

Kevin Heath High

kevin@hobbitsoft.net

GitHub Username: kevinheathhigh

Sailing Buddy

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Splash Screen](#)

[Main Screen](#)

[Portrait](#)

[Landscape](#)

[Overview Section](#)

[Tabbed Section](#)

[Local Stations List](#)

[Favorite Stations List](#)

[Search Screen](#)

[Station Detail Screen](#)

[Map Screen](#)

[Forecast Screen](#)

[Settings Screen](#)

[Widget Screen](#)

[Key Considerations](#)

[Permissions](#)

[Data Access](#)

[Sorting Favorites](#)

[External Libraries](#)

[Free vs Paid Version](#)

[Free Version](#)

[Paid Version](#)

[Analytics](#)

[Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Title bar Icons](#)

[Main Screen](#)

[Station Detail Screen](#)

[Forecast Screen](#)

[Lists on Main and Search screens](#)

[Task 4: Online Data and Database](#)

[NOAA Website](#)

[Buoy Station List](#)

[Database Table](#)

[Buoy Observations](#)

[Forecast](#)

[Tides and Currents API](#)

[Online/Offline Modes](#)

[Online](#)

[Offline Pre initialization](#)

[Offline Post initialization](#)

[Task 5: Location Routines](#)

[Task 6: Settings](#)

[Task 7: Favorite Widget](#)

[Task 8: Testing](#)

[Task 9: Free Tier Advertising](#)

Description

Plan your sail by getting up date NOAA data from both your local bouy and local weather stations.

Intended User

Perfect for all boating enthusiasts.

Features

- Display NOAA Buoy data from nearest or selected station
- Display Tides and Currents data from station nearest selected buoy station
- Display Weather from station nearest selected buoy station
- Save buoy station IDs as favorites for quick access

- Widget to display data from favorite station

User Interface Mocks

Only the Main Screen has a differing landscape screen. Use Room's ViewModel to maintain view on activities when screen rotates or becomes out of focus.

