EVENT MANAGEMENT SYSTEM ‘EventLog’

[**1. Introduction**](#_is38vualkzks) **3**

[1.1 Purpose](#_70p344itep76) 3

[1.2 Scope of the Project](#_1yqv9jxmggqw) 3

[1.3 Intended Audience](#_yyeqqizhgjy8) 4

[1.4 Definitions, Acronyms and Abbreviations](#_dyghiay90auu) 4

[**2.Overall Description**](#_36aemydistsm) **6**

[2.1 Product Perspective](#_ksot14qj967z) 6

[2.2 Product Functions](#_gls1q43z1tz6) 6

[2.3 User Characteristics](#_908so8pqnd00) 7

[2.4 Operating Environment](#_lh2dj2grpqja) 7

[2.5 Design and Implementation constraints](#_hw8wd3sytxe7) 8

[2.7 Assumptions and Dependencies](#_2pcjr2esgeci) 8

[**3.SPECIFIC REQUIREMENTS**](#_a5vcferwi0f0) **9**

[3.1 EXTERNAL USER REQUIREMENT](#_i2toj1b88sxq) 9

[3.1.1 Software interface](#_uymvnzh5whpm) 9

[3.1.1.1 Login Screen](#_d0q8kr741lgx) 9

[3.1.1.2 Select Interested Clubs](#_v2snqch43i23) 10

[3.1.1.3 All Events](#_nn2amb40wpnm) 10

[3.1.1.4 Expanded Event Details](#_e8l9826f0dow) 11

[3.1.1.5 Events for Me](#_e5ux8gi57jga) 11

[3.1.1.6 Club wise Events](#_n32p6onrz2s5) 12

[3.1.1.7 Registered Events](#_rgxvmoa888cg) 12

[3.1.1.8 Add Events (for Organizers)](#_delpo1xeu96n) 13

[3.1.1.9 Modify Events](#_ml01xj81mqoj) 13

[3.1.1.10 Delete Events](#_r0ujhwrv5ujf) 14

[3.1.2 Hardware interfaces](#_fr1qkg36jkdy) 14

[3.1.3 Software interfaces](#_tr7i6vh42x3j) 14

[3.1.4 Communicational interfaces](#_bkgv4kmw1xx1) 14

[3.2 FUNCTIONAL REQUIREMENTS](#_k8yefbo9ija5) 15

[3.2.1 User Class 1 - The User](#_itly3cot05ck) 15

[3.2.1.1 Functional requirement 1.1](#_ff7dx1qzvq5j) 15

[3.2.1.2 Functional requirement 1.2](#_jseqz5m5bgqo) 15

[3.2.1.3 Functional requirement 1.3](#_ujm8jnum5uav) 15

[3.2.1.4 Functional requirement 1.4](#_oowiyz8k6coy) 16

[3.2.1.5 Functional requirement 1.5](#_rcaxtq18u5zv) 16

[3.2.1.6 Functional requirement 1.6](#_5yzj5ou3m8ji) 16

[3.2.1.7 Functional requirement 1.7](#_yi7becvo7rkw) 16

[3.2.1.8 Functional requirement 1.8](#_jmt1zj8dpgzp) 17

[3.2.1.8 Functional requirement 1.10](#_ppvl7fx1yyu3) 17

[3.2.1.8 Functional requirement 1.11](#_6pqslsg93tcd) 17

[3.2.2 User Class 2 - The Organizer](#_wfirk2e96enb) 18

[3.2.2.1 Functional requirement 2.1](#_a5l7vxv5bc8) 18

[3.2.2.2 Functional requirement 2.2](#_hnnih3f5cq7) 18

[3.2.2.3 Functional requirement 2.3](#_3da8u1odue5) 18

[3.2.2.4 Functional requirement 2.4](#_8zplu8qg7nk7) 18

[3.2.2.5 Functional requirement 2.5](#_ua1fotflk505) 19

[3.2.2.1 Functional requirement 2.6](#_m17bq1xon45t) 19

[3.3 NON- FUNCTIONAL REQUIREMENTS](#_k62sf66kfi8q) 19

[3.3.1 Non- Functional requirement : Security](#_1bipf6qfx1ku) 19

[3.3.2 Non- Functional requirement : Maintainability](#_ngcpqglkz5pt) 20

[3.3.3 Non- Functional requirement : Portability](#_oakzggvms0hy) 20

[3.2.3 Non- Functional requirement : Performance](#_1526e4bg4f66) 20

[3.3.4 Non- Functional requirement : Availability](#_z1cqiwl8nttr) 21

[**5. Use Case Diagram**](#_msu4p4ef781y) **22**

[**6. ER Diagram**](#_lup5ean5mn66) **22**

[**7. Data-Flow Diagram**](#_oing9zs21elm) **24**

[**8. State Diagram**](#_26s4bwwpw99x) **30**

[**9. Activity Diagram**](#_jxc6me48vl8f) **31**

[9.1 User Activity Diagram](#_7ahkp2vk8xeu) 31

[9.2 Organizer Activity Diagram](#_e348h1guwkne) 32

[9.3 Add Event Activity Diagram](#_ld57iah14j8q) 33

[**10. Summary**](#_ot7s38ndrn9l) **36**

# 

# 1. Introduction

This section gives a scope description and overview of everything included in this SRS document. Also, the purpose for this document is described and a list of abbreviations and definitions is provided.

## 1.1 Purpose

The Software Requirements Specification (SRS) will provide a detailed description of the requirements for the Event Management System. This SRS will allow for a complete understanding of what is to be expected from the newly introduced system which is to be constructed. The clear understanding of the system and its functionality will allow for the correct software to be developed for the end user and will be used for the development of the future stages of the project. It is primarily intended to be proposed to a customer for its approval and provide a foundation for developing the first version of the system for the development team. It provides a common reference point for both the developer team and stakeholder community. The SRS will evolve over time as users and developers work together to validate, clarify and expand its contents.

## 1.2 Scope of the Project

The Event Management System, “EventLog” is a Web application which helps students and organizers belonging to a University to facilitate and manage the events in a more organized way. This web application should be free and accessible from any browser on desktop. This software includes all events occurring in the university and gives permission to the users to view and register through it. An Organizer also uses the web-portal in order to administer the system and manage events. It can also access all the functionalities of a user.

The objectives of the Event Management System is to simplify the management of all the data related to the various events that take place in a University. It would provide the students have access to every detail of the events updated by the organizers, just one click away. This would exterminate the idea of sending emails for activities at the university as a whole. Safety, ease of use and most importantly the efficiency of information display and retrieval are some benefits the development team are going to present with this system. This system should be user appropriate, easy to use, provide easy recovery of errors and have an overall end user high subjective satisfaction.

## 1.3 Intended Audience

This project is intended to be used by a particular University that includes faculty, students, staff, organizer of events, and so on. This SRS is intended for several audiences, including the customer, as well as the project manager, designers, developer and tester.

* The customer will use this SRS to verify that the developer team has created a product that is acceptable to the customer.
* The project manager of the developer team will use this SRS to plan milestones and a delivery date, and ensure that the developing team is on track during development of the system.
* The designer will use this SRS as a basis for creating the system’s design. The designer will continually refer back to this SRS to ensure that the system they are designing will fulfill the customer’s needs.
* The developer will use this SRS as a basis for developing the system’s functionality. The developer will link the requirements defined in this SRS to the software they create to ensure that they have created software that will fulfill all of the customer’s documented requirements.
* The tester will use this SRS to derive test plans and test cases for each documented requirement. When portions of the software are complete, the tester will run his tests on that software to ensure that the software fulfills the requirements documented in this SRS. The tester will again run his tests on the entire system when it is complete and ensure that all requirements documented in this SRS have been fulfilled.

