ASA HUB

By

Hassan Farooqi

Supervised By

Dr. Muhammad Usman

Department of computer sciences

Quaid-i-Azam University

Islamabad

2013-2017


بسم الله الرحمن الرحيم

QUAID-I-AZAM UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

Dated: _____

FINAL APPROVAL

This is to certify that we have read the final report submitted by Hassan Farooqi it is our judgment that this report is of sufficient standard to warrant its acceptance by the Quaid-i-Azam University, Islamabad for the degree of the Bachelor in Computer Science (BSCS).

Committee

EXTERNAL EXAMINER _____

SUPERVISOR _____

Dr. Muhammad Usman

Department of Computer Science

Quaid-i-Azam University, Islamabad

INCHARGE _____

Dr. Onaiza Maqbool

Department of Computer Science

Quaid-i-Azam University, Islamabad

ACKNOWLEDGMNET

In the name of ALLAH, Most Beneficent, Most Merciful.

First of all, 1 thanks to Almighty Allah because of His kindness and grace for completion of my project.

I would like to express my deepest gratitude to all those who provided me the possibility to complete this project. A special gratitude I give to my respected teacher and supervisor Dr. Muhammad Usman, whose guidance and kindness helped me to coordinate my project. Furthermore, I would like to thanks to all faculty members of computer science department for their help, giving proper attention and time.

I would like to express my gratitude to my parents and all family members specially my elder brother Muhammad Umar Farooq for his kind cooperation and encouragement, which helped me throughout my life.

My thanks and appreciation also goes to my class fellows and friends, who have willingly helped me out with their abilities.

Hassan Farooqi

2013-2017

ABSTRACT

ASA is the democratic body of Quaid-i-Azam University, which works for the betterment of the academic staff community. To achieve the basic objective of the association is very difficult to do in a manual way. The work of the association is currently being done manually. The basic problem faced by the association is managing the notifications. Different notifications are to be sent by President, Secretary and other office holders of the association. It is difficult to maintain the record of ASA and account details. The ASA members have no proper platform for discussing the topics related to ASA. No such system exists to reduce the burden of managing the whole workflow of the association.

Therefore system has been designed and develops which provides services, namely, notification management, a discussion forum for discussing topics related to ASA, members account management facility, and online or virtual meeting service. This system is a web-based application, therefore, the system has been implemented in ASP.NET 4.5 along with MS SQL Server Management server database. In order to access the system, users should have specific usernames and passwords. No external hardware is required to maintain the value of the system. The possible outcomes of the system that is would be time-saving and reduce the burden of manual work. Notification of ASA will be well managed and overcomes the burden of keep records. Yearly expenditure report of ASA will be managed and it also overcomes the meeting problem through online meeting facility and members will have a proper forum for discussing issues related to the association.

The system has a feature of maintainability in order to meet the future works. This system is for only for QAU limited to one organization and in future work, it can be extended to other organizations as well.

Table of Contents

List of Figure

List of Table

"Education is the best provision for the journey to old age" Aristotle (384-322 B.C)

Chapter 1

Software Project Management Plan

This chapter first introduces the Academic Staff Association (ASA). It then highlights the problem that has been addressed in this work along with the designed and developed solution. It also elaborates project organization and project planning. Finally, this chapter elucidates the scope and objectives of this project.

1.1 Problem Definition

ASA is known as Quaid-i-Azam University Academic Staff Association. The Vice-Chancellor is the patron-in-chief of the association. The basic objective of ASA is to promote social, educational and cultural activities, to collaborate with other organizations in furthering the interest of education. ASA works for to promote the interest of members and to organize discussions and lectures. To undertake such other activities as may be determined by the association to achieve its objective. All full-time salaried teachers of the University are eligible to be the members of the association. The members are required to pay a monthly subscription to be deducted from their salaries. The amount of monthly subscription is determined by the General body. The general body of the association is the supreme body of the association. The office bearers of the association are President, Vice-President, Secretary, Joint Secretary and Treasurer.

The "ASA HUB" project is a group project and has been assigned to a group of two students. The component of the project is based on modules, namely, members profile management, discussion forum, virtual/ online meetings, electronic-voting, meetings record (minutes) management, notification management and account management. From these components notification management, discussion forum, members account management and online/virtual meetings modules are assigned to me. In order to meet the objective, the association has to do every work manually. It is difficult to manage the notification and maintains the record of all notifications. Members of the association have to discuss the important issue as it is impossible to engage every member due to lack of discussion forum. Every member of the association cannot attend the every meeting therefore due to engagement in different academic activities. Association is facing difficulty in maintaining the record of distribution of budget and account details.

1.2 Proposed Solution

The system has been designed and developed for Academic Staff Association of QAU, named as, "ASA HUB", which is based on a web-based application for the ASA members, for the admin and other privileged users. The ASA HUB portal will facilitate the elected body of ASA and its members in a number of ways by providing services such as discussion forum in which member can discuss academic activities and other activities determined by ASA to achieve its objective, virtual/ online meetings service through which member can attend the meeting online, notification management facility through which members can be up to date about the upcoming events through notification, maintaining budget records and account management facility.

1.3 Scope

Scope of the system defines the boundaries of the system that what will the software deliver to users. [1] The scope is more towards the functional requirements, therefore, the system provides major functionality such as discussion forum, virtual/online meeting, notification management, and members account management. The system will allow the users to perform all the activities of ASA. The system will enable users to openly discuss and make comment on any topic related to the association in the form discussion forum facility. The system will provide online or virtual meeting service that enables users to join the meeting through internet. The system will provide notification management facility that makes users manage the notifications of the association. The user will view the all notifications of the association. The system will enable users to maintain records of whole budget of the association through account management facility. The user can view the account details of whole yearly budget of the association. The users will have specific username and password in order to access these services. This system is limited to faculty members of QAU, therefore, stakeholders are members of ASA. Only members of ASA can use the system.

1.4 Objectives

The primary objective of the systems is to facilitate the ASA. This system overcomes the burden of manual works. The users will have separate accounts and through specific accounts, a user can able to access the system. Through discussion forum facility user can discuss any issue related to other users of association. It would be time-saving and easily manageable. Virtual or online meeting facility overcomes the burden of arranging meeting and user can discuss important issues related to the association in order to achieve its objectives. Account management facility maintains the whole record of account and budget details of association. Notification can be managed easily and overcome the burden of saving records of all notifications of the association.

1.5 Project Organization

Project organization is a structure that facilitates the coordination and implementation of project activities. Its main reason is to create an environment that fosters interactions among the team members with a minimum amount of disruptions, overlaps, and conflict. Project organization is basically how the project is organized according to software engineering. Which process model is going to be followed and what are major roles and responsibilities. Finally which tools and techniques will be used in order to develop this system.

1.5.1 Software Process Model

The software process model is an abstract representation of a software process. It presents a description of a process from some particular perspective. The software process model is followed is Rapid Application Development (RAD) Model [2]. The reason behind choosing this model is when there are deadlines, requirements are well understood and business application can be modularized, it is reasonable to use this approach. This approach is quite suitable as compared to other approaches of software process model as waterfall model is used when requirements are well understood and this model does not follow modular approach. The incremental approach is used when requirements are well understood and increments can easily be defined. Evolutionary prototyping model is used when requirements are not well understood. Evolutionary spiral model is used when risk is high and needs to be resolved. The approach of component-based development follows when quality components are available.
RAD is basically a high-speed adaptation of the linear sequential model. The basic steps followed in this model are; communication, planning, modelling, construction and deployment. The project is divided into two modules and two modules further divided into two more modules one is documentation and other is construction and deployment. RAD model enables rapid delivery as it reduces the overall development time due to reusability of the components and parallel development. Only system that can be modularized can be built using RAD. The advantages of using this approach are; Delivers fully functional systems in very short periods it requires sufficient human resources and commitment, not all applications can be modularized and may not be appropriate with technical risks high (new technology). The software process model is an abstract representation of a software process. It presents a description of a process from some particular perspective.

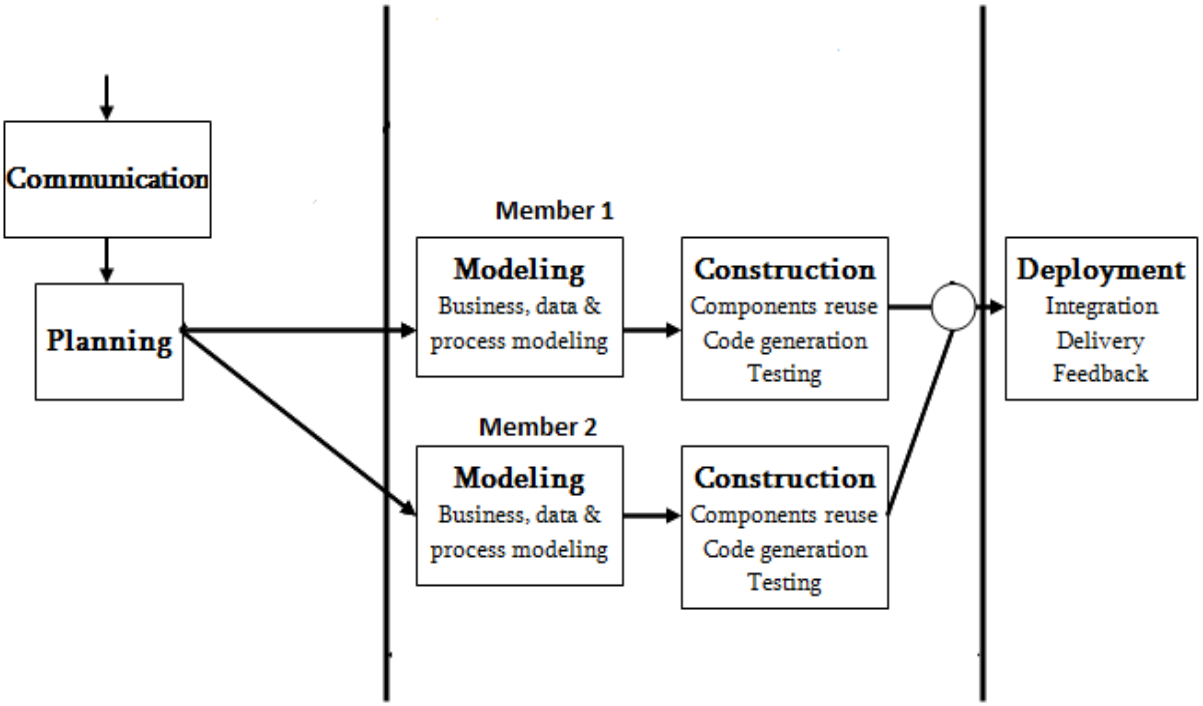

Figure 1.1 The Process Model

1.5.2 Roles and Responsibilities

The "ASA HUB" is basically a group project, therefore, roles and responsibilities are divided into a group of two students. The project contains two modules; one module contains discussion forum, notification management, online or virtual meeting and account management of the association. Second modules contain profile management, meeting records, and electronic voting.

One module is assigned to one student and the second module is assigned to another student. Following are role and responsibility on the basis of modules. Better communication with supervisor for a clear understanding of requirements. Meet the stakeholders for requirement gathering. After the complete understanding of system design a plan in order to meet the requirements. Make sure of the availability of all stakeholders for refinement and for getting updates regarding system and requirements. The supervisor has very important role in refinements of requirement and testing of the system according to given requirements. Implementation of the project according to requirements and verify that system. Test the system and finally deployment of that system for the end users. The following is the role and responsibilities of both students are; project plan, requirements specification, analysis, architecture specifications, component or object specification, source code, test Plan and final deliverable.

This system is developed for ASA QAU so all faculty members can access this system. In this one is admin and admin, in this case, will be secretary of the ASA and some other

privileged user as well. Privileged users have some other rights than the normal user. Secretary of ASA has all rights of the system. Secretary will manage the whole records of the meeting, notification management and other services of ASA as well. Privileged users have some special rights for example in the case of conducting election Librarian of the QAU has right to conduct the electronic voting for ASA.

1.5.3 Tools and Techniques

Tools that are using for implementation of this system are visual studio 2013 and SQL server database. Argo UML tool and Microsoft Visio for UML diagrams such as use case diagram, class diagram, activity diagram, domain model and Entity relationship diagram and for writing documentation Microsoft word are used. For designing a plan of the system, project libre is used.

This system will be implemented in language.NET framework asp.net [Active Server Page]. ASP.NET framework 4.5 will be used in order to fully implement this system. CSS will be used as cascading style sheets to style the contents. Bootstrap, JavaScript, JQuery will also be used.

1.6 Project Management Plan

Software project planning is task which is performed before the production of software actually starts. It is there for the software production but involves no concrete activity that has any direction connection with software production; rather it is a set of multiple processes, which facilitates software production.

	Documentation	**77 days**	**9/20/16 8:00 AM**	**1/4/17 5:00 PM**
	Chapter 1:Project Introduction	**10 days**	**9/20/16 8:00 AM**	**10/3/16 5:00 PM**
	Introduction	2 days	9/20/16 8:00 AM	9/21/16 5:00 PM
	Problem Definition	1 day	9/21/16 8:00 AM	9/21/16 5:00 PM
	Proposed Solution	1 day	9/22/16 8:00 AM	9/22/16 5:00 PM
	Scope	1 day	9/23/16 8:00 AM	9/23/16 5:00 PM
	Objective	1 day	9/24/16 8:00 AM	9/26/16 5:00 PM
	Project Organization	1 day	9/27/16 8:00 AM	9/27/16 5:00 PM
	Project Management Plan	4 days	9/28/16 8:00 AM	10/3/16 5:00 PM

Chapter 2:Requirements Gathering and Analysis	**20 days**	**10/4/16 8:00 AM**	**10/31/16 5:00 PM**
Introduction	**3.25 days**	**10/4/16 8:00 AM**	**10/7/16 10:00 AM**
Purpose	0.25 days	10/4/16 8:00 AM	10/4/16 10:00 AM
Stakehoders	0.25 days	10/5/16 8:00 AM	10/5/16 10:00 AM
Major Functions	0.25 days	10/6/16 8:00 AM	10/6/16 10:00 AM
Supported Functions	0.25 days	10/7/16 8:00 AM	10/7/16 10:00 AM
Major Inputs and Outputs	0.25 days	10/7/16 8:00 AM	10/7/16 10:00 AM
Overview	**3.25 days**	**10/10/16 8:00 AM**	**10/13/16 10:00 AM**
Overall Discriptions	0.25 days	10/10/16 8:00 AM	10/10/16 10:00 AM
Product Perspective	0.25 days	10/11/16 8:00 AM	10/11/16 10:00 AM
Product Functions	0.25 days	10/11/16 8:00 AM	10/11/16 10:00 AM
User Charactristics	0.125 days	10/11/16 8:00 AM	10/11/16 9:00 AM
Constaints	0.25 days	10/12/16 8:00 AM	10/12/16 10:00 AM
Assumptions and Dependencies	0.25 days	10/13/16 8:00 AM	10/13/16 10:00 AM
Specific Requirements	**12 days**	**10/14/16 8:00 AM**	**10/31/16 5:00 PM**
Functional Requirements	1 day	10/14/16 8:00 AM	10/14/16 5:00 PM
Non Functional Requirements	1 day	10/17/16 8:00 AM	10/17/16 5:00 PM
Use Case Diagram	2 days	10/18/16 8:00 AM	10/19/16 5:00 PM
Use Cases Description	8 days	10/20/16 8:00 AM	10/31/16 5:00 PM
Chapter 3:System Design	**18 days**	**11/10/16 8:00 AM**	**12/5/16 5:00 PM**
Architectural Diagram	1 day	11/10/16 8:00 AM	11/10/16 5:00 PM
Entity Relationship Diagram	5 days	11/11/16 8:00 AM	11/17/16 5:00 PM
Class Diagram	2 days	11/18/16 8:00 AM	11/21/16 5:00 PM
Sequence Diagrams	10 days	11/22/16 8:00 AM	12/5/16 5:00 PM
Chapter 4: Implementation	**5 days**	**12/5/16 8:00 AM**	**12/9/16 5:00 PM**
Introduction	1 day	12/5/16 8:00 AM	12/5/16 5:00 PM
Language Selection	1 day	12/5/16 8:00 AM	12/5/16 5:00 PM
Database Selection	1 day	12/5/16 8:00 AM	12/5/16 5:00 PM
Interfaces	4 days	12/6/16 8:00 AM	12/9/16 5:00 PM
Chapter 5:Testing	**5 days**	**12/12/16 8:00 AM**	**12/16/16 5:00 PM**
Test cases	5 days	12/12/16 8:00 AM	12/16/16 5:00 PM
Chapter 6	1 day	12/19/16 8:00 AM	12/19/16 5:00 PM
Conclusion and Future Work	2 days	12/20/16 8:00 AM	12/21/16 5:00 PM
Review	**10 days**	**12/22/16 8:00 AM**	**1/4/17 5:00 PM**
Review 1	5 days	12/22/16 8:00 AM	12/28/16 5:00 PM
Review 2	5 days	12/29/16 8:00 AM	1/4/17 5:00 PM

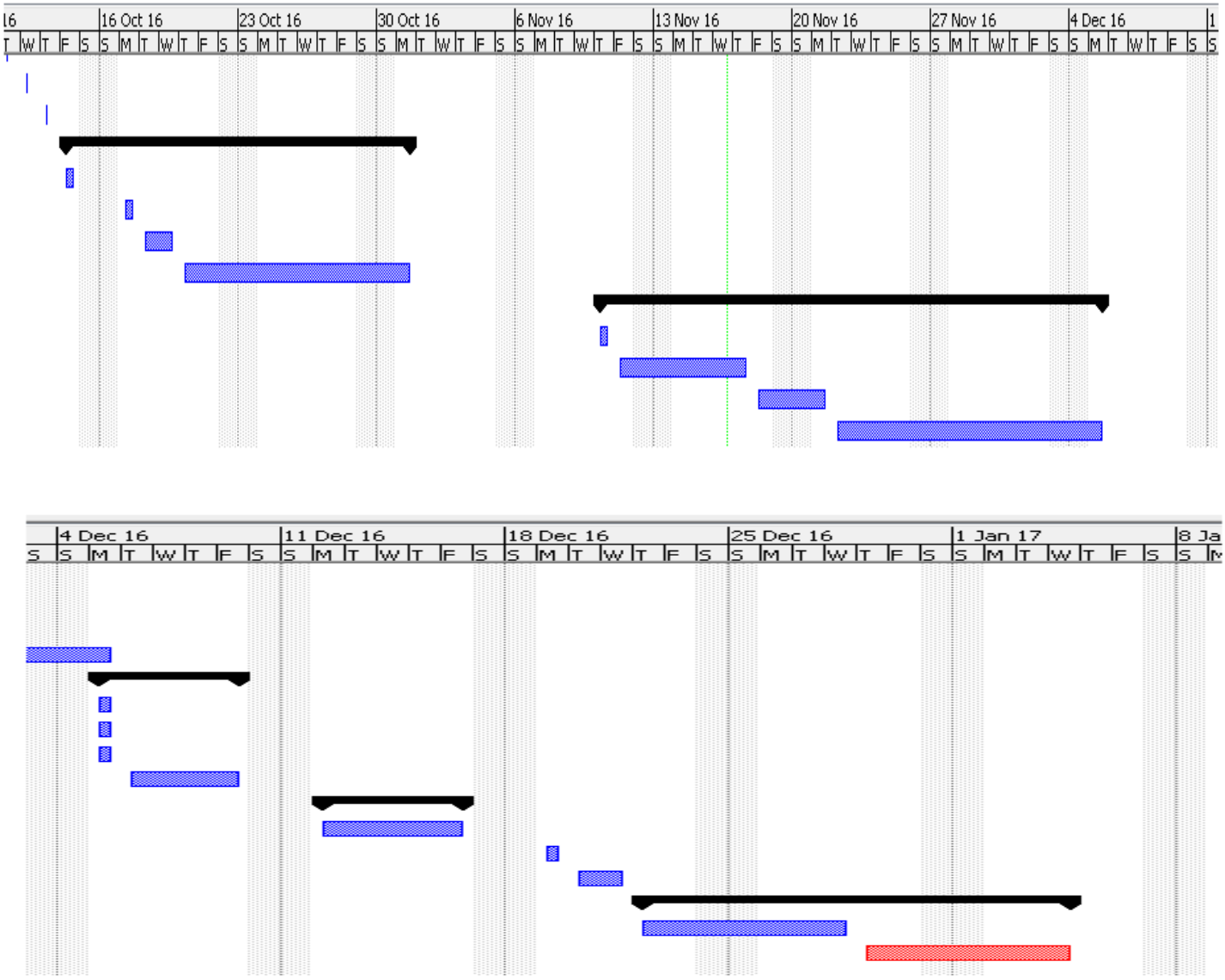

Figure 1.2 Project Management Plan

1.7) Report Structure

Chapter 1 briefly describes the introduction of the system, what are actual problem and proposed solution, its scope, objective, describe the organization of project and project management plan.

Chapter 2 describes the requirements of the system and to decide what the system should do and what the system should not do. Briefly describes complete details of functional and non-functional requirements specifications for the system.

