GitHub Username: bashayerAlsalman

MyGym

Description

The application allows all people who goes to the gym to set their training plan in daily bases.
By specifying the exercise, machine, weight, speed, and so on.
The application will be developed by java programming language.

Intended User

This application is designed for all people who goes to the gym, and have a training plan.

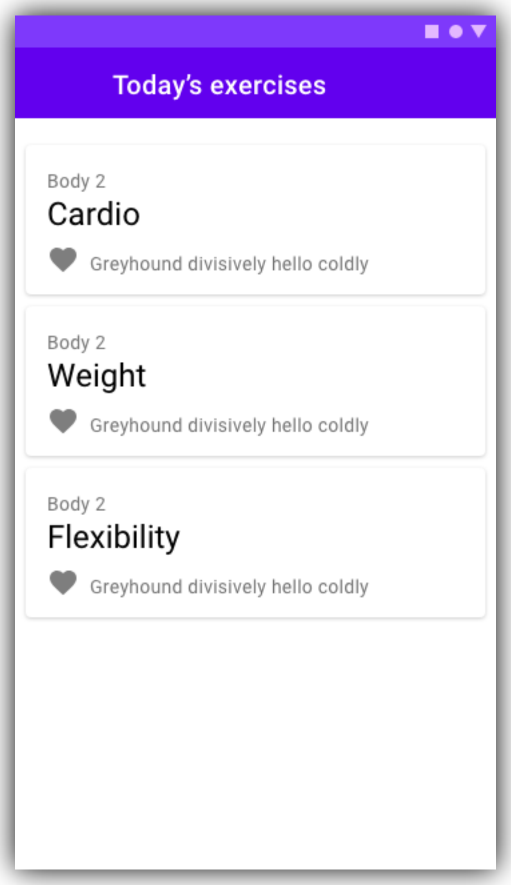
Features

- View the list of daily exercises
- Add new exercise
- Edit an exercise
- Delete an exercise
- Done/Undone an exercise

User Interface Mocks

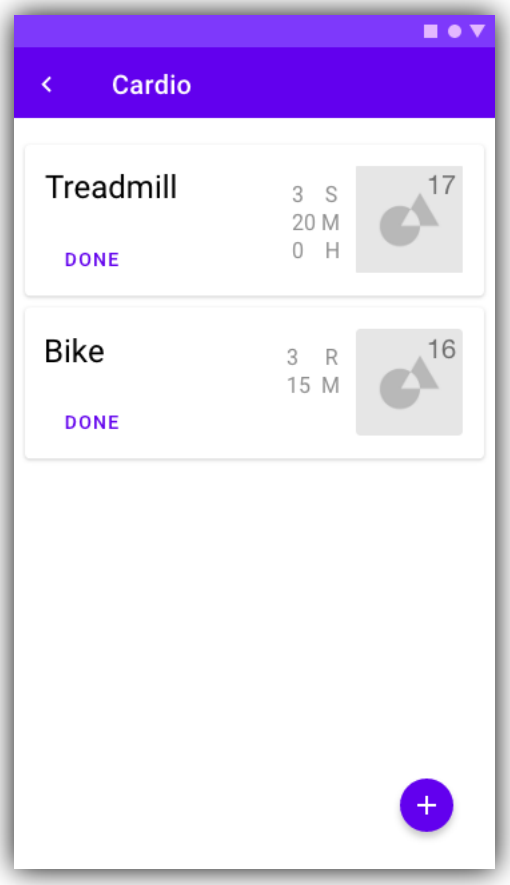
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1


Today's exercises
Body 2
Cardio
Greyhound divisively hello coldly
Body 2
Weight
Greyhound divisively hello coldly
Body 2
Flexibility
Greyhound divisively hello coldly