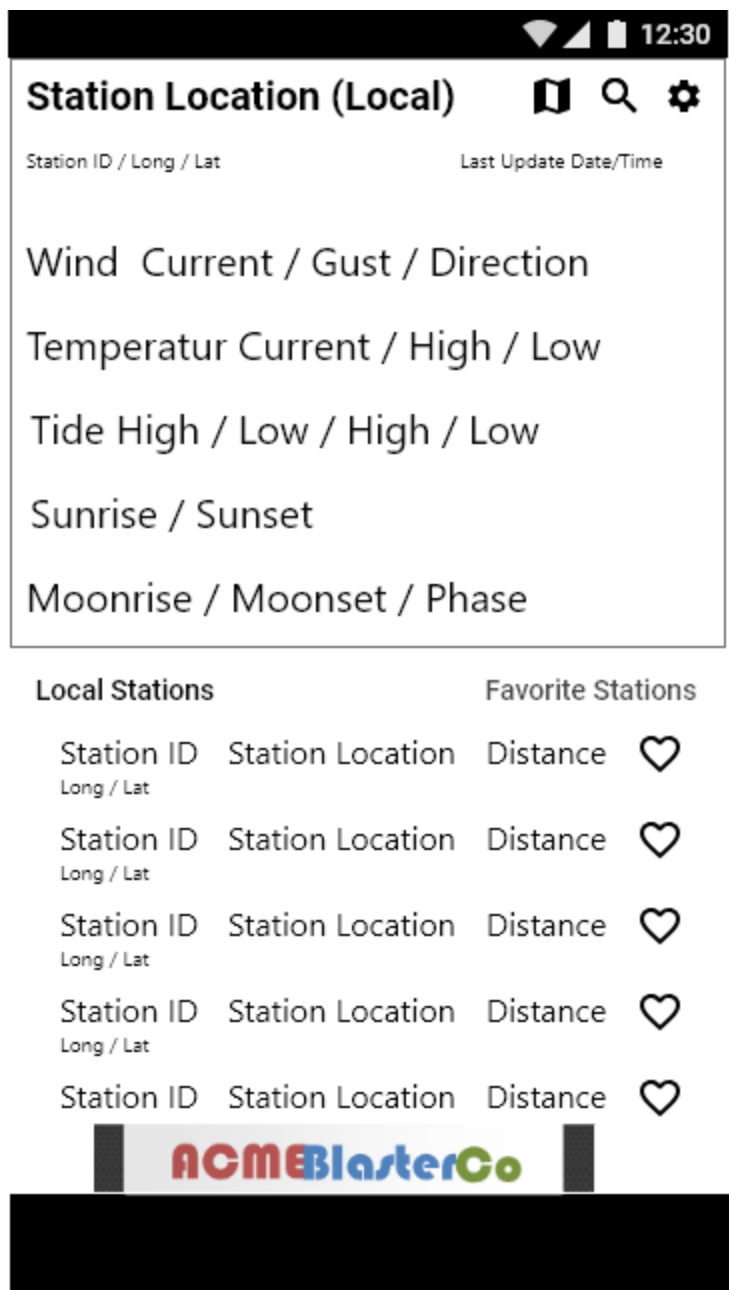
Splash Screen



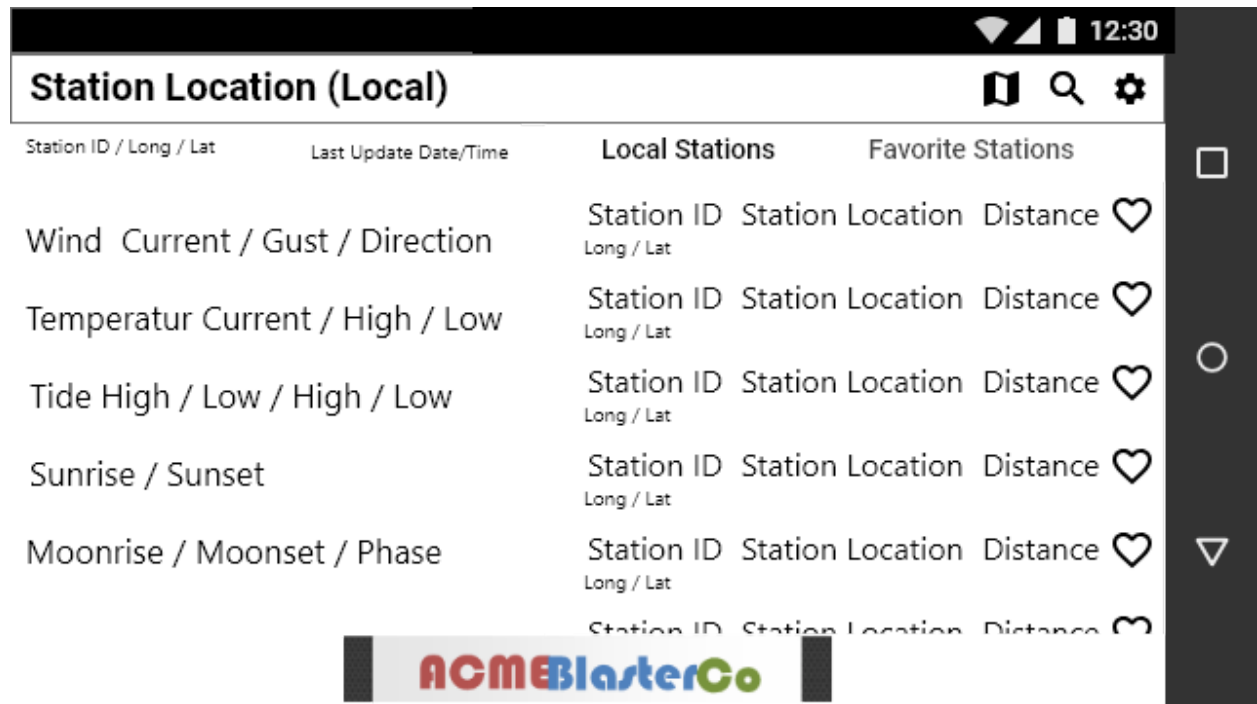
Initial Display when programs launches and sets up, retrieves data from NOAA, populates database, etc.

