# Software Requirements Specifications
# University Event Organizer

**Version 1.0**

**Whole Scope Solutions**

**January 30, 2017**

# Table of Contents

# Revision History

| Name | Date | Reason for Changes | Version |
|---|---|---|---|
| Specification Document | January 30, 2017 | Initial drafting | RS 1.0 |
| | | | |
| | | | |

# 1 Introduction

## 1.1 Purpose

This document is set to outline the requirements, functional and non-functional, as well as use cases for the University Event Organizer as laid out in the RFP received from Teck4. The University Event Organizer is a web app to be used by students and faculty at the University of Victoria for the purpose of event advertisement.

## 1.2 Project Scope

The software being proposed is a web application that manages events for the University of Victoria for the use of faculty, students, university clubs, and organizations. This software is meant to better organize and monitor school events and to make sure the activities taking place are good representations of the university's culture. The web application should provide students, faculty and anonymous users with access to legitimate University of Victoria hosted events.

## 1.3 Glossary of Terms

**UVic –** University of Victoria

**VPC –** Virtual Private Cloud

## 1.4 References

- *Request for Proposal, University Event Organizer, Version 1.2*. Teck4. [Online] Available https://wholescopesolutions.github.io/docs/teck4/rfp.pdf.

## 1.5 Overview

This document will contain an overarching description of the Event Planner project to be developed for Teck4. This description includes features, user groups, the system environment, constraints, and any dependencies the system may have. It will outline the features of the system as well as the functional requirements that go along with them, a description of the feature, and how it is intended to function.

Any external interfaces used by the system will be discussed in depth. Non-functional requirements will also be outlined as well as any other details necessary for the development of the system.

# 2 Overall Description

## 2.1 Product Perspective

The University Event Organizer is an entirely new system being built to fill a need at UVic (University of Victoria). This system is intended to centralize and regulate university event information. It is a stand-alone system which does not depend on any previously developed systems at UVic. The system will connect to Facebook and Google events to allow events to be shared or saved to the user's calendar.

## 2.2 Product Features

The University Event Organizer is being developed to allow UVic student clubs and groups as well as faculty to publicize events. All users will have the ability to view events as well as flag inappropriate events. Any student with a UVic email address can create an account but only pre-approved members of UVic student organizations or clubs can post or edit events. The system administrator and the club administrators will be able to approve events; however, only system administrators will be able ban users for posting inappropriate events and remove events from the page. Registered users may RSVP for events, which will be tracked in an exportable calendar.

## 2.3 User Classes and Characteristics

The expected users of the system can be classified by one of the following categories:

- **Anonymous User**: Any user who is not logged in to an account, a portion of UVic students who have chosen not to set up an account, and any community members who may or may not be associated with UVic. These users may view and flag events in the application.
- **Authenticated User**: Exclusive to UVic students, these are persons who have set up an account and authenticated successfully. These users may register for events, export events to external calendar applications and perform all actions of an anonymous user.
- **Privileged User**: Users who have been promoted by club administrators or administrators, and are generally trusted members (at the discretion of club administrators) who have the ability to create and edit event postings. These users may also perform all actions of an authenticated user.
- **Club Administrator**: Designated representatives of an organization who grants privileged user access rights and is responsible for moderating events created by their organization. These users may also perform all actions of a privileged user
- **System Administrator**: A super user role, which entitles said user to remove events, ban users, and register or deregister club administrators.

Most expected users should fall under the authenticated user class, while the club administrator class requires the most system functionality.

## 2.4 Operating Environment

The system will be hosted on an Amazon EC2 M4.2xlarge instance, which provides a 2.3 GHz Intel Xeon® E5-2686 v4 (Broadwell) processor, 32GB of memory and 1000 Mbps dedicated bandwidth. This hosted server will be running the latest stable version of Debian OS (v8.0) 64-bit. The EC2 instance will be hosted inside an Amazon VPC to ensure a greater level of security as well as the MySQL database which will also be hosted on a M4.2xlarge instance. Installed software will include but is not limited to: Nginx, NodeJS, and ReactJS.

## 2.5 Design and Implementation Constraints

The most pressing constraint on the design and implementation of the system is our limited access to UVic data. Information privacy and security must be considered for legal and ethical reasons in this publicly accessible system. Event calendar and export information must conform to external application interfaces. Anonymous users may access the system and flag events, which could potentially be abused.

## 2.6 Assumptions and Dependencies

The system's view event page will include a Facebook share option, as well as the ability to export the event to the users Google Calendar. To use either of these features the user will be required to have a registered account. The use of these external services is completely optional, and is not required for the core functionality of the application.

While the community outside of UVic may view events, registration and login is restricted to UVic students and faculty.

# 3 System Features

## 3.1 Create an Event

### 3.1.1 Description and Priority

This feature should allow privileged users, club administrators and administrators to create new events and post them to the web application. Event creation is high priority and should be one of the first features implemented.

### 3.1.2 Stimulus/Response Sequences

| | |
|---|---|
| Stimulus | User requests new event |
| Response | System returns the new event page with multiple inputs |
| Stimulus | User specifies event data and submits |
| Response | System exits to the main page, event is created |

### 3.1.3 Functional Requirements

**[R.3.1.1]** In the event that an event was created by a privileged user, it must be sent to the appropriate club administrator or the site administrator for approval. Events created by a club administrator or site administrator are automatically approved.

