

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**
-

Description

Intended User

Features

User Interface Mocks

Screen 1 - Login

Screen 2 – Register

Screen 3 – MainActivity Fitness Gods

Screen 4 – Exercises List

Screen 5 – Detailed Exercise Description

Screen 6 – Personal Workouts

Screen 7 – Custom Exercises

Screen 8 – About Us

Screen 9 – Widget

Key Considerations

In what language the app will be developed?

Libraries, Gradle and versions specifications.

How will your app handle data persistence?

Describe any edge or 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 or other external services.

Additional information

Next Steps: Required Tasks

Task 1: Project Setup

Task 2: Implement UI for Each Activity and Fragment

Task 3: FitnessGodsDBHelper Implementation

Task 4: LoginActivity Implementation

Task 5: RegistrationActivity Implementation

Task 6: MainActivity Implementation

Task 7: WorkoutsFragment Implementation

Task 8: ExercisesFragment Implementation

Task 9: DetailedExerciseFragment Implementation

Task 10: CustomWorkoutsFragment Implementation

Task 11: CustomExercisesFragment Implementation

Task 12: AboutUsFragment Implementation

Task 13: Widget Implementation

GitHub Username: <https://github.com/theoddprince>

FITNESS GODS

Description

Fitness Gods app is a personal trainer which will provide you with information that will help you achieve all of your goals and without paying a lot of money and to replace expensive instructors. Fitness Gods app will be a very friendly for beginners and pros and you can always find a suitable workouts.

This app allows you to create your own personal program that will fit your needs. The app is an extensive database of exercises for every muscle with detailed description and videos for each exercise.

Intended User

Fitness Gods app will be used by men and women equally since the app will contain workouts for both genders needs.

Features

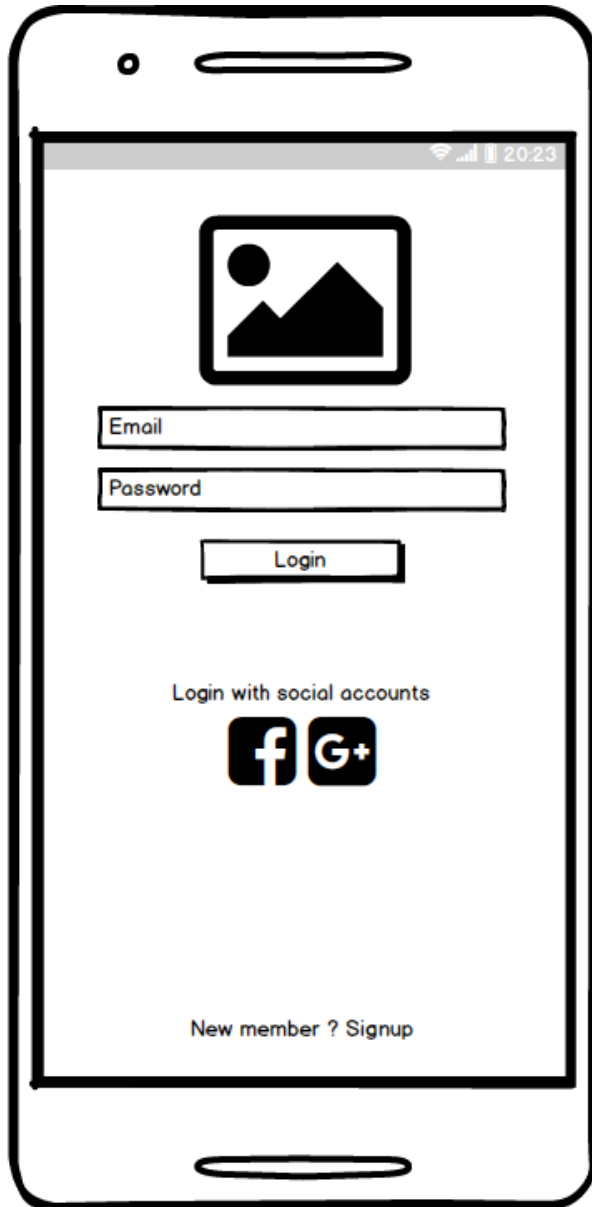
Main features of Fitness Gods app:

- Gym exercises guide with videos support for every workout.
- Text instruction with pictures.
- Various gym exercises database.
- A list of effective workouts for every muscle group.
- Exercises for home and gym.
- Allows you to add your own workouts with your custom exercise.

