OLLSCOIL TEICNEOLAÍOCHTA
BHAILE ÁTHA CLIATH
DUBLIN
TECHNOLOGICAL
UNIVERSITY DUBLIN

Software Engineering and Testing. BSC Year 2, 2024/2025

(Assignment 3 - 20%)

Assessment 3: Design and Draft Implementation

Submitted by:

B00160049 Matthew Keenan

B00159340 Stanislav Kolev

B0016388 Fabio Melilo

Submission date

24/3/25

Declaration

I herby certify that this material, which I now submit for assessment on the programme of study leading to the award of Ordinary Degree in Computing in the Institute of Technology Blanchardstown, is entirely my own work except where otherwise stated.

Author: B00160049 Matthew Keenan Dated: 24/3/25

Author: **B00159340 Stanislav Kolev** Dated: 24/3/25

Author: **B0016388 Fabio Melilo** Dated: 24/3/25

Table of Contents

Title:

Director’s Cut: Movie Booking and Recommendation System

Project Definitions

- Purpose of document

  This document will outline the design process using ERDs and class diagrams and implementation of the design using html and php.

- What is the project?

  Our project is a movie recommendation and booking which will allow the find new movies based on genre and then booking a screening of that movie. User will also be able to review movies they have seen, and there will be profile pages for identification of the reviewer for other users.

- Functional Specifications

  Buying Feature - Shopping

  - Allows users to add movie tickets with booking time and desired seat into shopping cart.
  - Shopping carts lead to a checkout screen which will prompt banking details.

  Recommendation System - Recommendations

  - Once clicked on a desired movie the user will be put into the times and seats screen this will also contain a section that recommends movies that are of the same genre using JSON.
  - The user will be able to navigate to the movies in this selection, this allows them to purchase tickets for the recommendations.

  Movie Browsing - Searching

  - The movie browsing page displays movies available for purchase through JSON database. The database can also be queried using search functions.
  - Each movie will lead to its own page that contains the price and times that are available, the user will also get to choose their desired seat.

  Reviews – Movie Reviews

  - Users can leave written reviews on movies they have watched.
  - Reviews can include a rating system (e.g., 1-5 stars).
  - Displayed reviews will be fetched from a database using JSON.

  Profiles Reviews - User Rating of Reviews

  - Each user profile will have a review system, for there reviews like a thumbs up and thumbs down on the reviews.

Only change to the functionality specifications is the removal of user friending which is replaced with profile reviews

- Main components of the software system

  Php and html for the website implementation

  Sql for the database inclusion

1. Document Revision

Rev. 1.0 19/3/25 – initial version

Rev. 2.0 22/3/25 – completed documentation

2. Methodology

System models – UML

The use of a UML diagram will allow us to visualize the intended use of the website systems such as the payment for the bookings or the reviewing system

Use of, and necessity of OOAD

The use of OOAD allows use to divide the implementation of the website in to individual class to and objects for example within a recommendations class each movie will be an object with title,genre,desc,price etc as variables within the movie object with will be stored in the recommendations class and OOAD will help us develop the filtering method within the class that will show movies based on the if the movie genre variable matches the selected genre.

Purpose of using classes / What is a class diagram?

Classes will open the ability to manage the objects and methods necessary for the website as shown above with the recommendation class example as the class manage the use of the movie object and functionality of the methods used within e.g the filtering method

The class diagram will visualize the total number of classes used within the website and outline the relationship between all the classes for example it will visually show that every booking object must include a movie object as you can't book a film screening without knowing the film or the price of the ticket

Static Versus Dynamic Case Diagrams?

A static diagram showcases the structure of a system in terms of relationships between the classes and each part of the system is used at a single point in time, without considering possible changes over time an example of these types of diagrams are class diagrams or ERDs as they focus on the relationship between classes for class diagrams or tables for ERDs

Whereas Dynamic case diagrams consider different points within the system for example a diagram laying out the difference in function in a logged in user and one that isn't

What is an ERD?

