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

Description
Intended User
Features
User Interface Mocks
Screen 1
Screen 2
Key Considerations
How will your app handle data persistence?
Describe any corner cases in the UX.
Describe any libraries you’ll be using and share your reasoning for including them.
Describe how you will implement Google Play Services.
Next Steps: Required Tasks
Task 1: Project Setup
Task 2: Implement UI for Each Activity and Fragment
Task 3: Your Next Task
Task 4: Your Next Task
Task 5: Your Next Task

GitHub Username: andreeaelena

Hydratic - Hydration tracking app

Description

Keeping hydrated is a very important task that we should be doing daily, but sometimes we need an extra motivation to stay on track. Hydratic helps you set a hydration level that you want to achieve daily and log every water intake throughout the day. You can also set reminders to make sure you don’t forget to drink water if you’re having a busy day.
Hydratic also lets you see your progress and figure out what are the days that need more attention from your side.

Intended User

This app is essentially for everyone who wants to have a healthy lifestyle but their busy day-to-day life gets in the way and distracts them from their goal. Hydratic will act as their hydration coach, making sure they stay on track with their water intake.
The app will be available to users from all countries supported by the Google Play Store.

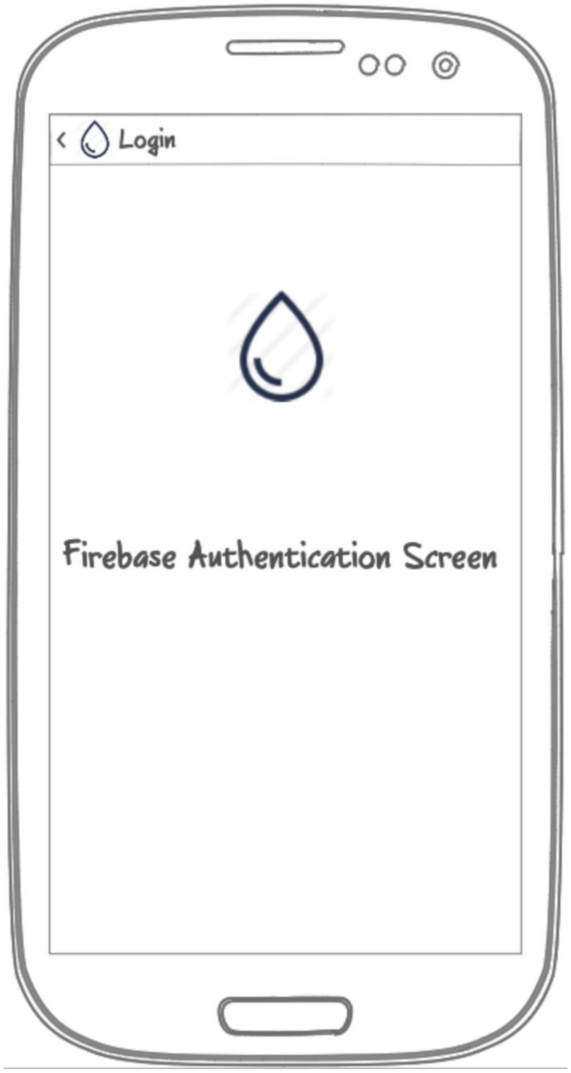
Features

- User authentication
- User onboarding - configures the app for the user
- Displays the current hydration level (daily value)
- Allows the user to input data (water intake as different containers: glass, cup, bottle, etc)
- Sets reminders and sends local notifications
- Displays historic data as a chart
- Displays useful information about why hydration is important
- Saves the user’s data to a database
- Displays basic info through a widget

