How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: "**Capstone_Stage1**"
3. Replace the text in green

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"

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.
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: gr8scott88

HIIT It!

Description

Get the most of your workout by utilizing High Intensity Interval Training (HIIT) to burn more calories in less time. HIIT It! Allows you to quickly start a simple interval training session, load a preset methods (such as Tabata) or customize your own workouts.

Intended User

This is an application for fitness minded individuals. The target user could have access to a gym, or could simply have a space in which to workout. The principles of HIIT training can easily be done using bodyweight style exercises. Typically those people utilizing HIIT are not purely casual gym goers, but are more interested in getting the most out of their time at the gym and really want to push themselves to their limits.

Features

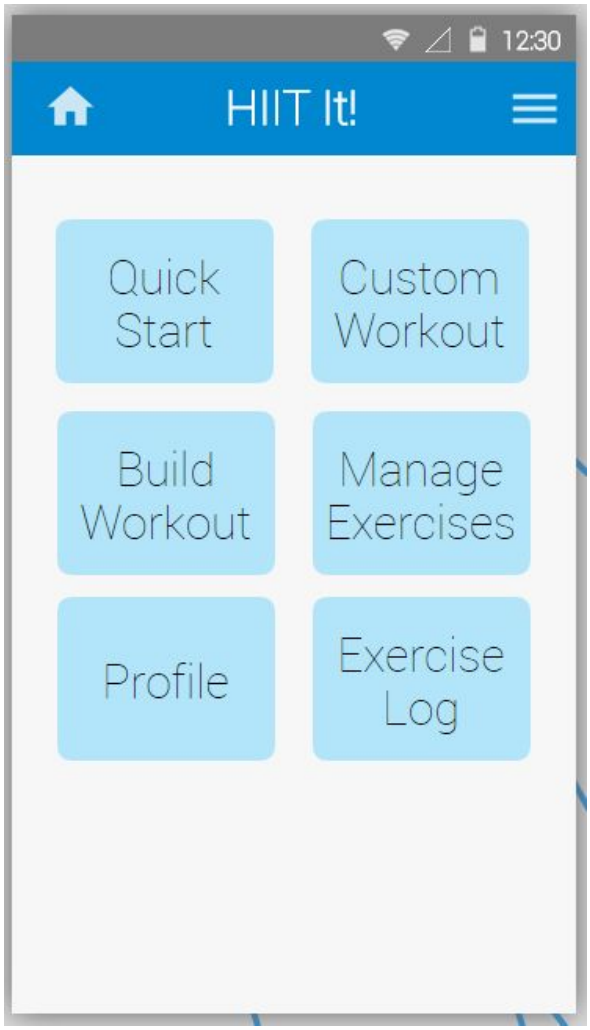
- Simple interval timing
- Pre-set intervals (Tabata, etc)
- Custom circuit training (custom timings that can be saved)
- Workout frequency tracking
    - Days in a month
    - Duration per day
- Personal fitness tracking (user input)
    - Weight
    - Waist size
    - Body fat %

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 Photoshop or Balsamiq.

