Use case - REZERVARI LOCURI

O institutie teatrala pune la dispozitia spectatorilor un sistem pentru rezervarea locurilor la spectacole. In fiecare zi, institutia are o singura reprezentatie, la care spectatorii pot rezerva locuri începand de dimineata. După fiecare rezervare, toate terminalele vor afisa situatia actualizata referitor la ocuparea salii si pret. Terminalele puse la dispozitia spectatorilor afiseaza intreaga configuratie a salii, precizând pentru fiecare loc pozitia, numarul, pretul si starea (liber sau rezervat). Folosind un astfel de terminal, spectatorul isi poate introduce datele personale, poate selecta unul sau mai multe locuri si poate declansa un buton pentru rezervarea lor. După fiecare rezervare, toate terminalele vor afisa situatia actualizata referitor la ocuparea salii.

ID and name	UC-01: Reserve Seats		
Primary actor	Spectator	Secondary actors	None
Description	This use case describes the process of a spectator reserving seats for a theatrical performance using the reservation system provided by the theater institution.		
Trigger	Spectator initiates the seat reservation process.		
Preconditions	- The theatrical institution has a scheduled performance for the day. - The reservation system is operational. - The spectator has accessed a terminal.		
Postconditions	- Reserved seats are recorded in the system. - The theater's seat occupancy status is updated.		
Normal flow	1. Spectator approaches the terminal and starts the reservation process. 2. The system displays the current configuration of the theater hall, showing seat positions, seat numbers, prices, and availability status (free or reserved). 3. Spectator selects one or more seats by indicating their position, number, and confirms the reservation. 4. The system verifies seat availability. 5. If the selected seats are available, the system prompts the spectator to enter personal details for the reservation.		

	6. Spectator enters personal details (name, contact information, etc.). 7. The system confirms the reservation, displaying a summary of the reserved seats and the total cost. 8. Spectator confirms the reservation. 9. The system updates the seat availability status, marking the reserved seats as occupied. 10. All terminals display the updated seat occupancy status.
Alternative flows	**- Invalid Seat Selection:** - If the spectator selects an already reserved seat, the system prompts for a different selection. **- Cancellation:** - Spectator cancels the reservation process at any point before confirmation. - The system returns to the initial state, and no seats are reserved. **- System Unavailability:** - If the reservation system is down, an error message is displayed, and the spectator is prompted to try again later.
Exceptions	**- Invalid Personal Details:** - If the spectator provides incomplete or invalid personal details, the system prompts for correction. - The system does not proceed until valid details are entered. **- Payment Failure:** - If a payment step is involved, and the payment fails, the system prompts the spectator to try an alternative payment method or cancel the reservation. - The system does not proceed until a successful payment is made.

Descriptions of template fields:

- **ID and name:** Title should be descriptive and should usually begin with a verb, e.g. order, calculate, input, etc. ID can have any format but must be unique among all use cases.
- **Primary actor:** Person that wishes to accomplish a goal through the use of the system. Only a single primary actor per use case.
- **Secondary actors:** Actors that have an interest in the completion of the goal but that do not directly interact with the system.

- **Description:** Concise description of the purpose of the use case.
- **Trigger:** Condition internal or external to the system that prompts the use case to start.
- **Preconditions:** Conditions that must be true before the use case starts. Each should be labeled with an ID unique to the use case.
- **Postconditions:** Conditions that must be true after the use case ends normally. Each should be labeled with an ID unique to the use case.
- **Normal flow:** Detailed step-by-step description of the logical flow of the use case. It should describe an explicit two way interaction, with the system prompting for input and the actor responding accordingly. Each step should be numbered.
- **Alternative flows:** Flows that achieve the same goal as the normal flow but are expected to be less common or lower priority.
- **Exceptions:** Conditions that result in the normal flow ending prematurely due to an unrecoverable condition in the system. The condition that causes the flow should be clearly stated, as should be any other decisions that the actor must make in this situation.

ID and name	UC-02: Manage Seat Configuration		
Primary actor	Admin	Secondary actors	None
Description	This use case outlines the steps an admin takes to manage the seat configuration for performances in the theater institution's reservation system.		
Trigger	Admin initiates the seat configuration management process.		
Preconditions	- The admin has valid credentials and is logged into the admin panel. - The seat configuration system is accessible and operational.		
Postconditions	- The seat configuration is updated in the system. - Any changes made are reflected in the reservation system for upcoming performances.		
Normal flow	1. Admin accesses the admin panel and navigates to the "Seat Configuration" section. 2. The system displays the current layout of the theater hall, indicating seat positions, seat numbers, and their current configuration.		

	3. Admin selects the desired performance or date for which they want to manage the seat configuration. 4. The system allows the admin to add, remove, or modify seats on the selected date/performance. 5. Admin makes the necessary changes to the seat configuration, specifying positions, numbers, and any other relevant details. 6. The system validates the changes to ensure they conform to the theater's seating constraints. 7. If validation is successful, the admin confirms the changes. 8. The system updates the seat configuration for the selected date/performance. 9. All terminals reflect the updated seat configuration for future reservations.
Alternative flows	**- Undo Changes:** - If the admin wants to revert the changes made during the current session, they can undo the modifications before confirming. **- Cancel Configuration:** - At any point before confirmation, the admin can cancel the seat configuration process, and no changes will be applied.
Exceptions	**- Invalid Configuration:** - If the proposed seat configuration violates constraints (e.g., exceeds the maximum capacity of the theater), the system alerts the admin and prevents the changes until valid adjustments are made. **- System Unavailability:** - If the seat configuration system is temporarily unavailable, the admin is notified, and they are prompted to try again later.

Extra step: Traceability

For this extra step, you will add traceability information for each use case by adding a new field to the template:

Method-level traces	<fully.qualified.ClassName>#<methodName> ...

Any method that implements the functionality described in the normal flow, alternative flow or exceptions should be included in this field. This means that the method that is initially executed and any methods of any classes that the work is delegated to should be included.

Examples for previous use cases:

UC-1:

Method-level traces	my.company.ordering.MenuWidget#dateClicked my.company.ordering.MenuWidget#completeOrder my.company.ordering.InventoryInterface#checkInventory ...

UC-5:

Method-level traces	my.company.payroll.PayrollInterface#checkEligibility my.company.payroll.RegistrationForm#confirm ...