Project Documentation

ISTE-330 Database Connectivity and Access

MoviePlanet

Document (filename): MoviePlanet

Author(s): Lučano Žganec, Patrik Đurasek, Bruno Mamić, Ivan Landeka, Ana Raguz

Date (last change): 20.11.2016

Version (last version): 1.3

|  |  |  |  |
| --- | --- | --- | --- |
| *Version* | *Description of Change* | *Author(s)* | *Date* |
| 1.0 | Creating the documentation | Ana & Ivan | 2016-11-5 |
| 1.1 | Description and Architecture Update | Lucano & Patrik | 2016-11-6 |
| 1.2 | Adding data dictionary and expanding business layer description | Ana, Patrik & Lucano | 2016-11-13 |
| 1.3 | Adding GUI description and preview | Patrik & Lucano | 2016-11-20 |
| 1.4 | Extending GUI description/user manual | Patrik & Lucano | 2016-11-27 |
| 1.5 | Extending the User Manual | Ivan & Ana | 2016-11-27 |
| 1.6 | installation | Ivan | 2016-12-03 |
| 1.7 | configuration and acceptance testing | Ana | 2016-12-04 |

Table of contents

[1 Introduction](#_gjdgxs)

[1.1 Overview](#_30j0zll)

[1.2 Purpose](#_1fob9te)

1.3 Scope

[1.4 Background](#_3znysh7)

[1.5 References](#_2et92p0)

[1.6 Document Overview](#_tyjcwt)

[2 Problem Description and Solution Architecture](#_3dy6vkm)

[2.1 Problem Description](#_1t3h5sf)

[2.2 Architectural Design](#_4d34og8)

[2.3 Database Layer and Database Abstraction Layer Description](#_2s8eyo1)

[2.4 Business Layer Description](#_17dp8vu)

[2.5 Presentation Layer Description](#_3rdcrjn)

[2.6 Areas of particular concern](#_26in1rg)

[3 Requirements](#_lnxbz9)

[3.1 Context](#_35nkun2)

[3.2 Functional Requirements](#_1ksv4uv)

[3.3 Other (Non-Functional) Requirements](#_2jxsxqh)

[4 User Documentation](#_z337ya)

[4.1 Graphical User Interface](#_3j2qqm3)

[4.2 User Manual](#_1y810tw)

[5 Installation, Configuration and Acceptance Testing](#_4i7ojhp)

[5.1 Installation](#_2xcytpi)

[5.2 Configuration](#_1ci93xb)

[5.3 Acceptance Testing](#_3whwml4)

[6 Final Remarks and Conclusion](#_2bn6wsx)

7 References

# Introduction

## Overview

*[Provide an overview of the project / application and additional information to place it in context – high level description of the domain and the problem and high level description of the desired functionality (deliverable 1).]*

This project will demonstrate an application with a collection of movies stored in a custom made database containing all information about movies themselves; including the year of release, budget, earnings, actors, directors, run time, and the overall grade of the movie, and offering a list to save the movies the users have seen to make their search easier. This application will have a GUI where a person will be able to search for a movie and get back all the necessary information about it, and then watch a trailer for it, if desired. The application will constantly communicate with the data-base which will be updated accordingly; upon release of a new movie. Movies will be sorted by Year, Genre and Name. They can be searched by keywords in the title. The purpose of this application is to make user’s lives easier by having a database of movies in one place, presented in a nice and interactive way.

## Purpose and Scope

*[Provide an overall description of the document and its purpose as well as intended audience. Discuss the scope of the document (deliverable 1).]*

The purpose of this document is to describe to the smallest details the scope of our project, the way the application will work, all parts that will be communicating together and the language it will be written in. The document will describe the design and all the parts of the database and their contents. These will all be explained as to what is their purpose, and for each part its intention. This document is intended for developers that wish to understand how the applications works and what its parts are, and contain required information for anyone who wishes extend the application features in the future. Also. any user that is interested in the details of the project could find all the necessary information in this document.

## Background

*[Describe who is producing the document and why (deliverable 1).]*

The document is produced by group members who are responsible for the design of the project and who have communicated with other team members and agreed upon the final idea before the design process starts. The document is produced to record the progress of the project from the very beginning to the end and final presentation/execution of the developed application.

