FrogCrew

Use Cases

Version <1.0>

*[Note: Text enclosed in square brackets and displayed in blue italics (style=InfoBlue) is included to provide guidance to the author and should be deleted before publishing the document. A paragraph entered following this style will automatically be set to normal (style=Body Text).]*

Use Cases

***Use Case ID and Name***

*Give each use case a unique integer sequence number identifier. State a concise name for the use case that indicates the value the use case would provide to some user. Begin with an action verb, followed by an object.*

***Author and Date Created***

*Enter the name of the person who initially wrote this use case and the date it was written.*

***Primary and Secondary Actors***

*An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks. Different actors often correspond to different user classes, or roles, identified from the customer community that will use the product. Name the primary actor that will be initiating this use case and any other secondary actors who will participate in completing execution of the use case.*

***Trigger***

*Identify the business event, system event, or user action that initiates the use case. This trigger alerts the system that it should begin testing the preconditions for the use case so it can judge whether to proceed with execution.*

***Description***

*Provide a brief description of the reason for and outcome of this use case, or a high-level description of the sequence of actions and the outcome of executing the use case.*

***Preconditions***

*List any activities that must take place, or any conditions that must be true, before the use case can be started. The system must be able to test each precondition. Number each precondition. Example: PRE-1: User’s identity has been authenticated.*

***Postconditions***

*Describe the state of the system at the successful conclusion of the use case execution. Label each postcondition in the form POST-X, where X is a sequence number. Example: POST-1: Price of item in the database has been updated with the new value.*

***Main Success Scenario/Normal Flow***

*Provide a description of the user actions and corresponding system responses that will take place during execution of the use case under normal, expected conditions. This dialog sequence will ultimately lead to accomplishing the goal stated in the use case name and description. Show a numbered list of actions performed by the actor, alternating with responses provided by the system. The normal flow is numbered “X.0”, where “X” is the Use Case ID.*

***Extensions:***

* ***Alternative Flows***

*Document other successful usage scenarios that can take place within this use case. State the alternative flow, and describe any differences in the sequence of steps that take place. Number each alternative flow in the form “X.Y”, where “X” is the Use Case ID and Y is a sequence number for the alternative flow. For example, “5.3” would indicate the third alternative flow for use case number 5. Indicate where each alternative flow would branch off from the normal flow, and if pertinent, where it would rejoin the normal flow.*

* ***Exceptions***

*Describe any anticipated error conditions that could occur during execution of the use case and how the system is to respond to those conditions. Number each alternative flow in the form “X.Y.EZ”, where “X” is the Use Case ID, Y indicates the normal (0) or alternative (>0) flow during which this exception could take place, “E” indicates an exception, and “Z” is a sequence number for the exceptions. For example “5.0.E2” would indicate the second exception for the normal flow for use case number 5. Indicate where in the normal (or an alternative) flow each exception could occur.*

***Priority***

*Indicate the relative priority of implementing the functionality required to allow this use case to be executed. Use the same priority scheme as that used for the functional requirements.*

***Frequency of Use***

*Estimate the number of times this use case will be performed per some appropriate unit of time. This gives an early indicator of throughput, concurrent usage loads, and transaction capacity.*

***Business Rules***

*List any business rules that influence this use case. Don’t include the business rule text here, just its identifier so the reader can find it in another repository when needed.*

***Associated Information***

*Identify any additional requirements, such as quality attributes, for the use case that may need to be addressed during design or implementation. Also list any associated functional requirements that aren’t a direct part of the use case flows but which a developer needs to know about. Describe what should happen if the use case execution fails for some unanticipated or systemic reason (e.g., loss of network connectivity, timeout). If the use case results in a durable state change in a database or the outside world, state whether the change is rolled back, completed correctly, partially completed with a known state, or left in an undetermined state as a result of the exception.*

***Assumptions***

*List any assumptions that were made regarding this use case or how it might execute.*

