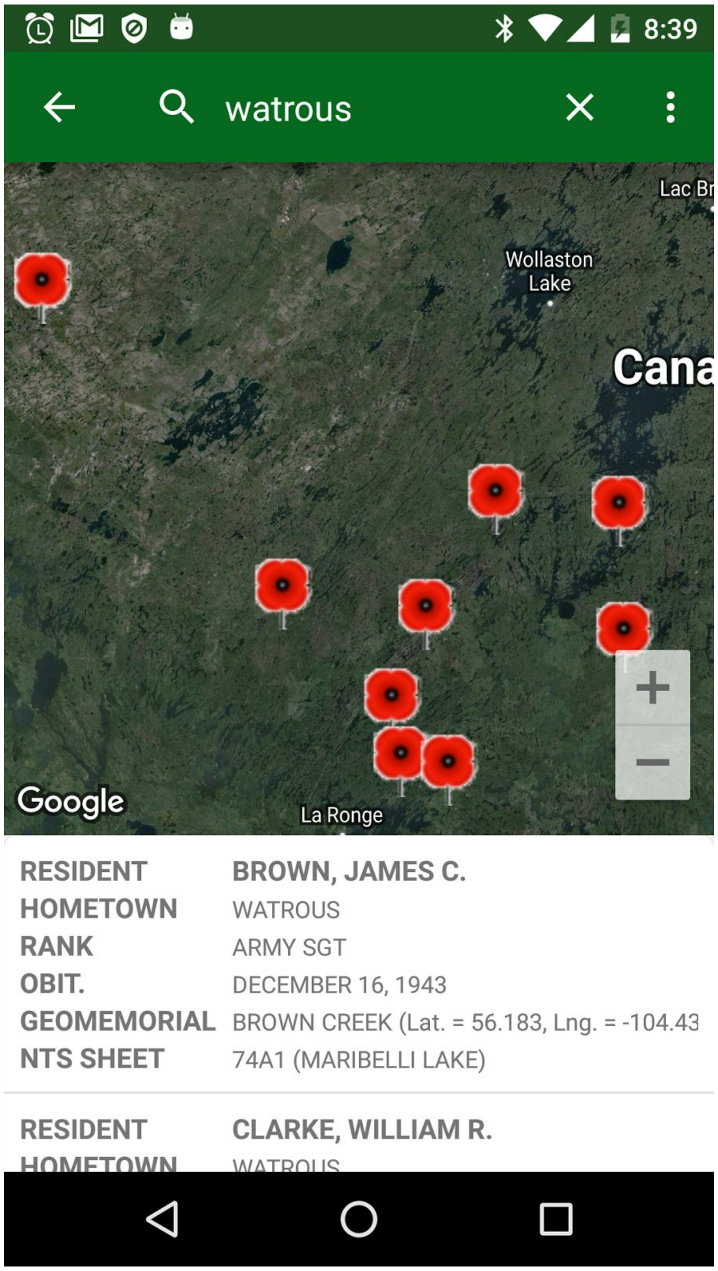


BRIESE LAKE (Lat. = 55.833, Lng. = -105.667)
Named After: BRIESE, RICHARD G.

RESIDENT	BRIESE, RICHARD G.
HOMETOWN	LANIGAN

Displaying Multiple Search Results

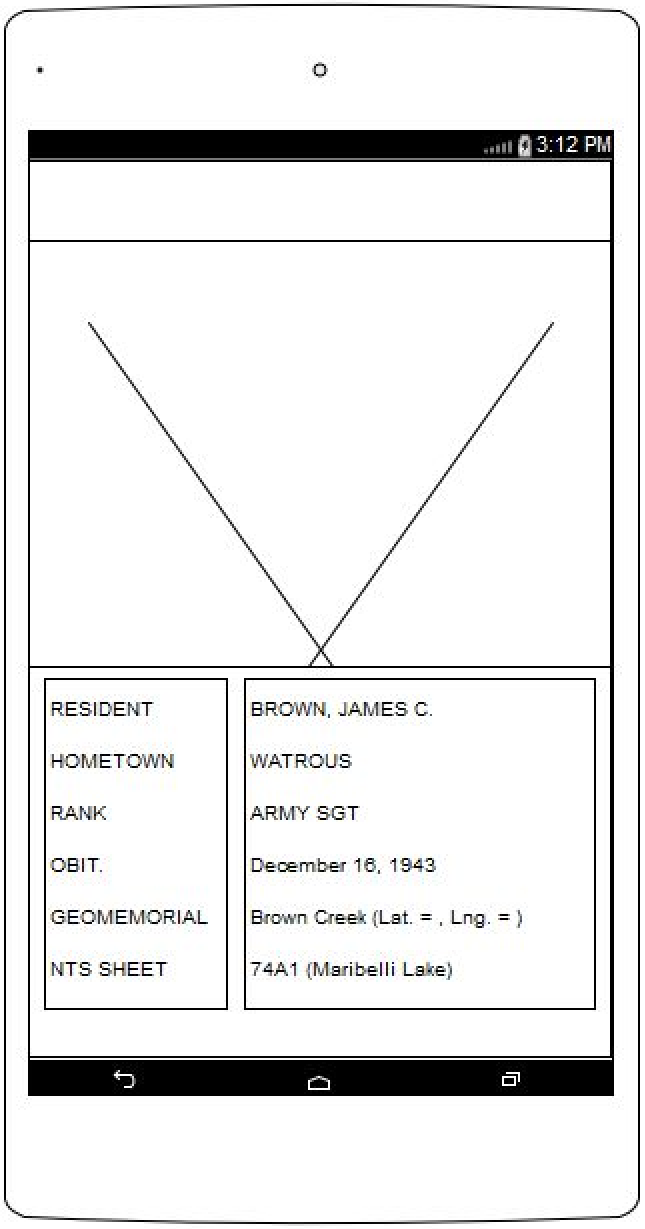
If more than one geo-memorial is displayed as a search result, an area bounding all of the geo-memorials will be shown to the user, as seen in this screenshot.



RESIDENT	BROWN, JAMES C.
HOMETOWN	WATROUS
RANK	ARMY SGT
OBIT.	DECEMBER 16, 1943
GEOMEMORIAL	BROWN CREEK (Lat. = 56.183, Lng. = -104.43
NTS SHEET	74A1 (MARIBELLI LAKE)
RESIDENT	CLARKE, WILLIAM R.
HOMETOWN	WATROUS

Main Activity Layout in Portrait Mode (Phone)

This wireframe shows the general layout of the MainActivity as it will appear on a Phone in portrait mode.

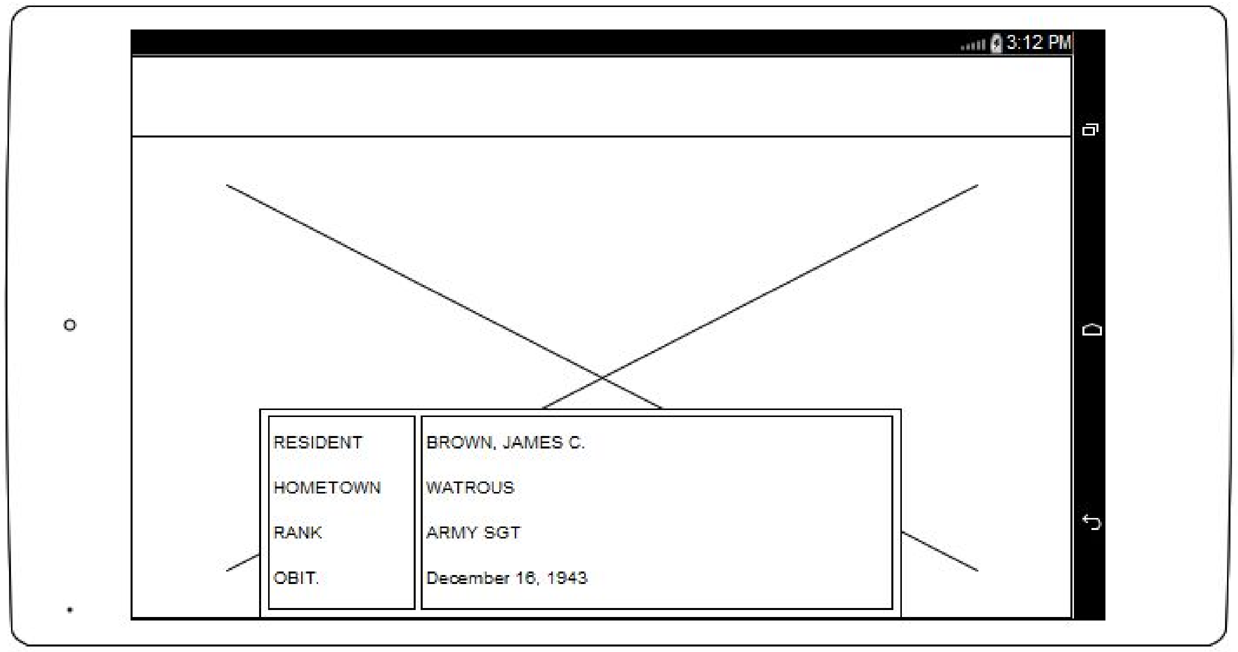


At the top of the screen is the app bar that users can enter their searches in or select a menu option. A Google Map control covers rest of the screen, but a Search fragment anchored to the bottom of the screen is laid on top of it.

NB: My deepest apologies for the quality of the diagram -- I am using Enterprise Architect as my diagramming tool, and, although it's a superb tool when it comes to drawing UML diagrams and workflow diagrams, its Android wireframing features are severely lacking in the current release.

Main Activity Layout in Landscape Mode (Phone)

This wireframe shows the general layout of the MainActivity as it will appear on a Phone in landscape mode.