## References

*[List references and other associated documents, including: documentation, standards, policies, deliverables, meeting summaries, white papers, etc. (deliverable 1)]*

We do not have any references at this point in time, but we have regular team meetings where we discuss our wishes and options in person and constantly work on developing our idea. We all give our suggestions and agree upon the best one in a fair vote.

## Document Overview

*[Provide a description of the document organization per chapters (deliverable 1).]*

The first chapter is an introduction to the overall issue we are dealing here with as a group and an explanation on why are we producing this project and who it is aimed at. The second chapter is focused on presenting the main “problem” of our project and our solution to the presented problem in form of the rough description/outline of the application, architecture of the software and the layers. Third chapter focuses on the requirements both functional and nonfunctional needed for the application. Chapter 4 describes the graphical user interface of our project and how to use it, also has a user manual for any users that are confused or stuck and need a solution to a common problem or a dilemma. Chapter 5 is focused on the prerequisites of the system which will run the application, the installation process and the configuration of the same one in order to see if the requirements have been met. Sixth chapter is our own personal reflection on our work, on the things we have done good and are happy with, the things we have done bad and could do better the next time.

# Problem Description and Solution Architecture

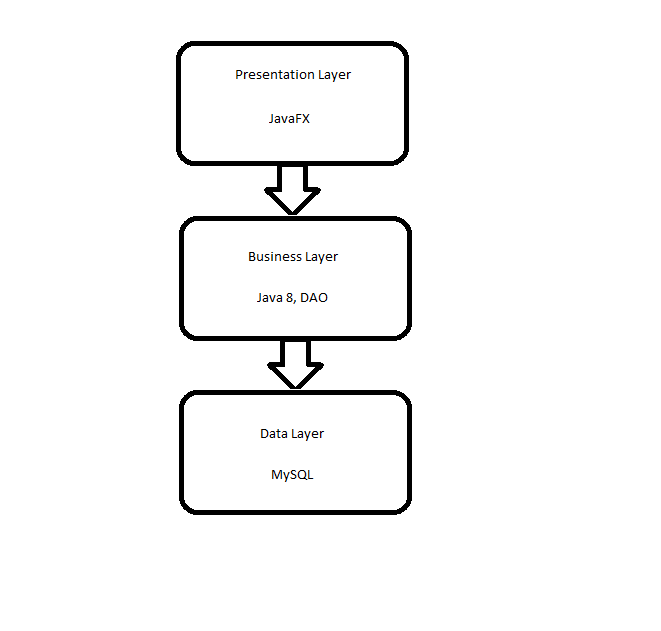
## Problem Description

*[Detailed description of the problem that will be solved (deliverable 1).]*

The “problem” which will be solved by producing this application is that people can access movie reviews all in one place, by looking for them by their name, their actors or some keywords, as well as being able to have the list of movies they have watched in the same place. People spend too much time trying to find a good honest description and review of the movie, and they often don’t remember if they have seen the movie they are currently reading about. The problem is that the information is usually spread among few sources and never composed into one in one place, this application will have it all in one. The purpose of the application is to have a nice friendly looking and easy to use interface which will attract the user and keep him entertained, and coming back.