**[R.3.1.2]** Any new event without the required information will not be created, if a user attempts to create an event without the required parameters the system will respond with a request for the user to submit the missing parameters.

**[R.3.1.3]** In the case that a user tries to input more than 5 images into a single event the system will respond with a prompt that that cannot add more than 5 images to an event and return the user to the new event page.

## 3.2 View Event

### 3.2.1 Description and Priority

This feature should allow any user of the system to view an event, this function can be accessed from searching or the user's existing schedule. This feature is of high priority because it is used to access many other features including the RSVP feature, the reporting events, flagging events and editing an event.

### 3.2.2 Stimulus/Response Sequences

| | |
|---|---|
| Stimulus | User requests to view an event |
| Response | System displays the event |

### 3.2.3 Functional Requirements

**[R.3.2.1]** Each event should track and display how many times it has been viewed

## 3.3 Register Club Administrator

### 3.3.1 Description and Priority

This feature should allow System Administrator users to appoint club administrators. This feature is of high priority, due to its required implementation for the system to work, as club administrators are required to create events which users register for. The current user must be an administrator for this action and the selected user must have an account in the current system.

### 3.3.2 Stimulus/Response Sequences

| | |
|---|---|
| Stimulus | Administrator requests a user profile |
| Response | System displays user profile |
| Stimulus | Administrator requests to upgrade user account |
| Response | System upgrades said user to Club Administrator |

### 3.3.3 Functional Requirements

**[R.3.3.1]** Once a user is registered as a club administrator, they should receive an email notifying them of their new privileges. This should only occur once.

## 3.4 Registering a Privileged User

### 3.4.1 Description and Priority

This feature allows a club administrator or the system administrator to upgrade a normal user account to a privileged user. To do this the user must already have an account with a UVic email address. This is considered to be a high priority feature as it is required for non-admin users to post events. The administrator must be logged in and have club administrator status or better to register a privileged user. The user to be promoted must have an existing registered account with a UVic email.

### 3.4.2 Stimulus/Response Sequences

| | |
|---|---|
| Stimulus | The administrator requests to view the user profile |
| Response | The system displays the requested user profile |
| Stimulus | The administrator requests to grant privileged status |
| Response | The system upgrades the user to privileged status |

### 3.4.3 Functional Requirements

**[R.3.4.1]** After the user has been given the privileged user permissions, they should be sent an email notifying them of their new permissions.

## 3.5 Register User

### 3.5.1 Description and Priority

This feature allows any UVic student without an account to create a new account to log in with. This action is of high priority since no authenticated users can exist without it.

### 3.5.2 Stimulus/Response Sequences

| | |
|---|---|
| Stimulus | The anonymous user requests registration |
| Response | The system responds with the appropriate registration form |
| Stimulus | User inputs registration details and submits the form |
| Response | System validates information, creates user, and sends verification email |
| Stimulus | User visits email verification page |
| Response | System enables account to allow login, and redirects user to login page |

### 3.5.3 Functional Requirements

**[R.3.5.1]** System must read user information from the database to avoid creating duplicate users.

**[R.3.5.2]** System must write user information to the database to create and enable users.

**[R.3.5.3]** System must sanitize and validate user data entered into registration form.

## 3.6 Log in

### 3.6.1 Description and Priority

This feature allows anyone with an account to access the system as a user. This action is of high priority since no user can access their special features without it.

### 3.6.2 Stimulus/Response Sequences

| | |
|---|---|
| Stimulus | The anonymous user requests to log in |
| Response | The system responds with the appropriate form |
| Stimulus | User inputs login details and submits the form |
| Response | System redirects to main page with additional features for logged in user |

### 3.6.3 Functional Requirements

**[R.3.4.1]** System must import user information from the database

## 3.7 Search Event

### 3.7.1 Description and Priority

This system feature enables all users to search for school events that are stored on the System's database. The user could provide keywords for different search fields as desired for different search results. The system will return events that matches all of the keywords provided by the user. This feature is to be considered High priority. The reason is that searching an event is one of the main method for users to find events in the system.

### 3.7.2 Stimulus/Response Sequences

| | |
|---|---|
| Stimulus | User request to search for events |
| Response | The system displays resulting events |

All users should be able to search for events. If the user does not provide any search parameters, the system will display all the available events. If a keyword was given by the user, the system will only display the events that matches the given keywords. If the user chooses to search by category, the system will return all events with the given category. Only privileged users are allowed to search for restricted events. Regular registered users can only search for age restricted events if they are 19 years or older.

### 3.7.3 Functional Requirements

**[R3.7.1]** The Search Event screen shall allow the users to search events by using keywords such as name, category, date.

**[R3.7.2]** The Search Event screen shall display the events that the users has searched for.

**[R3.7.3]** The Search Event screen shall allow users to select an events for viewing.

**[R3.7.4]** The Search Event screen shall allow privileged users to search for the restricted events.

# 3.8 Approve Event

### 3.8.1 Description and Priority

This feature provides the ability for system and club administrators to approve pending events proposed by privileged users. This feature is to be considered high priority.

