GitHub Username: btsochev

The Street Gym

Description

The Street Gym provides daily workout routines for all your main muscle groups. The app helps you to achieve impressive street workout skills and functional muscles.
In just a few minutes a day, you can build muscles and keep fitness at home or outside without having to go to the gym. No equipment or coach needed, all exercises can be performed with just your body weight.

Intended User

Fitness enthusiasts who like working out outside or people who can not or have no time to go to a gym.

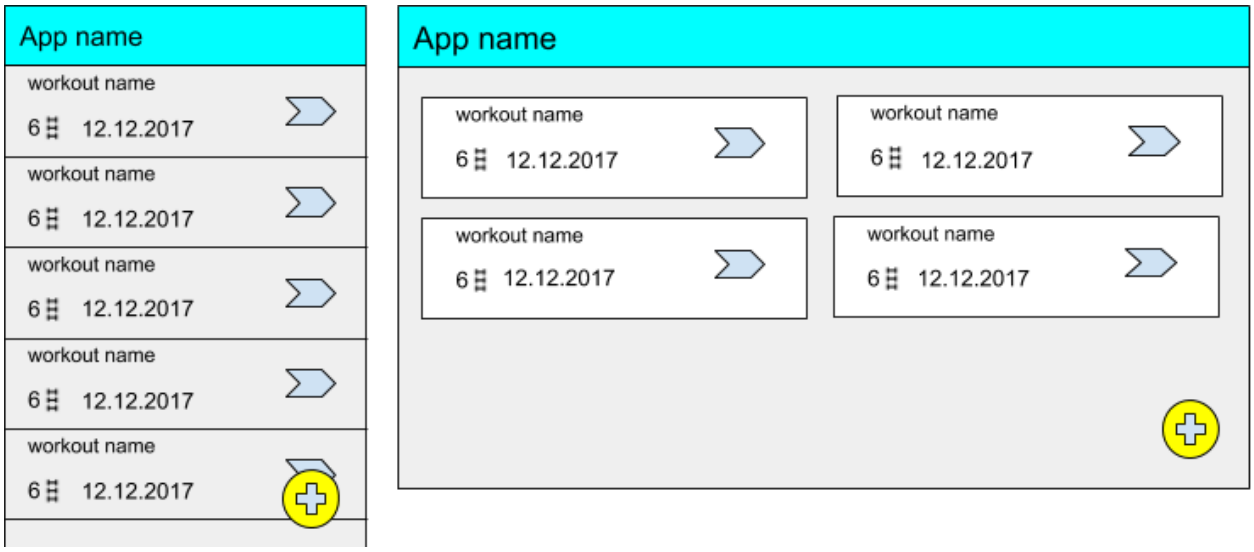
Features

- Smooth design allows for easy navigation

- Create your own exercices
- Create a daily routine
- Choose from already predefined routines
- Workout log

