CHAPTER-01: INTRODUCTION

This chapter is a part of our software requirement specification for the project "Assignment 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 Pharmacy 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 the customer finds acceptable. 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 the right track when developing 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 demands. 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 -02: INCEPTION OF AMS

2.1 INTODUCTION

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. The purpose of the document is to represent a short description of the small class business like pharmacy shop and identify the stakeholders of the pharmacy shop.

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

According to Sommerville and Sawyer [Som97], “Anyone who benefits in a direct or indirect way from the system which is being developed is a stakeholder.” This implies that stakeholders include the end users of the developed software as well as the people whose activities might be influenced by the tool. Towards the end of inception, the list of stakeholders is usually larger as every

stakeholder is allowed to suggest one or more individuals who might be probable stakeholders for the given problem.

To identify stakeholders, we consulted some teachers and students of Dhaka University, Bangladesh and asked them the following questions:

- Who will be using this web application?
- Whose work will these project affect?

We identified following stakeholders for our assignment system:

- Instructor
- Student

Instructor: Instructor is a person who is a creator of the group. S/he creating a code for the students to join his/her group, post assignment, check plagiarism of student's assignment, distribute mark and comment on post.

Student: Student is a person who can submit assignment, resubmit assignment, communicate with instructor and comment on post.

2.1.2 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:

INSTRUCTOR VIEWPOINTS

- User friendly and efficient system
- Computer based system
- Error free system
- Multiple login System: Instructor and student
- Strong Authentication
- Filtering option

- Plagiarism checking
- Easy to operate
- Notification for each post or comment
- Communicate with Student
- Future support from developers
- Give format of assignment
- Check format and requirements to submit
- Distribute mark evaluating plagiarism

STUDENT VIEWPOINTS

- User friendly and efficient system
- Computer based system
- Easy to operate
- Strong authentication
- Resubmission option
- Filtering option
- Post and comment
- Message to teacher
- Multiple access

2.1.3 WORKING TOWARDS COLLABORATION

Each of the stakeholder constituencies (and non-stakeholder constituency) contributes to the requirement engineering process. The greater the numbers of interactions with multiple stakeholders, the higher is the probability of inconsistency, conflicts and clashes of viewpoints. In such circumstances, requirement engineers finalize the requirements following some steps, which are listed below.

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

COMMON REQUIREMENTS

- User friendly
- Strong authentication
- Filtering option
- Resubmission
- Format of assignment
- Plagiarism check
- Multiple access
- Error free system
- Future Support from developers
- Check format and requirements to submit

CONFLICTING REQUIREMENTS

- Limited budget
- Post on the group
- Message to teacher
- Notification for each comment and post

FINIAL REQUIREMENTS

- User friendly
- Strong authentication
- Multiple access
- Filtering option
- Plagiarism checking
- Format of assignment
- Resubmission assignment
- Notification for post and comment
- Check format and requirements to submit
- Post and comment
- Message to teacher

2.1.4 REQUIREMENTS QUESTIONARE

In requirements engineering, the involved individuals can be broadly divided into two clusters: the developers and the stakeholders. Coming from different backgrounds, it will be obvious that these two parties will have different points of views regarding the problem. The stakeholders have more knowledge on facing the problem. Meanwhile, the developers are experienced with providing computerized solutions. Thus, in order to obtain an efficient solution to the problem, it is important to ‘loosen up’ or ‘break the ice’ between the two groups.

Following the ideal guidelines of requirement engineering, some context free questions were asked. The context free questions help throwing light on the stakeholders of the project. The next set of questions includes the context itself so that a better understanding of the problem is obtained. The stakeholder is encouraged to voice out his/her opinions about an alternate solution and also provide recommendations to the developer’s suggestions. The final set of questions focuses on the communication activity itself.

2.2 CONCLUSION

The Inception phase helped us to establish basic understanding about the pharmacy management system, 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-03: ELICITATION OF AMS

After discussing on the inception phase, we need to focus on 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, problems of scope 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. Requirements Elicitation (also called requirements gathering) combines 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 scenario
- Elicitation work products

3.2.1 COLLABORATIVE REQUIREMENTS GATHERING

We have met with stakeholders in the inception phase. The stakeholders are Instructor and student. Many different approaches to collaborative requirements gathering have been proposed by the stakeholders. To solve this problem, we have met with the stakeholders again to elicit the requirements. A slightly different scenario from these approaches has been found.

- The meeting was conducted with the teacher and students of different departments of University of Dhaka. They were questioned about their requirements and expectations.
- They were inquired about the problems with existing workflow.
- The final requirement list was delivered at the end of the meeting.

3.2.2 PROBLEM IN THE SCOPE

A number of problem were encountered in the course of preparing the software requirement specification and analysis of Assignment Management System.

What was done:

- The system is applicable for the soft document (pdf, docx, txt etc.) type assignment.
- Plagiarism checking will be performed between the submitted assignments only.

What was not done:

- The system is not applicable for the hand-written assignment.
- The plagiarism check will not be performed with the outer world.

3.2.3 QUALITY FUNCTION DEPLOYMENT

Quality Function Deployment (QFD) is a technique that translates the needs of the customer into technical requirements for software. It 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.

3.2.3.1 NORMAL REQUIREMENTS

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

- Assignment filtering with group
- Assignment submission

- Assignment resubmission
- Format checking when submit
- Mail sending for new assignment

3.2.3.2 EXPECTED REQUIREMENTS

- Comment and post
- Notification
- Authentication
- User friendly
- Multiple access
- Error free system

3.2.3.3 EXCITING REQUIREMENTS

- Plagiarism check
- Message to teacher

3.2.4 USAGE SCENARIO

Assignment Management System (AMS) is an automated system for the following purposes:

- Authentication
- Assignment management
- Group management
- Communication

3.2.4.1 AUTHENTICATION

SIGN UP

When user will access the system, s/he will view the sign up and sign in options. In this system, user will register with first name, last name, email, phone number and password for creating an account. After data entry, there will be a validity check. The password must contain minimum 8

characters and maximum 30 characters including numbers. The regex will be used to check the email address and phone number. User must have to give unique email id for sign up.

SIGN IN

User will sign in with his/her email id and password. If the entered data match with the corresponding data stored in the database, the user will be able to access the system. If the password is entered incorrectly more than 5 times the user has to recover his/her account.

Account recovery

If the user forgets his/her password, s/he can recover his/her account by "forgot password" option. When the user will select the "forgot password" option, s/he has to enter his/her email address. Then a link will be sent to his/her email. When the user will select the link, s/he will be redirected to a web page, where the user will set his/her new password. Then the user can log in to his/her account using the new password.

SIGN OUT

When a user will attempt to log out, the system will check if there are any running process. If there are any running process, the system will warn him/her to close running processes. If the user does not close the running processes, the system will close the running processes forcedly.

3.2.4.2 ASSIGNMRNT MANAGEMENT

POST ASSIGNMENT

When an instructor posts an assignment, s/he can give the students a format that students have to follow at the time of assignment submission. An instructor has to fill assignment title, description of assignment and deadline of submission to post an assignment. An instructor can attach files to an assignment. When an instructor posts an assignment, a message will be sent through email to all the students. All the students can view the assignment and comment on the assignment that the instructor posts. Instructor and students can post an announcement about a specific topic and all (instructor and student) can comment on that post. The instructor can search, download and view

the assignment of the individual student. Instructor can also view the submission time and date of every student. When all the students submit the assignments, all the assignments are stored in one folder. The instructor can download the folder in which all assignments of the students are stored. The instructor can filter the assignments by students and group wise and students can filter assignments by course wise.

ASSIGNMENT SUBMISSION

All the students can see the assignment given by the instructor. The students can submit the assignment within a time duration that is given by the instructor. If the instructor gives a format of an assignment, the students have to follow the specific format at the time of submission. Otherwise, the system will show an error message of submission. Students can cancel submission and resubmit the assignment within the deadline for submission. Students cannot submit assignment unless they cannot full fill all requirements (cannot submit 2 files of total 3). Students will not be able to submit after the deadline of submission time if the instructor not allows late submission.

PLAGIARISM CHECK

The instructor can check plagiarism of assignments that students will submit. Plagiarism will be checked by the MOSS (for a measure of software similarity), a free-level software to detect plagiarism. To check plagiarism using MOSS, instructor have to select “check plagiarism” option for an assignment. After checking plagiarism MOSS will send the result of similarity as a link. An instructor can see the percentage of similarity of every student.

MARK DISTRIBUITION

After the evaluation, the instructor can give marks to each assignment of the students in this system.Then instructor can publish the marks of assignments in the group and each member can view the marks if the instructor wishes.

3.2.4.3 GROUP MANAGEMNT

When a user creates a group, the system considers the user as an instructor. The instructor must have to sign in to his/her account to create a group. Then the user has to fill up a group name (class name/course name), section, and subject to create a group. After creating the group, the system will ask the instructor to create a code for his/her group. A student can join groups using code that is created by group instructor. A student can join many groups by using group-codes as s/he wants. Also, an instructor can create multiple groups. An instructor can remove or update his/her group. Instructor and students can post on the group

3.2.5 ELICITATION WORK PRODUCT

At first, we have to know whether the output of the Elicitation task may vary because of the dependency on the size of the system or the product to be built. Here, the Elicitation work product includes:

- Making a statement of our requirements for the Pharmacy Management System.
- Making a bounded statement of scope for our system.
- Making a list of users and other stakeholders who participated in the requirements elicitation.
- A set of usage scenarios that provide insight into the use of the system.
- Description of the system's technical environment

CHAPTER-04: SCENARIO BASED MODELING OF AMS

This chapter describes the Scenario Based Model for the Assignment 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 DEFINATION 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 DIAGRAM

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

4.3.1 LEVEL- 0 USE CASE DIAGRAM-AMS

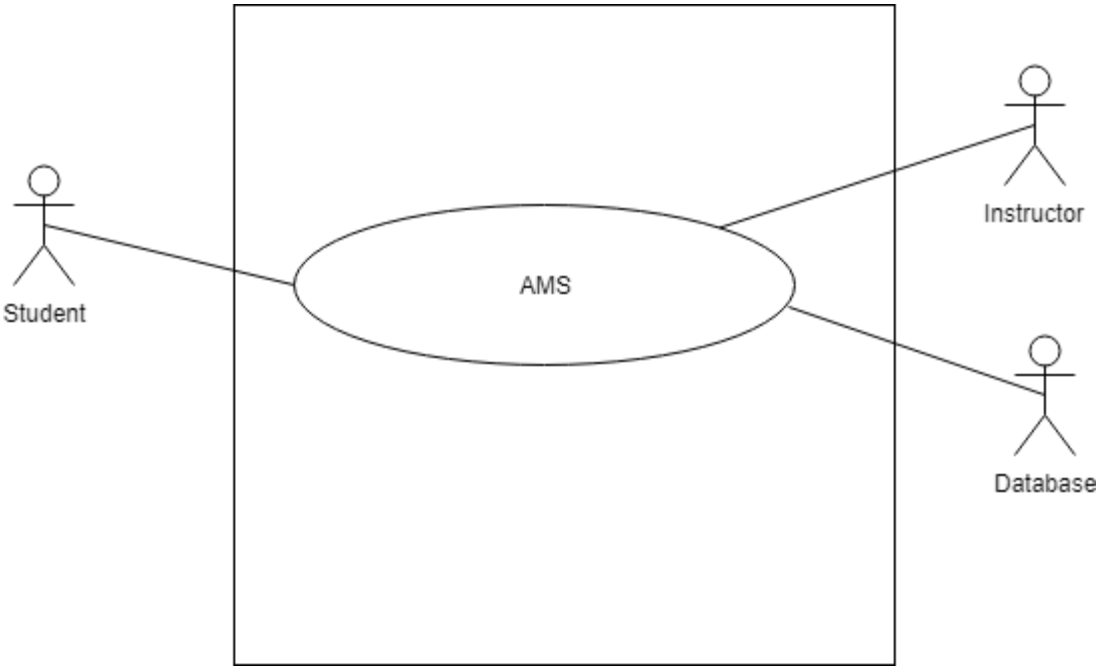


Figure-1: level 0 use case diagram- AMS

Name: Assignment Management System

Primary actor: Instructor, Student, Database

Secondary actor: Result Management System

DESCRIPTION OF USECASE DIAGRAM LEVEL-0

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

Following the actors of "Assignment Management System":

- Instructor
- Student
- Database

4.3.2 LEVEL -1 USECASE DIAGRAM-SUBSYSTEM

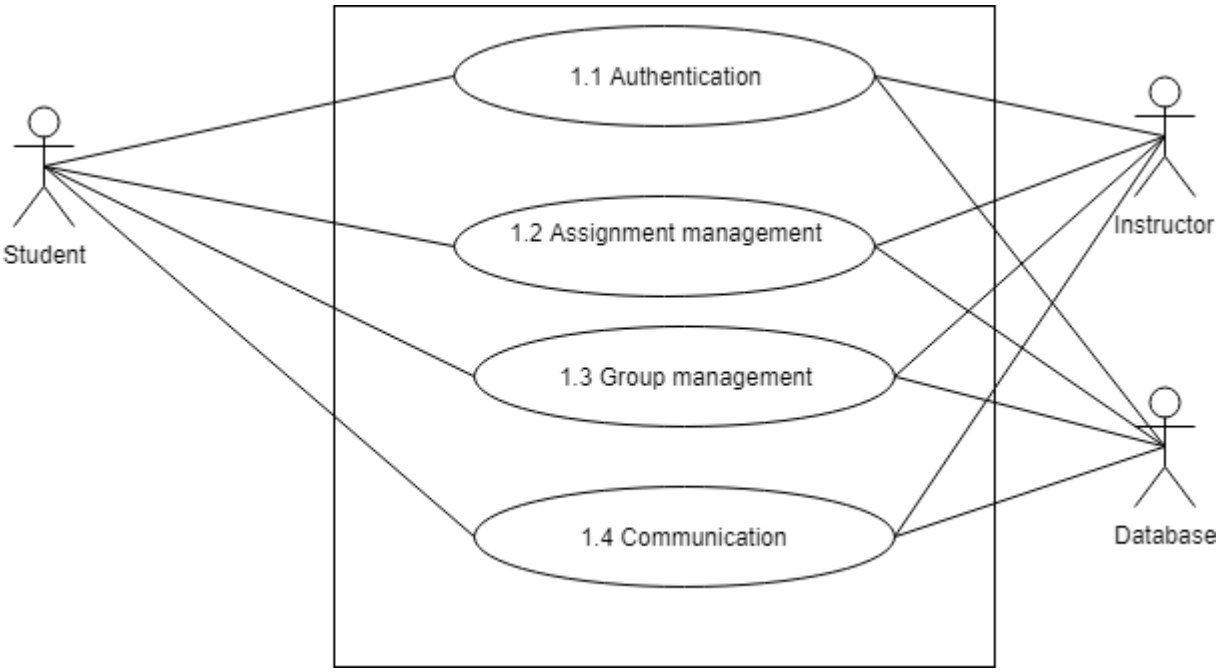


Figure-2: level 1 use case diagram - Subsystem

Name: Subsystem of AMS

Primary actor: Instructor, Student, Database

Secondary actor: N/A

There are 4 subsystems in the Assignment Management System. They are-

- Authentication
- Assignment management
- Group management
- Communication

4.3.3 LEVEL- 1.1 USE CASE DIAGRAM- AUTHENTICATION

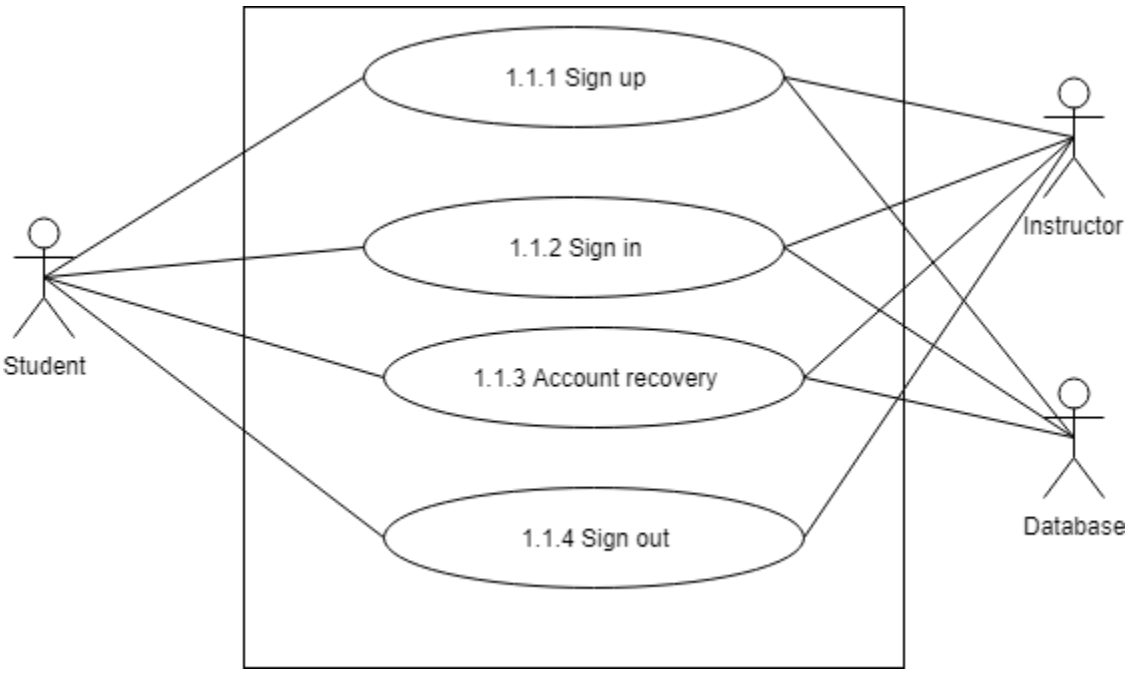


Figure-3: level 1.1 use case diagram – Authentication

Name: Authentication of AMS

Primary actor: Instructor, Student, Database

Secondary actor: N/A

DESCRIPTION OF LEVEL- 1.1 USE CASE DIAGRAM-

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 PMS can be divided into four parts. These are:

- Sign up
- Sign in
- Account recovery
- Sign out

1.1.1 SIGN UP

- Primary actor: Instructor, Student, Database
- Secondary actor: N/A

STUDENTS ACTION/REPLY

- Action: Student enter information to sign up.
- Reply: System check validity and store information.

INSTRUCTOR ACTION/REPLY

- Action: Instructor enter information to sign up.
- Reply: System check validity and store information.

DATABASE ACTION/REPLY

- Store valid data.
- Show data successfully store or not.

1.1.2 SIGN IN

- Primary actor: Instructor, Student, Database

- Secondary actor: N/A

STUDENTS ACTION/REPLY

- Action: Student enter information to sign in.
- Reply: System check validity and store information.

INSTRUCTOR ACTION/REPLY

- Action: Instructor enter information to sign in.
- Reply: System check validity and store information.