# Use Case 1: Crew Member Creates Crew Member Profile

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC-1: Crew Member Creates Crew Member Profile | | |
| Created By: | Aliya Suri | Date Created: | 10/01/24 |
| Primary Actor: | Crew Member | Secondary Actors: |  |
| Trigger: | The Crew Member clicks the registration link in the invitation email. | | |
| Description: | The Crew Member wants to set up an account, so that they can be crewed for games. | | |
| Preconditions: | PRE-1. An invitation email is sent to the Crew Member. | | |
| Postconditions: | POST-1. The Crew Member account is set up. | | |
| Main Success Scenario: | 1. The Crew Member clicks the registration link in the invitation email. 2. The System opens a new page and asks the Crew Member to enter the details of this new account according to the “Details” defined in the Associated Information of this use case. 3. The Crew Member enters the details of this new account and confirms that they have finished. 4. The System validates the Crew Member’s inputs according to the “Details” defined in the Associated Information of this use case. 5. The System displays the details of the new account and asks the Crew Member to confirm the registration. 6. The Crew Member either confirms the registration (continues the normal flow) or chooses to modify the details (return to step 3). 7. The System saves the information about the new account and informs the Crew Member that this account has been created. 8. The System redirects the Crew Member to the login page. 9. Use case ends. | | |
| Extensions: | **2a. The Crew Member has already set up the account:**  2a1. The System alerts the Crew Member that they have already set up their account and shall log in.  2a2. The System redirects the Crew Member to the login page.  **4a. Input validation rule violation:**  4a1. The System alerts the Crew Member that an input validation rule is violated and displays the nature and location of the error.  4a2. The Crew Member corrects the mistake and returns to step 4 of the normal flow. | | |
| Priority: | High | | |
| Frequency of Use: | Approximately 35-40 users, 1 usage per year. | | |
| Business Rules: | BR-1: A Crew Member must have a unique email address in the system.  BR-2: Crew Members can only edit their own profile.  BR-3: Phone numbers must follow the format 999-999-9999.  BR-4: All required fields (First Name, Last Name, Email, Phone Number, Role, Qualified Position) must be filled before saving changes. | | |
| Associated Information: | |  |  |  |  | | --- | --- | --- | --- | | **Property name** | **Data type** | **Editability** | **Validation rule** | | First Name | String | Yes | Required | | Last Name | String | Yes | Required | | Email | String | Yes | Required, Valid Email | | Phone Number | String | Yes | Required, 999-999-999 Fornat | | Password | String | Yes | Required | | Role | String | Yes | Required | | Qualified Position | String | Yes | Required | | | |
| Related Use Cases | UC 2: Crew Member Edits Profile  UC 14: Admin Invites Crew Member  UC 16: Admin Views Crew Members | | |
| Assumptions: |  | | |
| Open Issues: |  | | |

# Use Case 3: Crew Member Views A Crew Member’s Profile

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC-3: Crew Member Views A Crew Member’s Profile | | |
| Created By: | Aliya Suri | Date Created: | 10/01/24 |
| Primary Actor: | Crew Member | Secondary Actors: |  |
| Trigger: | The User indicates to view the details of a Crew Member’s profile. | | |
| Description: | The User wants to view the details of a Crew Member’s profile, so that they can contact that Crew Member. | | |
| Preconditions: | PRE-1. The User is logged into the System. | | |
| Postconditions: | POST-1. The details of the specified Crew Member profile are displayed to the User. | | |
| Main Success Scenario: | 1. The User indicates to view the details of a Crew Member profile. 2. The User finds a list of Crew Members through UC-12: Views crew members. 3. The User views the list and chooses to view the details of one specific Crew Member. 4. The System retrieves and displays details of this Crew Member profile according to the “Details” defined in the Associated Information and the “Security/access concerns” defined in the Business Rules of this use case. 5. The User views the details of this Crew Member profile. 6. Use case ends. | | |
| Extensions: |  | | |
| Priority: | Medium | | |
| Frequency of Use: | Approximately \*\*\* user, average of \*\*\* usages per week. | | |
| Business Rules: | BR-13: Crew Members can only access directory information of other Crew Members.  BR-37: Only Admins with the correct permissions can access sensitive data. | | |
| Associated Information: | Details:   * Same as the “Details” defined in the Associated Information of UC-1: Set up a crew member account | | |
| Related Use Cases: | UC 6: Crew Member Views Crew List  UC 16: Admin Views Crew Members | | |
| Assumptions: |  | | |
| Open Issues: |  | | |

