Assignment Management System

Software Project Lab-2

20/3/2018

Group member

Tulshi Chandra Das-811

Maloy Kanti Sarker-834

Supervised by

Amit Seal Ami

Lecturer, IIT, DU

IIT
University of Dhaka

ASSIGNMENT MANAGEMENT SYSTEM

SUBMITTED TO

SPL2 Committee
Institute of Information Technology
University of Dhaka

SUBMITTED BY

Tulshi Chandra Das BSSE0811
Maloy Kanti Sarkar BSSE0834

LETTER OF TRANSMITTAL

20 March 2018
The Coordinators, SPL2
Institute of Information Technology
University of Dhaka.

Subject: Submission of term report on “Assignment Management System”

Sir
With due respect, we are submitting the report on the above topic you have 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.

Sincerely yours
Tulshi Chandra Das BSSE0811

Maloy Kanti Sarkar BSSE0834

3rd Year, 5th Semester, 8th Batch
Institute of Information Technology
University of Dhaka

Supervisor

Amit Seal Ami
Lecturer, IIT, DU

ACKNOWLEDGEMENT

We are highly indebted for getting such a tremendous opportunity to prepare the report on Assignment Management System. We would like to thank our Supervisor, Amit Seal Ami, Lecturer, Institute of Information Technology, University of Dhaka, whole-heartedly, for giving us guidelines about how we can prepare this report and the coordinators of this course for giving guideline us. In completing this paper, we have collected data from some students and instructors of the university. We are thankful to all for the works cited.

ABSTRACT

The study is made for Assignment Management System. The scope of the study is to analyse on online Assignment Management System of Universities and design the SRS of this system. The object of this study is to develop an SRS (Software Requirements Specification and Analysis) of Assignment Management System.

TABLE OF CONTENTS

Content of figures

List of Tables

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 Assignment 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 designers, developers. The customer will use this SRS to verify that the developer team has created a product that the customer finds acceptable. 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. When portions of the software are complete, the developer 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 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 are. 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

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 this 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 creates a code for the students to join his/her group, post assignment, check plagiarism of student's assignment, distribute mark and comment on the post.

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

- Strong Authentication
- Filtering option
- Plagiarism checking
- Notification for each post or comment
- Communicate with Student
- Future support from developers

- Give format of the assignment
- Check format and requirements to submit
- Distribute mark evaluating plagiarism
- User-friendly and efficient system
- Computer-based system
- Error-free system
- Easy to operate

STUDENT VIEWPOINTS

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

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 a final decision about requirements

COMMON REQUIREMENTS

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

CONFLICTING REQUIREMENTS

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

FINAL REQUIREMENTS

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

2.1.4 REQUIREMENTS QUESTIONNAIRE

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 in 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 to throw 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 a basic understanding about the Assignment 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 the problems 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 in 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 requirements 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 a 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 ASSIGNMENT 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 for 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 DISTRIBUTION

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 MANAGEMENT

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 Assignment 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 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 the 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 the overall system.

4.3.1 LEVEL- 0 USE CASE DIAGRAM-AMS

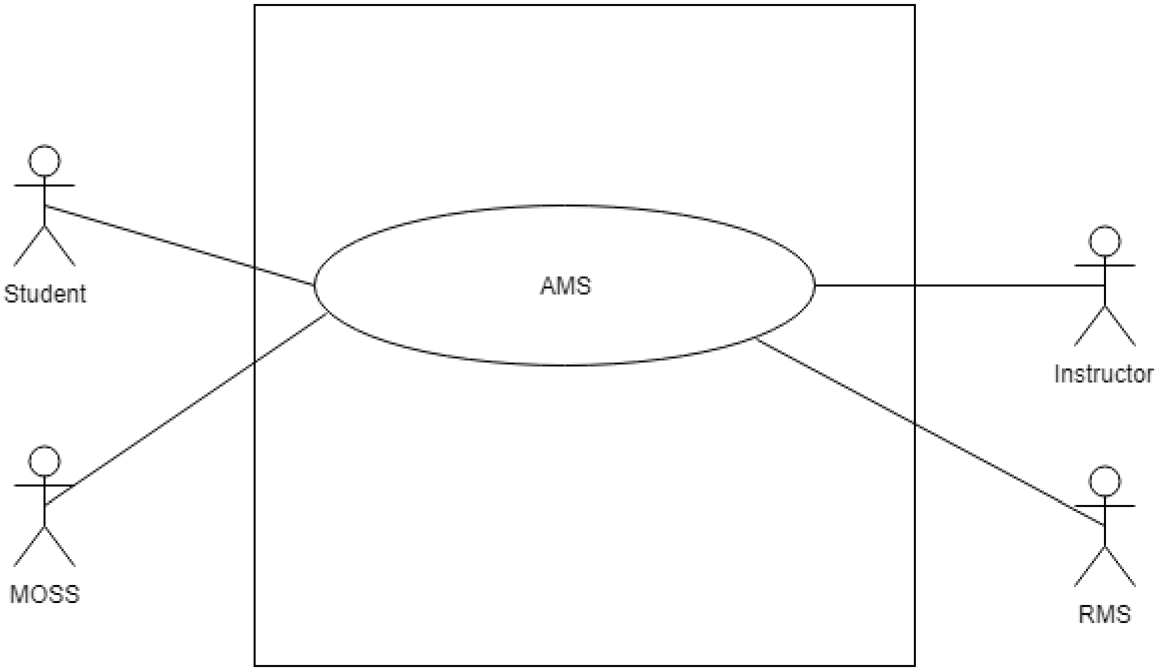


Figure 1: level 0 use case diagram- AMS

Name: Assignment Management System

Primary actor: Instructor, Student

Secondary actor: Result Management System, MOSS

DESCRIPTION OF USE CASE DIAGRAM LEVEL-0

After analyzing user story, we found five 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 the information.

Following the actors of "Assignment Management System":

- Instructor

- Student
- Database

4.3.2 LEVEL -1 USECASE DIAGRAM-SUBSYSTEM

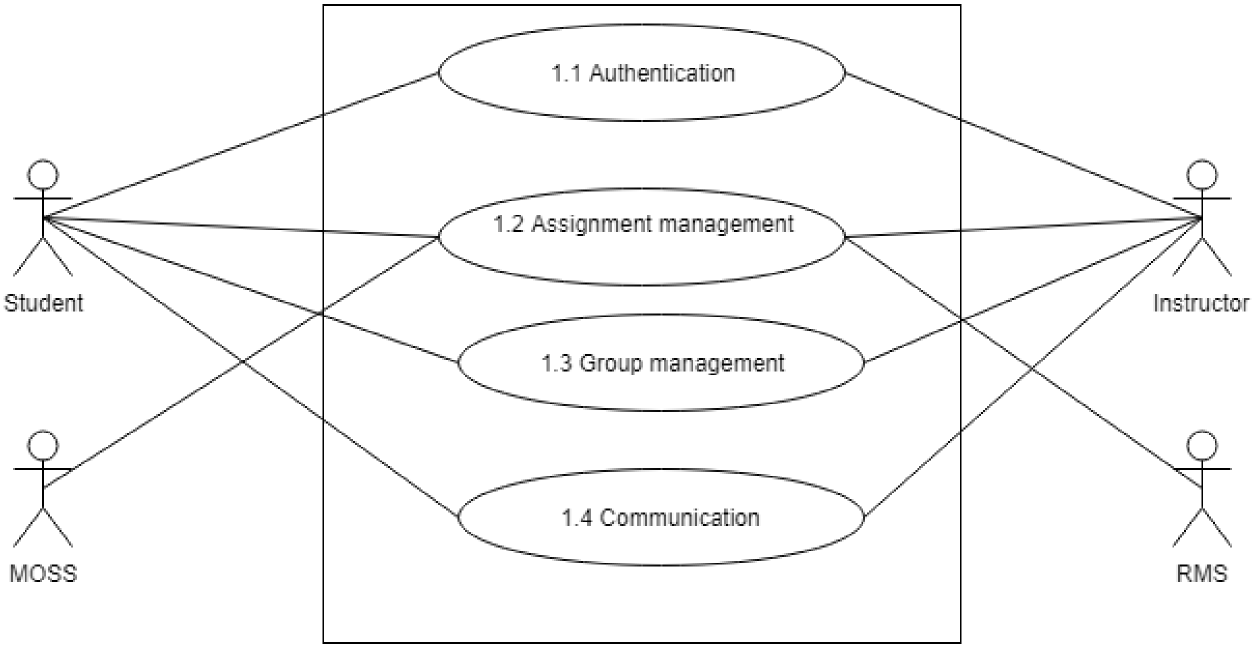


Figure-2: Figure 2level 1 use case diagram - Subsystem

Name: Subsystem of AMS

Primary actor: Instructor, Student

Secondary actor: MOSS, RMS

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

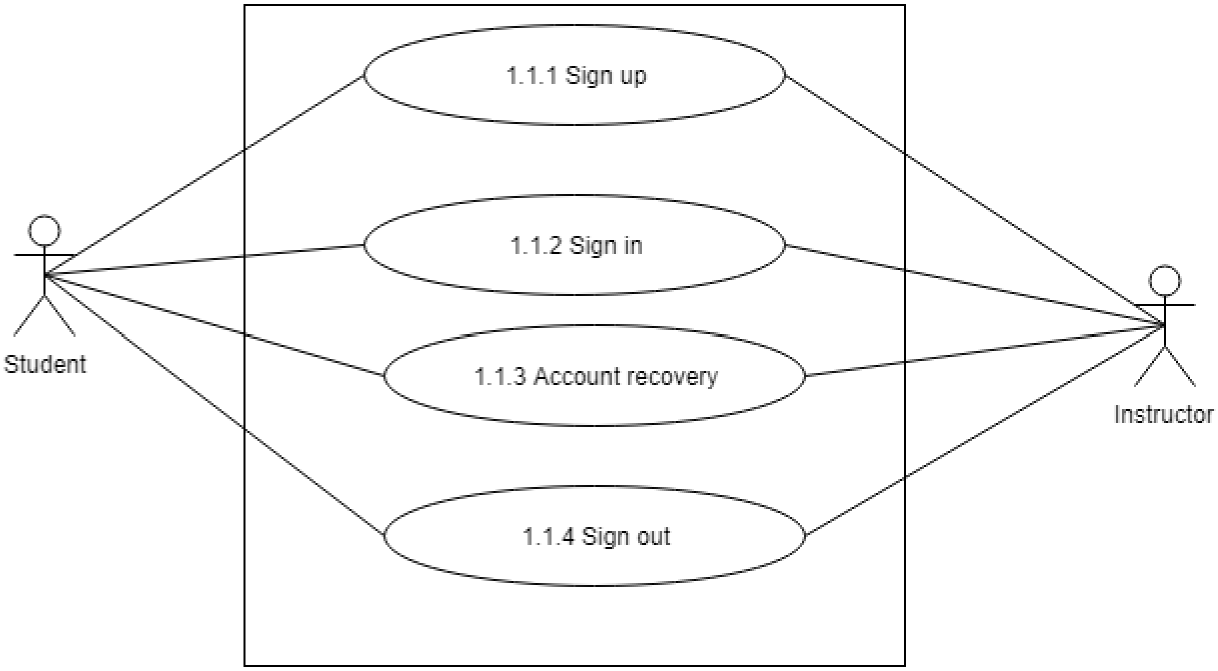


Figure-3: Figure 3level 1.1 use case diagram – Authentication

Name: Authentication of AMS

Primary actor: Instructor, Student

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 AMS 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
- Secondary actor: N/A

STUDENTS ACTION/REPLY

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

INSTRUCTOR ACTION/REPLY

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

1.1.2 SIGN IN

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

STUDENTS ACTION/REPLY

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

INSTRUCTOR ACTION/REPLY

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

1.1.3 ACCOUNT RECOVERY

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

STUDENTS ACTION/REPLY

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

INSTRUCTOR ACTION/REPLY

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

4.3.4 LEVEL -1.2 USE CASE DIAGRAM- ASSIGNMENT MANAGEMENT

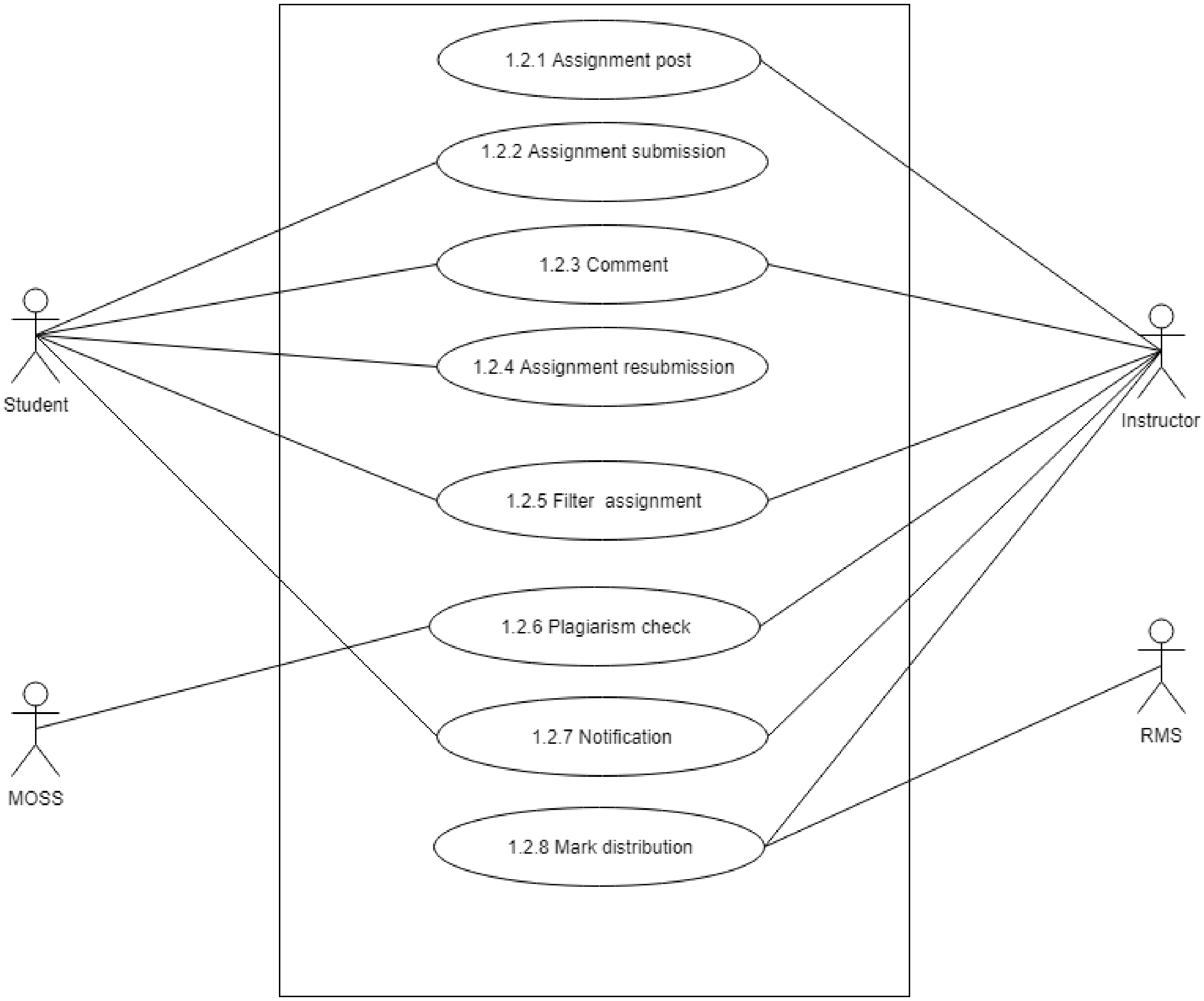


Figure 4: level 1.2 use case diagram- Assignment management

Name: Assignment management of AMS

Primary actor: Instructor, Student

Secondary actor: MOSS, RMS

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

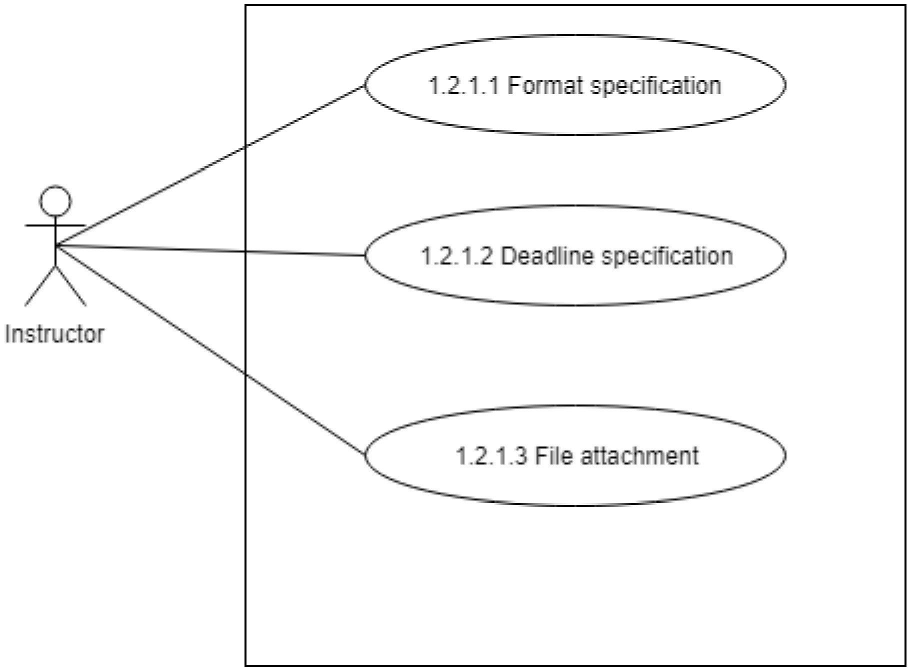


Figure 5: level 1.2.1 use case diagram- Post assignment

Name: Post Assignment of AMS

Primary actor: Instructor

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
- Secondary actor: N/A

INSTRUCTOR ACTION/REPLY

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

1.2.1.2 DEADLINE SPECIFICATION

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

INSTRUCTOR ACTION/REPLY

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

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.