DATABASE ACTION/REPLY

- Store valid data.
- Show data successfully store or not

1.1.3 ACCOUNT RECOVERY

- Primary actor: Instructor, Student, Database
- Secondary actor: N/A

STUDENTS ACTION/REPLY

- Action: Student enter email.
- Reply: System check validity and store information.

INSTRUCTOR ACTION/REPLY

- Action: Instructor enter information to sign in.
- Reply: System check validity and send pin.

DATABASE ACTION/REPLY

- Store modified data.
- Show data successfully store or not

4.3.4 LEVEL -1.2 USE CASE DIAGRAM- ASSIGNMENT MANAGEMENT

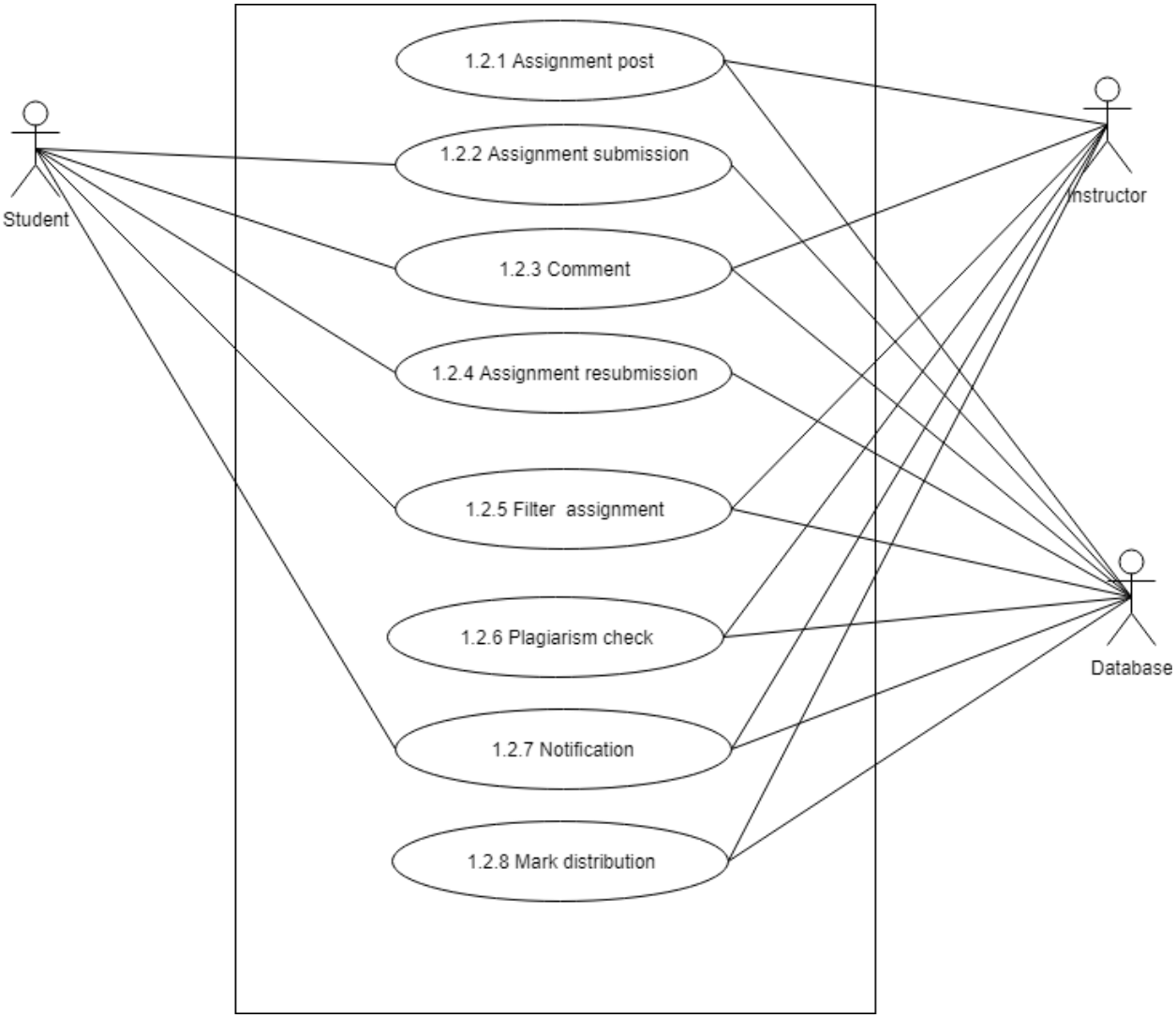


Figure-4: level 1.2 use case diagram- Assignment management

Name: Assignment management of AMS

Primary actor: Instructor, Student, Database

Secondary actor: N/A

DESCRIPTION OF LEVEL- 1.2 USE CASE DIAGRAM-

There are 8 subsystems in Assignment subsystem. These are-

- Assignment post
- Assignment submission
- Comment
- Assignment resubmission
- Filter assignment
- Plagiarism check
- Notification
- Mark distribution

4.3.5 LEVEL -1.2.1 USE CASE DIAGRAM- POST ASSIGNMENT

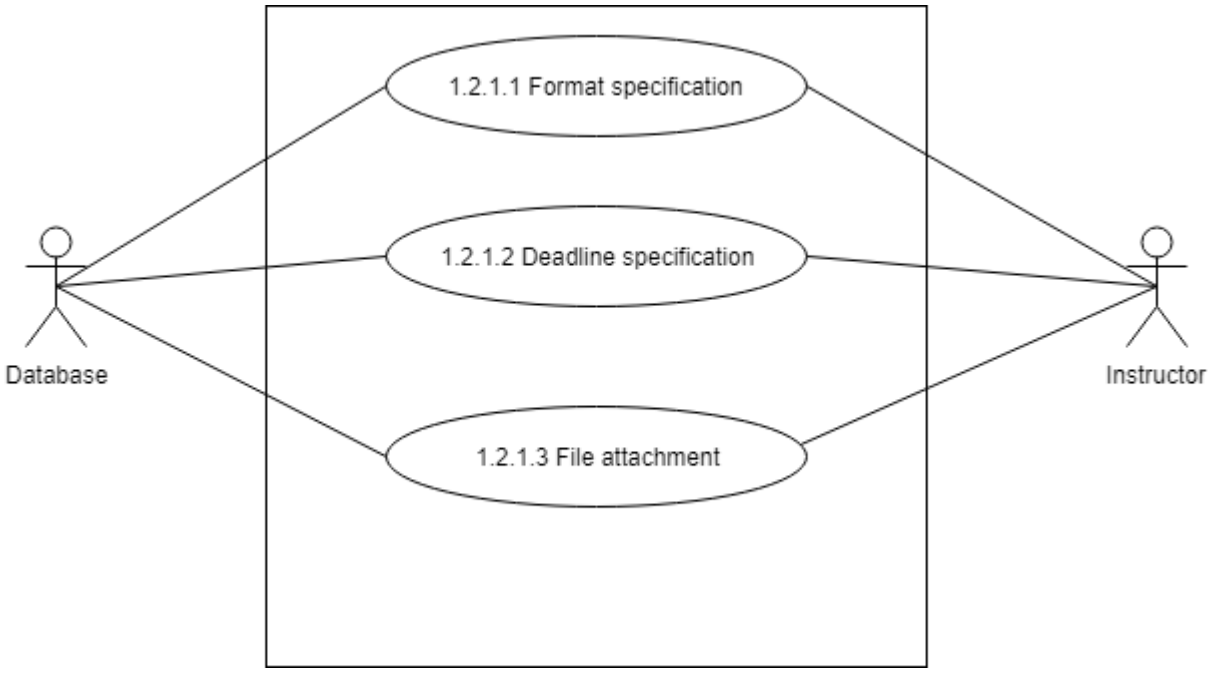


Figure-5: level 1.2.1 use case diagram- Post assignment

Name: Post Assignment of AMS

Primary actor: Instructor, Database

Secondary actor: N/A

DESCRIPTION OF LEVEL -1.2.1 USE CASE DIAGRAM-

There are 3 subsystems of Post assignment subsystems. These are-

- Format specification
- Deadline specification
- File attachment

1.2.1.1 FORMAT SPECIFICATION

- Primary actor: Instructor, Database
- Secondary actor: N/A

INSTRUCTOR ACTION/REPLY

- Action: Select a format.
- Reply: Format selected.

DATABASE ACTION/REPLY

- Action: Store format of assignment.
- Reply: Format stored.

1.2.1.2 DEADLINE SPECIFICATION

- Primary actor: Instructor, Database
- Secondary actor: N/A

INSTRUCTOR ACTION/REPLY

- Action: Give deadline for assignment.
- Reply: Deadline confirmed.

DATABASE ACTION/REPLY

- Action: Store deadline of assignment.
- Reply: Deadline stored.

1.2.1.3 FILE ATTACHMENT

- Primary actor: Instructor, Database
- Secondary actor: N/A

INSTRUCTOR ACTION/REPLY

- Action: Select a file to attach.
- Reply: File attached.

DATABASE ACTION/REPLY

- Action: Store attach file of assignment.
- Reply: File stored.

1.2.2 ASSIGNMENT SUBMISSION

- Primary actor: Student, Database
- Secondary actor: N/A

STUDENT ACTION/REPLY

- Action: Students submit assignment.
- Reply: System will check format and requirements of corresponding assignment and show a message whether it is submitted or not.

DATABASE ACTION/REPLY

- Action: Store the assignment.
- Reply: Show assignment successfully store or not.

1.2.3 COMMENT

- Primary actor: Instructor, Student, Database
- Secondary actor: N/A

INSTRUCTOR ACTION/REPLY

- Action: Instructor comment on any post
- Reply: Commented on post.

STUDENT ACTION/REPLY

- Action: Student comment on any post
- Reply: Commented on post.

DATABASE ACTION/REPLY

- Action: Store comment in the database.
- Reply: Show a message whether it successfully store or not.

1.2.4 ASSIGNMENT RESUBMISSION

- Primary actor: Student, Database
- Secondary actor: N/A

STUDENT ACTION/REPLY

- Action: Submit assignment.
- Reply: System will check format and requirements of corresponding assignment and show a message whether it is submitted or not.

DATABASE ACTION/REPLY

- Action: Store the assignment.
- Reply: Show assignment successfully store or not.

1.2.5 FILTER ASSIGNMENT

- Primary actor: Instructor, Student, Database
- Secondary actor: N/A

INSTRUCTOR ACTION/REPLY

- Action: Instructor select option to filter assignment.
- Reply: Assignment filtered.

STUDENT ACTION/REPLY

- Action: Student select option to filter assignment.
- Reply: Assignment filtered.

DATABASE ACTION/REPLY

- Action: View filtered assignment.
- Reply: Assignment viewed.

1.2.6 PLAGIARISM CHECK

- Primary actor: Instructor, Database
- Secondary actor: N/A

INSTRUCTOR ACTION/REPLY

- Action: Instructor select assignment folder to check plagiarism.
- Reply: System give corresponding folder path to MOSS.

DATABASE ACTION/REPLY

- Action: Evaluating plagiarism of file a link will store on the database.
- Reply: link stored in the database.

1.2.7 NOTIFICATION

- Primary actor: Instructor, Student, Database
- Secondary actor: N/A

INSTRUCTOR ACTION/REPLY

- Action: Instructor select notification to view.
- Reply: Notification viewed.

STUDENT ACTION/REPLY

- Action: Student select notification to view.
- Reply: Notification viewed.

Database action/reply

- Action: Store notification.
- Reply: Notification stored.

1.2.8 MARK DISTRIBUTION

- Primary actor: Instructor, Database
- Secondary actor: N/A

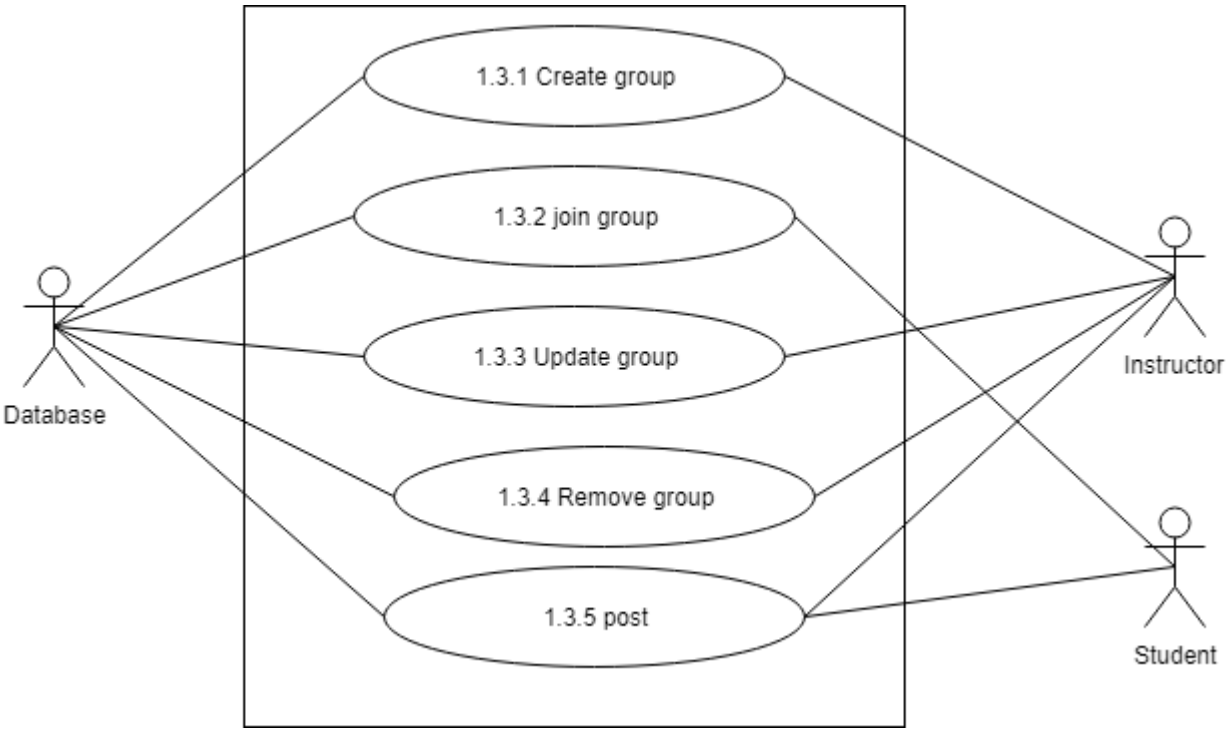
INSTRUCTOR ACTION/REPLY

- Action: Instructor select student to give mark.
- Reply: Mark was given.

DATABASE ACTION/REPLY

- Action: Store mark in the database
- Reply: Mark stored.

4.3.6 LEVEL -1.3 USE DIAGRAM- GROUP MANAGEMENT



. Figure-6: level 1.3 use case diagram- Group management

Name: Group management of AMS

Primary actor: Instructor, Student, Database

Secondary actor: N/A

DESCRIPTION OF LEVEL -1.3 USE CASE DIAGRAM-

There are 5 subsystems in group management subsystem. These are-

- Create group
- Join group
- Update group

- Remove group
- Post

1.3.1 CREATE GROUP

- Primary actor: Instructor, Database
- Secondary actor: N/A

INSTRUCTOR ACTION/REPLY

- Action: Instructor give group name, section and subject to create a group.
- Reply: Group was created.

DATABASE ACTION/REPLY

- Action: Store group information.
- Reply: Information stored.

1.3.2 JOIN GROUP

- Primary actor: Student, Database
- Secondary actor: N/A

STUDENT ACTION/REPLY

- Action: Student enter code to join group.
- Reply: System show a message whether student is entered or not.

DATABASE ACTION/REPLY

- Action: Store student information.
- Reply: Information stored.

1.3.3 UPDATE GROUP

- Primary actor: Instructor, Database
- Secondary actor: N/A

INSTRUCTOR ACTION/REPLY

- Action: Instructor edit group information to update.
- Reply: Group was updated.

DATABASE ACTION/REPLY

- Action: Store update group information.
- Reply: Information stored.

1.3.4 REMOVE GROUP

- Primary actor: Instructor, Database
- Secondary actor: N/A

INSTRUCTOR ACTION/REPLY

- Action: Instructor edit group information to update.
- Reply: Group was updated.

DATABASE ACTION/REPLY

- Action: Store update group information.
- Reply: Information stored.

1.3.5 POST

- Primary actor: Instructor, Student, Database
- Secondary actor: N/A

INSTRUCTOR ACTION/REPLY

- Action: Instructor give post.
- Reply: Posted successfully.

STUDENT ACTION/REPLY

- Action: Student give post.

- Reply: Posted Successfully.

DATABASE ACTION/REPLY

- Action: Store post.
- Reply: Post stored.

1.4 COMMUNICATION

- Primary actor: Instructor, Student, Database
- Secondary actor: N/A

INSTRUCTOR ACTION/REPLY

- Action: Instructor comment or message to communicate.
- Reply: Communicated successfully.

STUDENT ACTION/REPLY

- Action: Student comment or message to communicate.
- Reply: Communicated successfully.

DATABASE ACTION/REPLY

- Action: Store comment or message.
- Reply: Comment or message stored.

4.4 ACTIVITY DIAGRAMS

ACTIVITY DIAGRAM – 1: AUTHENTICATION

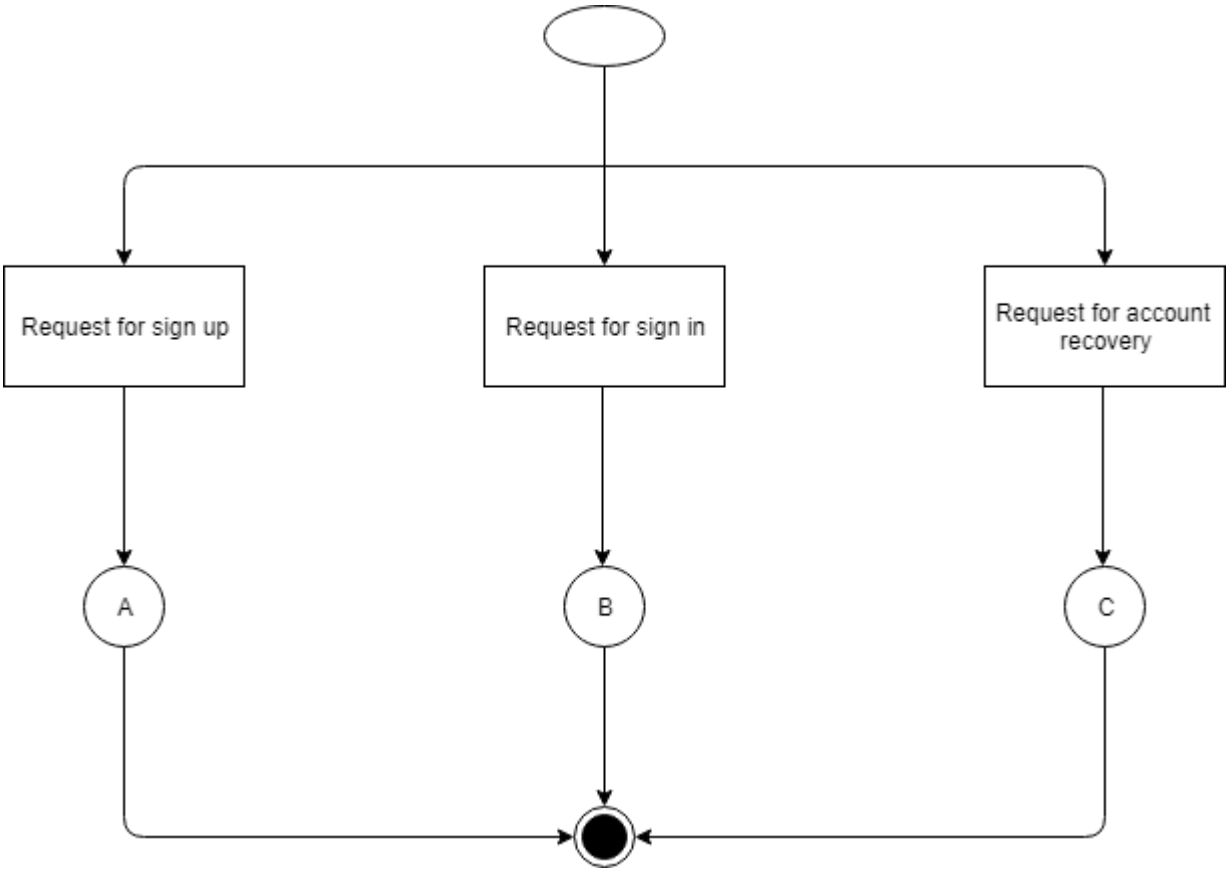


Figure – 7: Level 1.1 Activity diagram – Authentication.

