GitHub Username: dtsvetilov

YouFitness

Description

YouFitness is a fitness activity tracker app, that allows you to store your workouts. Create workouts, embed YouTube video and make your exercise plan. Keep track of every fitness activity and its details.

Intended User

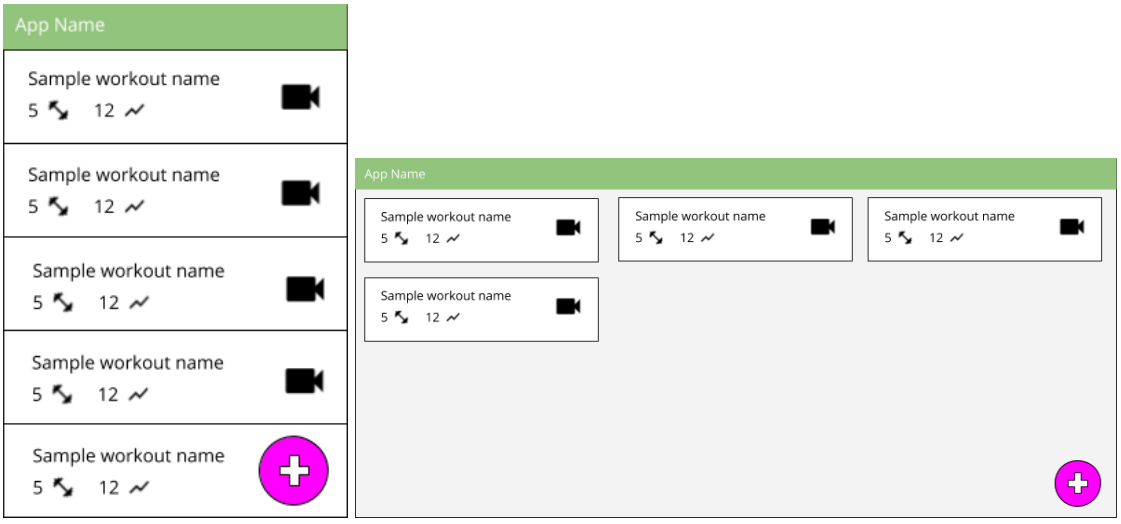
YouFitness can be used by every person who is working out in the gym. It doesn't matter the workout type, the app will keep track of it.

Features

- Data sync between devices
- Personal user account;
- Workouts list;
- Embed YouTube video for every workout;
- Plan exercise program for each workout;
- Keep track of time spent for each workout session.

User Interface Mocks

Workouts List

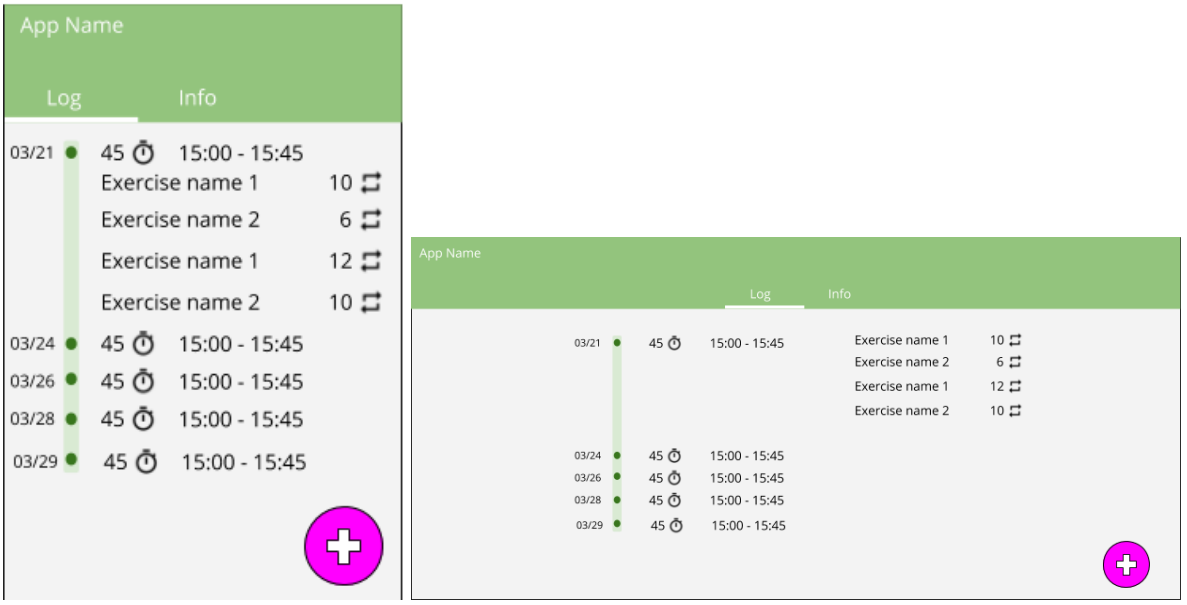


The workouts list will display collection of all user's workouts. On phone and portrait tablet the collection is displayed as list of tiles. On landscape tablet the collection is displayed as grid of cards. Evey workout item contains fields for:

- Workout name;
- Number of the exercises it is containing;
- Number of logs this workout was performed;
- Camera indicator if a YouTube video was embedded.

