Requirements and Analysis Document for training tracking app (android)

Victor Hui, Philip Lindström Rabia, Patrik Olsson, Valdemar Vålvik, Oscar Wallin

2021-09-16
version 1

1 Introduction

Moving past the time of the Corona pandemic and lockdowns, people start to get back to the gym. Therefore the demand for fast and agile training apps increases. The app provided by AEY investment group aims to give both experienced and inexperienced gym attendees. The tool to help with planning and tracking of workouts and progression. Through a sleek UI and easy functionality the aim is to, through the app, motivate the user to a more active lifestyle.

1.1 Definition, acronyms, and abbreviations

- PB/PR “Personal best/Personal record”
- reps “repetitions”
- sets “is the multiplier of repetitions separated by rest ex. 3 sets of 12 repetitions”
- 1-rep max “what is the heaviest someone can lift only 1 time”

2 Requirements

2.1 User Stories

S1: Workout plan

Implemented

- Yes/No

Description

As a user, I want to make a workout plan that contains x amount of workouts.

Functional

- There is a way to add/remove a workout plan
- There is a way to name the workout plan.
- There is a way to edit(add/remove workouts) the workout plan
- There is a way to view previously saved workout plans

Non-functional

- The user goes through a wizard-like process to create the workouts.
- The user can delete workouts by swiping left and then pressing a minus sign
- The user can add a workout by pressing a button.

S2: Workouts

Description

As a user, I want to be able to create workouts that contain exercises with sets and reps.

Functional

- There is a way to add/remove a workout
- There is a way to name the workout
- There is a way to edit(add/remove exercises) the workout
- There is a way to view previously saved workouts.

Non-functional

- I can delete exercises by swiping left then pressing the minus sign
- I can add an exercise by pressing a button

S3: Exercises

Implemented

- Yes/No

Description

As a user, I want to be able to create exercises. When creating an exercise I want it to be connected to an exerciseId. All exercises of the same type(e.g Bench Press, Deadlift) should have the same exerciseId.

Functional

- There is a way to add/remove exercises.
- There is a way to name the exercise.
- There is a way to add an exerciseId for identifying purposes.

- There is a way to edit(add/remove sets) the exercise.
- There is a way to view previously saved exercises.
- The exerciseId should be saved automatically in the database on creation.

Non-functional
- The user cannot have two exercises of the same name.

S4: Current Workout

Implemented
- Yes/No

Description
As a user, I want to create a current workout session where I can input data (weight, rep, sets) for each exercise.

Functional
- I can start a current workout by choosing a workout plan and then a workout from that plan.
- I can change information about the exercise such as sets, reps and weight during my workout.
- I can add exercises that weren't originally in the workout.
- Time that has passed since workout has started should be displayed(in minutes/seconds, and maybe hours/minutes/seconds eventually)
- After finishing a workout, the workout should be saved automatically.

Non-functional
-

S5: Saving PB's

Implemented
- Yes/No

Description
As a user, I want to save my PB's and write down the date, weight and reps as well as what exercise.

Functional
- I can add and update personal PB's for my account.
- After finishing a workout the app will check automatically if I have achieved a new PB on any exercise.

Non-functional
- The relevant PB-data can be accessed at the history tab

S6: PB graph

Implemented
- Yes/No

Description
As a user, I want to see the progress of my PB's through a graph.

Functional
- Use the PB-data from an exercise and display it as a graph.

Non-functional
- The relevant PB-data can be accessed at the history tab

S7: Calculating 1-rep max

Implemented
- Yes/No

Description
As a user, I want to be able to calculate my 1-rep max.

Functional
- Use a calculator and the following algorithm (insert here) to easily show the 1-rep max?
- A part of our PB section?

Non-functional
- The user has 2 input boxes to write nr of reps and weight. The calculator shows the corresponding 1-rep max as output.

S8: About Us page

Implemented
- Yes/No

Description
As developers, we want to be seen in suits on our 'About Us' page.

Functional
- Can I find and view an "about us" page?
- Is there relevant information on this page?

Non-functional

Story Identifier: S9
Story Name: Workout presets

Implemented
- Yes/No

Description
As a user, I want to be able to copy certain workouts both from a set of standard workouts or other famous workout presets

Functional

- Have a (database) with standard and other known workout presets
- Ability to swap between workout presets

Non-functional

- These could be available at the same place as the user created workouts/exercises

S10: Setting options

Implemented

- Yes/No

Description

As a user, I want to be able to change different settings.