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 the corresponding assignment and show a message whether it is submitted or not.

1.2.3 COMMENT

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

INSTRUCTOR ACTION/REPLY

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

STUDENT ACTION/REPLY

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

1.2.4 ASSIGNMENT RESUBMISSION

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

STUDENT ACTION/REPLY

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

1.2.5 FILTER ASSIGNMENT

- Primary actor: Instructor, Student
- 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.

1.2.6 PLAGIARISM CHECK

- Primary actor: Instructor
- Secondary actor: MOSS

INSTRUCTOR ACTION/REPLY

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

MOSS ACTION/REPLY

- Action: MOSS will receive a request of plagiarism check
- Reply: MOSS will send the result

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.

1.2.8 MARK DISTRIBUTION

- Primary actor: Instructor
- Secondary actor: RMS

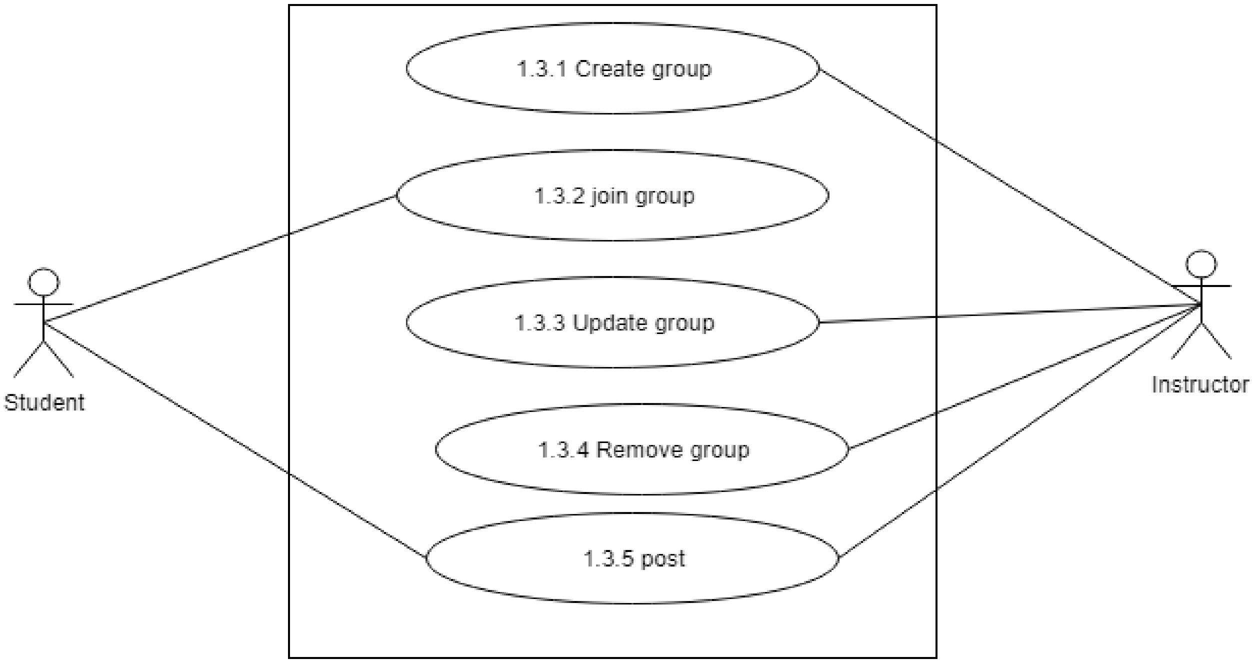
INSTRUCTOR ACTION/REPLY

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

RMS ACTION/REPLY

- Action: RMS will request for get marks
- Reply: RMS will the results sent by AMS

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

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
- Secondary actor: N/A

INSTRUCTOR ACTION/REPLY

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

1.3.2 JOIN GROUP

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

STUDENT ACTION/REPLY

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

1.3.3 UPDATE GROUP

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

INSTRUCTOR ACTION/REPLY

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

1.3.4 REMOVE GROUP

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

INSTRUCTOR ACTION/REPLY

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

1.3.5 POST

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

INSTRUCTOR ACTION/REPLY

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

STUDENT ACTION/REPLY

- Action: Student gives post.
- Reply: Posted Successfully.

1.4 COMMUNICATION

- Primary actor: Instructor, Student
- 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.

4.4 ACTIVITY DIAGRAMS

ACTIVITY DIAGRAM – 1: AUTHENTICATION

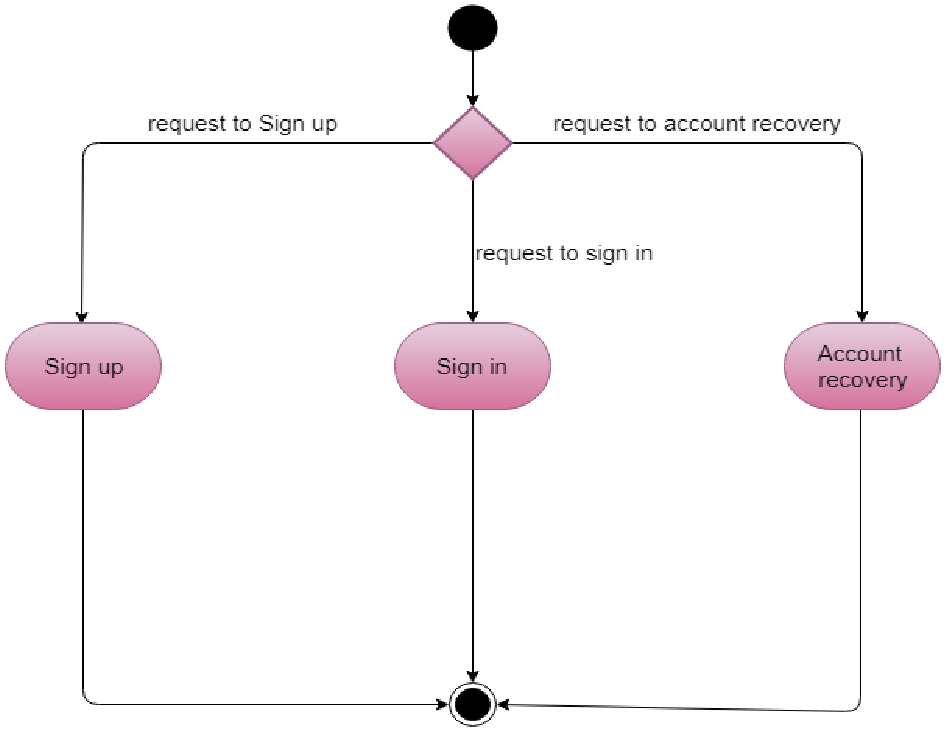


Figure 7: Level 1.1 Activity diagram – Authentication.

ACTIVITY DIAGRAM – 1.1: SIGN UP

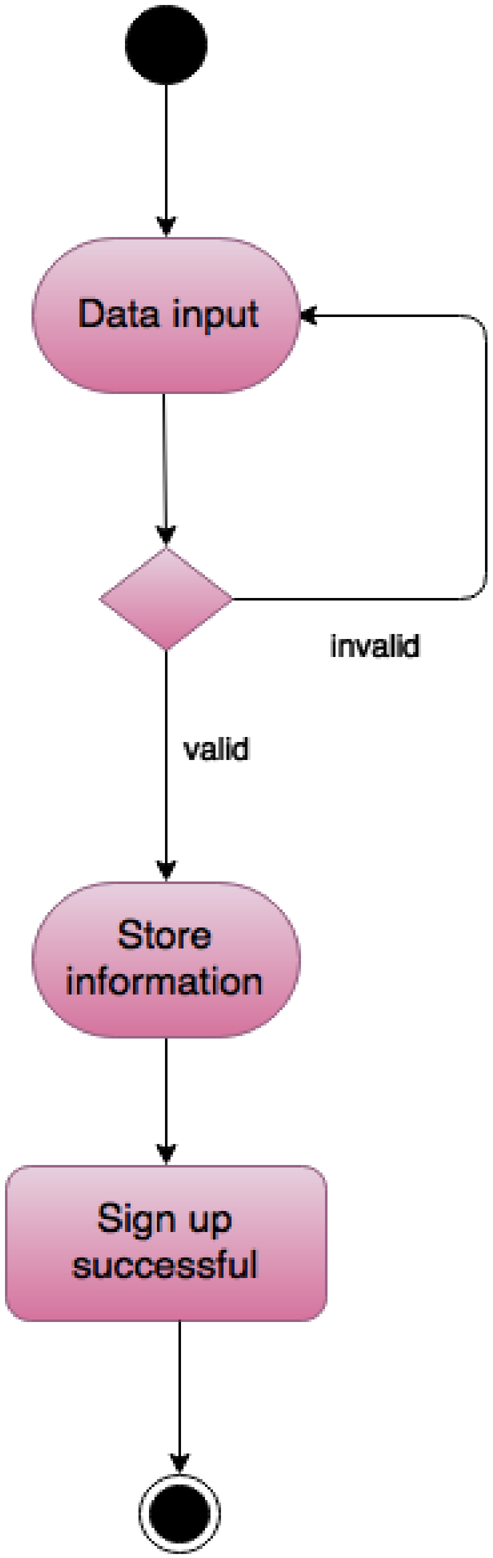


Figure 8: Level 1.1.1 Activity diagram – Sign up

ACTIVITY DIAGRAM – 1.2: SIGN IN

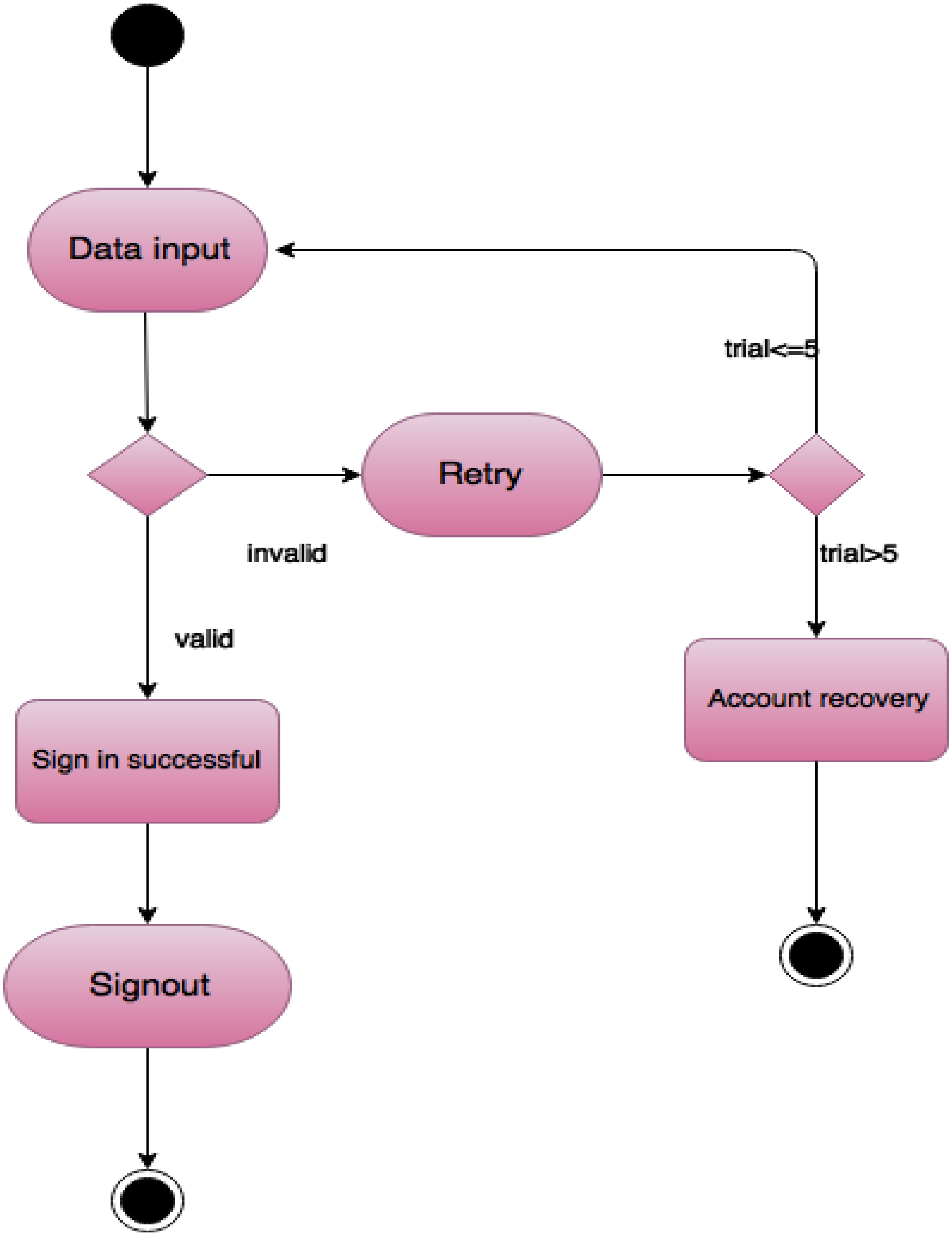


Figure 9: Level 1.1.2 Activity diagram – Sign in.