Main Screen

Portrait



Landscape



Overview Section

First view of data, initial screen shows the buoy/weather data from stations closest to current position

This will change when a station is selected from one of the list (Local/Favorites)

Tabbed Section

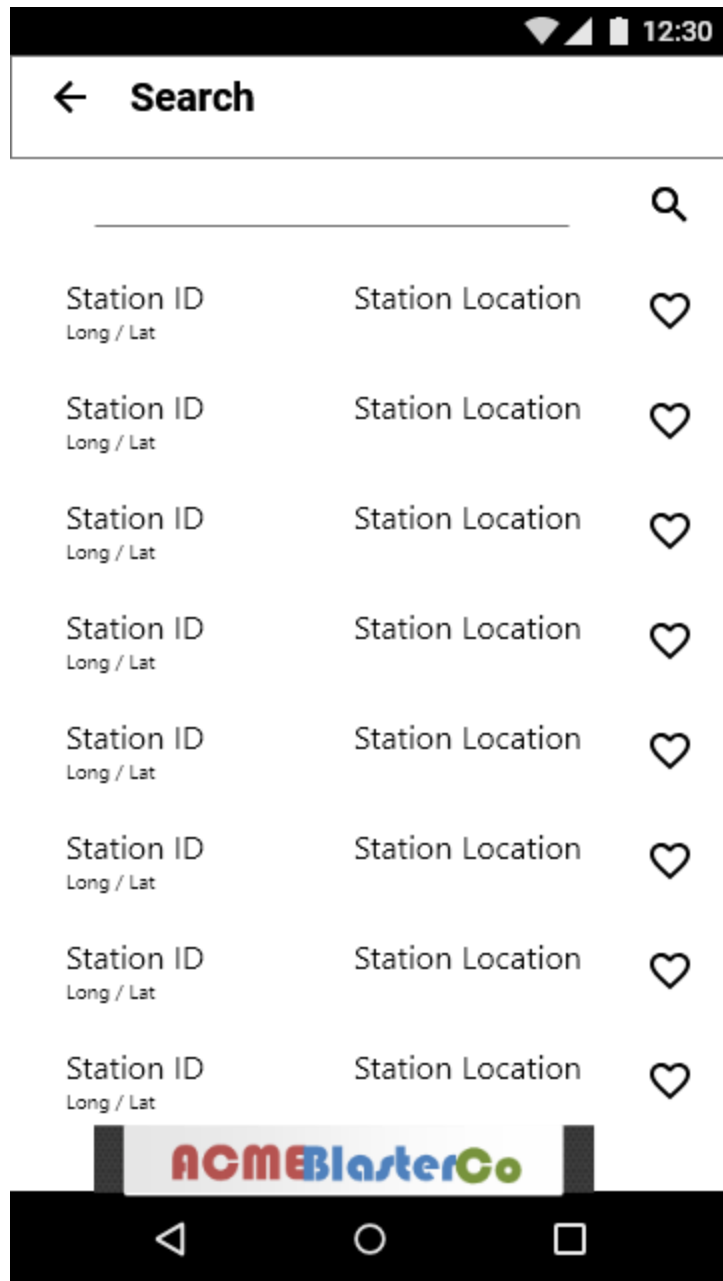
Local Stations List

This tab shows a scroll list of 10 closest stations (the first station on the list being the closest station current being view in the overview when app starts)