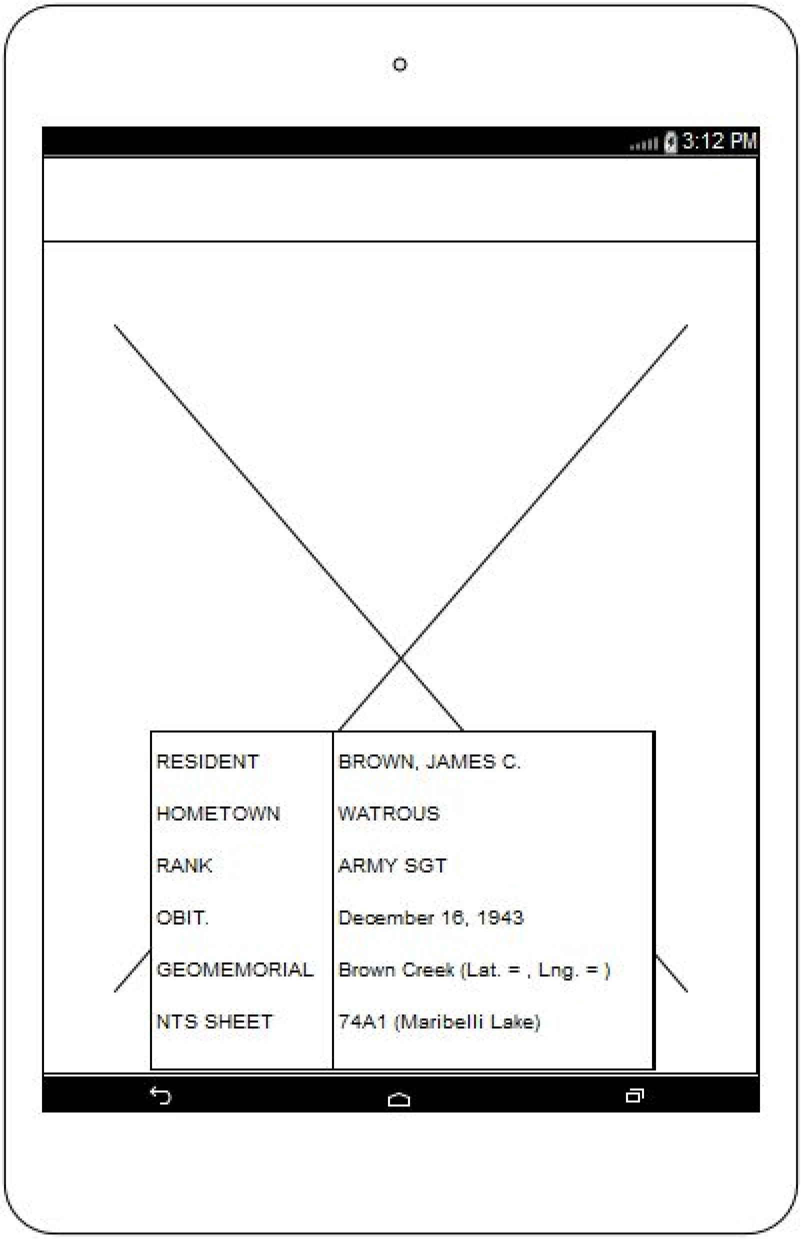


RESIDENT	BROWN, JAMES C.
HOMETOWN	WATROUS
RANK	ARMY SGT
OBIT.	December 16, 1943

This layout is similar in all respects to the portrait layout, except that the search fragment is given substantial left and right margins so as to appear centered on the device. This is to prevent enormous amounts of whitespace from appearing to the right of the search fragment, given the long form factor it has to fill.

Main Activity Layout in Portrait Mode (Tablet)

This wireframe shows the general layout of the MainActivity as it will appear on a Tablet in portrait mode.

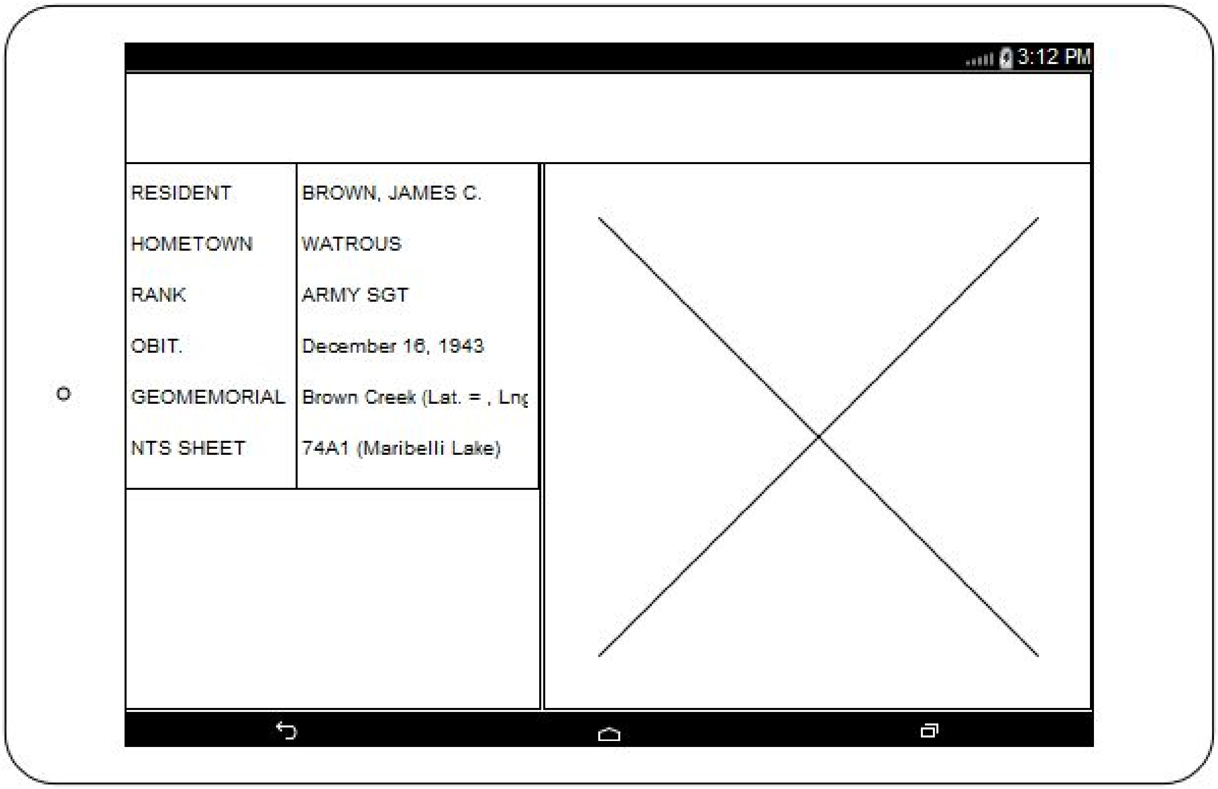


RESIDENT	BROWN, JAMES C.
HOMETOWN	WATROUS
RANK	ARMY SGT
OBIT.	December 16, 1943
GEOMEMORIAL	Brown Creek (Lat. = , Lng. =)
NTS SHEET	74A1 (Maribelli Lake)

This layout is similar in all respects to the phone’s landscape mode, except that the map fragment is taller than it is wide.

Main Activity Layout in Landscape Mode (Tablet)

This wireframe shows the general layout of the MainActivity as it will appear on a Tablet in landscape mode.



RESIDENT	BROWN, JAMES C.
HOMETOWN	WATROUS
RANK	ARMY SGT
OBIT.	December 16, 1943
GEOMEMORIAL	Brown Creek (Lat. = , Lng
NTS SHEET	74A1 (Maribelli Lake)

Unlike the other layouts, the search and map fragments here are positioned one right after another. The search fragment has a width of approximately 40% of the screen, while the map fragment takes up the remaining 60%.

4. Favorites Activity

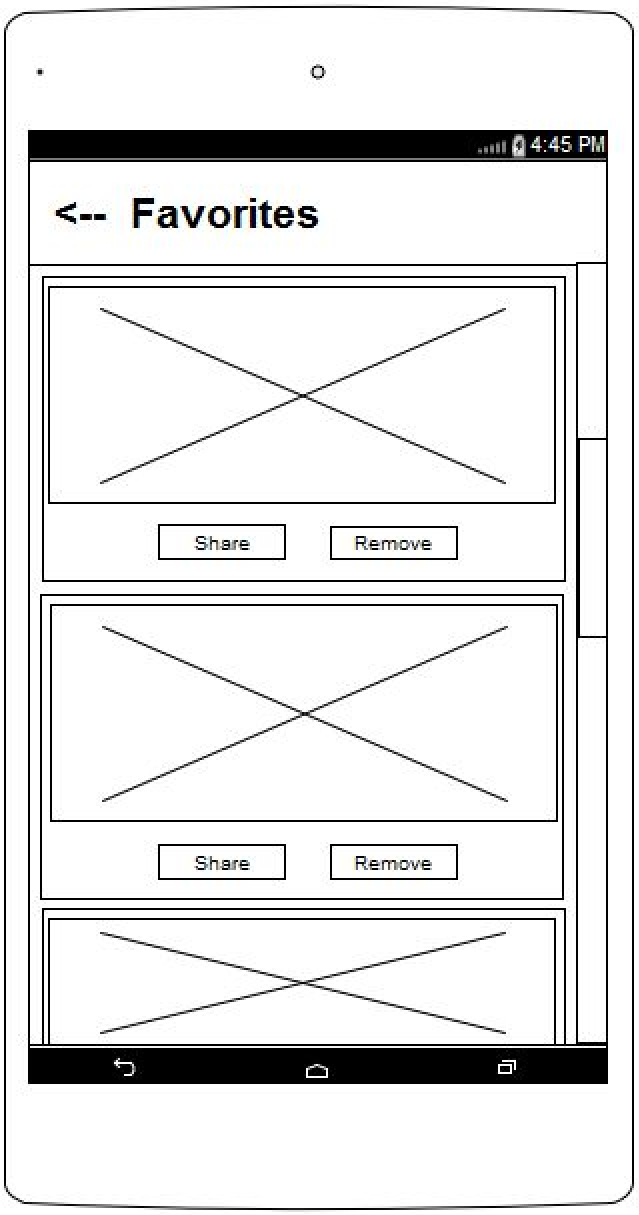
The Favorites Activity can be reached from the Navbar and is the place where users can view the geo-memorials they have selected as favorites, as well as share or remove them. Although the Favorites Activity is fairly straightforward in its design, its layout differs for both phones and tablets and for portrait and landscape modes.

The Share Button, when pressed, will create a Share action containing information on the selected geo-memorial such as its name and location as well as the resident it was named after.

The Remove Button, when pressed, will remove the geo-memorial from the user's list of favorites.

