## **Use case tables**

For an app that manages the seats for an event

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-1: Log In | | |
| **Primary actor** | Spectator/Admin | **Secondary actors** |  |
| **Description** | A user, either an admin or a spectator, tries to log into the application with a user name and a password. | | |
| **Trigger** | A spectator wants to access information about the available seats for a concert or the admin wants to access information about all planned concerts. | | |
| **Preconditions** | None | | |
| **Postconditions** | POST-1. On login success, the user is presented with the respective information. | | |
| **Normal flow** | 1. The user introduces the user name and password. 2. They press the login button. 3. They are logged into the application and can continue their activity there. | | |
| **Alternative flows** | 1. The user has no account and they decide to register -> UC-2 | | |
| **Exceptions** | 1. The user introduces a wrong user name or password. In this case the app will show a pop up notifying the user about this aspect. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-2: Register | | |
| **Primary actor** | Spectator | **Secondary actors** |  |
| **Description** | A user registers a new account and is logged in afterwards. He is asked for an email, a user name and a password. | | |
| **Trigger** | The user wants to make a new account. | | |
| **Preconditions** | None | | |
| **Postconditions** | POST-1. On register success, the user is presented with the respective information | | |
| **Normal flow** | 1. The user introduces the email, username and the password twice for confirmation. 2. They press the login button. 3. They are logged into the application and can continue their activity there. | | |
| **Alternative flows** | - | | |
| **Exceptions** | 1. The user introduces an email that already has an account associated with it, in which case the app notifies the user of this misuse. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-3: View seats | | |
| **Primary actor** | Spectator | **Secondary actors** |  |
| **Description** | The user can view the layout of the seats in the concert room. | | |
| **Trigger** | The user wishes to see the available seats for the event today. | | |
| **Preconditions** | PRE-1. The user must be logged in as a spectator. | | |
| **Postconditions** | POST-1. Seats are shown. | | |
| **Normal flow** | 1. The user is shown the layout of the room. | | |
| **Alternative flows** | 1. The user decides he wants to reserve some seats – he presses the Reserve button -> UC-5 2. The user decides to view his reservations – he presses the View Reservations button -> UC-6 | | |
| **Exceptions** | - | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-4: View clients | | |
| **Primary actor** | Admin | **Secondary actors** |  |
| **Description** | An admin accesses the User management tab of the app. | | |
| **Trigger** | The admin wants to verify whether a person has an account in the app or not. | | |
| **Preconditions** | PRE-1. Has to be logged in as an admin. | | |
| **Postconditions** | POST-1. Clients are shown. | | |
| **Normal flow** | 1. The admin views a list of clients. He cannot act upon this list. | | |
| **Alternative flows** | - | | |
| **Exceptions** | - | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-5: Reserve seats | | |
| **Primary actor** | Spectator | **Secondary actors** |  |
| **Description** | A spectator reserves a number of seats for today’s event. | | |
| **Trigger** | The spectator wants to reserve some seats for today’s event | | |
| **Preconditions** | PRE-1. The user is logged in as a spectator.  PRE-2. There is an active show. | | |
| **Postconditions** | POST-1. The seats the user reserves are marked as reserved in both the database and on the seat view interface. | | |
| **Normal flow** | 1. User presses the Reserve button. 2. A new window pops up. 3. The user inputs the seats he wants to reserve via the numbers assigned to each of them (ex : 1, 2, 3 -> Seats 1,2,3 ). 4. He presses the confirm button once he’s done. | | |
| **Alternative flows** | - | | |
| **Exceptions** | 1. One or more of the seats the user wrote has already been reserved. 2. The program launches a warning pop up and doesn’t register the user’s input. 3. He may try to reserve again in the same window. 4. The time of the event has passed. 5. The program launches a warning pop-up. 6. The user may not try to reserve again in the same window. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-6: View reservations | | |
