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: ffs1985

My Wod Book

Description

My Wod Book allows you to store and categorize your wod routines each time you enter a crossfit box. Crossfit requires you to log your trainings exercises in order to keep progress, and select the weight for the training you are going to perform. MyWodBook will allow you to add this information, and will automatically categorize it in order to you be able to estimate the weight to your next training. It will also store automatically your PR’s (Personal Records).

Intended User

This app is intended for crossfitters, but in the future can be adapted to any person that wants to train and log their works.

Features

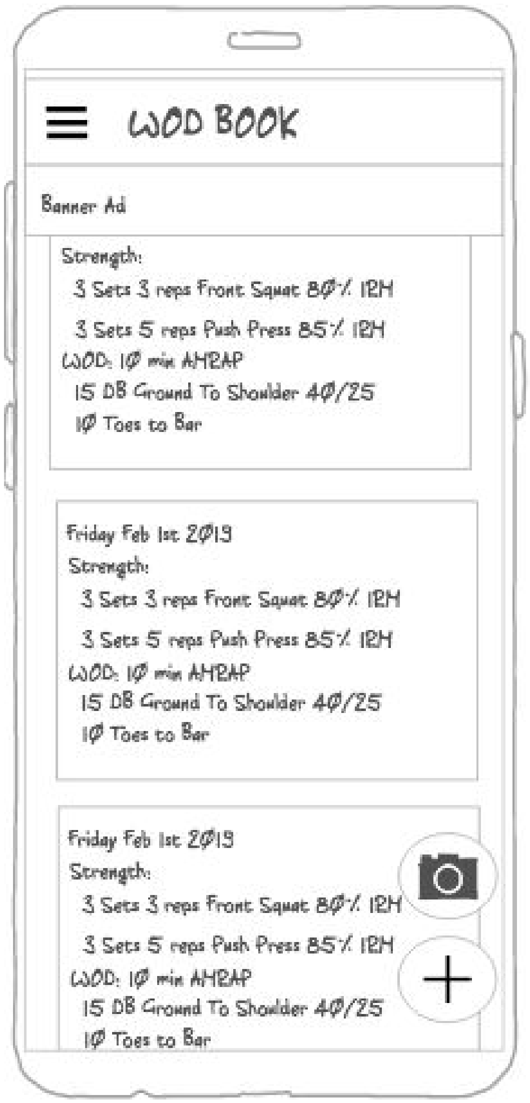
MyWodApp main features are:

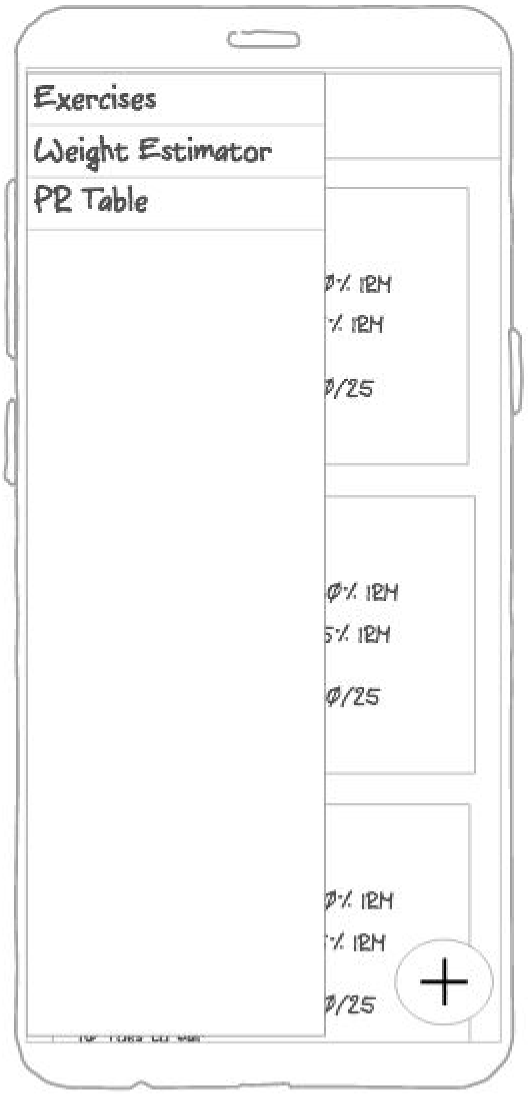
- Load WOD (Work Of the Day) information.
- Auto categorize data loaded.
- Provide an already populated exercise list of exercises.
- View Progress data with graphics per exercise.
- Filter data per categories like muscle group, date, exercise, etc.
- Show PR's information.
- Take pictures and pre-load data from there.

