

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.](#)

[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: KhamellT

Gymmee

Description

Gymmee is the app that allows athletes and first time gymmers to have a real personal trainer through an android app.

Personal trainers will give to their athletes ad-hoc exercises based on their past experiences and shape. Feedback will be granted in real time for trainers, while gymmers will have to wait for the trainer to give them exercises each week.

Exercises can be executed both at home and in a gym depending on the trainer's guidelines.

Intended User

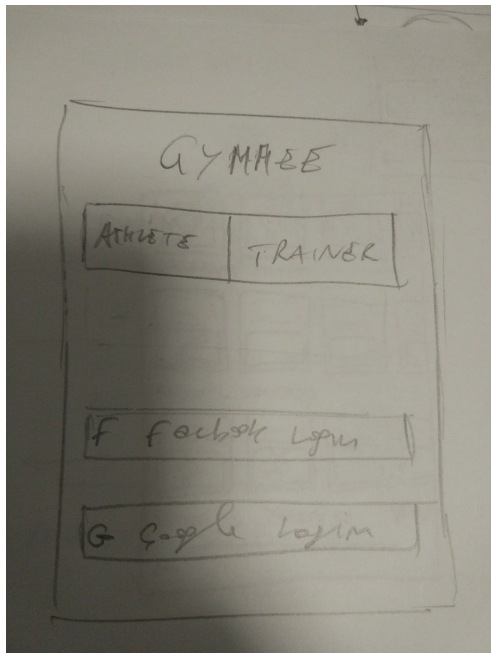
This is only for affirmed athletes or beginners.

Features

- Saves workouts, user and trainers information in cloud (Firebase)
- Saves locally current user data (Firebase)