User Interface Mocks

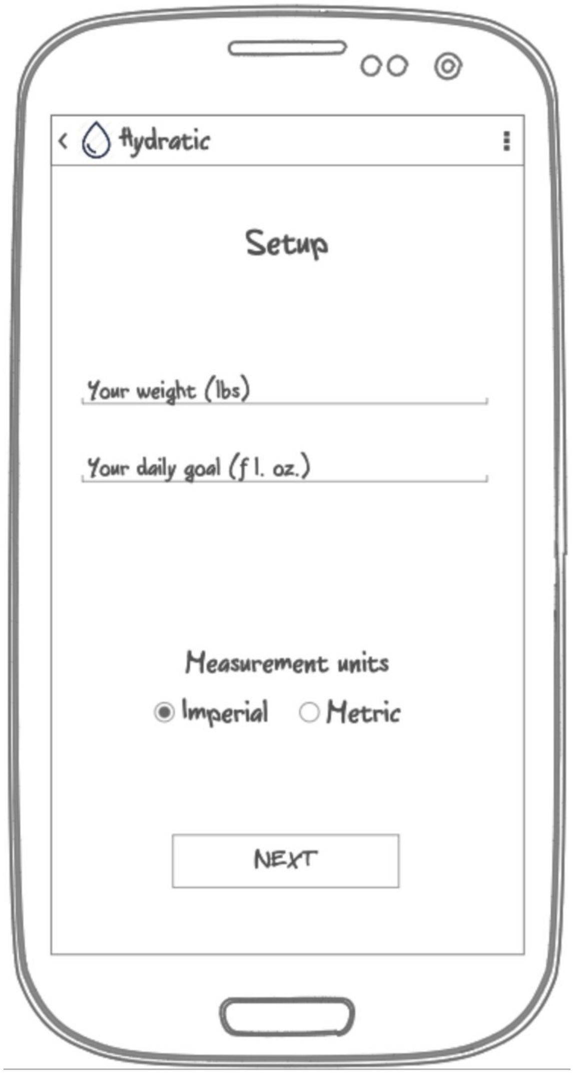
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1 - Firebase Authentication screen



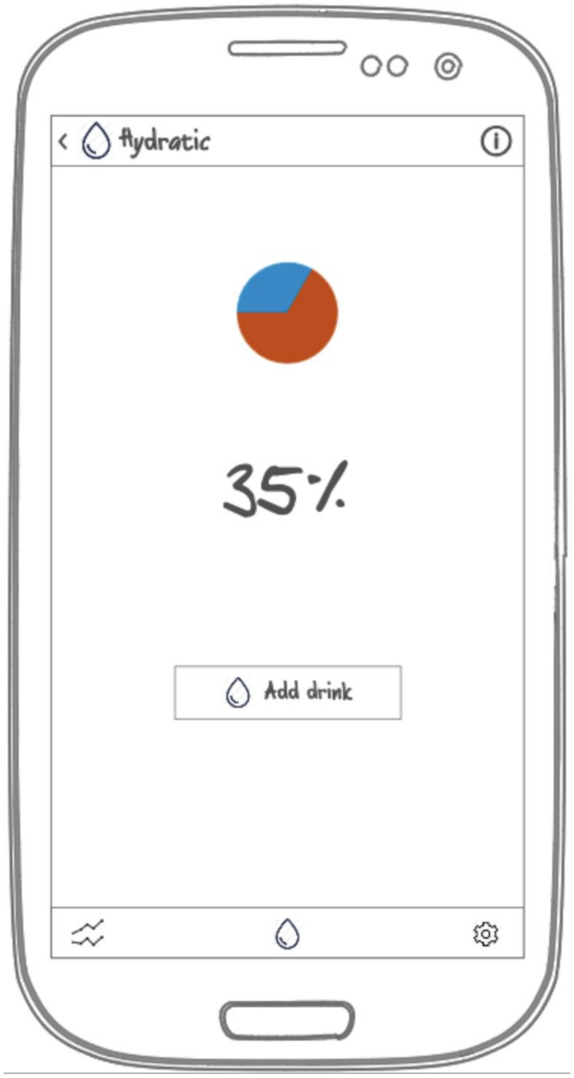
This is the screen shown to the user when the app is launched the first time. In order to use the app, the user must first authenticate. This is needed because the app uses Firebase Realtime Database to store the user’s data, so each user needs to be uniquely identified. To simplify this process we will use Firebase Authentication which should work seamlessly with the Realtime Database.

