Theatre Reservation System

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-1: Login | | |
| **Primary actor** | Admin | **Secondary actors** | Theatre reservation system |
| **Description** | The admin of the application can log in to be able to manage the show: delete unwanted reservations, add a new show, update the configuration | | |
| **Trigger** | The admin needs to manage the show | | |
| **Preconditions** | * none | | |
| **Postconditions** | POST-1: The functionalities for the admin are displayed on the screen | | |
| **Normal flow** | **1.0. Login**  1. The Admin opens the application  2. The Admin enters the credentials  3. TRS checks the data and logs in the Admin | | |
| **Alternative flows** | * none | | |
| **Exceptions** | **1.0.E1: The data is incorrect**  1. The Admin receives an error that the data they introduces is incorrect  **1.0.E2: The account is not the admin account**  1. The apps logs in as a usual User, no additional functionalities | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-2: Reserve a seat | | |
| **Primary actor** | User | **Secondary actors** | Theatre reservation system |
| **Description** | A User accesses the Theatre reservation System, views the available seats for the day, selects the seat they want, and makes a reservation for the chosen seat. | | |
| **Trigger** | A user logs in and wants to go to the show at the theatre | | |
| **Preconditions** | PRE-1. User is logged into COS.  PRE-2. There are available seats for the show | | |
| **Postconditions** | POST-1. Reservation is made for the chosen seats  POST-2. Available seats remaining are updated to reflect the reservation. | | |
| **Normal flow** | **2.0 Reserve a single seat**  1. User picks the option to see the configuration of the room and the available seats  2. TRS displays the room with all the seats and the available ones  3. User clicks on one seat that they want to reserve  4. TRS displays the price and state of the desired seat  5. If available, the User can choose to reserve the seat  6. The User enters their personal data  7. TRS updates the seats in the room on all devices | | |
| **Alternative flows** | **2.1 Reserve multiple seats**  1. User clicks on the reservation option (see 1.0.1-1.0.2)  2.The User selects multiple seats that they want to reserve  3. Return to step 6 of the normal flow | | |
| **Exceptions** | **2.0.E1 – All the seats are already reserved**  1. TRS informs the User that no seats are available and they cannot attend the show  **2.0.E2 – The seats configuration for the day have not yet been updated**  1. TRS informs the User that the seats are not yet available for reservation | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-3: Delete a reservation | | |
| **Primary actor** | User | **Secondary actors** | Theatre reservation system |
| **Description** | A User accesses the Theatre reservation System, views the reservations they have and chooses t odelete a reservation | | |
| **Trigger** | A user logs in and wants to delete a resrvation | | |
| **Preconditions** | PRE-1. User is logged into TSR.  PRE-2. The User has at least one reservation | | |
| **Postconditions** | POST-1. Reservation is deleted for the chosen seats  POST-2. Available seats remaining are updated to reflect the deletion. | | |
| **Normal flow** | **3.0 Delete a single seat**  1. User picks the option to see the reservations  2. TRS displays the reservations  3. User clicks on the reservation that they want to delete  4. TRS displays a warning  5. If they choose to continue, the reservation is deleted  6. TRS updates the seats in the room on all devices | | |
| **Alternative flows** | **2.1 Delete multiple seats**  1. User clicks on the reservations option (see 1.0.1-1.0.2)  2.The User selects multiple reservations that they want to delete  3. Return to step 4 of the normal flow | | |
| **Exceptions** | * none | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-4: View current reservations | | |
| **Primary actor** | User | **Secondary actors** | Theatre reservation system |
| **Description** | The User can see their reservations for the show | | |
| **Trigger** | The User wants to see their reservations | | |
| **Preconditions** | PRE-1: The user is logged in the application | | |
| **Postconditions** | * none | | |
| **Normal flow** | **4.0 – Display all reservations**  1. The User chooses the option to view the reservations  2. TSR dispalys a list of all the seats reserved by the User | | |
| **Alternative flows** | * none | | |
| **Exceptions** | * none | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-5: View all seats | | |
| **Primary actor** | User | **Secondary actors** | Theatre reservation system |
| **Description** | The User can see the seats for the show | | |
| **Trigger** | The User wants to see the room configuration | | |
| **Preconditions** | PRE-1: The user is logged in the application | | |
| **Postconditions** | * none | | |
| **Normal flow** | **5.0. – Display all seats**  1. The User chooses the option to view the seats  2. TSR dispalys a list of all the seats, with their core | | |
| **Alternative flows** | * none | | |
| **Exceptions** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-6: Add new show | | |
| **Primary actor** | Admin | **Secondary actors** | Theatre reservation system |
| **Description** | The admin creates a new configuration for a new show | | |
| **Trigger** | A new show for the day is planned | | |
| **Preconditions** | PRE-1: The admin is logged in to the application | | |
| **Postconditions** | POST-1: The new show is displayed to anyone that wants to see it | | |
| **Normal flow** | **6.0. Add a new show**  1. The admin introduces the data for the new show  2. The admin specifies the number of rows and number of seats in each row for the show  3. The admin saves the new show  3. TSR displays the room configuration, with all the seats marked as free  (The show is automatically deleted after the show’s end time) | | |
| **Alternative flows** | * none | | |
| **Exceptions** | **6.0.E1 – There already is a show planned for the day**  1. The admin receives an error (there can only be one show a day) | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-7: Update show | | |
| **Primary actor** | Admin | **Secondary actors** | Theatre reservation system |
| **Description** | The admin updates a show | | |
| **Trigger** | The show needs to be updated | | |
| **Preconditions** | PRE-1: The admin is logged in to the application | | |
| **Postconditions** | POST-1: The updated show is displayed to anyone that wants to see it | | |