## Architectural Design

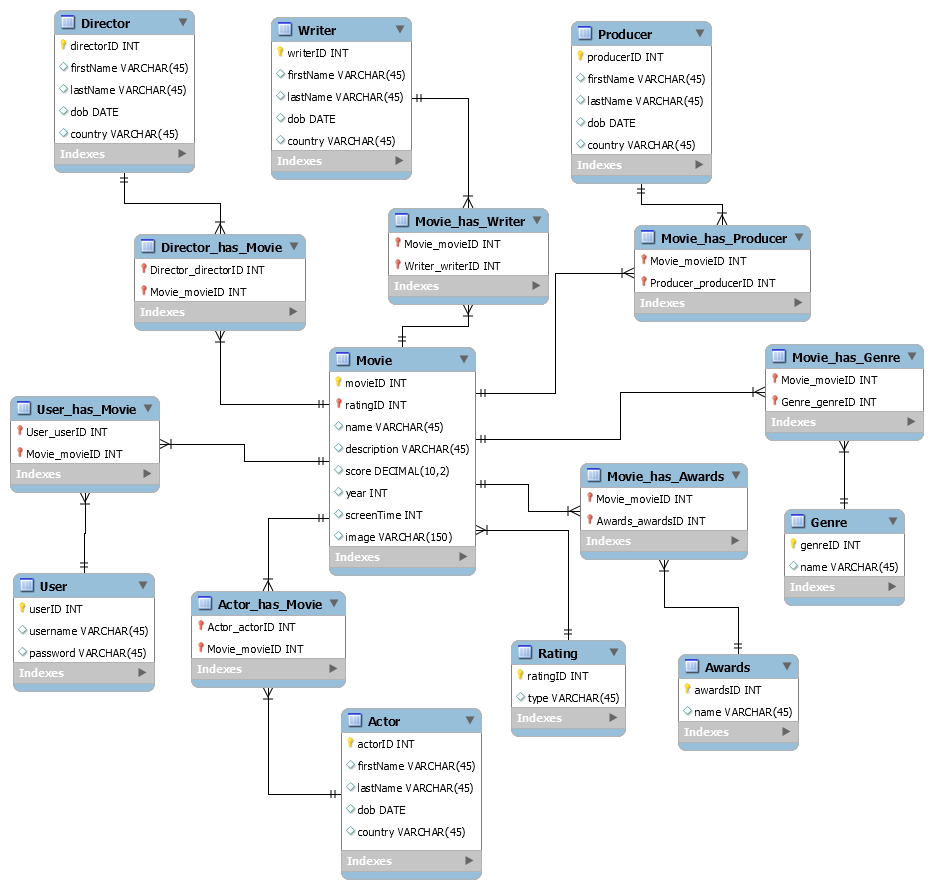
*[Capture important software design decisions (deliverable 1). Do not forget to explain the choice of language(s) and rationale for this design. Describe and sketch Software Architecture – decomposition of the software into layers/modules/units/components and provide detailed software design description for each part. ]*



For this solution we decided to use Java as our development language (with JavaFX platform for better looking GUI), since of its ‘write once, run anywhere’ concept, making our application portable. For database we will use mySQL. We will use the layered architecture approach, which will include a database layer, database abstraction layer, business layer and a presentation layer.

## Database Layer and Database Abstraction Layer Description

*[Provide database structure with ERD and database schema (physical), as well as data dictionary explanations of entities (tables) and their characteristics/attributes/properties (columns) (deliverable 1 and 2).]*



This is the database model that will be used for the application. Since the application is based on movies, movie table is set in the middle. Other tables, such as director, writer, genre, awards, and other are used for additional information provided as well as for various filters. User table is connected with movie table for creating the private list for each user that will hold the movies. Many tables are intersection tables due to many-to-many relationships.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Entity Descr.** |  |  |  |  |  |
| **Column name** | **Column descr.** | **data type** | **length** | **primary key** | **nullable** | **unique** |
| **Director** | A director of a specific or more than one movies |  |  |  |  |  |
| ID | His unique identification number | int | 11 | true |  | true |
| name | Name of the director | varchar | 45 |  |  |  |
| dob | date of birth | date |  |  |  |  |
| country | country of origin | varchar | 45 |  |  |  |
| **User** | A person visiting our site |  |  |  |  |  |
| ID | Unique identification number for a user | int | 11 | true |  | true |
| username | unique name for each user | varchar | 45 |  |  |  |
| password | unique combination of chars for each user | varchar | 45 |  |  |  |
| accessLevel | access level for a user | varchar | 45 |  |  |  |
| **Actor** | People that participated in a movie or more than one movie |  |  |  |  |  |
| ID | Unique identification number for an actor | int | 11 | true |  | true |
| name | name of the actor | varchar | 100 |  |  |  |
| dob | date of birth of the actor | date |  |  |  |  |
| country | country of the actor | varchar | 45 |  |  |  |
| **Movie** | A movie that will be in our “library” |  |  |  |  |  |
| ID | Unique identification number for a movie | int | 11 | true |  | true |
| name | name of the movie | varchar | 200 |  |  |  |
| description | description of the movie | varchar | 1000 |  |  |  |
| score | score of the movie (rating) | decimal | 10,2 |  |  |  |
| year | year of release | int | 11 |  |  |  |
| screenTime | movie duration | int | 11 |  |  |  |
| **Writer** | Writer of the plot of the movie |  |  |  |  |  |
| ID | Unique identification number for a writer of a movie | int | 11 | true |  | true |
| name | name of the writer | varchar | 45 |  |  |  |
| dob | date of birth | date |  |  |  |  |
| country | country of the writer | varchar | 45 |  |  |  |
| **Rating** | The grade of the movie based on its success |  |  |  |  |  |
| ID | Unique identification number for rating of the movie | int | 11 | true |  | true |
| type | type of the rating | varchar | 45 |  |  |  |
| **Awards** | The awards that a certain movie has won after its release date |  |  |  |  |  |
| ID | Unique identification number for the awards movie has won | int | 11 | true |  | true |
| name | name of the award | varchar | 45 |  |  |  |
| **Producer** | A person who produced a movie or more than one movie |  |  |  |  |  |
| ID | Unique identification number for a producer of a movie | int | 11 | true |  | true |
| name | name of the producer | varchar | 45 |  |  |  |
| dob | date of birth | date |  |  |  |  |
| country | country of producer | varchar | 45 |  |  |  |
| **Genre** | A type of the movie |  |  |  |  |  |
| ID | Unique identification number for a type of the movie | int | 11 | true |  | true |
| name | name of the genre | varchar | 45 |  |  |  |
| User\_has\_Movie |  |  |  |  |  |  |
| score | score for the movie the user gave | decimal | 10,2 |  |  |  |