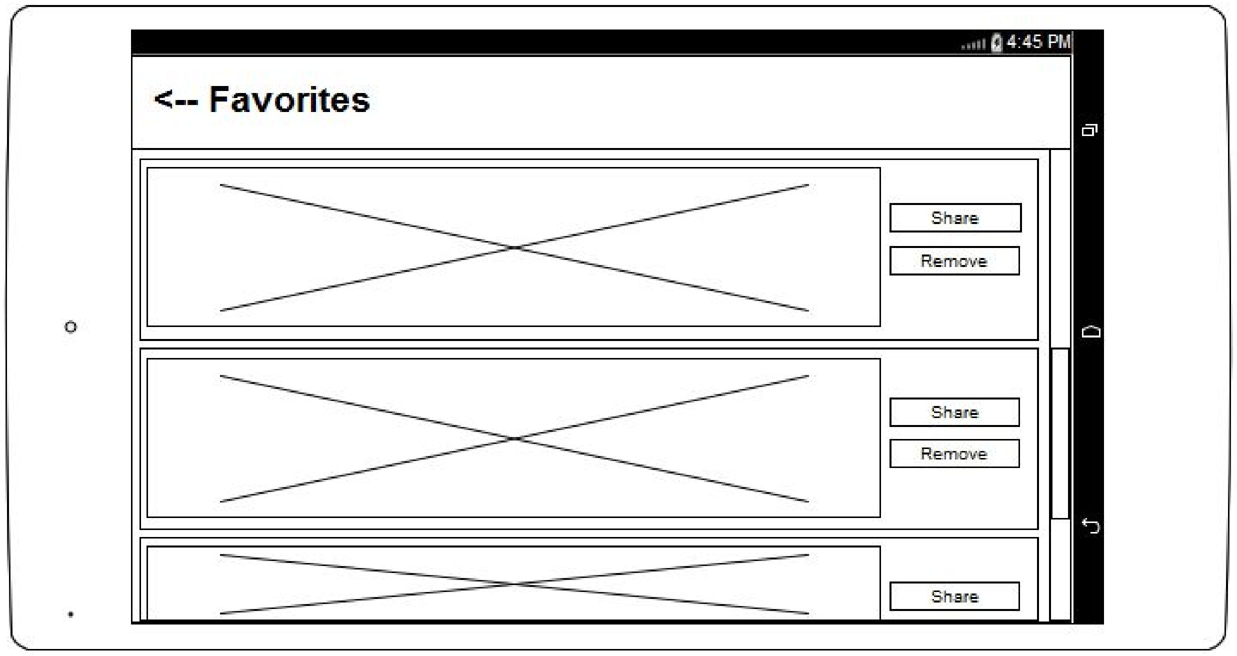
Favorites Activity in Portrait Mode (Phone)

This is a wireframe of the Favorites Activity as it will appear on a Phone being held in Portrait mode. It's essentially a vertical linear layout containing a scrollable list-view that displays maps of each favorite geo-memorial, as well as Share and Remove buttons. Since the width of a phone in portrait mode is at a premium, the Share and Remove buttons are positioned underneath their respective map controls.



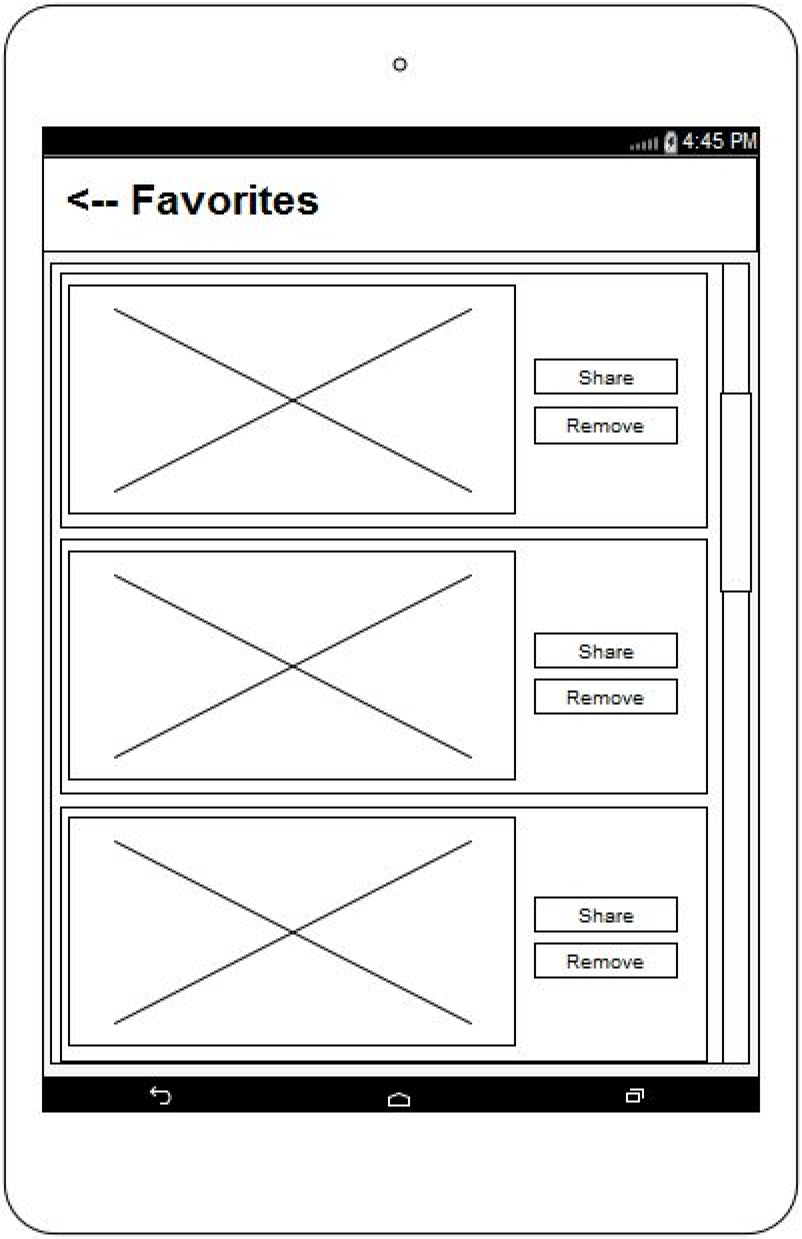
Favorites Activity in Landscape Mode (Phone)

This is a wireframe of the Favorites Activity as it will appear on phones being held in Landscape mode. Like the previous wireframe, this too is a vertical linear list containing a scrollable list-view of favorite geo-memorials. But since width is a little easier to come by in landscape mode, the Share and Remove buttons are displayed at the end of the map controls.



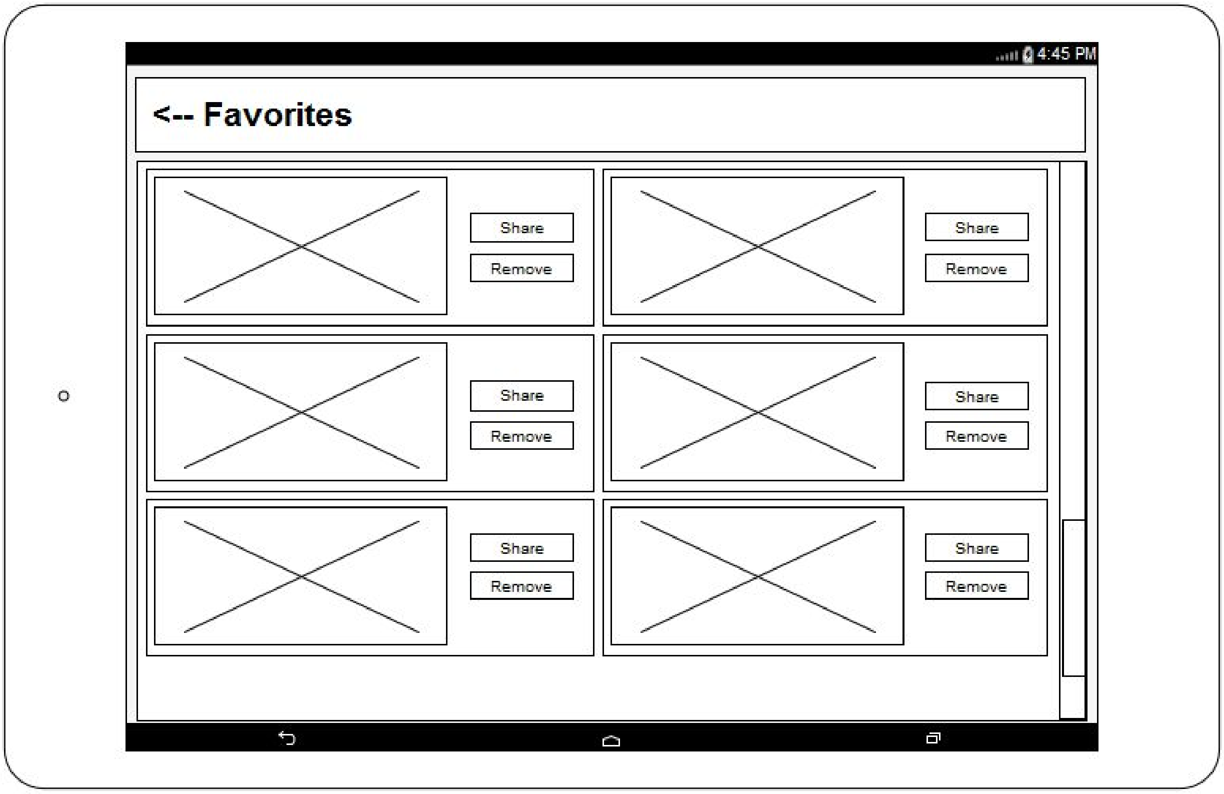
Favorites Activity in Portrait Mode (Tablet)

This is a wireframe of the Favorites Activity as it will appear on tablets being held in Portrait mode. It doesn't different in any substantial way from the layout of the Favorites Activity that appears on Phones being held in Landscape mode.



Favorites Activity in Landscape Mode (Tablet)

This is a wireframe of the Favorites Activity as it will appear on tablets being held in Landscape mode. It differs from the previous three layouts in that a two-column grid layout is used to display the favorites instead of a vertical linear layout. Although I picture the Share and Remove buttons at the end of the map controls, I may opt to display them underneath the map controls if things get a little too crowded -- I am not sure on what the best approach is yet.