| **Normal flow** | **7.0. Update a show**  1. The admin introduces the updated data for the show  2. The admin saves the new show  3. TSR displays the updated show details | | |
| **Alternative flows** | * none | | |
| **Exceptions** | **6.0.E1 – There already is a show planned for the day**  1. The admin receives an error (there can only be one show a day) | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-8: View User History | | |
| **Primary actor** | User | **Secondary actors** | Theatre reservation system |
| **Description** | The user sees a list of the show he/she has been to in the past | | |
| **Trigger** | The user wants to see the history | | |
| **Preconditions** | PRE-1: The user is logged in to the application | | |
| **Postconditions** | POST-1: The history is displayed | | |
| **Normal flow** | **8.0. View User History**  1. The user chooses the option of history  2. TSR displays a list of shows the user has previouslly been to | | |
| **Alternative flows** | **8.1. No History**  1. The user chooses the option of history  2. TSR displays „You haven’t been to any shows yet” | | |
| **Exceptions** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-9: Create subscription | | |
| **Primary actor** | User | **Secondary actors** | Theatre reservation system |
| **Description** | The user buys a subscription to the theatre | | |
| **Trigger** | The user wants to buy a subscription | | |
| **Preconditions** | PRE-1: The user is logged in to the application | | |
| **Postconditions** | POST-1: The subscription is bought | | |
| **Normal flow** | **9.0. Create subscription**  1. The user chooses the option of subscription  2. TSR displays a list of available subscriptions  3. The user chooses the desired subscription  4. TSR redirects the user to the payment page | | |
| **Alternative flows** | - | | |
| **Exceptions** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-10: Delete a subscription | | |
| **Primary actor** | User | **Secondary actors** | Theatre reservation system |
| **Description** | A User accesses the Theatre reservation System and chooses to delete a subscription | | |
| **Trigger** | A user logs in and wants to delete a subscription | | |
| **Preconditions** | PRE-1. User is logged into COS. | | |
| **Postconditions** | POST-1. Subscription is deleted for the user | | |
| **Normal flow** | **10.0 Delete a subscription**  1. User picks the option to see the subscriptions  2. TRS displays the subscriptions  3. User clicks on the subscription that they want to delete  4. TRS displays a warning  5. If they choose to continue, the subscription is deleted | | |
| **Alternative flows** |  | | |
| **Exceptions** | * none | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-11: Pay cash | | |
| **Primary actor** | User | **Secondary actors** | Theatre reservation system |
| **Description** | A User accesses the Theatre reservation System and chooses to pay for a reservation or subscription in cash | | |
| **Trigger** | A user logs in and wants to pay for a reservation or subscription in cash | | |
| **Preconditions** | PRE-1. User is logged into COS.  PRE-2. User selected a reservation or subscription to buy | | |
| **Postconditions** | POST-1. Subscription/reservation is paid | | |
| **Normal flow** | **11.0 Pay cash**  1. User is redirected to the payment page and chooses „Pay in cash” option  2. TSR displays a page where the User has to enter their personal data  3. User eneters the data  4. TSR dispalys the amount and time of day when the User has to got o the theatre and pay for the reservation/subscription | | |
| **Alternative flows** |  | | |
| **Exceptions** | **11.0.E1 – The personal data is incorrect**  1. The user receives an error | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-12: Pay card | | |
| **Primary actor** | User | **Secondary actors** | Theatre reservation system |
| **Description** | A User accesses the Theatre reservation System and chooses to pay for a reservation or subscription with the credit card | | |
| **Trigger** | A user logs in and wants to pay for a reservation or subscription with the credit card | | |
| **Preconditions** | PRE-1. User is logged into COS.  PRE-2. User selected a reservation or subscription to buy | | |
| **Postconditions** | POST-1. Subscription/reservation is paid | | |
| **Normal flow** | **12.0 Pay card**  1. User is redirected to the payment page and chooses „Pay with a credit card” option  2. TSR displays a page where the User has to enter their card data  3. User eneters the data  4. TSR takes the money from the card  5. TSR displays a success message | | |
| **Alternative flows** | - | | |
| **Exceptions** | **12.0.E1 – Payment could not be done**  1. The user receives an error  2. The User is directed to the payment options again | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-13: Pay with subscription | | |
| **Primary actor** | User | **Secondary actors** | Theatre reservation system |
| **Description** | A User accesses the Theatre reservation System and chooses to pay for a reservation with a subscription | | |
| **Trigger** | A user logs in and wants to pay for a reservation with a subscription | | |
| **Preconditions** | PRE-1. User is logged into COS.  PRE-2. User selected a reservation to buy | | |
| **Postconditions** | POST-1. Reservation is paid | | |
| **Normal flow** | **13.0 Pay with a subscription**  1. User is redirected to the payment page and chooses „Pay with a subscription” option  2. TSR displays a page where the User has to choose a valid subscription  3. User chooses the subscription  4. TSR dispalys a success message | | |
| **Alternative flows** | - | | |
| **Exceptions** | - | | |

Planned iterations:

* First iteration:
* Create structure of application
* Create domain classes for users and shows
* Create repository for users and shows
* Create service for users and shows
* Implement login page and logout button
* Implement page for add and update shows
* Second iteration:
* Create domain classes for reservations
* Create repository for reservations
* Implement the service for the reservations
* Implement page for available seats
* Implement page for view reservations
* Implement page for delete reservations
* Implement page for reserving a seat
* Third iteration:
* Create domain classes for history and subscription
* Create repository for history and subscription
* Implement service for history and subscription
* Implement subscription pages
* Implement payment pages
* Create the entire flow of the application