Developer Manual - *FitnessTracker*

Latest revision: 08-06-2014

Getting started

git clone https://github.com/matbern/FitnessTracker

Dependencies

- Java 6 SE development environment
- Android SDK
- A (virtual) Android device

Android SDK targets

- Minimum SDK: 16 (Jelly Bean)
- Target SDK: 19 (Kitkat)

Building and installing

After cloning the git repository import the project to Eclipse through File > New > Project > Android > Android Project from Existing Code, then navigate to the cloned git repository and import the FitnessTracker project. To build the project run the code on a (virtual) Android device with at least API level 16 (first release of the Jelly Bean Operating System).

Developing

FitnessTracker uses GitHub as revision control. Due to the modular nature of the application development takes place in a separate branch for each developer.

Branching

When developing a new feature this is done in a personal development branch. Should the feature not interfere with already existing code the branch can then be merged with master without notifying other developers. When merging with master, always pull the master branch to your development branch before pushing.

Should the feature alter already functional code no merge with master will be done before consulting the other developers.

Definition of “Done”

A document defining the project’s definition of “Done” can be found in the Documentation folder in the project root.

Release procedure

Releasing a new version of FitnessTracker can be differentiated between two cases.

Minor release

Minor releases include fixes and updating of the internal logic of the application and they don't directly introduce new functionality. Every minor release should include information about bug fixes and updated functionality.

Major release

Major releases introduce new functionality to the application and should as such be tagged as a new version upon release. A changelog is to be included containing

- New features,
- Changed features,
- Removed features,
- Known issues.

Testing

Due to the modular nature of the application most testing is done as manual black-box and white-box tests since independent features are added all the time. A test suite is included in the project, concerning the database, which can be further built upon to include test cases for new internal logic that might be introduced later.

Architecture

The application code is organized by area. Since the application currently only uses the device's internal storage there are three main packages; database, slidingmenu and fitnesstracker.

Database

Contains code related to setting up and handling the internal SQLite database used by the application.

SlidingMenu

Contains code related to the main menu of the application.

FitnessTracker

Contains the main activity as well as all additional fragments used to access the different functions of the application.

Future development

The goals for the future are separated into two categories; application and testing goals.

Application goals

The biggest goal for the future is to allow users to share their statistics with others by introducing a client/server protocol to the application. With all vital data stored in an SQLite database this should be achievable given the time required to do so.

The GUI will always have room for improvements and additional features will be added until the application reaches its completed stage.

Testing goals

Setting up a solid test suite for the internal structure of the application is something we want to achieve. Manual testing will be difficult to escape with how the application works but limiting this to the bare minimum is the biggest goal for the future related to testing.