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.

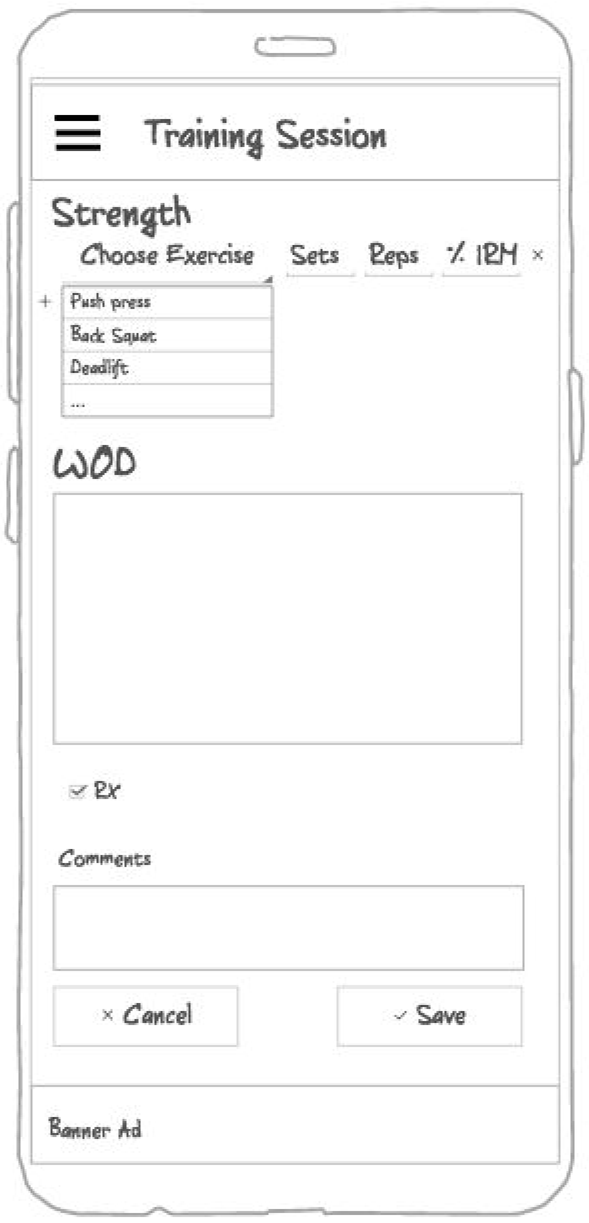
Main Screen


WOD BOOK
Banner Ad
Strength:
3 Sets 3 reps Front Squat 8Ø% 1RM
3 Sets 5 reps Push Press 85% 1RM
WOD: 1Ø min AMRAP
15 DB Ground To Shoulder 4Ø/25
1Ø Toes to Bar
Friday Feb 1st 2Ø19
Strength:
3 Sets 3 reps Front Squat 8Ø% 1RM
3 Sets 5 reps Push Press 85% 1RM
WOD: 1Ø min AMRAP
15 DB Ground To Shoulder 4Ø/25
1Ø Toes to Bar
Friday Feb 1st 2Ø19
Strength:
3 Sets 3 reps Front Squat 8Ø% 1RM
3 Sets 5 reps Push Press 85% 1RM
WOD: 1Ø min AMRAP
15 DB Ground To Shoulder 4Ø/25
1Ø Toes to Bar



