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: harshadpune

GetFit

Description

GetFit will help users to remain fit. The app can be used at home and do not require any equipment. This app will feature a daily motivational video which will keep its users motivated for workout. If user likes the video they can bookmark it and watch it later.

Intended User

Intended for everyone who want to get fit or remain fit.

Features

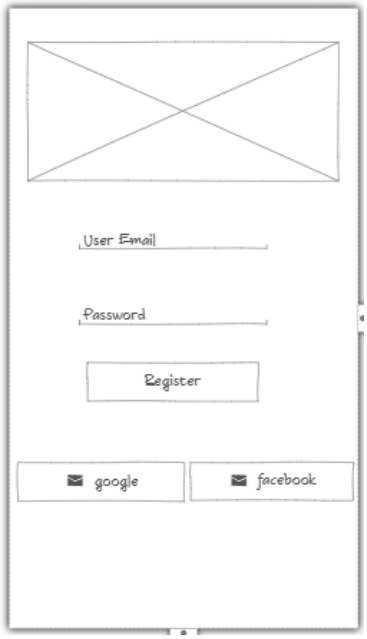
Features for this app:

- Authenticates user using firebase either via email, google or facebook
- Shows daily motivational video to the users
- Motivational video will be played using Youtube API within the app.
- Bookmarks the motivational videos as per user request
- Saves the fitness instructions locally using Room which can be used offline
- Saves the user workout information to Firebase database
- Shows information report of user workout fetched from Firebase database
- Shows the timer and elapsed time in a workout

User Interface Mocks

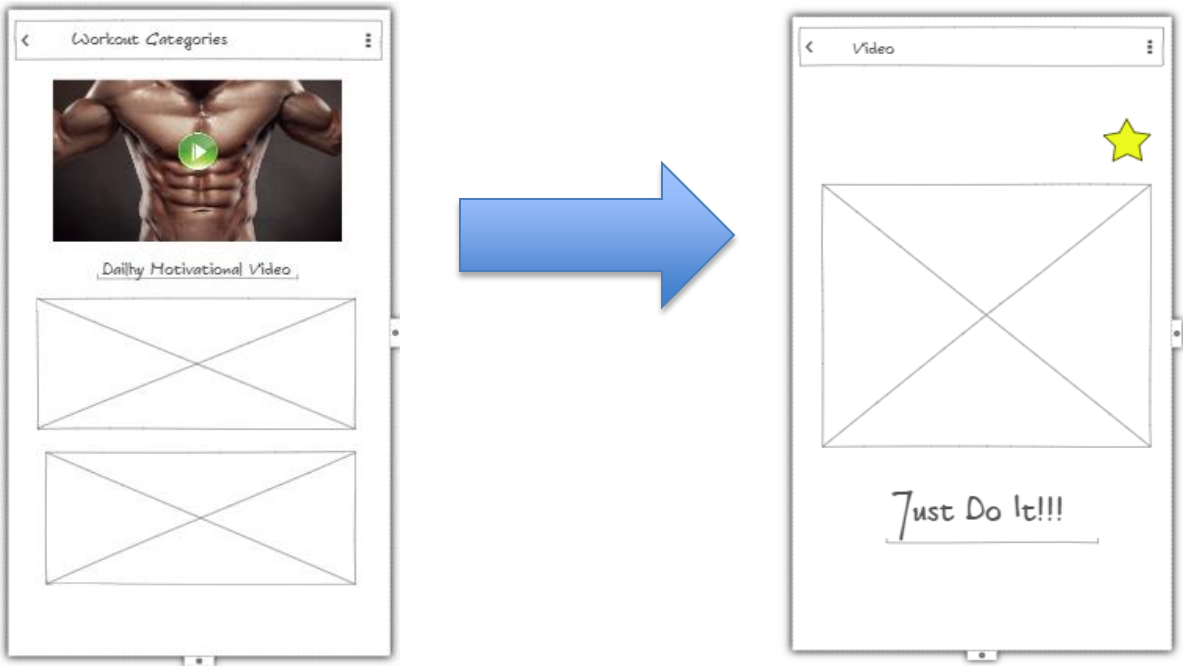
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1 : Login Screen



This login screen will let your login with their email id password or directly via google or facebook account. It will help user tack their workout details.

Screen 2 : Workout Categories

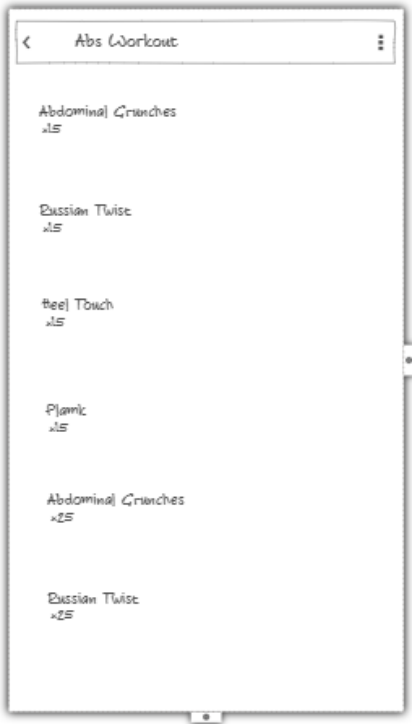


User will see this once they are logged in. It will show a Daily motivational video at the top. It will show the CardView list of categories of workouts (e.g Abs, Chest, Legs). Clicking on motivational video will play the video within app and allow user to bookmark the video. Clicking on workout category will take user to workout list (Refer Screen4).