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



The First page shown to the user is the login page in the first time after installing the app. The page needs email and password inputs from the user in order to login into Fitness God app. At the bottom screen there is a Signup link which will direct the user into the Registration page in case he does not have any account ready yet. The user has the option to login using his social media accounts such as Facebook or Gmail+.

Screen 2 - Register

REGISTER

Email

Password

Confirm Password

Register

Already a member ? Login Here

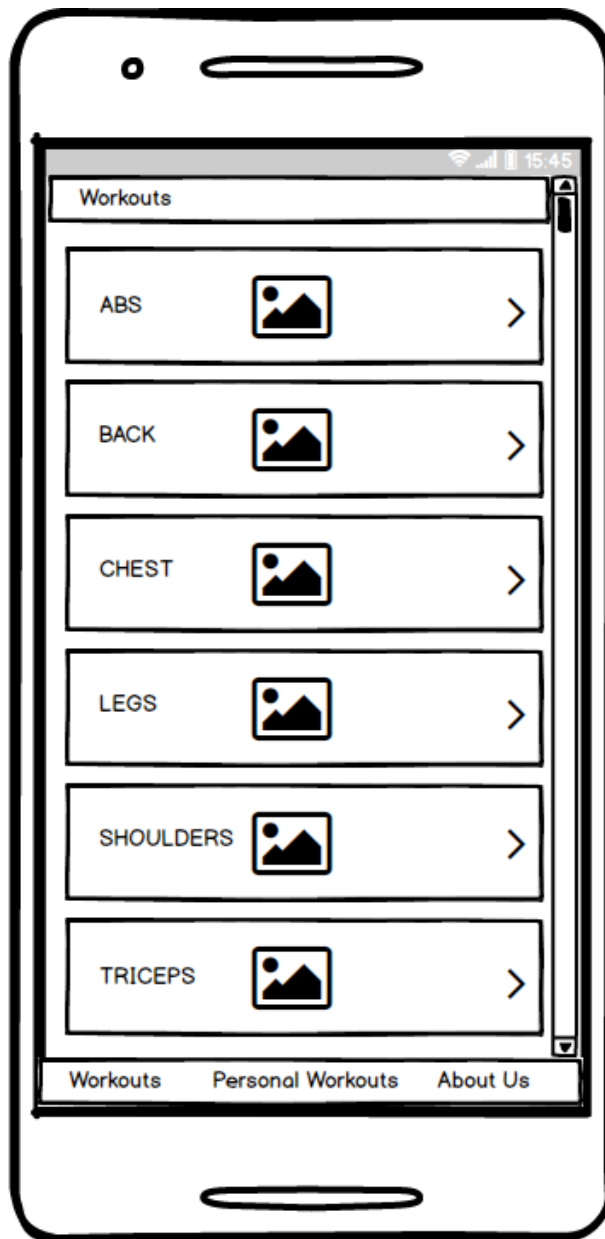
The registration page will be entered once the user clicks on the Signup link at the bottom screen of the Login page.

The user must provide a valid email and a password for the registration part.

In case the user has an account already he can click on the Login Here at the bottom screen of the layout.

After registration successful the user will be logged in instantly to the main activity page.

Screen 3 - MainActivity Fitness Gods



The main activity page will be directly entered after registration or login.