Screen 2 - User onboarding / Setup



This is the screen that gets launched only the first time the user opens the app after installing it and logging in. It is used to obtain the minimum information from the user that is needed to setup and run the application. The user will need to input the weight and select the daily hydration goal. The measurement units can also be configured between Imperial and Metric.

Screen 3 - Main screen



This is the main screen of the app. It will be the first screen the user will see after completing the onboarding process, and every time the app is launched from that point on.

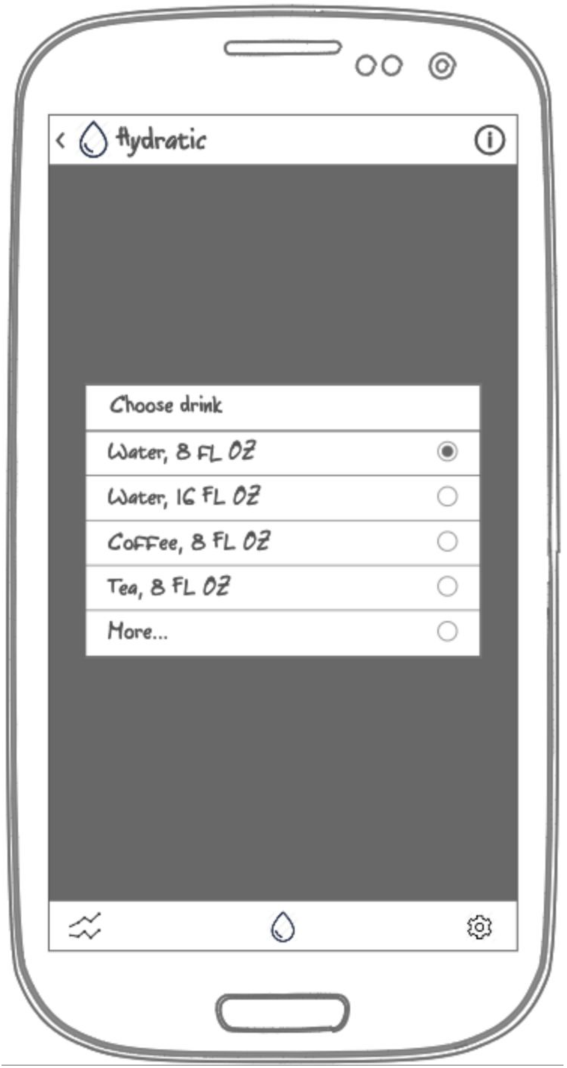
The screen will display the hydration percentage in both a written and graphical way to the user (note that the pie chart from the mock is there just to show that there will be a visual way of showing the progress, but the actual progress will be displayed in a more animated and pleasant way).

From this screen the user can also log a new drink that will add up towards the goal.

This screen also contains a bottom navigation bar that can take the user to three different screens. From left to ring: History, Main Screen, Settings

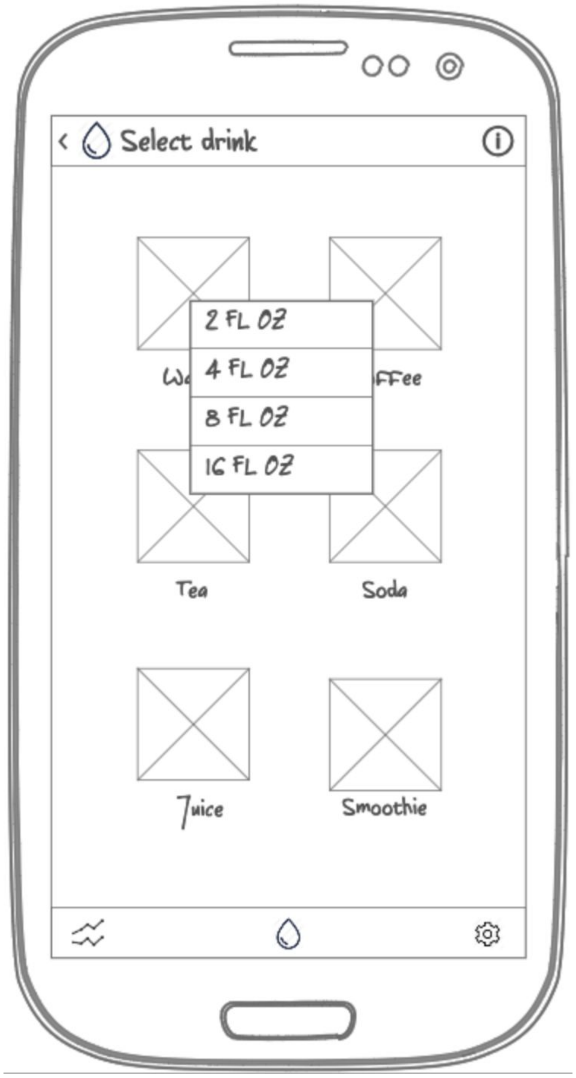
The Toolbar contains an action button that can open a Tips screen, where the user can read more useful information about why is it important to keep hydrated.