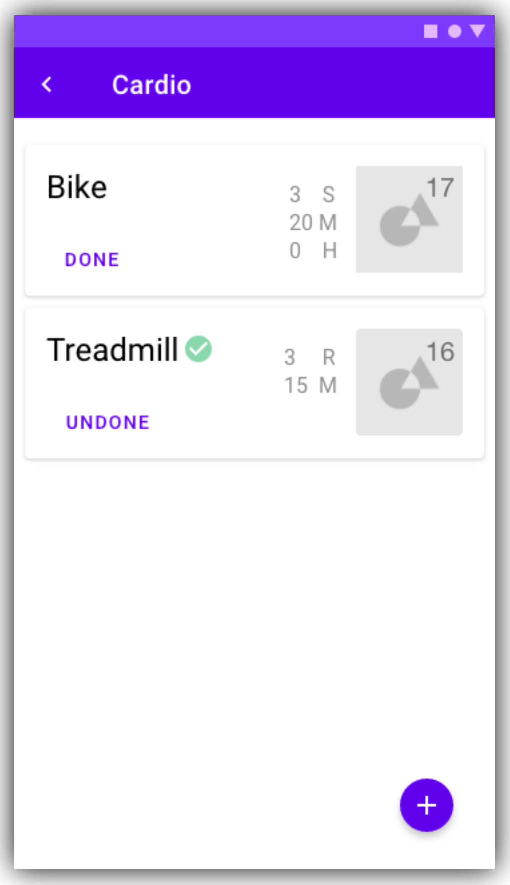
This is the main screen which contains the categories of all exercises.

Screen 2



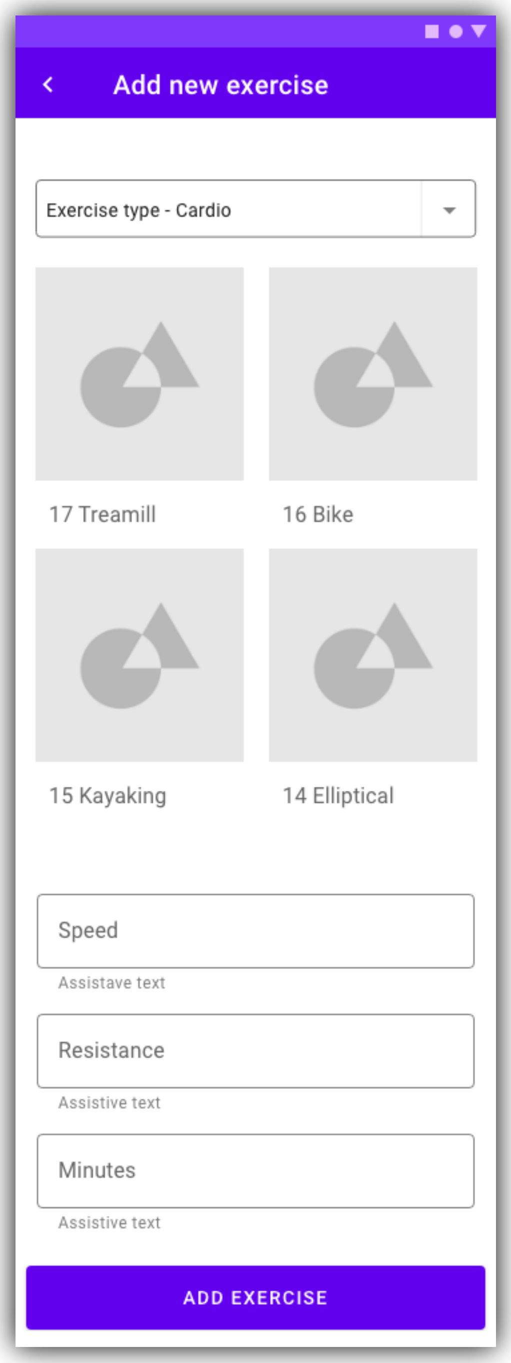
This screen appears when the user clicks on one of the exercises categories as here in the screen “Cardio”. Which will show all exercise under the selected category with the specification of speed, Hight, minutes, and resistance. As well as with the machine picture and number.