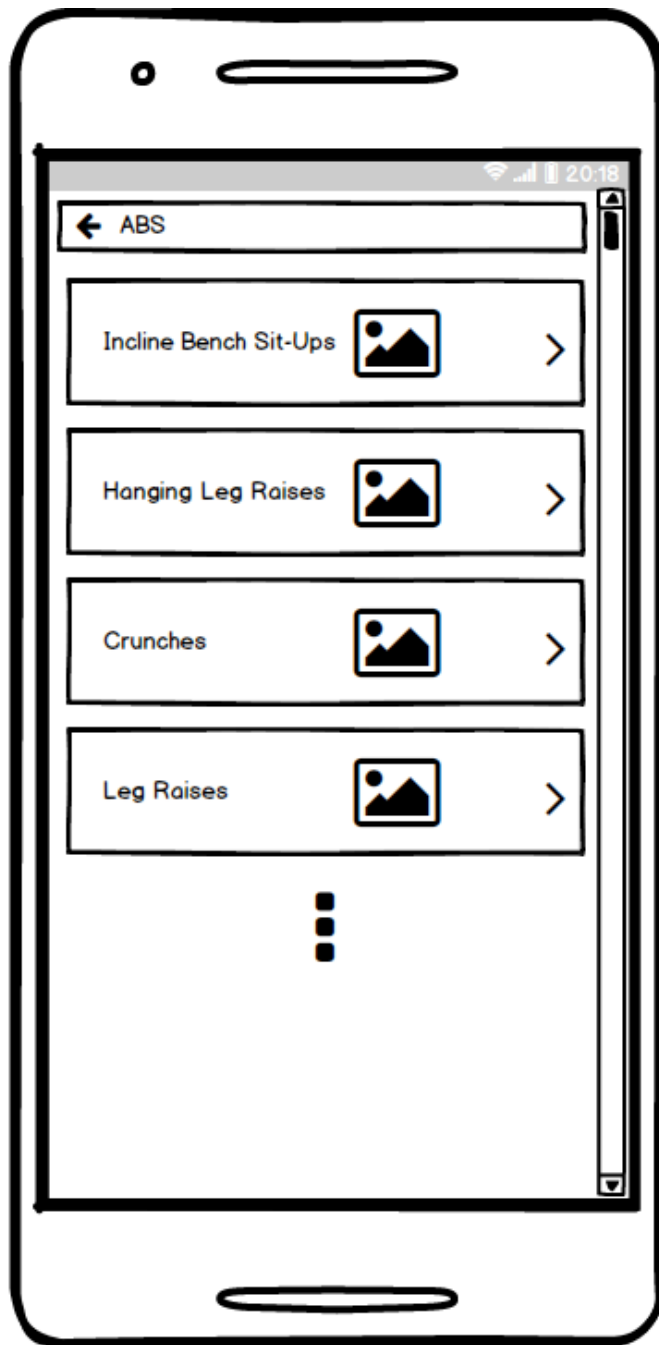
The page will contain a list of workouts fetched from the firebase database and viewed to the user. Each workout can be clicked and it will redirect the user to the desired workout sets.

At the bottom of the screen there is a menu, and it provides three fragment pages (Pager).

The first page is the default Workout page, the second page will contain a customized workouts which the user selects as his favorite. The third page will contain some information about the app and the developer.

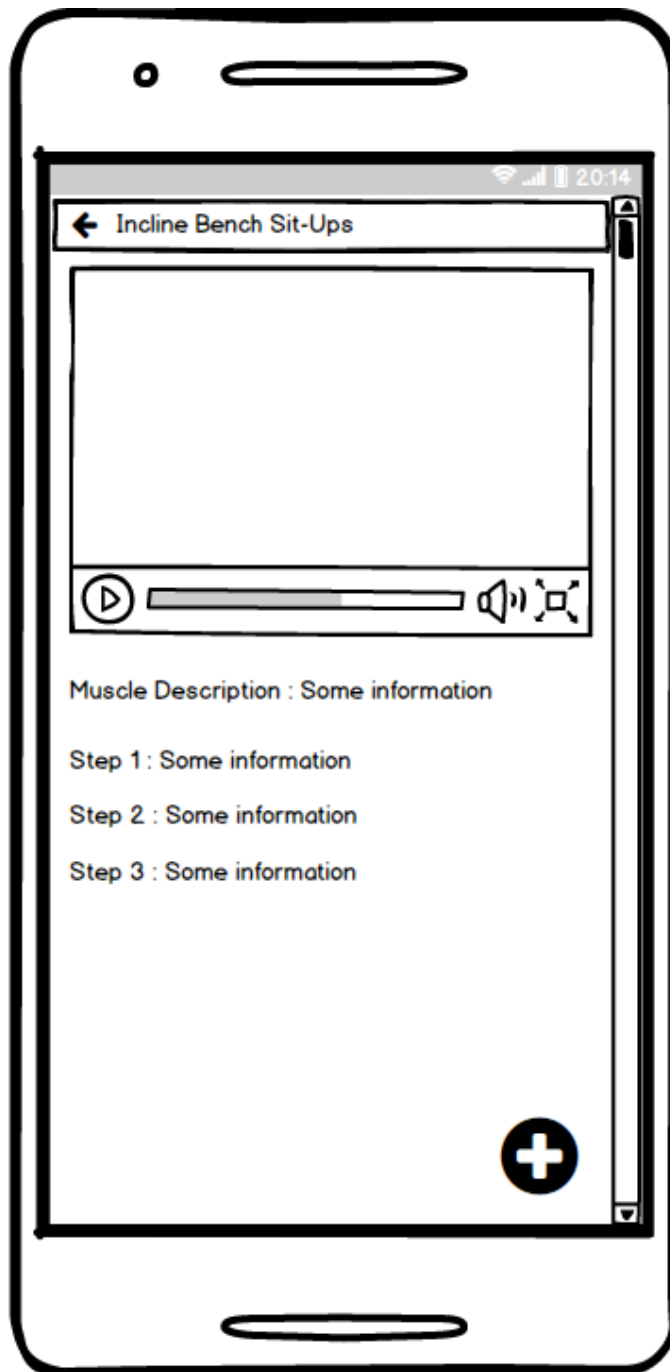
I will change the menu text to icons, as for the mockup I wrote text for explanation.