Favorite Stations List

This tab shows stations that have been marked as favorites

Search Screen



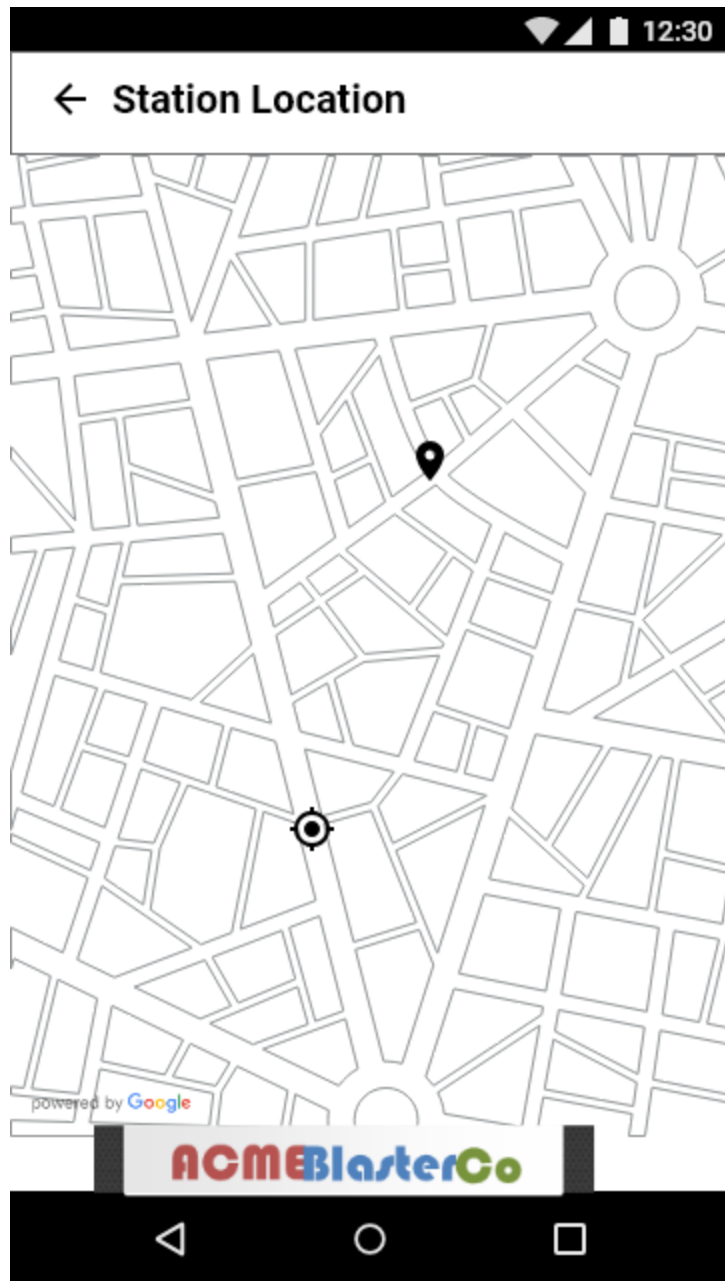
Allows for the searching of station by either station id or station name

