SOFTWARE PROJECT LAB MANAGEMENT SYSTEM

Software Requirement Specification

Institute of Information Technology
University of Dhaka

SPL COURSES AUTOMATION

Submited to

Md. Saeed Siddik

Assistant Professor
Institute of Information Technology
University of Dhaka

Submited by

Sabbir Ahmed
BSSE 1041

Md. Hasanuzzaman
BSSE 0943

5th Semester

Institute of Information Technology

University of Dhaka

Submission date: 22nd October 2020

Letter of Transmittal

22nd October 2020

Md. Saeed Siddik

Assistant Professor

Institute of Information Technology

University of Dhaka

Subject: Submission of term report on “Software Project Lab (SPL) Management System”.

Sir,

With due respect, we are submitting the report on the above topic you assigned to us. In this report, we have given our best effort albeit some shortcomings. We earnestly hope that you would excuse our errors and oblige thereby.

Yours sincerely

Sabbir Ahmed	BSSE 1041
Md. Hasanuzzaman	BSSE 0943

5th Semester

Institute of Information Technology

University of Dhaka

Session: 2020-21

Acknowledgement

We are highly indebted for getting such a tremendous opportunity to prepare the report on SPL courses automation. We would like to thank whole-heartedly, Md Saeed Siddik, Assistant Professor for giving us guidelines about how we can prepare this report. In completing this paper, we have collected various important data and information. We are thankful to all of the works cited.

Abstract

The study is made for Software Project Lab courses of Institute of Information Technology automation. The scope of the study is to study the full process of Software Project Lab 1 and 2 and design the SRS of this system. The object of this study is to develop an SRS (software requirements and specification) of Software Project Lab Management System.

Table of Contents

Contents

Table of Figures

List of Tables

CHAPTER 1: INTRODUCTION

This chapter is a part of our software requirement specification for the project "SPL Management System". In this chapter we focus on the intended audience for this project.

1.1 PURPOSE

This document briefly describes the Software Requirement Analysis of SPL Management System. It contains functional, non-functional and supporting requirements and establishes a requirements baseline for the development of the system. The requirements contained in the SRS are independent, uniquely numbered and organized by topic. The SRS serves as an official means of communicating user requirements to the developer and provides a common reference point for both the developer team and the stakeholder community. The SRS will evolve over time as users and developers work together to validate, clarify and expand its contents.

1.2 INTENDED AUDIENCE

This SRS is intended for several audiences including the customers as well as the project managers, designers, developers, and testers.

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

and ensure that all requirements documented in this SRS have been fulfilled.

1.3 Conclusion

This analysis of the audience helped us to focus on the users who will be using our analysis. This overall document will help each and every person related to this project to have a better idea about the project.

CHAPTER 2: INCEPTION OF SPLMS

In this chapter, the Inception part of the SRS will be discussed briefly.

2.1 INTRODUCTION

Inception is the beginning phase of requirements engineering. It defines how a software project gets started and what the scope and nature of the problem to be solved is. The goal of the inception phase is to identify concurrent needs and conflicting requirements among the stakeholders of a software project. At project inception, we establish a basic understanding of the problem, the people who want a solution, the nature of the solution that is desired and the effectiveness of preliminary communication and collaborations between the other stakeholders and the software team.

To establish the groundwork we have worked with the following factors related to the inception phases:

- List of stakeholders
- Recognizing multiple viewpoints
- Working towards collaboration
- Requirements questionnaire

2.1.1 List of Stakeholders

Stakeholder refers to any person or group who will be affected by the system directly or indirectly. Stakeholders include end-users who interact with the system and everyone else in an organization that may be affected by its installation. At inception, a list of people who will contribute input as requirements are elicited. The initial list will grow as stakeholders are contacted because every stakeholder will be asked: “Whom else do you think I should talk to?”

To identify the stakeholders we consulted with Manager and asked him following questions:

- Who is paying for the project?
- Who will be using the project outcomes?
- Who gets to make the decisions about the project (if this is different from the money source)?
- Who has resources I need to get the project done?

- Whose work will my project affect? (During the project and also once the project is completed).

We identified the following stakeholders for our automated SPL management system of Institute of Information Technology.

The committee: A committee is formed with 6 members. There are two managers and four members. Managers plays a vital role in SPL Management System. Almost all the decisions are taken by the managers.

Students: Here we are considering 3rd and 5th semesters students of Institute of Information technology.

Teachers: All the teachers of Institute of Information Technology

Developers: Developers are one of the stakeholder because they are also affected by this system. They develop this system and work for further development. If there occurs any system interruption, they will find the problem and try to solve it.

2.1.2 Recognizing Multiple Viewpoints

Different stakeholders achieve different benefits from the system. Consequently, each of them has a different view of the system. So we have to recognize the requirements from multiple points of view, as well as multiple views of requirements. Assumptions are given below:

Committee's viewpoint:

- User friendly and efficient system
- Error free system
- Easy to operate
- Minimum maintenance cost
- Store information easily

Student's viewpoint:

- Easy to access
- User friendly
- Use softcopy instead of hard copy

Teacher's viewpoint:

- Easy to access
- User friendly

Developer's viewpoint:

- Easy to develop
- No ambiguous requirement

Admin's viewpoint:

- User friendly
- Efficient system
- Easy to access

2.1.3 Working towards Collaboration

Every stakeholder has their own requirements. There are some common and conflicting requirements of our stakeholder. That's why we followed the following steps to merge these requirements-

- ✓ Find the common and conflicting requirements
- ✓ Categorize them
- ✓ List the requirements based on stakeholder's priority points
- ✓ Make final decision about requirements

Common requirements:

- ✓ User friendly
- ✓ Efficient system
- ✓ Error free system

Conflicting requirements:

- ✓ High security of the system
- ✓ Easy access (Different stakeholder wants different type of access)

Final requirements: We finalize the following requirements based on stakeholder's priority point:

- Easy to access

2.1.4 Requirements Questionnaire

We first ask the stakeholder some context free questions to understand the project's overall performance and goals. These questions are mentioned in section 2.1.1. These questions help us to identify the stakeholders of the project. Then we ask our next set of questions to better understand the problem and take stakeholder's opinion about the solution. The final set of question focused on the effectiveness of the communication activity itself.

2.2 CONCLUSION

The Inception phase helped us to establish basic understanding about the Software Project Lab management system of Institute of Information technology, identify the stakeholders who will be benefited if this system becomes automated, define the nature of the system and the tasks done by the system, and establish a preliminary communication with our stakeholders.

In our project, we have established a basic understanding of the problem, the nature of the solution that is desired and the effectiveness of preliminary communication and collaboration between the stakeholders and the software team. More studies and communication will help both sides (developer and client) to understand the future prospect of the project. Our team believes that the full functioning document will help us to define that future prospect.

CHAPTER 3: ELICITATION OF SPLMS

After discussing on the Inception phase, we need to focus on the Elicitation phase. So this chapter specifies the Elicitation phase.

3.1 INTRODUCTION

Requirements Elicitation is a part of requirements engineering that is the practice of gathering requirements from the users, customers and other stakeholders. We have faced many difficulties, like understanding the problems, making questions for the stakeholders, limited communication with the stakeholders due to a short amount of time and volatility. Though it is not easy to gather requirements within a very short time, we have surpassed these problems in an organized and systematic manner.

3.2 ELICITING REQUIREMENTS

We have seen Question and Answer (Q&A) approach in the previous chapter, where the inception phase of requirement engineering has been described. The main task of this phase is to combine the elements of problem solving, elaboration, negotiation and specification. The collaborative working approach of the stakeholders is required to elicit the requirements. We have finished the following tasks for eliciting requirements-

- Collaborative Requirements Gathering
- Quality Function Deployment
- Usage Scenarios
- Elicitation work products

3.2.1 Collaborative Requirements Gathering

We have met with many stakeholders in the Inception phase such as the manager, teachers and students. These meetings created an indecisive state for us to elicit the requirements. To solve this problem we have met with the stakeholders (who are acting a vital rule in the whole process) again to elicit the requirements. A slightly different scenario from these approaches has been found.

Following activities have been completed to accomplish this task.

- The meetings were conducted with the manager, teachers, admin and students; they were questioned about their requirements and expectations from the SPL Management System.
- They were asked about the problems they were facing with the current manual system.
- Lastly we selected our final requirement list from the meetings.

3.2.2 Quality Function Deployment

Quality Function Deployment (QFD) is a technique that translates the needs of the customer into technical requirements for software. Ultimately the goal of QFD is to translate subjective quality criteria into objective ones that can be quantified and measured and which can then be used to design and manufacture the product. It is a methodology that concentrates on maximizing customer satisfaction from the software engineering process. So we have followed this methodology to identify the requirements for the project. The requirements, which are given below, are identified successfully by the QFD.

3.2.2.1 Normal Requirements

Normal requirements are generally the objectives and goals that are stated for a product or system during meetings with the customer. The presence of these requirements fulfills customers' satisfaction. These are the normal requirements for our project.

1. User-friendly design.
2. Allow new teachers to register for system.
3. Students and teachers can upload documents easily.
4. Teachers and manager can give feedback easily in online.
5. User can see previous year’s projects topics.

3.2.2.2 Expected Requirements

These requirements are intrinsic to the product or system and may be so elementary that the customer does not explicitly state them. Their absence will be a cause for significant dissatisfaction. Below the expected requirements for our project are briefly described.

1. IDs is provided so that only valid users can safely login to the system and see notification.
2. Manager can change academic calendar.
3. Manager can change group's presentation time.
4. Every user will have an account to access the system.

3.2.2.3 Exciting Requirements

These requirements are for features that go beyond the customer's expectations and prove to be very satisfying when present. Following are some exciting requirements of our project.

1. Admin will upload documents in GitHub at the end of the courses.

3.2.3 Usage Scenario

Every year the 3rd and 5th semester students of IIT take two mandatory courses SPL1 and SPL2. In the courses the students complete a project. Software Project Lab Management System (SPLMS) will be an automated system to maintain the courses and track the projects.

There are two separate committee to maintain SPL1 and SPL2 courses. Each committee will consist of 6 members (4 general members, 2 managers) . Committee members are teachers.

There is an admin who can update the whole process assigning new committee members and managers. She can assign new admin also.

Authentication and Registration

There are three types of user. The teachers, students (roll no, batch) and admin. The users need to sign in to the system. To sign in first they need to register. To register the user need to submit their name, e-mail address, password and course role. An e-mail will be sent to that mail and the mail will contain a link. By clicking the link the registration process will finish. After registration the user can sign in the system by submitting the e-mail address and password. If any user forget her password she can retrieve the password. A mail consisting of her password will be sent.

Initialization

The manager will start the course with academic calendar. The academic calendar includes- 1st midterm date/proposal date, 2nd midterm date, draft report submission date, final report submission date, final presentation date.

She will invite all the teachers to form groups via notification. When the teachers will sign in, they will see the notification.

Before presentation date the manager will declare the presentation time of the groups, the name of the teachers who will attend the group's presentation. The manager can change the academic calendar and number of groups that one teacher can supervise.

Forming Groups

After getting the notification, the teacher can create a group. To create a group, she will have to set-group name, project topic, group member/s (student).

Then she will invite the selected student/s via notification to join the group. One teacher can supervise multiple groups. The number of groups one can supervise will be decided by the managers. The students can join their group after getting the notification. All the students need to register to get the notification.

The managers can create a group with group details and supervisor name and will notify the students. The students will join the through the notification. The teacher of the group will be treated as the supervisor of the group.

Feedback

Before presentation the group will upload their documents to the system. The documents can be four types -proposal, SRS report (only for SPL2), draft report, final report. The committee will submit the feedback of the draft report. The assigned teacher can give their feedback at presentation time. They can give their feedback through web interface. If any group is resubmitted at final presentation the manager will set a resubmission date and time of the group's presentation. According to the feedback the committee will generate the marks sheet manually. The managers will upload the softcopy of the marks sheet.

Updating Course and Forming New Committee

At the start of the semester the admin will update the courses assigning new committee. She will assign two different committee to maintain the two courses. Then the new committee managers can start the courses.

3.2.4 Scope of the Project

The system will work only for SPL1 and SPL2 of IIT.

At the time of registration the user will get a mail for confirmation.

The invitation will occur via notification.

The manager can change the academic calendar or the group formation rule.

The marking system is manual. The manager will upload the softcopy of the mark sheet.

CHAPTER 4: SCENARIO BASED MODELING OF SPLMS

This chapter describes the Scenario Based Model for the Dhaka University Club management System.

4.1 INTRODUCTION

Although the success of a computer-based system or product is measured in many ways, user satisfaction resides at the top of the list. If we understand how end users (and other actors) want to interact with a system, our software team will be better able to properly characterize requirements and build meaningful analysis and design models. Hence, requirements modeling begins with the creation of scenarios in the form of Use Cases, activity diagrams and swim lane diagrams.

4.2 DEFINITION OF USE CASE

A Use Case captures a contract that describes the system behavior under various conditions as the system responds to a request from one of its stakeholders. In essence, a Use Case tells a stylized story about how an end user interacts with the system under a specific set of circumstances. A Use Case diagram simply describes a story using corresponding actors who perform important roles in the story and makes the story understandable for the users.

The first step in writing a Use Case is to define that set of “actors” that will be involved in the story. Actors are the different people that use the system or product within the context of the function and behavior that is to be described. Actors represent the roles that people play as the system operators. Every user has one or more goals when using system.

Primary Actor

Primary actors interact directly to achieve required system function and derive the intended benefit from the system. They work directly and frequently with the software.

Secondary Actor

Secondary actors support the system so that primary actors can do their work. They either produce or consume information.

4.3 USE CASE DIAGRAMS OF SPLMS

Use Case diagrams give the non-technical view of overall system.

4.3.1 Level-0 Use Case Diagram-SPLMS

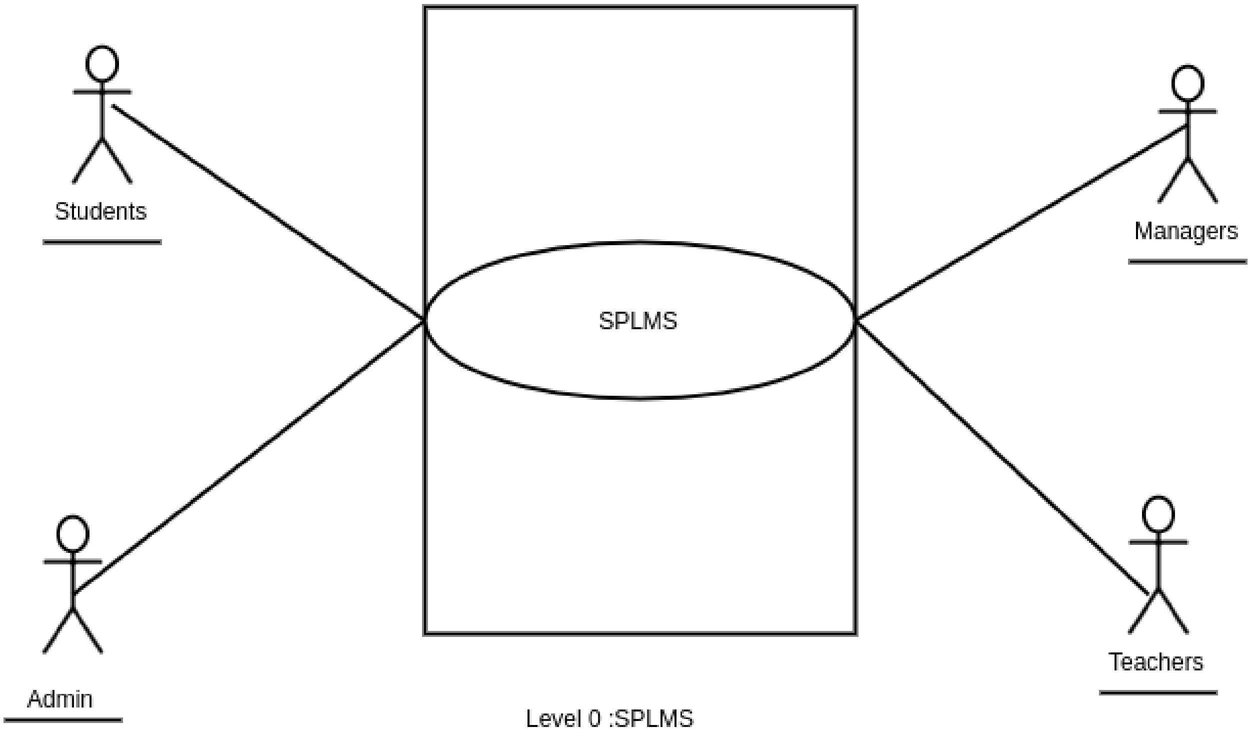


Figure 1 Level-0 Use Case Diagram

Name:	SPL Management System
ID:	SPLMS-L-0
Primary Actors:	Student, Manager, Teacher, Admin

Description of Use Case Diagram Level 0:

After analyzing the scenario, we found four actors who will directly use the system as a system operator. Primary actors are those who will play action and get a reply from the system whereas secondary actors only produce or consume information.