Screen 4 - Exercises List



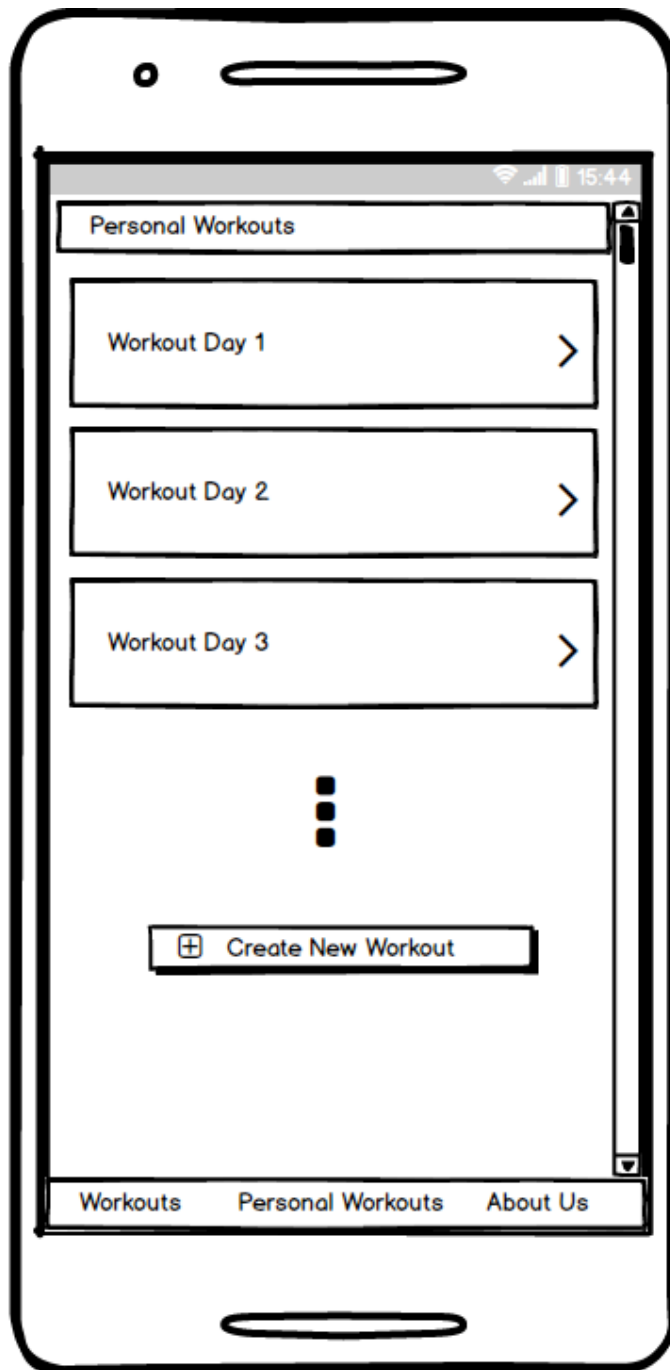
This page will contain an exercises list for each workout selected from the main page.