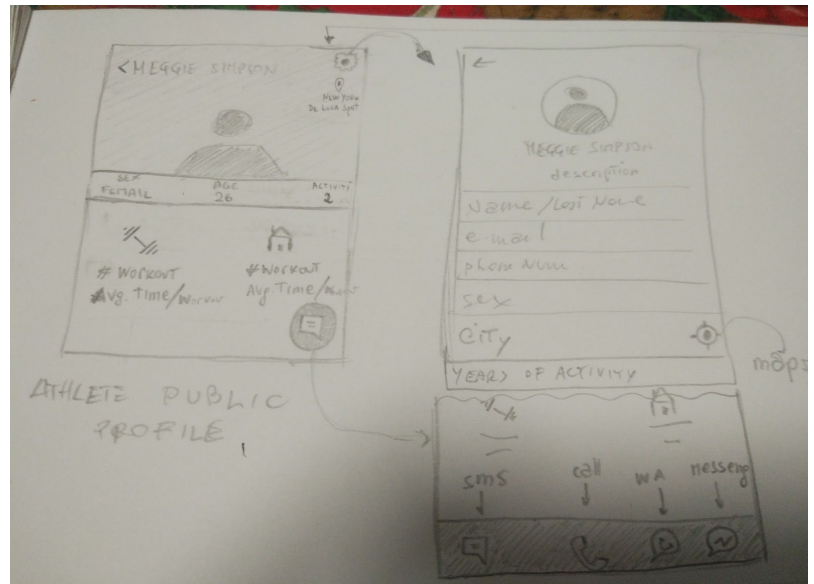
User Interface Mocks

Screen 1 - Login + Trainer/Athlete choice



Login with facebook or google to go to Screen 2

Screen 2 - User Public Profile



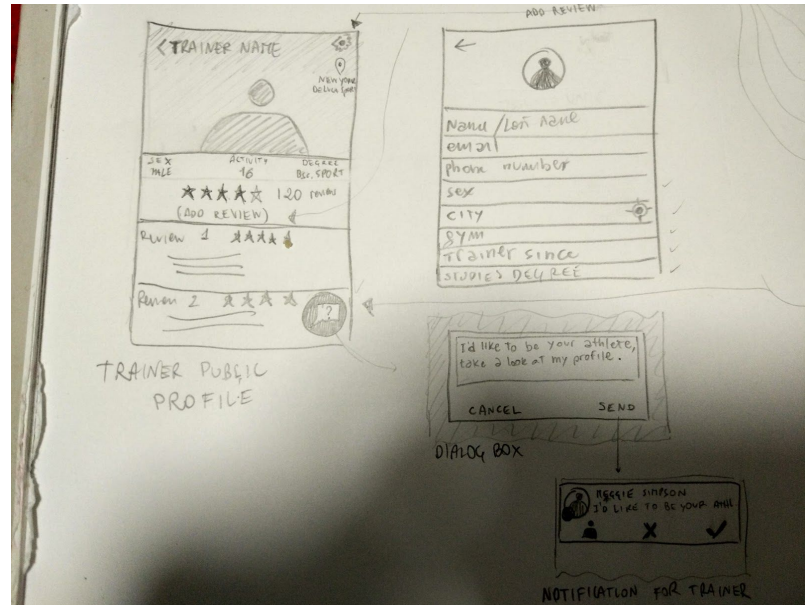
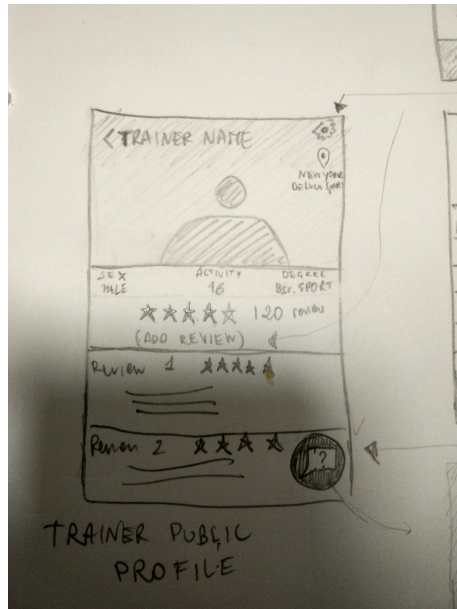
Click on gear icon top right to go to screen 3

Click on fab to open communication panel (only trainer)

Screen 3 - User Edit Data



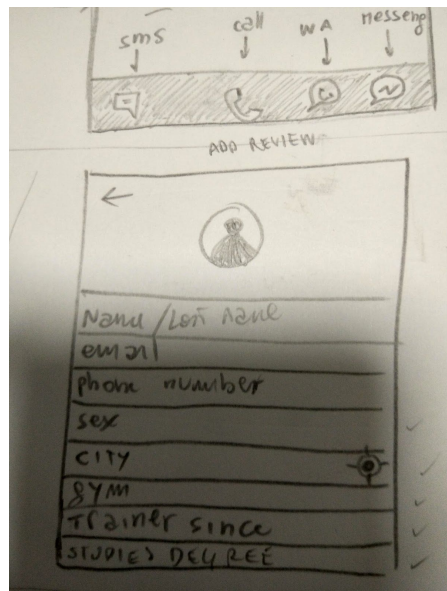
Screen 4 - Trainer Public Profile



Click gear Icon to go to screen 5

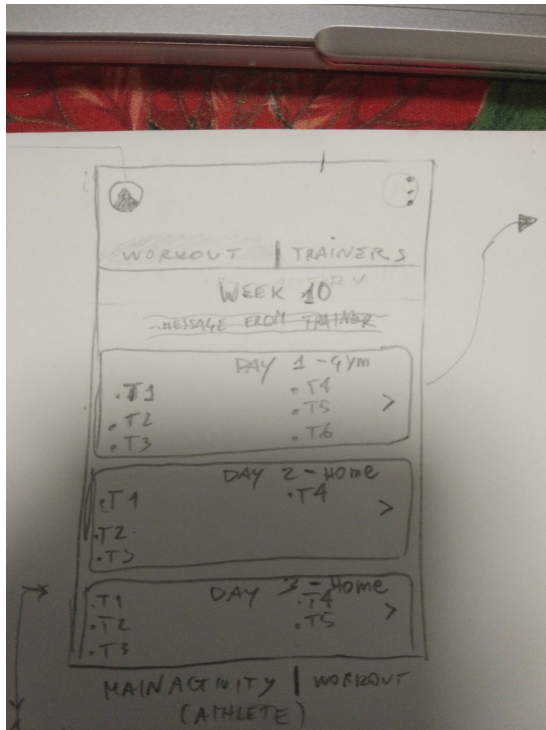
Click the fab to ask the trainer to be his/her athlete

