SE 3XA3: Software Requirements Specification
MovieGuide

Team 35, PGH software solution
Hamid Ghasemi ghasemih
Pratyush Bhandari bhandarp
Gazenfar Syed syedg1

October 5, 2018

Contents

List of Tables

List of Figures

Table 1: **Revision History**

Date	Developers	Change
Oct 3	Hamid, Gazanfar	Non-functional requirments, Functional requirements
Oct 5	Pratyush, Hamid	Project Drivier and Project issues, Filling latex file

1 Project Drivers

1.1 The Purpose of the Project

In our world, time and money are the two most valuable things to people. These days, people are working longer hours and free time is very valuable. As such people are looking for something worth-while to do in their free time. A very popular pass-time is watching movies, whether it be going to the theatres or sitting at home on Netflix; the average run-time of a movie is close to two hours and people need to be informed on whether watching a certain movie will be a good investment of their time. PGH Software Solutions aims to solve this problem by developing an application that allows users to easily obtain information such as trailers, rating and synopses of their favourite movies to make an informed decision about how to spend their time.

1.2 The Stakeholders

1.2.1 The Client

The stakeholders of this project include the client, customer and the developers of the project. The main clientele of this project are movie theatres and movie streaming services who have an interest in boosting their user satisfaction. With our application movie watchers will be able to better make the right decision when watching a movie thus increasing user satisfaction at movie theatres or movie streaming services.

1.2.2 The Customers

The main customers of this project are people with a busy lifestyle, who enjoy watching movies in their pass-time. As well as our main intended customer base, regular users will also find the application helpful.

1.2.3 Other Stakeholders

The last stakeholders of this project are the software developers building the application, because the success of this app can potentially bring large earnings to those building the application.

1.3   Mandated Constraints

Description: The product shall operate on a mobile operating system
Rationale: The product is marketed for busy customers who will find it to
be a burden to use their laptops to find a suitable movie to watch.
Fit criterion: The product shall be compatible with most android devices

Description: The product shall use an API to fetch data from a server
Rationale: Databases are too big to feasibly store on the application
Fit criterion: The product will fetch a few movies from the server at a once.
This will allow users to find the movies they want without using too much
storage.

Description: The product shall run only when connected to the internet
Rationale: Movie information needs to be collected from the online database
which is not possible without an internet connection.
Fit criterion: The product will only run if their internet connection is suffi-
cient enough to run the application.

1.4   Naming Conventions and Terminology

None.

1.5   Relevant Facts and Assumptions

The program we are referencing contains over 2000 lines of Java code

User characteristics should go under assumptions.

2   Functional Requirements

2.1   The Scope of the Work and the Product

2.1.1   The Context of the Work

Our main task is to recreate the MovieGuide open-source Android applica-
tion whilst fulfilling the functional and non-functional requirements. The

work has been divided into weekly deliverables to manage time effectively. During the first few weeks, the deliverables involved analysis of the source code, problem identification, and development planning. These deliverables helped establish issues and obstacles that we may experience during our implementation. After outlining the requirements, we will be ready to produce a proof of concept. Lastly, we will formulate a test plan to ensure our proof of concept works as expected and document the entire development process.

2.1.2   Work Partitioning

Our development process will be partitioned into two major steps: logic and user interface. The logic will be broken into implementation of the movie API, redirection to YouTube for trailers, searching for movies, and sorting movies by name or rating. The user interface will be divided into development of the home page, organization of information provided by the API, and graphics/animations. All of these steps will be completed using the Android Studio Editor.

2.1.3   Individual Product Use Cases

Searching:
The user may search for a specific movie by using the search bar provided in the user interface. The program will then take the user input and search for the name within the list of movies. If there is a search hit, the movie along with its information (provided by the API) will be displayed to the user. If there is a search miss, the program will display a message to the user notifying them that there was no such movie found.

Sorting:
The user may choose to have the movies sorted by name, genre, or rating. The program will sort the movies based on the users selection and display the movies back to the user in sorted order.

Watching:
The user may watch trailers for movies by clicking on the video thumbnail. The program will redirect the user to the corresponding video on YouTube so that they may view the trailer. The movie application will run in the background while the user watches the trailer on YouTube or a browser of

their choice. This means that the application does not close upon redirection.

2.2   Functional Requirements

Requirement Number: FR1
The software shall provide a synopsis of every movie along with its rating and a video trailer
Rationale: If the user cannot gather information about the movie, the application is not performing its fundamental purpose.

Requirement Number: FR2
The software shall retrieve movies based on query passed by user into search bar.
Rationale: Allows user to gather information about a specific movie.

Requirement Number: FR3
The software shall notify the user if there are no movies that match the search query.
Rationale: User should know if there is a search miss, so that they do not wait for the program to output a result.

Requirement Number: FR4
The software shall redirect user to trailer when the trailer thumbnail is clicked on.
Rationale: Allows user to gain a deeper insight to what the movie is like.

Requirement Number: FR5
The software shall sort the movies according to user specification.
Rationale: Allows user to find movies by name, rating, or genre.

3 Non-functional Requirements

3.1 Look and Feel Requirements

3.1.1 Appearance Requirement

The PGH application shall exhibit the logo of the company in the starting point. It will hold a dark color in the background to make the picture of the movies more visible which gives the vibe of the movie before getting chosen. The application must fit the size of the mobile phones.

