FYP Automation System for SE Department

Nafeesa 11540
Salman Khan 11610
Ehtisham Ali 11636

Supervised By

Dr. Raheel Zafar

Submitted for the partial fulfillment of BS Software Engineering degree to the Faculty of Engineering

NATIONAL UNIVERSITY OF MODREN LANGUAGES

ISLAMABAD

March, 2021

ABSTRACT

Final year project is a difficult and challenging task for the students. They are busy in their projects along with different courses. So time management is the key issue for them. Most of the graduate students face a lot of problems during the final year of their degree. The students have to meet the final year project committee members and supervisor repeatedly for guidance. Student's does not have the schedule of supervisors and project committee members. It is difficult for the students to trace the project committee, and they have to check project committee again and again. Students have to select a unique idea for FYP (Final year project) which require visiting thesis section which is very time consuming. In addition to that students have frequent queries about eligibility criteria for FYP and guidelines related to documentation, availability of supervisors and many more. The final year project committee members and supervisors face equal problems in this regard. The students barely show up to present their work to the supervisors.

As a matter of fact, the traditional method of dealing with final year projects is not effective enough to produce required results, therefore a system is needed to solve the issues of students, supervisors and final year project committee members and committee head will help to resolve the issues. It is a web based system that will help the students through automating FYP standard operating procedures.

The web application has been developed using Node Js and react native. Mongo DB has been used for the development of database connection and storing data.

Final Approval

It is certified that project report titled '**FYP Automation System for NUML**' submitted by **Nafeesa, Ehtisham Ali, Salman Khan** for the partial fulfillment of the requirement of **"Bachelor's Degree in Software Engineering"** is approved.

COMMITEE

Dr. Basit Shehzad
Dean Engineering & CS Signature: _____

Dr. Muzaffar Khan
HoD Engineering Signature: _____

Mr. Naveed Ahmed
Head Project Committee Signature: _____

Dr. Raheel Zafar
Supervisor Signature _____

DECLARATION

We hereby declare that our dissertation is entirely our work and genuine / original. We understand that in case of discovery of any PLAGIARISM at any stage, our group will be assigned an F (FAIL) grade and it may result in withdrawal of our Bachelor's degree.

1. Nafeesa _____
2. Ehtisham Ali _____
3. Salman Khan _____

TABLE OF CONTENTS

Chapter **Page**

Chapter 1: Introduction 1

1.0 Introduction 2

1.1 Motivation 2

1.2 Problem Statement 2

1.3 FYP Automation System For SE Department 2

1.4 Goals and Objectives 3

1.5 Scope of the Study 4

1.6 Nature of the Project 4

1.7 Overview of the Report 5

Chapter 2: Background and Existing Work 6

2.0 Introduction 7

2.1 Important Constructs of the Application Domain 7

2.1.1 Node Js 7

2.1.2 Client Server Architecture 7

2.1.3 Mongo DB 7

2.2 Existing Systems and Their Limitations 8

2.2.1 Project Management System 8

2.2.1.1 Features 8

2.2.2 Limitations of Learning Management System NUML 8

2.2.2.1 Features 8

2.3 Limitations of the Existing Systems 9

2.3.1 Limitations of Project Management System 9

2.3.2 Limitations of Learning Management System NUML 9

2.4 Comparison of Existing System and Proposed System Features 9

2.5 Summary 10

Chapter 3: Requirements Specification 11

3.0 Introduction 12

3.1 Interface Requirements 12

3.2 Hardware Interface Requirements 14

3.2.1 Software Interface Requirements 15

3.2.1.1 NodeJs 15

3.2.1.2 Mongo DB 15

3.3 Functional Requirements....16
3.4 Use Cases....16
3.4.1 Use Case Diagram....17
3.4.2 Login Use Case....20
3.4.3 Register Use Case....21
3.4.4 Manage Group Invitation Use Case....22
3.4.5 Make Group....24
3.4.6 Submit Document Use Case....25
3.4.7 Review Document Use Case....26
3.4.8 Chat Use Case....27
3.4.9 Manage Meeting Log Use Case....29
3.4.10 Guidelines Use Case....30
3.4.11 View Group Details Use Case....31
3.4.12 Check Meeting Log....32
3.4.13 Project Evaluation....32
3.4.14 Manage List....32
3.5 Non-Functional Requirements....32
3.5.1 Security....32
3.5.2 Safety....32
3.5.3 Flexibility....33
3.5.4 Testability....33
3.5.5 Maintainability....33
3.5.6 Portability....33
3.6 Resource Requirements....33
3.6.1 Hardware Requirements....33
3.6.2 Software Requirements....33
3.6.2.1 Node Js....34
3.6.2.2 Human Effort....34
3.7 Database Requirements....35
3.7.1 Mongo DB....35
3.8 Project Feasibility....36
3.8.1 Technical Feasibility....36
3.8.2 Operational Feasibility....36
3.8.3 Legal & Ethical Feasibility....36
3.9 Summary....36
Chapter 4: System Modelling…....38
4.0 Introduction....39

4.1 High Fidelity Prototype 40
4.1.1 Dashboard 41
4.1.2 Login 42
4.1.3 Add Guidelines 43
4.1.4 Register 44
4.1.5 Student List 45
4.1.6 Supervisor List 46
4.1.7 Make Group 47
4.2 Design Approach 39
4.2.1 Top down Design Approach 39
4.2.2 Bottom up Design Approach 38
4.3 Logical View 44
4.3.1 Class Diagram 44
4.3.2 Entity Relation Diagram (ERD) 45
4.4 Process View 46
4.4.1 Activity diagram 46
4.4.1.1 Manage List activity diagram 47
4.4.1.2 Project Evaluation activity diagram 47
4.4.1.3 Login Activity Diagram 47
4.4.1.4 Register activity 48
4.4.1.5 Manage Group Invitation activity 49
4.4.1.6 Make Group activity 50
4.4.1.7 Submit Document activity 51
4.4.1.8 Review Document 52
4.4.1.9 Chat 54
4.4.1.10 Manage Meeting Log 54
4.4.1.11 Guidelines 55
4.4.1.12 View Group Details 55
4.4.1.13 Check Meeting Log 55

4.5 State machine diagram....56
4.5.1 Login state machine diagram....56
4.5.2 Register state machine diagram....57
4.5.3 Manage Group Invitation diagram....58
4.5.4 Make Group diagram....59
4.5.5 Submit Document state machine diagram....60
4.5.6 Review Document state machine diagram....62
4.6 Sequence Diagram....66
4.6.1 Login Sequence Diagram....66
4.6.2 Register Sequence Diagram....67
4.6.3 Manage Group Invitation Sequence Diagram....67
4.6.4 Make Group Sequence Diagram....68
4.6.5 Submit Document Sequence Diagram....69
4.6.6 Review Document Sequence Diagram....69
4.6.7 Chat Sequence Diagram....69
4.6.8 Manage Meeting Log....69
4.6.9 Guidelines Sequence Diagram....69
4.6.10 View Group Details Sequence Diagram....69
4.6.11 Manage list Sequence Diagram....69
4.6.12 Repository Sequence Diagram....69
4.6.13 Project Evaluation Sequence Diagram....69
4.7 Development view....70
4.7.1 Component diagram....70
4.8 Physical view....71
4.8.1 Deployment Diagram....71
4.12 Summary....71

LIST OF FIGURES

Figure **Caption** **Page**

4. 1: Login 39
4. 2: Dashboard Prototype.....40
4. 3: Add Guidelines 40
4. 4: Register 41
4. 5: Student List 41
4. 6: Supervisor List 42
4. 7: Make Group 42
4. 8: Class diagram. 44
4. 9: Entity Relationship Diagram (ERD) 45
4. 10: Manage List Activity 46
4. 11: Project Evaluation activity 47
4. 12: Login activity 48
4.13: Registration…….....49
4. 14: Manage Group Invitation 50
4. 15: Make Group. 51
4. 16: Submit Document 52
4. 17: Review document. 53
4. 18: Chat 54
4. 19: Manage meeting Log. 55
4. 20: Check group details activity. 56
4. 21: Register. 57
4. 22: Add Dummy Face 58
4. 23:Add Filters. 59
4. 24: Displaying Prominent Parts. 60
4. 25: Displaying Filters. 61
4. 26: Sketch Image.....62
4. 28: Generate Report. 63
4. 29: System State Machine Diagram. 64
4. 22: Add Dummy Face 65
4. 23:Add Filters. 66
4. 24: Displaying Prominent Parts. 67
4. 25: Displaying Filters. 68
4. 26: Sketch Image.....62
4. 28: Generate Report. 63
4. 29: System State Machine Diagram. 64
4. 22: Add Dummy Face 58
4. 23:Add Filters. 59
4. 24: Displaying Prominent Parts. 60
4. 25: Displaying Filters. 61
4. 26: Sketch Image.....62
4. 28: Generate Report. 63
4. 29: System State Machine Diagram. 64

LIST OF TABLES

Table **Caption** **Page**

2. 1: Comparison of Existing System and Proposed System Features 9
3. 1: Login Use Case 20
3. 2: Register Use Case 21
3. 3: Manage Group Invitation Use Case 22
3. 4: Make Group Use Case 24
3. 5: Submit document Use case 25
3. 6: Review Document Use Case 26
3. 7: Chat Use case 27
3. 8: Manage Meeting Log Use Case 29
3. 9:Guidelines 30
3. 10: View Group Details Use case 31
3. 11: Check meeting Log……32
3. 12 : Project Evaluation……..…..32
3. 13: Manage List……..…..33
3. 14: Task gantt chart…….….34
3. 15: Work Division…….…..35

CHAPTER 1

INTRODUCTION

1.0 Introduction

This chapter delivers introduction to the system report. It provides proper explanation to goals, views, objectives and scope statement of the system. This chapter provides the information about the objectives behind development of the system. The advancement inn technology and the ongoing pandemic has increased the need for an automated Final year project handling system, that can automate the manual processes related to the final year projects of software engineering department. The manual approach use in the university and its drawbacks became more clear in the pandemic where every mode of education needed to be automated also the departments increasing strength and the increasing burden on supervisors and committee members requires a system that automate the core functionalities for the students and FYP's management. This chapter provides complete details about how FYP automation system will facilitate its users in monitoring, controlling and managing final year project related tasks.

1.1 Motivation

In the past few years' technology in educational institutes, which has greatly influenced the traditional approaches to data handling and managing students work. Still in developing countries traditional approaches were used for managing the final year projects of students until the recent pandemic came in. Institutions and government authorities are now left with no other choice rather than developing appropriate system that helps students and supervisors work in a hybrid environment, while still being able to stay in complete touch with progress. "FYP automation system for Se department" aims to provide students, supervisors, committee members and heads with a platform where managing and monitoring the student's Projects and reports will be much easier and less time consuming than before, with less physical interaction.

1.2 Problem Statement

In the existing world online or technical interaction with students has become a necessity. With the increasing number of students in software engineering department and the final year projects by those students, it has not become vital to have a system that automates the sops. Due to increasing responsibilities on supervisors and committee heads it's not possible to manually go through every project or manually arrange meetings for this purpose. Looking at all these issues it is really vital to have a system that does it for the stakeholders.

With "FYP automation system for Se department" it would become easier and less time consuming for both students and supervisors to manage the final year projects. Supervisors will now have an ease to impose all the guidelines and view progress with in an automated way rather than wasting time of arranging physical meetings.

1.3 Goals and Objectives

The goal of "FYP automation system for Se department" is to help the FYP committee head, final year project supervisors and students to automate the manual processes and provide ease. The objectives of "FYP automation system for Se department" is to automate the sops related to final year project of students, providing better control over student's projects to supervisors and committee members.

1.4 Scope of the Study

"FYP automation system for Se department" is a web based application. Through this developed system the software department can easily manage the sops related to final year projects. This will allow the students and other stakeholders involved to perform different activities using the same system. It has features i.e. Evaluation, Deadline imposing, guidelines uploading. The features will make sure everything done manually before will now be automated. The deliverable of this system is web based application.

1.5 Process Model

Incremental model is followed in the development of " FYP Automation System For SE Department". In incremental model an incremental approach is used in developing the system, system is built in iterations and every iterations ends up in usable system. Moreover, the iterations can be easily tested and the system is developed entirely step by step. Incremental development is performed step by step from analysis, designing, implementation, testing, maintenance. Incremental Model is the process model in software development in which all the requirements are broken down in atomic or standalone modules. This model is used for " FYP Automation System For SE Department" because usable software is produced earlier and it is easier to integrate requirement changes afterwards. The process of testing and removing bugs is also simpler in the model.

1.6 Nature of the Project

"FYP automation system for Se department" is a web based application. The outcome is a web application that has two ends client and server. Node js along with react native is used for the development of the web application. The system is being supported by database at server end.

1.7 Overview of the Report

The chapter is related to introduction, scope, objectives, problem statement, process model used and purpose behind developed system. The second chapter contains thorough details related to the problem domain, the problem domain existing systems available and their comparison with the proposed system " FYP automation system for SE department".

The third chapter is related to beginning of use case modelling, functionalities, assorted and nonfunctional system requirements of "FYP Automation System For SE Department" has been explained. Chapter four is related to the details about designing and modelling of "FYP Automation System For SE Department" with the use of 4+1 view model, that is supported by UML diagrams.

CHAPTER 2

BACKGROUND AND EXISTINGWORK

2.0 Introduction

A vast number of systems have been developed to support educational institutes around the world, meaningful progress has taken place in developing distance learning and different systems have been developed across the globe. In this chapter abstract of existing systems are discussed.

2.1 Important Constructs of the Application Domain

2.1.1 Node Js

Node Js platform is used for the GUI. Node Js is an open source thus easily accessible platform. It can work on numerous operating system i.e. Windows, Macos, Linux etc. This desktop application is developed using this platform due to its convenient nature.

2.1.2 Client Server Architecture

The client server architecture is developed for the system based on the kind of requirements defined earlier.

2.1.3 Mongo DB

JavaScript is used as query language in Mongo DB which is more convenient with the kind of data we have. The rapidly growing data requires this database design. It's a document database which stores data in Jason-like documents.

2.2 Existing Systems and Their Limitations

There are a few existing systems that provides the students project management facilities but they a lot of restrictions and flaws. Those systems are discussed below:

2.2.1 Project Management System

The project management system was developed to handle and monitor student's projects. The Project Management System provides some basic functionalities related to user records and project management task like add, delete and update. This application is not suitable for FYP management.

2.2.1.1 Features

- Add, Delete, Update user record.
- Add, Delete, Update project record.

2.2.2 Learning Management System NUML

The LMS for NUML is very suitable and efficient system for providing online management facilities but it does not provide any sort of facility to user in terms of FYP management.

2.2.2.1 Features

- Students/teacher management
- Course enrollment.
- Uploading Assignments quizzes

2.3 Limitations of the Existing Systems

The major limitation or setbacks of the existing systems are as follows:

2.3.1 Limitations of Project Management System

Not suitable for FYP Management. Providing some basic feature that are limited for a management system Does not facilitate any FYP Committee Members and supervisor's. Do not provide any sort of guidelines and seminars related to FYP and its documentation.

2.3.2 Limitations of Learning Management System NUML

It can't be used for long period of time without interruption, because its session expires in every 20 minutes and the user has to login again. It is suitable for online classes and online management of student's assignment. It doesn't provide any option regarding the management of final year project.

2.4 Comparison of Existing System and Proposed System Features

Table 2.1 describe the comparison between proposed system and existing system functionalities. The proposed system covers all the features while the existing system fails to perform all the features.

Table 2. 1: Comparison of Existing System and Proposed System Features

Features	**Project Management system**	**LMS NUML**	**Proposed System**
Authentication	✓	✓	✓
Evaluation	✗	✗	✓
Monitoring	✗	✗	✓
Scheduling	✗	✗	✓
Communication Modes	✗	✓	✓
Guidelines	✗	✗	✓
FYP archives	✗	✗	✓
Dashboard	✗	✓	✓

2.5 Summary

In this chapter the vital constructs used in developing the system are explained. The current systems including project management system and Learning Management System are explained along with features and limitations.

CHAPTER 3

REQUIREMENTS SPECIFICATION

3.0 Introduction

In this chapter system modelling is done and use case diagram is designed. System modelling is the predicting different view of the system by developing abstract models of the system. Every model represents a specific view or prospective of that system. System modelling done here by using some kind of graphical notation, that graphical notations are based on notation in the Unified Modelling Language (UML).

System modelling provides analyst comprehensive understanding of the system and it helps the developer in understanding the working flow of the software and modelling approach which are used to interact with users. The system design reveals that in which environment system works. The belief of modeling is really helpful in the progress of the system as it serves to have a traditional method to discuss the ideas underlying the abundant various types of modelling.

3.1 Interface Requirements

The client-server architecture is developed for the system.

3.1.1 Hardware Interface Requirements

The following hardware requirement for development and testing of the system are required.

- Laptop/Desktop
- RAM (Random Access Memory) 4GB or higher
- Dual-core 2.5 GHz processor or higher
- Hard drive minimum 300GB
- Window 7.0 or above

3.1.2 Software Interface Requirements

The required software interface requirements for the system are:

3.2.2.1 Node JS

Node js is based on JavaScript runtime environment that executes JavaScript code. Node Js

is an open source platform, which makes it easily available without wasting resources. Complexity of writing server applications is reduced in Node Js through APIs. Primarily Node Js is used to build web servers. The functionalities of Node Js are unblocking. Node Js support different operating systems including Macos, Linux, Windows and UNIX servers.

3.2.2.2 Mongo DB

Mongo Db is an open source, document orientated and cross platform database program. Mongo DB is classified as a NoSQL database. Json-like documents are used with schemas for data storage in Mongo DB, which is developed by Mongo DB Inc. Mongo DB supports many features including Indexing, replication, load balancing, file storage, aggregation, capped collections and transactions. This database approach is more suitable for rapidly changing data.

3.2 Functional Requirements

The Basic functionalities of the system are:
System shall allow to upload images. System shall only allow authorized access. System must be able to decrease the size of images. Application should be able to compare new image with existing images. System shall crop most prominent face parts. System should be able to compare prominent parts with present prominent parts and save unmatched parts. System shall display prominent parts. System shall be able to drag and drop prominent parts to make a sketch. System should be able to

3.3 Use Cases

The use cases are used to represent discrete functionality of a system or package. How the external actors communicate with the system is called use case. The high level functionalities of a system and the interaction of user with them. Use cases are the main ideas or concepts of Unified Modelling Language.

Use case can be explained as collection of defined sequences which may include every possible viable state of software for the purpose of circulating implementation controls.

Use cases help the development team to understand each information point and requirements of the system to be developed. Use case writing contain multiple steps which include following: Name, Brief description, Preconditions, Post conditions, Actors, Basic flows and Exceptional flows.

To achieve a specific goal, the communication that needs to be taken place between system and actors is known as use case. Actors might not necessarily be humans an actor can be a hardware or software as well that is interacting with a system. Every stakeholder cannot be classified as an actor but every actor can be classified as a stakeholder.

Detailed and well-designed UML USE case diagrams are really vital in successful completion of a project. Use cases gives an overview and idea of the system its main stakeholders and the interaction between them.

3.4 Use Case Diagram

Use case diagram of the developed system is explained below.

This figure 3.1 describes the system modules including Make Group, Project Evaluation, Guidelines, Manage Group Invitation, Meeting Log, Fyp groups etc.

3.4.1 Login Use Case

Table 3.1 describes how the Admin/user login to the system. This use case does not require any pre- conditions.

Table 3. 1: Login Use Case

Use Case – 1	Log in Use Case		
Goal	This use case explain how the user logs in to the system.		
Brief Description	User need to have an account to perform activities and viewing different information.		
Pre-Conditions	User must be registered before logging in.		
Post Conditions	Dashboard displayed successfully.		
Failed End Condition	The user isn't able to login.		
Primary, Secondary Actors	Admin, Student, Supervisor.		
Dependency	Includes Register Use Case.		
Basic Flow	**Steps**	**Actions**	**System Response**
	1	The user starts the system to login.	The system asks the user to return user name and password.
	2	The user enters his/her login credentials i.e. username and password.	The System checks if the entered password and username is correct.

	3	The user is now logged in and directed to dashboard.	System now allows the user to see confidential and authorized information access.
Alternative Flow	**Steps**	**Actions**	**System Response**
	2a	Invalid password/ Username, the user cannot login and is informed of a login error.	The system will ask user to re-enter the specific credential and add a correct one.

3.4.2 Register Use Case

Table 3.2 Explains logging in to the system. Register use case has no pre condition and only admin can register users.

Table 3. 2: Register Use Case Table

Use Case – 2	Register use case.
Brief Description	Admin registers new user into the system.
Goal	To register a new user.
Pre-conditions	Null.
Post conditions	The admin fills registration form credentials and adds a new user.
Failed End Condition	Registering user isn't successful.
Primary,	Admin

Secondary Actors			
Dependency	None		
Basic Events Flow	**Steps**	**Actions**	**System Response**
	1	Admin opens the application.	The System asks the Admin for register new account.
	2	The Admin chooses register option.	The System asks admin for registration related information, i.e. Username, password, etc.
	3	The admin provides the required information.	Admin authenticates information and account of user is created.
Alternative Flow	**Steps**	**Actions**	**System Response**
	3a	Admin selects the cancel option	The admin is redirected by the system to login page.
	3b	Admin doesn't confirm the registration process.	Registration is denied.