Screen 4 - Choose drink screen (dialog)



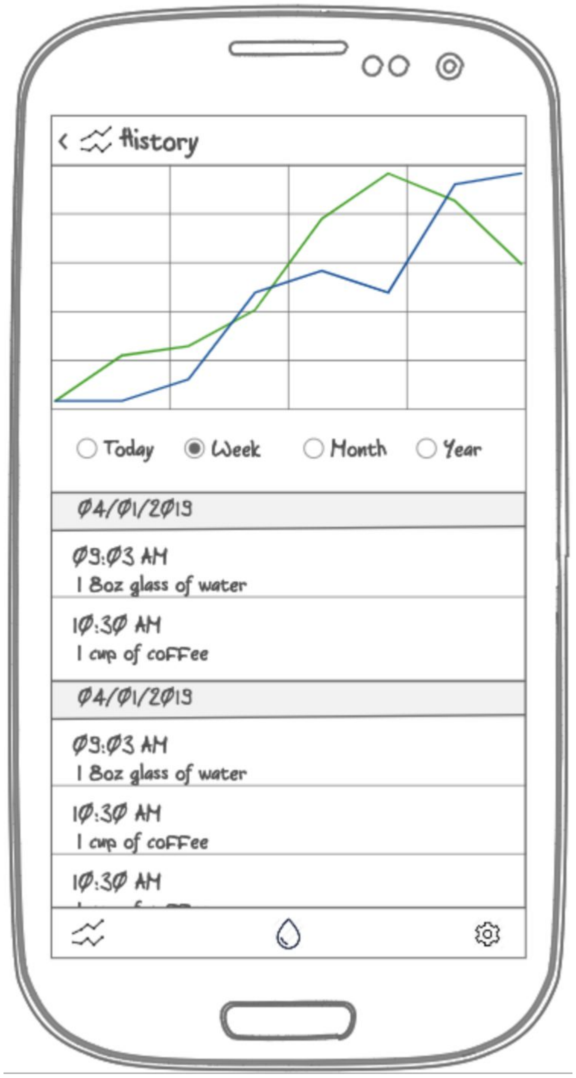
This dialog is launched when the user clicks on the “Add drink” button from the main screen and lets the user select a drink type that should be logged from a predefined list. The user can also choose the “More…” option to access more customizable options (this will take the user to the Select drink screen).

Screen 5 - Select drink screen



From this screen the user can select more drink options, which can be customized in terms of volume. The volume values are displayed in the selected units (selected from the onboarding screen or from the settings screen).
Each drink type will have its own specific icon so it can be easily recognizable.

