Project Title	**Online Movie ticket booking**
Technologies	MERN
Domain	Entertainment
Project Level	Difficult

Table

Contents

1. Problem Statement:

The online ticket ordering system was created to give customers another option for prepurchasing movie tickets. It is an automated system, and to demonstrate it, we will give use case diagrams that include the system's requirements and features, as well as some process descriptions and a data dictionary.

1.1. Introduction:

An online movie ticket serves as a conduit between the potential customer and the organization in charge of online ticket sales. It seeks to increase efficiency and minimizes the complexity involved as much as possible.

2.Goal of our system:

- Record performance details.
- Record Customer details.
- Record tickets sold.
- Print tickets.
- Print address labels for telephone booking.

3.Purpose:

seeking for a complete online solution to manage the purchase of online movie tickets.

- Online movie statistics.
- Online movie booking.
- Online movie review.

4.scope:

- Customers can buy ticket online and cancel the seats at a suitable time (1 hour before the show).
- Before buying tickets, all the customer has to register and become a member.
- If the customers cancelling tickets will not be given money back; instead, the amount of money will be recorded in the customer account for further use.
- Customer will login to the system.
- Customer will choose the seats positions.
- Customer will choose the film, time and venue.
- Customer needs a card to complete a buying transaction.
- Confirm and show the transaction number to the customer • Customer can modify the data.

4.1Functional Requirements:

- User should be able to view the list of movies which are running near to his location (based on GPS).
- User should be able to select the seat as per his choice.
- User should have different types of payment.

4.2: Registration and Login:

- Customer must be registered before watch the movie.
- With valid user id and password, customer login to the system.

4.3: Changes to cart:

User can change or cancel the booking.

4.4: Payment:

There are many types of secure billing will be available.

5. Technical issues:

System work on client Server Architecture. It will need internet server.

6. Project Evaluation metrics:

6.1. Code:

- You are supposed to write code in a modular fashion ● Safe: It can be used without causing harm.
- Testable: It can be tested at the code level.
- Maintainable: It can be maintained, even as your codebase grows.
- Portable: It works similarly in every environment (operating system).
- You have to maintain your code on GitHub.
- You must keep your GitHub repo public so anyone can check your code.
- Proper readme file you have to maintain for any project development.
- You should include the basic workflow and execution of the entire project in the readme file on GitHub.
- Follow the coding standards.

6.2. Database:

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

6.3. API Details or User Interface:

You have to expose your complete solution as an API or try to create a user interface for your model testing. Anything will be fine for us.

6.4. Deployment:

Deploy the application on your preferred service.

6.5. Solutions Design:

You have to submit complete solution design strategies in High-level Document (HLD), Lowlevel Document (LLD), and Wireframe documents.