ACTIVITY DIAGRAM – 1.1: SIGN UP

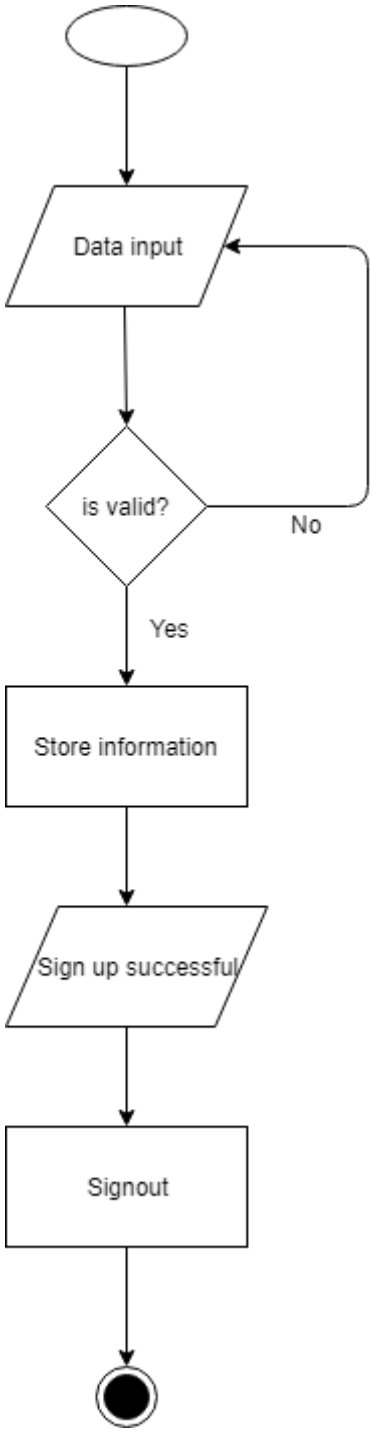


Figure – 8: Level 1.1.1 Activity diagram – Sign up.

ACTIVITY DIAGRAM – 1.2: SIGN IN

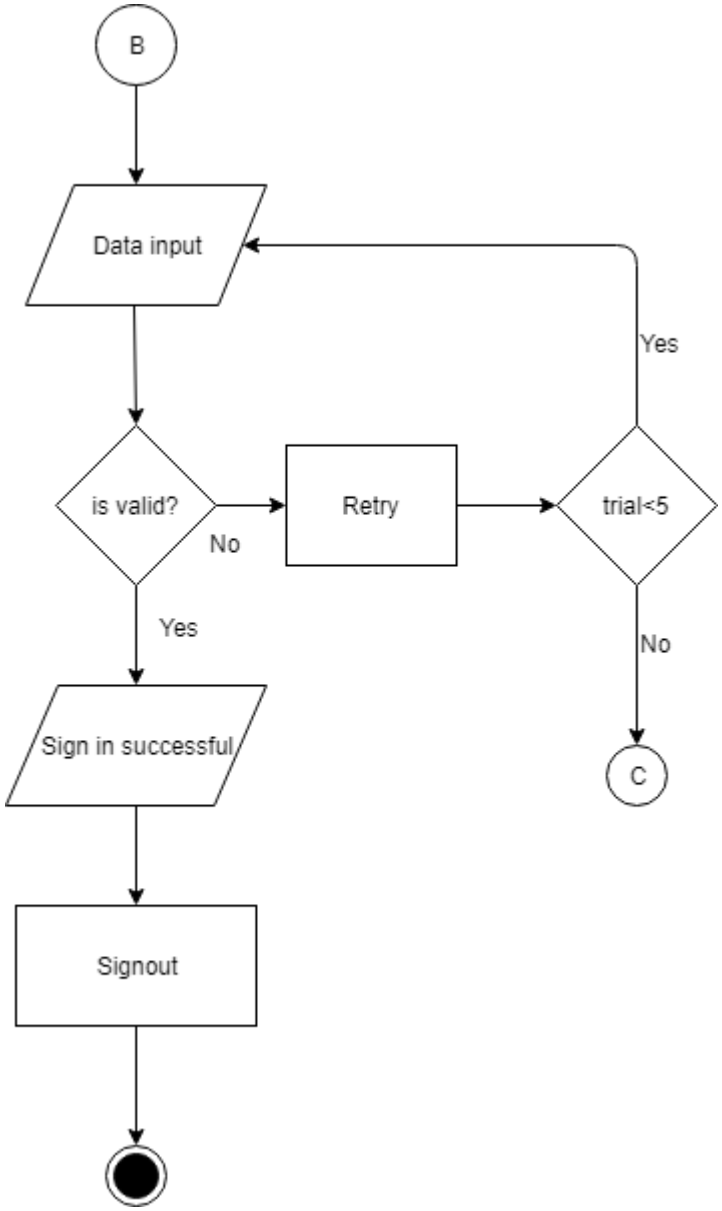


Figure – 9: Level 1.1.2 Activity diagram – Sign in.

ACTIVITY DIAGRAM – 1.3: ACCOUNT RECOVERY

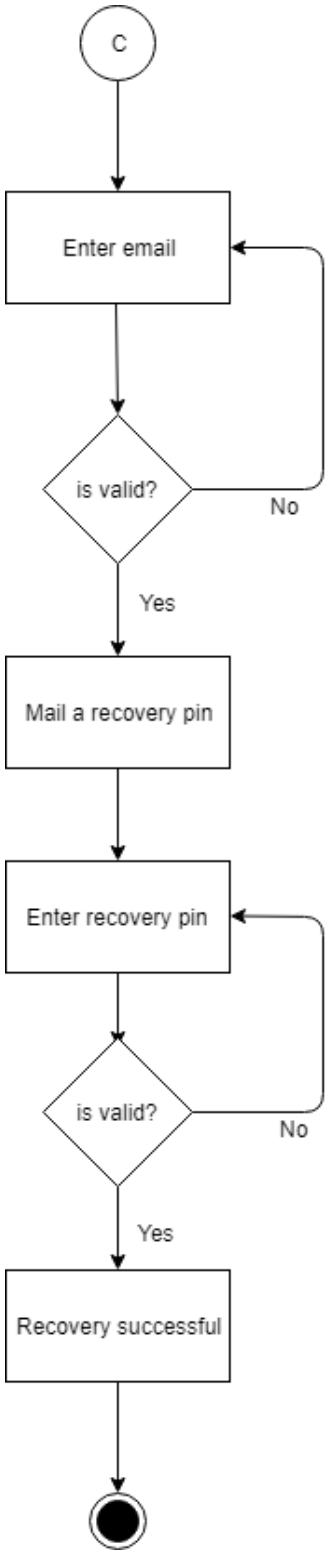


Figure – 9: Level 1.1.3 Activity diagram – Account recovery

ACTIVITY DIAGRAM – 2: ASSIGNMENT MANAGEMENT

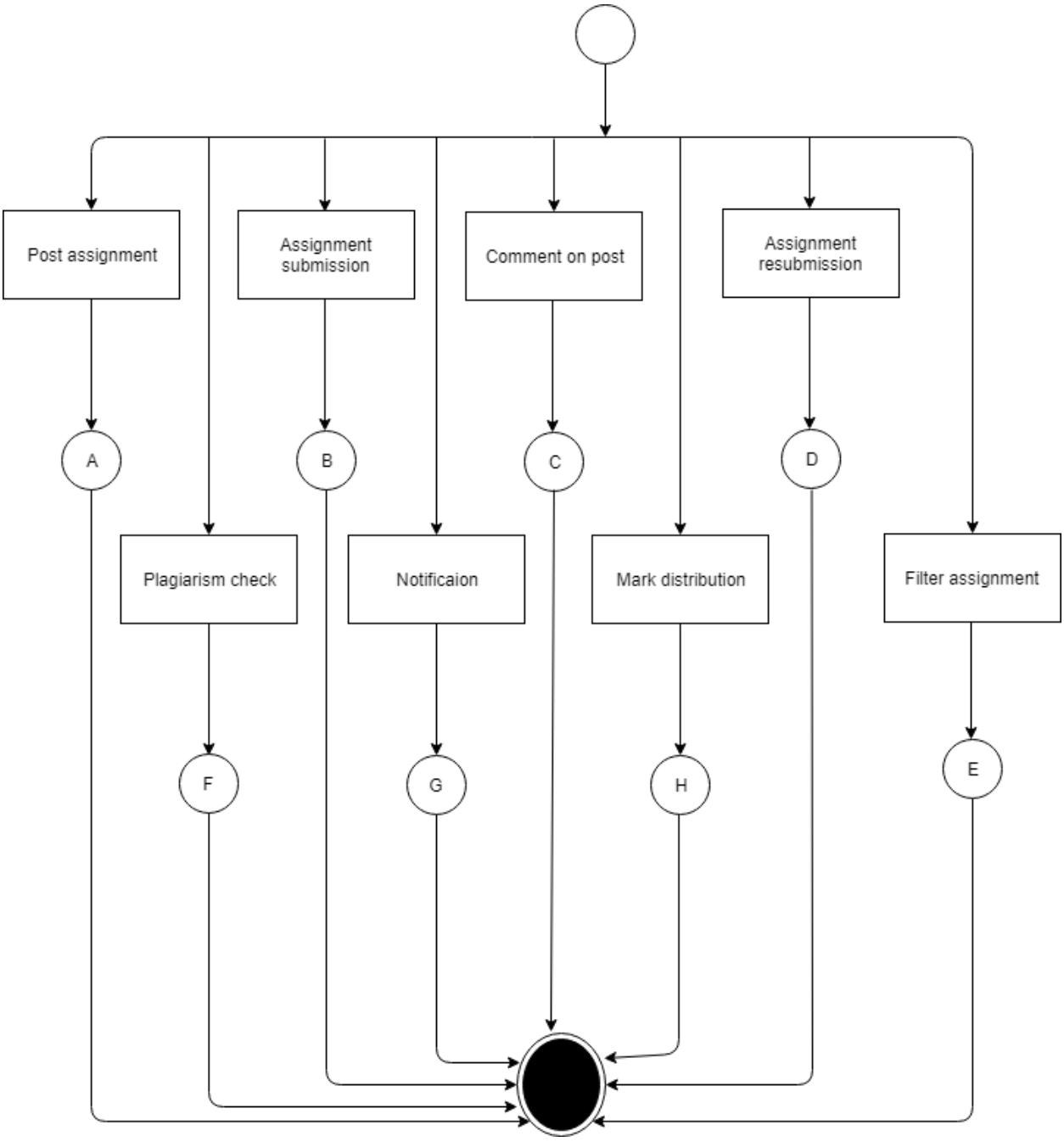


Figure – 10: Level 1.2 Activity diagram – Assignment management

ACTIVITY DIAGRAM – 2.1: ASSIGNMENT POST

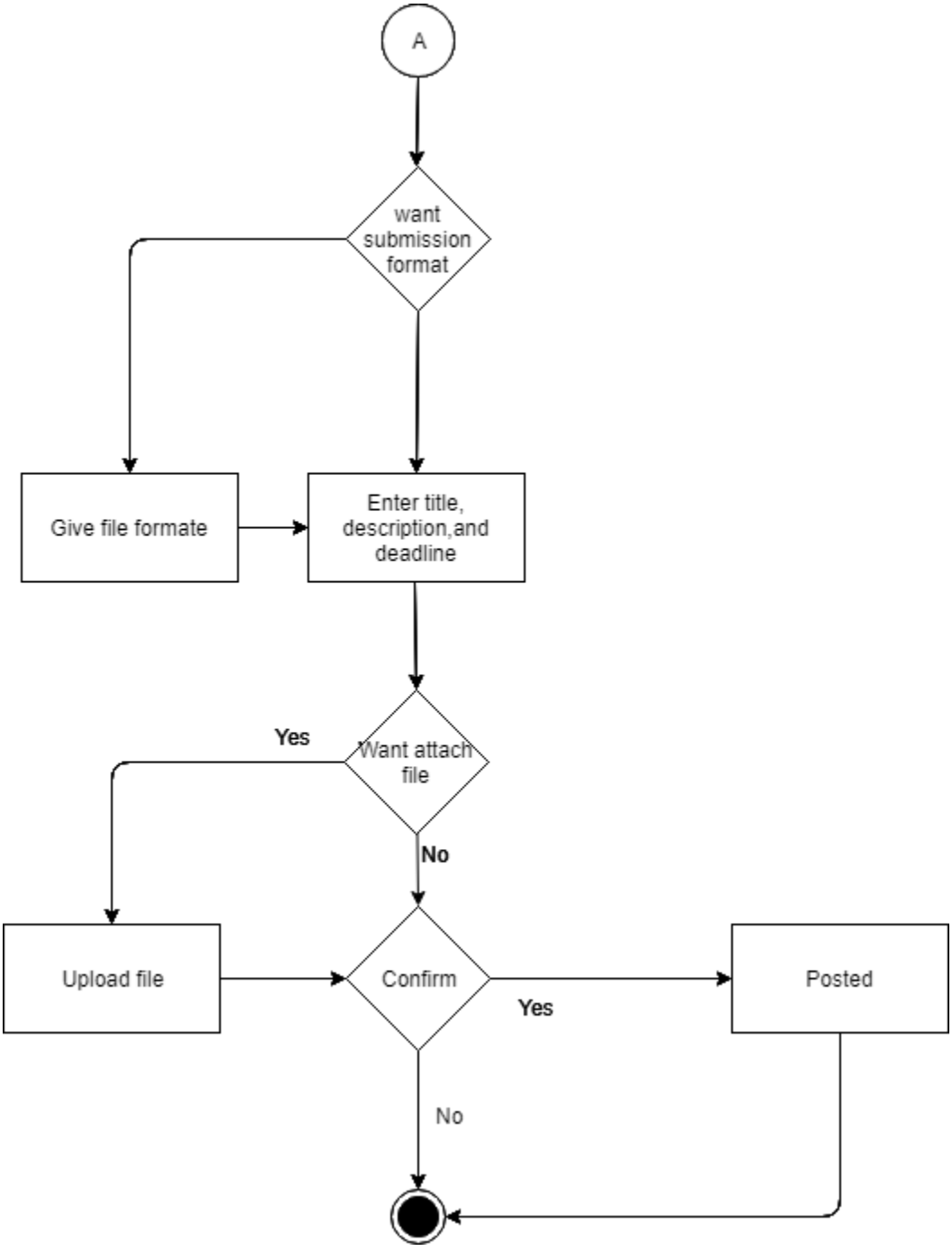


Figure – 11: Level 1.2.1 Activity diagram – Assignment post

ACTIVITY DIAGRAM – 2.2: ASSIGNMENT SUBMISSION

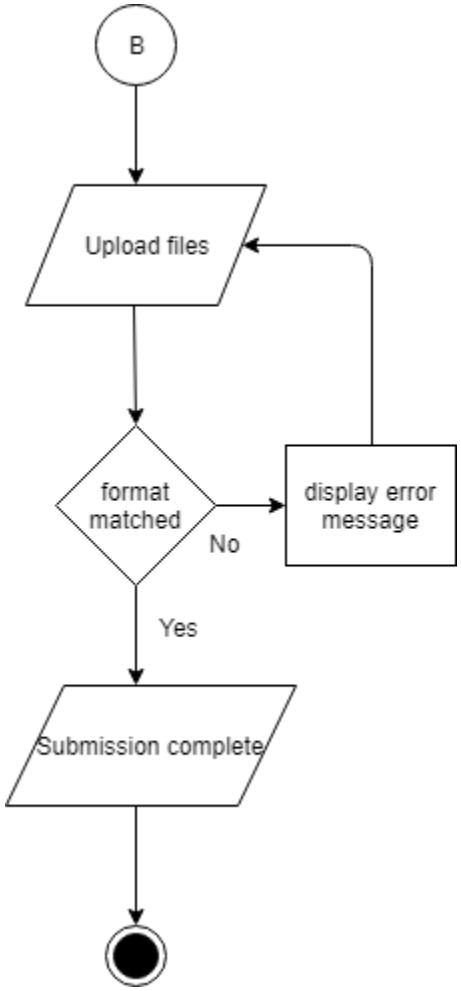


Figure – 12: Level 1.2.2 Activity diagram – Assignment submission

ACTIVITY DIAGRAM – 2.3: COMMENT

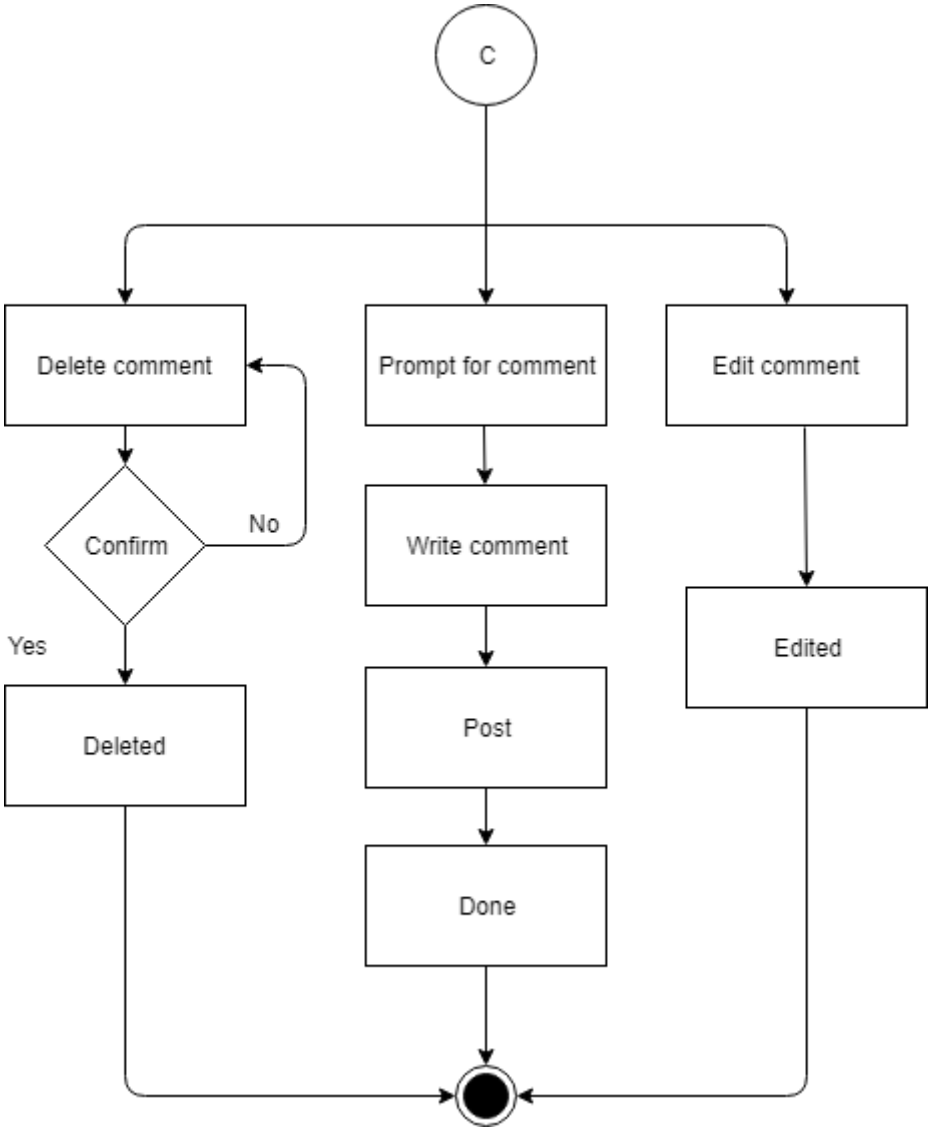


Figure – 13: Level 1.2.3 Activity diagram – Comment.