ACTIVITY DIAGRAM – 1.3: ACCOUNT RECOVERY

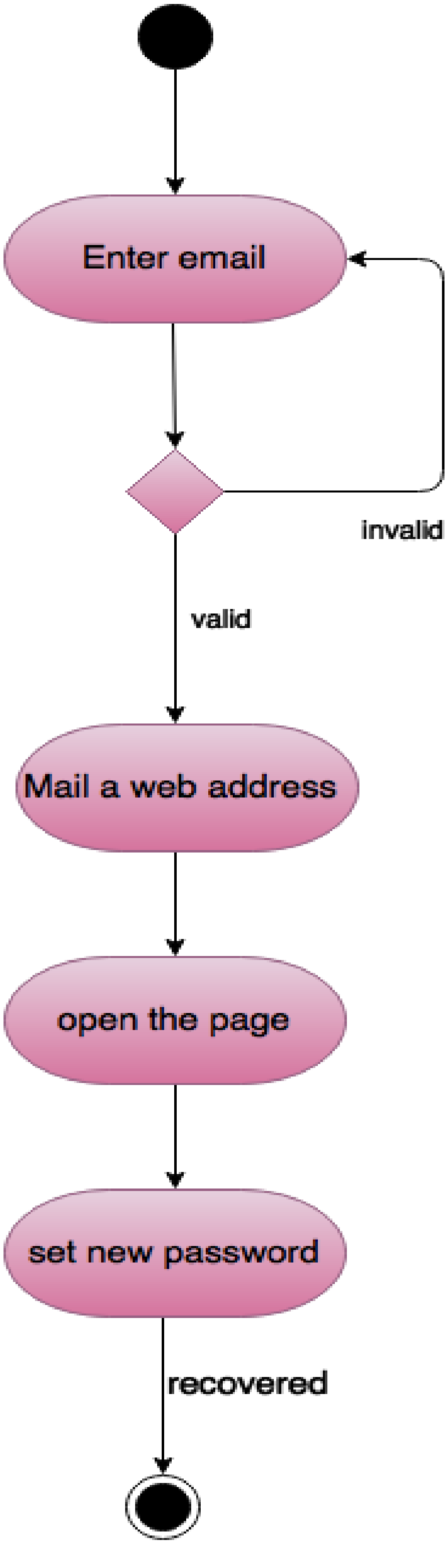


Figure 10: Level 1.1.3 Activity diagram – Account recovery

ACTIVITY DIAGRAM – 2: ASSIGNMENT MANAGEMENT

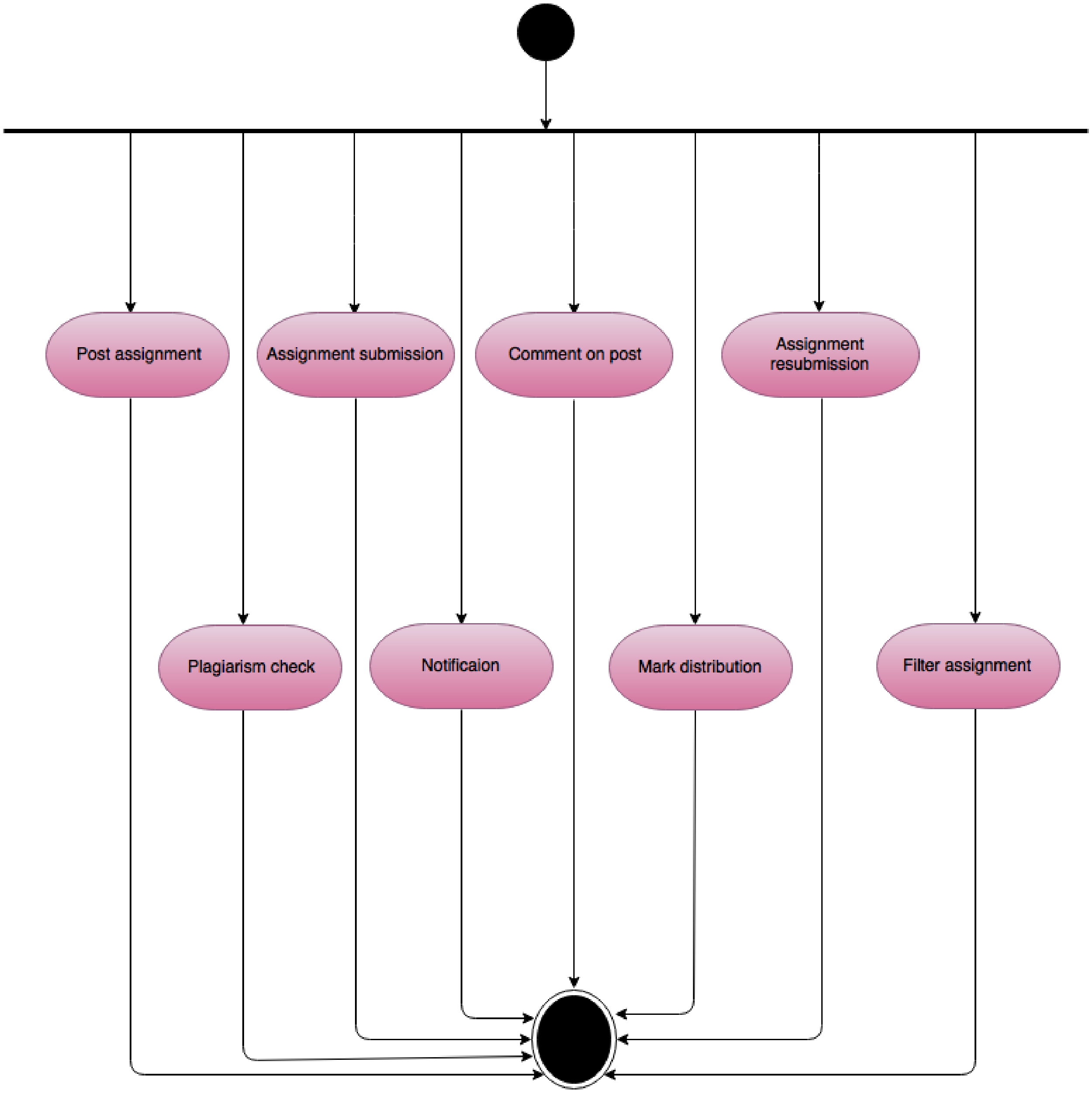


Figure 11: Level 1.2 Activity diagram – Assignment management

ACTIVITY DIAGRAM – 2.1: ASSIGNMENT POST

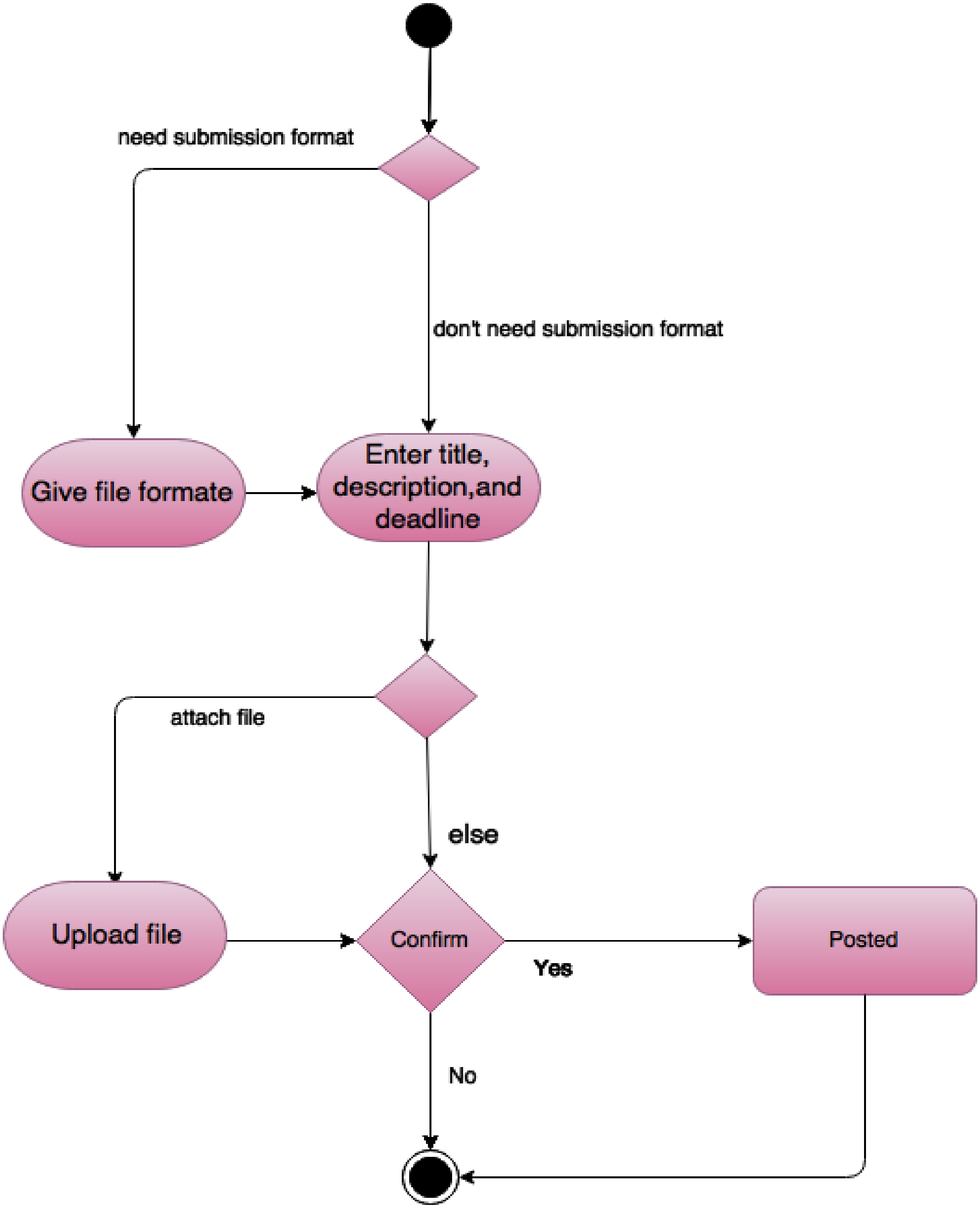


Figure 12: Level 1.2.1 Activity diagram – Assignment post

ACTIVITY DIAGRAM – 2.2: ASSIGNMENT SUBMISSION

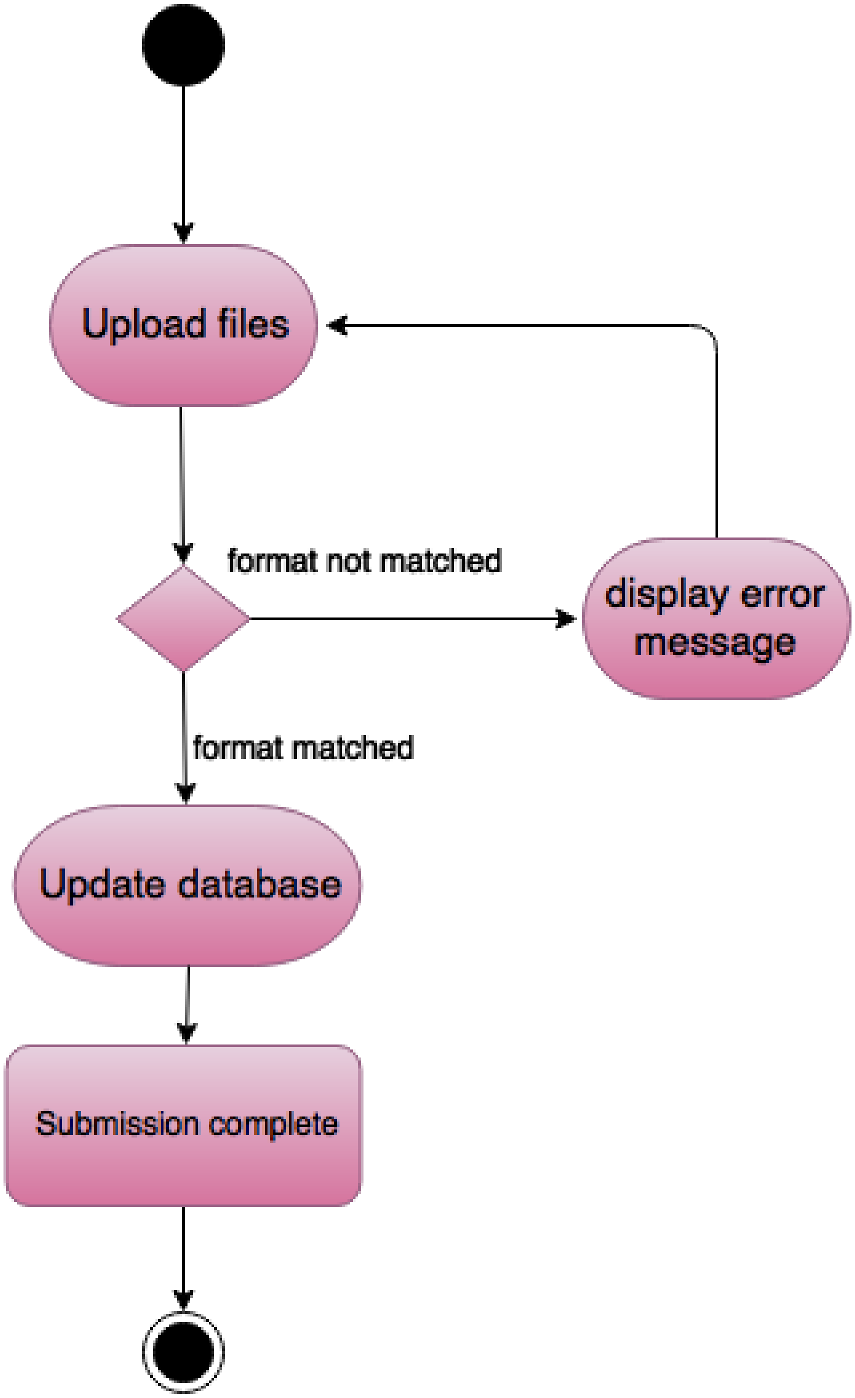


Figure 13: Level 1.2.2 Activity diagram – Assignment submission

ACTIVITY DIAGRAM – 2.3: COMMENT

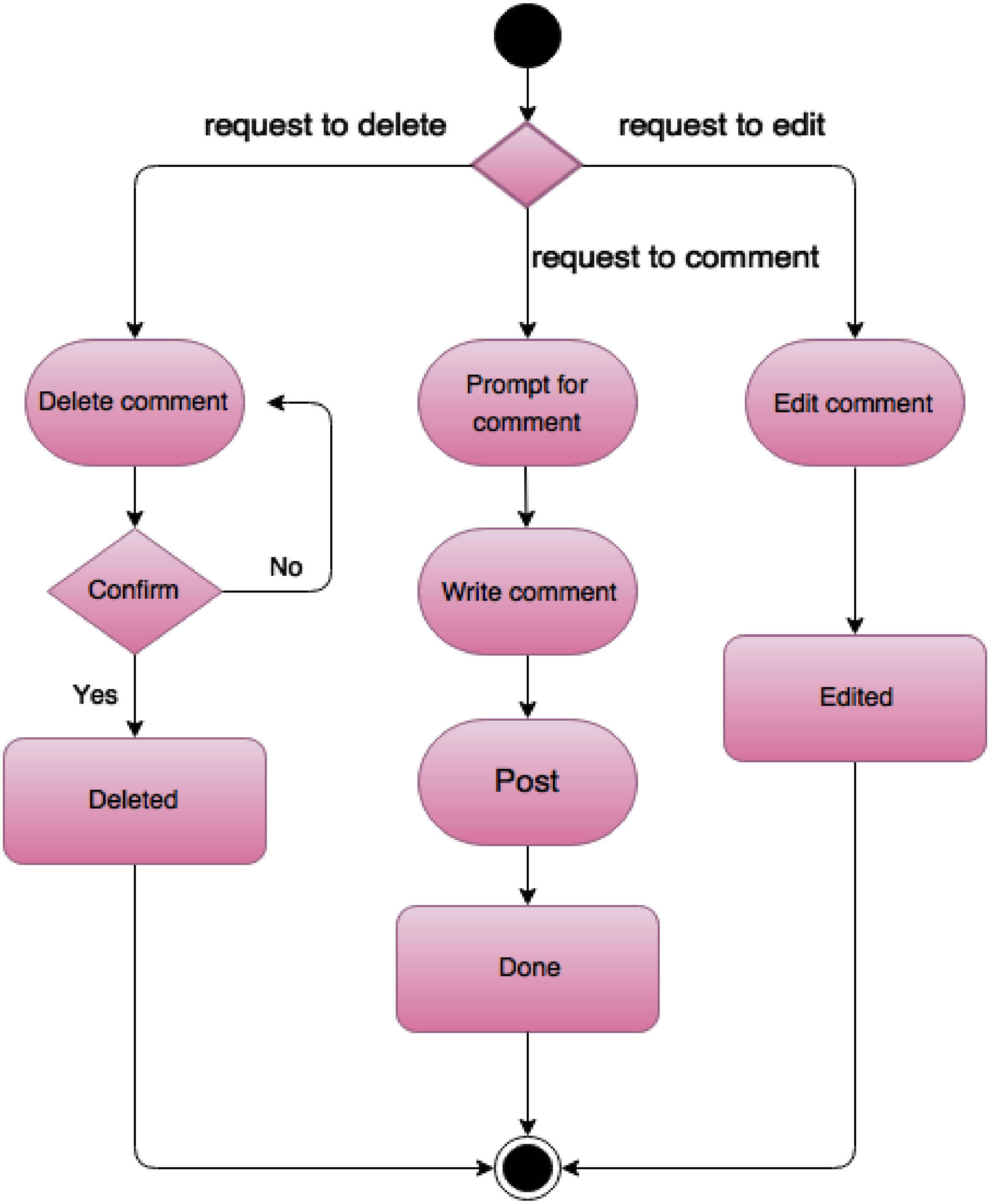


Figure 14: Level 1.2.3 Activity diagram – Comment.

ACTIVITY DIAGRAM – 2.4: ASSIGNMENT RESUBMISSION

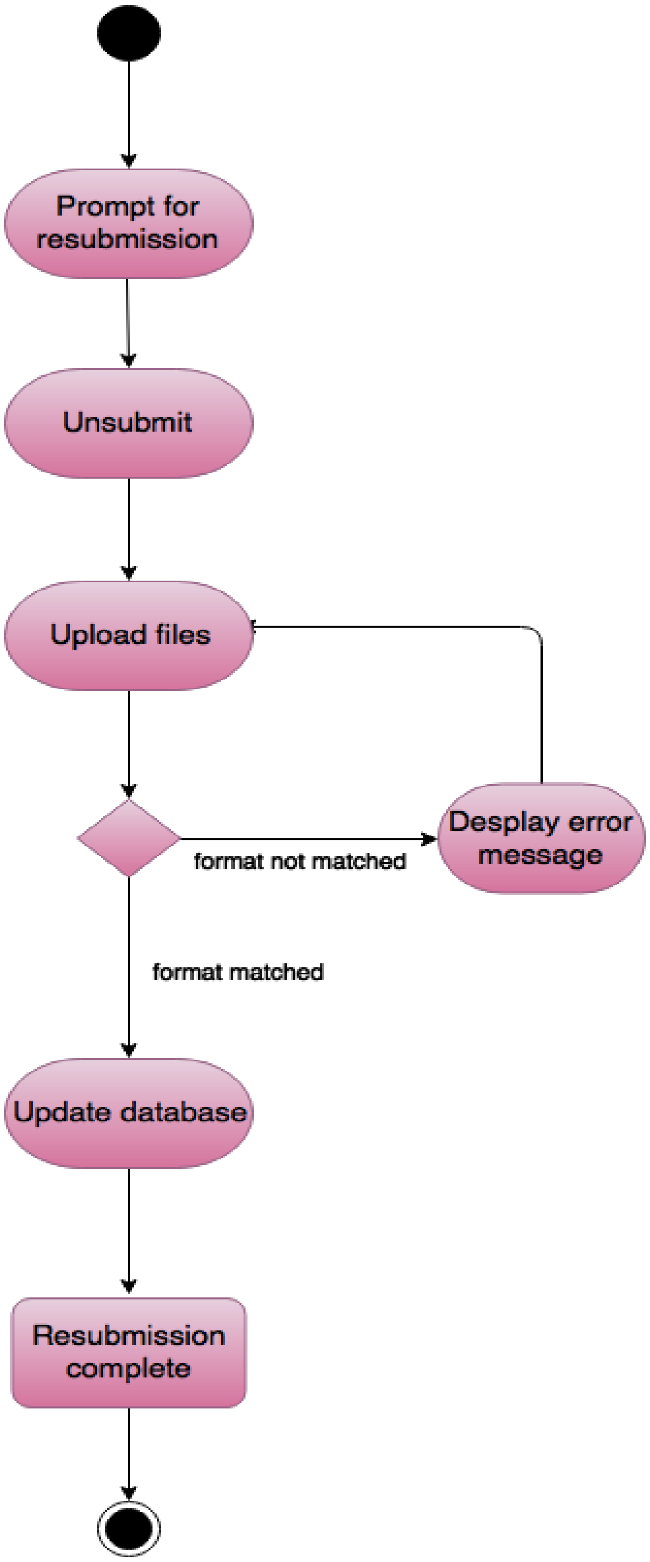


Figure 15: Level 1.2.4 Activity diagram – Assignment resubmission.