An ERD is a diagram that visualizes the relationship between different parts of the system and outlines the type of relationships for example one movie has many reviews. It also has the elements of the system needed to create a database for example the review section will have a review score, review text and review id excluding FK's

Purpose of using classes?

Classes allow for management and use of data and methods will allow its use within the system for example a movie class will manage the title genre and price from the database, all information on the movies

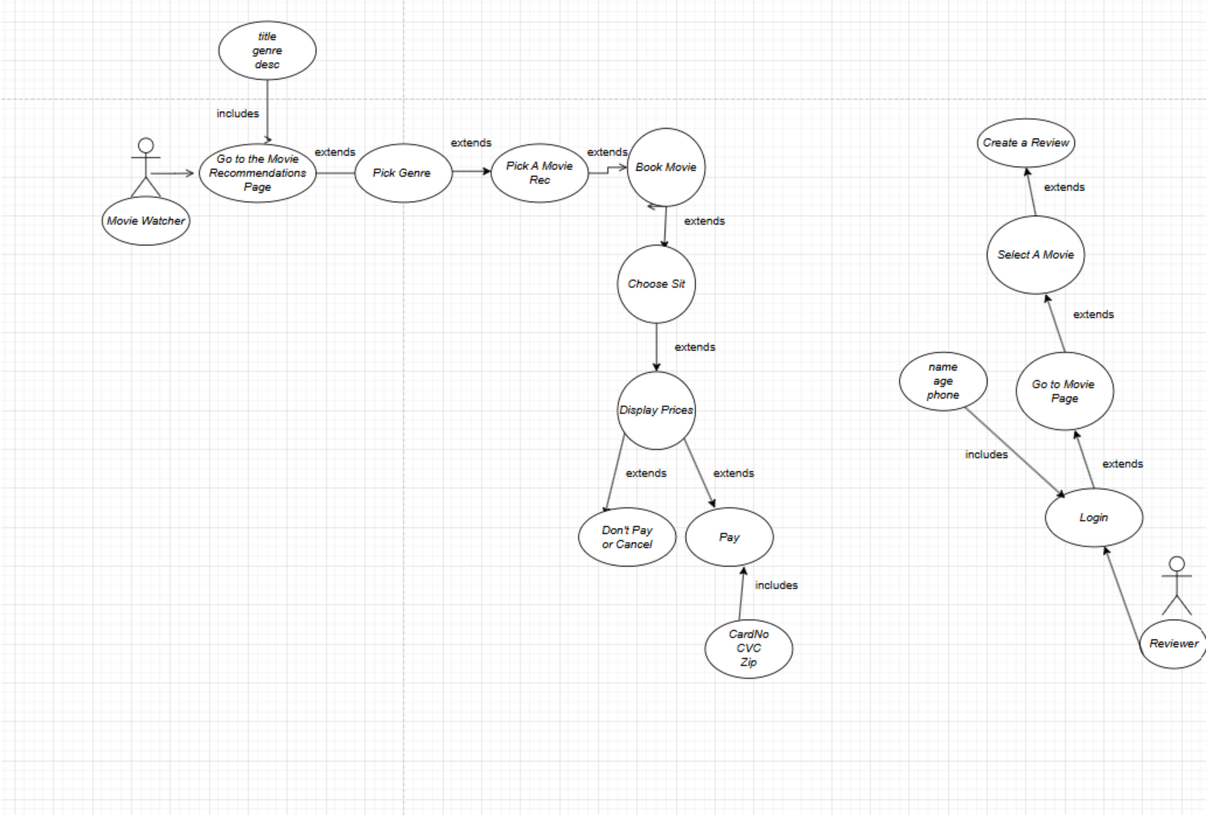
Volatile versus Persistent storage – Object Instances / Database?

Persistent storage is storage that is maintained after the system is shut down for example the dbdirectordb used to store the information on the movies doesn't disappear after SQL is shutdown the data is maintained for future use, unless manually deleted. Were as volatile storage disappear after a system is shut down for example to test if the matchDetails method works the instance of userTest only exist when the program is run.