# Use Case 5: Crew Member Views General Game Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC-5: Crew Member Views General Game Schedule | | |
| Created By: | Michala Rogers | Date Created: | 2/25/25 |
| Primary Actor: | Crew Member | Secondary Actors: |  |
| Trigger: | The User indicates to view the game schedule. | | |
| Description: | The Crew Member wants to view the game schedule to check upcoming games, event details, and available positions. This ensures they are aware of upcoming events and can plan their availability accordingly. | | |
| Preconditions: | PRE-1: A game schedule has been created and published.  PRE-2: The Crew Member is logged into the system. | | |
| Postconditions: | POST-1: The details of the general game schedule are displayed to the Crew Member.  POST-2: The Crew Member can view key game details such as date, time, location, opponent, and required crew positions. | | |
| Main Success Scenario: | 1. The Crew Member navigates to the "View Game Schedule" section. 2. The System retrieves and displays the general game schedule 3. The Crew Member reviews the schedule. 4. The Crew Member can filter, sort, or search for specific games. 5. The Crew Member exits the schedule view. 6. Use case ends. | | |
| Extensions: | 3a. No games are currently scheduled:   * 3a1. The system displays a message: "No upcoming games available at this time." * 3a2. The Crew Member can return to the main dashboard.   5a. The Crew Member wants to view more details about a specific game:   * 5a1. The Crew Member selects a game from the list. * 5a2. The system retrieves and displays additional information about that game (e.g., assigned crew, reporting time, and contact details). | | |
| Priority: | Medium | | |
| Frequency of Use: | Approximately 50 users, averaging 5-10 views per week. | | |
| Business Rules: | BR-5: Crew Members can only view their own schedule.  BR-9: Only Admins can modify the schedule. | | |
| Associated Information: | Game Schedule Information:   * Game Date * Game Time * Opponent * Venue * Required Crew Positions   Sorting Options:   * Sort by Date (Newest to Oldest) * Sort by Opponent * Sort by Venue   Notifications:   * If a schedule update occurs, the Crew Member receives a notification. | | |
| Related Use Cases: | waiting until use case numbers are solidified | | |
| Assumptions: | UC 4: Crew Member Views Their Scheduled Games  UC 22: Admin Schedules Crew  UC 23: Admin Publishes Game Schedule | | |
| Open Issues: | The game schedule is finalized before being displayed to Crew Members.  The Crew Member has the correct permissions to access the schedule. | | |

