



**NANYANG  
TECHNOLOGICAL  
UNIVERSITY**  

---

**SINGAPORE**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING  
NANYANG TECHNOLOGICAL UNIVERSITY**

**SC2002: Object-Oriented Design & Programming  
2021/2022 SEMESTER 1 Group Assignment Report**

**Building an OO Application  
MOvie Booking and LIsting Management Application  
(MOBLIMA)**

**Lab Group: SS10 Group 2**

**Group Members:**



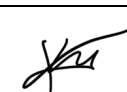

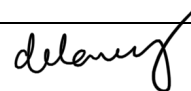
AGNES TAN ZIQI (U2121633F)  
ALAN WONG KUAN MING (U2121088B)  
ANG JUN YING KEITH (U2121419F)  
LEE ZHIYONG (U2122492A)  
QUEK YING EN DELANEY (U2121440D)

### **Declaration of Original Work for 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 (CE2002 or CZ2002)	Lab Group	Signature /Date
<b>AGNES TAN ZIQI</b>	SC2002	SS10	 13/11/2022
<b>ALAN WONG KUAN MING</b>	SC2002	SS10	 13/11/2022
<b>ANG JUN YING KEITH</b>	SC2002	SS10	 13/11/2022
<b>LEE ZHIYONG</b>	SC2002	SS10	 13/11/2022
<b>QUEK YING EN DELANEY</b>	SC2002	SS10	 13/11/2022

## **Design Considerations**

Our application will start by asking the user to choose between two options, a movie goer or admin staff. Logging in as staff will allow the user to access functionalities like editing movies, cinemas, showtimes, etc. Movie\_Goers are not allowed to access these functionalities as we will not display those options if the user does not log in.

### **Assumptions made:**

1. One member has only one unique phone number – we use the phone number as the key for Person class
2. Each cinema only shows 1 movie at all showtimes.

### **Principles and OOP concepts used:**

#### **Single Responsibility Principle**

In our application, the Price class consists of all the information required to calculate the price of each type of ticket. The sole responsibility of this class is to produce an array of these prices for TransactionBooking class to use.

Seating class also adheres to the single responsibility principle as it is only responsible for booking a seat and nothing else.

#### **Interface Segregation Principle, Dependency Injection Principle, and Open-Closed Principle**

In our MOBLIMA, we define a base interface Menu\_Interface and classes like Menu\_Admin Menu\_User, Menu\_Start. These classes implement the Menu interface and define their own versions of the abstract methods in the Menu interface (eg. ...). The interface is essentially closed for modification but opens for extension as any kind of menu can implement this interface and modifications specific to who the menu is for will be done in the class itself and not the interface. This also means that the higher-level modules, (Menu\_Admin and

Menu\_User) do not depend upon the low-level modules but rather depend upon abstractions. By using an interface, it keeps coupling low and the impact of change is smaller, which promotes reusability.

### Inheritance and Liskov Substitution Principle

We have implemented a subclass class MovieEdit which inherits from Movie class. Movie class contains all the necessary accessor methods while MovieEdit subclass has additional mutator methods for admins to be able to make changes to movie details. This also encourages SRP because Movie is only responsible for viewing movie details and the editing is handed over to MovieEdit subclass. This also promotes LSP as MovieEdit subclass can make good use of the methods in Movie class. This is because before changes are made, one would usually like to view the details before modifying it, invoking the use of the accessor methods defined in Movie class before using the mutator methods in MovieEdit subclass.

### Encapsulation

In several of our classes, we have helper functions like read and write to help classes access and mutate flat file content. These functions are made private so that functions outside of the class cannot access as not every function should be able to read and edit a flat file.

All of our attributes in classes are declared as private and implements public get and set methods to access and mutate these attributes.