User Interface template chosen and how it can aid in executing the functional specification of the project.

3. Requirements

4.1 Use Cases



<u>CASE 1: Movie Watcher:</u>

<u>Find A Movie:</u>

. go to the recommended movies page

. filter by the type of movie you want:

Title

Genre

Price

. choose from recommended movies

. click on movie

<u>Book Movie:</u>

. click on movie page

. click booking option

. select appropriate time

. login to allow you to book

. select sitting

. continue to payment screen

Display Prices

. shows the user the cost of the ticket

. display option to enter payment details

Pay

. Input Payment Information:

- Name
- card number
- cvc
- zip code

. Confirm payment transaction

. Pay for the movie

Don't Pay

. Asked to Input Payment Details

. Click decline

. return to movie page

CASE 2: Reviewer

Login

. Select register/login

. Input login details

-email

-name

-age

-password

-phone number

. confirm information

. profile is created

Pick A Movie

. Go to a movie page you have previously watched

. click on the movie page

. click on review button

Review Movie

. Fill out description on your thoughts on the Movie

. Select score of enjoyment

. Confirm you want to finalize and publish review by clicking confirm

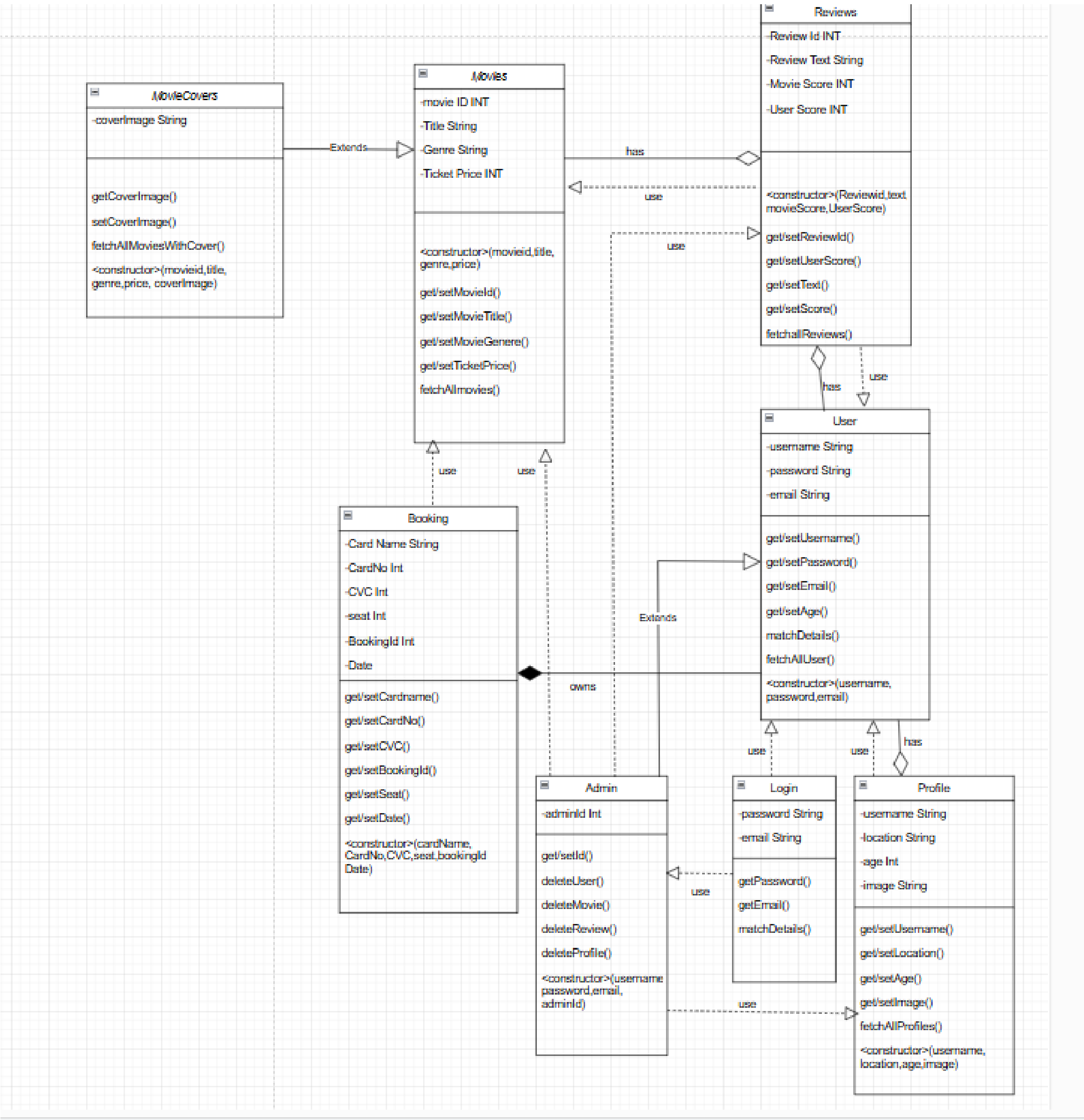
4.2 Use Case Specifications

(Specifically – how use case specifications have been used as a means to develop the ***classes/attributes/methods*** and database ***tables***)

- The use case has been used to develop the movie class as on the main page as the title genre and price is grabbed from the ThedirectorsDB and displayed on the page
- The use case has been used to develop userTest and login class however currently only the email and password are in use to test the matchDetails method
- The Use case has been used to develop will be used to develop the booking class to check and store the card Name, Cardno and cvc but currently the variable are only create not functional

4. Case Diagrams

Class Diagram – Show all relationships, multiplicities, associations, generalisations (inheritance), aggregations (compositions) - See lecture 4.



Paragraph to explain ALL design decisions.

Movie-Review

- Every movie has a review, and the review uses the movie variables to identify the movie being reviewed

Cover-Movie

Cover image extends movie as its an element of movie as it gets the image of the movie

Booking-Movie

- To make a booking the booking needs to use the movie variables to identify the which movie the booking is for and the price of the film

User-Booking

- Because a user pays for a booking, they own the booking and have access to the booking to confirm ownership

Review-User

- Users have user reviews and to make a user review the review class needs to use the user variables e.g. username

User-Login

- For the login class to check if the user details are correct, it needs to be able to use those variables e.g. email and password

User-Profile

- A user has a profile when they first login to an were they can add information to the profile