Following are the actors of Software Project Lab management system –

- Admin (primary)
- Manager (primary)
- Teacher (primary)
- Student (primary)

4.3.2 Level-1 Use Case Diagram-Subsystems

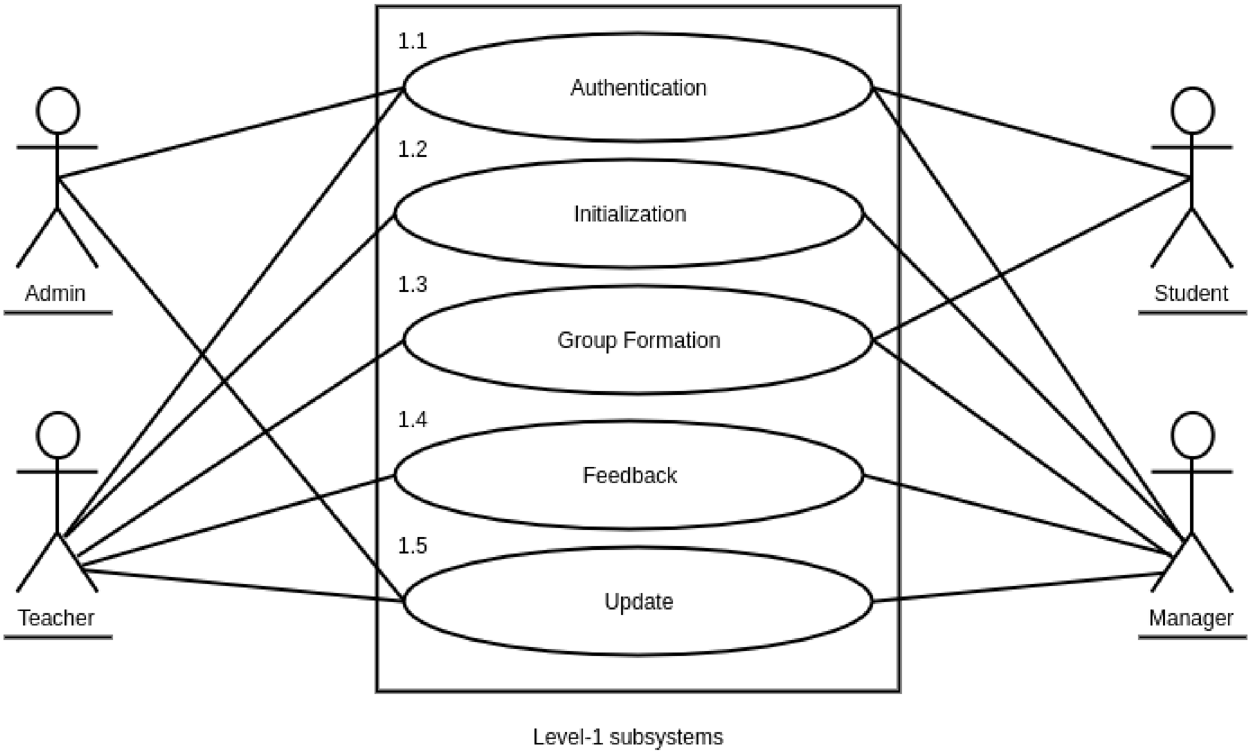


Figure 2: Level 1 Use Case Diagram Subsystems

Name:	Subsystems of SPL Management System
ID:	SPLMS-L-1
Primary Actors:	Student, Manager, Teacher, Admin

Description of Use Case Diagram Level 1:

There are five subsystems in the Club Management System. They are:

1. Authentication
2. Initialization
3. Group Formation

4. Feedback
5. Update

4.3.3 Level-1.1 Use Case Diagram-Authentication

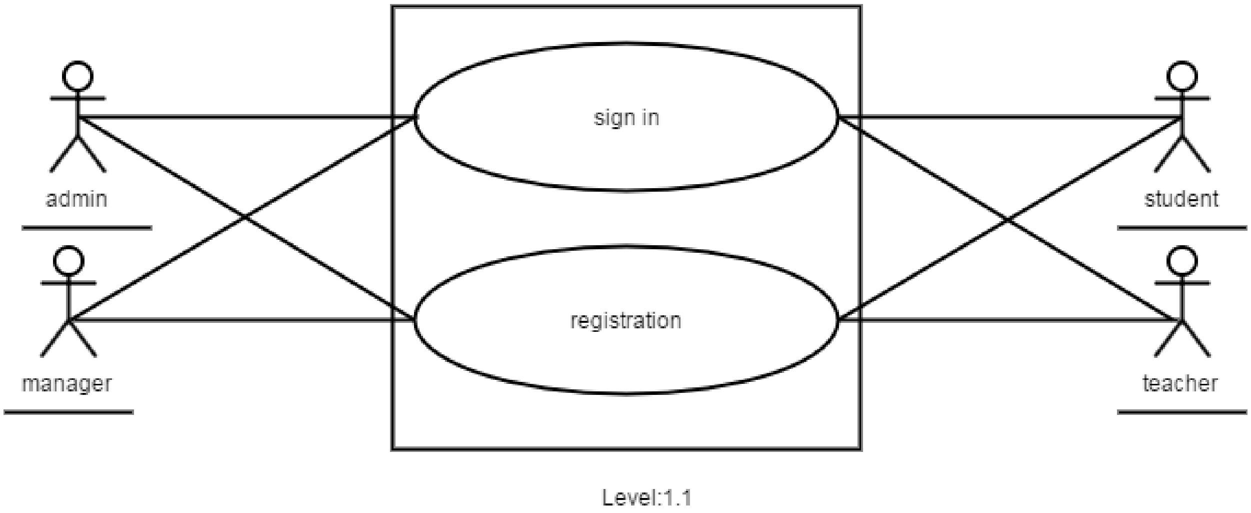


Figure 3: Level 1.1 Use Case Diagram Authentication

Name:	Authentication
ID:	SPLMS-L-1.1
Primary Actors:	Student, Manager, Teacher, Admin

Description of Use Case Diagram Level 1.1:

Authentication is the process of determining whether someone or something is, in fact, who or what it is declared to be. The authentication subsystem of SPL Management System can be divided into two parts. They are:

1. Authentication
2. Registration

If the user is already registered than she can sign in to the system using her mail and password.

To register the user need to give her information. Then a mail will be sent to her.

Action-Reply of Use Case Diagram Level 1.1:

Action: User will input their e-mail address and password to sign in.

Reply: If the input is correct the user will sign in to the system.

Action: The user will input their name, e-mail address, password and course role to register.

Reply: A mail will be sent to the e-mail address for confirmation

4.3.4 Level-1.2 Use Case Diagram-Initialization

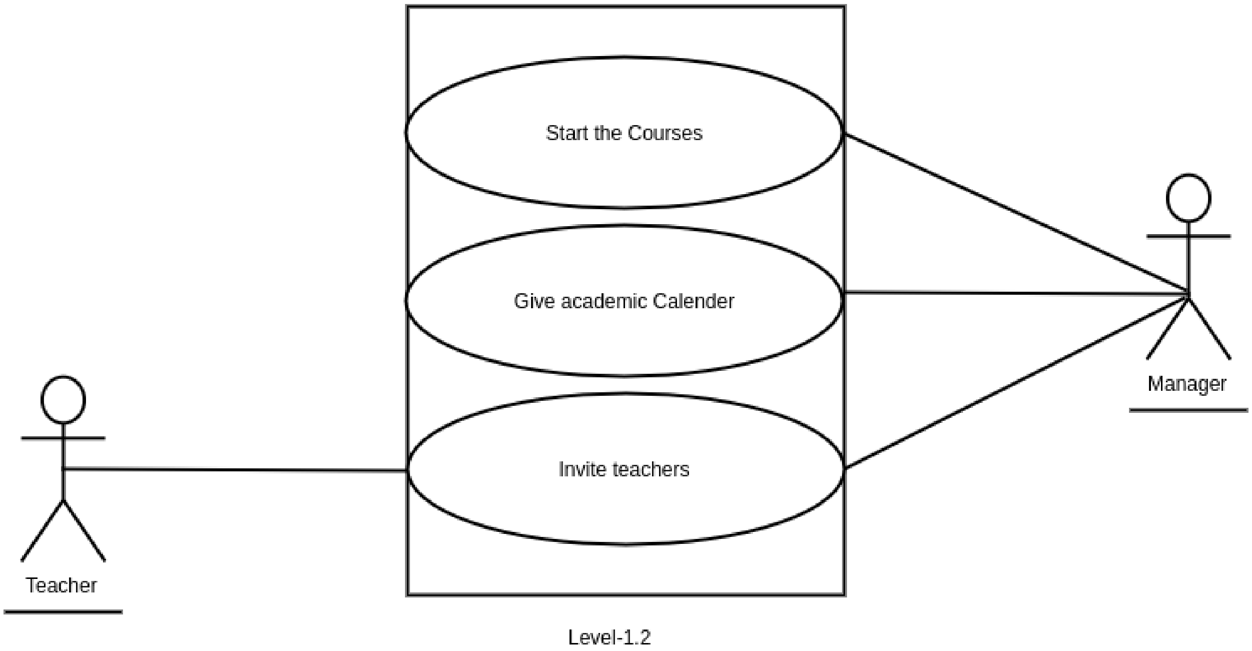


Figure 4: Level 1.2 Use Case Diagram Initialization

Name:	Initialization
ID:	SPLMS-L-1.2
Primary Actors:	Manager, Teacher

Description of Use Case Diagram Level 1.2:

The manager will initialize the process. She will create academic calendar and start the course. Then she will invite the teacher.

Action-Reply of Use Case Diagram Level 1.2:

Action: The manager will give the input to create academic calendar and sent an invitation.

Reply: An invitation notification will be sent to the teacher.

4.3.5 Level-1.3 Use Case Diagram-Group Formation

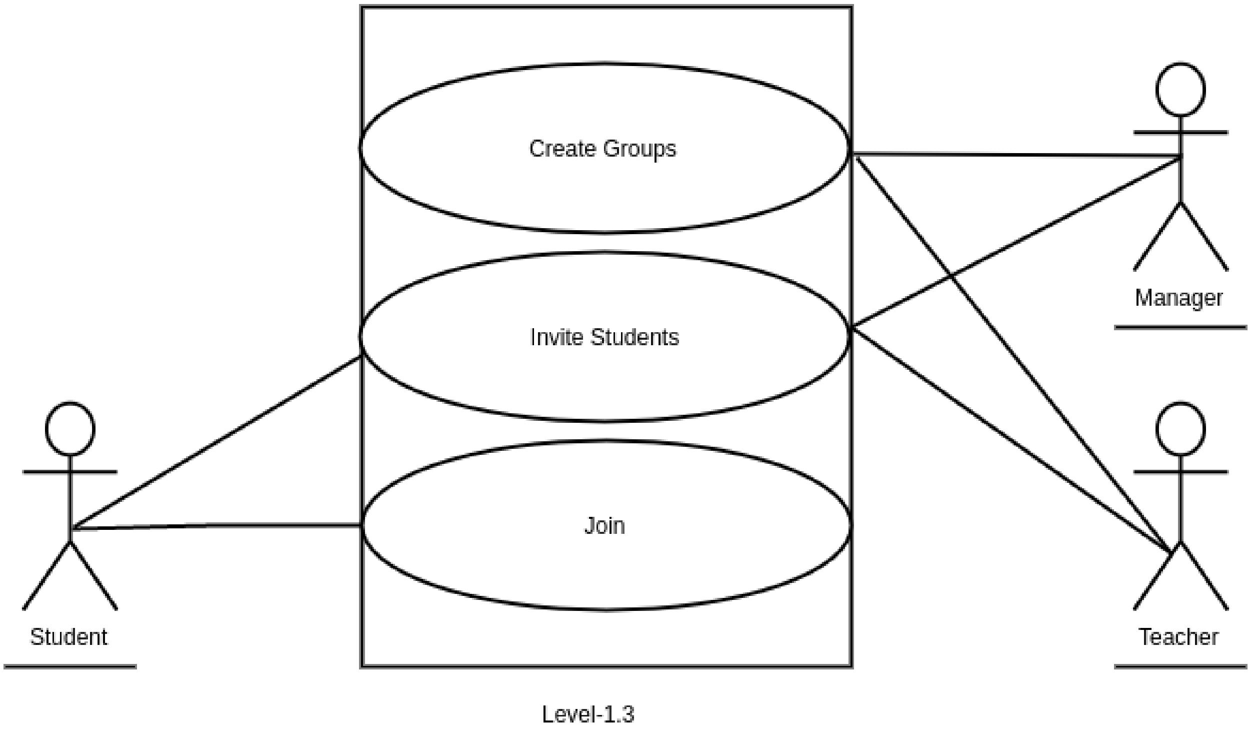


Figure 5: Level 1.3 Use Case Diagram of Group Formation

Name:	Group Formation
ID:	SPLMS-L-1.3
Primary Actors:	Manager, Teacher, Student

Description of Use Case Diagram Level 1.3:

The teacher create groups and sent invitation to student. After getting the invitation the student can join the group.

Action-Reply of Use Case Diagram Level 1.3:

Action: The teacher will input to create group and sent invitation to the selected student.

Reply: The selected student will get a notification and will join the group.

4.3.6 Level-1.4 Use Case Diagram-Feedback

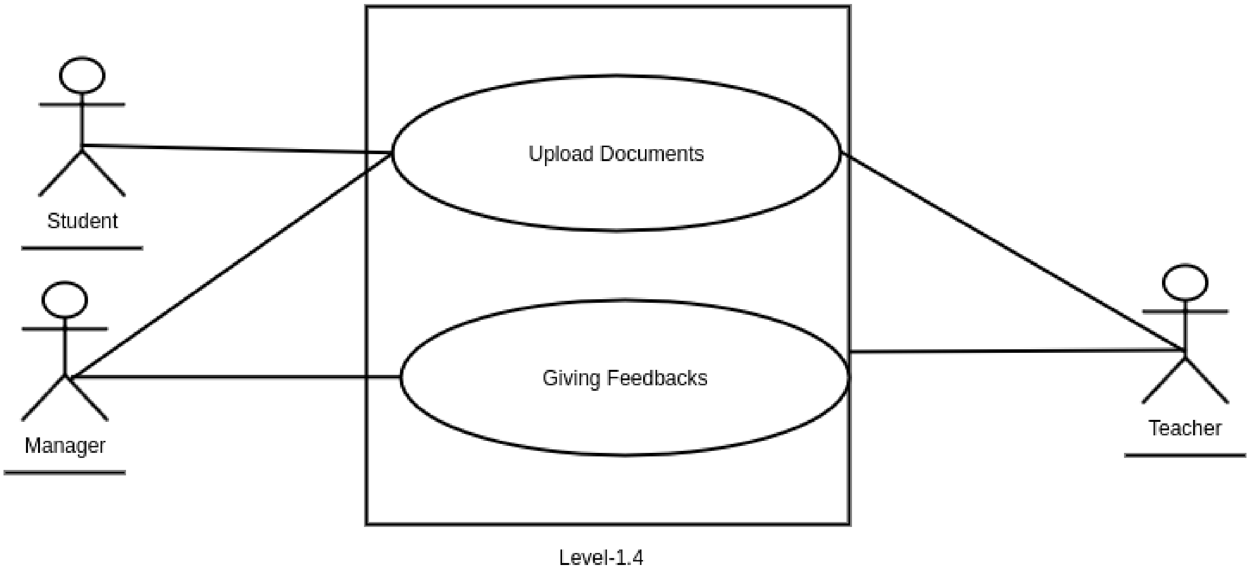


Figure 6: Level 1.4 Use Case Diagram of Feedback

Name:	Feedback
ID:	SPLMS-L-1.4
Primary Actors:	Student, Teacher, Manager

Description of Use Case Diagram Level 1.4:

The student will upload their document. At presentation the teacher can give feedback.

Action-Reply of Use Case Diagram Level 1.4:

Action: The student upload the document. The teacher upload document. The teacher can give feedback.

Reply: The document and feedback will store in database.

4.3.7 Level-1.5 Use Case Diagram-Update

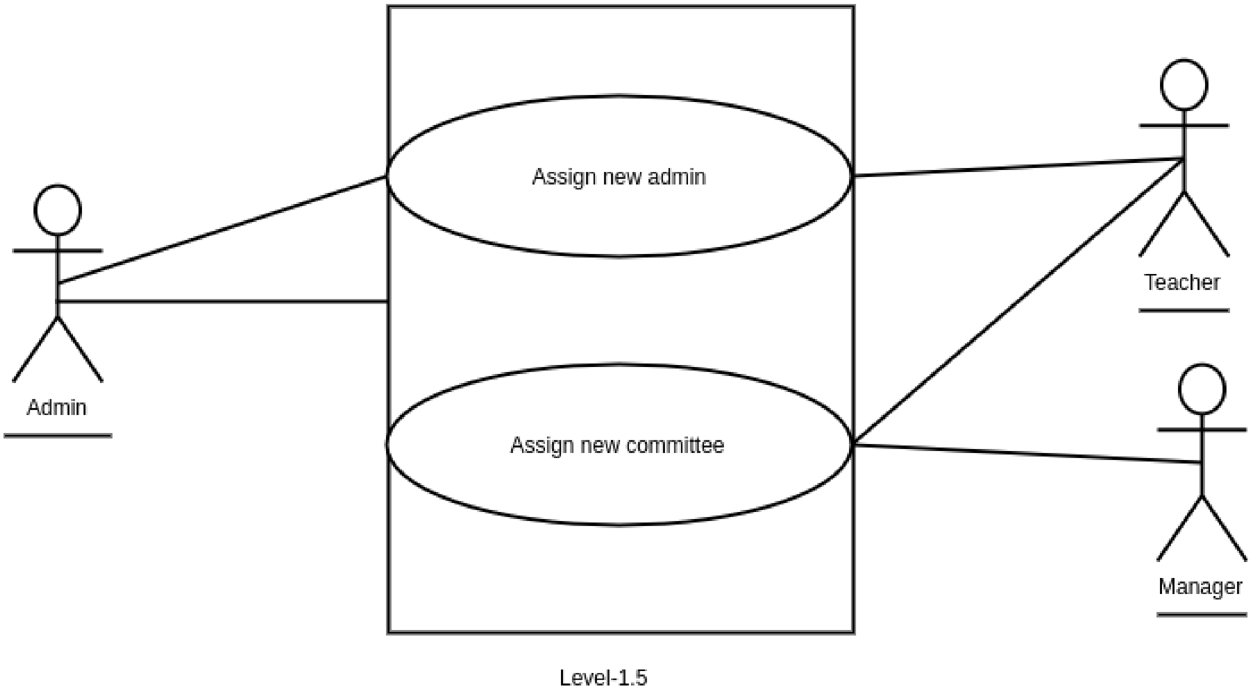


Figure 7: Level 1.5 Use Case Diagram of Update

Name:	Update
ID:	SPLMS-L-1.5
Primary Actors:	Admin, Teacher, Manager

Description of Use Case Diagram Level 1.5:

The admin can change the committee. She can change the admin also.

Action-Reply of Use Case Diagram Level 1.5:

Action: The admin will assign new committee members and new admin.