### 3.8.2 Stimulus/Response Sequences

| | |
|---|---|
| Stimulus | System or Club administrator approves event |
| Response | User is notified of events approval status change |

### 3.8.3 Functional Requirements

**[R3.8.1]** When an event is approved, the event information needs to be made publicly available and searchable.

**[R3.8.2]** Event acted upon is marked as approved in the database.

# 3.9 Remove an Event

### 3.9.1 Description and Priority

This feature gives the system administrator the ability to remove posted events. This feature is high priority.

### 3.9.2 Stimulus/Response Sequences

| | |
|---|---|
| Stimulus | System administrator selects to remove event |
| Response | System prompts for confirmation of removal |
| Stimulus | System administrator confirms |
| Response | System dismisses prompt and redirects user to landing page |

### 3.9.3 Functional Requirements

**[R3.9.1]** When an event is removed, the event information needs to be made not publicly available and not searchable.

# 3.10 RSVP For an Event

### 3.10.1 Description and Priority

This system feature allows registered users register for posted events. This action will add the event information to that user's application calendar information which will be available for export in other features. This feature will update the information regarding the attendance quantity. This feature is medium priority.

### 3.10.2 Stimulus/Response Sequences

Stimulus User selects an event to register for
Response User is notified of successful (or failed) registration

### 3.10.3 Functional Requirements

**[R3.10.1]** When a student RSVPs for an event, this information needs to be saved to the event so that other users can see how many people are planning to attend an event.

## 3.11 Share an Event

### 3.11.1 Description and Priority

This system feature allows any logged in user to share a posted event through Facebook. This feature is medium priority.

### 3.11.2 Stimulus/Response Sequences

Stimulus User selects to share an event
Response A Facebook prompt is displayed where they can share the event

Stimulus User shares event
Response Facebook prompt dismisses and regular system resumes

### 3.11.3 Functional Requirements

**[R3.11.1]** Sharing the event to Facebook does not affect any existing data in the database.

## 3.12 Export an Event to Calendar

### 3.12.1 Description and Priority

This system feature allows any logged in user to export their calendar of events they have registered for to their Google Calendar. The user must have a Google account. This feature is medium priority.

### 3.12.2 Stimulus/Response Sequences

Stimulus User selects to export their calendar
Response A Google prompt is displayed for exporting options

Stimulus User finalizes export
Response Google prompt is dismissed

### 3.12.3 Functional Requirements

**[R3.12.1]** Exporting the calendar does not affect any existing data in the database.

## 3.13 Edit an Existing Event

### 3.13.1 Description and Priority

This feature allows system administrators, club administrators and privileged users to edit information of events. The events a user may edit varies on the user's level:

- System administrators may edit any event in the system.
- Club administrators may edit any event that is registered under their domain, such as created by them, or their derived privileged users.
- Privileged users may only edit events they created.

This feature is medium priority.

### 3.13.2 Stimulus/Response Sequences

| | |
|---|---|
| Stimulus | User selects to edit an event they have the ability to edit |
| Response | System takes user to the event edit view |
| Stimulus | User modifies desired information and saves changes |
| Response | System saves changes and redirects user to normal viewing of event |

### 3.13.3 Functional Requirements

**[R3.13.1]** The event information is updated in the database and displayed correctly on normal viewing of the same event.

**[R3.13.2]** If the event has been previously approved the event returns to it's previously state and can now be flagged or reported

## 3.14 Remove a Privileged User's Privileges

### 3.14.1 Description and Priority

This feature allows a system or club administrator to remove a privileged users privileges from the system. A system administrator may revoke club administrators and privileged user's privileges. A club administrator may only remove privileges from privileged users in which they initially granted. This feature is medium priority.

### 3.14.2 Stimulus/Response Sequences

| | |
|---|---|
| Stimulus | The administrator selects to revoke a user's privileges |
| Response | The system prompts for confirmation of the revocation. |
| Stimulus | The administrator confirms. |
| Response | The system dismisses the prompt. |

### 3.14.3 Functional Requirements

**[R3.14.1]** The revoked privileged user's information in the database is updated to reflect new user status.

## 3.15 Remove a Club Administrators Privileges

### 3.15.1 Description and Priority

This feature allows a system or club administrator to remove a privileged users privileges from the system. A system administrator may revoke club administrators and privileged user's privileges. A club administrator may only remove privileges from privileged users in which they initially granted. This feature is medium priority.

### 3.15.2 Stimulus/Response Sequences

Stimulus The administrator selects to revoke a user's privileges
Response The system prompts for confirmation of the revocation.

Stimulus The administrator confirms.
Response The system dismisses the prompt.

### 3.15.3 Functional Requirements

**[R3.15.1]** The revoked privileged user's information in the database is updated to reflect new user status.

## 3.16 Ban a User

### 3.16.1 Description and Priority

This feature allows the system administrator to ban a user from the service for various reasons. This feature is low priority.

### 3.16.2 Stimulus/Response Sequences

Stimulus System administrator enters the user email of the account to be banned.
Response The system prompts for confirmation of the user ban.

