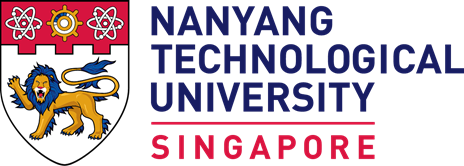
****

**SC/CE/CZ2002 Object-Oriented Design & Programming**

**Group Assignment**

**Movie Booking and Listing Management Application (MOBLIMA)**

**Submitted on:**

11 Nov 2022

**By:**

**SS10**

**Group 3**

Chew Zhi Qi (U2120266D )

Eddy Cheng Kuan Quan (U2121953D)

Gan Hao Yi (U210769F)

Sanskkriti Jain (U2123542L)

Tan Jue Lin (U2111250D)

**Declaration of Original Work for SC/CE/CZ2002 Assignment**

We hereby declare that the attached group assignment has been researched, undertaken, completed and submitted as a collective effort by the group members listed below.

We have honored the principles of academic integrity and have upheld Student Code of Academic Conduct in the completion of this work.

We understand that if plagiarism is found in the assignment, then lower marks or no marks will be awarded for the assessed work. In addition, disciplinary actions may be taken.

| Name | Course | Lab Group | Signature/Date |
| --- | --- | --- | --- |
| CHEW ZHI QI | Computer Science Y2 | SS10 |  |
| EDDY CHENG KUAN QUAN | Computer Science Y2 | SS10 |  |
| GAN HAO YI | Computer Science Y2 | SS10 |  |
| SANSKKRITI JAIN | Computer Science Y2 | SS10 |  |
| TAN JUE LIN | Computer Science Y2 | SS10 |  |

**Design Consideration**

1. Project Assumptions (Excluding stated assumptions in Project Description):
2. There is a default admin account written in the database.
3. There is a default ticket price charge written in the database.
4. There is a default environment variable stored in the database.
5. There are default cineplexes stored in the database.
6. Implementation of the SOLID design principles:
7. Single Responsibility Principle (SRP)

This principle focuses on the cohesion of the classes such that any one class is only responsible for its own functionality and attributes. To implement this it must be noted that a class doesn’t take up more than one responsibility. Throughout our project, there are many examples of this Design Principles implementation. One such example is our TextDB class, which is mainly responsible for the interactions with the text files to add/edit/delete any data from the databases. This is not implemented in the respective UserInterface classes to ensure the UserInterface classes like MoviesUI, AdminUI etc are only responsible for getting or displaying data to the users and TextDB is the one that actually interacts with the database.

1. Open - Closed Principle (OCP)

This principle is implemented in classes to allow them to change what they do without modifying their source code allowing us to prevent causing bugs. This has been implemented at various instances in our Application. One such example is the Movie Class, this class is implemented to store all the constructors and assessors required for multiple details for a movie listing. This class defines variables like movieTitle, movieSynopsis, and includes setter and getter methods as well. If we need to add additional attributes in the future, we can extend this class and make any modifications instead of directly editing the class’s source code itself.

1. Interface Segregation Principle (ISP)

Interface Segregation Principle refers to the practice of classes not depending on interfaces that they do not use. In our application this has been implemented by the frontend and backend being separate classes. The user interface classes like MovieListingUI, ReviewUI, AdminUI, etc. are used for the user to be able to interact with the application and the classes such as Customer, Admin, Movie, etc. are used to handle the backend functions which doesn’t require the user to directly access it and instead are responsible for the manipulation and/or data storage.

1. Use of Object Oriented Programming Concepts
   1. Encapsulation

Encapsulation is used in our application in almost every defined class to help protect any object's data from being incorrectly accessed by any other class. This has been aptly reinforced in our MOBLIMA application by making various attributes within a class private. These can be accessed or modified only by using the setter or getter methods in the object’s class. For example, in the Customer class, if you would like the MovieGoer to be able to update their details like their name or phone number, you would have to go through the setMovieGoer and setMobileNumber methods. In terms of the getter method, an example includes using the Movie class, wherein to print the Movie Name or other details in the tickets, the getChosenMovie or method is called to return the value.

* 1. Inheritance

The use of Inheritance in any program is to avoid duplication of the codes and helps avoid creating multiple classes. Our application uses the concept of Inheritance which allows the new class to inherit all the public methods and attributes from a created class. In our application, an example of this concept includes its use in the AllCineplex class. Using the line **“public class AllCineplex extends Settings{}”**, the AllCineplex Class inherits attributes and methods from the Settings Class. Given that the Settings class holds global environmental variables for the Reviewing and Ranking system, by inheriting the Settings class, the AllCineplex class gets access to all the public attributes like the boolean variables, isSale or IsRating etc.