ACTIVITY DIAGRAM – 2.4: ASSIGNMENT RESUBMISSION

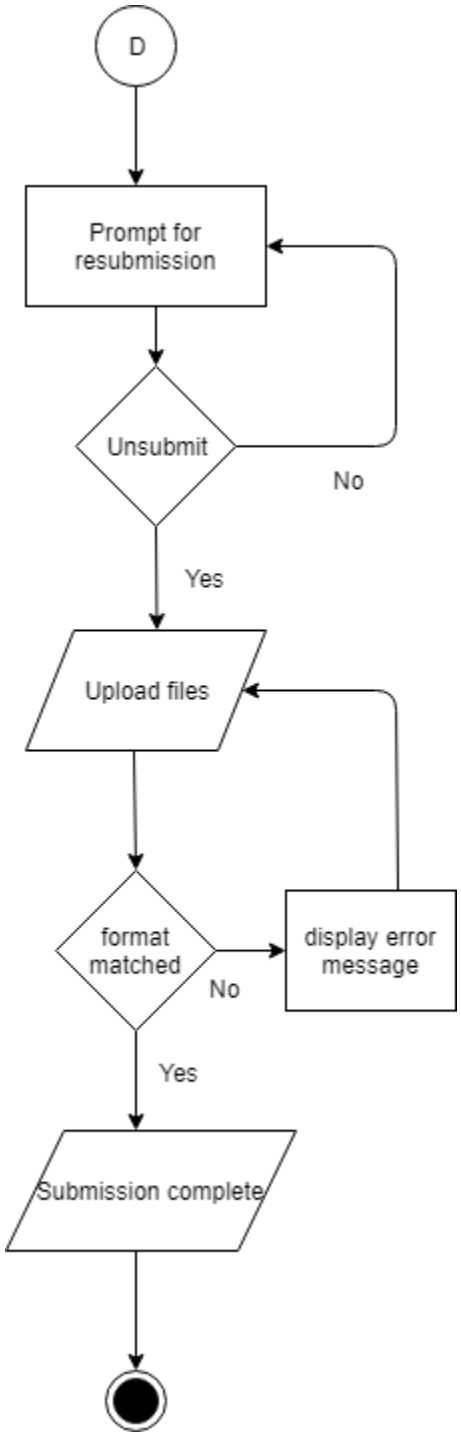


Figure – 14: Level 1.2.4 Activity diagram – Assignment resubmission.

ACTIVITY DIAGRAM – 2.5: FILTER ASSIGNMENT

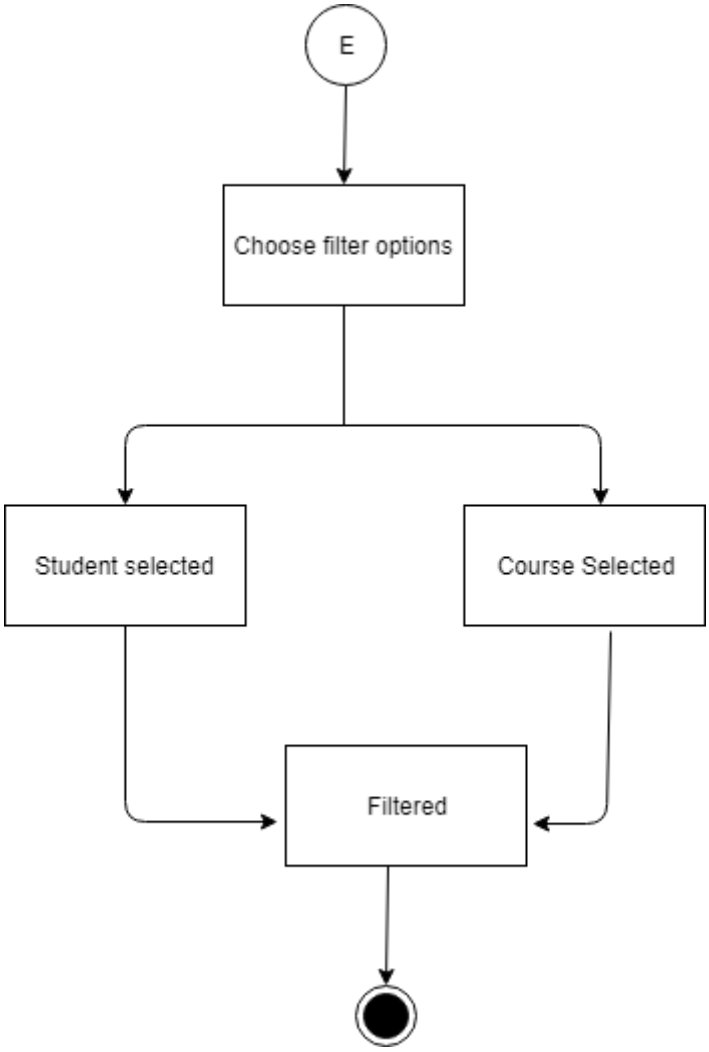


Figure – 15: Level 1.2.5 Activity diagram – Filter assignment

ACTIVITY DIAGRAM – 2.6: PLAGIARISM CHECK

ACTIVITY DIAGRAM – 2.7: NOTIFICATION

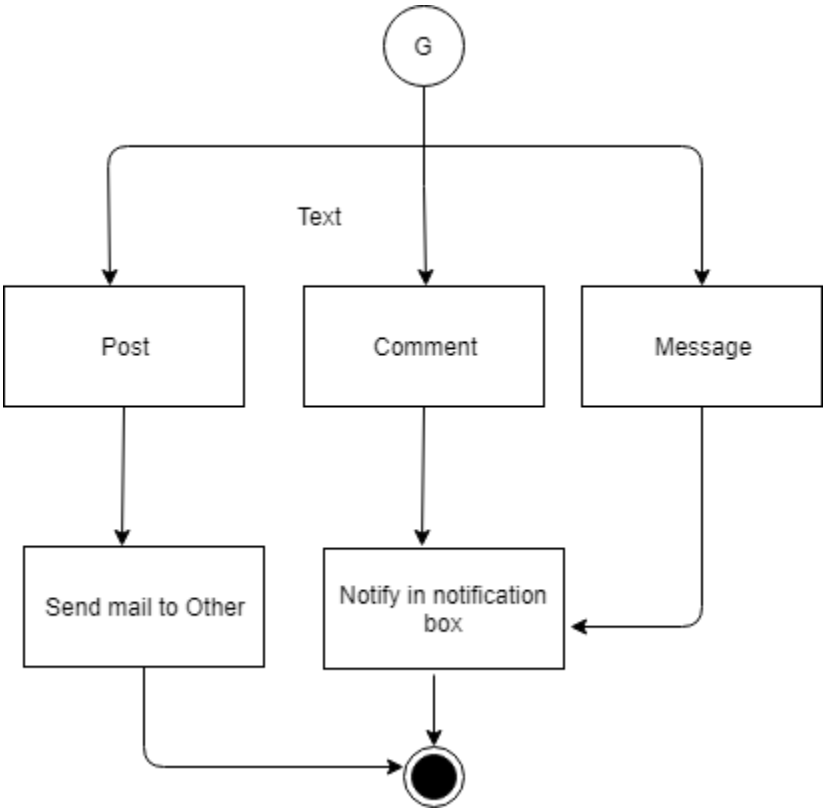


Figure – 17: Level 1.2.7 Activity diagram –Notification

ACTIVITY DIAGRAM – 2.8: MARK DISTRIBUTION

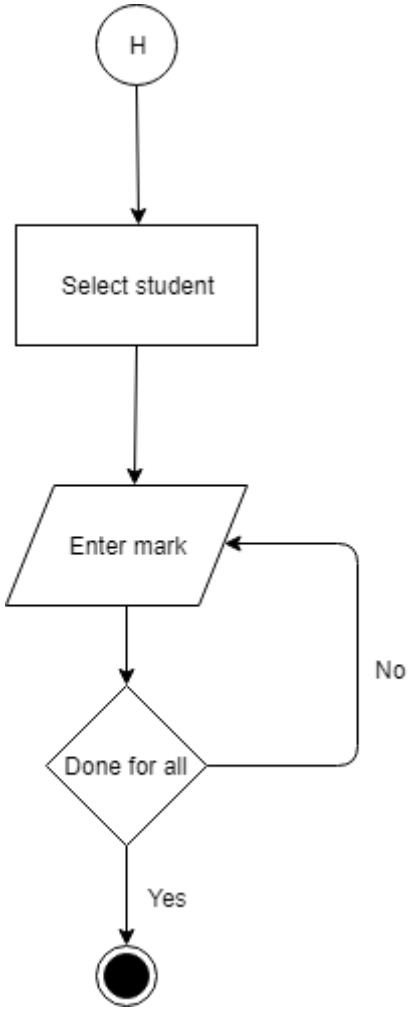


Figure – 18: Level 1.2.8 Activity diagram – Mark distribution.

ACTIVITY DIAGRAM – 3: GROUP MANAGEMENT

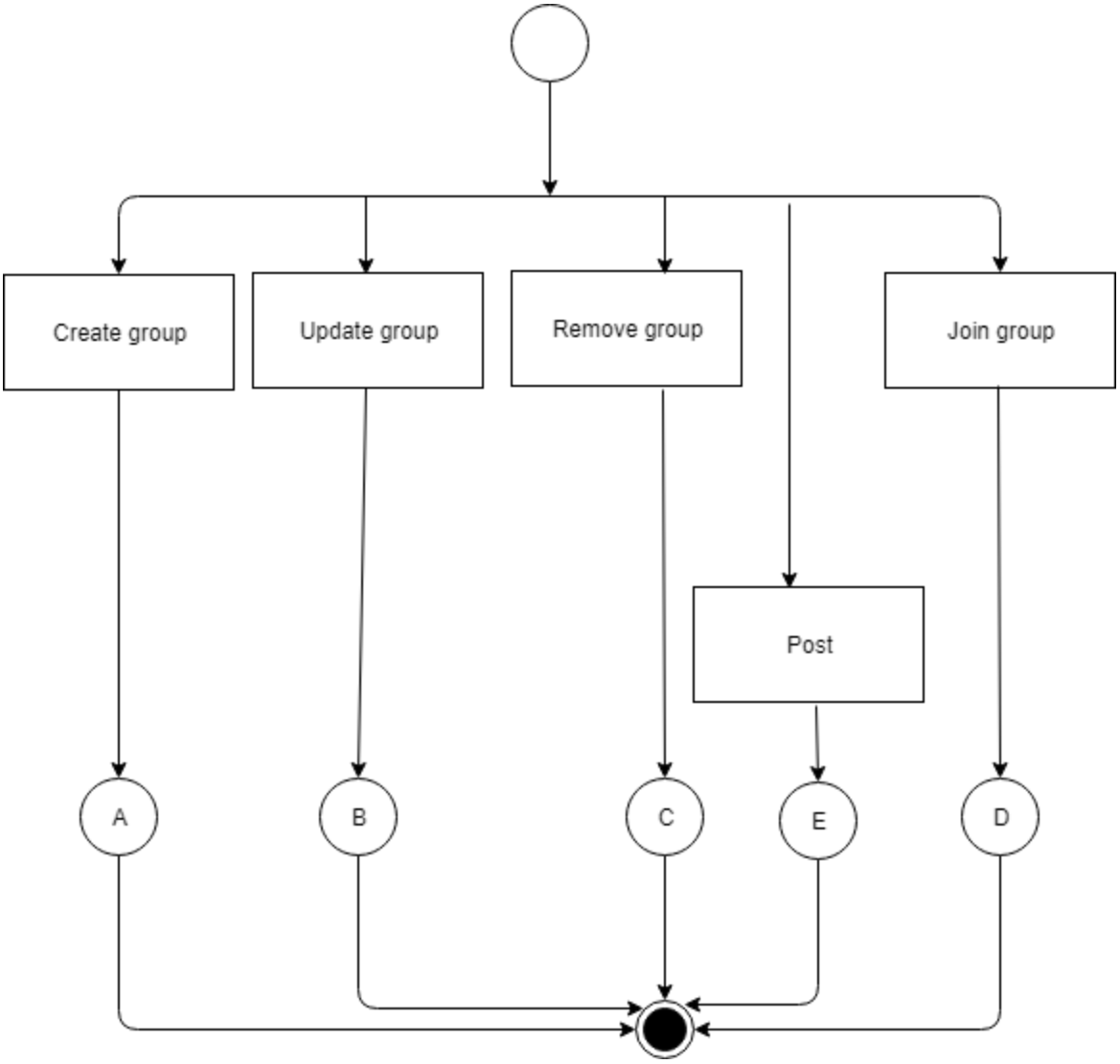


Figure – 19: Level 1.3 Activity diagram – Group management.

ACTIVITY DIAGRAM – 3.1: CREATE GROUP

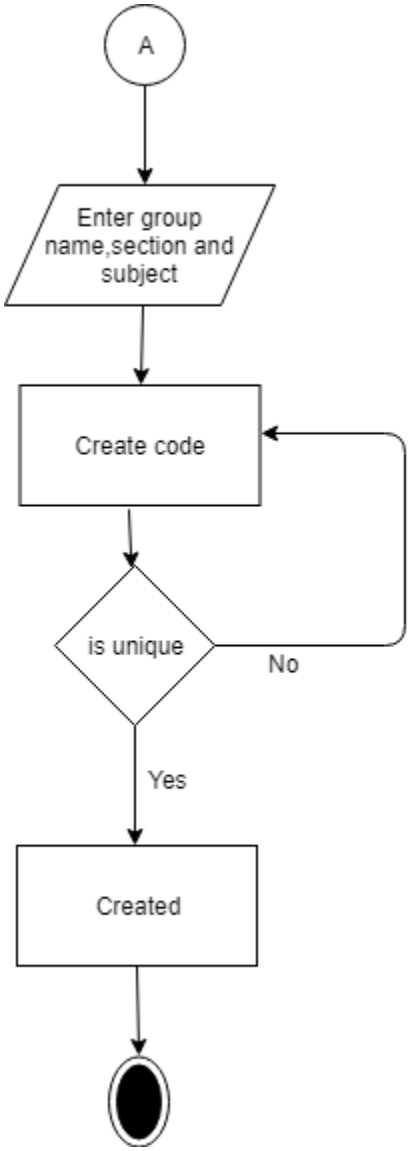


Figure – 20: Level 1.3.1 Activity diagram – Create group

ACTIVITY DIAGRAM – 3.2: JOIN GROUP

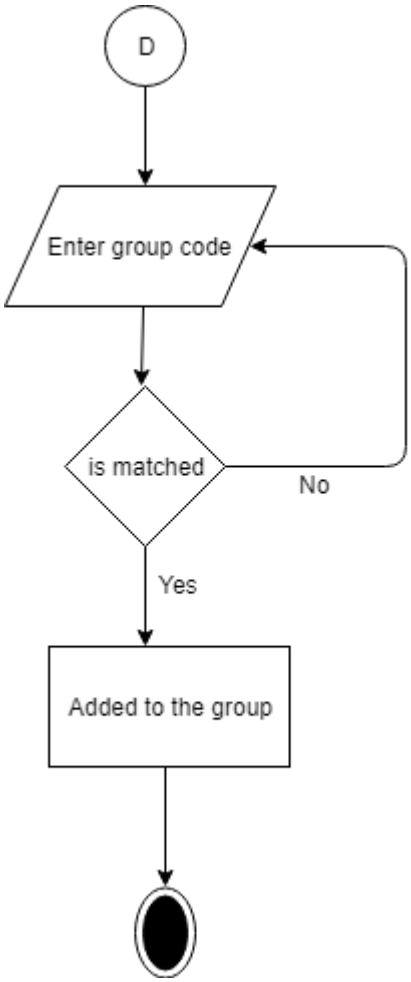


Figure – 21: Level 1.3.2 Activity diagram – Join group

ACTIVITY DIAGRAM – 3.3: UPDATE GROUP

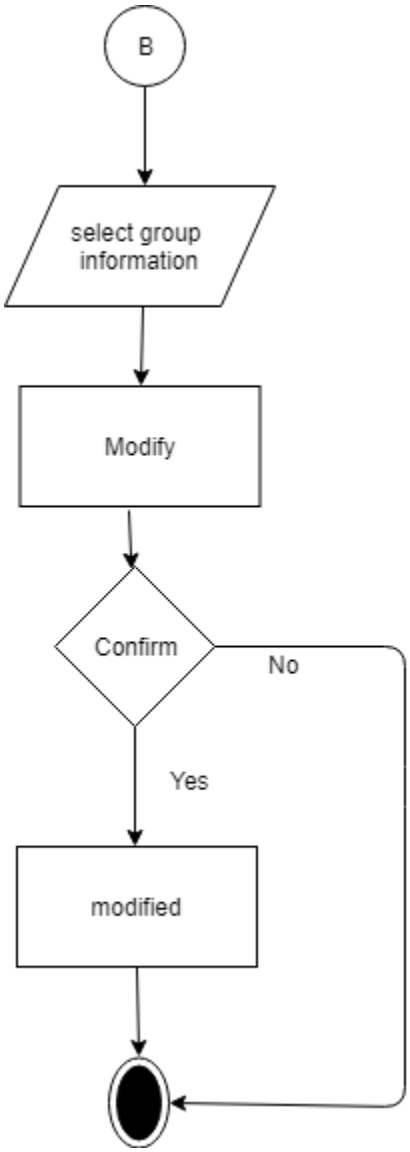


Figure – 22: Level 1.3.3 Activity diagram – Update group.