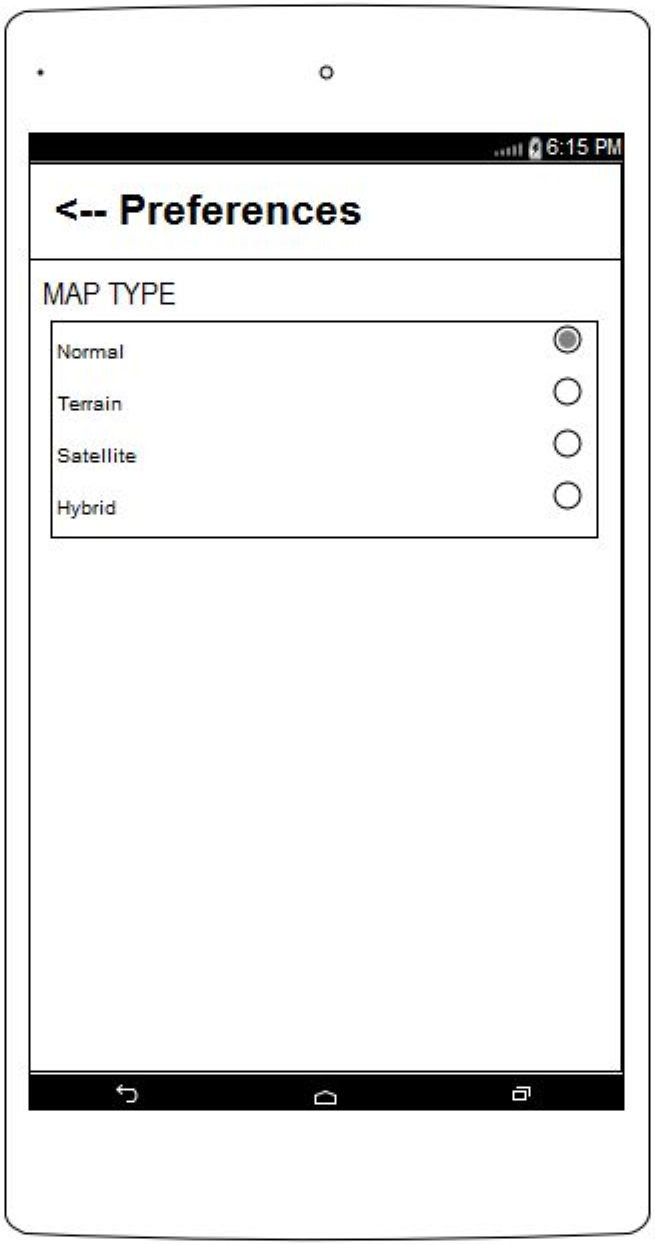
5. Preferences Activity

The Preferences Activity can be reached from the Navbar and is the place where users can set their application preferences. Because there is only one preference that can be set at the present time -- Map Type -- the screen has a simple design that doesn't change across phones or tablets, portrait modes or display modes.

When the user goes to the Preferences page, they will be asked to select one of four map types to render maps in. Once they have selected a Map Type, all they need to do is press the Up button to return to the Main Activity.

Preferences Activity in Portrait Mode (Phone)

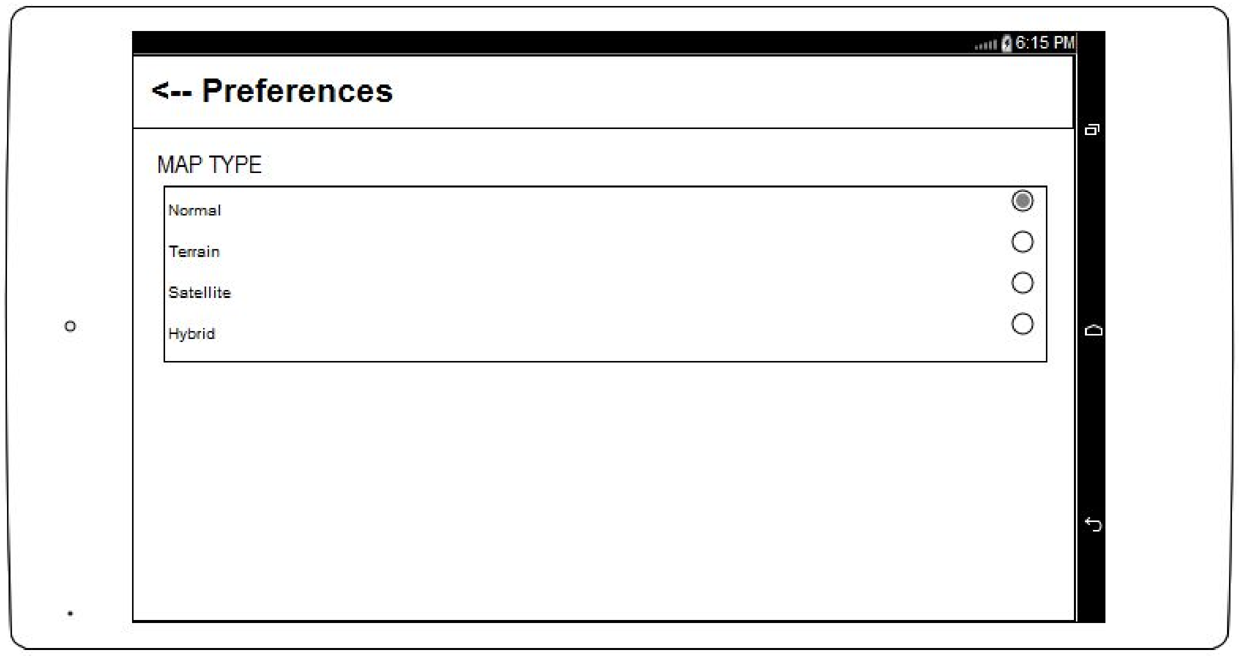
This is a wireframe of the Preferences Activity as it will appear on phones being held in portrait mode.



MAP TYPE: [✓ Normal] Terrain Satellite Hybrid

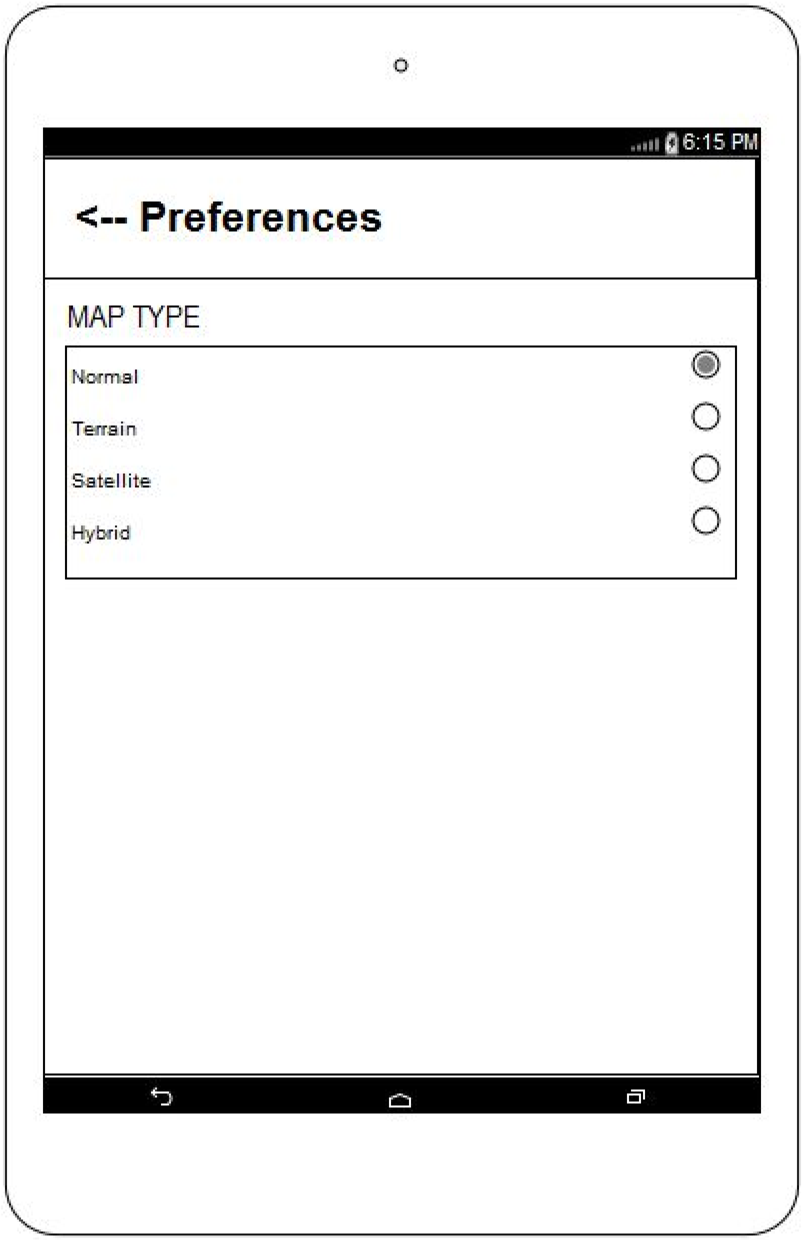
Preferences Activity in Landscape Mode (Phone)

This is a wireframe of the Preferences Activity as it will appear on phones being held in landscape mode.



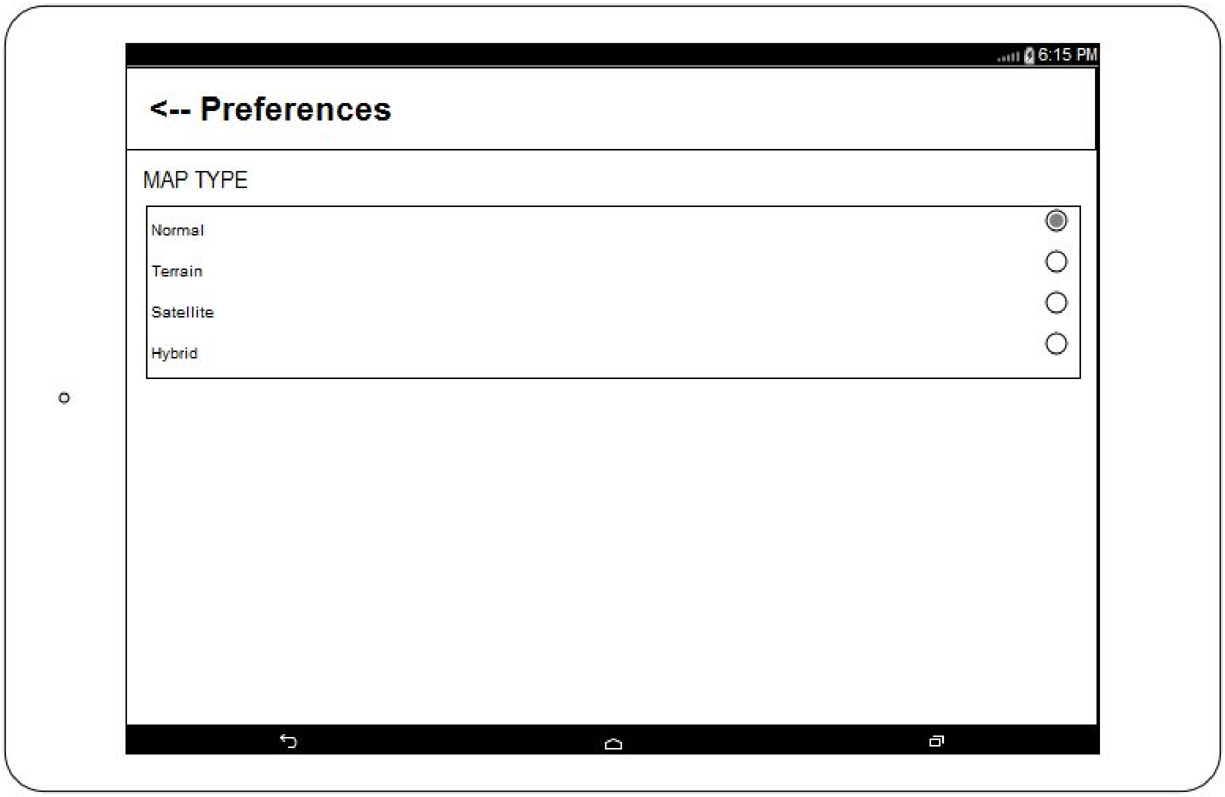
Preferences Activity in Portrait Mode (Tablet)