Functional

- Can I change the color theme?
- Can I change measurement units?
- Can I change the app based on acceptability problems, like color blindness?

Non-functional

- The settings will be available at the settings tab. A usage of checkboxes or some other input box will be used for the different setting options.

S11:Startpage

Implemented

- Yes/No

Description

As a user, i want to get some fun and inspiring information from the apps startpage

Functional

- Can I view a graph of some interesting performance of past workouts?
- Can I get some inspirational quotes/texts?

Non-functional

S12: Navigation Bar

Implemented

- Yes

Description

As a user, I want to be able to easily change between the different parts of the applicaton.

Functional

- There is a way to switch between the different parts of the program.
- The bar for changing to different parts is located at the bottom of the application.
- There is a way to see where you currently are by looking at the navigation bar.

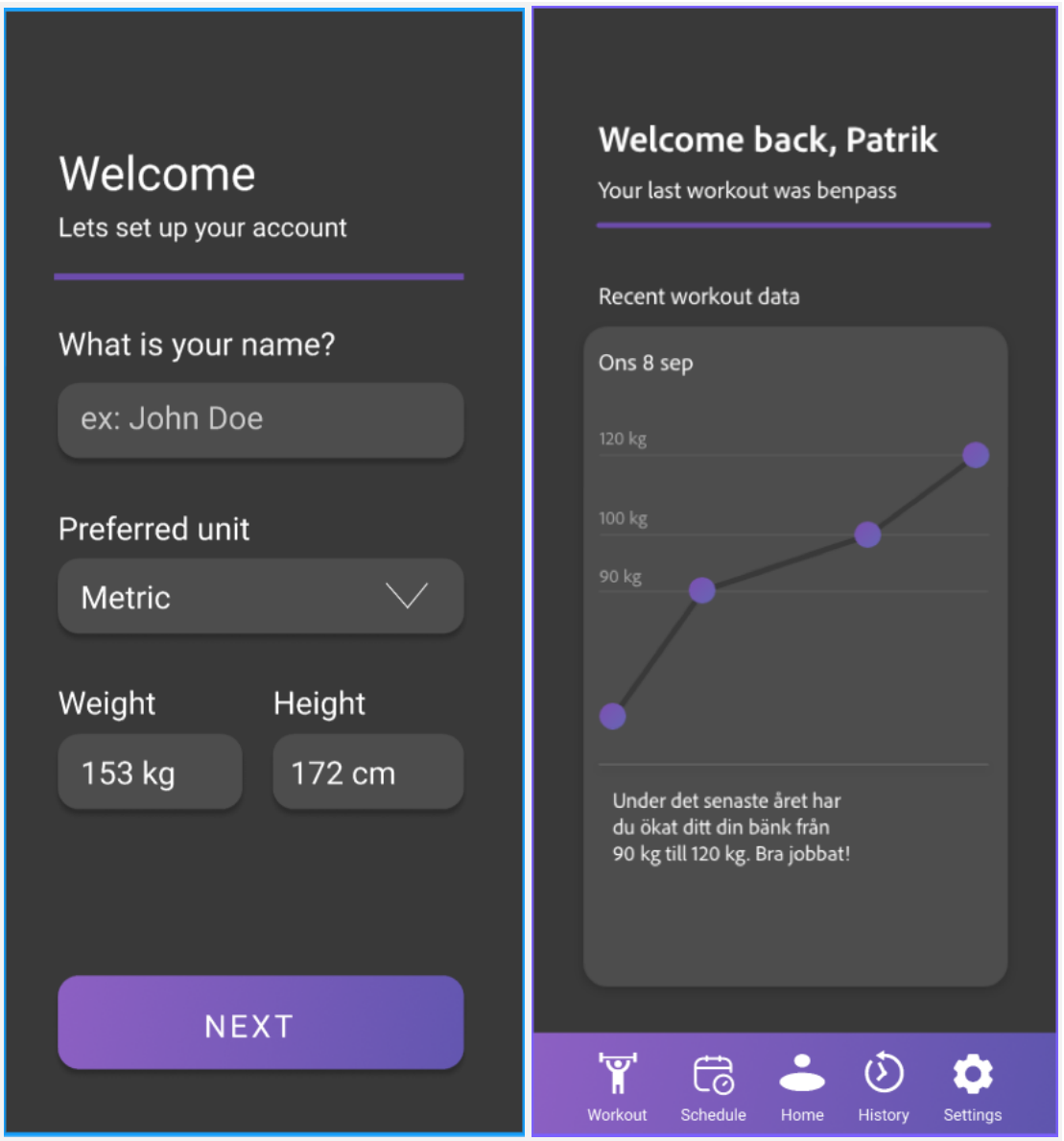
Non-functional

- You should be able to see the text and animation of the nav-bar at all times.