3.4.3 Manage Group Invitation Use Case

Table 3.4 shows how the management of invitation by the supervisor.

Table 3. 3: Manage Group Invitation Use Case

Use Case – 4	Manage Group Invitation.		
Goal	Supervisor wants to accept or decline group invitation.		
Brief Description	This use case deals with the management of invitation by the supervisor.		
Pre-conditions	Supervisor must be logged in.		
Post conditions	Supervisor will be added into a new group if accepts otherwise the invitation list will be updated.		
Failed End Condition	The supervisor couldn't update the invitation list.		
Primary, Secondary Actors	Supervisor Database		
Dependency	Includes login use case.		
Basic Flow	**Steps**	**Actions**	**System Response**
	1	Supervisor logs in to the system.	System display dashboard to supervisor.
	2	Supervisor selects group invitation option.	The system displays list of invitations received by the supervisor.
	3	Supervisor accepts the invitation.	The systems updates the group list and adds the specific group in group

			details.
	4	Supervisor declines the invitation.	The systems updates the group invitation list and removes the specific invitation from request list.
Alternating Flow	**Steps**	**Actions**	**System Response**
	2a	Supervisor do not choose any option.	The system doesn't perform any action

3.4.4 Make Group Use Case

Table 3.5 describes the Make Group use case. The use case explains that how to allow student to make group by adding members and group details.

Table 3.4: Make Group Use case

Use Case – 5	Make Group.
Brief Description	The use case aims to allow student to make group by adding members and group details.
Goal	Student wants to make group.
Pre-conditions	Student must be logged in.
Post conditions	The student has made a group and a new group is added to groups list.
Failed End Condition	Student doesn't select any option to make group.
Primary, Secondary Actors	Student, Database

Dependency	Includes login use case.		
Basic Flow	**Steps**	**Actions**	**System Response**
	1	Student login in the system.	System directs student to dashboard.
	2	Student use make group option.	System asks user to add member, add group details and select supervisor.
	3	The student enters the required information and completes the steps.	The system successfully adds a new group and saves details of group in FYP group.
Alternating Flow	**Steps**	**Actions**	**System Response**
	2a	User does not select make group option.	System doesn't perform any action in response.

3.4.5 Submit Document Use Case

Table 3.6 explain Submit document use case.This use case allows the student to submit a document by uploading it.

Table 3. 5: Submit Document Use Case

Use Case – 6	Submit Document.
Brief Description	This use case allows the student to submit a document by uploading it.
Goal	The student wants to upload a document.
Pre-conditions	The student must be logged in.
Post conditions	The document is successfully uploaded.

Failed End Condition	Student couldn’t upload the document.
Primary, Secondary Actors	Student.
Dependency	Includes login use case.

Basic Events Flow	**Steps**	**Actions**	**System Response**
	1	The user selects upload document option.	The system asks the user to browse document.
	2	The student browse and uploads the document.	The system shows the document in the uploaded documents.
Alternative Flow	**Steps**	**Actions**	**System Response**
	2a	The student doesn’t browse the document successfully.	System doesn’t upload anything.

3.4.6 Review Document Use Case

Table 3.7 describes how the document is reviewed.

Table 3. 6: Review Document Use case

Use Case – 7	Review Document Use case.
Brief Description	This use case explains how the document is reviewed by the supervisor.
Goal	The goal is to successfully review document by supervisor.
Pre-conditions	Supervisor must be logged in.

Post conditions	The document required is successfully displayed to supervisor.		
Failed End Condition	The supervisor is unable to review document.		
Primary, Secondary Actors	Supervisor.		
Dependency	Includes login and submit document use cases.		
Basic Flow	**Steps**	**Actions**	**System Response**
	1	Supervisor login to the system.	System takes the supervisor to the dashboard.
	2	The supervisor selects the document to review.	The system displays the document.
Alternating Flow	**Steps**	**Actions**	**System Response**
	2a	The supervisor does not select any document to review.	The system does not display anything.

3.4.7 CHAT Use Case

Table 3.8 describes that how chat takes place In system.

Table 3. 7: Chat use case

Use Case – 8	Edit and set prominent part as image.
Brief Description	This use case deals with allowing the user to chat.
Goal	The goal is to perform editing if needed
Pre-conditions	User must be logged in.
Post conditions	Chatting has successfully finished and messages were delivered successfully.

Failed End Condition	The user can't chat with another user		
Primary, Secondary Actors	Student, Supervisor		
Dependency	Includes login use case.		
Basic Flow	**Steps**	**Actions**	**System Response**
	1	The user clicks chat option.	The system asks user to select user to chat.
	2	The user selects the specific user to chat with.	The system asks the user to enter the message.
	3	The user types the text and presses send.	The system deliver the texts to the other side.
Alternating Flow	**Steps**	**Actions**	**System Response**
	1a	The user doesn't select any user.	The system doesn't display anything in response.

3.4.8 Manage Meeting Log Use case

Table 3.9 describes that how meeting log is managed.

Table 3. 8: Manage Meeting Log Use case

Use Case – 9	Manage Meeting Log.
Brief Description	This use case is executed to allow students to manage their meeting Logs with supervisors. Supervisor can review the work progress through meeting logs.

Goal	The goal is to manage meeting logs.		
Pre-conditions	The user must be logged in.		
Post conditions	The meeting log is created.		
Failed End Condition	Not executed if editing is not needed.		
Primary, Secondary Actors	Student Supervisor		
Dependency	Includes login Use case.		
Basic Flow	**Steps**	**Actions**	**System Response**
	1	User login to the system.	System directs user to the dashboard.
	2	The user selects meeting log form option.	System asks the user to fill the form for meeting Log containing meeting details.
	3	Supervisor selects review the work progress.	The system directs supervisor to meeting Logs.
Alternating Flow	**Steps**	**Actions**	**System Response**
		None	None.

3.4.9 Guidelines Use case

Table 3.10 describes that guidelines are managed and displayed.

Table 3. 9: Guidelines Use case

Use Case – 10	Guidelines Use Case		
Brief Description	This use case deals with the guidelines module which includes adding , updating or editing the guidelines.		
Goal	Adding, updating or editing the guidelines.		
Pre-conditions	The system must be in ready state.		
Post conditions	The guidelines have been successfully modified.		
Failed End Condition	The guidelines have not modified successfully.		
Primary, Secondary Actors	Admin		
Dependency	Includes login Use Case		
Basic Flow	**Steps**	**Actions**	**System Response**
	1	The user selects add guidelines option.	The system asks user to new guidelines.
	2	The user selects update guidelines option.	The system asks user to update guidelines.
	3	The user selects delete guidelines option.	The system asks user to choose guidelines to delete.

Alternating Flow	Steps	Actions	System Response
	1a	The user doesn't select valid option.	The system doesn't perform anything in return.

3.4.10 View Group Details Use Case

Table 3.11 describes Generate report use case. This use case describes how sketch record are obtained, entered in system and saved in database as criminal record report

Table 3. 10: View Group Details Use case

Use Case – 11	View Group details Use Case.
Brief Description	This use case deals with allowing to view Groups available.
Goal	To view list of available FYP groups.
Pre-conditions	The User must be logged in.
Post conditions	The list has been displayed and viewed .
Failed End Condition	List modification wasn't successful.
Primary, Secondary Actors	Student, Supervisor

Dependency	Includes login Use Case.		
Basic Flow	**Steps**	**Actions**	**System Response**
	1	The user selects view Groups option.	The system displays the list of FYP groups.
Alternating Flow	**Steps**	**Actions**	**System Response**
	3a	None	None

3.4.11 Check meeting Log Use Case

Table 3.12 describes Generate report use case. This use case describes how sketch record are obtained, entered in system and saved in database as criminal record report

Table 3. 11: Manage List Use case

Use Case – 12	Manage List Use Case.
Brief Description	This use case deals with management of list of supervisors and students by admin. The use case is executed to allow admin to add, update or delete those lists.
Goal	To successfully make changes to the lists including addition, updating or deletion of list records.
Pre-conditions	The Admin must be logged in.
Post conditions	The student or supervisor list was modified successfully.
Failed End Condition	List modification wasn't successful.

Primary, **Secondary Actors**	Admin Database		
Dependency	Includes login Use Case.		
Basic Flow	**Steps**	**Actions**	**System Response**
	1	The user selects, add to student/supervisor list option.	The system asks the user for necessary details.
	2	The user adds the new details to the list.	The system stores the new details into the database.
	3	The user selects, update student/supervisor list option.	System allows user to enter details in the already existing list.
	4	The user adds the modification to the list details.	The system stores the modified list details into the database.
	5	The user selects, delete from student/supervisor list option.	System allows user to delete any already existing record in list.
	6	The user deletes the necessary records from the list.	The system stores the modified list details into the database.
Alternating Flow	**Steps**	**Actions**	**System Response**
	3a	User click on Cancel option.	Information is removed.

3.4.12 Project Evaluation Use Case

Table 3.13 describes Project Evaluation use case. This use case describes how project is evaluated by admin.

Table 3. 12: project evaluation Use case

Use Case – 12	Project evaluation.		
Brief Description	This use case deals with project evaluation of FYP groups which include grading documents.		
Goal	To successfully grade and evaluate project documents.		
Pre-conditions	The Admin must be logged in.		
Post conditions	The project was evaluated successfully.		
Failed End Condition	Project evaluation wasn't successful.		
Primary, Secondary Actors	Admin Database		
Dependency	Includes login Use Case.		
Basic Flow	**Steps**	**Actions**	**System Response**
	1	The Admin selects group.	The system displays the group and its data.
	2	The admin selects document to view and grade.	The system displays the document.

	3	The admin grades the document.	System saves the grading information.
Alternating Flow	**Steps**	**Actions**	**System Response**
	2a	User click on Cancel option.	Information is removed.

3.4.13 Manage List Use case

Table 3.14 describes Manage List use case. This use case describes how list of students and supervisors is managed.

Table 3. 13: Manage List Use case

Use Case – 12	Manage List Use Case.
Brief Description	This use case deals with management of list of supervisors and students by admin. The use case is executed to allow admin to add, update or delete those lists.
Goal	To successfully make changes to the lists including addition, updating or deletion of list records.
Pre-conditions	The Admin must be logged in.
Post conditions	The student or supervisor list was modified successfully.
Failed End Condition	List modification wasn't successful.

Primary, Secondary Actors	Admin Database

Dependency	Includes login Use Case.		
Basic Flow	**Steps**	**Actions**	**System Response**
	1	The user selects, add to student/supervisor list option.	The system asks the user for necessary details.
	2	The user adds the new details to the list.	The system stores the new details into the database.
	3	The user selects, update student/supervisor list option.	System allows user to enter details in the already existing list.
	4	The user adds the modification to the list details.	The system stores the modified list details into the database.
	5	The user selects, delete from student/supervisor list option.	System allows user to delete any already existing record in list.
	6	The user deletes the necessary records from the list.	The system stores the modified list details into the database.
Alternating Flow	**Steps**	**Actions**	**System Response**
	3a	User click on Cancel option.	Information is removed.

3.5 Non-Functional Requirements

Non-functional requirements describe the system attributes. Some of the non-functional requirements of the developed system are:

3.5.1 Security

The system is used for educational purpose so the system needs to control unauthorized access. The back of data kept by the system makes the user experience more reliable and trustworthy.

3.5.2 Flexibility

The system is supposed to have enough reliability to spot minor errors made by the user and effectively help user to resolve them.

3.5.3 Testability

The system can be tested in any way or by using any testing method for conformance to user requirements and system performance.

3.5.4 Maintainability

The system is supposed to maintain records of every stakeholder involved.

3.5.5 Portability

The system is designed to work on windows and must work on it effectively.

3.6 Resource Requirements

The following requirements are used mentioned below. For the successful accomplishment of the project many hardware and software resources are required. The programming language used for developing the system is Node Js.

3.6.1 Hardware Requirements

The following hardware specification are required to develop and test the system.

- Desktop/Laptop.
- RAM (Random Access Memory) 4GB or higher
- Dual-core 2.5 GHz processor or higher
- Hard drive minimum 300GB
- Window 8 or above

3.6.2 Software Requirements

The following mentioned software resources are required for system.

3.6.2.1 Node Js

Node js is based on JavaScript runtime environment that executes JavaScript code. Node Js is an open source platform, which makes it easily available without wasting resources. Complexity of writing server applications is reduced in Node Js through APIs. Primarily Node Js is used to build web servers. The functionalities of Node Js are unblocking. Node Js support different operating systems including Macos, Linux, Windows and UNIX server.

3.6.2.2 Mongo DB

Mongo DB is classified as a NoSQL database. Json-like documents are used with schemas for data storage in Mongo DB, which is developed by Mongo DB Inc. Mongo DB supports many features including Indexing, replication, load balancing, file storage, aggregation, capped collections and transactions. This database approach is more suitable for rapidly changing data. More suitable for ad-hoc approach.

3.6.2.3 Human Effort

Table 3. 14: Task gantt chart

Tasks	**Sep 2020**	**Oct 2020**	**Nov 2020**	**Dec 2020**	**Jan 2021**	**Feb 2021**	**Mar 2021**	**Apr 2021**	**May 2021**	**Jun 2021**
Problem Identification										
Requirement Gathering and Analysis										
Proposal report And presentation										
Prototyping and user interface										
Web Application Development										
Initial integrated testing										

Progress Report & Presentation									
Final Testing And demo									

Table 3. 15: Work Division

	Salman Khan	**Ehtisham Ali**	**Nafeesa**
Requirement Analysis and Information Research	Yes	Yes	Yes
Node Js Research	Yes	Yes	Yes
Proposal Report	Yes	Yes	Yes
Interface Design	Yes	Yes	Yes
System modeling	Yes	Yes	Yes
Database Server Setting Up	Yes	Yes	Yes
Guidelines	Yes	Yes	Yes
Repository	Yes	Yes	Yes
Authentication	Yes	Yes	Yes
Dashboard	Yes	Yes	Yes
Make Group Module	Yes	Yes	Yes
Progress Report	Yes	Yes	Yes
Presentation and Demonstration	Yes	Yes	Yes

3.9 Database Requirements

The database stores all the data retrieved from the clients. Mongo DB database is used for this project.

3.9.1 Mongo DB

MySQL is an open source relational database management system (RDBMS) which is backed by oracle and Structured Query Language (SQL) is used to perform operations in MySQL. For this system MySQL is used on window and MySQL runs virtually on all platforms. Although it is used in extensive variety of applications, MySQL is mostly linked with web applications and online issuing

3.10 Project Feasibility

Feasibility study of the project is accomplished to examine whether the project is feasible within budget and time allocated.

3.10.1 Technical Feasibility

The system is based on desktop so it is more feasible to run it on windows operating system as the features are designed for desktop application.

3.10.2 Operational Feasibility

The system is operationally quite feasible as it is developed using framework and algorithms that provide all of the features for successful completion of project.

3.10.3 Legal & Ethical Feasibility

It doesn't break any rule and regulation of state. The project is legally and ethically feasible because it does not harm anyone in any context, break any law nor harm religious beliefs of any ethnic group.

3.11 Summary

The use case model describes the communication among actors and use cases and also discuss the functional and non- functional requirements. In chapter 3 we have explained the complete and comprehensive flow of this system's working. Requirement specifications are the foundation for a successful and well managed project. We have designed use case diagram and explained use cases of this system in chapter 3 comprehensively, which gives a clear idea about the working of the system and the stakeholders involved into it. Functionalities of the system has been identified clearly including nonfunctional and functional requirements. Legitimacy of the system is also indicated through feasibility are discussed in detail.

CHAPTER 4

SYSTEM MODELING

4.0 Introduction

In this chapter we will look at 4+1 view design architecture model. System modelling contains numerous models or diagrams, with each model explaining different aspect of system. All the models in this chapter are based on Unified Modelling Language(UML).

The non-technical stakeholders including the users can more easily understand the system with the help of system modelling. Quality system modelling is very helpful in future changes in the system as well as successful and smooth completion of project.

4.1 Hide Fidelity Prototype

These prototypes are higher model and shows the working and work progress of the system along with effectiveness of system.

4.1.1 Dashboard

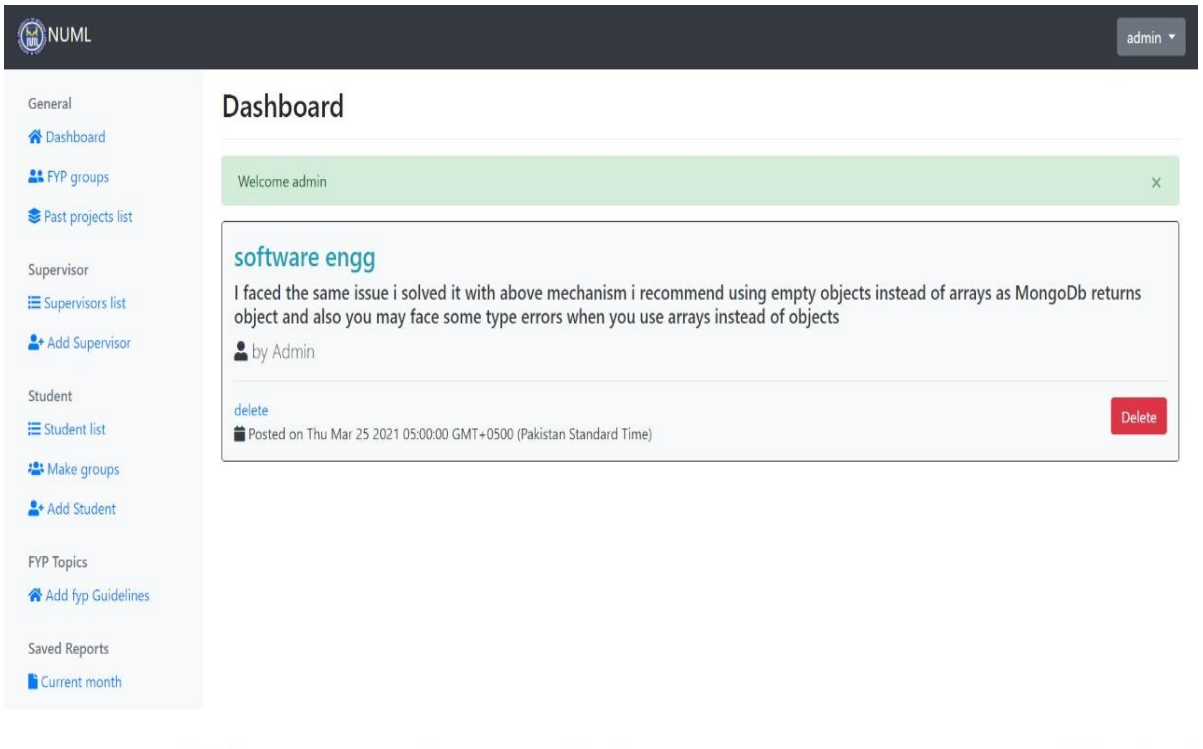


Figure 4. 1: Dashboard

4.1.2 Login

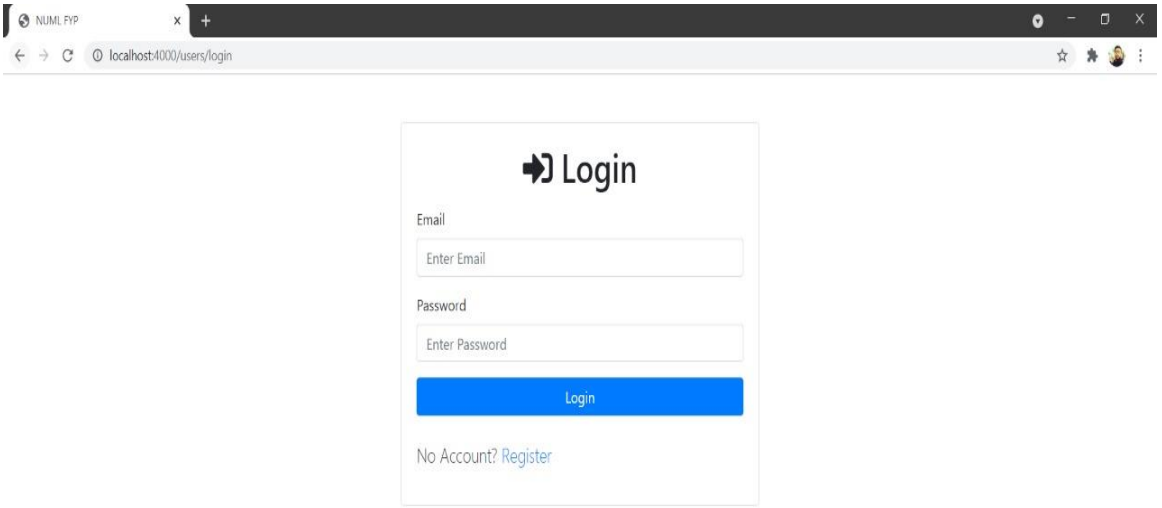


Figure 4. 2: Login.

