SC2002 MOBLIMA Project

Chen Ruxing, Kwa Wee Teck, Wong Chu Feng, Yuen Jia Qing Ivan

School of Computer Science and Engineering, Nanyang Technological University

Lab group: MACS

Assignment Group 8

SC2002 MOBLIMA Project

MOBLIMA s an application to computerize the process of making online booking and purchase of movie tickets, listing of movies and sale reporting. It will be used by the movie-goers and cinema staff. The application acts as a centralized ‘location’ for making online bookings for all the Cineplexes in different location managed by the vendor.. We will explain our design concepts and analyse our implementation of the program.

# Design considerations

In Java, there are 5 SOLID design principles:

* Single Responsibility Princple
* Open Closed Principle
* Liskov Substitution Princple
* Interface Segregation Princple
* Dependency Inversion Princple

These 5 principles ensure that the code written is “up to standard”. We will examine all 5 of them.

## Single Responsibility Princple

This principle states that the classes should be responsible for a single part or functionality of the program. This is true in our MOBLIMA program, with reference to the UML diagram(attached at the back), each class represents one function. For example, we have the program class itself, MoblimaApp, base classes like movie and Customer define a single entity and have cinemaApp

## Open Closed Princple

This principle states that classes, modules and functions should be open for extension but closed for modification. The base classes in this project like, video, Showtime, Review, Seat can be used for extension. For example, if a user wants to create another project for the zoo, it is possible to implement the above mentioned classes. If the zoo requires a trailer, video class can be implemented. In a zoo with different showtimes for animals, one can use the Showtime class and Seat class to store data of where the visitors are going to sit at different show timings. Review class can also be used for the visitors to give their feedback after the show.

## Liskov Substitution Princple

This principle ensures that when a base class is used for extension, the functions’ properties are not modified and the behaviour of the overridden methods must not change. In the movie class, it extends the base class video where it inherits the attributes of the video like name, the director, the cast and the video description and its functions to assign data to the mentioned attributes. The extension is done without modifying the original functions’ purpose.

## Interface Segregation Princple

This principle states that a user is not forced to implement methods that they will not use. As this project was built from ground up, we expanded the functionalities and attributes when it is required. Thus, there is no overlap of functions that are left “unused”.

## Dependency Inversion Princple

This principle states that dependencies within the system are built based on abstractions. In our project, movie classes extends class video. Since a movie and a video represents a “is-a” relationship, a movie is an extension of a video with more attributes like hall number, price, parental rating guidance, sales number, rating and it’s status (preview, coming\_soon, now\_showing and end\_of\_showing).

# Assumptions

1. This is a single-user application and concurrent access is not considered.
2. The currency will be in Singapore Dollar (SGD) and inclusive of Good and Services Tax (GST).
3. Payment will always be successful.
4. There is no need to interface with external system, e.g. Payment, printer, etc.
5. Online purchase is made without validation of identity or age.
6. All movies are no longer than 3 hours.
7. Each cinema hall will only show one movie.

# Further improvements

Looking back, 2 new features that we can implement are:

1. Emailing booking ticket to user. Since we already have the user’s email, after they have made a booking, automatic sending of the booking ticket can be made possible instead of the user having to use the “View booking history” function. With that, it can be extended to emailing the user giving them notifications to remind them of the upcoming movie they have booked.
2. User can search movies with different filters. When user is choosing a movie, he search by movie type. For example, user can input “3D” and “PG13” to search for any movies that are in those categories.

# Screen captures

* Test cases were included in the demonstration video.

UML Class Diagram

­Diagram

Description automatically generated