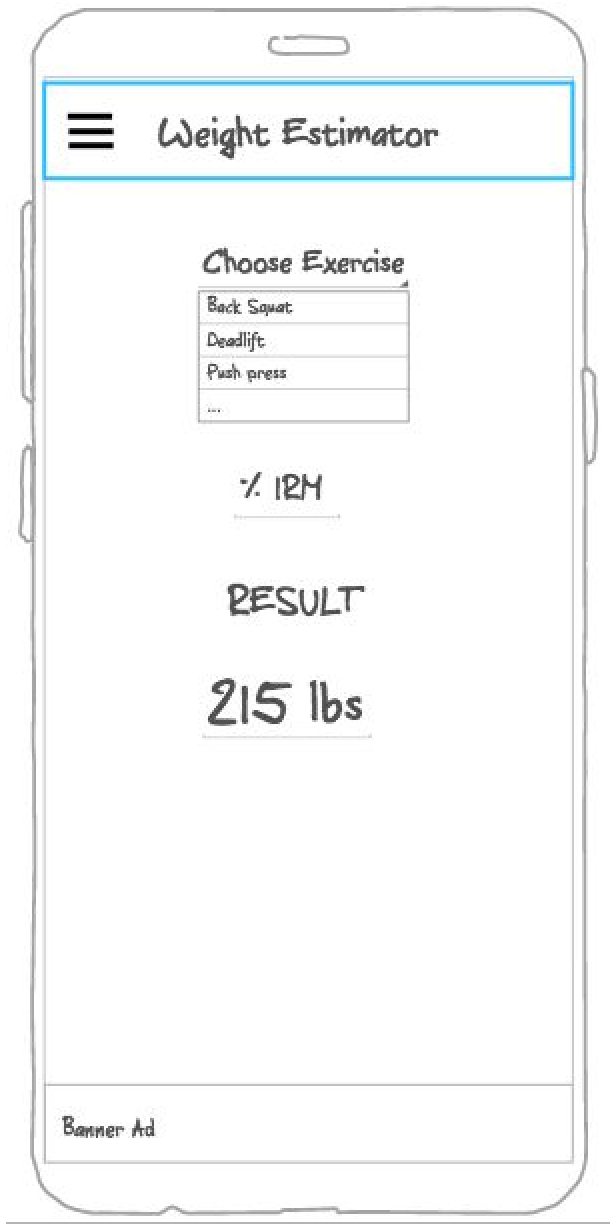
Main Screen have a list of the logged trainings ordered by date.

Add/Edit Training Session



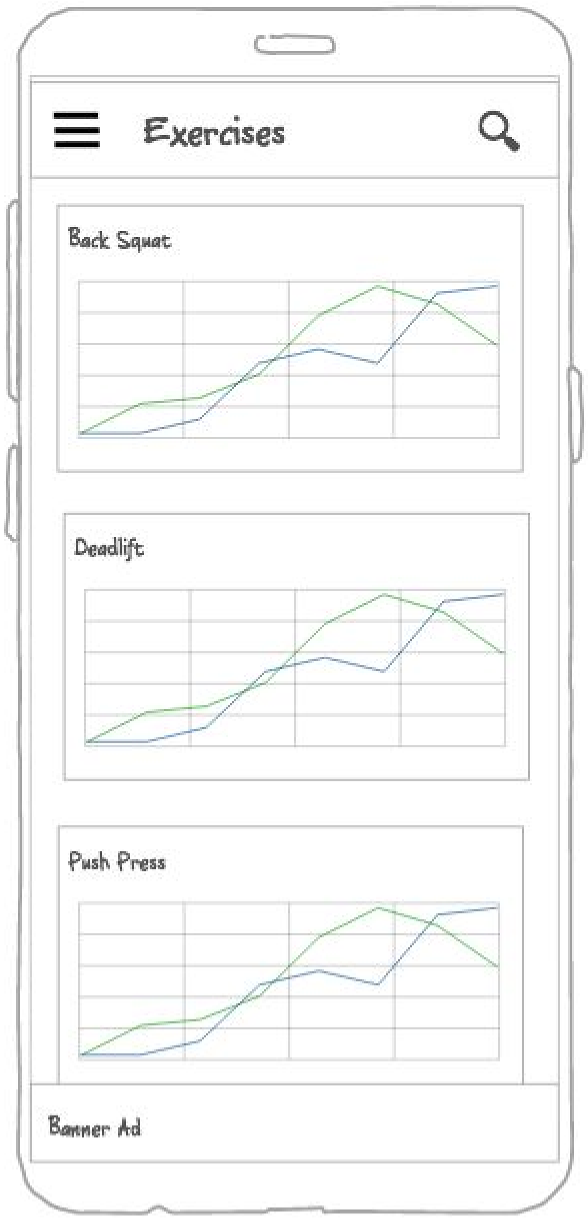
This screen allows you to add or edit the information of a training session.