## 1.4 Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| **TERM** | **DEFINITION** |
| User | Someone who interacts with the web application |
| Organizer | Someone who is given specific permission for managing the events in the system. |
| Web-Portal | A web application which presents all the events and can be used by both user and organizer |
| Stakeholder | Any person who has interaction with the system who is not a developer |
| SRS | Software Requirement Specification |
| SQL | Structured Query Language |
| DEP | Dependency |
| DESC | Description |
| RAT | Rational |
| TAG | A unique, persistent identifier |
| GIST | A short, simple description of the concept |
| SCALE | The scale of measure used by the requirement |
| METER | The process or device used to establish location on a SCALE |
| MUST | The minimum level required to avoid failure |
| PLAN | The level at which good success can be claimed |
| WISH | A desirable level of achievement that may not be attainable through current approaches |
| DEFINED | The official definition of a term |

# 2.Overall Description

## 2.1 Product Perspective

The Event Management System is a new self-contained software product which will be produced by the project team. The newly introduced system will provide easy access to the system and it will contain user friendly functions with attractive interfaces. This system handles all event management activities and provides ease for the user to get all the desired information on one platform and helps in preventing the need of unnecessary mails that leads to loss of important information.

The system consists of a web application with different interfaces for users and organizer to facilitate the required management. Since this is a data-centric product it will need somewhere to store the data. For that, a database will be used. The web portal will communicate with the database and will add and modify data. All of the database communication will go over the Internet.

## 2.2 Product Functions

The main purpose of the product is to provide one stop platform for all the activities, events and updates happening in the university and thereby, reducing the amount of unnecessary emails and preventing the loss of important information and therefore, increasing the efficiency and convenience to the user.

This product lets registered users to view, show interest and register in different categories of events and activities occurring at the university mainly including the club category-wise events, events that the user has registered, upcoming talks/workshops and suggested events specific to the user. It also lets registered organizers manage these events by adding, modifying and deleting the details of events and can manage and view the user information mainly including the registered users of different events and students interested in specific clubs.

The web portal will provide functionality to manage the system and the event information. It will also provide information about the system, for example show when there is a new update.

## 2.3 User Characteristics

There are two types of users that interact with the system: users including students of the University, and organizers of events. Each of these two types of users has different use of the system so each of them has their own requirements.

The user that includes the students of the university can view the upcoming and ongoing events at the university with the help of university email only. Every user just needs to give the interested club list (min 3) to view the application. He/she can view different types of categories of events listed including - suggested events specifically for the user, club category-wise events and registered events and can register for the same.

The organizer is expected to be familiar with the interface of the tech support system as in how to manage the information pertaining to the events. Also must take the responsibility of posting the proper details of the event. He/she also has the access to view the details of users that are registered for the events the organizer added in the past. The organizer can also register as a user and can view and register for events.

## 2.4 Operating Environment

This is a web based system and hence will require the operating environment for a client and server GUI.

**Software**

* Website is designed to run on any platform which supports web-browsers like Google Chrome and Safari.
* A running XAMPP server if wanted to host it on localhost.
* Database is installed on phpmyadmin therefore requires a server to host it.

**Hardware**

* Operating System Supports all known operating systems, such as Windows, Linux or Mac.
* Computer 512MB+ RAM, monitor with minimum resolution of 1024x768, keyboard, and mouse.

## 2.5 Design and Implementation constraints

The Internet connection is a constraint for the application. Since the application fetches data from the database over the Internet, it is crucial that there is an Internet connection for the application to function.

Decisions regarding which database to use should be taken considering the fact that data being exchanged or stored is large, and the appropriate data management system will yield efficient performance.

This system has a langage constraint and intended to be executed in ‘English’ language. Hence, it does not provide multilingual support to users.

## 2.7 Assumptions and Dependencies

One assumption about the product is that it will always be able to be accessed by an Internet Browser on Desktop. The dependency is that it requires Internet based servers for its web-based features to be displayed over the website. Users should have basic information and knowledge of how to connect to an Internet and go to a website. Users and organizers will only be able to access the system by the University Email Id to make sure that the person is authorized and an eligible viewer that is a part of the university.

In the case of Databases, it will be dependent on SQL servers for information management of the system.

It is also assumed that clients could change that decision on the next phases of the software development. According to the agile methodology, clients and developers have the flexibility to incorporate changes in the system design and implementation as and when required.

Another technical assumption is that the organizer will himself first authorize the event information from the head of the department and then add the validated details of the events for the users to view. This feature comes under exciting requirements and will be incorporated in the system in the further increments of the product.

# 3.SPECIFIC REQUIREMENTS

This section provides the functional and quality requirements of the system. It gives the detailed description of the system and its features.

## 3.1 EXTERNAL USER REQUIREMENT

This section provides a detailed description of all the inputs and outputs from the system. It also gives a description of the software communication interfaces and provides basic prototypes of the user interface.

#### 3.1.1 Software interface

The user’s browser should be compatible for a satisfactory user experience.

