

# COMP642 Advanced Programming

## Software Project Part 2 – Implementing the Design

---

<b>Worth:</b>	40%
<b>Due:</b>	<b>Friday, 3 Nov 2023 5:00pm.</b>
<b>Late Penalty:</b>	Work not received by the due time attracts an immediate penalty of up to 25% of the marks available. No work will be accepted after <b>Sunday, 5 Nov 2023 5:00pm.</b>
<b>Submission:</b>	Zip your completed files and submit the .zip through the link on COMP642 LEARN page.

### Overview

Create a Python application which uses the model and controller classes that you designed in Project Part 1 to implement an online movie ticket booking system. Please refer to the feedback provided for Project Part 1 to improve your design before developing this project.

### Requirements

1. Adapt as necessary your design from Project Part 1 so that the following scenarios can be implemented for guests, customers and front desk staff and admin staff. Create the necessary files to populate the application.

#### Guests

1. Guests navigate to the online movie ticket booking system.
2. Guests browse/search movies based on title, language, genre, and release date.
3. Guests view selected movie.
4. To book a movie, guests must register as a new member.

#### Customers (booking movie tickets)

1. Customers navigate to the online movie ticket booking system.
2. Customers log in to the system.
3. Customers browse/search movies based on title, language, genre, and release date.
4. Customers view selected movie.
5. Customers view the selected movie schedules.
6. Customers select the date and time for the selected movie screening.
7. Customers make a booking.
8. Customers select one or more available seats, according to their preferred seat location.
9. Customers add discount coupon if any.
10. Customers pay for the movie tickets using cash, credit card or debit card.
11. Customers get a notification of the booking.

**Customers (cancelling movie tickets)**

1. Customers navigate to the online movie ticket booking system.
2. Customers log in to the system.
3. Customers click on the booking tab.
4. Customers click on the selected booking to cancel.
5. Customers get a notification of the cancellation.

**Front Desk Staff (Book movie tickets)**

1. Staff member navigates to the online movie ticket booking system.
2. Staff member logs in to the system.
3. Staff member selects the movie screening and make a booking.
4. Staff member selects one or more available seats, according to the customer's preferred seat location.
5. Staff member adds discount coupon if any.
6. Staff member receives payment from the customer in cash, credit card, or debit card.

**Front Desk Staff (Cancel movie tickets)**

1. Staff member navigates to the online movie ticket booking system.
2. Staff member logs in to the system.
3. Staff clicks on the booking tab.
4. Staff member selects booking to cancel.
5. Staff member provides refund to the customer.

**Admin Staff (adding new movie)**

1. Admin navigates to the online movie ticket booking system.
2. Admin logs in to the system.
3. Admin clicks on add new movie.
4. Admin enters the relevant details for the new movie.
5. Admin confirms adding the new movie.

**Admin Staff (adding a show for a movie)**

1. Admin navigates to the online movie ticket booking system.
2. Admin logs in to the system.
3. Admin selects a movie.
4. Admin adds a screening to that movie.
5. Admin confirms adding that screening.

**Admin Staff (cancelling a movie)**

1. Admin navigates to the online movie ticket booking system.
2. Admin logs in to the system.
3. Admin clicks on cancel a movie.
4. Admin selects the movie to cancel.
5. Admin confirms cancelling the selected movie.

### **Admin Staff (cancelling a show)**

1. Admin navigates to the online movie ticket booking system.
  2. Admin logs in to the system.
  3. Admin selects a movie.
  4. Admin selects the screening for that movie to cancel.
  5. Admin confirms cancelling the selected screening for the selected movie and issues a refund to those customers who have booked the screening.
- 
2. Implement an appropriately designed view. Your view should be an interface with appropriate controls, useful feedback and exception handling. You may use tkinter or web forms to implement the view.
  3. Write and test the code for all the model classes and the controller in your design using **pytest**.
  4. Include test cases to test the scenarios described above.
  5. Your code must be clear and easy to maintain and appropriately commented.
  6. Your application **does not** have to provide facilities for admin staff to add new customers, delete existing customers or amend customer/movie details. It also **does not** have to provide facilities for a customer to amend their details.

## Marking

Criteria	Marks (out of 100)
Application functionalities: <ul style="list-style-type: none"><li>• Guest (5)</li><li>• Customer (10)</li><li>• Front Desk Staff (5)</li><li>• Admin Staff (5)</li></ul>	25
Application Interface: <ul style="list-style-type: none"><li>• Interface Design (7)</li><li>• Use of appropriate widgets (3)</li><li>• Input validation and feedback (5)</li></ul>	15
Code <ul style="list-style-type: none"><li>• Model classes (25)</li><li>• Controller (15)</li><li>• View (5)</li><li>• Error handling and preventions (5)</li><li>• Testing (10)</li></ul>	60