GitHub Username: karejn

Fit4u

Description

Want to get fit but don't know where to begin?
Fit4u is an app that provides you with a range of workout routines based on your own fitness level and targeting specific areas of the body. Most routines have pictures to guide you and a digital personal trainer (videos) to go with you through it.
You can create your own personal program that will fit your needs.

Intended User

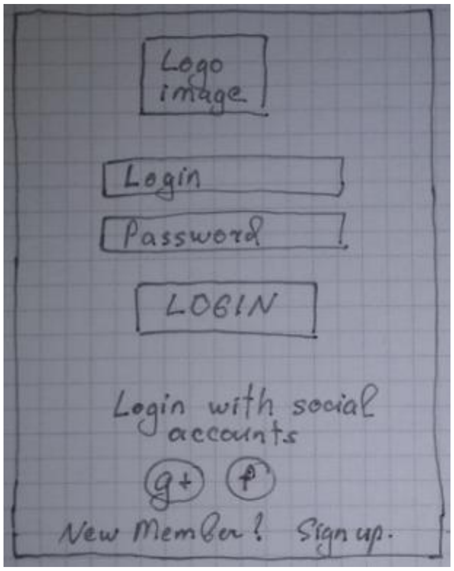
Men and women: every person who wants to get fit.

Features

- Exercises database: for home and gym.
- Exercises with videos support.
- Written instruction with detailed pictures and videos.
- A list of effective workouts for every muscle group.
- A possibility to create your own sets with your favorite exercises.

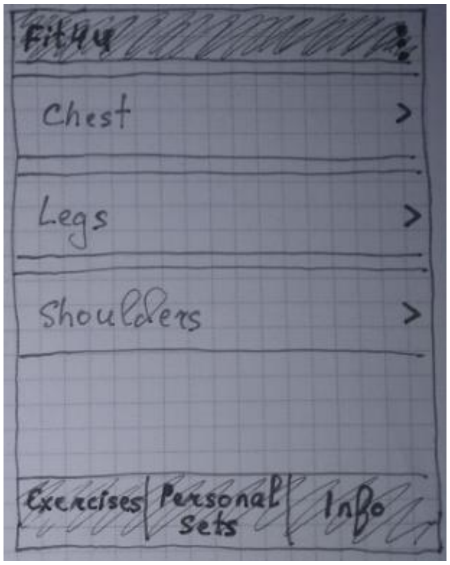
User Interface Mocks

Screen 1 - Login



The login page that the user sees the first time after installing the app.
At the bottom of the page there is a link to registration form where the user sees almost the same screen with Login and Password fields.

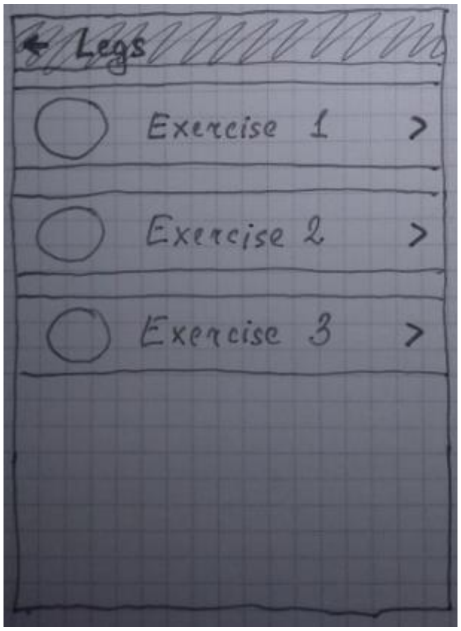
Screen 2 – Main Activity



A list of exercise sets. Each set could be clicked, and it will redirect user to the activity shown on Screen 3.

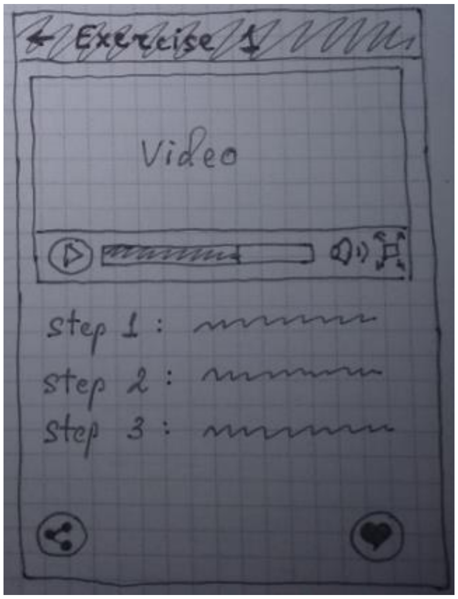
At the bottom of the screen there is a menu, and it provides three fragment pages.