Chapter 3 gives the description of software design. It entails the description about the chosen architecture design, user interface design. Briefly describes the interaction and relation between the between actor and system objects through sequence and class diagram.

Chapter 4 gives the description related to system implementation. Briefly describes how systems is implemented, which system framework in used. Also entails the description related to language selection, software used and screen images.

Chapter 5 gives the description of the testing of the system. Briefly describe the test approaches, test plan, testing tool and environment and finally the test cases of the system.

Chapter 6 contains the conclusion and future work related to this system. Overall conclusion and future enhancement what can be further implemented or improvements in this application.

In this chapter basic introduction of the system is described in detail. Basic problem and proposed solution to that problem as well. A complete overview of system including the scope of the system and possible outcomes of the system. This chapter covers the basic introduction and requirement gathering and analysis will be discussed in next chapter.

"You can never be overdressed or overeducated" Oscar Wilde (1854-1900)

Chapter 2

Software Requirements Specifications

The purpose of this requirement gathering and analysis is to clear the requirements of the system and to decide what the system should do and what the system should not do. To clear the requirements like function and non-functional requirements for the system and understand the major inputs and outputs for the system.

2.1 Product Overview

ASA HUB" is system for ASA of QAU. This system is basically a web-based application. It can run on every system. This system will be implemented in ASP.NET. The user can use this system for different purposes regarding association. Admin (Secretary of the Organization) will manage the whole system. Admin can manage the whole record of association as well. The user will be able to use this system in order to discuss different topic regarding academic and other matters related to the association. This product will enable the user to attend the online meeting from anywhere and will provide different services wiz, managing notifications, view notifications and association account management.

2.2 Major Functions

Sign up users to the system and sending registration number and password to the user through email when user is registered. Manage the whole notifications such as add notification. Manage discussion forum such as add topic for discussion and can comment on that topic. Manage the account details of the association and also providing a facility of attending an online meeting to users from anywhere.

2.3 Supported Functions

Users and admin can perform the different queries regarding notification management. User and admin can view the old notification, view account details from the specific period.

2.4 Major Inputs and outputs

The user gives some inputs to system and system will generate the response to those inputs respectively.

2.4.1 Major Inputs

Admin sign up the users and user enters the specific username and password to log into system. Management of notification from the user, adds account detail from the treasurer, manage

discussion forum and manage meeting virtually. The user can request to view the notification, add notification. The user can also perform different action regarding discussion forum and meetings respectively. The user can request to view the account details as well. The user can request to view the old record of ASA in the form of notification, discussion or meeting.

2.4.2 Major Outputs

User registered to the system. The system gives response to specific queries of users. Information regarding to user added to system. Different acknowledge will be generated for different input. User request for account details for last five years, one or two years then record of the specific year will be displayed to the user.

2.5 Definitions, Acronyms and Abbreviations

Table 2.1 Abbreviations

ASA Hub	Academic Staff Association Hub
QAU	Quaid-i-Azam university
User	All members of ASA
Admin	Secretary of the Association
Treasurer	Account manager of the Association

2.6 Overview

The rest of topics contain the detailed information about functional, non-functional, performance requirements, the overall functionality of the system, use cases and their description, domain model, and ER diagram.

2.7 User Characteristics

Users of this system are all the members of the association. This is assumed that all the users have basic knowledge of computer or laptop and knowledge of web application. Users and admin must have knowledge of how to use web-based applications and can able to perform certain tasks. Admin must be familiar with the web-based system and understands the basic working of the system.

2.8 Constraints

User should any mobile device, smart phone, computer or laptop and internet connection to access the system.

2.9 Assumptions and Dependencies

This web-based application is depends on the availability of internet. It is assumed that the user and admin both have any computer or laptop or smartphone devices to access the system.

2.10 Specific Requirements

2.10.1 Functional Requirements

Functional requirements are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. The functional requirements are that the system should allow only to admin managing the whole system. Only admin can make signup the new user and assign them specific username and password. The system allows user to access the system through proper username and password. The system gives the proper message of any invalid entry. The system allows admin to manage notifications, manage the account details. The system enables users to view notifications, and to view account details. The system allows user to join meeting virtual or online through specific IP address and give the valid message over invalid entries.

2.10.2 External Interface Requirement

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

2.10.3 User Interfaces

Through user interface user will be able to interact with the system. This will be web-based application for the user. User can use this application through internet. User would have unique username and password to access this application.

2.10.4 Software Interfaces

ASA HUB is web-based application and it will be implemented in ASP.NET, therefore, this system will be run on any operation system. The internet is required to access the system. The system can access through any internet browser.

2.10.5 Communication Protocol

Communication protocols required for this system are; Hypertext transfer protocol (http) for communication over the internet and Simple Mail Transfer Protocol (SMTP) is an internet standard for electronic mail transmission.

2.11 Software System Attributes

2.11.1 Reliability

System should be reliable. There should be no occurrence of the failure. The system should be able to work properly all-time means the extent to which it works as and when needed. The system should give the proper response to every query perform by user. The system is approximately ninety one percent reliable.

2.11.2 Availability

System should be available to every user at any time. All the members of the associations are able to access the system at any time.

2.11.3 Security

Since this system will be hosted on server, all the user data will be kept on that server. Product should be able to protect privacy of user data. User should only be accessed the system through user own credentials and any other user should not be able to access to the user private data. All input need to be encoded and validated to prevent SQL injection. User can only perform operation under the permission. Some user groups can be configured that they can never have certain permission.

2.11.4 Maintainability

There should be aspect of maintainability for the system. In some cases, maintainability involves a system of continuous improvement learning from the past in order to improve the ability to maintain systems, or improve reliability of systems based on maintenance experience. The application should be easy to extend. The code should be written in a way that it favours implementation of new functions. In order for future functions to be implemented easily to the application.

2.11.5 Portability

This is web-based system therefore main purpose of developing web-based system is to improve the portability of system. To improve portability, system should run on variety of platforms and variety of connection speeds. The system should be lightweight therefore that it can run on a machine with slow internet connection. To make the web application lightweight, simple libraries and tools should be used at developing phase.

2.11.6 Performance

Since this system is going to web-based, it does require a powerful server machine with high band internet access. Server machine should have a powerful CPU and high speed internet access therefore that it can handle multiple users at the same time. Another performance requirement is the storage space. Higher storage space means more user and bigger workspace per user therefore higher the storage, better the performance. Performance requirement by the user side is, web application should be developed as a lightweight web app therefore that it can work on almost any platform even with slower internet connections. Expected number of simultaneous user should be at least 200. System should be able to deal with at least 200 users at the same time. Also database of the system should handle at least a five hundred of users at any periods and response to any query with in limited time.

2.12 Product Functions

2.12.1 Manage User Information

Users that are member of ASA are to be registered to the system. Admin then register the user to the system after verifying its form data and user's account's registration number and password will be sent to the user by email. After getting password, user is eligible to login through its web application. Admin can update, search and view all records of the users.

2.12.2 Add Notifications

Admin will add the notification data like notification from, date, notification to the system. Admin can also update and view all records of notification. Only admin can add or send notifications.

2.12.3 View Notifications

User can view the notifications that are to send or add by admin, User can only view the notification as user cannot delete notification.

2.12.4 Add Topic

User can add topic for open discussion related to association.

2.12.5 Leave Comment

User can give comment of added topic. User can add his/her opinion of respective topic regrading ASA.

2.12.5 Join Meeting

User can attend the online meeting through this system. Only authorized user can access the system and join the meeting from anywhere. President of the association first join the meeting then other user can join the meeting. In this case the president system is work as server and through the IP address of the system other user can join the meeting. Before starting the meeting admin will send a unique four digit token to the respective users, in order to enable them to join the meeting. User will enter the unique token if the token is valid then user will able to join the meeting.

2.12.7 Leave Meeting

After the meeting is over user can leave the meeting, therefore user sign out from the online meeting.

2.12.8 Manage Account

The treasurer will be responsible of managing the account. He will be responsible for framing the budget. The treasurer will exercise the supervision on the income and expenditure of the association and manage the account of all users.

2.12.9 Request to view Account Details

User can view the whole account details and system will display the account details of particular month, years or any selected period.

2.12.10 Request for login

User enters its registration number and password to use the application. Only authentic users will be able to use the application.

2.12.11 Request for log out

User clicks on logout button and user log out from the system

2.13 Use Case Diagram

Use-case diagram is a graphical representation of user's interaction with the system. Use-case diagram can portray the different types of users of the system and ways that they interact with system. [3]

List of use-cases mentioned in use-case diagram are described in detail, therefore that we are able to look more precisely that how user can interact with system to perform tasks. As this application is multiple users based and there are two main types of users.

Administrator: Administrator manages and analyses all information regarding ASA.

User: all members of ASA.

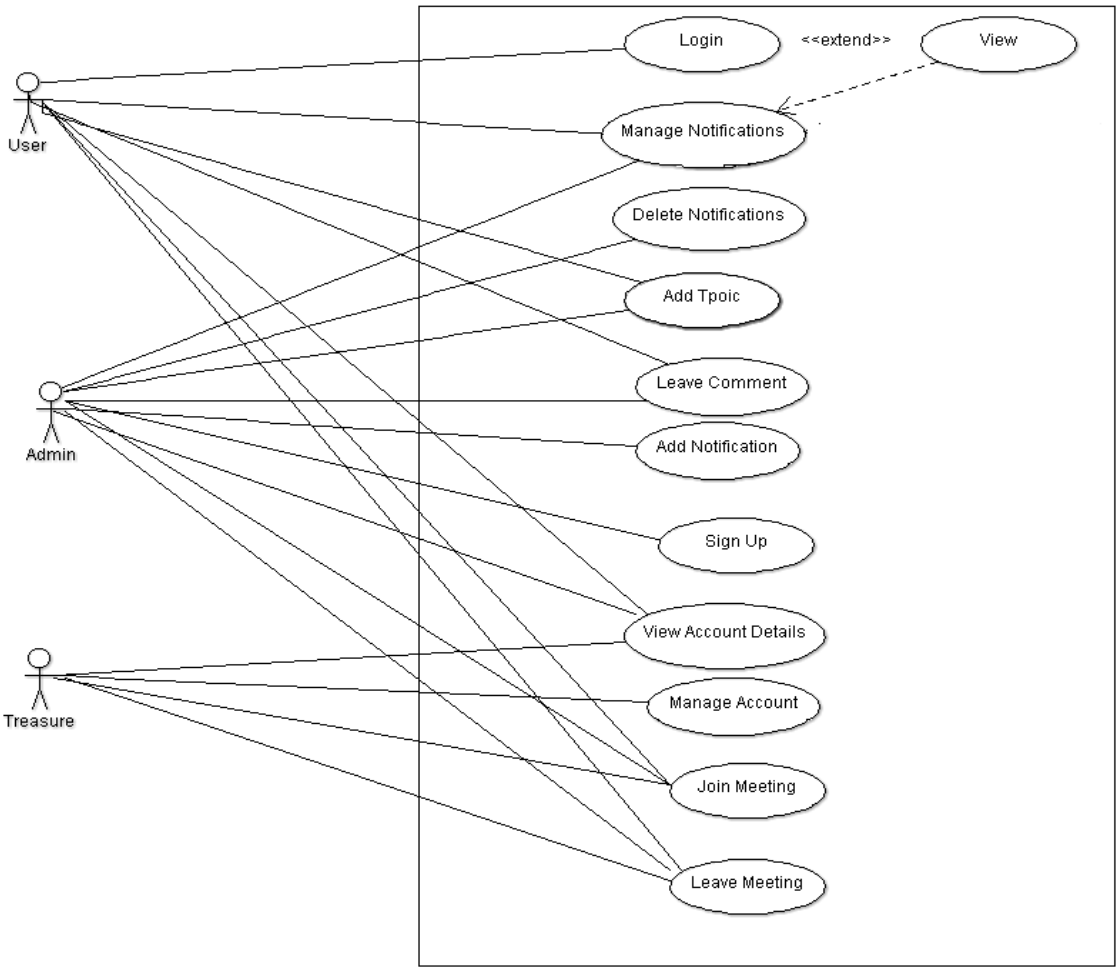

Figure 2.1 Use Case Diagram

2.14 Use Case Description

2.14.1 Use case 1: Admin Sign up

ID	**UC1**
Name	Admin Sign up
Primary Actor	Admin
Pre-Conditions	i. Admin must be member of association ii. Admin should on login screen
Post-Conditions	i. Information of the Admin will be stored in the system ii. Admin can logged in to the system through proper username and password
Main Success Scenario	1. Admin enters the data in given fields (username, password, email) for sign up 2. Admin clicks on sign up button. 3. Information related to Admin will be added to system
Alternative flows or Extensions	1a**.** If admin enters the email that is already sign up 1. System gives message of already sign up. 2a**.** If admin clicks on Sign up button without entering user name and password 1. System gives error message asks user to enter valid user name and password.
Frequency	Could be nearly continuous

In order to access the system admin first sign up to the system, then admin got specific username and password through that username and password admin can access the whole system. Admin will be registered to the system and can perform the different functions related to ASA. In order to access the system admin enters the required data to given field like first name, last name, email and department. Appropriate message or flows will be shown if there is an error. This use case is to make access of the user to the system.

2.14.2 Use case 2: User Login

ID	**UC2**
Name	Login
Primary Actor	User
Pre-Conditions	i. User is registered and its information is saved in the system ii. user is on log in screen
Post-Conditions	i. User log in to system ii. Home screen will be displayed to user
Main Success Scenario	1. User enters user name and password. 2. User clicks on login button. 3. User log in to system 4. Home screen will be displayed to user
Alternative flows or Extensions	1a**.** if user enters the invalid user name 1. System asks user to enter the user name again. 1b**.** if user enters the invalid password. 2. System signals invalid password and ask to re-enter password.
Frequency	Could be nearly continuous

For user login use case only the members of ASA have access to the system, therefore through proper username and password member can access the system. In order to use application members must have login to the system. Only registered user can logged in to the system. This use case for users in order to access the system

2.14.3 Use case 3: User Logout

ID	**UC3**
Name	Logout
Primary Actor	User
Pre-Conditions	User must log in to system
Post-Conditions	User log out to system
Main Success Scenario	1. System displays log out option. 2. User selects the option. 3. User is logged out of the system.

Alternative flows or Extensions	None
Frequency	Could be nearly continuous

For the sake of security and maintain privacy user log out to the system. After using the system user can log out to the system. Once the user logged out to the system then for accessing the system user must login to the system by entering username and password.

2.14.4 Use case 4: Add Topic

ID	**UC4**
Name	Add Topic
Primary Actor	User
Pre-Conditions	User must login to system
Post-Conditions	Topic for discussion will be added to system
Main Success Scenario	1. System display multiple option to the user 2. User select the Add topic option 3. Input field like text area is open for entering topic for discussion and user click on post button to add topic 4. Topic is added to system 5. User can view the updated information
Alternative flows or Extensions	3. a) If user clicks on post button without entering the topic or text in text area 1. System gives error message and asks user to add text/topic in given field
Frequency	Could be nearly continuous

ASA portal provides a discussion forum in order to discuss the topics related to ASA therefore for these purpose users first add the topic in add topic section. Any registered user can add topic for discussion. User follow the some step and respective message will be displayed in case any error. This use case will be used in discussion forum module.

2.14.5 Use case 5: Leave Comment

ID	UC5
Name	Leave Comment
Primary Actor	User
Pre-Conditions	User must login to system
Post-Conditions	Comment for respective topic will be posted/ added
Main Success Scenario	1. User select the Comment option 2. Input field like text area is open for entering comment on respective topic for discussion and user click on post button to add comment 3. Comment is added to system 4. User can view the updated information
Alternative flows or Extensions	None
Frequency	Could be nearly continuous

For giving comments of posted topic user can make comments on it. User can give his./her opinion on any topic that are posted for discussion. User first log in to the system then user can comment of respective topics. Comment will be saved in to the system and other user can also made comment on it. Proper message will also be displayed in case of error or invalid entry. This use case will be used discussion forum module.

2.14.6 Use case 6: Add Notification

ID	UC6
Name	Add Notification
Primary Actor	Admin
Pre-Conditions	Admin must login to system
Post-Conditions	Notification will be added
Main Success Scenario	1. System display multiple option to the Admin 2. Admin selects the Add Notification option 3. Input field like text area is open for entering the notification and user click on send button to add Notification 4. Notification is added to system and notification sent through mail to all members 5. User and Admin can view the updated information and notification

Alternative flows or Extensions	3. a) If user clicks on send button without entering the Notification or text in text area 1. System gives error message and asks user to add text/Notification in given field
Frequency	Could be nearly continuous

Admin have only access to add the notification, Admin logged in to the system. System displays the multiple option to the admin and admin select the add notification and add the notification details respectively. Notification will be send to all or specific members through email and also stored in the system, User can view the notification as well. This use case is used in notification management system.

2.14.7 Use case 7: View Notification

ID	**UC7**
Name	View Notification
Primary Actor	User
Pre-Conditions	User must login to system
Post-Conditions	All notification will be shown
Main Success Scenario	1. System display multiple option to the user 2. User select the View Notification option 3. System displayed all the notification to the user if notifications are posted.
Alternative flows or Extensions	None
Frequency	Could be nearly continuous

As notification is added by the admin, now user can view the notification as well. Latest notification will be on the top when user views the notifications. All the added notifications are displayed by the system to users. This use case will be used in notification management system.

2.14.8 Use case 8: Login

ID	UC8
Name	Login
Primary Actor	Admin
Pre-Conditions	Admin is registered in the system
Post-Conditions	Admin log in to system Home screen will be displayed to user
Main Success Scenario	1. Admin enters user name 2. Admin enters password. 3. Admin clicks on login button. 4. Admin log in to system 5. Home screen will be displayed to user
Alternative flows or Extensions	1a**.** if user enters the invalid user name 1. System asks user to enter the user name again. 1b**.** if admin enters the invalid password. 2. System signals invalid password and ask to re-enter password.
Frequency	Could be nearly continuous

In order to access the system, users have to login to the system. User enters the username and password to access the system respectively. Proper message will be displayed to the user in case of invalid entry or data like incorrect username or password. This use case is for admin to access the system.

2.14.9 Use case 9: Logout

ID	UC9
Name	Logout
Primary Actor	Admin
Pre-Conditions	Admin is registered in the system
Post-Conditions	Admin log in to system Home screen will be displayed to user
Main Success Scenario	1. System displays log out option. 2. Admin selects the option. 3. Admin is logged out of the system.
Alternative flows or Extensions	None

Frequency	Could be nearly continuous

When user logged in to the system can also log out from the system. User clicks on logout button and log out from the system. For further accessing the system user must enter username and password to login. This use case for admin to log out from the system

2.14.10 Use case 10: Join Meeting

ID	**UC10**
Name	Join Meeting
Primary Actor	User
Pre-Conditions	i. User is logged in to system ii. President must join the meeting
Post-Conditions	User will be joined the meeting.
Main Success Scenario	1. System displays the option of join meeting. 2. User joins the meeting online
Alternative flows or Extensions	2a**.** if meeting is not being president then member should unless president is online to preside the meeting.
Frequency	Could be nearly continuous during meeting time

In order to join the meeting online user first log in to the system. System displays the option of join meeting and through which user can join the meeting. User has to enter the given token for joining the meeting. President first joins the meeting then other members of association can join the meeting. Proper message will be displayed by the system in case of any error or failure. This use case will be used in online/ virtual meeting module.

2.14.11 Use case 11: Leave Meeting

ID	**UC11**
Name	Leave Meeting
Primary Actor	User
Pre-Conditions	User has joined the meeting
Post-Conditions	User will be sign out from the meeting
Main Success Scenario	1. System displays the option of leave meeting 2. User clicks on Leave meeting button 3. User sign out from the meeting

Alternative flows or Extensions	None
Frequency	During or after meeting hours

Once the user has joined the meeting then he can also leave the meeting as well. System displays the option of leave meeting to the user. When the meeting is done then user click on leave meeting option then user will be sign out from the meeting. User cannot leave the meeting between the meeting when meeting is done and user leave the meeting. This use case will be used in online/ virtual meeting.

2.14.12 Use case 12: Manage Account

ID	**UC12**
Name	Manage Account
Primary Actor	Admin
Pre-Conditions	Admin logged in to the system
Post-Conditions	Information related to account updated to the system
Main Success Scenario	1. System displays the option of Manage account 2. User clicks on Leave meeting button 3. User sign out from the meeting
Alternative flows or Extensions	None
Frequency	During or after meeting hours

Treasurer has responsibility to maintain the account of all users. Farming a budget and income details of the association will be managed by the treasures which have special right to manage the account. This use case will be used in member account management

2.14.13 Use case 13: View Account Details

ID	**UC13**
Name	View Account Details
Primary Actor	User
Pre-Conditions	User logged in to the system
Post-Conditions	Account detail will be displayed to the user

Main Success Scenario	1. System displays the option of view account 2. User clicks on view Account Details button 3. User enter the time period [Date, months or years etc.] from which he/she wants to view the account details 4. Account details will be displayed to user.
Alternative flows or Extensions	None
Frequency	Could be nearly continuous

User can view the account details of the users. Income and budget of the association can be viewed by the user. User can view the expenditure and income of the association. User can view the account details from a specific year, months as well. This use case will be used in account management module.