This is a wireframe of the Preferences Activity as it will appear on tablets being held in portrait mode.



Preferences Activity in Landscape Mode (Tablet)

This is a wireframe of the Preferences Activity as it will appear on tablets being held in landscape mode.



<-- Preferences

MAP TYPE

Normal [✓]
Terrain
Satellite
Hybrid

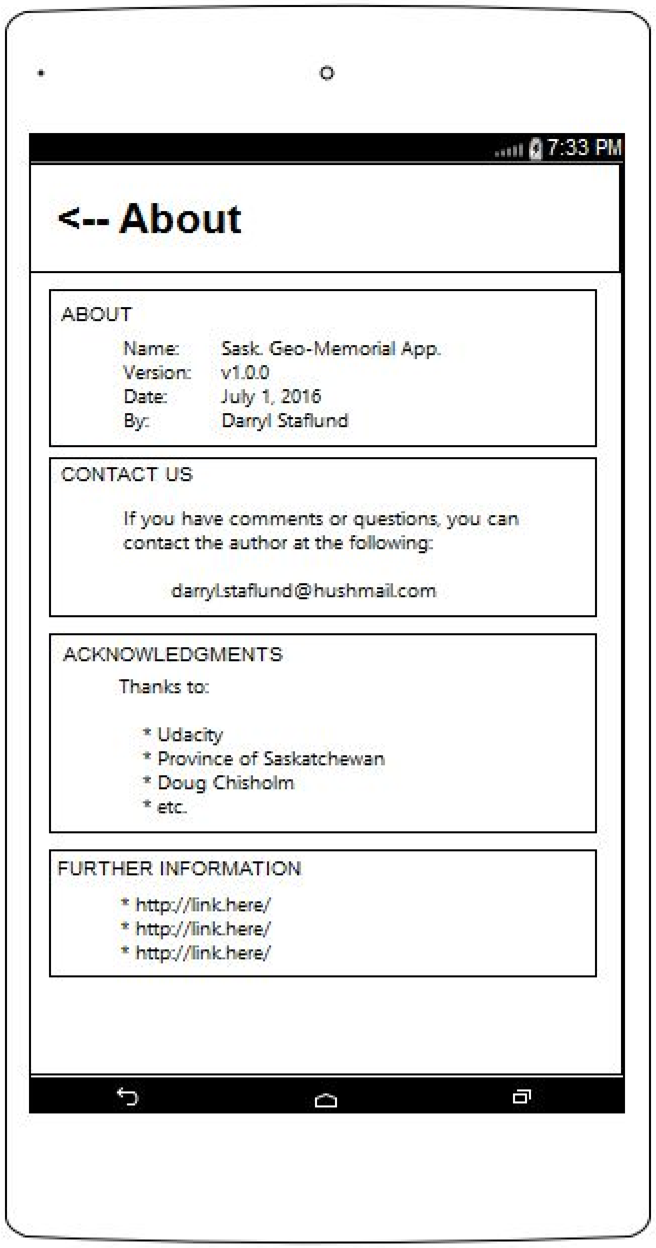
6. About Activity

The About Activity can be reached from the Navbar and is the place where users can view build, acknowledgement, and contact information as well as find links to more sites about the Saskatchewan Geo-Memorial project. Help information will also be visible although I haven't yet decided what the help section will look like.

With the exception of some odd links on this activity, this is a read-only page that the end user can leave by pressing the Up button to return to the Main Activity.

About Activity in Portrait Mode (Phone)

This is a wireframe of the Help Activity as it will appear on phones being held in portrait mode. It uses a vertical linear layout to display four types types of information to the user.



<-- About

ABOUT

Name:	Sask. Geo-Memorial App.
Version:	v1.0.0
Date:	July 1, 2016
By:	Darryl Staflund

CONTACT US

If you have comments or questions, you can contact the author at the following:

darryl.staflund@hushmail.com

ACKNOWLEDGMENTS

Thanks to:

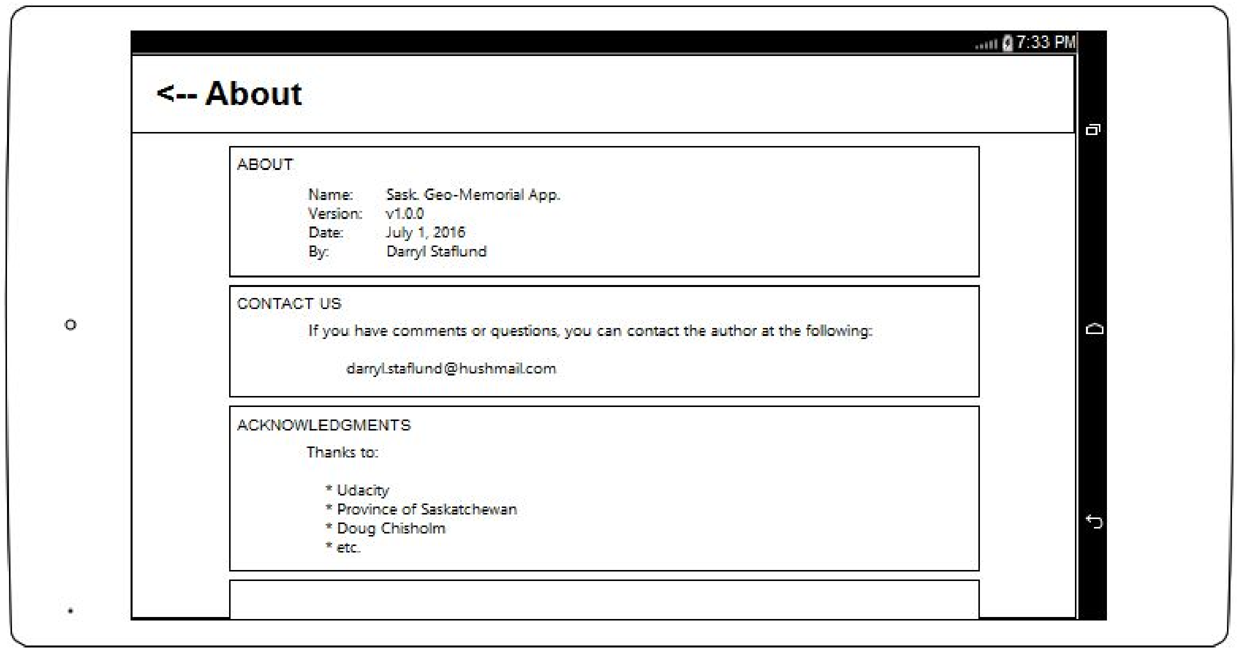
* Udacity
* Province of Saskatchewan
* Doug Chisholm
* etc.

FURTHER INFORMATION

* http://link.here/
* http://link.here/
* http://link.here/

About Activity in Landscape Mode (Phone)

This is a wireframe of the Help Activity as it will appear on phones being held in landscape mode. This layout is similar to the previous one, except that wide margins are used on both side of the liner list to prevent large swaths of whitespace at the end of each list item.



<-- About

ABOUT

Name:	Sask. Geo-Memorial App.
Version:	v1.0.0
Date:	July 1, 2016
By:	Darryl Staflund

CONTACT US

If you have comments or questions, you can contact the author at the following:

darryl.staflund@hushmail.com

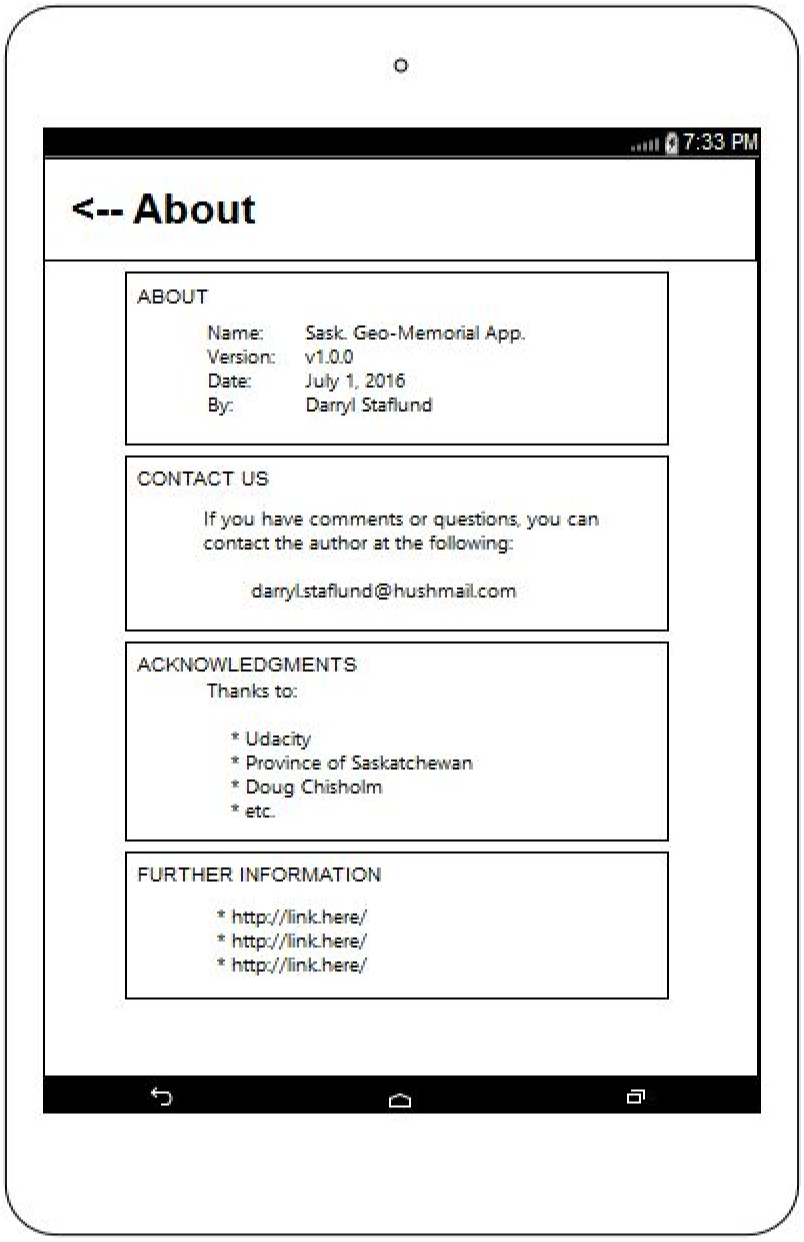
ACKNOWLEDGMENTS

Thanks to:

* Udacity
* Province of Saskatchewan
* Doug Chisholm
* etc.

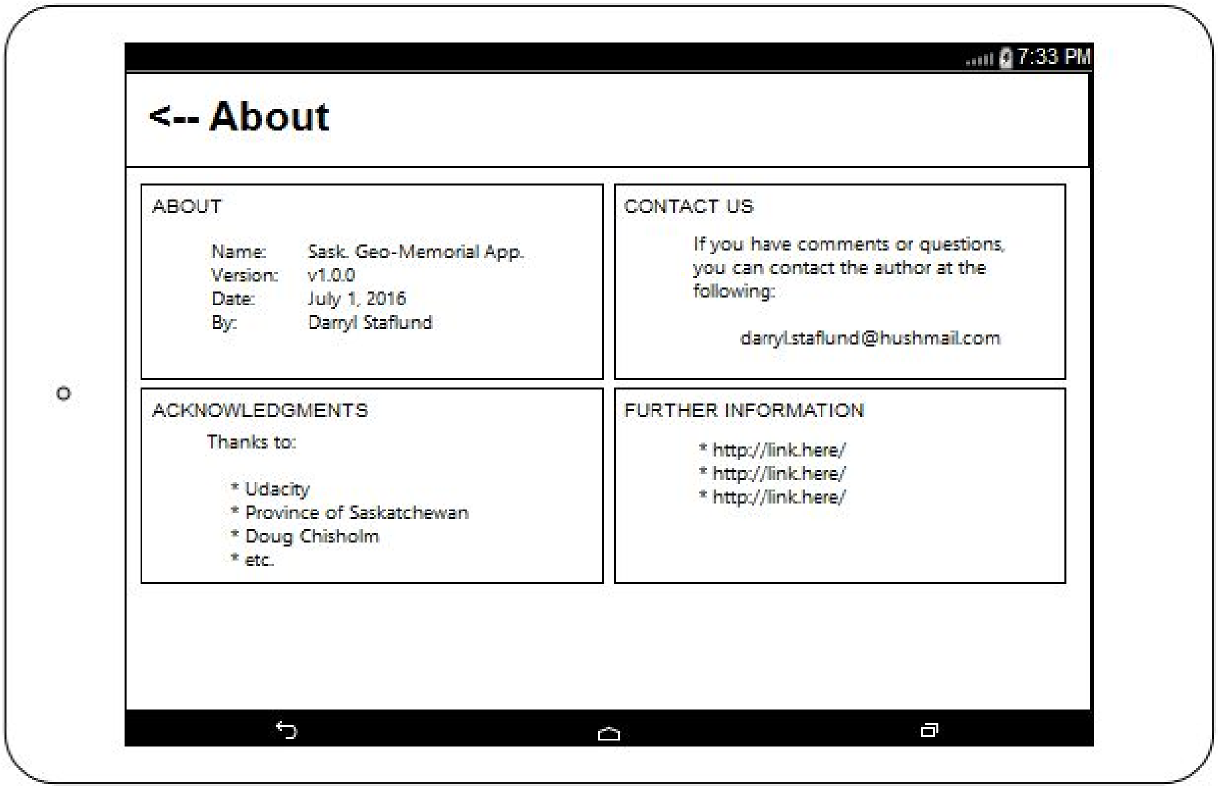
About Activity in Portrait Mode (Tablet)

This is a wireframe of the Help Activity as it will appear on tablets being held in portrait mode. This layout is similar to the phone's layout in landscape mode, except that the orientation is longer than wide.



About Activity in Landscape Mode (Tablet)

This is a wireframe of the Help Activity as it will appear on tablets being held in landscape mode. In this layout, the vertical linear layout is replaced with a two-column grid layout in order to make better use of space.



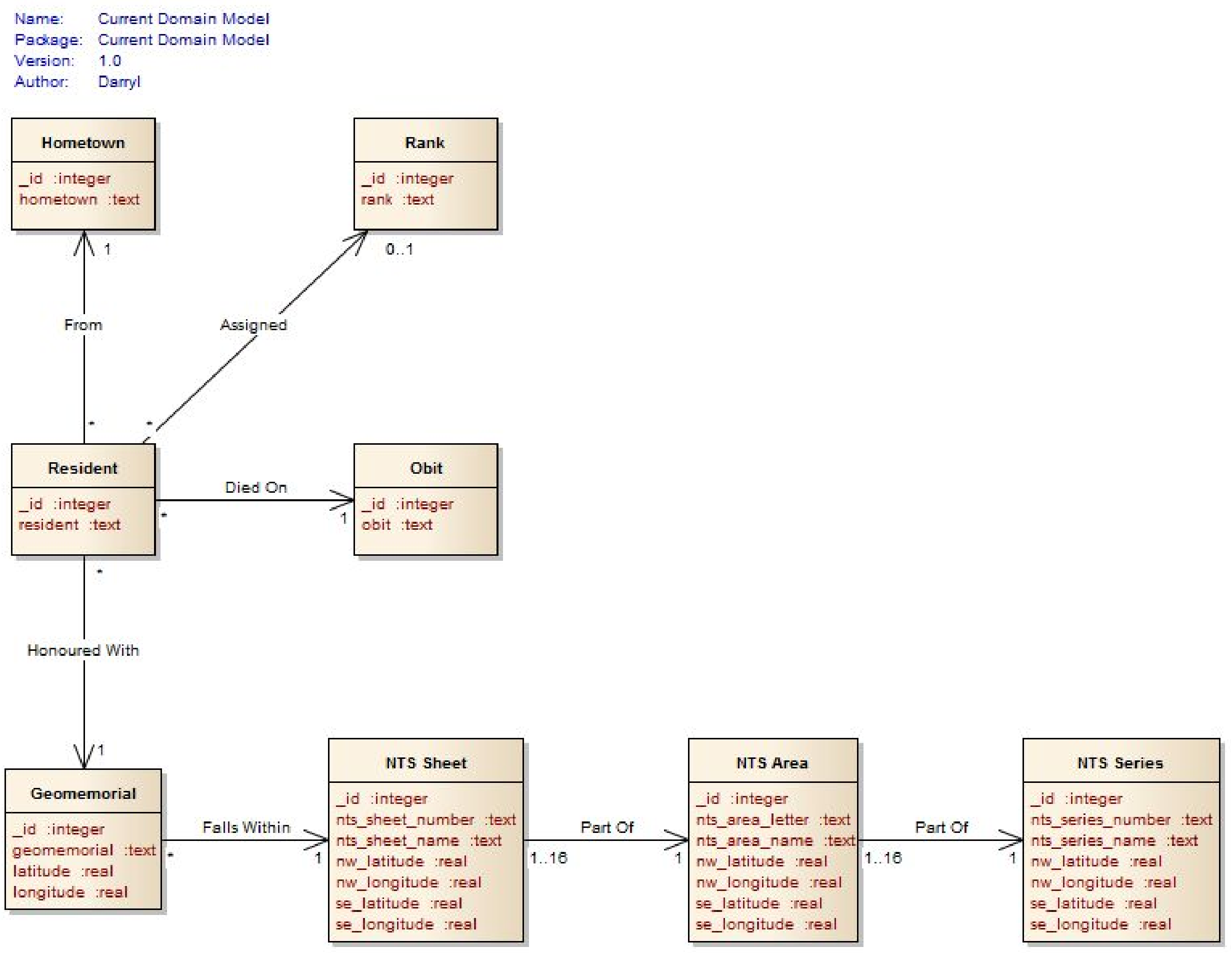
Key Considerations

How will your app handle data persistence?

Because the majority of geo-memorials are in the northern parts of Saskatchewan that cannot be accessed by road and are not covered by cell towers, I have chosen to include the geo-memorial data within the app (to allow for offline use) and use a Content Provider to search it.

Domain Model

The overall domain model for the application is as follows:



For every resident ("resident") that the Province has honoured with a geo-memorial ("geomemorial"), information on their hometown ("hometown"), rank ("rank"), and obit date ("obit") will be stored by the application. Information on the resident's geo-memorial ("geomemorial") will also be kept. This information can be searched.

Since hikers, canoers, pilots, and those who live in the north use National Topographic System (NTS) maps to travel from Point A to Point B, I associate each geo-memorial ("geomemorial") with the NTS maps they can be found on. There are three such maps:

- NTS Series maps ("nts_series") are drawn at a scale of 1:1,000,000 and cover 4 degrees of latitude and 8 degrees of longitude. They are referred to by name (ex: NTS 62 Series) or by number (ex: 62), and are carved up into 16 NTS Area maps.

- NTS Area maps ("nts_area") are drawn at a scale of 1:250,000 and cover 1 degree of latitude and 2 degrees of longitude. They are referred to by name (ex: Weyburn) or by a letter appended to the number of the NTS Series that the area is part of (ex: 62E). NTS

Area maps are carved up into 16 NTS Sheet maps.

