System design document for Blue Java

Table of Contents:

Version: 1.3.3.7

Date: 140515

Author: David Bergström
Marcus Trigell
Tobias Andersen
Axel Niklasson

This version overrides all previous versions.

1 Introduction

1.1 Design goals

The design should follow Android-programming standards.
The design should be easy to modify i.e. to add further functionality.

1.2 Definitions, acronyms and abbreviations

- GUI, Graphical User Interface
- API, Application Programming Interface
- Android, Mobile operating system
- Java, Platform independent programming language

2 System design

2.1 Overview

The application will use Android standards, resulting in a MVP (model view presenter) pattern. General use of Android standards and functionality provided with the Android SDK.

2.1.1 API - themoviedb.org

All data is collected from themoviedb.org's API by several HTTP requests. All queries used to send the requests are constructed on the fly, by a class representing the API.

2.1.2 Database

Movies marked as favorites are stored in a SQLite database on the phones internal storage.

2.1.3 Layouts

Android provides a straightforward XML vocabulary that corresponds to the View classes and subclasses. Thus the layouts are built in XML, connected to View classes (Java).

2.1.4 Internal representation of values

All texts and default values are stored in XML files. This allows for easy translation and modification.

2.1.5 Event handling

All events are handled by Android activities. An activity is a single, focused thing that the user can do. To start new activities, intents (representing the intention of the new activity) are created. Activities take care of creating a window, which then can be filled with a layout.

2.2 Software decomposition

2.2.1 General

System consisting of packages:
Activities, Adapters, API:s, Fragments, Helpers, Interfaces, Models, SQLite
See Fig 2 in appendix.

2.2.2 Decomposition into subsystems

Adapters, extending the BaseAdapter class, are used to fill various lists in different views. Instead of creating a list component for each item, the list components are reused to save memory.

Inflaters are used to insert and update views on the fly.

WiFi Services from the Android system are used.

2.2.3 Layering

All related classes used to get data from themoviedb.org could be seen as a layer. Another set of classes that could be seen as a layer include the sorting classes.

2.2.4 Dependency analysis

Android SDK is the base plattform.
Picasso is used to help show pictures in various situations.
Facebook SDK is used to help with all Facebook-related functions.
See Fig 1 in appendix.

2.3 Concurrency issues

The UI runs on single thread. All data collecting are done through asynchronous HTTP requests, using Android AsyncTasks.

2.4 Persistent data management

All persistent data are stored in an SQLite database on the phones internal storage. The only table used is called movie and stores basic details, such as ID, title, rating, vote count etc.

2.5 Access control and security

NA

2.6 Boundary conditions

NA

3 References

1. Android Developer, see http://developer.android.com
2. MVP pattern, see http://en.wikipedia.org/wiki/Model–view–presenter

APPENDIX

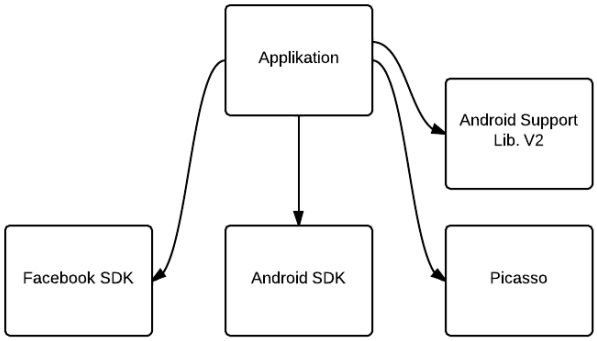

Fig 1. Dependancy analysis model

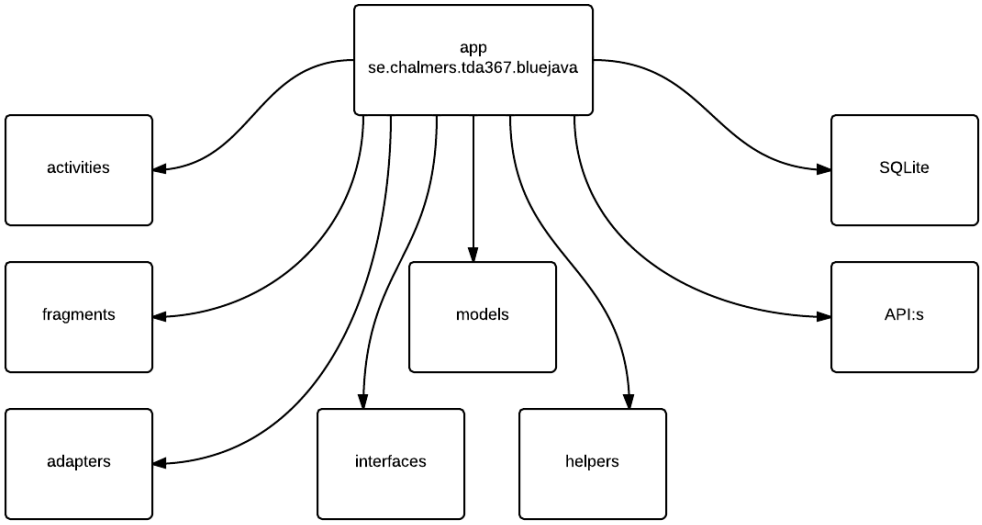

Fig 2. Main package diagram