ACTIVITY DIAGRAM – 2.5: FILTER ASSIGNMENT

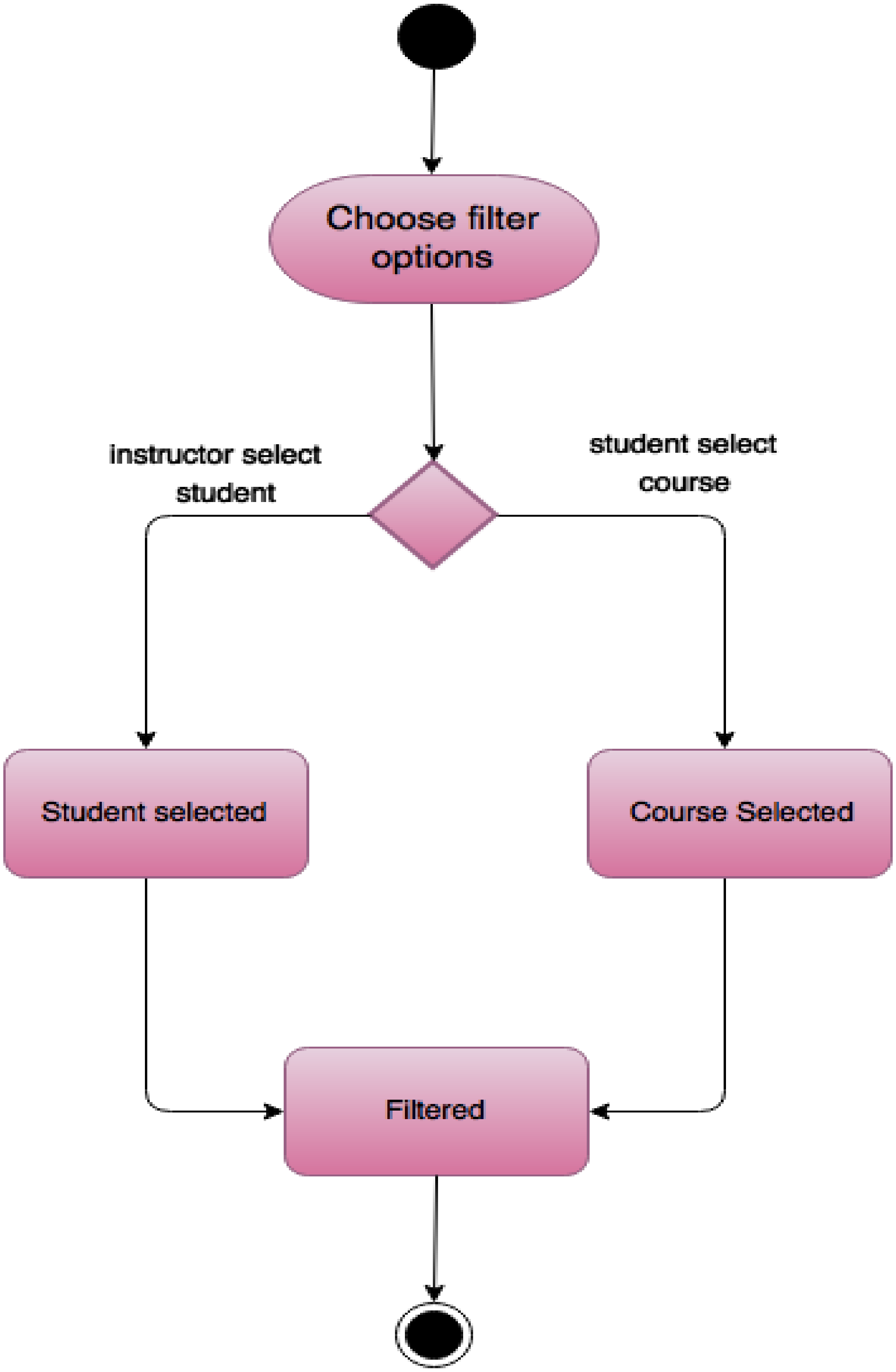


Figure 16: Level 1.2.5 Activity diagram – Filter assignment

ACTIVITY DIAGRAM – 2.6: PLAGIARISM CHECK

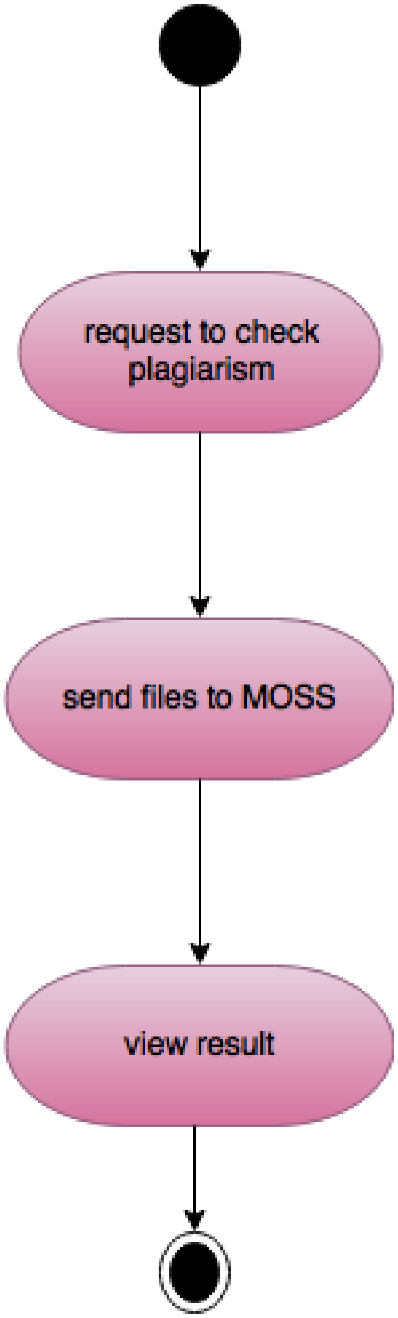


Figure 17: Level 1.2.6 Activity diagram – Plagiarism check

ACTIVITY DIAGRAM – 2.7: NOTIFICATION

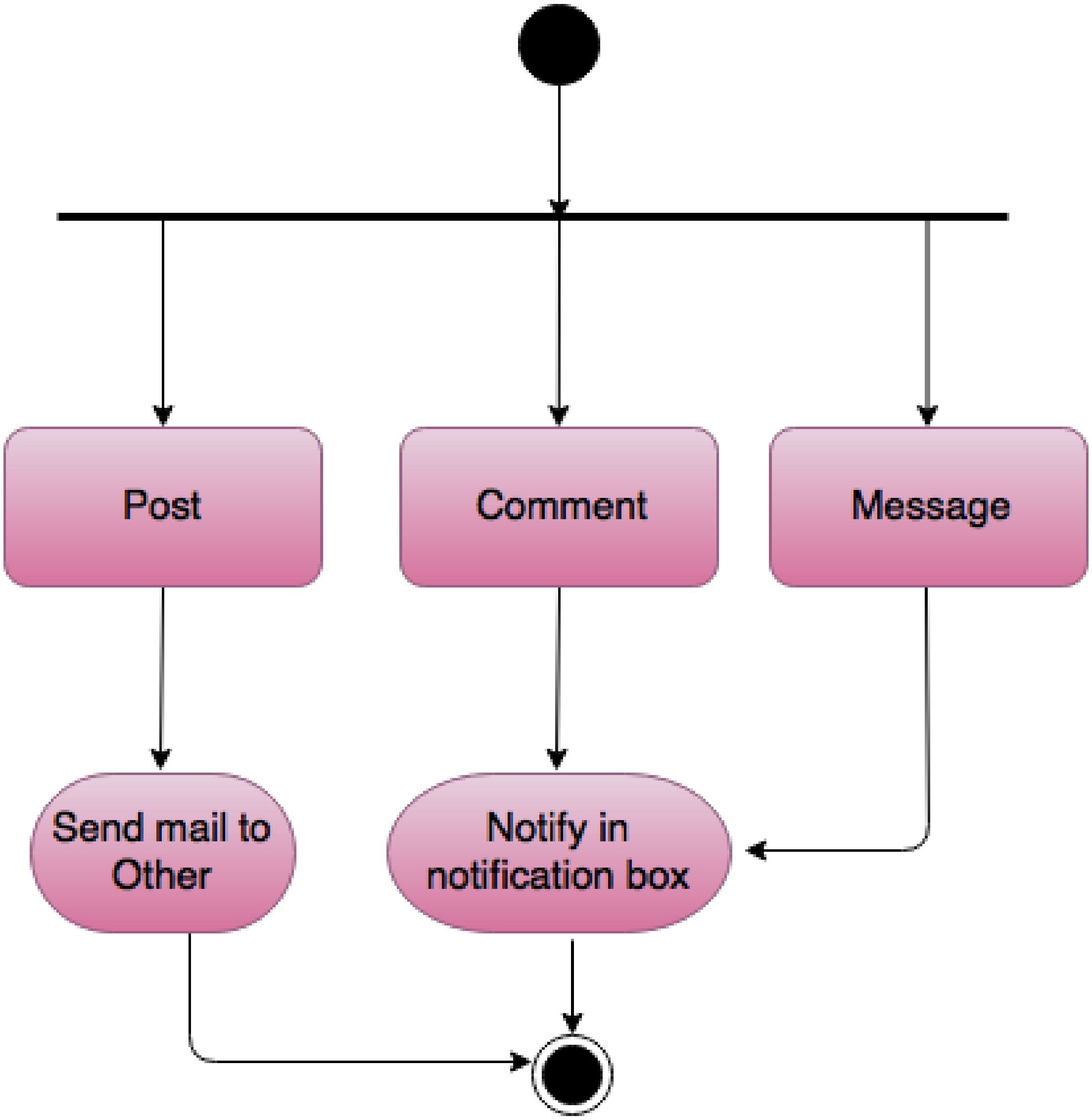


Figure 18: Level 1.2.7 Activity diagram –Notification

ACTIVITY DIAGRAM – 2.8: MARK DISTRIBUTION

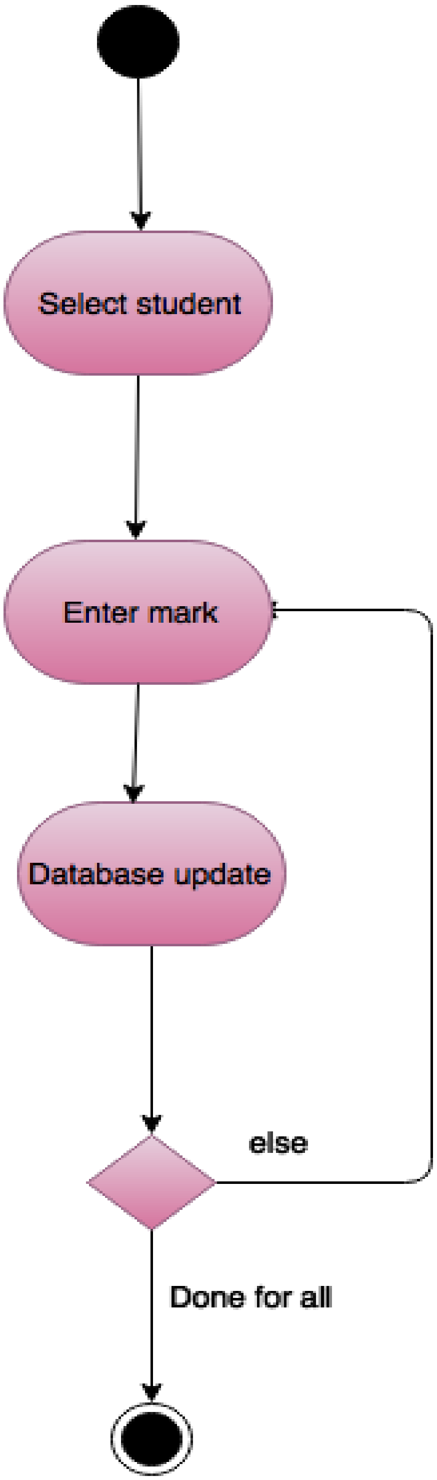


Figure 19: Level 1.2.8 Activity diagram – Mark distribution.

ACTIVITY DIAGRAM – 3: GROUP MANAGEMENT

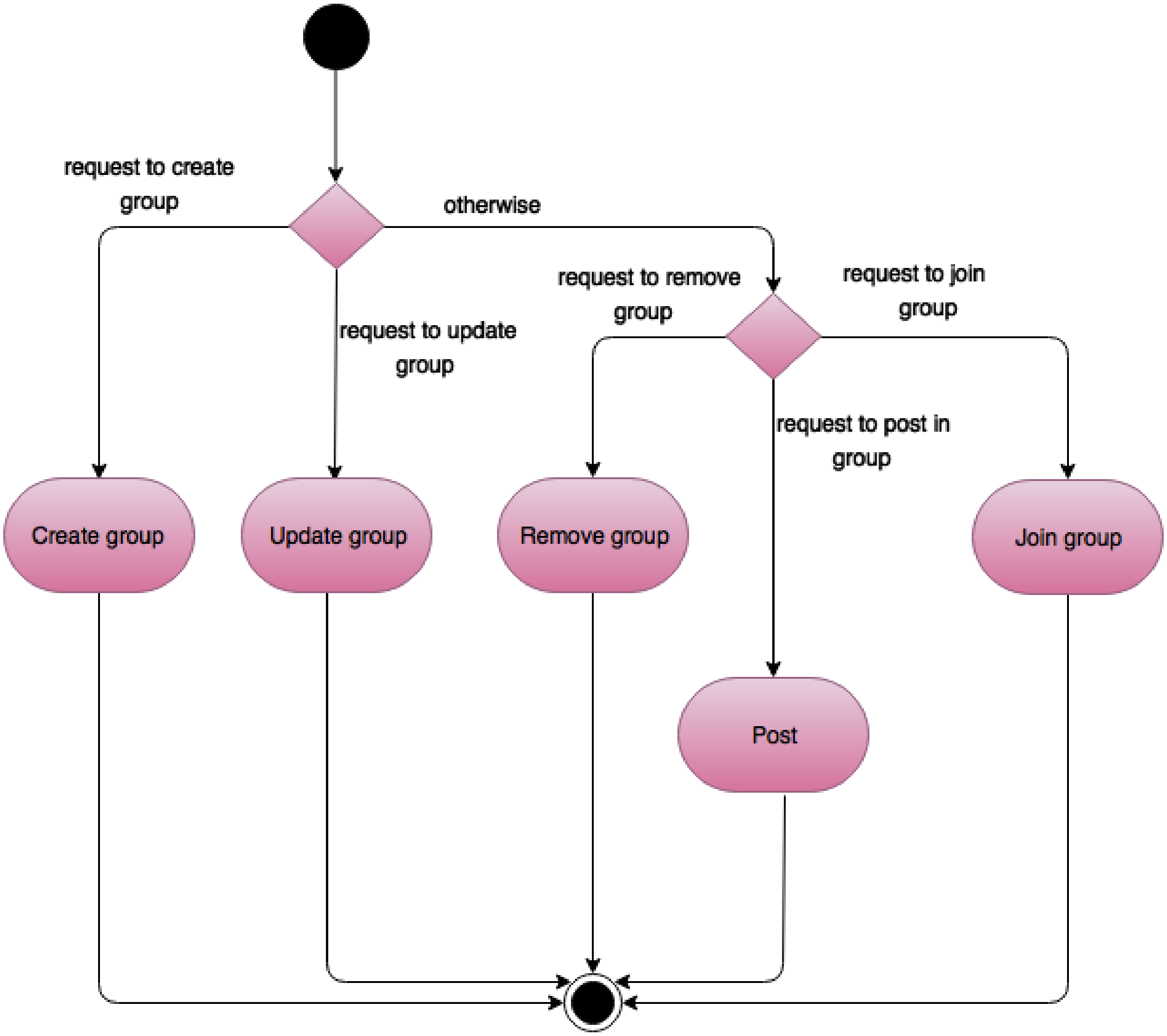


Figure 20: Level 1.3 Activity diagram – Group management.