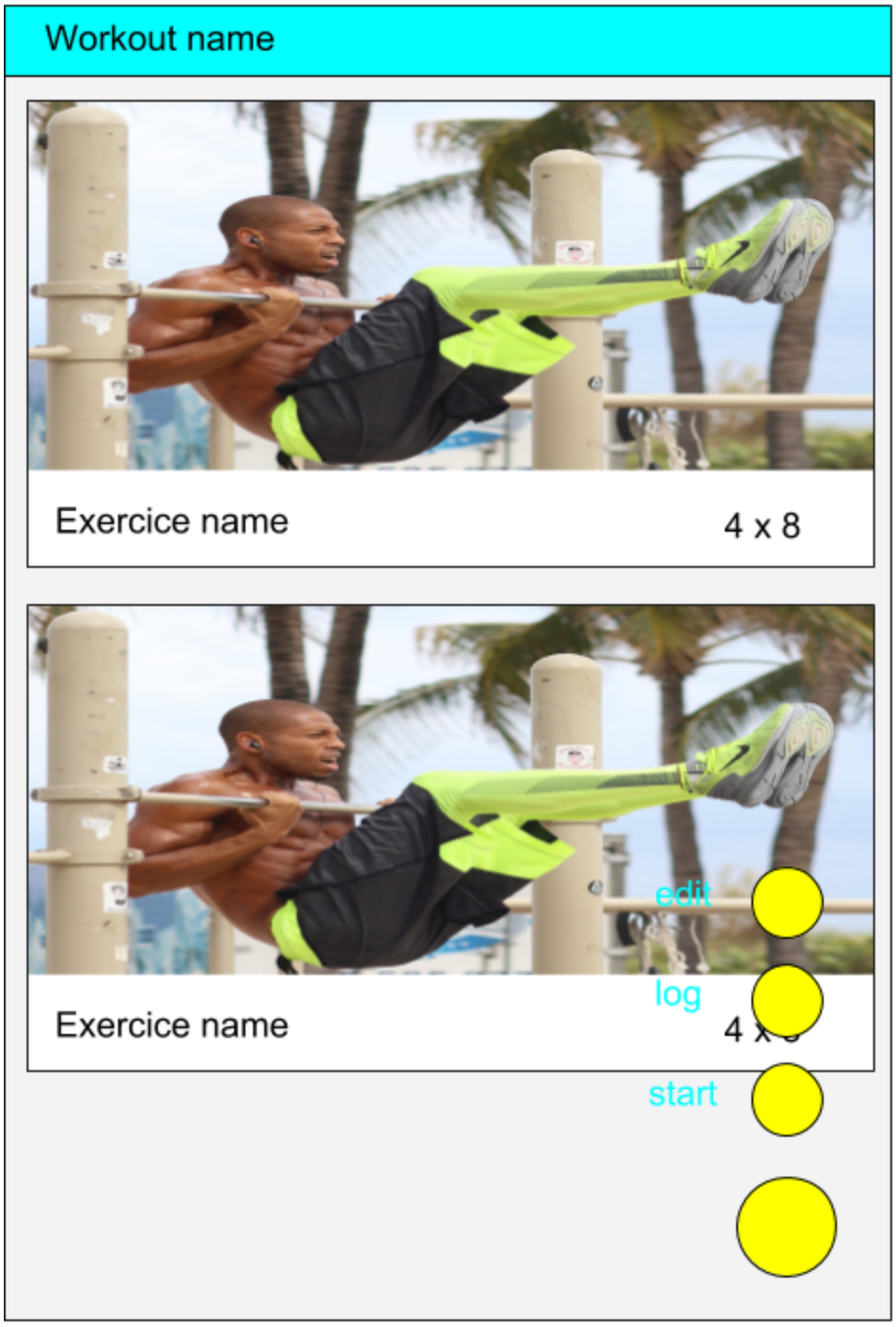
User Interface Mocks

Workouts list



List of all user's workouts. Each workout item has workout name, number of exercises and last date when workout was done. On phone items are displayed as list, on tablet the collection is displayed as grid.
Floating action button for adding a new workout.

