- Description
- Intended User
- Features
- User Interface Mocks
  - Screen 1
  - Screen 2
  - Screen 3
  - Screen 4
  - Screen 5
  - Screen 6
  - Screen 7
- Key Considerations
    - How will your app handle data persistence?
    - Describe any corner cases in the UX.
    - Describe any libraries you'll be using and share your reasoning for including them.
- Next Steps: Required Tasks
  - Task 2 Register/sign-in feature
  - Task 3 Fill predefined workouts and exercises
  - Task 4 Rest countdown timer
  - Task 5 Retrieve and display the log data from the server
  - Task 6 Finding friends by email
  - Task 7 Sharing exercises option
  - Task 8 Support Tablet

GitHub Username: Buli1212

PersonalGymLog

Description

‘PersonalGymLog is a great alternative for the ‘paper and pen’ approach. The advantage over existing training logs is the ability to create your own workout routines. It allows monitoring the rest time between sets by displaying a countdown timer. The user is given the option to save the results and compare them with friends. ‘PersonalGymLog’ provides easy, uncluttered interface which doesn’t distract from a training. All of the user data is saved on the server so the app can be used across different devices.

Intended User

The gym application ‘PersonalGymLog’ is a perfect tool for every person performing a weight training at the gym.

Features

The features include :

- creating fully customized training routines ,
- set of predefined workouts for new gym users ,
- saving the training info in a log which includes date , type of exercise and weights lifted,
- countdown timer for tracking down rest times between sets ,
- sound and vibration notifications when the rest time is over ,
- sharing results with friends,
- user data stored online

User Interface Mocks

Screen 1

When a user starts the app he/she will be present with the option to register using email/password combination or to log in. After the user logs in for the first time the credentials will be saved on the device and the user will be taken to the main activity automatically.

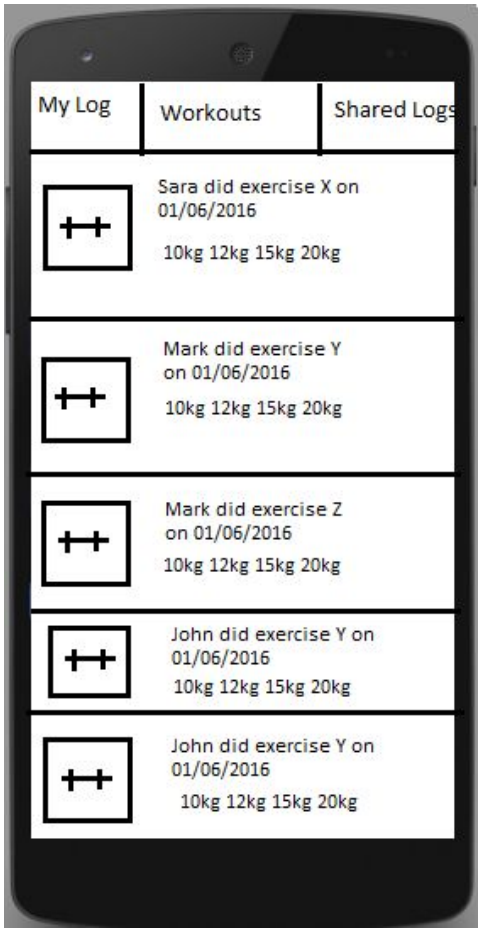
Screen 2

After logging in the user will be taken to the main screen consisting of three swipeable views. The default view will be the selection of workouts. The user will be able to add a new workout here and access each of the predefined workouts. After choosing the workout selection of available exercises will be shown. For each exercise there will be a different image representing it.

Screen 3

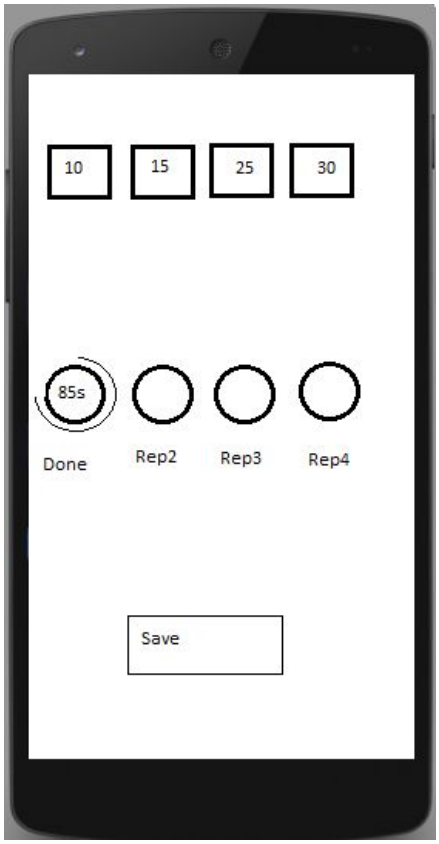
Another view from the main screen. This one will show only exercises completed by the logged user of this application.