Stimulus The system administrator confirms the ban.
Response The system bans the user and displays a confirmation or failure message.

### 3.16.3 Functional Requirements

**[R3.16.1]** The user being banned must be flagged as banned in the database and prevented from logging in again

**[R3.16.2]** If the User is logged in when they are banned the system must log them out and flag them as banned in the database.

## 3.17 Report an Event

### 3.17.1 Description and Priority

This system feature any user with a valid account with the system to report events that they deem inappropriate in nature. This is a low priority feature as much of the system can function without this feature implemented.

### 3.17.2 Stimulus/Response Sequences

| | |
|---|---|
| Stimulus | User selects report event on a currently selected event |
| Response | System responds with an input form |
| Stimulus | User writes concerns about the event and submits the form |
| Response | System closes the form with an appropriate message |

### 3.17.3 Functional Requirements

**[R3.17.1]** Once an event is reported, the system marks the event in the database as flagged, and logs the user message.

**[R3.17.2]** The event, is added to the club administrators queue along with the ser messages if there was more than one report made.

## 3.18 Flag an Event

### 3.18.1 Description and Priority

This system feature allows any level of user to flag events they believe to be inappropriate or requires attention. This feature is to be considered Low priority.

### 3.18.2 Stimulus/Response Sequences

| | |
|---|---|
| Stimulus | User selects to flag event |
| Response | System marks event as flagged |

### 3.18.3 Functional Requirements

**[R3.18.1]** Once an event is flagged, the system marks the event in the database as flagged.

**[R3.18.2]** The event, if not already in the queue, is added to the respective club administrator's queue of flagged events which require review.

# 4 External Interface Requirements

## 4.1 Hardware Interfaces

The application must be web responsive and compatible with mobile and desktop devices. Any device running this software must have a compatible web browser installed. Keyboard, touch, and mouse input must be captured for this application to function properly.

## 4.2 Software Interfaces

The software uses a MySQL database for the storage of user data, event statistics and events. This software should run well on operating systems including Windows Vista and later, and OS X 10.8 Mountain Lion and later. Software should also run well on current Linux distributions produced after 2010. The web application should utilise existing APIs for Google Calendar exporting and Facebook sharing. Calendar exporting should use a call to the Google Calendar API to export the users schedule or event listing to their Google account. Event sharing should use Facebook's API to share a selected event to a user's friends. Event title, descriptions, date, time and location should be shared when the event sharing functionality is called. For the calendar update function, title, description, date and time should be shared to Google Calendar. Tools for development will include any web browser with adequate inspection tools, Sublime Text and Visual Studio Professional Edition 2015.

## 4.3 Communication Interfaces

Each user account must register with a valid UVic email accounts for account verification and notifications. The software uses TCP connections with HTTP/HTTPS responses for network server communications. This application will use a Google and Facebook connection to share event information. The function will be compatible with modern web browsers such as Google Chrome, Firefox, Opera, and Internet Explorer, and any other browser compatible with HTML 5.

# 5 Other Non-Functional Requirements

## 5.1 Performance Requirements

Frequent use of the system is expected during 9 A.M to 10 P.M. all days of the week, thus the regular system maintenance will be performed at times after 12 A.M. and before 6 A.M. Required physical maintenance will be minimal, as hosting is provided by external providers, following the specifications in section 2.4 (Operating Environment). Therefore, the system will operate during the specified frequent use interval.

## 5.2 Safety Requirements

The system allows a user with privileged or higher status to create events. Events are approved by administrators, but this does not eliminate the possibility of an inappropriate event being posted and seen by users. To combat this issue registered users can report inappropriate events, and all users can flag events. Administrators are notified of reported and flagged events. Administrators have the power to remove any event.

Users who post inappropriate events are tracked. To avoid registered users from flagging legitimate events their accounts are tracked of reporting history. The administrator can decide if they need to demote or ban a user.

## 5.3 Security Requirements

To create an account a user needs a UVic email address. To maximize user privacy only a minimal amount of user information will be stored. Only administrators can view a user's account information. User information includes but not limited to age, event creation history, event reporting history, and RSVPs.

The servers will be physically secured by Amazon as the system is running on AWS. Data must be encrypted at rest, and communication between the client interfaces and central servers must be encrypted. Internal audits will be conducted quarterly to ensure continued commitment to security guidelines.

## 5.4 Software Quality Attributes

### 5.4.1 Availability

**Description**: The amount of time the system is up and running correctly

**Metric**: The percentage of time the system is up in a year

**Goal**: The system should be available for at least 95% of the time in a year

### 5.4.2 Maintainability

**Description**: Issues should be able to be fixed and new functionalities should be able to be added to the system without the system going down. Maintenance should be done in a test environment and the updates pushed to the live system only if they are confirmed to be error-free after a thorough testing process.