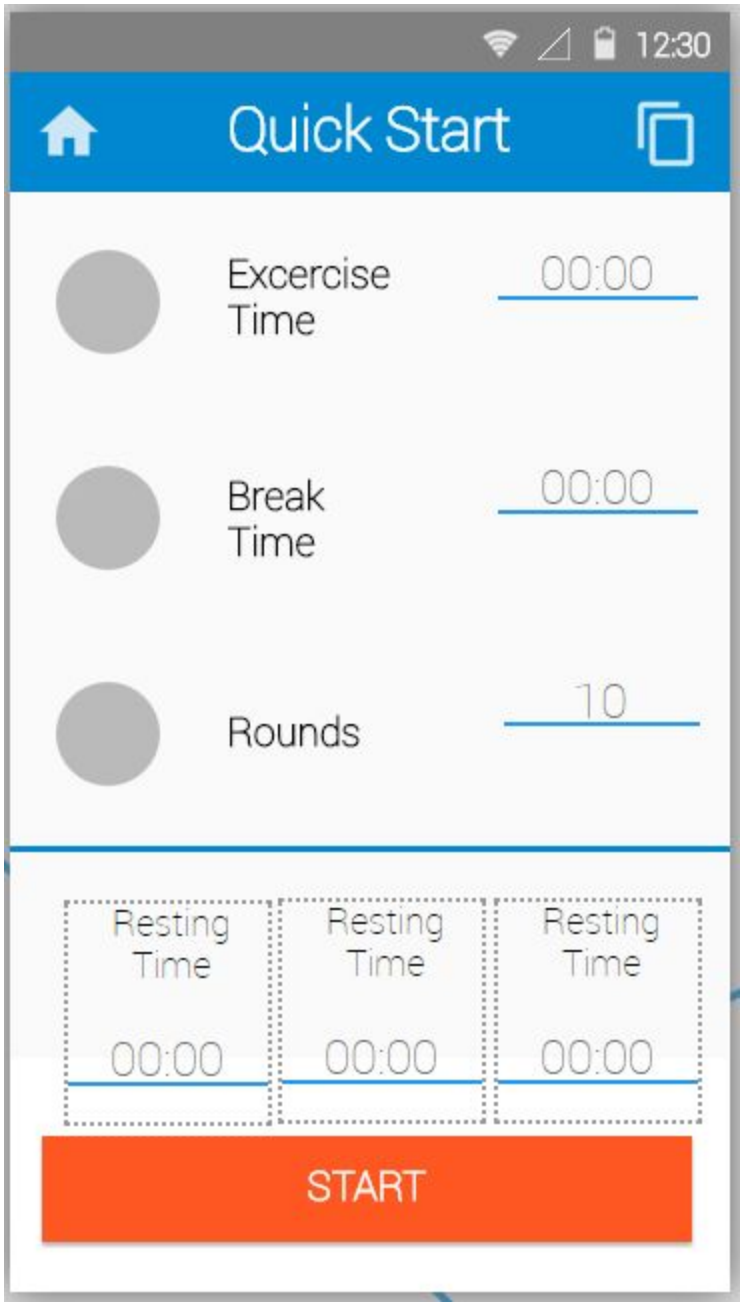
All screens mocked with www.fluidui.com

Screen 1



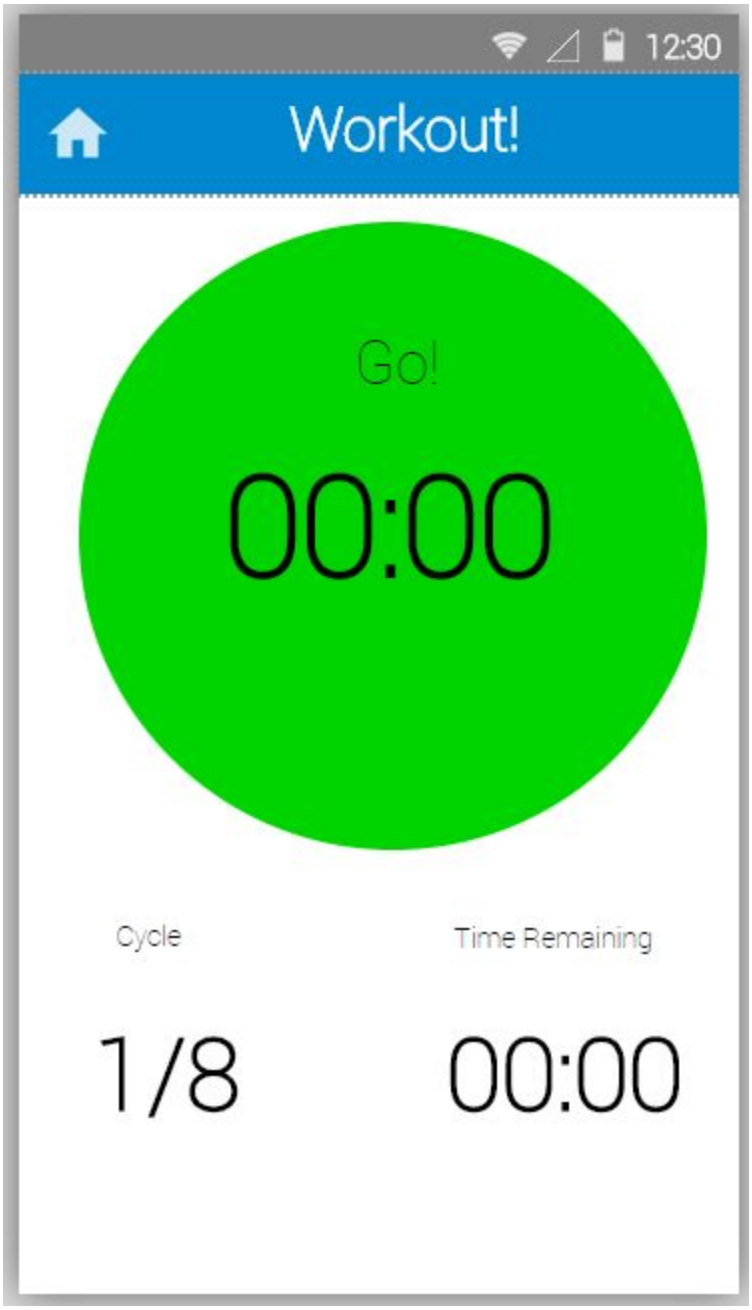
The main screen just provides a grid of options for the user. This will inevitably change aesthetically I just don't have a good feel for it yet.