Reply: The new assigned committee member and admin will get a notification

4.4 ACTIVITY DIAGRAMS OF SPLMS

4.4.1 Level-1.1 Activity Diagram Authentication

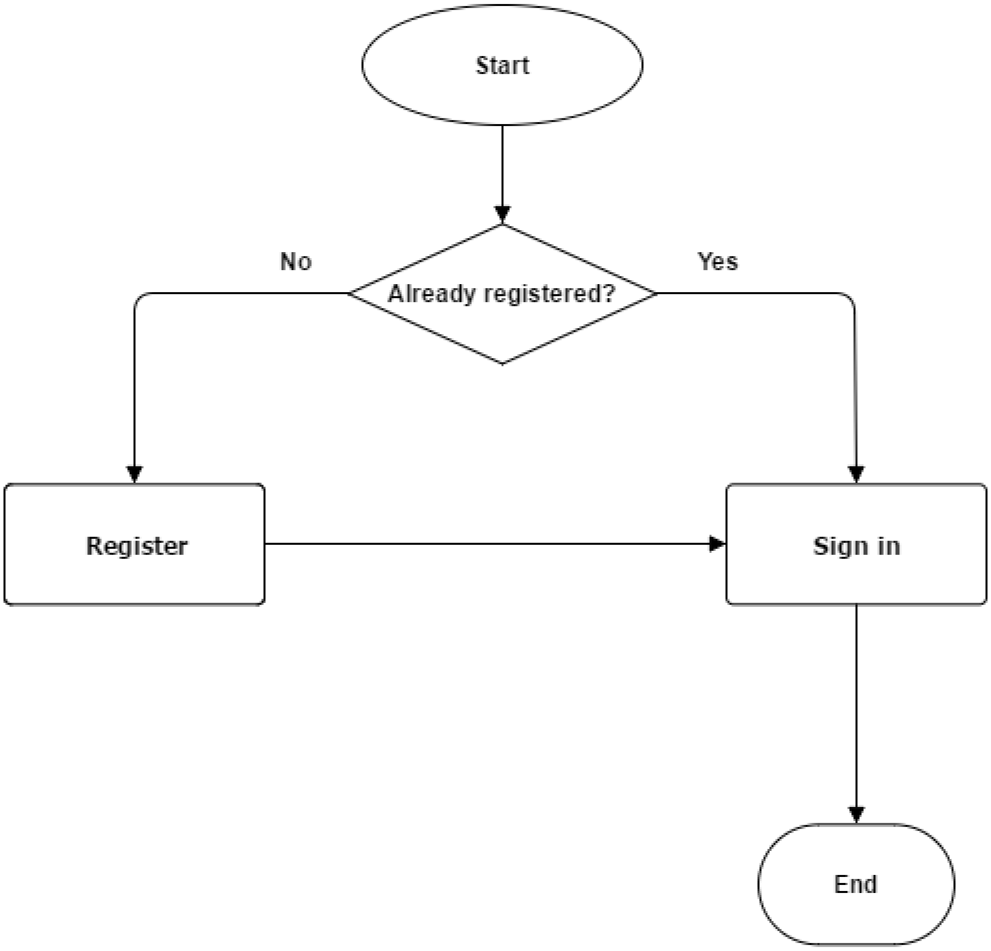


Figure 8: Activity Diagram 1.1: Authentication

4.4.2 Level-1.1.1 Activity Diagram Sign in

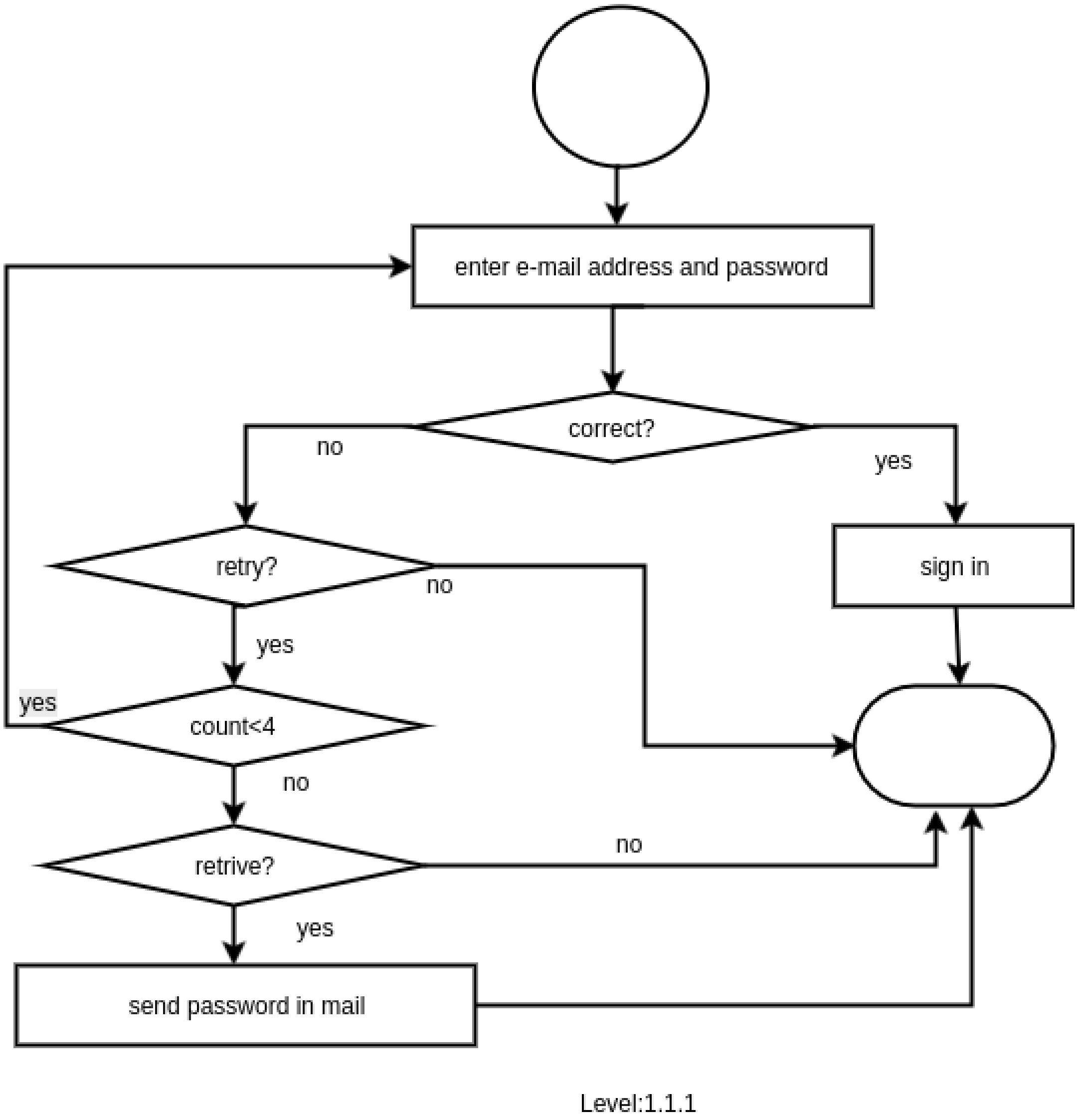
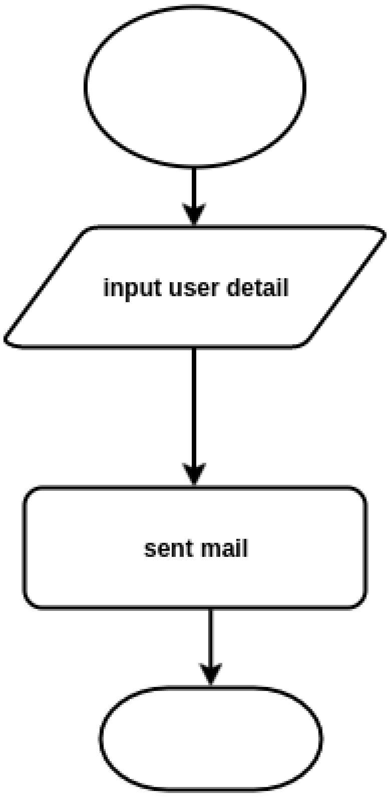


Figure 9: Activity Diagram 1.1.1: Sign in

4.4.3 Level-1.1.2 Activity Diagram Registration



level: 1.1.2

Figure 10: Activity Diagram 1.1.2: Registration

4.4.4 Level-1.2 Activity Diagram Initialization

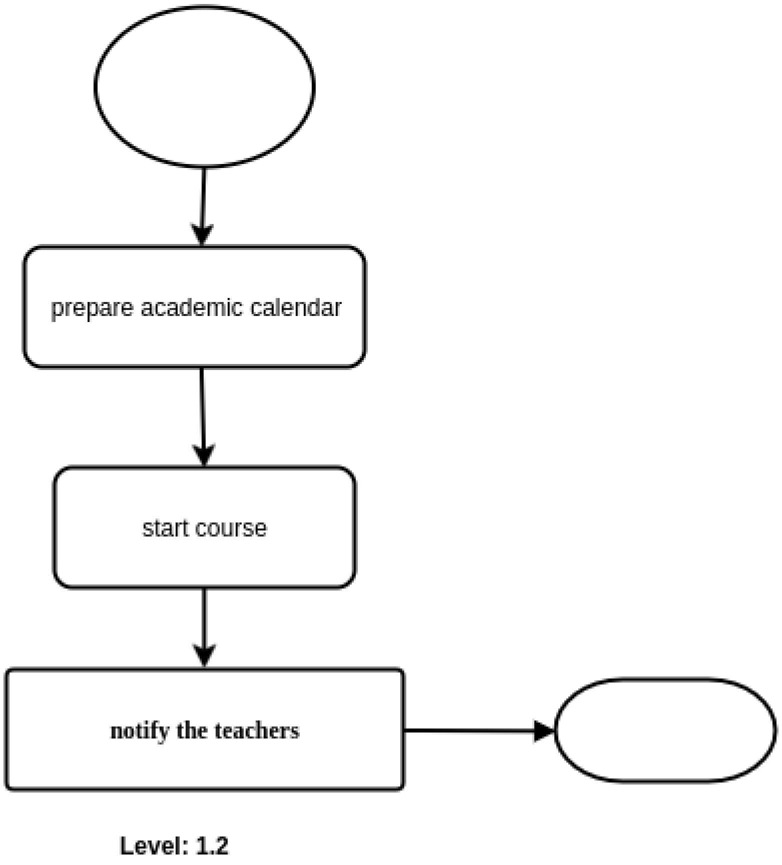
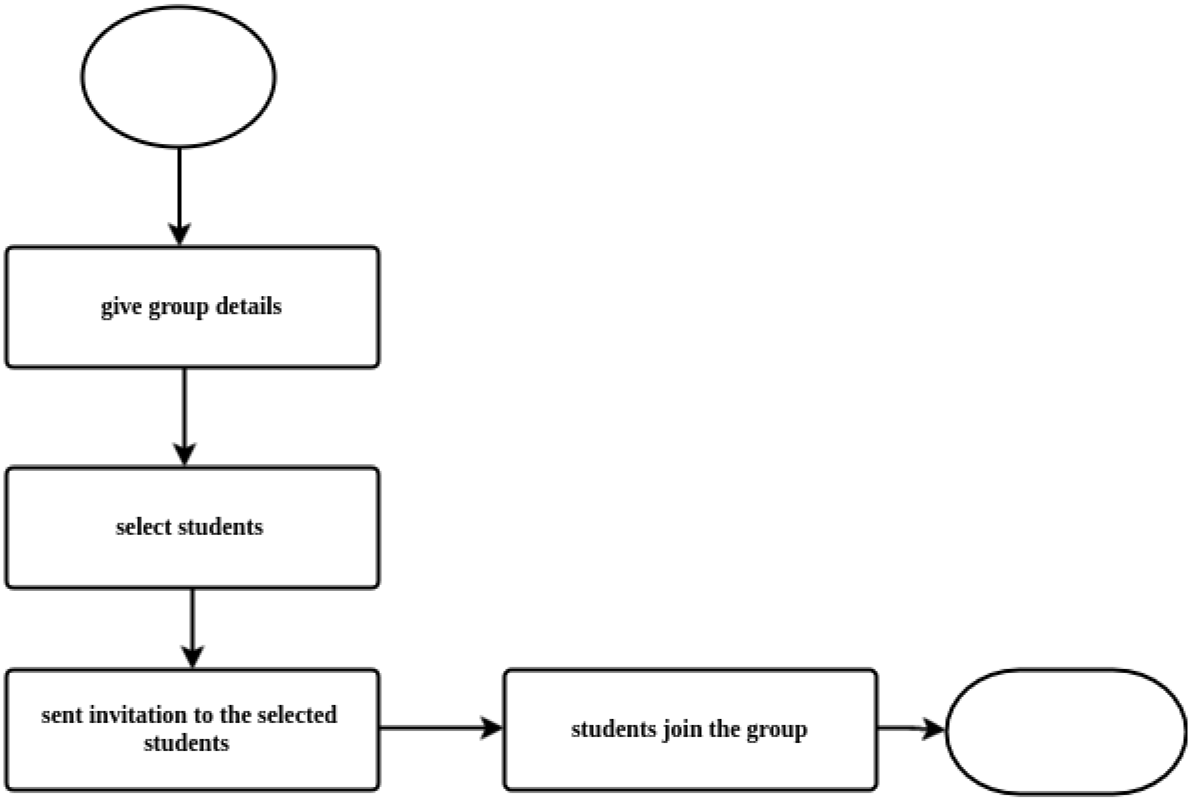