##### 3.1.1.1 Login Screen

##### 3.1.1.2 Select Interested Clubs

##### 3.1.1.3 All Events

##### 3.1.1.4 Expanded Event Details

##### 3.1.1.5 Events for Me

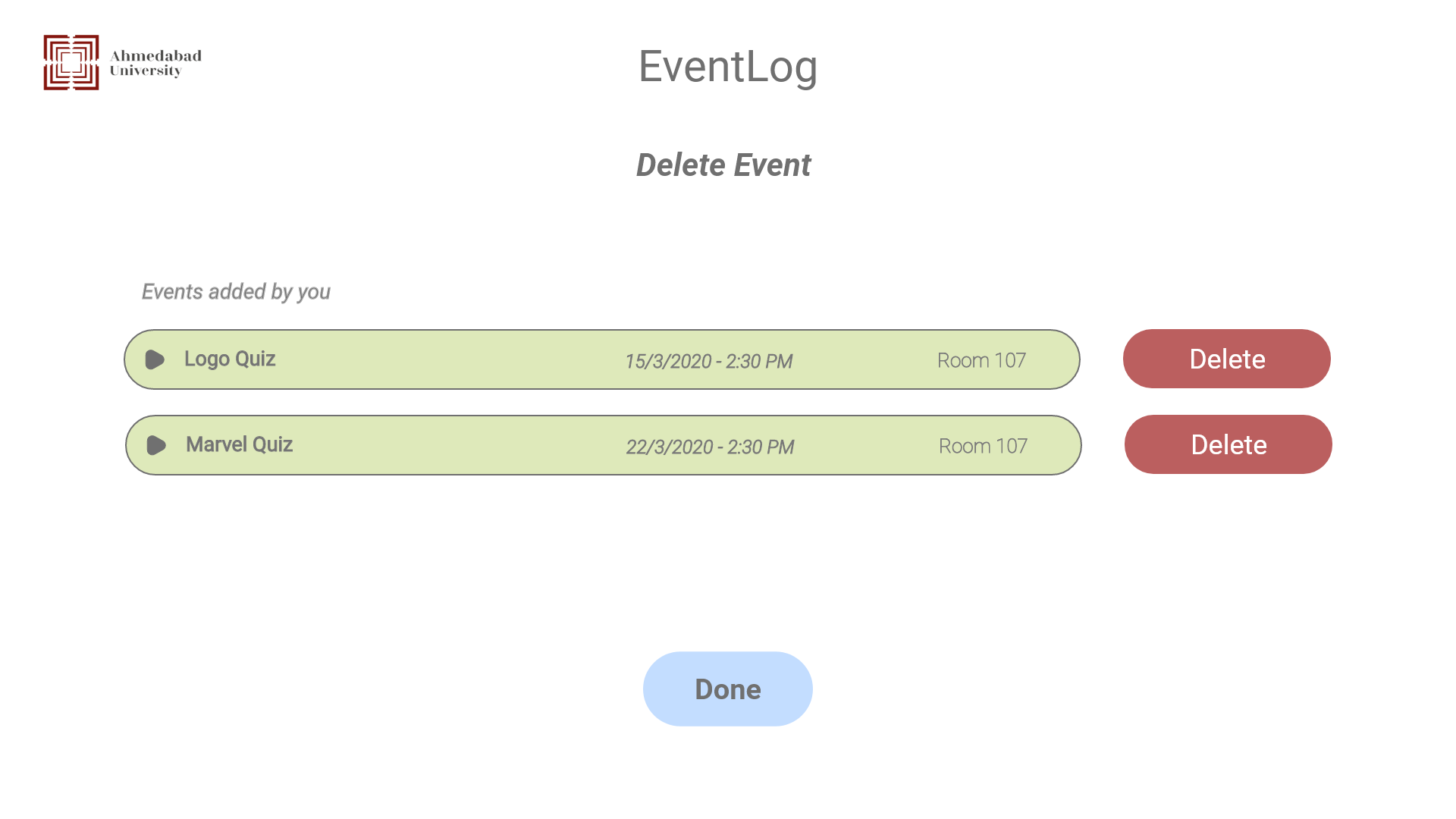
##### 3.1.1.6 Club wise Events

##### 3.1.1.7 Registered Events

##### 3.1.1.8 Add Events (for Organizers)

##### 3.1.1.9 Modify Events

##### 3.1.1.10 Delete Events



#### 

#### 3.1.2 Hardware interfaces

As none of the web applications require any specific hardware interfaces as such. The internet connection is managed by the operating system and the database server is managed by the underlying operating system on the web server.

#### 

#### 3.1.3 Software interfaces

The communication between the database and the web portal consists of operation concerning both reading and modifying the data, while the communication between the database and the web application consists of only reading operation.

#### 3.1.4 Communicational interfaces

The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems for the web portal.

## 3.2 FUNCTIONAL REQUIREMENTS

This section includes the requirements that specify all the fundamental actions of the software system.

#### 3.2.1 User Class 1 - The User

##### 3.2.1.1 Functional requirement 1.1

**ID: FR1**

**TITLE**: View Web application

**DESC:** A user should be able to view the web application through any browser on the mobile phone.

**RAT**: In order for a user to view the web application.

**DEP**: None

##### 3.2.1.2 Functional requirement 1.2

**ID: FR2**

**TITLE**: User Login - Web application

**DESC:** A user should be able to login in the system by doing google sign-in with his/her University Email Id. Since it is google sign-in, the user will be able to get logged in automatically if it has been logged in in google mail.

**RAT**: In order for a user to login in the web application.

**DEP**: FR1

##### 3.2.1.3 Functional requirement 1.3

**ID: FR3**

**TITLE**: Select ‘Interested Clubs’

**DESC:** A user should be able to select at least 3 and at most 5 clubs which the user finds interesting on the first time of sign-in.

**RAT**: In order for a user to select interested clubs for the first time of login.

**DEP**: FR1

##### 3.2.1.4 Functional requirement 1.4

**ID: FR4**

**TITLE**: View All Events

**DESC:** A user should be able to select ‘all events’ lined up by the most recent events.

**RAT**: In order for a user to login in the web application.

**DEP**: FR1, FR2

##### 3.2.1.5 Functional requirement 1.5

**ID: FR5**

**TITLE**: View Club Wise Events

**DESC:** A user should be able to select a club category from the bar menu and view the respective club events by double-clicking on the club name.

**RAT**: In order for a user to view all club events category-wise.

**DEP**: FR1, FR2

##### 3.2.1.6 Functional requirement 1.6

**ID: FR6**

**TITLE**: View Registered Events

**DESC:** A user should be able to view the events that the user has registered by clicking on the ‘Registered Events’ in the bar menu.

**RAT**: In order to view registered events of that user.

**DEP**: FR1, FR2

##### 3.2.1.7 Functional requirement 1.7

**ID: FR7**

**TITLE**: View Suggested Events / ‘Events for Me’

**DESC:** A user should be able to view the suggested events by clicking on the ‘Events for Me’ in the bar menu. It consists of all the events related to the clubs that the user has previously selected as ‘Interested Clubs’.

**RAT**: In order for a user to view suggested events.

**DEP**: FR1, FR2

##### 3.2.1.8 Functional requirement 1.8

**ID: FR8**

**TITLE**: View details of events

**DESC:** A user should be able to view the detailed description and details of the events by double clicking on the specific event part. It includes a detailed description, poster of the event, venue, date, time, fee, number of applicants registered and left seats, names of faculties or mentors if involved, and more.

**RAT**: In order for a user to like the events shown.

**DEP**: FR3, FR4, FR5, FR6, FR7

##### 

##### 3.2.1.9 Functional requirement 1.9

**ID: FR9**

**TITLE**: View the number of applicants registered

**DESC:** A user should be able to view the total number of applicants registered and total number of seats remaining for the particular event.

**RAT**: In order for a user to view the number of applicants registered.

**DEP**: FR3, FR4, FR5, FR6, FR7, FR8

##### 3.2.1.10 Functional requirement 1.10

**ID: FR10**

**TITLE**: Register for the events

**DESC:** A user should be able to register for the events by going

**RAT**: In order for a user to view the number of applicants registered.

**DEP**: FR3, FR4, FR5, FR6, FR7, FR8

#### 

#### 3.2.2 User Class 2 - The Organizer

##### 3.2.2.1 Functional requirement 2.1

**ID: FR11**

**TITLE**: Organizer Log in

**DESC**: An organizer should be able to log in with his/er university email id with an organizer registered account.

**RAT**: In order for an organizer to log into the system.

DEP: None

##### 3.2.2.2 Functional requirement 2.2

**ID: FR12**

**TITLE**: Add Events

**DESC:** Given the organizer is logged in, he/she should be able to add events by inputting required information such as Venue, Date, Time, Capacity of users, Faculty or mentors if involved, fees, poster, and so on.

**RAT**: In order for an organizer to add an event into the system.

**DEP**: FR2

##### 3.2.2.3 Functional requirement 2.3

**ID: FR13**

**TITLE**: Modify Events

**DESC:** Given the organizer is logged in, he/she should be able to modify events by modifying the information such as Venue, Date, Time, Capacity of users, Faculty or mentors if involved, fees, poster, and so on.

**RAT**: In order for an organizer to modify an event of the system.

**DEP**: FR12, FR13

##### 3.2.2.4 Functional requirement 2.4

**ID: FR14**

**TITLE**: Cancel Events

**DESC:** Given the organizer is logged in, he/she should be able to cancel the events after stating valid reasons for the same.

**RAT**: In order for an organizer to cancel an event of the system.

**DEP**: FR12, FR13

##### 3.2.2.5 Functional requirement 2.5

**ID: FR14**

**TITLE**: View Registered Users

**DESC:** Given the organizer is logged in, he/she should be able to view the registered users of the events that he/she added in the system.

