**Use Case ID:** 007 Kick Priveleges

**Use Case Level:** User Privileges

**Details:**

* **Actor:** Organizer.
* **Pre-conditions:**
  + Organizer has successfully logged onto the system.
  + The application is open.
  + There is at least one member part of the organization.
* **Description:**

**Trigger:**

1. Use case begins when Organizer clicks on the member management tab.
2. The system shall provide the Organizer with a list of members that are sorted.
3. The Organizer will click on the member that they want to kick out.
4. The Organizer will then click on the kick button in the member description.
5. The club adminstrator will provide a short description to the member why they are being kicked from their club.
6. The club adminstrator will send the request by selecting the send button.
7. The system shall notify the Organizer if the request was submitted correctly.
8. Use case ends when the system will removes the member from the club.

* **Relevant requirements:**

None

* **Post-conditions:**

1. The request to kick the member is saved by the system.
2. When the kicked member logs in they will receive a message notifying why they have been kicked from said club.

**Alternative Courses of Action**

1. In step D.6 (step 6 of Description section) the user has the option to cancel the kick request.
2. In step D.5 if the description is left blank the system will provide the user with a message to give a short reason why the member is being kicked.
3. In step D.2 the list of users can be sorted alphabetically or by ranking.

**Exceptions:**

1. There are no members in the club to kick.

**Related Use Cases:** None.

**---------------------------------------------------------------------------------------------------------------------**

**Decision Support**

**Frequency:** On average 50 kick requests are made monthly by Organizer.

**Criticality:** High. Allows the club adminstrator to kick inactive members to make space for other people that will contribute to their organization.

**Risk:** Medium. Implementing this use case requires web-based technology.

**Constraints:**

* Usability:
  + No previous training time.
  + On average the user should take 2 minutes to complete the kick request to the system.
* Reliability
  + Mean time to failure – 5% failures for every month of operation is acceptable.
  + Availability – Down time for Login Back-up 30 minutes in a 24 hour period.
* Performance
  + Request should be sent and saved within 6 seconds.
  + System should be able to handle 100 request in 1 minute.
* Supportability
  + The Event Creation should be supported by Chrome, Mozilla, and IE.
* Implementation
  + The implementation shall use JS React for front-end, and Java-based software for back-end.

---------------------------------------------------------------------------------------------------------------------

**Modification History**

**Owner:** Anthony Sanchez-Ayra

**Initiation date:** 09/03/2019

**Date last modified:** 09/03/2019

**Use Case ID:** 008 Promotion Privileges

**Use Case Level:** User Privileges

**Details:**

* **Actor:** Organizer.
* **Pre-conditions:**
  + Organizer has successfully logged onto the system.
  + The application is open.
  + There is at least one member part of the organization.
* **Description:**

**Trigger:**

1. Use case begins when Organizer clicks on the member management tab.
2. The system shall provide the Organizer with a list of members that are sorted.
3. The Organizer will click on the member that they want to promote.
4. The Organizer will then click on the promote button in the member description.
5. The club adminstrator will provide a short congratulatory message for the user receiving the promotion.
6. The club adminstrator will send the request by selecting the send button.
7. The system shall notify the Organizer if the request was submitted correctly.
8. Use case ends when the system promotes the member of the club to the desired position.

* **Post-conditions:**

1. The request to promote the member is saved by the system.
2. When the promoted member logs in they will receive a message notifying them of their promotion.

**Alternative Courses of Action**

1. In step D.6 (step 6 of Description section) the user has the option to cancel the promotion request.
2. In step D.5 if the congratulatory message is left blank the system will provide the user with a message to give a short reason why the member is being promoted.
3. In step D.2 the list of users can be sorted alphabetically or by ranking.

**Exceptions:**

1. There are no members in the club to promoted.
2. The Organizer attempts to promote the leader of the club.
3. The Organizer attempts to promote a member to a position that has already been filled within the organization.

**Concurrent Use Cases:** None.

**Related Use Cases:** None.

**---------------------------------------------------------------------------------------------------------------------**

**Decision Support**

**Frequency:** On average 10 promotion requests are made monthly by Organizer.

**Criticality:** High. Allows the club adminstrator to promote active members to make sure that the club can runs operations smoothly.

**Risk:** Medium. Implementing this use case requires web-based technology.

**Constraints:**

* Usability:
  + No previous training time.
  + On average the user should take 2 minutes to complete the promotion request to the system.
* Reliability
  + Mean time to failure – 5% failures for every month of operation is acceptable.
  + Availability – Down time for Login Back-up 30 minutes in a 24 hour period.
* Performance
  + Request should be sent and saved within 6 seconds.
  + System should be able to handle 100 request in 1 minute.
* Supportability
  + The Event Creation should be supported by Chrome, Mozilla, and IE.
* Implementation
  + The implementation shall use JS React for front-end, and Java-based software for back-end.