## **Proposed Future Features**

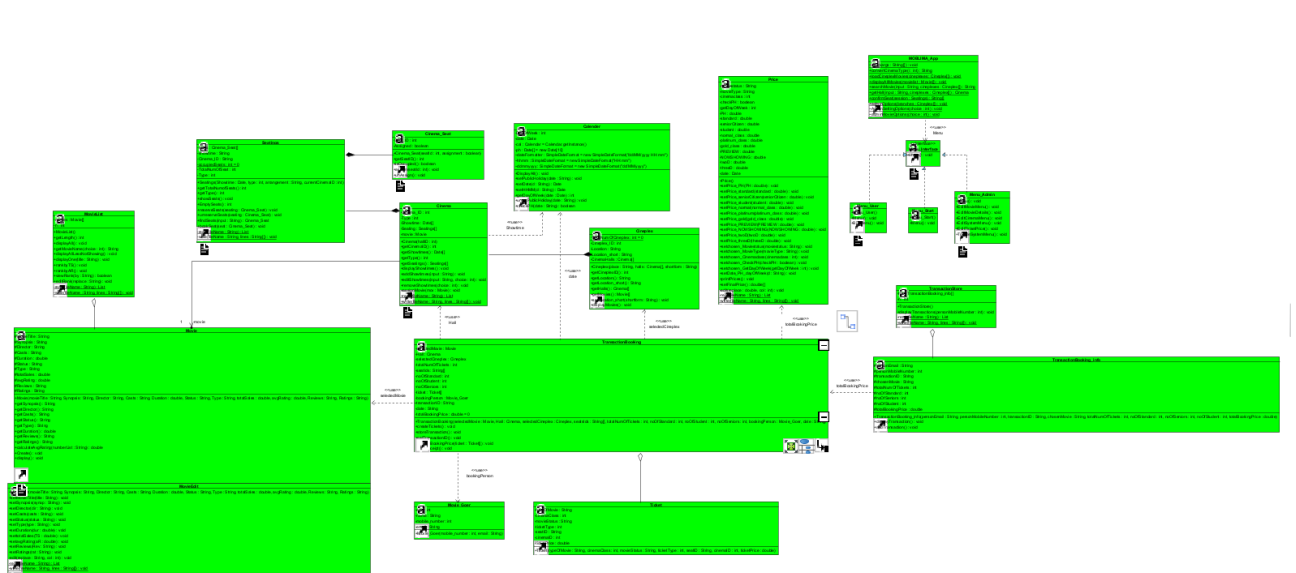
### 1) Member privilege

In our design, we regard all users at the same level. In reality, users belonging to a certain organization get special prices and are entitled to certain promotions. For example, Shaw theatres have a separate login function for SAFRA members which unlocks special ticket prices just for them. We can create another class just for these members which inherit from our Movie\_Goer class. Objects of this class will be created when a user logs in and can then enjoy a lower ticket price which will be calculated in our Price class. (eg. 30% off)

### 2) Different movies in one cinema for different showtimes

In reality, different cinemas play different movies at different showtimes. In our case, we assume that one cinema plays the same movie for all the showtimes on that same day. To implement this, we will need to use an array of Movie objects as an attribute in our Cinema class and adjust our flatfiles accordingly.

## UML Class Diagram (separate file outside of report)

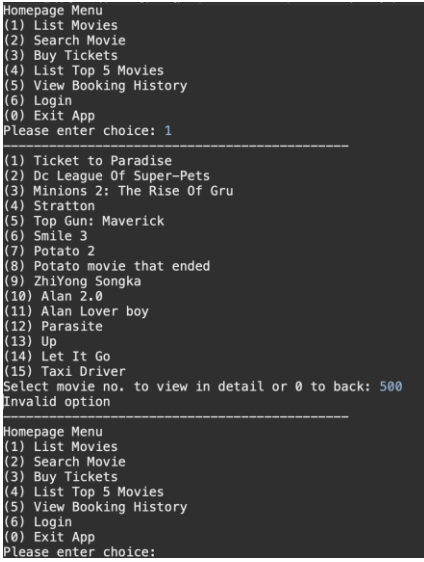
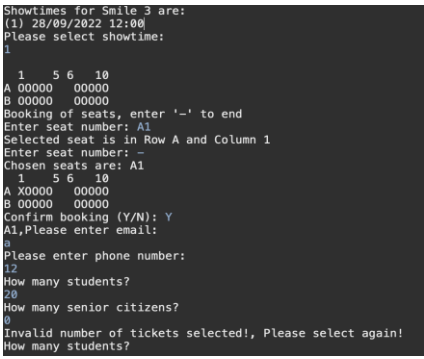


## Testing

### Testcases demonstrated in the video

- Listing all movies and viewing one movie details
- Check seat availability and trying to book an unavailable seat
- View booking history
- List top 5 movies (both ways)
- Login as admin incorrectly and correctly
- Create/update/remove movie listing
- Create/update/remove cinema showtimes and movies to be shown
- Configure system settings (set public holiday, edit public holiday pricing and toggling top 5 ranking)

