

# Clubs Events

## Requirements Specification and Analysis

v.1.0

12 March 2018

### Team Members

Murat Akalın

Begüm Bilgin

Murat Yiğit

Mert Çetinkaya

Prepared for  
SE302 Software Engineering



IŞIK UNIVERSITY  
COMPUTER  
SCIENCE AND  
ENGINEERING

## Table of Contents

1.	Introduction.....	1
1.1.	Purpose of the System .....	1
1.2.	Scope of the System .....	1
1.3.	Objectives and Success Criteria of the Project .....	1
1.4.	Definitions, Acronyms, and Abbreviations .....	2
1.5.	Overview .....	2
2.	Current System.....	2
3.	Proposed System .....	2
3.1.	Overview .....	2
3.2.	Functional Requirements.....	3
3.3.	Nonfunctional Requirements .....	4
	Usability.....	4
	Reliability .....	4
	Performance.....	4
	Supportability .....	4
	Implementation.....	4
	Interface.....	5
	Packaging .....	5
	Legal.....	5
3.4.	System Models.....	5
	Scenarios.....	5
	Use case model .....	9
	Object model.....	34
	Dynamic model.....	35
	User interface—navigational paths and screen mock-ups .....	35
4.	Glossary .....	42
5.	References.....	43

## REQUIREMENTS ANALYSIS DOCUMENT [1]

### 1. Introduction

#### 1.1. Purpose of the System

Club Events is a web-based system, whose purpose is to provide an efficient and online meeting point for school clubs and students / people. Also, students/ people can follow clubs' events, clubs' information on web.

#### 1.2. Scope of the System

The system provides functionalities to students / people and school clubs. Also, the system includes a superuser( Admin) whom has extended functionalities and permissions, such as having direct access to managing records hold in the database.

As stated above, the system includes three actors which are students/people, system moderator and an admin. Since each of these system users have different functionalities, but they use same interface. Opening new parts like moderator panel with additional privileges inside.

Since all our actors share certain attributes in the database regarding their personal information ( Email, Password, etc..) they are all inherited from an abstract table named " User". All actors are an instance of " User" and they share these attributes regarding personal information. They are specialized as " Normal User", " Moderator" or "Admin" with a discriminator role column in the database. Login authentication is a common functionality for all actors. While logging in to the system, "User" table is searched if the login information is correct, and the system redirects the user to the main page.

There are two actors except admin. First actor is normal user, second actor is Moderator. Moderator is a normal user, also have an authority to adding clubs/ editing clubs/ disable clubs and adding events/editing events/ disable events. They are approved by admin at registration process.

Every user logged in to the system can view their personal information. They can't change their personal information. They are allowed to change their password at any time. If a user forgets their password and cannot authenticate, the login screen will have an option to send them an e-mail.

Normal User has limited authorities on system. Normal User can add comments on events, can show all clubs and their events. And also it can add own favorite clubs and events.

Admin has many authorities on system. Admin can add, change or delete clubs and events. Admin can display all members (*both normal users and moderators*). They can be deleted from the system by admin. Moderator is approved or rejected by admin. Admin can approve or reject any comment from the users.

### 1.3. Objectives and Success Criteria of the Project

The main arguments required for the system to be successful are:

- The system should be used by the people who is related to domain
- The system should keep protected and guarantee every user's information and their data.
- The system has implementations understandable, clear and efficient
- The general design of system in order to have fast, efficient system
- Well association between front-end and back-end
- To provide reliable, efficient and lossless data.

### 1.4. Definitions, Acronyms and Abbreviations

- DB is short version of database term
- Normal User is an actor on Clubs Events, who is default user in system
- Moderator is an actor on Clubs Events, who is approved by admin
- HTTP is a protocol for secure communication over a computer network which is widely used on the internet
- RAD: Requirements Analysis Document

### 1.5. Overview

The rest of our RAD documentation contains Current System Section, Proposed System section, overview of Clubs Events section, Functional Requirements section ( *includes high-level functionality of the system* ) , Nonfunctional Requirements section ( *includes usability, reliability, performance, supportability, implementation , interface, operational, packaging and legal requirements*), System Models section, Object Model section, Dynamic Model section and Glossary.

In System Model section , we described scenarios and models of Clubs Events. These models include use case models, object model, dynamic model and user interface view( mockup). Each of these models helps us better understand and analyze the system in different ways. Scenarios tell us the functional requirements in detail.

## 2. Current System

There is no existing web site like Clubs Events in our school. Our school is using only phone applications. Işık University has two online systems that the students and moderators in the faculty use, Campus Online and Course Online. We are accustomed to both of these systems and some of the functions in these systems will also be in Clubs Events. Our purpose is the project will integrate into the Campus Online or Course Online.

## 3. Proposed System

We are going to try to develop a social network that is completely free. This web site is user friendly , and useful for everyone. Everyone can register in web site for free. The point is creating a communication between the school' clubs and students and everyone. You do not have to study at the same school. You can still register web site. We will try give all clubs information and clubs events information to visitors and registered users.

### 3.1 Overview

Clubs Events is a online platform for communication between the Işık University clubs and students and everyone. The system also has an admin, whom manages the system and has full authentication to the system. The admin also sets who will be moderator and who will add comment on web site. The next section( Functional Requirements) contain a list of all the functions about what any actor will be able to do in the system . Nonfunctional Requirements describes the systems behavior, which will affect the system success criteria and performance , and includes system , interface and implementation details. Basically, a functional requirement describes what our system should do , while non-functional requirements place constraints on how our system will do so.

### 3.2 Functional Requirements

This section contains what each of our actors in the system must be able to do in the system.

Functional Requirements describe the high-level functionality of the system. They are the requirements the system must have, and corresponding actors are able to perform. Each actor has different functional requirements, and there are some common requirements for all actors. These common functionalities are associated with account-related functionalities.

#### **Functional Requirements shared by all actors:**

- All actors in the system can see all clubs' information and clubs' events and details.

#### **Functional Requirements for the Normal User:**

- Normal User shall be able to see all clubs and events
- Normal User shall be able to comment clubs' events.
- Normal User must have a password recovery option in case they forget their password. And are unable to authenticate, with their password reset and sent to them via e-mail.

#### **Functional Requirements for the Moderator:**

- Moderator shall be able to can access Moderator Panel.
- Moderator can add comment clubs' events.
- Moderator can add club on web site.
- Moderator can edit club on web site
- Moderator can disable club on web site
- Moderator can create event on web site
- Moderator can edit event on web site
- Moderator can disable event on web site

#### **Functional Requirements for the Admin:**

- Admin shall be able to login
- Admin shall be able to create a new admin

- Admin shall be able to access Moderator Panel and Django Admin Panel.
- Admin shall be able to display personal information and account information of Clubs, Events and all users
- Admin shall be able to add, edit or delete clubs information from the system.
- Admin shall be able to add, edit or delete event information from the system.
- Admin shall be able to add, edit or delete user (both Normal User and Moderator) information from the system.
- Admin shall be able to delete comments.
- Admin can add comments clubs' events.
- Admin can check events and clubs status.

### **3.3 Nonfunctional Requirements**

This section describes user-level requirements those are not directly related to functionality.

#### **3.3.1 Usability**

Clubs & Events will include just one interface. Every users use same interface. This interface will include all the functions an actor must perform. For usability, this interface should be easy to use and understand, with the actor being able to perform their tasks without explanation. They should also be appealing to the eye while being simple in design. Users should be able to reach their goals with minimal number of clicks.

#### **3.3.2 Reliability**

Clubs & Events shall be secure, and will not allow any unauthorized user to enter the system. Only users permitted and defined by the admin in the database shall be able to authenticate to the system, using their account information. Visitors should be able to do some stuff. Yet, registered visitors which are normal user and moderator, should be provided with unique e-mails and password that are appropriate for password criteria. The password should be hit as hashed password in database. Also visitors can register the system. We are designing the web site for everyone. All information will keep Django Framework database. . In addition, for it to be reliable, the system should not be down or crash in case of errors, and should be running 90% of the time.

#### **3.3.3 Performance**

Clubs & Events is designed to be used by multiple numbers of users. And also, should be responsive in different devices which are mobile phones, tablet and computers. The web-site is going to be dynamic content, so there should not be complicated queries in back end to not decrease performance. Responsive time should be fast, with a maximum of 4 seconds waiting time in regular site traffic. HTTP GET or POST request may be used to receive information and feedback, and task completions of users without reloading the page.

#### **3.3.4 Supportability**

The system should be managed by admin. The system should be reachable over any browser (maybe with the exclusion of older versions of internet explorer) in a standard computer. Management and maintenance of Clubs & Events belongs to the system administrator. Maintenance of Clubs & Events should be able to be done easily, and in certain time

intervals. Interfaces for database altering, tables and instances should be shown in administrator panel.

### **3.3.5 Implementation**

The system will be implemented on Django Frameworks. Python and HTML will be used as the programming language and Pycharm will be used as a framework in the process. . We will be using a “code-first” approach, which means we will construct the Models first and the virtual relations between them, and the database will be created based on these models. Model-view-controller design methodology is going to be implemented in this project. We will check input over controller, our model is satisfied object relation model over SQL.HTML5, CSS and Boostsrp should be used to implement. User Interface should be web-based (accessible via WWW Browser).

### **3.3.6 Interface**

The interface should be easy to use for all actors, as usability is the main concern. There should be no guide to instruct the actor about what and how to do their tasks. The interface shouldn't be too colorful and eye-tiring, instead a simple color scheme should be used. The colors of school might be used. Menu navigation should change according to actor. There will be a side menu navigation in our interfaces. Similar tasks and functions should be put together.

### **3.3.7 Packaging**

Admin should install the system. Also, the system is a web site so the site should uploaded on the server. There should no time constraints on the installation. We are planning for the project to be used by the school clubs and SKS(Department of Health, Culture and Sports), with the project fulfilling the functional requirements of entering clubs information , events information, but other , but other constraints on the actual delivery of the system are not determined. Thus, packaging requirements will be decided in the future.

### **3.3.8 Legal**

The Clubs & Events does not use any license or licensed stuff. It is a social network. The software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and no infringement.

## **3.4 System Models**

### **3.4.1 Scenarios**

A scenario is an instance of a use case. These scenarios will describe the use case instances in detail, and involves a situation where an actor takes a given functionality or path to resolve the scenario.

Obtained actor instances are:

- I. Visitor: Ali
- II. User: Murat
- III. Moderator: Mustafa
- IV. Admin: Hurşit

### Scenario 1:

<i>Scenario Name:</i> Make Comment
<i>Participant actor instances:</i> <u>Murat: User, Mustafa: Moderator, Hurşit: Admin</u>
<i>Flow of events:</i> <ol style="list-style-type: none"><li>1. Murat logs in as user.</li><li>2. He sees the event flow at the home page, every event has a button to make comment and she clicks on it.</li><li>3. After clicking on the button she sees a new page with a text input and writes her own thoughts to comment.</li><li>4. He clicks on the comment button.</li><li>5. System returns him to the home page if the comment is added, if not, no response returns.</li></ol>

### Scenario 2:

<i>Scenario Name:</i> Show Clubs
<i>Participant actor instances:</i> <u>Murat: User, Hurşit: Admin, Mustafa: Moderator, Ali: Visitor</u>
<i>Flow of events:</i> <ol style="list-style-type: none"><li>1. Murat logs in as user.</li><li>2. He sees the Clubs at the home page and clicks on it.</li><li>3. After he clicks on the Clubs button he displays all the clubs.</li></ol>

### Scenario 3:

<i>Scenario Name:</i> Add Club
<i>Participant actor instances:</i> <u>Mustafa: Moderator, Hurşit: Admin</u>
<i>Flow of events:</i> <ol style="list-style-type: none"><li>1. Mustafa logs in as user.</li><li>2. He sees a navigation bar at the home page and clicks on Moderator Panel at the navigation bar.</li><li>3. After he clicks on the Moderator Panel button he displays a page where he can Add/Edit Clubs and Add/Edit Events.</li><li>4. He clicks on Add Club button.</li><li>5. He enters a Club Name and a Club Info and adds a Club emblem, then he clicks on the Save button.</li><li>6. System returns him to the Moderator Panel page.</li></ol>

### Scenario 4:

<i>Scenario Name:</i> Edit Club
<i>Participant actor instances:</i> <u>Mustafa: Moderator, Hurşit: Admin</u>
<i>Flow of events:</i> <ol style="list-style-type: none"><li>1. Mustafa logs in as user.</li><li>2. He sees a navigator bar at the home page and clicks on Moderator Panel at the navigator bar.</li><li>3. After he clicks on the Moderator Panel button he displays a page where</li></ol>



- he can Add/Edit Clubs and Add/Edit Events.
- 4. He clicks on Edit Club button.
- 5. After he clicks on the Edit Club button, he displays all the clubs with a button to edit the clubs and clicks on the button.
- 6. He displays the information of the Club he has chosen and changes Club Info or the Club Emblem and clicks on the Save button.
- 7. System returns him to the Moderator Panel page.

