Exception Handling Lab Exercise

(This is a group task)

Imagine you are developing an online movie ticket reservation system for a popular cinema chain. The system allows customers to browse a list of movies, each identified by a unique movie code, along with its available dates and showtimes. For simplicity, assume there are six showtimes each day: two in the morning, two in the afternoon, and two in the evening. Customers can select a movie, date, and showtime, specify the number of tickets, and choose their preferred seats.

However, during the booking process, various errors can occur that the system should handle gracefully. For example:

- If a customer enters a movie code that does not exist, the system should throw an InvalidMovieCodeException and prompt the user to reenter the correct code.
- Customers may accidentally select an invalid date or showtime. In such cases, appropriate exceptions should be thrown, giving users the chance to correct their input.
- The system should ensure that the number of tickets entered is a positive integer and does not exceed the available seats. If the quantity is invalid, prompt the user to enter a valid number.
- If a customer tries to reserve more tickets than are available for a particular showtime, the system should throw an OverbookingException and alert the user.

Additionally, the system must handle user inactivity. If a user remains inactive for a specified period, their session should be automatically saved, ensuring that their selected seats and ticket quantities remain reserved for a limited time.

Once the user completes the booking, the system should calculate the total bill and prompt the user to enter their email address for billing purposes. The bill should then be generated as a PDF and emailed to the user.

In this lab, your database is a CSV file given with the lab description.

Your Tasks:

1. Identify all the functional and non-functional requirements of the movie ticket reservation system mentioned above.
2. Design and implement the Java-based system for the given scenario.
   a. You must use your knowledge of object-oriented programming to design the solution for this lab.
   b. You should include all the classes in your program in a single file. The class containing the main method should be named "MovieTicketReservationGroup_<<Your Group Name>>". Please note that including all classes in a single file is considered bad coding practice; however, we do it in this lab to simplify the grading process.
   c. At the top of your file, include a Javadoc comment that lists your identified requirements and explains the overall functionality of the program.

d. Each class and method in the program should also be accompanied by a Javadoc comment explaining its functionality and parameters.
e. Ensure your code is readable with comments and proper class names, variables and methods.
f. You must use all the best practices you have learned so far.

Save your Java code as a PDF file, rename it with your group name, and upload it to the link.

THIS IS A TURNITIN ASSIGNMENT, AND CODE PLAGIARISM WILL BE CHECKED AUTOMATICALLY. YOUR SIMILARITY SHOULD BE LESS THAN 20%, AND IF NOT, YOUR SUBMISSION WILL NOT BE GRADED.