4.1.3 Add Guidelines

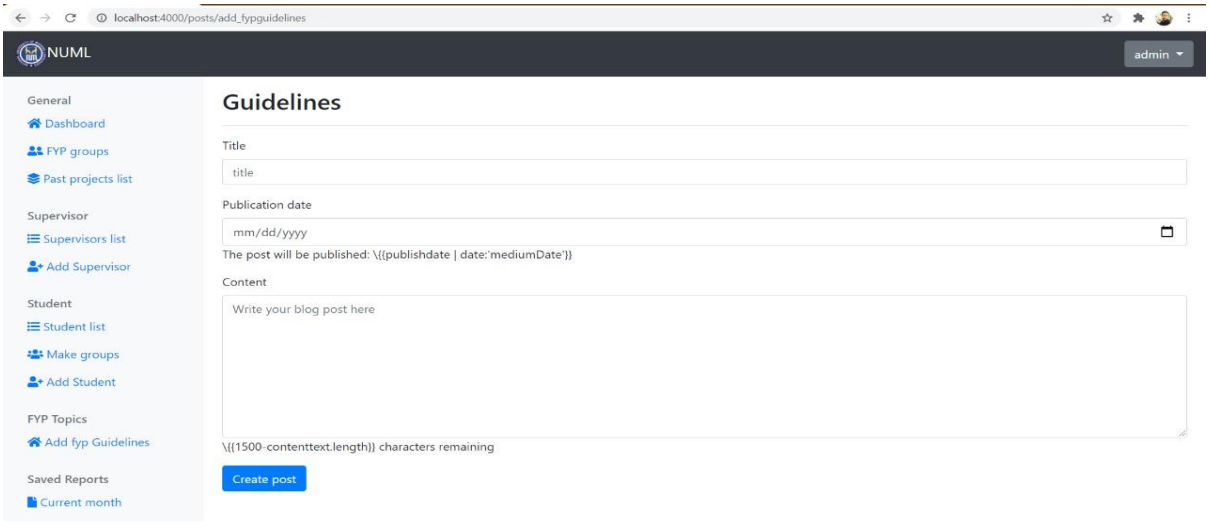


Figure 4. 3 Add guidelines.

4.1.4 Register

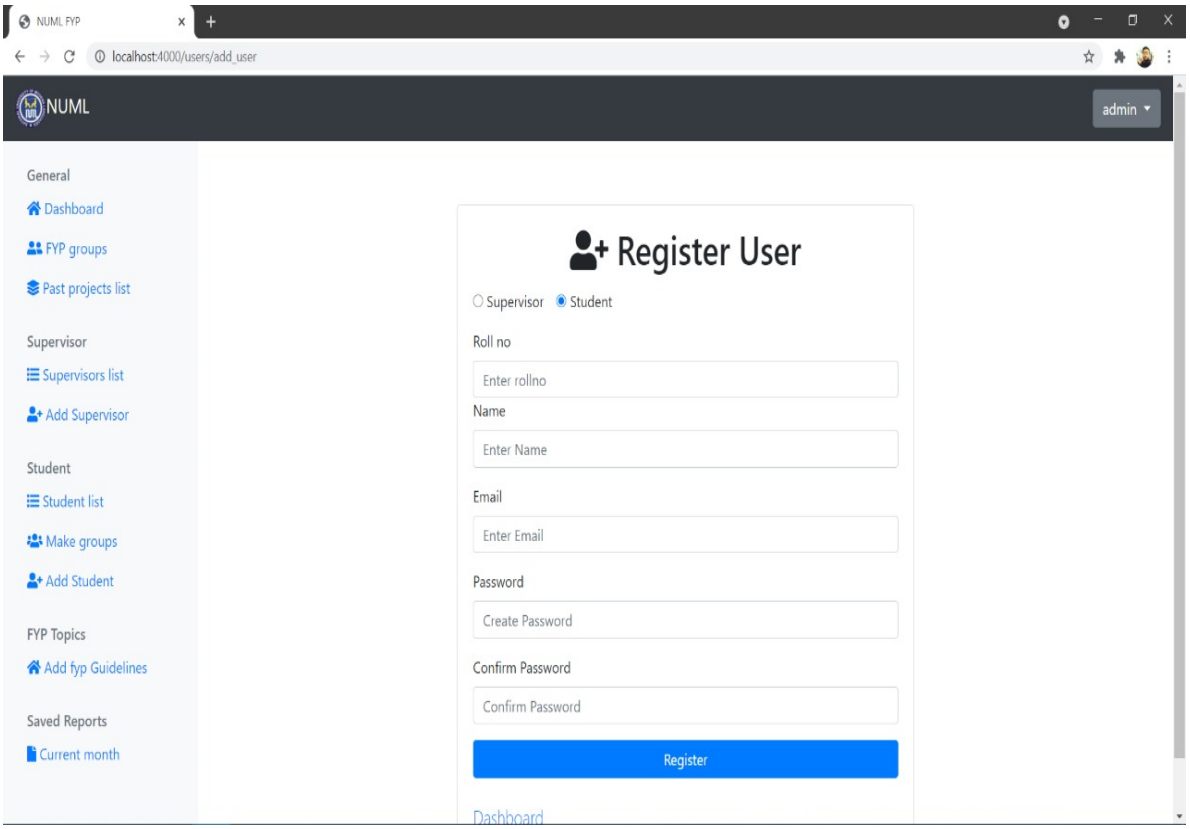


Figure 4. 4: Register.

4.1.5 Student List

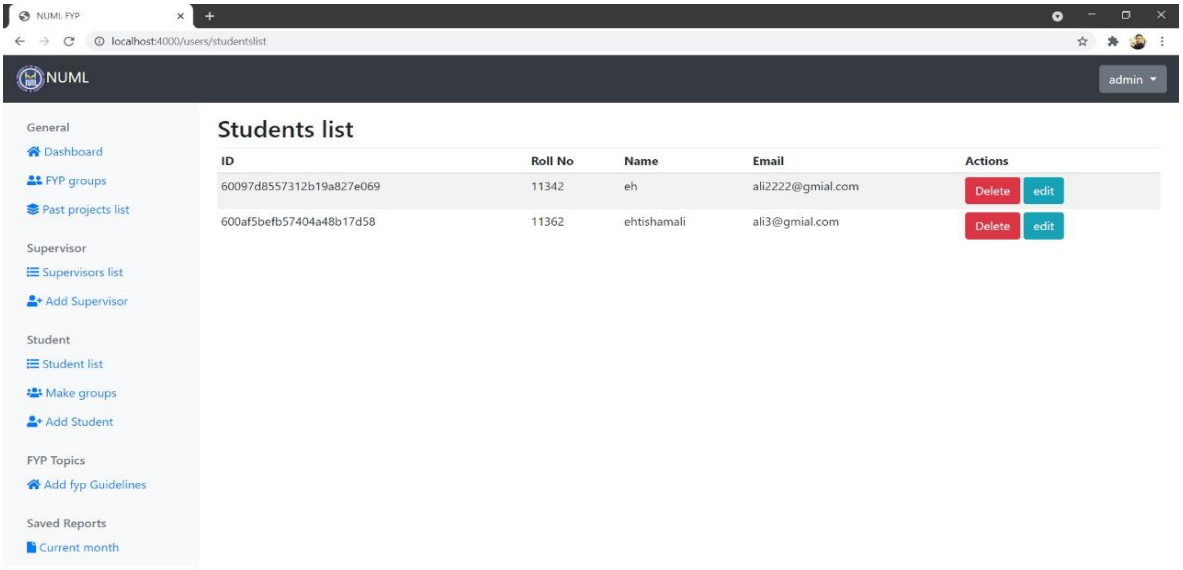


Figure 4. 5: Student List.

4.1.6 Supervisor List

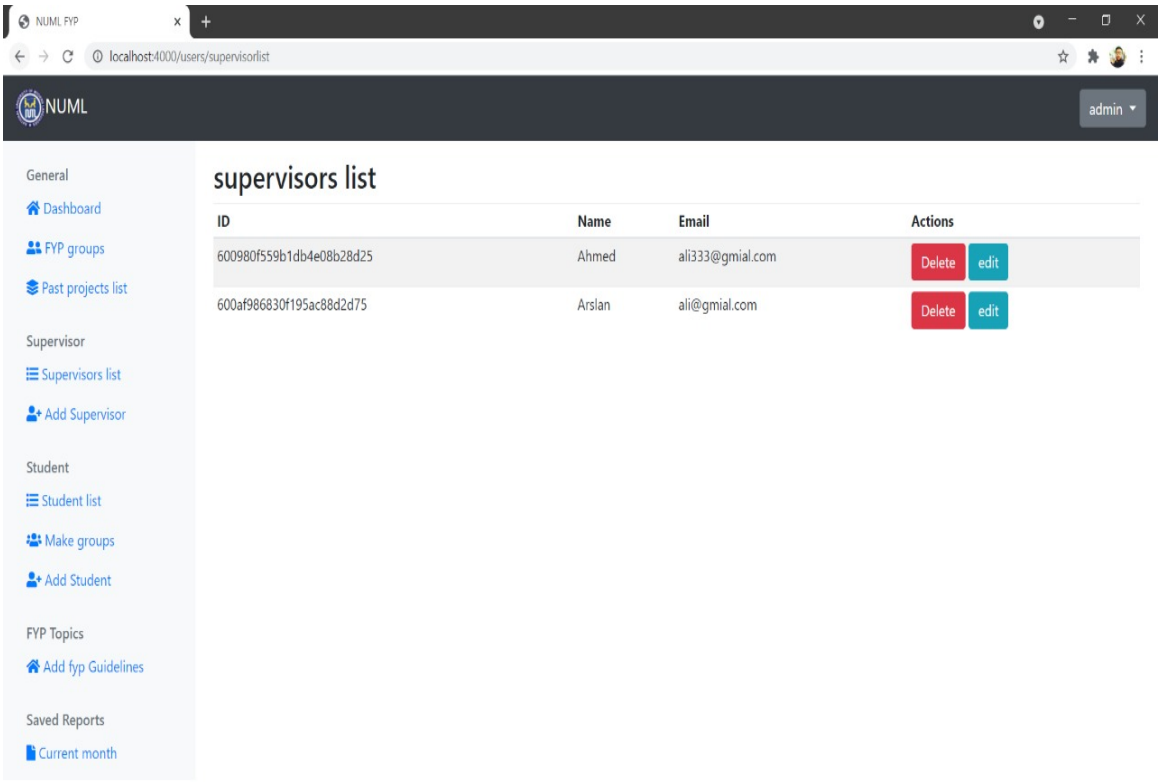


Figure 4. 6: Supervisor List

4.1.7 Make Group

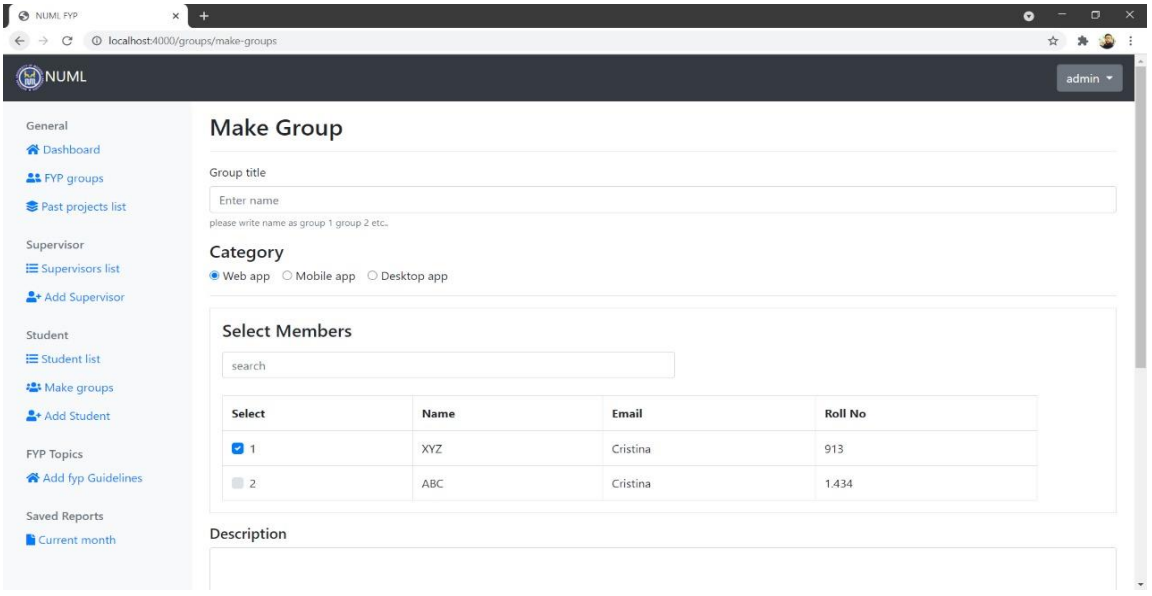


Figure 4. 7: Make Group.

4.2 Design Approach

There are two basic generalized design approaches in software engineering.

- Top down Design Approach.
- Bottom up Design Approach.

4.2.1 Top down Design Approach

In top down design approach the system division is performed on the basis of requirements, after listing the requirements the system is divided in standalone modules providing a structured control of the system. In bottom up design approach we only know the basic components of the system, by integrating these components we develop a better system.

4.2.2 Bottom up Design Approach

Bottom up design approach requires the very basic knowledge of system components and modules and leads towards a product. In this approach, we have all the basic components of the system known, and we develop and integrate them which leads towards a product. The top down design approach is adopted to develop the system. In Bottom Up

4.3 Logical View

The logical architecture mainly supports the functional requirements of a system, which includes services provided by system to its users. Logical view contains: Class diagram.

- Entity Relation Diagram (ERD).

4.3.1 Class Diagram

The class diagram which consists of attributes, classes and methods are the most commonly use diagrams in system development.

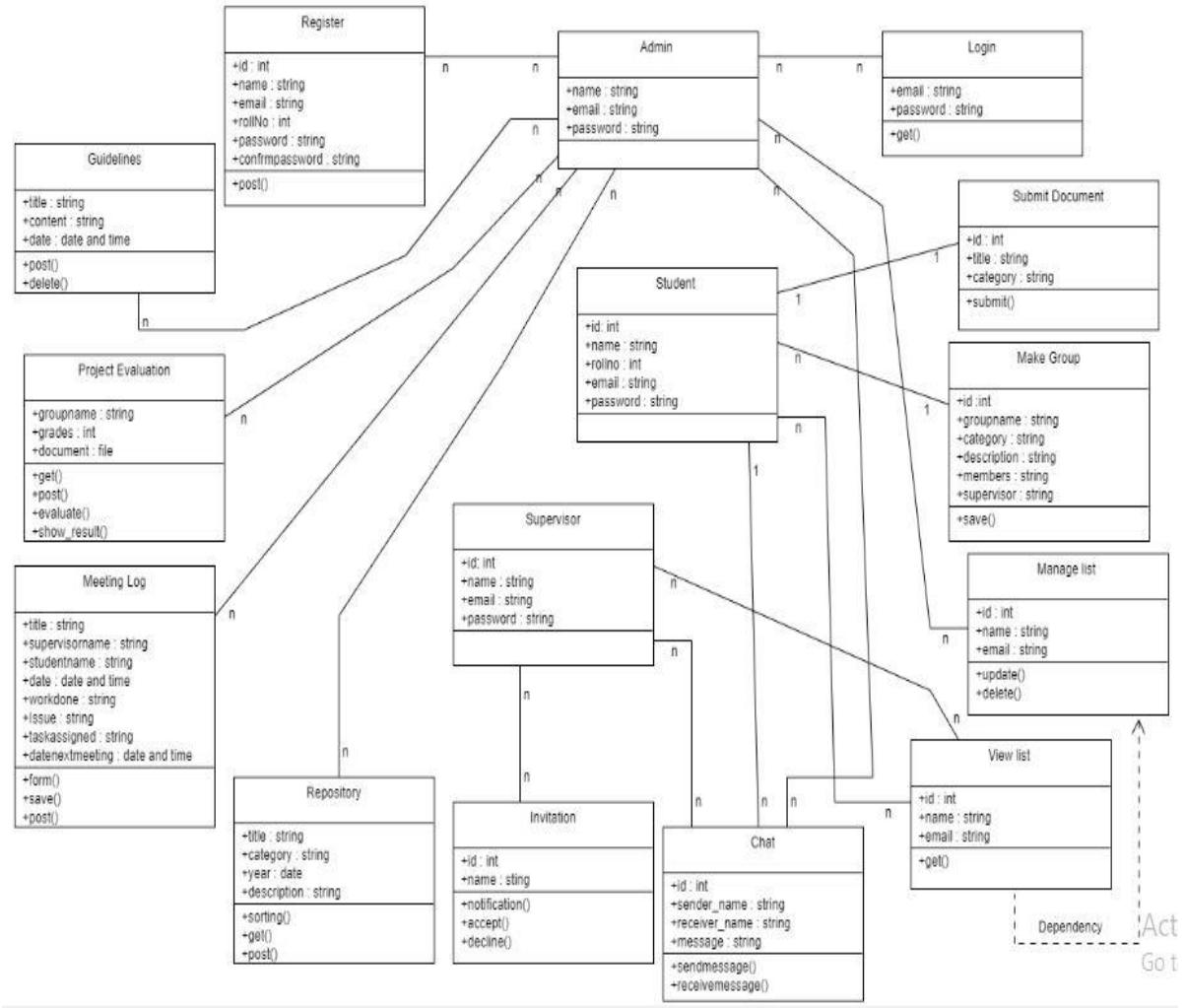


Figure 4. 8: Class diagram.

4.3.2 Entity Relation Diagram (ERD)

Entity relationship diagram shows how different entities related to one another and how they work together. It is used to describe the elements of a system and their relationships. The elements of ERD includes:

- Entity.
- Attributes.
- Relationships.

The Figure 4.10 describes the elements of the system and their relationship

4.3.2.1 Supervisor ERD

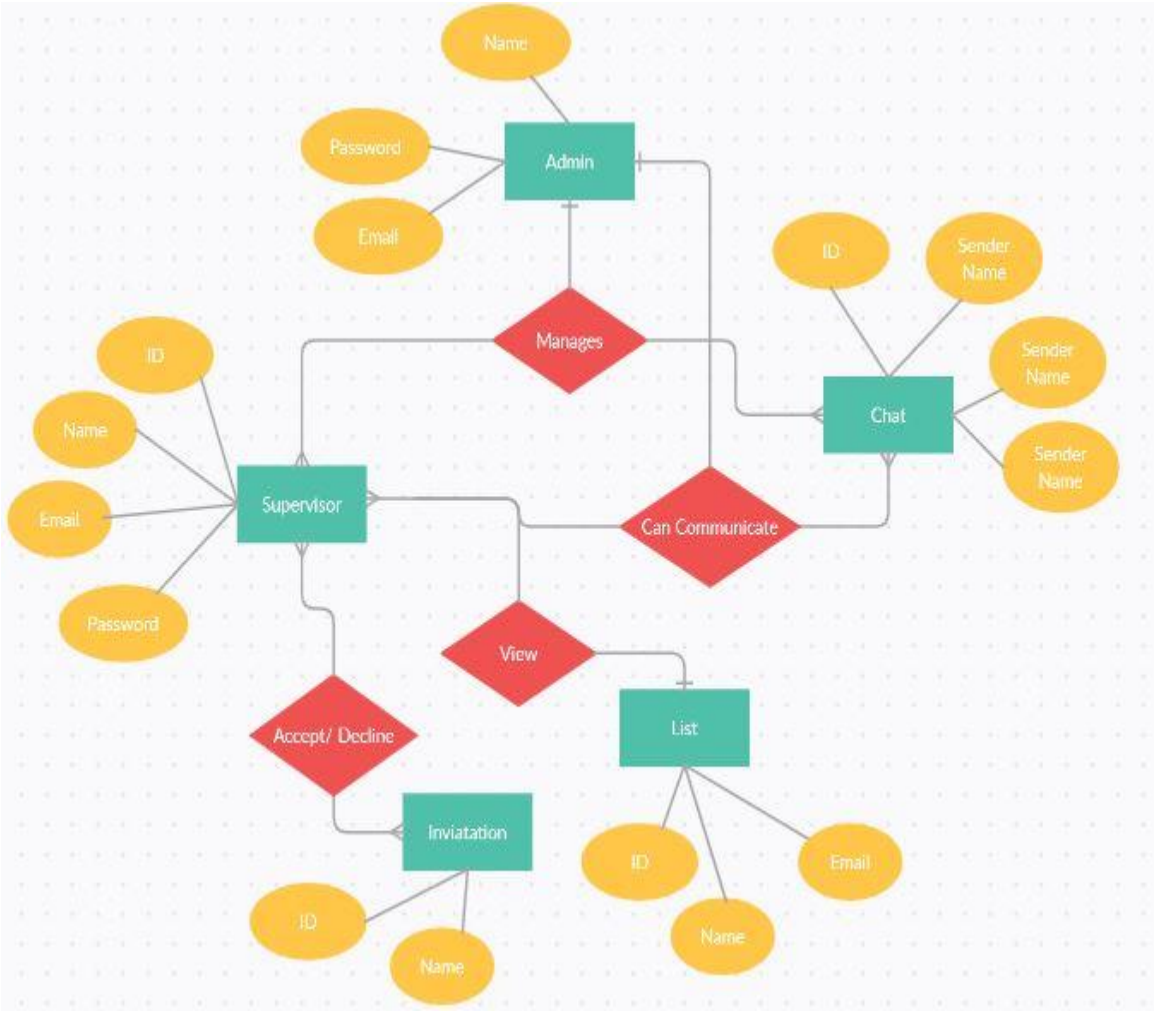


Figure 4. 9: Entity Relationship Diagram (ERD)