#### Scenario 5:

Scenario name: Login
Participating Actor : Murat : User, Mustafa: Moderator, Hurşit: Admin
<p>Flow of Events:</p> <ol style="list-style-type: none"> <li>1.) Murat, Mustafa or Hurşit is at main page. They clicks to login button and directed at login page. At that page, he enters username/e-mail and password in the area and clicks on the login button.</li> <li>2.) They entered his username/email true and he logins to the system and redirected to main page.</li> </ol>
<p>Extentions:</p> <p>2a.) If Murat, Mustafa or Hurşit leaves any area blank, or enters his password or email/username wrong, he will see a warning message like “email or password is wrong”.</p>

#### Scenario 6:

Scenario name: Log-out
Participating Actor : Murat : User, Mustafa: Moderator, Hurşit: Admin
<p>Flow of Events:</p> <ol style="list-style-type: none"> <li>1.) Murat, Mustafa or Hurşit clicks the log-out button, exits his account.</li> <li>2.) They will direct to the main page as logged out</li> </ol>

#### Scenario 7:

Scenario name: Register
Participant actor instances: Ali : Visitor
<p>Flow of events:</p> <ol style="list-style-type: none"> <li>1.) Ali is at main page. He clicks register button and he is directed to theregister page.</li> <li>2.) He enters information of his name/username and enters password twice for checking. Clicks save button.</li> <li>3.) His registiration complited</li> </ol>

Extentions:

2a) If he enters an allready used username/email, he will see an error message.

2b) If he enters a password that is shorter than 8 characters or fully numeric or similar to the username/email, he will see an error message.

### Scenario 8:

Scenario name: Add event
Participant actor instances: Hurşit : Admin, Mustafa: Moderator
<p>Flow of events:</p> <ol style="list-style-type: none"><li>1.) Hurşit or Mustafa is at main page. After he logins, he clicks the “moderator panel” button at the right side of the navigation bar.</li><li>2.) He clicks the “adding event” button.</li><li>3.) At the page that opened, he selects club which will have the new event. Then, he enters the event name, created date, day, location and description.</li><li>4.) He clicks the “save” button. New event for a selected club is created.</li></ol>
<p>Extentions:</p> <p>4a) If he leave any area blank, he will see an error message.</p>

### Scenario 9:

Scenario name: Edit event
Participant actor instances: Hurşit : Admin, Mustafa: Moderator
<p>Flow of events:</p> <ol style="list-style-type: none"><li>1.) Hurşit or Mustafa is at main page. After he logins, he clicks the “moderator panel” button at the right side of the navigation bar.</li><li>2.) He clicks the “editing event” button under the “events” title.</li><li>3.) At the page that opened he can see all events so he selects event which he wants to edit. He can change event name, created date, day, location and information at that page.</li><li>4.) He clicks the “save” button. The event is edited.</li></ol>
<p>Extentions:</p> <p>4a) If he leaves any area blank, he will see an error message.</p>

### Scenario 10:

Scenario name: Delete Club
Participant actor instances: Admin : Hurşit
<p>Flow of events:</p> <ol style="list-style-type: none"><li>4.) Hurşit is at main page. After he logins, he types “/admin” as following to the address of the page so he goes to the django admin panel.</li><li>5.) He clicks “clubs” button.</li><li>6.) He selects clubs that he wants to delete.</li><li>7.) He clicks the action bar and selects the delete action.</li><li>8.) He clicks “done” button.</li><li>9.) He deleted selected clubs.</li></ol>
<p>Extentions:</p> <p>5a.) If he clicks done button without doing nothing, an error message will occur.</p>

### Scenario 11:

Scenario name: Delete event
Participant actor instances: Admin : Hurşit
<p>Flow of events:</p> <ol style="list-style-type: none"><li>1.) Hurşit is at main page. After he logins, he types “/admin” as following to the address of the page so he goes to the django admin panel.</li><li>2.) He clicks “events” button.</li><li>3.) He selects events that he wants to delete.</li><li>4.) He clicks the action bar and selects the delete action.</li><li>5.) He clicks “done” button.</li><li>6.) He deleted selected events.</li></ol>
<p>Extentions:</p> <p>5a.) If he clicks done button without doing nothing, an error message will occur.</p>

### Scenario 12:

Scenario name: Delete comment
Participant actor instances: Admin : Hurşit
<p>Flow of events:</p> <ol style="list-style-type: none"> <li>1.) Hurşit is at main page. After he logins, he types “/admin” as following to the address of the page so he goes to the django admin panel.</li> <li>2.) He clicks “comments” button.</li> <li>3.) He selects comments that he wants to delete.</li> <li>4.) He clicks the action bar and selects the delete action.</li> <li>5.) He clicks “done” button.</li> <li>6.) He deleted selected comments.</li> </ol>

### Scenario 13:

<i>Scenario Name:</i> Deactivate Comment
<i>Participant actor instances:</i> <u>Mustafa: Moderator, Hurşit: Admin</u>
<p><i>Flow of events:</i></p> <ol style="list-style-type: none"> <li>1. Mustafa logs in as user.</li> <li>2. He sees the event flow at the home page, every event has a button to make comment and he clicks on it.</li> <li>3. After clicking on the button he sees a new page with a text input and a status bar with accepted, waiting and rejected properties, chooses rejected status and clicks on the Save button.</li> </ol>

### Scenario 14:

<i>Scenario Name:</i> Deactivate Club
<i>Participant actor instances:</i> <u>Mustafa: Moderator, Hurşit: Admin</u>
<p><i>Flow of events:</i></p> <ol style="list-style-type: none"> <li>1. Mustafa logs in as moderator.</li> <li>2. He sees a navigation bar at the home page and clicks on Moderator Panel at the navigation bar.</li> <li>3. After he clicks on the Moderator Panel button he displays a page where he can Add/Edit Clubs and Add/Edit Events.</li> <li>4. He clicks on Edit Club button.</li> <li>5. After he clicks on the Edit Club button, he displays all the clubs with a button to edit the clubs and clicks on the button.</li> </ol>

6. He displays the information of the Club he has chosen and below all the things he can change, he sees a dropdown menu with published & draw properties, chooses draw and clicks on the Save button.
7. System returns him to the Moderator Panel page.

#### Scenario 15:

<i>Scenario Name:</i> Deactivate Event
<i>Participant actor instances:</i> <u>Mustafa: Moderator, Hurşit: Admin</u>
<i>Flow of events:</i> <ol style="list-style-type: none"> <li>1. Mustafa logs in as user.</li> <li>2. He sees a navigation bar at the home page and clicks on Moderator Panel at the navigation bar.</li> <li>3. After he clicks on the Moderator Panel button he displays a page where he can Add/Edit Clubs and Add/Edit Events.</li> <li>4. He clicks on Edit Event button.</li> <li>5. At the page that opened he sees all events and selects the event which he wants to edit.</li> <li>6. He displays the event name, created date, day, location and information at that page and he sees a dropdown menu with published &amp; draw properties, chooses draw and clicks on the Save button.</li> <li>7. System returns him to the Moderator Panel page.</li> </ol>

#### Scenario 16:

Scenario name: Delete User
Participant actor instances: Admin : Hurşit
Flow of events: <ol style="list-style-type: none"> <li>1.) Hurşit is at main page. After he logins, he types “/admin” as following to the address of the page so he goes to the django admin panel.</li> <li>2.) He clicks “Users” button.</li> <li>3.) He selects users that he wants to delete.</li> <li>4.) He clicks the action bar and selects the delete action.</li> <li>5.) He clicks “done” button.</li> <li>6.) He deleted selected users.</li> </ol>
Extentions: <p>5a.) If he clicks done button without doing nothing, an error message will occur.</p>

#### Scenario 17:

Scenario name: Accept comment
-------------------------------

Participant actor instances: Admin : Hurşit
<p>Flow of events:</p> <ol style="list-style-type: none"> <li>1.) Hurşit is at main page. After he logins, he types “/admin” as following to the address of the page so he goes to the django admin panel.</li> <li>2.) He clicks “comments” button.</li> <li>3.) He selects comments that he wants to accept.</li> <li>4.) He clicks the action bar and selects the accept action.</li> <li>5.) He clicks “done” button.</li> <li>6.) He accepted selected comments.</li> </ol>
<p>Extentions:</p> <p>5a.) If he clicks done button without doing nothing, an error message will occur.</p>

### Scenario 18:

<i>Scenario Name:</i> Update a User to Moderator
<i>Participant actor instances:</i> <u>Hurşit: Admin</u>
<p><i>Flow of events:</i></p> <ol style="list-style-type: none"> <li>1. Murat logs in as admin.</li> <li>2. He sees user authentication and authorization options at the page he arrived by logging in.</li> <li>3. He clicks on the Change button near Users.</li> <li>4. He clicks on the user he wants to update as a moderator.</li> <li>5. He picks the “Staff Status” at the “Permissions” section and clicks on the Save Button.</li> <li>6. System returns him to the User page.</li> </ol>

### Scenario 19:

<i>Scenario Name:</i> System Installation
<i>Participant actor instances:</i> <u>Hurşit: Admin</u>
<p><i>Flow of events:</i></p> <ol style="list-style-type: none"> <li>1. Hurşit logs in as admin.</li> <li>2. He will installation the system on the web.</li> <li>3. Everyone can reach on www. Browser.</li> </ol>

## Use Case Scenarios

### Use Case 1:

Use case name: Login
Participant actors: Initiated by User : Murat, Admin : Hurşit, Moderator : Mustafa
<p>Flow of events:</p> <ol style="list-style-type: none"> <li>1. User, Moderator or Admin clicks to the login button.</li> <li>2. Club Events System displays the login screen of the website.</li> <li>3. User, Moderator or Admin enters username/e-mail and password to the login fields.</li> <li>4. User, Moderator or Admin clicks "Login" button.</li> <li>5. If the username and password match with the username and password which is stored in database. User, Moderator or Admin authenticates.</li> <li>6. Club Events System fetches the dashboard of student.</li> </ol>
Entry Condition: User visits the website.
Exit Condition: User, Moderator or Admin clicks "Login" button and entered information should be correct.
<p>Quality Requirements:</p> <p>If User, Moderator or Admin leaves one or more fields empty, system displays a warning message, like " Please fill this area."</p> <p>If the information's checked from database are not true, system displays a warning message, like " Please enter a correct username and password. Note that both fields may be case-sensitive."</p>

## Use Case 2:

Use case name: Logout
Participant actors: Initiated by User: Murat, Admin: Hurşit , Moderator: Mustafa
<p>Flow of events:</p> <ol style="list-style-type: none"> <li>1.) User, Moderator or Admin clicks the "Logout" button.</li> <li>2.) System redirects User, Moderator or Admin to the main page.</li> </ol>
Entry Condition: User, Moderator or Admin logs in to the system.
Exit Condition: : Admin clicks "logout".
Quality Requirements: None

### Use Case 3:

Use case name: Register
Participant actors: Initiated by Visitor: Ali
Flow of events: <ol style="list-style-type: none"><li>1.) Visitor is at login page. Visitor has not any account for Club Events System.</li><li>2.) Visitor clicks register label, enters information of username/e-mail and password. Visitor. clicks register button.</li><li>3.) He registers.</li></ol>
Entry Condition: Visitor visits the website.
Exit Condition: Visitor clicks save button.
Quality Requirements: Visitor can't have a name over 150 characters. If visitor doesn't fill the confirmation password area, system displays a warning message, like " Please fill this area." If visitor enters a pasword that is shorter than 8 characters, system will display a warning message like "This password is too short. It must contain at least 8 characters." If visitor enters a common password, system will display a warning message like "This password is too common." If visitor enters a password that is fully numeric, system will display a warning message like "This password is entirely numeric." If user enters a password that is similar to the username/e-mail, system will display a warning message like "Your password can't be too similar to your other personal information."