Figure 11: Activity Diagram 1.2: Initialization

4.4.5 Level-1.3 Activity Diagram Group Forming



level: 1.3

Figure 12: Activity Diagram 1.3: Group Forming

4.4.6 Level-1.4 Activity Diagram Feedback

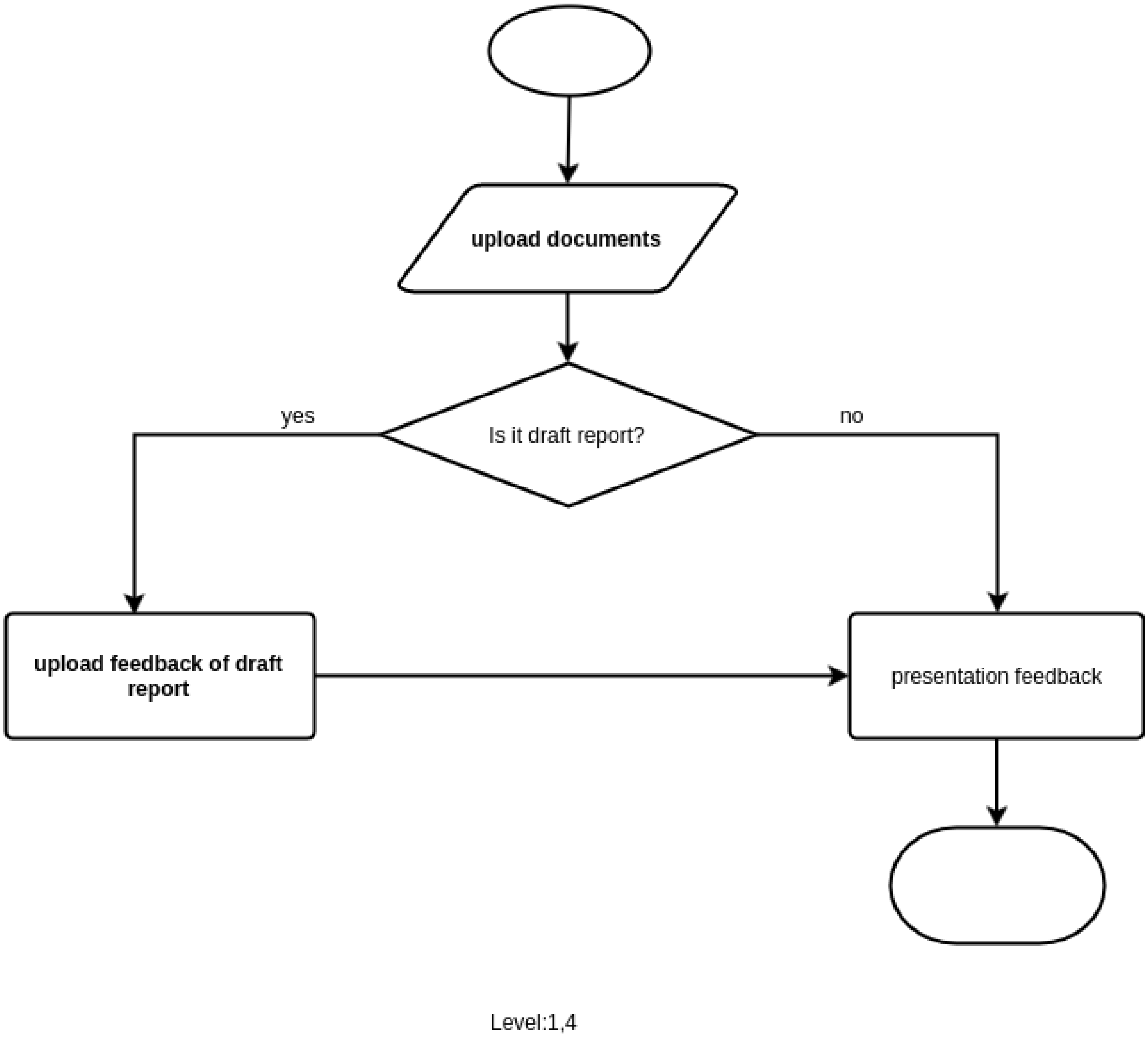


Figure 13: Activity Diagram 1.4: Feedback

4.4.7 Level-1.1 Activity Diagram Update

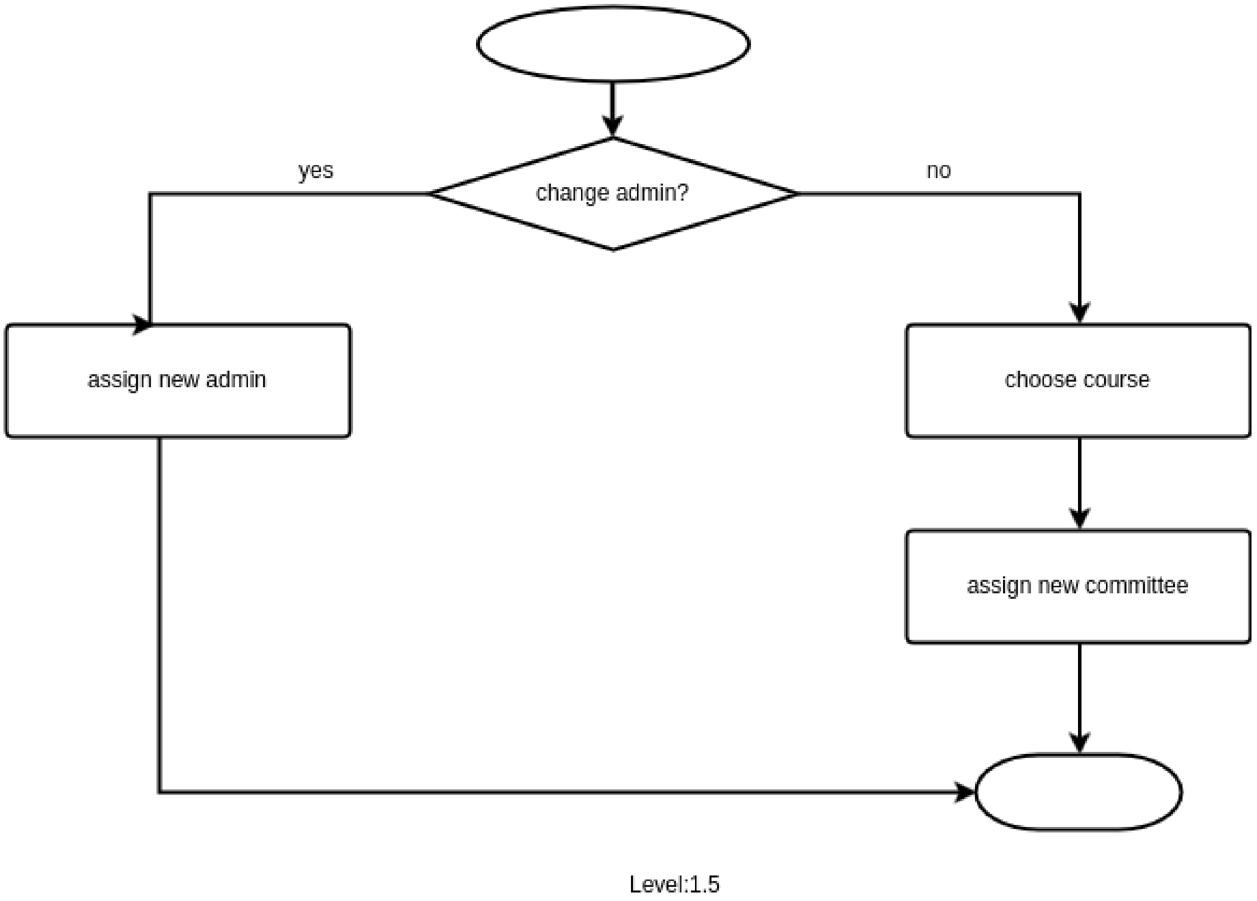


Figure 14: Activity Diagram 1.5: Update

4.5 Swimlane Diagrams of SPLMS

4.5.1 Level-1.1 Swimlane Diagram Authentication

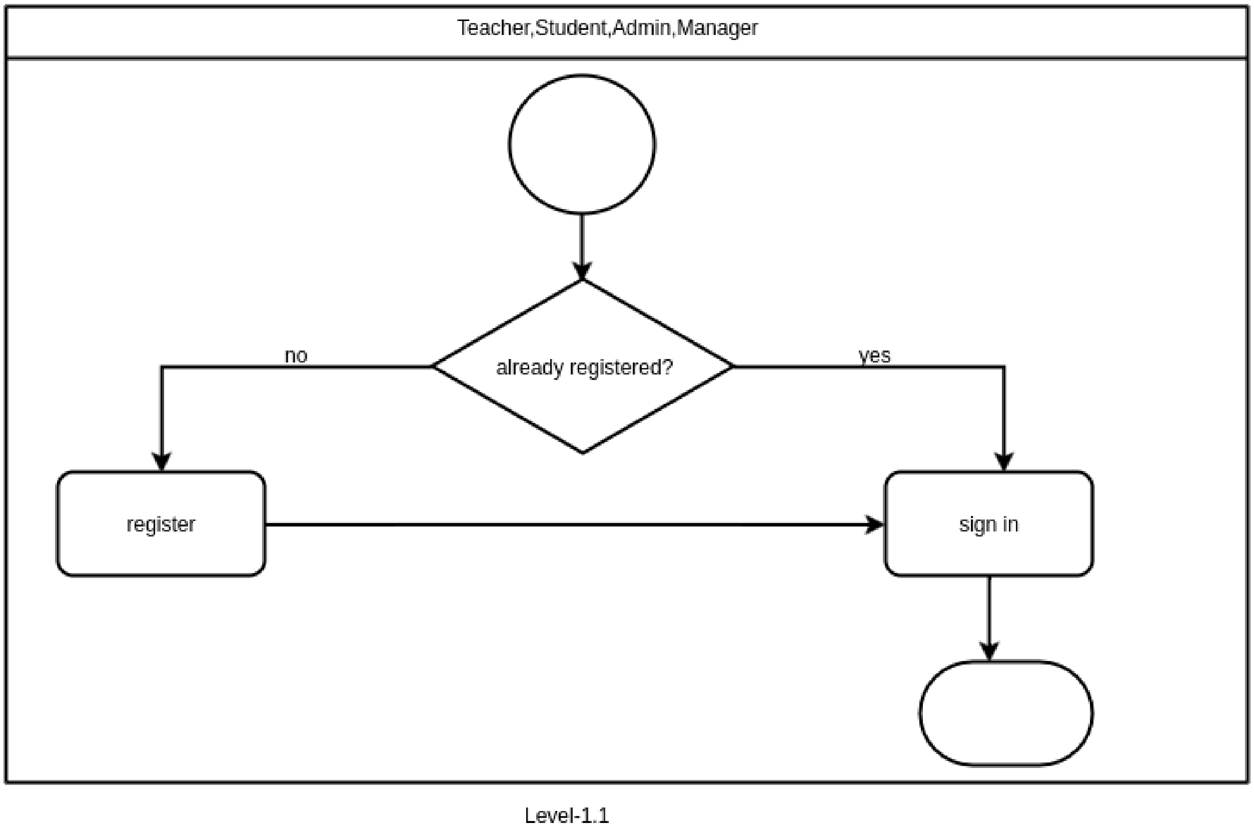


Figure 15: Swimlane Diagram 1.1: Authentication

4.5.2 Level-1.1.1 Swimlane Diagram Sign in

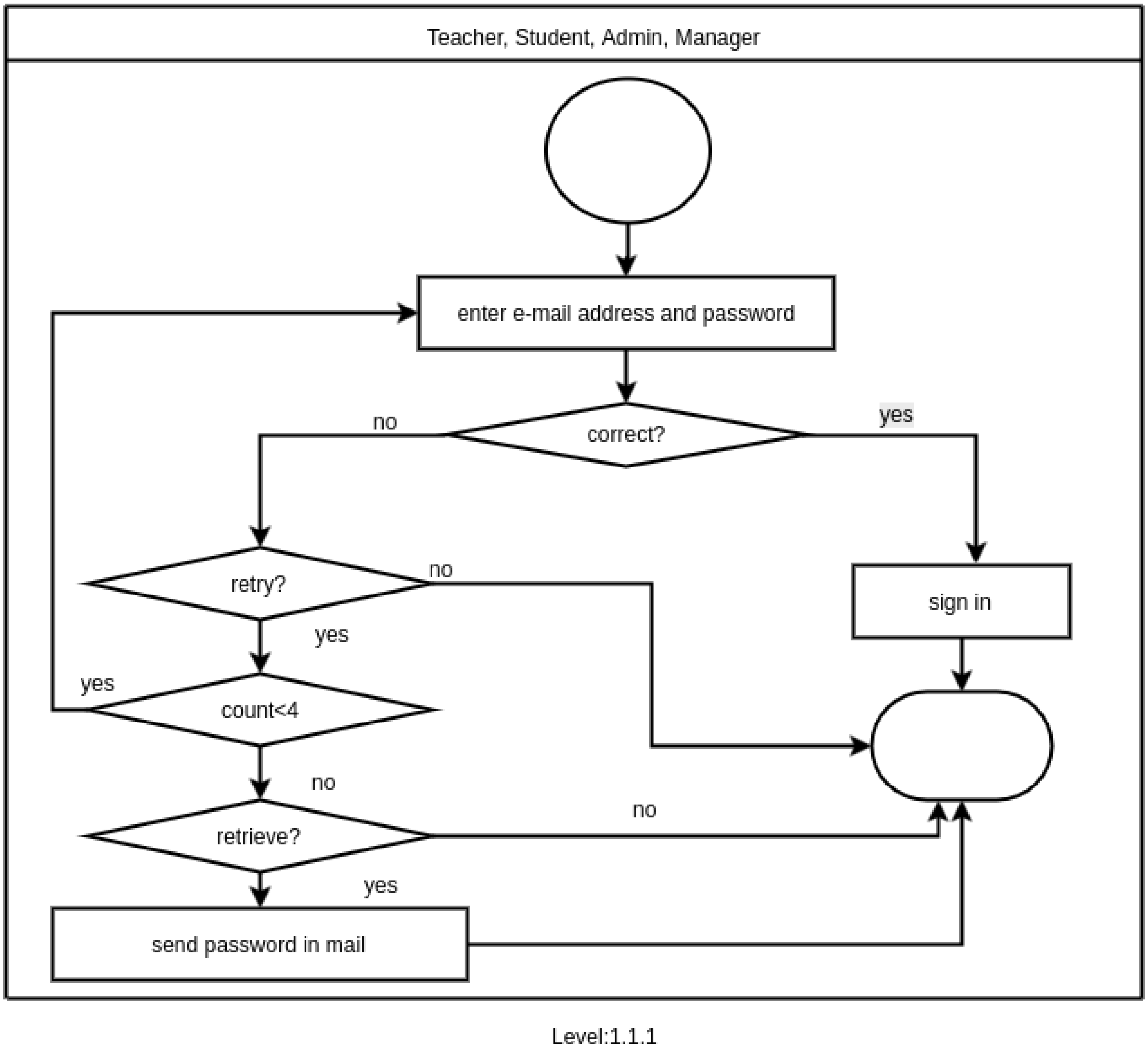


Figure 16: Swimlane Diagram 1.1.1: Sign in

4.5.3 Level-1.1.2 Swimlane Diagram Registration

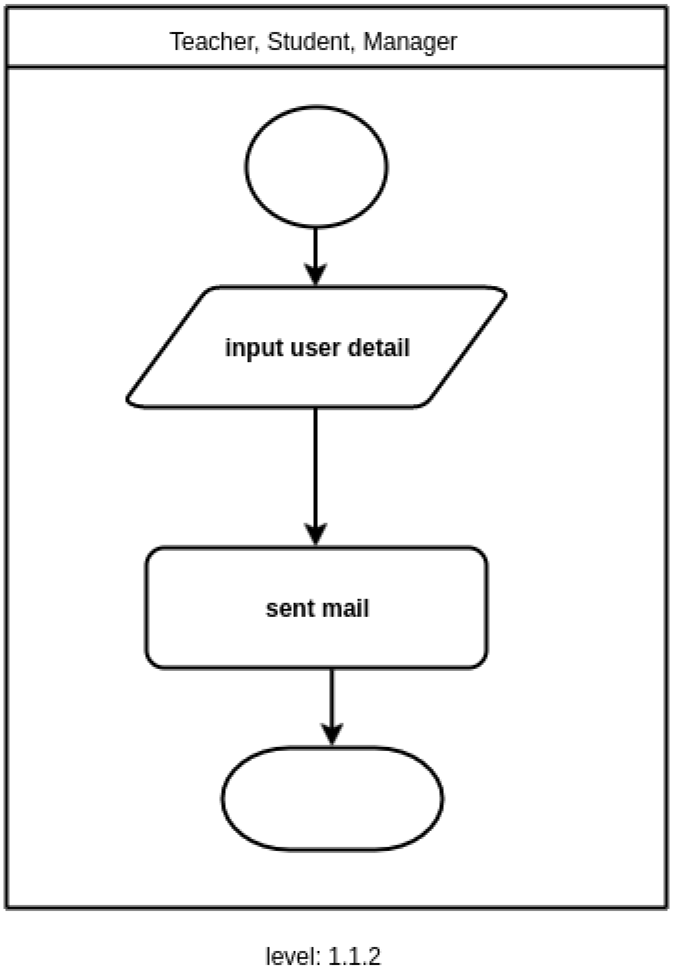


Figure 17: Swimlane Diagram 1.1.2: Registration

4.5.4 Level-1.2 Swimlane Diagram Initialization

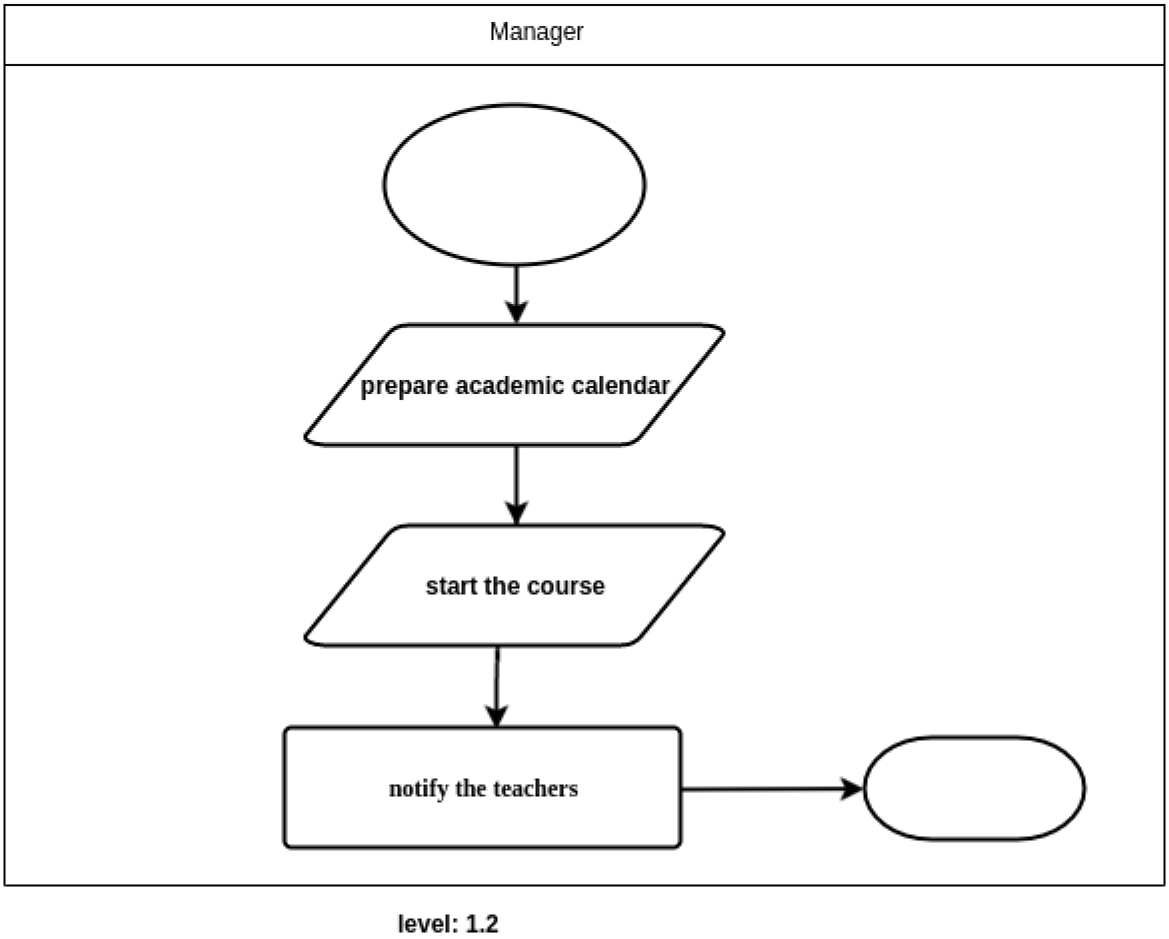
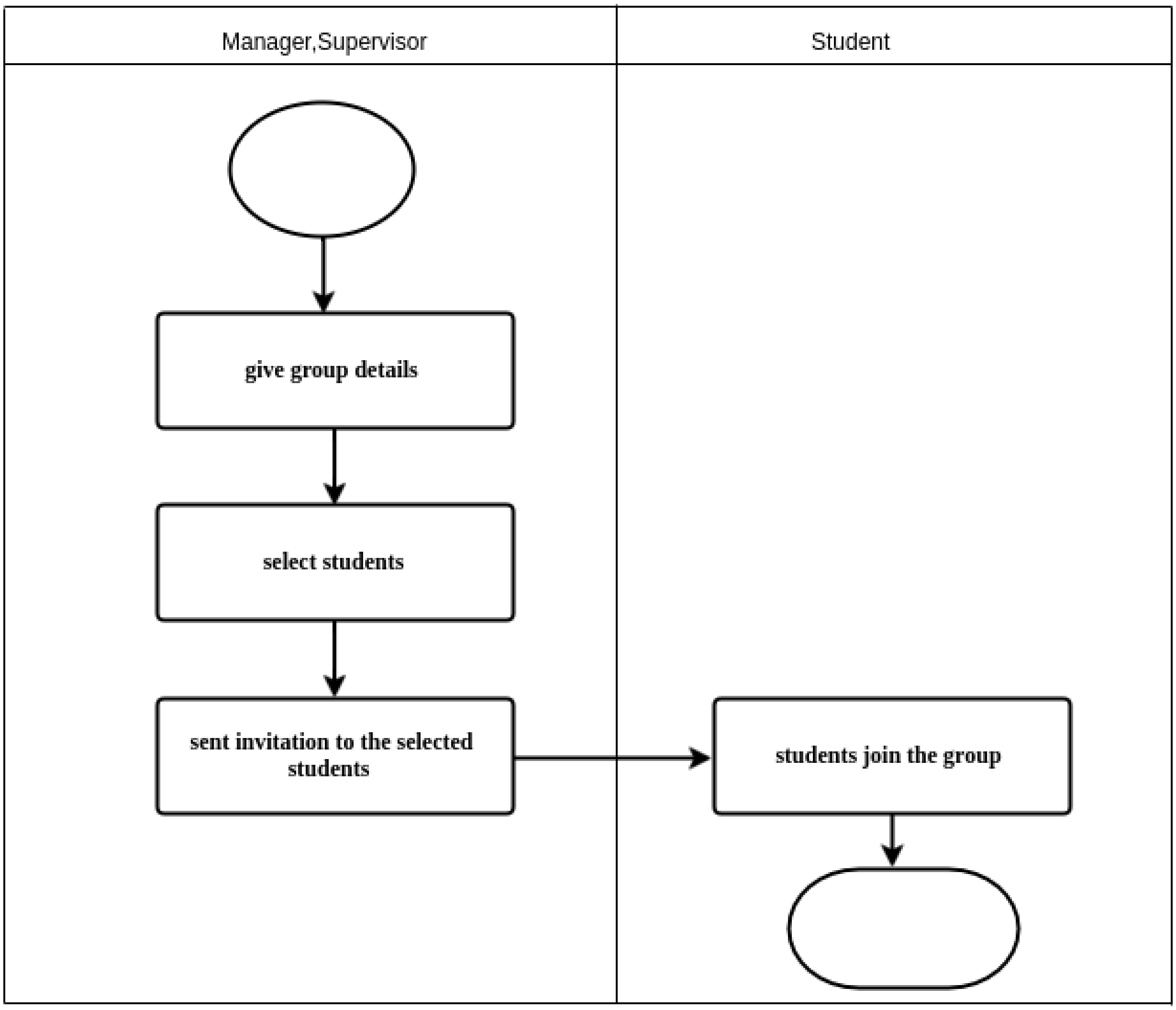