### Testcases not demonstrated in video

S/N	Test case description	Expected Outcome	Actual Outcome
1	List movie and choose a non-existent option <ul style="list-style-type: none"> <li>- Only 15 movies present in movie list</li> <li>- Chose movie number 500</li> </ul>	Prints appropriate error message  And ask to select menu again	 <pre> Homepage Menu (1) List Movies (2) Search Movie (3) Buy Tickets (4) List Top 5 Movies (5) View Booking History (6) Login (0) Exit App Please enter choice: 1 ----- (1) Ticket to Paradise (2) Dc League Of Super-Pets (3) Minions 2: The Rise Of Gru (4) Stratton (5) Top Gun: Maverick (6) Smile 3 (7) Potato 2 (8) Potato movie that ended (9) ZhiYong Songka (10) Alan 2.0 (11) Alan Lover boy (12) Parasite (13) Up (14) Let It Go (15) Taxi Driver Select movie no. to view in detail or 0 to back: 500 Invalid option ----- Homepage Menu (1) List Movies (2) Search Movie (3) Buy Tickets (4) List Top 5 Movies (5) View Booking History (6) Login (0) Exit App Please enter choice:           </pre>
2	Choosing more ticket types than expected <ul style="list-style-type: none"> <li>- Selected to buy 1 seat</li> <li>- Chose 20 tickets</li> </ul>	Display error message and ask to select tickets again	 <pre> Showtimes for Smile 3 are: (1) 28/09/2022 12:00 Please select showtime: 1 1 5 6 10 A 00000 00000 B 00000 00000 Booking of seats, enter '-' to end Enter seat number: A1 Selected seat is in Row A and Column 1 Enter seat number: - Chosen seats are: A1 1 5 6 10 A X0000 00000 B 00000 00000 Confirm booking (Y/N): Y A1, Please enter email: a Please enter phone number: 12 How many students? 20 How many senior citizens? 0 Invalid number of tickets selected!, Please select again! How many students?           </pre>

3	Keying in a non-existent mobile number to view booking history	Shows there is no transaction booking	<pre> (4) List Top 5 Movies (5) View Booking History (6) Login (0) Exit App Please enter choice: 5 ----- Please enter phone number: 00000 Past Transactions: Sorry! We have no transactions with 0 ----- </pre>
4	Adding reviews/ratings	Input is appended behind existing reviews	<pre> (10) Reviews (11) Ratings (0) Exit Edit Movie Menu Select option to edit: 10 Edit review in progressNot bad Reviews updated! (10) Taxi Driver (11) SC2002 Project Select movie no. to view in detail or 0 to back: 11 Selected movie is:SC2002 Project SC2002 Project SHOWING Running Time: 120.0 Type: 2D Synopsis: Filled synopsis Main Cast: US Reviews: Good Movie, Not bad Ratings: 1,2,3 </pre>
5	Selecting a non-existent movie while booking seats	Error message showing that movie is non-existence	<pre> Homepage Menu (1) List Movies (2) Search Movie (3) Buy Tickets (4) List Top 5 Movies (5) View Booking History (6) Login (0) Exit App Please enter choice: 3 ----- Please select a choice a movie to watch 1. Ticket to Paradise 2. Dc League Of Super-Pets 3. Minions 2: The Rise Of Gru 4. Stratton 5. Top Gun: Maverick 6. Smile 3 7. SC2002 Project 8 Buy Menu (1) View movie details (2) Check movie availability (3) Buy tickets (0) Exit Search Menu 1 No such movie in database. </pre>
6	Searching for a movie not in database	Tell user there is no such movie	<pre> Homepage Menu (1) List Movies (2) Search Movie (3) Buy Tickets (4) List Top 5 Movies (5) View Booking History (6) Login (0) Exit App Please enter choice: 2 ----- Enter movie name: Spiderman No such movie in database. ----- Homepage Menu </pre>



7	When editing prices, input negative number	Will show error message and prompt for input again	<pre> (3) Edit Ticket Price Menu Enter your choice: 2 Enter new pricing: -20 Please enter an appropriate amount  PublicHoliday 70.0 Standard Price Senior Citizen Student 6.0   5.0   4.0 2D   3D 1.0   2.0 Standard   Platinum   Gold 1.0   1.5   1.25 Preview   Now Showing 1.5   1.0 Edit Ticket Price Menu (1) Edit Public Holiday Pricing (2) Edit Standard Pricing (3) Edit Senior Citizen Pricing </pre>
---	---	--	---

**Video Link:** <https://youtu.be/xVxl38MBmSs>