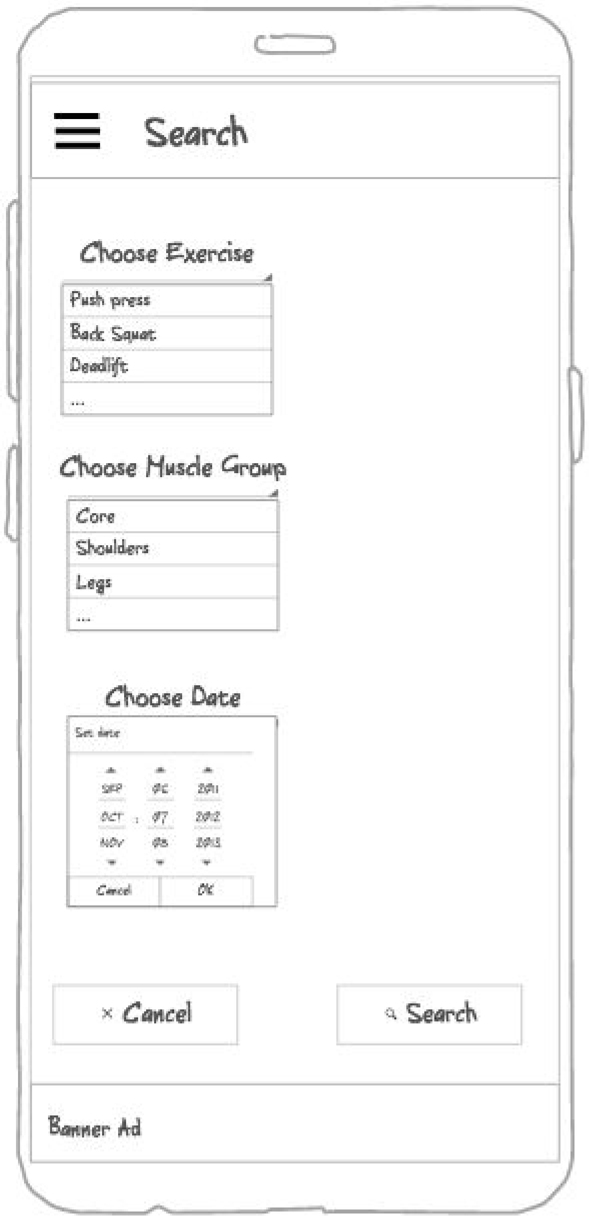
Estimate Training Weight



This screen allows you to obtain an estimated weight to select on a certain training.