2.15 Database Requirements

In order to use this web-based system there is at least 3GB hard disk and 1GB RAM. MS SQL Server will be used as SQL engine and database. An Entity Relationship Diagram (ERD) is a snapshot of data structures. An Entity Relationship Diagram shows entities (tables) in a database and relationships between tables within that database. For a good database design it is essential to have an Entity Relationship Diagram. Entities are the things for which we want to store information. An entity is a person, place, thing or event. Attributes are the data we want to collect for an entity. Relationships describe the relations between the entities.

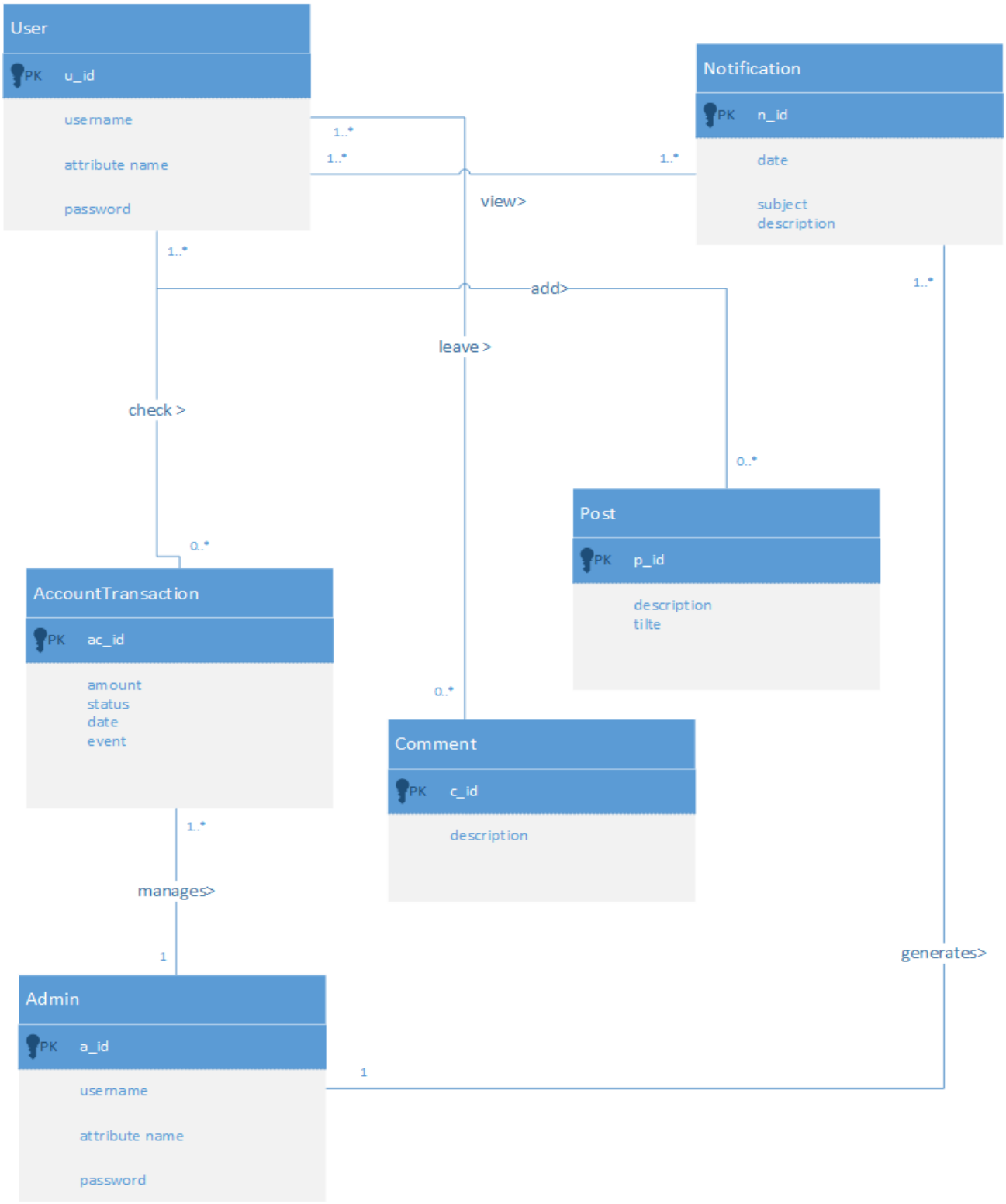

Figure 2.2 ER Diagram

This chapter has given complete information of requirement gathering and analysis is to clear the requirements of the system and to decide what the system should do and what the system should not do. Complete information of specific requirements including functional and non-functional requirements. Design and implementation of system will be discussed in the next chapter.

"An investment in Knowledge pays the best interest " Benjamin franklin (1706-1790)

Chapter 3

Software Design Description

This chapter first gives the complete description of software design. It then elaborates the architectural design and detailed description of components of system. Finally, this chapter elucidates the user interface design and interaction diagrams mainly system sequence diagram and class diagram.

3.1 Introduction

Software Design Description (SDD) is the representation of a software design to be used for communication design information to its stakeholders. It shows how the software system will be structured to satisfy the requirements. The SDD is performed in two stages. The first is a preliminary design in which the overall system architecture and data architecture is defined. In the second stage, that is the detailed design stage, more detailed data structures are defined and algorithms and codes are developed for the defined architecture.

3.1.1 Design Overview

Design is meaningful engineering representation of something that is to be built. The design mainly focus on four major areas of concern mainly data, architecture, interfaces and components [4]. Software design is an iterative process through which requirements are translated into a blueprint for constructing the software. It shows how end user can interact with the system therefore mainly focus on user interface design. Design begins with requirement model and at each stage, software design work product are reviewed for clarity, correctness, completeness and consistency with the requirements and with one another. Software design sits at the technical kernel of software engineering and is applied regardless of the software process model that is used. The requirements translated clearly through designing class diagram, sequence diagram, system sequence diagram and user interface interactions.

3.1.2 Requirement Traceability Matrix

Requirements traceability matrix is a matrix in which we describe that which requirement is mapping with which sequence diagram, test case, and method of class diagram. The purpose of traceability matrix is that when requirements have to be updated then one can update that requirement using traceability matrix instead of going through the whole document. It is often used with high-level requirements and detailed requirements of the product to the matching parts of high-level design, detailed design, test plan, and test cases.

Table 3.1 Requirement Traceability Matrix

Requirement Id	Requirement Name	Sequence Diagram	Test Case	Class Diagram	Interface	Domain Model
UC:1	Login	Sequence Diagram login	T1	Login	Login Interface	Login
UC:2	Add Notification	Sequence Diagram add notification	T4	Notification	Notification Interface	Notification
UC:3	Add topic	Sequence diagram add topic	T2	Post	Discussion forum interface	Post
UC:4	Leave comment	Sequence diagram comment	T3	Comment	Discussion forum interface	Comment
UC:5	Join meeting	Sequence diagram join meeting	T7	Online Meeting	Online Meeting interface	Online Meeting
UC:6	Leave meeting	Sequence diagram leave meeting		Online Meeting	Online Meeting Interface	Online Meeting
UC:7	Update Notification	Sequence diagram update notification	T6	Notification	Notification interface	Notification
UC:8	View account details	Sequence diagram view account detail	T9	Account Transaction	Account Manage interface	Account Transaction
UC:9	Manage Account	Sequence diagram manage account	T8	Account Transaction	Account Manage interface	Account Transaction

3.2 System Architecture Design

Architectural design is defines the relationship between major structural elements of the software. It defines the design patterns that can be used to achieve the requirements that have been defined for the system. Architecture design entails the manner in which these components interact and the structure of data that are used by the components. Components or modules are generalized to represent major system elements and their interactions.

3.2.1 Chosen System Architecture

The chosen architecture for this system is Three Tier Architectural pattern Three-tier architecture is a software architecture pattern in which the user interface (presentation),functional process logic (business rules), computer data storage and data access are developed and maintained as independent modules Interacting between components of system is shown in diagram. The singular quality of a three-tier architecture is the separation of the application logic into a distinct logical middle tier of software. The interface tier is relatively free of application processing; windows or web pages forward task requests to the middle tier. The middle tier communicates with the back-end storage layer. [5]

A three-tier architecture separates its tiers from each other based on the complexity of the users and how they use the data present in the database. In database tier, the database resides along with its query processing languages. This tier also have the relations that define the data and their constraints at this level. The middle tier reside the application server and the programs that access the database. For a user, this application tier presents an abstracted view of the database. End-users are unaware of any existence of the database beyond the application. At the other end, the database tier is not aware of any other user beyond the application tier. Hence, the application layer sits in the middle and acts as a mediator between the end-user and the database. In user interface tier end-users operate on this tier and they know nothing about any existence of the database beyond this layer. At this layer, multiple views of the database can be provided by the application. All views are generated by applications that reside in the application tier.

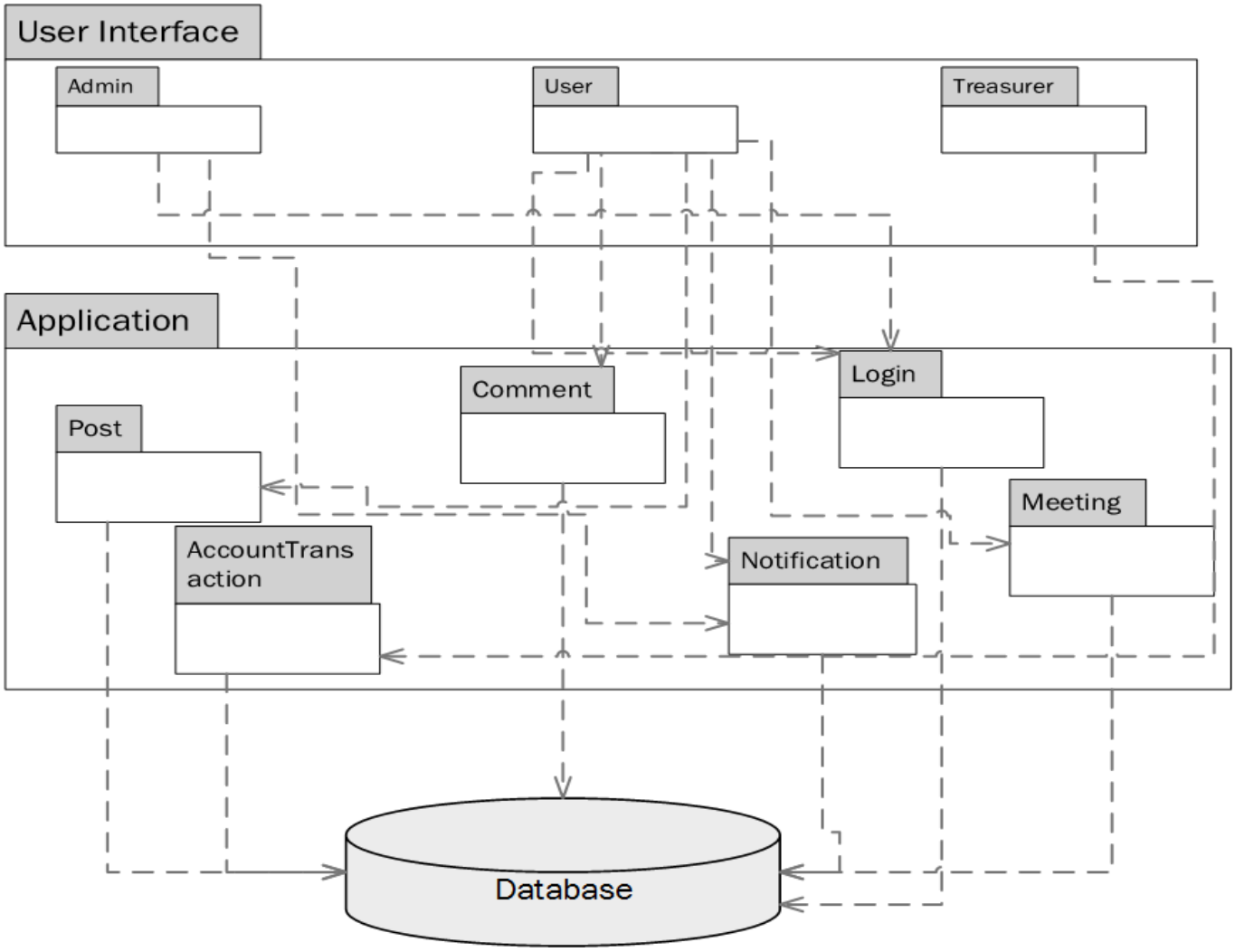

Figure 3.1 Architecture Diagram

3.3) Domain Model

A domain model is a system of abstractions that describes selected aspects of a sphere of knowledge, influence, or activity.[6] The model can then be used to solve problems related to that domain. The domain model is a representation of meaningful real-world concepts pertinent to the domain that need to be modelled in software. The concepts include the data involved in the business and rules the business uses in relation to that data.

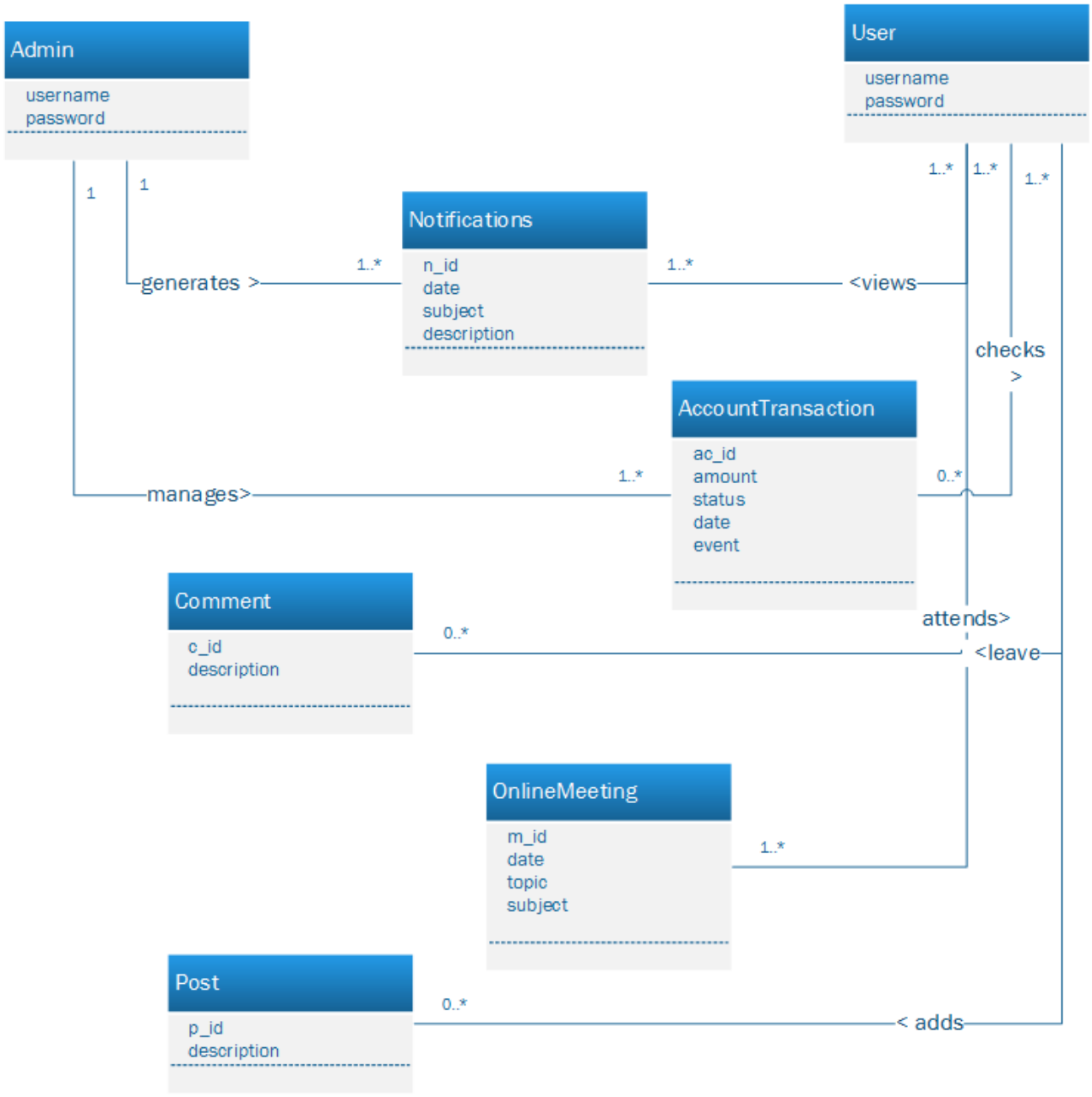

Figure 3.2 Domain Model

3.4 Component Diagram

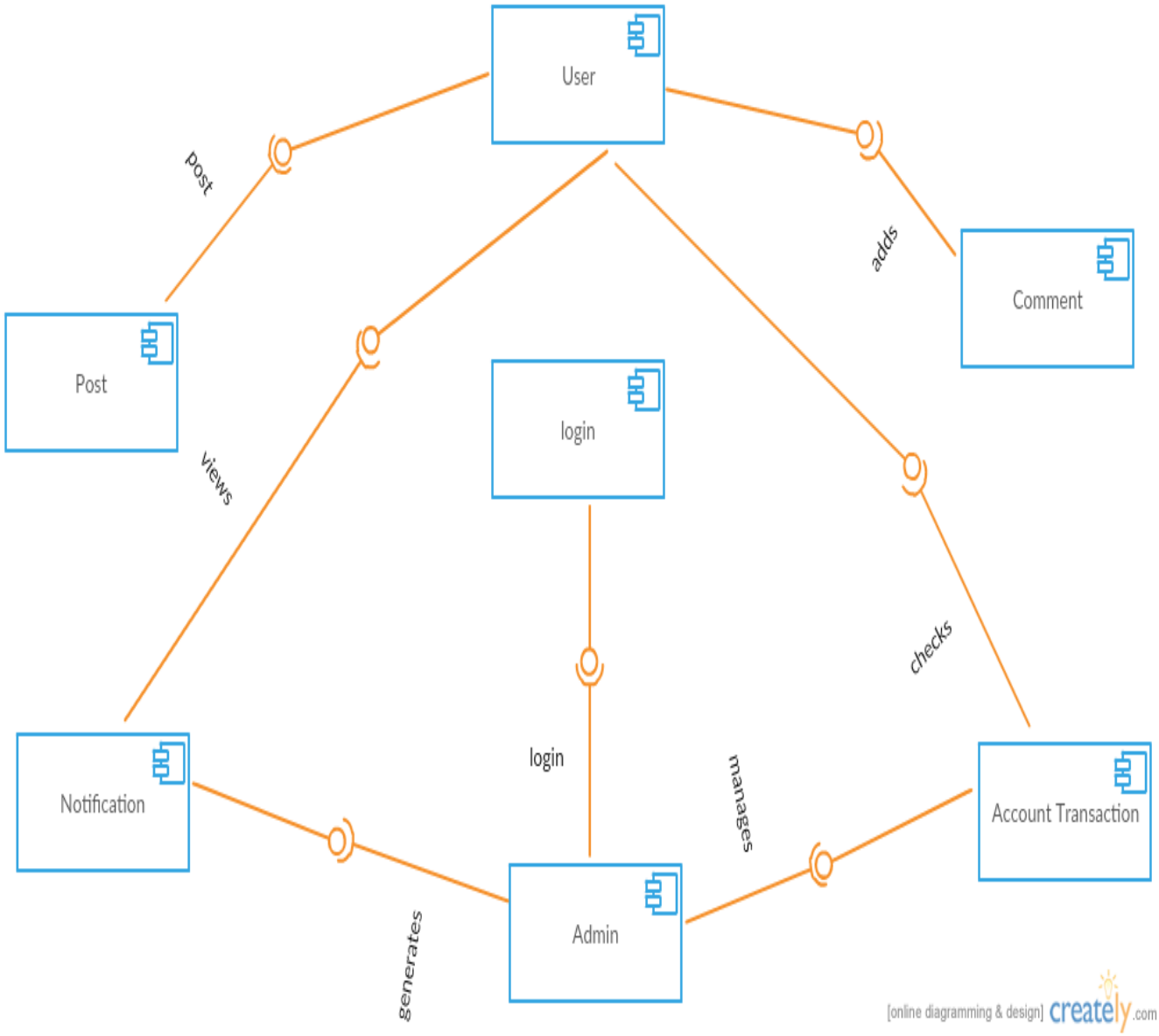

Figure 3.3 Component Diagram

3.5 User Interface Design

User interface design creates an effective communication between user and a computer. User interface design begins with the identification of user, task, and environmental requirements.

3.5.1 Description of User Interface Design

User interface is the part of software and is design in such a way that is expected to provide the user insight of the software. User interface provide the fundamental platform for human-computer interaction. User interface can be graphical, command line and text-based. There are large number of activities performs for designing user interface. These activities includes interface analysis and modelling, interface design, design construction and design validation. Interface analysis and modelling focus on profile of user that will interact with the system. Interface design is to define a set of interface objects and actions that enable a user to perform all defined tasks in a manner that meets every usability goal defined for the system. Interface construction normally begins with the creation of a prototype that enables usage scenarios to be evaluated. Finally interface validation focuses on the ability of the interface to implement every user task correctly, to accommodate all task variations, and to achieve all general user requirements. User interface divided in to two categories; graphical user interface and command line interface. Command line interface provides a command prompt, the place where the user types the command and feeds to the system. Graphical User Interface (GUI) provides the user graphical means to interact with the system. GUI can be combination of both hardware and software.