* 1. Abstraction

Abstraction is the process wherein the error handling and implementation details are hidden from the user and only the functionality is shown to them. It helps reduce the complexity and usability of the program making it more user-friendly. In our Application, this has been used in the login system. The Admin Class handles the input from the user and the corresponding output to be displayed in the console. In this class, the user is also shown the prompts to input their username and password mainly using the printing methods but they are unaware of the error handling using the Scanner Object or the hashing of their passwords using the SHA256 Algorithm both of which are included in the class.

* 1. Polymorphism

Polymorphism refers to the ability of an object to be referred to different types, and the appropriate type depends on the Inheritance hierarchy. Polymorphism is demonstrated through method overloading and method overriding. In our application, we demonstrated method overloading through our TextDB class, where we have several methods with the same name such as the functions readFromFile and WriteToTextDB. Method overloading is done by implementing different method signatures. For instance, WriteToTextDB(String fileName, Movie movie) and WriteToTextDB(String fileName, Customer customer) uses the same method name but different parameters.

**Future Implementations**

1. Recommendation System

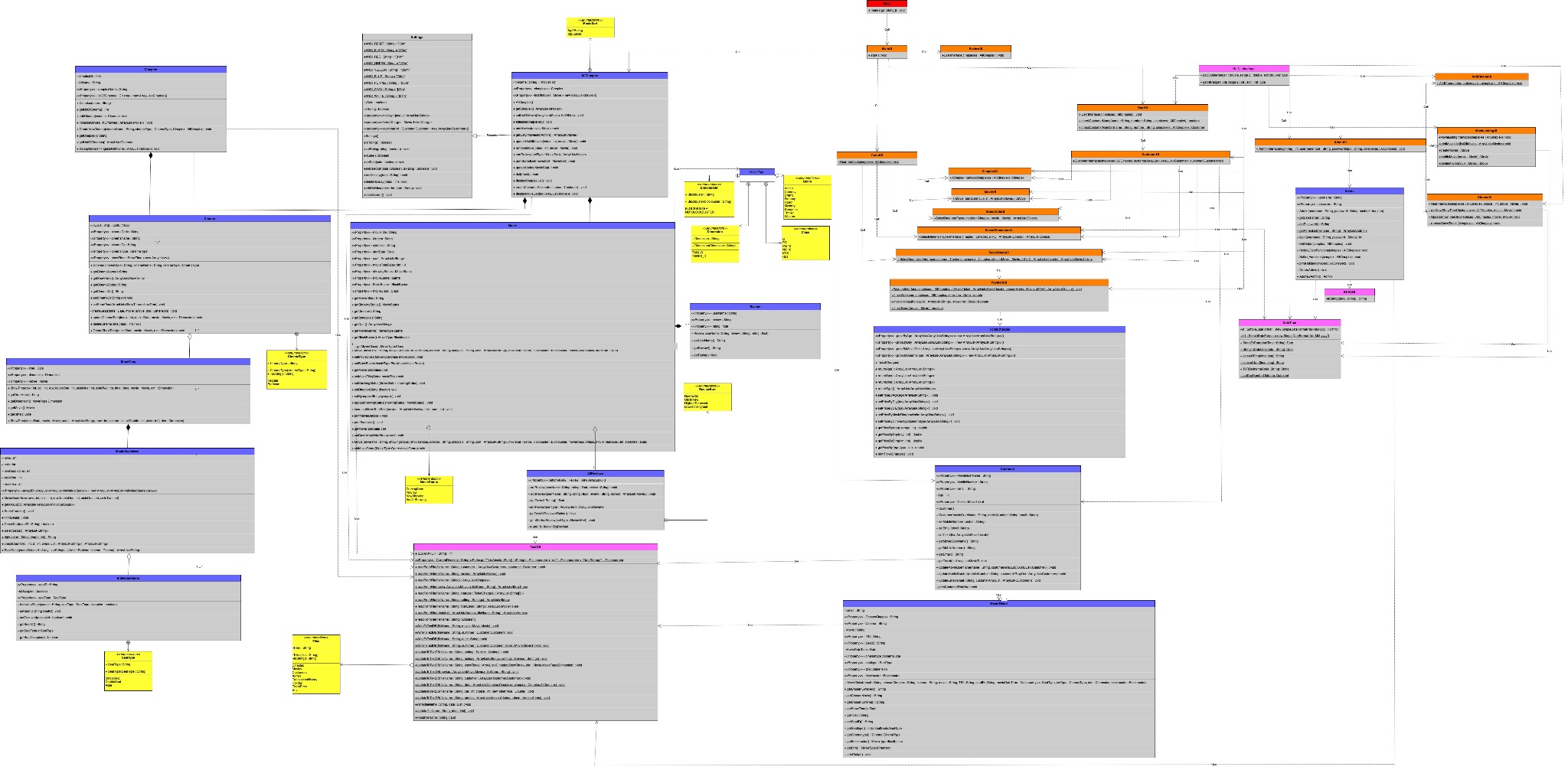
The system could suggest movies the customer would prefer based on past purchases and search history. By collecting data on the movies that the customer watches and searches for, the system could implement an algorithm to suggest movies that the customer might be interested in. For example, if the customer often watches the Horror / Thriller genre, the system could suggest new Horror / Thriller releases when the customer logs in. The implementation of the Open-Closed principle in our program, as well as the code reusability allows us to apply this extension easily.