Screen 5 - Detailed Exercise Description



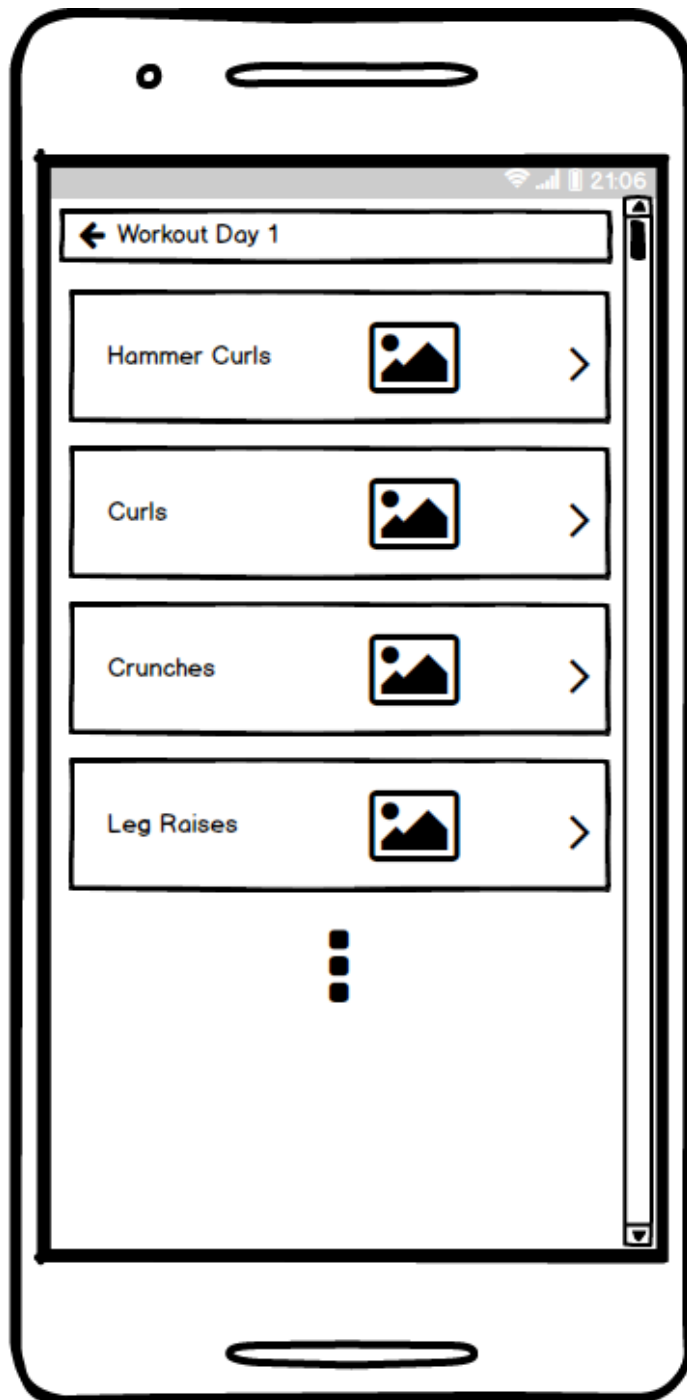
This page which will contain a detailed description of the desired exercise and steps for how to do. Demonstration video will be available and the user can add the exercise to his customized exercises page by clicking on the plus icon at the bottom left of the screen.

Screen 6 - Personal Workouts



Personal Workouts page will allow the user to create new workouts with any desired name in order to give him the ability to save his favorite exercises.