Admin-Login-User

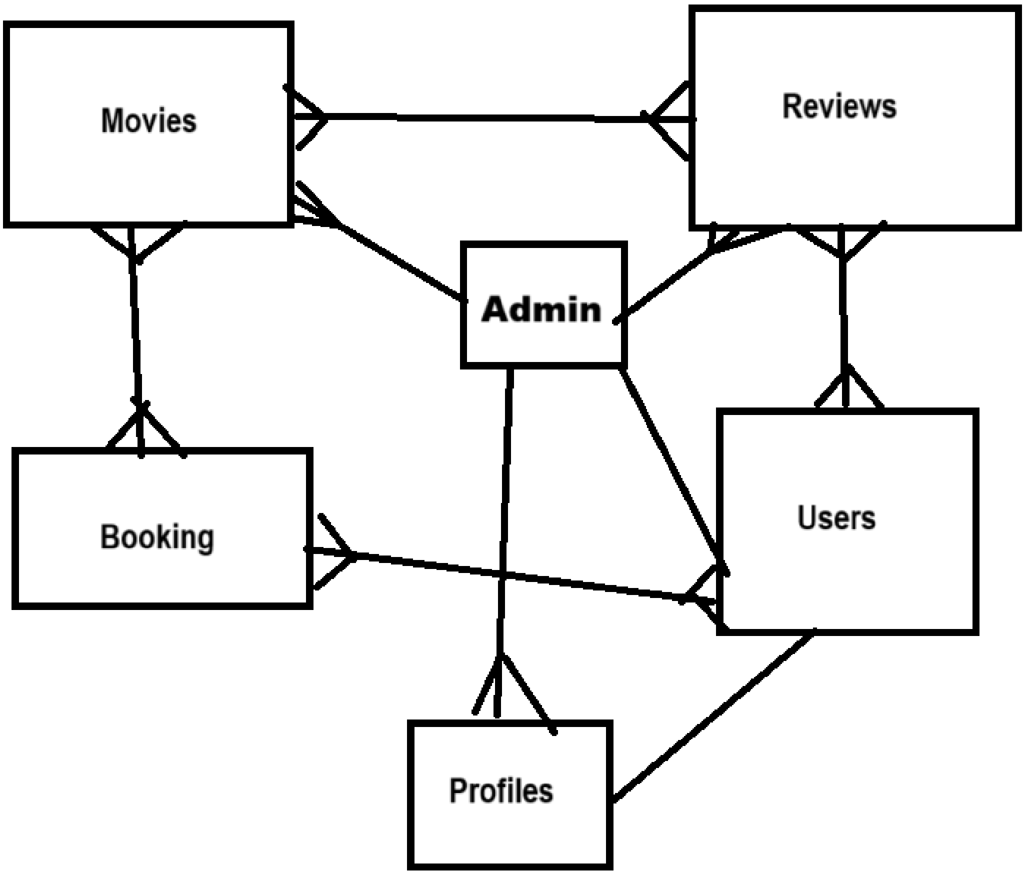
- For the login class to check if the admin details are correct, it needs to be able to use those variables e.g. email and adminId
