## Examples

For a hypothetical *Cafeteria Ordering System*1:

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-1: Reserve Seats | | |
| **Primary actor** | Spectator | **Secondary actors** | Admin |
| **Description** | Allows a spectator to reserve seats for a performance. | | |
| **Trigger** | Spectator indicates the desire to reserve seats. | | |
| **Preconditions** | Spectator is logged into the system. | | |
| **Postconditions** | Seats are reserved in the system with a status of "Reserved." | | |
| **Normal flow** | **1.0 Reserving a seat**   1. Spectator accesses the system to reserve seats. 2. System displays the seating configuration. 3. Spectator selects one or more seats. 4. Spectator confirms the reservation. | | |
| **Alternative flows** | **1.1 Reserving a taken seat**  If the selected seats are not available, inform the spectator and prompt them to choose other seats.  If the spectator is not logged in, prompt them to log in or create an account. | | |
| **Exceptions** | - | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-2: View Seat Availability | | |
| **Primary actor** | Spectator | **Secondary actors** | None |
| **Description** | Enables spectators to view the availability of seats for a performance. | | |
| **Trigger** | Spectator indicates the desire to view seat availability. | | |
| **Preconditions** | None. | | |
| **Postconditions** | Spectator views the current seat availability. | | |
| **Normal flow** | **1.0 Reserving a seat**   1. Spectator selects the option to view seat availability. 2. System displays a graphical representation of the seating configuration with available and reserved seats. | | |
| **Alternative flows** | **1.1 Reserving a taken seat**  If there is an issue retrieving seat information, display an error message to the spectator. | | |
| **Exceptions** | - | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-3 : Manage User Account | | |
| **Primary actor** | Spectator | **Secondary actors** | Admin |
| **Description** | Allows spectators to manage their user accounts. | | |
| **Trigger** | Spectator indicates the desire to manage their account. | | |
| **Preconditions** | PRE-1. Spectator is logged into the system. | | |
| **Postconditions** | POST-1. User account information is updated as per the spectator's actions. | | |
| **Normal flow** | **2.0 Account data editing**   1. Spectator selects the option to manage their account. 2. Spectator can update their profile information, change password, or delete the account. | | |
| **Alternative flows** | If the spectator encounters any issues while updating their profile or changing the password, display an error message. | | |
| **Exceptions** | - | | |

### **Use Case Planning over 3 Iterations:**

1. Auth implementation.
   1. Library DB implementation
   2. User CRUD
   3. Log in/Register
2. Seats reserving system.
   1. Seats CRUD
   2. Reserve Seats
   3. Book seat
   4. Plata rezervare
3. Front End Implementation.
   1. Modify the price of seats