

**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.

E-Ticket Signature - Ticketing

* After a purchase is successful an E-Ticket signature will be created. That could theoretically be used to print tickets at the cinema in person.
* The signature is a randomly generated string using JavaScript.
* Printable Ticket

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.

Profiles Friending - Friending

* Users will be able to friend each other and look at each other’s profiles.

1. **Document Revision**

Rev. 1.0 19/3/25 – initial version

1. **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.

1. **Requirements**

4.1 Use Cases

A diagram of a company

AI-generated content may be incorrect.

**CASE 1: Movie Watcher:**

Find A Movie:

. go to the recommended movies page

. filter by the type of movie you want:

Title

Genre

Price

. choose from recommended movies

. click on movie

Book Movie:

. 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

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

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

1. **Case Diagrams**

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

Paragraph to explain ALL design decisions.

**Entity Relationship Diagram** – Show all relationships, multiplicities,

A computer screen shot of a computer

AI-generated content may be incorrect.

Paragraph to explain ALL design decisions.

1. **Conclusions**

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

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

Additional sections: Table of Contents, Table of Figures, References, Index

Checklist: Is your document complete and correct?

*Content:*

* Does the design include all requirements from the customers’ needs
* Are you satisfied with all parts of the document?
* Do you believe all parts have been implemented?
* Have you explained your methodology and design choices?
* Have you clearly articulated your understanding of the purpose of all diagrams created ?
* What are these diagrams? Why you need them? How were they developed?
* Is each part of the document in agreement with all other parts?
* Does the design create a solution for the initial proposal?

*Completeness*:

* Are all the necessary components specified?
* Are the design specifications precise enough?
* Are all sections from the document template included – if changed, why?

*Clarity*:

* Is the design reasonable?
* Is the level of details for each design section appropriate?
* Is the design written in a language appropriate to the intended audience of software engineering teams?
* Are all items clear and unambiguous?