---------------------------------------------------------------------------------------------------------------------

**Modification History**

**Owner:** Anthony Sanchez-Ayra

**Initiation date:** 09/03/2019

**Date last modified:** 09/03/2019

**Use Case ID:** 009 Notifications

**Use Case Level:** High-Level

**Details:**

* **Actor:** Member.
* **Pre-conditions:**
  + Member has an account in the system.
  + Member is part of at least one organization and is subscribed to events.
* **Description:**

1. Use case begins when member clicks on the clubs tab.
2. The system shall provide the member with a set of cards that represent the clubs that they are a part of.
3. The member will click on the club that they want to obtain notifications for.
4. The member will click on get event news button on the club description page.
5. The system shall notify the member that the request was submitted correctly.
6. Use case ends when the system allows the user to receive notifications for events of the club.

* **Relevant requirements:**

None

* **Post-conditions:**

1. The request to receive notifications from the club is saved in the system.

**Alternative Courses of Action**:

None.

**Exceptions:**

None.

**Concurrent Use Cases:** None.

**Related Use Cases:** None.

**---------------------------------------------------------------------------------------------------------------------**

**Decision Support**

**Frequency:** On average 30 notification requests are made daily by the member.

**Criticality:** High. Allows the member to know when the organization that they are a part of is conducting events.

**Risk:** Medium. Implementing this use case requires web-based technology.

**Constraints:**

* Usability:
  + No previous training time.
  + On average the user should take 2 minutes to complete the notification request to the system.
* Reliability
  + Mean time to failure – 5% failures for every month of operation is acceptable.
  + Availability – Down time for Login Back-up 30 minutes in a 24 hour period.
* Performance
  + Request should be sent and saved within 6 seconds.
  + System should be able to handle 100 request in 1 minute.
* Supportability
  + The Event Creation should be supported by Chrome, Mozilla, and IE.
* Implementation
  + The implementation shall use JS React for front-end, and Java-based software for back-end.

---------------------------------------------------------------------------------------------------------------------

**Modification History**

**Owner:** Anthony Sanchez-Ayra

**Initiation date:** 09/03/2019

**Date last modified:** 09/03/2019

**Use Case ID:** 010 Create Clubs

**Use Case Level:** High-Level

**Details:**

* **Actor:** Organizer
* **Pre-conditions:**
  + Organizer has an account in our application.
  + Organizer is successfully logged into the application.
* **Description:**

**Trigger:**

1. Use case begins when Organizer clicks on the clubs tab.
2. The system shall provide the Organizer with a set of cards that represent the clubs that they are a part of and a Create Club option.
3. The Organizer will click on the Create Club option.
4. The system shall provide the Organizer with a form to fill out, asking for the following details:
   1. Club Name
   2. Club Description
   3. Requirements for joining
   4. Privacy of the club (whether its open to others or not).
5. The system shall notify the Organizer that the request was submitted correctly.
6. Use case ends when the system allows the Organizer to see that they have created a new club.

* **Post-conditions:**

1. The request to create a club is stored in the system.
2. The club is shown to members depending on its privacy settings.

**Alternative Courses of Action**:

1. In step D.4 the user has the option to cancel the creation of their club.
2. In step D.5 if any of the fields are left blank the system will provide the user with a message to fill in all the fields.
3. In step D.5 the system shall ask the user to confirm if they would like to create a club.

**Exceptions:**

1. If the Organizer tries to make a club that already exists then they will get an error message.

**Concurrent Use Cases:** None.

**Related Use Cases:** None.

**---------------------------------------------------------------------------------------------------------------------**

**Decision Support**

**Frequency:** On average 20 club creation requests are made monthly by the Organizer.

**Criticality:** High. Allows the Organizer to create a club which allows new communities to grow around campus.

**Risk:** Medium. Implementing this use case requires web-based technology.

**Constraints:**

* Usability:
  + No previous training time.
  + On average the user should take 2 minutes to complete the notification request to the system.
* Reliability
  + Mean time to failure – 5% failures for every month of operation is acceptable.
  + Availability – Down time for Login Back-up 30 minutes in a 24 hour period.
* Performance
  + Request should be sent and saved within 6 seconds.
  + System should be able to handle 200 request in 1 minute.
* Supportability
  + The Event Creation should be supported by Chrome, Mozilla, and IE.
* Implementation
  + The implementation shall use JS React for front-end, and Java-based software for back-end.