Screen 6 - History screen



○ Today ● Week ○ Month ○ Year

04/01/2013

09:03 AM
1 8oz glass of water

10:30 AM
1 cup of coffee

04/01/2013

09:03 AM
1 8oz glass of water

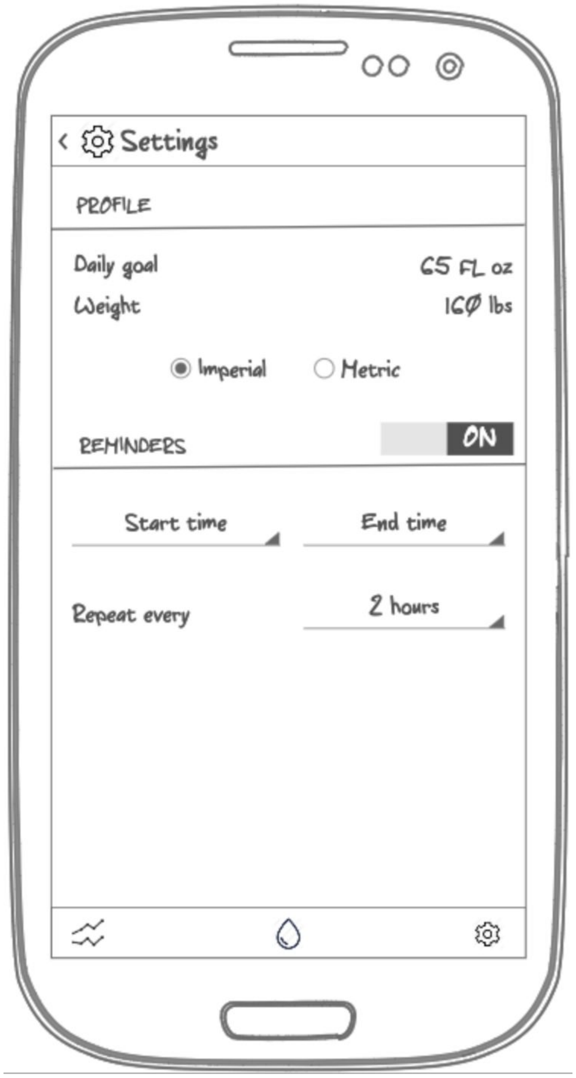
10:30 AM
1 cup of coffee

10:30 AM

In this screen the user can see the progress made during the current day, week, month or year via a line chart.

Below the chart the user can have access to a list containing all the data points that were logged in each day, each one displaying information like the drink type, volume and the hour when it was logged.