ACTIVITY DIAGRAM – 3.1: CREATE GROUP

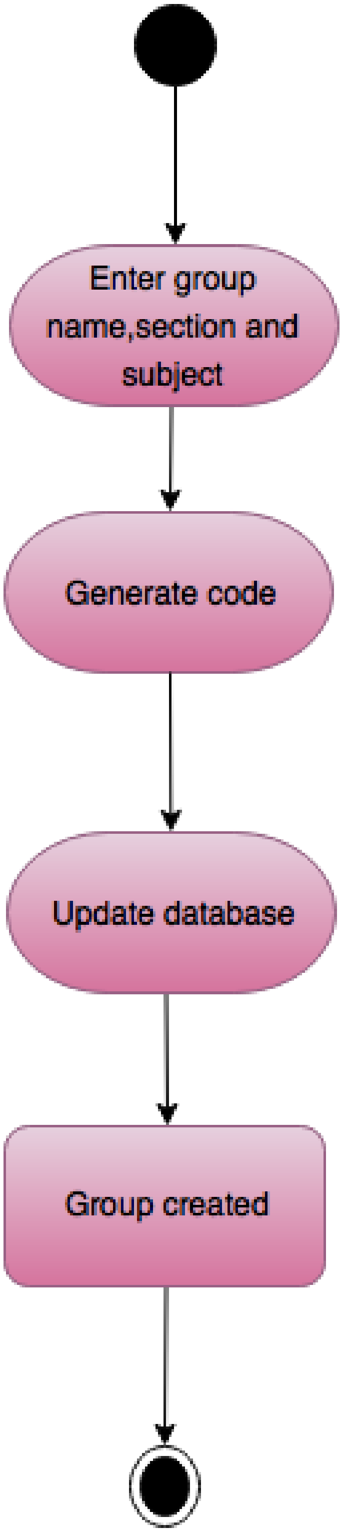


Figure 21: Level 1.3.1 Activity diagram – Create group

ACTIVITY DIAGRAM – 3.2: JOIN GROUP

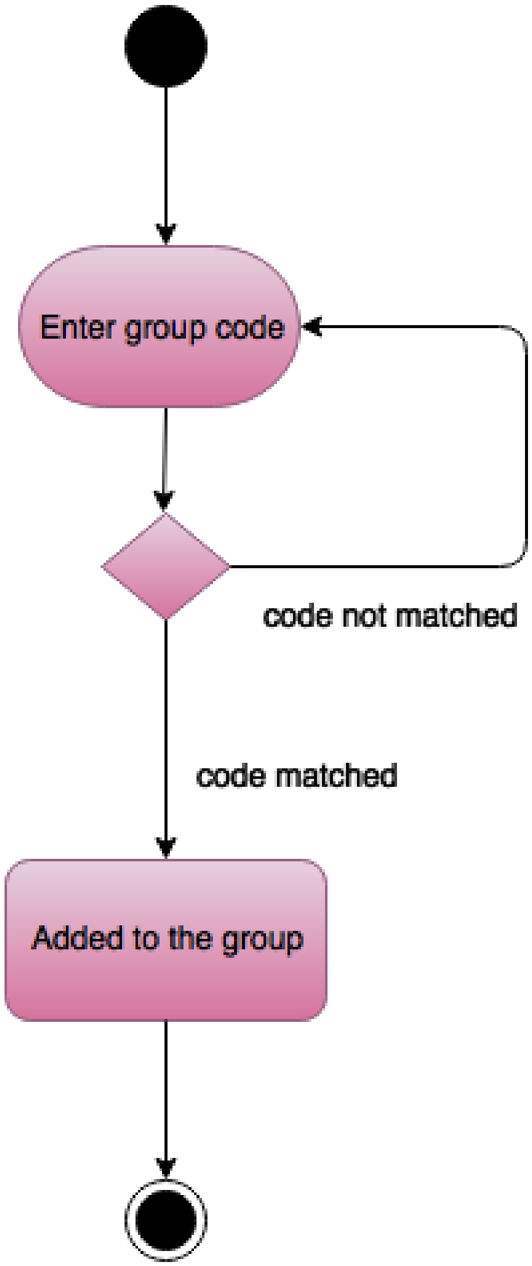


Figure 22: Level 1.3.2 Activity diagram – Join group

ACTIVITY DIAGRAM – 3.3: UPDATE GROUP

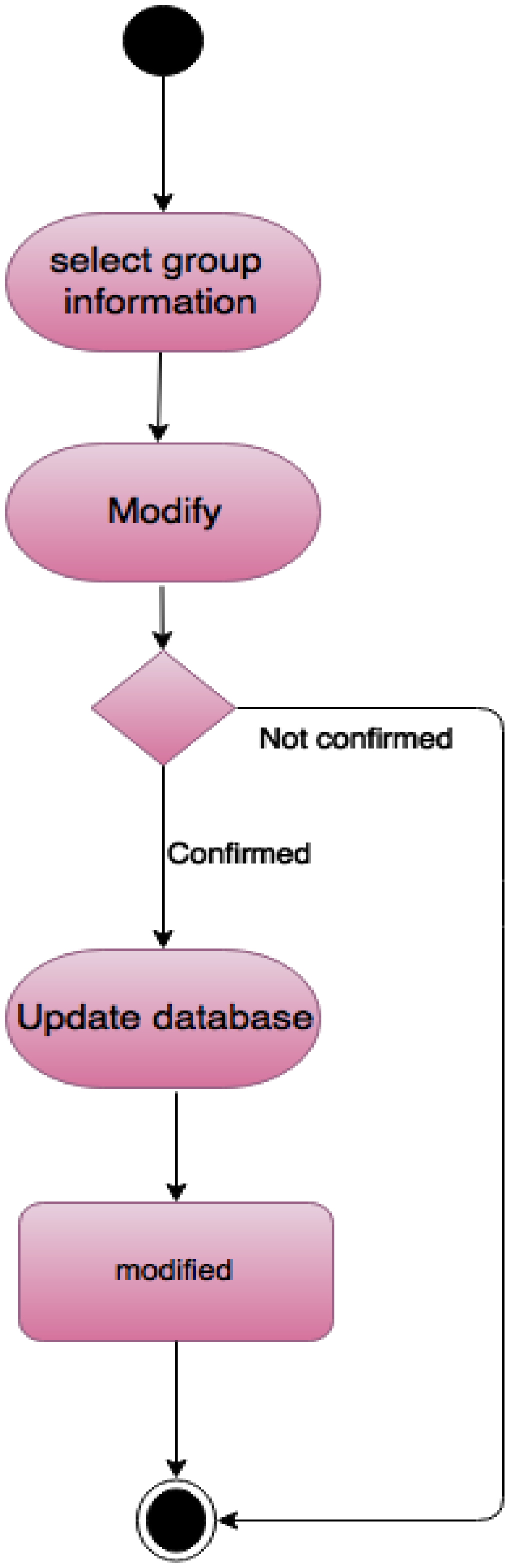


Figure 23: Level 1.3.3 Activity diagram – Update group.

ACTIVITY DIAGRAM – 3.4: REMOVE GROUP

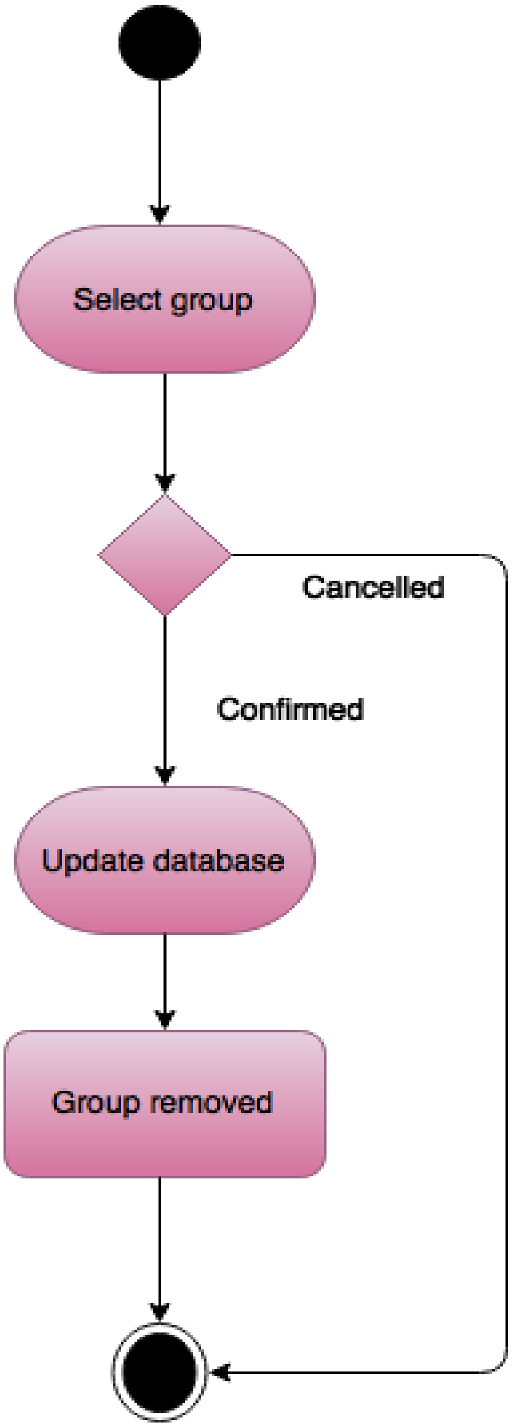


Figure 24: Level 1.3.4 Activity diagram – Remove group.

ACTIVITY DIAGRAM – 3.5: POST

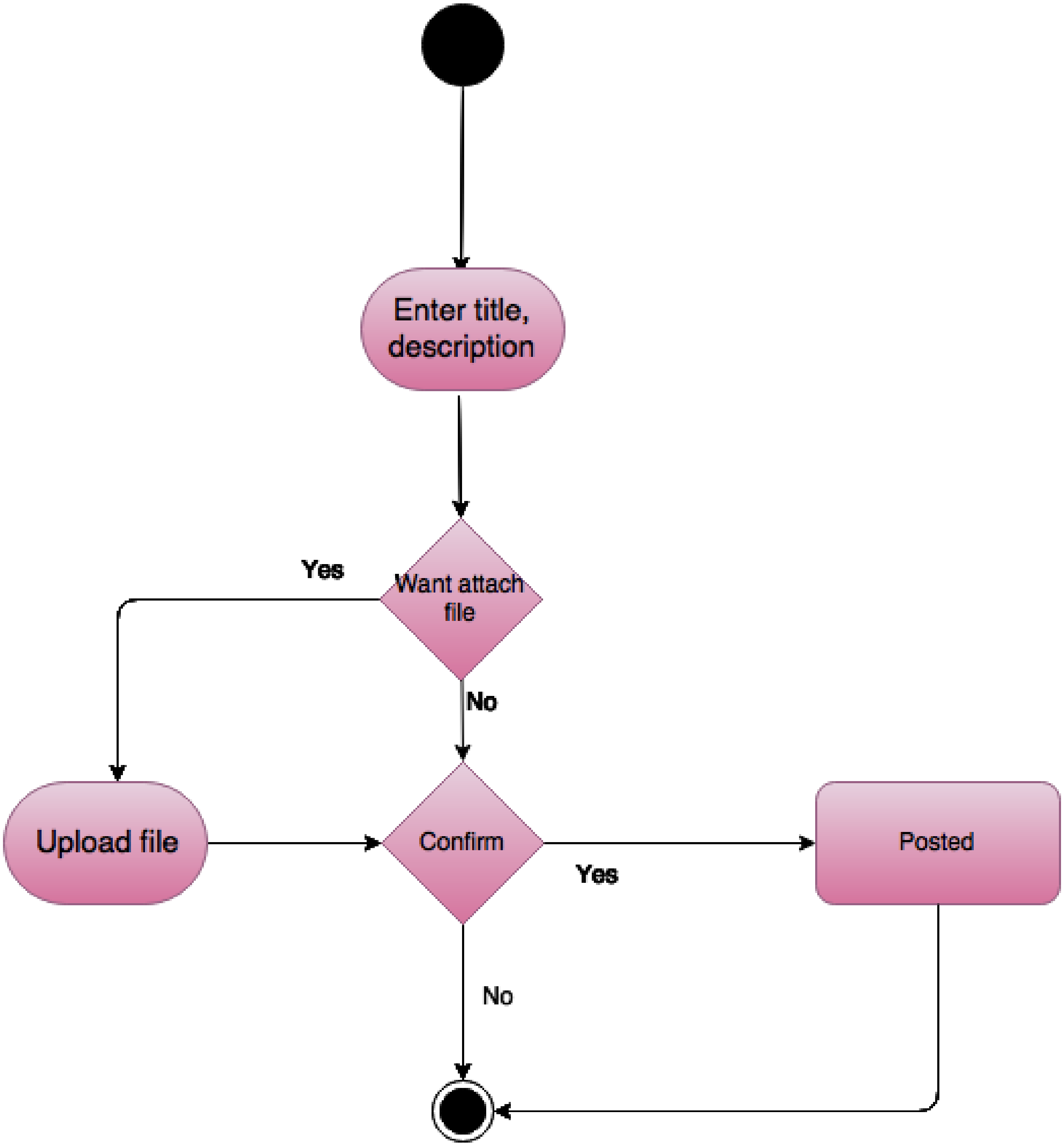


Figure 25: Level 1.3.5 Activity diagram – Post.

4.4 SWIM LANE DIAGRAMS

SWIM LANE DIAGRAM – 1: AUTHENTICATION

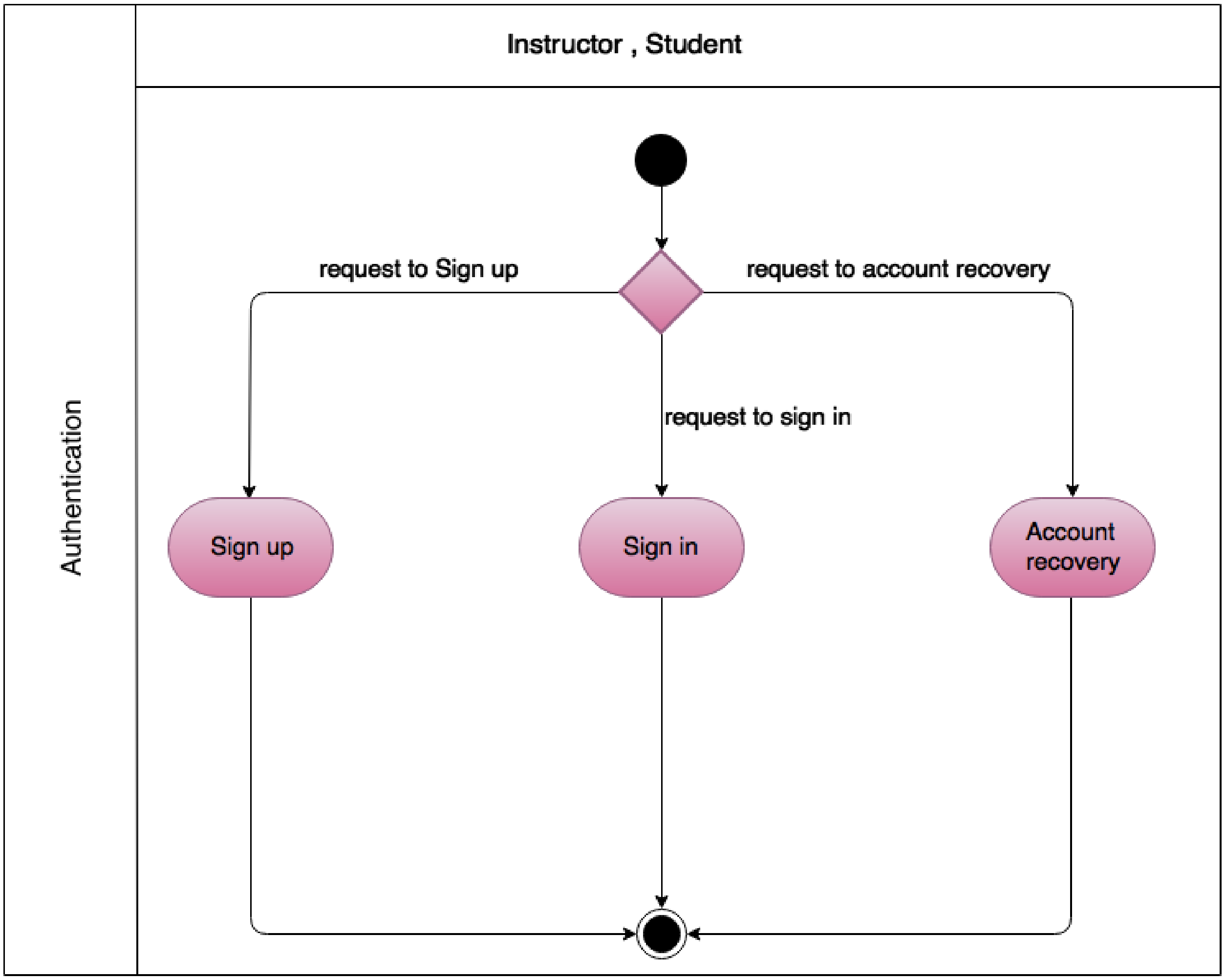


Figure 26: Level 1.1 Swim lane diagram – Authentication.