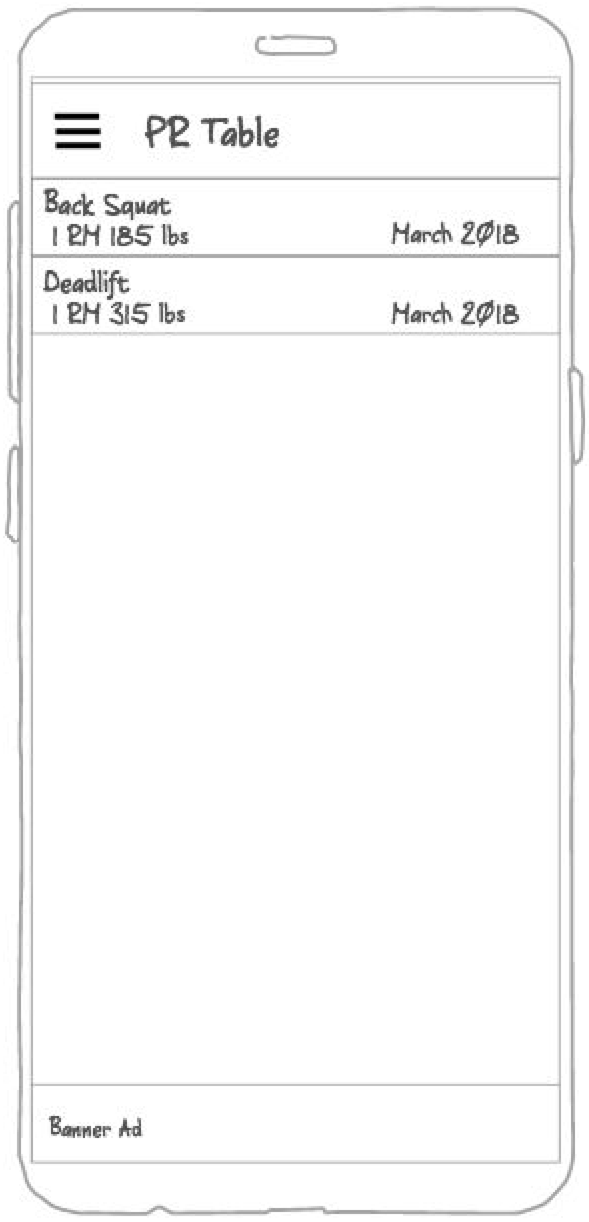
Exercise History Data





This screen shows you the information of your training sessions for a particular exercise.

PR Records



This screen shows you the list of your personal PR based on the information logged.

Key Considerations

How will your app handle data persistence?

We are going to store the user information and all the logged information of the training sessions made by that user.

To achieve that we are going to use Android Architecture Components Room and Lifecycle. For that we are going to generate each class for each table to store with @Entity annotation. Also we are going to provide @Dao interfaces to perform all the operations with the database. Also we are going to use LiveData in order to return the information to ensure the data we are seeing is updated.

Describe any edge or corner cases in the UX.

For tablet and big screens, we are going to show a grid instead of a list for the main log screen, and the exercises table.

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

Butterknife to bind field methods and views. This will allow us to avoid boilerplate code.
Icepick to save and restore instance states.
Dexter to simplify the process of requesting permissions at runtime.
Retrofit - To make http connections simpler.
GraphView - To plot and show chart information of the logged training data.

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

To recognize text from photos we are going to try with 2 approaches.
First we are going to try to use Firebase ML Kit. This allows us to do this on the cloud or in the device itself if there is no connection. The only issue with this is as my latest test the accuracy of this approach wasn't the expected. The 2nd possible approach is going directly to use the GCP Vision API. This api use Auto ML Vision capabilities of GCP and seems like works better with OCR text recognition.

To show Ads we are going to use Admob Firebase. For this are going to follow this steps for a Banner Ads
https://developers.google.com/admob/android/banner

Next Steps: Required Tasks

Task 1: Project Setup

This task consist in create the initial Project structure with all the required libraries and define the initial skeleton of the app.

- Configure libraries
- Design architecture visual components solution
- Implement initial skeleton of the app.

Task 2: Implement UI for Each Activity and Fragment

This task will consist in create all the Activities, Fragments and all associated classes and xml files to visually implement all the view components of the app

- Build UI for Main Logs View
- Build UI for Add/Edit Training Session
- Build UI for Exercises View
- Build UI for Search Exercise View
- Build UI for PRs Table View
- Build UI for Camera Picture View

Task 3: Implement Database layer

This task will create all the required classes to handle the persistence layer.

- Entities
- DAOs
- Database
- ViewModels

Task 4: Implement Service layer

- Create Exercise list service
- Create the service to extract info of the picture.

Task 5: Connect Persistence with UI

- Connect ViewModel with Activity, Fragments and Adapters

Task 6: Implement AdMob

Implement AdMob in all the views.

Task 7: Testing

- Add unit test for services.
- Add integration tests for main flows.

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**"