4.3.2.2 Admin ERD

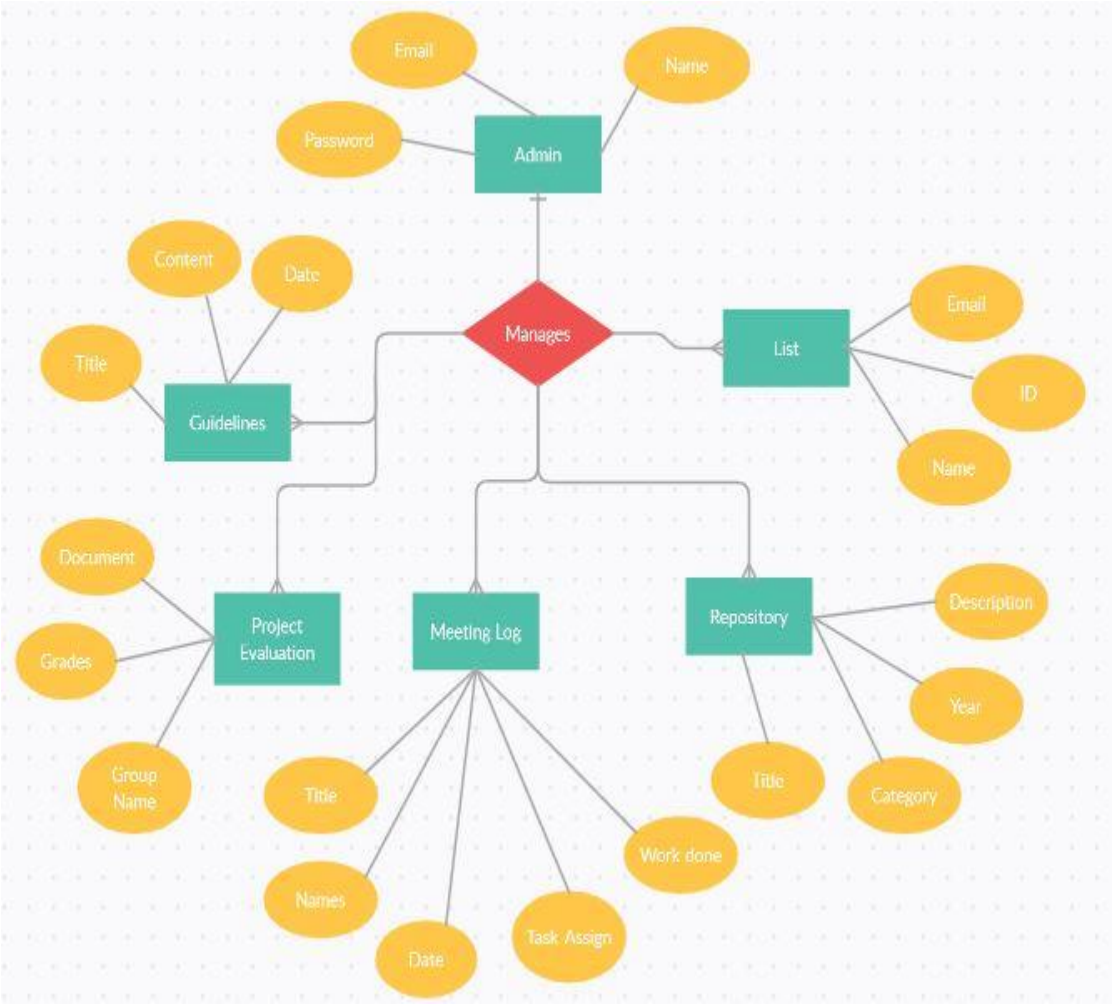


Figure 4. 10: Entity Relationship Diagram (ERD)

4.3.2.3 ERD student

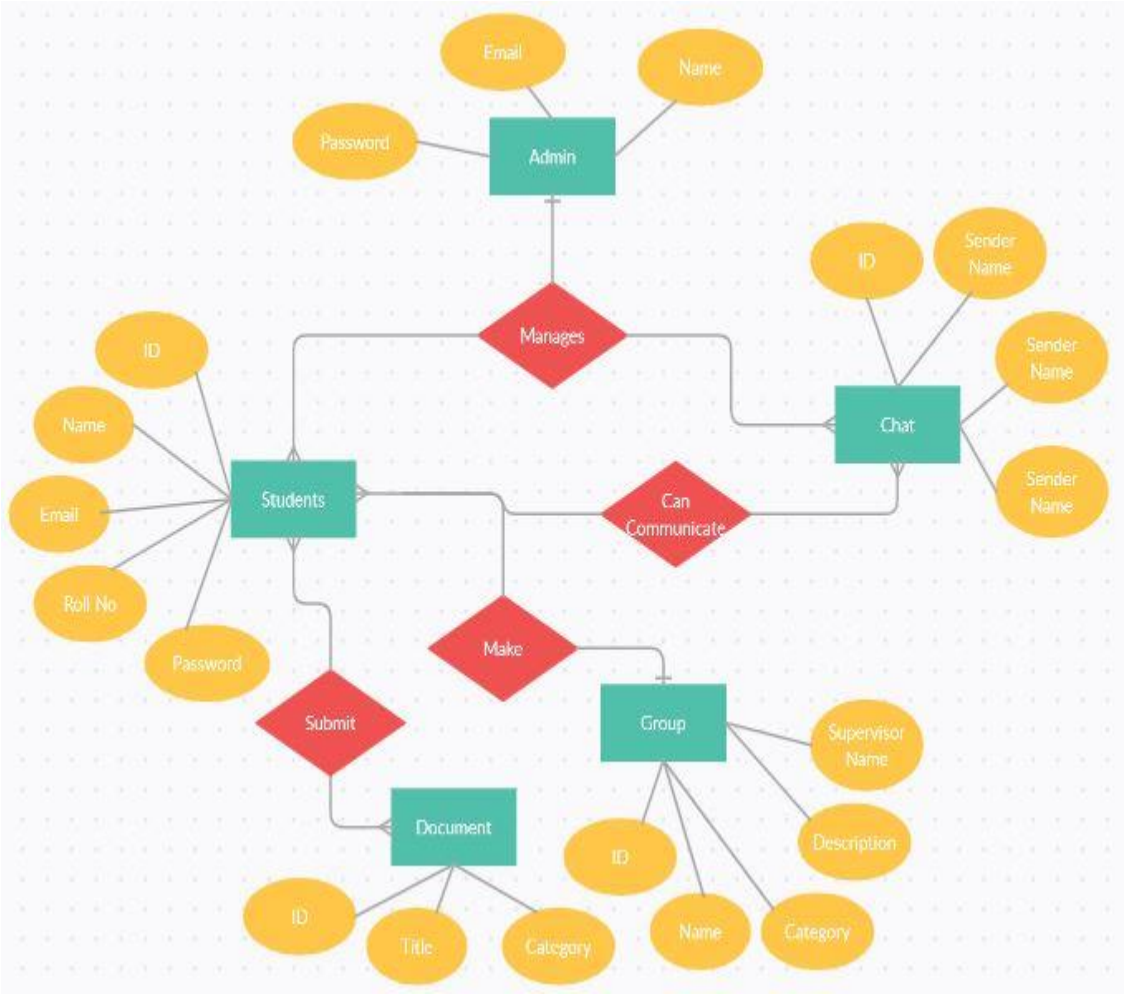


Figure 4. 11: Entity Relationship Diagram (ERD)

4.4 Process View

The process architecture takes into account some non-functional requirements, such as performance and availability. It addresses issues of concurrency and distribution, of system's integrity, of fault-tolerance, and how the main abstractions from the logical view fit within the process architecture. The process view includes:

- Activity diagram.
- State machine diagram.
- Sequence diagram.

4.4.1 Activity diagram

Activity diagram describes the dynamic aspects of the system. Activity diagram is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. It also deals with the dynamic behavior of the system by focusing on the runtime behavior of the system. Activity diagrams are mainly used as a flowchart that consists of activities performed by the system.

4.4.1.1 Manage List

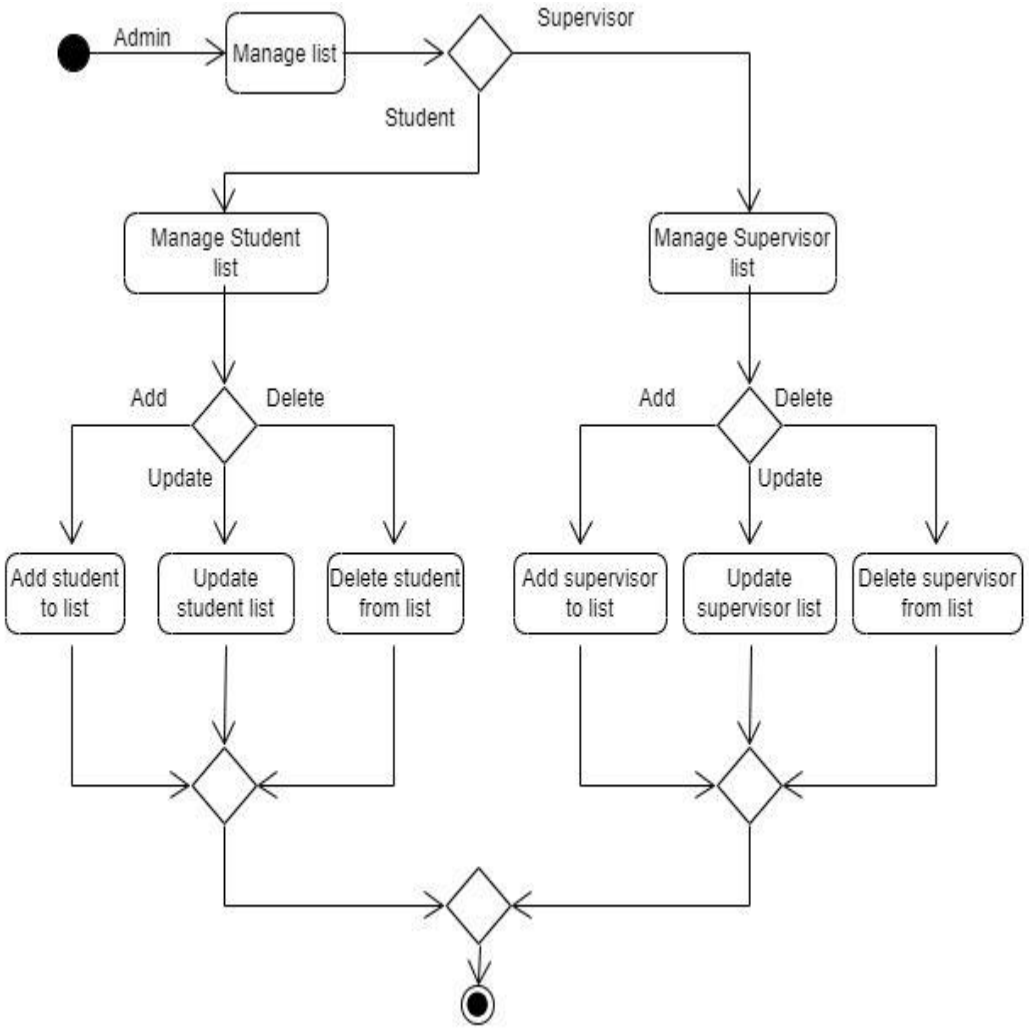


Figure 4. 12: Make List

4.4.1.2 Project Evaluation

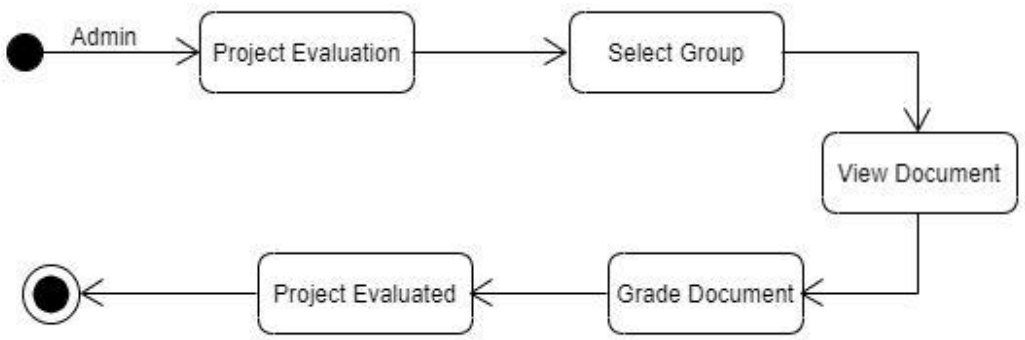


Figure 4. 13: Project Evaluation

4.4.1.3 Login

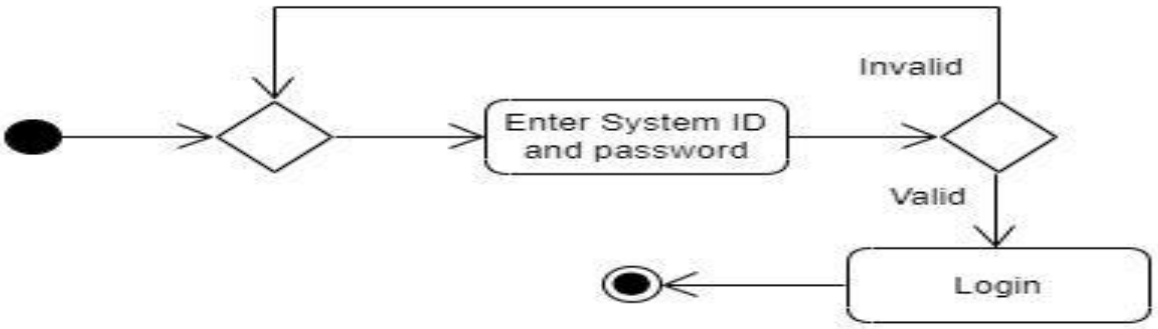


Figure 4. 14: Login Activity.

4.4.1.4 Registration activity diagram

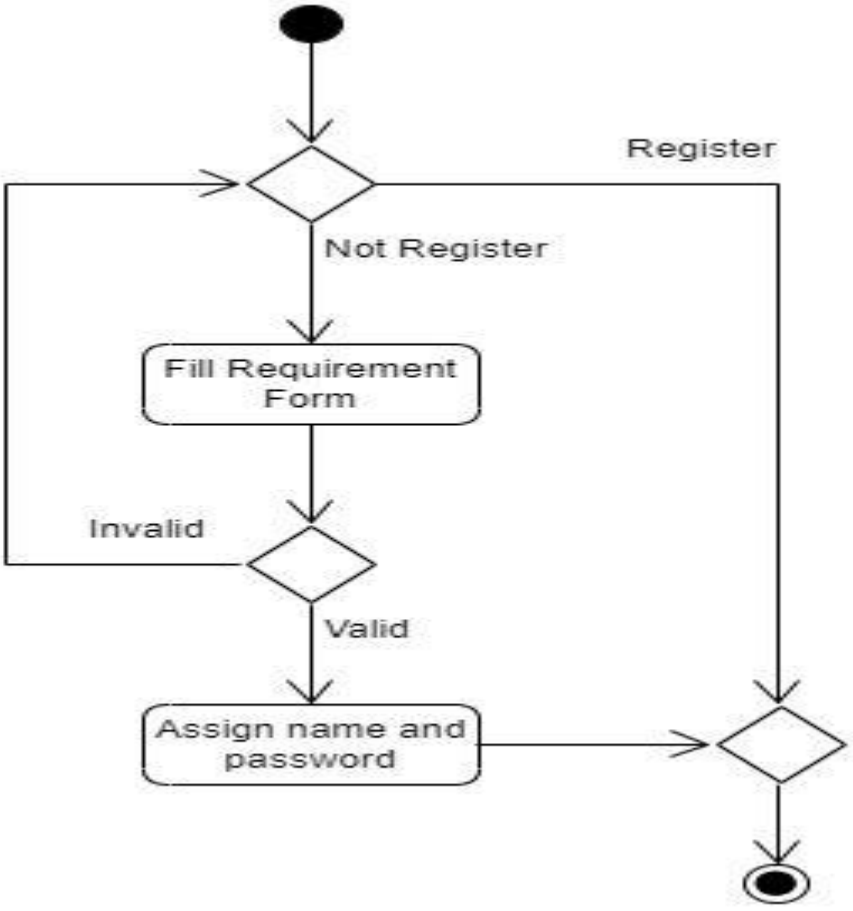


Figure 4. 15: Register activity.

4.4.1.5 Manage Group Invitation

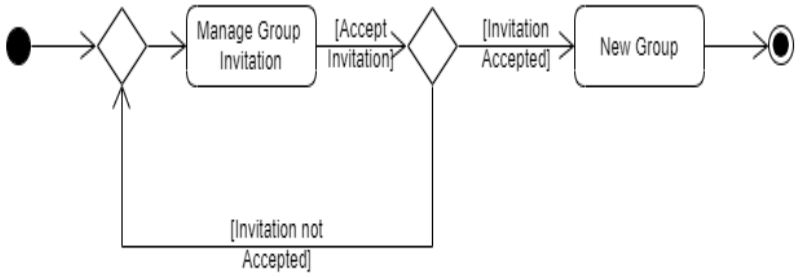


Figure 4. 16: Manage Group Invitation activit

4.4.1.6 Make Group

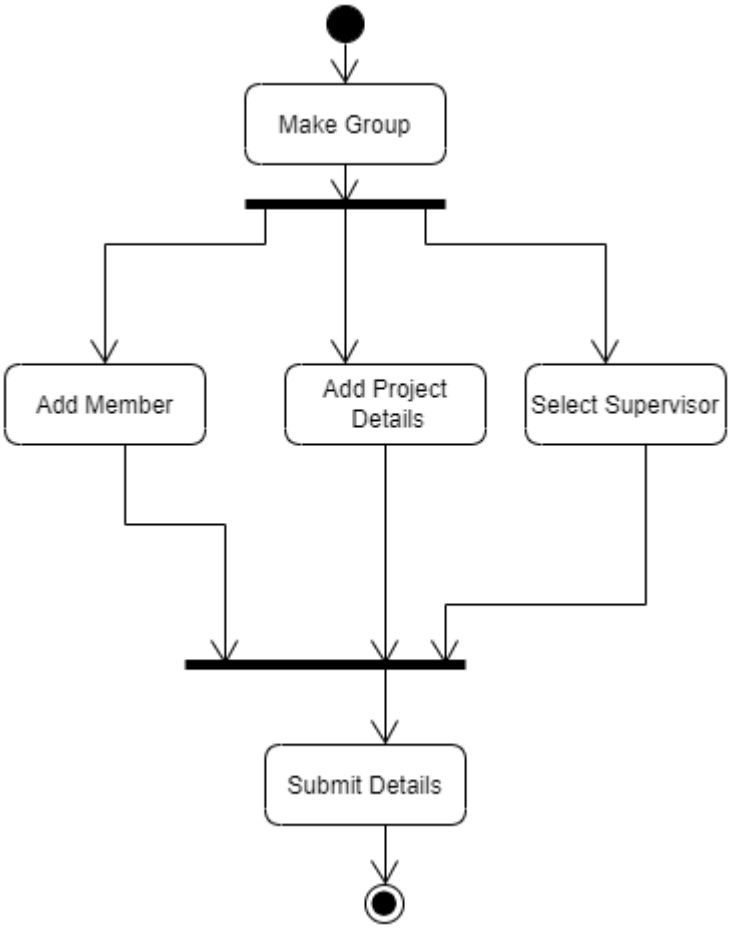


Figure 4. 17: Make Group Activi

4.4.1.7 Submit Document activity

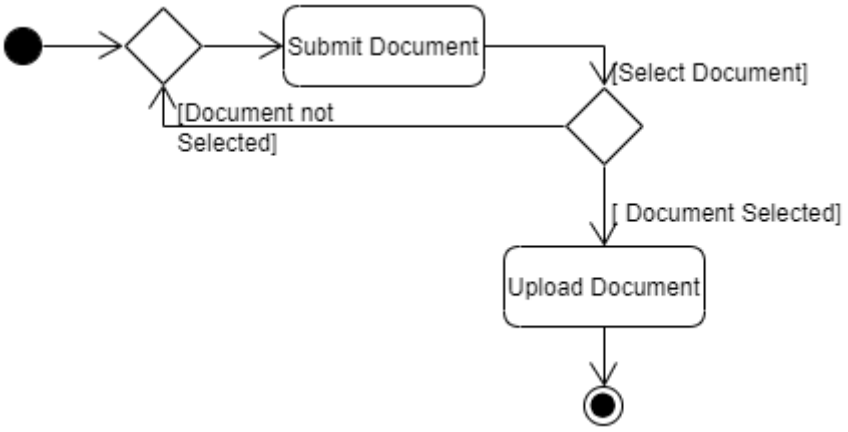
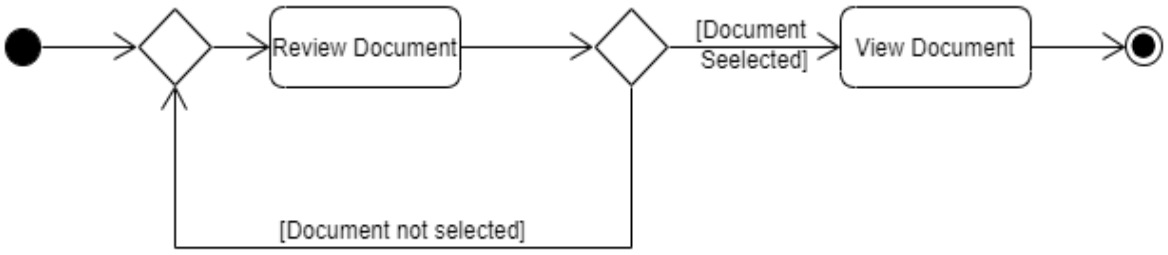


Figure 4. 18: Submit Document

4.4.1.8 Review Document

The user/admin are allowed to select the filters according to their need. The selected filter will be displayed in the side bar. The Figure 4.16 describes to select the filters and display the selected filter accordingly. Only the available filters will be displayed.



