GitHub Username: scott7462

WorkHard

Description

Workhard is an agile app that aims to assist athletes calisthenics with the creation and customization of their workouts. And keep tracking your daily activities quickly and simply. While allowing social sharing daily and overall achievements.

Workhard not want to revolutionize calisthenics workouts. He wants to simplify and become the place where an athlete can quickly select a workout as needed and do it to focus on each day be stronger and more agile.

Intended User

Workhard is an app designed for athletes of all levels, from novice to advanced. Have a clear objective, to be better every day in their practice routines.

Features

- Create and edit workouts.
- List preset workouts.
- Save custom workouts.
- View history of workouts performed.
- Detailed training conducted.
- Share results of testing at Facebook and Google Plus.
- Perform tracking workouts.
- Log in.
- Register for the app.

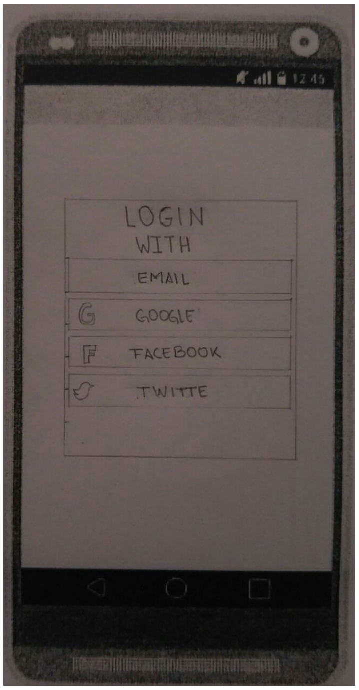
User Interface Mocks

Launch

Launch Screen: Home screen of the app. This screen will take the user each time the application is started.

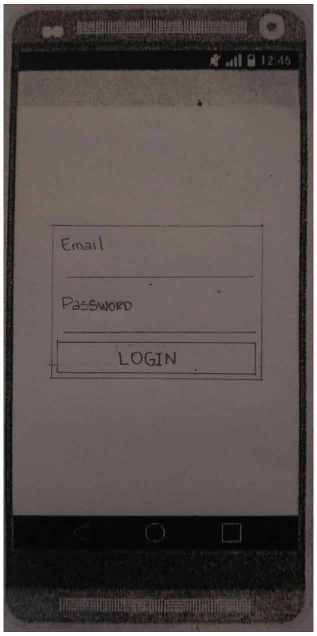
Login

Login: login screen. You can authenticate with social networks like Facebook or Google Plus. You can also press login by email to access your own email profile.



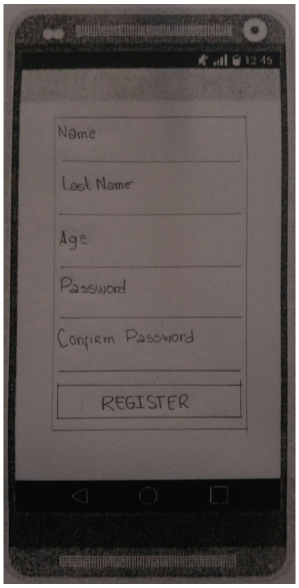
Login by Email

Login: login screen. On this screen the user can enter credentials through email and password or you can also press the record button to access the registration screen of the app.



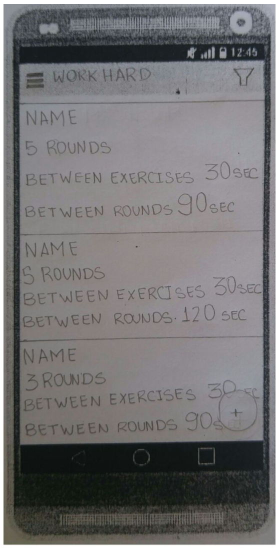
Register

Register: On this screen the user will enter your Name, Last Name, Email, age and password to create authentication credentials within the app. By registering you will automatically enter into the main application screen.



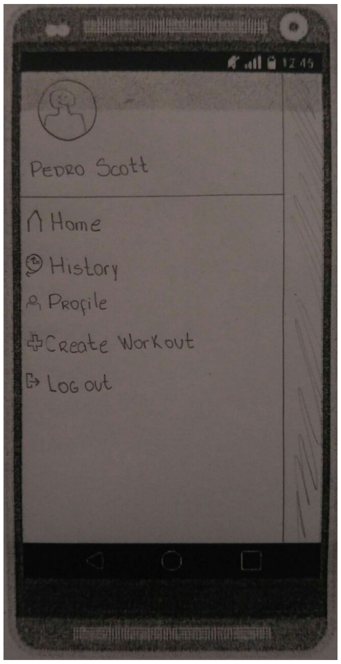
Home

Home: User choose a list of default by the server app workouts. These can be filtered by: most used or by the body where the training focus. You can also access the menu navigation app.



