Mobile App Documentation

for

Salish Sea Water Weather Station

Created by R. Lingg, S. VanZwol, M. Grappo, and K. Stewart

*Advised by Dr. Erik Fretheim*

*Western Washington University*

Version 1.0

2019-10-23

# Table of Contents

[**Table of Contents**](#_mslvh2y1d8hf) **1**

[Revision History](#_6j27nldm1k7i) 2

[**1 Development Environment**](#_smfx75kue2mz) **2**

[1.1 Android Studio](#_wakc85cac3wu) 2

[1.2 SDK Support](#_37q2k7osll5c) 2

[2 Application Build](#_h4aaaexpfndp) **2**

[2.1 Used Repositories](#_hnu64ww17kiu) 2

[2.2 Gradle Layout](#_b0lke4bia1jg) 3

[2.3 Build Dependency Specifics](#_zecaehshgrod) 3

[**3 Manifests Directory**](#_3f86vtjkpbt) **3**

[3.1 Android Manifest](#_yd1x7frw0wi2) 3

[**4 Java Source Code Directory**](#_9uk12so7q65p) **3**

[4.1 Authentication](#_fgbfpuy9tup5) 3

[4.2 Repository](#_ijinua47k8e6) 4

[4.3 Buoy Connection](#_7xhx9tee2564) 4

[4.3.1 Bluetooth Low Energy Service](#_b5rrc6owc6zj) 4

[4.3.2 Device Control Activity](#_nwm1rhkwq1f) 4

[4.3.3 Device Scan Activity](#_4n18296qhllo) 5

[4.3.4 Encryption Unit](#_j0os2uol12l9) 5

[**5 Resources Directory**](#_3pq0kl2ss7rw) **5**

[5.1 Drawable](#_2wd3e459ehz2) 5

[5.2 Layout](#_2wd3e459ehz2) 5

[5.3 Menu](#_2wd3e459ehz2) 5

[5.4 Mipmap](#_2wd3e459ehz2) 6

[5.5 Values](#_2wd3e459ehz2) 6

## 

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Authors | Date | Description | Version |
| RL | 2019-10-23 | Document Created | 1.0 |
|  |  |  |  |
|  |  |  |  |

## 

# 1 Development Environment

## 1.1 Android Studio

Compiled in Android Studio 3.5.1, Build #AI-191.8026.42.35.5900203, built on September 25, 2019, JRE: 1.8.0\_202-release-1483-b49-5587405 x86\_64, JVM: OpenJDK 64-Bit Server VM by JetBrains.

## 1.2 SDK Support

This application is compiled with the android build settings set to compileSdkVersion 27, buildToolsVersion '28.0.3', minSdkVersion 19, and targetSdkVersion 27.

# 2 Application Build

## 2.1 Used Repositories

This application uses resources from the google, jcenter, and maven repositories. These repositories are pulled before the application starts building and are used for resources in later stages of construction.

## 2.2 Gradle Layout

The gradle-wrapper.properties file contains the resources for where to source the gradle tool for use. The build.gradle(weather-station-app) contains the settings for the build dependencies, required repositories, and specified code versions, as well as the virtual machine version to run the code with. The build.gradle (Module: Application) holds information about how to use the information from the build.gradle (weather-station-app) when the gradle is building the compiled version.

## 2.3 Build Dependency Specifics

This application is built with the 3.5.1 android gradle tool. This application is built with the 3.5.1 android gradle tool. The android support version for this application is v4:27.1.1 which is the most up to date version of the api 27 support and uses the version 28.0.0 annotations. The kotlin version used in the application is 1.3.50, the kotlin jvm version is 1.3.50, and the kotlin core routines version is 1.1.1.

# 3 Manifests Directory

## 3.1 Android Manifest

The AndroidManifest.xml lives in this directory and hold the summary and descriptive information about the application. Such things in this xml file include the application’s version code and name, required feature directories, required permissions from the user, and general application design information. In the application design area of this file, the layout, activities, icons, services, and orientations are specified.

# 4 Java Source Code Directory

## 4.1 Authentication

There are three main components in the authentication directory. Firstly, the current user file holds the information locally about the users key and their ID. Second, the login activity file holds the code that lets the user try to sign in, handle responses on sign in attempts, and processes the request to the server with the users login information and returns an accept or decline with an error message. The login activity also handles the flow of the application for creating the page, logging in, authenticating, and moving to the next page. Thirdly, the user registration fragment holds the control flow for the user trying to create a new account and the errors that could arise.

## 4.2 Repository

The Repository Class in the repository folder holds the main functionalities for directly accessing the web server database and using the associated api. The Repository objects are used in the access of the database via the creation of Repository objects. The main actions in the Repository class include talking to the server about user login, user registration, getting current stations and building the api for usage. The Repository Class uses the other classes and interfaces in the repository directory which. The classes used include a skeleton for the Data Visualization Activity, the Encryption Dialogue Activity, the Error Parser, and the Request Interceptor. The interfaces that are used in the implementation of the Repository class are a Data Source interface which enforces the user login, registration and buoy listing to be implemented for buoy data sources, as well as a Weather API interface to enforce user login, registration and buoy listing to be implemented with HTTP functionalities for the storage server.

## 4.3 Buoy Connection

### 4.3.1 Bluetooth Low Energy Service

The Bluetooth Low Energy Service file contains the code for creation of a Bluetooth manager, general search for devices, general establishment of a connection, and general disconnect as well as management of callbacks to the application.

### 4.3.2 Device Control Activity

The Device Control Activity holds the code for communication from the board to the app and from the app to the database as well as the code for the actions to start this process. The transfer of data is performed by 1) the reception of the whole symmetric key (the key-part and the initial vector) from the buoy, 2) creating the secret key pair, 3) receiving the encrypted data packets from the buoy, 4) displaying the encrypted data, 5) decrypting the data, 6) displaying the unencrypted data, 7) turning the data points into a json format, and 8) sending the json array to the web server. The data sending, notification, and service UUIDs for communication with the buoy are stored in this file.

### 4.3.3 Device Scan Activity

The Device Scan Activity holds the code and actions for scanning for potential Bluetooth Low Energy access points and displaying them to the screen. This includes the connection to established buoys as well as the registration of new buoys with the webserver. As such this area has many Bluetooth protocols as well as a direct reference to the web servers ip and port for connections.

### 4.3.4 Encryption Unit

The Encryption unit is an object that is instantiated and given to the device control activity for usage in the decryption of data from the buoy. As such it handles the creation of new key pairs and cyphers, the distribution of the created public and private keys, and the decryption of data from the buoy. It also holds an encryption functionality for testing purposes only that should be removed from the final build.

# 5 Resources Directory

## 5.1 Drawable

The drawables directory holds the xml files for how to draw shapes or images to the screen for usage by the user interfaces. The images used for backgrounds as well as pop ups are stored in this area.

## 5.2 Layout

The layout directory holds the xml files for the user interfaces for each page of the application. This includes the buttons and their routing, text entry areas, and popups with descriptions about the properties of each component in the pages.

## 5.3 Menu

The menu directory holds the files pertaining to the top navigation menu that holds the individual pages inside the content areas of this wrapper. This is where the buttons to scan Bluetooth devices and registration of device popdown menus are held.

## 5.4 Mipmap

The mipmap directory holds the icons that are used. In the current state of the project this is only holding the WWU icon in a variety of image sizes for flexible sized icon use.

## 5.5 Values

The values directory holds the xml templates for general values and styles used by the other directories in resources and throughout the app. These templates include the values for referenced styles, colors, and strings.