SWIM LANE DIAGRAM – 1.1: SIGN UP

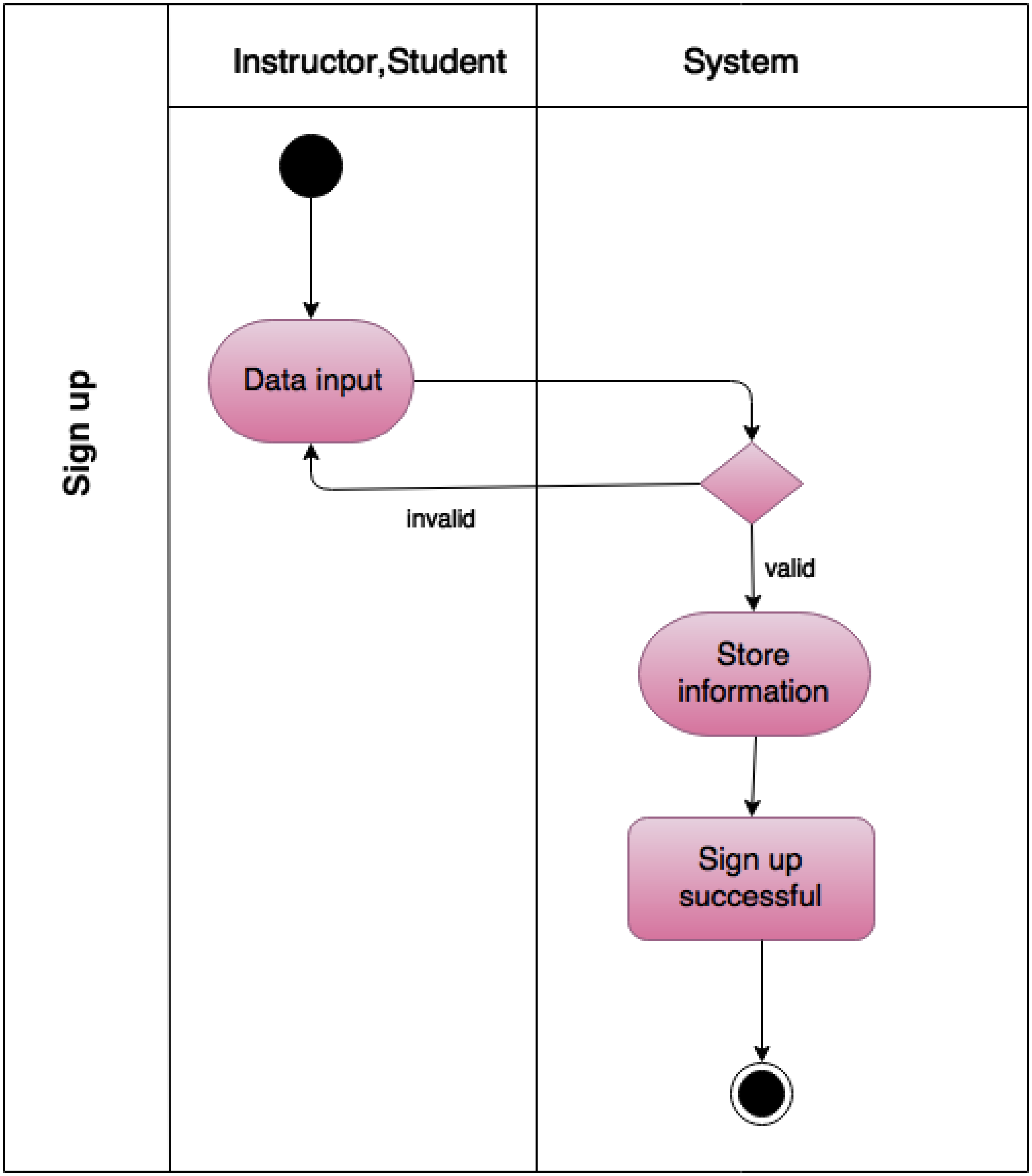


Figure 27: Level 1.1.1 Swim lane diagram – Sign up.

SWIM LANE DIAGRAM – 1.1: SIGN IN

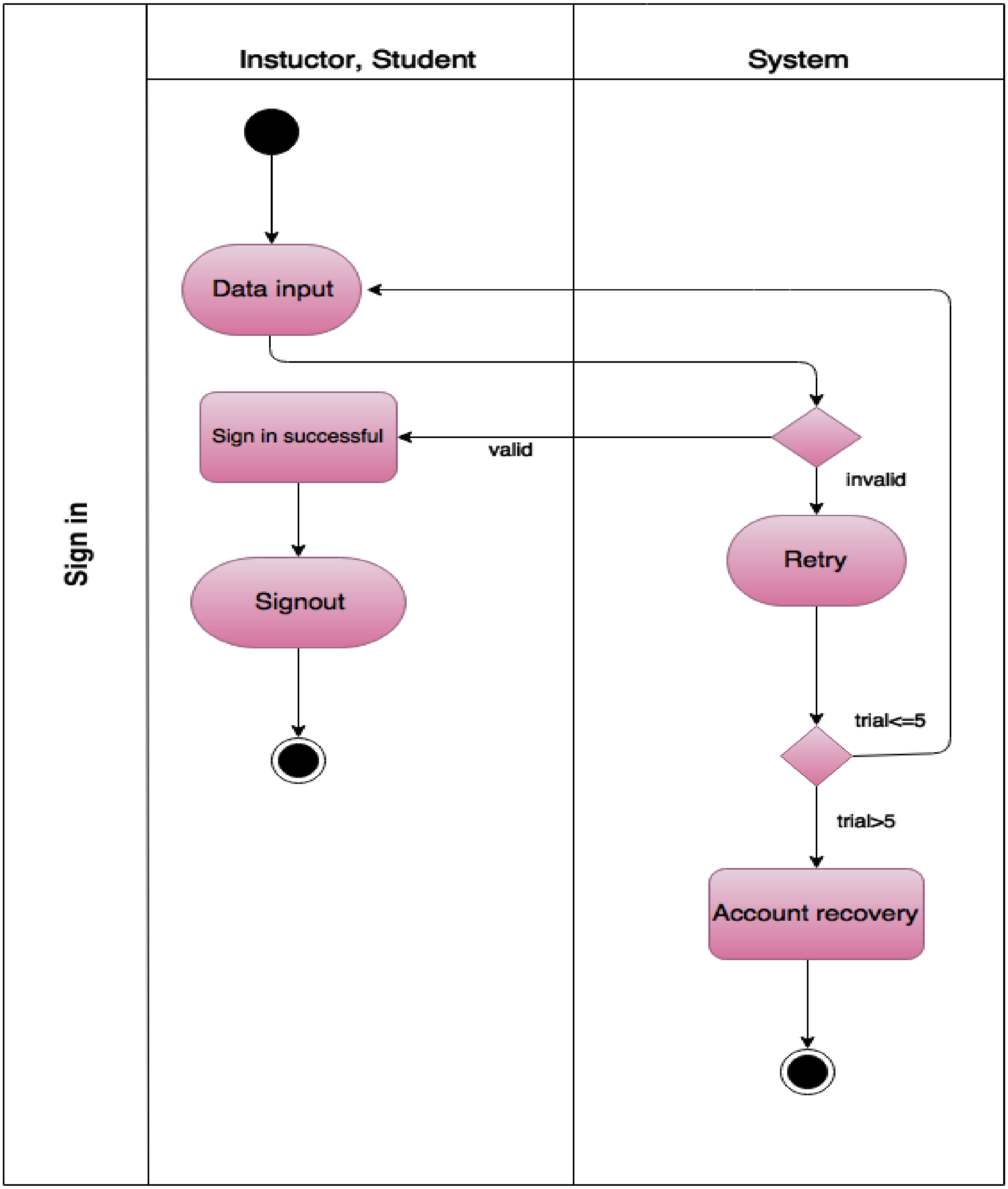


Figure – 28:Figure 28: Level 1.1.2 Swim lane diagram – Sign in.