Characteristic	**Description**
Window	Multiple windows allow different information to be displayed simultaneously on the user's screen.
Icon	Icons different types of information. On some systems, icons represent files; on others, icons represent processes.
Menu	Commands are selected from a menu rather than typed in a command language.
Pointing	A pointing device such as a mouse is used for selecting choices from a menu or indicating items of interest in a window.
Graphics	Graphical elements can be mixed with text on the same display.

Interface 1: Login

User first login to the system.

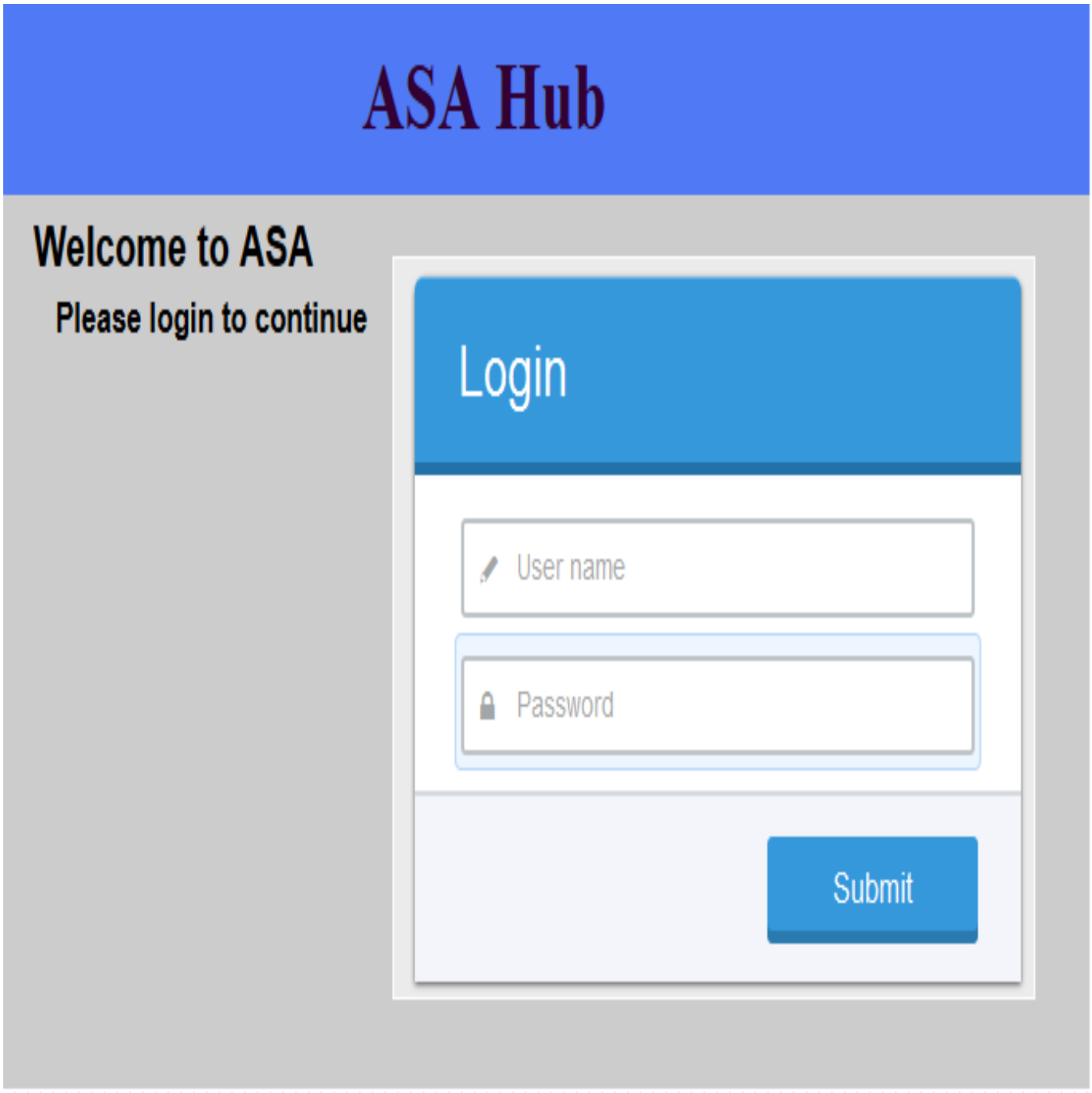

Figure 3.4 Login Interface

Interface 2: Home Page

After Log in to the system, home page will appear to the user.

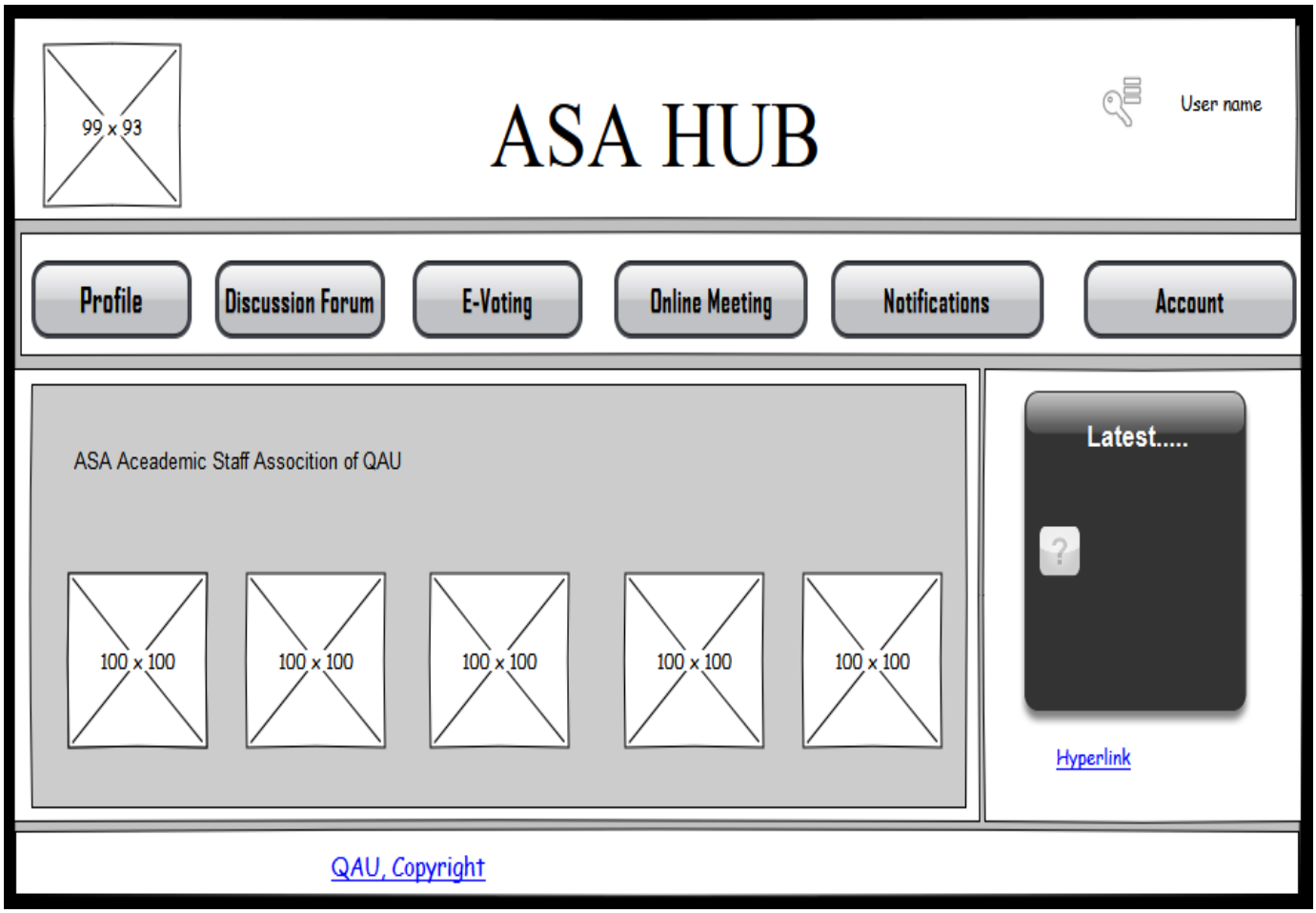

Figure 3.5 Home Page Interface

Interface 3: Discussion Forum

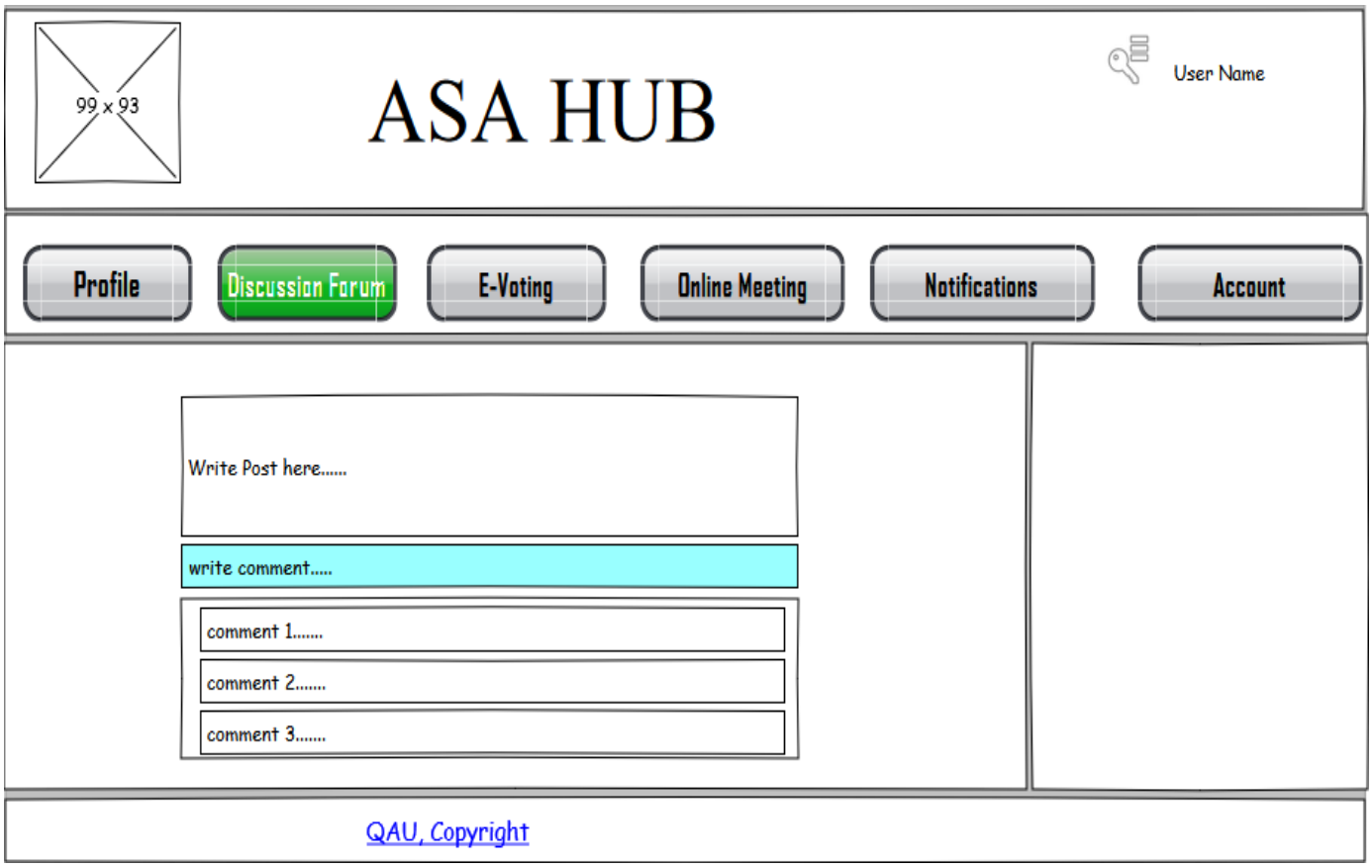

Figure 3.6 Discussion Forum Interface

Interface 4: Online or Virtual Meeting

Figure 3.7 Online Meeting Interface

Interface 5: Notification Management

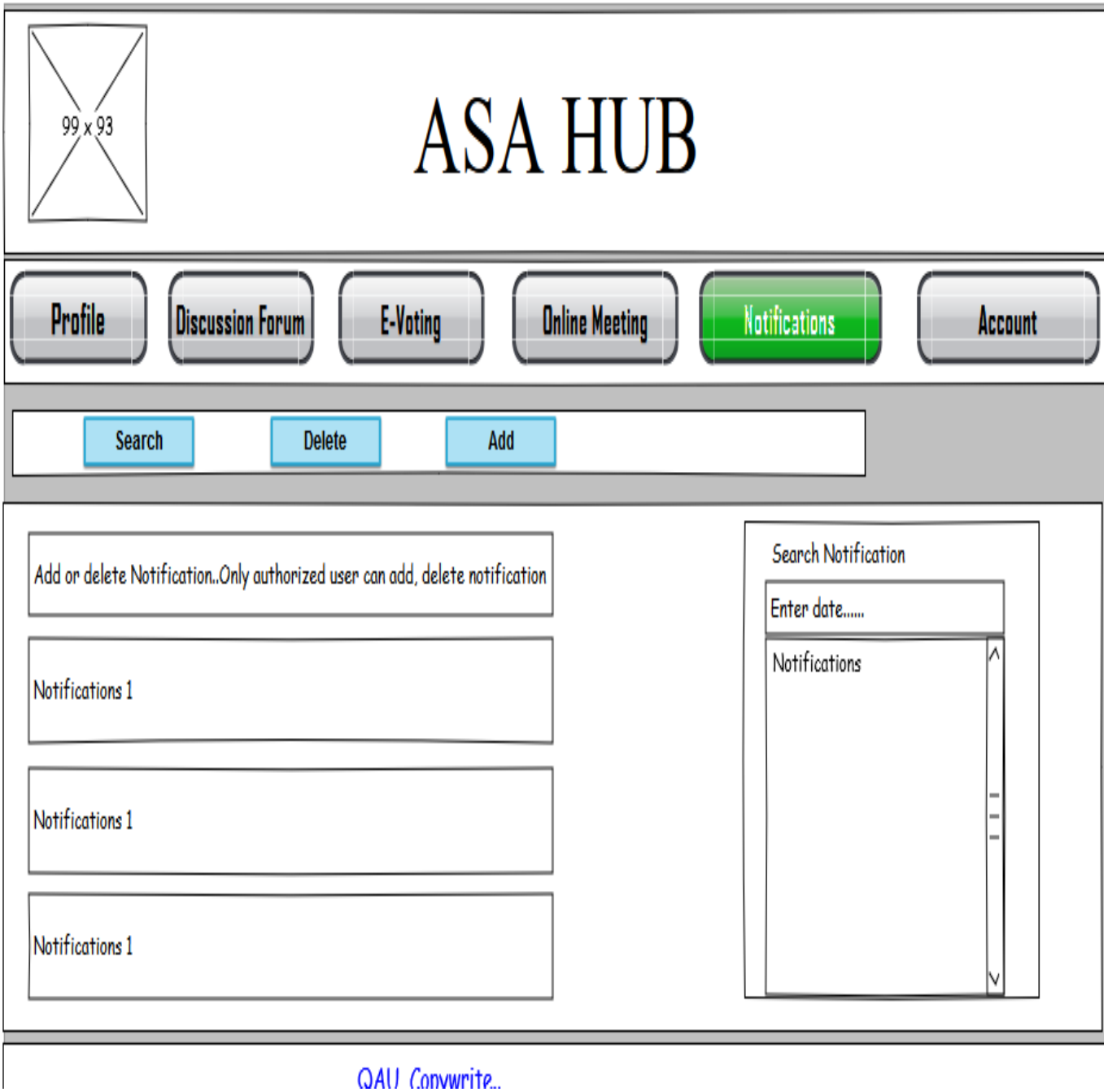

Figure 3.8 Notification Interface

Interface 6: Account Management

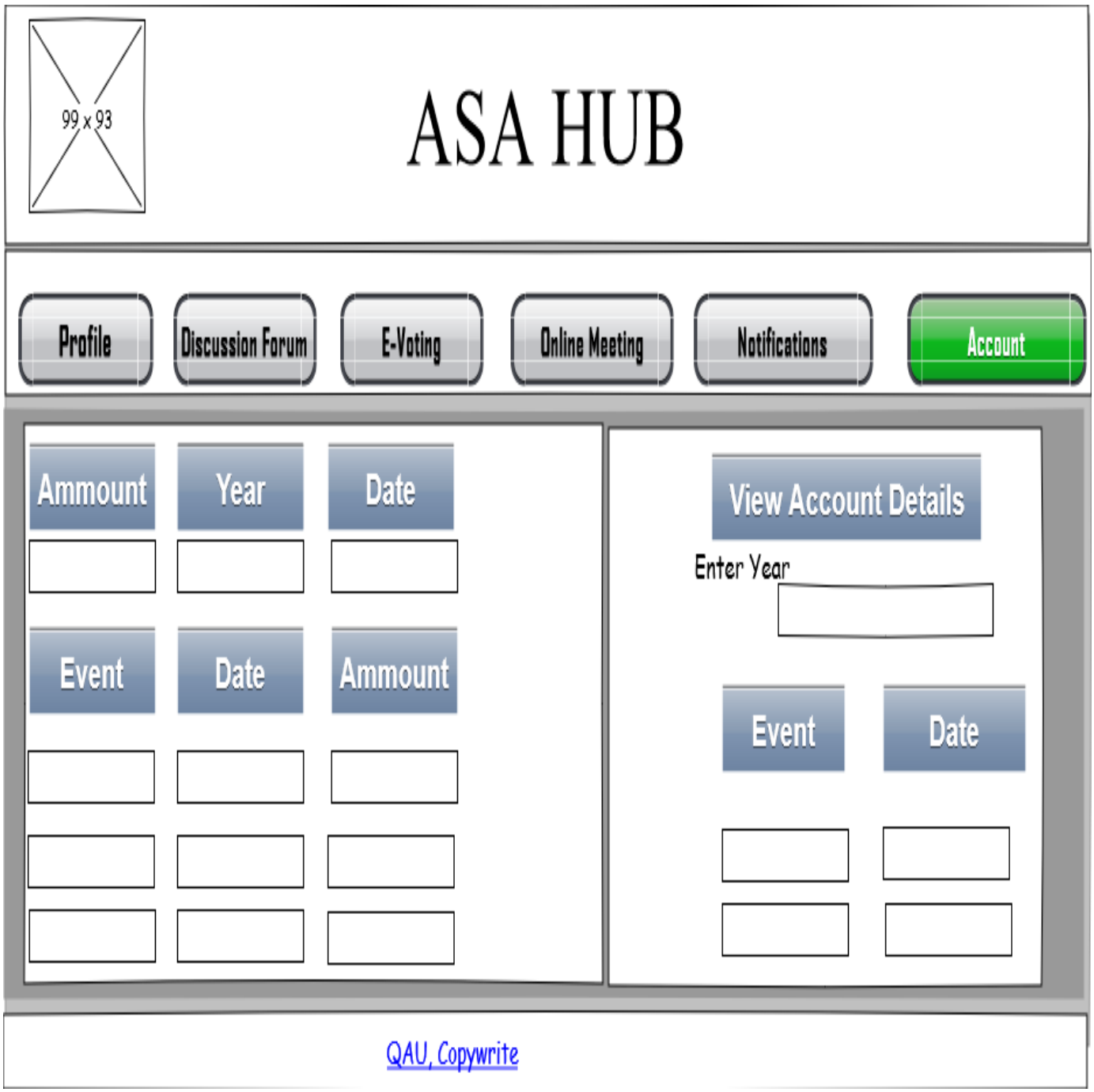

Figure 3.9 Account Manage Interface

3.6 Sequence Diagram

Sequence diagram are used to model the interaction between the actors and the objects in a system and interaction between the objects themselves [7]. A sequence diagram shows that interactions that take place during a particular use case or use case scenario.