1. Ticket Price Comparison

The customer could be allowed to compare ticket pricing between different days of the week, cinema type, movie dimension etc. They can decide whether spending a bit more on any combination is worth it or not and which is more suitable for them. So if they have originally chosen a normal cinema they can instantly compare the price with a premium cinema and decide on whether they would prefer to pay the extra amount to watch the movie and make a booking in the premium cinema instead. The implementation of the open-closed principle of our program in the classes MovieTicket, TicketCharges make it easier to implement this. The existing class doesn't have to be modified, an extension of the current class is needed to allow this function to be implemented.

| ID | Description | Test Case | Test Data | Expected Outcome |
| --- | --- | --- | --- | --- |
| Admin Options | | | | |
| 01 | Check User/Admin Login with valid and invalid option | 1. User key in “5” and Enter.  2. User enters a valid username and password | “Valid Option” -  Login Input: Username = ant  Password = 12345  “Invalid Option” -  Login Input:  Username = ant  Password = 7890  “Valid Option” -  Login Input: Username = admin/ant  Password = 12345  “Invalid Option” -  Login Input:  Username = admin/ant  Password = 7890 | Valid: User allowed to enter User Portal  Invalid: Error message, asks user to re-enter value  Valid: Admin user allowed to enter Admin Portal  Invalid: Error message, asks user to re-enter value |
| 02 | Ticket Prices Display/Edit | Edit price by category:  1.Admin enters “5” and enters  2. Admin enters a valid admin username and password  3. User key in “1”  Display:  3.1. Admin key in “1” and press Enter  4. Admin to key in a valid/invalid option when prompt to  5. Admin to key in a valid/invalid option when prompt to  6. User key in new value | “Valid Option” -  Login Input: Username = admin/ant  Password = 12345  1 - Display ticket prices  Category:  “Valid Option” -  1- Age (Adult , student, Senior Citizen)  2-Day of the week(Mon-Fri,Sat-Sun,PH)  3-2D/3D movie  4-Regular/Premium  Any invalid:  “Invalid option” - any number out of range displayed. | Display:  “1” - Display ticket prices  Category:  Valid options are shown to admin  Display:Ticket price updated  Any invalid:  “Invalid option” - any number out of range displayed. |
| 03 | Choose to create/edit/delete movie | 1.Admin enters “5” and enters  2. Admin enters a valid admin username and password  3. Admin key in “3”  4. Admin to key in a valid/invalid option when prompt to | “Valid Option” -  1 - Create movie  2 - Update movie  3 - Delete movie  “Invalid option” - any number out of range displayed. | Valid:  1,2,3-Prompted to choose which movie  1- To enter all movie details (Movie, name, movie status,director,synopsis,cast,genre,blockbuster,ratings,start date)  Results-Display new movie details  2-To choose the movie property to change(Movie, name,movie status,director,synopsis,cast,genre,blockbuster,ratings,start date)  Results-Display update movie details  3-To choose the movie to delete(Change movie to end of showing)  Results-Movie deleted  Any Invalid value: Re-enter value |
| 04 | Admin able to view Ranking of movie | 1. Admin key in “5” and Enter.  2. Admin enters a valid admin username and password  3. Admin key in “4” and Enter  4. Admin key in valid input | “5” - Login with your Account.”  valid username and password  “4” - Movie Listing by rank  Valid option:  “1” - Display Top 5 movie rankings by rating  “2” - Display Top 5 movie rankings by ticket sales | Top 5 Ratings would be displayed according to admins chosen setting |
| 05 | Admin able to add new cinema | 1.Admin key in “5” and Enter.  2.Admin enters a valid admin username and password  3.Admin key in “5” and Enter  4.Admin key in valid input to select cineplex to add cinema  5.Admin key in cinema name  6.Admin key in regular/regular cinema type | “5” - Login with your Account.”  valid username and password  “5” - Movie Listing by rank  “Valid Option” -  Valid options are shown to admin  “Invalid option” - any number out of range displayed. | Valid: To display all the seating arrangement  Invalid: User should be prompt with an error and asked to key in valid input within range.  Users should only be able to enter next options once they enter a valid option. |
| 06 | Admin able to change what the user is able to view (Top 5 by sale/rating) or register new admin. | 1.Admin key in “5” and Enter.  2.Admin enters a valid admin username and password  3.Admin key in “6” and Enter  4.Admin key in valid input to select change user view setting or add new admin | “5” - Login with your Account.”  valid username and password  “Valid Option” -  1-Control the display of movie rankings to customers  2-Register new Administration Account | Valid:  1- Admin prompted to choose user can view by (Display by rating,Display by ticket sales,Display both,Disable top rating display)  Result- User interface will be shown  2-Admin prompted to key in Admin username and password.   Results-Admin will be created and stored to db with hashed password |