Figure 18: Swimlane Diagram 1.2: Initialization

4.5.5 Level-1.3 Swimlane Diagram Group Forming



level: 1.3

Figure 19: Swimlane Diagram 1.3: Group Forming

4.5.6 Level-1.4 Swimlane Diagram Feedback

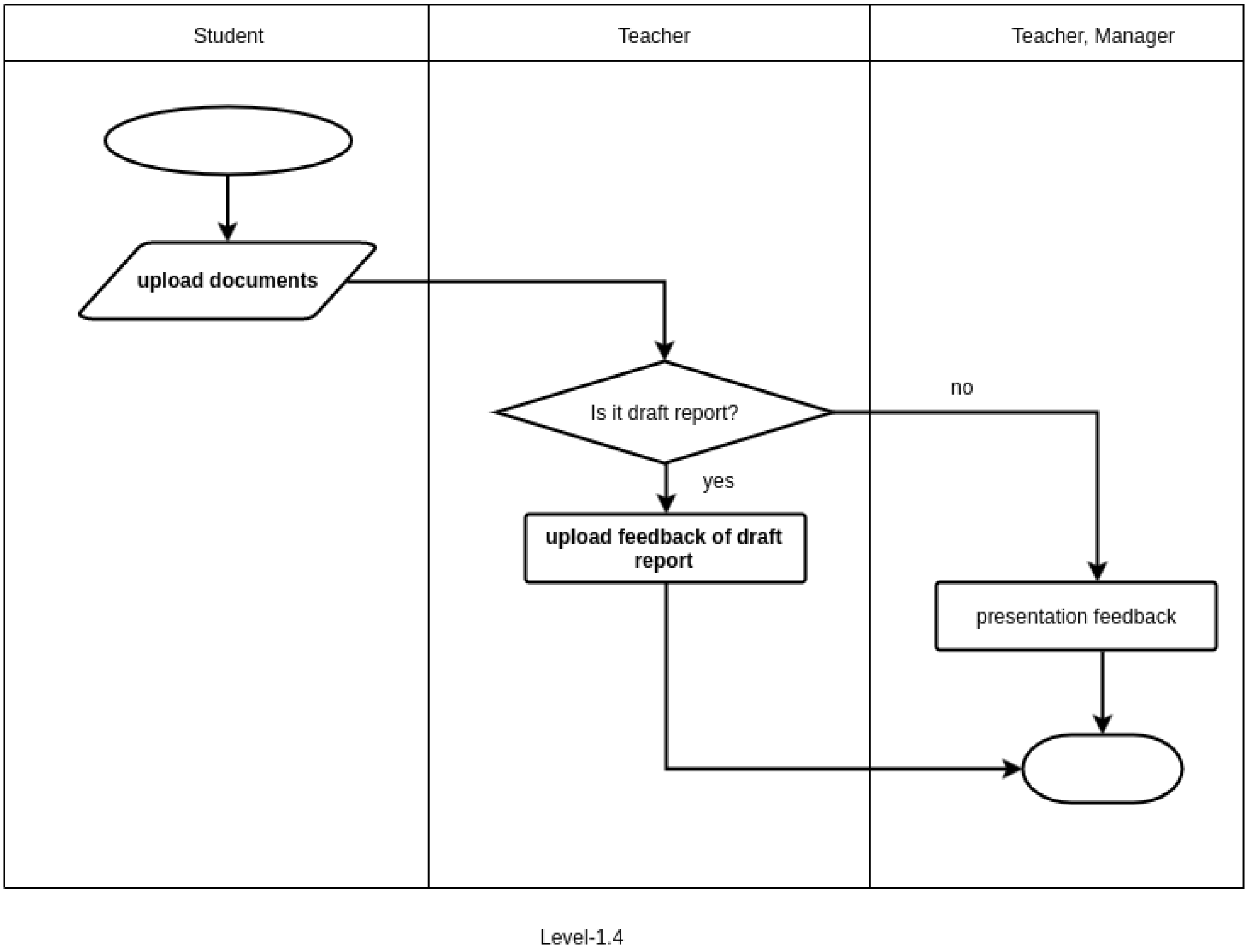
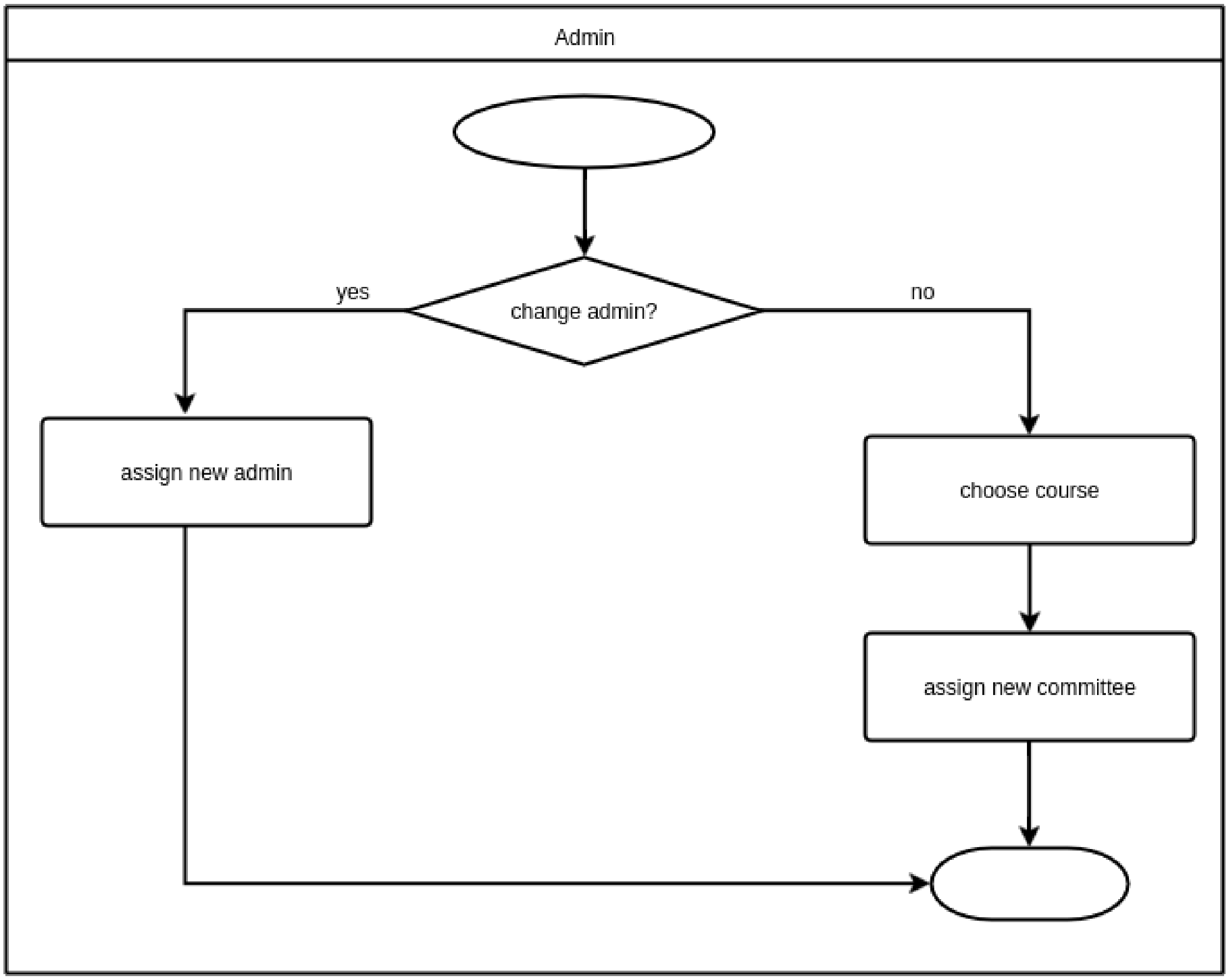


Figure 20: Swimlane Diagram 1.4: Feedback

4.5.7 Level-1.5 Swimlane Diagram Update



Level:1.5

Figure 21: Swimlane Diagram 1.5: Update

CHAPTER 5: DATA MODELING OF SPLMS

5.1 DATA MODELING CONCEPT

If software requirements include the necessity to create, extend or interact with a database or complex data structures need to be constructed and manipulated, then the software team chooses to create data models as part of overall requirements modeling. The entity-relationship diagram (ERD) defines all data objects that are processed within the system, the relationships between the data objects and the information about how the data objects are entered, stored, transformed and produced within the system.

5.2 DATA OBJECTS

A data object is a representation of composite information that must be understood by the software. Here, composite information means an information that has a number of different properties or attributes. A data object can be an external entity, a thing, an occurrence, a role, an organizational unit, a place or a structure.

5.2.1 Noun identification

We identified all the nouns whether they are in problem space or in solution space from our usage scenario.

Table 1: Noun Identification for Data Modeling

NO	Noun	Problem Domain/ Solution Space	Attributes
1	Software Project Lab Management System	P	
2	The committee	S	37
3	Admin	S	6-9
4	Student	S	6-11
5	Teacher	S	6-9,38
6	Name	S	
7	E-mail address	S	
8	Password	S	
9	Course role	S	
10	Roll no	S	
11	Batch	S	
12	Group	S	13-19,26-28
13	Group name	S	
14	Project topic	S	
15	Group members	S	
16	Supervisor	S	
17	Feedback	S	
18	Members mark	S	
19	Members grade	S	
20	Documents	S	21-25
21	Proposal	S	
22	SRS report	S	
23	Draft report	S	
24	Feedback of draft report	S	
25	Final report	S	
26	Presentation time	S	
27	Examiner's list	S	
28	Resubmission date	S	
29	Managers	S	
30	Members	S	
31	Academic calendar	S	32-36
32	Proposal/1st midterm date	S	
33	2nd midterm date	S	

34	Draft report submission date	S	
35	Final report submission date	S	
36	Final presentation date	S	
37	Committee role	S	
38	Number of groups	S	
39	Database	S	
40	Registration	S	
41	Authentication	S	
42	GitHub	P	
43	Notification	S	
44	Mail	S	45
45	Link	S	

5.2.2 Potential Data Objects

- ✓ Committee: 37
- ✓ Admin:6-9
- ✓ Student : 6-11
- ✓ Teacher : 6-9,38
- ✓ Group : 13-19,26-28
- ✓ Documents: 21-25
- ✓ Academic calendar: 32-36
- ✓ Mail : 45

5.2.3 Analysis for finalizing Data Objects:

- The attributes of Admin, Student and Teacher are almost same. So there common attributes will be stored as User data object.
- To store the extra attributes of student and teacher there will be data object student and teacher.
- As the admin have no extra attributes so it will be referred as user.
- In committee data object there will be year, course code to identify the committee.
- To store the documents there will be four different data object..
- To identify the examiners of presentation we will have a data object presentation

5.2.4 Final Data Objects

Table 2: Final Data Objects

NO	Name	Attributes
1	User	User id, name, e-mail address, password, course role
2	Student	User id, roll no, group id
3	Teacher	User id, number of groups
4	Committee	Committee id, Year, course code, teacher user id, committee role
5	Groups	Group id, Group name, year, project topic, teacher user id
6	Academic calendar	Year, Course code, Proposal date, 2nd midterm date, draft report submission date, Final report submission date, Final presentation date
7	Resubmission	Resubmission id, Group id, teacher user id, state, claim, resubmission date
8	Presentation	teacher user id, group id, presentation time
9	Document	Document id, group id, document type, user id
10	Notification	Notification id, sender id, receiver id, notification

5.3 Data Object Relations

Data objects are connected to one another in different ways.