Station Detail Screen



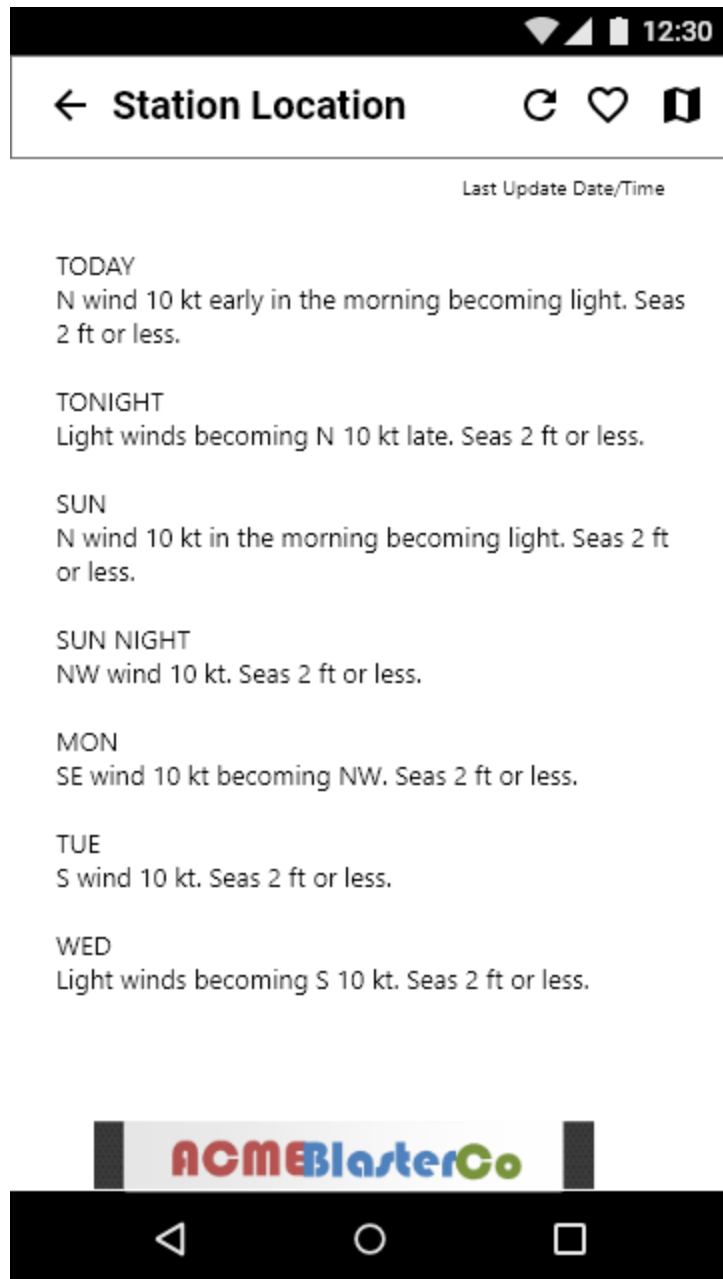
Show detail information with regards to select Station Buoy, this is dynamic given that not all buoys display the same information.

Map Screen



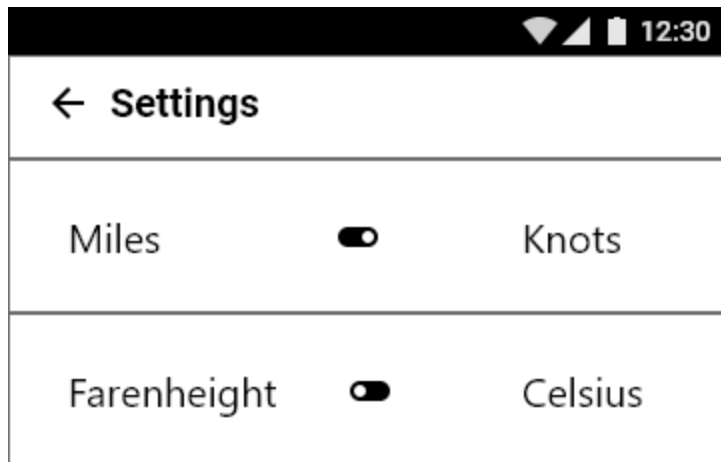
Map screen shows location of selected Station on a map relative to current position.

Forecast Screen



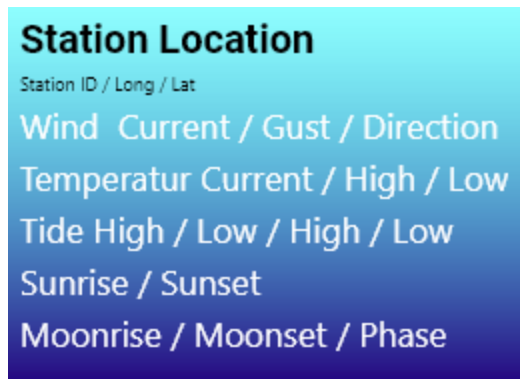
Forecast screen shows current reported forecast

Settings Screen



Settings screen shows user configurable settings, perhaps make this an overlay, instead of a separate activity. Though will need space if displaying licenses is required.