| 05 | User are able to view seats | 1. User key in “4” and press Enter.  2. User key in valid Cineplex Options  3. User key in valid Movie options  4. User key in valid Date options  5. User key in valid Time options  6. User key in valid Dimensions option  7. User key in valid number of pax  8. User to key in valid number for selected seats. | “Valid Option” - Within the range of numbers shown.  “Invalid option” - any number out of range displayed.  “Invalid option” - user would be prompt if seat selected are taken | Valid: Admin is prompted to key in cinema name and cinema type  Invalid: User should be prompt with an error and asked to key in valid input within range.  Users should only be able to enter next options once they enter a valid option. |
| 06 | Guest Booking:  Users are able to view ticket details & prices. | 1. User key in “4” and press Enter.  2. User key in valid Cineplex Options  3. User key in valid Movie options  4. User key in valid Date options  5. User key in valid Time options  6. User key in valid Dimensions option  7. User key in valid number of pax  8. User to key in valid number for selected seats.  9. User to key in “1” and press Enter  10. User to key in valid Username, mobile number and email  11. User to key in valid number of elderly/student ticket | “Valid Option” -  “1”- Yes confirm  “2” - No don’t confirm  “Invalid option” - any number out of range displayed. | Valid: User to view ticket details and total prices. |
| 07 | Customer Booking:  User can to confirm seat booking | 1. User key in “5” and Enter.  2. User enters a valid/invalid username and phone number to login  3. User enters “1” and Enter  4. User key in valid Cineplex Options  5. User key in valid Movie options  6. User key in valid Date options  7. User key in valid Time options  8. User key in valid Dimensions option  9. User key in valid number of pax  10. User to key in valid number for selected seats.  11. User to key in valid/invalid options to confirm booking | “Valid Option” -  Login Input: Username = admin/ant  Password = 12345  “Valid Option” -  “1”- Yes confirm  “2” - No don’t confirm  “Invalid option” - any number out of range displayed. | Valid:  “1” - will prompt User to enter choose number of Student and Elderly  “2” - go back to main page  Invalid: User should be prompt with an error and asked to key in valid input within range.  Users should only be able to enter next options once they enter a valid option. |
| 08 | User able to view ranking of movie | General User:  1. User key in “6” and press Enter | “6” - Display Movies by Ranking | Top 5 Ratings would be displayed according to chosen setting |
| 09 | Customer able to view Purchase History | 1. User key in “5” and Enter.  2. User enters a valid/invalid username and phone number to login  3. User enters “6” | (3) Valid Option: “5”  Invalid Option: “ >6” | The user is shown their booking history and brought back to main menu |
| 10 | User able to view List of Cineplex/Cinema/Movie | 1. User key in valid/invalid option input and press Enter. | Valid input:  “1” - Display List of Cineplexes.  “2” - Display List of Movies.  “3” Display List of Showtime  “Invalid option” - any number out of range displayed. | Will show selected valid input.  Invalid: User should be prompt with an error and asked to key in valid input within range. |
| 11 | Display Movie review based on option chose | 1. User key in “2” and press Enter.  2. User key in valid option and press Enter 3. User key in valid/invalid option input and press Enter | “1” - Newest to oldest  “2” - Oldest to newest  “3” - Highest rating to lowest rating  “4” - Lowest rating to highest rating  “Invalid option” - any number out of range displayed. | Review from chosen option will display accordingly  Invalid: User should be prompt with an error and asked to key in valid input within range. |

| 01 | Check User/Admin Login with valid and invalid option | 05 | | User are able to view seats |
| --- | --- | --- | --- | --- |
|  | |  | | |
| 06 | | Guest Booking: User able to view ticket details & prices. |
|  | | |
| 08 | | User able to view ranking of movie |
|  | | |
| 02 | Ticket Prices Display/Edit | 07 | | Customer Booking: User can to confirm seat booking |
|  | |  | | |
| 09 | | Customer able to view Purchase History |
|  | | |
| 03 | Choose to create/edit/delete movie |  | | Admin able to view Ranking of movie |
|  | |  | | |
| 05 | | Admin able to add new cinema |
|  | | |
| 06 | Admin able to change what the user is able to view (Top 5 by sale/rating) or register new admin. | 11 | | Display Movie review based on option chose |
|  | |  | | |
| 10 | User able to view List of Cineplex/Cinema/Movie | | | |
|  | | | | |



\*see attached image.