- NTS Sheet maps (“nts_sheet”) are drawn at a scale of 1:50,000 and cover ¼ degree of latitude and ½ degree of longitude. They are referred to by name (ex: Oxbow) or by a number appended to the number / letter combination of the NTS Area that the area is a part of (ex: 62E1). NTS Sheets are the maps that people use on a daily basis and every geo-memorial can be located on an NTS sheet.

These NTS objects serve two roles in the system:

- Travellers interested in planning a trip can search for geo-memorials on their route by entering the names or numbers of the NTS maps they are working with. For example, a canoer travelling through the area covered by NTS Sheet “74E13” will cause the application to display all geo-memorials that fall within the NTS Sheet’s coordinates.

- The application can use the NTS maps to display counts of the geo-memorials that reside within the visible display area of the Google Map. For instance, if the entire Province of Saskatchewan is displayed within Google Map’s display area, counts of the geo-memorials within each of the NTS Series can be displayed on the map. Then, as the user successively zooms in to view smaller and smaller areas of the province, NTS Area counts can be shown, and then NTS Sheet counts, until finally the viewable area is small enough that markers for all of the geo-memorials contained within it can be shown to the user.

Data Model

SQLite database tables will be created for each of the domain objects shown above. In addition, a virtual FTS3 table will be used to provide fast text-based searches of all data in the system, and a “search_suggestions” table will be used to provide real-time search suggestions when a user is typing in their search criteria. These search suggestions will be categorized by domain object (ex: resident, hometown, NTS Area, etc.)

Population of the SQLite Database

Since the script used to populate the database is over 11,000 lines long, I will not be embedding the script as text within the ***SQLiteOpenHelper*** implementation. Rather, I will be treating the script as a raw resource (i.e. as a file located in ‘/res/raw’) which I will then read from within the ***SQLiteOpenHelper*** implementation to populate the database.

Content Provider

The application will have a single Content Provider extending ***SearchRecentSuggestionsProvider*** that will be used to perform the following queries:

- Text Searches (i.e. the free-form searches selecting off the FTS3 virtual table)
- Search Suggestions
- Resident Detail Searches
- Geo-memorial Detail Searches
- NTS Series Counts
- NTS Area Counts
- NTS Sheet Counts

Inserts, updates, and deletions will not be supported.

User Messages to the Developer

I will provide contact information in the form of an email address that users can send suggestions to.

Favorites and User Preferences

I hope to implement Google Sign-In for this application but make its use optional. If a user chooses not to log onto the application, I will use Android's Shared Preference capabilities to store and retrieve Favorites and User Preferences. But if a user does log onto the application, I hope to have Google store these values in the Cloud. I have to do some more reading on Google Sign-In to learn what this all entails.

Describe any corner cases in the UX.

User search returns too many geo-memorials to display on the map

I have found in testing that Google Maps can only display around 500 or so markers before the application crashes, and even displays of a couple hundred can overwhelm the user. So the number of markers displayed to the user has to be limited. What I will do is limit the maximum number of markers to 100 and use a toast or snackbar to inform users when their searches return too many results.

Google Maps is displaying an area with over 100 geo-memorials

There are times when a user might just want to interact with Google Maps instead of using the search feature. If the visible area contains many hundreds or thousands of geo-memorials, the application will crash if it tries to show them all. What I will do is display counts of the number of geo-memorials found in the NTS Series that fall within the map's display area. Then, as the users zooms closer and closer to the ground, I will replace NTS Series counts with NTS Area counts, and then NTS Area counts with NTS Sheet counts. Then, when we get to a zoom-level where no more than 100 geo-memorials are within the viewable area, I will replace the NTS Sheet counts with geo-memorial markers. The reverse procedure will also apply.

Offline Use

Since most of northern Saskatchewan is not covered by cell towers, people wanting to use the application in these areas will have to use it offline. For this to work, users will have to use Google Map's offline feature to download the areas of the map that they'd like to work with offline while still in an area with coverage. In order to make this easier, I will inform users of the need to do this in my introductory screens.

Markers are continually added to the map until it crashes

I have found in testing that as a user successively moves from one place to another on the map, the number of markers affixed to the map-- inside and outside the visible area -- can accumulate until the application crashes. In order to address this, I will design the application to remove all markers from the map that come to fall outside the display area as a user pans around, and add in their place the other markers that come to fall within the display area.

Heavy text display on map icons can slow down map refreshes

I found in testing that heavy use of custom text balloons in a Google Map control can seriously impact performance -- map refreshes tend to slow down and and it can take upwards of one to two seconds to display the balloons. I have chosen to deal with this issue in several ways. First, I will only display text balloons under very special conditions -- for example, when a user clicks on a marker, the balloon for that marker will be displayed. Second, I will use the Google Maps Android API utility library (http://googlemaps.github.io/android-maps-utils/) to generate images for these balloons (using its Icon Generator feature), which I will then cache and re-use in the background.

The Map zooms to close to the surface when moving to a single marker

I found in testing that the map zooms too close to the surface when it is programmed to move from one location to another. So, a user wanting to see the location of a lake isn't able to view the location of the lake with respect to other lakes in the area. Instead, the user is shown a detailed surface of the lake from which they need to zoom out in order to see the surrounding area. To deal with this, instead of zooming in as far as I can go onto a single coordinate, I will zoom down to a bounded area covering ¼ degrees latitude and ½ degrees longitude (i.e. the area covered by an NTS sheet) that has the point of interest in the middle. Then, if the person wants to zoom in for a closer view they can do so.

Google Maps is too damned accurate!

It turns out that the geographic coordinates for each of the Saskatchewan geo-memorials are only accurate to three decimal places (for example, Loranger Bay is (57.967, -103.367) because most of the coordinates were taken at the end of 1945 when measurements didn't need to be so precise. But Google Maps is so precise (upwards of 8+ decimals in precision) that the positioning of Markers in my Map control suffer from two weaknesses. First, if a user drills down to get a pretty close view of a geo-memorial, they might see that its marker is ½ a centimetre or so off from the associated landmark. Second, if a large number of geo-memorials fall within the same small geographic area, the markers tend to line up in rows and columns when displayed on the map -- something that looks a bit weird. I am not really sure how to deal with this, and I don't think it's a deal-breaker, but I am aware of the issue and I will make sure to tell users about it.

Describe any libraries you'll be using and share your reasoning for including them.

I will be using the following libraries to develop this app:

Group Id	***Artifact Id***	***Reason***
com.android.support	support-annotations	To use its Nullness annotations in order to minimize NPEs To use its Typedef annotations to implement enumerations
com.android.support	support-v4	To make use of its Fragment compatibility features
com.android.support	appcompat-v7	To make use of its ActionBar compatibility features where needed
com.android.support	design	To make use of its Material

		Design features such as FABs, Navigation Drawers, and so on
com.android.support	recyclerview-v7	To use its RecyclerView control when displaying search results
com.google.android.gms	play-services	To make use of the Google Play service APIs
com.google.android.gms	play-services-auth	To make use of Google's Sign-In feature
com.google.android.gms	play-services-maps	To make use of Google Maps
com.google.maps.android	android.maps.utils	To use its Icon Generator feature in order to improve map performance when displaying lots of info balloons in the map display area

Next Steps: Required Tasks

Task 1: Create a Google Developer Account

Create a Google Developer Account

1. Go to http://console.developers.google.com/ and create an account
2. Open a Billing Account.

Create a Project Under the Account

1. Create a Project called 'Saskatchewan Geo-Memorial Project'
2. Associate Project with Billing Account
3. Create an Android API Key for the project

Enable Google Services to the Account

1. Enable the Google Maps Android API for the project
2. Enable the Google+ API for the project

Secure the Services

1. Generate a development certificate
2. Associate its SHA1 key with the project's Android API key
3. Obtain a production certificate
4. Associate its SHA1 key with the project's Android API key

Task 2: Project Setup

Repository Setup

1. Create Github repository named 'Capstone Project'
2. Install git on development laptop
3. Clone Github repository onto development laptop
4. Create an appropriate .gitignore file if needed

Android Studio Setup

1. Create 'geomemorial' project in git workspace
2. Add the following libraries to the 'geomemorial' project:

    a. com.android.support:support-annotations
    b. com.android.support:support-v4
    c. com.android.support:appcompat-v7
    d. com.android.support:design
    e. com.android.support:recyclerview-v7
    f. com.google.android.gms:play-services
    g. com.google.android.gms:play-services-auth
    h. com.google.android.gms:play-services-maps
    i. Com.google.maps.android:android.maps.utils

3. Add the following map-related entries to the 'geomemorial' manifest:

    a. Permissions

        i. android.permission.ACCESS_FINE_LOCATION
        ii. android.permission.ACCESS_NETWORK_STATE
        iii. android.permission.INTERNET
        iv. android.permission.WRITE_EXTERNAL_STORAGE

    b. Features

            i. glEsVersion = 0x00020000

      c. Metadata (To Google Map Activity)

            i. Com.google.android.gms.version
            ii. com.google.android.geo.API_KEY

4. Commit the project to local git repository
5. Push the project to Github

Task 3: Content Setup

SQL Script Creation

1. Create DDL Set-Up Script
    a. Create Script in '/res/raw'
    b. Add table creation DDL
        i. For RANK table
        ii. For OBIT table
        iii. For HOMETOWN table
        iv. For GEOMEMORIAL table
        v. For RESIDENT table
        vi. For NTS_SERIES table
        vii. For NTS_AREA table
        viii. For NTS_SHEET table
        ix. For SEARCH_SUGGESTIONS table
        x. For MARKER_INFO virtual table (FTS3)
    c. Add view creation DDL
        i. For PROVINCIAL_COUNTS view
        ii. For NTS_SERIES_COUNTS view
        iii. For NTS_AREA_COUNTS view
        iv. For NTS_SHEET_COUNTS view
2. Create DDL Tear-Down Script
    a. Add view drop statements
        i. For NTS_SHEET_COUNTS view
        ii. For NTS_AREA_COUNTS view

        iii. For NTS_SERIES_COUNTS view
        iv. For PROVINCIAL_COUNTS view

    b. Add table drop statements

        i. For MARKER_INFO virtual table
        ii. For SEARCH_SUGGESTIONS table
        iii. For NTS_SHEET table
        iv. For NTS_AREA table
        v. For NTS_SERIES table
        vi. For RESIDENT table
        vii. For GEOMEMORIAL table
        viii. For HOMETOWN table
        ix. For OBIT table
        x. For RANK table