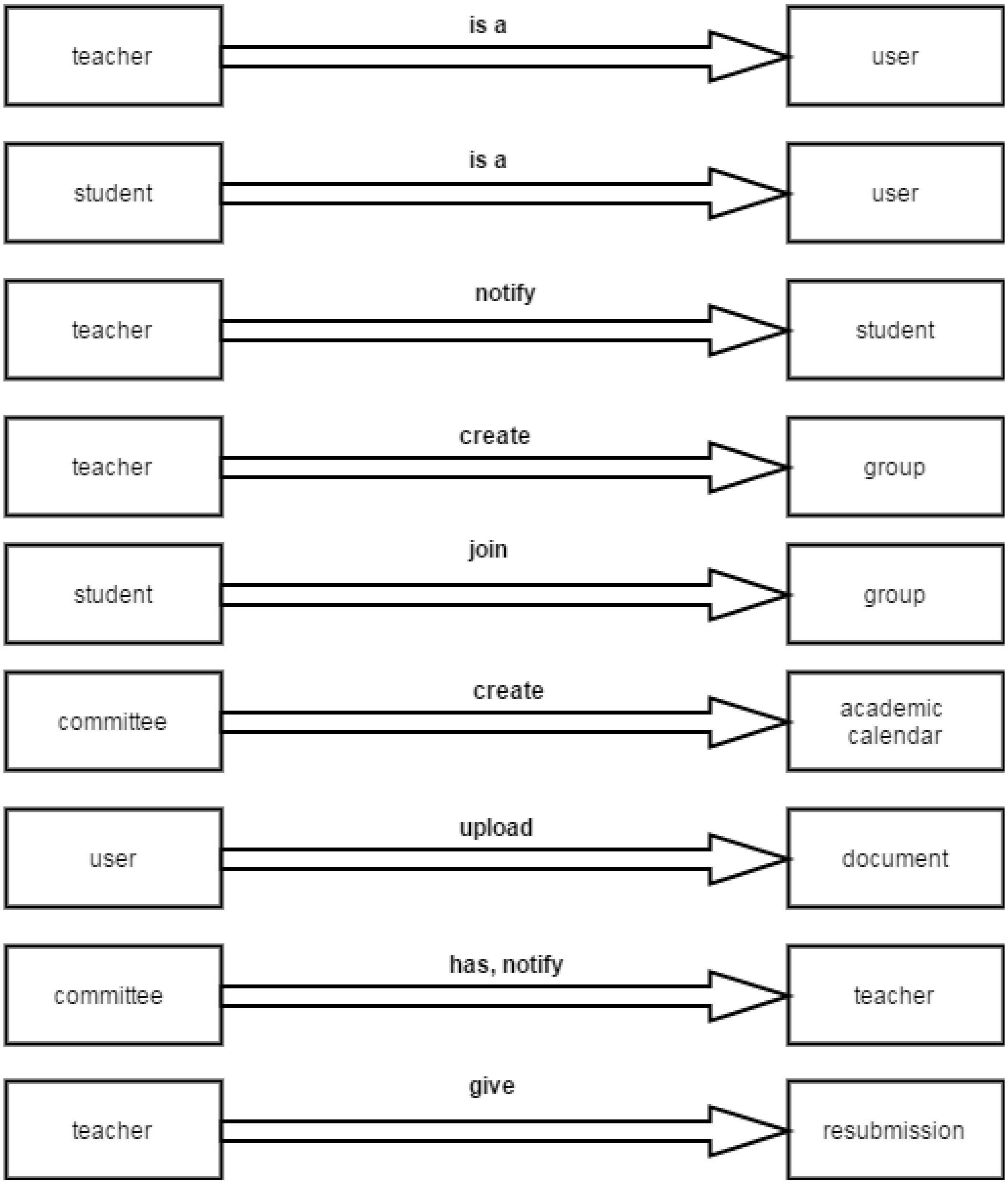


Figure 22: Relationships between Data Objects

5.3 ENTITY RELATIONSHIP DIAGRAM

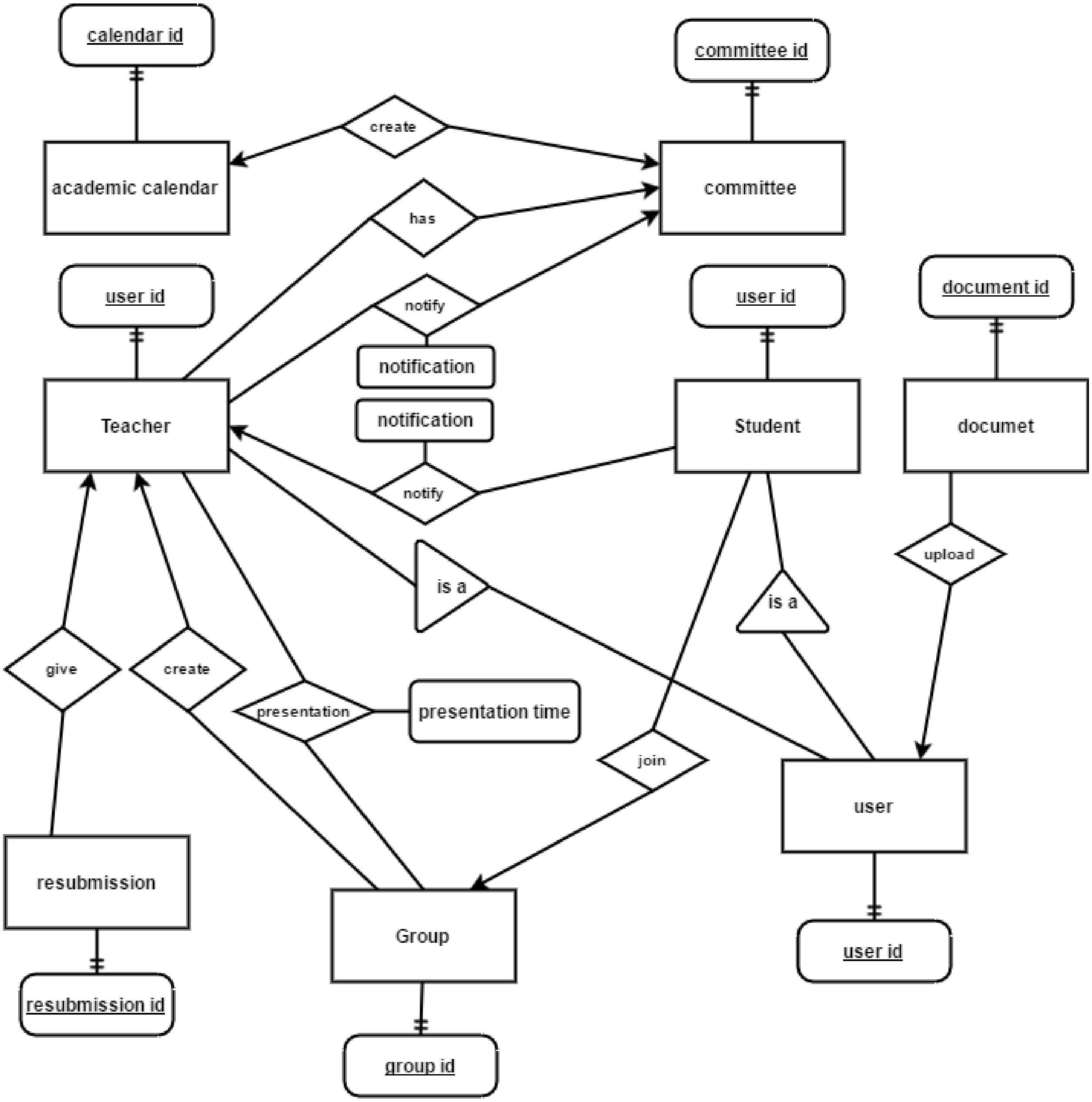


Figure 23: ER Diagram of SPLMS

5.4 SCHEMA DIAGRAMS

Table 3: Schema for User

User		
Attributes	Type	Size
User Id	NUMBER	5
Name	VARCHAR	20
E-mail address	VARCHAR	20
Password	VARCHAR	15
Course role	VARCHAR	15

Table 4: Schema for Student

Student		
Attributes	Type	Size
User Id	NUMBER	5
Roll no	NUMBER	4
Batch	NUMBER	3
Group id	NUMBER	5

Table 5: Schema for Teacher

Teacher		
Attributes	Type	Size
User Id	NUMBER	5
Number of groups	NUMBER	2

Table 6: Schema for Committee

Committee		
Attributes	Type	Size
Committee id	NUMBER	5
Teacher Id	NUMBER	5
Year	NUMBER	5
Course code	NUMBER	5
Committee role	VARCHAR	10

Table 7: Schema for Group

Group		
Attributes	Type	Size
Group id	NUMBER	5
Group name	VARCHAR	15
Project topic	VARCHAR	30
Year	NUMBER	5
Teacher Id	NUMBER	5

Table 8: Schema for Document

Document		
Attributes	Type	Size
Document id	NUMBER	5
Group id	NUMBER	5
File type	VARCHAR	20
User id	NUMBER	5

Table 9: Schema for Notification

Notification		
Attributes	Type	Size
Notification id	NUMBER	5
Sender id	NUMBER	5
Receiver id	NUMBER	5
Notification	VARCHAR	100

Table 10: Schema for Presentation

Presentation		
Attributes	Type	Size
User Id	NUMBER	5
Group name	VARCHAR	15
Presentation time	TIME	

Table 11: Schema for Resubmission

Resubmission		
Attributes	Type	Size
Year	NUMBER	5
Teacher id	NUMBER	5
Group id	NUMBER	5
State	VARCHAR	15
Claim	VARCHAR	100
Resubmission date	DATE	

Table 12: Schema for Academic Calendar

Academic calendar		
Attributes	Type	Size
Calendar id	NUMBER	5
Year	NUMBER	5
Course code	VARCHAR	10
Proposal date	DATE	
Midterm date	DATE	
Draft report submission date	DATE	
Final report submission date	DATE	
Final presentation date	DATE	

CHAPTER 6: CLASS BASED MODELING OF SPLMS

This Chapter is intended to describe class based modeling of Dhaka University Club Management System.

6.1 CLASS BASED MODELING CONCEPT

Class-based modeling represents the objects that the system will manipulate, the operations that will applied to the objects, relationships between the objects and the collaborations that occur between the classes that are defined.

6.2 GENERAL CLASSIFICATIONS

To identify the potential classes we have first selected the nouns from the solution space of the story. These were then characterized in seven general classifications. The seven general characteristics are as follows:

1. External entities
2. Things
3. Events
4. Roles
5. Organizational units
6. Places
7. Structures

Following are the specifications of the nouns according to the general classifications:

Table 13: General Classifications of Nouns

NO	Noun	Problem Domain/ Solution Space	Generale classification
1	The committee	S	5
2	Admin	S	4,5
3	Student	S	4,5
4	Teacher	S	4,5
5	Name	S	
6	E-mail address	S	
7	Password	S	
8	Course role	S	
9	Roll no	S	
10	Batch	S	
11	Group	S	5
12	Group name	S	
13	Project topic	S	
14	Group members	S	
15	Supervisor	S	4,5
16	Feedback	S	3
17	Documents	S	2
18	Proposal	S	2
19	SRS report	S	2
20	Draft report	S	2
21	Feedback of draft report	S	2
22	Final report	S	2
23	Presentation time	S	

24	Examiner's list	S	
25	Resubmission date	S	
26	Managers	S	4
27	Members	S	4
28	Academic calendar	S	2
29	Proposal/1st midterm date	S	
30	2nd midterm date	S	
31	Draft report submission date	S	
32	Final report submission date	S	
33	Final presentation date	S	
34	Committee role	S	
35	Number of groups	S	
36	Database	S	2
37	Registration	S	3
38	Notification	S	3
39	Authentication	S	3

6.3 Selection Criteria

The potential classes were then selected as classes by six Selection Criteria. A potential class becomes a class when it fulfills all six characteristics.

1. Retained Information

2. Needed Services

3. Multiple Attributes

4. Common attributes

5. Common operations

6. Essential requirements

Table 14: Selection Criteria of Potential Classes

NO	Noun	Problem Domain/ Solution Space	Selection Scriteria	Accepted / rejected
1	The committee	S		×
2	Admin	S	2,3,4	√
3	Student	S	1,2,3,4,5	√
4	Teacher	S	2,3,4,5	√
5	Group	S	1,3,5	√
6	Supervisor	S	2,3,4,5	√
7	Feedback	S	1	×
8	Documents	S	3,4,6	√
9	Proposal	S	3,4,6	√
10	SRS report	S	3,4,6	√
11	Draft report	S	3,4,6	√
12	Feedback of draft report	S	3,4,6	√
13	Final report	S	3,4,6	√
14	Managers	S	2,3,4,5	√
15	Members	S	2,3,4,5	√
16	Academic calendar	S	1,3	×
17	Database	S	1,2,6	√
18	Registration	S	2	×
19	Notification	S	1	×
20	Authentication	S	2	×

6.4 Associate Noun and Verb Identification

We now identify the nouns and verbs associated with the potential classes to better find out the attributes and methods of each class.

Table 15: Associate Noun and Verb Identification

NO	Potential Class	Nouns	Verbs
1	Teacher	1. name 2. e-mail address 3. password 4. course role	1. create group 2. notify students 3. feedback 4. see previous projects
2	Student	1. name 2. e-mail address 3. password 4. roll no 5. batch 6. course role	1. join group
3	Admin	1. name 2. e-mail address 3. password 4. course role	1. assign new committee 2. store project in gitHub 3. assign new admin
4	Manager	1. name 2. e-mail address 3. password	1. start course 2. create calendar 3. notify teachers 4. edit calendar 5. upload documents 6. edit group formation rules
5	Members	1. name 2. e-mail address 3. password	1. create group 2. notify students 3. feedback 4. see previous projects
6	Supervisor	1. name 2. e-mail 3. address 4. password 5. course role	1.feedback 2.see previous projects
7	Documents	1. name 2. size 3. extension	1. upload documents

8	Database	1. database name 2. password 3. database user	1. updates information 2. gets information
9	Groups	1. name 2. project topic	1. Store group details in database
10	Proposal	1. name 2. size 3. extension	1. upload documents
11	SRS report	1. name 2. size 3. extension	1. upload documents
12	Draft report	1. name 2. size 3. extension	1. upload documents
13	Feedback of draft report	1. name 2. size 3. extension	1. upload documents
14	Final report	1. name 2. size 3. extension	1. upload documents

6.5 Attribute and Method Selection

After identifying the classes, we have specified their attributes and methods.

Table 16: Attribute and Method Selection of Classes

No	Class	Attribute	Method
1	Teacher	1. name 2. emailAddress 3. password 4. courseRole	1. createGroup() 2. notifyStudents() 3. feedback() 4. seePreviousProjects() 5. uploadDocuments() 6. getNotification()
2	Student	1. name 2. emailAddress 3. password 4. roll no 5. batch 6. courseRole	1. joinGroup() 2. uploadDocuments() 3. getNotification()
3	Admin	1. name 2. emailAddress 3. password 4. courseRole	1. assignNewCommittee() 2. storeDocumentsInGitHub() 3. assignNewAdmin() 4. chooseOption()
4	Manager	1. name 2. emailAddress 3. password	1. startCourse() 2. createCalendar() 3. notifyTeachers() 4. editCalendar() 5. uploadDocuments() 6. editGroupFormationRules()
5	Member	1. name 2. emailAddress 3. password	1. createGroup() 2. notifyStudents() 3. feedback() 4. seePreviousProjects() 5. uploadDocuments()
6	Supervisor	1. name 2. email 3. address 4. password 5. courseRole	1. feedback() 2. seePreviousProjects() 3. uploadDocuments()

7	Document	1. name 2. size 3. extension	1. storeDocument() 2. checkDocuments() 3. storeToDatabase()
8	Database	1. databaseName 2. password 3. databaseUser	1. createConnection() 2. updateInformation() 3. getInformation() 4. insertInformation() 5. closeConnection()
9	Group	1. name 2. project topic	1. joinGroup() 2. uploadDocuments()
10	Proposal	1. name 2. size 3. extension	1. storeDocument() 2. checkDocuments() 3. storeToDatabase()
11	SRS report	1. name 2. size 3. extension	1. storeDocument() 2. checkDocuments() 3. storeToDatabase()
12	Draft report	1. name 2. size 3. extension	1. storeDocument() 2. checkDocuments() 3. storeToDatabase()
13	Feedback of draft report	1. name 2. size 3. extension	1. storeDocument() 2. checkDocuments() 3. storeToDatabase()
14	Final report	1. name 2. size 3. extension	1. storeDocument() 2. checkDocuments() 3. storeToDatabase()

6. 6 FINALIZING CLASSES

To identify the final classes we need to check if there can be any hierarchies or merges. These identifications are given below:

1.'Teacher' have common attributes and methods with 'supervisor' and 'Member' so we merge them as 'Teacher'.
2. There will be an abstract class named 'user' where the common operations(inputLoginCredential(), logout(),register())will occur.
3.'Document' have common attributes and methods with 'final report', 'feedback of draft report', 'draft report', 'SRS report', 'proposal'. So we merge them as 'Document'.
4.The manager have some extra operations than the teacher. So the manager class will extends teacher class.

So the final classes are-

Table 17: Final Classes with Methods and Attributes

No	Class	Attribute	Method
1	User	1. name 2. emailAddress 3. password 4. courseRole	1. inputLoginCredentials() 2. logout() 3. register()
2	Teacher extends User	1. numberOfGroups	1. createGroup() 2. notifyStudents() 3. feedback() 4. seePreviousProjects() 5. uploadDocuments() 6. getNotification()
3	Student extends User	1. roll no 2. batch	1. joinGroup() 2. uploadDocuments() 3. getNotification()
4	Admin extends User		1. assignNewCommittee() 2. storeDocumentsGitHub() 3. assignNewAdmin() 4. chooseOption()
5	Managers extends teacher		1. startCourse() 2. createCalendar() 3. notifyTeachers() 4. editCalendar() 5. uploadDocuments() 6. editGroupFormationRules()

6	Document	1. name 2. size 3. extension	1. storeDocument() 2. checkDocuments() 3. storeToDatabase()
7	Database	1. databaseName 2. password 3. databaseUser	1. createConnection() 2. updateInformation() 3. getInformation() 4. insertInformation() 5. closeConnection()
8	Group	1. name 2. project topic	1. joinGroup() 2. uploadDocuments()

6.7 CLASS CARDS

After identifying our final classes we have generated the following class cards.

Table 18: Class Card of User

User	
Attributes	**Methods**

name
emailAddress
password
courseRole
	inputLoginCredintials()
	logout()
	register()
	sentMail()

Responsibilities	**Collaborative Class**
Registration	Database
Authentication	Database, Admin, Teacher, Student

Table 19: Class Card of Document

Document	
Attributes	**Methods**

name
size
extension
	storeDocument()
	checkDocuments()
	storeToDatabase()

Responsibilities	**Collaborative Class**
Store document	Database

Table 20: Class Card of Admin

Admin	
Attributes	**Methods**

name
emailAddress
password
course role
	assignNewCommittee()
	uploadDocumentsInGitHub()
	assignNewAdmin()
	chooseOption()

Responsibilities	**Collaborative Class**
Assign new committee	Teacher, manager
Assign new admin	Teacher

Table 21: Class Card of Teacher

Teacher	
Attributes	**Methods**

name
email address
password
course role
	createGroups()
	notifyStudents()
	feedback()
	uploadDocuments()
	seePreviousProjects()
	getNotification()

Responsibilities	**Collaborative Class**
Create group	Group
Giving feedback	Database
Upload documents	Documents

Table 22: Class Card of Student

Student	
Attributes	**Methods**

name
emailAddress
password
course role
roll no
batch
	joinGroups()
	uploadDocuments()
	getNotification()

Responsibilities	**Collaborative Class**
Joining group	Group
Upload documents	Documents

Table 23: Class Card of Manager

Manager	
Attributes	**Methods**

name
emailAddress
password
Course role
	createGroups()
	notifyStudentds()
	feedback()
	seePreviousProjects()
	uploadDocuments()
	startCourse()
	createCalendar()
	notifyTeachers()
	editCalendar()
	editGroupFromationRules()

Responsibilities	**Collaborative Class**
Create course	-
Upload documents	Documents

Table 24: Class Card of Database

Database	
Attributes	**Methods**

DatabaseName
DatabasePassword
DatabaseUser
	createConnection()
	updateInformation()
	getInformation()
	insertInformation()
	closeConnection()

Responsibilities	**Collaborative Class**
Update information	User, Admin, Teacher, Student, Manager, Document
Give information	User, Teacher, Student

Table 25: Class Card of Group

Group	
Attributes	**Methods**

name
projectTopic
	JoinGroup()
	uploadDocuments()

Responsibilities	**Collaborative Class**
Storing group detail	Database

6.8 CRC DIAGRAM

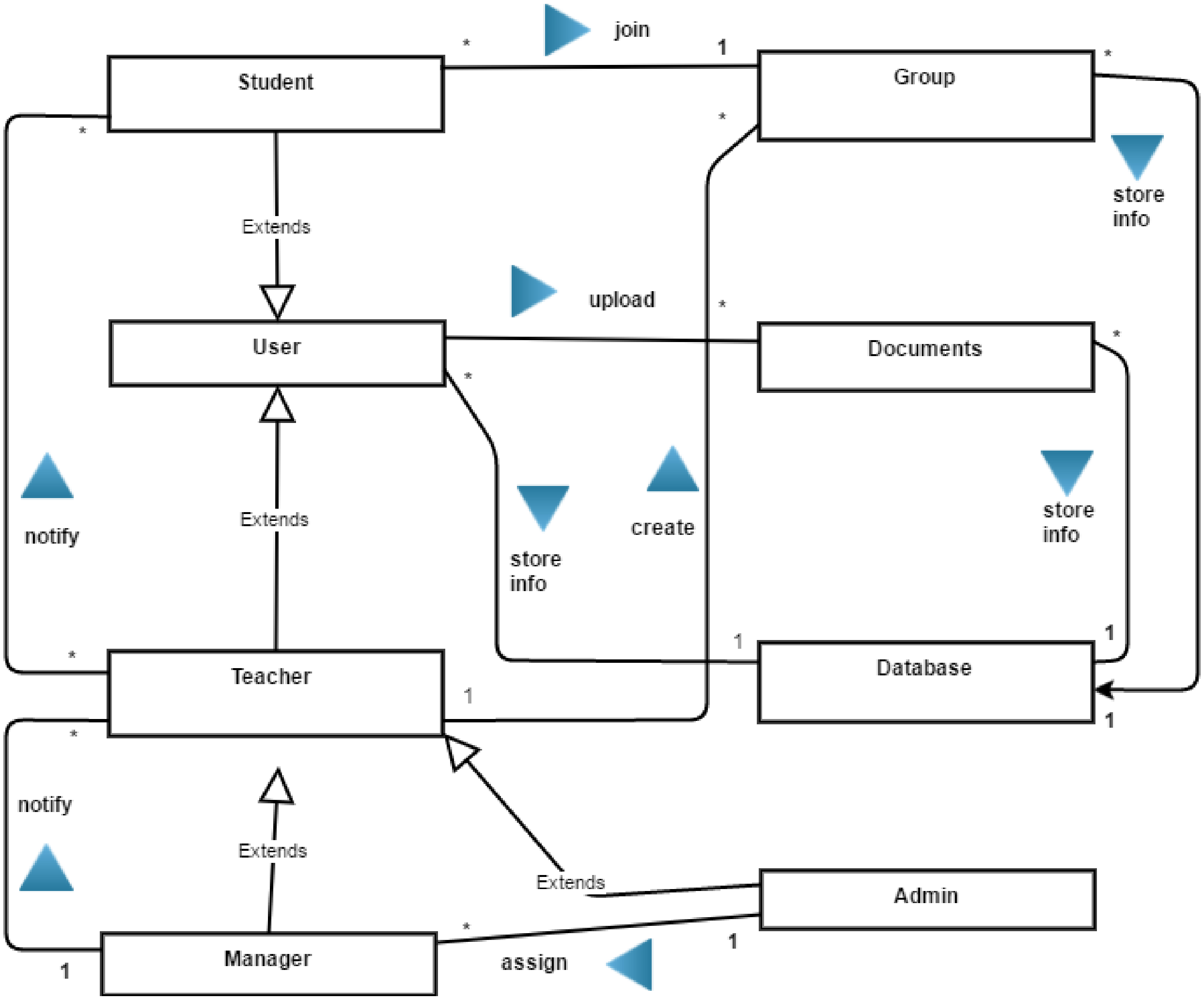


Figure 24: CRC Diagram of SPLMS

CHAPTER 7: BEHAVIORAL MODELING OF SPLMS

7.1 STATE TRANSITION DIAGRAM

State diagram represents active states for each class the events (triggers). For this we identified all the events, their initiators and collaborators.

