Software Design Document

Grupp 22

Adrian Lindberg
adrlin@student.chalmers.se
Institutionen för Informationsteknik

Jonatan Nylund
nylundj@student.chalmers.se
Institutionen för Informationsteknik

Jonathan Sundkvist
jonran@student.chalmers.se
Institutionen för Informationsteknik

Sebastian Nilsson
sebnils@student.chalmers.se
Institutionen för Informationsteknik

May 27, 2016

1 Introduction

1.1 Design Goals

The application in it's core is a database. It's an app containing movies and users, where the user saves his movie ratings. Since ratings and movies are data that will be shared with all users; the entities needs to be stored. Which means that the model has to be integrated with a database layer.

The main goal of this application is to integrate the domain model with a database and with an object oriented approach. The system needs to be scalable, there should be no problems adding a new entity or a new type of database.

1.2 Definitions, acronyms and abbreviations

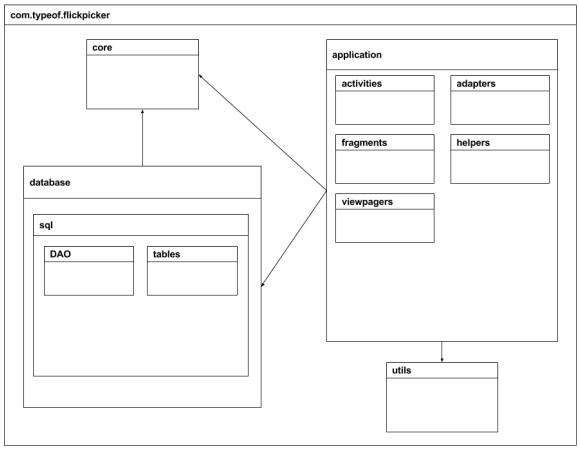
- DAO - Data Access Object
- XML - eXtensible Markup Language
- Android - Mobile operating system by Google
- Activitiy - Androids Controller
- Fragment - Androids Controller
- SQL - Structured Query Language
- SQLite - SQL Database Engine

2 System Design

2.1 Software decomposition

The design is split into four main packages; application, core, database and utils.

2.1.1 General



2.1.2 Decomposition into subsystems

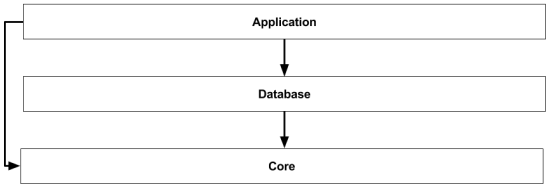
The core package contains the classes that constitutes the applications business logic, which consists of container classes that represents entities in the domain.

The application package is where the Android application resides. The classes describe what activities the app consists of. Activities is connected to layout files written in XML and represents what the user see and interact with in the app.

To populate the user interface with content like movies, users, ratings etc. the application package uses the database package. The database package consists of an interface layer that provides methods for reading and writing to the database. These methods uses entities from the core package to save and retrieve data to and from the database. The interface layer does however not specify what kind of database that's being used.

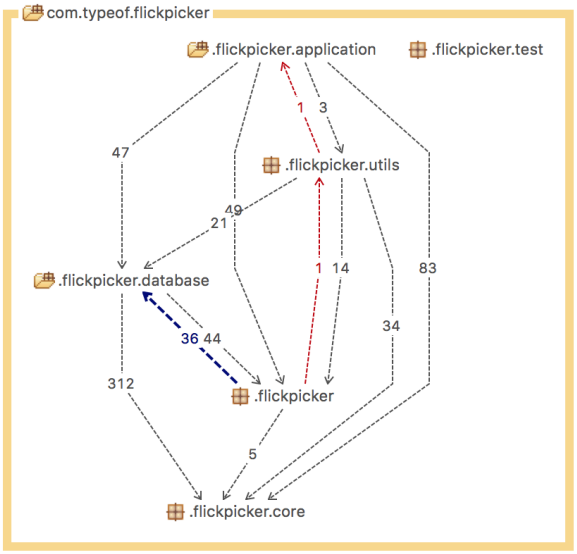
What kind of database is being used is decided in the AndroidManifest.xml file, which is a configuration file for the app. When the application starts the main application class, App, reads the manifest and creates instances of the type given by the manifest.

2.1.3 Layering



There are three main layers in the application. The application layer is the top layer and handles user interaction. This is where the Android application lives. The application layer uses the container classes in the Core when querying the Database layer.

2.1.4 Dependency analysis



2.2 Concurrency issues

The application sometimes loads large amounts of data from the database. The data is painted onto the screen, and this is preformed on the main thread. This leads to a loss in responsiveness for some of the views in the app.

Some of the calculations preformed by the algorithms in the application could have been placed in a secondary thread in order optimize performance on the main thread.

2.3 Persistent data management

The app is very much centered around persistent data in form of a database. The database is stored locally on the users phone. It's an SQLite database and the query language is SQL. The reason for the usage of SQLite and the decision to store the database on the device comes from the standard database API that comes with the Android framework.

2.4 2.5 Access control and security

N/A

2.5 2.6 Boundary conditions

The application needs the Android Operating System to run. The application can be installed onto a device with the Android SDK Platform Tools. An alternative is using the Android Studio IDE, which builds and installs the application automatically.

3 References

Android Activities
`http://developer.android.com/reference/android/app/Activity.html`

Android Fragments
`https://developer.android.com/guide/components/fragments.html`

SQL - Structured Query Language
`https://sv.wikipedia.org/wiki/Structured_Query_Language`

SQLite - SQL Database Engine
`https://www.sqlite.org/about.html`