How to Use this Template

1. Create a new document, and copy and paste the text from this template into your new document [Select All → Copy → Paste into new document]
2. Name your document file: "**Capstone_Stage1**"
3. Replace the text in green

GitHub Username: nicke2668

RealWeather

Description

A weather app that allows the user to set a location and has a widget

Intended User

Everyone

Features

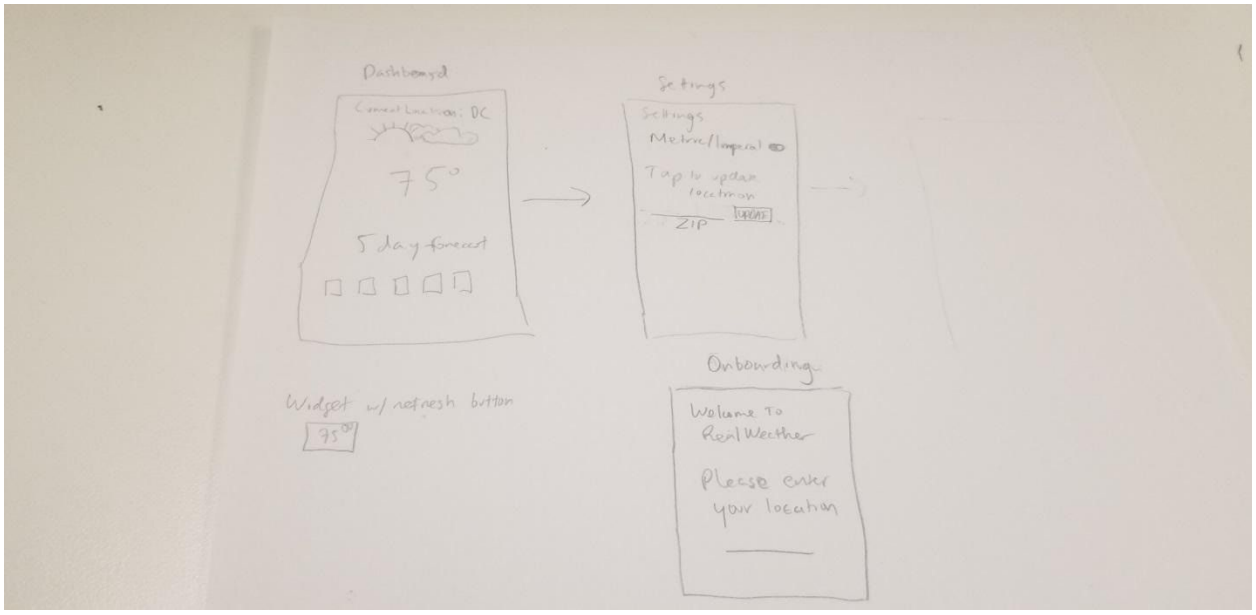
List the main features of your app. For example:

- Saves location
- Updates weather
- Allows location to be updated
- Has widget

User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1



Key Considerations

How will your app handle data persistence?

SharedPreferences - for temperature unit preferences
Room - for weather data

Describe any edge or corner cases in the UX.

Rotation – weather data will be persisted.

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

**App will be written solely in the Java Programming language.

Versions:

Android Studio: 3.5.3
Gradle: 5.6.2

Picasso: 2.5.2
Retrofit:2.6.0
Gson: 2.8.6
Room: 1.1.1

Picasso or to handle the loading and caching of images. Will have an imageset that will reflect various conditions of the weather.

Retrofit for API calls

Gson to deserialize network response

Describe how you will implement Google Play Services or other external services.

OpenWeatherAPI – to retrieve weather data via WorkManager
Location API - To retrieve user location if permission granted.
Admob - To display test ads

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

- Get OpenWeatherAPI key
- Configure libraries
- Implement UI designs
- Set up network calls
- Set up persistence model
- Set up WorkManager scheduled updates
- Observe network calls from UI using livedata

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for MainFragment
- Build UI for PreferenceFragment (give option for Metric/Imperial, etc)
- Build UI for widget

Task 3: Set up Free and Paid Flavors

- Create free flavor with ads
- Create paid flavor with no ads

Task 4: Set up MVVM architecture

- Set up ViewModel scoped to a navigation graph
- Use single activity architecture
- Set up repositories with access to retrofit instance

Task 5: Wire up data to UI

Communicate weather data results to UI via live data – ensure that weather data is updated. Ensure that widget is updated when refreshed.

Task 6: Handle accessibility

Ensure that all imageviews have content descriptions.

If the system does not pass focus to the appropriate view when navigating in a given direction, specify which view should receive focus with the following attributes:

android:nextFocusUp
android:nextFocusDown
android:nextFocusLeft
android:nextFocusRight

Task 7: Handle resources

Strings will be stored in strings.xml
Colors will be stored in colors.xml
Themes will be stored in styles.xml

Add as many tasks as you need to complete your app.

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**"