### Use Case 4:

Use case name: Edit event
Participant actors: Initiated by Admin: Hurşit, Moderator: Mustafa



<p>Flow of events:</p> <ol style="list-style-type: none"> <li>1.) Admin or Moderator is at main page.</li> <li>2.) Admin or Moderator clicks the moderator panel button</li> <li>3.) System redirects Admin or Moderator to the moderator page.</li> <li>4.) Admin or Moderator clicks editing event button</li> <li>5.) System shows all the events and he clicks the pencil button near the event that he wants to edit.</li> <li>6.) Admin or Moderator edits the filed club, event name, created date, day, location and information areas.</li> </ol>
Entry Condition: Admin or Moderator logins to the system.
Exit Condition: Admin or Moderator clicks save button.
Quality Requirements: If Admin or Moderator leaves one or more fields empty, system displays a warning message, like " Please fill this area".

## Use Case 5:

Use case name: Adding an event
Participant actors: Initiated by Admin: Hurşit, Moderator: Mustafa
<p>Flow of events:</p> <ol style="list-style-type: none"> <li>1.) Admin or Moderator is at main page.</li> <li>2.) Admin or Moderator clicks the moderator panel button.</li> <li>3.) System redirects Admin or Moderator to the moderator page.</li> <li>4.) Admin or Moderator clicks adding event button</li> <li>5.) Admin or Moderator fills the club, event name, created date, day, location and information areas</li> </ol>
Entry Condition: Admin or Moderator logins to the system.
Exit Condition: Admin or Moderator clicks save button.

Quality Requirements: If Admin or Moderator leaves one or more fields empty, system displays a warning message, like " Please fill this area".

#### Use Case 6:

<b>Use case name:</b> Make Comment
<b>Participant actors:</b> Initiated by User, Admin, Moderator
<b>Flow of events:</b> <ol style="list-style-type: none"><li>1. User clicks on the comment button.</li><li>2. System displays a new page with a text input.</li><li>3. User and writes her/his thoughts to comment and clicks on the comment button.</li></ol>
<b>Entry Condition:</b> User should be logged in with the web site.
<b>Exit Condition:</b> System returns her/him to the home page if the comment is added, if not, no response returns.
<b>Quality Requirements:</b> None.

### Use Case 7:

<b>Use case name:</b> Show Clubs
<b>Participant actors:</b> Initiated by User, Admin, Moderator, Visitor
<b>Flow of events:</b> 1. User clicks on Clubs at the navigation bar.
<b>Entry Condition:</b> None
<b>Exit Condition:</b> System displays a new page which contains the Clubs' information.
<b>Quality Requirements:</b> None.

### Use Case 8:

<b>Use case name:</b> Add Club
<b>Participant actors:</b> Initiated by Moderator, Admin
<b>Flow of events:</b> 1. Moderator sees a navigation bar at the home page and clicks on Moderator Panel at the navigation bar. 2. System displays a page where the Moderator can Add/Edit Clubs and Add/Edit Events. 3. Moderator clicks on Add Club button. 4. System displays a page where the Moderator can enter Club Name, Club Info and Club emblem. 5. Moderator enters a Club Name and a Club Info and adds a Club emblem, then s/he clicks on the Save button.
<b>Entry Condition:</b> Moderator should be logged in with the web site.
<b>Exit Condition:</b> System returns her/him to the Moderator Panel page.

**Quality Requirements:**

Moderator has to fill the Club Name and Club Info areas, if not, system gives an error message saying "Please fill this area."

Moderator also has to upload a Club Emblem, if not, system gives an error message saying "Please select a file."

**Use Case 9:**

<b>Use case name:</b> Edit Club
<b>Participant actors:</b> Initiated by Moderator, Admin
<b>Flow of events:</b> <ol style="list-style-type: none"><li>1. Moderator sees a navigation bar at the home page and clicks on Moderator Panel at the navigation bar.</li><li>2. System displays a page where the Moderator can Add/Edit Clubs and Add/Edit Events.</li><li>3. Moderator clicks on Edit Club button.</li><li>4. System displays all the clubs with a button to edit the clubs and clicks on the button.</li><li>5. Moderator displays the information of the Club s/he has chosen and changes Club Info or the Club Emblem and clicks on the Save button.</li></ol>
<b>Entry Condition:</b> Moderator should be logged in with the web site.
<b>Exit Condition:</b> System returns her/him to the Moderator Panel page.
<b>Quality Requirements:</b> <p>Moderator cannot leave any areas empty as it is like in Adding Clubs.</p>

**Use Case 10:**

<b>Use case name:</b> Deactivate Club
<b>Participant actors:</b> Initiated by Moderator, Admin
<b>Flow of events:</b> Moderator sees a navigation bar at the home page and clicks on Moderator Panel at the navigation bar. System displays a page where the Moderator can Add/Edit Clubs and Add/Edit Events. Moderator clicks on Edit Club button. System displays all the clubs with a button to edit the clubs and clicks on the button. Moderator displays the information of the Club s/he has chosen and changes the status as draw and clicks on the Save button.
<b>Entry Condition:</b> Moderator should be logged in with the web site.
<b>Exit Condition:</b> System returns her/him to the Moderator Panel page.
<b>Quality Requirements:</b> None.

### Use Case 11:

<b>Use case name:</b> Deactivate Event
<b>Participant actors:</b> Initiated by Moderator, Admin
<b>Flow of events:</b> <ol style="list-style-type: none"><li>1. Moderator sees a navigation bar at the home page and clicks on Moderator Panel at the navigation bar.</li><li>2. System displays a page where the Moderator can Add/Edit Clubs and Add/Edit Events.</li><li>3. Moderator clicks on Edit Event button.</li><li>4. System displays all the events with a button to edit the events and clicks on the button.</li><li>5. Moderator displays the information of the Event s/he has chosen and changes the status as draw and clicks on the Save button.</li></ol>
<b>Entry Condition:</b> Moderator should be logged in with the web site.
<b>Exit Condition:</b> System returns her/him to the Moderator Panel page.
<b>Quality Requirements:</b> None.

### Use Case 12:

<b>Use case name:</b> Deactivate Comment
<b>Participant actors:</b> Initiated by Admin, Moderator
<b>Flow of events:</b> <ol style="list-style-type: none"> <li>1. Moderator clicks on the comment button.</li> <li>2. System displays a new page with a text input and input and a status bar with accepted, waiting and rejected properties.</li> <li>3. Moderator chooses rejected status and clicks on the Save button..</li> </ol>
<b>Entry Condition:</b> Moderator should be logged in with the web site.
<b>Exit Condition:</b> System returns her/him to the Moderator Panel page.
<b>Quality Requirements:</b> None.

### Use Case 13:

<b>Use case name:</b> Update a User to Moderator
<b>Participant actors:</b> Initiated by Admin
<b>Flow of events:</b> <ol style="list-style-type: none"> <li>1. Admin sees user authentication and authorization options at home page of the admin panel and clicks on the Change button near Users.</li> <li>2. System displays the Users page.</li> <li>3. Admin clicks on the user he wants to update as a moderator.</li> <li>4. System displays a page where admin can change all the personal information of the user selected.</li> <li>5. Admin picks the “Staff Status” at the “Permissions” section and clicks on the Save Button.</li> </ol>
<b>Entry Condition:</b> Admin should be logged in with the Django Admin Panel.
<b>Exit Condition:</b> System returns him to the User page.
<b>Quality Requirements:</b> None.

#### Use Case 14:

<b>Use case name: Delete Club</b>
<b>Participant actors:</b> Initiated by Admin: Hurşit
<b>Flow of events:</b> <ol style="list-style-type: none"><li>1.) Admin is at main page.</li><li>2.) Admin types “/admin” as following to the address of the page and clicks enter.</li><li>3.) System redirects Admin to the django admin page.</li><li>4.) Admin clicks clubs button so the system shows all the clubs.</li><li>5.) Admin selects clubs that he wants to delete.</li><li>6.) Admin selects delete action and clicks done button.</li><li>7.) System deletes selected clubs.</li></ol>
<b>Entry Condition:</b> Admin logins to the system.
<b>Exit Condition:</b> Admin clicks done button.
<b>Quality Requirements:</b> If Admin clicks done without selecting any choice, system displays a warning message.

#### Use Case 15:

<b>Use case name: Accept comment</b>
<b>Participant actors:</b> Initiated by Admin: Hurşit
<b>Flow of events:</b> <ol style="list-style-type: none"><li>1.) Admin is at main page.</li><li>2.) Admin types “/admin” as following to the address of the page and clicks enter.</li></ol>



3.) System redirects Admin to the django admin page. 4.) Admin clicks comments button so the system shows all the comments. 5.) Admin selects comments that he wants to accepts. 6.) Admin selects accept action and clicks done button. 7.) System accepts selected comments.
<b>Entry Condition:</b> Admin logins to the system.
<b>Exit Condition:</b> Admin clicks done button.
<b>Quality Requirements:</b> If Admin leaves one or more field empty, system displays a warning message.

#### Use Case 16:

<b>Use case name:</b> Reject comment
<b>Participant actors:</b> Initiated by Admin: Hurşit
<b>Flow of events:</b> <ol style="list-style-type: none"> <li>1.) Admin is at main page.</li> <li>2.) Admin types “/admin” as following to the address of the page and clicks enter.</li> <li>3.) System redirects Admin to the django admin page.</li> <li>4.) Admin clicks comments button so the system shows all the comments.</li> <li>5.) Admin selects comments that he wants to reject.</li> <li>6.) Admin selects reject action and clicks done button.</li> <li>7.) System rejects selected comments.</li> </ol>
<b>Entry Condition:</b> Admin logins to the system.
<b>Exit Condition:</b> Admin clicks done button.
<b>Quality Requirements:</b> If Admin leaves one or more field empty, system displays a warning message.

#### Use Case 17:

<b>Use case name: Deactivate club</b>
<b>Participant actors:</b> Initiated by Admin: Hurşit
<b>Flow of events:</b> <ul style="list-style-type: none"><li>1.) Admin is at main page.</li><li>2.) Admin types “/admin” as following to the address of the page and clicks enter.</li><li>3.) System redirects Admin to the django admin page.</li><li>4.) Admin clicks clubs button so the system shows all the clubs.</li><li>5.) Admin selects clubs that he wants to deactivate.</li><li>6.) Admin selects deactivate action and clicks done button.</li><li>7.) System deactivates selected clubs.</li></ul>
<b>Entry Condition:</b> Admin logs in to the system.
<b>Exit Condition:</b> Admin clicks done button.
<b>Quality Requirements:</b> If Admin leaves one or more field empty, system displays a warning message.

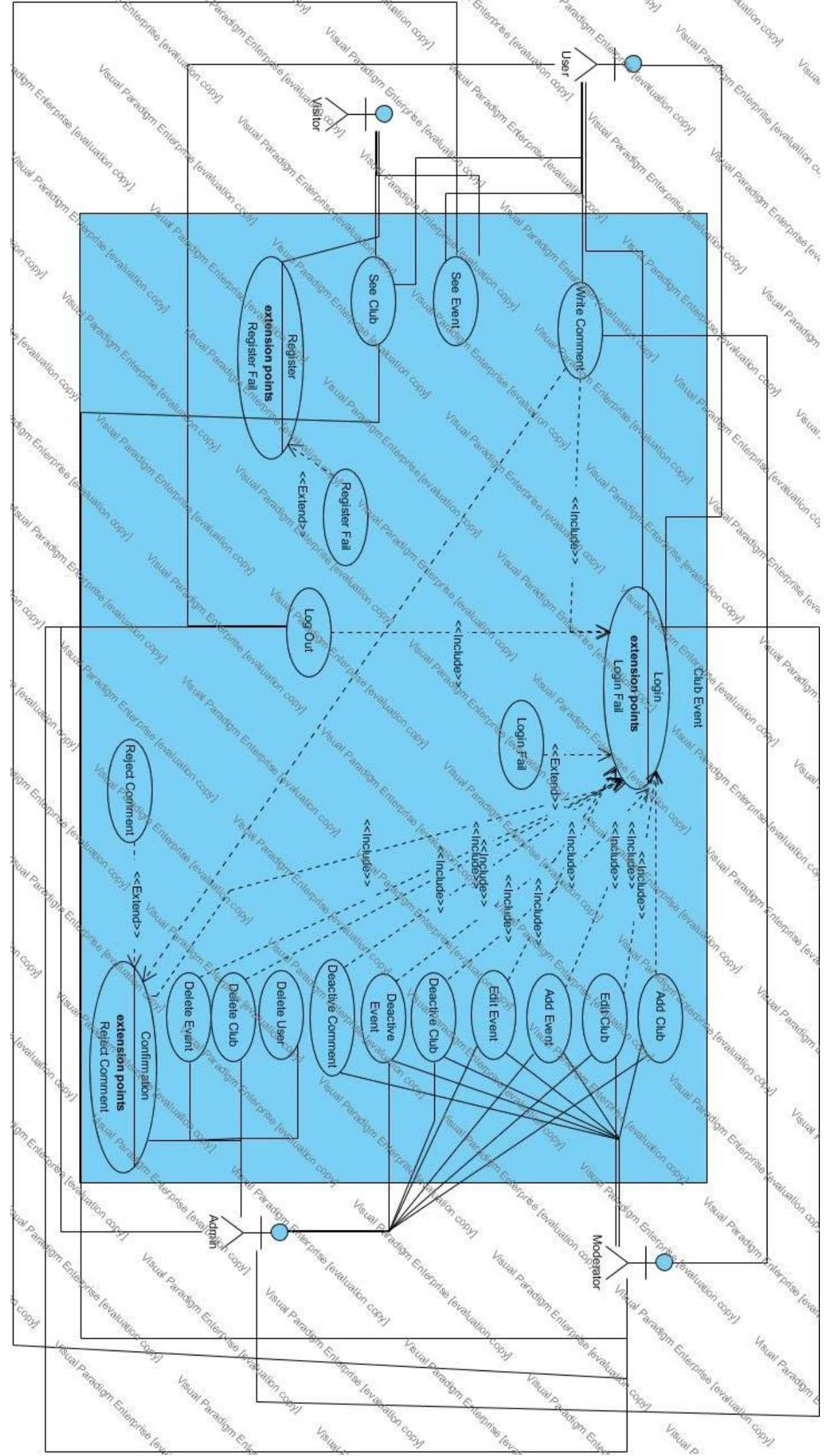
#### Use Case 18:

