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: prateekcode

GymTrainee

Description

GymTrainee is workout app that keeps you filtered Youtube workout videos. It helps you as a trainer. The app contains a workout for Home as well as Gym. So you don't need to ask for someone's assistant. One needs a solution.
The app have all types of workout like Full Body Workout, Abs Workout, Lower Body Workout, Biceps & Triceps workout, Back workout, and all other necessary ones. You just have to watch videos from the section and you're done. You can save your workout too for later use.
When you're done just click on the check button which keeps you motivated while doing the workout.

Intended User

This app for age group more than age of 10. This app for those who are interested in fitness and workout.

Features

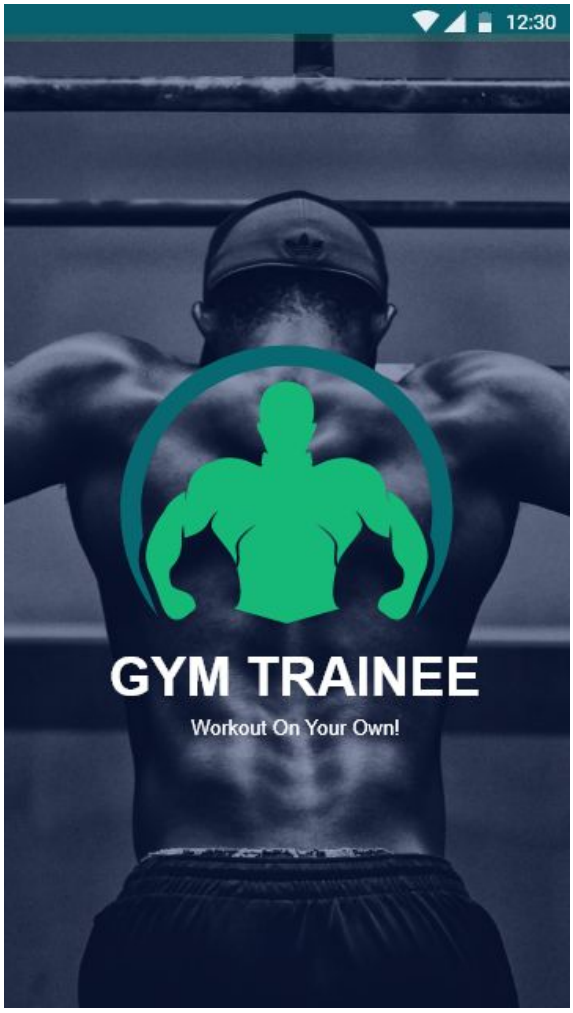
The main features are:

- Workout videos
- Save favorite workout videos
- Search workout
- Keep motivated with videos

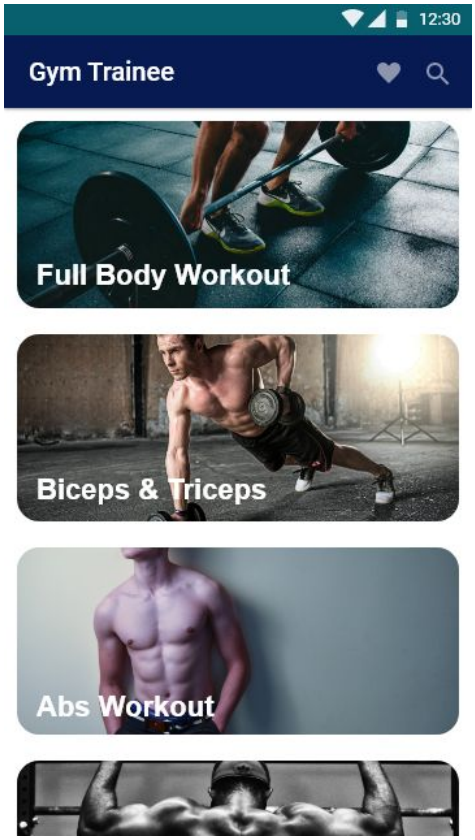
User Interface Mocks

The Mockup of the app created in Adobe Xd tools.