ACTIVITY DIAGRAM – 3.4: REMOVE GROUP

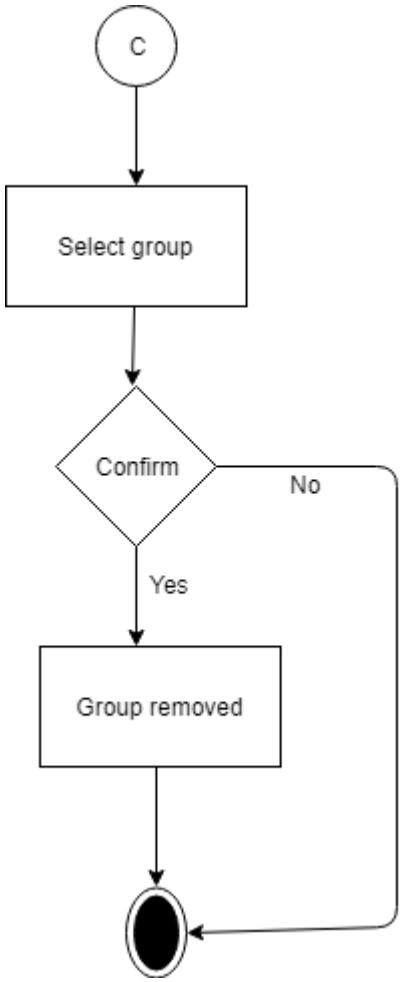


Figure – 23: Level 1.3.4 Activity diagram – Remove group.

ACTIVITY DIAGRAM – 3.5: POST

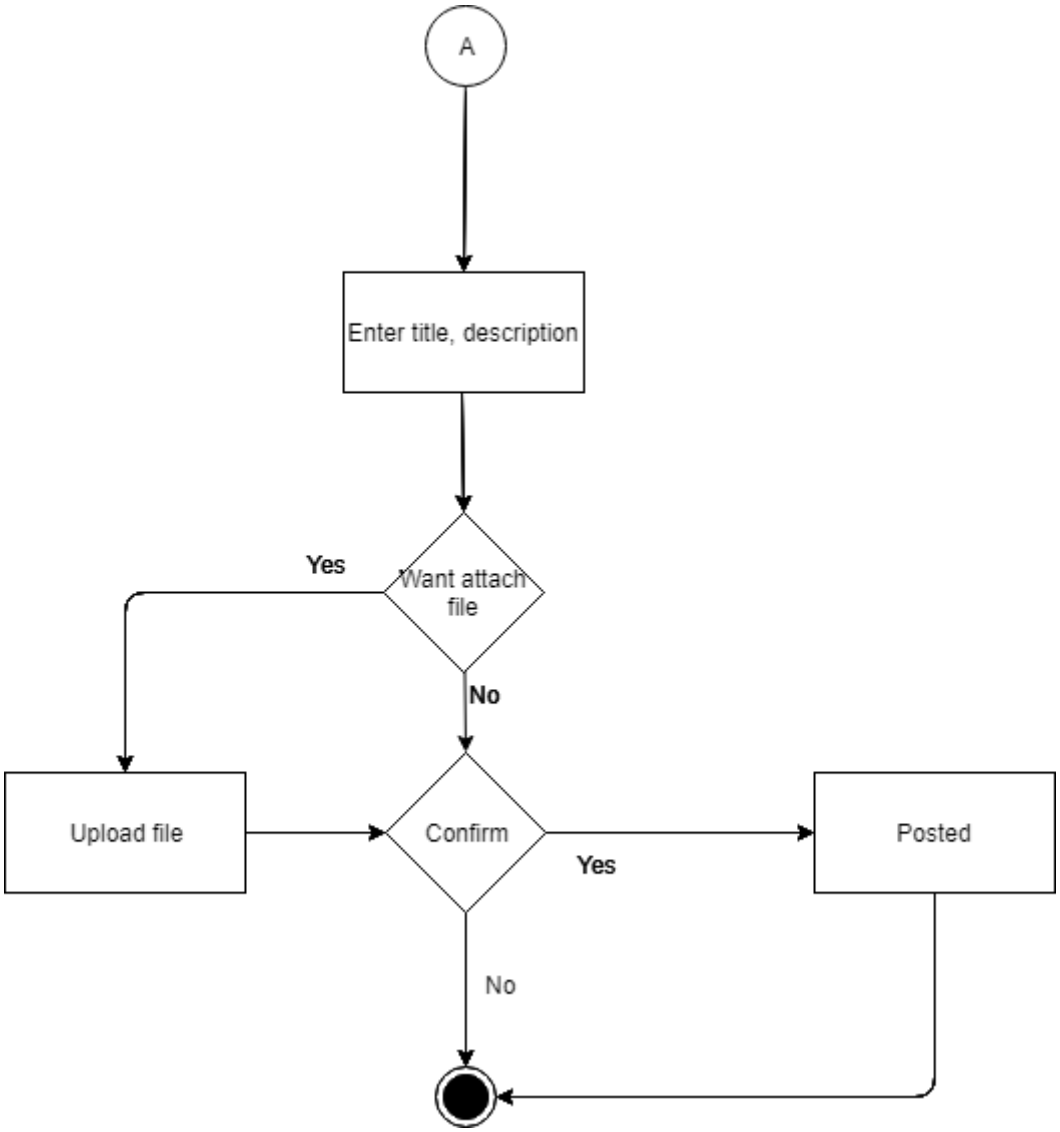


Figure – 24: Level 1.3.5 Activity diagram – Post.

4.4 SWIM LANE DIAGRAMS

SWIM LANE DIAGRAM – 1: AUTHENTICATION

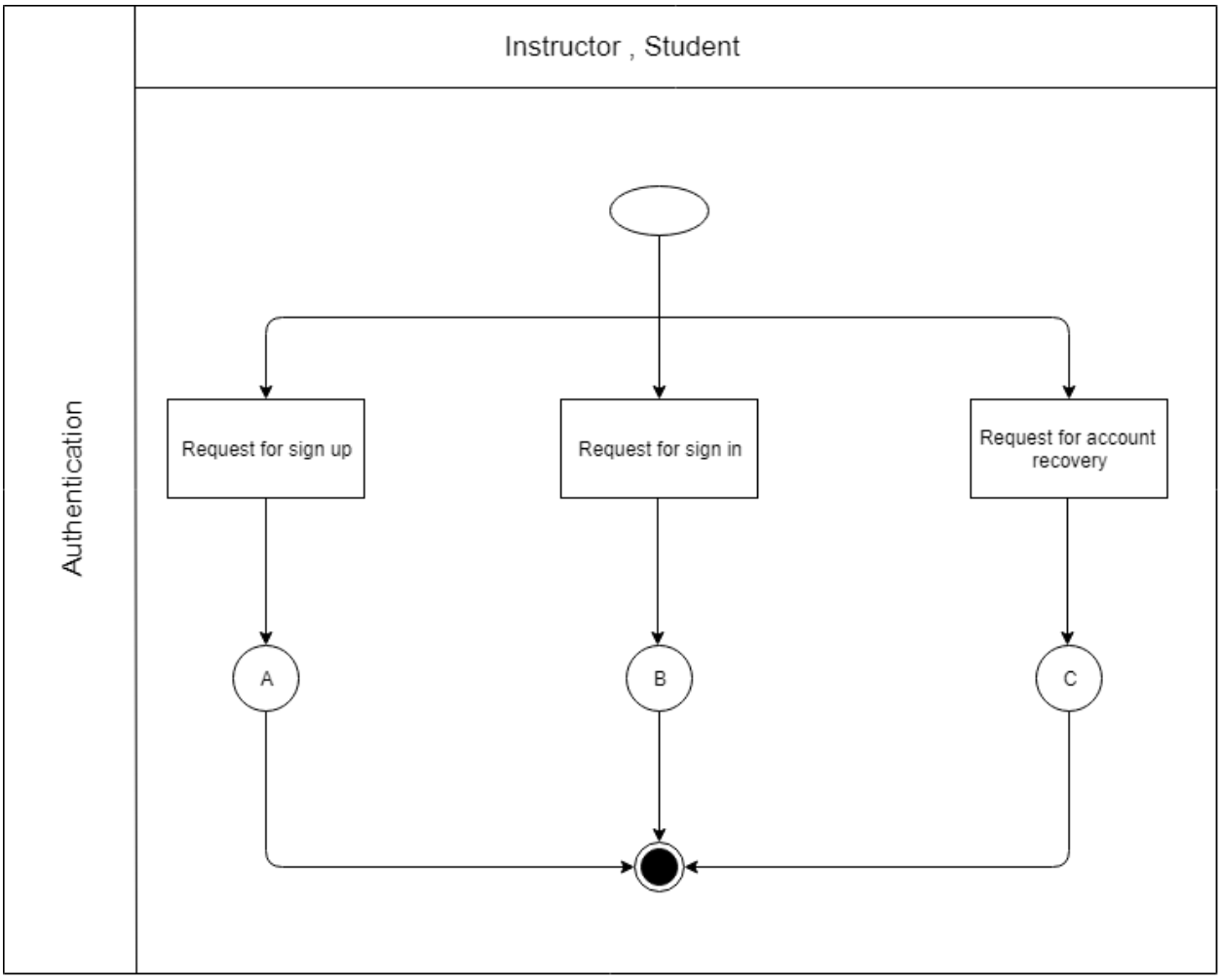


Figure – 25: Level 1.1 Swim lane diagram – Authentication.

SWIM LANE DIAGRAM – 1.1: SIGN UP

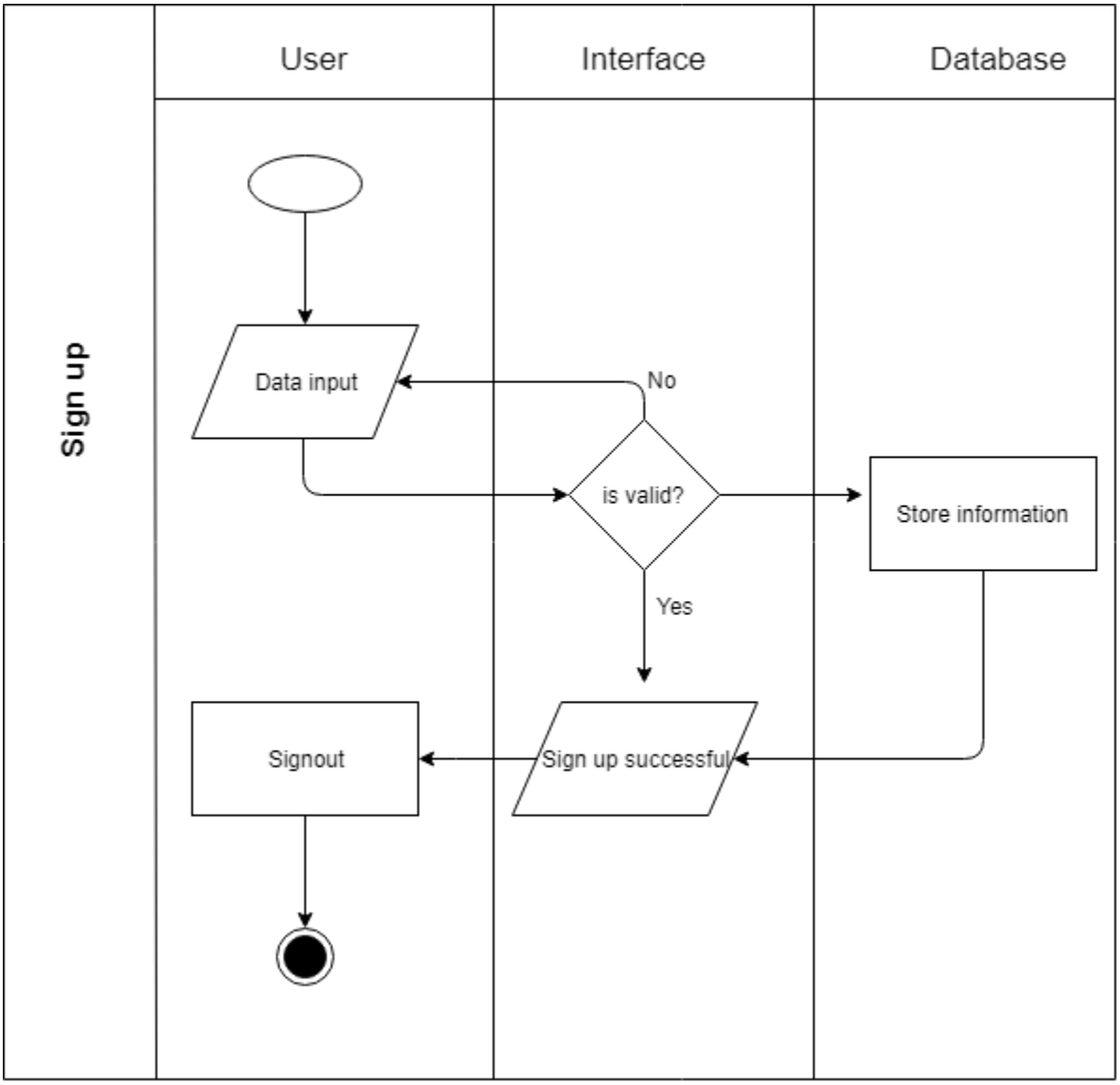


Figure – 26: Level 1.1.1 Swim lane diagram – Sign up.

SWIM LANE DIAGRAM – 1.1: SIGN IN

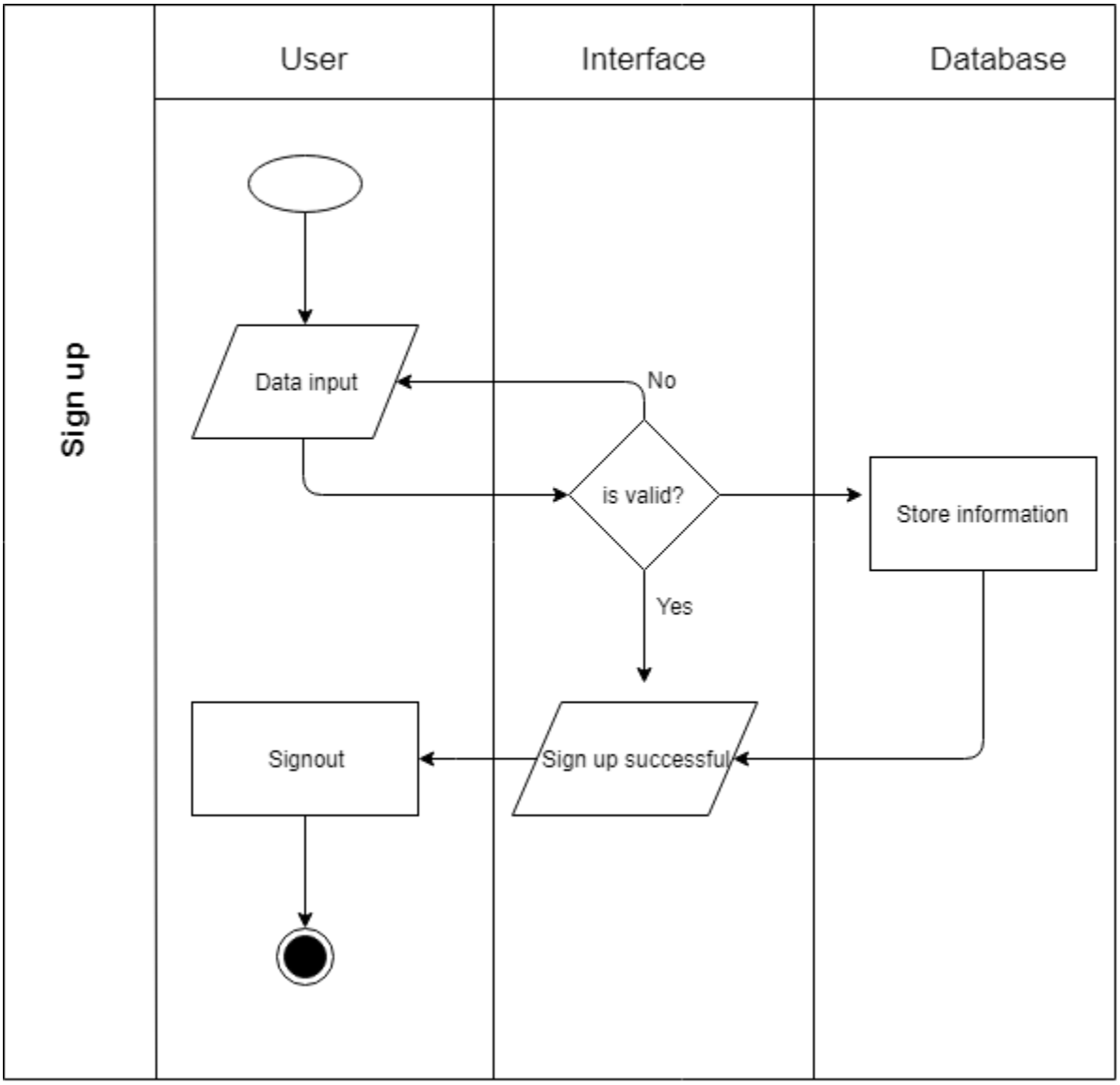


Figure – 27: Level 1.1.2 Swim lane diagram – Sign in.