Screen 7 - Custom Exercises



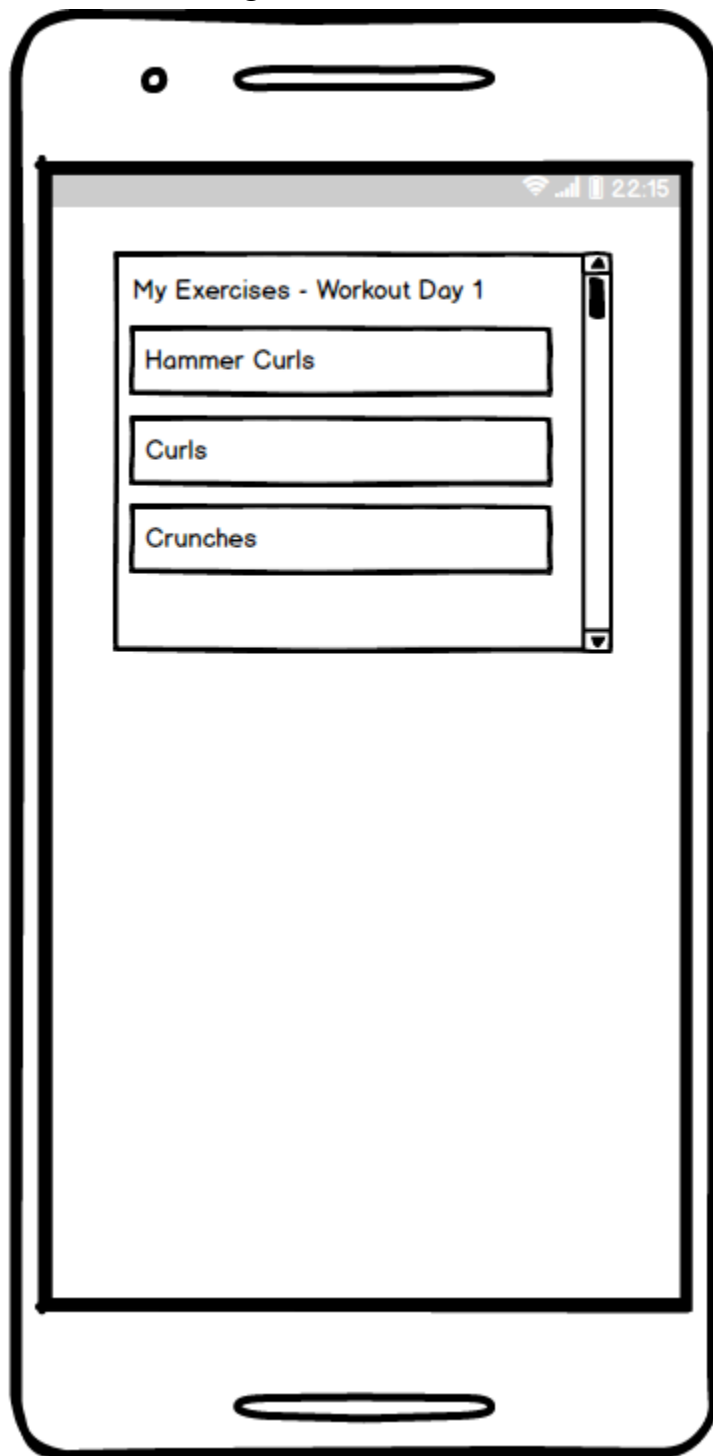
The Custom Exercise page will contain all of the favorite exercises for the user. Clicking on any exercise will direct the user to screen 5 (Detailed Exercise Description). There will be an option to the user to remove any exercise from the list.

Screen 8 - About Us



About us page will be available from the bottom main menu that will contain some information about the app.

Screen 9 - Widget



Widget which can be an option for the user to see his desired exercises of any workout he created on the home screen, he can click on any exercise and be redirected to the detailed page for the exercise.

Key Considerations

In what language the app will be developed?

The app will be developed in Java using android studio.

Libraries, Gradle and versions specifications.

| Name | Version |
|--|---------|
| Android Studio | 3.1.2 |
| com.android.tools.build:gradle | 3.1.2 |
| com.google.firebase:firebase-core | 16.0.1 |
| com.google.gms:google-services | 4.0.0 |
| com.squareup.picasso:picasso | 2.5.2 |
| com.android.support:cardview-v7 | 27.0.2 |
| com.android.support:recyclerview-v7 | 27.0.2 |
| com.google.android.exoplayer:exoplayer | r2.2.0 |

How will your app handle data persistence?

The app will use a Firebase Real-time Database in order to store and fetch all of the videos and the information needed that will be presented to the user inside the app.

The app will use a local SQLite DB in order to store the users' favorite exercises and his login information.

The app will use content provider.

Describe any edge or corner cases in the UX.

Since I will be using ExoPlayer in my app, I will face several edge cases:

- 1) When clicking at the home button while the video was running I will need to store the video state and position in case I get back to the app.
- 2) When clicking on the back button I will need to release the player in order to avoid crashes.
- 3) In case of screen rotation when video is running needs to be handled as well.
- 4) In case of using tablets I need to maintain some different layouts presentation for specific pages.

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

- 1) I will be using Picasso library in order to handle the loading and caching of images that I'm going to present in the application.