SWIM LANE DIAGRAM – 1.1: ACCOUNT RECOVERY

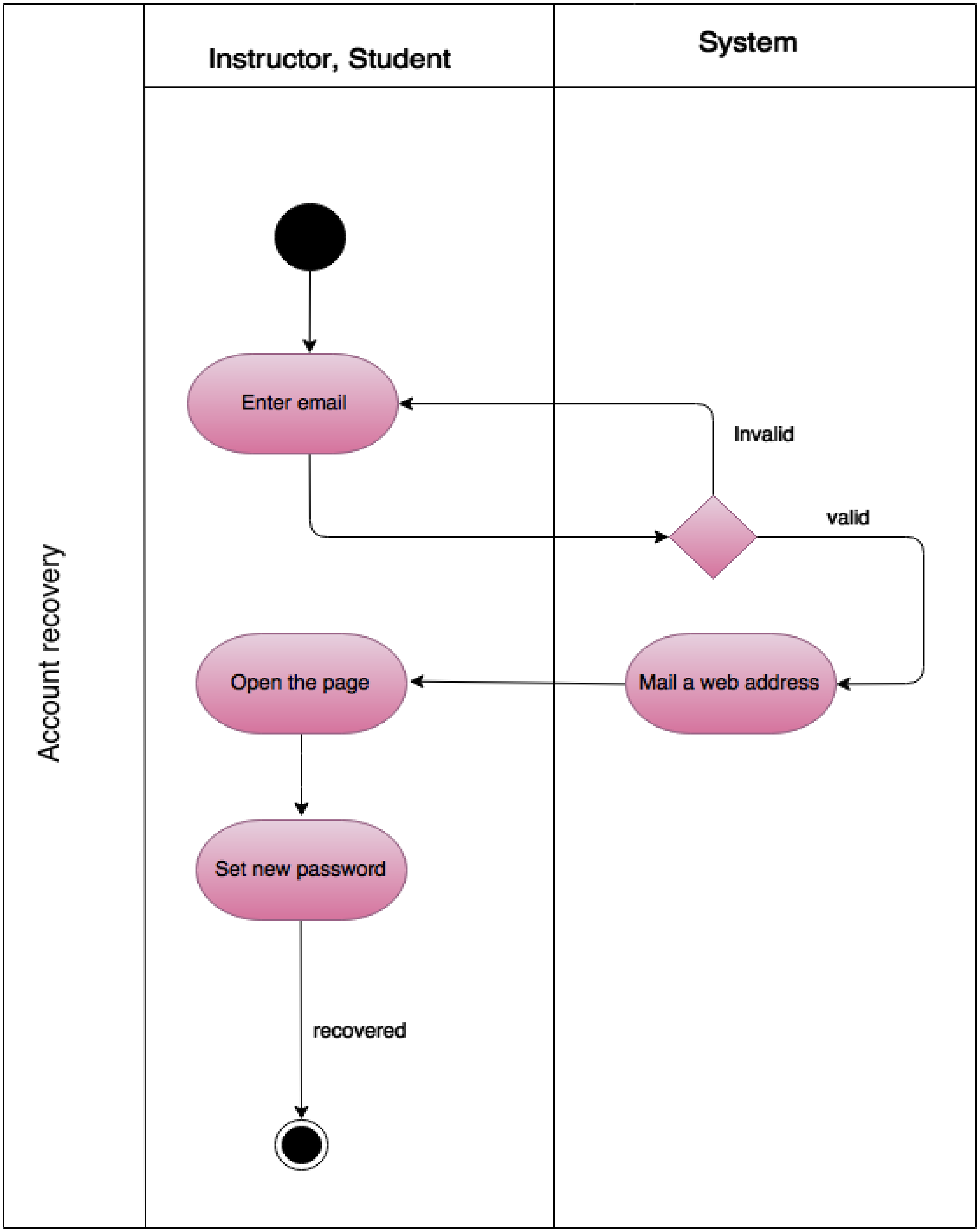


Figure 29: Level 1.1.3 Swim lane diagram – Account recovery

SWIM LANE DIAGRAM – 2: ASSIGNMENT MANAGEMENT

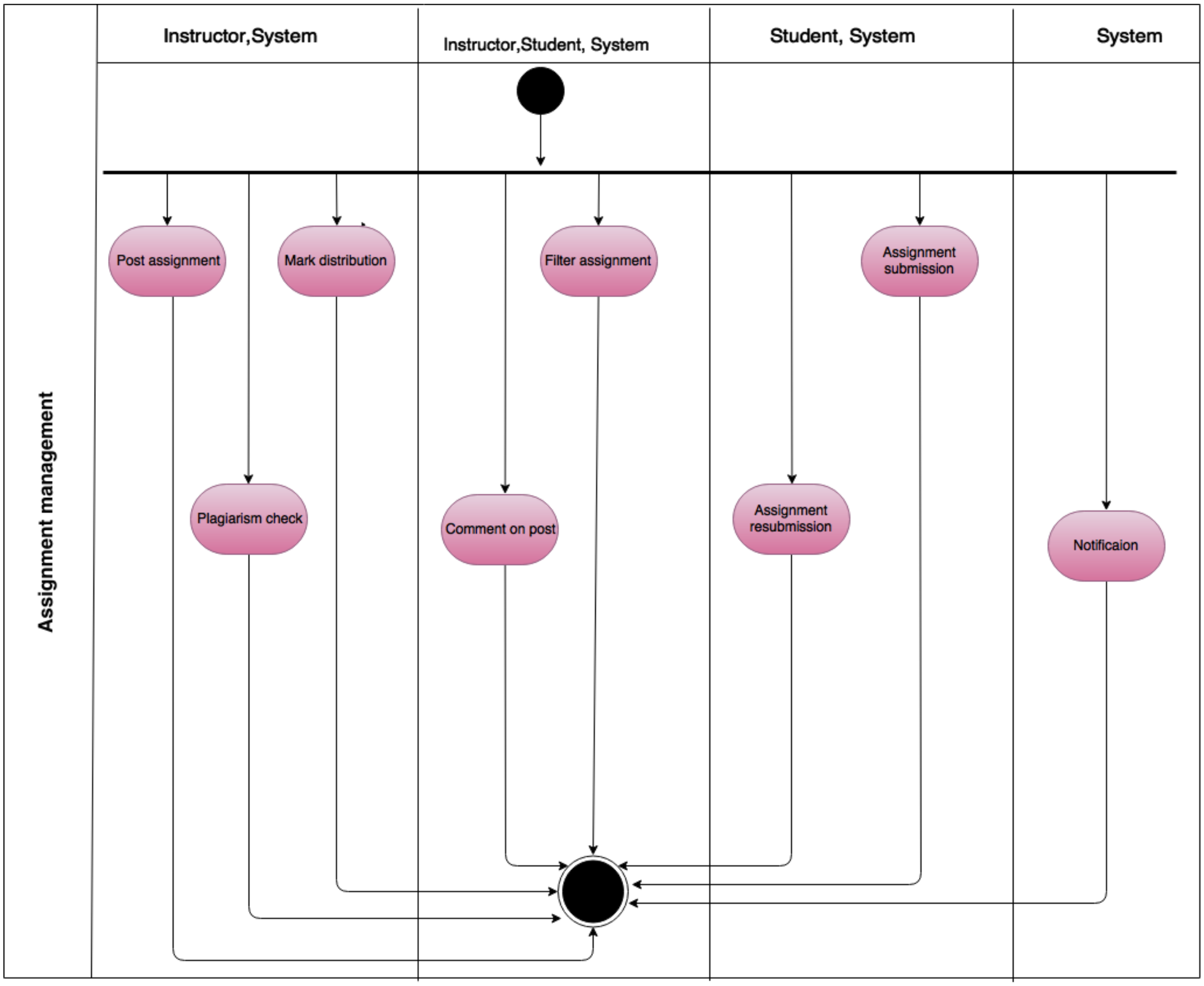


Figure 30: Level 1.2 Swim lane diagram – Assignment management

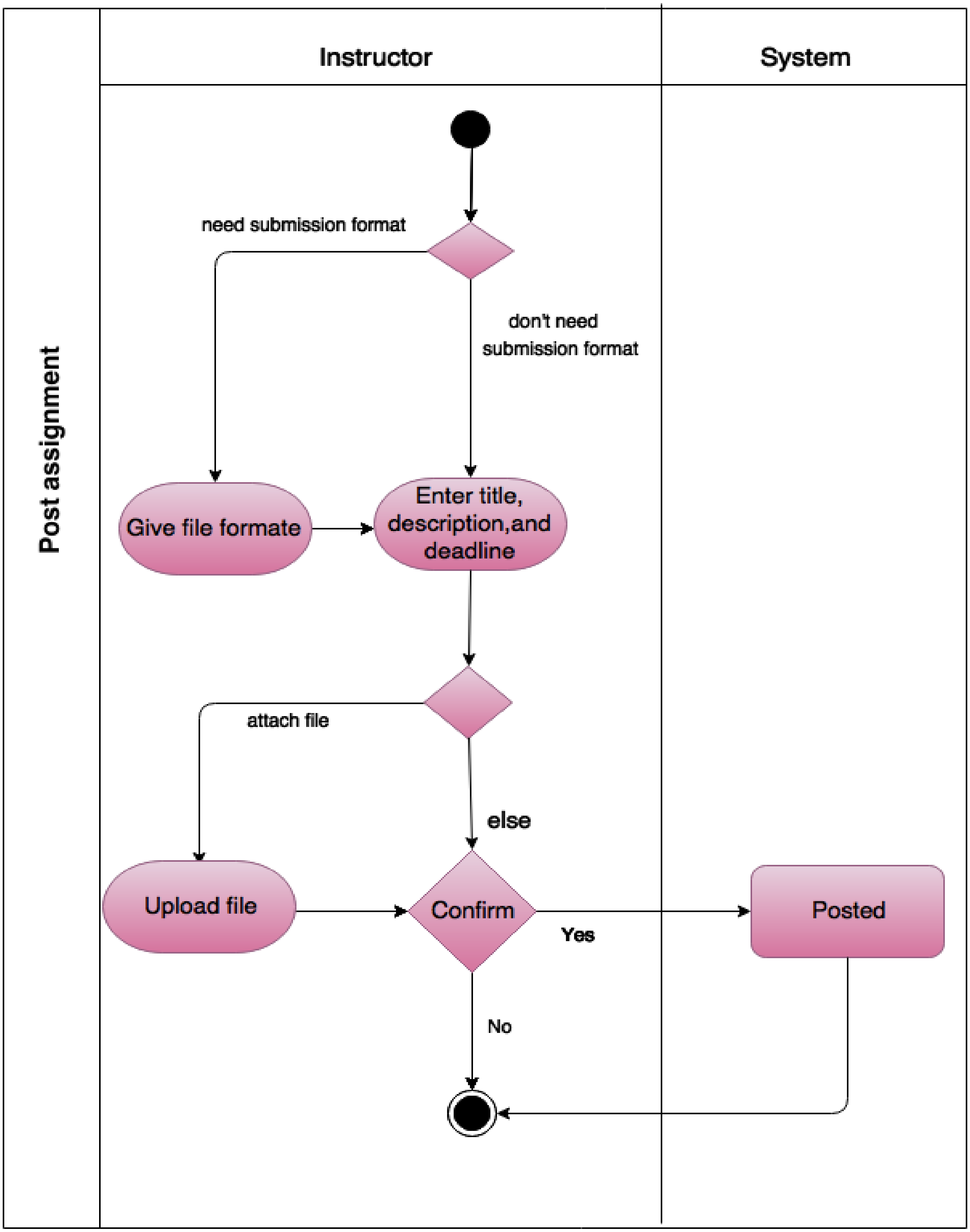


Figure 31: Level 1.2.1 Swim lane diagram – Assignment post.

SWIM LANE DIAGRAM – 2.2: ASSIGNMENT SUBMISSION

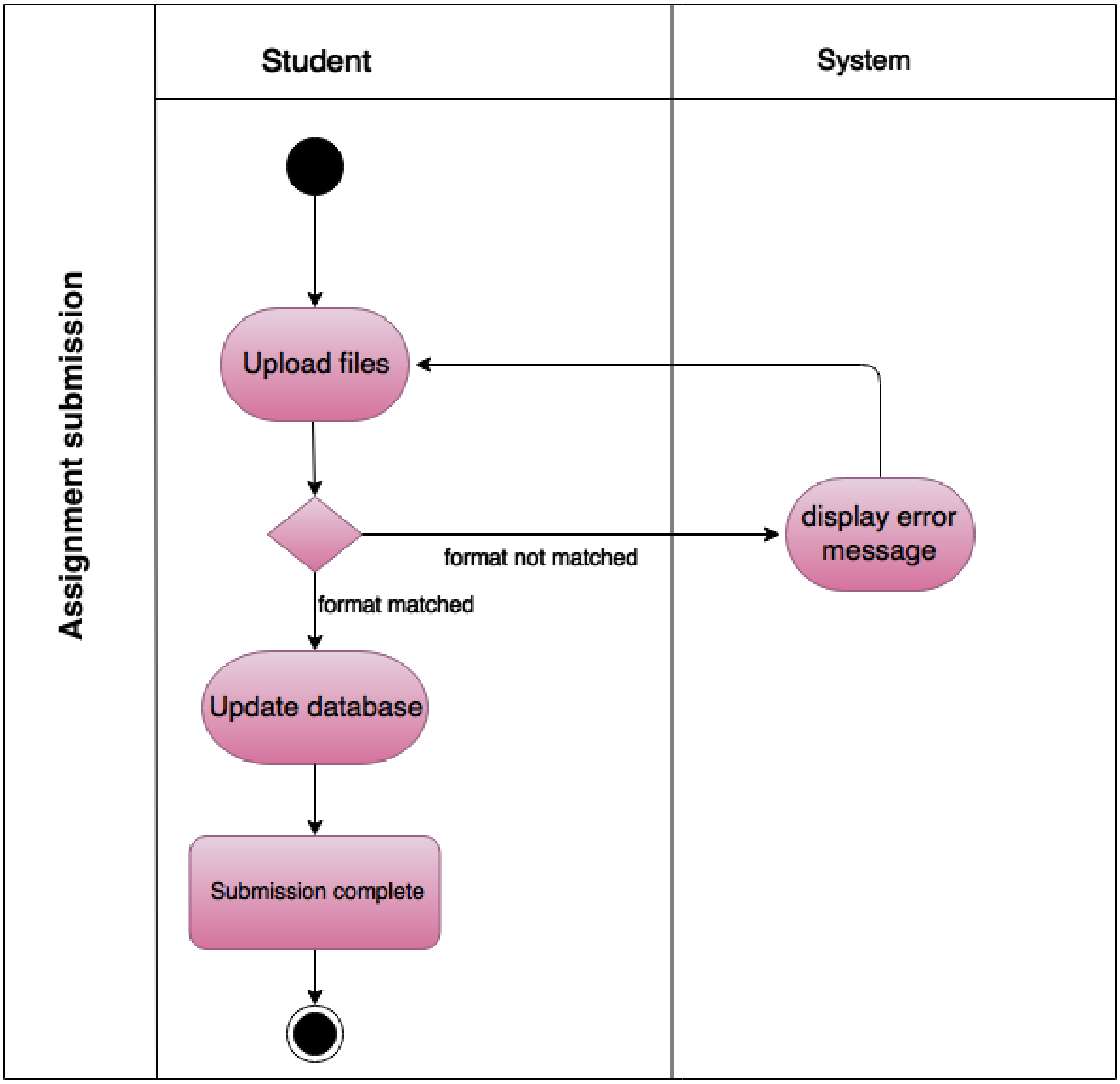


Figure 32: Level 1.2.2 Swim lane diagram – Assignment submission.

SWIM LANE DIAGRAM – 2.3: COMMENT

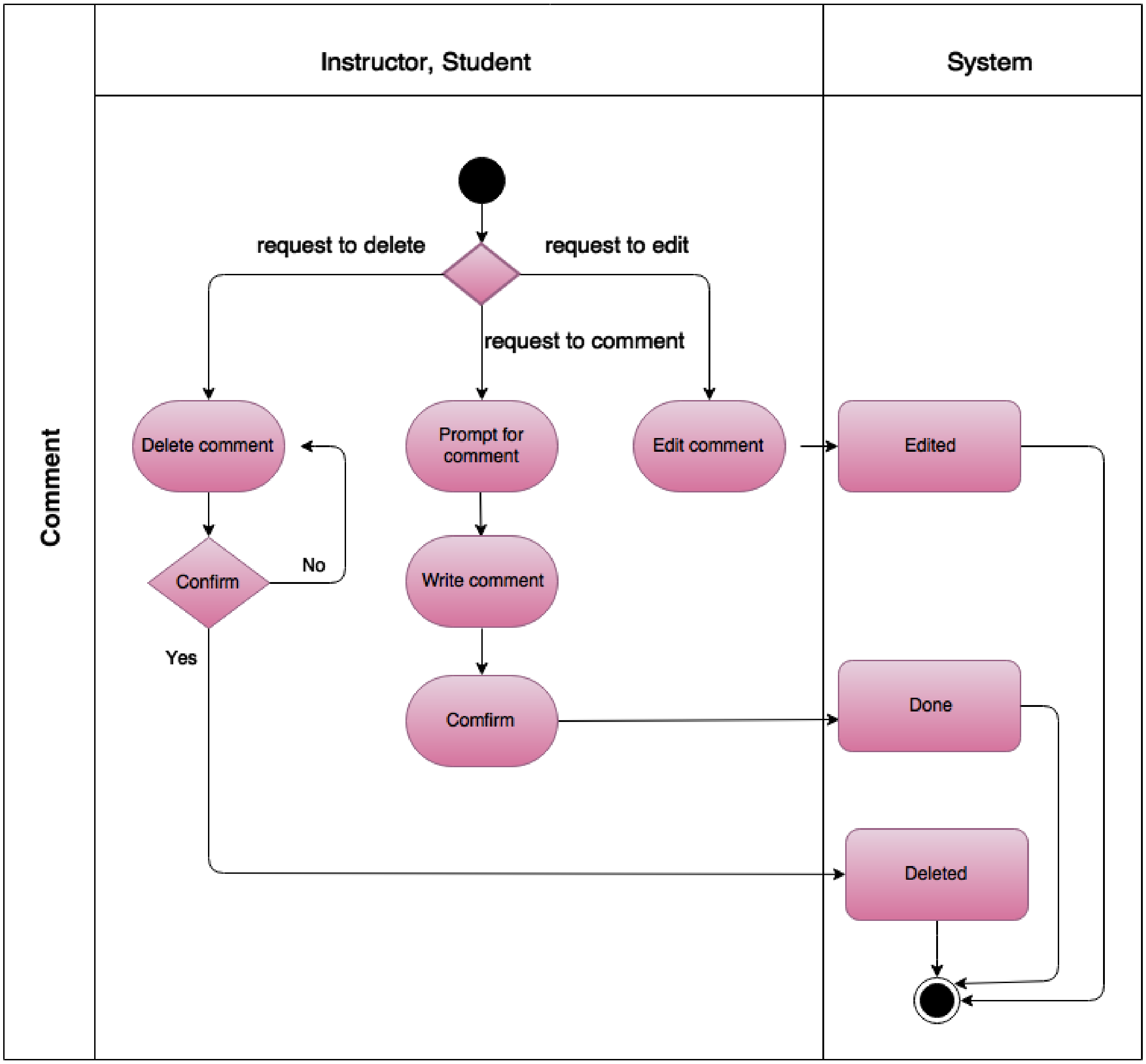


Figure 33: Level 1.2.3 Swim lane diagram – Comment.

SWIM LANE DIAGRAM – 2.4: ASSIGNMENT RESUBMISSION

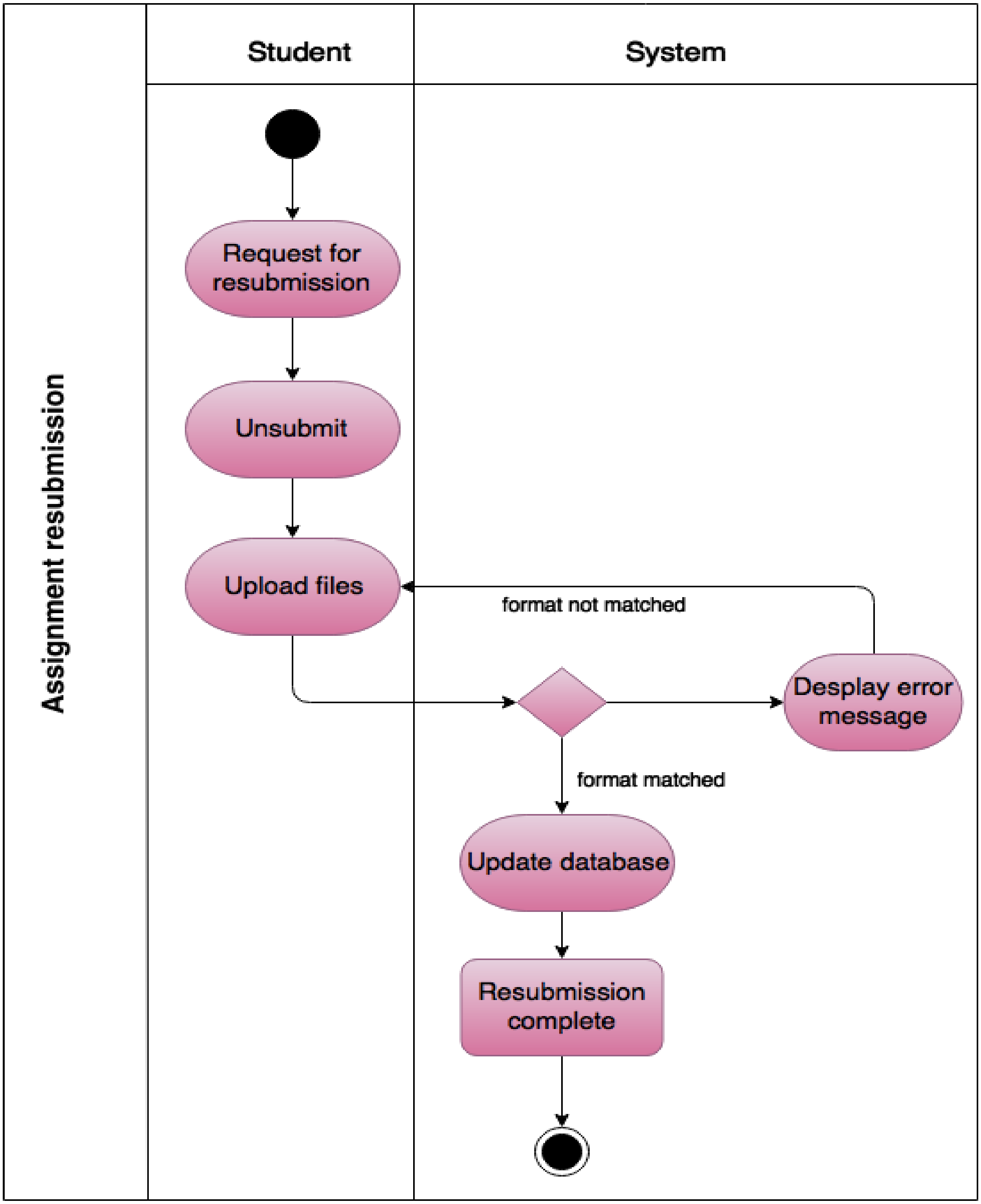


Figure 34: Level 1.2.4 Swim lane diagram – Assignment resubmission.

SWIM LANE DIAGRAM – 2.5: FILTER ASSIGNMENT

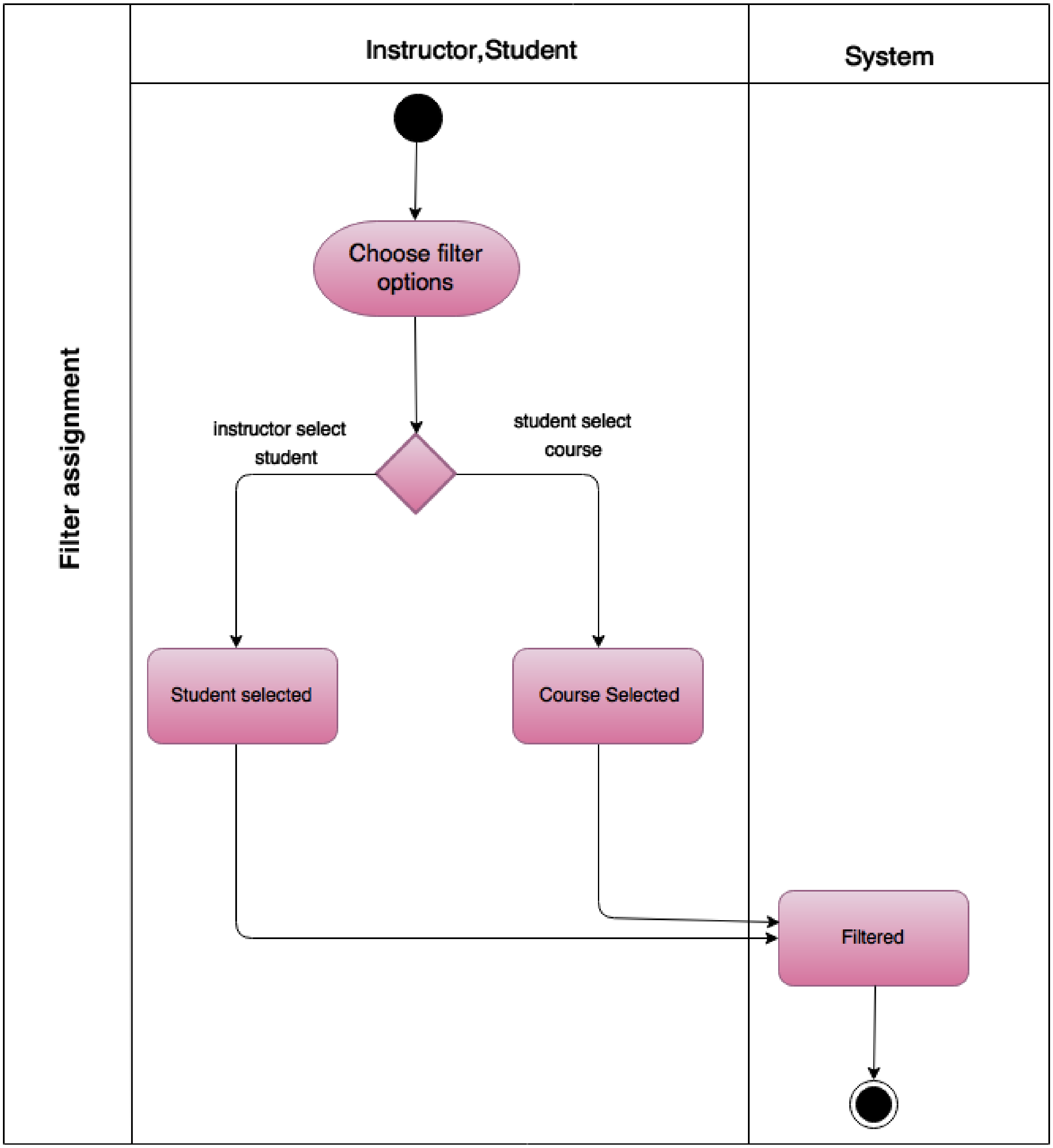


Figure 35: Level 1.2.5 Swim lane diagram – Filter assignment.