SWIM LANE DIAGRAM – 1.1: ACCOUNT RECOVERY

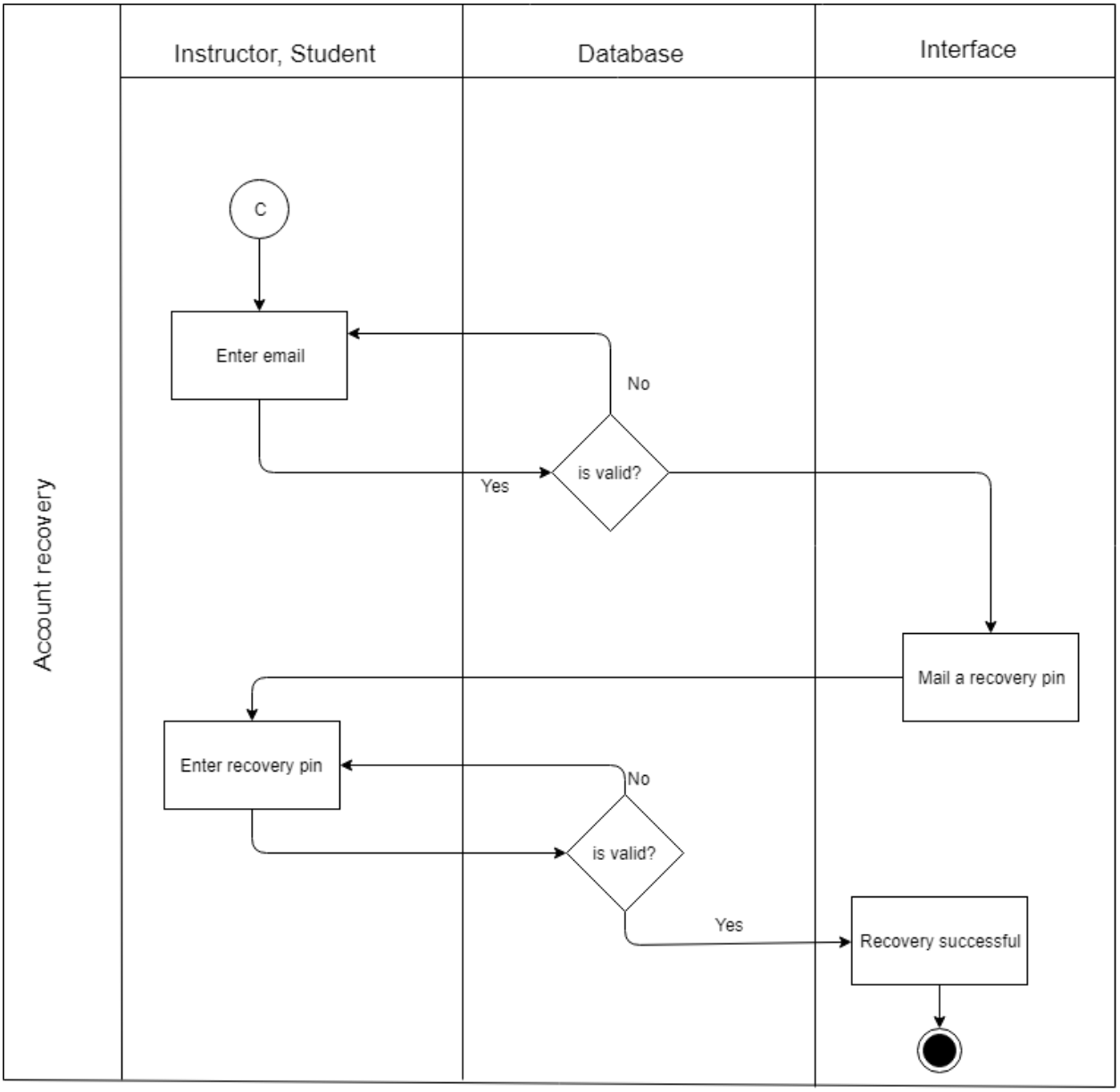


Figure – 27: Level 1.1.3 Swim lane diagram – Account recovery

SWIM LANE DIAGRAM – 2: ASSIGNMENT MANAGEMENT

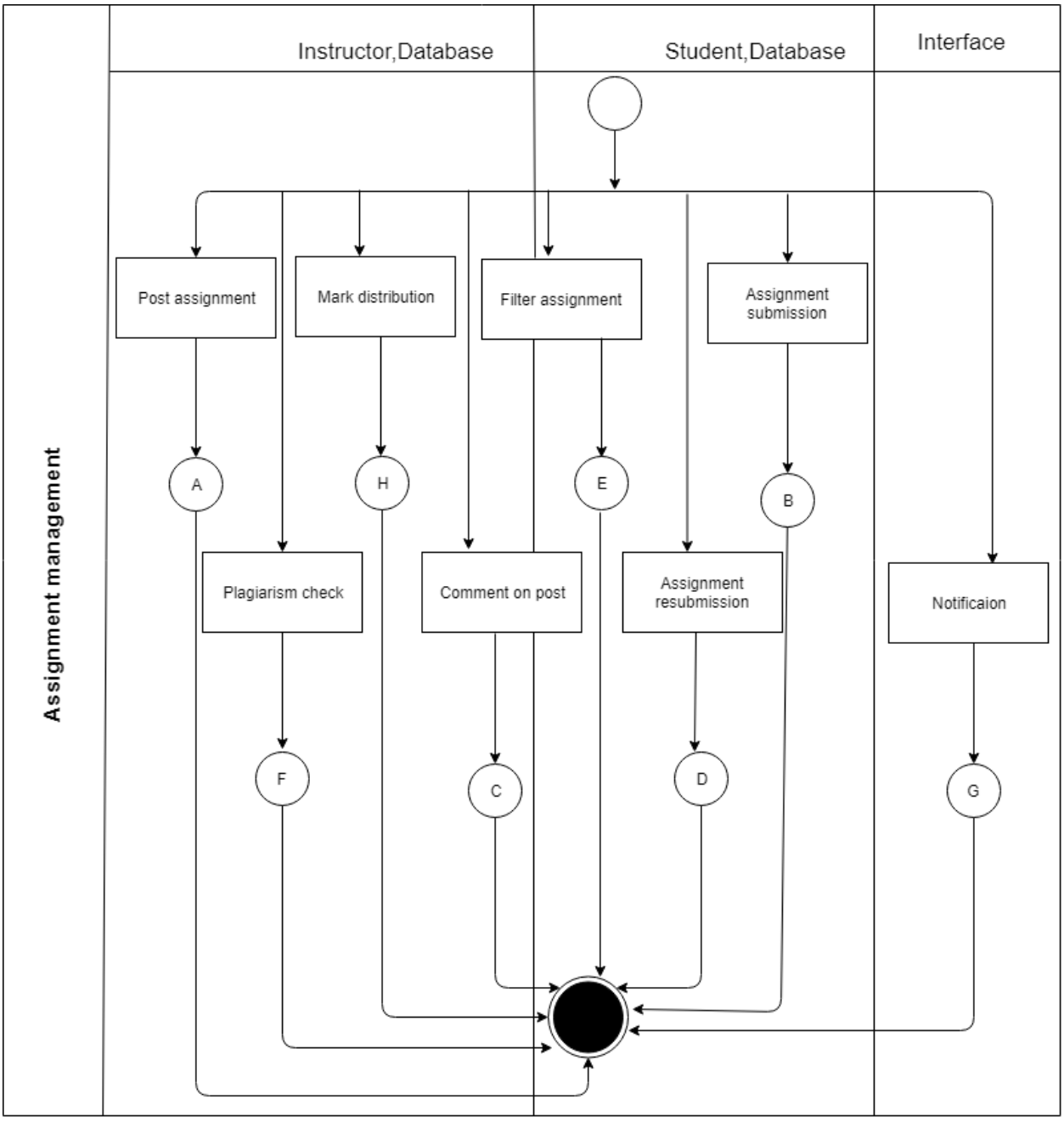


Figure – 28: Level 1.2 Swim lane diagram – Assignment management

SWIM LANE DIAGRAM – 2.1: ASSIGNMENT POST

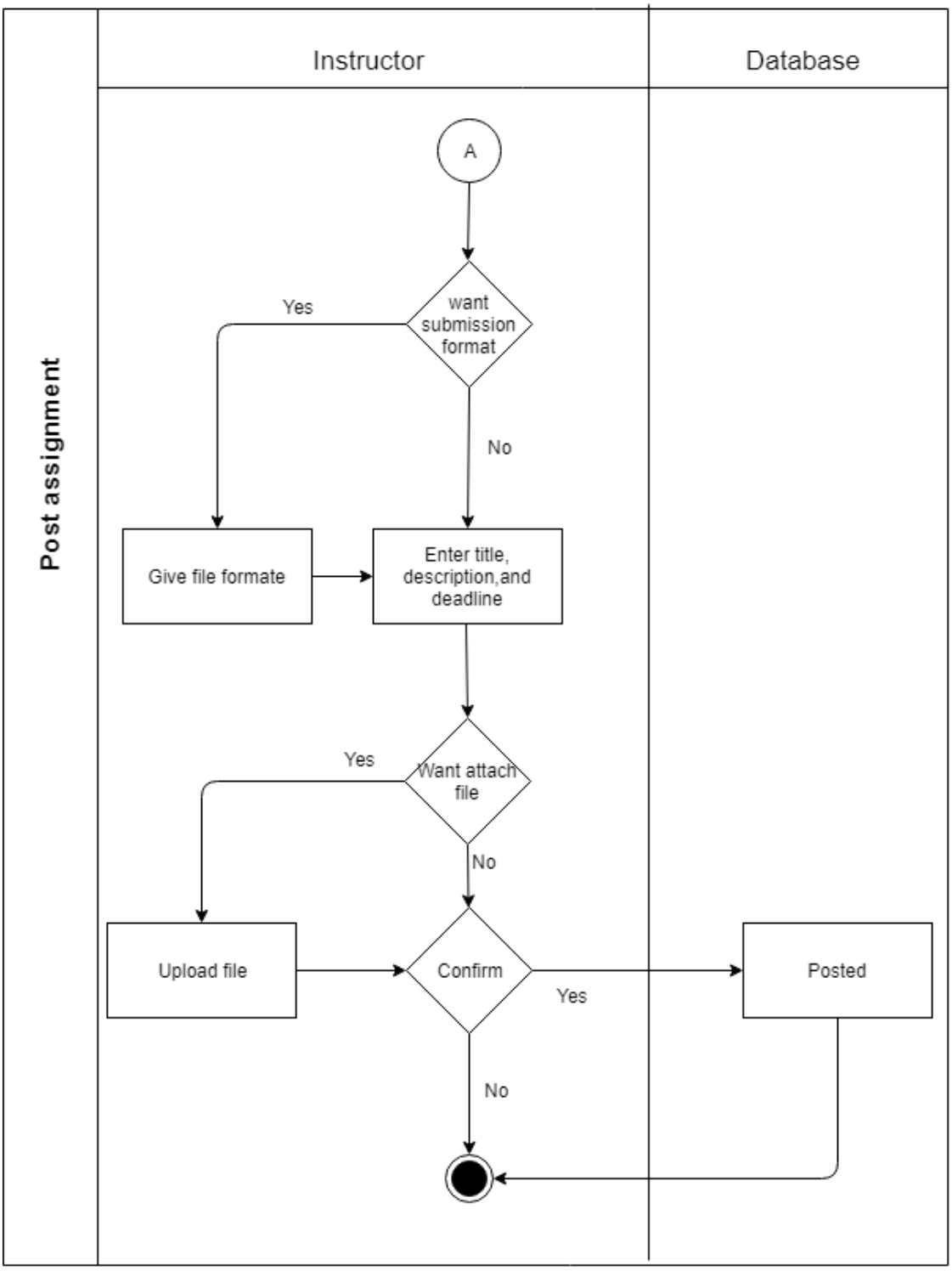


Figure – 29: Level 1.2.1 Swim lane diagram – Assignment post.

SWIM LANE DIAGRAM – 2.2: ASSIGNMENT SUBMISSION

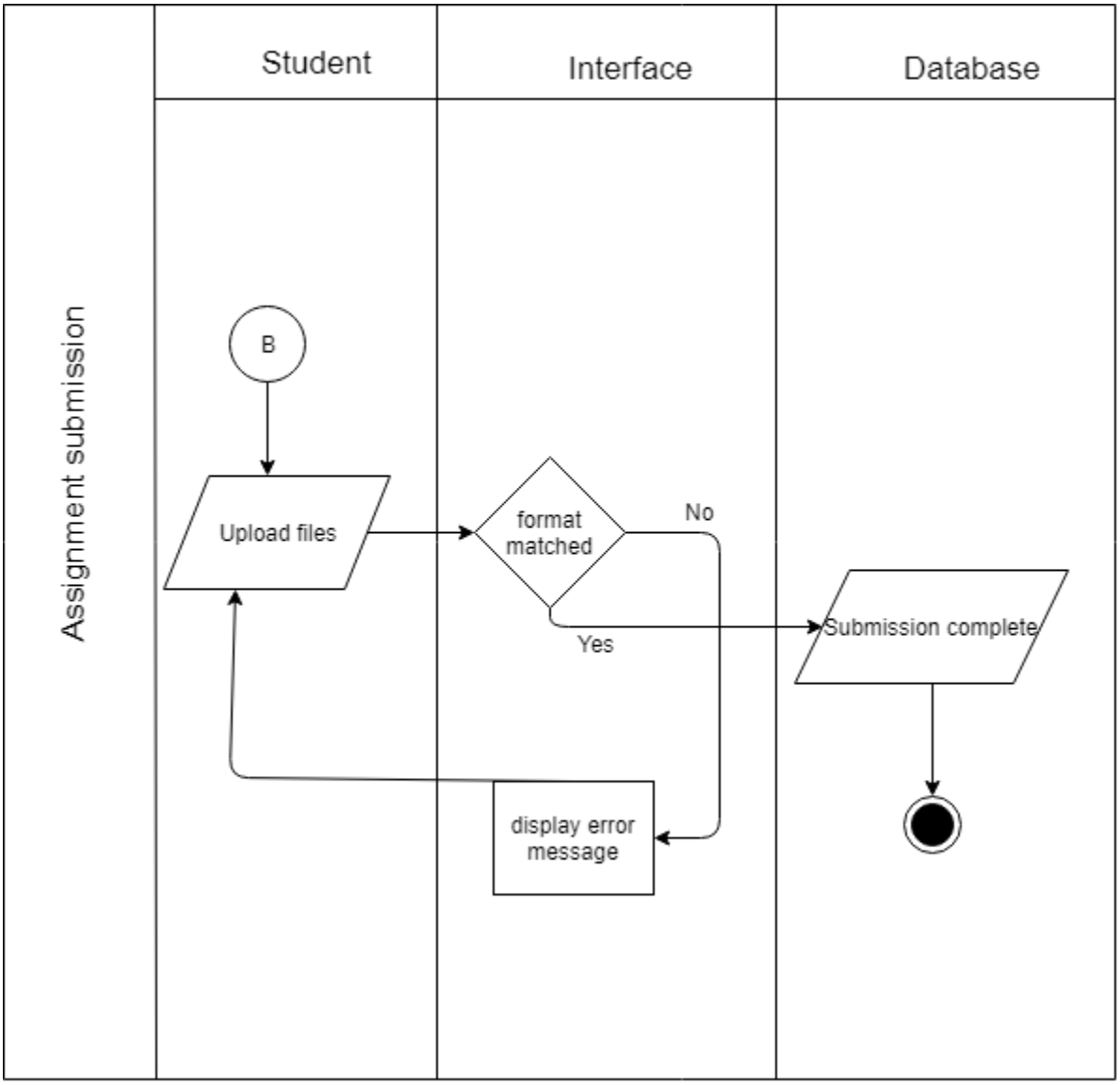


Figure – 30: Level 1.2.2 Swim lane diagram – Assignment submission.

SWIM LANE DIAGRAM – 2.3: COMMENT

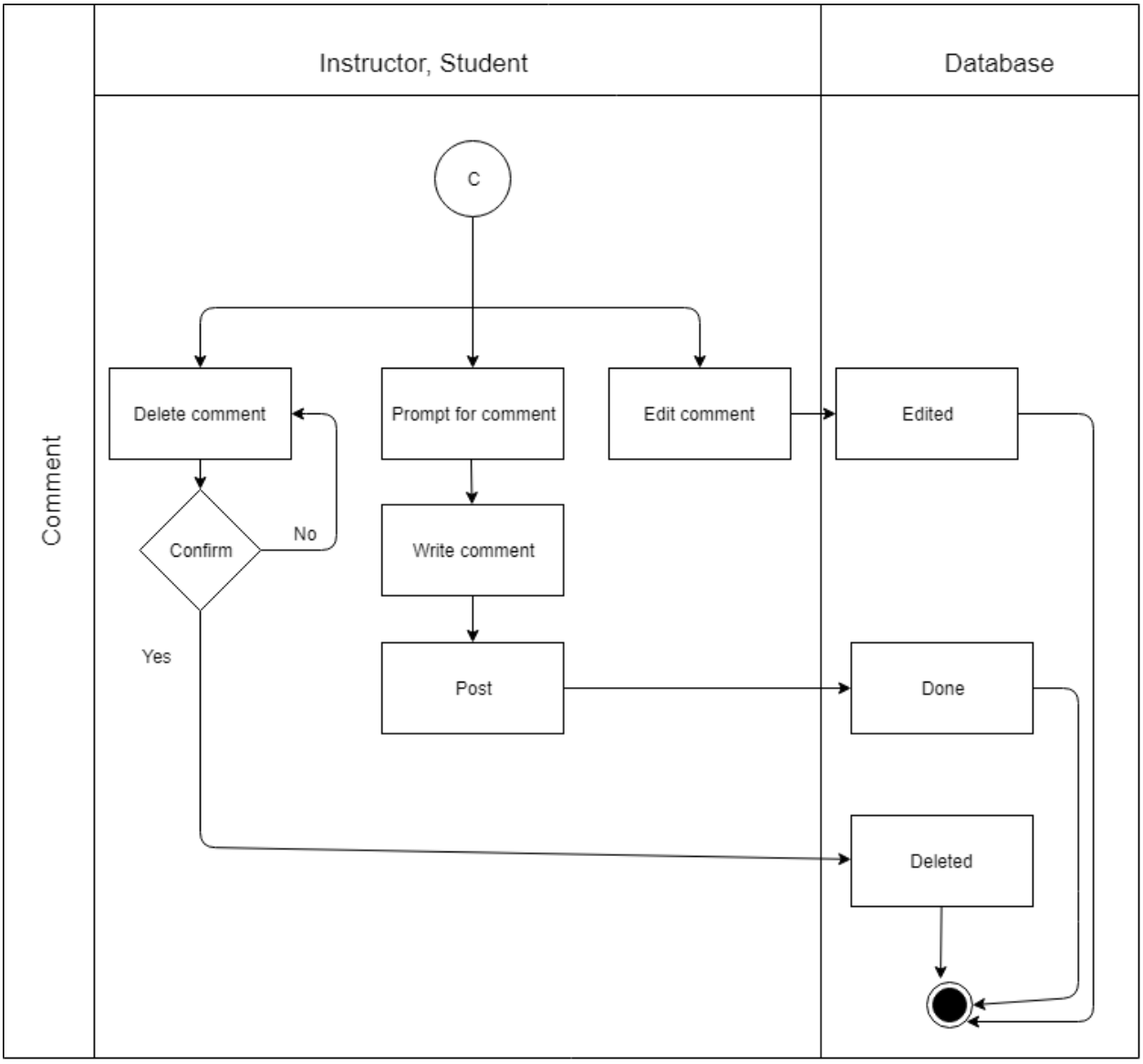


Figure – 31: Level 1.2.3 Swim lane diagram – Comment.