3. Create DML Data Population Script

    a. Add Table DML statements

        i. For RANK table
        ii. For OBIT table
        iii. For HOMETOWN table
        iv. For GEOMEMORIAL table
        v. For RESIDENT table
        vi. For NTS_SERIES table
        vii. For NTS_AREA table
        viii. For NTS_SHEET table

    b. Add Virtual Table (FTS3) data population statements
    c. Add SEARCH_SUGGESTIONS data population statements

Database Contract Creation

1. Define Common Database Settings

    a. Content authority
    b. Base content uri
    c. Database name
    d. Database version

2. Define Classes

a. For each table

   i. Rank
   ii. Obit
   iii. Hometown
   iv. Geomemorial
   v. Resident
   vi. NTS Series
   vii. NTS Area
   viii. NTS Sheet
   ix. Search Suggestions
   x. Marker Info

b. For each view

   i. Provincial Counts
   ii. NTS Series Counts
   iii. NTS Area Counts
   iv. NTS Sheet Counts

3. Add the following to each of these classes:

   a. Name
   b. Content Uri
   c. Content Type
   d. Content Item Type
   e. Column Names
   f. Constraints
   g. Default Projection
   h. Default Projection Order

Content Helper Setup

1. Create SQLiteOpenHelper implementation
2. Override onCreate method and have it use scripts to create and populate the database.
3. Override onUpgrade method and have it use scripts to drop database objects

Content Provider Setup

1. Create SearchRecentSuggestionsProvider implementation

2. Create static MatchCodes class containing integer constants for each of the query types supported by the Content Provider
3. Create static MatchPaths class defining match paths for each of the supported query types
4. Define a UriMatcher implementation that will be used to match URIs supported by this content provider
5. Define a Return Types array that will associate match codes with the types of content items to be returned for them
6. Add a custom constructor that calls the setupSuggestions method of the underlying base class
7. Override the onCreate method
8. Override the getType method
9. Override the query method

Task 4: Create Common Files

Create Styles (/res/values/styles.xml)

Contains the default styles of the application.

Create Styles (/res/values-large/styles.xml)

Contains style override specific to tablets. For instance, start and end margin widths would be defined here.

Create Styles (/res/values-land/styles.xml)

Contains style overrides specific to phones or tablets being held in landscape mode. For instance start and end margin widths would be defined here.

Task 5: Create Main Activity

Create Google Maps Api File (/res/string/google_maps_api.xml)

Contains the API key needed to use Google Maps in production.

Create Navbar Layout (/res/layout/fragment_navbar.xml)

This declares the navigation bar layout.

Create Navbar Fragment (/java/NavbarFragment.java)

Responsible for managing user interactions with navbar components.

- Attaches an **OnItemClickListener** implementation to menu control defined in the Navbar layout that is responsible for doing the following:
  - Integrating the Google Sign-In API into the application (NB: I have to research this a bit more to find out how it is done.)
  - Creating an explicit intent to display the Favorites Activity when the Favorites menu item is clicked.
  - Creating an explicit intent to display the Preferences Activity when the Settings menu item is clicked.
  - Creating an explicit intent to display the About Activity when the Help & Feedback menu item is clicked.
- Managing the saving and restoring of Navbar fragment state

Create Customer Map Marker Drawable (/res/drawable/marker_map.xml)

Defines a custom shape that will be used to display a custom map marker.

Create Map Layout (/res/layout/fragment_map.xml)

Responsible for rendering the Google Map control.

- Declares a <fragment /> definition that subclasses **com.google.android.gms.maps.SupportMapFragment**.

Create Map Fragment (/java/MapFragment.java)

Responsible for managing map interactions and the map lifecycles.

- Implements **GoogleMap.OnMapLoadedCallback** so that a reference to the initialized map can be stored for further use.
- Implements **OnMapReadyCallback so that** so that the initial view of the map to an area that bounds the Province of Saskatchewan.
- Implements **GoogleMap.OnMarkerClickListener** so that information windows will be displayed when a user clicks on a marker
- Implements **GoogleMap.OnCameraChangeListener** to facilitate animations and, perhaps, to facilitate marker caching

- Contains a static class that implements **GoogleMap.InfoWinderAdapter**. This class generates a custom information window when a marker is clicked.

- Responsible for saving and restoring the fragment's state.

Create Search List Item (/res/layout/list_item_search.xml)

Responsible for displaying information about a single geo-memorial list item shown in the search layout. Contains the following fields:

- Resident Name
- Hometown
- Rank
- Obit. (i.e. Date of Death)
- Geo-Memorial Name and Location
- Name of NTS Sheet on which the geo-memorial can be found

Create Search Layout (/res/layout/fragment_search.xml)

Defines the scrollable list control that will be used to display search results.

Create Search Fragment (/java/SearchFragment.java)

Responsible for managing search result interactions and to managing the fragment's lifecycles.

- Implements **AdapterView.OnItemClickListener** so that clicks made to a search result cause the map to zoom to its geo-memorial location.

- Contains an inner class extending **CursorAdapter** that is responsible for binding search results to the scrollable list view declared in the fragment's layout.

- Contains an inner static **ViewHolder** class that streamlining access to a search item's UI controls.

- Responsible for saving and restoring the fragment's state.

Create Search Configuration (/res/xml/searchable.xml)

Responsible for defining how the search provider will behave.

Create Main Layout (/res/layout/activity_main.xml)

This is the default layout of the Main Activity. It uses a frame layout to anchor the search fragment on top of the bottom half of the map fragment.

Create Main Layout (/res/layout-large-land/)

This is the layout of the Main Activity as seen on tablets held in landscape mode. It replaces the frame layout with a horizontal linear layout so that search and map fragments appear side by side.

Create Main Menu (/res/menu/menu_main.xml)

This declares the menu items that appear on the Main Activity.

- It declares map type entries that users can click to select a map type
- It contains an entry that users can click to clear their search history.

Create Main Activity (/java/MainActivity.java)

Responsible for coordinating interactions between search and map fragments. Also responsible for handling search queries as well as menu selections.

- Extends **AppCompatActivity** to support the navbar and other material design features.
- Overrides **onNewIntent** so that calls made to it by the Search Provider cause a search to be performed or results to be displayed.
- Overrides **onOptionsItemSelected** so that selections in the Options Menu will be handled.
- Delegates search requests to the Content Provider.
- Delegates search results to both the Search fragment (for textual display of search results) and the Map fragment (for geographic display of search results).
- Responsible for saving and restoring the activity's state.
- Responsible for displaying the App Bar.

Task 6: Create Favorites Activity

Create a Favorites List Item (/res/layout/list_item_favorites.xml)

Declares the default layout of a geo-memorial favorite as consisting of a map control with Share and Remove buttons underneath it.

Create a Favorites List Item (/res/layout-normal-land/list_item_favorites.xml)

Declares an alternative Favorites List Item layout that displays the Share and Report buttons at the start or end of the map on phones held in landscape mode.

Create a Favorites List Item (/res/layout-large/list_item_favorites.xml)

Declares an alternative Favorites List Item layout that displays the Share and Report buttons at the start or end of the map on tablets.

Create Favorites Layout (/res/layout/fragment_favorites.xml)

Declares a Favorites fragment as consisting of a vertical linear layout containing a scrollable list of favorites list items.

Create Favorites Layout (/res/layout-large-land/fragment_favorites.xml)

Displays an alternative Favorites layout that uses a two-column grid layout to display list items instead of a vertical linear layout.

Create Favorites Fragment (/java/FavoritesFragment.java)

Responsible for managing the lifecycle and user interactions with the Favorites layout.

- Defines a **CursorAdapter** responsible for binding search favorites to the fragment's linear layout.
- Defines a **ViewHolder** responsible for optimizing the rendering of favorites list items.
- Attaches an **OnItemClickListener** implementation to each of the Share buttons that create a Shareable Action when button has been clicked.
- Attaches an **OnItemClickListener** implementation to each of the Remove buttons that removes the geo-memorial from the user's list of favorites.
- Responsible for saving and restoring fragment state

Create Favorites Activity (/java/FavoritesActivity.java)

Responsible for loading the Favorites Fragment and for saving and restoring activity state.

Task 7: Create Preferences Activity

Create Preferences Layout (/src/layout/fragment_preferences.xml)

Declares how the Preferences Fragment will be displayed on screen. Simply consists of radio buttons at the moment.

Create Preferences Fragment (/java/PreferencesFragment.java)

Responsible for managing the lifecycle and user interactions with the Preferences layout.

- Retrieves and stores preferences from Shared Preferences in the case of unauthenticated users.
- Retrieves and stores preferences from the user's Google Sign-In profile in the case of authenticated users (NB: I still need to learn the details of this.)
- Responsible for saving and restoring fragment statement.

Create Preferences Activity (/java/PreferencesActivity.java)

Responsible for loading the PreferencesFragment and for saving and restoring activity state.

Task 8: Create About Activity

Create About Layout (/src/layout/fragment_about.xml)

Declares how the About fragment will be displayed on screen. Simply consists of a number of cards for display build and version information, contact details, acknowledgments, and links to further information on the Saskatchewan Geo-Memorial project.

Create About Layout (/src/layout-large-land/fragment_about.xml)

An alternative layout that uses a two-column grid layout to display the information cards instead of a vertical linear layout.

Create About Fragment (/java/AboutFragment.java)

Responsible for managing the lifecycle and user interactions with the About layout.

- Responsible for saving and restoring fragment statement.

Create About Activity (/java/AboutActivity.java)

Responsible for loading the AboutFragment and for saving and restoring activity state.

Task 9: Other Required Tasks

9a. Localization and Internationalization

1. Move all strings into strings.xml

2. Provide sufficient context for declared strings
3. Mark message parts that should not be translated
4. Support RTL layouts and text
5. Use system-provided formats for dates, times, numbers, and currency
6. Include a full set of default resources

9b. Accessibility

1. Describe user interface controls
2. Enable focus-based navigation
3. Add secondary feedback-mechanism in addition to audio

9c. Material Design

1. Ensure that surface composition is appropriate
2. Ensure that shadows and elevations are appropriate
3. Use a primary colour and an accent colour
4. Ensure that the status bar is coloured appropriately
5. Ensure that margins and spacing are appropriate
6. Ensure that font sizes and font families are appropriate
7. Ensure that feedback motions and effects are appropriate
8. Ensure that transitions and animations are appropriate
9. Use a FAB button when appropriate
9. Ensure that the navbar is designed appropriately

9d. Stability and Reliability

1. Validate input where needed (ex: # of results displayed on map)
2. Handle database exceptions and timeouts
3. Handle network exceptions and timeouts

Task 10: Test, Debug, Repeat