Screen 4



The last view in the main screen. This one will show only exercises completed by friends.

Screen 5

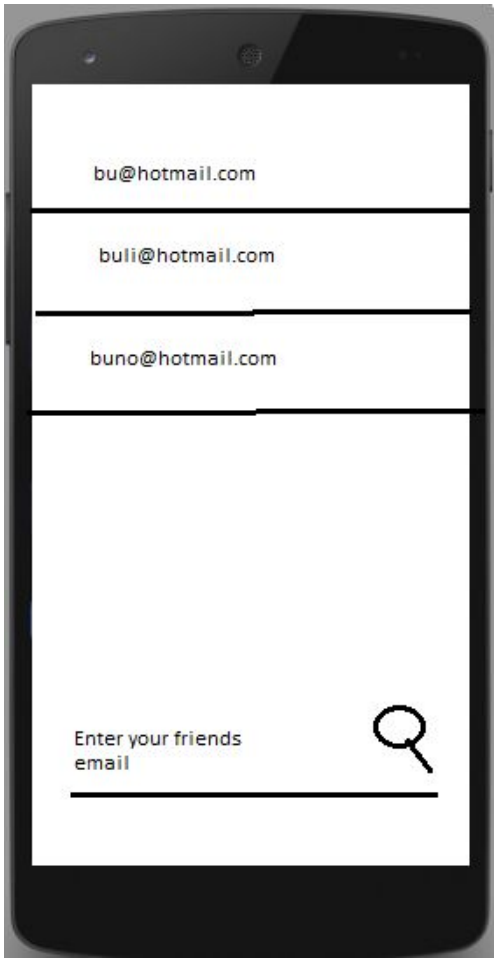


This screen will be displayed after a user chooses an exercise to perform. Here the user will be able to choose the weight attempted for every set(top of the screen). In the middle countdown timers will be displayed for every set. Whenever the user completes the set he/she will be able to start another rest timer. By clicking on Save the exercise along with scores will be saved in the user's log and shared with friends if chosen so.

Screen 6

Here a user will be able to choose which friends should see his/her completed exercises. Clicking on ‘Add New Friend’ button will take a user to another screen where more friends may be added.

Screen 7



This screen will allow the user to search his/her friends in the online database using their email address.

Key Considerations

How will your app handle data persistence?

All of the user data will be stored in online Firebase Database. However the Firebase data persistence option will be enabled allowing data to be displayed even if the device is offline.

Describe any corner cases in the UX.

The back button will provided for every screen going deeper than the main activity. The countdown timers for rest times will be running in the background even if the screen is locked.

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

Firebase for storing and retrieving data from online database as well as user authentication.
Todd-Davies:ProgressWheel for countdown timers animation.
RecyclerViewMergeAdapter to include data sets from different sources in one recyclerview list.e.g. loading predefined and user-created workout plans in one list.

Next Steps: Required Tasks

Task1 Horizontal swiping and workout list

- Setting up horizontal swiping between three main fragments.
- Including recyclerview in each fragment
- Configuring Firebase adapter to sync workout data

Task 2 Register/sign-in feature

- Configure input validation and account creation/authentication using Firebase
- Add 'log out' option

Task 3 Fill predefined workouts and exercises

- Provide predefined workouts and exercises
- Give option to add and remove workouts/exercises

Task 4 Rest countdown timer

- Use ProgressWheel library for countdown timers
- Add a weight selector for each set
- Save the data(weights/exercise name/time) on the server using Firebase

Task 5 Retrieve and display the log data from the server

- Retrieve the the exercises completed by the user of the app and display them in the list with the latest on the top

Task 6 Finding friends by email

- Create an activity for querying users' emails. The user will be able to add a friend by email.
- Validate input
- Add the friends' emails to the Firebase database and retrieve the list displaying it to the user

Task 7 Sharing exercises option

- Let the user choose when and with whom he/she can share the completed exercises
- Each exercise completed by the user should now land in the personal log database and in the friend's database
- Display all of the shared exercises shared with the user in one of three fragments in main activity

Task 8 Support Tablets

- Add tablet specific layouts
- Display the workout selection activity on half of the screen and exercises for the selected workouts on the other