Movie Center Android application
Functional Specification

Anas Alagtal
14511433

30/11/18

Functional Specification Contents

0. Table of contents

A table of contents with pages numbers indicated for all sections / heading

1. Introduction

1.1 Overview: Watching movies is one of the most common form of entertainment. This activity can be enhanced with the presence of an application that displays all the necessary information regarding all past movies and the upcoming ones too including the name, plot, trailer, reviews, ratings etc. That is what my “MovieCenter” android application provides which is very similar to “IMDB” except for the simplicity of UI my application provides as well as a built in recommender system to recommend user movies it thinks he/she may like to watch based on the users information, what he/she rated well, searched, reviewed etc. My application will connect to an online database that contains new/updated information that goes through the server. User information like login details and information about movies they have watched or reviewed etc will be stored in the database. Users can log in by registering through the app or registering using their existing google account. The recommender system, which will be written in python will read the information regarding the user from the database to be able to recommend movies specifically to the user.

1.2 Business Context: This application is targeted really to a market involving anyone who is old enough to use a smartphone and has any interest in movies. As said previously it will enhance any users movie watching experience. It is an android application so can only be used on android smartphones by downloading from the google play store.

1.3 Glossary:

API: *“a set of functions and procedures that allow the creation of applications which access the features or data of an operating system, application, or other service.”*

Android studio: *“Android Studio is the official Integrated Development Environment (IDE) for Android app development“*

The movie db: *a third part website the offers an api which allows access to a full range of movies database and information regarding the movies*

2. General Description

2.1 Product / System Functions:

Once the application is downloaded and installed into the android device from the google play store the user then launches the application and chooses whether to register using his/her email and entering their details manually or choosing the login with google email option. Once registered, the user is logged in and their user information is recorded into my database. They can then use any feature within the application. Whether that is to search a movie and they like to see reviews/rating, trailer, whether to give their own reviews/rating or whether to see a list of recommended movies based on what they have previously enjoyed.

2.2 User Characteristics

The user needs to have basic knowledge of how to install an android application from the google play store. The entire application including title, plot, trailers etc will be in English so they need to be familiar with the language. They would need to have a personal email or a google account to be able to register or log in. from then on everything should be straightforward.

2.3 User Objectives

The objective from the user point of view would be a simple app that would display movies in an organised and clear manner. Information regarding any movie would be easily attainable and a search bar that allows them to search for a preferred movie.

2.4 Operational Scenarios:

2.4.1 Registration/ login Unsuccessful:

Current System state: The app is waiting for a user to enter details
Informal scenario: The user enters incorrect credentials for email/ google registration or for logging in.
Next Scenario: Application alerts the user that information is incorrect and asks to enter again

2.4.2 User registers:

Current System state: The app is waiting for a user to enter details
Informal scenario: The user enters correct credentials
Next Scenario: The application user interface is accessed and user information is saved.

2.4.3 Clicking tab:

Current System state: App is on “upcoming movies tab” and waiting for user to click an option.
Informal scenario: The user clicks on popular movies tab.
Next Scenario: Application displays information and awaits user.

2.4.4 Searching incorrect Movie:

Current System state: App is displaying popular movies and waiting for user to click an option.
Informal scenario: The user types in the search box an inexistant movie name.
Next Scenario: Application displays an error page saying no results for search.

.

2.4.5 Searching Movie:
Current System state: App is displaying popular movies and waiting for user to click an option.
Informal scenario: The user types in the search box a movie name.
Next Scenario: Application displays movies that contains the name searched in the title of the movie.

2.4.6 Logging out:
Current System state: App is displaying searched movies and waiting for user to click an option.
Informal scenario: the user chooses to click the logout button.
Next Scenario: Application logs the user out and displays login section again.

2.5 Constraints

Data: The recommended feature of the application requires the user to enter a sufficient amount of rating and reviews etc for movies to be able to recommend a good range of movies that are within the users interests.

Time: the applications recommender system will be very difficult to implement in the time given between when the functional spec is due and when the project is due along with other modules and exams

Encryption: User passwords will need to be encrypted which is a learning curve for me as I've never done it before.

Device: This application is built is android studio meaning that it can only be used on android smartphones or devices. They would be therefore required to have a google account to be able to install from the google play store

3. Functional Requirements

3.1. The movie database API service:

Description:
This refers to the API of the of the “The movie db” from which I would be getting my movie information.
Criticality:
Very critical. They entire application rests on the success of the api. There would be no content to show within the application if there was no API.
Technical issues:
Technical issues could arise from The Movie db’s end which could interfere with my application
Dependencies with other requirements:
The API depends on the UI of the application. The movie information that is receives depends on the UI to showcase it.