---------------------------------------------------------------------------------------------------------------------

**Modification History**

**Owner:** Anthony Sanchez-Ayra

**Initiation date:** 09/04/2019

**Date last modified:** 09/04/2019

**Use Case ID:** 011 Attendance Tracker

**Use Case Level:** High-Level

**Details:**

* **Actor:** Member
* **Pre-conditions:**
  + Member has an account in our application.
  + Member is successfully logged into the application.
  + Member is part of a club and is attending an event hosted by said club.
* **Description:**

**Trigger:**

1. Use case begins when member clicks on the events tab.
2. The system shall provide the member with a sorted list of events that the user has signed up for.
3. The member will click on the event that they are currently attending.
4. The system shall provide the member with a description of the event as well as a button that says “I’m here!”
5. The user shall click on the “I’m here” button.
6. The system shall process the request for the click.
7. Use case ends when the system notifies the user that their attendance at the event was noted.

* **Post-conditions:**

1. The attendance request is saved in the system, along with arrival time.
2. The member is awarded a certain amount of points for attending the event.

**Alternative Courses of Action**:

1. In step D.10 the “I’m here” button will only appear if the user is at the location where the event is occuring.
2. In step D.8 the sorted list provided by to the user can be sorted by date the event will take place on or by club name.

**Exceptions:**

1. If the member tries to click the I’m here button 15 minutes before the event is ending they will not get credit for attending the event.

**Concurrent Use Cases:** None.

**Related Use Cases:** None.

**---------------------------------------------------------------------------------------------------------------------**

**Decision Support**

**Frequency:** On average 100 attendance requests are made weekly by the Organizer.

**Criticality:** High. Allows the member to notify their club that they are active in their organization.

**Risk:** High. Implementing this use case requires web-based technology and GPS tracking.

**Constraints:**

* Usability:
  + No previous training time.
  + On average the user should take 2 minutes to complete the notification request to the system.
* Reliability
  + Mean time to failure – 5% failures for every month of operation is acceptable.
  + Availability – Down time for Login Back-up 30 minutes in a 24 hour period.
* Performance
  + Request should be sent and saved within 6 seconds.
  + System should be able to handle 1000 request in 1 minute.
* Supportability
  + The Event Creation should be supported by Chrome, Mozilla, and IE.
* Implementation
  + The implementation shall use JS React for front-end, and Java-based software for back-end.

---------------------------------------------------------------------------------------------------------------------

**Modification History**

**Owner:** Anthony Sanchez-Ayra

**Initiation date:** 09/04/2019

**Date last modified:** 09/04/2019

**Use Case ID:** 012 Cancel an event.

**Use Case Level:** User Goal

**Details:**

* **Actor:** Organizer
* **Pre-conditions:**
  + Organizer has an account in our application.
  + Organizer is successfully logged into the application.
  + Organizer is part of a club.
* **Description:**

**Trigger:**

1. Use case begins when organizer clicks on the event that they want to cancel.
2. The system shall redirect the organizer to the Event Description view, which shall present them with a button labeled cancel event.
3. The organizer will click on the cancel event button.
4. The organizer will click yes on the validation message displayed by the system.
5. The system shall notify the organizer that the event was cancelled.
6. End case ends when the system removes the event from being viewed.

* **Post-conditions:**

1. The cancellation request is saved in the system.
2. The system notifies all users that subscribed to the event that it has been cancelled.

**Alternative Courses of Action**:

1. In step D.3 the system will prompt the organizer with a validation message to confirm that they actually want to cancel the event.

**Exceptions:**

1. The database is not active.
2. The Event Description view is not active.
3. The validation message is not active.

**Concurrent Use Cases:** None.

**Related Use Cases:** None.

**---------------------------------------------------------------------------------------------------------------------**

**Decision Support**

**Frequency:** On average 5 cancellation requests are made weekly by the organizer.

**Criticality:** High. Allows the organizer to cancel an event whenever necessary.

**Risk:** High. Implementing this use case requires web-based technology.

**Constraints:**

* Usability:
  + No previous training time.
  + On average the user should take 2 minutes to complete the notification request to the system.
* Reliability
  + Mean time to failure – 5% failures for every month of operation is acceptable.
  + Availability – Down time for Login Back-up 30 minutes in a 24 hour period.
* Performance
  + Request should be sent and saved within 6 seconds.
  + System should be able to handle 100 request in 1 minute.
* Supportability
  + The Event Creation should be supported by Chrome, Mozilla, and IE.
* Implementation
  + The implementation shall use JS React for front-end, and Java-based software for back-end.

---------------------------------------------------------------------------------------------------------------------

**Modification History**

**Owner:** Anthony Sanchez-Ayra

**Initiation date:** 09/04/2019

**Date last modified:** 09/04/2019