**Metric:** Number of changes made directly to a production system without proper staging.

**Goal:** No changes should be made directly to a production system without proper staging.

# 6 Other Requirements

TBD

# Appendix A – Analysis Models

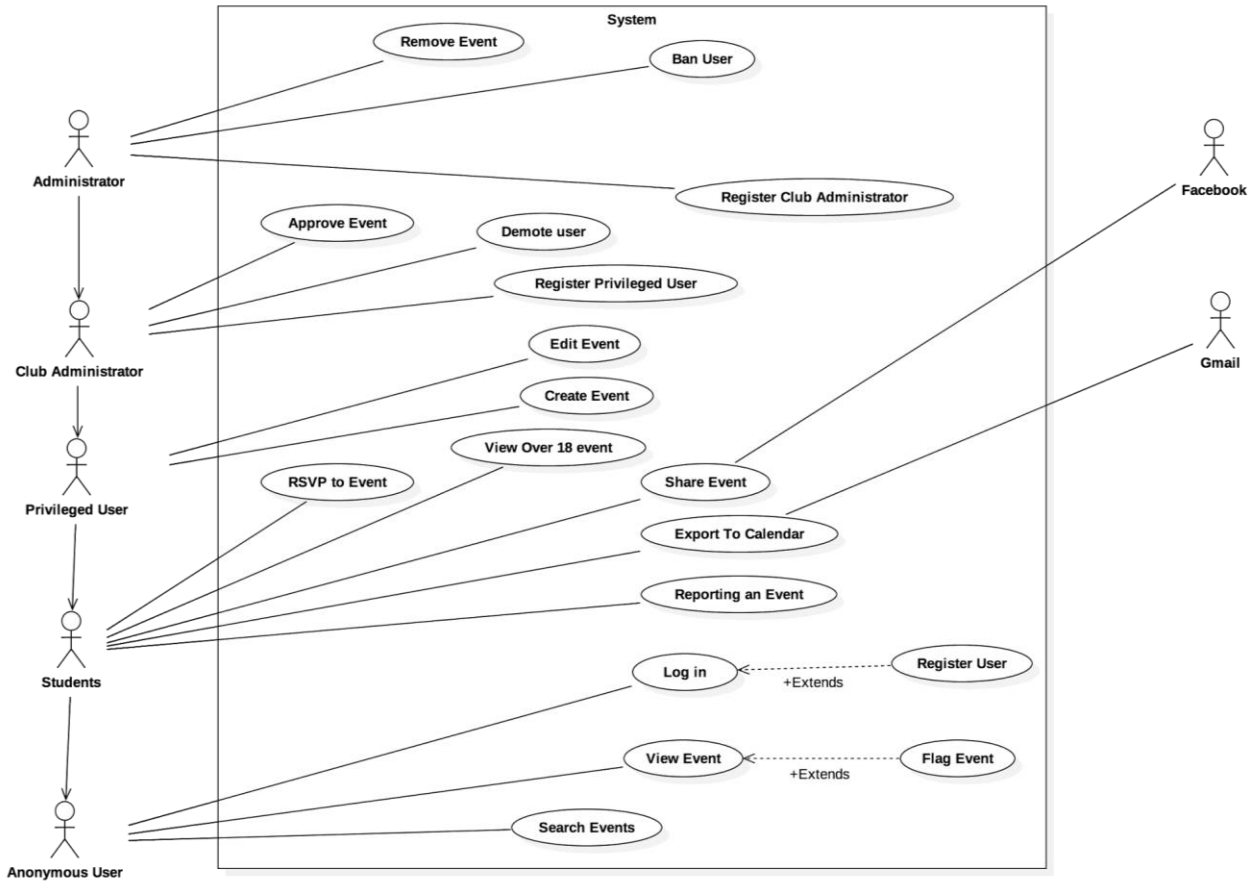


*Figure 1: Use Case Diagram*

## A.1 – Create Event

**Use Case Name**

Create event

**Description**

A Privileged User, Club Administrator or an Administrator uses the web application to create a new event.

**Actors**

Privileged User, Club Administrator or Administrator

**Pre-Conditions**

Users must have an account with privileged access or greater, know event location, date, time, categories, description, event tags and age restriction.

**Main flow**

1. User opens the web application
2. <User Authentication> User logs in with a Privileged user account
3. System redirects back to main application with user information
4. User selects option to create and event
5. <User input> System displays new event page with several form options
6. User inputs Date(s) of event
7. User specifies event location (address, room number etc)
8. Adds time(s) the event is being run
9. User adds a description of the event
10. User inputs the permitted age for the event (18+)
11. User adds some relevant event categories
12. Actor includes some event tags
13. Actor selects add an image
14. <import image> System prompts for an image file
15. User chooses an image to upload
16. <load file> System includes the image in the post
17. User choses to allow RSVP to the event specifies target email
18. User submits the form
19. System returns user to main application
20. User sees that the event has been created successfully with correct information

**Post-Conditions**

Only signed in users 18+ can view the post (if 18+ was chosen), post shows an RSVP option to signed in users.