A floating Action Button will be placed on the bottom right corner of the screen. It is used for creating new workout.

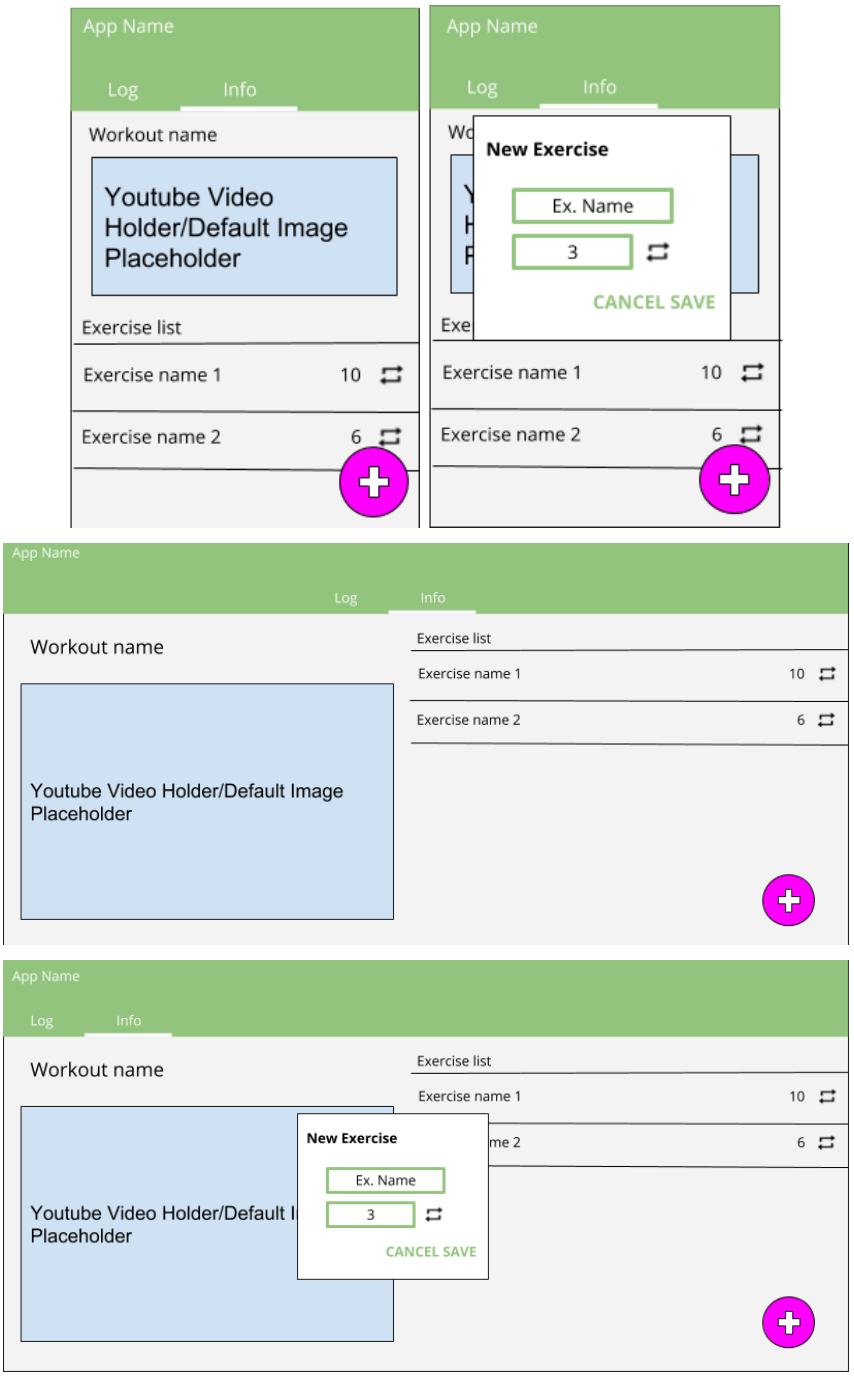
Workout Detail - Log Tab



The workout detail screen contains two tabs. The first is the workout log. It will show a log of the workout's activity. The log is designed as timeline. At the beginning each log item shows the date, the minutes duration and the time of the activity. By clicking on the item. It shows a detailed exercise information. This information contains the name and the number of repeats for every exercise.