**RAT**: In order for an organizer to view the registered users of the added events.

**DEP**: FR12, FR13

##### 3.2.2.1 Functional requirement 2.6

**ID: FR12**

**TITLE**: User functionality to organizer

**DESC:** An organizer should be able to do all the user functions including viewing events, and registering for events.

**RAT**: In order for an organizer to access all the user functionality also.

**DEP**: None

## 

## 3.3 NON- FUNCTIONAL REQUIREMENTS

#### 3.3.1 Non- Functional requirement : **Security**

**ID: NR1**

TAG: Admin Login Account

GIST: Security of accounts.

SCALE: If an admin tries to log in to the web portal with a non-existing account then the admin should not be logged in. The admin should be notified about log-in failure.

MUST: 100% of the time.

**ID: NR2**

TAG: User Login Account

GIST: Security of accounts.

SCALE: If a user tries to log in to the web portal with an email account other than the university email then the user should not be logged in. The user should be notified about log-in failure.

MUST: 100% of the time.

#### 3.3.2 Non- Functional requirement : **Maintainability**

**ID: NR3**

TITLE: Application extensibility

DESC: The application should be easy to extend. The code should be written in a way that it favors implementation of new functions.

RAT: In order for future functions to be implemented easily to the application.

DEP: none

**ID: NR4**

TITLE: Application testability

DESC: Test environments should be built for the application to allow testing of the applications' different functions.

RAT: In order to test the application.

DEP: none

#### 3.3.3 Non- Functional requirement : **Portability**

**ID: NR5**

TITLE: Application portability

DESC: If hosted, the application should be able to run in any browser.

RAT: The adaptable platform for the application to run on.

DEP: none

#### 3.2.3 Non- Functional requirement : **Performance**

**ID: NR6**

TAG: Response Time

GIST: The fastness of the changing between different web pages

SCALE: The response time of the different output webpage after performing some event.

MUST: No more than 5 seconds 100% of the time.

WISH: No more than 2 seconds 100% of the time.

**ID: NR7**

QR8 TAG: System Dependability

GIST: The fault tolerance of the system.

SCALE: If the system gets some strange input or loses the connection to the Internet, the user should be informed.

MUST: 100% of the time.

#### 

#### 3.3.4 Non- Functional requirement : **Availability**

**ID: NR8**

TAG: System Availability

GIST: Accounting the availability of the system of the times it is used

SCALE: The average system availability without taking into consideration the network failure.

MUST: More than 98% of the time.

WISH: 100% of the time

**ID: NR9**

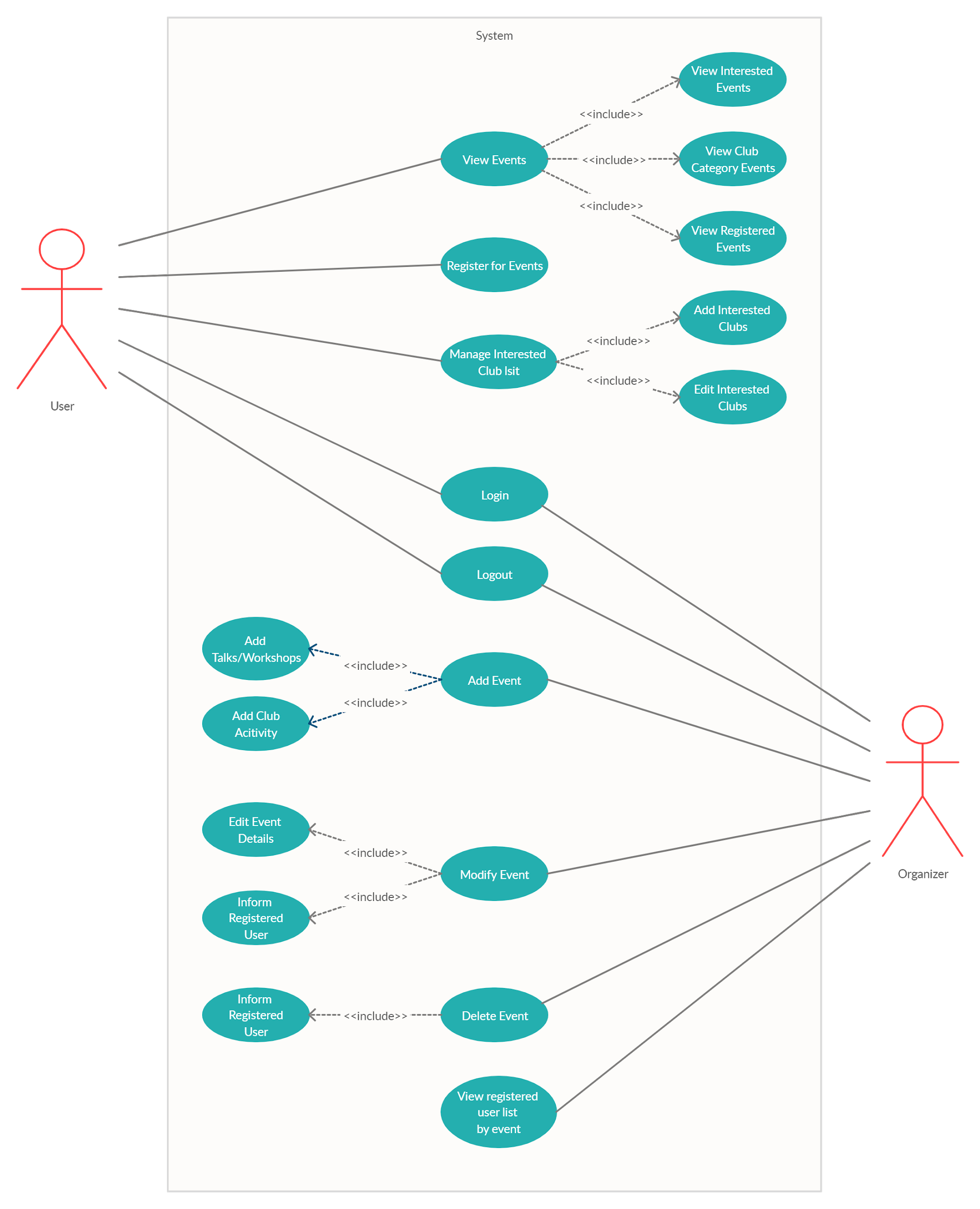
TITLE: Internet Connection Requirement

DESC: The web application should be connected to the Internet.

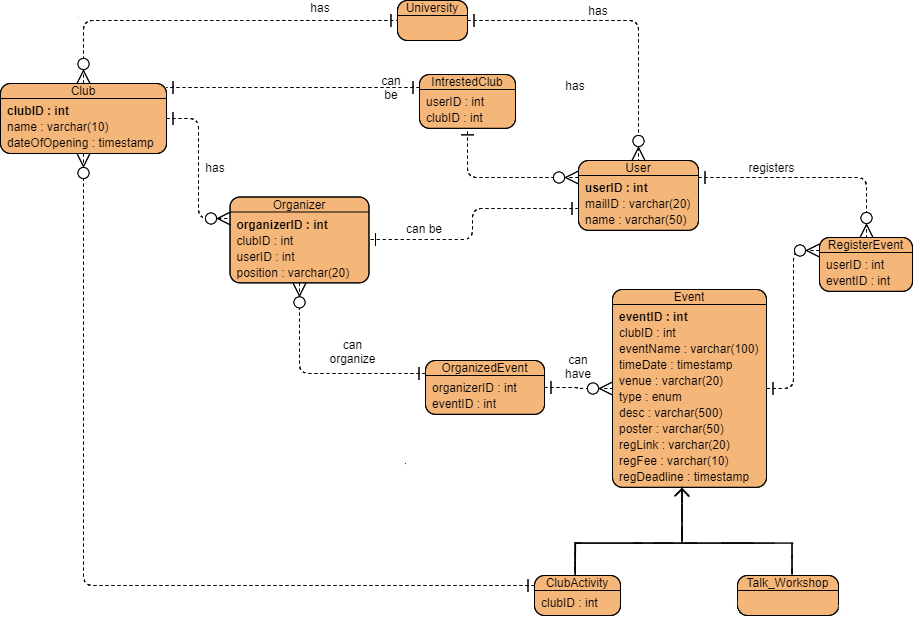
RAT: In order for the application to communicate with the database.