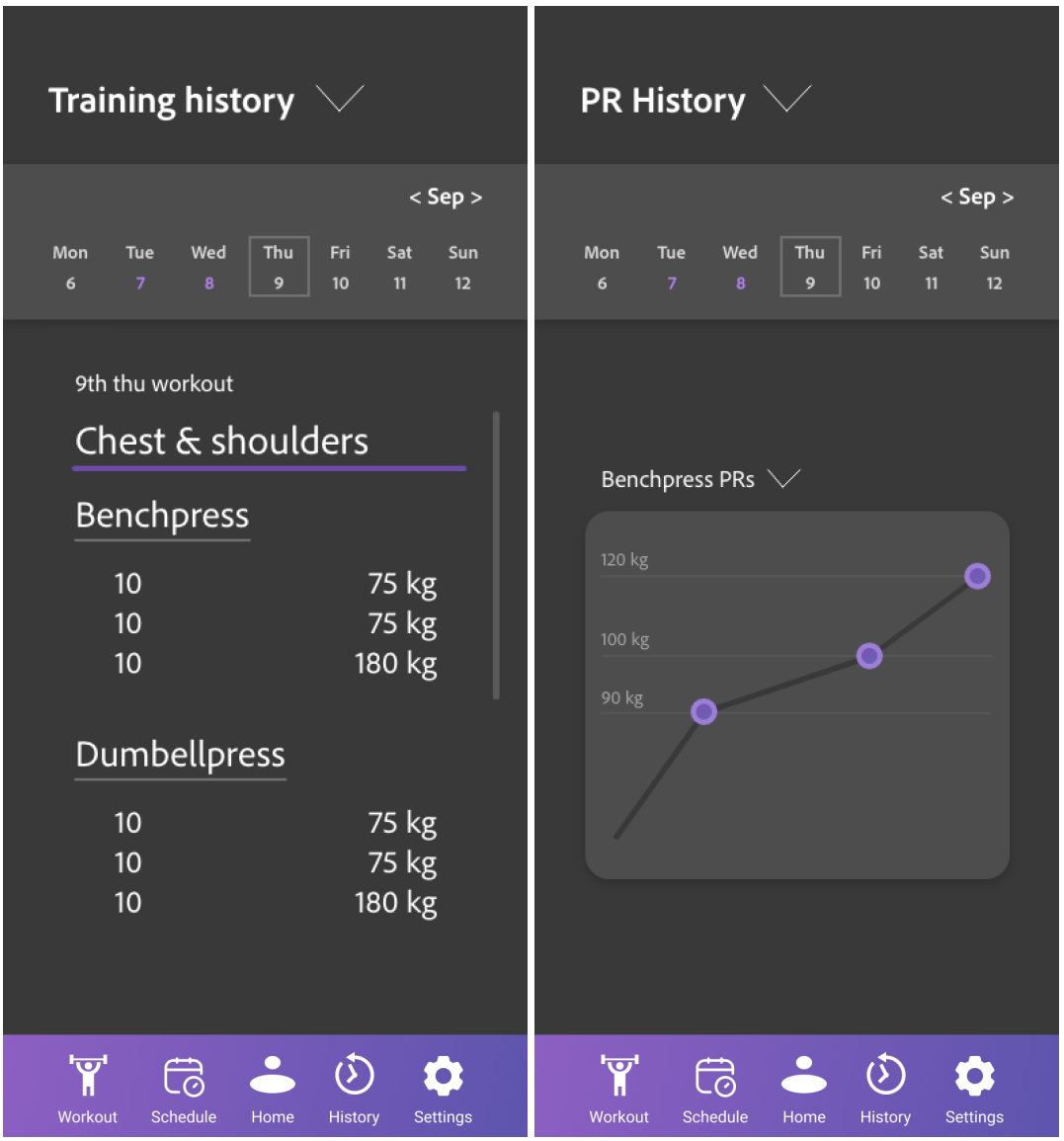
2.2 Definition of Done

- All the acceptance criterias are met for the user stories
- All the co-authors to the user stories commits are given
- The user story-specific code is merged to the main branch
- The code should be executable
- The code should be documented