3.6.1 Admin Login

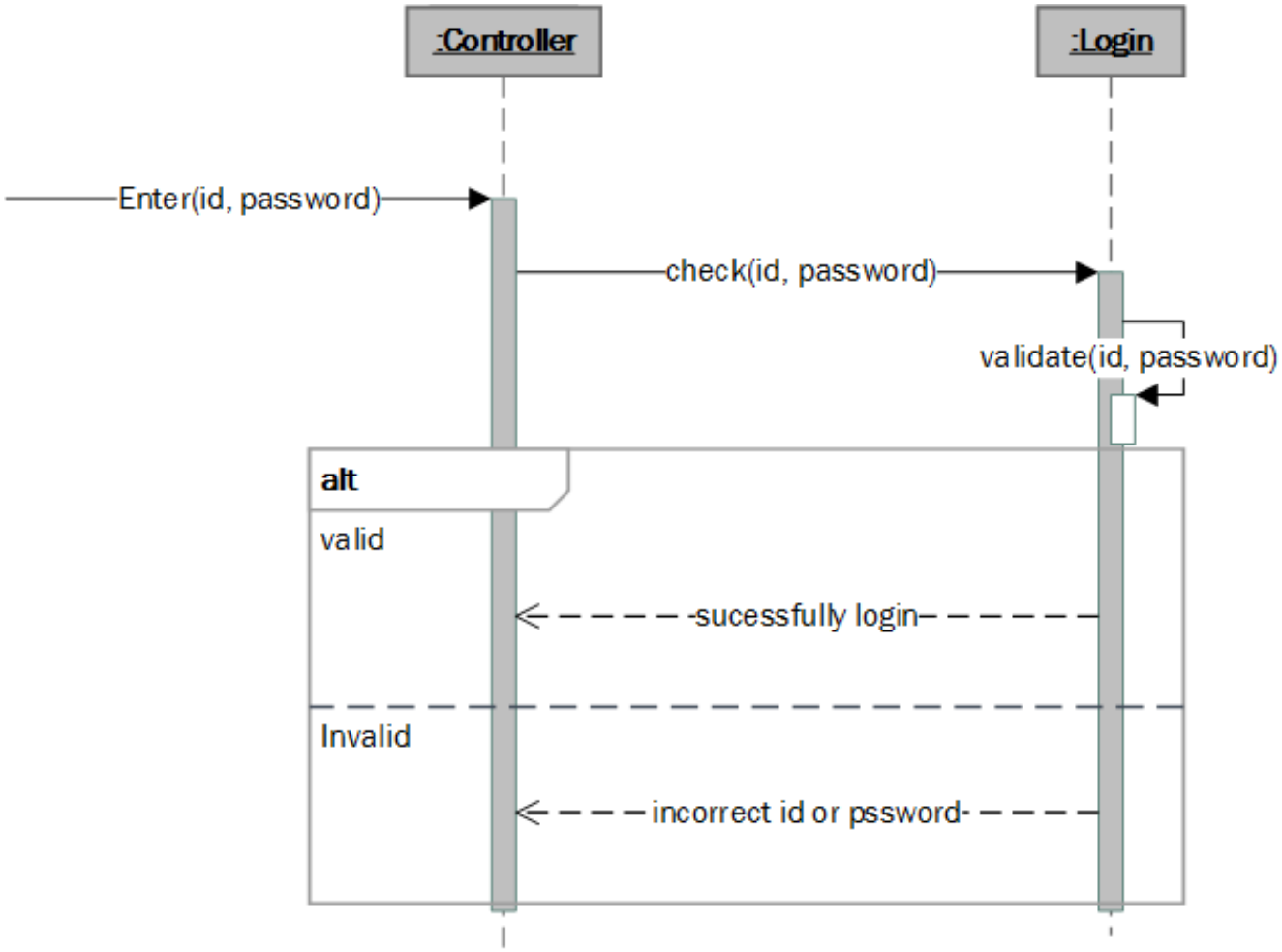

Figure 3.10 Sequence Diagram Admin Login

Sequence diagram of admin login shows that sequence of interaction that take place when the admin wants to login to the system. User enters id and password to the system then system validates id and password. If admin is authorized then admin will be successfully login to the system else error message will be displayed to the user.

3.6.2 User Login

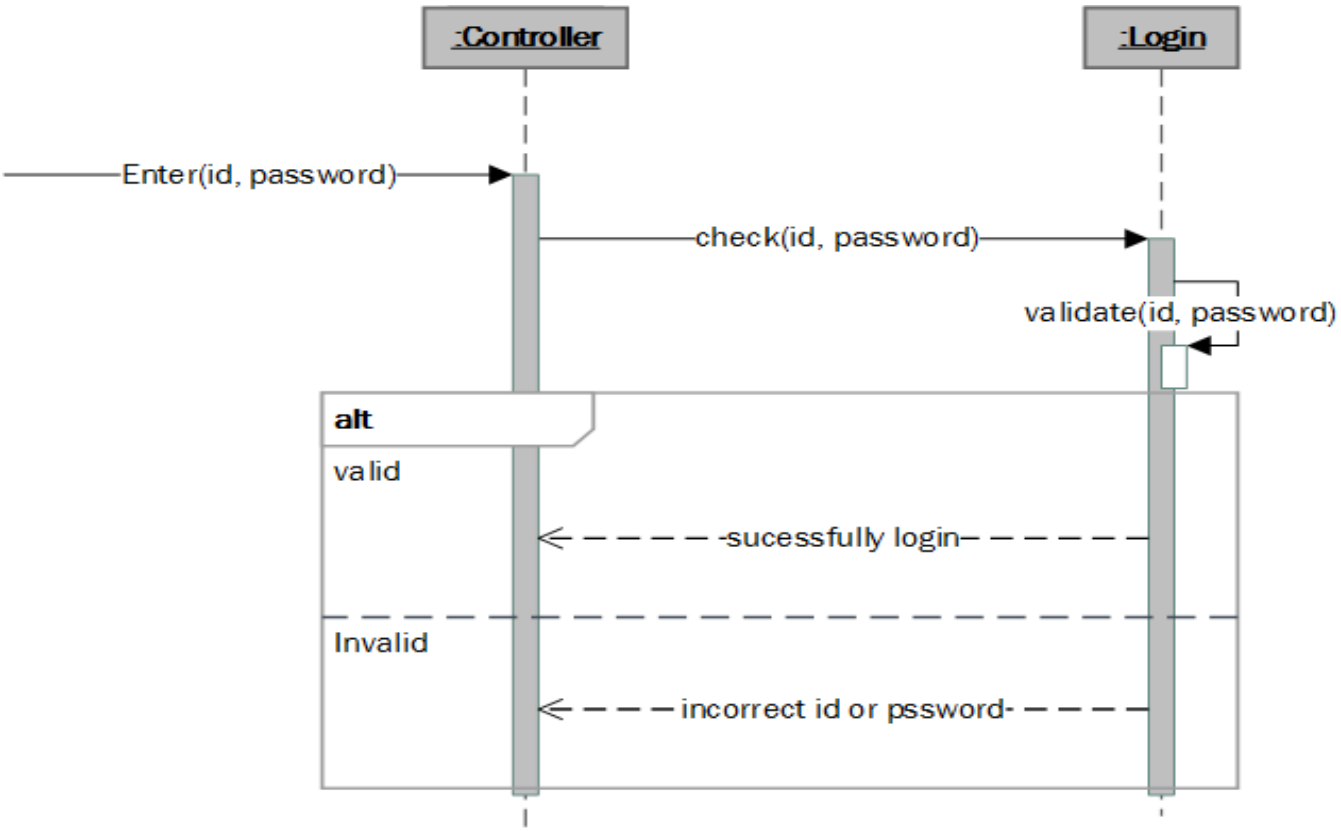

Figure 3.11 Sequence Diagram Add topic

Sequence diagram for user login shows the interaction that how user can able to log into the system. User first enters the user name and password, after validating the user name and password user will successfully login to the system.

3.6.3 Add topic

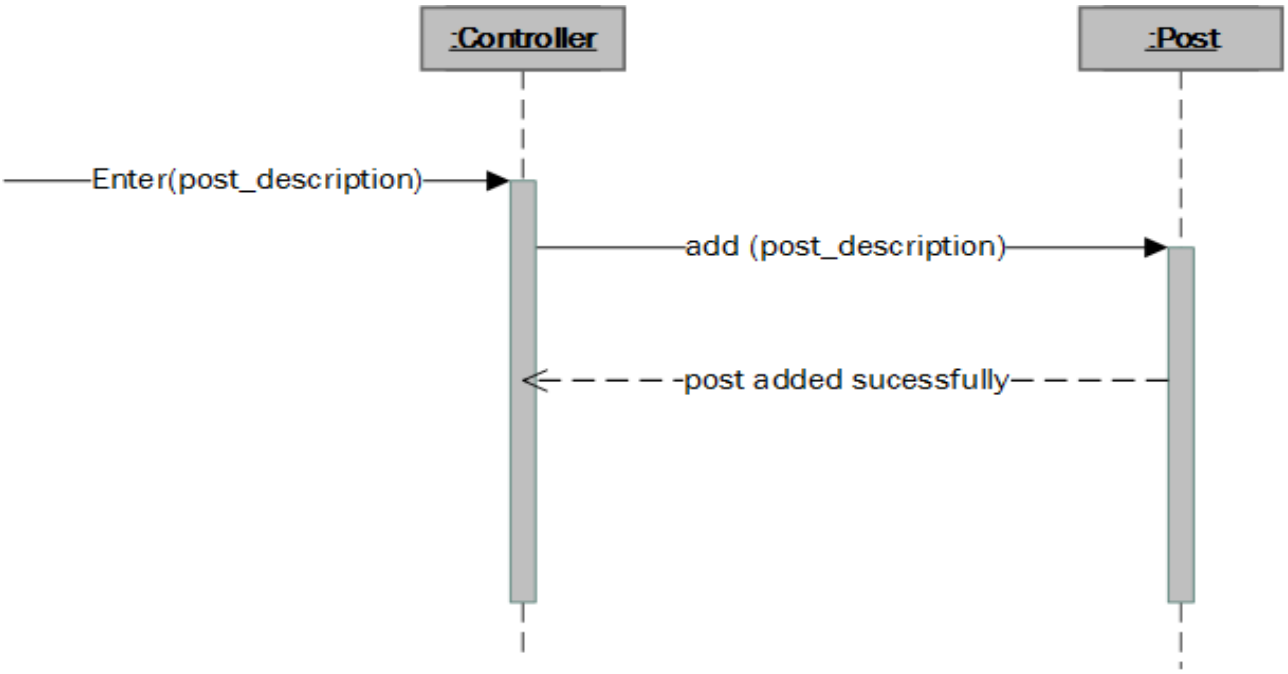

Figure 3.12 Sequence Diagram Add topic

The sequence diagram of add topic or post shows that the sequence of steps that when a user wants to add post to the system. User enters the post description to the controller as an input then controller send that input to post class and finally post added successfully to the system.

3.6.4 Leave Comment

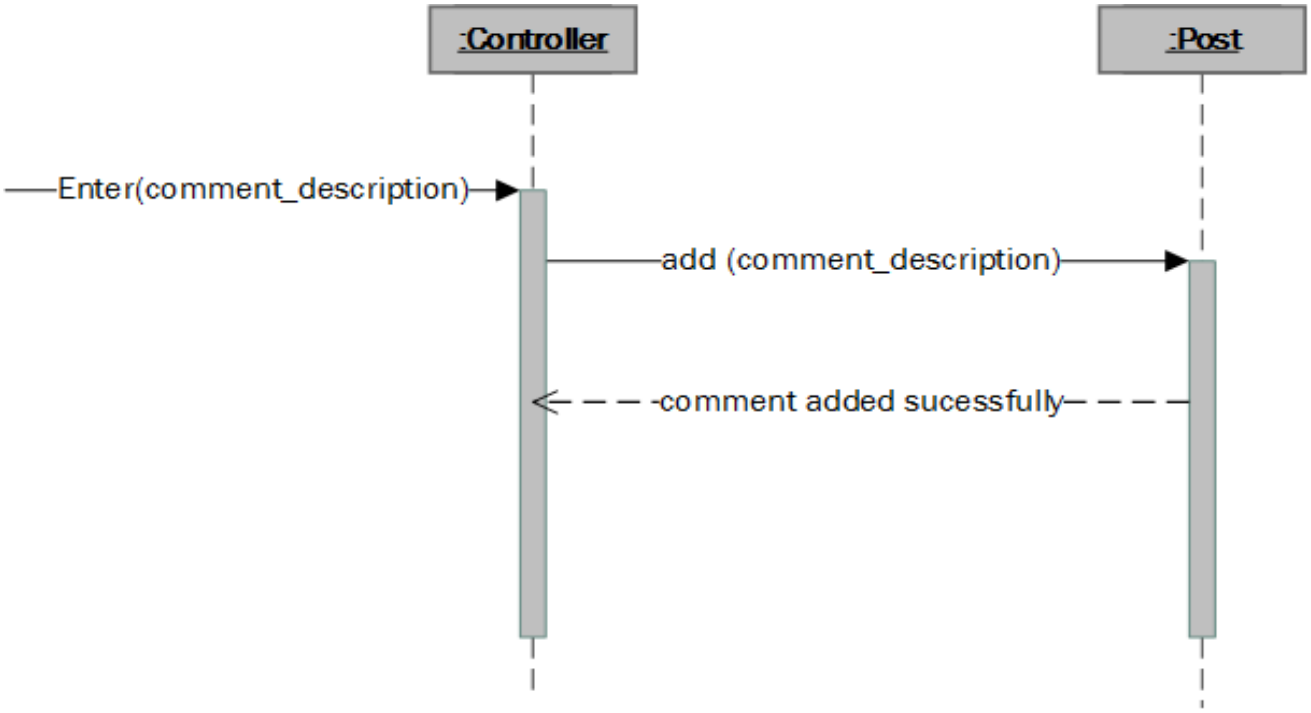

Figure 3.13 Sequence Diagram Comment

After the post the user can comment on that post. Sequence diagram of leave comments shows that how a comment can be added on respective post. User enters the comment description in the comment box then that comment will be added against respective post.

3.6.5 Join Meeting

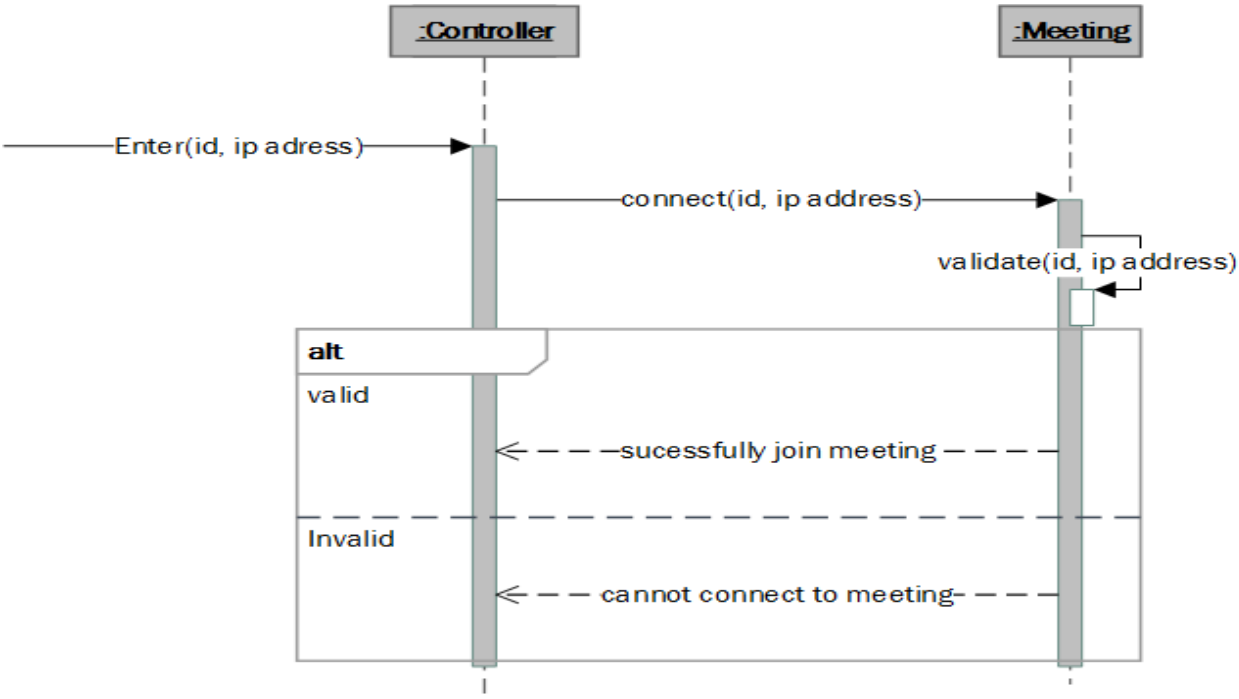

Figure 3.14 Sequence Diagram Join Meeting

Sequence diagram of join meeting shows the sequence of interactions that how a user can join meeting. In order to join the meeting ip address of the server is required. User enters id or respective ip address. After validating the ip address or id user successfully join the meeting.

3.6.6 Leave Meeting

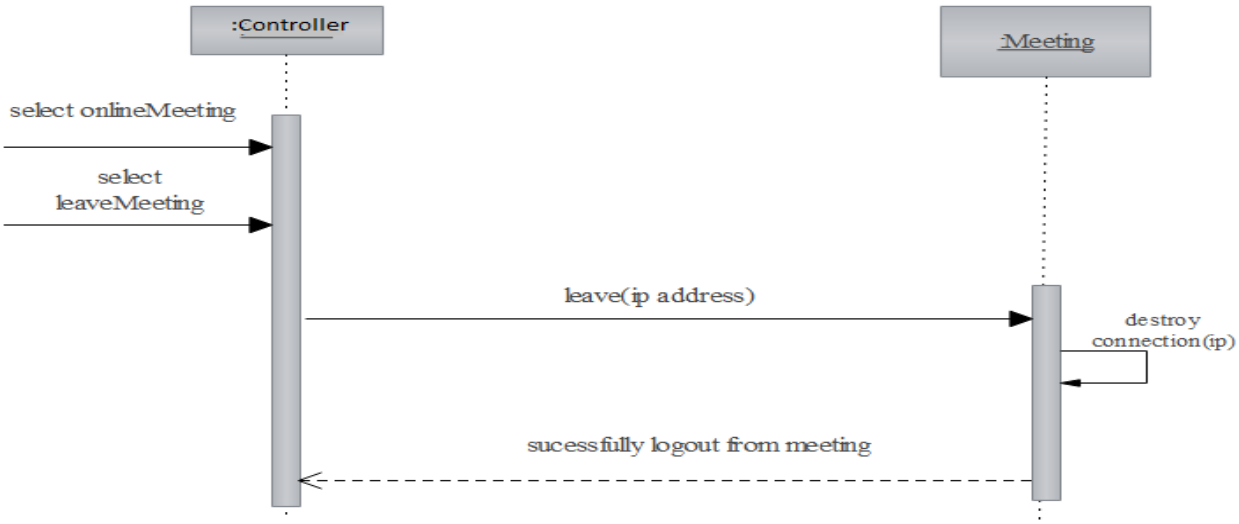

Figure 3.15 Sequence Diagram Leave Meeting

In order to leave the meeting, system destroy the session of respective user and user logout from the meeting.