3.2. Storing user information:

Description:
This refers to the information of the users who register to be stored in a database, alongside all of their likes, dislikes, rating, reviews etc.
Criticality:
Very critical. User information needs to be stored so that the app can tailor recommendations to each unique user.
Technical issues:
Storing user information requires password encryption which could fail depending on which database I decide to use which I still have not figured out yet
Dependencies with other requirements:
Depends on the database that I use and the log-in functions of google.

3.3. User Interface/Graphics:

Description:
This refers to the graphics of the application and the ease of use in terms of seeing/ choosing the options. Clicking a movie should bring them to a screen where the movie is displayed alongside plot, ratings, trailer etc. Creating clear easy to see options will take some research into design of the app and colour usage so that the contrast of colours doesn’t provide difficulty for the user. Graphics also must be done in the best way to ensure user satisfaction and so that they don’t use competitor’s applications.
Criticality:
User interface is always critical. If an app was not user friendly an easy to use and navigate then it would have little to no users. UI is critical also with my app. Movies need to be displayed at a seperate contrast to the background. Movie options need to be of clear text size, font and colour. The actual functionality of any app is irrelevant if the app is not user friendly.
Technical issues:
Good amount of time will be put into research to ensure there is little to no technical issues within the UI of the application. With good preparation and planning I will have little technical issues when it comes to user interface.

Dependencies with other requirements:
The UI will mainly depend on what version of android studio I use. But Ii will be using the newest so that shouldn't be a problem

3.4. Android version:

Description:
The application requires the user to have an android device because it is an android application.
Criticality:
Very critical
Technical issues:
The user needs to be running an android version that is compatible with my application which will be detailed in the user manual.
Dependencies with other requirements:
None

3.5. Internet Connection:

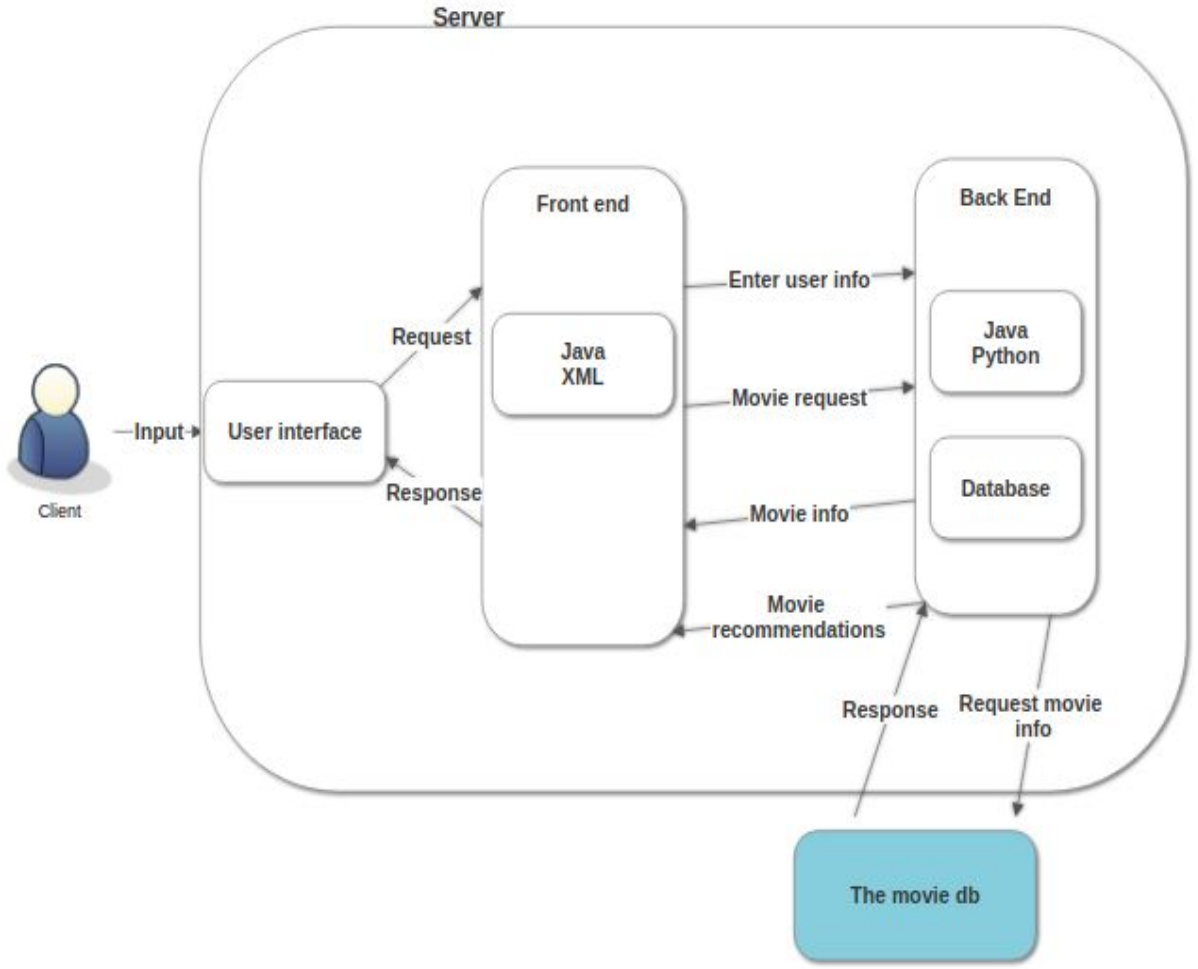
Description:
The user must have internet connection to be able to use the application as all the movie information is retrieved through an API. Whether that is through mobile network or WiFi.
Criticality:
Very critical
Technical issues:
Without an internet connection the application is not able to showcase and movie or movie information rendering the application useless
Dependencies with other requirements:
None