| **Primary actor** | Spectator | **Secondary actors** |  |
| **Description** | A spectator views his reservations. | | |
| **Trigger** | A user wants to see the reservations he has made and maybe cancel one of them. | | |
| **Preconditions** | PRE-1. User must be logged in as spectator. | | |
| **Postconditions** | POST-1. Reservations for user are shown. | | |
| **Normal flow** | 1. The user presses the View Reservations button. 2. A new window with a list of his reservations is open. | | |
| **Alternative flows** | 1. Upon opening the list of reservations the user can select one and go to UC-7 | | |
| **Exceptions** | - | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-7: Cancel reservation | | |
| **Primary actor** | Spectator | **Secondary actors** |  |
| **Description** | The part of the application that handles the removing of a reservation and the consequent freeing of the seats. | | |
| **Trigger** | A user wants to cancel one of their reservations. | | |
| **Preconditions** | PRE-1. User must be logged in as spectator.  PRE-2. User must have at least one reservation. | | |
| **Postconditions** | POST-1. The database and the seat view are updated as a consequence of the reserved seats being freed. | | |
| **Normal flow** | 1. The user presses the View Reservations button. 2. User selects a reservation. 3. They press the Remove button. 4. The window closes with a pop-up confirming the cancelation. | | |
| **Alternative flows** | - | | |
| **Exceptions** | 1. User has no reservations and presses delete button. 2. Upon pressing the delete button the reservations window is closed and a pop-up warns the user about the situation. 3. The user tries to cancel a reservation for an event that has already passed. 4. The user is notified of his mistake via a pop-up and no changes take place. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-8: View events | | |
| **Primary actor** | Admin | **Secondary actors** |  |
| **Description** | All the events and their corresponding dates are shown in a table. | | |
| **Trigger** | An admin wants to view the events. | | |
| **Preconditions** | PRE-1. User must be logged in as an admin. | | |
| **Postconditions** | POST-1. Evets are shown. | | |
| **Normal flow** | 1. Upon logging in and choosing the View Events button, the user is presented with a list of all events. 2. The user may browse up and down the list of events. | | |
| **Alternative flows** | 1. The user selects an event -> UC-10 2. The user presses the Add button -> UC-9 | | |
| **Exceptions** | - | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-9: Add an event | | |
| **Primary actor** | Admin | **Secondary actors** |  |
| **Description** | This feature adds an event with empty reservations to the database | | |
| **Trigger** | A new upcoming event is announced an must be logged into the app. | | |
| **Preconditions** |  | | |
| **Postconditions** | POST-1. An event with all seats free is added. | | |
| **Normal flow** | 1. User presses the Add event button. 2. They are presented with a window with a name and date input. 3. The user presses the Confirm button. 4. The information is read and a new event is added. | | |
| **Alternative flows** | - | | |
| **Exceptions** | 1. The date input has passed. 2. The user is warned and is taken back to change their input. 3. No changes take place. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-10: Reset seat reservations for an event. | | |
| **Primary actor** | Admin | **Secondary actors** |  |
| **Description** | The user selects an event and frees all seats for it. | | |
| **Trigger** | This is a last resort, in case of a bug or of an organizational error. | | |
| **Preconditions** | PRE-1. User must be logged in as an admin.  PRE-2. The list of events must not be empty.  PRE-3. The selected event must not have passed. | | |
| **Postconditions** | POST-1. All reservations for the selected event are canceled.  POST-2. All seats are freed.  POST-3. Any other user logged in may see this change. | | |
| **Normal flow** | 1. User selects an event. 2. They press the Reset button. 3. They are warned of the consequences of such and action an that it is not safe. 4. They confirm that they understand the warning. 5. The changes take place. | | |
| **Alternative flows** | - | | |
| **Exceptions** | 1. The selected event has no reservations. 2. The program warns the user and applies no changes. 3. The selected event has passed. 4. The user is warned and no changes take place. 5. There is no selected event upon pressing the Reset button. 6. No changes take place. | | |