A floating Action Button is placed on the bottom right corner of the screen. It is used for creating new workout activity.

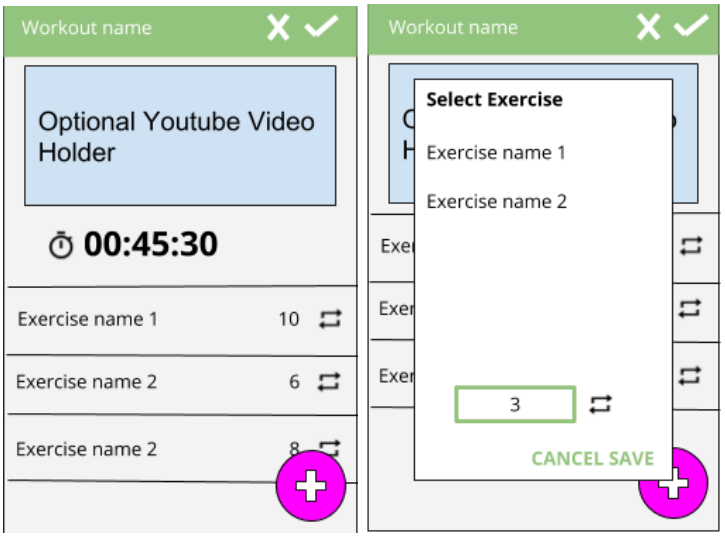
Workout Detail - Info Tab

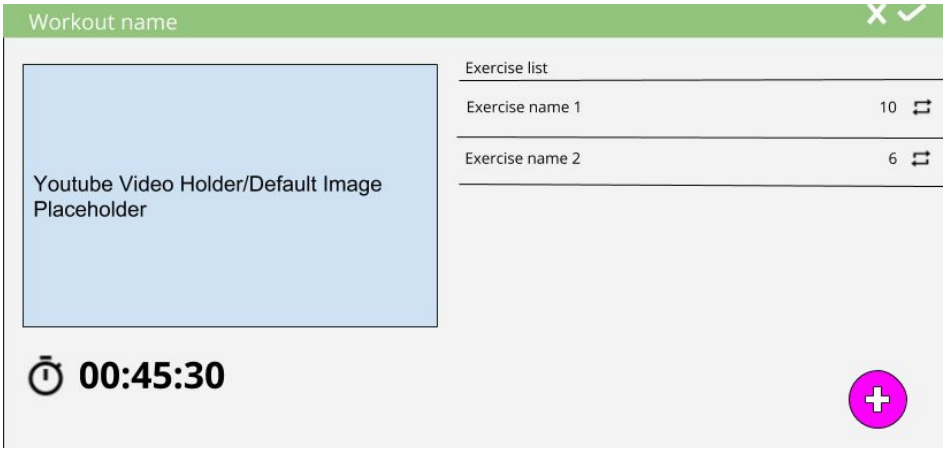


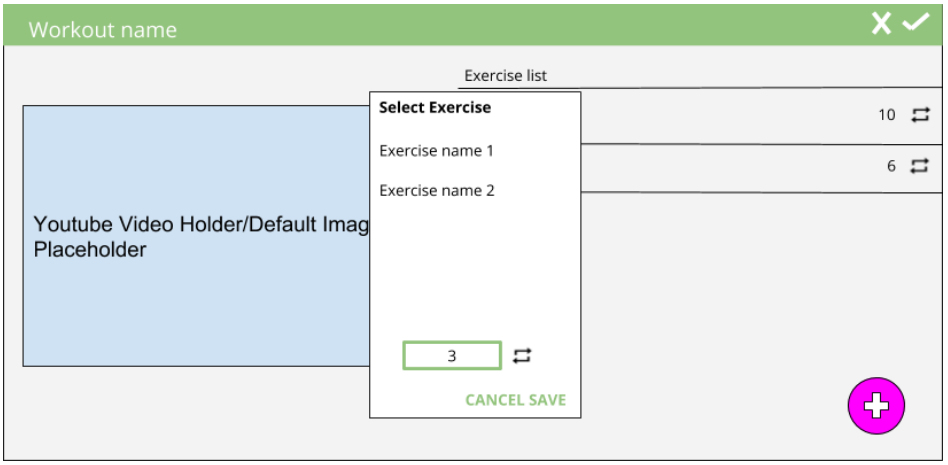
The workout detail screen contains two tabs. The second is the workout info. It shows a field for the workout name. The name could be edited by clicking on the name field. A dialog is displayed with a single text input field. User enters the workout name and clicks save. The newly entered value is displayed in the workout name field.
Below the name field is the YouTube video placeholder/player. If a video is embedded then the placeholder is displayed as a YouTube video player with embedded video preloaded. If YouTube video is not embedded then a default image placeholder will be displayed.
The screen displays a workout plan with a list of the exercises included in the plan. Every exercise information contains the name and the planned number of repeats.