<b>Use case name: Deactivate event</b>
<b>Participant actors:</b> Initiated by Admin: Hurşit
<b>Flow of events:</b> <ol style="list-style-type: none"> <li>1.) Admin is at main page.</li> <li>2.) Admin types “/admin” as following to the address of the page and clicks enter.</li> <li>3.) System redirects Admin to the django admin page.</li> <li>4.) Admin clicks enets button so the system shows all the events.</li> <li>5.) Admin selects events that he wants to deactivate.</li> <li>6.) Admin selects deactivate action and clicks done button.</li> <li>7.) System deactivates selected events.</li> </ol>
<b>Entry Condition:</b> Admin logins to the system.
<b>Exit Condition:</b> Admin clicks done button.
<b>Quality Requirements:</b> If Admin leaves one or more field empty, system displays a warning message.

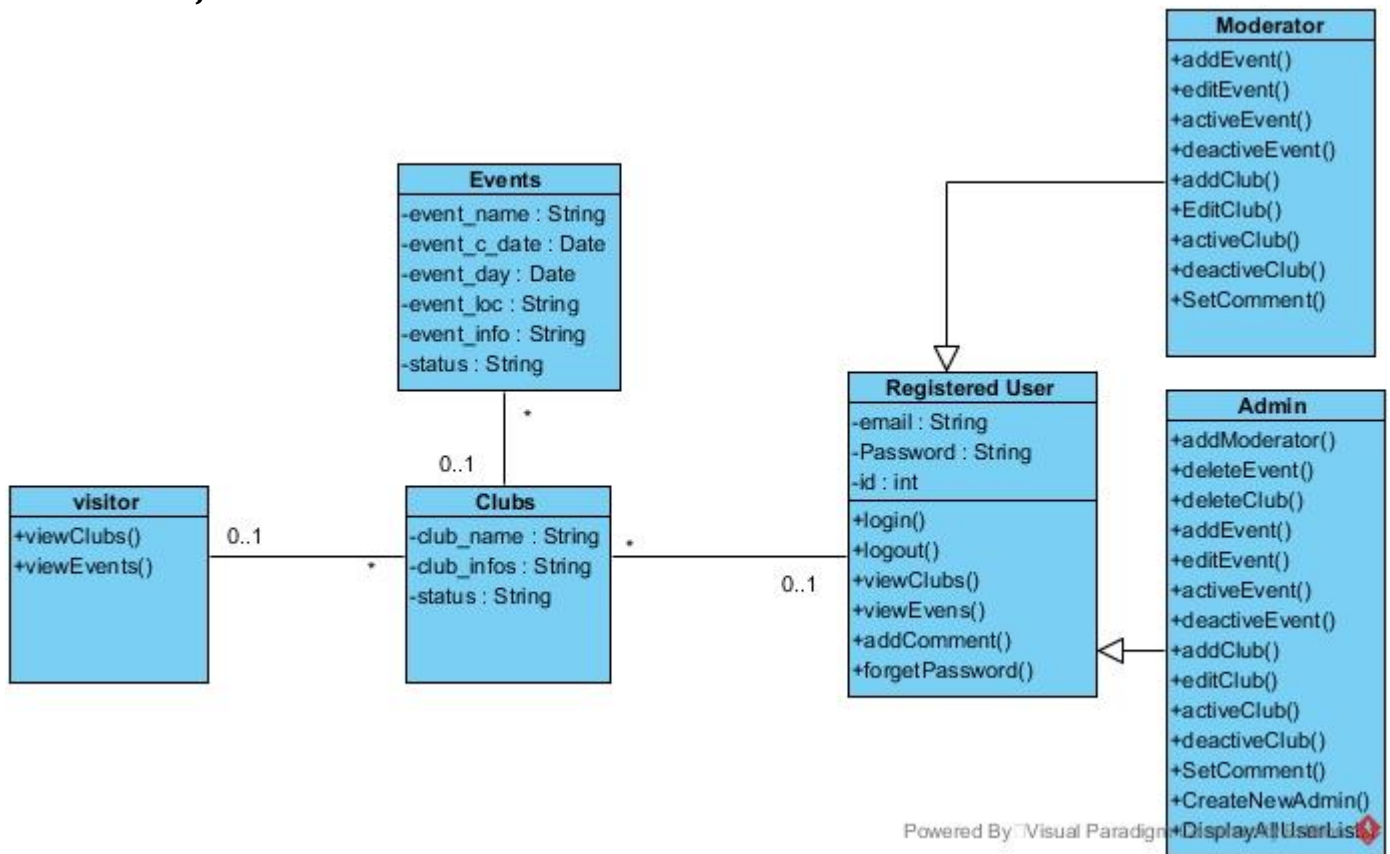
#### Use Case 19:

<b>Use case name: System Installation</b>
<b>Participant actors:</b> Initiated by Admin: Hurşit
<b>Flow of events:</b> <ol style="list-style-type: none"> <li>1.) Admin ready the system activates.</li> <li>2.) With this operation, everyone can Access on www. Browser.</li> </ol>
<b>Entry Condition:</b> Actor must be Admin
<b>Exit Condition:</b> None