Identifying Events

Table 26: Event Identification

No	Events	Initiator	Collaborator
1	Input login credentials	User	Database
2	Given information	Database	User
3	Correct input	User	-
4	Incorrect input	User	-
5	Retried	User	-
6	Logged In	User	Manager, Student, Admin, Teacher

7	Account created	Manager	Database
8	Information updated	Database	Manager, teacher, student
9	Course created	Manager	-
10	Notified teachers	Manager	Teachers
11	Notified students	Teachers	Students
12	Got notification	Students	-
13	Joined group	Students	Group
14	Created group	Teachers	-
15	Uploaded documents	Teachers, Students	Document
16	Checked documents	Document	-
17	Stored documents	Document	Database
18	Given feedback	Teacher	Database
19	Option chosen	Admin	-
20	Stored in GitHub	Admin	-
21	Assigned New Committee	Admin	Database, Teacher
22	Assigned New Admin	Admin	Database, Teacher

7.1.1 State Transition Diagram User

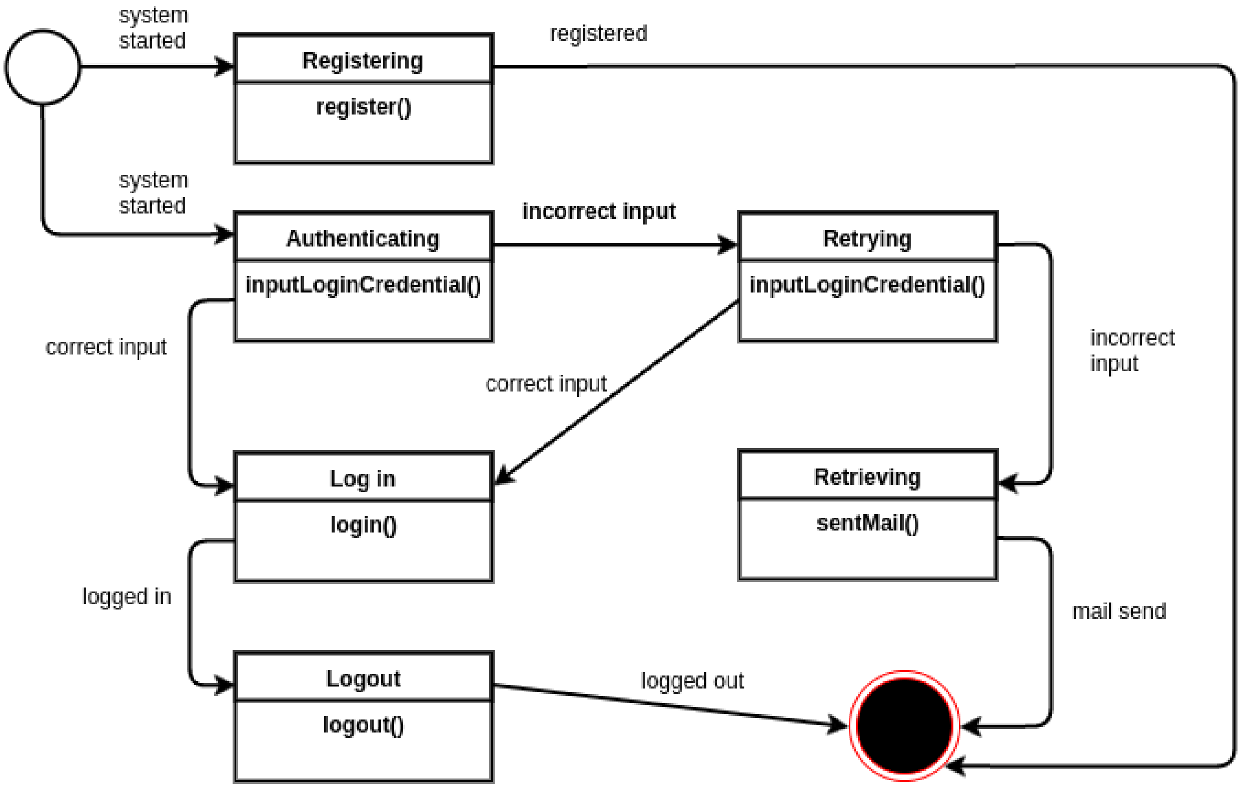


Figure 25: State Transition Diagram User

7.1.2 State Transition Diagram Admin

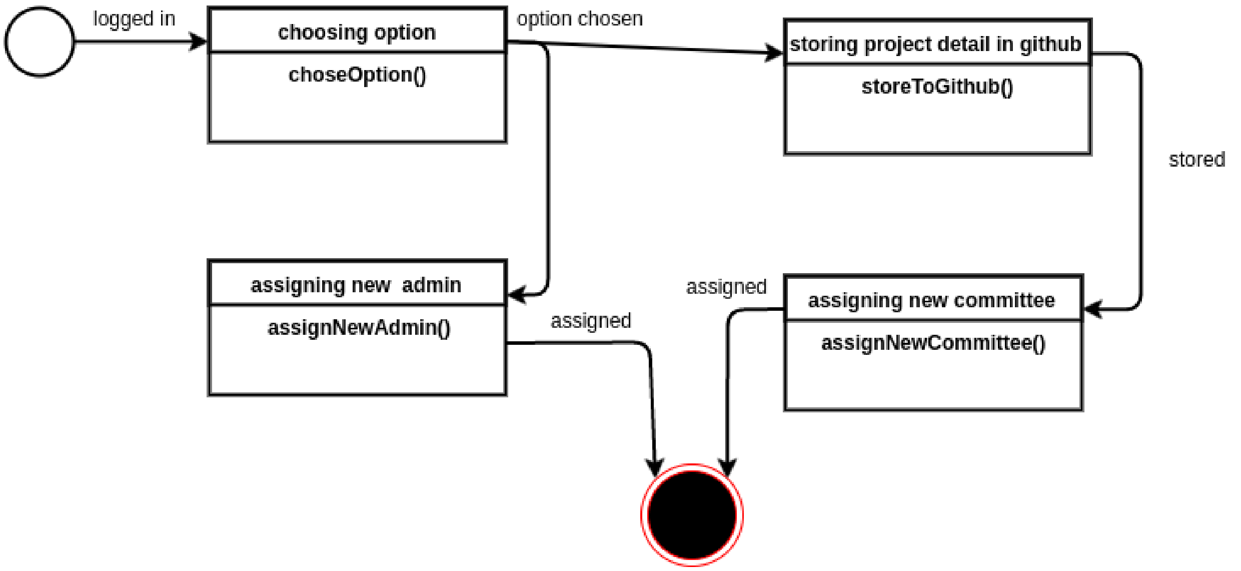


Figure 26: State Transition Diagram Admin

7.1.3 State Transition Diagram Teacher

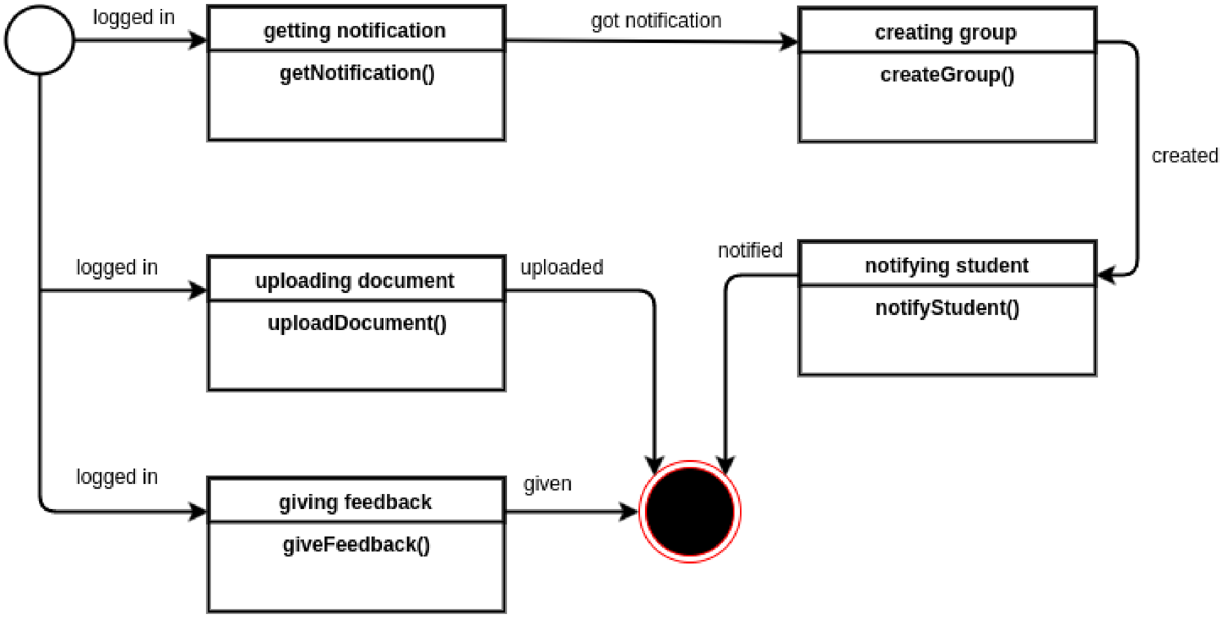


Figure 27: State Transition Diagram Member

7.1.4 State Transition Diagram Manager

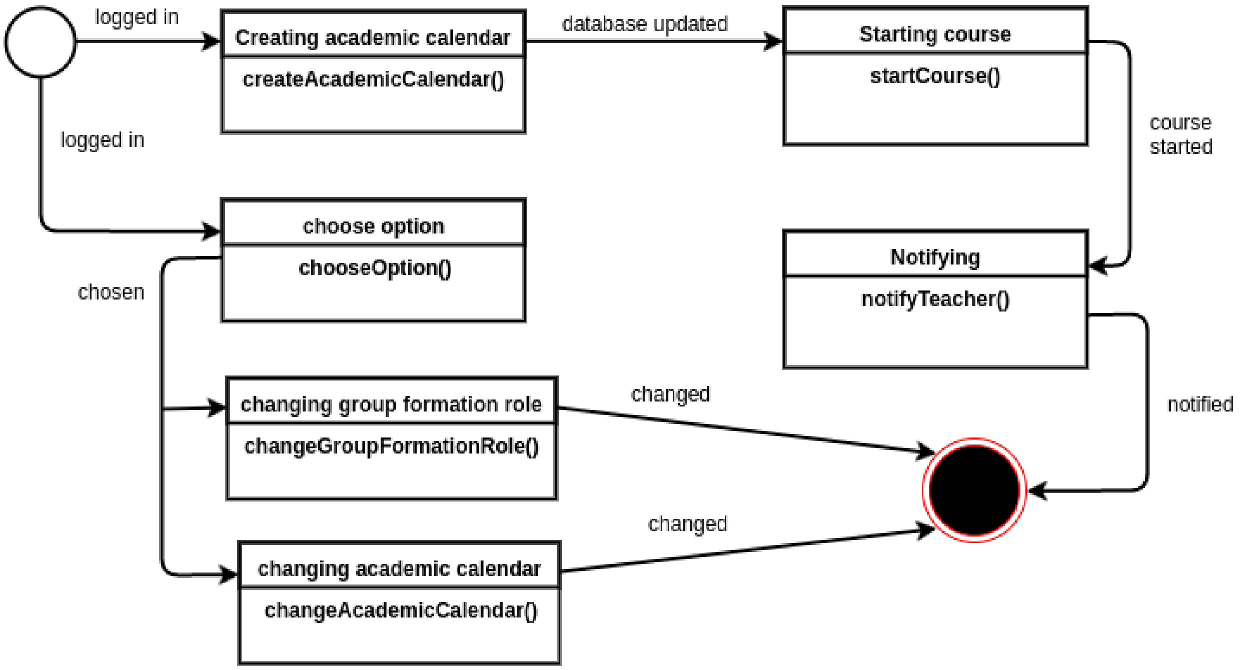


Figure 28: State Transition Diagram Manager

7.1.5 State Transition Diagram Student

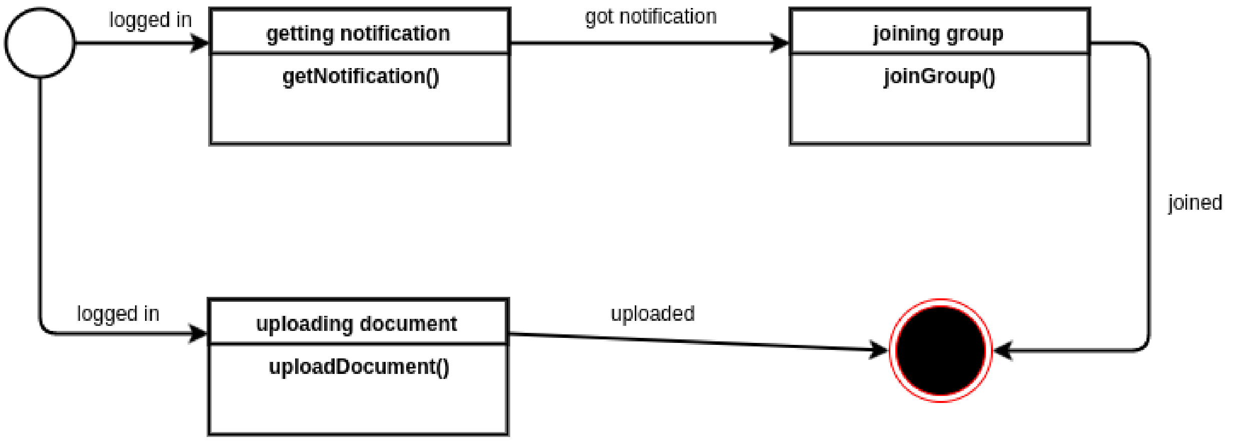


Figure 29: State Transition Diagram – Student

7.1.6 State Transition Diagram Document

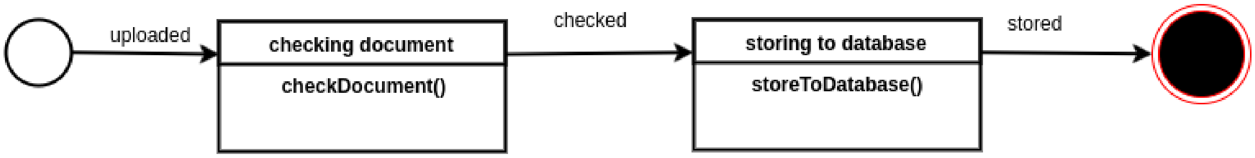


Figure 30: State Transition Diagram Document