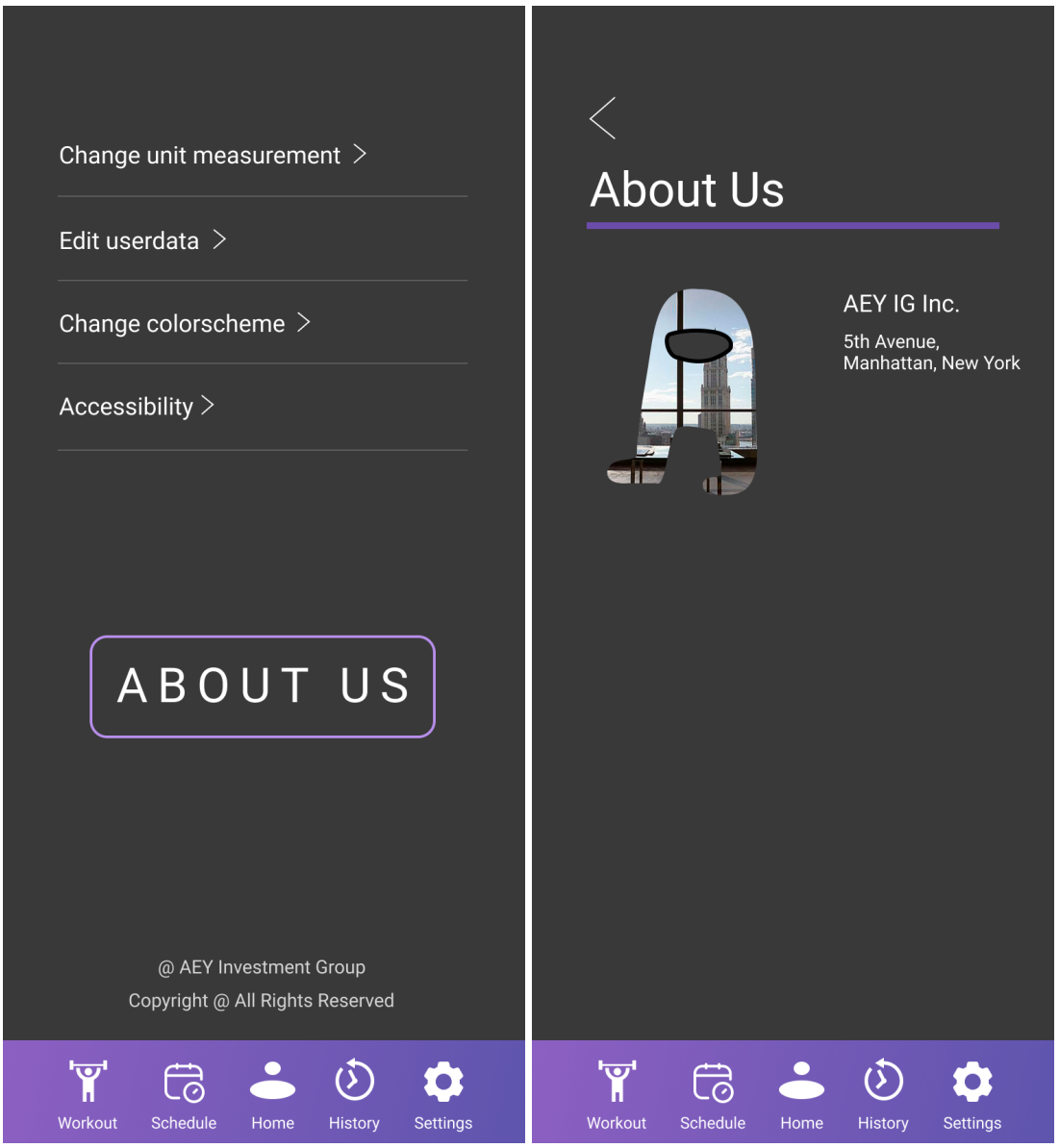
2.3 User interface



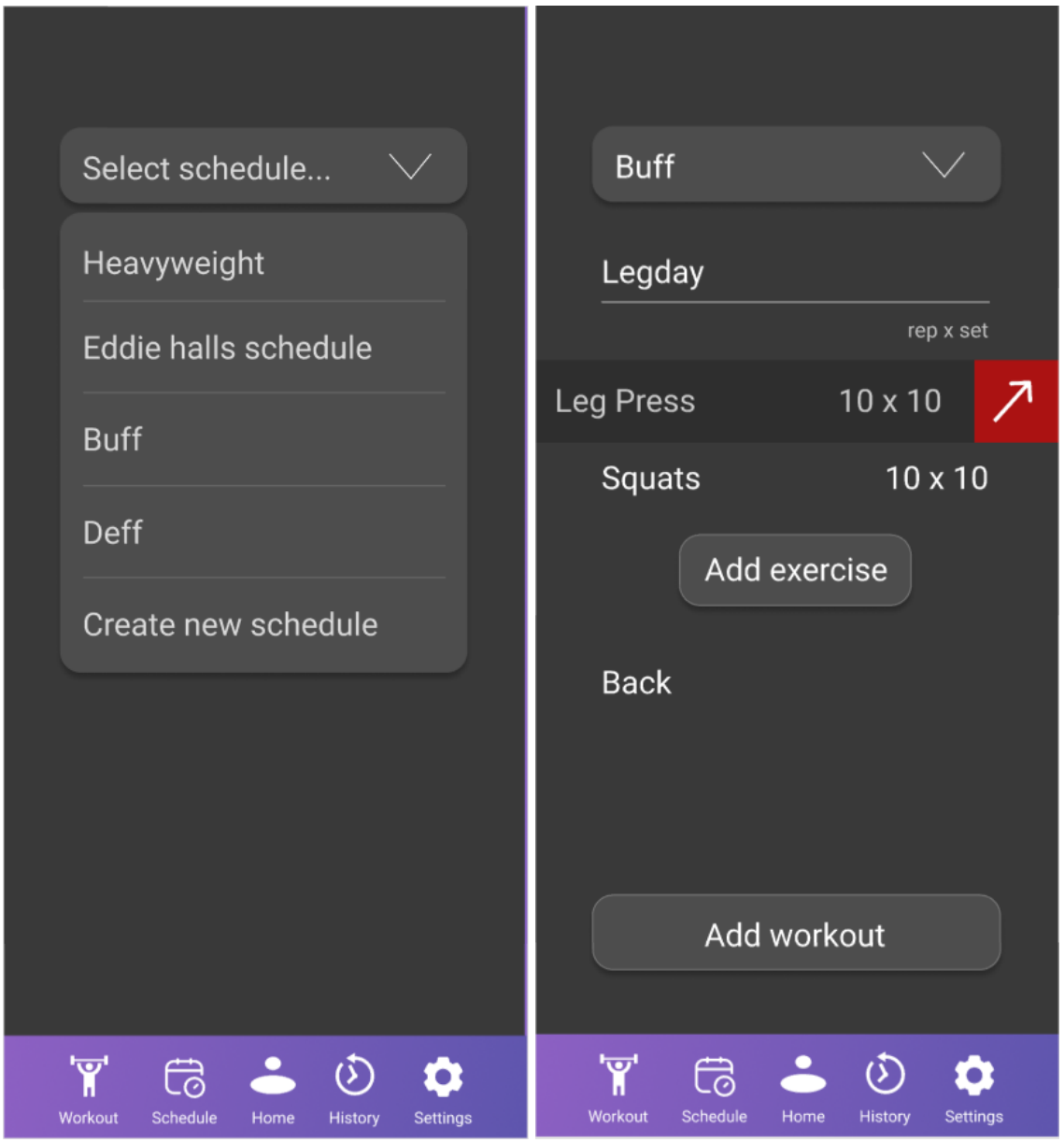
The first time using the app, the user needs to provide some information (left). The application is divided into 5 tabs as seen in the bottom navigation menu (right). The one in the picture is the “Home”-tab and provides some motivating information.