Screen 5 - Trainer edit data



Click any field to edit it

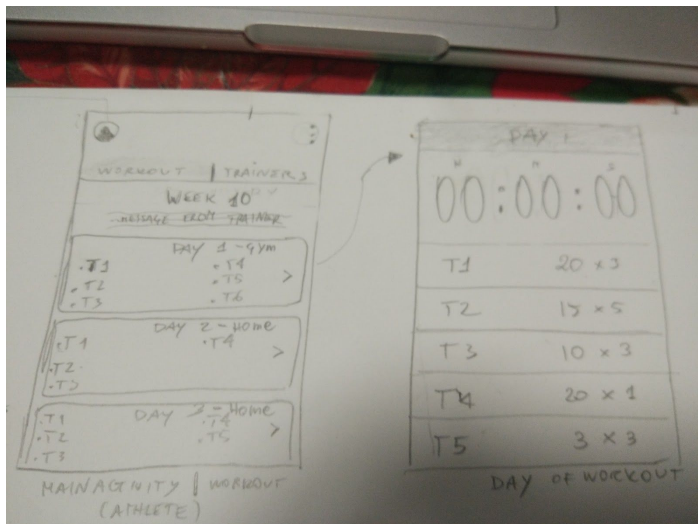
Screen 6 - User MainActivity tab Workout



Click a day Cardview to go to screen 7

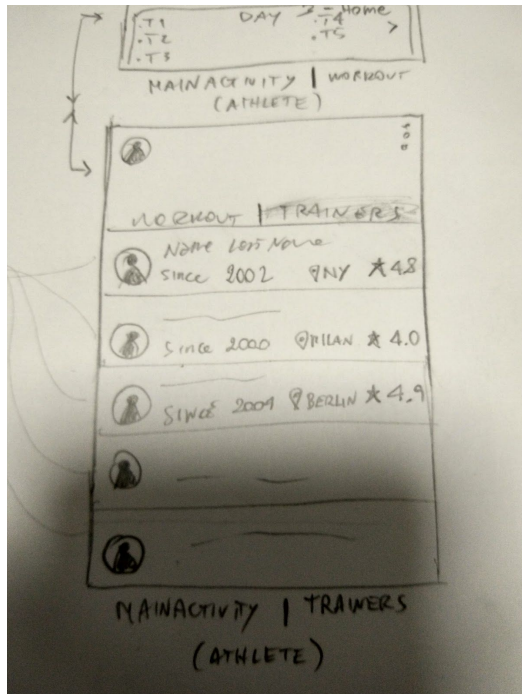
Click User profile photo in the top left corner to go to Screen 2