A exercise can be edited or deleted by clicking on it. A dialog will be displayed with fields for editing the name and repeats of the exercise. The dialog will contain also a extra button for deleting the exercise.
A floating Action Button is placed on the bottom right corner of the screen. It is used for creating new plan exercise. A dialog will be displayed with fields for editing the name and repeats of the exercise.

Workout Activity

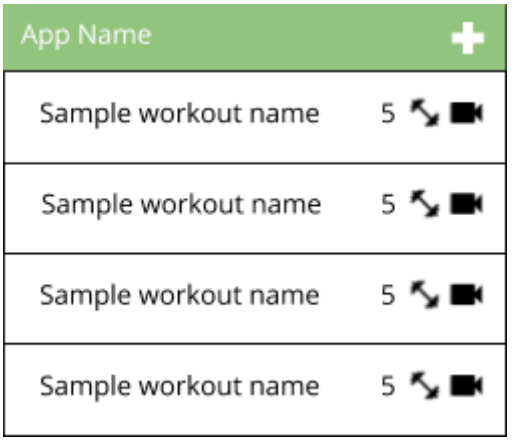






The workout activity screen contains a YouTube video (if embedded in the workout info), list of exercises completed in the activity and a timer showing the time passed since the activity start. Every exercise contains the name and the number of repeats.
A exercise can be edited or deleted by clicking on it. A dialog will be displayed with fields for editing the name and repeats of the exercise. The dialog will contain also a extra button for deleting the exercise.
Two action buttons will be placed on the toolbar. The first is for saving the activity, the second is for canceling the activity.
A floating Action Button is placed on the bottom right corner of the screen. It is used for creating new activity exercise. A dialog will be displayed with list of all available exercises for this workout and a text enter field for editing the repeats count.

App Widget



The app will provide 4x3 widget. The widget will be displayed on phone/tablet home screen. It will contain a list of available workouts. By clicking on workout item the user will be redirected to the app with a new workout activity started.
An action button will be placed on top right corner of the widget. It will function as a new workout button. Pressing it will redirect the user inside the app on the new workout screen.

Key Considerations

How will your app handle data persistence?

The app will use Firebase Realtime Database. User will be able to sign in with phone, email or Google account. All user data will be stored in the cloud.

Describe any edge or corner cases in the UX.

When the app initially starts a sign in/sign up screen is shown. After a successful authentication the user is redirected to the main screen of the app - the Workouts list. The user presses the create workout button ar selects a specific workout. The app then shows the Workout Detail screen. If a new workout is creating then the top right action buttons will be SAVE and CANCEL (also Workout Activity Log is empty and the new Workout Activity button is hidden), otherwise EDIT and DELETE. User can now add exercises to the Workout Plan or start a new Workout Activity. The app navigates to the Workout Activity screen. The Workout Activity can be saved or canceled.

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

YouTube Player API - embeds and plays the workout related YouTube video.
Firebase API - responsible for data persistence and user authentication.
Butter Knife - field and method binding for views

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

On startup the app will pull its initial data. The data will be pulled from the Firebase Rest API with a single request. The data will contain preloaded exercise names so the user can easily browse and choose exercise he prefers.
The app will use Google Firebase Service. It will handle data persistence and user authentication using its Android API. The service provides mechanisms for writing, reading and querying data stored on the cloud.
YouTube Player API will be used for playing embedded YouTube videos. The service provides player view which plays the video in a way the user is familiar. The player provides all the controls needed for managing the video playback.

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 project development will begin by adding dependencies to the libraries that will be used in the project (Firebase, YouTube Player and Butter Knife). The project configuration must be specified (Min SDK, Target SDK and etc.)

Task 2: Implement initial data pull

On app startup the initial data will be pulled via API call. The data is stored in Firebase Database. The pull request will be performed by executing an AsynTask that will call Firebase REST API. The result data will be stored in the app and will be used for further user input.

Task 3: Implement Activities and Fragments

- Authentication - implement Firebase Authentication UI;
- Workouts List - implement recyclerview for workout items and floating action button for new workout action;
- Workout Detail - implement tablayout and two fragment for each tab - Workout Activity Log and Workout Info
    - Workout Activity Log - implement timeline recyclerview with expansion feature;
    - Workout Info - implement YouTube Player API, recyclerview for workout plan exercises and new exercise dialog;
- Workout Activity - implement YouTube Player API, recyclerview for workout activity exercises and timer view.
- App Widget - implement widget functionality and UI.

Task 4: Implement Free and Paid flavors

Implement two different flavors for free and paid version of the app. The free version will contain ads and the paid one will be without ads.
Implement Ads throughout the app. A banner ad will be placed in Workout List and Workout Detail screens.