## Business Layer Description

*[Provide Business layer design and explanation (deliverable 1 and 2), as well as connections to Database layer and Presentation layer.]*

Each entity in the ERD will be represented as a database object class with CRUD methods. Users interacting with the presentation layer will receive/pass data to these objects, and the objects are the ones communicating with the database layer, which is in charge for making the changes and fetching/returning the data from the actual database.  
Once the solution is implemented, a more precise and detailed description will be added.

## Presentation Layer Description

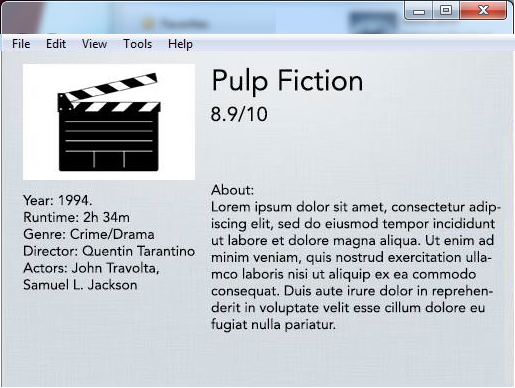
*[Provide Presentation layer graphical user interface design, structure, layout and explanation (no screenshots, could use prototypes) (deliverable 4). Include menus and options (deliverable 1 and 4).]*

The application will have a list of movies and filtering options. Upon selecting a movie a new window will open, containing information about the movie as well as a trailer. The menu will be generic one with information about the application and closing the application, and maybe few options for user’s preference in using the application.

Main Screen



Selected Movie



## Areas of particular concern

*[Provide Identification of areas of particular note or concerns (deliverable 1).]*

In the discussions we had as a group, we were talking about the possible performance issues due to a high number of possible joins when conducting queries. However, this might be optimized in the future updates.

Another possible concern that has been on our minds, is the embedding of videos into our Java desktop application, which might be troublesome due to the fact we have to use 3rd party APIs which might produce some issues.

# Requirements

## Context

*[Provide a description of the application in the context, with explanations as applicable. The context of a system refers to the connections and relationships between the application and its environment (deliverable 1, 3, and 5).]*

The application will be run on desktop. It will work in a way that the users will have a search bar where they will be able to enter some keywords or name of the movie and the movie will show up as a image on the main screen which the user can click in order to get all the information on the movie he or she searched, or they can search for a movie by genre search which will be a menu that is available to the left of the main screen. There will also be available some searches such as new or top rated movie which when the user clicks he or she will get all the new movies of the newest date and if they click on top rated they will get movies with the highest scores grouped by ratings from 5(highest rating) to 1 (lowest rating). The information which is displayed on the application when a movie is clicked will then display the movie title, the year of release, its grade, its main actors, budget, earnings, genre, and a description. Once the users have read all the information they can go on and search for another movie.