**Alternative Flows**

A. At <User Authentication>, if the user entered incorrect login details
    a. The system displays a message saying “the username/password was not correct please try again”
    b. Return to <User Authentication>
B. At <User Input>, if the user neglects to include one or more of the steps 5-9 and submits the form
    a. The system displays a message saying “you are missing some fields” and specifying what fields were missed and does not leave the page.
C. At <User Input>, if the user neglects to include one or more steps 10-16 and submits the form
    a. The system submits the post as is with the specified information
D. At <load file>, if the user chooses to add another image
    a. Loop back to step 13
    b. When file is loaded system allows user to choose a preferred image
E. At <load file>, user chooses to add an image when they already have 5 added
    a. System displays a message ”cannot add more than 5 images to an event”

b. System returns to new event form

## A.2 – Edit an Event

**Use Case Name**

Edit an existing event

**Description**

A Privileged User, Club Administrator or an Administrator uses the web application to edit an existing event.

**Actors**

Privileged User, Club Administrator or Administrator

**Pre-Conditions**

Users must have an account with privileged access or greater.

**Main Flow**

1. User opens the web application
2. <User Authentication> User logs in with a Privileged user account
3. System redirects back to main application with user information
4. User selects option to edit an event
5. <User input> System displays event page with several form options
6. User edits intended information fields
7. User saves information
8. System returns user to main application
9. User sees that the event has been created edited with correct information

**Post-Conditions**

Only signed in users 18+ can view the post (if 18+ was chosen), post shows an RSVP option to signed in users.

**Alternative Flows**

A. At <User Authentication>, if the user entered incorrect login details
    a. The system displays a message saying “the username/password was not correct please try again”
    b. Return to <User Authentication>
B. At <User Input>, if the user clears a required field
    a. The system displays a message saying “you are missing some fields” and specifying what fields were missed and does not leave the page

## A.3 – Search for an Event

**Use Case Name**

Search for an event

**Description**

A user wants to find an event posted on the system.

**Actors**

Anonymous user

**Pre-Conditions**

Any level of user may search for an event using the system. User must have internet access. To see 18+ events users must be signed in and 18+.

**Main flow**

1. User opens the web application
2. <Search Criteria> Fills a search criteria with keywords for event they are seeking
3. System displays events related to keywords

**Post-Conditions**

To continue to register for events, users must be signed in.

**Alternative Flows**

A. At <Search Criteria>, if the user is signed in
    a. There will be suggested filters or search history recommended

## A.4 – RSVP to Event

**Use Case Name**

RSVP to event

**Description**

Allow a user with an account can rsvp to an event

**Actors**

Student, Privileged User, Club Administrator or Administrator

**Pre-Conditions**

The user must have an account

**Main flow**

1. User opens the web application
2. <User Authentication> User logs in

3. System redirects back to main application with user information
4. User searches for an event with relevant criteria
5. <Returns Events> system returns events matching the input criteria
6. User selects an event with the option to rsvp
7. <View Event> system displays event information
8. User chooses to rsvp to event
9. System displays that the rsvp action was performed successfully

**Post-Conditions**

User's calendar should update with the date and time of the event

**Alternative flows**

A. At <view event> user cancels the rsvp
    a. Use case ends
B. At <view event> event is at full capacity
    a. System prompts user "event is full you can't rsvp at this time"
    b. Event owner is notified that the event is at full capacity in the application

## A.5 Register Privileged User

**Use Case Name**

Register Privileged User

**Description**

Elevate a Student User to a Privileged User

**Actors**

Administrator or Club Administrator, Student user, Privileged user

**Pre-Conditions**

User must have Club administrator access or higher

**Main flow**

1. User opens the web application
2. <User Authentication> Administrator logs in
3. System redirects back to main application with user information
4. Administrator selects option to register a Privileged User
5. The system displays list of eligible users the administrator has authority over
6. <User Input> The system will filter users with keywords
7. Select the user to give privileged status
8. <Confirm Action> The system displays message prompt to confirm action
9. Administrator confirms the action and the selected user is now registered as a Privileged User

**Post-Conditions**

The new Privileged User can now Create and Edit Events

**Alternative flows**

A. At <User Authentication>, if the user entered incorrect login details
    a. The system displays a message saying “the username/password was not correct please try again”
    b. Return to <User Authentication>
B. At <User Input>, if the admin clears a required field
    a. The system displays a message saying “you are missing some fields” and specifying what fields were missed and does not leave the page
C. At <Confirm Action> if the administrator selects Cancel
    a. The selected user will not be given privileged status

## A.6 – Register Club Administrator

**Use Case Name**

Register Club Administrator

**Description**

Elevate a Student or Privileged user to a Club Administrator

**Actors**

Administrator, Club Administrator, Student user, Privileged user

**Pre-Conditions**

User must have Administrator Account

**Main flow**

1. User opens the web application
2. <User Authentication> Administrator logs in
3. System redirects back to main application with user information
4. Administrator selects option to register a Club Administrator
5. The system displays list of eligible users the administrator has authority over
6. <User Input> The system will filter users with keywords
7. Select the user to give Administrator status
8. <Confirm Action> The system displays message prompt to confirm action
9. Administrator confirms the action and the selected user is now registered as a Club Administrator

**Post-Conditions**

The new Club Administrator can now create and edit (if they were previously a student), register privileged users and approve events

**Alternative flows**

A. At <User Authentication>, if the user entered incorrect login details
    a. The system displays a message saying “the username/password was not correct please try again”
    b. Return to <User Authentication>
B. At <User Input>, if the admin clears a required field
    a. The system displays a message saying “you are missing some fields” and specifying what fields were missed and does not leave the page
C. At <Confirm Action> if the administrator selects Cancel
    a. The selected user will not be granted club administrator access
    b. The use case ends

## A.7 – Share Event (Facebook)

**Use Case Name**

Share Event (Facebook)

**Description**

A user wants to share an event to their Facebook timeline so that it is visible to their friends

**Actors**

Student, Privileged User, Club Administrator or Administrator, External Application (Facebook)

**Pre-Conditions**

The user must have an account in the system, a pre existing Facebook account, and there must be an event already in the system for the user to view.