### 3.4.2 Use Case Model

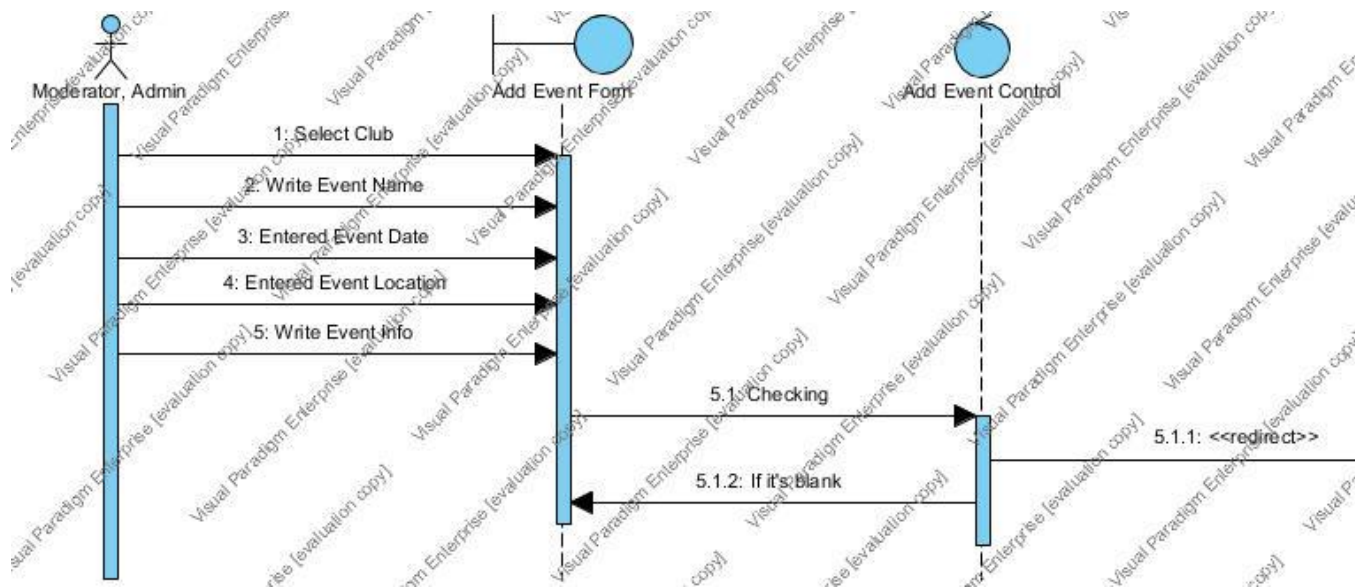


### 3.4.3 Object Model



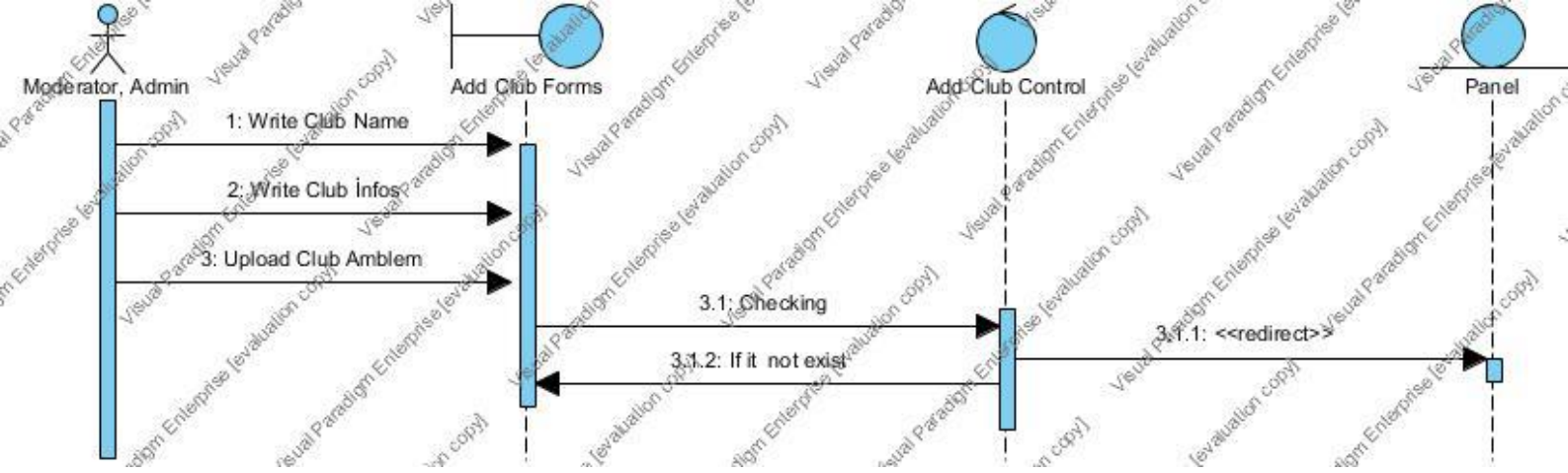
### 3.4.4 Dynamic Model

#### Add Event

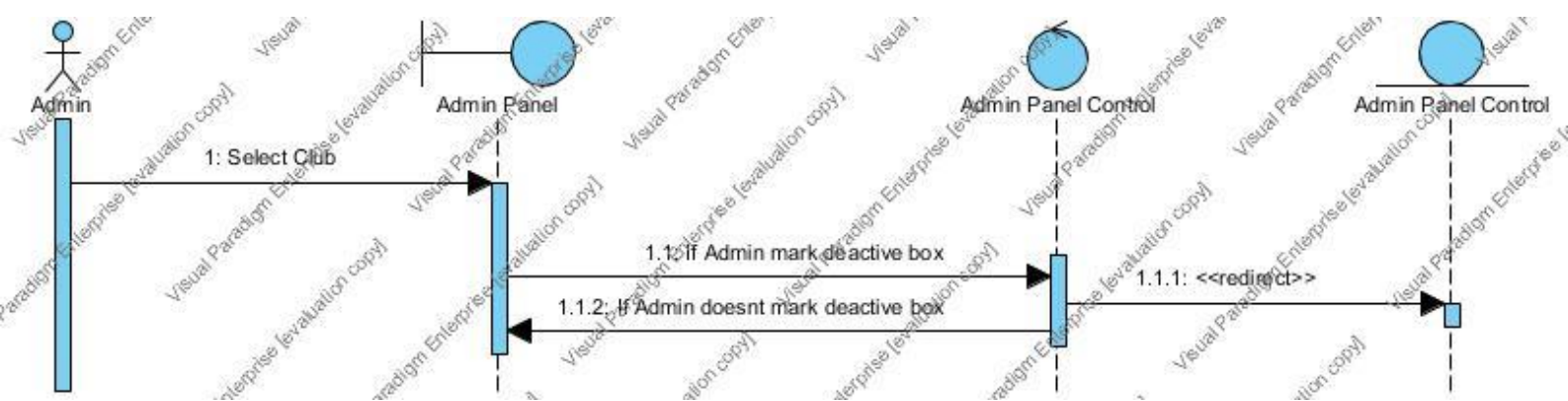


#### Add Club

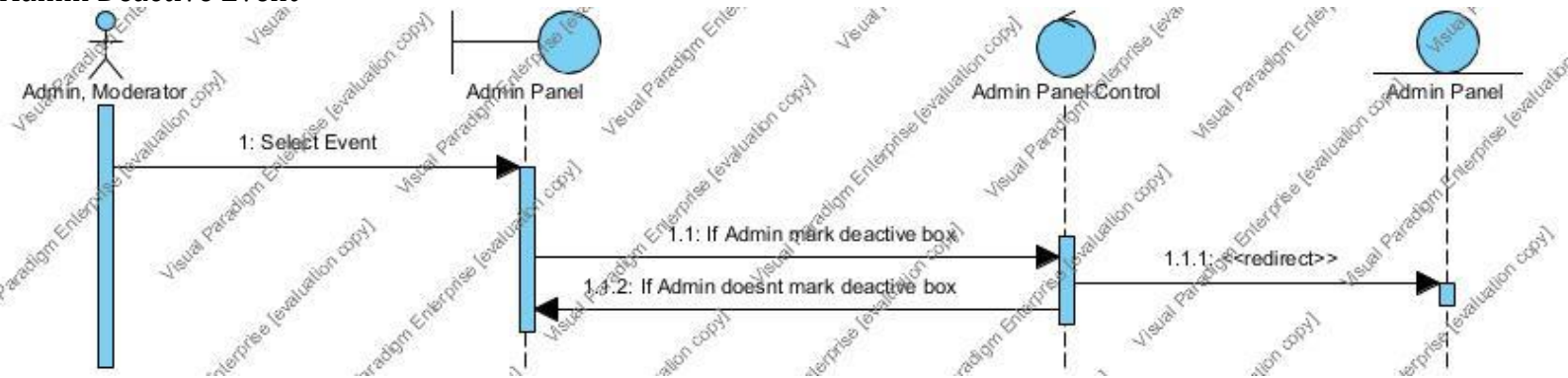




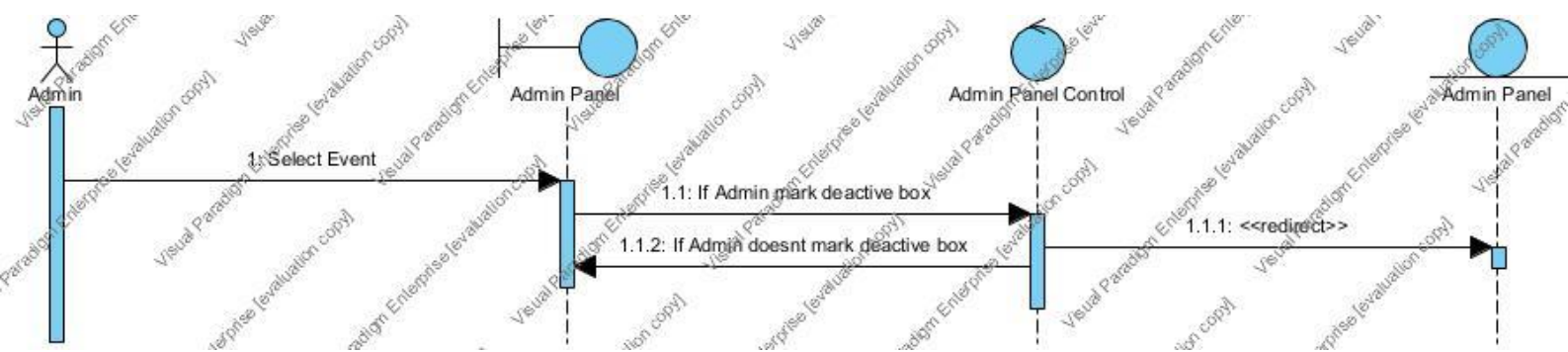
## Admin Deactive Club



## Admin Deactive Event



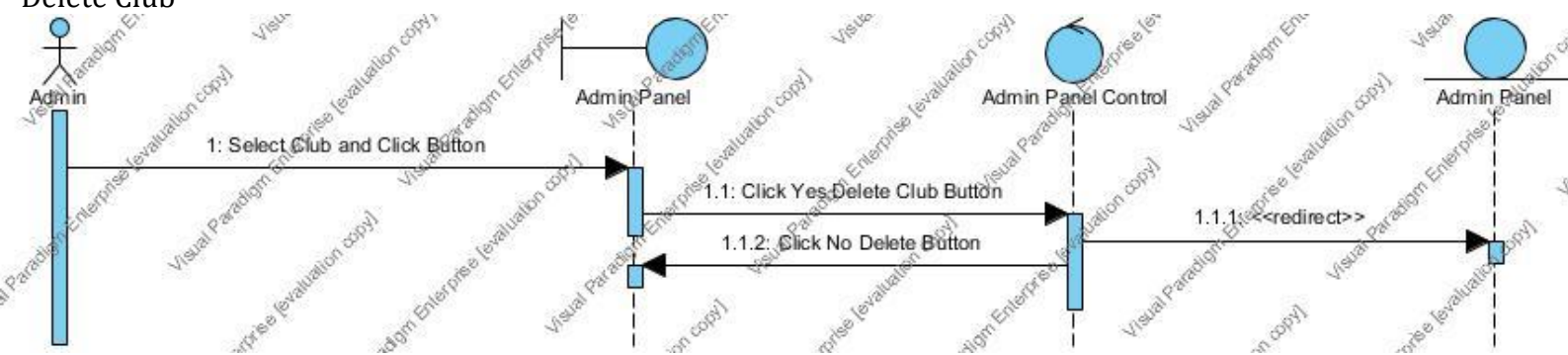
## Admin Delete Event



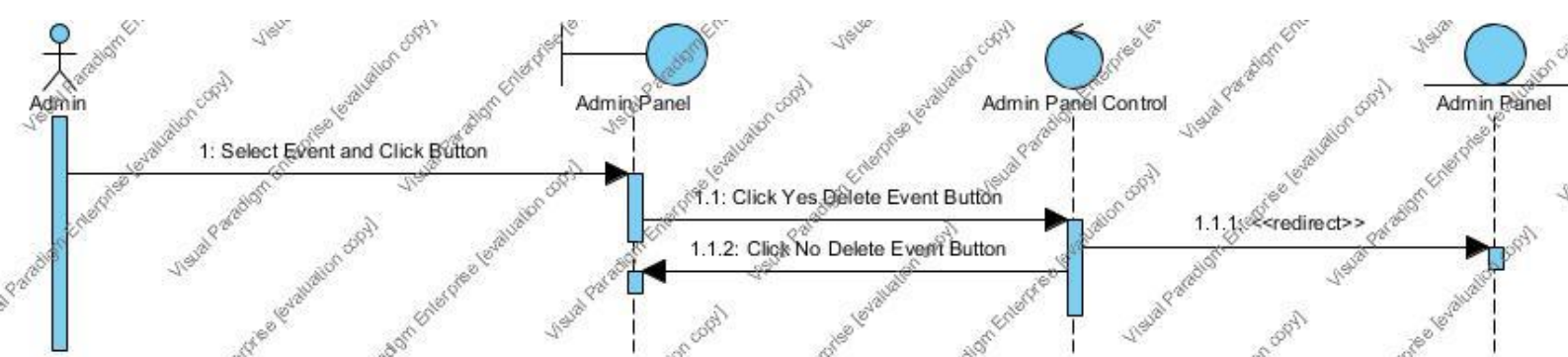
## Comment



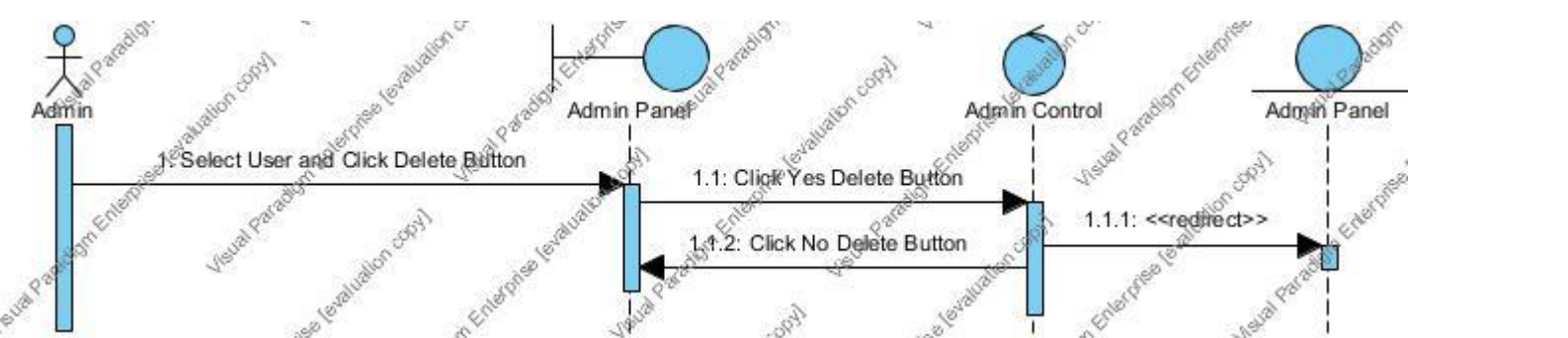
## Delete Club



## Delete Event

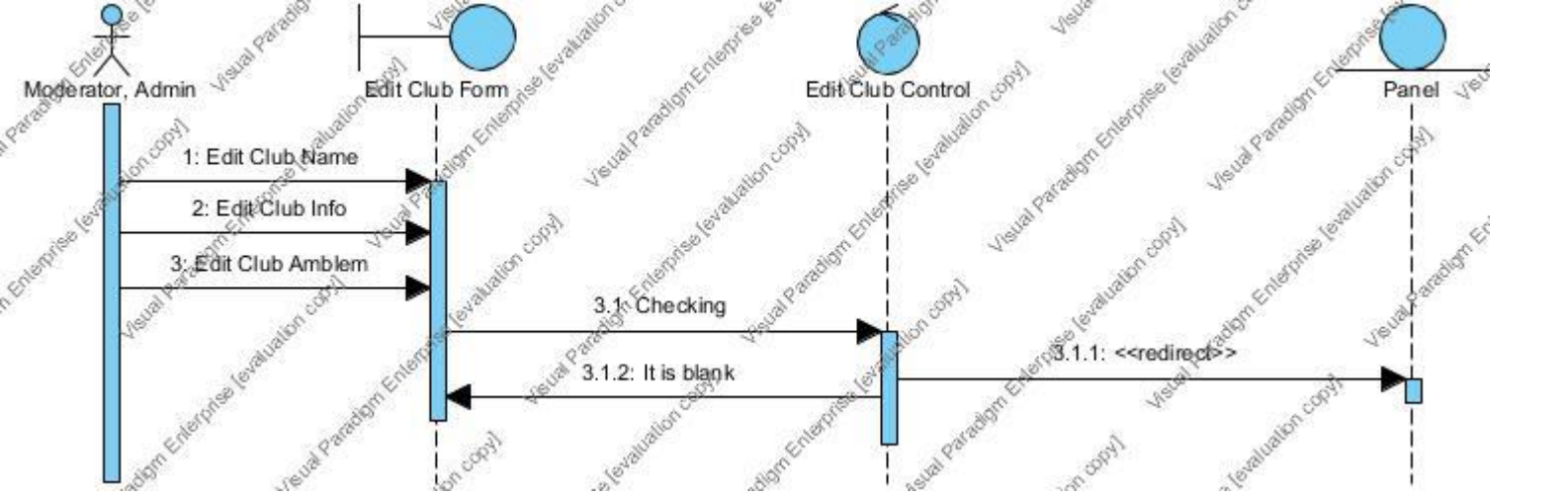


## Delete User

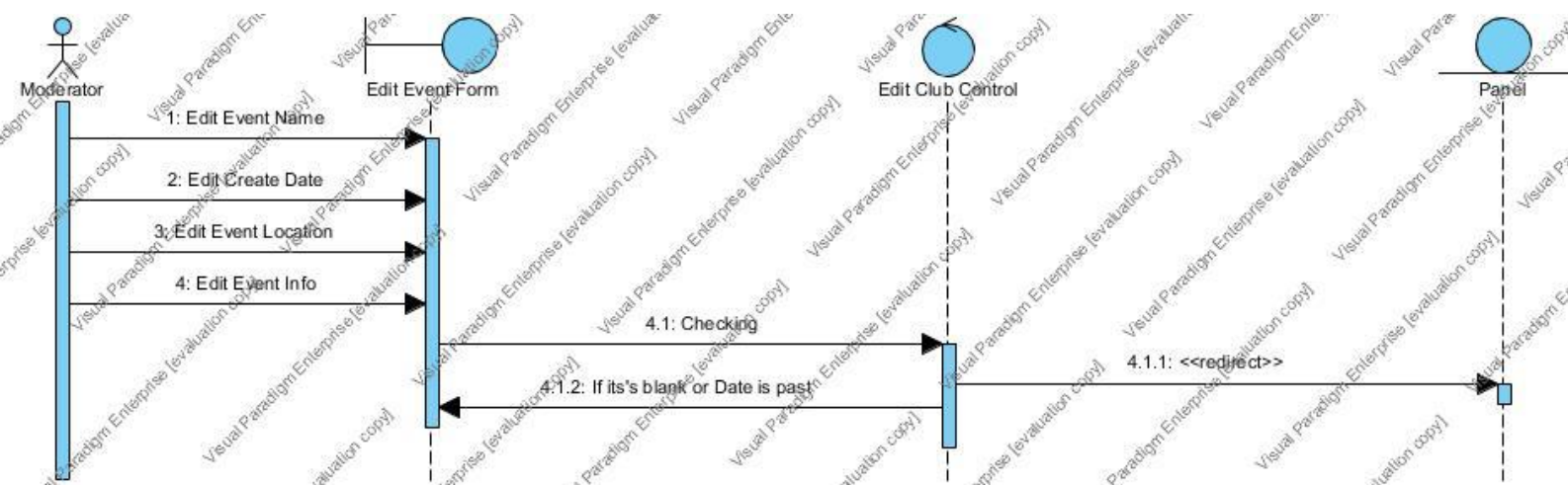


## Edit Club

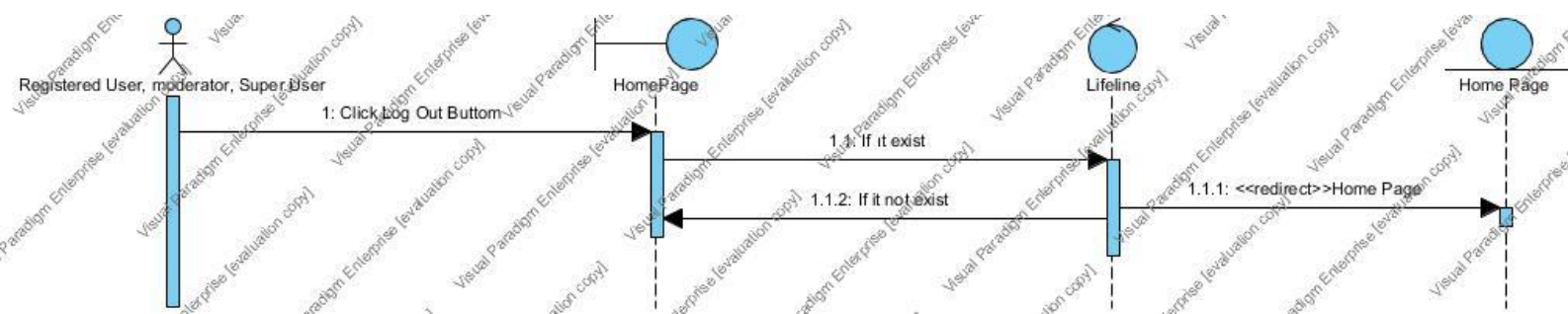




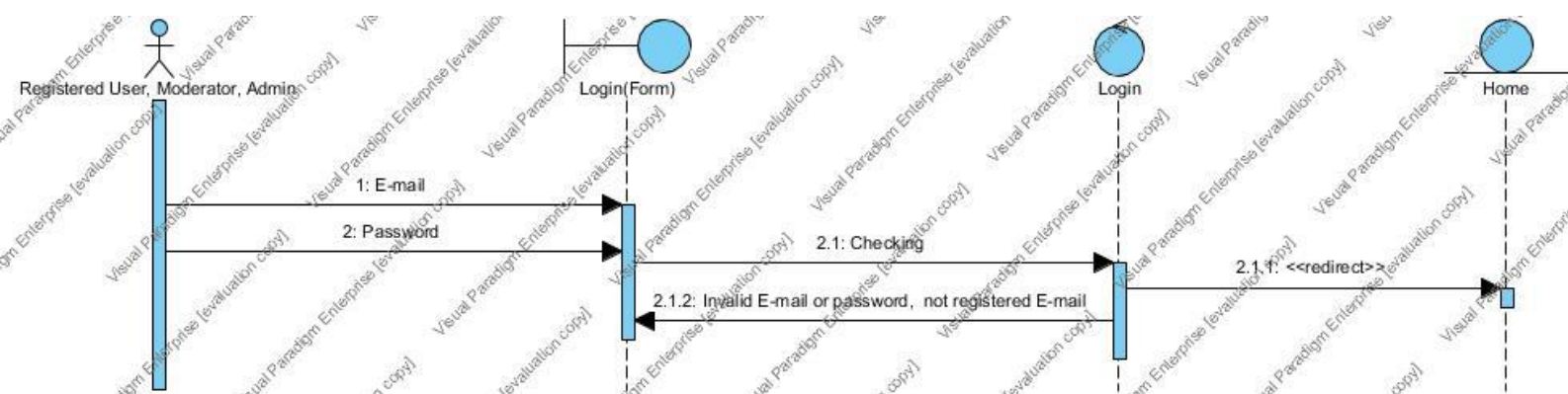
## Edit Event



## Log Out

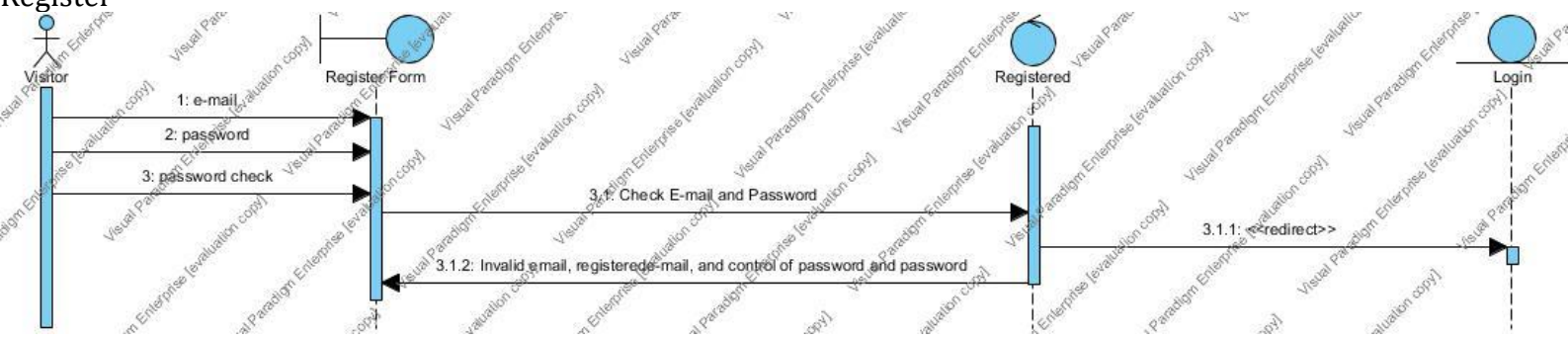


## Login

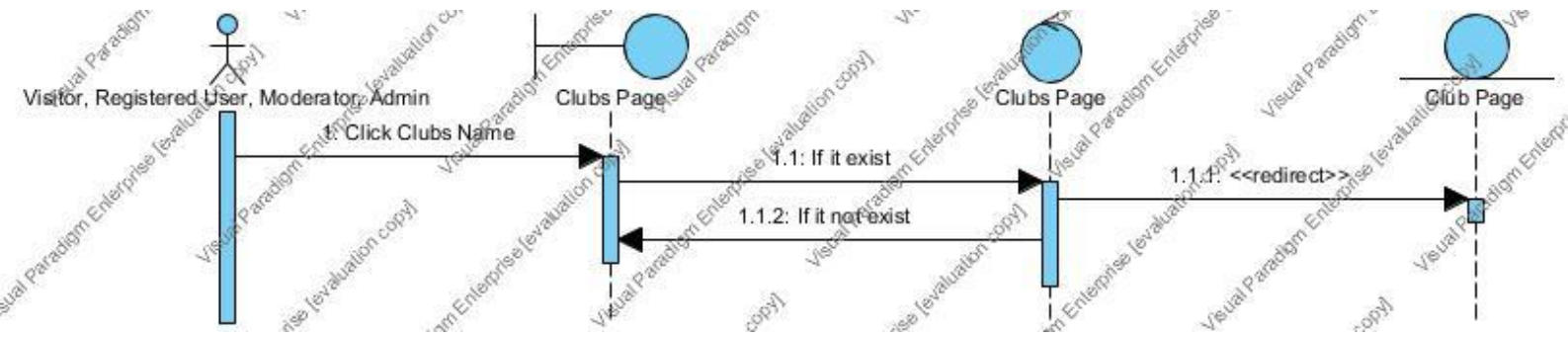




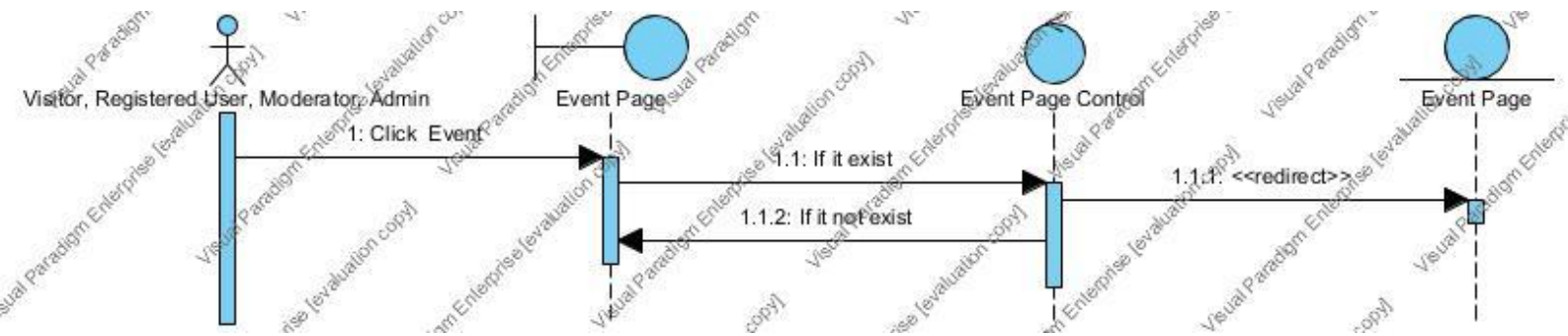
## Register



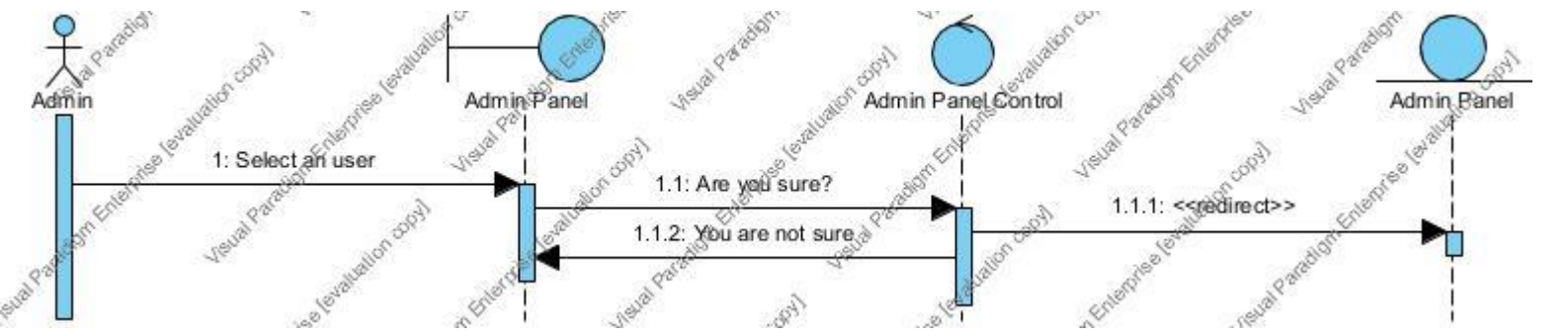
## See Clubs



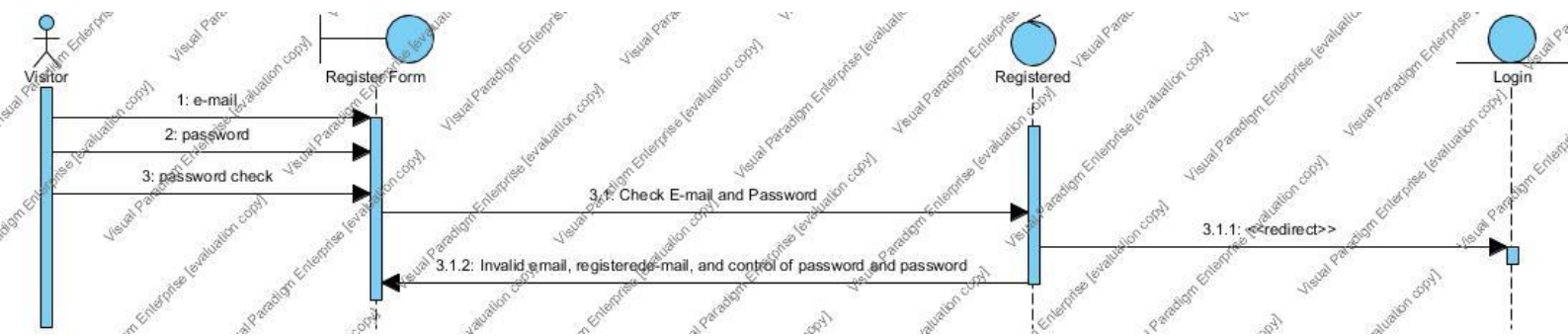
## See Events



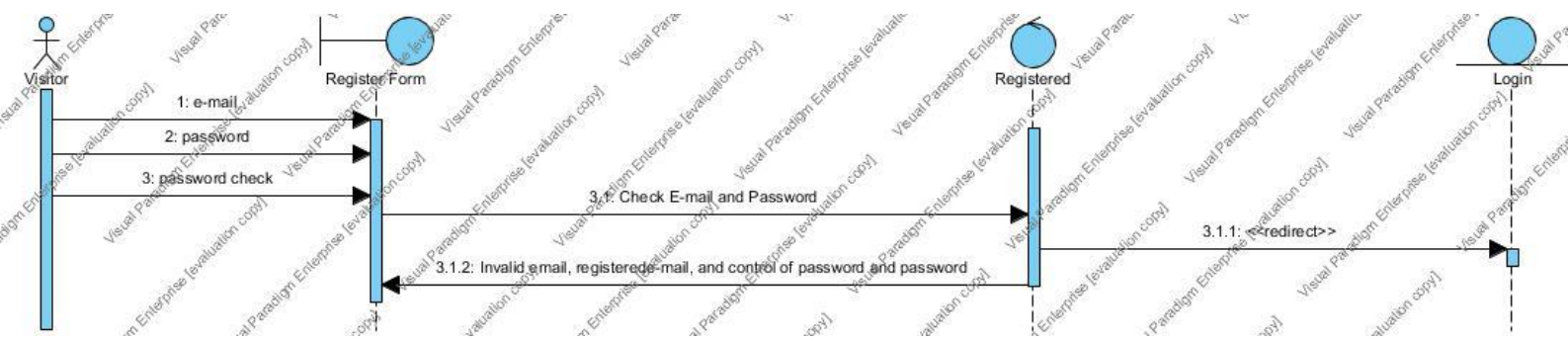
## Upgrade Status



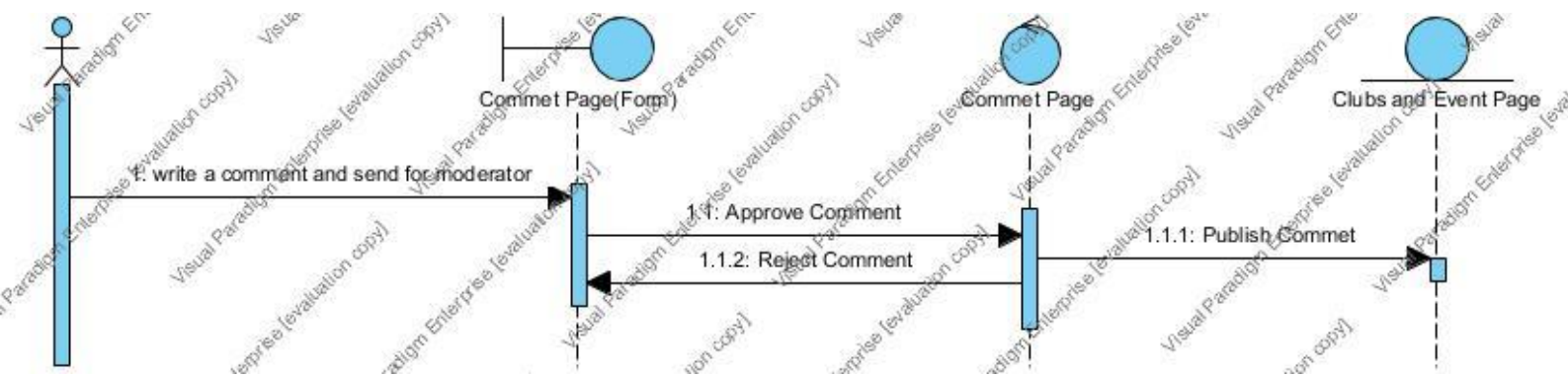
## User Registration



## Visitor Registration

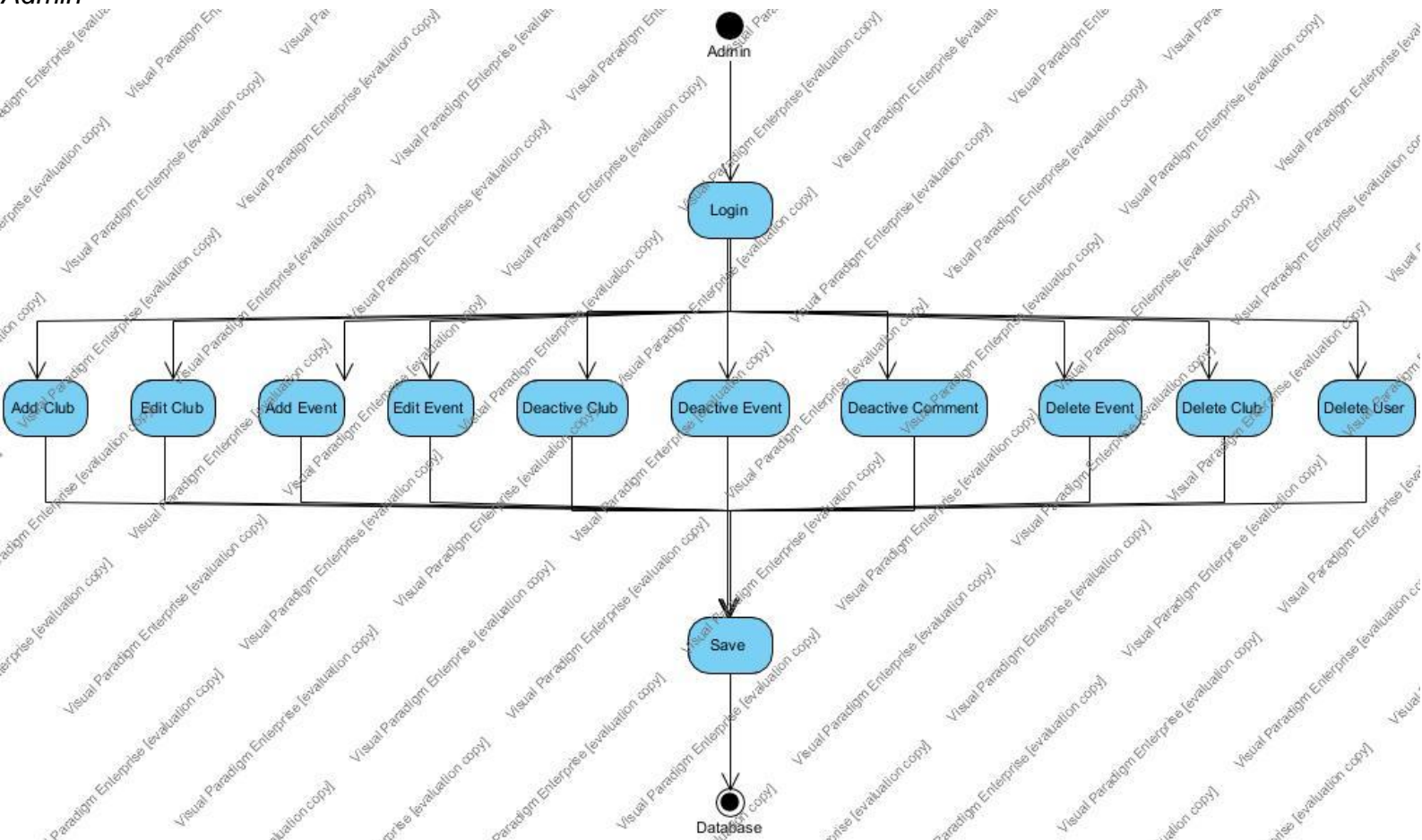


## Write Comment



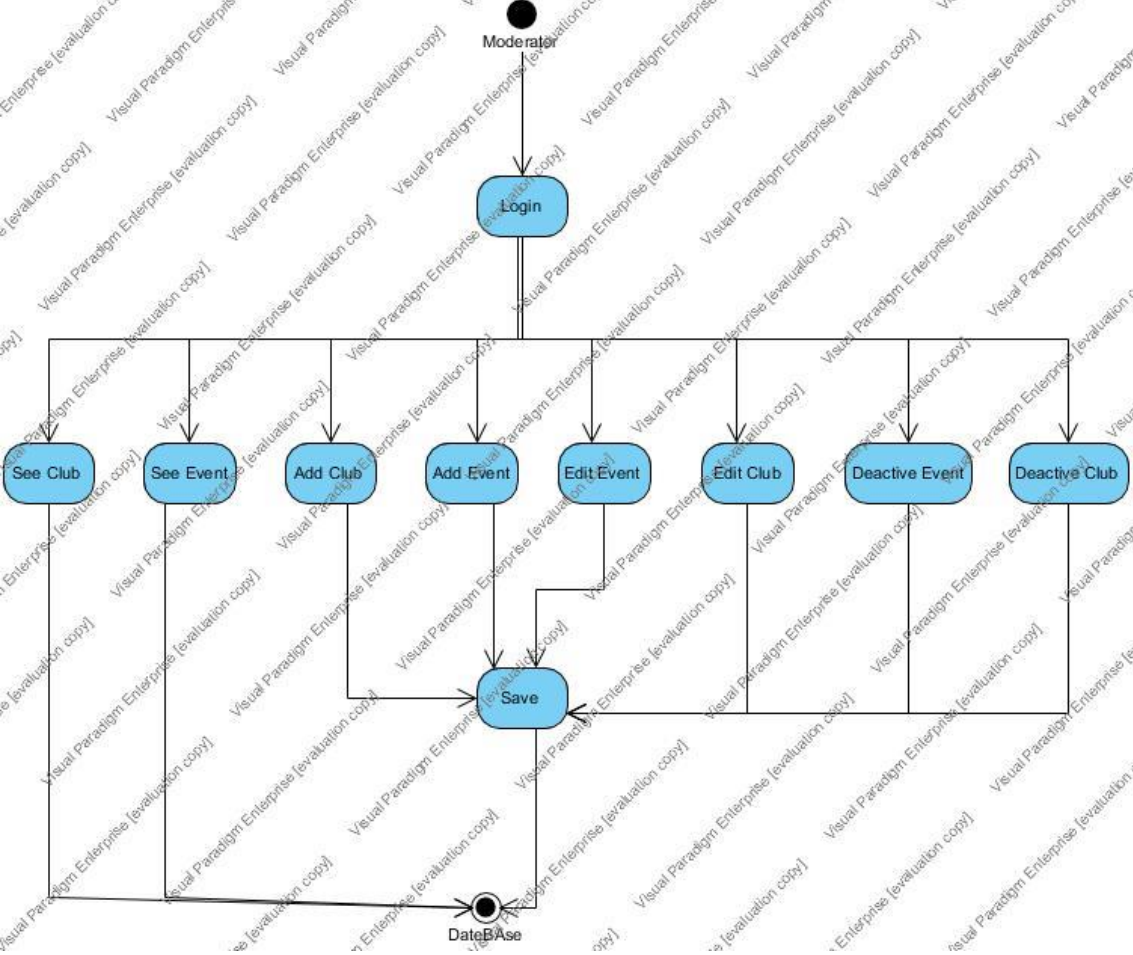
## Activity Diagrams

### Admin

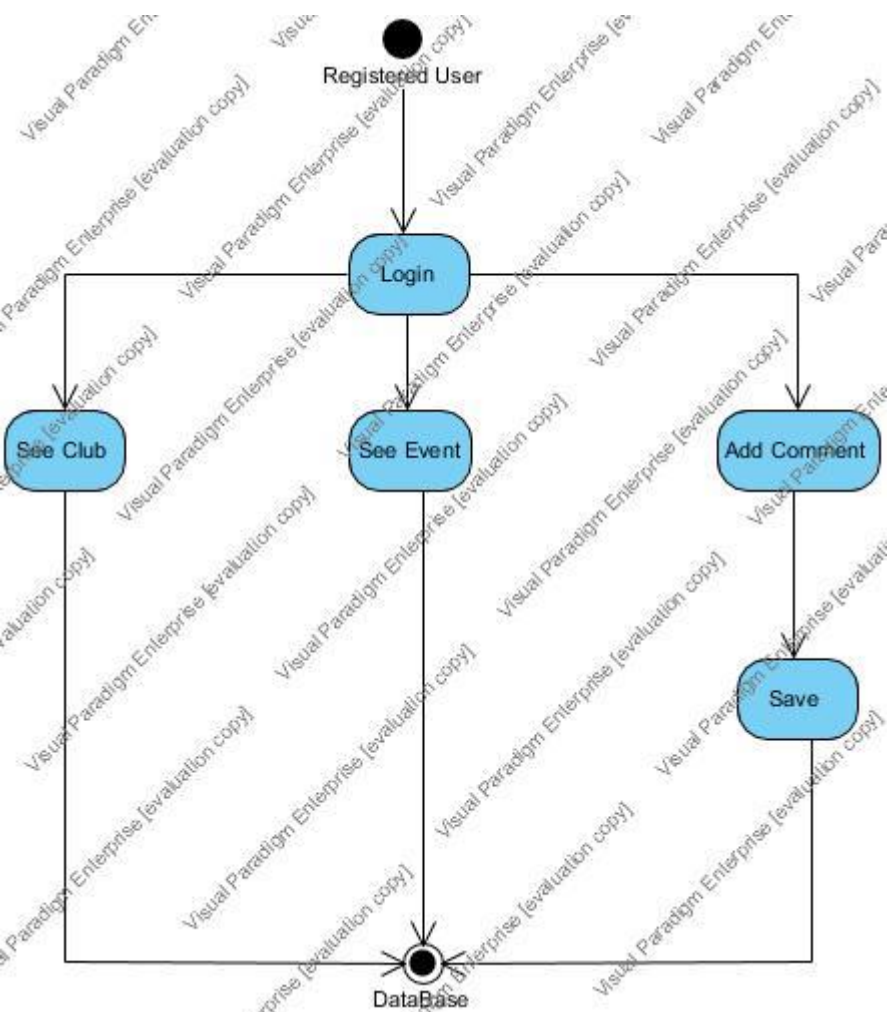


### Moderator

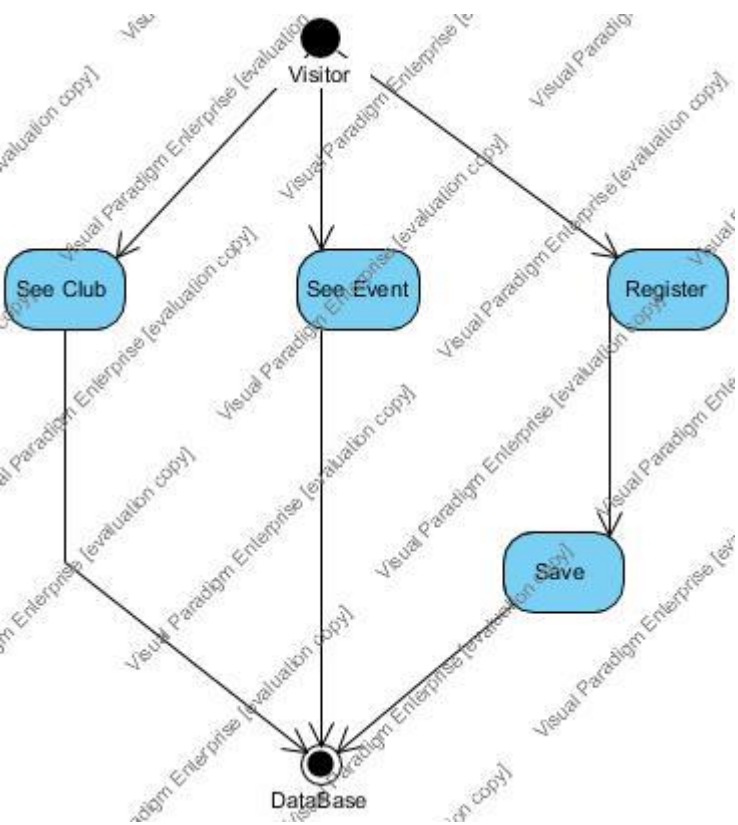




## Registered User



## Visitor



### 3.4.5 User interface—navigational paths and screen mock-ups



Hosgeldiniz. Murat Akalın

Hava Durumu