# Use Case 6: Crew Member Views Crew List

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC-6: Crew Member Views Crew List | | |
| Created By: | Kate Bednarz | Date Created: | 01/25/25 |
| Primary Actor: | Crew member | Secondary Actors: |  |
| Trigger: | The user indicates to view a crew list. | | |
| Description: | The user wants to view a crew list. | | |
| Preconditions: | PRE-1: A crew list has been created. | | |
| Postconditions: | POST-1: The details of the specified crew list are displayed to the user. | | |
| Main Success Scenario: | 1. The user indicates to view a crew list for a specific game. 2. The system retrieves and displays details of the crew list according to the “Details” defined in the Associated information. 3. The user views the details of the crew list. 4. Use case ends. | | |
| Extensions: |  | | |
| Priority: | Medium | | |
| Frequency of Use: | Approximately 30 users, average of 2 usages per week. | | |
| Business Rules: | BR-13: Crew Members can only access directory information of other Crew Members. | | |
| Associated Information: | Details:   |  |  |  |  | | --- | --- | --- | --- | | **Property name** | **Data type** | **Editability** | **Validation rule** | | Sport | String | Yes | Required | | Opponent | String | Yes | Required | | Game date | Datetime | Yes | Required | | Game time | Datetime | Yes | Required | | Position | String   * Producer * Asst Prod * Director * Asst Director * Technical Dir * Graphics * Bug Op * Replay EVS * EIC * Video * Audio * Camera * Utility * Tech manager * TOC * Observer | Yes | Required | | Name | String | Yes | Required | | Report time | Datetime | Yes | Required | | Report location | String | Yes | Required | | | |
| Related Use Cases: | UC 3: Crew Member Views Other Crew Member’s Profile  UC 16: Admin Views Crew Members  UC 23: Admin Publishes Game Schedule | | |
| Assumptions: |  | | |
| Open Issues: |  | | |

# 

# Use Case 7: Crew Member Submits Availability

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC-7: Crew Member Submits Availability | | |
| Created By: | Kate Bednarz | Date Created: | 09/29/2024 |
| Primary Actor: | Crew Member | Secondary Actors: | Admin |
| Trigger: | The Crew Member indicates to submit availability for a game. | | |
| Description: | The Crew Member wants to submit their availability for a game, so that they can be scheduled to work a game. | | |
| Preconditions: | PRE-1. The Crew Member is logged in to the System.  PRE-2. The Admin has published a game schedule. | | |
| Postconditions: | POST-1. The Crew Member’s availability is stored in the System. | | |
| Main Success Scenario: | 1. The Crew Member indicates to submit their availability for a game. 2. The System displays games that they haven’t submitted availability for. 3. The Crew Member selects the games they are available for.. 4. The System validates the Crew Member’s inputs according to the “Details” defined in the Associated Information of this use case. 5. The System saves the new availability, and informs the Crew Member that their availability has been saved. 6. The System notifies relevant actors about the creation of the availability according to the “Notification” defined in the Associated Information of this use case. 7. Use case ends. | | |
| Extensions: | **4a. Input validation rule violation:**  6a1. The System alerts the Crew Member that an input validation rule is violated and displays the nature and location of the error.  6a2. The Crew Member corrects the mistake and returns to step 6 of the normal flow. | | |
| Priority: | High | | |
| Frequency of Use: | All crew members are users, average of 1 usage per user per new schedule release. | | |
| Business Rules: | BR-6: Crew Members must submit availability before being scheduled.  BR-8: Availability updates notify the Admin. | | |
| Associated Information: | Details:   |  |  |  |  | | --- | --- | --- | --- | | **Property name** | **Data type** | **Editability** | **Validation rule** | | Available | Boolean | Yes | Required | | Comment | String | Yes |  |   Notification:   * The System sends a notification to the Admin | | |
| Related Use Cases | UC 8: Crew Member Edits Availability  UC 17: Admin Views Crew Member’s Availability  UC 22: Admin Schedules Crew | | |
| Assumptions: |  | | |
| Open Issues: |  | | |