Widget Screen



Widget displays overview data of selected station from Favorites list. Clicking on Widget will launch detail screen for appropriate station. Widget should refresh once an hour

Key Considerations

Permissions

Internet Access

Location

Data Access

Initial list of Stations will be retrieved on startup and populated to a local SQLite database. This should happen in the background while the splash screen is being displayed. This data should not be required to be every time since it does not change that often. Will need to resolve how to check for data updates to the main staton table. All downloaded cached data needs to be time-stamped and displayed so user knows

Each time the app is run it should check current location and resolve the closest Stations, download and cache appropriate data for local Stations. This will allow for quick display of current data. This should also happen during splash screen.

Caching Data

Add a Table to the database to cache accessed data so that the app does not have to go back out to the internet every time a station is selected. Since the data does not change often (~ every hour). New data should only be checked if the current data is older than an hour. Both a refresh button as well as the ability to 'pull down' on the view to force a refresh. Automatic refresh will not be enable.

When data is refreshed, update the data in the background and use Live Data to actually refresh the displayed data.

Activity Data

Though their appears to be a lot of data being displayed, ultimately it is small and text based and easily passable via intents.

Sorting Favorites

Maintain a tally of how often a favorite is accessed and sort the favorites list based on how often used, i.e. when the user actually selects the favorite from the favorites list.

Internal Libraries

Database / Persistent Storage

Room Database Service

<https://developer.android.com/topic/libraries/architecture/room>

GPS Location Services

<https://developer.android.com/reference/android/location/Location>

JSON Parsing

<https://github.com/google/gson>

External Libraries

Use Butterknife for binding view objects.

<http://jakewharton.github.io/butterknife/>

Free vs Paid Version

Free Version

Ad Display

<https://developers.google.com/ads/>

Display a banner ad at the bottom of every screen

Periodically display a full page (video?) ad when switching screens

Always display a full page (video?) ad before the forecast screen

Paid Version

No Ads anywhere

Analytics

Incorporate Google Firebase Analytics

Steps: Required Tasks

Task 1: Project Setup




Create a Master/Detail Android Application

Task 2: Implement UI for Each Activity and Fragment





- Splash Screen (displayed while downloading updated information)
- MainActivity
 - Overview Layout Fragment for displaying Current Location data
 - Location
 - Wind Speed/Direction
 - Temperature
 - Sunrise/Sunset Times
 - Moon Info Rise/Set/Phase
 - Pull down refresh of data
 - Local Station Lists (Based on location information)
 - Favorites
- Detail Screens
 - Buoy Data
 - Tides
 - Currents
 - Forecast
 - Sun/Moon Rise/Set Times and Phase
 - Other information provided by Bouy
 - Pull down refresh of data
- Search Screen
 - List of All Stations (Sorted Alpha)
 - RegEx to narrow list by Location or ID
- Forecast Screen
 - Pass coordinates
 - Retrieve forecast from nearest weather station
 - Pull down refresh of data
- Map Screen
 - Pass Station coordinates and local coordinates

Task 3: Title bar Icons




Main Screen

-  Shows Map Screen for displayed location
-  Shows Search Screen
-  Shows Settings Screen

Station Detail Screen

-  Refresh Station Details (from Internet)
-  Add or remove current Station from Favorites
-  Shows Forecast Screen local to displayed Station
-  Shows Map Screen for displayed Station

Forecast Screen

-  Refresh Forecast Screen (from Internet)
-  Add or remove current Station from Favorites
-  Shows Map Screen for displayed Station