- As an admin is a type of user the admin class extends user, also so it has access to the class to delete a user

Admin-Movie-Profile

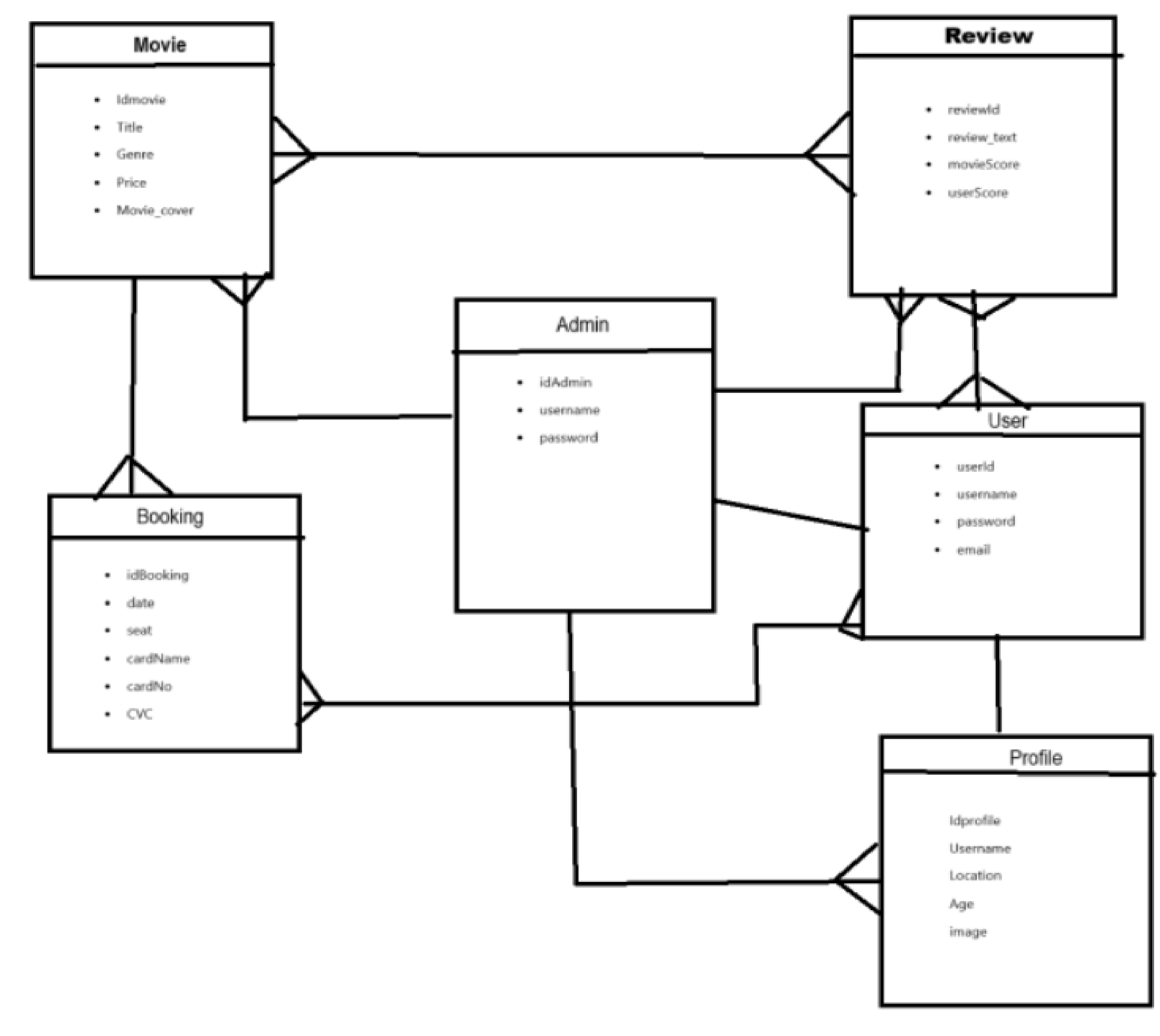
- The admin has to be able to use the movie and profile so they can be deleted