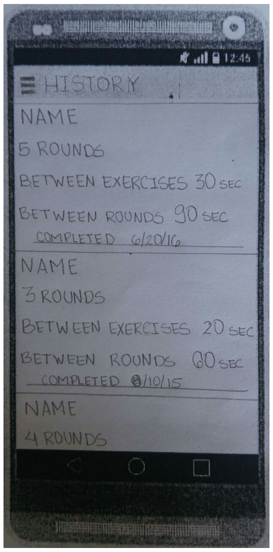
Menu

Navigation menu: On screen the user can navigate through the main screens of the app. Look at your profile picture on top of the screen.



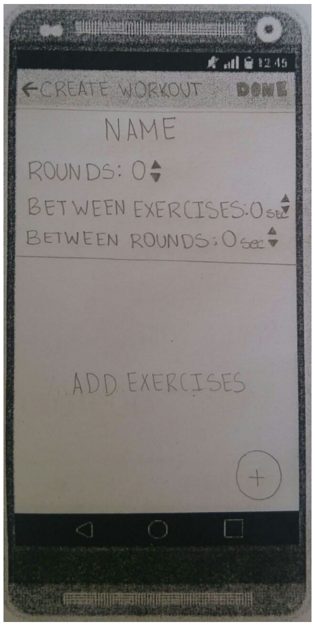
Workout History

Workout history: In this section the user can view the log of all training that previously made and filter this list by muscle worked body or date.



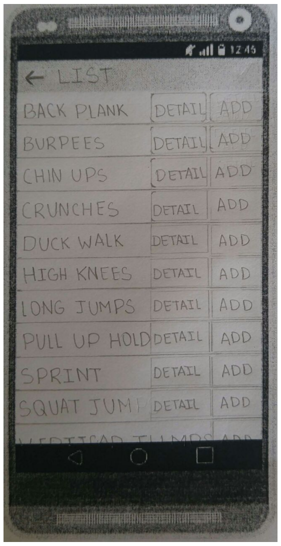
Create workout

Create Training: The training create screen has a form where the user can add the details of your workout as the number of rounds, the rest time between rounds, the name of training. you can also find a list of exercises to choose and where you can specify the desired number of repetitions. To add an exercise to the list, the user can press the floating button to access the search list of exercises.



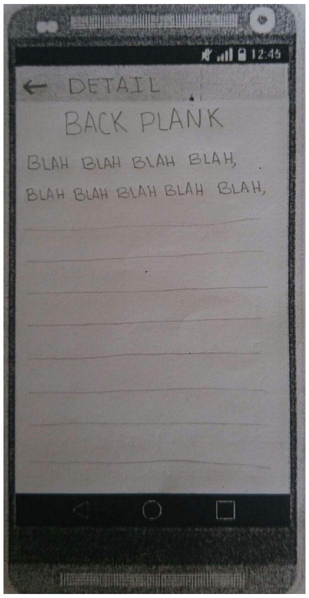
List of exercise to workout

List of exercises: The user can view all available exercises. By giving one click, this is added to the training that is being created. This list can be filtered by name through a sought in actionbar or can be filtered by muscle group.



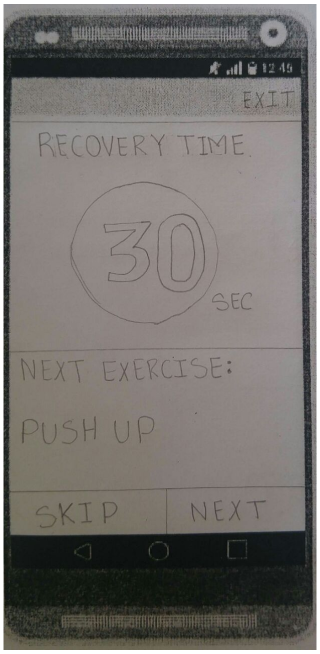
Detail of exercise

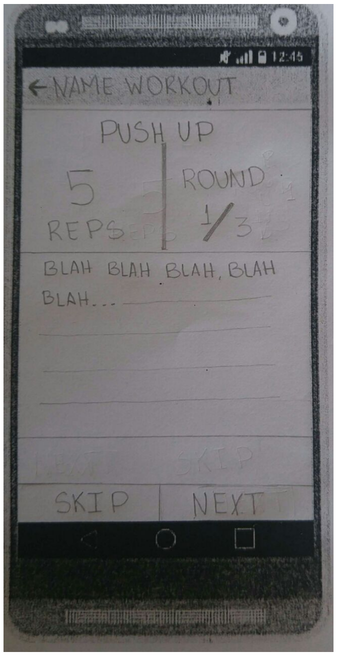
Detail of exercise: In this screen you can see a brief description of the exercise, with a group of three sample images of the same.



Do Workout.