3.6.7 Add Notification

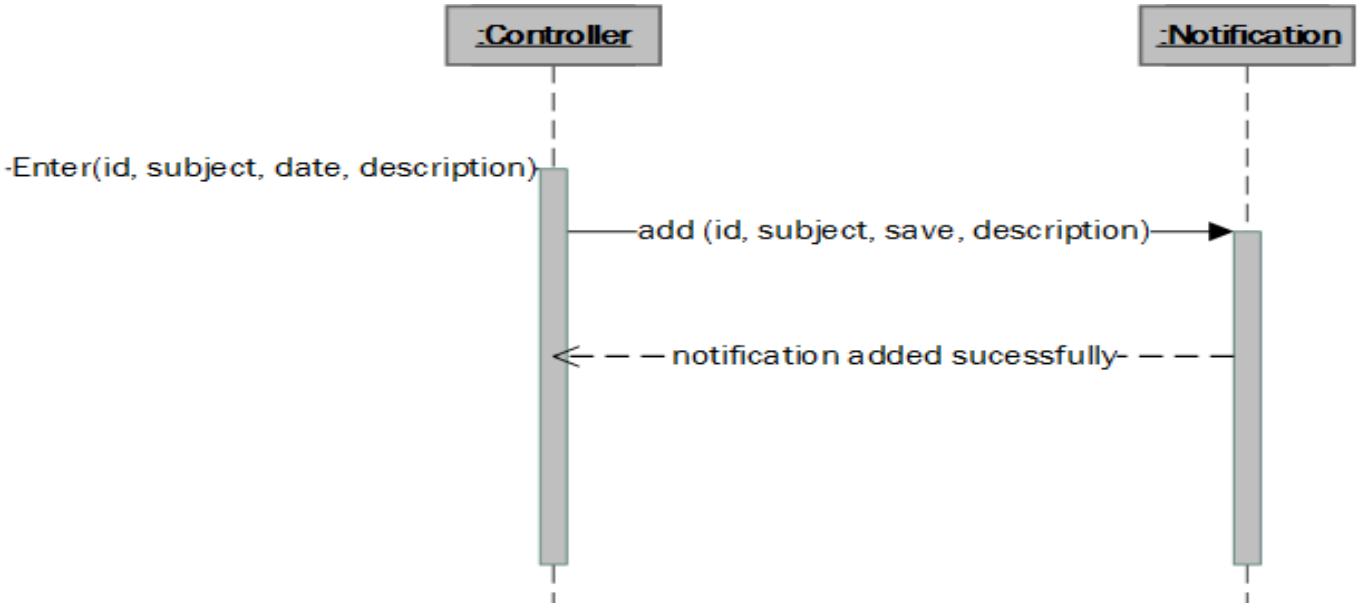

Figure 3.16 Sequence Diagram Add Notification

Sequence diagram for adding notification shows the interaction between the objects that how a notification added to the system. Admin has right to add notification to the system. Admin enters notification_id, date and description then message of notification added successfully will be displayed.. User enter the id of the notification or date, after executing the query, if notifications are available then notification will displayed

3.6.9 Manage Account

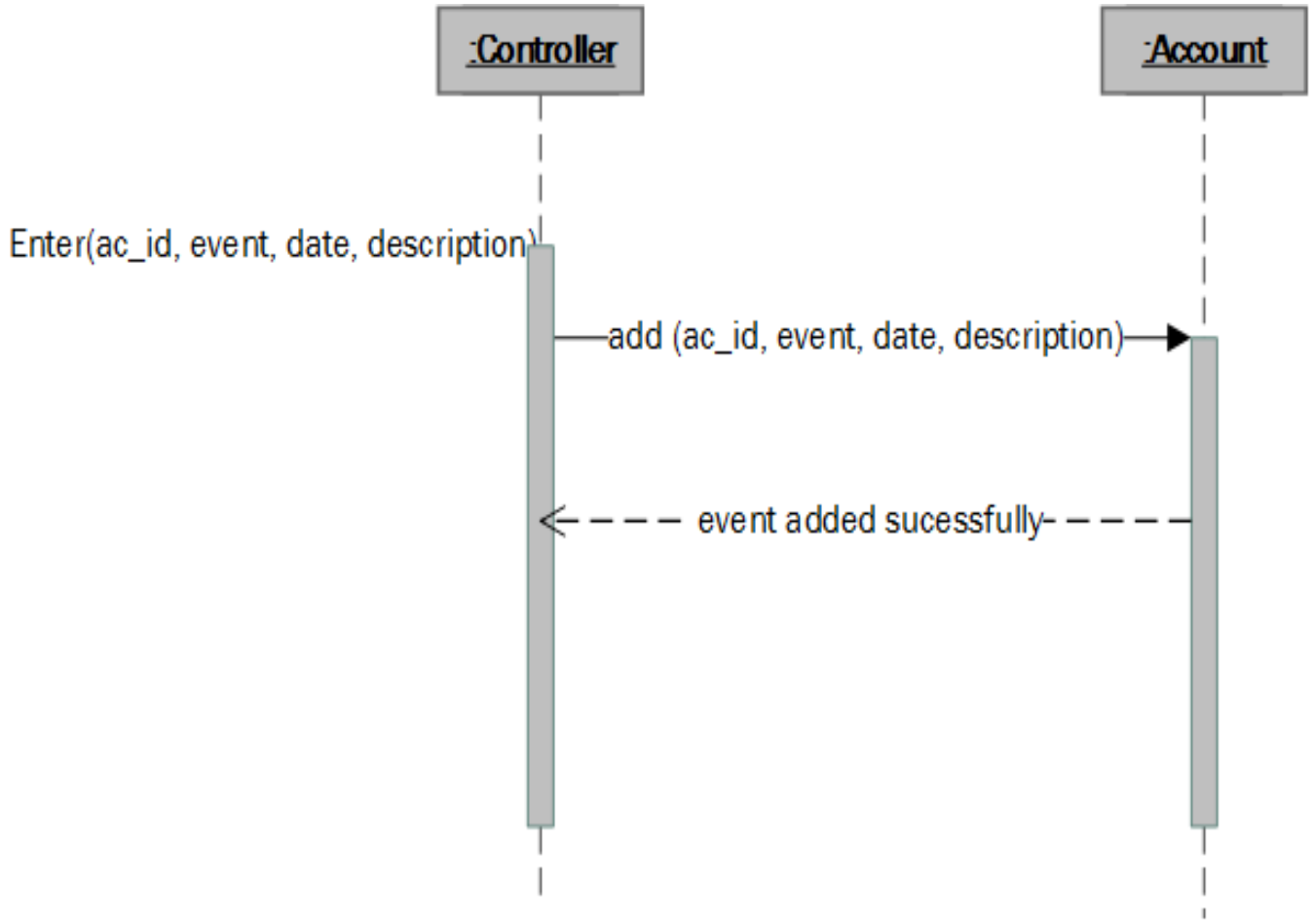

Figure 3.18 Sequence Diagram Manage Account

Sequence diagram for manage account shows the sequence of interaction between the instances that how account is managed. Treasurer enters the event, date, description means total budget or cost of event to the system. Amount of respective event added to the system

3.6.10 View Account Detail

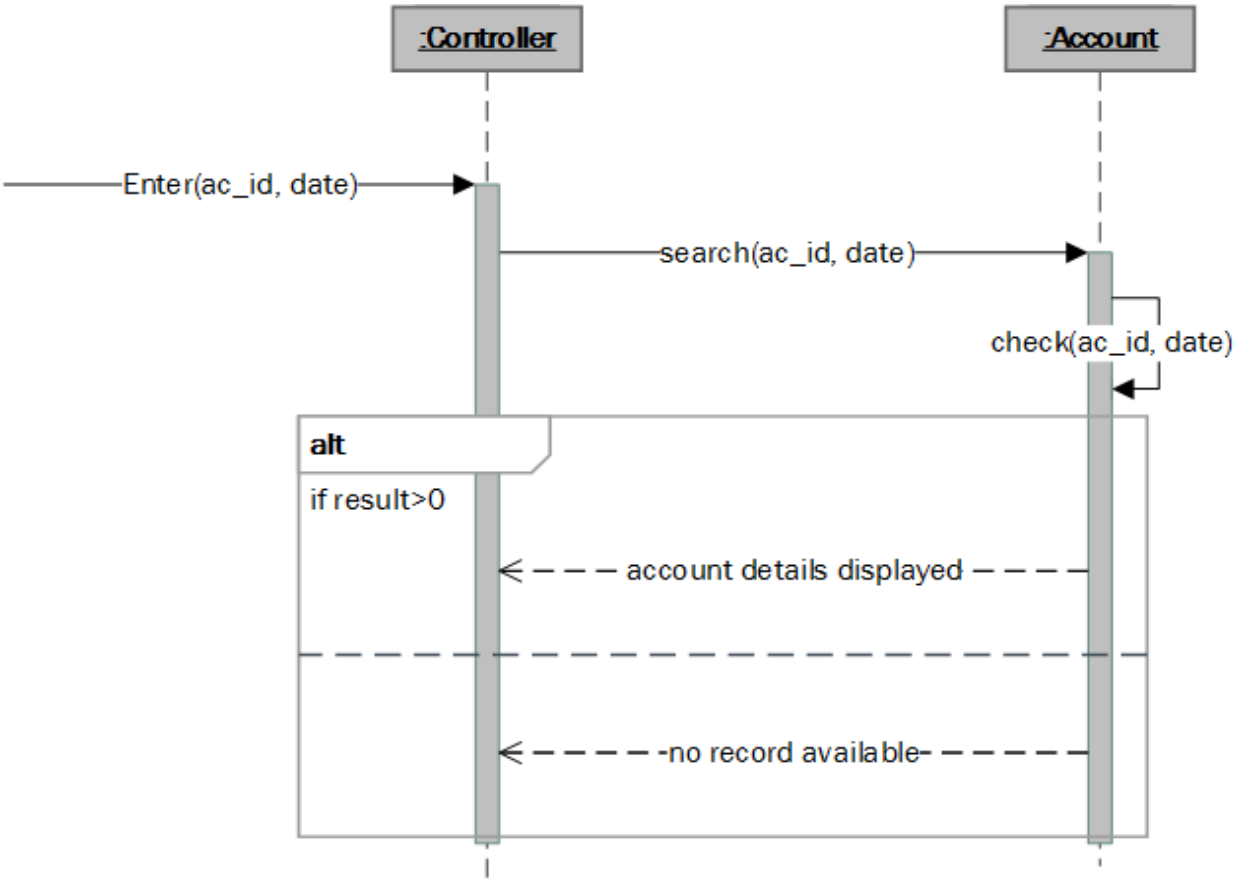

Figure 3.19 Sequence Diagram View Account Detail

View Account details lies under manage account modules. Sequence diagram of view account details shows the interaction that how user can view the account details. User enters the date and after checking the result against respective date account details displayed to user.

3.7 Class Diagram

The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application. The class diagram shows a collection of classes, interfaces, associations, collaborations and constraints. It is also known as a structural diagram. The purpose of the class diagram is to model the static view of an application. The class diagrams are the only diagrams which can be directly mapped with object oriented languages and thus widely used at the time of construction [8].

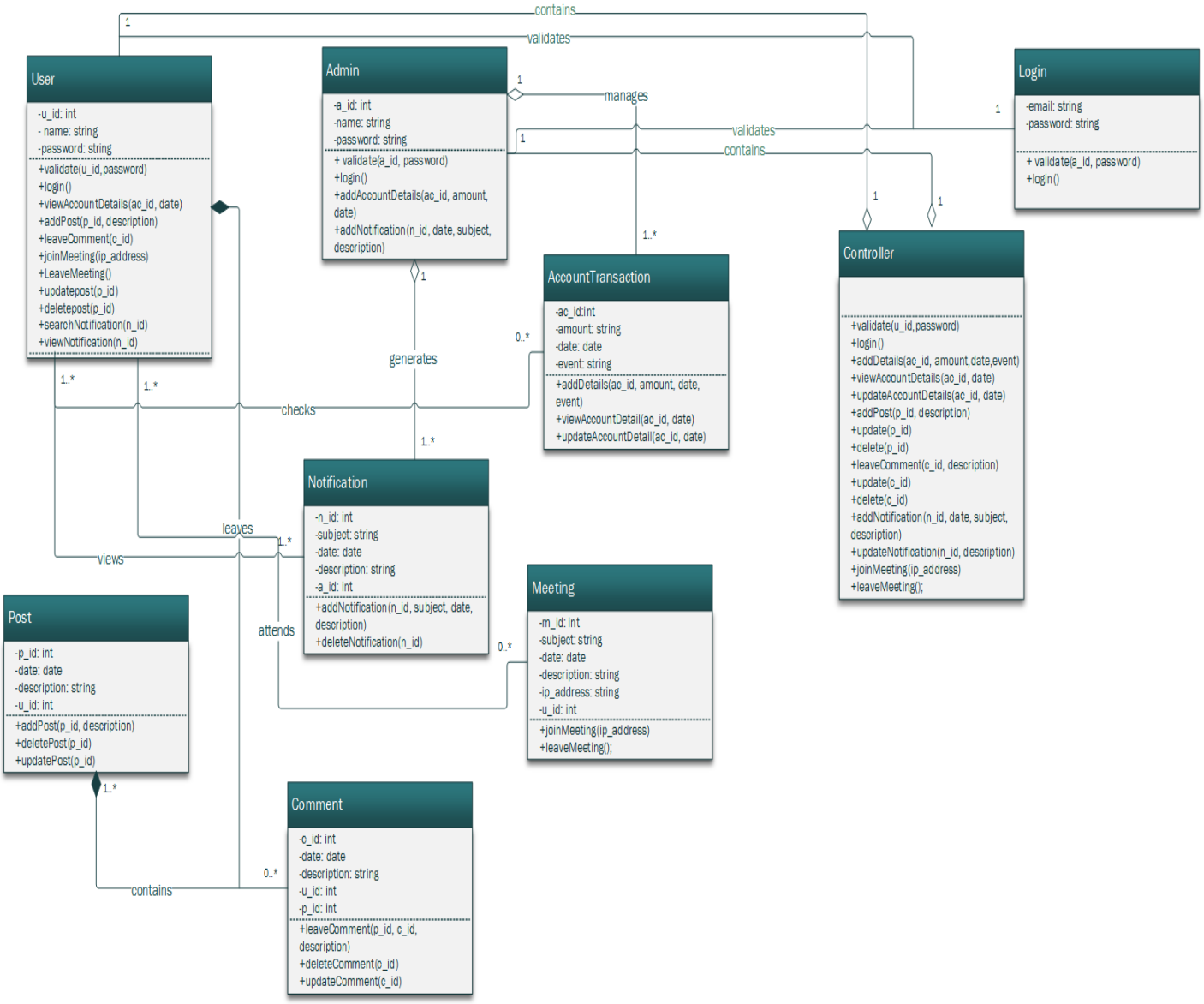

Figure 3.20 Class Diagram

This chapter has given the complete description of software design. It has given details description regarding architecture design, components of the system and user interface description. Finally, interaction between the object and human actor are shown by interaction diagram and relationship between the instances is shown by class diagram. System testing and test cases will be discussed in next chapter.

"Education is a progressive discovery of our own ignorance" Will Durant (1885-1981)

Chapter 04

System Implementation

This chapter is all about the implementation phase including language selection, framework selection, database selection, and software used, and screen snap shots. After the design phase, in implementation phase, which techniques and tools are used. It is a process that defines that how the system should be built. It also ensuring that the information system is operational and used. It also entailing the information system meets quality standard. This chapter is related to the actual implementation of the application.

4.1 Framework Selection

The dot net framework is used to develop this system. The .Net framework is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It helps in designing portable scalable and robust applications. The .NET's class library consists of classes, interfaces and value types the help in speeding up the development process and provides access to system functionality. Dot Net Framework which is a software development framework from Microsoft and being heavily used in Software Industry by millions of software developers. It allows software developers to use multiple programming languages in the same application. A software developer can develop application in an easier way without taking care of the performance and memory leakage issues because they are being taken care by the framework implicitly.

4.2 Language Selection

Asp.net c# is used to develop this system. ASP.NET is a web development platform, which provides a programming model, a comprehensive software infrastructure and various services required to build up robust web applications for PC, as well as mobile devices. ASP.NET works on top of the HTTP protocol, and uses the HTTP commands and policies to set a browser-to-server bilateral communication and cooperation. ASP.NET is a part of Microsoft .Net platform. ASP.NET applications are compiled codes, written using the extensible and reusable components or objects present in .Net framework. These codes can use the entire hierarchy of classes in .Net framework. ASP.NET is used to produce interactive, data-driven web applications over the internet. It consists of a large number of controls such as text boxes, buttons, and labels for assembling, configuring, and manipulating code to create HTML pages. Asp.net c# web forms extend the event-driven model of interaction to the web

applications. The browser submits a web form to the web server and the server returns a full mark-up page or HTML page in response. All client side user activities are forwarded to the server for state full processing. The server processes the output of the client actions and triggers the reactions.

4.3 Database Selection

Microsoft SQL Server management is used in this project as a database. MS SQL Server is a relational database management system (RDBMS) developed by Microsoft. This product is built for the basic function of storing retrieving data as required by other applications. It can be run either on the same computer or on another across a network. This tutorial explains some basic and advanced concepts of SQL Server such as how to create and restore data, create login and backup, assign permissions, etc. Each topic is explained using examples for easy understanding. SQL Server Management Studio is a workstation component\client tool that will be installed if we select workstation component in installation steps. This allows you to connect to and manage your SQL Server from a graphical interface instead of having to use the command line. In order to connect to a remote instance of an SQL Server, you will need this or similar software. It is used by Administrators, Developers, Testers, etc.

4.4 Software Used

Microsoft Visual Studio 2012 IDE

Visual Studio 2012's comprehensive set of development and debugging tools as well as its integrated. Unit testing and load testing tools can unquestionably help you develop high-quality applications.

Microsoft SQL Server Management Studio

Microsoft SQL Server is a relational database server, developed by Microsoft. It is a software product whose primary function is to store and retrieve data as requested by other software application. It is platform dependent. It is both GUI and command based software. It supports SQL (SEQUEL) language which is an IBM product, non-procedural, common database and case insensitive language.

