*Use Case Template for main Theater Booking use cases*

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC-1: Manage bookings | | |
| Primary actor | Admin | Secondary actors | Theater Booking System |
| Description | An Admin accesses the Theater Booking System from the Theater intranet or the internet, then he logs in his account, where he can see the bookings made, and can access the menu for updating information or making a report about the situation. | | |
| Trigger | An admin indicated that he wants to manage the bookings | | |
| Preconditions | PRE-1. Admin is logged in his account | | |
| Postconditions | POST-1. The changes made in the bookings are updated and stored  POST-2. The report of the seats is created | | |
| Normal flow | * 1. Admin checks the bookings  1. Admin requests to manage bookings 2. The System shows all the bookings made 3. The System shows the show events available 4. Admin requests the situation of empty seats 5. The System provides the requested information | | |
| Alternative flows | * 1. Admin changes the event show details  1. Steps 1-3 from normal flow 2. Admin requests to update/delete/add an event show 3. The system lets admin to add changes 4. The system stores all the changes 5. The system shows a message of “success” | | |
| Exceptions | 1.1.E1. There are no show events  1.The systems returns message  1.2.E1. There are no show events  1.The system returns message  1.2.E2. The update is invalid   1. The system returns message   2a. If the admin accepts it the system goes back to managing a booking menu  2b. Else the system waits for valid updates | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC-2: Make a booking | | |
| Primary actor | Spectator | Secondary actors | Theater Booking System |
| Description | A spectator accesses the Theater Booking System from the internet, then he logs in his account, where he can see the configuration of the event show space. He can select the free seats he wants to book, but before that he must introduce his personal data. | | |
| Trigger | A spectator indicated that he wants to make a booking | | |
| Preconditions | PRE-1. Spectator is logged in his account | | |
| Postconditions | POST-1. The bookings are updated and stored  POST-2. The number of seats is updated | | |
| Normal flow | * 1. Spectator selects make a booking to the show event   1. System shows the configuration of the theater and the show event  2.The spectator select the seats  3.The system shows a window of booking  4.The spectator introduce the personal data  5. The system make the booking and store it | | |
| Alternative flows | * 1. Spectator checks the event show details  1. Steps 1 from normal flow 2. The system shows that there are no empty seats 3. The spectator goes back | | |
| Exceptions | 1.1.E1. There are no show events  1.The systems returns message  1.1.E2. There are no seats   1. The system returns message   1.2.E1. There are no show events  1.The system returns message | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC-3: Login | | |
| Primary actor | Any theater participant | Secondary actors | Theater Booking System |
| Description | A Theater participant introduce his login data, then enters his account. | | |
| Trigger | A Theater participant indicates he wants to login | | |
| Preconditions | - | | |
| Postconditions | POST-1. The Theater participant is logged in his account | | |
| Normal flow | * 1. Theater participant logs in  1. Theater participants introduce his login data 2. The system checks if the data is valid 3. Theater participant logs in in his account | | |
| Alternative flows | * 1. Theater participant doesn’t have an account  1. Theater participant introduce his data 2. The system shows message that the account doesn’t exist 3. The system goes back | | |
| Exceptions | 1.1.E1. The username/password is incorrect  1.The systems returns message  1.2.E1. There is no account  1.The system returns message | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC-4: Introduce personal data | | |
| Primary actor | Spectator | Secondary actors | Theater Booking System |
| Description | A spectator accesses the Theater Booking System from the Theater intranet or the internet, then he logs in his account, where he can see the event show, and can make a booking by introducing his personal data. | | |
| Trigger | A spectator indicated that he wants to make a booking, and introduce his personal data | | |
| Preconditions | PRE-1. The spectator is logged in his account | | |
| Postconditions | POST-1. The personal data is saved for the booking | | |
| Normal flow | * 1. Spectator introduce his personal data  1. Spectator introduces the data 2. The system validates the data 3. The system saves the data 4. The system goes back to making a booking | | |
| Alternative flows | - | | |
| Exceptions | 1.1.E1. The data isn’t valid  1.The systems returns message | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC-5: Select seats | | |
| Primary actor | Spectator | Secondary actors | Theater Booking System |
| Description | A spectator accesses the Theater Booking System from the internet, then he logs in his account, where he can see the configuration of the event show space. He can select the free seats he wants to book, but before that he must introduce his personal data. | | |
| Trigger | A spectator indicates that he wants to select seats for booking seats at an event show | | |
| Preconditions | PRE-1. Admin is logged in his account  PRE-2. Spectator introduced his personal data | | |
| Postconditions | POST-1. The seats are selected and saved for the next step | | |