# se Case 14: Admin Invites Crew Member

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC-14: Admin Invites Crew Member | | |
| Created By: | Manuel Burciaga | Date Created: |  |
| Primary Actor: | Admin | Secondary Actors: | Crew Member (receiving the invite) |
| Trigger: | The Admin indicates to invite new crew members. | | |
| Description: | The admin invites a new crew member by sending a link. The invited crew member receives an email with a link to create their profile. | | |
| Preconditions: | PRE-1. The Admin is logged into the System. | | |
| Postconditions: | **POST-1:** The crew member receives an email invitation with a registration link. | | |
| Main Success Scenario: | 1. The admin navigates to the "Invite Crew Member" section. 2. The admin provides email(s). 3. The system sends an email invitation to the crew member with a registration link. 4. Use case ends. | | |
| Extensions: | **4a. Invalid email address:**   * **4a1:** The system detects the email is invalid and prompts the Admin to correct it. | | |
| Priority: | High | | |
| Frequency of Use: | As needed when new crew members are added. | | |
| Business Rules: | BR-1: A Crew Member must have a unique email address in the system. | | |
| Associated Information: | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Details | Property Name | Data Type | Editability | Validation Rule | | Crew Member Name | String | Yes | Required | Must not Exceed 50 characters | | Crew Member Email | String | Yes | Required | Must be a valid, and unique email | | Crew Member Role | String | Yes | Required | Crew member must select one or more roles |   Notification:   * The system sends an email invitation to the new crew member. * The system sends a confirmation to the crew member once the profile is approved. * The system notifies the admin if the crew member fails to complete the profile within a certain time frame (e.g., 7 days). | | |
| Related Use Cases: | UC 1: Crew Member Creates Profile  UC 16: Admin Views Crew Members | | |
| Assumptions: |  | | |
| Open Issues: |  | | |

# Use Case 15: Admin Deletes A Crew Member

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC-15: Admin Deletes A Crew Member | | |
| Created By: | Michala Rogers | Date Created: | 2/5/2025 |
| Primary Actor: | Admin | Secondary Actors: | Crew Member |
| Trigger: | The Admin initiates the removal of a Crew Member from the system. | | |
| Description: | The Admin removes a Crew Member who is no longer part of the team. This ensures that only active members remain in the system and prevents scheduling or communication issues with inactive users. | | |
| Preconditions: | PRE-1: At least one Crew Member is registered in the system.  PRE-2: The Admin is logged into the system.  PRE-3: The Crew Member is not assigned to any future scheduled games. | | |
| Postconditions: | POST-1: The Crew Member’s account and associated data are removed from the system.  POST-2: System notifications are sent to relevant users informing them of the deletion.  POST-3: If required, historical data related to the Crew Member (e.g., past game assignments, reports) remains accessible for record-keeping. | | |
| Main Success Scenario: | 1. The Admin navigates to the "Manage Crew Members" section. 2. The Admin selects the Crew Member to be deleted. 3. The System prompts the Admin with a confirmation message, including any warnings about active assignments. 4. The Admin confirms the deletion. 5. The System checks if the Crew Member has any pending or assigned shifts.    1. If no pending assignments exist, proceed to Step 6.    2. If assignments exist, prompt the Admin to reassign or remove the Crew Member from schedules. 6. The System deletes the Crew Member’s profile from the active crew list. 7. The System notifies the Admin that the deletion was successful. 8. The System sends a notification to the Crew Member informing them that their account has been removed (if applicable). 9. Use case ends. | | |
| Extensions: | 5a. The Crew Member has pending or scheduled shifts:   * 5a1. The System alerts the Admin that the Crew Member is assigned to upcoming shifts. * 5a2. The Admin is prompted to reassign or remove the Crew Member from these shifts before proceeding. * 5a3. Once reassigned or removed, the Admin can confirm the deletion.   6a. The Crew Member has historical work data that must be retained:   * 6a1. Instead of full deletion, the System archives the Crew Member’s account and marks it as inactive. * 6a2. The System ensures past assignments remain accessible for reporting purposes.   7a. Admin cancels the deletion:   * 7a1. The System retains the Crew Member’s account with no changes. | | |
| Priority: | Low | | |
| Frequency of Use: | Approximately 1 user, average of 1 usage per week. | | |
| Business Rules: | BR-16: Admins can only delete Crew Members who do not have upcoming scheduled games.  BR-17: Admins cannot delete Crew Members if they are assigned to financial reports. | | |
| Associated Information: | Notification:   * The System notifies the Admin of the successful deletion. * The System notifies the Crew Member of the deletion.   The Admin shall be able to cancel the process at any time prior to confirming it. | | |
| Related Use Cases: | UC 16: Admin Views Crew Members  UC 17: Admin Views Crew Member’s Availability  UC 22: Admin Schedules Crew | | |
| Assumptions: | The Admin has the necessary permissions to delete a Crew Member.  The System provides a warning before deletion if the Crew Member has upcoming shifts.  Crew Member deletions do not impact past payroll or financial reports. | | |
| Open Issues: | Should the Crew Member receive an email notification when their account is deleted?  Should Admins be allowed to restore deleted Crew Members within a certain timeframe?  Should there be an automated check preventing deletion if the Crew Member has outstanding payroll or other obligations? | | |