Screen 7 - Workout Day detail with timer



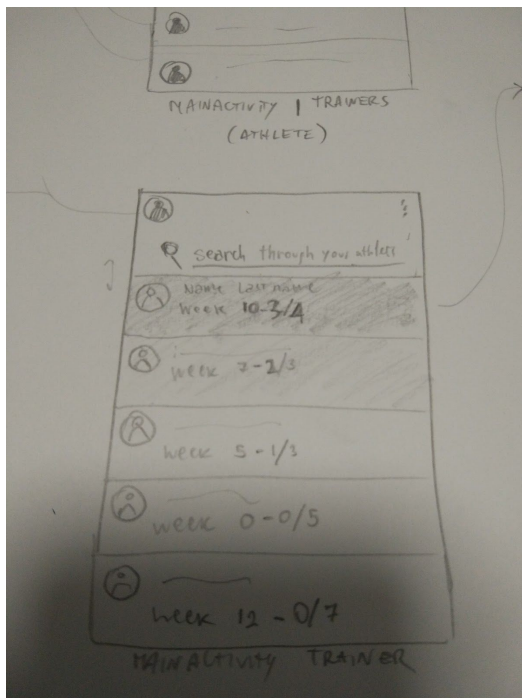
The one on the right.

Screen 8 - User Mainactivity tab Trainers



Click on trainer element to go to screen 4 (User can leave a review)

Screen 9 - Current Trainer's Athletes list

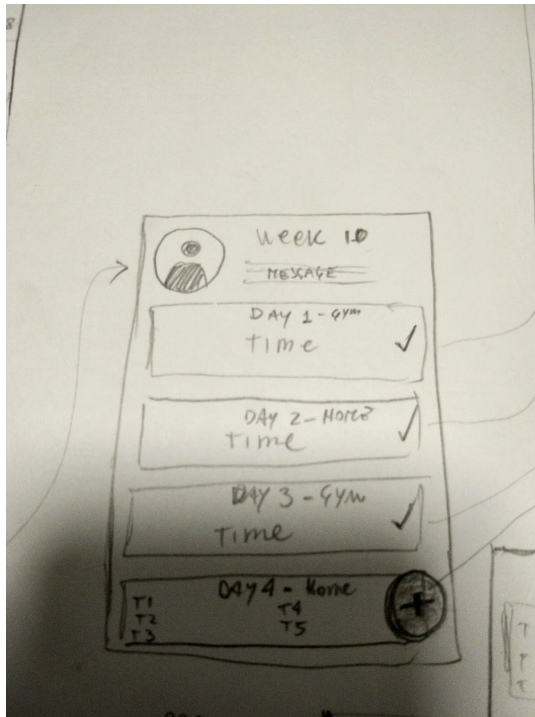


Click Trainer icon in top left corner to go to screen 3

Click Athlete element to go to screen 9

Use Searchbar to find specific athlete. Ordered by week workout completion

Screen 10 - Athlete Workout schedule seen by Trainer

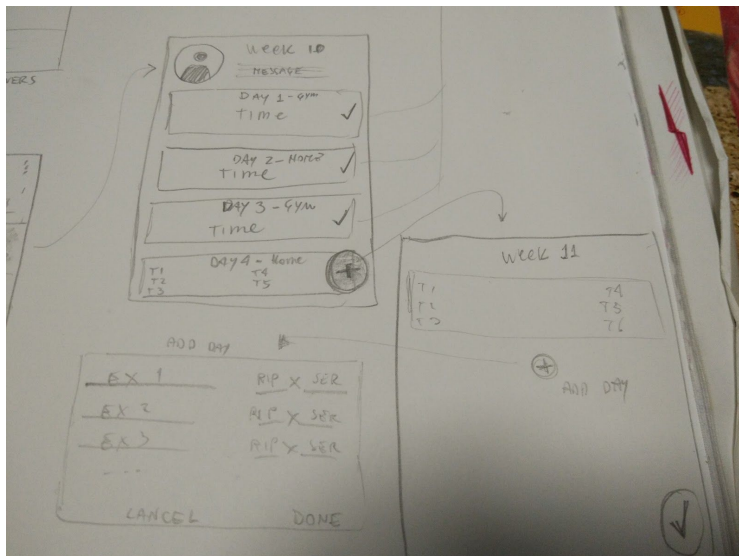


Click message field to put a message for athlete

Click day workout element to go to screen 7 (Screen 7 now editable)