**Main flow**

1. User opens the web application
2. <User Authentication> User logs in
3. System redirects back to main application with user information
4. User selects to share an event to Facebook
5. External application opens, prompting user for credentials
6. User is prompted by external application to confirm that they want to share the event to their timeline
7. The user confirms that they would like the post the event to their Facebook timeline

**Post-Conditions**

User’s Facebook timeline should now display the event that was just shared

**Alternative flows**

A. At <User Authentication>, if the user entered incorrect login details
    a. The system displays a message saying “the username/password was not correct please try again”
    b. Return to <User Authentication>

## A.8 – Ban User

**Use Case Name**

Ban User

**Description**

In which an Administrator uses the web application interface to disable a user from accessing the system.

**Actors**

Administrator

**Pre-Conditions**

Users must have an account with administrator access, the highest privilege in the system.

**Main flow**

1. Administrator opens a application to access the system.
2. <User Authentication> User logs in with a Administrator account.
3. System redirects back to main application with the administrator user interface.
4. User selects the option Ban User.
5. The system display a list of all users in the current system.
6. <User input> The system will filter the list of users with keywords.
7. The system will display only the users matching the given keywords.
8. Administrator disables the correct user by selecting Ban User.
9. <Confirm Action> The system prompts a message to reassure the Ban User action.
10. The Administrator confirms the action and the correct user is banned.

**Post-Conditions**

The user which was banned from the administrator can no longer gain access to the system. Also, the event hosts will receive a notification on the information of the banned user.

**Alternative flows**

A. At <User Authentication>, if the user entered incorrect login details
    a. The system displays a message saying “the username/password was not correct please try again”
    b. Return to <User Authentication>
B. At <User Input>, if the admin clears a required field

a. The system displays a message saying “you are missing some fields” and specifying what fields were missed and does not leave the page

C. At <Confirm Action>, if the administrator selects No
   a. The selected user will not be disabled from the system

## A.10 – Export to Google Calendar

**Use Case Name**

Export to Google calendar

**Description**

A user exports the event times they’re registered for to Google Calendar.

**Actors**

Student, Privileged User, Club Administrator or Administrator, Google Calendar

**Pre-Conditions**

The user must have an account in the system, and an existing account for Google Calendar, and be registered for at least one event in their calendar.

**Main flow**

1. User opens the web application
2. <User Authentication> User logs in
3. System redirects back to main application with user’s information
4. User reviews calendar and exports it to External Calendar Application
5. The use case ends.

**Post-Conditions**

User’s external calendar app should have the correct information that was exported.

**Alternative flows**

A. At <User Authentication>, if the user entered incorrect login details
   a. The system displays a message saying “the username/password was not correct please try again”
   b. Return to <User Authentication>

## A.11 – Remove Event

**Use Case Name**

Remove Event

**Description**

Removal of an existing event on the web application by an administrator

**Actors**

Administrator

**Pre-Conditions**

A user must have administrator access

**Main flow**

1. User opens the web application
2. <User Authentication> User logs in as an administrator
3. System redirects back to main application with an administrative interface
4. User selects option to view list of events
5. System displays list of all events with flagged events highlighted at the top of the list
6. User selects an event from the list to view details of the event
7. User selects the option of delete event
8. <Confirm Action> System prompts user to confirm removal of event
9. User confirms action and the event is removed

**Post-Conditions**

Event is removed and is no longer visible by any user of the system. Event hosts will be notified that their event has been removed.

**Alternative flows**

A. At <User Authentication> , if the user entered incorrect login details
    a. The system displays a message saying “the username/password was not correct please try again”
    b. Return to <User Authentication>
B. At <Confirm Action>, if the administrator selects No
    a. Selected event will not be removed
    b. Use case ends

## A.12 – Approve Event

**Use Case**

Approve Event

**Description**

Approve an event from an un-approved poster

**Actors**

Administrator and Club Administrator

**Pre-Conditions**

A user must have at least club administrator access

**Main flow**

1. User opens the web application
2. <User Authentication> User logs in as an administrator or club administrator
3. System redirects back to main application with an administrative interface
4. User selects option to view list of unapproved events
5. System displays list of all unapproved events
6. User selects an event from the list to view details of the event
7. User selects the option of approve event
8. <Confirm Action> System prompts user to confirm approval of event

**Post-Conditions**

Event is approved and visible to any user of the system.

**Alternative flows**

A. <At view event> User has the option of removing event instead of approving it.
    a. Use case ends

## A.13 – Log in

**Use Case Name**

Log in

**Description**

A user logs into their account on the web application.