Screen 7 - Settings screen



Daily goal 65 FL oz

Weight 160 lbs

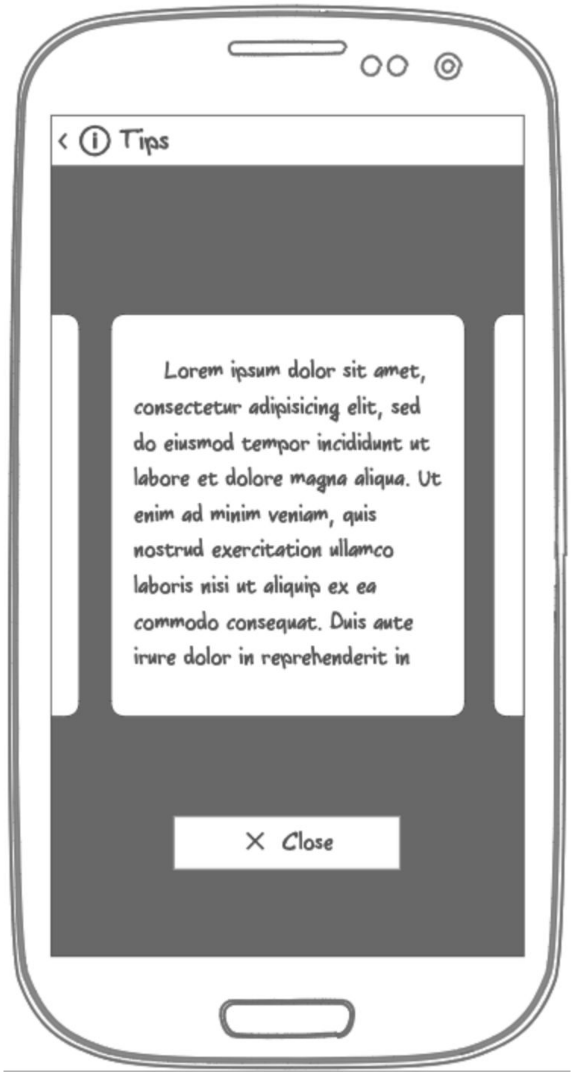
● Imperial ○ Metric

REMINDERS ON

Repeat every 2 hours

This screen allows the user to configure two main things: the user's profile and the reminders.
The settings from the user's profile are the daily hydration goal and the user's weight.
The reminders section has the following settings: the ability to turn reminders on or off, set an interval between the user would like the reminders to take place, and a frequency with which the reminders should occur.
This screen will also include a way to Sing Out the user.

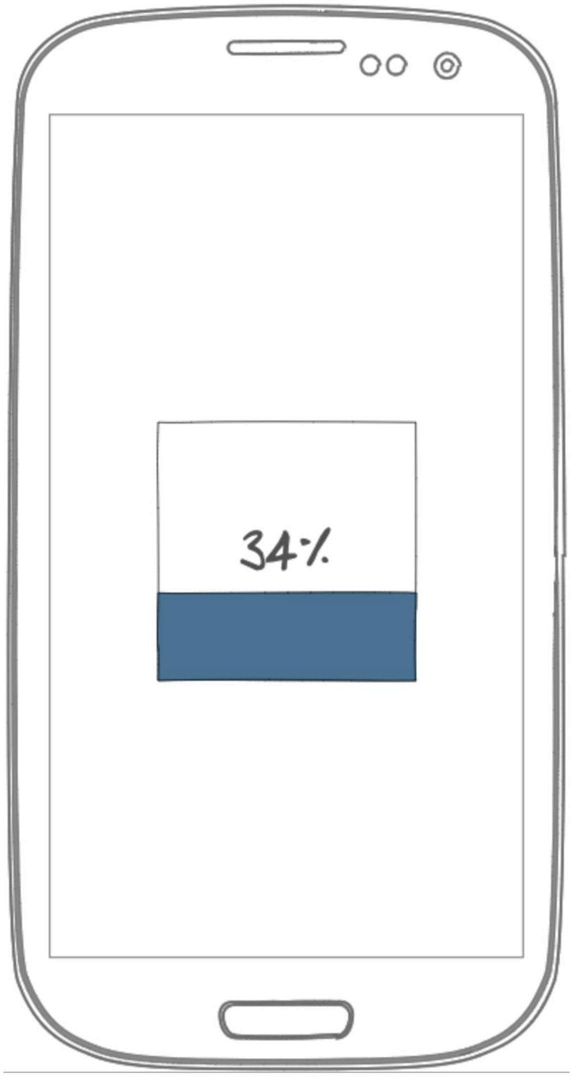
Screen 8 - Tips screen



Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in

The tips screen displays some useful tips for the user to get more familiar with hydration and why is it important. The tips are displayed in a ViewPager so the user can easily swipe between them. To exit this screen the user can either click on the back button or on the “Close” button present on screen.

Screen 9 - Widget



The widget will be a simple resizable rectangle that can display the hydration percentage both as a numeric value and as a graphical progress bar.
Tapping on the widget should take the user to the app’s Main Screen.

Key Considerations

How will your app handle data persistence?

The data that the user inputs will be saved inside a Firebase Realtime Database. To make sure that each user can access only his data, we will implement an authentication screen using Firebase Authentication, to create a seamless experience for the user.

For smaller pieces of data, like whether or not the Onboarding screen was shown, we will use the Android SharedPreferences.

Describe any edge or corner cases in the UX.

After logging in the first time, the user will be taken to the Onboarding / Setup screen to input some data needed to configure the app for the first use. Starting with the second time the app is launched, we need to make sure that the onboarding screen is not shown to the user. We can do this by saving a key-value pair inside the SharedPreferences the first time the Onboarding screen was shown to the user, and then only show this screen again the there is no such value in SharedPreferences.