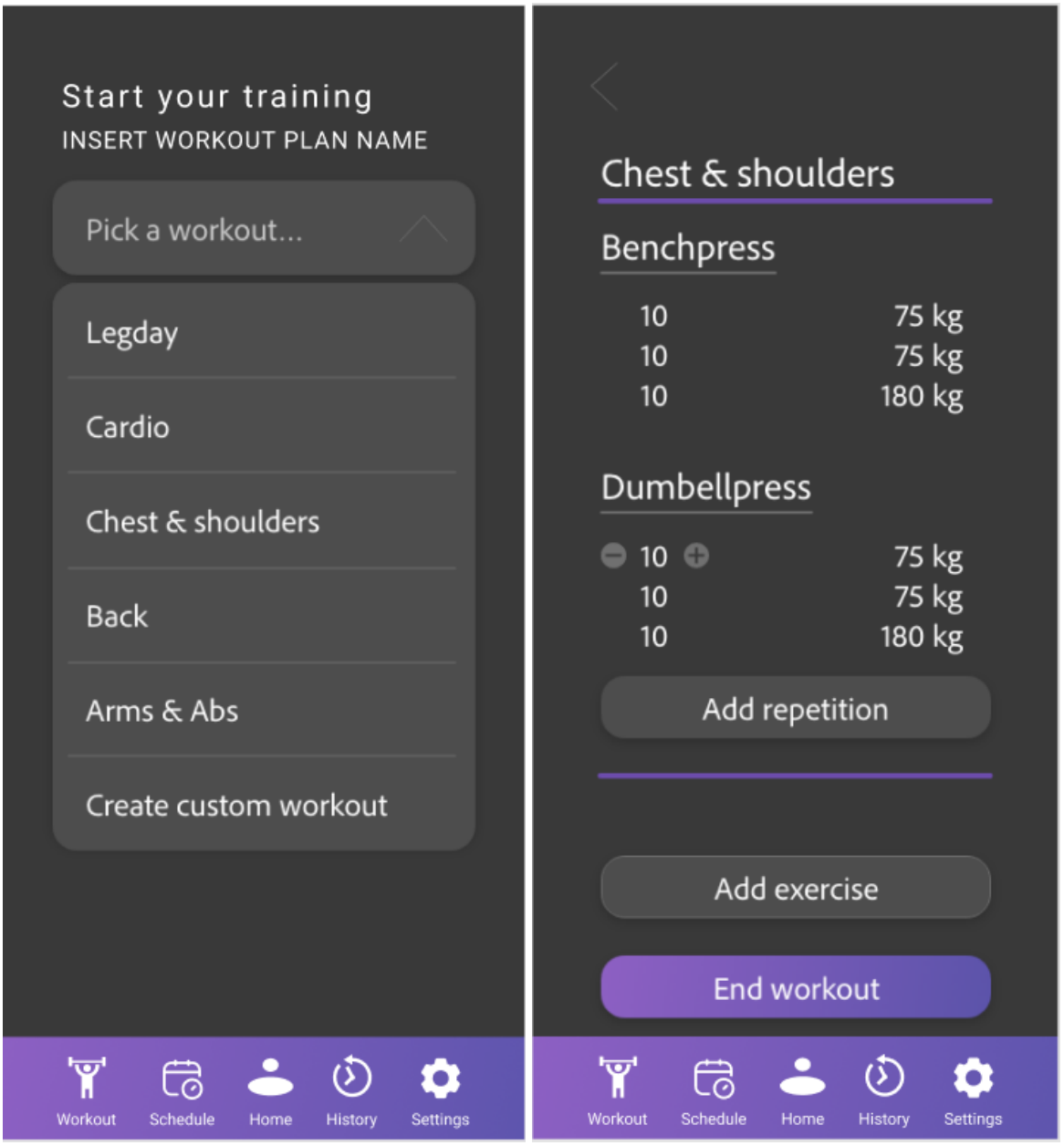
These two pictures show the view in the “History”-tab. The user can alternate between training history (left), which shows the workout performed on a specific date and pr history (right), which shows the pr-progression (pb).



This view shows the “Settings”-tab (left) and “About us”-tab (right). In settings the user is able to change some variables presented in the picture above.