Screen 3 – A list of exercises



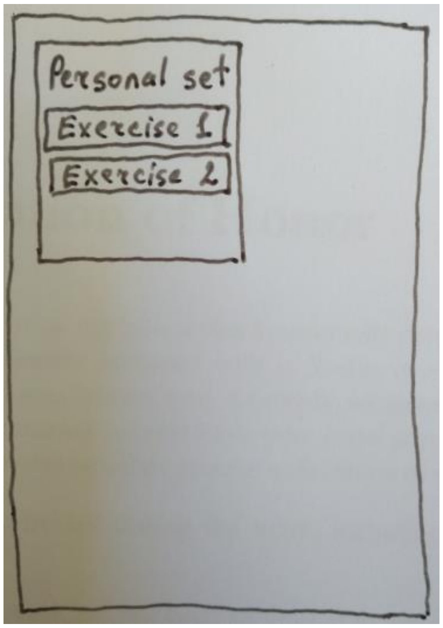
An exercises list for each set selected from the activity with sets.

Screen 4 – Detailed exercise description



A detailed description for each exercise. Instead of video for some exercises will be provided a photo or (if there is no video, nor photo) a typical placeholder with app logo will be shown.
At the bottom of the screen there is a share button and a button to add an exercise to a personal set.

Screen 5 – Widget



Widget provides an option for the user to see his favorite exercises, the user can click on any exercise and be redirected to the detailed description for the exercise.

Key Considerations

How will your app handle data persistence?

The app will use a Firebase Real-time Database to store and fetch all the videos and the information about exercises.
The app will use a local SQLite DB to store the users' favorite exercises and login information.
The app will use a content provider.

Describe any edge or corner cases in the UX.

The main corner cases relate to playing videos option:
- In case of screen rotation when video is running the video should be handled properly.
- When clicking on the back button the player should be released to avoid crashes.
- When clicking at the home button while during the video playing the video state should be stored in case the user gets back to the app.

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

- ExoPlayer - it supports features not currently supported by Android's MediaPlayer API, including DASH and SmoothStreaming adaptive playbacks. Unlike the MediaPlayer API, ExoPlayer is easy to customize and extend, and can be updated through Play Store application updates.
- Picasso - to handle the loading and caching of images.
- Butter Knife - Field and method binding for Android views.

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

- I will be using the Google play services to make Firebase work and for integrating Google sign-in.

Additional information

- App is written solely in the Java Programming Language.
- App keeps all strings in a strings.xml file and enables RTL layout switching on all layouts.
- The app is using IntentService since the data fetching will be done once or on a per request basis.
- App includes support for accessibility. That includes content descriptions and navigation using a D-pad.
- Libraries, Gradle and versions specifications.

Android Studio Version	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

Next Steps: Required Tasks

Task 1: Project Setup

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

Task 2: Implement UI for Each Activity and Fragment

- Build UI for LoginActivity
- Build UI for RegistrationActivity
- Build UI for MainActivity
- Build UI for SetsFragment
- Build UI for ExercisesFragment
- Build UI for CustomSetsFragment
- Build UI for CustomExercisesFragment
- Build UI for DetailedExerciseFragment
- Build UI for InfoFragment

Task 3: DBHelper Implementation

- Implementing SQLite DB.
- Create table for users’ information.
- Create table for exercise sets 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: (Exercise Sets, Exercises and Info).
- Implementing Logout button at the app upper bar.
- Implementing Pager to work with the menu for better user experience divided into three fragments (SetsFragment, ExercisesFragment, InfoFragment).

Task 7: SetsFragment Implementation

- Create layout for SetsFragment.
- Implementing RecyclerView to present all of sets fetched from the firebase DB.
- Implementing data fetching function for the sets and store them into the sets DB.
- Implementing SetsAdapter which extends RecyclerView.Adapter.
- Implementing OnSetClickListener interface.

Task 8: ExercisesFragment Implementation

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

Task 9: DetailedExerciseFragment Implementation

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

Task 10: CustomSetsFragment Implementation

- Create layout for CustomSetsFragment.

- Implementing RecyclerView for the presentation of the created workout.
- Implementing CustomSetsAdapter which extends RecyclerView.Adapter.
- Implementing OnCustomSetsClickListener interface.
- Implementing a new button for creating a new set.
- Implementing set remove.

Task 11: CustomExercisesFragment Implementation

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

Task 12: InfoFragment Implementation

- Create layout for InfoFragment
- Implementing some information about the app (license, contact email, etc.)

Task 13: Widget Implementation

- Create layout for widget
- Implementing widget which contains the list of the created custom exercises.

.