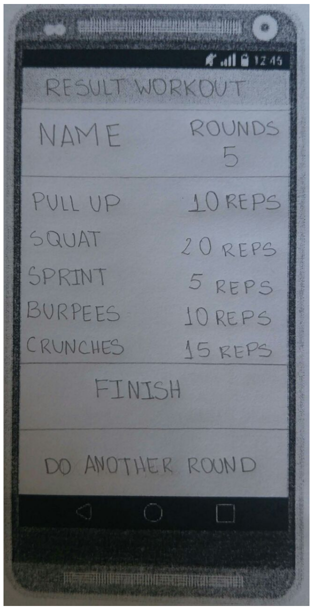
Exercise: This display accompanies the athlete during the course of the period showing the details of each and every time a stopwatch rest. This timer is you can increase or decrease the same, on the same screen as it may increase or decrease the number of repetitions to run.





Result Workout.

Training result: This screen is where the results of the exercise carried out by the athlete will be. By clicking the button finish training sail training the history screen.



RESULT WORKOUT

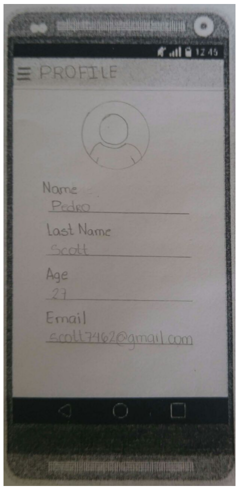
NAME	ROUNDS 5
PULL UP	10 REPS
SQUAT	20 REPS
SPRINT	5 REPS
BURPEES	10 REPS
CRUNCHES	15 REPS

FINISH

DO ANOTHER ROUND

Profile.

Training result: This screen is where the results of the exercise carried out by the athlete will be. By clicking the button finish training sail training the history screen.



PROFILE

Name
Pedro

Last Name
Scott

Age
27

Email
scott7462@gmail.com

Key Considerations

Data Persistence:

This app manages a database based on realm, where the results of recent calls to the server will be saved and will handle all the data necessary for the user to perform all actions of the app, and is performing workouts or create new ones. Later the app server updates information.

Libraries to use:

- Realm: This library is for the database manager of the app.
- Glide: glide With the app can load images for the network.
- Rxjava: With the rx java app the manager can process in the background threads.
- Retrofit: This library can manager the connection to web services.
- Data Binding: This library for inject views and create relation of model and view.
- Timber: With the app can stamp the logs from manager debug and release.
- Joda-Time: This utility can help to manage the objects to date reference.
- RxPermissions: This app is used to manage the logic With rx permission java.

Task 1: Project Setup

The MVP architecture of the project and the initial project organization will be created:

- Start project for android studio to have the basis of the project structure.
- Create package functionality, where each fragment, activity or adapter it is stored.
- Create the package and major classes of presenters to communicate the views with the models.
- Presenters create data for communication between views and data handler.
- Create Busprovider with rxjava observers.

Task 2: Create data models.

Creating persistence management classes and communication with the server:

- Create classes of data models (Pojos) for the project logic.
- Create classes and interfaces for handling server connection with retrofit and rxJava.
- Create class for handling preferences.
- Create class for handling database.
- Creating the Data Manager for communication between presenters data and different data sources within the app.

Task 3: View and Navigation.

Construction of XML layouts, basic interaction and navigation app:

- Building XML layouts for launcher Screen.
- Building XML layouts for the Login Screen.
- Building XML layouts for the Register Screen.
- Building XML layouts for Drawable Navigation Screen.
- Building XML layouts for Home Screen.
- Building XML layouts for the Create Workout Screen.
- Building XML layouts for the History of workouts Screen.
- Building XML layouts for the History Details Screen.
- Building XML layouts for the History Details Screen.

- Building XML layouts for the Create Workout Screen.
- Building XML layouts for the List of Exercises Screen.
- Building XML layouts for Exercises Details Screen.
- Building XML layouts for Workout Do Screen.
- Building XML layouts for Workout Result Screen.
- Granting the interaction of all views to run the complete nevagion of the app.

Task 4: Connection and Web services functionality.

Creating methods for connecting web services:

- Development of web service user registration.
- Web development service Login using different methods, Facebook, Google Plus or email.
- Developing web service to get user profile.
- Developing web service to get list of preset workouts.
- Developing web service to get list of user training.
- Developing web service to get list of exercises.
- Development of web creation service training.
- Web development service training send results performed.
- Developing web service to get workout history.
- Development of web editing service training.
- Development of web service user profile editing.

Task 5: Construction management functionality database.

Construction of methods for handling database within the app:

- Creating methods for access to user data.
- Creating methods for data access Workouts.
- Creating methods for data access Exercises.
- Creating methods for handling data created so offline.

Task 6: Creating business logic in data presenters.

Development of methods and classes for handling business logic for each screen:

- Rx develop through all interactions between presenters and data models, with the implementation of the observables necessary for iteration with every view of the app.

Task 7: Integration apis.

Operating configuration and construction of external apis app:

- Integration of Google Plus and Facebook for login.
- Google analytics integration in the project.
- Google integration abs.

Task 7: Integration apis.

User testing and automated test creation:

- Implementation of unit tests for the operation of data models.
- Testing in real scenarios.