3.6. Recommender System:

Description:
This is the feature that recommends movies to a user based on their preferences.
Criticality:
Important
Technical issues:
The algorithm parameters must be give accurate weightings and must be taught well in order to give accurate recommendations
Dependencies with other requirements:
The database containing all the users information.

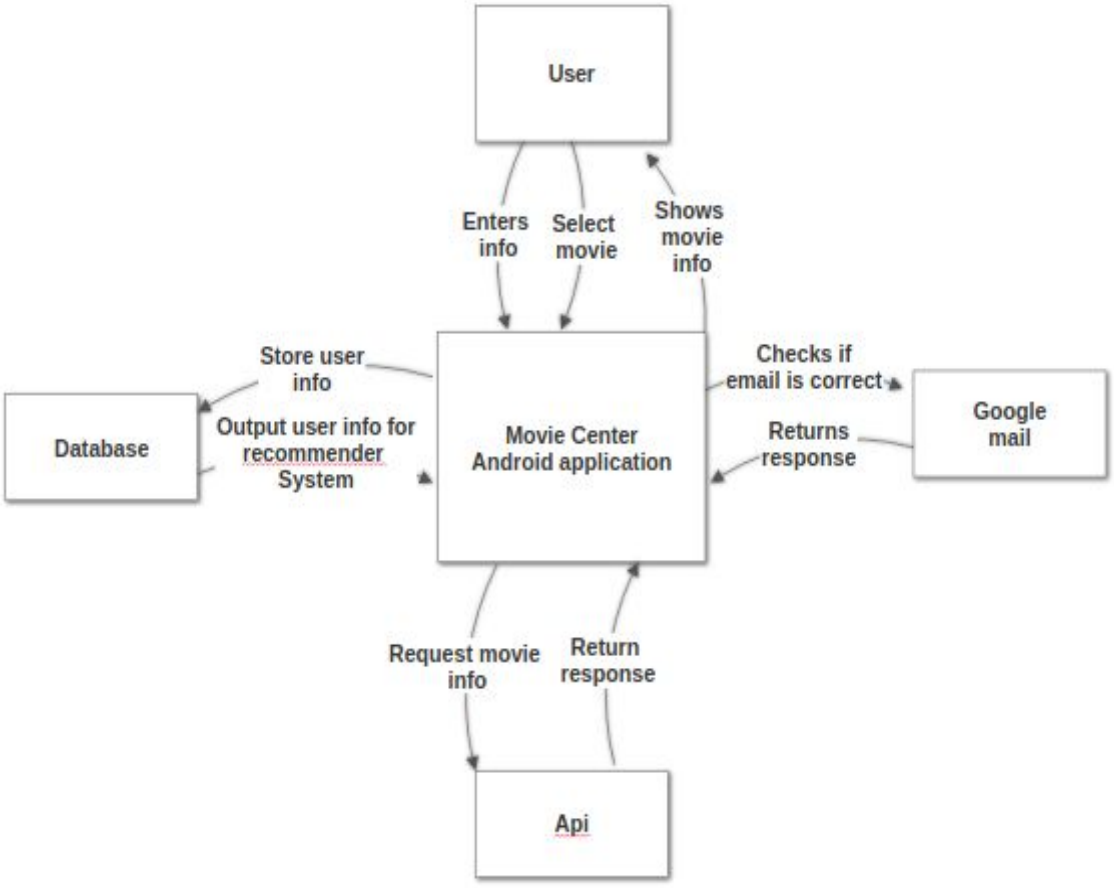
4. System Architecture

This section describes a high-level overview of the anticipated system architecture showing the distribution functions across system modules.