4.4 Screen Shots

4.4.1 Login

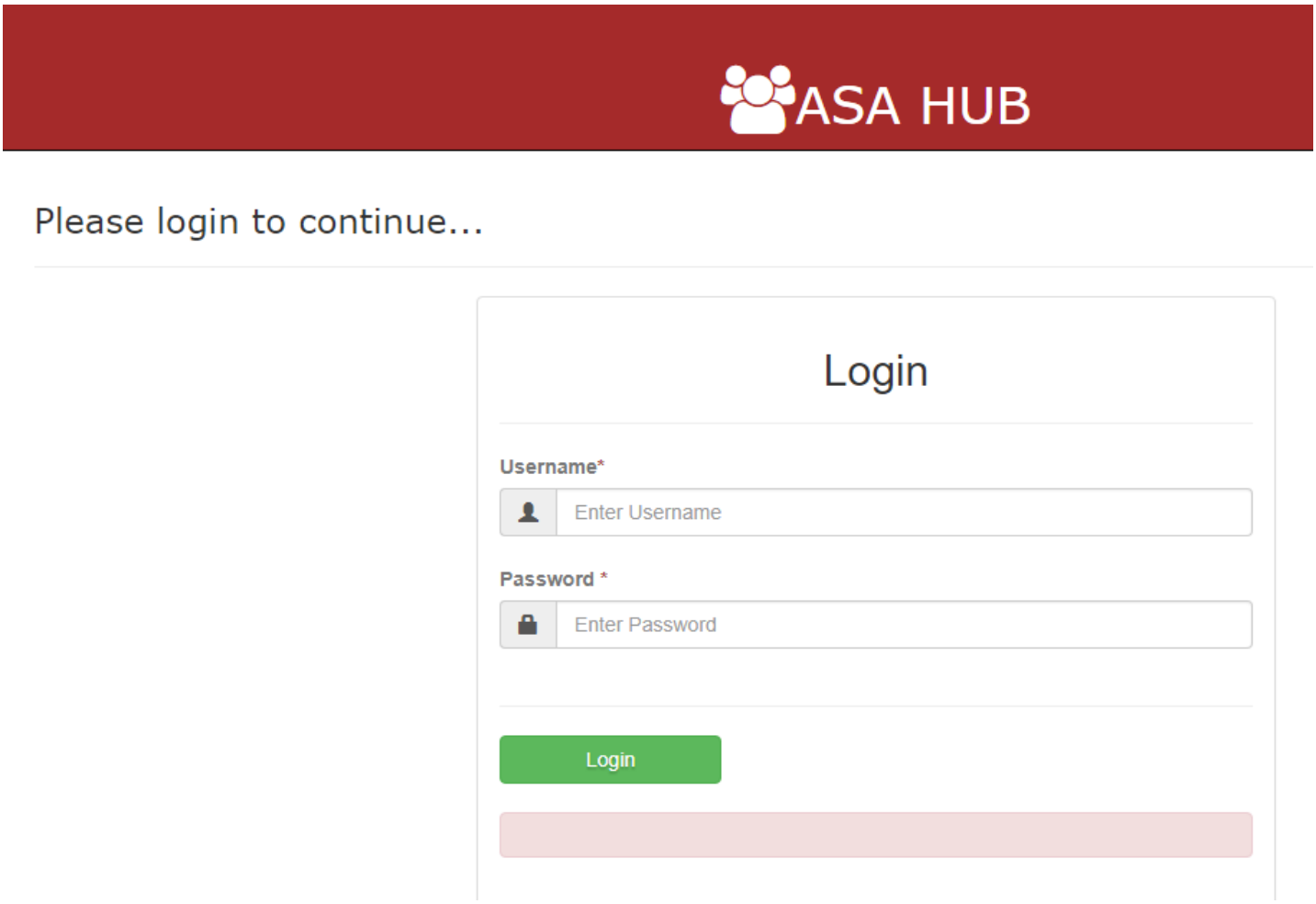

Figure 4. 1 Login screen

4.4.2. Add Notification

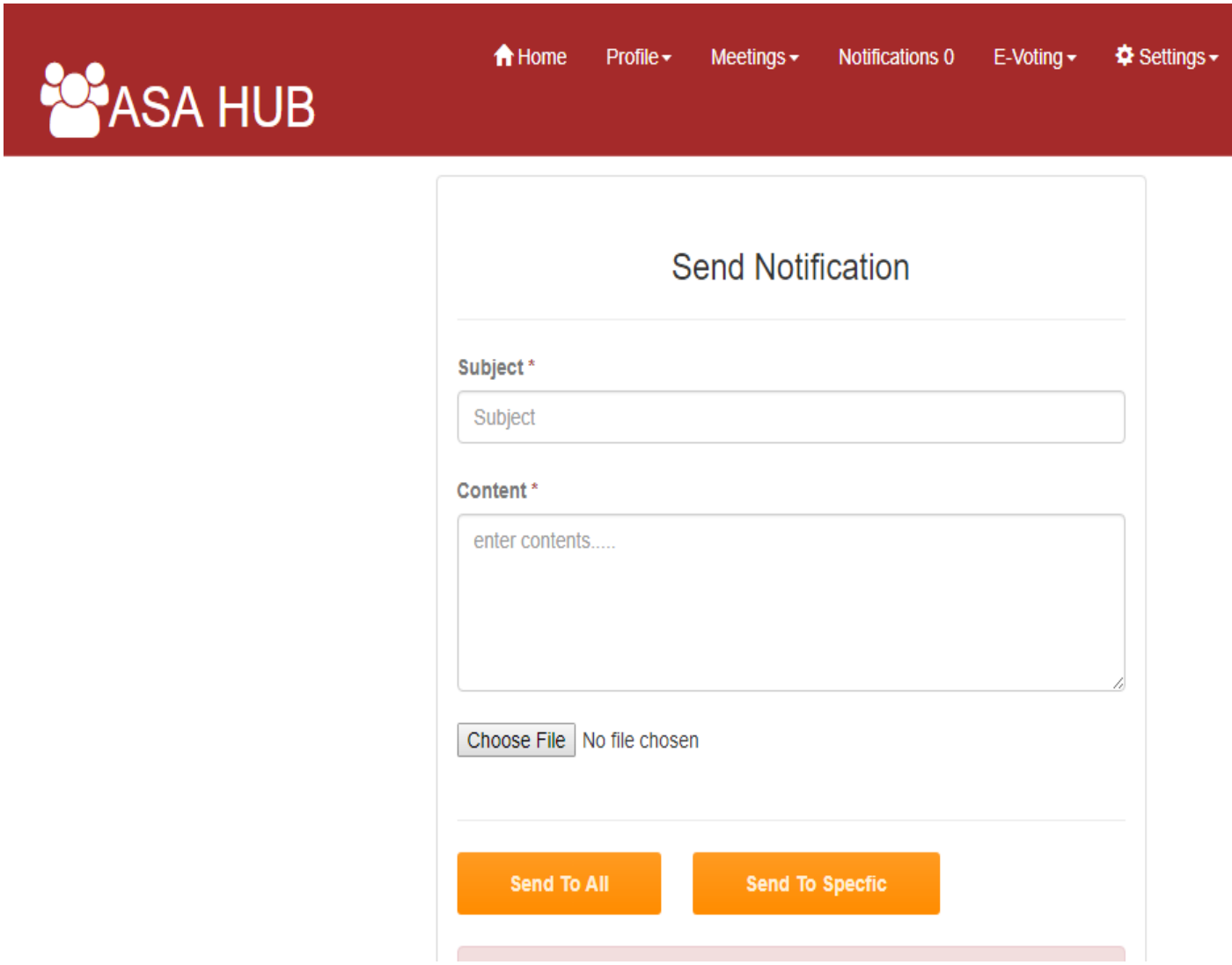

Figure 4. 2 Add Notification Screen

4.4.3 View Notification

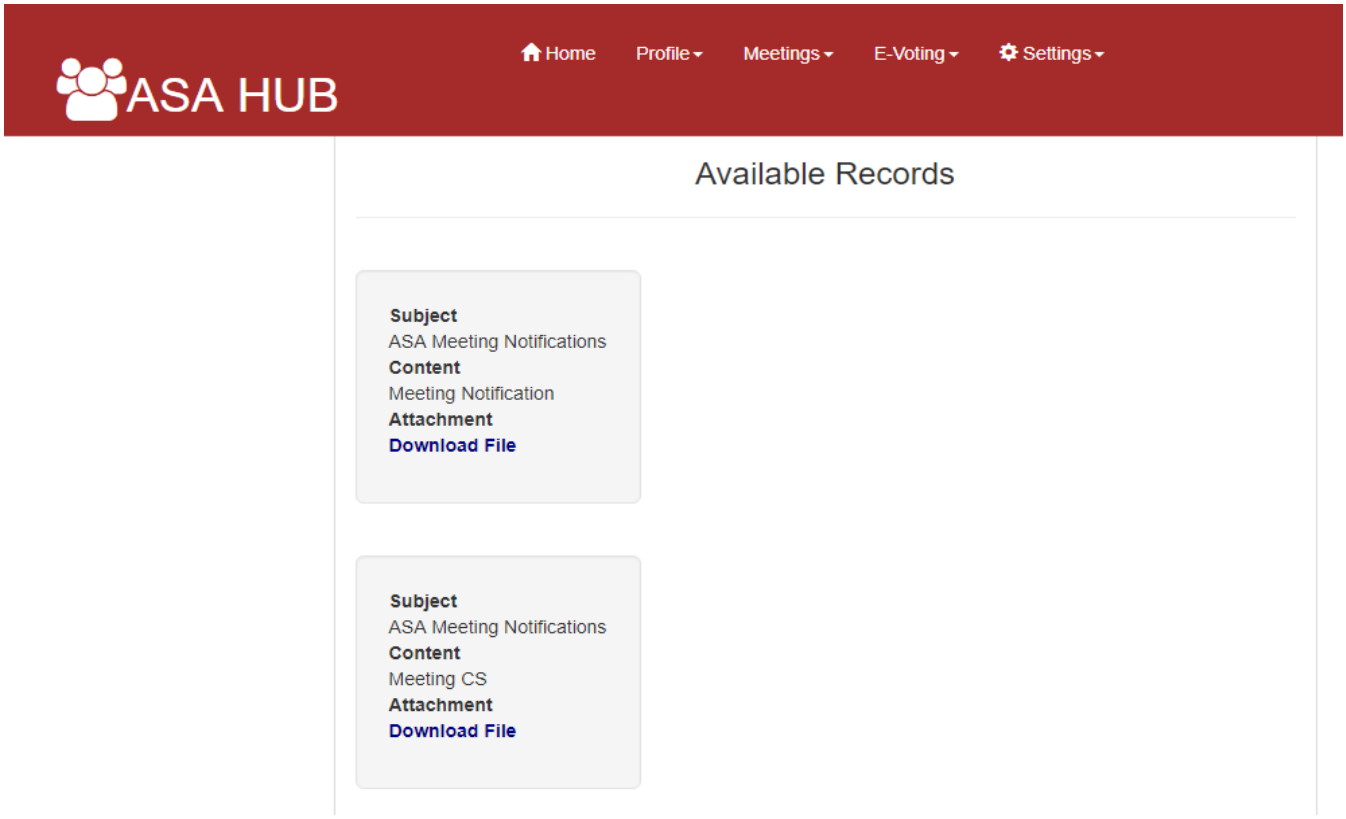

Figure 4. 3 View Notification Screen

4.4.4 Discussion Forum

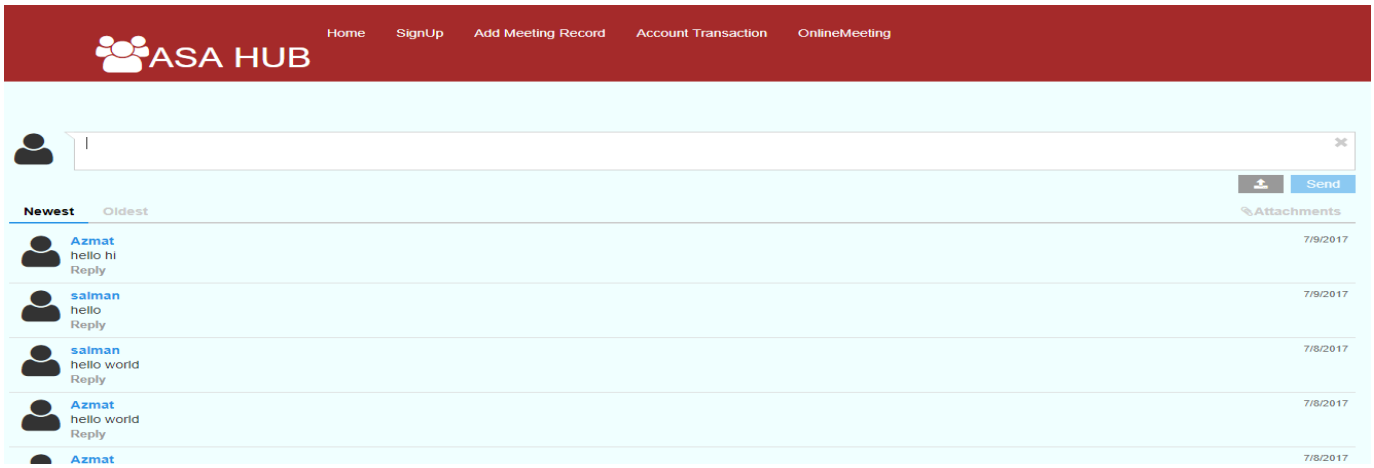

Figure 4. 4 Discussion Forum Screen

4.4.5 View Account Transaction

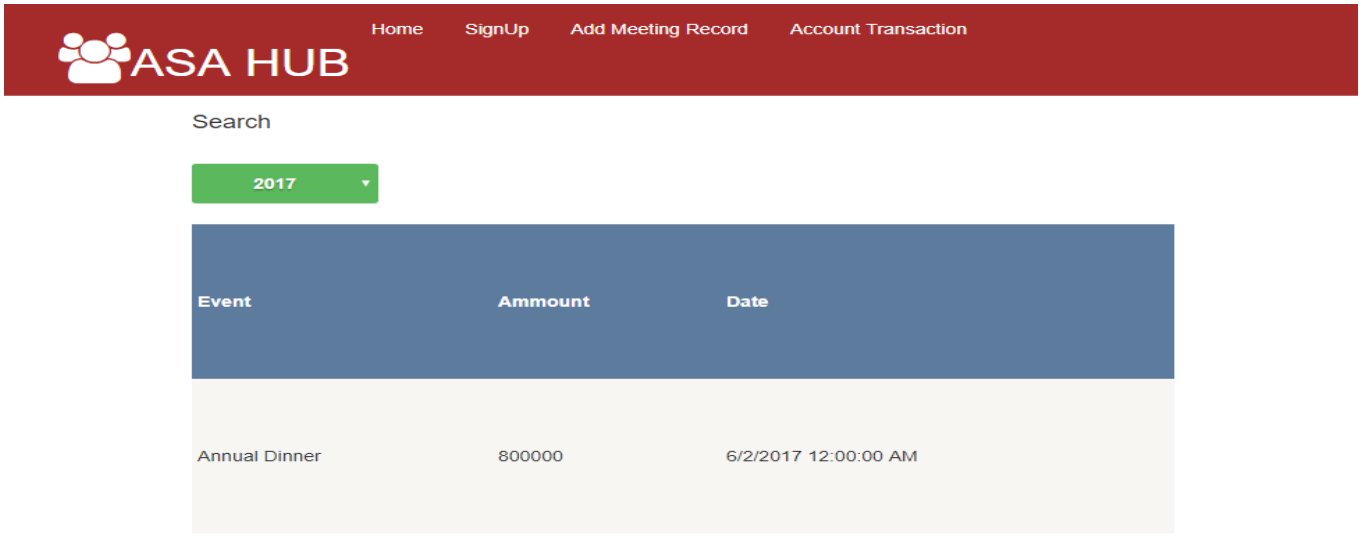

Figure 4. 5 View Account Screen

4.4.6 Manage Account

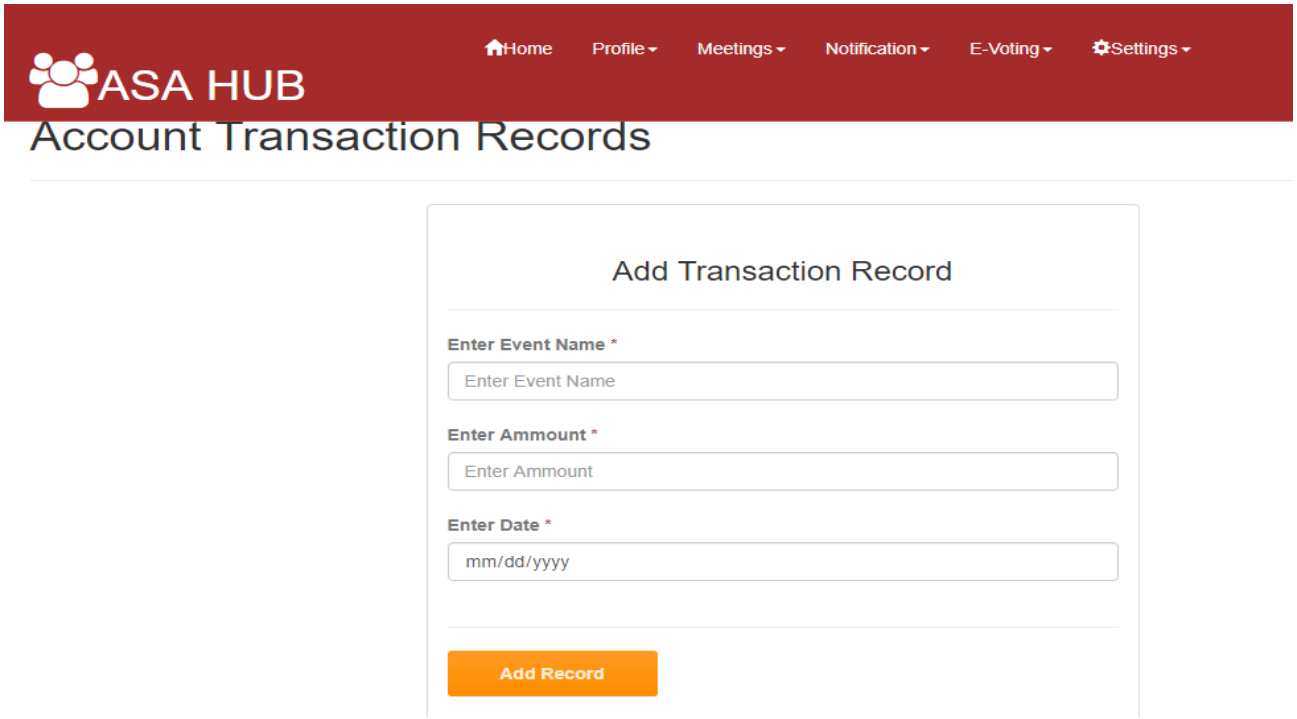

Figure 4. 6 Manage Account Screen

4.4.7 Sending Meeting Token

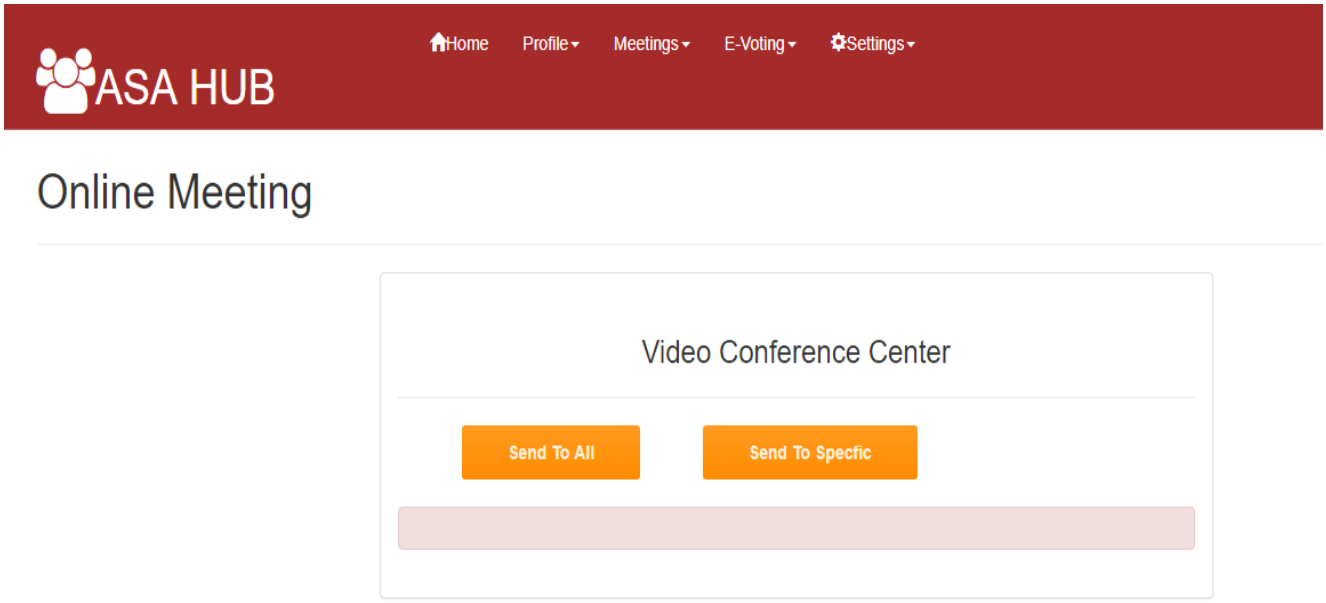

Figure 4. 7 Meeting Token Screen

4.4.8 Validate Token

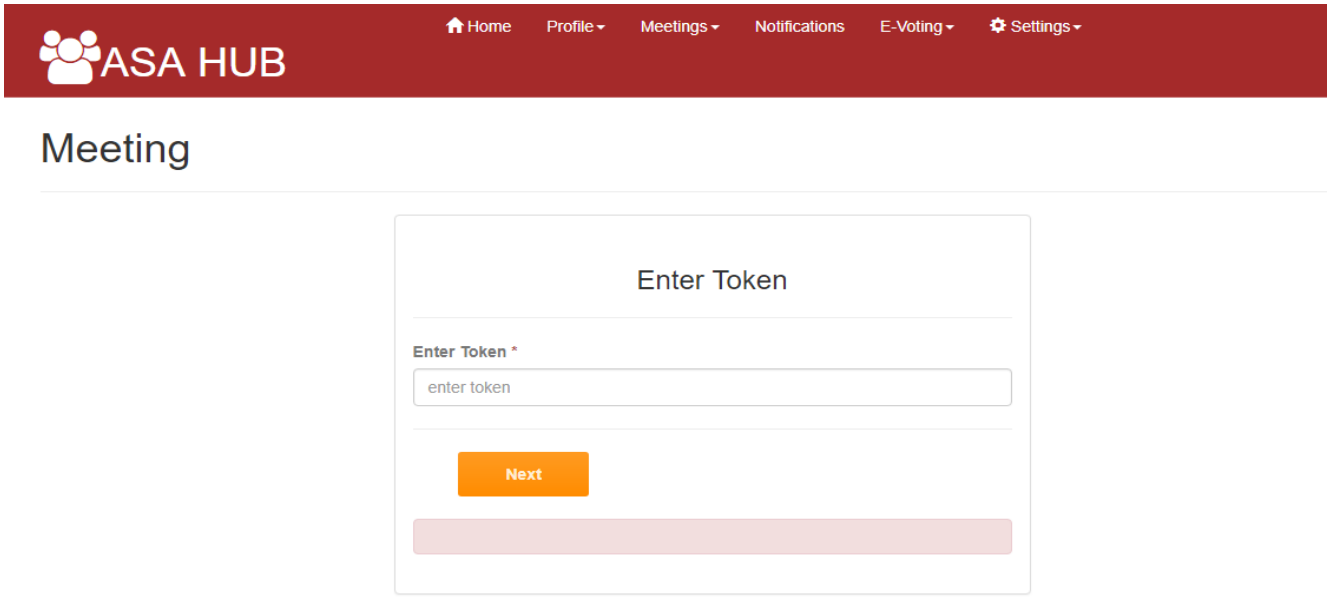

Figure 4. 8 Validate Token Screen

4.4.9 Start Meeting

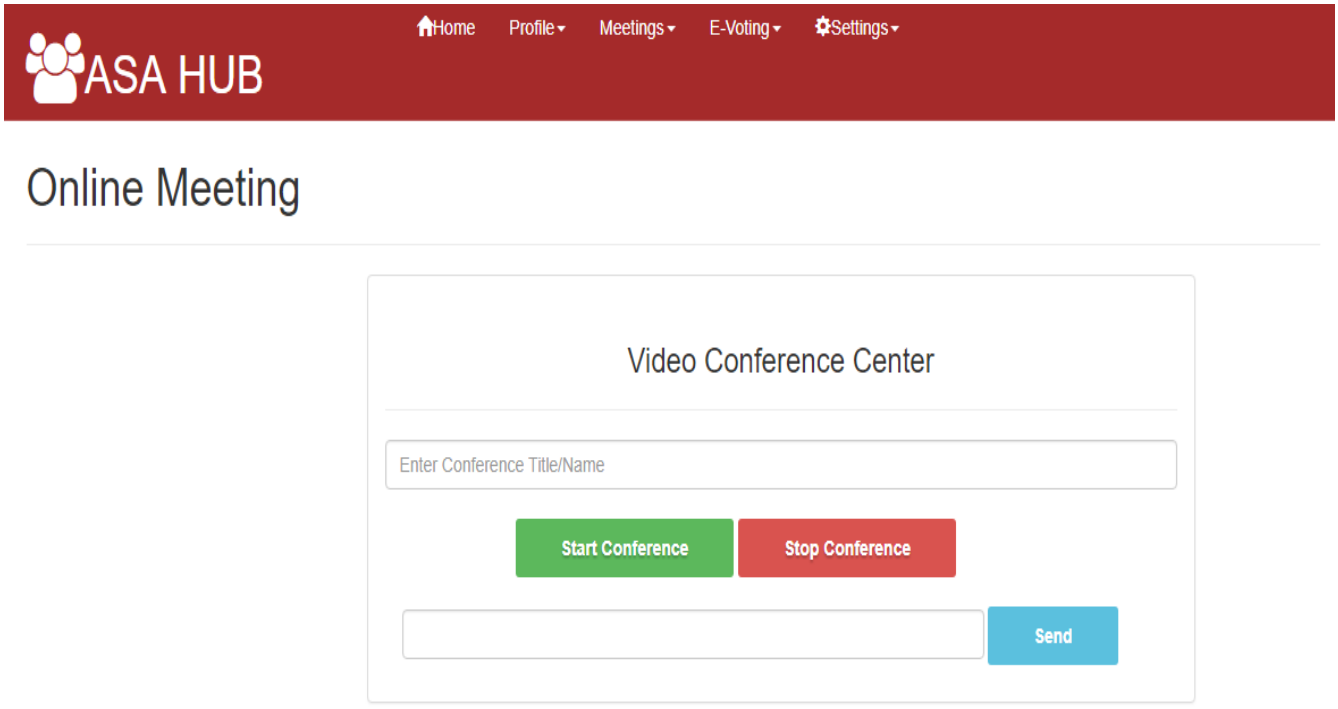

Figure 4. 9 Start Meeting Screen

4.4.10 Join Meeting

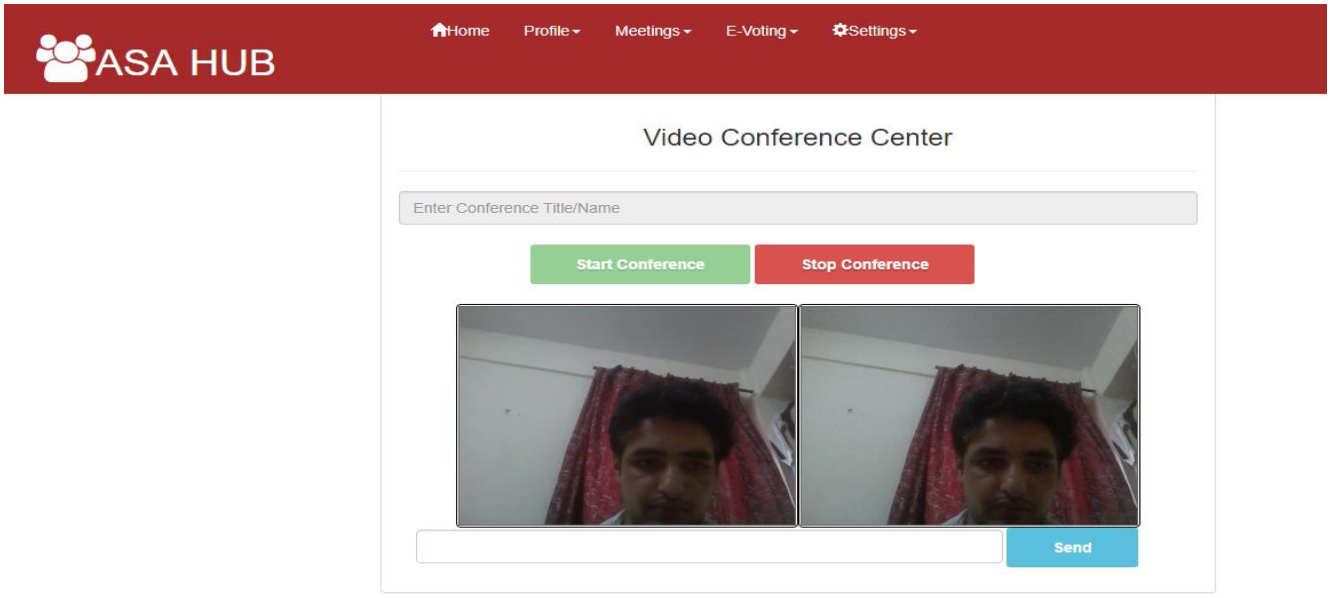

Figure 4. 10 Join Meeting Screen

4.4.11 Leave Meeting

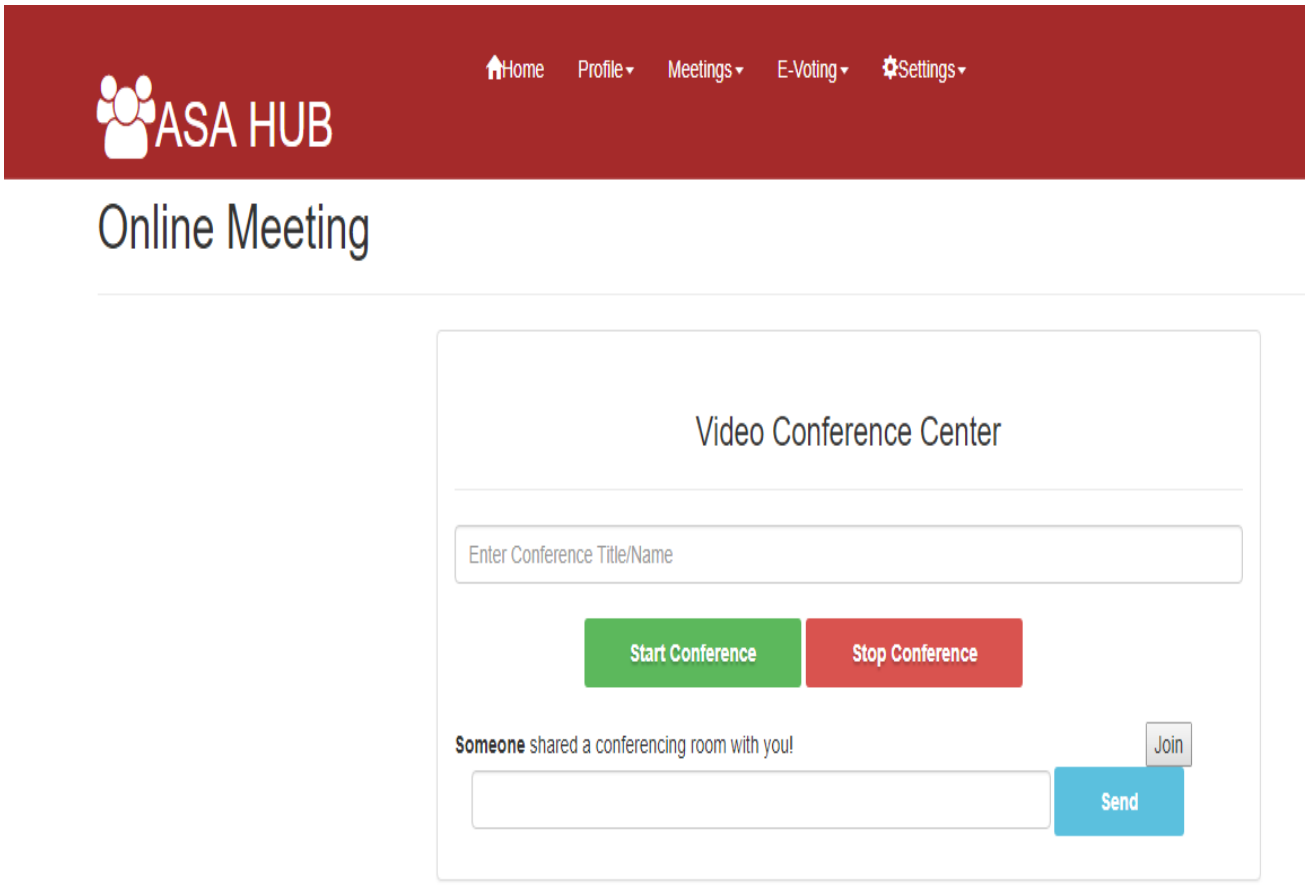

Figure 4. 11 leave Meeting Screen

This chapter has given complete description of system implementation. It has elaborated the operational and functions of the system. This chapter further explained the different approaches, language selection, tools and environment for complete implementation a system and finally contained the details description of frame work selected, software used and screen shots of the application.

*"To teach is to learn twice" Joseph Joubert (*1754-1824*)*

Chapter 05

Software Test Document

This chapter describes software testing processes. This chapter further elaborates the acceptance test cases which are used to test the functional and non-functional requirements after coding of software.

5.1 Introduction

Testing can only shows the presence of errors in the program. It cannot demonstrate that there are no remaining faults [9]. Software test document involves the documentation of artefacts that should be developed before or during the testing of software. Software testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

5.1.1 Test Approach

The testing approach describes the process of conducting a test. There are two major approaches in which the testing can be done. One on be basis of execution and other is on the basis of application level. On the basis of executions contains static testing, symbolic testing and dynamic testing. On the basis of application level contains system testing and unit testing. Static testing of program is done without executing the program. It is typically done by a compiler which checks for syntax errors and control flow errors such as unreachable code.

Symbolic testing is carried out by providing symbolic inputs to the software and executing the code by symbolically evaluating the program variables. Dynamic testing requires execution of the program using input data. Here the usual approach is to select the input data values such that desired control paths are executed. Since there can be infinite number of control paths in a program, dynamic test cases are designed to satisfy a minimal number of conditions that indicate the extent of control paths or alternative criteria that are covered in the test cases. System testing is carried out for the entire application and verifies that the product an assemblage of components works as a cohesive whole to satisfy the user requirements. Unit testing, on the other hand, carries out tests at the component (unit) level.

The selected approach for testing is unit testing. In unit testing is the process of testing program components, such as methods or object classes [10]. A unit test case provides the input parameter values and also provides the expected results when the code is executed. The unit test is carried out to verify the results of the module against the expected results.

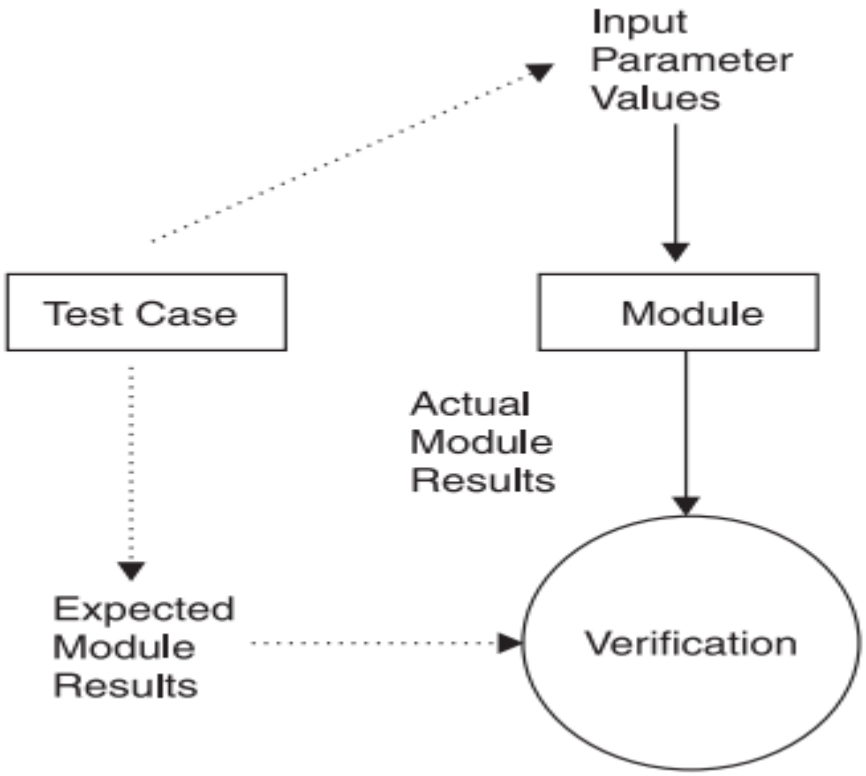

Figure 5.1 The Unit Test

Unit testing contains functional (black box) testing and structural (white box) testing. Black box testing are those that do not make use of knowledge of the internal logic of the module is unknown. Testing can be made on interface, testing based on function to be computed and input domain testing. White box testing use internal logic of the modules and it can be system testing, branch testing, and conditional testing.

5.2 Test Plan

Test planning is an activity that ensures that there is initially a list of tasks and milestones in a baseline plan to track the progress of the project. Test plan determines the scope and the risk that need to be tested and are not to be tested. Deciding fail and pass criteria.

5.2.1 Features to be tested

Features of the system that are to be tested using a specific approach. These features includes, login (admin/user), add post, add comment, update post, delete post, add notification, update notification, view notifications, join and leave meeting and manage account transaction and view account transaction.

5.3 Test Cases

A test case is a document, which has a set of test data, preconditions, expected results and post conditions, developed for a particular test scenario in order to verify compliance against a specific requirement. Test Case acts as the starting point for the test execution, and after

applying a set of input values, the application has a definitive outcome and leaves the system at some end point or also known as execution post condition [11].

5.3.1 Login

ID	T1
Description	User, Admin and treasurer can login to the system. It shows that login of user, admin or treasure can only be possible if username and password are correct.
Actor	User/Admin
Setup	1. Register admin with Username: Usman Password: usman001. 2. Register an user with Username: Hassan Password: hassan001 3. Register an treasurer with Username: Kamran Password: Kamran 05
Inputs:	1. Enter Username: Usman and usman001 2. Press login 3. Enter Username: Hassan and hassan001 4. Press login 5. Enter Username: Kamran and Kamran001 6. Press login
Expected Results	Usman as admin should log in successfully Hassan as user should log in successfully Kamran as treasurer should log in successfully
Observed Result	Usman as admin should log in successfully Hassan as user should log in successfully Kamran as treasurer should log in successfully
Frequency	Frequency of occurrence 5 or 6
Result/Verdict	Passed

5.3.2 Add Topic

ID	T2
Description	Topic or post for discussion will be added to the system. User, admin and treasurer can all post for discussion.
Actor	User
Setup	Login as user, admin and treasurer
Inputs:	1. Selects discussion forum 2. Selects add post 3. Enter description 4. Press Add post button or press Enter (Keyboard)
Expected Results	Post will be uploaded to the system
Observed Results	Post will be uploaded to the system
Frequency	Frequency of occurrence 5 or 6
Output/Verdict	Passed

5.3.3 Leave Comment

ID	T3
Description	Comment on added topic will be added to the system. User, admin and treasurer can all post for discussion.
Actor	User
Setup	Login as user, admin and treasurer
Inputs:	1. Selects discussion forum 2. Selects comment 3. Enter comment 4. Press Add comment button or press Enter (Keyboard)
Expected Results	Comment will be uploaded to the system
Observed Results	Comment will be uploaded to the system
Frequency	Frequency of occurrence 5 or 6
Output/Verdict	Passed

5.3.4 Add Notification

ID	T4
Description	Topic or post for discussion will be added to the system. Admin can only add notification.
Actor	Admin
Setup	Login as admin
Inputs:	1. Selects Notification 2. Selects add notification 3. Enter description(id, date, subject, detail description) 4. Press Add Notification button.
Expected Results	Post will be uploaded to the system and will be sent to all user
Observed Results	Post will be uploaded to the system and will be sent to all user