Screen 3 : Workout Menu

Workout screen has 3 menu options – Favorite videos, Report and Logout

Screen 4 : Workout List

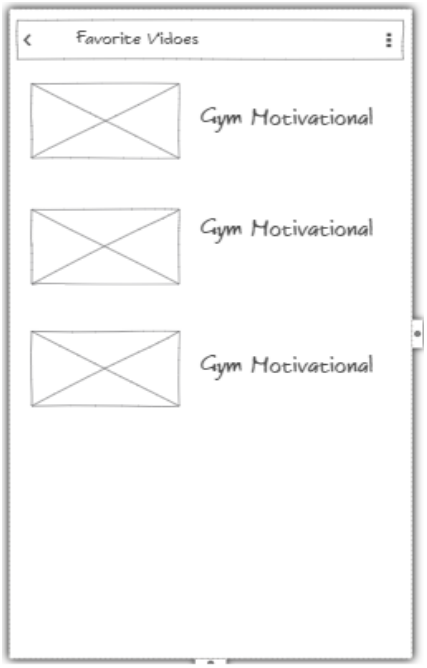


This screen will show list of workouts in a selected workout category. E.g List of Abs workout Plank, Heel touch etc. It will also show the number of repetitions to be carried out.

Screen 5 : Workout Start/Stop

User can start the workout. Timer will run to log the elapsed time in workout. Once workout is finished they can mark it as done. Workout will be logged in Firebase Realtime database once it is finished. User can navigate to next/pervious workout from this screen.

Screen 6 : Favorite Videos



User can save daily motivational videos and can see them in this Menu -> Favorite Videos section. The Video information will be stored locally using Room database. Video will be played within app using Youtube API.

Screen 7 : Workout Report

User can access their report in Menu -> Workout Report. This will show user how much time they have spent in a workout category and the date.

Screen 8 : Widget

Widget (3x2) will be provided by the app which will inform user that they have not done todays workout. If workout is done then it will show the report of today.

Key Considerations

How will your app handle data persistence?

App will use Firebase real-time database to save the details of user workout. App will use Room database to store the workout details locally once it is received from mock server.

Describe any edge or corner cases in the UX.

When workout is started and user presses back button a dialog will be displayed to user for confirmation. No data will be stored in that case.

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

Picasso – To handle image loading and caching
Retrofit – Retrofit API to get data mock data from server.
Mockapi.io – Mock api to host the get the Worktout details in JSON format.
Room – To store the workout details locally for offline use.

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

Firebase Auth - To login and authenticate user using email, google and facebook
Firebase Realtime Database- To save workout details of user
Youtube API – To play motivational video within the app.

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

The app will use Firebase to authenticate user. App will get Workout details from Mockapi.io in JSON format. We will need to user Youtube api to play the motivational video within the app.

- Setup project dependencies in gradle
- Populate Mockapi JSON response
- Setup firebase for the project
- Setup Youtube api for the project

Task 2: Implement UI for Each Activity and Fragment

Implement the UI for following Activities and Fragment.

- Login Activity
- Workout Category Activity
- Workout Details Activity
- Workout Start/Stop Activity and Fragments
- Favorite Videos Activity
- Reports Activity
- Youtube Player Activity

Task 3: Implement Firebase Auth

Once Firebase is setup for the project, email authentication will be added using Firebase auth.

- Implement Firebase Auth for Email
- Implement Firebase Auth using Google
- Implement Firebase Auth using Facebook
- Navigations to next screen

Task 4: Implement Server Connection

Connect to Mockapi.io to get Fiteness Data

- Connect to Mockapi.io using Retrofit api
- Implement Room to store the fetched data locally.
- Populate the video and workout categories cards

Task 5: Implement workout category and List cards

Use Picssso to display the workout photos.

- Picasso implementation to download and cache the images
- Show timer on workout screen
- Implement workout fragments

Task 6: Implement Firebase Real-time Database

Implement Firebase Real-time and Room database to store workout details.

- Implement Firebase real-time database to store stats (Column userid, workout, duration, date)
- Implement Room database if required to store the data locally for offline usage.

Task 7: Implement Reports Screen

Implement reports screen

- Fetch data from Firebase Database and display the report.

Task 8: Implement Widget

Implement widget to show user the stats about today's workout. It will show a screen informing user that they need to do workout if no workout is done.,

- Implement widget
- Make necessary db connection to show workout report data.

Task 9: Implement Logout

Implement logout functionality

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**"