Screen 2



This screen will fill the exercise times with the last used exercise profile. The icon in the top right allows for an alternate timer profile to be loaded. The bottom values calculate the total workout information.

Screen 3:

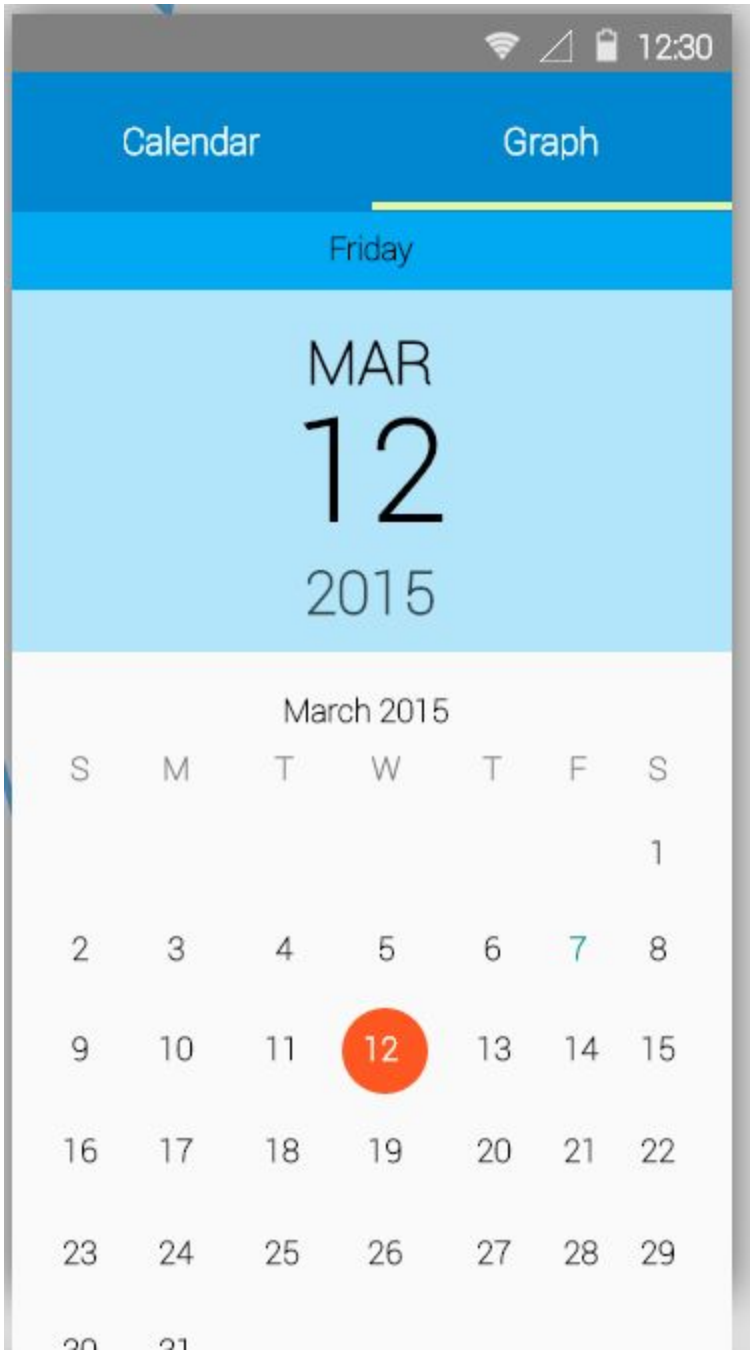


This screen will show a simple timed workout. The “simple” workout does not keep track of exercises. The timers will show round, time remaining, and current interval step (resting, going)