## Functional Requirements

*[List, name and explain all key functionalities. Could use UML use-case diagrams or use-case descriptions only including use-case name, actors, events flow, exceptions and special requirements. Include user requirements if necessary (users, roles, privileges) and associate with specific functional requirements (deliverable 1, 3, and 5).]*

|  |  |  |
| --- | --- | --- |
| **Name** | **No.** | **Functionality** |
| Starting application | 1.0 | When opening the application the application will display a list of all movies that are recorded in the database. The database has some pre-input records, but the admin(admin-role) can also add more movies to the database. When a movie is added to the database, the usual search and filters will have that record immediately. |
| Organizing movies/filtering movies | 1.1 | The application will have movies organized by various filters such as name, rating, score, genre, newest, most popular etc. Users(public-role) will be able to click on how they want to filter their movies and how they want to order them. |
| Description of movies | 1.2 | When a user(public-role) selects a movie from the home screen of the application, the user will see the movie title, names of actors, release year, generes, awards and ratings for all movies. |
| Movie score | 1.3 | Movie score will be calculated upon users grades, and even possibly suggest content that the user may like. |
| Search for movies | 1.4 | The user(public-role) will be able to search movies in the database by different criteria such as rating, genre, duration etc. |
| Roles | 2.0 | There are three types of roles available on the application. One is admin (Admin-role), one is registered user (user-role), and one is unregistered user (public-role).  The admin role are we and we have access to everything and are able to change, update and delete everything. Then there is the registered user role which allows any user that registered login and they are then able to save movies they have seen and are able to rate other movies. In the end there is also the unregistered user role or demo user which isn’t registered and anyone can user that user by default but you are able to only see movies and their descriptions. |
| Defining roles | 2.1 | One is Admin roles which only we will be able to have and we are capable of doing anything to the application, such as add/edit/delete movies to the database, edit information about people, add new genre etc. The second role is a registered user, or user-role which will be able to create an account on the application and rate movies (which will affect the “top rated” filter for example), and save movies in a list to see which movies they already have seen. And then there is the unregistered user role which is using the application without registering and just use the application to see movie descriptions without being able to rate or save movies. |

## Other (Non-Functional) Requirements

*[Describe the non-behavioral and non-functional requirements, including hardware and software requirements (platforms needed to support the system), interfaces, and operational requirements (how the system will run and communicate with operations personnel). Could also provide information about application availability, general performance, capacity, error handling, conventions used, security and similar if necessary (deliverable 1, 3, and 5).]*

|  |  |  |
| --- | --- | --- |
| **Name** | **No.** | **Non-Functionality** |
| General | 1.0 | The solution will be a desktop application, therefore, it will always be available to the user. |
|  | 1.1 | Since the solution will be written in Java, the host will have to have Java SE 8 SDK installed on their PC. |
|  | 1.2 | The application shall not stop communication with the database at any moment. |
|  | 1.3 | Network access will be required to use all the features of the application(trailers, 3rd party plugins etc.). |
| Roles | 2.0 | The users with admin role (we) can access the database and add, edit, delete movies etc. using proper credentials. |
|  | 2.1 | Registered users will have to have proper credentials to use the user panel where they can access their own list of movies they have seen and rate other movies. |
|  | 2.2 | Unregistered users will need no credentials input, they will just need to open the application and use it. They can only see the descriptions of the movies. |
| Error | 3.0 | If a user or admin enters the wrong credentials they will get a pop-up box that their username or password is incorrect. (Until they type the right credentials they won’t be able to login). |
| Security | 4.0 | All registered user credentials and admin credentials will be stored in our database that we only have access to. And only the admin may edit, delete, add new users. |

