**MVC architecture description**

Events - Model

The purpose is the handling of data when users put in information regarding a particular event. The event class will handle all validation necessary to ensure an event is valid i.e. event date cannot be prior to today’s date.

User - Model

The purpose is handling of data the user has put in. The user class manages the data, logic and rules of the application to ensure data integrity for all users i.e. usernames must be unique.

Club - Model

The purpose is handling data relating to clubs. It controls the integrity of the data related to a club i.e. every club is required to have an administrator.

Events\_controller - Controller

Purpose is to be able to create, read, edit and delete certain events. It will have a destroy function where an admin may remove the event from the listing.

Users\_controller - Controller

Purpose is to be able to create, read, edit and delete a user. It will have an update method so that the user may change information about the user to add more information.

Clubs\_controller – Controller

Purpose is to be able to create, read, edit and delete a club. It will have an add method that will allow users to be added to a specific club.

User\_new - View

Purpose is to show visually the page for creating a new user. Has simple heading and input box to put in information. There is also a "back" button that will return you to the homepage.

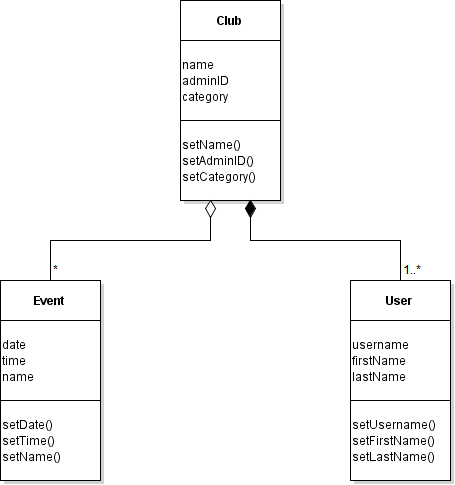
Club\_show – View

Purpose is to show a list of all the current users that are part of the club. It will have links to add/remove users from a club for admin users.

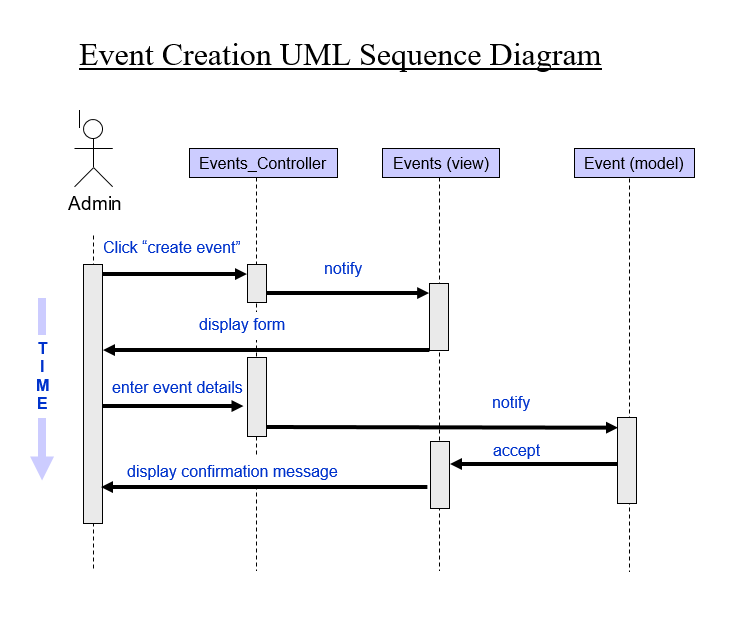
Events\_show - View

Show the view of created events in a list. Event location, name and date variables are created to store the expected values. Link to edit the events will be available for admins.

**UML Class Diagram**



**UML class diagrams and sequence diagram**



**Class descriptions**

**MODEL CLASSES:**

**Event**: Event objects will represent records in the events table. It will allow easy manipulation of the data and will enforce the integrity of any event related data.

*Fields:*

o Date: Stores the date of a particular event.

o Time: Stores the time of a particular event.

o Name: Stores the name of the event.

*Methods:*

o SetDate: Sets the event date. Validates that date entered is not less than today’s date.

o SetTime: Sets the time of the event. Validates time is a valid time.

o SetName: Sets the name of the event.

**Club**: Club objects will represent records in the clubs table. It will allow easy manipulation of the data relating to clubs.

*Fields*:

o Name: Stores the name of a particular club.

o AdminID: Stores the user ID of the administrator.

o Category: Stores the category of the club i.e. Sports Club, Fraternity etc.

*Methods*:

o SetName: Sets the name of a particular Club (duplicate names are ok).

o SetAdmin: Sets the user ID of the administrator of the club.

o SetCategory: Sets the club category.

**User:** User objects will represent records in the users table. These objects will allow for easy retrieval or manipulation of data relating to users.

*Fields*:

o Username: Stores the username of a particular user.

o FirstName: Stores the user’s first name.

o LastName: Stores the user’s last name.

*Methods*:

o SetUsername: Sets the username of a user. It ensures that the username is unique.

o SetFirst: Sets the user’s first name.

o SetLast: Sets the user’s last name.

**VIEW CLASSES:**

**Events**: Event view objects are used to display event details or to display how the user can enter data to create/modify an event.

*Fields*:

o Date: Stores the date of a particular event.

o Time: Stores the time of a particular event.

o Name: Stores the name of the event.

*Methods*:

o New: displays the form for creating a new event.

o Show: display the list of events that a user is part of.

o Edit: displays the form with data filled out for a particular event that can be edited.