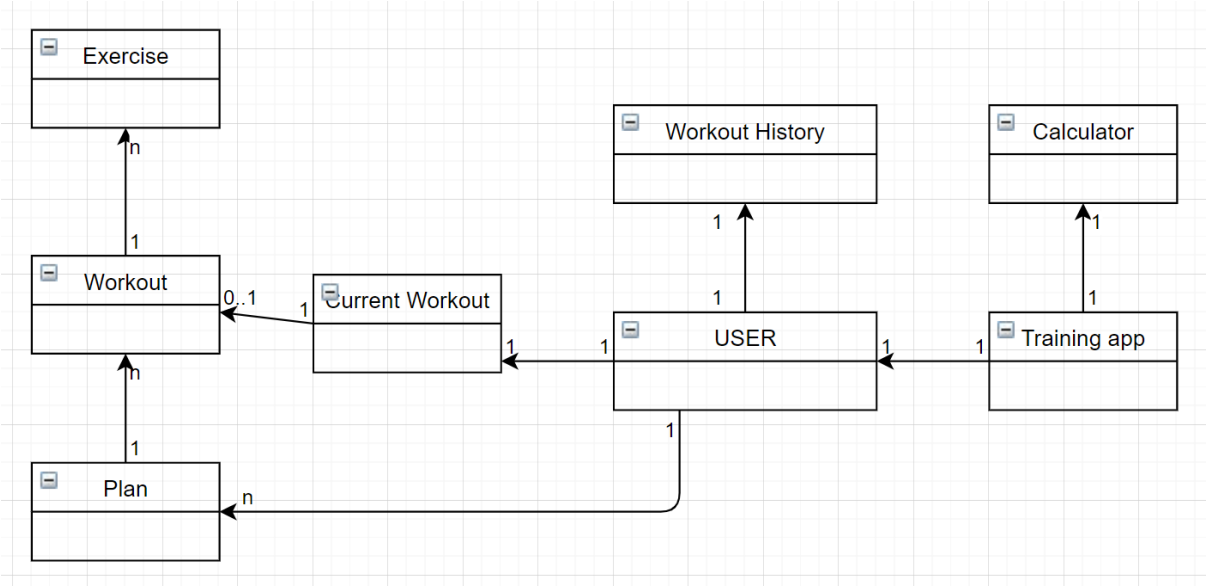


This is the “Schedule”-tab. Here the user can create a new schedule or enter previously saved ones (left). When pressing to either enter an existing or creating a new schedule, the user is presented with options to add new workouts and furthermore exercises inside these workouts (right).



This is the “Workout”-tab. At first the user is presented by a list of available workouts based on the preplanned plan from the schedule (left). After selecting a workout, all the exercises are displayed in that workout (right). In this stage the user can edit weights and reps depending on how the actual workout went. If the user detours from the workout, the option to add exercises to the current workout is available. This does not affect the original exercises saved in a workout.

3 Domain model



3.1 Class responsibilities

Training app is supposed to act as a datahandler that connects all the pieces of the model and makes it easy to pass it along to the different viewmodels.

USER is seen as a step down from **Training app**. It connects all the classes that handle everything from creating a workoutplan to storing it in the **Workout History**.

Plan are the base for a schedule of actual workouts. One **Plan** is made out of one or more workouts.

Workout is the building block for plans. In turn, **Workout** is made out of one or more **Exercise**.

Exercise is the bottom building block for a **Plan**. Here is information of what exercise, and how to perform it through sets and reps, given.

Current Workout acts as a copy of a selected **Workout** in progress by the user. This is because the user can give more information about how the workout actually played out, and therefore the app should not edit the originally saved **Workout**. Some information that can change/be added in **Current Workout** is the number of reps the user could do (maybe not the same as the original **Workout** due to fatigue etc.). Also what weight was used and if the user decided to add an **Exercise** not pre planned.

Workout History is supposed to handle the logic of displaying the completed **Current Workout**s in a way that is seen in the *2.3 User Interface* chapter at the “History”-tab.

Calculator is a simple class that calculates the possible 1-rep max from a formula that takes the amount of reps and the weight that was used.

4 References

- **Circle Ci**
  - Used for continuous integration, alternative to travis
  - https://circleci.com/
- **Figma for prototyping**
  - Prototyping tool
  - https://www.figma.com/
- **Google DrawIO**
  - Diagram software used for UML, domain model
  - https://app.diagrams.net/
- **Android studio** (IDE)
  - Integrated development environment for Google's Android operating system
  - https://developer.android.com/studio
- **JUnit**
  - Unit testing framework for the Java programming language.
  - https://junit.org/junit5/
- **Gradle**
  - Build tool used with Android Studio
  - https://gradle.org/