- 2) ExoPlayer: provides an alternative to Android's MediaPlayer API for playing audio and video both locally and over the Internet.

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

- 1) I will be using the Google play services in order to make Firebase work, the reason for that is because google services plugin reads the configuration information for the google services dot JSON file that has been added to the project.
- 2) Google play services will be used also for integrating google Sign-In into the Fitness Gods App.

Additional information

- 1) The app will keep all strings in a strings.xml file and enables RTL layout switching on all layouts.
- 2) Content Provide will be implemented in order to store the data locally.
- 3) Since I'm going to use content provide in the app I will use Loader to move its data to its views.
- 4) The app will be using IntentService since the data fetching will be done once or on a per request basis.

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

- Creating a new project at Android Studio for Fitness Gods app.
- Creating a new project at firebase for Fitness Gods app.
- Configuring the firebase project (Creating tables, inserting data).
- Configuring libraries (Picasso, ExoPlayer, Google Services ...).

Task 2: Implement UI for Each Activity and Fragment

- Build UI for LoginActivity
- Build UI for RegistrationActivity
- Build UI for MainActivity
- Build UI for WorkoutsFragment
- Build UI for ExercisesFragment
- Build UI for CustomWorkoutsFragment

- Build UI for CustomExercisesFragment
- Build UI for DetailedExerciseFragment
- Build UI for AboutUsFragment

Task 3: **FitnessGodsDBHelper Implementation**

- Implementing SQLite DB.
- Create table for user information.
- Create table for workouts information.
- Create table for the user favorite exercises.

Task 4: **LoginActivity Implementation**

- Create layout for the LoginActivity.
- Implementing login and connect it with firebase.
- Implementing Google Play Service for the login from Google+.
- Implementing the Signup button which will redirect the user into the registration screen.

Task 5: **RegistrationActivity Implementation**

- Create layout for the RegistrationActivity.
- Implementing the register button to fetch data inserted by the user and connect it with firebase.
- Implementing the Login here button which will redirect the user into the Login screen.

Task 6: **MainActivity Implementation**

- Create layout for the MainActivity.
- Implementing bottom menu which will contain: (Workouts, Exercises and About Us).
- Implementing logout button at the app upper bar.
- Implementing Pager to work with the menu for better user experience divided into three fragments (WorkoutFragment, ExercisesFragment, AboutUsFragment).

Task 7: **WorkoutsFragment Implementation**

- Create layout for WorkoutsFragment.
- Implementing RecyclerView to present all of workouts fetched from the firebase DB.
- Implementing data fetching function for the workouts and store them into the workouts DB.
- Implementing WorkoutsAdapter which extends RecyclerView.Adapter.
- Implementing OnWorkoutClickListener interface.

Task 8: ExercisesFragment Implementation

- Create layout for ExercisesFragment.
- Implementing RecyclerView to present exercises specified for each workout.
- Implementing ExercisesAdapter which extends RecyclerView.Adapter.
- Implementing OnExerciseClickListener interface.

Task 9: DetailedExerciseFragment Implementation

- Create layout for DetailedExerciseFragment.
- Implementing ExoPlayer (Initialization, release).
- Implementing onDestroy, onResume, onPause, onSaveInstanceState.
- Implementing data loading for a specific Workout (Steps and Video).
- Implementing a button which will add the exercise to the desired custom workout as a favorite.

Task 10: CustomWorkoutsFragment Implementation

- Create layout for CustomWorkoutsFragment.
- Implementing RecyclerView for the presentation of the created workout.
- Implementing CustomWorkoutsAdapter which extends RecyclerView.Adapter.
- Implementing OnCustomWorkoutClickListener interface.
- Implementing a new button for creating new workout.
- Implementing Workout remove.

Task 11: CustomExercisesFragment Implementation

- Create layout for CustomExercisesFragment.
- Implementing RecyclerView for the presentation of all the favorite exercises.
- Implementing CustomExercisesAdapter which extends RecyclerView.Adapter.
- Implementing OnCustomExerciseClickListener interface.
- Implementing exercise remove.

Task 12: AboutUsFragment Implementation

- Create layout for AboutUsFragment
- Implementing some information about the app (Developer name, email, etc...)

Task 13: **Widget Implementation**

- Implementing widget which will contain the list of the created custom exercises.

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**"