DEP: none

# 5. Use Case Diagram

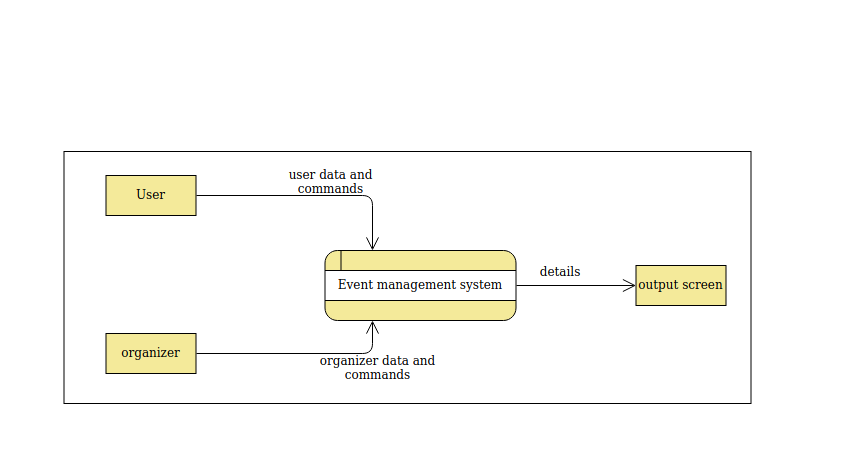


# 6. ER Diagram

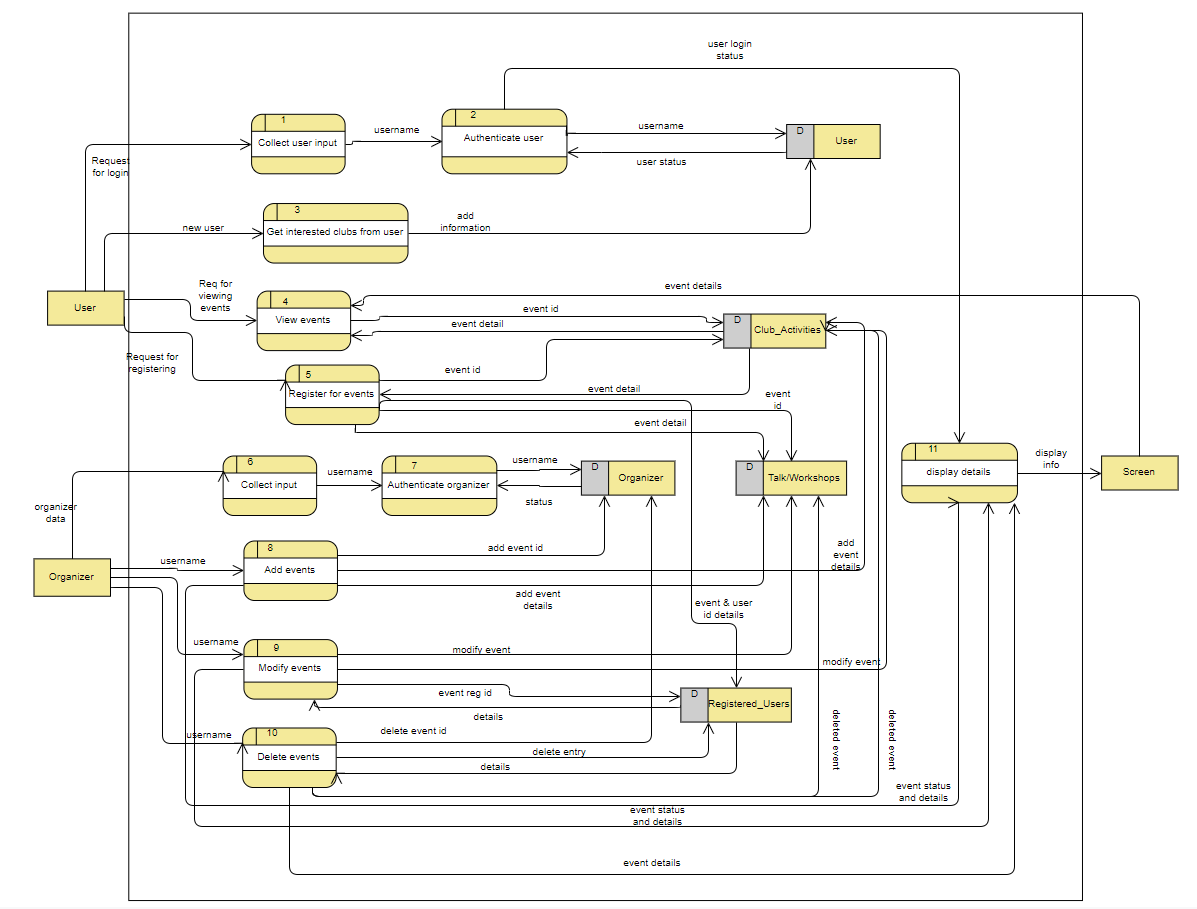


# 7. Data-Flow Diagram

7.1 Level 0 - Context Level Diagram

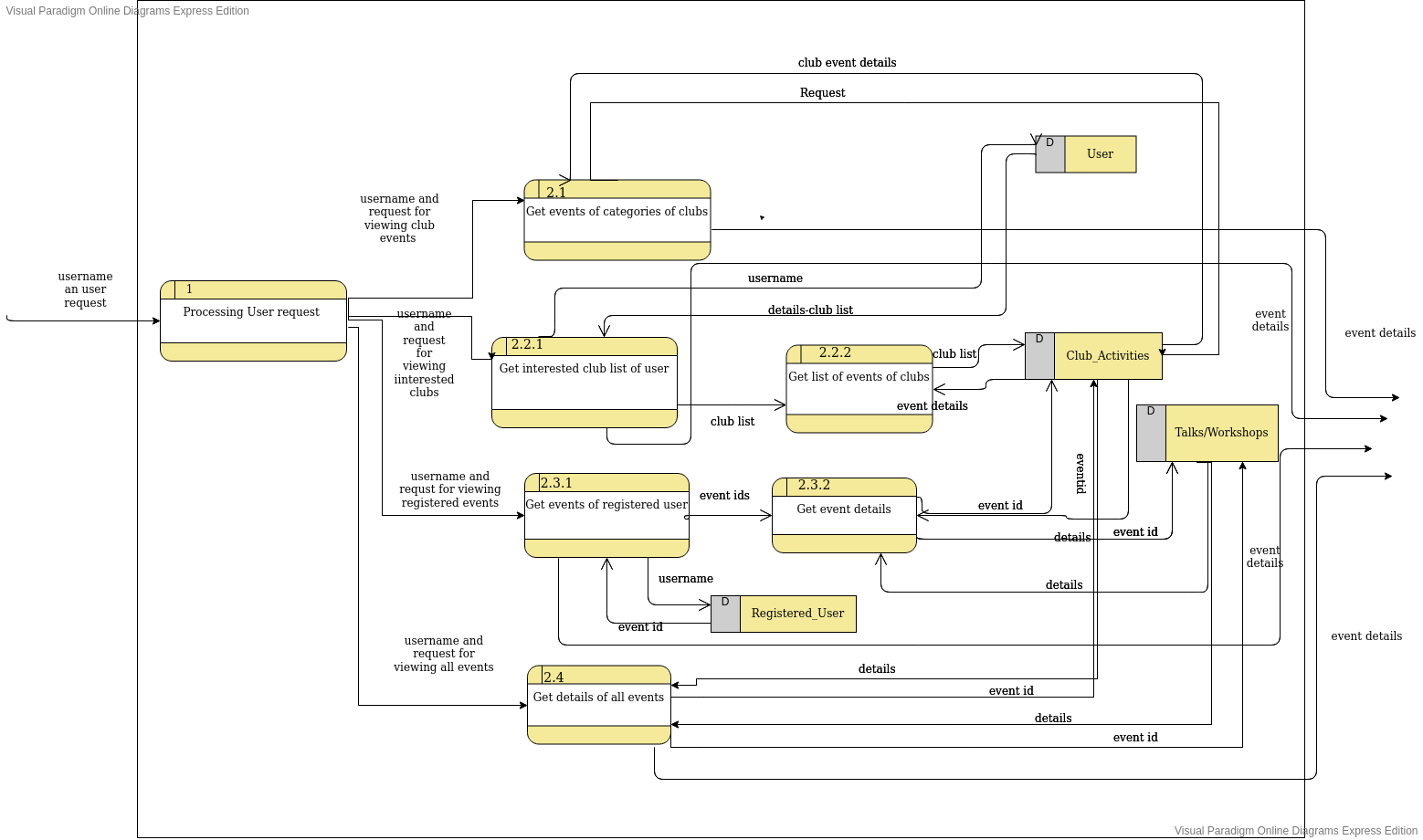


7.2 Level - 1

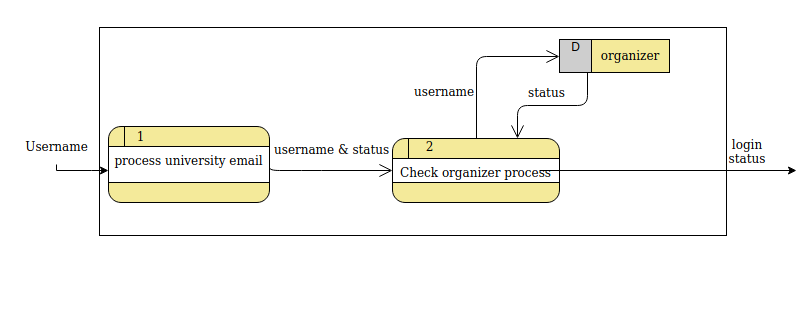


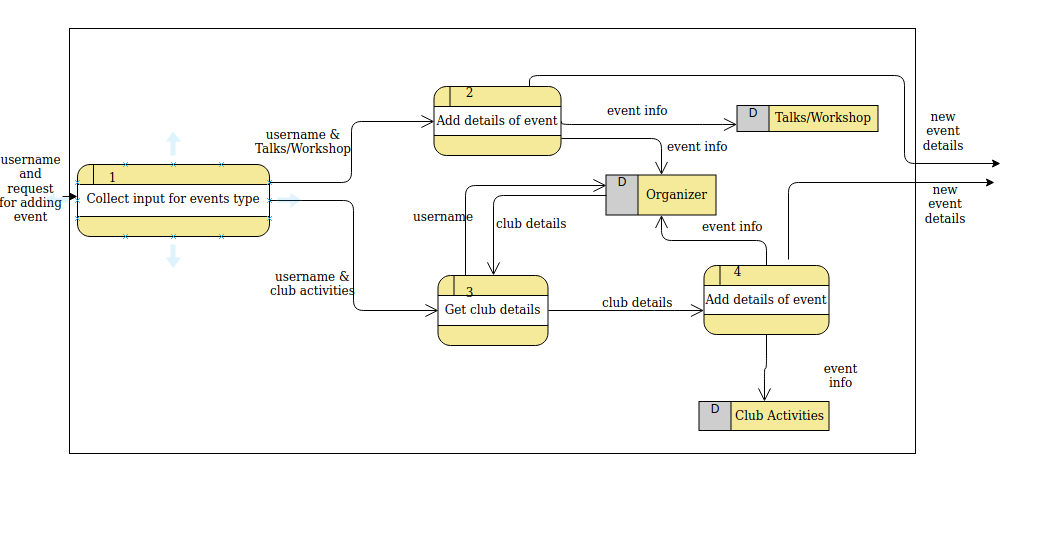
7.3 Level 2

7.3.1 Level 2 - View Events Process

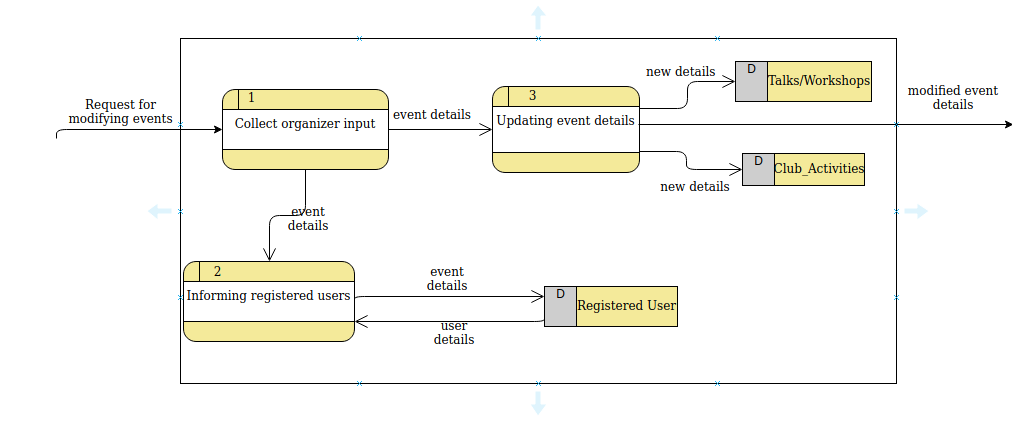


7.3.2 Level 2 - Authorize Organizer process

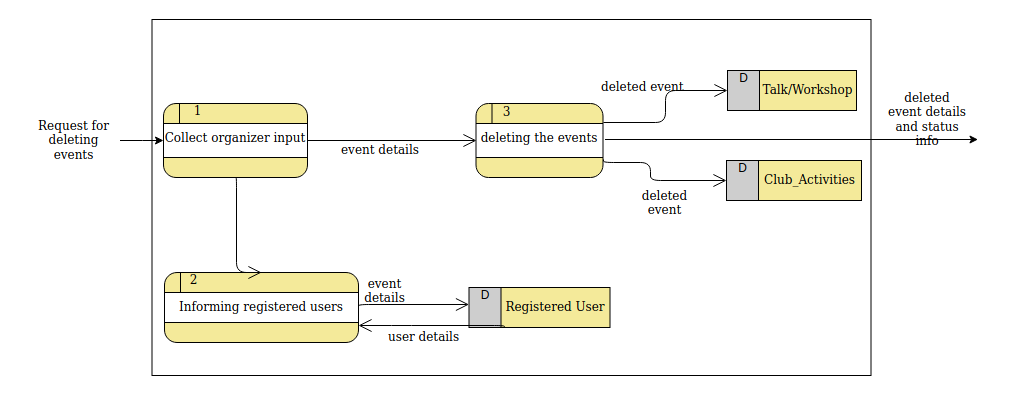