# User Documentation

## Graphical User Interface

*[User design and experience considerations (deliverable 3 and 4).]*

15151136_10211157781288752_1463962818_n.jpg

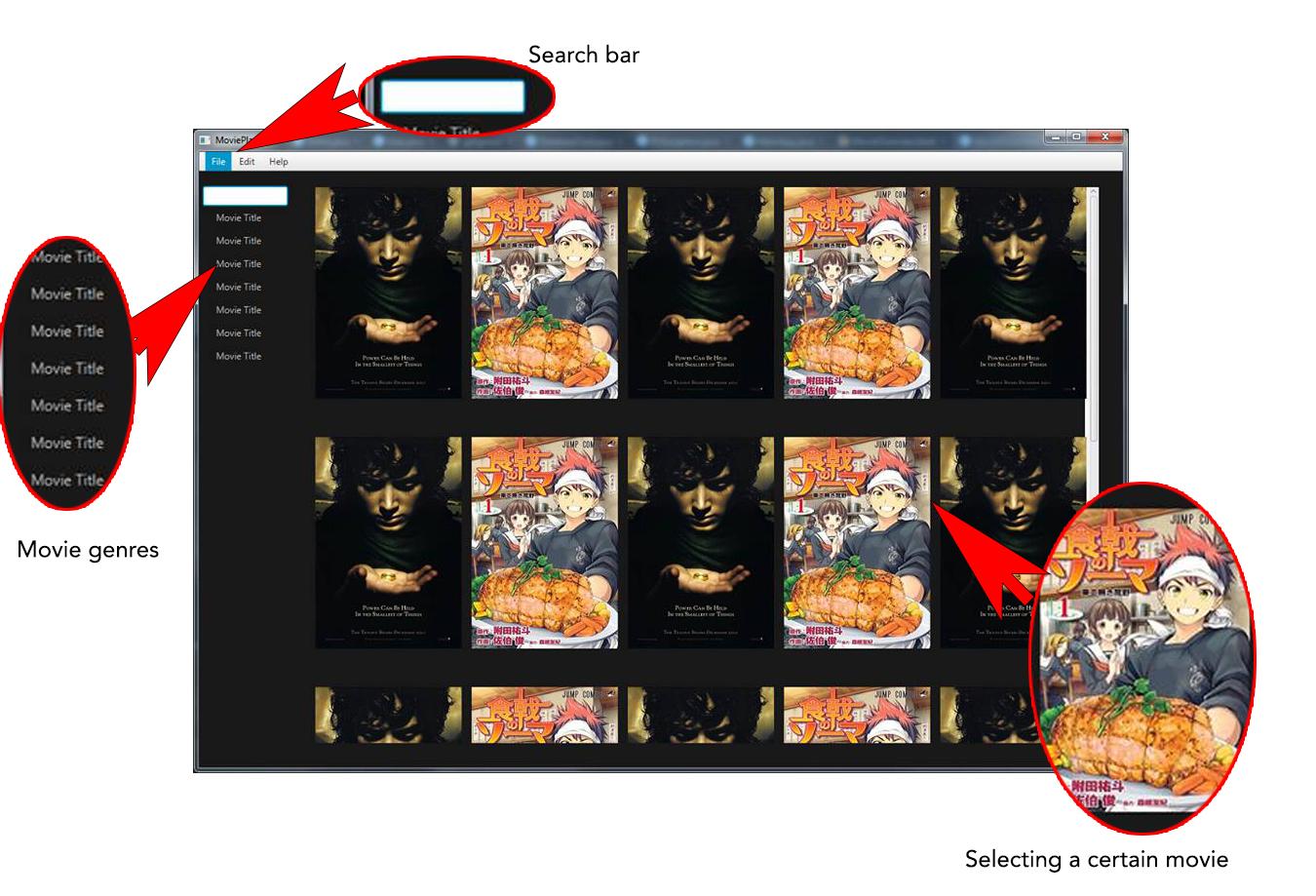
This is the GUI that will be our final design for this application. On the left-hand side will be the different sorting categories, starting with Genre, Year, Rating, etc… In this search bar in the upper left hand corner, the user will be able to search for a movie by its name. Clicking on the genre will load the movies in “top rated first” order. One of the filters will also be “Top Movies”, which serves as home screen. Clicking on the movie tile will open up the information about the movie (duration, actors, director, etc.). Admin will have a button that opens up the admin panel where new movies can be added or existing movies edited. Registered users will have a button to open up a list of the movies they have added to the list. In the “Help” menu, the user will get his User Manual which will contain all the necessary information needed to navigate this application correctly. If they are stuck at any moment in the time.



By selecting a certain movie, there will be provided all the information about the movie itself, containing a trailer, name, rating, year of release, director, actors, etc.. it will be interactive and users will be able to provide their own rating of the movie.

## User Manual

*[Expected usage of the available functionalities, could be divided per user roles, and could include screenshots with descriptions (deliverable 3).]*



**Menu #1:**

Here we see the main screen of the application. On the top we can see the menu with menu items “File”, “Edit” and “Help”. When you click on “File” a list will drop down and will provide us with some general functionalities like “logout”, “settings” and “exit”. “Edit” will provide the user some general editing functionalities like “Edit Profile”, “Edit Saved Movies” etc. The “Help” menu item when clicked will provide the user with Help instructions for the application how to use it and what functionalities are available.