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

Libraries used:

- ButterKnife v10.1.0: https://github.com/JakeWharton/butterknife
  - It is used for view binding using annotation processing to generate boilerplate code
- MPAndroidChart v3.0.3: https://github.com/PhilJay/MPAndroidChart
  - It will be used in the History screen to display a line chart of the hydration levels for the current day, week, month or year.
- Lottie v2.7.0: https://github.com/airbnb/lottie-android
  - It will be used to add delightful animations for the user when certain actions are performed.

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

External services:

- Firebase Authentication: https://firebase.google.com/docs/auth/
  - Will be used to authenticate users, so that we can easily use Firebase Realtime Database
- Firebase Realtime Database: https://firebase.google.com/docs/database/
  - Used to store the user's data after being logged in.

Accessibility

The app will be built with Accessibility in mind, so the following must be achieved:

- Set content descriptions for any ImageViews
- Make sure the app supports navigation using a D-pad, by making sure the UI elements are focusable and can be accessed in the correct order that makes sense to the user.

Project Resources

All resources will be stored in the Android **res** folder, in their appropriate subfolders and files. E.g.: values/colors.xml, values/strings.xml, values/styles.xml
Note that if different resources will be required for different screen orientations, for different screen sizes or for different languages, specific **values** folders will be used (i.e. values-land for landscape resources).

IntentService implementation

For any tasks that might block the UI thread, an IntentService will be used to make sure these tasks are made asynchronously. All the requests will be sent to the IntentService by calling the startService() method with the Intent describing the action and containing the data as the argument.
The onHandleIntent() method will be implemented, that will receive all the intents, launch a worker thread and execute the asynchronous task.

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

- Create the project using **Android Studio v3.3.2** and configure it with the following:
  - targetSdkVersion: **API Level 27**
  - minSdkVersion: **API Level 22**
  - Gradle version: **v3.1.1**
- Configure the project so that the app is written solely in the **Java Programming Language**
- Set up as git repository and push to GitHub
- Add and configure libraries
- Configure Firebase services

Task 2: Implement UI for Each Activity and Fragment

- Build UI for Login Activity
- Build UI for Onboarding Activity
- Build UI for Main Activity
- Build UI for the Add Drink Dialog

- Build UI for the Select Drink Activity
- Build UI for the History Activity
- Build UI for the Settings Activity
- Build UI for the Tips Fragment

Task 3: Create Firebase Authentication Activity

- Create layout
- Integrate Firebase Authentication
- Handle success and error cases

Task 4: Setup Firebase Realtime Database

- Define the data that needs to be stored
- Setup database

Task 5: Create the Onboarding Flow

- Create layout
- Add logic to convert from Imperial to Metric units and vice versa.
- Store the user’s data inside the database

Task 6: Create the Main Activity

- Create layout
- Configure the Toolbar
- Add and configure the Bottom Navigation Bar
- Read the user’s data from the Firebase Realtime Database and populate the UI
- Add a graphical representation of the hydration percentage (probably by using a Lottie animation)
- Setup the “Add drink” button to launch the Dialog used to select a drink

Task 7: Create the Select Drink Activity

- Create layout
- Add icons for each supported drink type
- Add sub-menu that is launched by clicking on either drink type to select the volume of the recipient

Task 8: Create the History Activity

- Create layout
- Implement the MPAndroidChart view to display a line chart of the user's history
- Add logic that can populate the line chart with data from different periods of time, based on the user's selection.
- Create RecyclerView and Adapter to display all the data that the user has saved

Task 9: Create the Settings Activity

- Create layout
- Implement logic that updates the user's data inside the Firebase Realtime Database in case the user changes any profile data.
- Implement an AlarmManager to send notifications to the user between a preselected period of time.
- Implement a Sing Out mechanism

Task 10: Setup Tips Fragment

- Create layout
- Implement a ViewPager and Adapter that holds cards through which the user can swipe and read useful information.

Task 11: Setup Widget

- Implement an Android Widget that displays the hydration percentage as both a numeric value and as a graphical progress bar.
- Set the Widget's on click action to open the app to the Main Screen

---

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**"