4.4.1.9 Chat

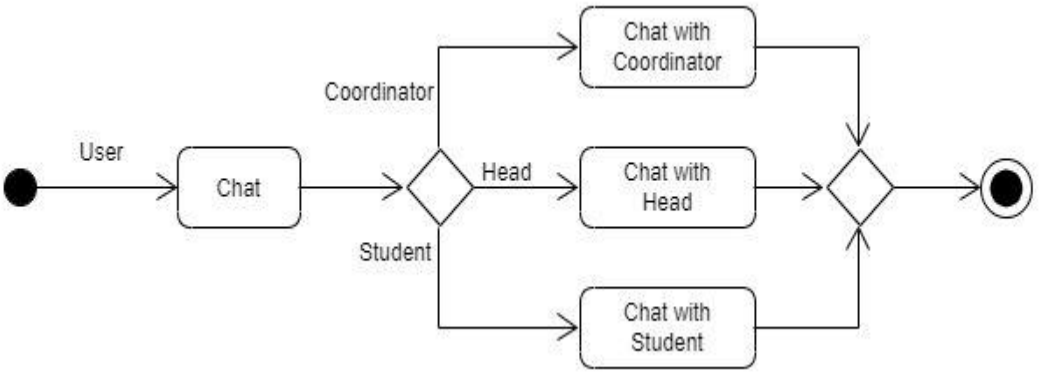


Figure 4. 20: Chat.

4.4.1.10 Manage Meeting Log

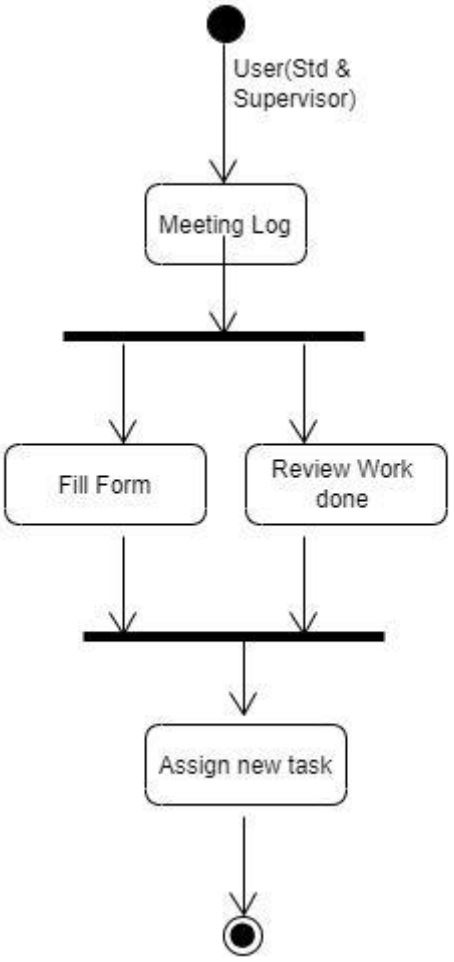


Figure 4. 21: Manage Meeting Log.

4.1.1.11Check Group Details

The system allows only the admin to make changes in database. The Figure 4.19 describes to select the dummy face to database. The admin select the face to load into database.

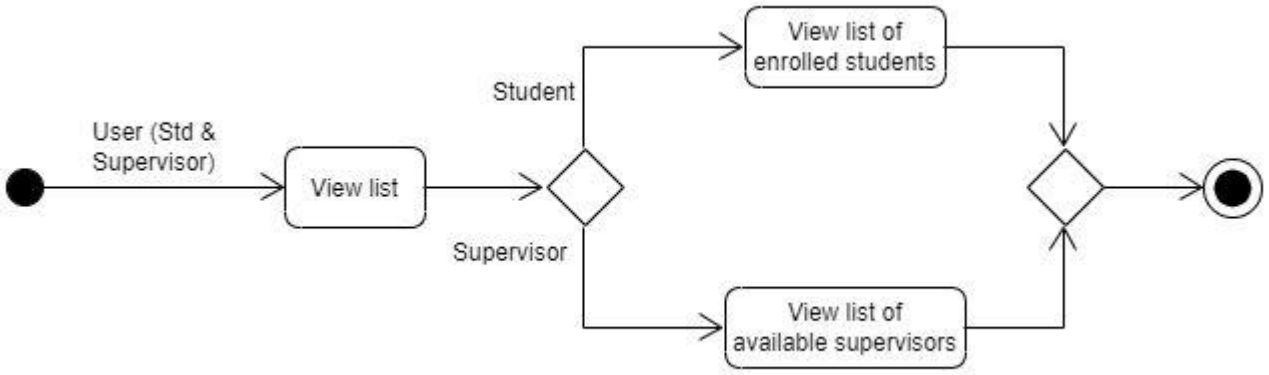


Figure 4. 22: Guidelines.

4.5 State machine diagram

It defines behavior of single object in detail. State machine diagram shows dynamic behavior of system, also called high level behavior.

4.5.1 Login state machine diagram

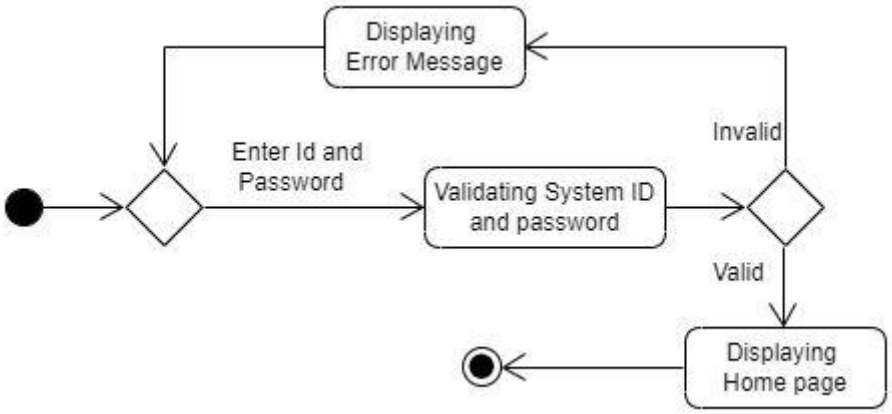


Figure 4. 23: Login

4.5.2 Register state machine diagram

Figure describes the register state for the user.

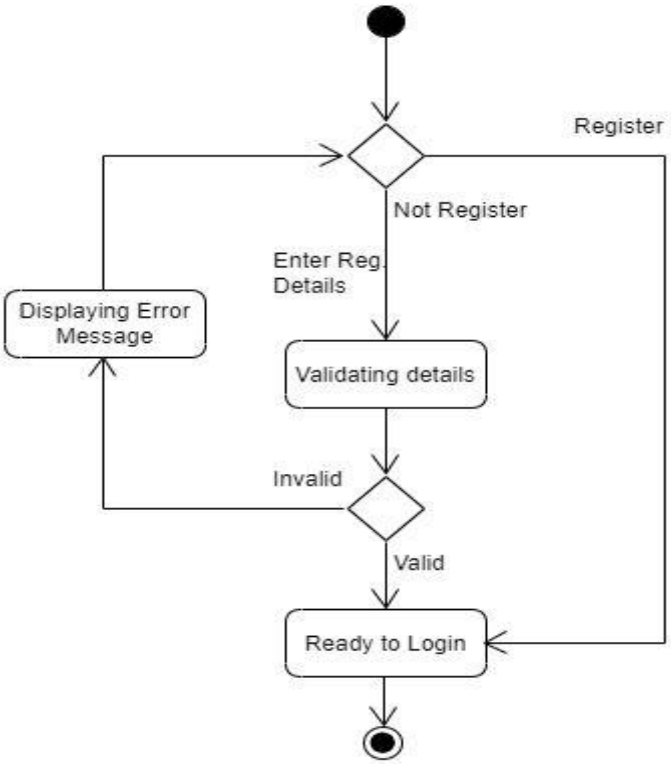


Figure 4. 24: Register.

4.5.3 Manage Group Invitation state machine diagram

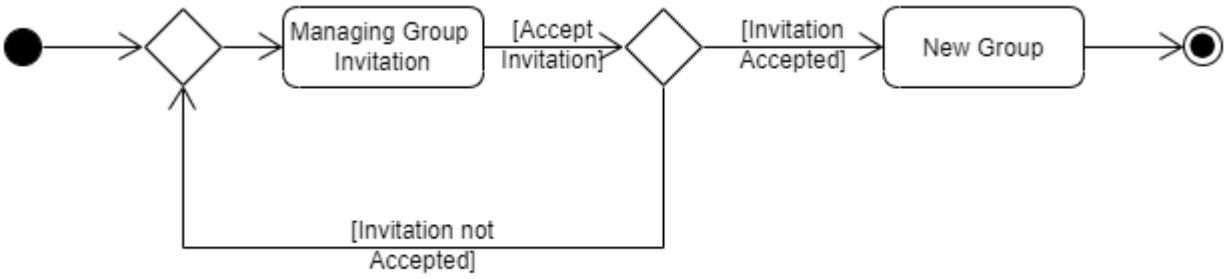


Figure 4. 25: Manage Group Invitation.

4.5.4 Submit Document state machine diagram

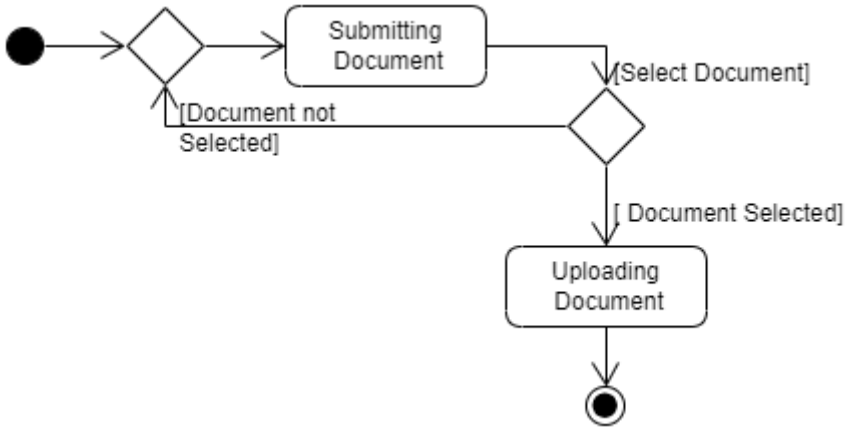


Figure 4. 26: Submit Document.

4.5.5 Review Document state machine diagram

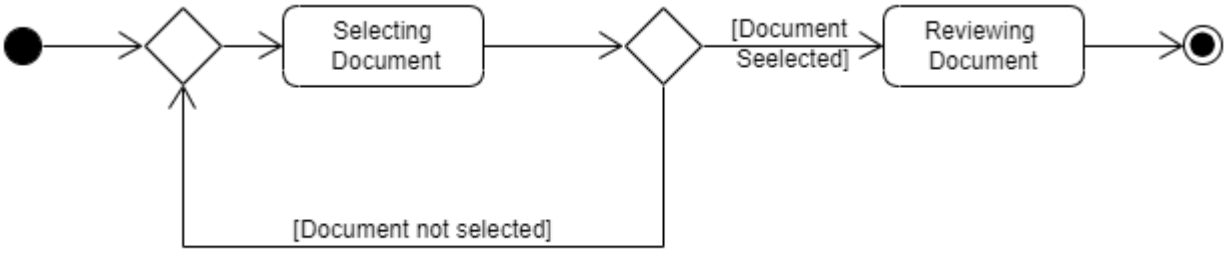


Figure 4. 27: Review Document.

4.5.6 Chat state machine diagram

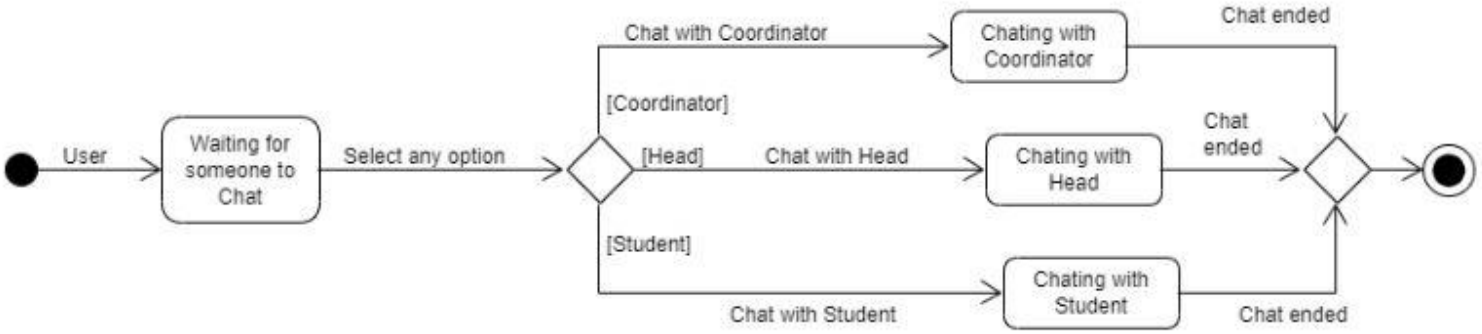


Figure 4. 28: Chat.

4.5.7 Manage Meeting Log state machine diagram

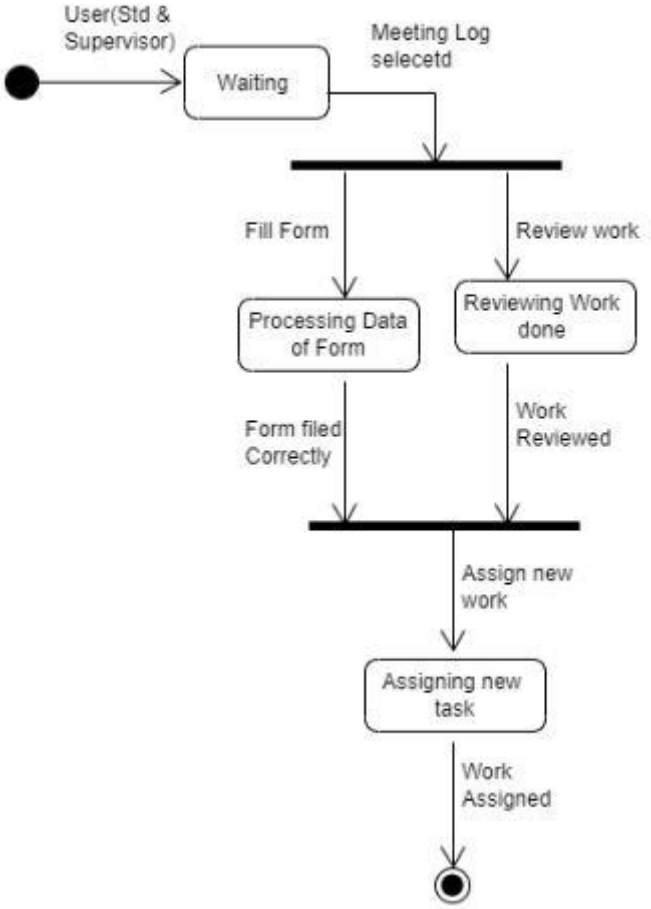


Figure 4. 29: Manage Meeting Log.

4.5.8 Guidelines State machine diagram

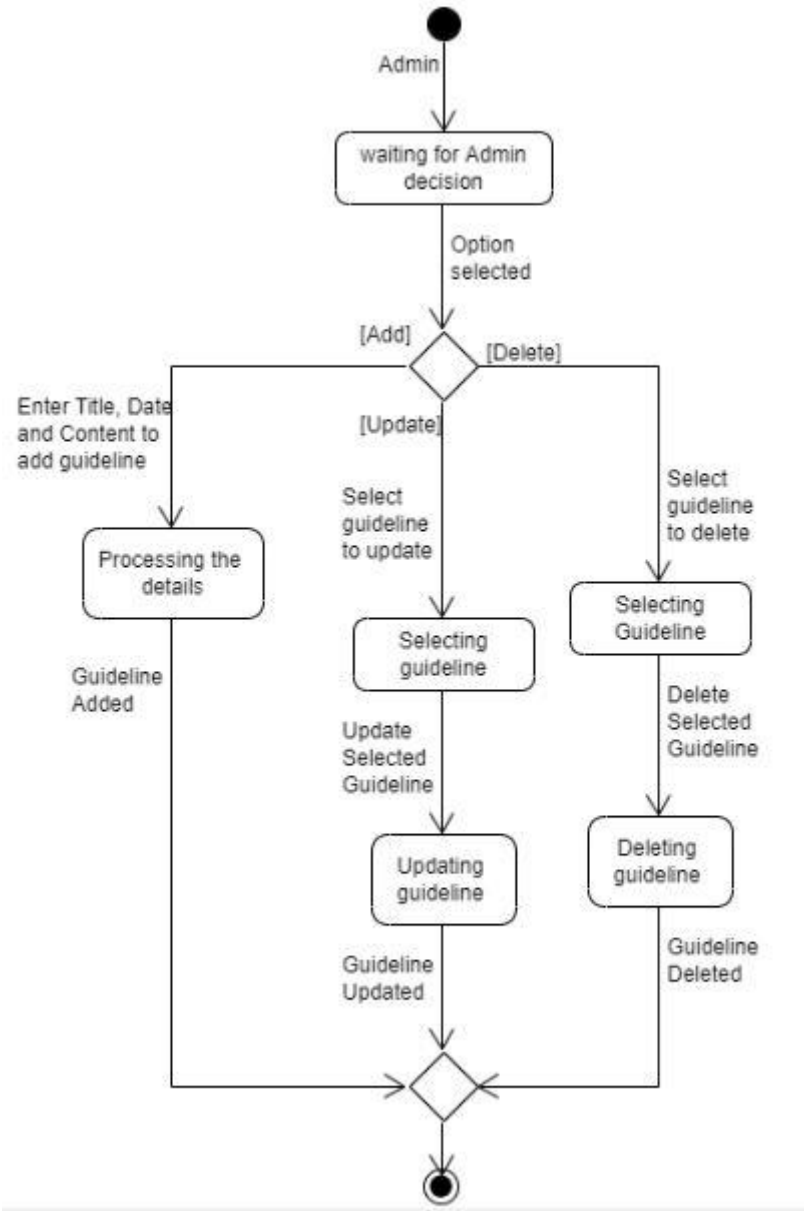


Figure 4. 30: Guidelines..

