

Requirements and Analysis Document for BlueJava

Table of Contents

Version: 1.3.3.7

Date: 14xxxx

Authors: David Bergström
Marcus Trigell
Tobias Andersen
Axel Niklasson

This version overrides all previous versions.

1 Introduction

This section gives a brief overview of the project.

1.1 Purpose of application

The project aims to provide an Android application for information about movies. It should be easy to find the movie that is intended to find and it should go fast. A stable, fast Android application is the main goal.

1.2 General characteristics of application

The application will be an Android application. It will use the phones network connection to get the information from themoviedb.org and present it in a structured and

1.3 Scope of application

The application will not store any information in the local storage of the phone. All information about the movies will be fetched in real time and therefore an active Internet connection is a must in order for the application to work.

1.4 Objectives and success criteria of the project

You should be able to:

- find movies using search and browse features
- filter and sort search results
- see detailed information about a movie

1.5 Definitions, acronyms and abbreviations

- GUI - Graphical User Interface

2 Requirements

In this section we specify all requirements

2.1 Functional requirements

The user should be able to:

1. Search for movies
2. Browse movies
3. Show details about a movie
4. Show user profile.
 - a. Recommendations
 - b. Save favorites
 - c. Settings
5. Leave the app and continue where he/she left off

2.2 Non-functional requirements

Usability is high priority. Users should be able to easily find their way around in a well designed app.

2.2.1 Usability

Usability is very high priority. Since it is a smartphone application, it needs to be fast and easy to use. It should be easy for the user to pick up the phone and search for a movie and avoiding unnecessary steps is very important. It should be possible to find given movie within seconds and quickly find the information he/she is looking for.

2.2.2 Reliability

The app needs to be stable and not crash spontaneously.

2.2.3 Performance

Mobile applications need to work fast and any actions that is initiated by the user should be completed in least amount of seconds possible.

2.2.4 Supportability

The application must be implemented so that it is possible to use it on various phone models and get the same user experience. There should be automated tests varifying all use cases and code related to the graphical user interface should be tested manually. However, the testing regarding the GUI should be included in the final documentation.

2.2.5 Implementation

To achieve the mobile phone model independence the application will be use the Java environment along with the Android SDK. Therefore, in order to use the application, the user must run it on an Android phone.

2.2.6 Packaging and installation

The application will be delivered as an Android application:

1. An APK file.

2.2.7 Legal

NA

2.3 Application models

2.3.1 Use case model

See appendix for UML diagram and text descriptions.

2.3.2 Use cases priority

Use cases are prioritized in the following order.

High:

- Search movies
- Sort
- Browse movies
- Show detailed view
- Rate movie
- Show credits

Mid:

- Search TV-shows
- Watch trailer
- Change settings
- Login via Facebook
- See soundtrack
- See Twitter movie feed
- See Twitter actor feed
- User profile, Recommendations
- User profile, Save favorites
- Klick "Seen this movie"
- Rate movies

Low:

- View a random movie
- Cinema nearby
- Chat

2.3.3 Analysis model

See APPENDIX.

2.3.4 User interface

Application will consist of a standard Android GUI. The GUI needs to be able to adapt to various screen resolutions, since Android is available on many different phones. The GUI also needs to consider very small screen sizes, so the information architecture must be well

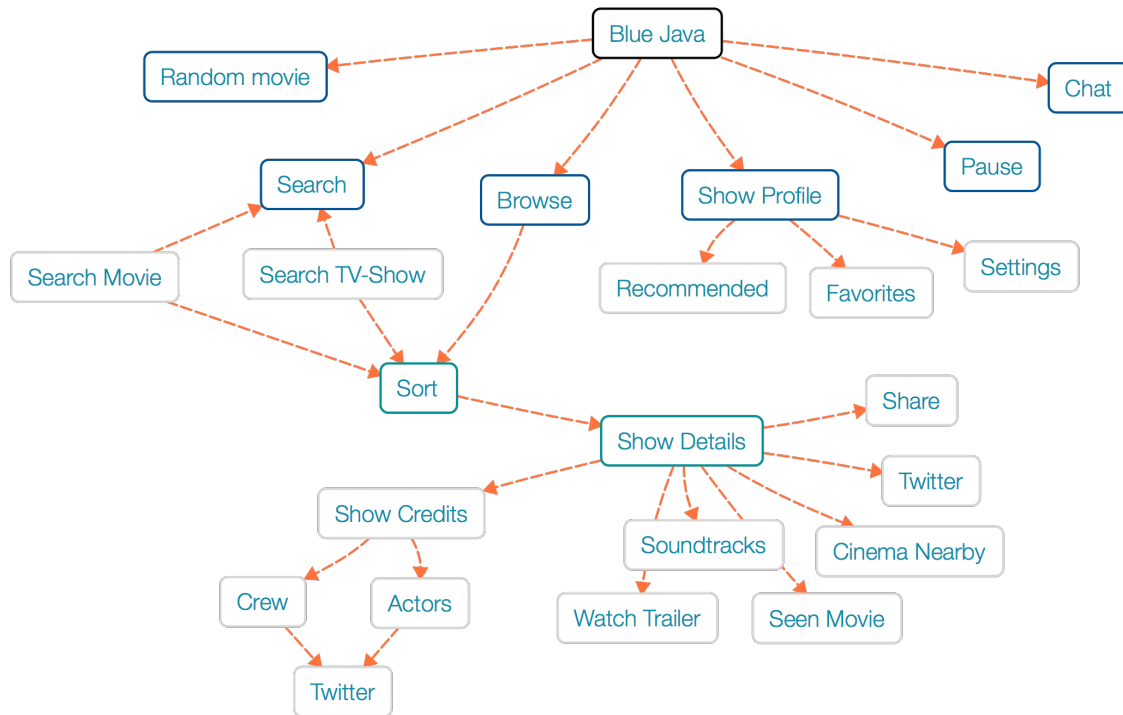
designed. As for “look and feel”, the standardized Android version will be used.

2.4 References

APPENDIX

Use Cases

Overview



Use Case Texts

See attachment 1.

GUI



Analysis model