# Use Case 16: Admin views crew members

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC-16: Admin views crew members | | |
| Created By: | Michala Rogers | Date Created: | 2/5/2025 |
| Primary Actor: | Admin | Secondary Actors: | Crew Member |
| Trigger: | The Admin initiates the action to view the list of all active Crew Members. | | |
| Description: | The Admin needs to access a list of all registered Crew Members in the system. This allows for quick reference, filtering, and management of crew assignments, ensuring efficient scheduling and coordination. | | |
| Preconditions: | PRE-1. There exists at least one Crew Member registered in the System.  PRE-2. The Admin is logged into the System. | | |
| Postconditions: | POST-1: The system retrieves and displays the list of current Crew Members.  POST-2: The Admin can apply filters or sorting options to organize the displayed list. | | |
| Main Success Scenario: | 1. The Admin navigates to the "Manage Crew Members" section. 2. The Admin selects the option to view all active Crew Members. 3. The system retrieves the names and details of all Crew Members from the database. 4. The system displays the Crew Member list based on predefined sorting mechanisms. 5. The Admin can apply filters (e.g., by role, availability, experience, or name) to refine the results. 6. The Admin can click on a Crew Member's name to view detailed profile information. 7. The Admin completes the review process. 8. Use case ends. | | |
| Extensions: | 6a. No Crew Members found in the system:   * 6a1. The system displays a message: "No Crew Members registered in the system." * 6a2. The Admin is prompted to invite new Crew Members.   7a. The Admin wants to export the Crew Member list:   * 7a1. The system provides an option to download the list in CSV or Excel format. * 7b. The Admin wants to perform bulk actions (e.g., send messages, update roles, deactivate accounts):   + 7b1. The system provides checkboxes next to each Crew Member’s name for bulk selection.   + 7b2. The Admin selects the desired action and confirms. | | |
| Priority: | Medium | | |
| Frequency of Use: | Approximately 1 user, average of 1-2 usages per day during active scheduling periods. | | |
| Business Rules: | BR-18: Admins can view, edit, or delete Crew Members' profiles. | | |
| Associated Information: | Sorting & Filtering Mechanisms:   * Sorting Criteria:   + Crew Member Role   + First Name   + Last Name   + Availability Status * Filtering Options:   + Active vs. Inactive Crew Members   + Availability for upcoming games   + Experience Level (e.g., New, Experienced, Senior)   + Assigned vs. Unassigned Crew | | |
| Related Use Cases: | UC 1: Crew Member Creates Profile  UC 2: Crew Member Edits Profile  UC 3: Crew Member Views Other Crew Member’s Profile  UC 6: Crew Member Views Crew List  UC 14: Admin Invites Crew Member  UC 15: Admin Deletes Crew Member | | |
| Assumptions: | The Admin has the necessary permissions to view the full Crew Member list.  Crew Members have already been registered in the system.  The system supports real-time updates to the Crew Member list. | | |
| Open Issues: | Should the Admin have the ability to archive Crew Members instead of deleting them?  Should the system allow multiple Admins to view and manage the list simultaneously?  Should Admins be able to assign or remove Crew Members from schedules directly from this view? | | |