Click fab in the bottom right corner to go to screen 11

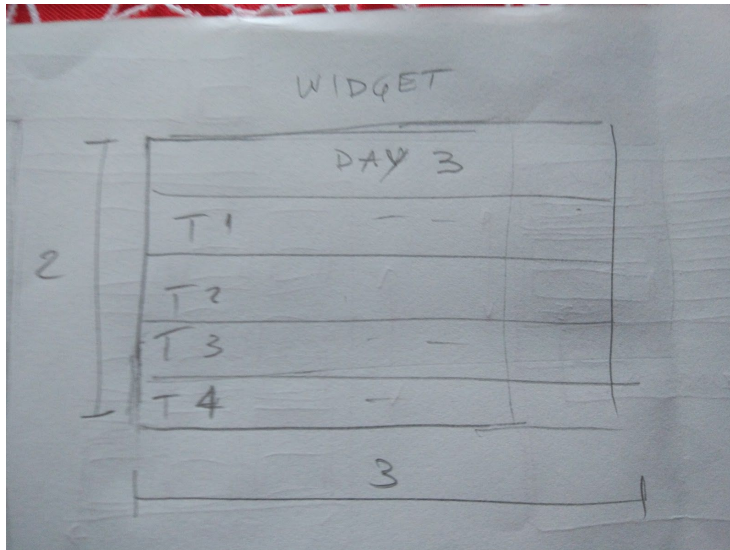
Screen 11 - Add Week/Day workout



Click tic fab in the bottom right corner to save

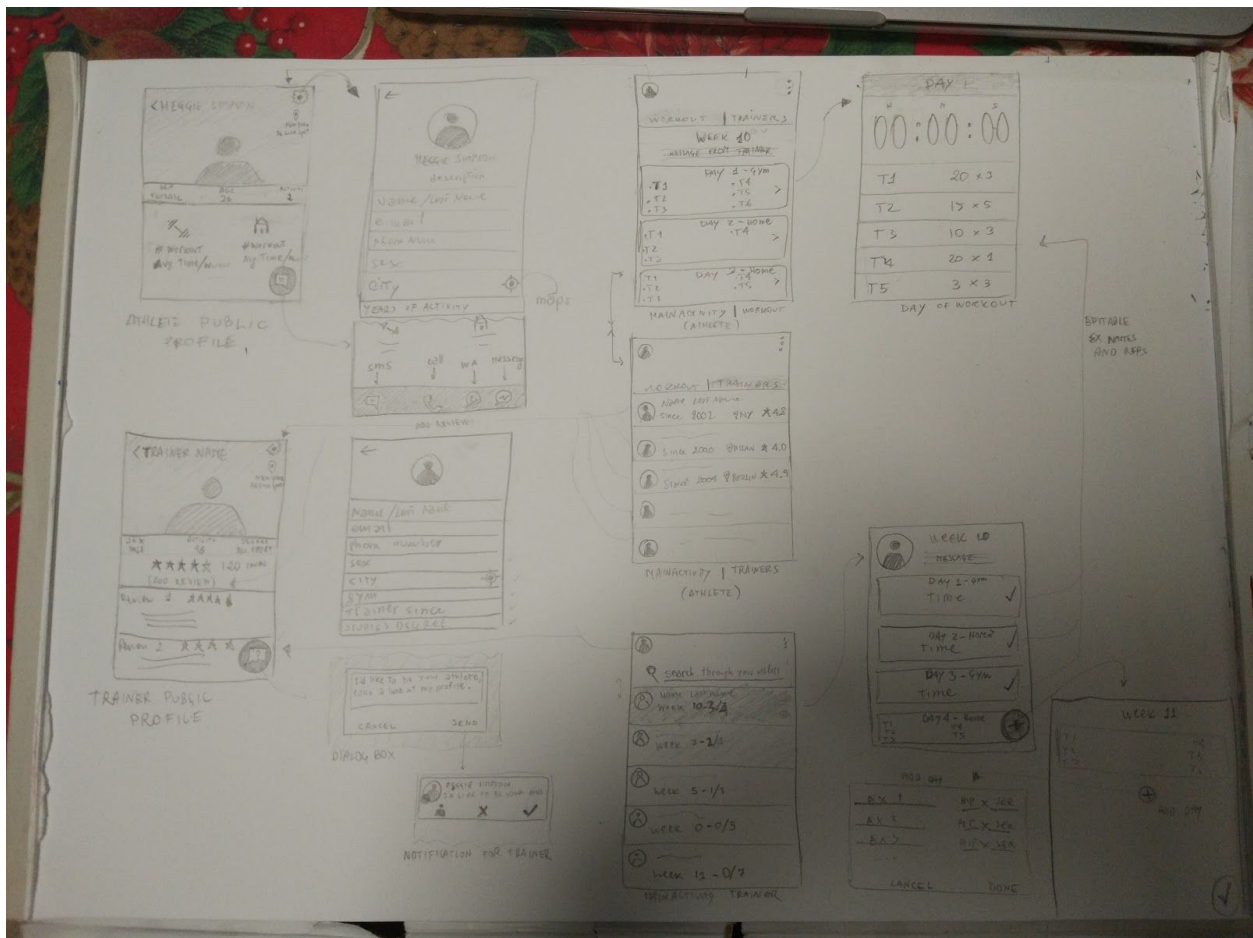
Click little add icon to add another day → opens a Dialog box to add single exercises