[Anasayfa](#)[Kulüpler](#)[Web\\_Mail](#)[Google\\_Maps](#)[HızlıErişim](#)

En Etkin Kulüpler

**Fotoğrafçılık Kulübü**

Fotoğrafçılık Kulübünün amacı, amatör fotoğrafçılığı yaymak, tanıtmak ve öğretmektir. Kulüp üyeleri bu amacı gerçekleştirmek için çalıştay, konferans, panel, dış mekan çekimleri için geziler ve bireysel ya da karma sergiler düzenlemektedir.

**Müzik Kulübü**

Işık Üniversitesi Müzik Kulübünün amacı, öğretim üyeleri ve öğrencileri bir araya toplamak, üniversite bünyesinde oluşan gruplara konser düzenlenmesini sağlamaktır. Ayrıca dönem içerisinde profesyonel grupları kampüse getirmektedir ve konserlerin organizasyonunu yapmaktadır.

**Sosyal Aktiviteler Kulübü**

Işık Üniversitesi Sosyal Aktiviteler Kulübü, öğrencilerin sosyal ilişkilerini geliştirmesine katkıda bulunmak amacı ile kurulmuştur. Öğrencilerin ders dışı zamanlarını değerlendirebileceği parti, gezi ve etkinlikler düzenlemektedir.



Hosgeldiniz. Murat Akalın

Hava Durumu

[Anasayfa](#) [Kulüpler](#) [Web\\_Mail](#) [Google\\_Maps](#) [HızlıErişim](#)

Son Eklenenler ☐ Tarihi geçmiş etkinlikler

## Kulüpler

- ☐ Fotoğrafçılık Kulübü
- ☐ Müzik Kulübü
- ☐ Sosyal S.P. Kulübü
- ☐ Psikoloji Kulübü
- ☐ İnşaat Kulübü
- ☐ Mimarlık ve T. Kulübü
- ☐ Minik Eller Kulübü
- ☐ IT & MIS Kulübü
- ☐ Dans Kulübü
- ☐ Uluslararası İlişkiler K.
- ☐ IEEE Kulübü
- ☐ Sosyal Aktiviteler K.
- ☐ Türk Kızılayı Kulübü
- ☐ Makina Müh. Kulübü
- ☐ Işıklı Kartallar Kulübü
- ☐ Işıklı Trabzonsporlular K.

## Etkinlikler

28.11.2018 Sosyal S.P. Kulübü Metin Sabancı Özel Eğitim Okulları  
3. Geleneksel.. Etkinliği Metin Sabancı Özel Eğitim Okulu