Screen 3



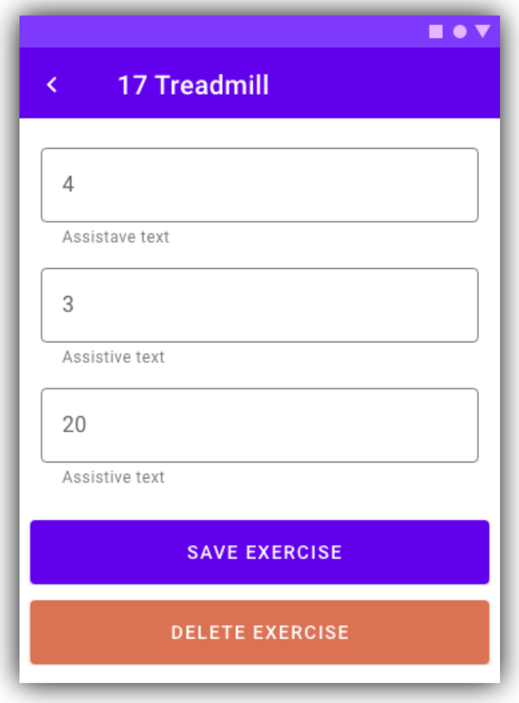
This screen shows the effect on clicking done in one of the exercises.

Screen 4



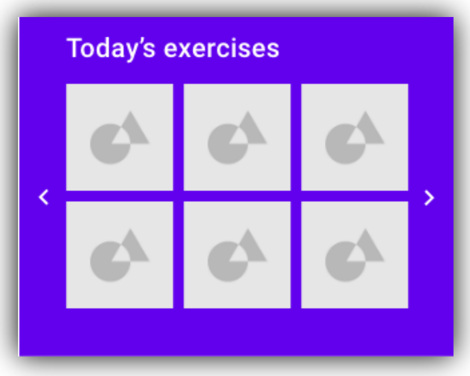
When user click on create new exercise.

Screen 5



This screen appears when the user clicks on one of the exercises to edit/delete it.

Screen 4



The widget will display the all exercises added by the user.

Key Considerations

How will your app handle data persistence?

The exercises information will be predefined in the application, and user will be able to create/edit his or her plan.
The data will be saved in room database.
The application will use LiveData and ViewModels when required.

Describe any edge or corner cases in the UX.

The user will be able to return to the previous screen by clicking on the back arrow.
And the user will be returned to the pervious screen after adding/editing/deleting an exercise.

The application will show an error message when the user tries to return from add/edit exercise screen without saving the information by showing this message “are you sure want to discard. The changes?”.

The application will show an error message when user enter incorrect number format or range “the number shall be from ** to **“

The user can not add new exercise without internet connection, it will show error message “please check your internet connectivity”

The user can user all application features without internet connectivity except adding new exercise.

The application will support mobile phones only in terms of UI.

The application will be designed for portrait orientation.

A content description attribute will contain small description of the xml item displayed.

Two languages [English | Arabic] will be provided in the application based on the default language of the mobile.

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

Library	**version**
Glide	4.9.0
Retrofit	2.6.0
Butterknife	10.1.0
RxJava	2.1.1
play-services-ads	17.2.1

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

The application main colors will be managed from firebase. By having remote configuration service by firebase.

AnMob to display adds in the add new exercise list.

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 the needed libraries
- Setup the application flavors

Task 2: Implement UI for Each Activity and Fragment

Build UI screen for the following

- MainActivity
- ExercisesListActivity
- AddNewExerscieActivity
- EditExerscieActivity
- Widget
- Exercise item

Task 3: Define exercise data

- Create Exercise mock API which return the information of exercises that needed to add new exercise. By this service https://www.mocky.io/
- Create constraint api which define the specification of each exercise as the speed ,Hight ,and so on. In the same service https://www.mocky.io/

Task 4: Integrate with services

- Adds:
    - Configure adds in the applications
    - Display the add in the top of add new exercise list.
- Firebase:
    - Configure what needed to use firebase.
    - When loading the application the color of the application will be read from firebase and save in sharedpreference.
- Exercise service:
    - Integrate the application will get all exercise service along with categories.
    - Integrate the application with get exercise specification.
    - App uses retrofit to call API services.
    - App uses RxJava to handle multiple threads.

Task 5: Write business logic

Build application business logic for the following:

- Display the exercises that added by the user in the exercises list.
- The DONE exercises will be displayed at the end of the list.
- Each specification will be displayed based on the selected exercise.
- Connect all screens be intent and pass the needed data between activities.

Task 6: Write validation

- All the text fields only accepts numerical values.
- Each exercise cannot be added multiple times.

Task 7: Implement widget

- The widget will display all added exercises in the application by the user.
- The exercises list will be updated after 30 minutes.
- 6 exercises at a time.