UECS2354 SOFTWARE TESTING
202406 – ASSIGNMENT (updated)

Project details:

You are to develop and test a **movie ticket booking system**. Consider the following requirements (the draft interface layout as in the Appendix A):

- A user must be a registered member in order to make a booking through this system.
- User can select the movies from the listing.
- User then can select the available showtime.
- The details of the movie date, time, the cinema hall number and available seats will be displayed.
- Booking can be made according to the available number of seats.
- User then key in number of tickets for each user category.
- The total number of tickets and total price will be calculated and shown accordingly.
- The ticket pricing is based on the following conditions:

  (a) User category:
  - Adult: Normal ticket price
  - OKU: Have 5% discount on the normal ticket price
  - Senior (55 years and above): RM9 per movie, except on Wednesday
  - Student (with valid student card): RM9 per movie, for showtime before 6pm
  - Children (3-12 years old): RM9 per movie, except on Wednesday

  (b) Showtime:
  - Before 1pm on weekdays (except on Wednesday): RM9
  - Every Wednesday, all day long: RM8
  - Weekend (Saturday & Sunday) and Public Holidays: additional RM2 on top of the normal price

  (c) Special movie category:
  - Normal or 2D: Normal ticket price
  - 3D: additional RM4 on top of the normal price
  - IMAX: additional RM4 on top of the normal price

- Payment can be made through credit card or online banking through a third-party payment gateway.
- Reservation considers successful when payment made successfully.
- Upon successful reservation, tickets will be emailed to user with movie details, seat number, payment details and QR code.
- Users are allowed to cancel their reservation ONLY if the reservation or payment is unsuccessful.

UECS2354 SOFTWARE TESTING
202406 – ASSIGNMENT (updated)

Assignment Deliverables:

Part A: Test Plan

Based on the project details and requirement, plan for the testing activities. Your test plan must contain scope, objective, test basis, features to be tested and not to be tested, test conditions, test entry and exit criteria. Refer to the test plan template **and guideline uploaded in WBLE (already discussed in the lecture class)**

Part B: Test Design

Based on the project details, **create decision table(s)**, and **design the test cases** for the testing phase. There must be clear description in every test case. Refer to the test case template **and guideline uploaded in WBLE (already discussed in the lecture class)**.

Part C: Application code and Test Code

Jar files: Place all JAR files (jUnitParams & Mockito) in C:\ jar_files

1. Application code:
   Your application code must have the following modules:
   i. cinema hall
      - store details such as Hall Number, No of Seats, No of Available Seats, No of Booked Seats, Status of the Hall
      - Status of the Hall: Fully Booked, Available, Not Available, Repair

   ii. movie:
      - Store details such as movie name, category, normal ticket price
      - Movie categories are: Normal, 3D, IMAX

   iii. showtime:
      - Store details such as movie name, time, date, hall number, status
      - Status of the showtime: Available, Not Available, Fully Booked, Cancelled

   iv. member:
      - Store details of registered member, such as name, birthday, email

   v. booking:
      - store details such as booking id, member name, email, movie details (name, category), showtime (time, date, hall), number of seats booked total price, status
      - Status of booking: booked, payment successful, payment unsuccessful, cancelled
      - **The total ticket price calculation should be done in this module.**
      - **Once the total ticket price is being calculated, the details will be sent to Payment module's `makePayment()` method**

Extra details on the modules not ready for testing:

vi. payment: **(Not Ready for Testing)**

- Store details such as total price, payment status, payment method
- Once payment is successful, will send the status to booking and email notification module
- Payment module will have the following methods:

(i)
```
public String makePayment(int bookingID,
double totalTicketPrice, String
userEmail)
```

*/**this method will call the third-party payment getaway to proceed with payment (either credit card or online payment methods) and will return the payment status if it is successful or not successful*/*

*/**if the payment is successful, this method then will call the following methods:*/*

1. Email notification module:

(ii)
```
public void sendEmail(int bookingID,
String paymentStatus, String userEmail)
```

//this method will trigger an email to user on the payment status

2. Booking module:

(iii)
```
public String updatePaymentStatus (int
bookingID, String paymentStatus)
```

//this method will update the payment status in booking module

vii. email notification: **(Not Ready for Testing)**

- Email will be triggered if the payment status is successful.
- Email Notification module will have the following methods:

(i)
```
public void sendEmail(int bookingID, String
paymentStatus, String userEmail)
```

//this method will trigger an email to user on the payment status

Create a separate class for each module. You are responsible to complete and implement the classes in the application.

Note: **The 'payment' module and 'email notification' modules are not ready for testing.** However, you can use stub and driver to test the integration with in these modules.

2. Test code:
   Write test code using jUnit Framework and Mockito to test the application code.

Note:
The particular testing issues that you need to pay attention to and their associated marks are shown in the marking scheme (*STA_Marking.docx*). Include comments in test code to clearly specify which aspect of these testing issues you are addressing. For example, when creating parameterised tests, include comments to state the approach that you are using (e.g. boundary value analysis, etc.).

The focus is NOT to write as many tests as possible, but rather to create the right number of tests necessary to verify the functionality of the methods. For example, if you are using equivalence partitioning, writing tests that use all the inputs within the same partition that produce identical output from a method is a waste of space.

Submission details:

Due date:
06 September 2024 (Friday – W12)

This is a group assignment. Form a group of 4 members in a group. Register your group member's name in the google sheet:
https://docs.google.com/spreadsheets/d/1n9tq69MA98ai6aTbnDWx3XySLgeaa2PP7JzNA0Yrqxw/edit?usp=sharing

Submit the following items through WBLE (link will be created in WBLE course page).
Create a folder with your *GroupNumber*, place the following items and zip the folder before submit.