Workout detail

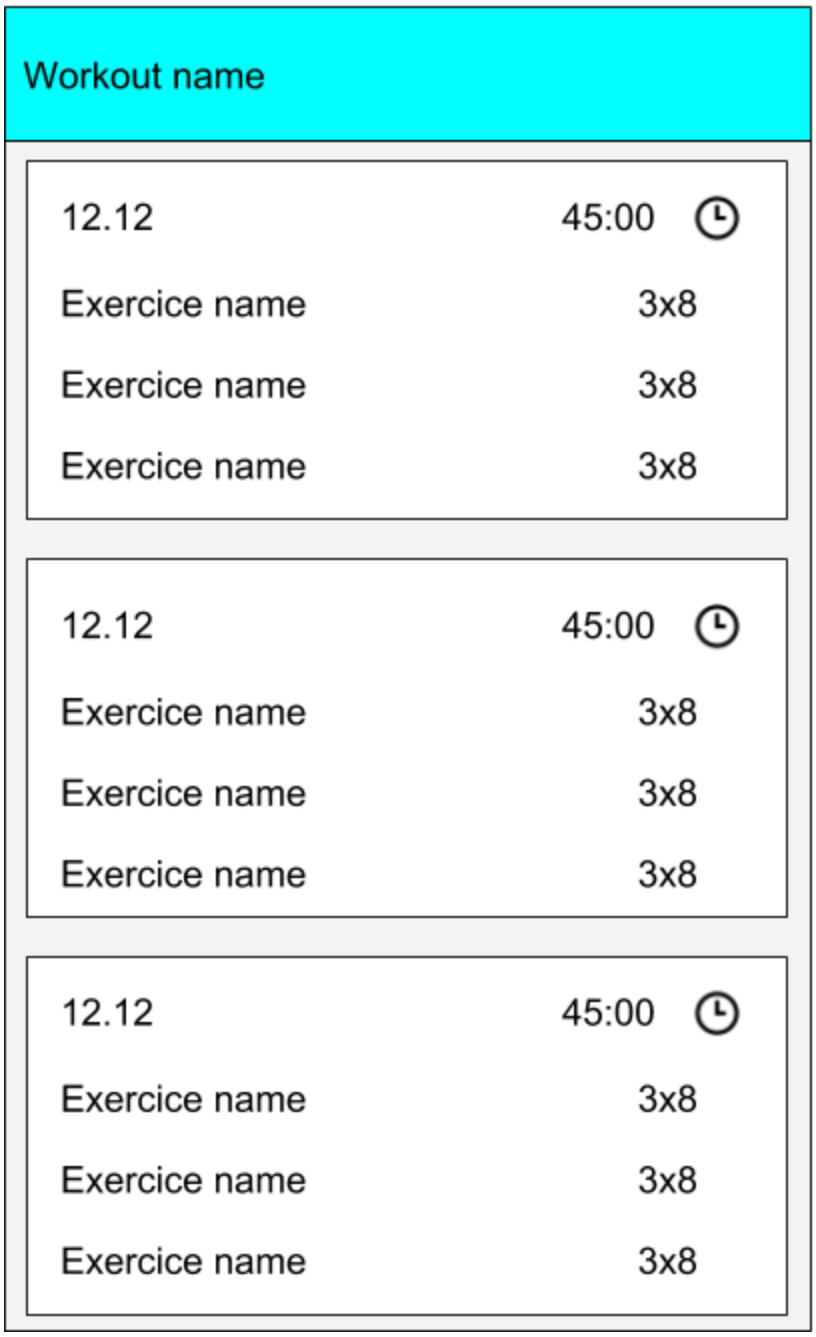


Workout detail screen. Here the user can see the whole workout. There is floating action menu with 3 options:

- Edit workout: The user goes to Edit Workout Screen. It is almost the same but all is editable, he can add/remove exercice, change exercice image, change repetition and series.
- Log: User goes to Log Workout Screen, where he can see Log of his workout.

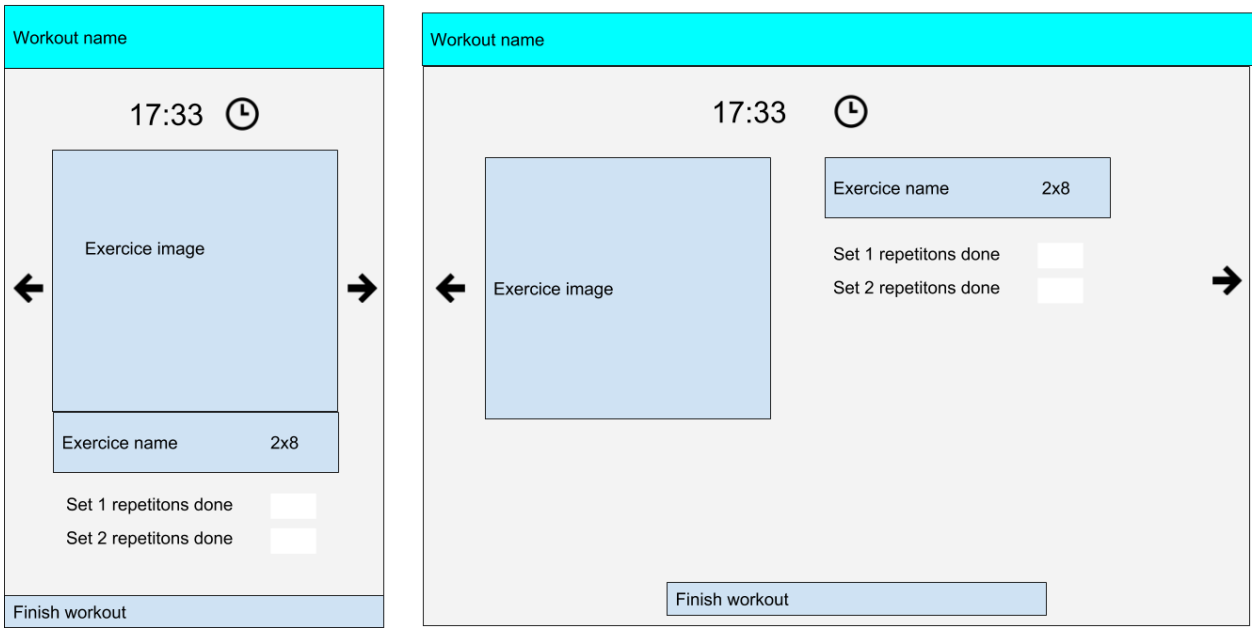
- Start Workout: user goes to Workout Activity.