5. High-Level Design

5.1 Context diagram:



6. Preliminary Schedule

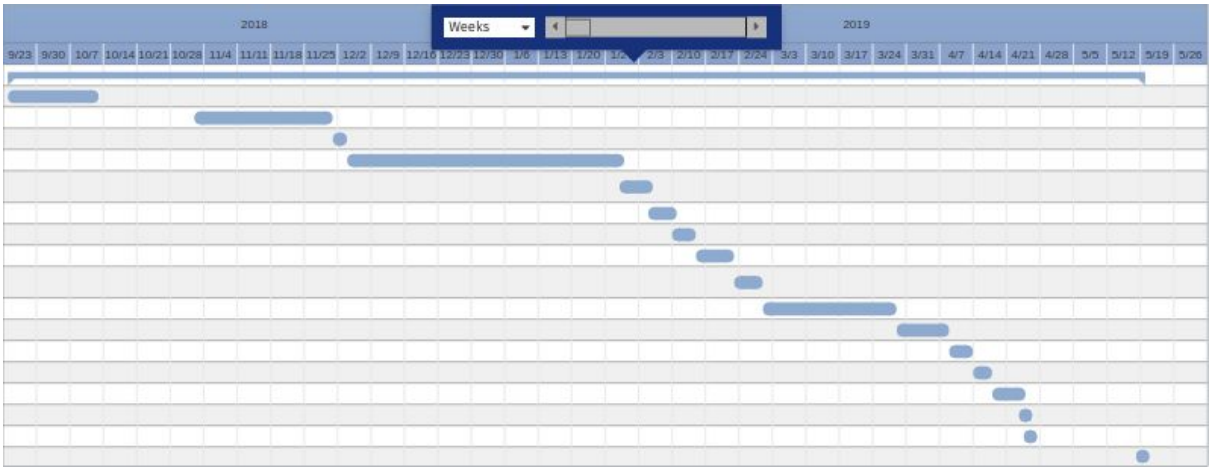
This section provides an initial version of the project plan, including the major tasks to be accomplished, their interdependencies, and their tentative start/stop dates.

6.1 Preliminary Schedule

Task	Dependencies	Start	End	Duration
Project		9/24/18	5/19/19	164
Submit Proposal		9/24/18	10/12/18	14
Functional spec	1	11/2/18	11/30/18	20
Template Android application		12/1/18	12/3/18	1
User interface layout With exam study	3	12/4/18	1/30/19	39
API call to the movie database to retrieve movie info	3	1/30/19	2/5/19	5
Set up user log ins	3	2/5/19	2/10/19	4
Set up server		2/10/19	2/14/19	4
Set up database		2/15/19	2/22/19	5
Connection to the database and server from the app	7,8	2/23/19	2/28/19	4
Set up recommender system		3/1/19	3/28/19	20
Train and test recommendations	10	3/29/19	4/8/19	7
Ensure success of connections	7,8,3	4/9/19	4/13/19	4
Bug fixes and improvements		4/14/19	4/17/19	3
Testing		4/18/19	4/24/19	5
User Manual		4/24/19	4/25/19	2
Technical spec		4/25/19	4/26/19	2
Video walkthrough		5/19/19	5/20/19	

6.2 Gantt chart

The project plan is accompanied by the following GANTT chart.



7. Appendices

Specifies other useful information for understanding the requirements.

7.1 Books :

Android Programming: The Big Nerd Ranch Guide (3rd Edition)
Recommender Systems: The Textbook. Authors: Aggarwal, Charu C.
The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform

7.2 Videos:

https://www.androidhive.info/2016/10/android-working-with-firebase-realtime-database/

https://www.youtube.com/watch?v=EM2x33g4syY

7.3 Links:

https://aws.amazon.com/ec2/getting-started/

https://firebase.google.com/docs/database/android/start/

https://medium.com/recombee-blog/machine-learning-for-recommender-systems-part-1-algorithms-evaluation-and-cold-start-6f696683d0ed