**Actors**

Student, Privileged User, Club Administrator or Administrator

**Pre-Conditions**

The user must have an account in the system.

**Main flow**

1. The user opens the web application
2. The enters the email and password associated with their account.
3. <User Authentication> The user is then logged into the application.

**Post-Conditions**

The user is redirected to the home page.

**Alternative flows**

A. At <User Authentication>, if the user entered incorrect login details
    a. The system displays a message saying “the username/password was not correct please try again”

    b. Return to <User Authentication>
B. At <User Authentication>, if the user entered email does not have an account associated with it:
    a. The system redirects the user to the register page.

## A.14 – Register User

**Use Case Name**

Register User

**Description**

Create a new account on the system to log in with

**Actors**

Anonymous users

**Pre-Conditions**

User must have a UVic email address which they can verify (via a unique link sent by email)

**Main flow**

1. User opens the web application
2. User navigates to the register page
3. <Enter Information> User enters required registration information:
    a. Email address
    b. Desired password
    c. Date of birth
4. <Submit information> User submits the registration information, system checks that entered data is valid
5. <Create Account> System creates user in database with provided information and hashed password, account is not marked as enabled
6. System sends confirmation email to provided address with unique link
7. Users visits unique link
8. System marks user as enabled
9. End of use case

**Post-Conditions**

User is able to log in with chosen email address and password

**Alternative flows**

A. At <Submit Information> if email address does not have uvic.ca as a domain
    a. The system displays a message prompting the user to enter a uvic.ca email
    b. Return to <Enter Information>
B. At <Submit Information> if email is in database as an existing user
    a. If the existing user is marked as enabled:

i. The system displays a message notifying the user the specified email is in use. System prompts them to enter new email address or visit the login page to reset their password.
ii. Return to <Enter Information>

b. If the existing user is not marked as enabled:
i. Overwrite existing data with new data
ii. Proceed to <Create Account>

C. At <Submit Information> if password is less than 8 characters in length
a. Display message prompting the user to choose a password which is 8 characters or longer.
b. Return to <Enter Information>.

## A.15 – Report Event

**Use Case Name**

Report Event

**Description**

Create a new account on the system to log in with

**Actors**

Authenticated User, Privileged User, Club Administrator, System Administrator

**Pre-Conditions**

The user must have successfully logged in

**Main flow**

1. User views an event which they believe to be inappropriate
2. <Report Event> User chooses the “Report Event” option
3. <Choose Reason> User selects the reason why they have reported the event
4. The system notifies the user their report has been submitted, and an administrator will review the event
5. The system notifies the administrator that an event was reported
6. <Review Event> The administrator reviews the event
7. The event is removed
8. End of use case

**Post-Conditions**

TBD

**Alternative flows**

A. At <Report Event> if the event has already been approved by an administrator
a. If the event has been edited since the last approval:

        i. Proceed to <Choose Reason>
    b. If the event has not been edited since the last approval:
        i. The system displays a message notifying the user the event has already been approved, and they cannot report it.
        ii. End of use case.

B. At <Review Event> if the administrator determines the event is appropriate:
    a. The event is not removed.
    b. The event can no longer be reported (unless edited).
    c. End of use case.

## A.16 – Reset Password

**Use Case Name**

Reset Password

**Description**

Set a new password for a user account

**Actors**

Anonymous User, Authenticated User, Privileged User, Club Administrator, System Administrator

**Pre-Conditions**

The user has an existing account

**Main flow**

1. User navigates to log in page
2. User chooses the “Reset Password” option
3. <Enter Email> User enters their email address
4. The system notifies the user that an email will be sent to that email if an associated account exists
5. <Send Email> The system sends a unique link to the email address
6. The user visits the link
7. <Choose Password> The user enters a new password
8. <Update Password> The system updates the user account with the new password
9. The user is notified their password has been reset
10. End of use case

**Post-Conditions**

The user is able to log in with their new password

**Alternative flows**

A. At <Enter Email> if the email does not use uvic.ca as the domain:
    a. The user is notified that only uvic.ca emails are accepted

b. Return to <Enter Email>

B. At <Send Email> if the email address is not associated with an existing account:
    a. Do not send an email or notify the user.
    b. End of use case.
C. At <Update Password>, if the password is less than 8 characters in length
    a. Display message prompting the user to choose a password which is 8 characters or longer.
    b. Return to <Choose Password>.

## A.17 - Flag Event

**Use Case**

Flag Event

**Description**

Any user can flag an event when they select an event, if the user does flag an event the system shouldn't allow them to flag again in the same session. Flagging an event sends that event to the Administrators queue for review.

## A.18 – View over 18 event

**Use Case**

View over 18 event

**Description**

When a user isn't signed in they can't view 18+ events, if they are logged in any user that is old enough can filter events in a similar way to searching. These events may typically involve alcohol, this feature is included for legal reasons.

## A.19 – View event

**Use Case**

View event

**Description**

After any user searches, or if they are on their main calendar view they have the chance to select an event. After an event is selected the system responds with the information. The event has a date and time as well as a description and various other fields; when on this page the user can decide to perform more actions depending on their access level.

# Appendix B – Issues List

TBD