4.5.9 Manage List state machine diagram

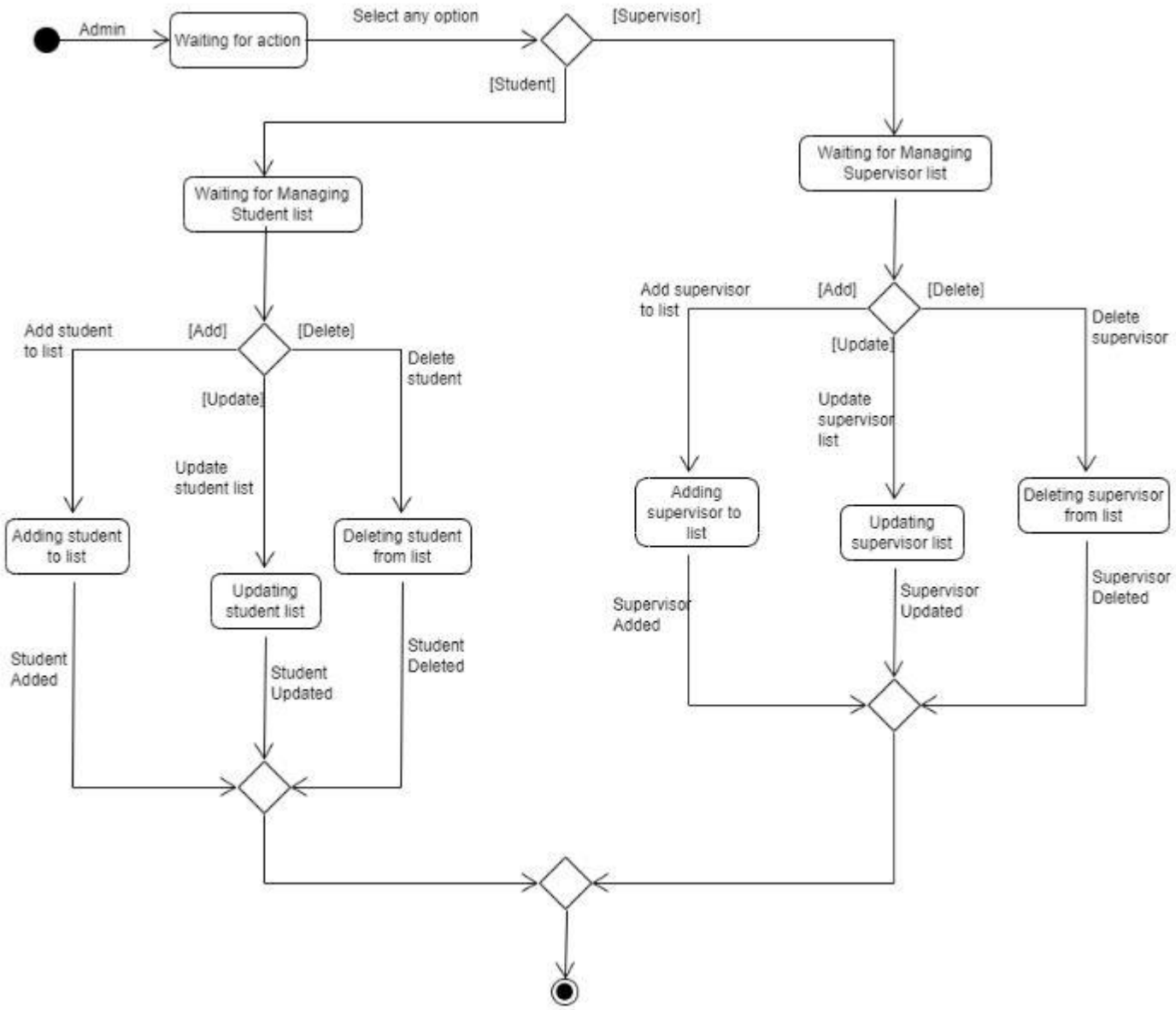


Figure 4. 31: Manage List

4.5.10 Manage Repository

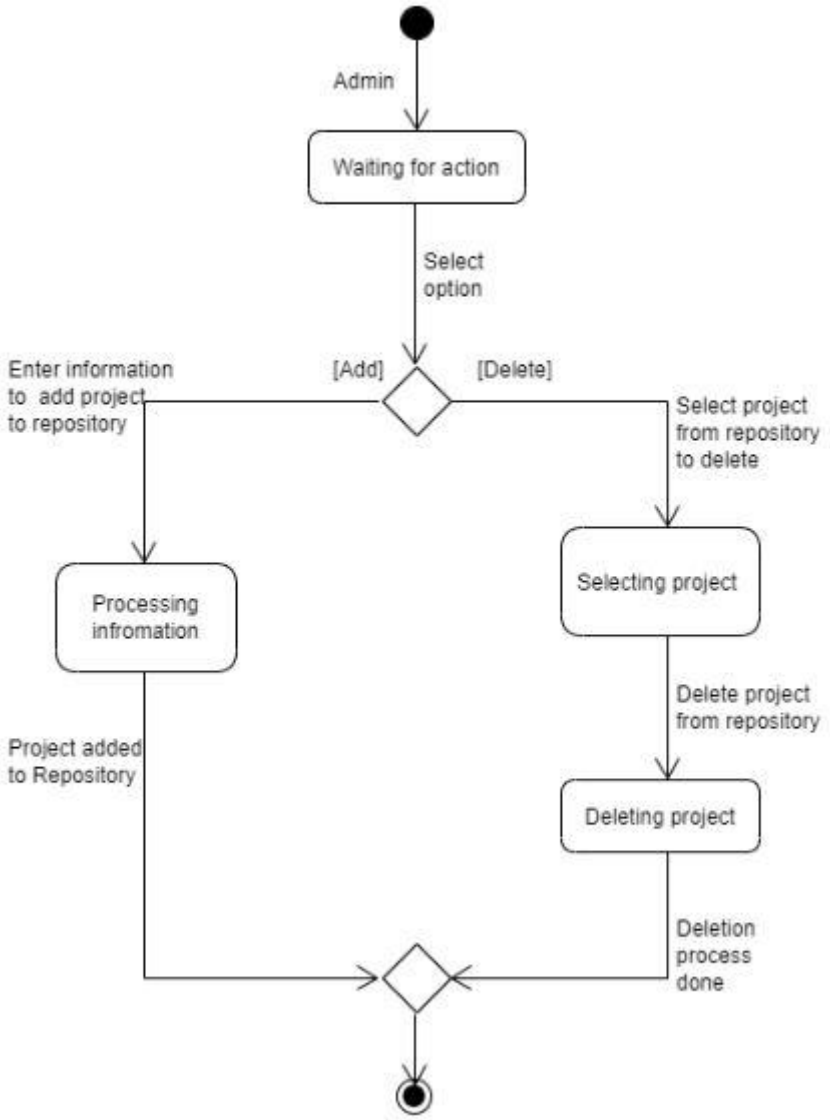


Figure 4. 31: Manage Repository.

4.5.11 View List

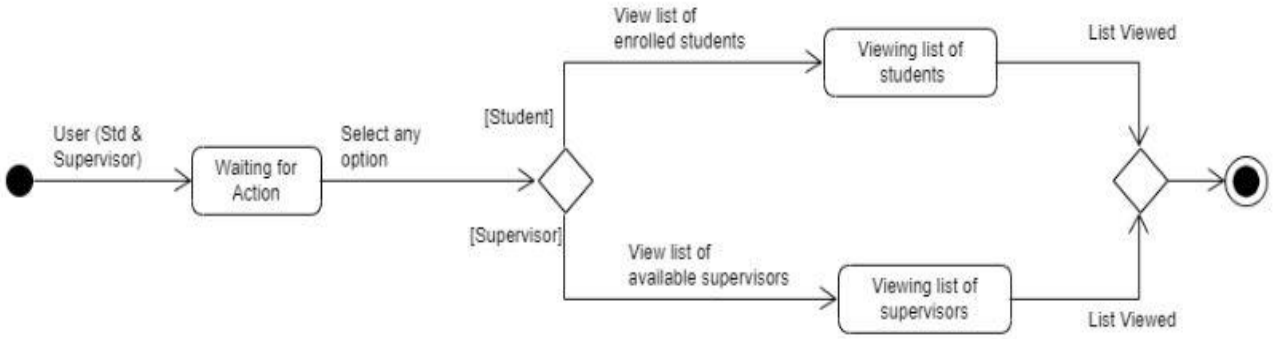


Figure 4. 32: view List

4.5.12 Project Evaluation state machine diagram

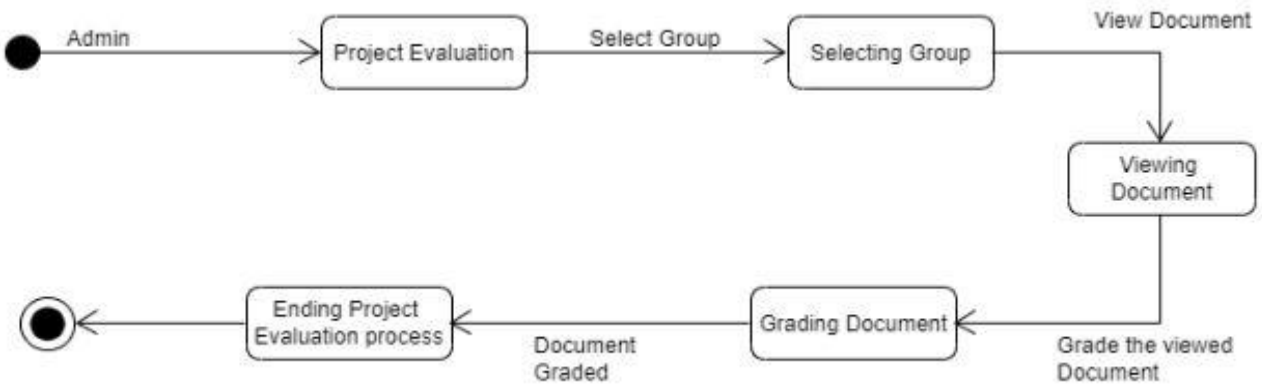


Figure 4. 33: project evaluation.

Sequence Diagram:

Communication between different objects in multiple situations is shown in sequence diagram. The emphasis of sequence diagram is time ordering the message.

4.5.13 Login Sequence Diagram

Figure 4.30 defines information flow between system application and user. The user request to the login page and the system check whether the user is registered or not. If the user is registered then the user will be successfully logged in otherwise user will register himself and then the user will be successfully logged in.

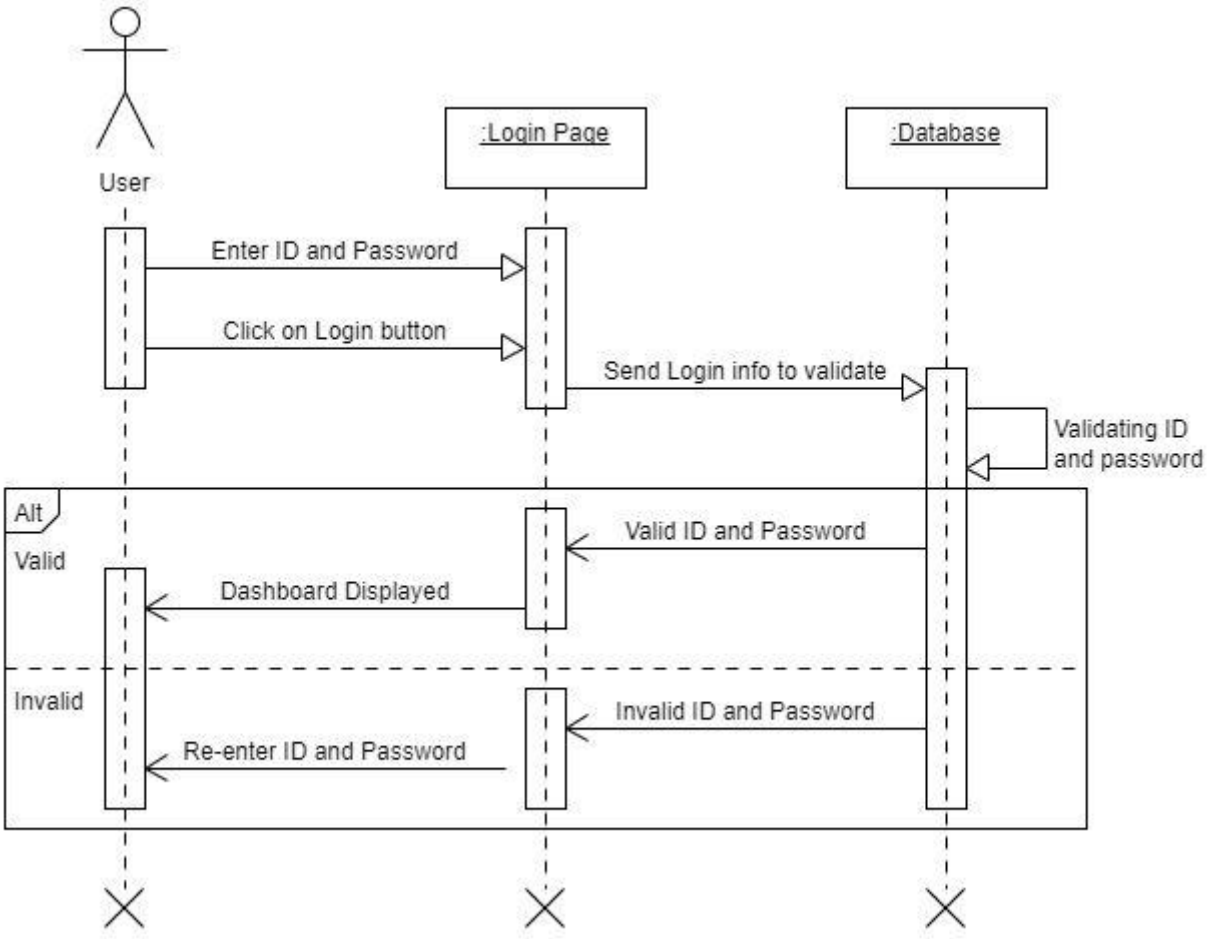


Figure 4. 34: Login Sequence Diagram

4.5.14 Register Sequence Diagram

Figure 4.31 describes the Register information flow for the user.

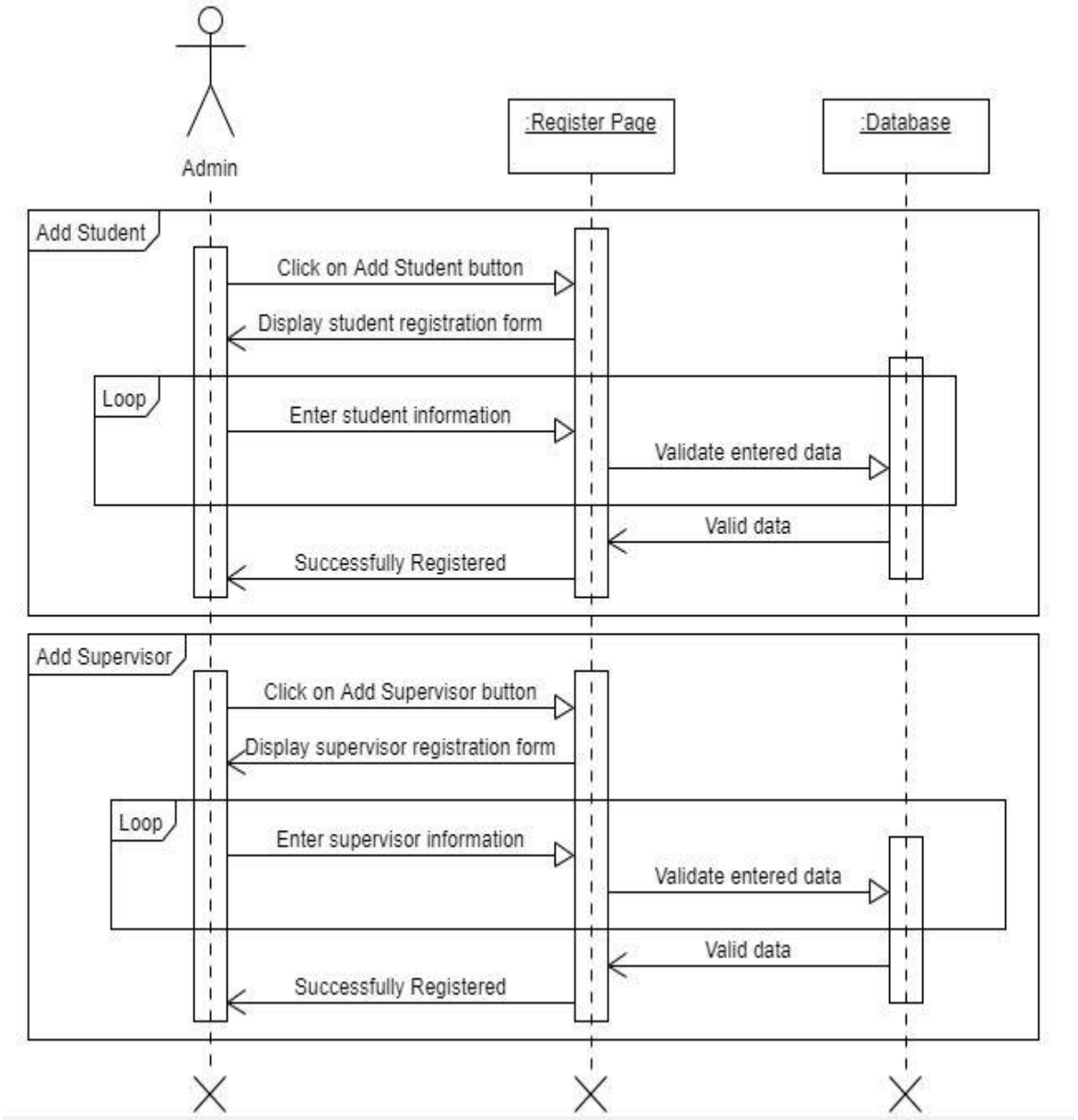


Figure 4. 35: Register Sequence Diagram

4.5.15 Manage Group Invitation Sequence Diagram

Figure 4.32 describes the information flow of manage group invitation to user.