24.11.2018 Müzik Kulübü Sahne Ekipmanları Atölyesi Müzik  
Stüdyosu


22.11.2018 Fotoğrafçılık Kulübü Yedigöller Etkinliği Yedigöller


11.11.2018 Fotoğrafçılık Kulübü Kastamonu&Safranbolu Gezisi  
 Kastamonu-Safranbolu
2018  
Sat, Nov 17

NOVEMBER 2018						
S	M	T	W	T	F	S
					1	2
					3	4
					5	6
					7	8
					9	10
					11	12
					13	14
					15	16
					17	18
					19	20
					21	22
					23	24
					25	26
					27	28
					29	30

[CANCEL](#) [OK](#)


Hosgeldiniz. Murat Akalın

Hava Durumu

[Anasayfa](#) [Kulüpler](#) [Web\\_Mail](#) [Google\\_Maps](#) [HızlıErişim](#)

Kulübün tüm yaptığı ve yapacağı etkinlikleri buradan görebilir, kulup hakkında detaylı bilgi edinebilirsiniz.



## Fotoğrafçılık Kulübü

Fotoğrafçılık Kulübünün amacı, amatör fotoğrafçılığı yaymak, tanıtmak ve öğretmektir. Kulüp üyeleri bu amacı gerçekleştirmek için çalıştay, konferans, panel, dış mekan çekimleri için geziler ve bireysel ya da karma sergiler düzenlemektedir.

24.11.2018 Fotoğrafçılık Kulübü Yedigöller Etkinliği Yedigöller



02.10.2018 Fotoğrafçılık Kulübü Fener&amp; Balat Fotoğraf Gezisi Eminönü Yeni Camii Önü



11.08.2018 Fotoğrafçılık Kulübü Tanışma Toplantısı Cep Sineması

## 4. Glossary

### Actors in the System

*Actor*: Defined user on the system (i.e. Admin, Moderator, and Visitor).

*User*: An abstract table on the database of the system. All the existing actors in the system are inherited from this table. Actors in this table share certain account attributes. All actors are Users, then they are specialized in their respective tables according to their roles.