# Use Case 18: Admin Creates Game Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC-18: Admin Creates Game Schedule | | |
| Created By: | Michala Rogers | Date Created: | 2/3/2025 |
| Primary Actor: | Admin | Secondary Actors: |  |
| Trigger: | The Admin initiates the process to create a new game schedule. | | |
| Description: | The Admin wants to create a game schedule, assigning specific games, dates, venues, and other details necessary for organizing crew assignments. This schedule will be used for crew assignment, shift planning, and event coordination. | | |
| Preconditions: | PRE-1: The Admin is logged into the system. | | |
| Postconditions: | POST-1: A new game schedule is created in the system.  POST-2: The schedule is saved as a draft and can be edited before publishing.  POST-3: The schedule is accessible for future assignments and updates. | | |
| Main Success Scenario: | 1. The Admin navigates to the "Create Game Schedule" section. 2. The system prompts the Admin to enter key schedule details 3. The Admin inputs the schedule details and submits them for validation. 4. The system verifies the input data for completeness and validity. 5. The system saves the schedule 6. Use case ends. | | |
| Extensions: | 4a. Missing or invalid game details:   * 4a1. The system alerts the Admin of missing or incorrect fields. * 4a2. The Admin corrects the issue and resubmits the schedule. | | |
| Priority: | High | | |
| Frequency of Use: | Approximately 1 user, average of 5 usages per week (during scheduling periods). | | |
| Business Rules: | BR-19: Only Admins with scheduling permissions can create a game schedule.  BR-20: A game schedule must have at least one valid game entry before saving.  BR-22: A schedule cannot be published until it is reviewed and finalized. | | |
| Associated Information: | Schedule details must follow predefined formats for dates, times, and venues.  Crew position requirements should align with event needs.  Schedule Details   * Sport type * Game date and time * Venue * Opponent (if applicable) * Required crew positions | | |
| Related Use Cases: | UC 19: Admin Edits Crew Member’s Profile  UC 20: Admin Adds Games to Game Schedule  UC 21: Admin Edits Game Schedule  UC 22: Admin Schedules Crew | | |
| Assumptions: | The system contains accurate game details and venue information.  Admins have the necessary permissions to create schedules. | | |
| Open Issues: | Should the system auto-suggest crew assignments based on previous schedules?  Should the schedule support recurring events for ongoing sports seasons? | | |

# 

# Use Case 20: Admin Adds Games To Game Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC-20: Admin Adds Games To Game Schedule | | |
| Created By: | Michala Rogers | Date Created: | 2/3/2025 |
| Primary Actor: | Admin | Secondary Actors: |  |
| Trigger: | The Admin initiates the action to add one or more games to an existing game schedule. | | |
| Description: | The Admin wants to add new games to an existing game schedule, ensuring the schedule remains updated with the latest game events. This allows for ongoing adjustments, including scheduling new games, rescheduling postponed games, or filling gaps in the current season’s schedule. | | |
| Preconditions: | PRE-1: The Admin is logged into the system.  PRE-2: A game schedule already exists in the system. | | |
| Postconditions: | POST-1: The new game(s) are successfully added to the existing game schedule.  POST-2: The schedule reflects the updated list of games.  POST-3: The system saves changes and updates relevant crew assignment workflows. | | |
| Main Success Scenario: | 1. The Admin navigates to the "Manage Game Schedule" section. 2. The Admin selects an existing game schedule to update. 3. The system displays the current list of scheduled games. 4. The Admin clicks “Add New Game” and enters the required details 5. The system validates the new game details. 6. The Admin confirms the addition of the game(s). 7. The system saves the changes and updates the schedule. 8. The system notifies relevant actors (e.g., crew members, admins) of the updated schedule. 9. Use case ends. | | |
| Extensions: | 6a. Missing or invalid game details:   * 6a1. The system alerts the Admin of missing or incorrect fields. * 6a2. The Admin corrects the issue and resubmits.   8a. The Admin wants to save changes as a draft before finalizing:   * 8a1. The system allows the Admin to save the updated schedule as a draft. | | |
| Priority: | High | | |
| Frequency of Use: | Approximately 1 user, average of 5-10 usages per week (during scheduling periods). | | |
| Business Rules: | BR-21: New games cannot overlap with existing scheduled games unless explicitly allowed.  BR-22: A schedule cannot be published until it is reviewed and finalized. | | |
| Associated Information: | Added games should follow system-defined formats for dates, times, and locations.  Crew position requirements must align with event needs.  Required Details:   * Sport type * Game date and time * Venue * Opponent (if applicable) * Required crew positions | | |
| Related Use Cases: | UC 18: Admin Creates Game Schedule  UC 21: Admin Edits Game Schedule  UC 22: Admin Schedules Crew | | |
| Assumptions: | The system contains accurate game details and venue information.  Admin has the game schedule to add.  Admins have the necessary permissions to modify schedules. | | |
| Open Issues: | Should the system allow bulk additions of multiple games at once?  Should the system auto-suggest potential game slots based on venue availability? | | |