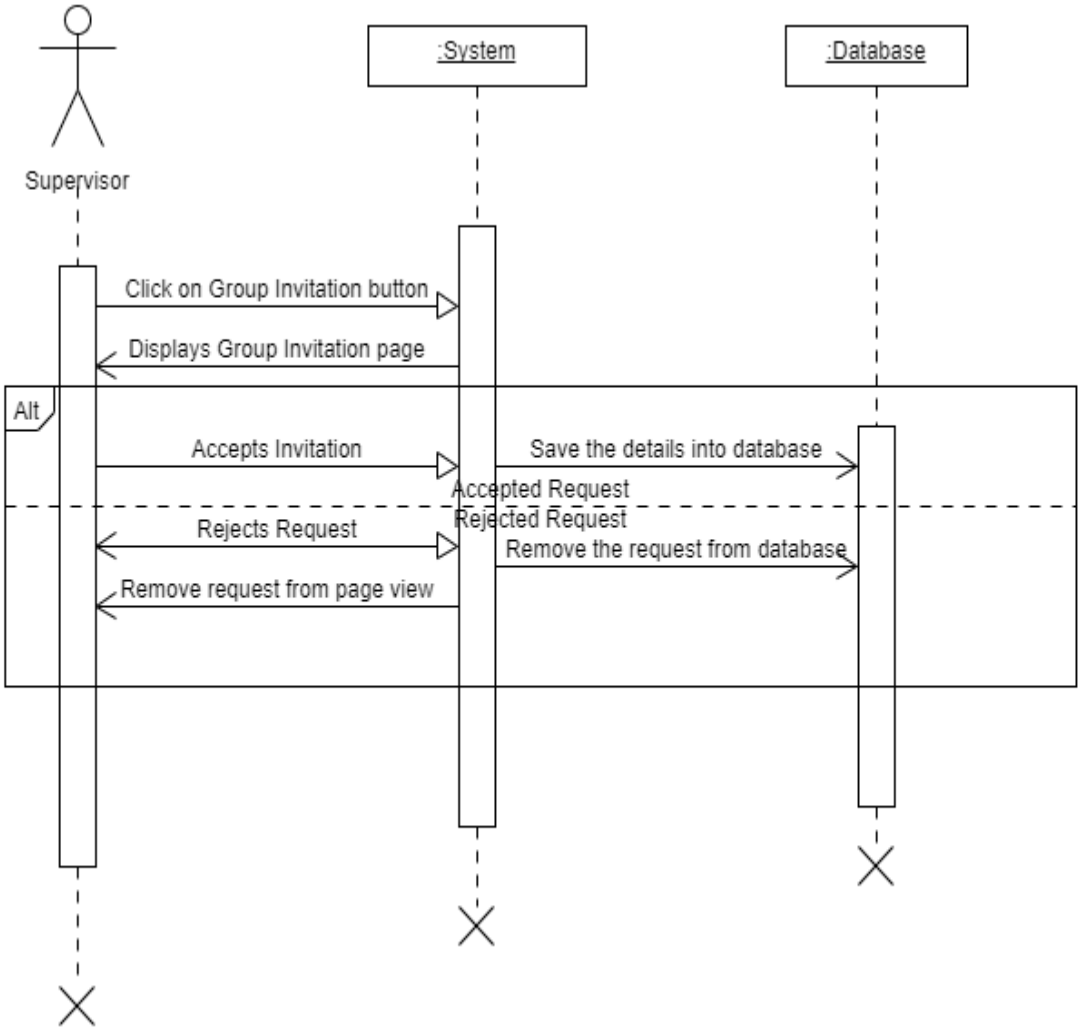


Figure 4. 36: Manage group Invitation

4.5.16 Make Group Sequence Diagram

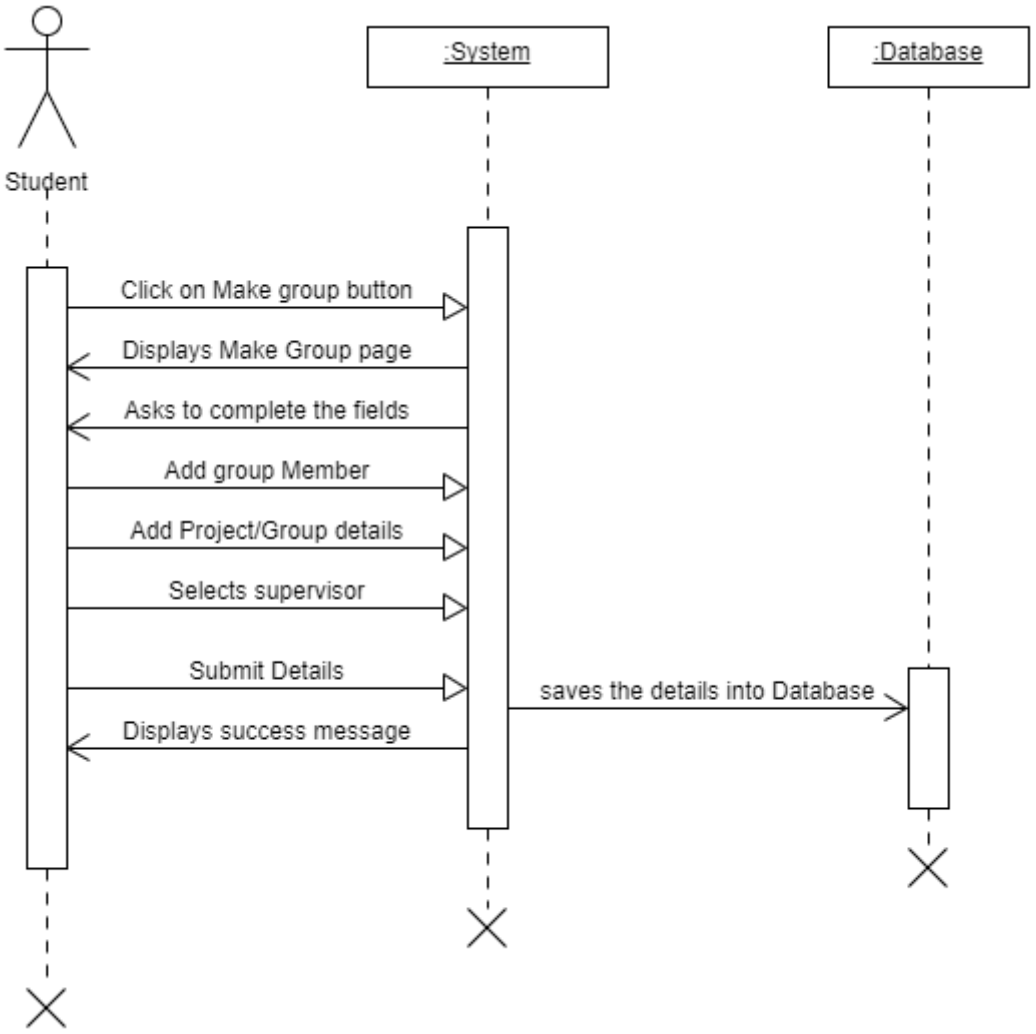


Figure 4. 37: Make group Sequence Diagram

4.5.17 Submit Document Sequence Diagram

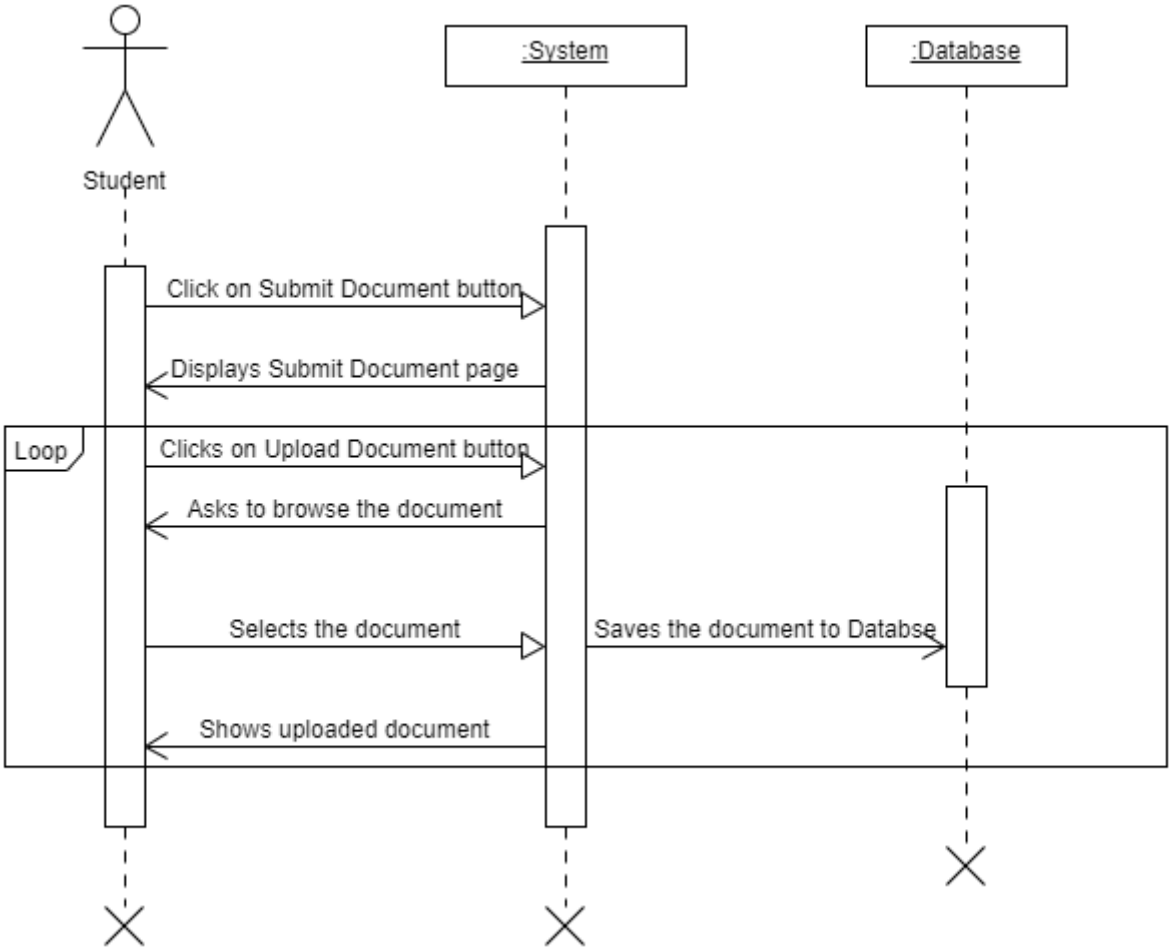


Figure 4. 38: submit document

4.5.18 Review Document Use case

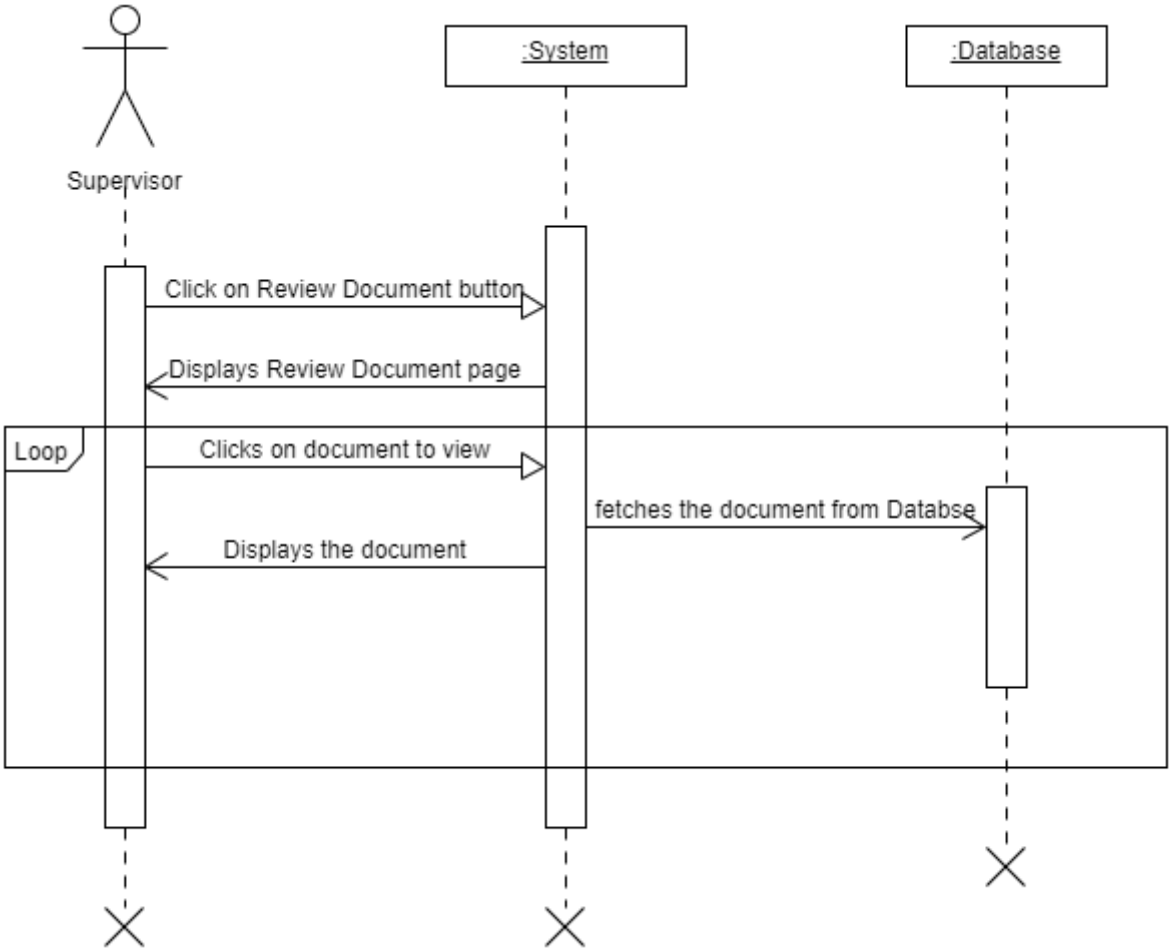


Figure 4. 39: submit document

4.5.19 Chat Sequence Diagram

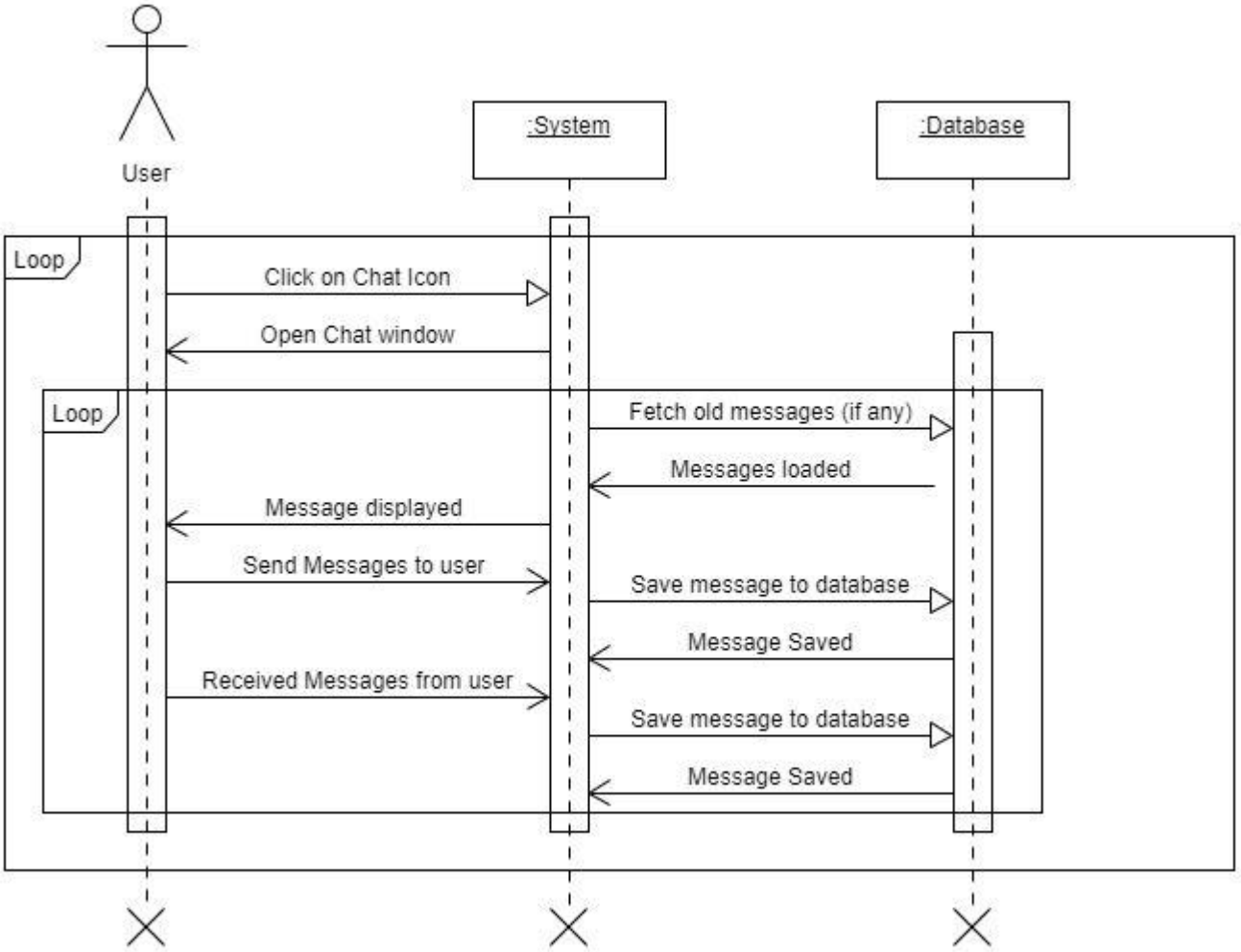


Figure 4. 40: Chat Sequence Diagram

4.5.20 Manage Meeting Log Sequence Diagram

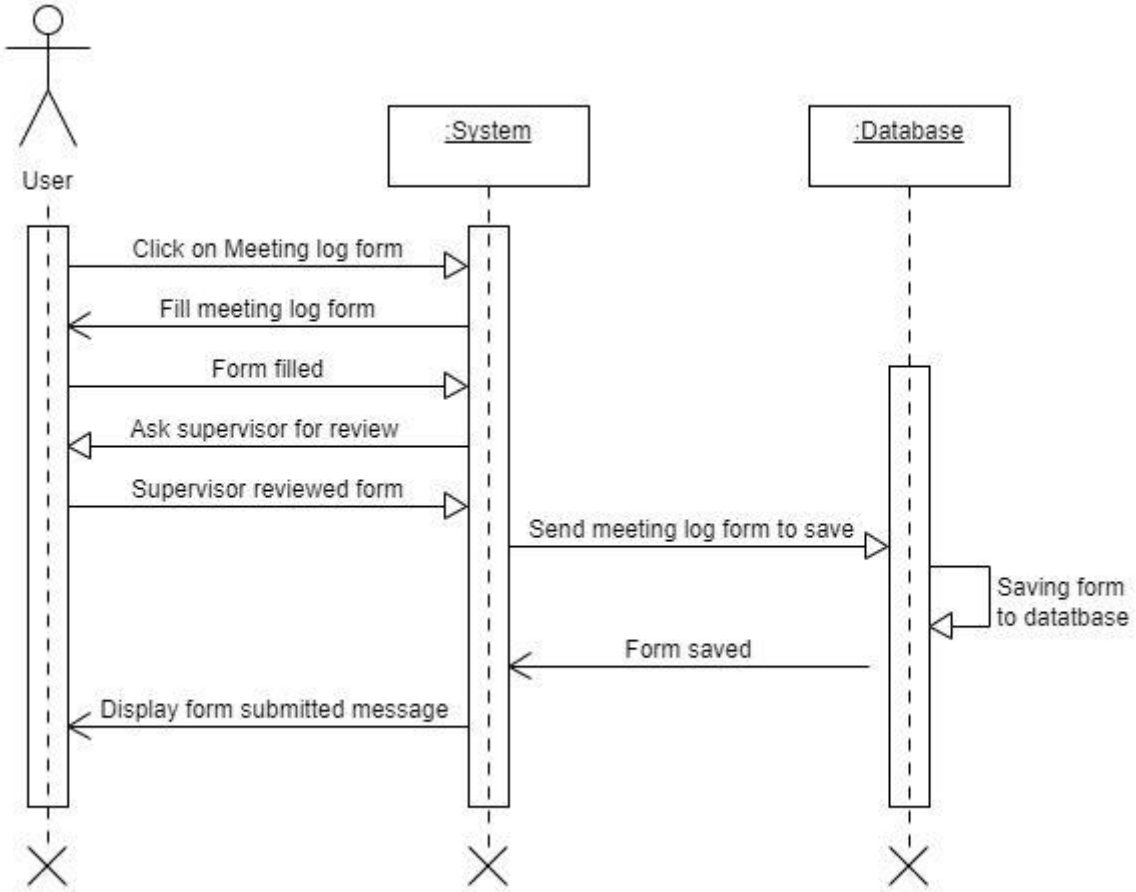


Figure 4. 41: Manage log Sequence Diagram

4.5.21 Guidelines

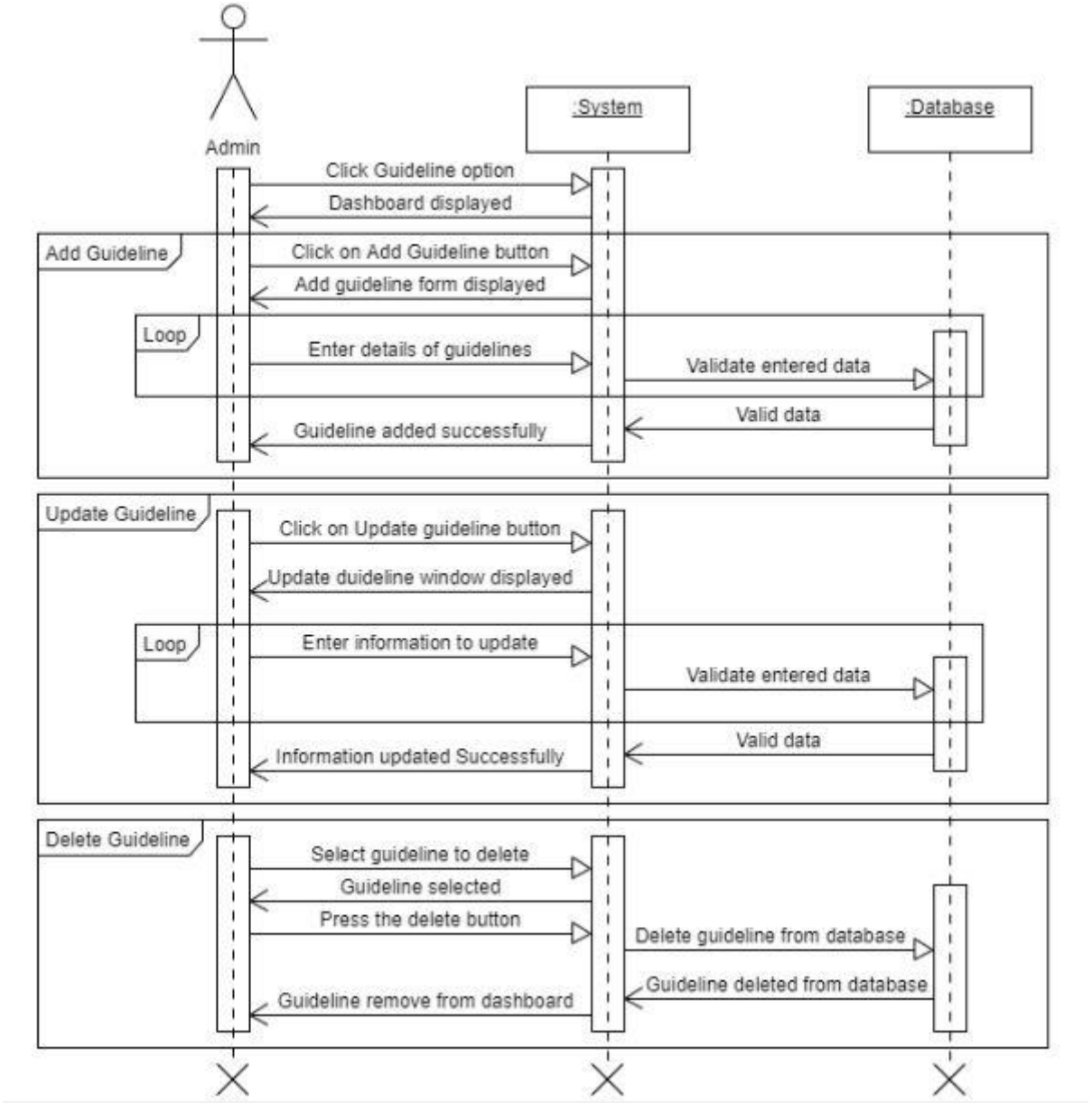


Figure 4. 42: Guidelines Sequence Diagram

4.5.22 View Group Details

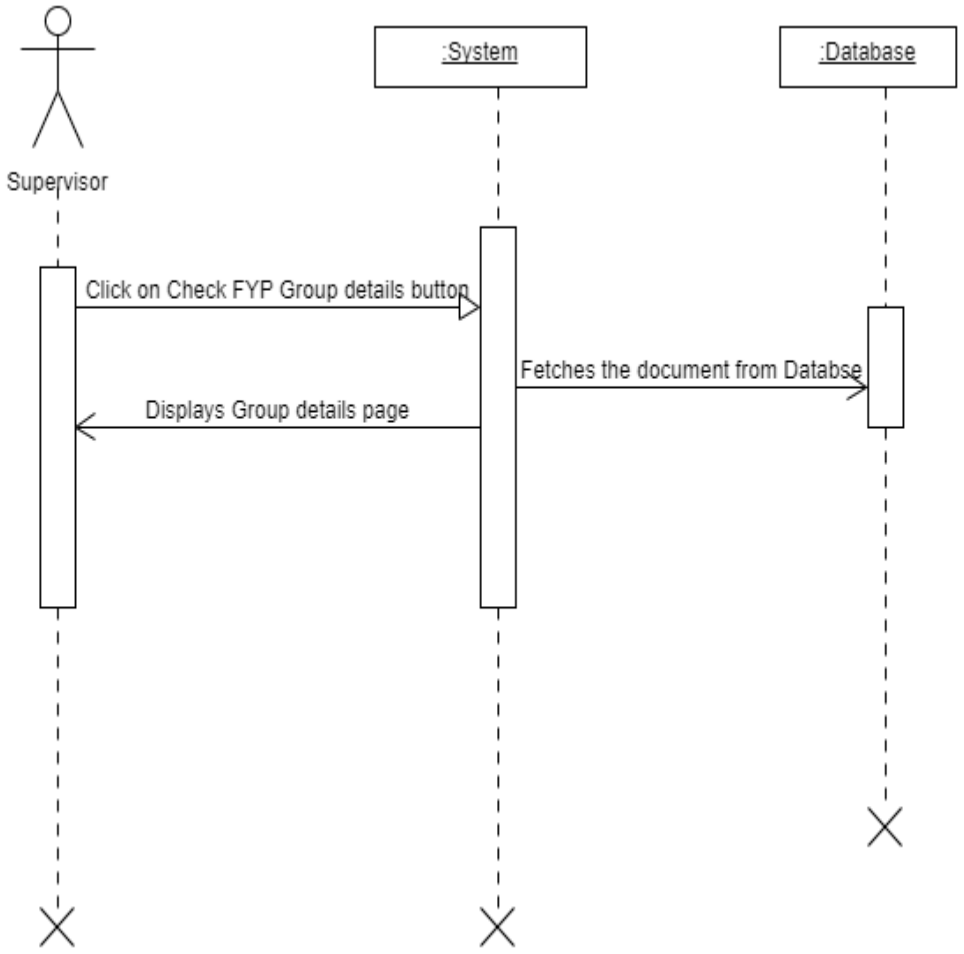


Figure 4. 43: FYP group details Sequence Diagram

4.5.23 Manage List Sequence Diagram

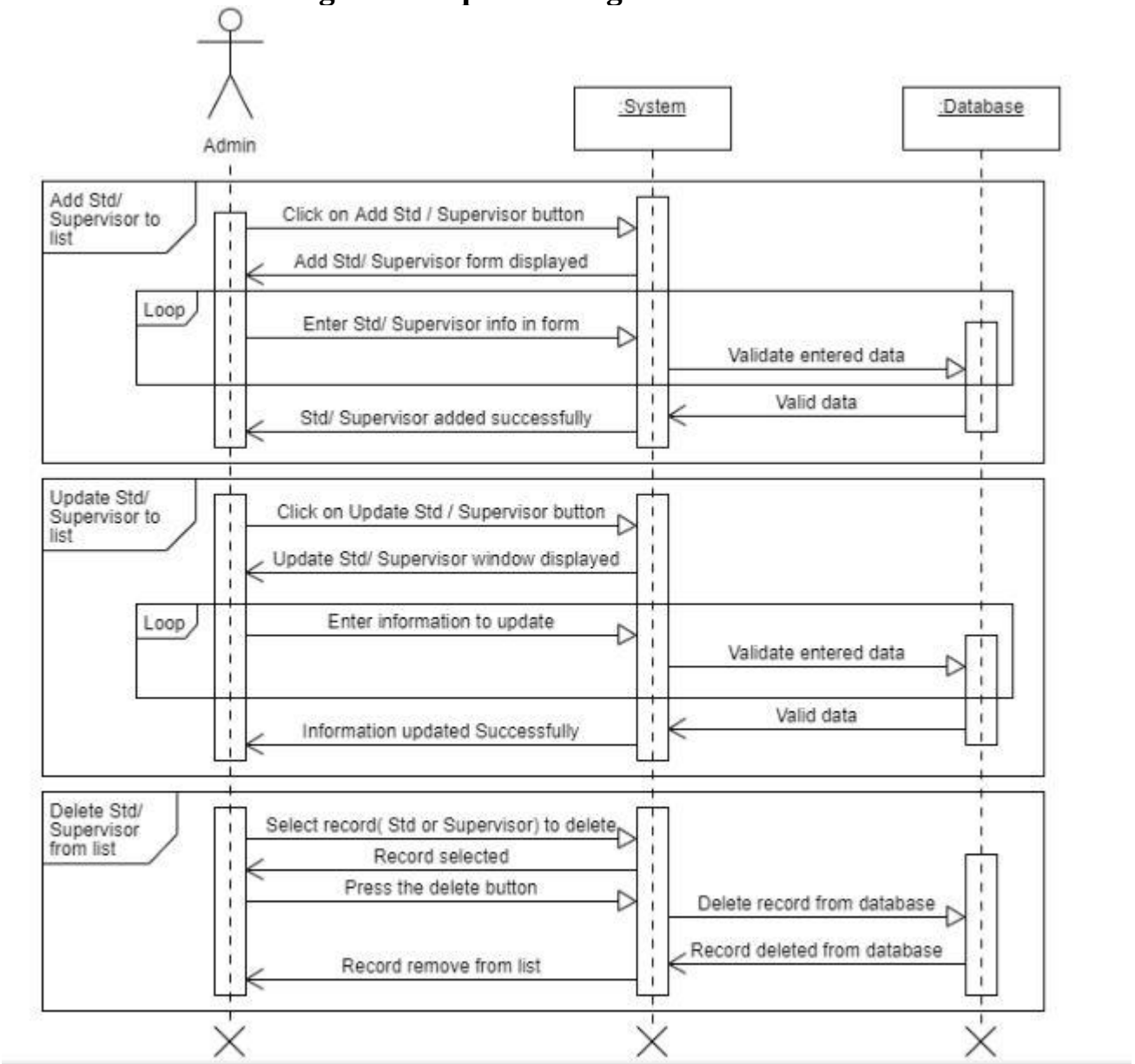


Figure 4. 44: Manage List Sequence Diagram

4.5.24 Repository Sequence Diagram

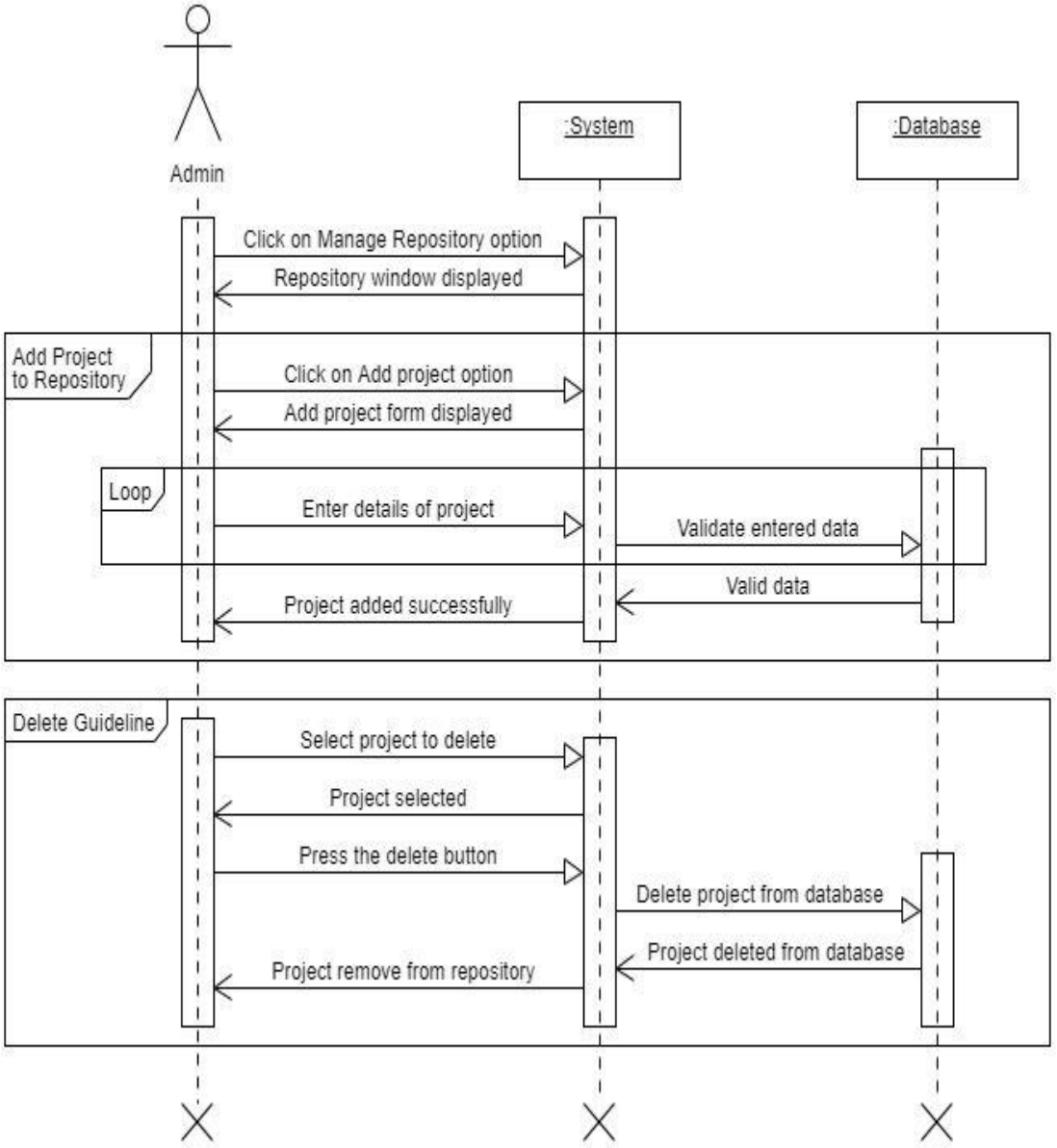


Figure 4. 45: Repository Sequence Diagram

4.5.25 Project Evaluation Sequence Diagram

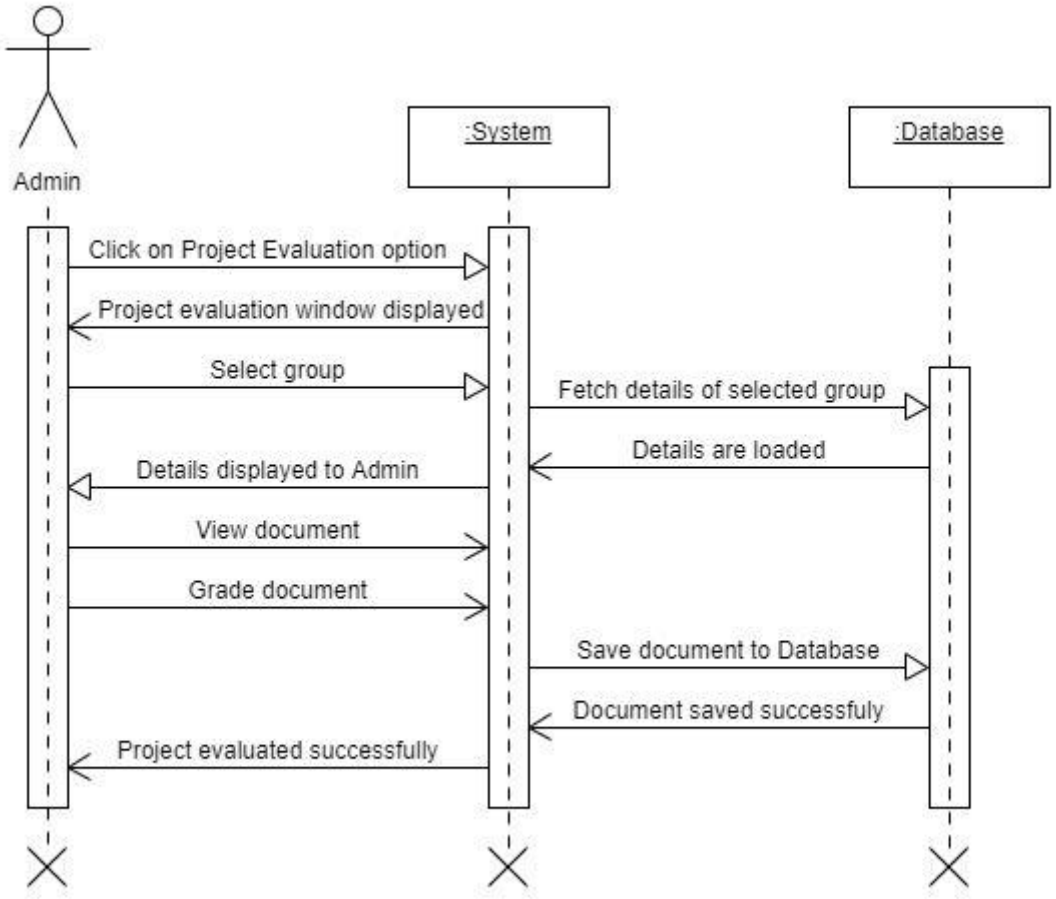


Figure 4. 46: Project Evaluation Sequence Diagram

4.6 Development view

The development architecture focuses on the actual software module organization on the software development environment. The software is packaged in small chunks; program libraries, or subsystems; that can be developed by one or a small number of developers. The development view includes:

4.6.1 Component diagram

The component diagram is a static view diagram of components of the system and the relationships between those components. A node in this diagram represents a component with one or more embedded classes with a well-defined interface. Figure 4. Shows a component diagram of this system.

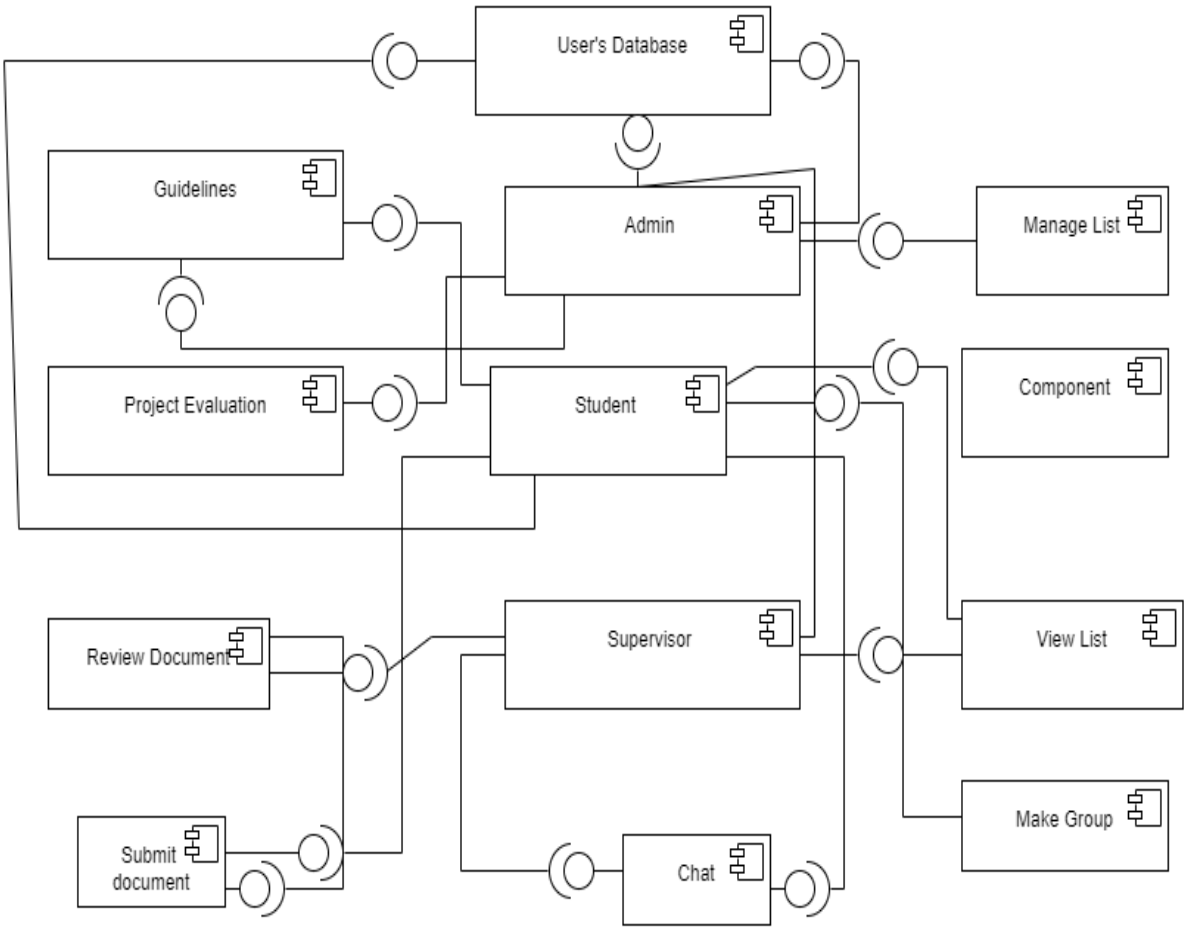


Figure 4. 47*: Component Diagram.*

4.7 Physical view