Workout log



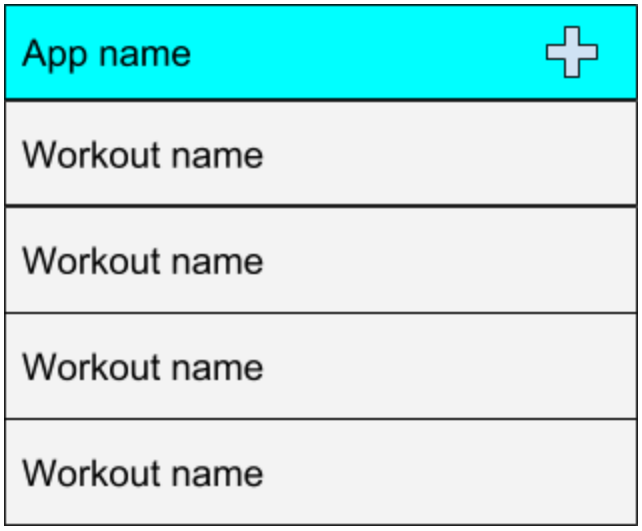
The log shows the date, the duration of the workout and detailed exercice information.

Workout activity



The workout activity has a timer showing the time passed since the start of the workout. The user can see exercise's image and number of sets and repetitions. After a set is done the user logs how many repetitions he did. When the user thinks he is ready, he can finish the workout.

Widget



The app will implement widget. It will show a list with most popular workouts. When a user clicks on a workout item, the app will start and he will be redirected to workout detail screen. The widget has button in the top right corner which will open the app with the new workout screen displayed.

Key Considerations

How will your app handle data persistence?

The app will handle data persistence with Firebase Realtime Database. It will implement login with email or phone.

Describe any edge or corner cases in the UX.

When app starts for the first time, login/sign up screen is displayed. The user authenticate himself and then the workout list is displayed. The user can select a workout or add a new one. The workout detail is displayed. If it is a new workout the have to add exercices and than save the workout. If it is not a new workout the user can edit it, see the log or start a workout activity. In the workout activity the user can slide through exercices, populate the repetitions done and finish the workout.

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

Firebase API - for data persistence. All data will be stored in the cloud.
Picasso - for image loading.

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

The app will be using Google Firebase Services. The user will authenticate with Firebase Authentication. The sign in methods will be set using FirebaseUI Auth. The app will pull all initial data with single request from the Firebase Rest API. The user will have already predefined workouts from which he can choose. All the data will be stored on the cloud.

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

First the build will be configured.

- Setup MIN SDK, Target SDK, BuildTools.
- Adding the libraries.
- Make sure everything compiles and there are no dependency conflicts.

Task 2: Implement Firebase Authentication

Implement Firebase Authentication UI

Task 3: Initial data pull

After successful authentication the app will pull all the initial data with an API call to the Firebase Rest API. This will be made with an AsyncTask. The data will be stored in the memory for further use.

Task 4: Implement UI for Each Activity and Fragment

- Build UI for Workout List - RecyclerView with FAB button.
- Build UI for Workout Detail - List of exercises with floating menu.
- Build UI for Workout Log- RecyclerView with cards.
- Build UI for Workout Activity - ViewPager for exercices and Timer.
- Build the App widget and its functionality.

Task 5: Free and Paid flavors

The app will have free and paid version. The paid version will be Ads free. The free version will contain ad banner at the bottom of the Workout List.