Widget



Tap to go to current day workout (Day 3 is just a placeholder for current day)

Complete Flow (No Login)



Key Considerations

How will your app handle data persistence?

User/Trainer personal data will be saved in **preferences**.

Workouts, Users and **trainers** data will be saved on **Firebase Realtime DB**

Users and **Trainers** photos will be stored in **Firebase Storage**

Workout Data saved in a local SQLite DB for offline access

Describe any corner cases in the UX.

Can't see corner cases (Maybe I didn't well understand what you mean)

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

Glide for Loading, Caching and retrieving images. Easy to use, fast and efficient.

Firebase Storage, to save users and trainers photos.

Firebase Realtime Database, to save trainers and workouts data.

RecyclerView, CardView and Design libraries to implement Material design and make the app unique

AppCompat for retrocompatibility

EasyPermissions to implement marshmallow permissions easily.

Facebook sdk for login

Guava to easily create keys to better organize firebase database

Describe how you will implement Google Play Services.

Google play service for:

Locations to compare trainers and athletes position. Trainers and athletes will see the city where each other is located.

Authentication in conjunction with Firebase Auth.

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

- Configure libraries
- Get all drawables needed
- Design screen protectors (gradients, etc.), shapes, background
- Write SQLite Contract

Task 2: Implement UI for Each Activity, Fragment and list item

- Build UI for AthleteMainActivity
- Build UI for AthleteWeekWorkoutFragment
- Build UI for TrainersListFragment
- Build UI for ActiveWorkoutActivity
- Build UI for SubscribedAthletesListActivity
- Build UI for NewWeekWorkoutActivity
- Build UI for TrainerMainActivity
- Build UI for AthleteProfileActivity
- Build UI for TrainerProfileActivity
- Build UI for Athlete/TrainerEditDataActivity
- Build UI for Dialog Boxes, Notification and list items
- Build UI for LoginActivity

Task 3: Write API for Firebase and local DB Interface

- Getters and setters for user/trainer data
- Getter and setters for workout local data
- Queries Algorithms
- Login communication with Firebase: Facebook/Google

Task 4: Write Activities and Fragments logic

- Bind views from xml
- Connect activities to fragments
- Write Callbacks for activity/fragment communication
- Build the rest of the logic

Task 5: Handle Error Cases

- No connection
- No data available
- No permissions, etc.

Task 6: Animations

- Shared elements and screen transitions
- Activities and fragments internal animations (Fabs motions, color changes, etc)

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**"