7.3.3 Level 2 - Add Events process

7.3.4 Level 2 - Modify Events process

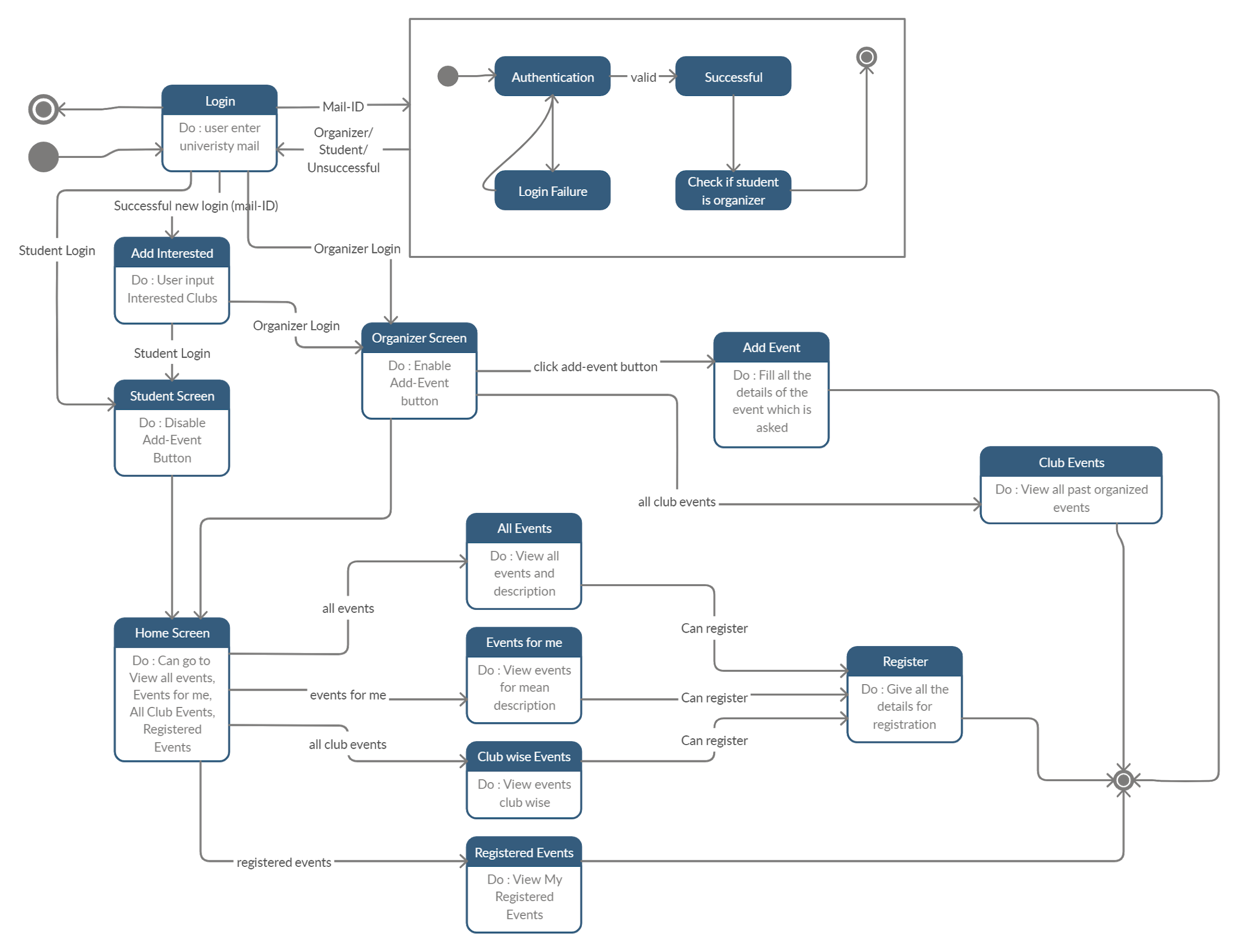


7.3.5 Level 2 - Delete Events process



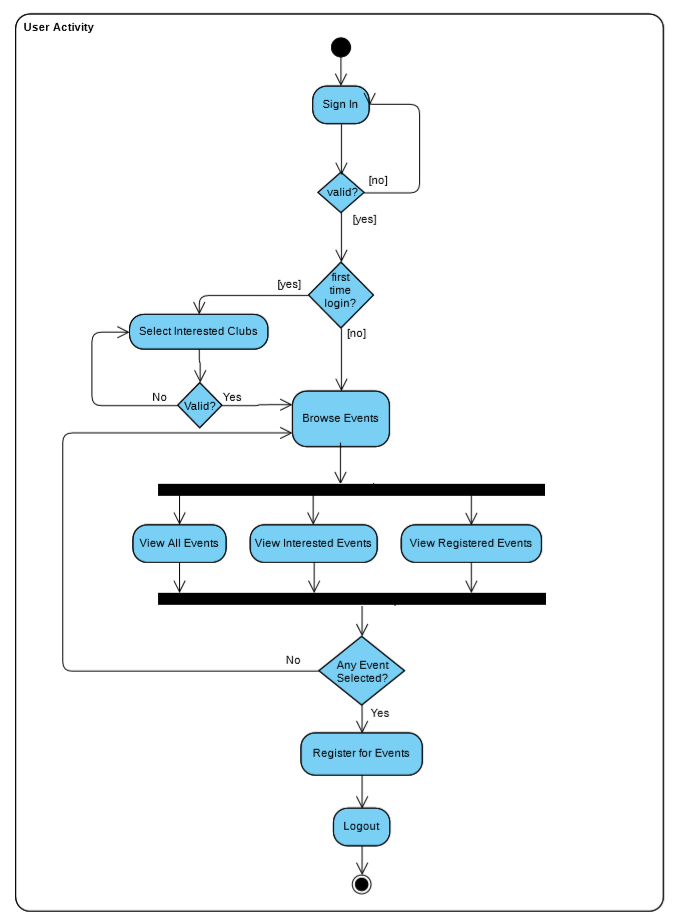
# 

# 8. State Diagram

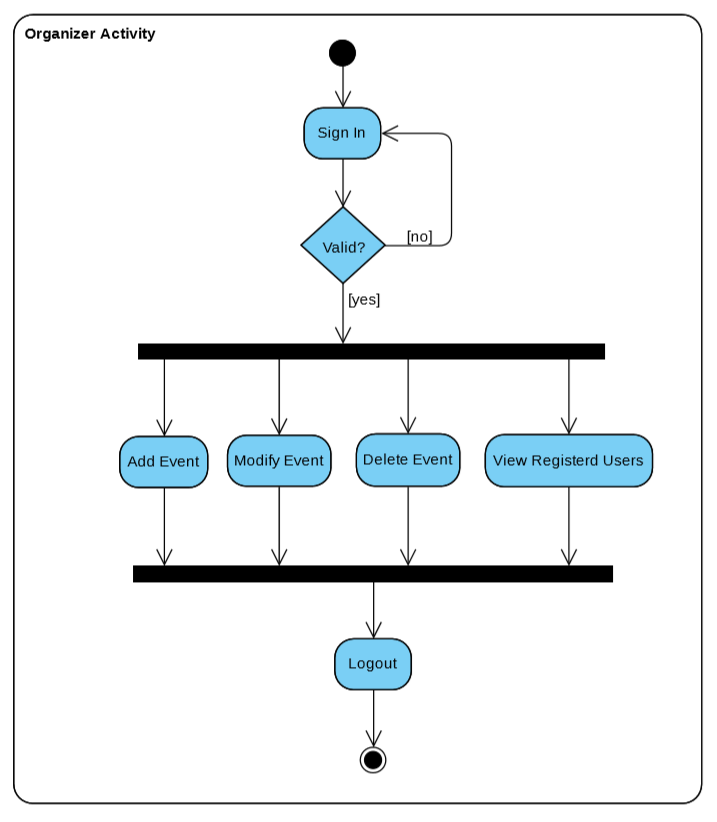


# 9. Activity Diagram

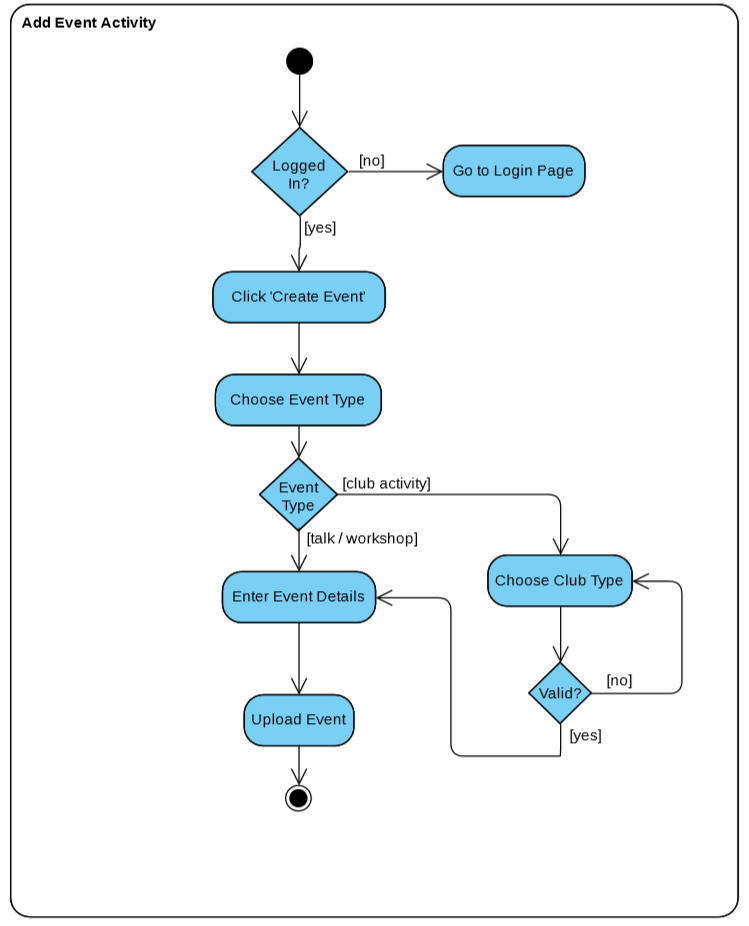
## 9.1 User Activity Diagram

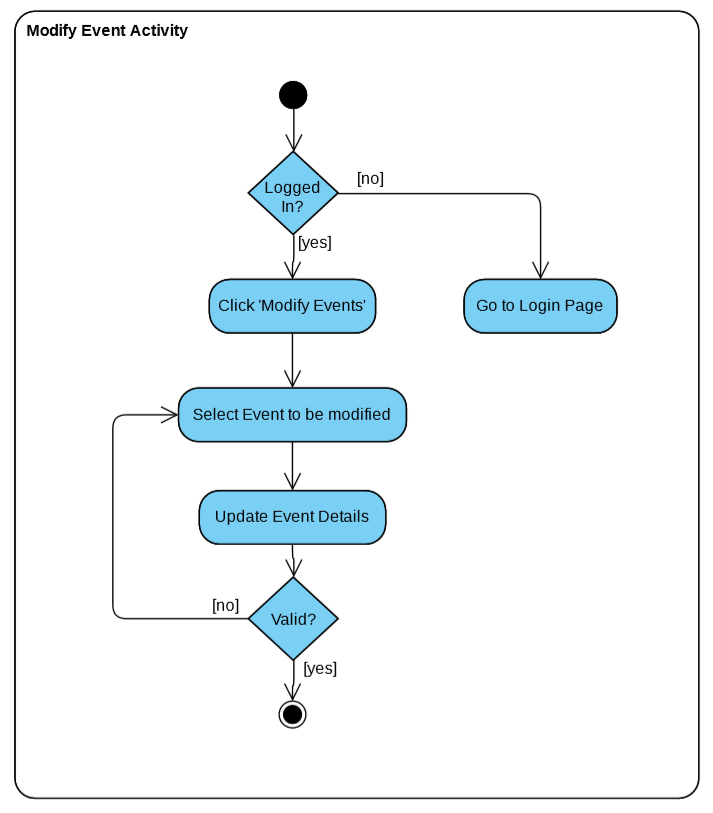


## 9.2 Organizer Activity Diagram

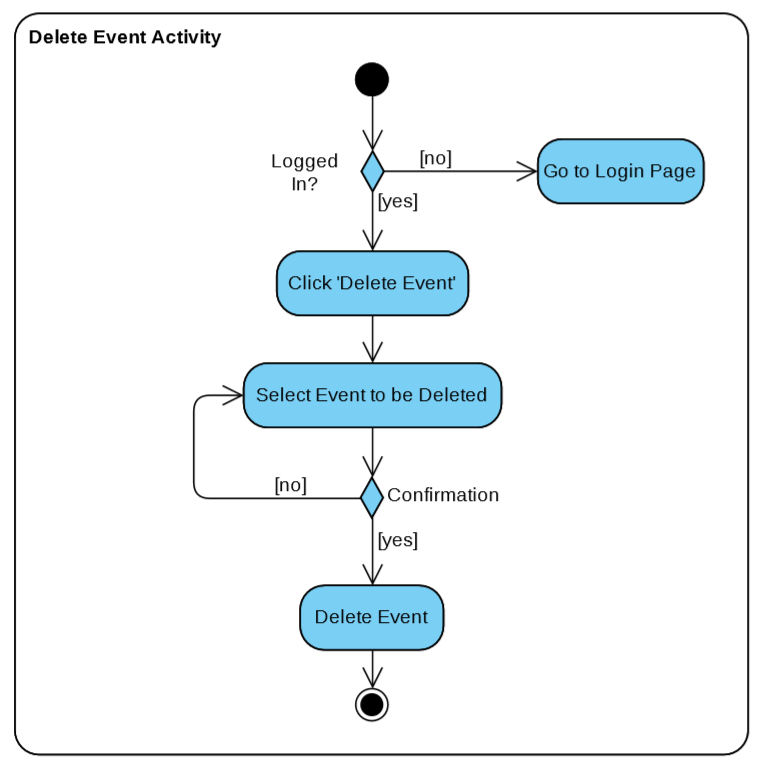


## 9.3 Add Event Activity Diagram



9.4 Modify Event Activity Diagram 

9.5 Delete Event Activity Diagram



# 10. Summary

The SRS provides overall description and an overview of the system functionality and system interaction with other systems. It also describes the probable user interfaces for the better understanding of the appearance of the system to the end users and stakeholders. It also provides the requirements specification in detailed terms and a description of the different system interfaces. Different specification techniques are used in order to specify the requirements more precisely for different audiences. It also depicts the system with user case scenario - use case diagrams, shows how data in the system will flow with - data flow diagram, changing of states due to action of events - using state diagram, relationship among entities of the system for database - Entity Relationship diagram and Flow of activities using- Activity diagram. The SRS document helps the developer to gain better understanding of the system and a strong reference point for the developing and implementation stage of the project.