**Clubs**: Club objects will be used to control how to display data regarding clubs.

*Fields*:

o Name: Stores the name of a particular club.

o AdministratorID: Stores the user ID of the administrator.

o Category: Stores the category of the club i.e. Sports Club, Fraternity etc.

*Methods*:

o New: displays the form for creating a new club.

o Show: shows all attributes regarding a specific club i.e. all the users belonging to a club.

o Edit: displays the form to edit data elements pertaining to a club.

**Users:** User objects will represent records in the users table. These objects will allow for easy retrieval or manipulation of data relating to users.

*Fields*:

o Username: Stores the username of a particular user.

o FirstName: Stores the user’s first name.

o LastName: Stores the user’s last name.

*Methods*:

o SetUsername: Sets the username of a user. It ensures that the username is unique.

o SetFirst: Sets the user’s first name.

o SetLast: Sets the user’s last name.

**CONTROLLER CLASSES:**

**Events\_Controller:** Objects of this class will control the data entry/manipulation for events.

*Fields*:

o Date: used for entering a date for an event.

o Time: used for entering a time for an event.

o Name: used for entering a name for an event.

*Methods*:

o Create: this method is used to create a new event along with all required attributes.

o Update: this method is used to update data elements for a specific event.

o Destroy: this method is used to delete an event.

**Clubs\_Controller:** Objects of this class will control the data entry/manipulation for clubs.

*Fields*:

o Name: used for entering the name of a club.

o AdministratorID: used for entering the administrator of a club.

o Category: used for entering the category for a club.

*Methods*:

o Create: this method creates a new club.

o Update: this method updates information about a club.

o Show: this method shows all users belonging to a club.

**User\_Controller:** Objects of this class will control the data entry/manipulation for users.

*Fields*:

o Username: used for entering the username of a user.

o FirstName: used for entering the first name of a user.

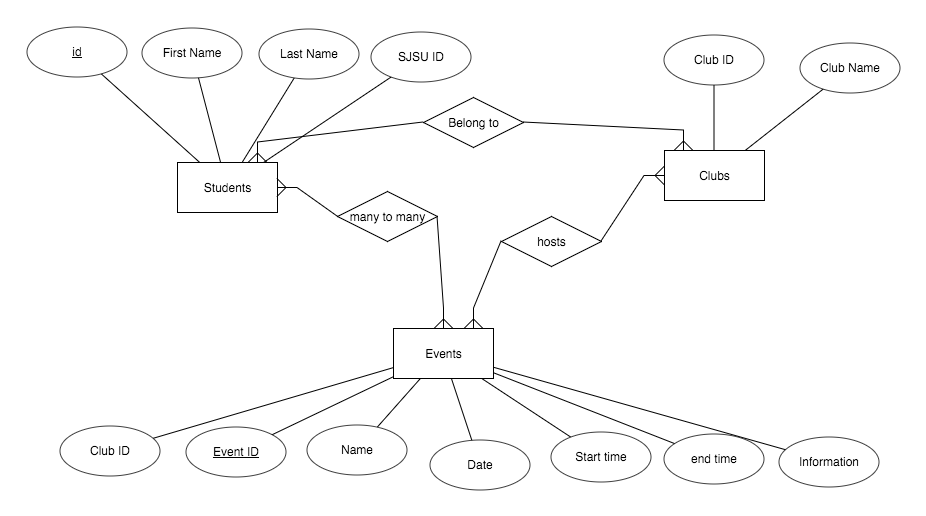
o LastName: used for entering the last name of a user.

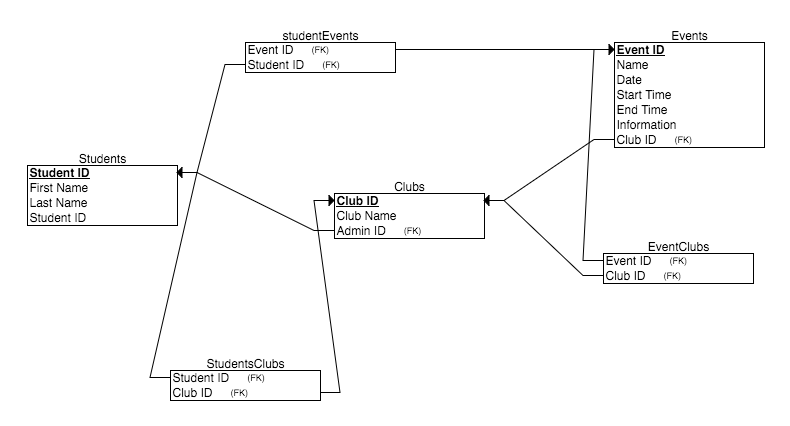
*Methods*:

o Create: this method creates a new user.

o Update: this method updates data elements for a specific user.

o Destroy: this method deletes a user.

**Conceptual data model in the form of an ER diagram** 

**Logical data model in the form of a relational schema** 

**Database table descriptions**

* **Students:** A table containing basic student student information
* **StudentsClubs:** Many students can be apart of a club, and students can join many clubs, this many to many table helps organize them
* **Clubs:** Currently Clubs need a unique key and a club name
* **Events:** Clubs will be hosting events that students can attend to, and students will be going to many events. This will allow students to be organized on a per event basis, regardless of what club they are a part of
* **EventClubs:** This many to many table is designed to assign events to clubs, but allows for clubs to join together and go to a single event together
* **StudentEvents:** Clubs will be hosting events that students can attend to, and students will be going to many events. This will allow students to be organized on a per event basis, regardless of what club they are a part in due to the eventClub relationship