Entity Relationship Diagram – Show all relationships, multiplicities,

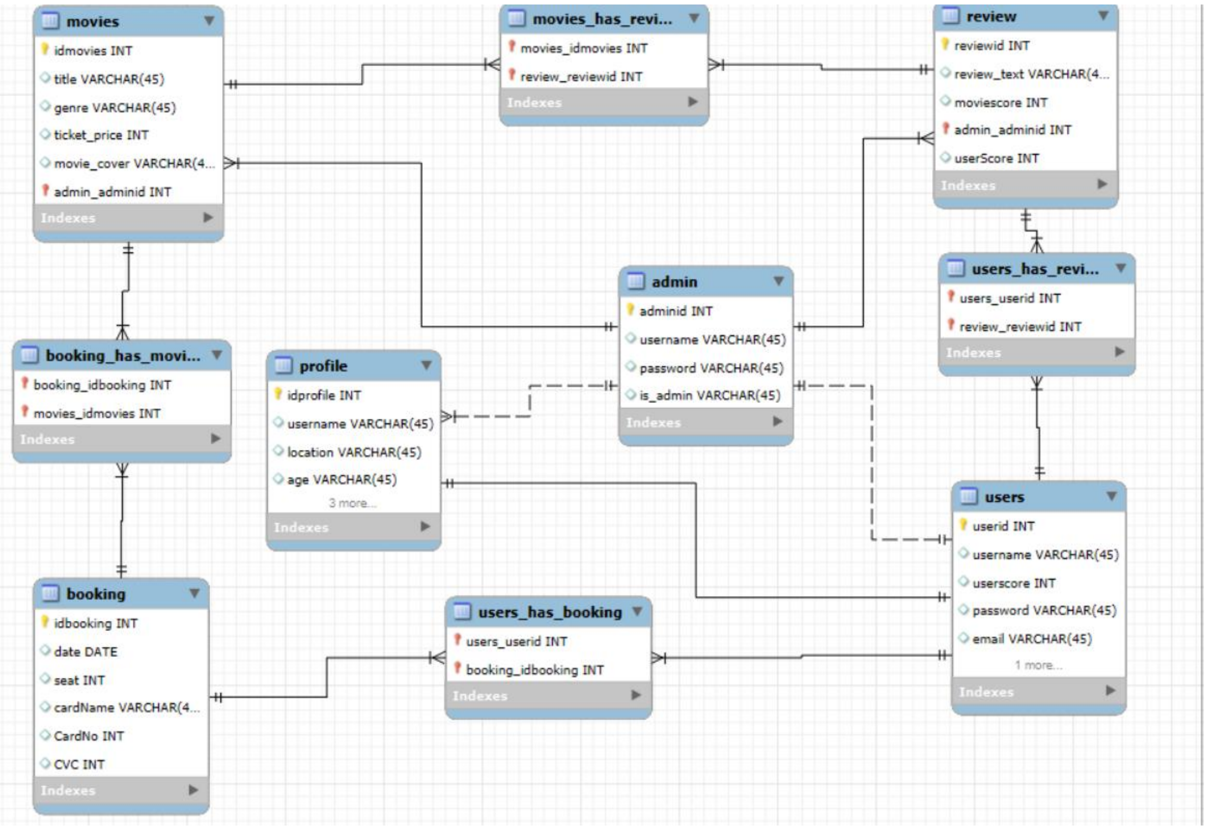
Conceptual



Logical


Movie
Idmovie
Title
Genre
Price
Movie_cover
Review
reviewId
review_text
movieScore
userScore
Admin
idAdmin
username
password
User
userId
username
password
email
Booking
idBooking
date
seat
cardName
cardNo
CVC
Profile
Idprofile
Username
Location
Age
image

Physical



Paragraph to explain ALL design decisions.

Movie-Review-User

- A single movie or user can have many reviews

Movie-Booking-users

- Many movies can be booked many times by a single or many different users

Users-Profile

- Every user has a profile after registering as a user

Admin-User

- Every admin is a user just with a higher privilege

Admin-Movie-Profiles

- An Admin can delete many movies and profiles

5. Conclusions

Your conclusions and recommendations as to how far the project has progressed.

The project is going as planned with a slight delay in implementation of the design shown in diagrams. The recommendation for future projects is to not over complicate design simple design is more effective

Your changes to the original proposal that the design has revealed and necessitated.

- Removal of user friending in favour of user reviews
- simplified use case diagram

Checklist: Is your document complete and correct?

Content:

- Does the design include all requirements from the customers' needs

  It allows the user to book review and be recommended movies as outlined

- Are you satisfied with all parts of the document?

  yes

- Do you believe all parts have been implemented?

  As of now no as the project is still on going

- Have you explained your methodology and design choices?

  Methodology and design process are explained

- Have you clearly articulated your understanding of the purpose of all diagrams created ?

  All design for the diagrams has been outlined

- What are these diagrams? Why you need them? How were they developed?

  Use Case- helps understand the use of the system and helps creation of class diagram

  Class- Helps outline how the classes relate to each other and helps to showcase how implementation should be done

  EDR- Gives an outline of the relationship between tables with a database

- Is each part of the document in agreement with all other parts?

  All the sections match up besides some class names as they were changed later on in the class digram

- Does the design create a solution for the initial proposal?

  The design does allow for users to book, review and be recommended movies yes

Completeness:

- Are all the necessary components specified?

  All components are specified

- Are the design specifications precise enough?

  The specifications allow the system to work as intended

- Are all sections from the document template included – if changed, why?

  Nothing was missed beside the second table of contents as the table of contents is outlined earlier in the document

Clarity:

- Is the design reasonable?

  The requirements needed to run the movie booking and recommendation the design is reasonable as it includes everything necessary for it function properly

- Is the level of details for each design section appropriate?

  Each design section is outlined in good detail

- Is the design written in a language appropriate to the intended audience of software engineering teams?

  All language is appropriate to the intended audience

- Are all items clear and unambiguous?

  Everything is outlined clearly