*Moderator*: An actor on the system. Represents an moderator of the Clubs/Events.

*Admin*: The system administrator. The admin has full authorization on the system, and can alter the database, or perform maintenance on the system.

### Development

*Back-end*: Server side of the system. All functionality runs in here.

*Front-end*: Client side of the system. All content & interfaces viewable by the users are displayed in here.

*Template*: Viewable contents.

*CRUD*: Create Read Update Delete. Mostly, this describes admin functions, as the admin can create new entries, edit them and delete them in the database, for all tables.

### Server

*Request*: From Client to Server

*Response*: From Server to Client

*Server*: Receive Request and Send Response

*Client*: Send Request and Receive Response

### Frameworks, Technologies & Approaches

*MVC*: A modern design pattern: model-view-controller.

*Html5*: New version of hypertext making language.

*CSS*: An approach to web design for make-up.

*Bootstrap*: A front-end framework for web design for responsiveness.

*JavaScript*: the programming language of HTML and the Web, provides responsiveness

*AJAX*: A technique for accessing web servers from a web page.

*JQuery*: A javascript library, to simplify javascript programming while coding.

## 5. References

- Bruegge B. & Dutoit A.H. (2010). *Object-Oriented Software Engineering Using UML, Patterns, and Java*, Prentice Hall, 3<sup>rd</sup> ed.
- Campus Automation System (2002) University of Işık, [Campus Online](#)
- Course Homepages Management System of ISIK University (2003), [Course Online](#)