7.1.7 State Transition Diagram Group

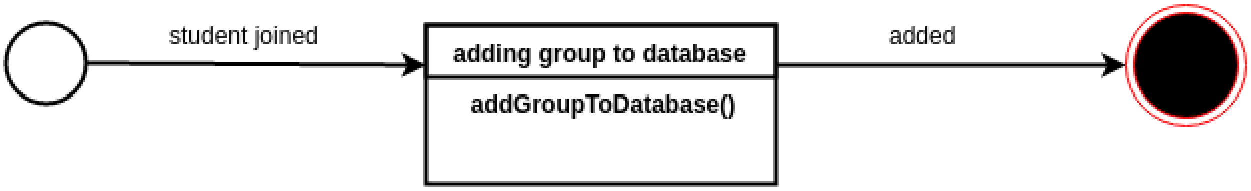


Figure 31: State Transition Diagram Group

7.1.8 State Transition Diagram Database

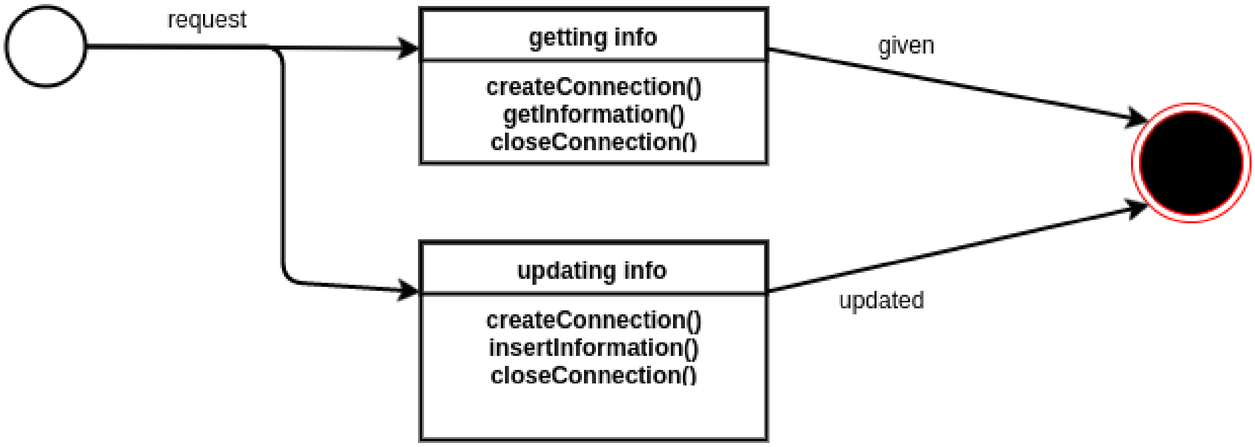


Figure 32: State Transition Diagram Database

7.2 Sequence Diagrams of Modules of SPLMS

7.2.1 Sequential Diagram Authentication

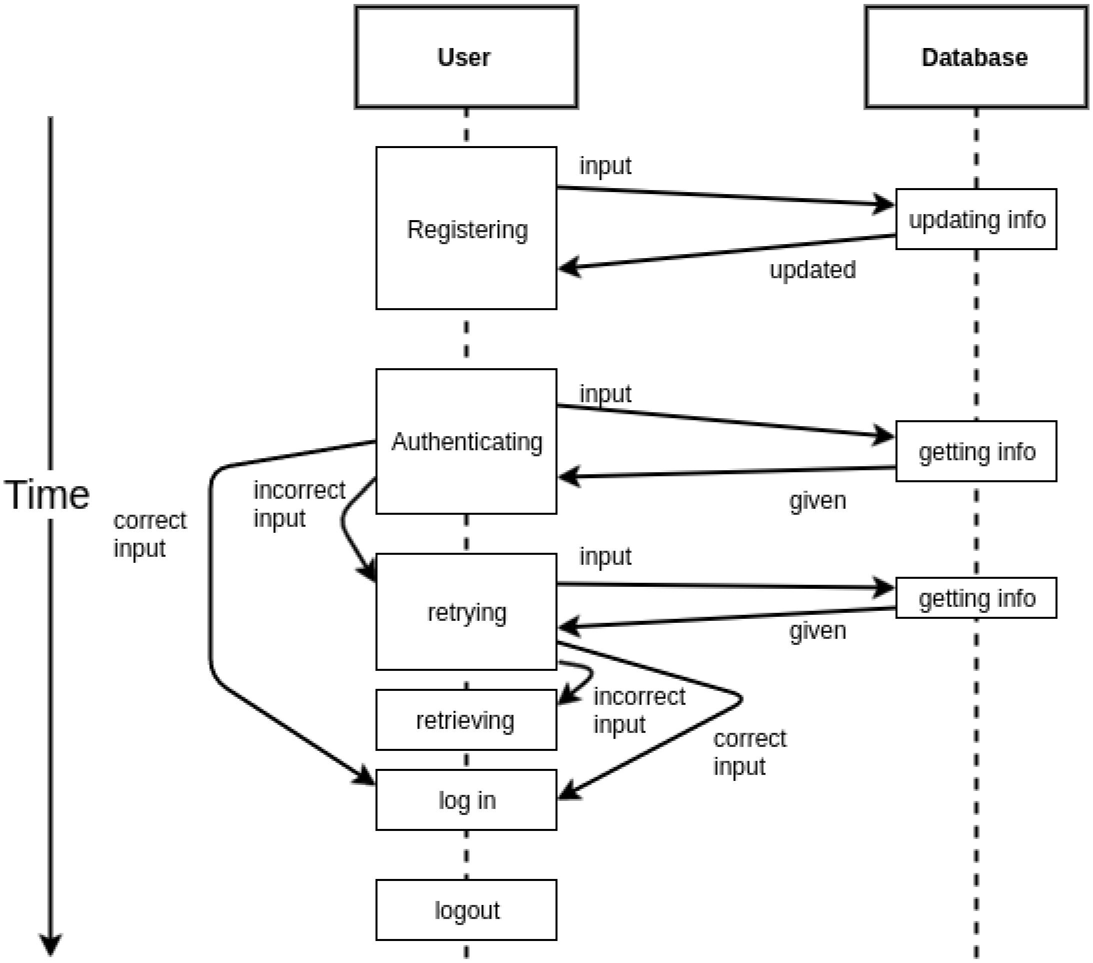


Figure 33: Sequence Diagram of Module-Authentication

7.2.2 Sequential Diagram Initialization

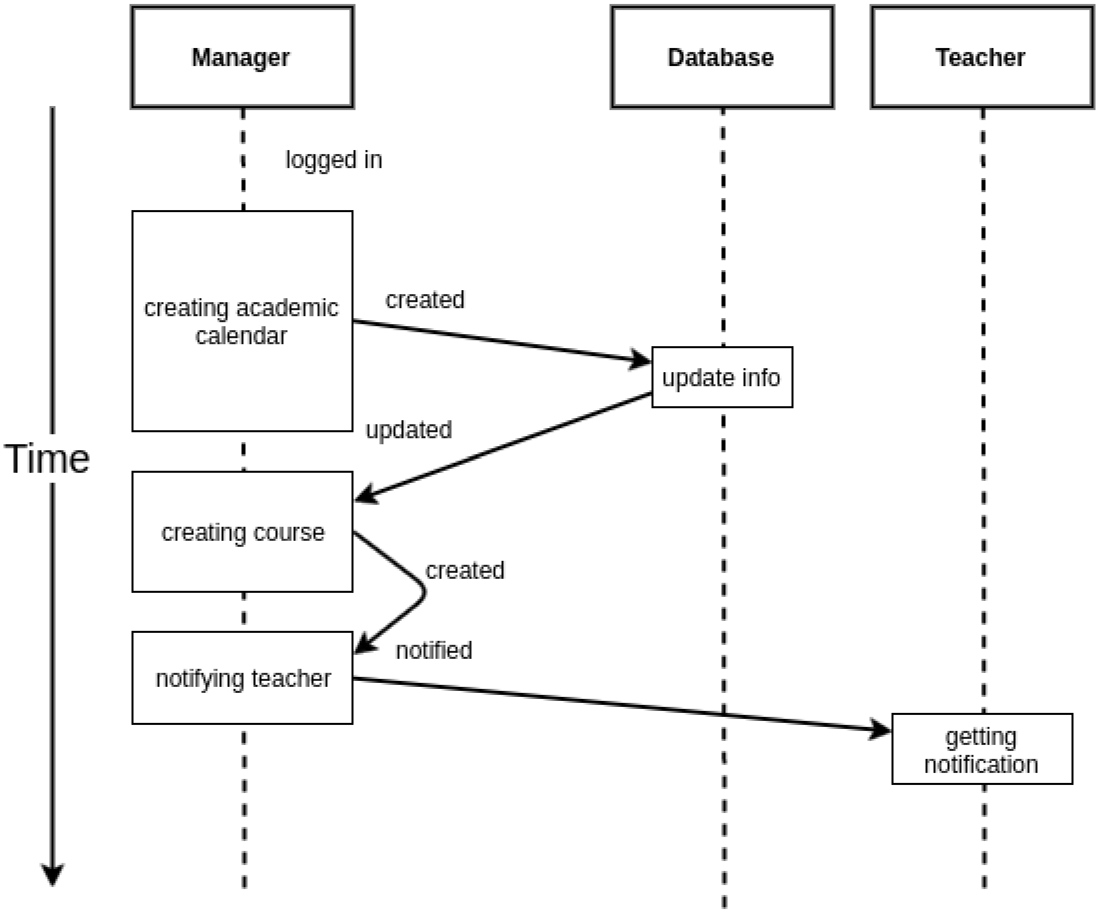


Figure 34: Sequence Diagram of Module-Initialization

7.2.3 Sequential Diagram Group Forming

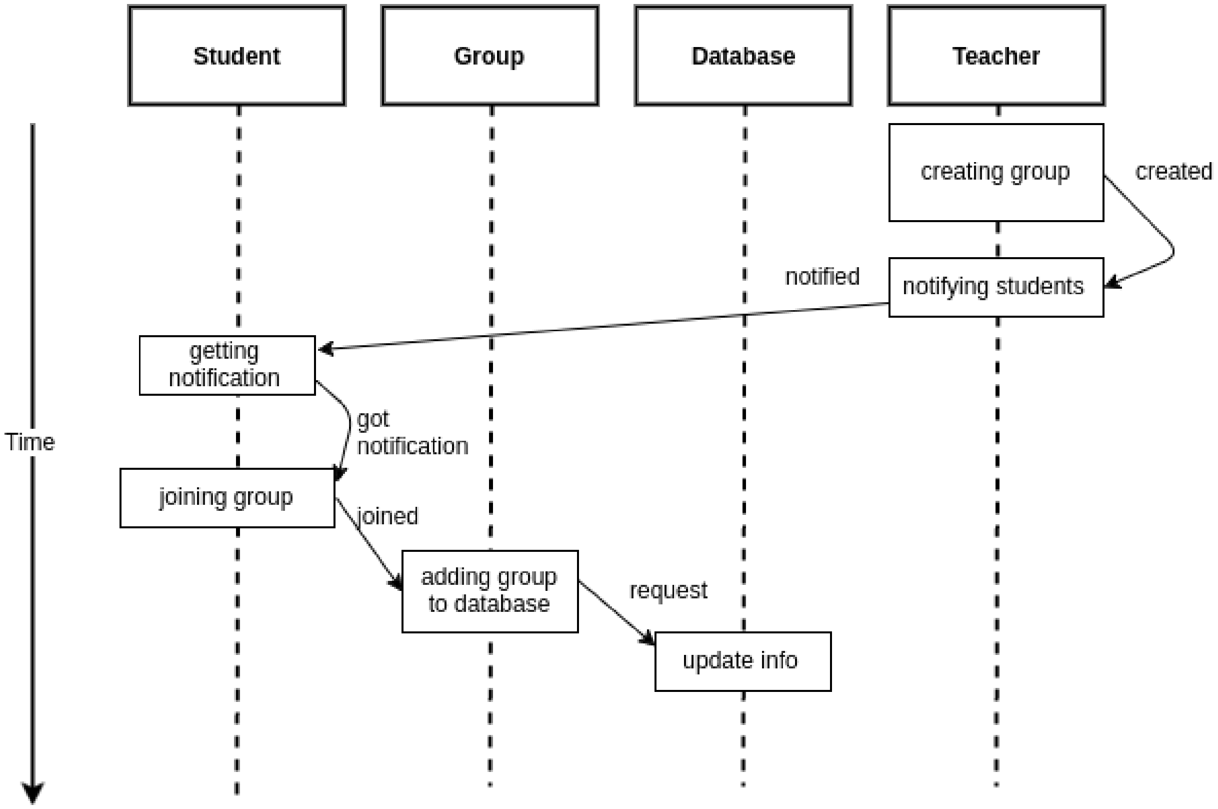


Figure 35: Sequence Diagram of Module-Group Formation

7.2.4 Sequential Diagram Feedback

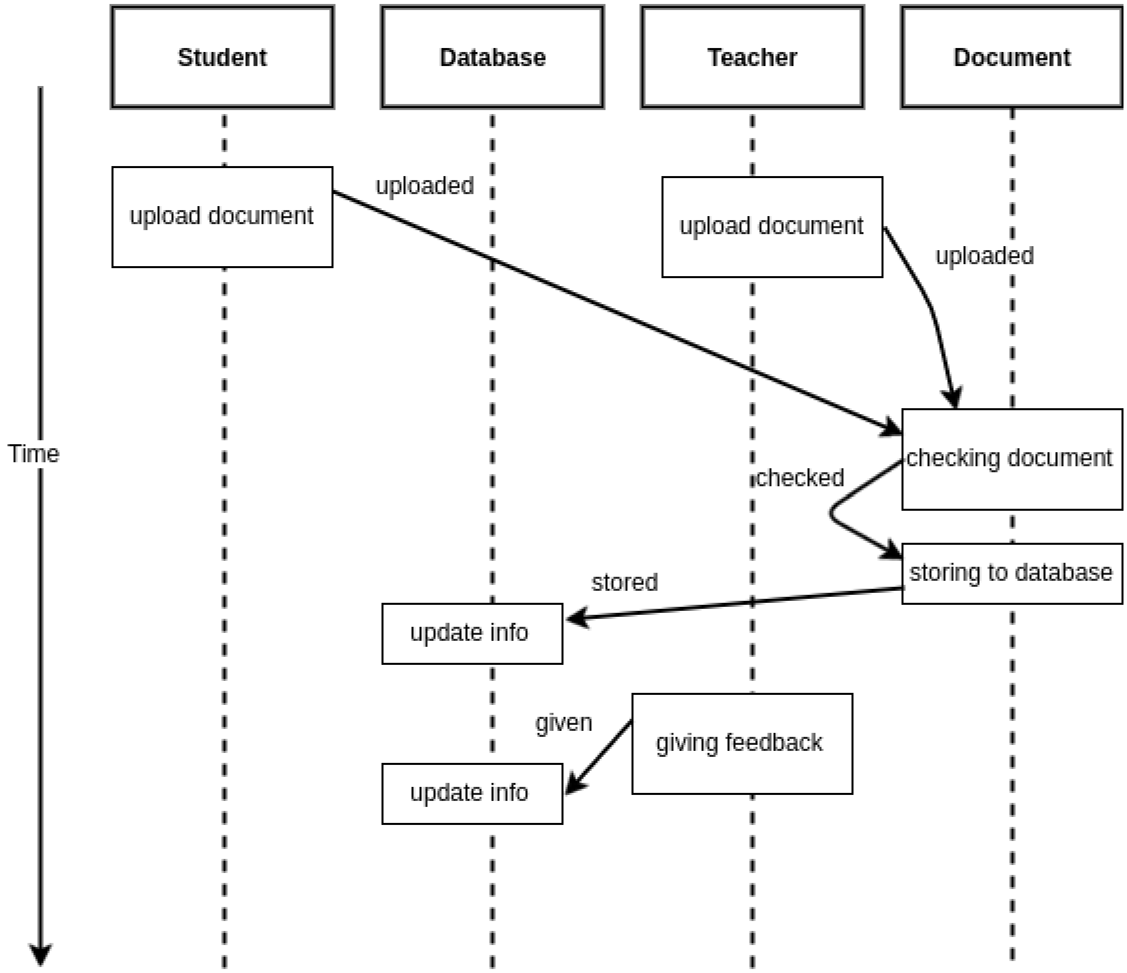


Figure 36: Sequence Diagram of Module-Feedback

7.2.5 Sequential Diagram Update

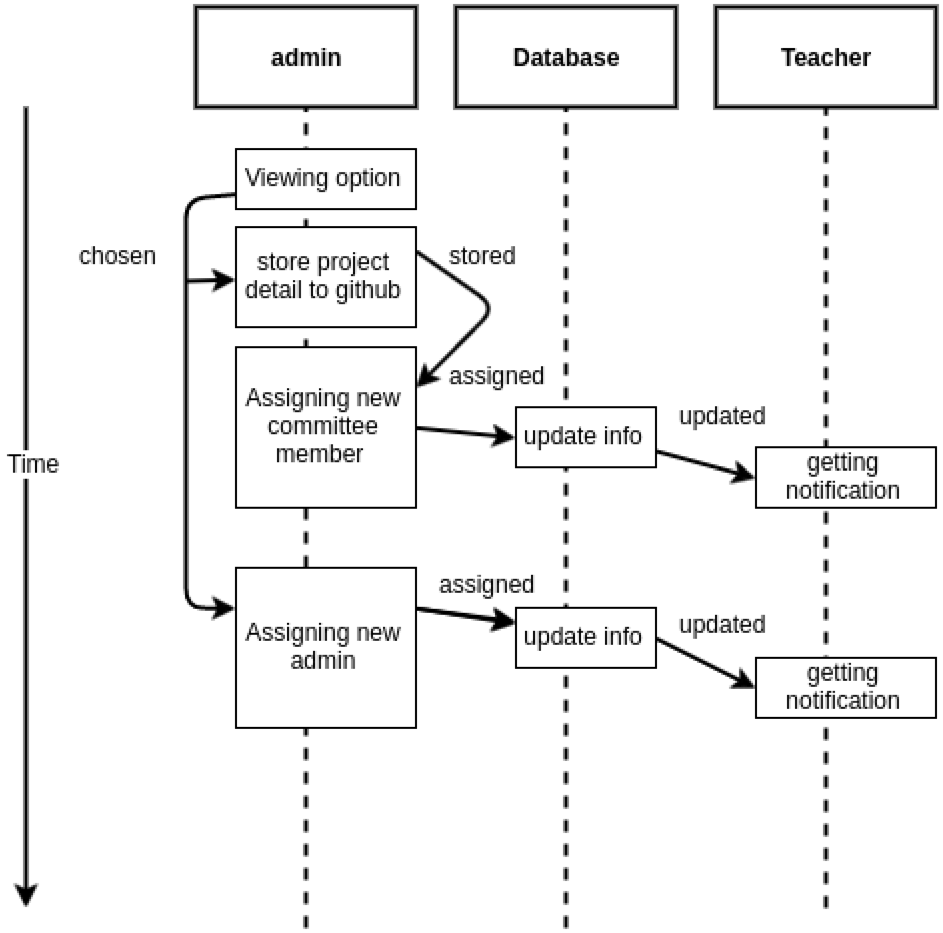


Figure 37: Sequence Diagram of Module-Update

CHAPTER 8: CONCLUSION

We are pleased to submit the final SRS report on Software Project Lab Management System. From this, the readers will get a clear and easy view of the overall system of Software Project Lab courses of IIT. This SRS document can be used effectively to maintain the software development cycle. It will be very easy to conduct the whole project using this SRS. We tried our best to remove all dependencies and make an effective and fully designed SRS. We believe that the reader will find it in order.–

CHAPTER 9: REFERENCES

- Pressman, Roger S. Software Engineering: A Practitioner's Approach (7th Edition)