The physical constructs consider predominantly the nonfunctional requirements of the system which include, reliability, security, performance and scalability. The physical view consists of following:

4.7.1 Deployment Diagram

Figure 4.37 describe the deployment view of the system developed. The deployment diagram provides a visualization of the topology of physical components of a system. So this diagram is used to describe static view of a system.

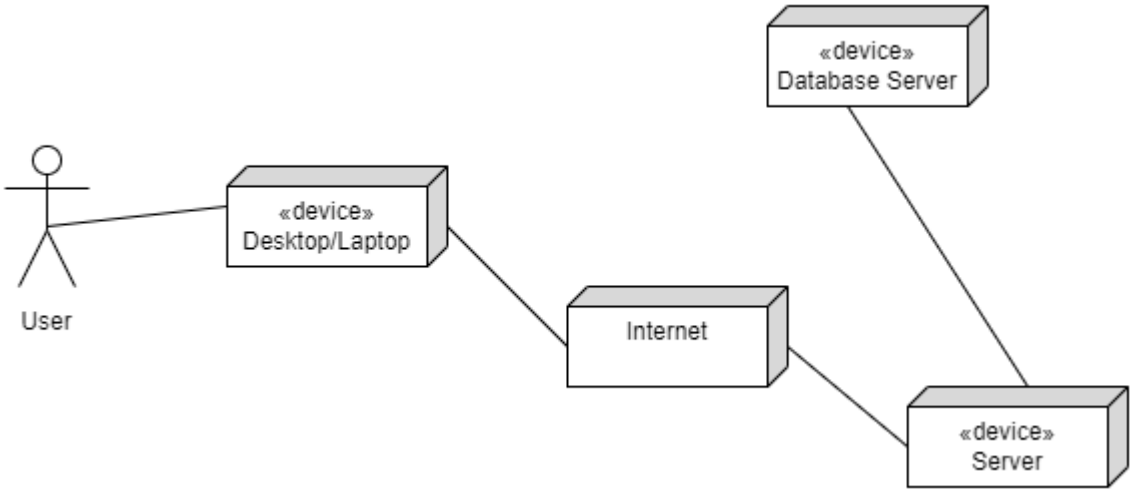


Figure 4. 50*:* Deployment Diagram

4.8 Summary

This chapter covers modelling of system. 4+1 View is described. This chapter includes the modelling of the system. The proper modelling through UML diagrams has been covered in this chapter. There are different diagrams developed such as activity diagram which shows flow from one activity to another in system, Component diagram which shows a static view of system components and the relationship among those components, Class diagram describes classes of system and their attributes, ERD shows logical data representation. All the diagrams are modelled through UMLET tool, we decided to use this tool because of its convenient features and diverse options.

References

[1] "Learning Management System: Log in to the site", *Lms2.numl.edu.pk*, 2021. [Online]. Available: https://lms2.numl.edu.pk/login/index.php. [Accessed: 29- Mar- 2021].

[2] "UMLetino - Free Online UML Tool for Fast UML Diagrams", *Umlet.com*, 2021. [Online]. Available: http://www.umlet.com/umletino/umletino.html. [Accessed: 29- Mar- 2021].

[3]*R.search.yahoo.com*, 2021. [Online]. Available: https://r.search.yahoo.com/_ylt=A2KIbMdYwGFgfwgA16NXNyoA;_ylu=Y29sbwNiZjEEcG9zAzEEdnRpZANDMTYxN18xBHNlYwNzcg--/RV=2/RE=1617047768/RO=10/RU=https%3a%2f%2fnodejs.org%2f/RK=2/RS=DmAbX0N_w9nIggOIWiol_c0iBnY-. [Accessed: 29- Mar- 2021].

[4]"The most popular database for modern apps", *MongoDB*, 2021. [Online]. Available: https://www.mongodb.com/. [Accessed: 29- Mar- 2021].