# Use Case 23: Admin Schedules Crew

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC-23: Admin Schedules Crew | | |
| Created By: | Michala Rogers | Date Created: | 2/3/2025 |
| Primary Actor: | Admin | Secondary Actors: | Crew Member |
| Trigger: | The Admin initiates the process to assign crew members to scheduled games. | | |
| Description: | The Admin assigns crew members to available roles in a game schedule. This ensures that each game has the required staff based on availability, qualifications, and predefined roles. Once scheduled, crew members receive notifications confirming their assignments. | | |
| Preconditions: | PRE-1: The Admin is logged into the system.  PRE-2: A game schedule exists with games requiring crew assignments.  PRE-3: Crew members have submitted availability for the scheduled games.  PRE-4: The system contains up-to-date records of crew members and their qualified roles. | | |
| Postconditions: | POST-1: Crew members are successfully assigned to scheduled games.  POST-2: The system saves the updated crew schedule.  POST-3: Crew members receive notifications regarding their assigned games. | | |
| Main Success Scenario: | 1. The Admin navigates to the "Schedule Crew" section. 2. The Admin selects a game from the list of scheduled events. 3. The system displays a list of available crew positions for the selected game. 4. The Admin assigns crew members to each required position based on availability and qualifications. 5. The system validates the assignments to ensure:    1. Crew members meet role qualifications.    2. There are no scheduling conflicts. 6. The Admin confirms the crew assignments. 7. The system updates the game schedule with assigned crew members. 8. The system sends notifications to scheduled crew members. 9. Use case ends. | | |
| Extensions: | 6a. Crew member is unavailable or has scheduling conflict:   * 6a1. The system alerts the Admin about conflicts. * 6a2. The Admin selects an alternative crew member.   7a. The Admin wants to save progress but not finalize assignments:   * 7a1. The system allows the Admin to save the schedule as a draft. | | |
| Priority: | High | | |
| Frequency of Use: | Approximately 1 user, average of 10-15 usages per week (during scheduling periods). | | |
| Business Rules: | BR-22: A schedule cannot be published until it is reviewed and finalized. | | |
| Associated Information: | Crew roles must align with predefined event requirements.  The system should allow batch scheduling for multiple games at once. | | |
| Related Use Cases: | UC 4: Crew Member Views Their Scheduled Games  UC 5: Crew Member Views General Game Schedule  UC 6: Crew Member Views Crew List  UC 22: Admin Schedules Crew | | |
| Assumptions: | The system maintains an updated list of available crew members.  Admins have full visibility into crew availability before scheduling. | | |
| Open Issues: | Should the system suggest optimal crew assignments based on previous schedules?  Should there be an automatic backup system in case a crew member cancels last-minute? | | |

# 