**Search bar:**

There is a search bar on the left side of the application. It’s functionality will be to search through the database of movies and find what the user has input in the search bar. It’s a simple search engine which when a user inputs a string or integer it will pop out only the movies related to that search.

**Menu #2:**

The menu on the left side of the application will be used for grouping movies. There will be grouping by different genres(Sci-Fi, Horror, Comedy, Action etc.), newest (the newest movies will appear), A-Z (alphabetical ordering of movies), year (ordering by year when the movie came out) and by rating (depending on scores of the movie from 1 - 5 ).

**Movies Frame:**

Most of the application is taken by the movies frame which is listing all the movies in our database. Here a user will be able to select a certain movie and then it will open up the information about that movie the year, actors, directors, producers, short description, rating and user comments.

# Installation, Configuration and Acceptance Testing

## Installation

*[Technical manual – prerequisites and installation process (deliverable 5).]*

***Prerequisites****:* Internet Access and Java SE 8 SDK.

***Installation****:*

|  |  |
| --- | --- |
| Step | Description |
| 1 | You will be directed to a webpage from which you will be able to download our app |
| 2 | On the webpage you will be able to download our application installation folder as a zip file |
| 2 | Once the zip file is downloaded the person will need to extract it all to a desired destination. (Ex: Desktop) |
| 3 | Once the files are extracted, load the mySQL database, open the project in IntelliJ and run the application |

## Configuration

*[Technical manual – configuration and defaults (deliverable 5).]*

Hardware requirements: For the installation of our application, it is required of you to have a desktop machine that can run Java applications since mobile is not supported yet.

Software requirements: for best experience, we advise you have the latest version of your OS installed/updated (Windows, Linux or OSx), a java .jdk, and for opening the project use IntelliJ.

## Acceptance Testing

*[Testing done to determine if the requirements are met – describe typical usage and tests (deliverable 5).]*

For the acceptance testing we have laid out the expected outcomes of our application, and behaviours it should exhibit when it is operated. First we started with the download process and went ahead and downloaded our app from web. We unzipped it and got it running. When the application opens there you have to enter your credentials to continue using the application. Right after that a main screen opens as the one shown above in the screenshot in the user manual. From there you are free to pick a movie that is of your interest. When you click on it, all the information about a movie appears. You can then add it to your list if you wish, and in your list you can add your rating. You can see all the movies you have added on your list. The admin panel allows an admin to add a movie, an actor, user, producer, writer, etc. and connect them to a certain movie.

# Final Remarks and Conclusion

*[Summarize the experiences, both in terms of the produced results and work on the project. List project deliverables, as well as positive (and negative) experiences and concerns. Comment on missing functionalities and possibilities for improvement and extensions. Estimate project effort (person-hours) and how it was distributed in time and per team roles (deliverable 5).]*

“I liked this experience because I like working on group projects as it contributes to my teamwork abilities. I liked the idea of the project, and the fact that we developed an application on our own that is functional. We have made this documentation and the application all together and had no troubles communicating and meeting up about any of the topics or areas of concern that we have had on our way to the end. Everyone was available at most of the times and helpful towards each other. I feel as though we did a good job, but there is always room for improvement as there is in every project. Some of the things I feel like we could do better next time are more functionalities to the application but that is for some later time and some greater knowledge. We put much effort into this, and had people working on all areas. Some more on the programming and others more on the design of the application and the documentation itself. Overall I am satisfied with my team and with this project, it was a good learning experience” - Ana Raguz.

“This group project was like a journey to me where I have learnt new skills and improved some others. I improved my team skills and working and organizing as a team member. We had to organize and see who does what some where more working on some areas, depending on what they are more skillful at. Still everyone managed to do their part of the project and everyone was helping everyone when needed. The results in the end were amazing we made a application which shows movie descriptions to users just as we wanted. Throughout this journey I’ve mostly experienced positive experiences such as working in a team helping each other and learning new stuff. There were no negative experiences maybe only lack of knowledge which I then learned throughout the project. The only missing functionally that we thought about adding but didn’t is the comments section so that every registered user should be able to comment on every movie. We could improve the design of the application to make it look more beautiful, we could add more filters for searching movies like trending, most seen etc. Overall I’ve spent 24 hours on making this project for my role. I worked on the documentation and planning most of my time throughout this project.” - Ivan Landeka