| Normal flow | * 1. Spectator selects empty seats  1. The spectator selects the seats he wants to book 2. The system saves the seats and returns spectator to make a booking step | | |
| Alternative flows | * 1. Spectator selects no empty seats  1. Step 1 from normal flow 2. The system returns message that the seats aren’t available | | |
| Exceptions | 1.1.E1. There is no personal data introduced  1.The systems returns message  1.2.E1. There is no personal data introduced  1.The system returns message  1.2.E2. The seats aren’t available   1. The system returns message   2a. If the spectator accepts it the system goes back to selecting seats  2b. Else the system returns spectator to his main page | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC-6: Add show event | | |
| Primary actor | Admin | Secondary actors | Theater Booking System |
| Description | An Admin accesses the Theater Booking System from the Theater intranet or the internet, then he logs in his account, where he can see the bookings made, and can access the menu for adding show event. | | |
| Trigger | An admin indicated that he wants to add a show event | | |
| Preconditions | PRE-1. Admin is logged in his account  PRE-2. Admin selected manage bookings | | |
| Postconditions | POST-1. The show event was added and stored | | |
| Normal flow | * 1. Admin adds a show event   1. Admin requests to add an event  2.The System shows the form for introducing  3.Admin puts the event details  4.The system validates the data introduced  5.The System adds the show event and stores it | | |
| Alternative flows | * 1. Admin wants to add an event, but one already exists  1. Step 1 from normal flow   2. The system shows a message or the admin  3. The system returns admin to main page | | |
| Exceptions | 1.1.E1. The data aren’t valid  1.The systems returns message  1.2.E1. There is an existing show event  1.The system returns message | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC-7: Delete show event | | |
| Primary actor | Admin | Secondary actors | Theater Booking System |
| Description | An Admin accesses the Theater Booking System from the Theater intranet or the internet, then he logs in his account, where he can see the bookings made, and can access the menu for deleting show event. | | |
| Trigger | An admin indicated that he wants to delete a show event | | |
| Preconditions | PRE-1. Admin is logged in his account  PRE-2. Admin selected manage bookings | | |
| Postconditions | POST-1. The show event was deleted from the database. | | |
| Normal flow | * 1. Admin deletes a show event   1. Admin requests to delete an event  2.The System search for that event and deletes it  3.The System shows message for deleting with “success” | | |
| Alternative flows | * 1. Admin wants to delete an event, but it doesn’t exist  1. Step 1 from normal flow   2. The system shows a message or the admin  3. The system returns admin to main page | | |
| Exceptions | 1.1.E1. The system didn’t find the show event  1.The systems returns message  1.2.E1. There is not an existing show event  1.The system returns message | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC-8: Update show event | | |
| Primary actor | Admin | Secondary actors | Theater Booking System |
| Description | An Admin accesses the Theater Booking System from the Theater intranet or the internet, then he logs in his account, where he can see the bookings made, and can access the menu for updating a show event. | | |
| Trigger | An admin indicated that he wants to update a show event | | |
| Preconditions | PRE-1. Admin is logged in his account  PRE-2. Admin selected manage bookings | | |
| Postconditions | POST-1. The show event was updated and stored | | |
| Normal flow | * 1. Admin updates a show event   1. Admin requests to update an event  2.The System searches the event and shows the form for introducing the new data  3.Admin puts the event details  4.The system validates the data introduced  5.The System updates the show event and stores it | | |
| Alternative flows | * 1. Admin wants to update an event, but the data aren’t valid  1. Steps 1-4 from normal flow   2. The system shows a message or the admin  3. The system returns admin to main page | | |
| Exceptions | 1.1.E1. The event doesn’t exist  1.The systems returns message  1.2.E1. The event doesn’t exist  1.The system returns message | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC-9: Count empty seats situation | | |
| Primary actor | Admin | Secondary actors | Theater Booking System |
| Description | An Admin accesses the Theater Booking System from the Theater intranet or the internet, then he logs in his account, where he can see the bookings made, and can access the menu for generating a report of the empty seats. | | |
| Trigger | An admin indicated that he wants to count empty seats situation | | |
| Preconditions | PRE-1. Admin is logged in his account  PRE-2. Admin selected manage bookings | | |
| Postconditions | POST-1. The report of the empty seats was generated and displayed | | |
| Normal flow | 1.1. Admin counts empty seats  1. Admin requests generating report of the seats  2.The System returns the empty seats situation report | | |
| Alternative flows | - | | |
| Exceptions | 1.1.E1. There is no show event  1.The systems returns message | | |

Planificarea cazurilor de utilizare:

Iteratia 1) Login, Select seats, Make a booking, Introduce personal data

Iteratia 2)Manage bookings, Add show event

Iteratia 3)Delete show event, Update show event, Count empty seats situation

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.