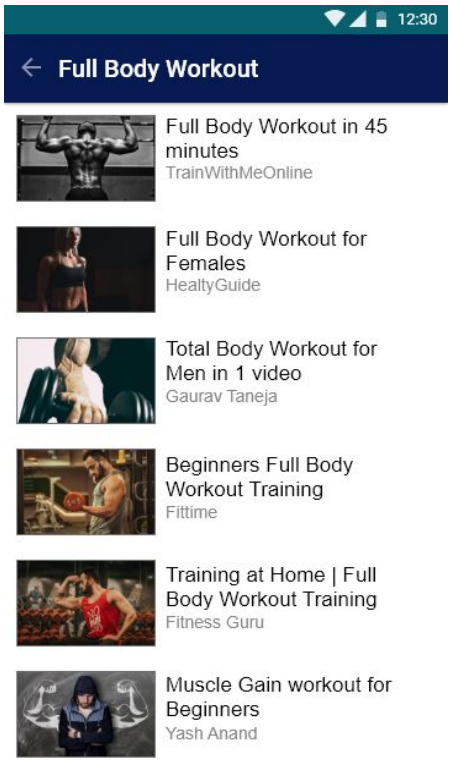
Splash Screen



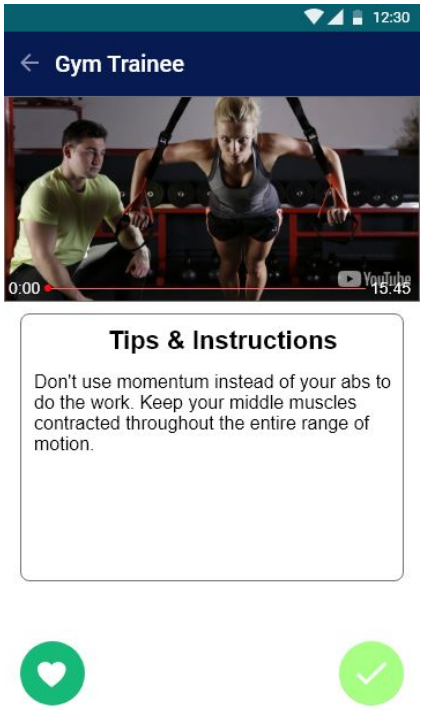
Home Screen



Workout Screen



Player Screen



Home Screen Widget

Key Considerations

1. Using Android Studio with all upgraded Gradle files and dependencies
2. Written in Java and XML

How will your app handle data persistence?

For Data Persistence, I'm using Room, Live Data, and View Model to save and retrieve favorite workout videos.

Describe any edge or corner cases in the UX.

This app uses a Youtube player, for this I'm using a third-party library to play video inside the app. App has simple elegant and simple UI.

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

I'm using a couple of the library in my Project.

1. Glide - For loading images
2. Room - To store data
3. Retrofit - For API calls
4. RxJava - For API calls as well
5. YouTube player view - For Playing Videos
6. ButterKnife - For View Bindings

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

I'm using AdView in my apps that use Google Play Service and Google Analyst

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

Using Android Studio to create a new project and first of all, make XML files and implement other Gradles files. Along with that I added logic for my every code. Start my project using XML and Java.

Task 2: Implement UI for Each Activity and Fragment

- Implement UI for MainScreen (i.e. MainActivity)
    1. Use RecyclerView to load workout
    2. Made all other layout using XML.

- Implement UI for Workout Playlist Activity:
    1. Use YouTube API to retrieve videos from YouTube
    2. Make the required layouts

- Implement UI for PlayerActivity:
    1. Use YouTube Player view to play videos using video id
    2. Add a CardView to show instructions and tips
    3. Add two FAB, one for Done and another one to Mark Favorite

Task 3: Create Room Database

In this step, I will create Room Database to store favorite workout videos and stored at one place which can be easily accessible from MainActivity.

- Make a Database class that extends RoomDatabase
- Make the Entity for the room database in a class and annotate it with @Entity.
- Make Dao for the database and annotate with @Dao. It provides methods annotation that leads to save, delete, update and retrieve data.
- Just Init your Activity with this database and you are ready for arranging your data in the room database

Task 4: Network Calls

- Making API calls like getting video using youtube API with the help of retrofit and RxAndroid
- Make an interface that handles API calls
- Implement interface with retrofit
- Get result from YouTube API using RxAndroid
- Successfully Make a list and load to recycler View using an adapter

Task 5: Home Screen Widget

Home screen widget for the user that can easily find favorite workout videos.

- Make XML for Home Screen widget
- Add layout to Home Screen widget
- Making a class that extends AppWidgetProvider that provided to create and update and delete widget on the home screen
- Defining receiver & meta data for home screen widget in Mainfest file

Task 6: Debugging and Testing

Debugging:

- Use debugging Breakpoints to check java code lines in android studio
- Debug and start building gradle with all files
- Install apk and it's time to checking all breakpoints

Testing:

- Test apk using android studio logcat
- Test apk on different devices using firebase test lab

Task 7: Important points will be considered

1) Specify that App keeps all strings in a strings.xml file and enables RTL layout switching on all layouts.

2) Must implement at least one of the three

If it regularly pulls or sends data to/from a web service or API, app updates data in its cache at regular intervals using a SyncAdapter or JobDispacter.

OR

If it needs to pull or send data to/from a web service or API only once, or on a per request basis (such as a search application), the app uses an IntentService to do so.

OR

It performs short duration, on-demand requests(such as search), the app uses an AsyncTask.

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**"