3.1.2 Style Requirement

The PGH application is a touchscreen program In order for program to make a strong relation with users. To choose each movie, user must tap on the movies designated place, then it will direct them to the summary, rating, and trailer of the movie.

3.2 Usability and Humanity Requirements

3.2.1 Ease of Use Requirement

The program shall be simple for any person with age more 8 years old.

3.2.2 Personalization and Internationalization Requirements

N/A

3.2.3 Learning Requirements

The program shall be able to utilize by any person and it shall operate in any android phone or device that has touchscreen ability.

3.2.4 Understandability and Politeness Requirements

The program shall be able to utilize by any person and it shall operate in any android phone or device that has touchscreen ability

3.2.5 Accessibility Requirements

The program shall be usable by partially sighted users.

3.3 Performance Requirements

3.3.1 Speed and Latency Requirements

The application must operate as soon as the user touches the screen, and it shall respond quick to the user. It shall respond less than 1 second.

3.3.2 Safety-Critical Requirements

The application shall not compromise the user data, and shall not interfere with phone operator system.

3.3.3 Precision or Accuracy Requirements

The application shall respond to user request swiftly, and it shall execute it precisely using integer.

3.3.4 Reliability and Availability Requirements

The application shall be available for all the users with android phone connected to internet.

3.3.5 Robustness or Fault-Tolerance Requirements

The program shall be able to operate without consideration of physical state of the end user.

3.3.6 Capacity Requirements

The application shall not overload the server.

3.3.7 Scalability or Extensibility Requirements

The code can be changed during increased size of the product

3.3.8 Longevity Requirements

the product shall function for life time in android phone.

3.4 Operational and Environmental Requirements

3.4.1 Expected Physical Environment

The application expected to be used anywhere for all the users with android phone that has access to internet, it shall be used in any environment that has android operating system.

3.4.2 Requirements for Interfacing with Adjacent Systems

The program shall be built for android phones though it can be operate in any device that has android operating system.

3.4.3 Productization Requirements

The product shall be of a size such that it can fit on phones RAM.

3.4.4 Release Requirements

The application shall alter yearly and it shall updated every day based on the new movies coming out, and for maintenance it shall get tested.

3.5 Maintainability and Support Requirements

Make maintenance of the program minimal. For supportablity, ensure the majority of users are able to run the software on their phone and users around the world can get access to it.

3.6 Security Requirements

3.6.1 Access Requirements

The program shall keep user information private and protect all the information which has been provided.

3.6.2 Integrity Requirements

The program shall not alter its source code. The product shall protect itself from intentional abuse.

3.6.3 Privacy Requirements

The application shall not store any data from users mobile phone without informing the user and shall not alter anything on the users phone without informing the user. The application shall not violate any rights from the movies, moreover the application shall not provide any wrong information about any movies.

3.6.4 Audit Requirements

N/A

3.6.5 Immunity Requirements

The program shall update at least every week to ensure that the software doesnt get virus.

3.7 Cultural Requirements

This product shall not be offensive to any religious or ethnic groups. This product will be available in English and it shall be hosted from the North American servers.

3.8 Legal Requirements

The application shall not violate any laws.

3.9 Health and Safety Requirements

This application shall adhere to the MIT Open License.

4 Project Issues

4.1 Open Issues

Our investigation into finding the correct number of movies to load at once from the online database is not yet complete.

Our research on how to do animations in java for an android application are yet to be done.

Our research in finding the optimal searching and sorting algorithms within the application are yet to be done.

4.2 Off-the-Shelf Solutions

A major existing product that can be referenced is the app we are basing our program on, MovieGuide. Many of the issues we are having can be resolved by taking a look at the implementation of the MovieGuide application.

4.3 New Problems

A major problem that our application may face is that users may believe that our application will ensure they end up watching a movie they like. Our application is only meant as a guide and cannot guarantee that a user will be satisfied with their movie choice a 100 % of the time. Another major issue that can arise is from our implementation of the API which gives our application access to the movie database. A larger growing database may make searching and sorting movies slower over time.

4.4 Tasks

Proof of Concept Implementation: Due Oct 15, 2018

The proof of concept will involve simply loading a list of movies from the online database using an API, and then displaying them on a phone screen. The purpose of this step is to see if we can get the essential backbone of our application working.

Java Design: Due Oct 30, 2018

The java design step will involve coming up with a list of required classes and functions needed to build the application. UML diagrams will prove to be very handy in this step.

Java Implementation: Due Nov 19, 2018

The implementation step will finally put all of the required functions and classes into one working project.

Above is a rough outline of the major tasks that need to be completed to bring this application a to a completion. A more thorough outline can be seen on our team Gantt chart.

4.5 Migration to the New Product

None.

4.6 Risks

The only major risk that has been foreseen is the possibility of ratings for current movies not being refreshed every couple of weeks. This will make the application less useful for a potential user.

4.7 Costs

None.

4.8 User Documentation and Training

None. The app should be designed in a way where a user can use the application without any instruction.

4.9 Waiting Room

None.

4.10 Ideas for Solutions

The application can try to read the network strength on a users phone and accordingly pick a certain amount of movies to load at once. This will allow phones with faster connections to run the application faster, but still allow phones with slower connections to use the application.

References

5 Appendix

5.1 Symbolic Parameters