Lists on Main and Search screens

-  Add or remove current Station from Favorites

Task 4: Online Data and Database

NOAA Website

https://www.ndbc.noaa.gov/docs/ndbc_web_data_guide.pdf

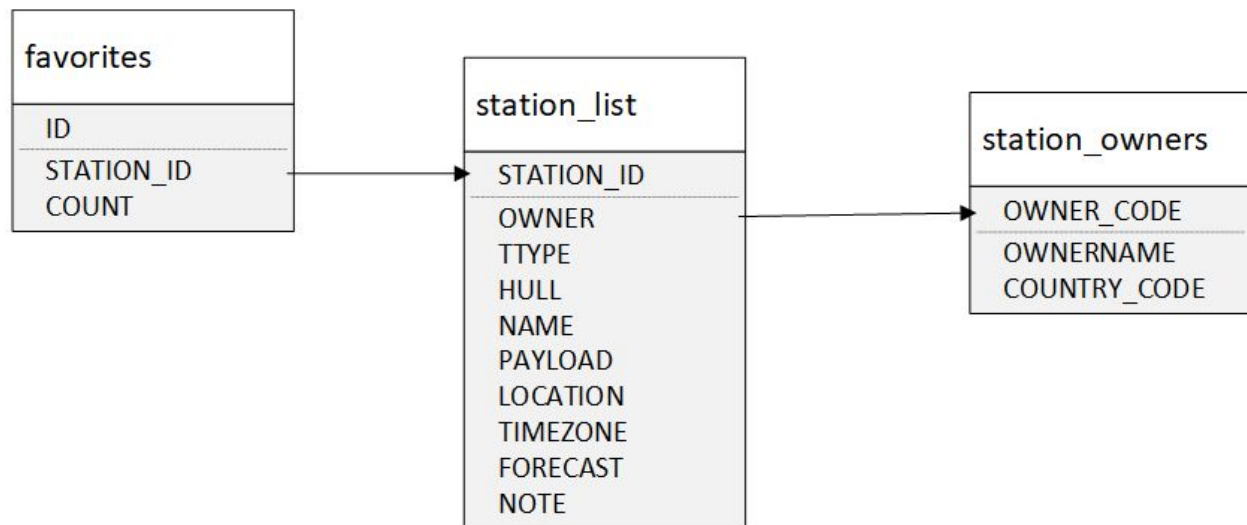
Buoy Station List

https://www.ndbc.noaa.gov/data/stations/station_table.txt

https://www.ndbc.noaa.gov/data/stations/station_owners.txt

Database Table

- station_list
- station_owners
- favorites



Caching Database Tables

- station_data
- forecast_data

Buoy Observations

https://www.ndbc.noaa.gov/data/latest_obs/<station id>.txt

Forecast

https://www.ndbc.noaa.gov/data/DAB_Forecasts/<station id>fc.html

Tides and Currents API

<https://tidesandcurrents.noaa.gov/api/>

Online/Offline Modes

Online

- Initialize and populate database the first time
- Update any data
- Update buoy information for non-favorite buoys
- Cache current downloaded data for availability if offline next instance

Offline Pre initialization

- Display Error about needing to be online the first time
- Monitor Network connection, initialize when connected

Offline Post initialization

- If Station data has been cached from previous online activity, display cached data
- If Station data has not been cached, show “need to be online” message

Task 5: Location Routines

- Routines to resolve location distances from current location
- Routines to resolve Station to Station distances.
-

Task 6: Settings

Two user defined settings are currently identified.

- Miles vs. Knots (default)
- Fahrenheit (default) vs. Celsius

Allow the user to choose the units the information is displayed.

Conversion routines

Defaults represent the data as retrieved from NOAA

Task 7: Favorite Widget

- When adding Widget, need to have a dialog to choose which favorite to display
- Use Overview Layout
- Display a refresh button

Task 8: Testing

- UI Testing
 - Each Screen is displayed when respective action is taken
 - Actions within each screen is responsive
 - Refresh actions go out and retrieves latest data
- Network Testing
 - Test access to NOAA data
 - Ability to finding Stations List/ Station Data/ Forecast/ Tides/ Currents
- Database Testing
 - Test creation/existence of database
 - Test SELECT statements against database
- Location Services
 - Test Location Services
 -

Task 9: Free Tier Advertising

<https://developers.google.com/admob/android/banner>

- Sign up for AdMob

- Integrate and display on each page
- Resolve integration for displaying full page ads when Forecast Screen is selected

Submission Instructions

- After you've completed all the sections, download this document as a PDF [File → Download as PDF]
 - Make sure the PDF is named "**Capstone_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
- Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"