GitHub Username: xloli20

Fitness App

Description

Work out anywhere, anytime, no equipment needed. Favourite your best workouts to reach it locally. Set a reminder so you can't forget to your daily workout.

Intended User

This app is for people who want to stay in shape and have no time to go to gym.

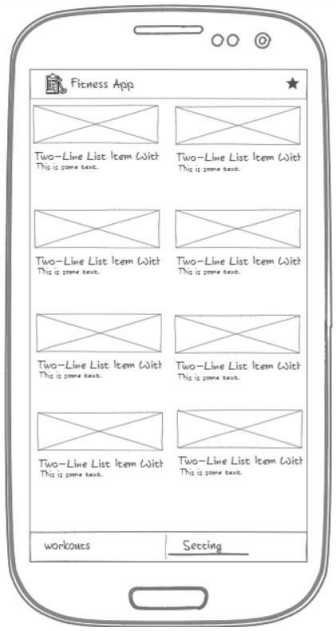
Features

List the main features of your app. For example:

- Shows a list of workouts
- Store Workout data.
- Sets reminder
- Plays Videos
- Switches language
- provides a widget to provide the favourite workouts to the user on the home screen.

User Interface Mocks

Screen 1

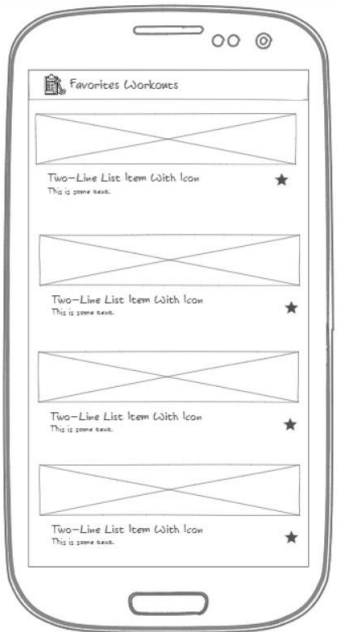


This is the Main Activity. There's Navigation Bar at the bottom so users can navigate between fragments. And this is the Workouts Fragment; it has Grid list for the Workouts.

Screen 2

This is the Detail Activity for each workout when user click on one item from the Grid list in the Workout Fragment. It has Collapsing Action Bar that has a picture for the Workout and its title. Also a YouTube Player and description Text View. And user can favourite an workout by pressing in floating action button to store it locally.

Screen 3



This is the Favourites Activity; it has a List of workouts that user has favourited it.

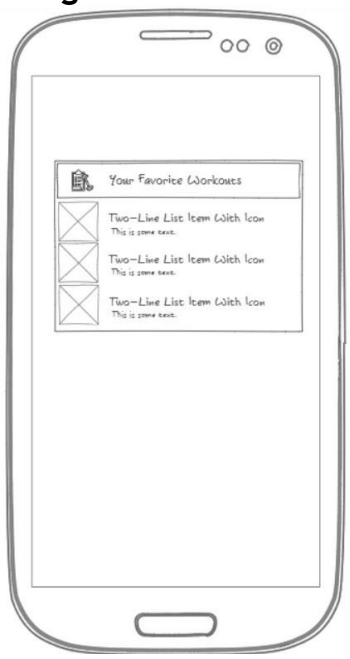
Screen 4

This is the Setting Fragment. There is switched On/Off for the reminder when it's off, SET TIME REMINDER Label will disappear. And a Spinner to change the language from English to Arabic.

Screen 5

When the user press on SET TIME REMINDER, this Dialog will appear to pick a time for the reminder.

Widget



This is the Widget for the app. It shows the Favourites Workouts.

Key Considerations

How will your app handle data persistence?

Room Database.

Describe any edge or corner cases in the UX.

if no internet is there, the app will show a message that the user should check his Internet connection.

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

Library	Why?	Stable version number
Glide	to handle the loading and caching of images	4.9.0

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

- Firebase Cloud Messaging, to send reminders to the users who enable it and set a time for it.

- AdMob, to display a banner for ads.

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

- Configure libraries.
- Add the needed permissions.
- Build Java Library for the Workouts Data.
- App keeps all strings in a strings.xml file.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity.
- Build UI for the Navigation between the fragments (WorkoutsFragment, SettingFragment).
- Build UI for WorkoutsFragment.
- Build UI for SettingFragment.
- Build UI for WorkoutDetailActivity
- Build UI for FavoritesActivity.
- Make sure to enables RTL layout switching on all layouts.

Task 3: Connect Java Library with the App

- Add the dependency to app build gradle.
- Retrieve the data from the Java Library and display it to RecyclerView in the WorkoutsFragment using Adapter.
- Use an async task to run image loading in the background.

Task 4: Implement Room Database

- Create Database.
- Design the Database.
- Create Dao Interface.

Task 5: Implement Firebase Cloud Messaging

- Connect your app with Firebase

- Add the needed dependencies.
- Edit your app manifest.
- Access the device registration token.
- More Instructions here: https://firebase.google.com/docs/cloud-messaging/android/client

Task 6: Create Widget

- Declaring an App Widget in the Manifest.
- Adding the AppWidgetProviderInfo Metadata.
- Create Widget layout.
- Create AppWidgetProvider.
- Pinning App Widgets.
- Creating an App Widget Configuration Activity.
- More instructions here: https://developer.android.com/guide/topics/appwidgets
- Read data from Room database and display it in RecyclerView using Adapter.
- Implement pending intent when the user clicks on a workout to lunch the app.