1. **Java project folder**:
   Archive of Eclipse **project folder** that contains source code for both the application code and the test code. The application code and test code should be placed in two separate source directories.
   - Project Name: *GroupNumber*

   Note: Do not include jar files as it may be blocked.

2. **Report**:
   Your report should contain the following:
   - Marking sheet (.docx) as the front page of the documentation
   - Assumptions (if any)
   - Class diagram for the application code *(If you update, add, remove any details for the classes given above, reflect it in the class diagram)*
   - The application code and jUnit test code, should be attached at the end of the documentation.

3. **Test plan (Part A) – (in word or pdf format)**
4. **Decision table (Part B) – (in excel format)**
5. **Test cases (Part B) – (in excel format)**

Total mark

The total mark of this practical assignment is 100. The 100 marks will contribute 30% of your final mark. It's your responsibility to understand the requirements of the tasks and prepare well for your submission. You might be asked questions about the works you submit to ensure that you understand them.

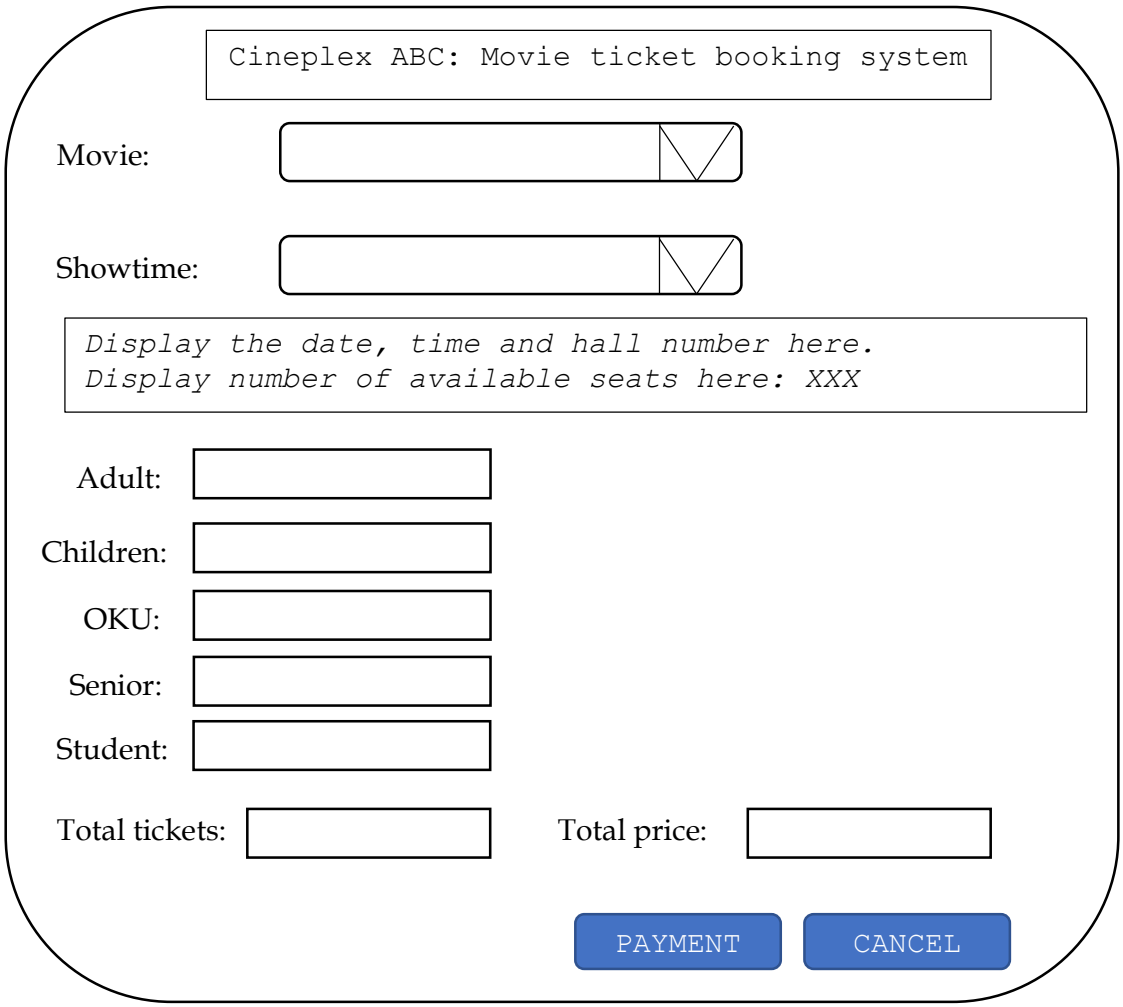
Late Submission

No late submission of assignment is allowed. Assignment received after the due date without valid reasons will be penalized using the following policy: 5 marks will be deducted for every day the assignment is overdue.

Plagiarism

It is important that your solutions to the practical assignment be your own work. It is perfectly acceptable to seek help and advice when completing the practical assignment, but this must not be taken to the point where what is submitted is in part someone else's work. Any group found to have committed plagiarism or cheating by copying program codes from other sources will be given a failing grade.

Appendix A:


Cineplex ABC: Movie ticket booking system
Movie:
Showtime:
Display the date, time and hall number here.
Display number of available seats here: XXX
Adult:
Children:
OKU:
Senior:
Student:
Total tickets:
Total price:
PAYMENT
CANCEL