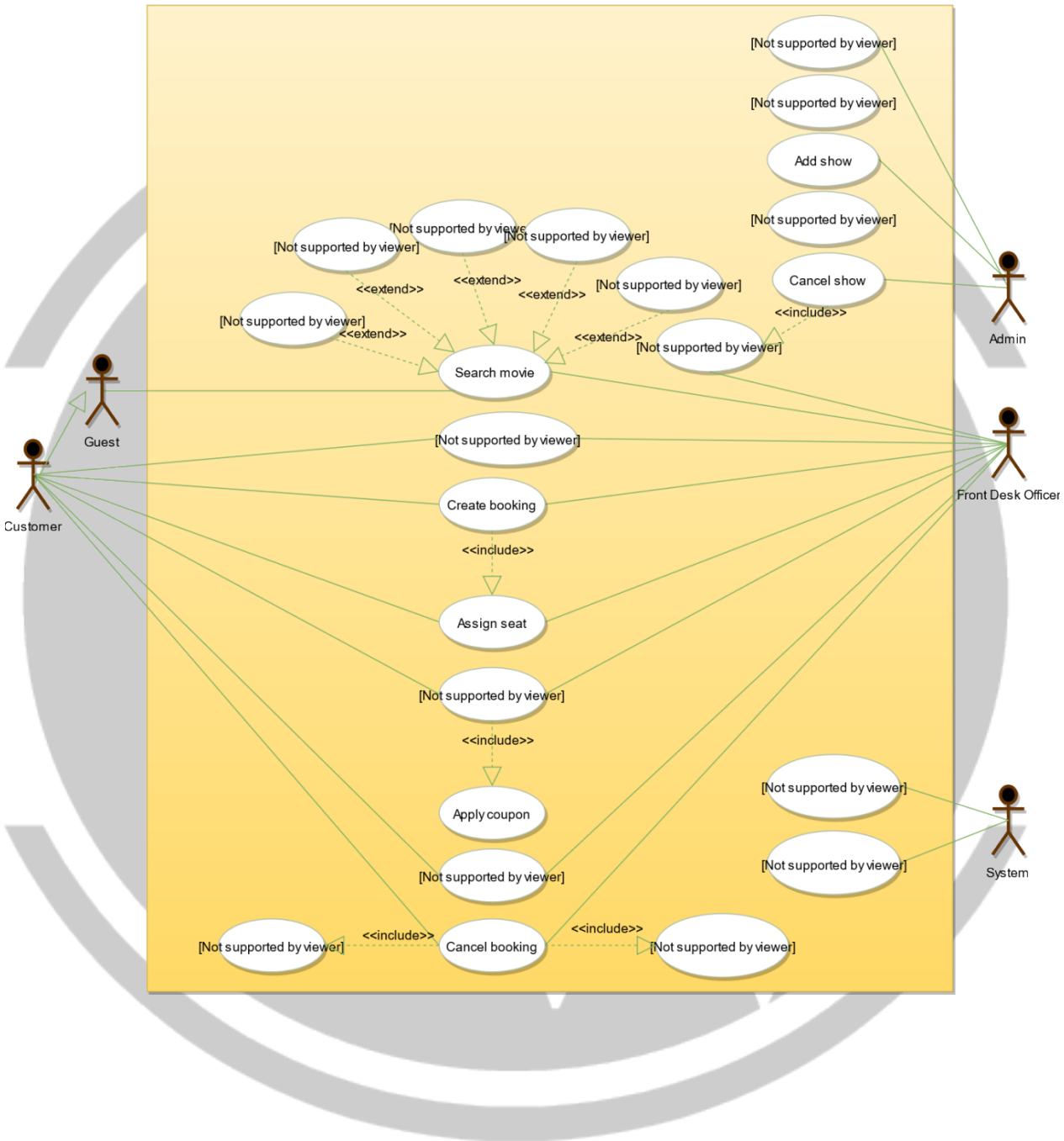
6.6. System Architecture:

You have to submit a system architecture design in your wireframe document and architecture document.

6.7. Optimization of solutions:

Try to optimize your solution on the code level, and architecture level, and mention all of these things in your final submission.

7. Use case Diagram:



8. Submission requirements:

8.1. High-level Document:

You have to create a high-level document design for your project. You can reference the HLD form below the link.

Sample link: HLD Document Link

8.2. Low-level document:

You have to create a Low-level document design for your project; you can refer to the LLD from the link below.

Sample link: LLD Document Link

8.3. Architecture:

You have to create an Architecture document design for your project; you can refer to the Architecture from the link below.

Sample link: Architecture sample link

8.4. Wireframe:

You have to create a Wireframe document design for your project; refer to the Wireframe from the link below.

Demo link: Wireframe Document Link

8.5. Project code:

You have to submit your code to the GitHub repo in your dashboard when the final submission of your project.

Demo link: Project code sample link

8.6. Detail project report:

You have to create a detailed project report and submit that document as per the given sample.

Demo link: DPR sample link

8.7. Project demo video:

You have to record a project demo video for at least 5 Minutes and submit that link.

The project LinkedIn a post:

You have to post your project details on LinkedIn and submit that post link in your dashboard in your respective field.