SWIM LANE DIAGRAM – 2.4: ASSIGNMENT RESUBMISSION

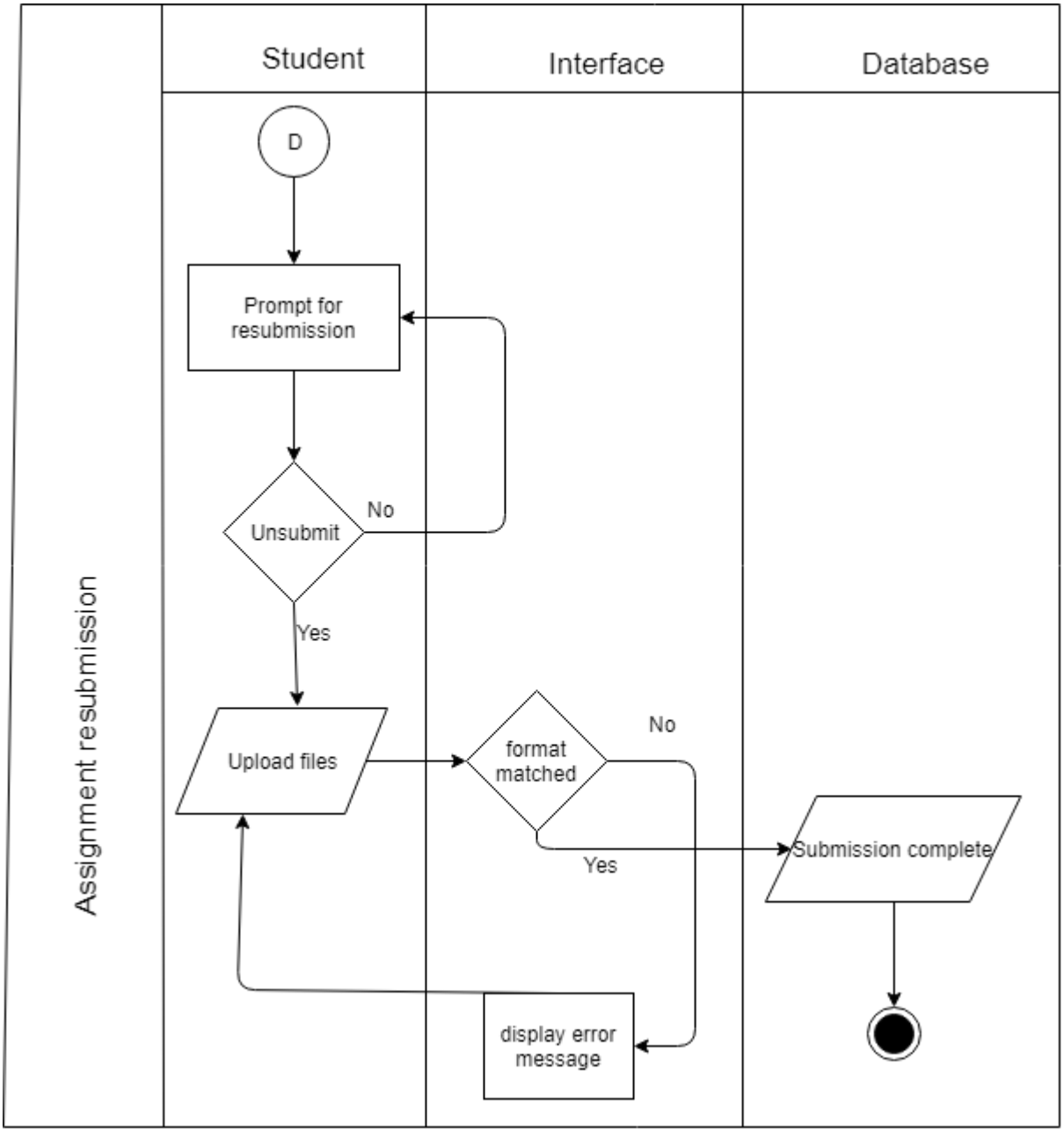


Figure – 32: Level 1.2.4 Swim lane diagram – Assignment resubmission.

SWIM LANE DIAGRAM – 2.5: FILTER ASSIGNMENT

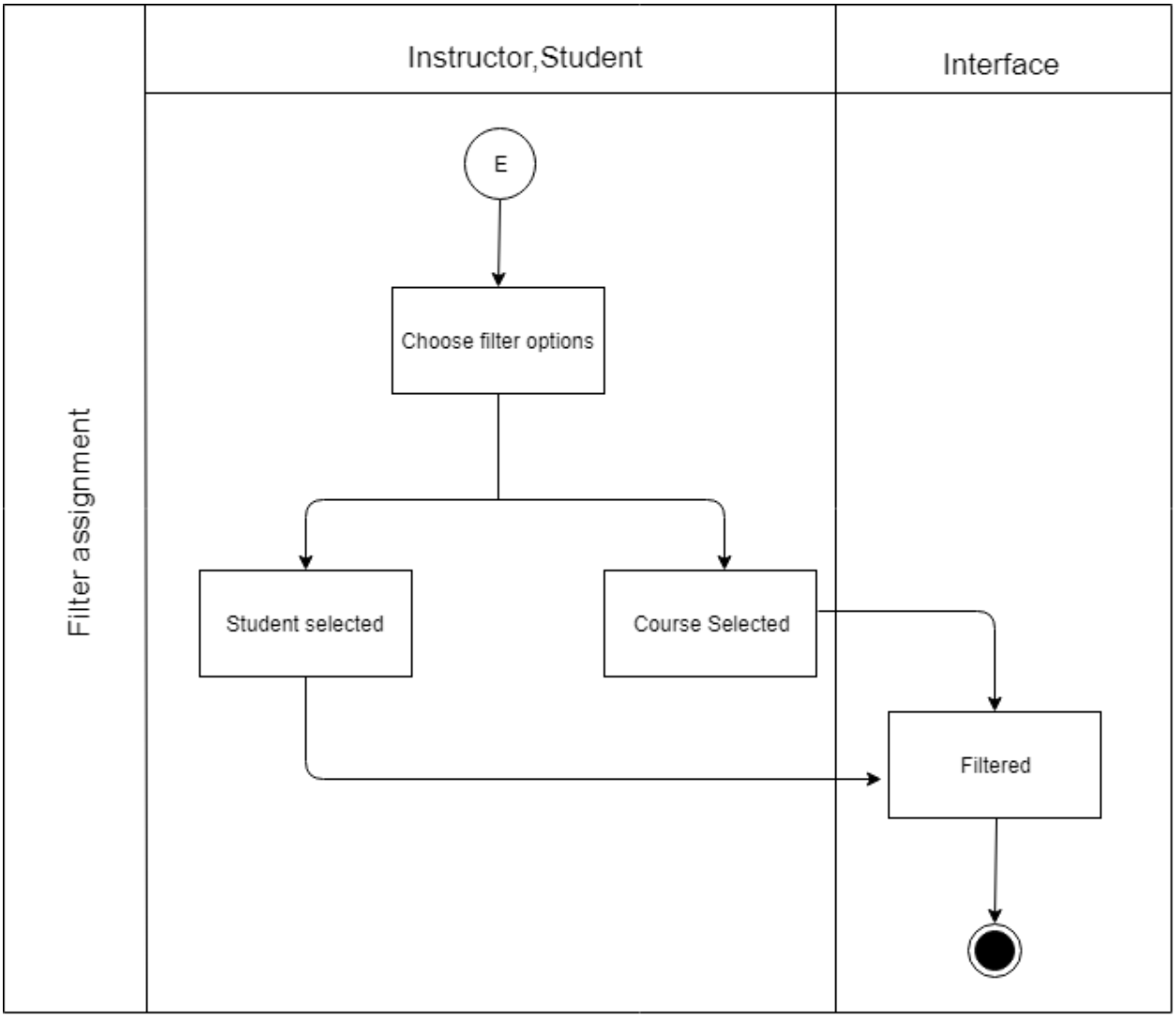


Figure – 33: Level 1.2.5 Swim lane diagram – Filter assignment.

SWIM LANE DIAGRAM – 2.6: PLAGIARISM CHECK

Figure – 34: Level 1.1.2 Swim lane diagram – Sign in.

SWIM LANE DIAGRAM – 2.7: NOTIFICATION

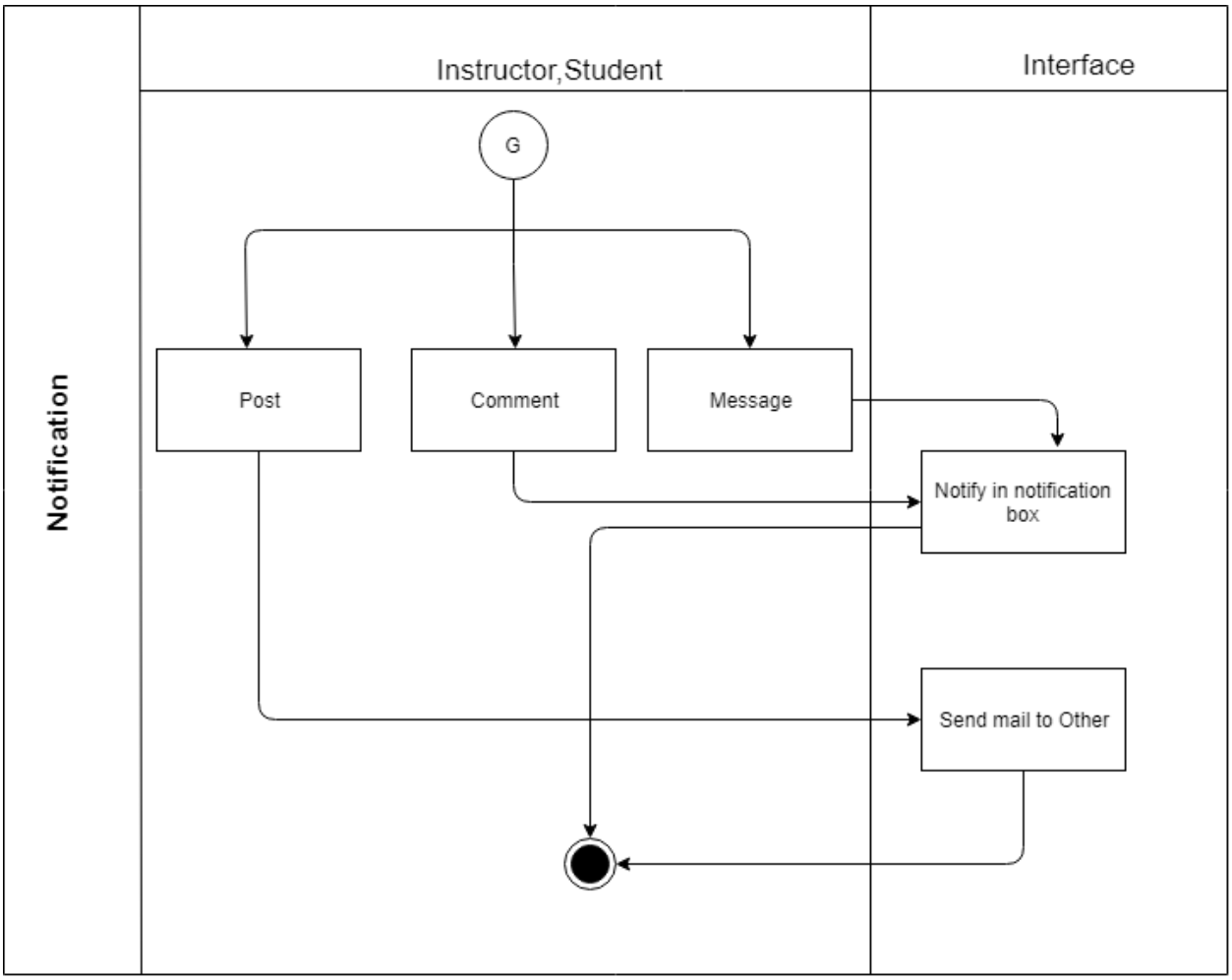


Figure – 35: Level 1.2.7 Swim lane diagram – Notification.