Frequency	Frequency of occurrence 5 or 6
Output/Verdict	Passed

5.3.5 Update Notification

ID	T6
Description	Notification will be deleted from the system. Admin can only delete the notification
Actor	Admin
Setup	Login as admin
Inputs	1. Selects Notification 2. Selects delete notification 3. Enter date or id 4. Press delete notification button
Expected Results	Notification will be deleted from the system.
Observed Results	Notification will be deleted from the system.
Frequency	Frequency of occurrence 5 or 6
Output/Verdict	Passed

5.3.6 Join Meeting

ID	T7
Description	User, admin and treasurer will join meeting
Actor	User
Setup	Login as admin, user or treasurer
Inputs	1. Selects Online Meeting 2. Selects join meeting

	3. Enter ip address or id 4. Press join meeting button.
Expected Results	User will able to join meeting
Observed Results	User will able to join meeting
Frequency	Frequency of occurrence 5 or 6
Output/Verdict	Passed

5.3.7 Manage Account

ID	T8
Description	Details of account will be added to the system. Admin can only manage account.
Actor	Admin
Setup	Login as admin
Input	1. Selects Manage Account 2. Selects Add details 3. Enter description(date, event, amount) 4. Press Add details button
Expected Results	Details will be uploaded to the system.
Observed Results	Details will be uploaded to the system.
Frequency	Frequency of occurrence 5 or 6
Output/verdict	Passed

5.3.8 View Account Details

ID	T9
Description	Details of account will be displayed to user. Admin, user and treasurer can view the account details
Actor	User
Setup	Login as admin, user or treasurer
Input	1. Selects Manage Account 2. Selects view account details 3. Enter description(date) 4. Press view account detail button
Expected Results	Details of the account will be displayed to the user
Observed Results	Details of the account will be displayed to the user
Frequency	Frequency of occurrence 5 or 6
Output/Verdict	Passed

This chapter has given complete description of software testing of the system. It elaborated the test case approach, test plan and test case of the features that to be tested. This chapter further has explained the testing approaches, tools and environment for testing a system and finally contained the details description of test cases of the product that to be built.

"Education is a progressive discovery of our own ignorance" Will Durant (1885-1981)

Chapter 06

Conclusion and FutureEnhancement

This system is useful for the ASA members to automate their manual work. This system almost most of the covers all the requirements of the ASA. The basic objective of ASA is to promote social, educational and cultural activities, to collaborate with other organizations in furthering the interest of education. ASA works for to promote the interest of members and to organize discussions and lectures. The "ASA HUB" project is a group project and has been assigned to a group of two students. The components of the project are based on modules, namely, members profile management, discussion forum, virtual/ online meetings, electronic-voting, meetings record (minutes) management, notification management and account management. From these components notification management, discussion forum, account transaction management and online/virtual meetings modules are part of this phase of project.

The ASA HUB portal will facilitate the elected body of ASA and its members in a number of ways by providing services such as discussion forum in which member can discuss academic activities and other activities determined by ASA. This project provides basic facilities to the members of ASA. By using this project members of ASA can perform following functions including discussion forum, for discussing issues, topics related to academic activities and other general issues. Member can also use notifications for sending and receiving or viewing notifications. Admin can send notifications to all and a specific user, and other users can only view notifications. Notifications will also be sent to a specific user as well. Admin can send notification to send to all or send to a specific. In notification send to all, notification will be sent to all registered member of the user. In case of send to a specific, before sending notification admin will select email id of respective user, therefore, notification will be send to a specific user. Notification badge will be show to the user according to notification send. Notification send to specific user will only show to that user. User can view notifications.

In order to resolve difficulties in meeting, user can participate in meeting by using online or virtual meeting. Admin, president of the ASA will initiate the meeting, Admin send a four digit unique token to the members by sending sms or email. A four digits unique token will be sent to the user. Before joining the meeting user first enter the received token either on mobile, through sms or email. If user enters correct token, then system will allow user to joint meeting. Similarly user can also view and manage account transaction. System will also maintain the report of budget of ASA.

Future Enhancement

It can be possible to enhance this application in future in many ways. In the case of discussion forum, further improvement can be made regarding layout. In this application discussion forum is aimed to develop like of Facebook and more functionality like of Facebook can be added to this module. Similarly in case of notification system, send and view notification are done successfully and search notification can also be implemented in future. In the case of online meeting, further improvements can be made in such ways, many user will join the meeting, and user name will be shown. In future, signal strengthen of different connection can also be shown. In addition to this, many user will join the meeting and user can view image of the member that will speaking at that instance.

References

[1] P. K.J. Mohapatra, Software Requirement Specification, in Title of Software Engineering A lifecycle Approach, 5ed, New Delhi, India, New Age International Publishers, 2010, Chapter 10, Sec.10.3, PP. 228-229

[2] C. Larman, Use Case Model: Writing Requirement in Context, in Title of Applying UML and Patterns. 2nd ed, Toronto, Canada, Prentice Hall, Chapter 6, Sec.6.4, PP.61-64

[3] R.S. Pressman, Design Concepts and Principles, in Title of Software Engineering A Practitioner's Approach, 5ed, New Delhi, India, Thomas Casson, Chapter 13, Sec 13.3, PP. 513-519

[4] C. Larman, Design Model, in Title of Applying UML and Patterns. 2nd ed, Toronto, Canada, Prentice Hall, Chapter 17, Sec.17.3, PP.261-264

[5] P. K.J. Mohapatra, Concept underlying Software Architecture, in Title of Software Engineering A lifecycle Approach, 5ed, New Delhi, India, New Age International Publishers, 2010, Chapter 16, Sec.16.3, PP. 379-380

[6] C. Larman, Domain Model, in Title of Applying UML and Patterns. 2nd ed, Toronto, Canada, Prentice Hall, Chapter 7, Sec.7.7, PP.117-119

[7] P. K.J. Mohapatra, Class Diagrams, in Title of Software Engineering A lifecycle Approach, 5ed, New Delhi, India, New Age International Publishers, 2010, Chapter 14, Sec.14.5, PP. 321-336

[8] I.Sommerville, Software Testing, in Title of Software Engineering, 9th ed, USA, Pearson Education Addision Wesley, Chapter 8, Sec.8.2, PP.221-224

[9] R.S. Pressman, Software Testing Techniques, in Title of Software Engineering A Practitioner's Approach, 5ed, New Delhi, India, Thomas Casson, Chapter 17, Sec 17.3, PP. 482-486

[10] C. Larman, Component Diagram, in Title of Applying UML and Patterns. 2nd ed, Toronto, Canada, Prentice Hall, Chapter 38, Sec.38.1, PP.591-596

[11] P. K.J. Mohapatra, Software Testing Techniques, in Title of Software Engineering A lifecycle Approach, 5ed, New Delhi, India, New Age International Publishers, 2010, Chapter 19, Sec.19.5, PP. 406-415