Screen 4:

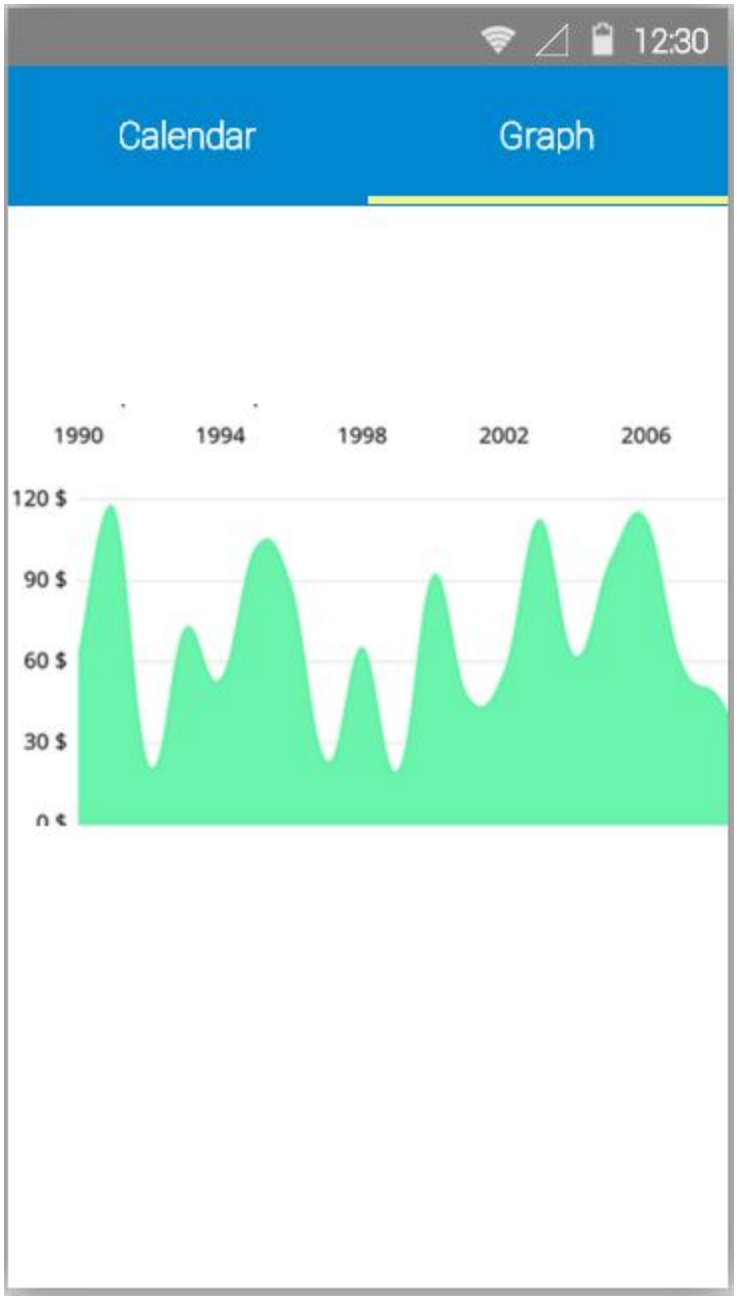

12:30
Load Workout
Workout Timing 1
Workout Timing 2
Workout Timing 3
Workout Timing 4

Screen 5-1:



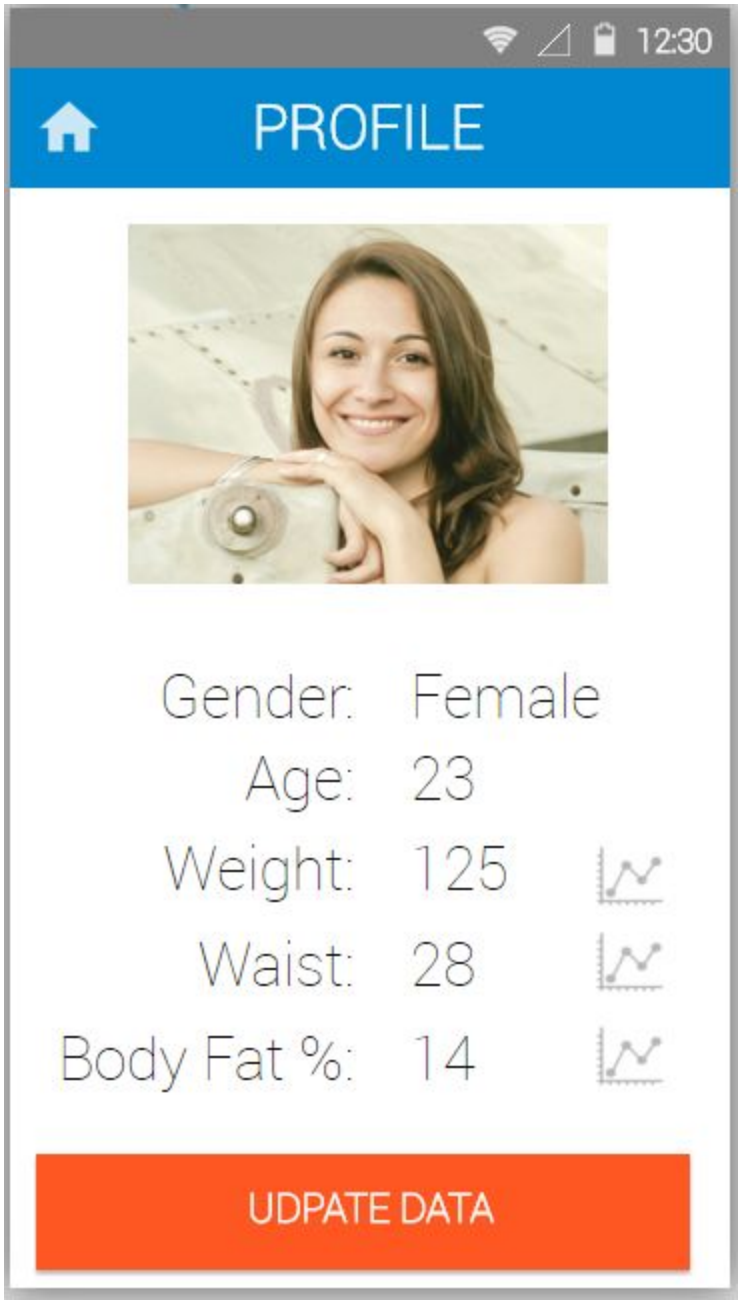
This screen will show the user what days they have logged exercises on. All days will be highlighted in orange.

Screen 5-2:



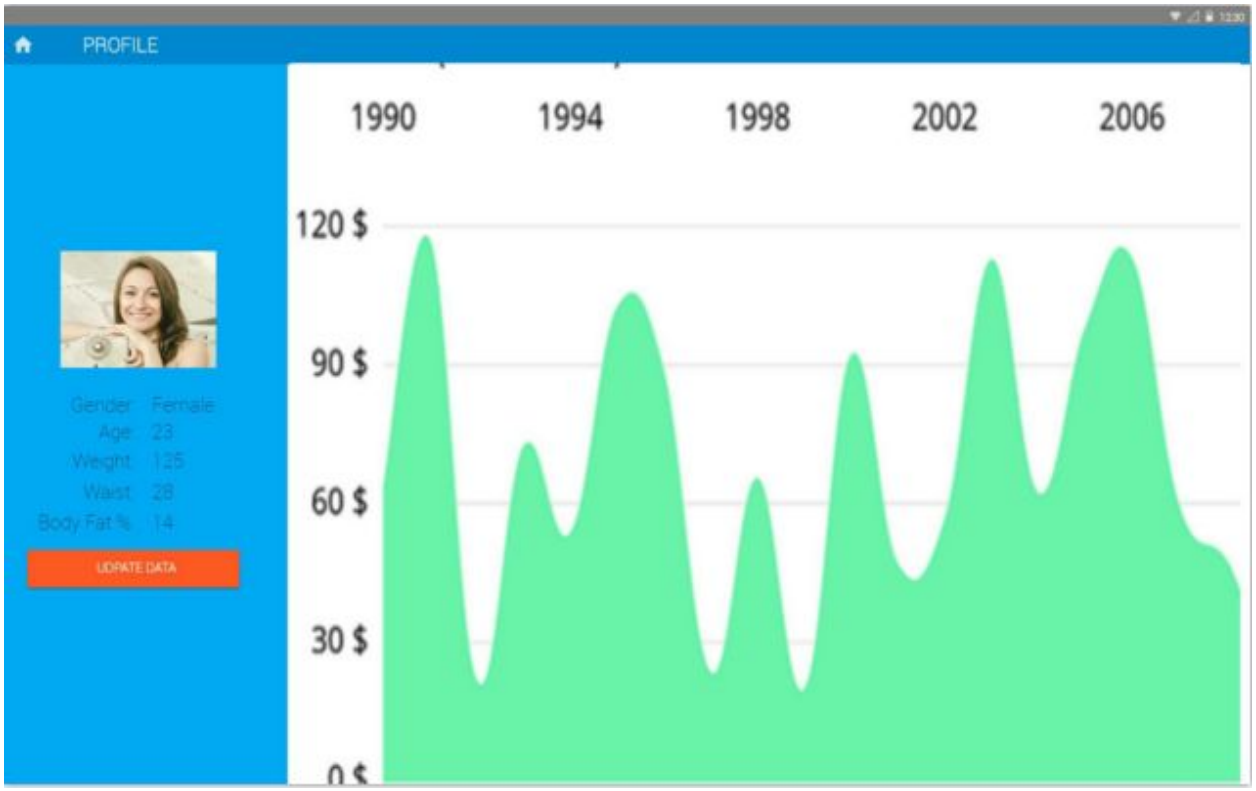
This screen will show the user their workout duration as a function of time. The x-axis will be date and the y-axis will be duration.

Screen 6:



This screen will show the user profile (mainly limited to physical stats). Each stat will be tracked and the user can view a graph of the statistic as a function of time. The button will allow the user to enter / update their stats.

Tablet Screen 1:



As this is primarily a fitness app, primary utilization is intended to be via a phone sized device. A tablet view makes the most sense when viewing historic information. For that reason, the profile view will be optimized to work on a table screen such that plots can quickly be referenced without having to go to a separate screen.

Key Considerations

How will your app handle data persistence?

Data will be handled via a content provider and stored in SQLite database.

Describe any corner cases in the UX.

Ensure it works well on very small screens (as this is a phone based device, it is possible that a user has a low powered device with low res screen)

Ensure that the application continues to function when a user closes the app via the back stack while an interval is running (interval should continue to run and “re-connect” if the application is brought back to focus)

Ensure that workout information is saved even if the interval completes while the application is not in focus.

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

I intend to use “MPAndroidChart” for the purposes of plotting. I’ll also make use of “Butter Knife” for data binding to minimize boilerplate code.

Next Steps: Required Tasks

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

Task 1: Project Setup

- Import required libraries through the gradle files.
- Stub primary activity files.

Task 2: Implement UI for Each Activity and Fragment

- Function only implementation of UI for activities and fragments:

- Main Screen
- Quick Workout Screen
- Workout Screen
- Profile Screen
- Workout History Screen
- History Tracking Screen

Task 3: Implement Basic UI Functionality and Dummy Data

- Link all screens through click handlers, ensure proper “flow” of activities and fragments
- Generate dummy datasets for preliminary functionality:
  - Profile info
  - Pre-configured workouts
- Implement dummy data into basic UI

Task 4: Create DataProvider and SQLITE Database

- Generate SQLITE database and necessary tables to hold real data
- Create DataProvider to query stored data and store new data

Task 5: Implement Timing Functionality

- Create service to implement timing functionality
  - This ensures that the app will continue to function even if it is closed while an interval is running

Task 6: Materialize it

- Go back and revamp UI to ensure it meets material design requirements
  - Re-color
  - Modify text sizes as necessary

Task 7: Implement Accessibility Features

- Ensure all features have proper accessibility information
  - Poor vision
  - RTL vs. LTR text orientation

Task 8: Extended functional testing

- Check for bugs, fix as necessary

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"