SWIM LANE DIAGRAM – 2.8: MARK DISTRIBUTION

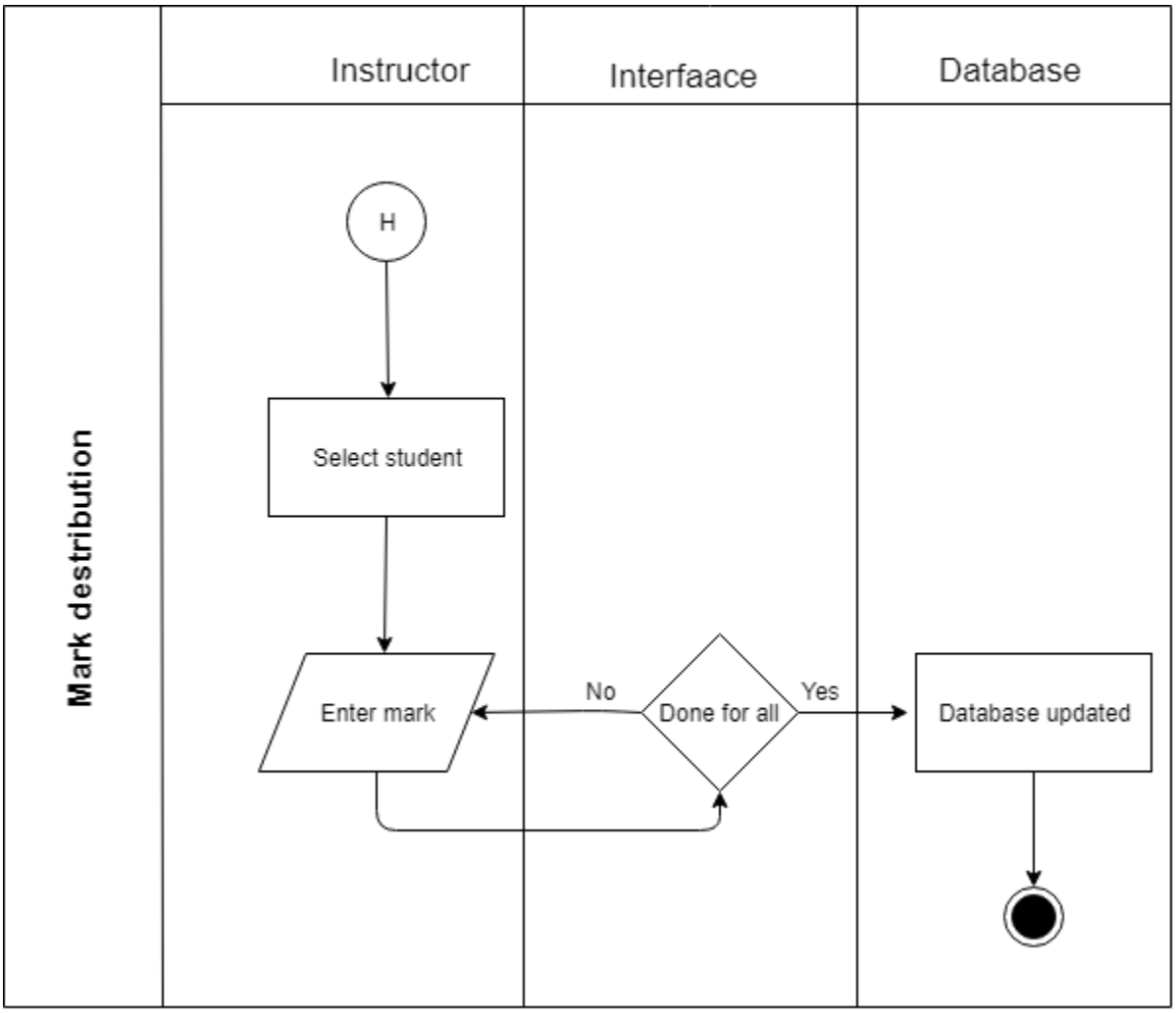


Figure – 36: Level 1.2.8 Swim lane diagram – Mark distribution

SWIM LANE DIAGRAM – 3: GROUP MANAGEMENT

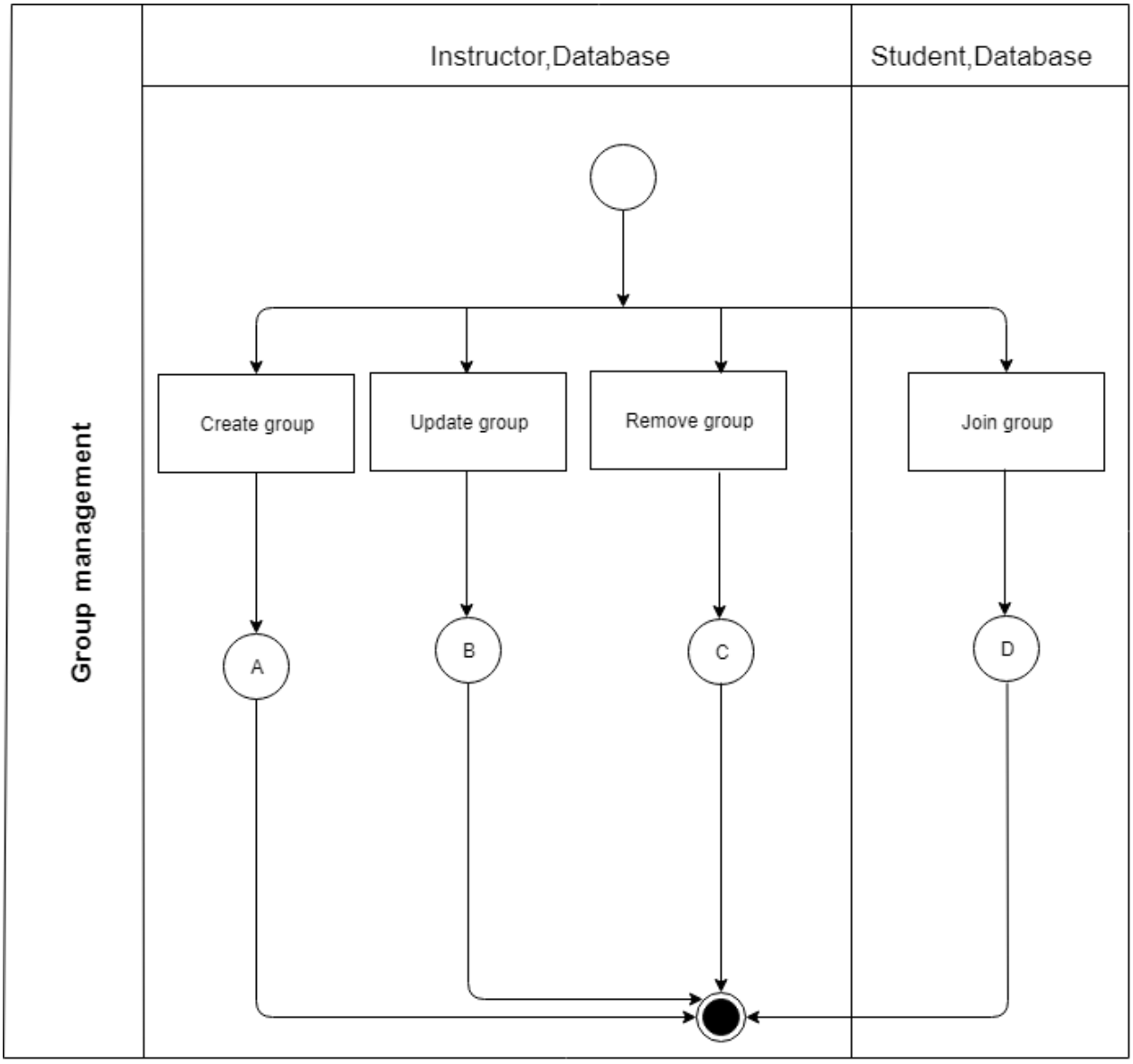


Figure – 37: Level 1.3 Swim lane diagram – Group management

SWIM LANE DIAGRAM – 3.1: CREATE GROUP

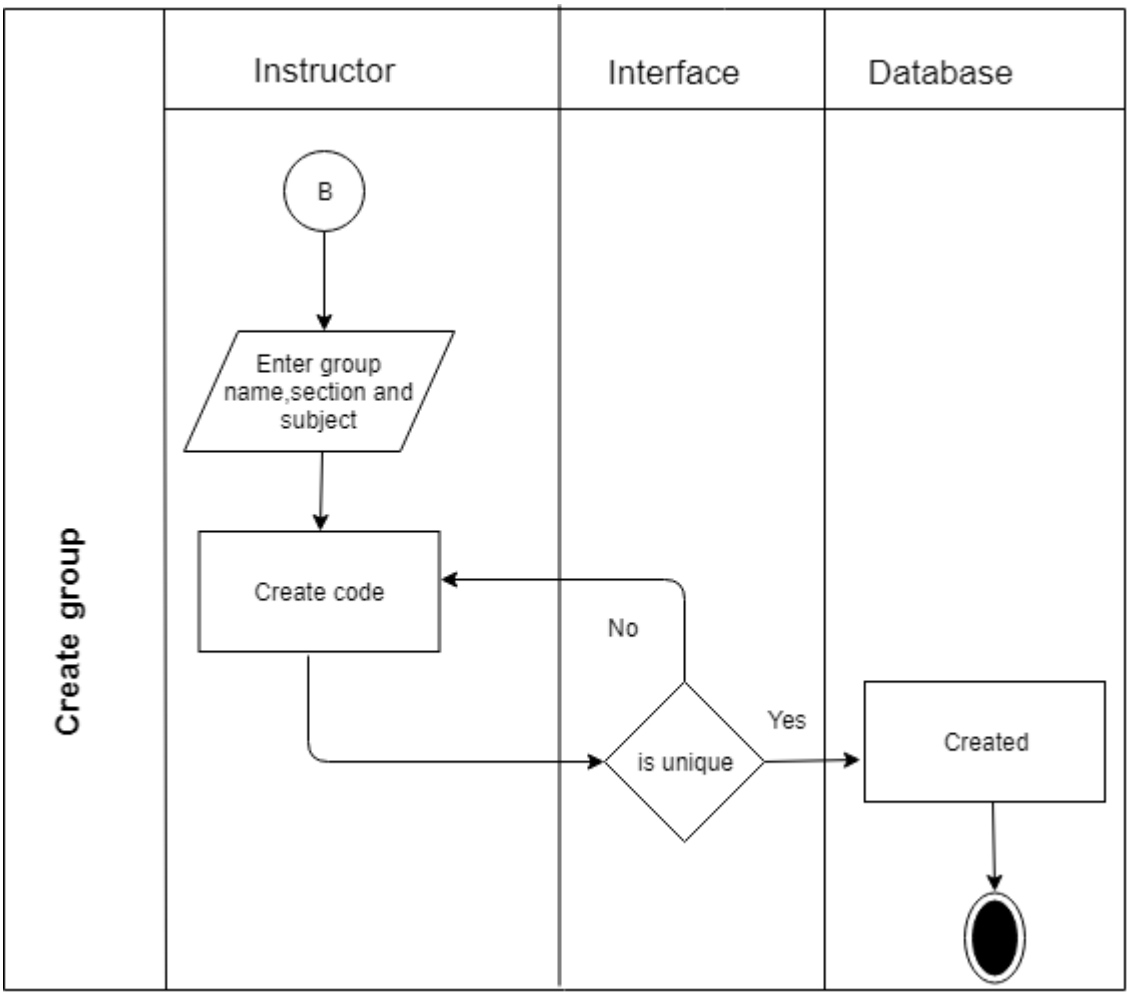


Figure – 38: Level 1.3.1 Swim lane diagram – Create group

SWIM LANE DIAGRAM – 3.2: JOIN GROUP

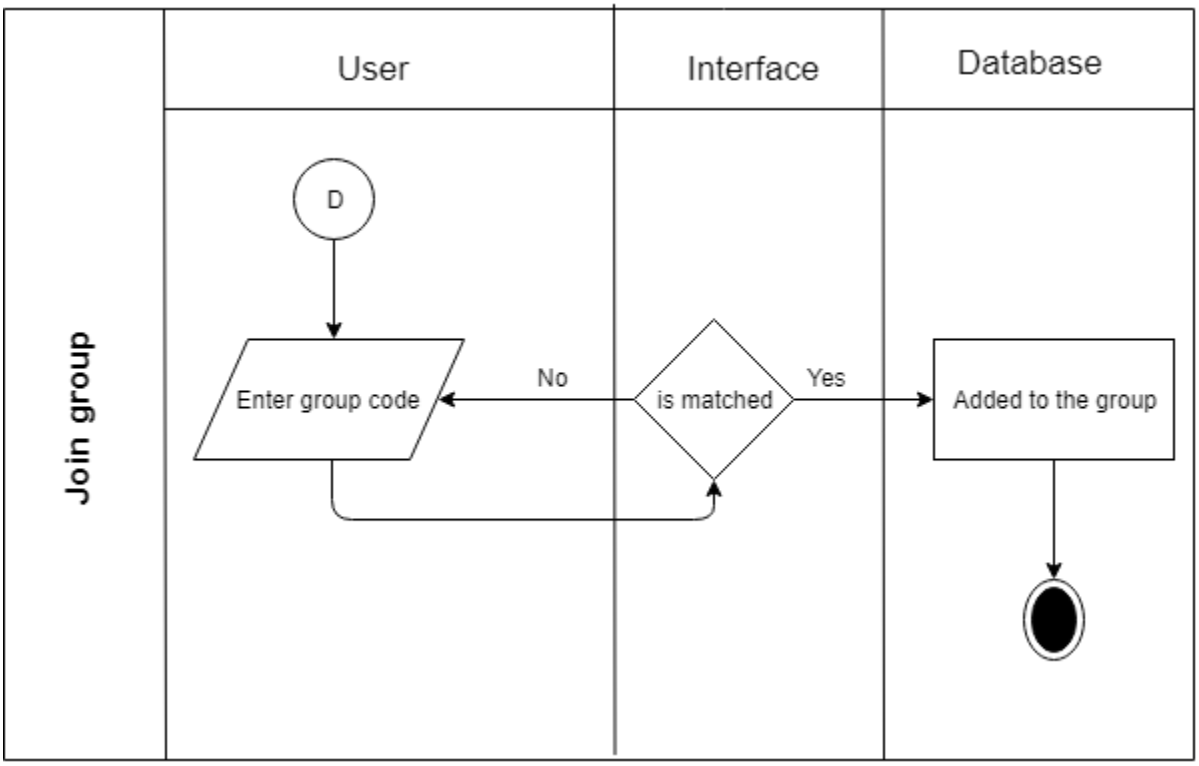


Figure – 39: Level 1.3.2 Swim lane diagram – Join group

SWIM LANE DIAGRAM – 3.3: UPDATE GROUP

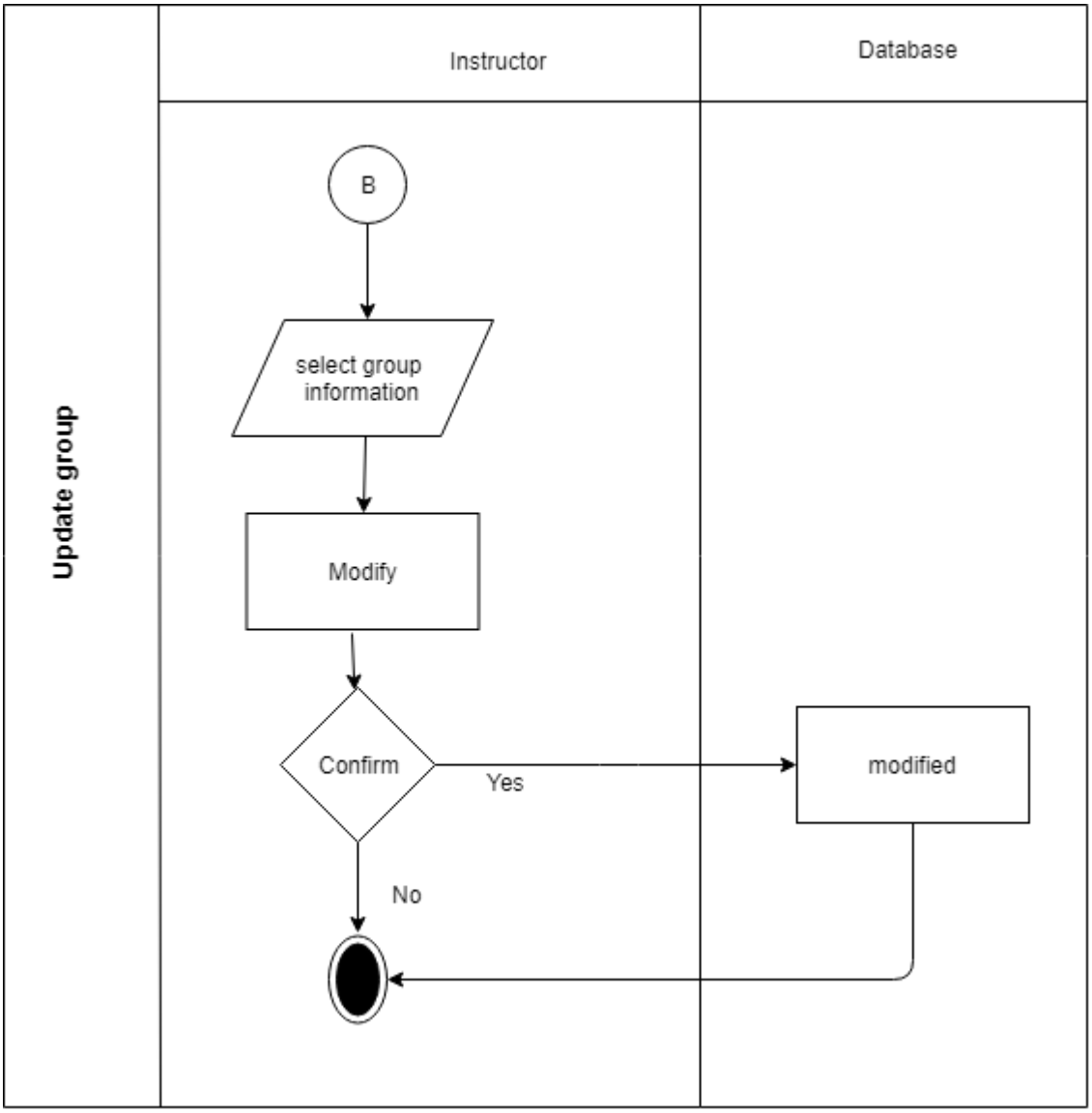


Figure – 41: Level 1.3.3 Swim lane diagram – Update group

SWIM LANE DIAGRAM – 3.4: REMOVE GROUP

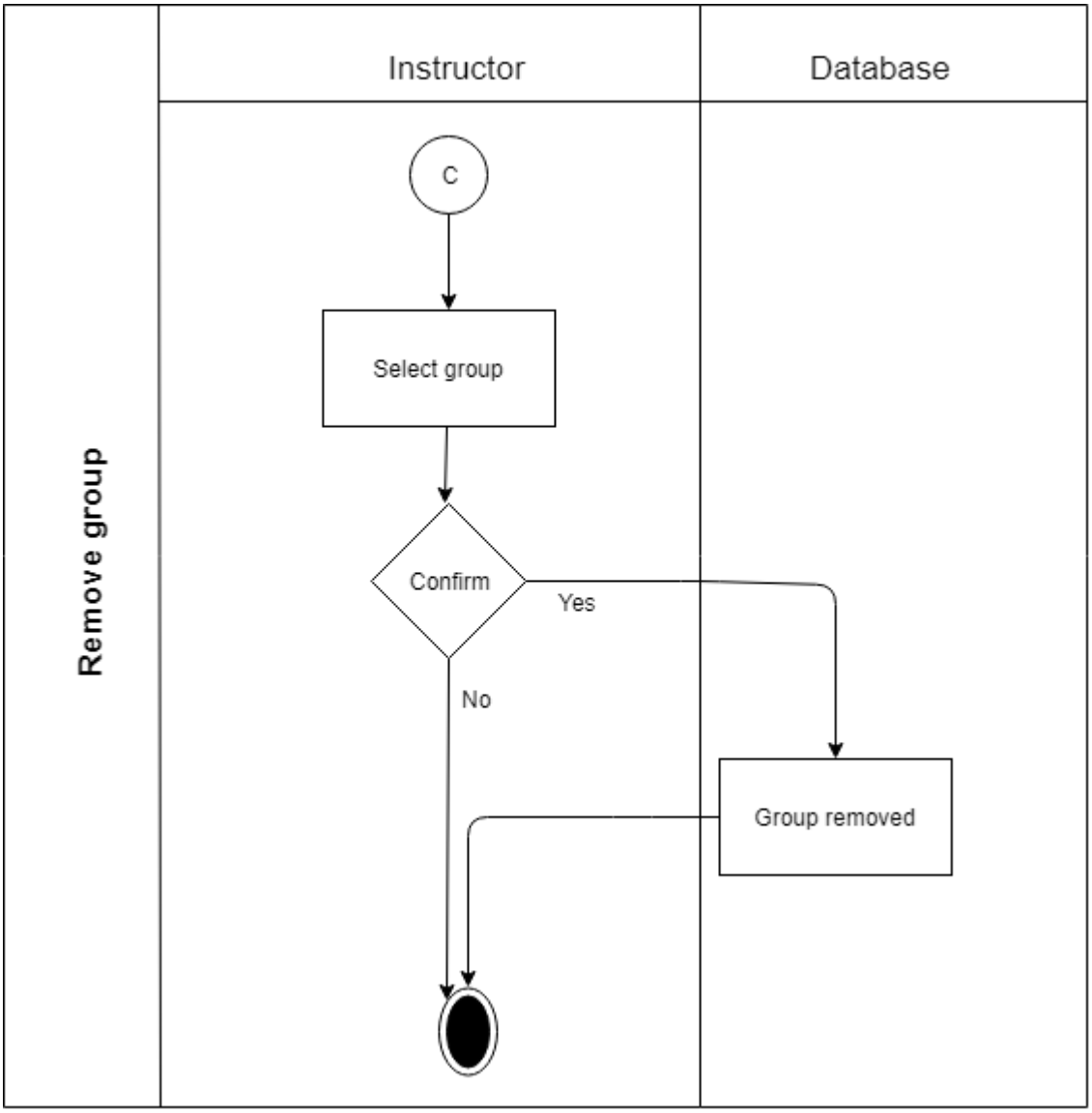


Figure – 40: Level 1.3.4 Swim lane diagram – Remove group

SWIM LANE DIAGRAM – 3.5: POST

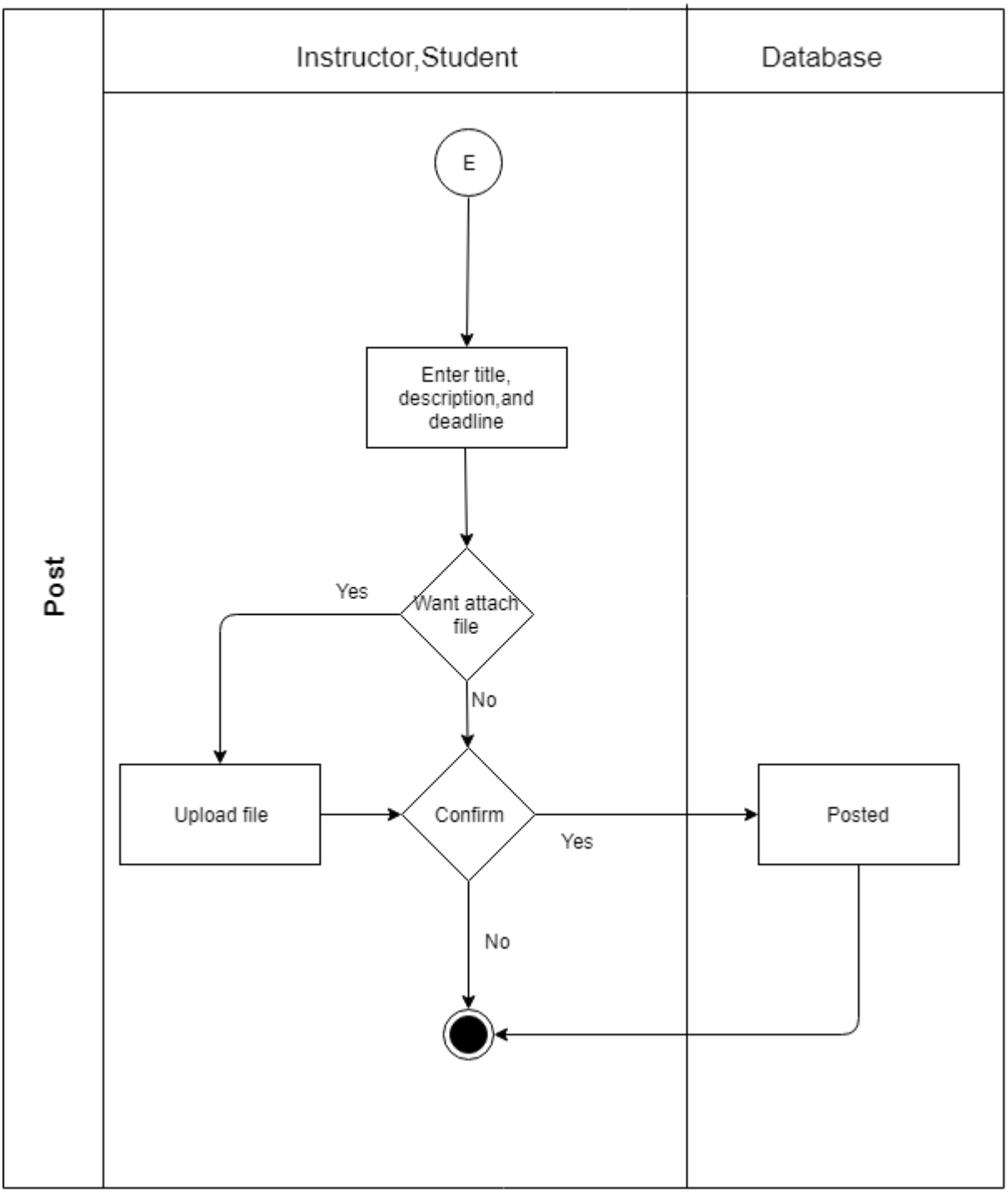


Figure – 42: Level 1.3.5 Swim lane diagram – Post