SWIM LANE DIAGRAM – 2.6: PLAGIARISM CHECK

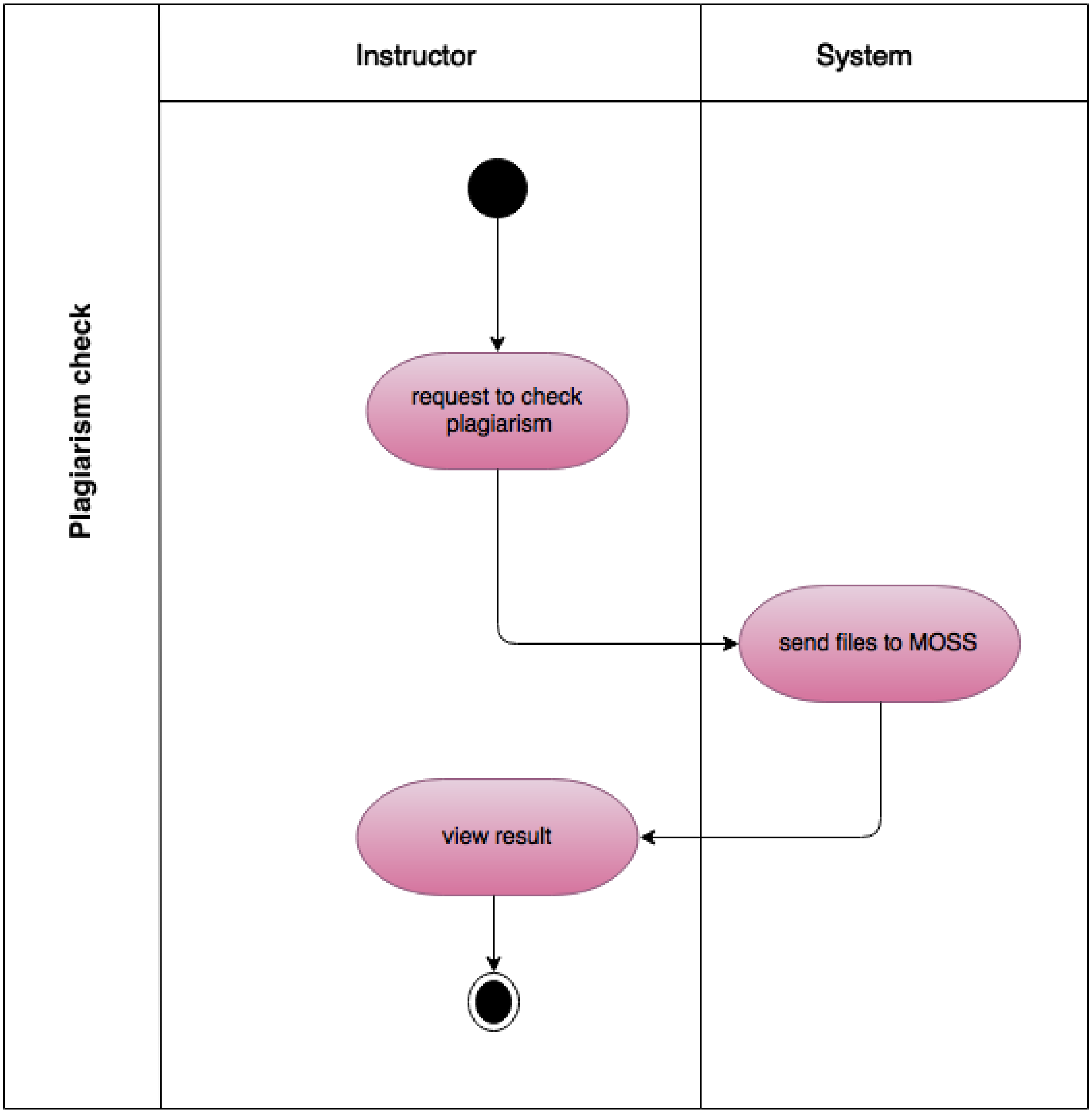


Figure 36: Level 1.2.6 Swim lane diagram – Sign in.

SWIM LANE DIAGRAM – 2.7: NOTIFICATION

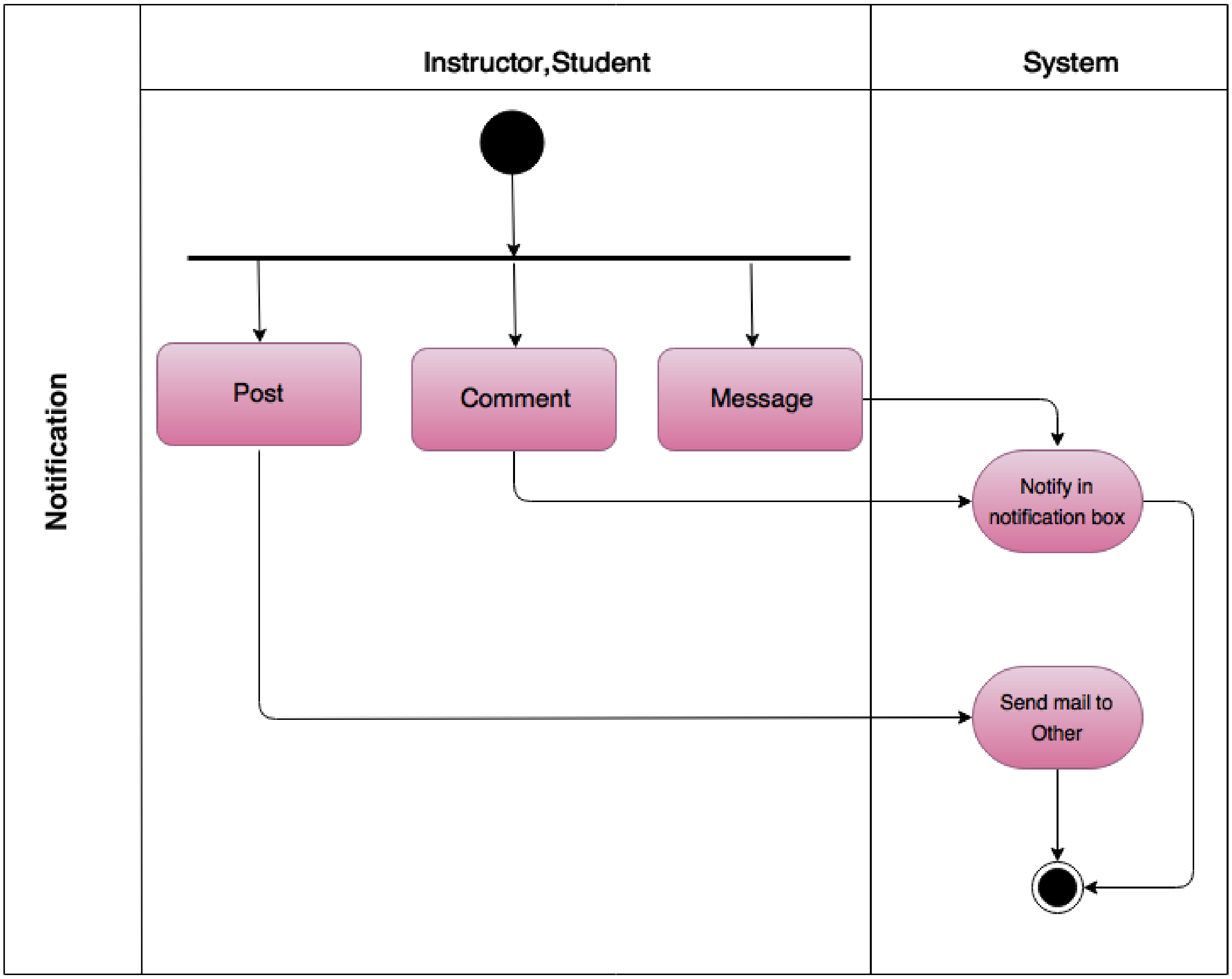


Figure 37: Level 1.2.7 Swim lane diagram – Notification.

SWIM LANE DIAGRAM – 2.8: MARK DISTRIBUTION

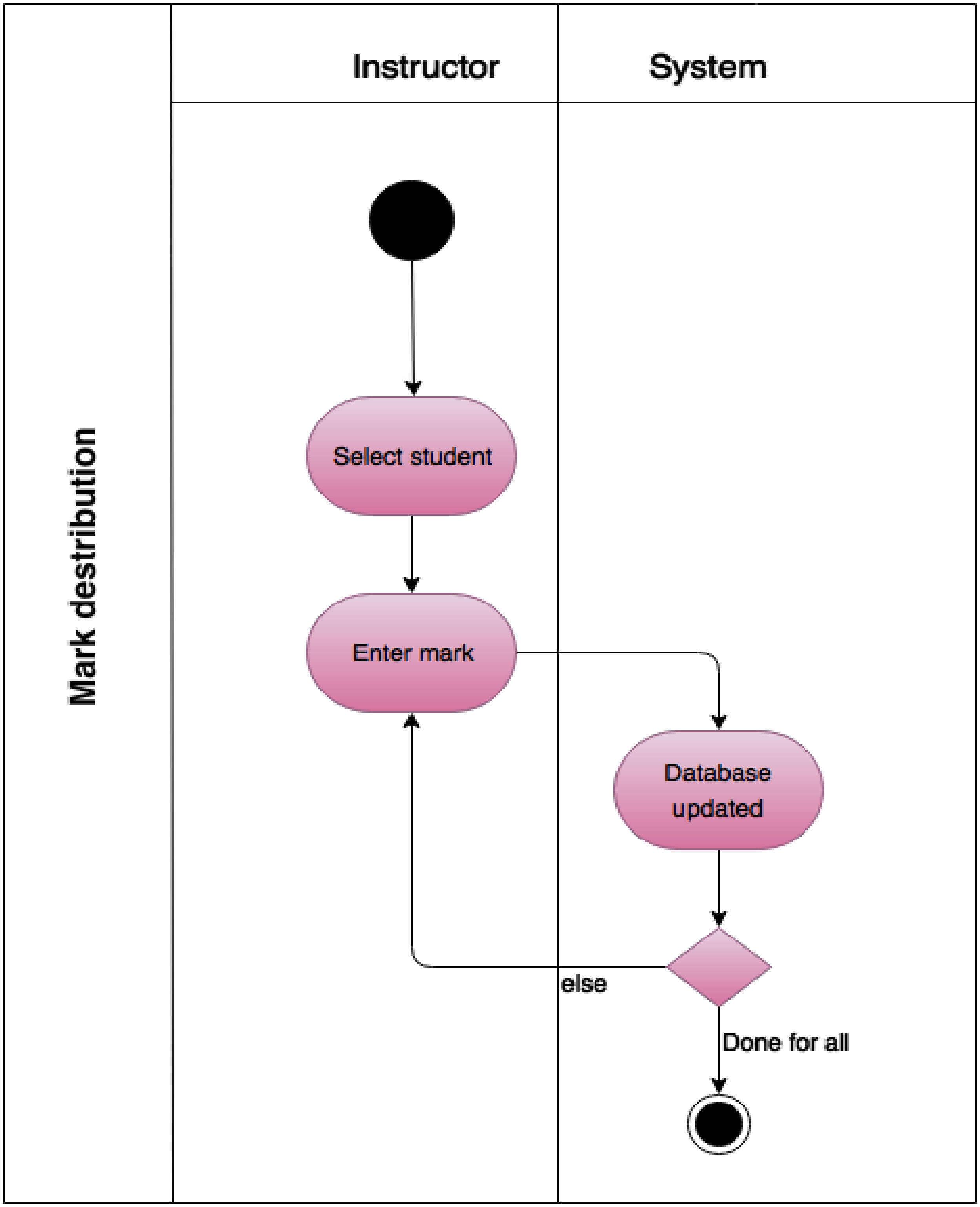


Figure 38: Level 1.2.8 Swim lane diagram – Mark distribution

SWIM LANE DIAGRAM – 3: GROUP MANAGEMENT

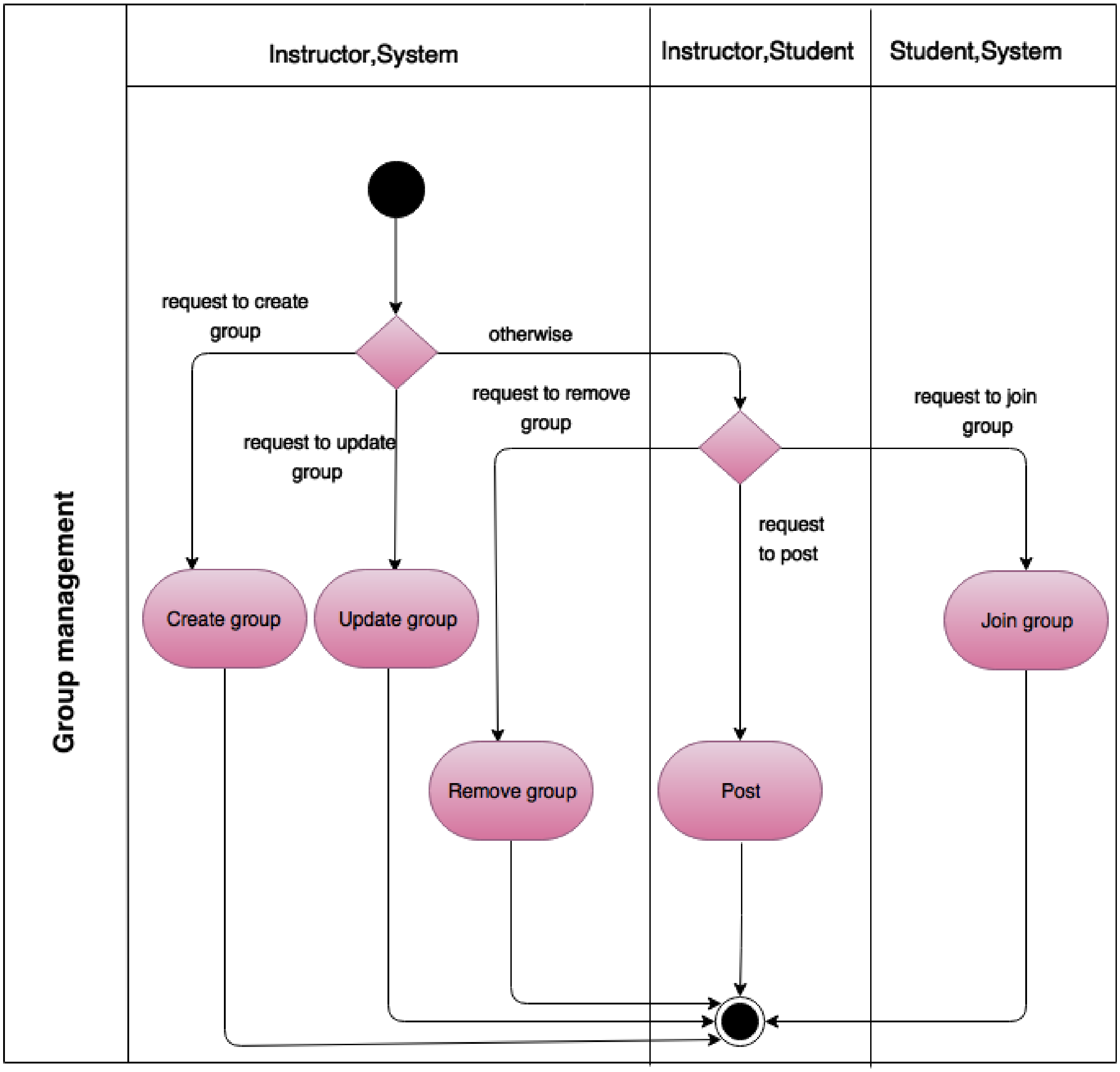


Figure 39: Level 1.3 Swim lane diagram – Group management

SWIM LANE DIAGRAM – 3.1: CREATE GROUP

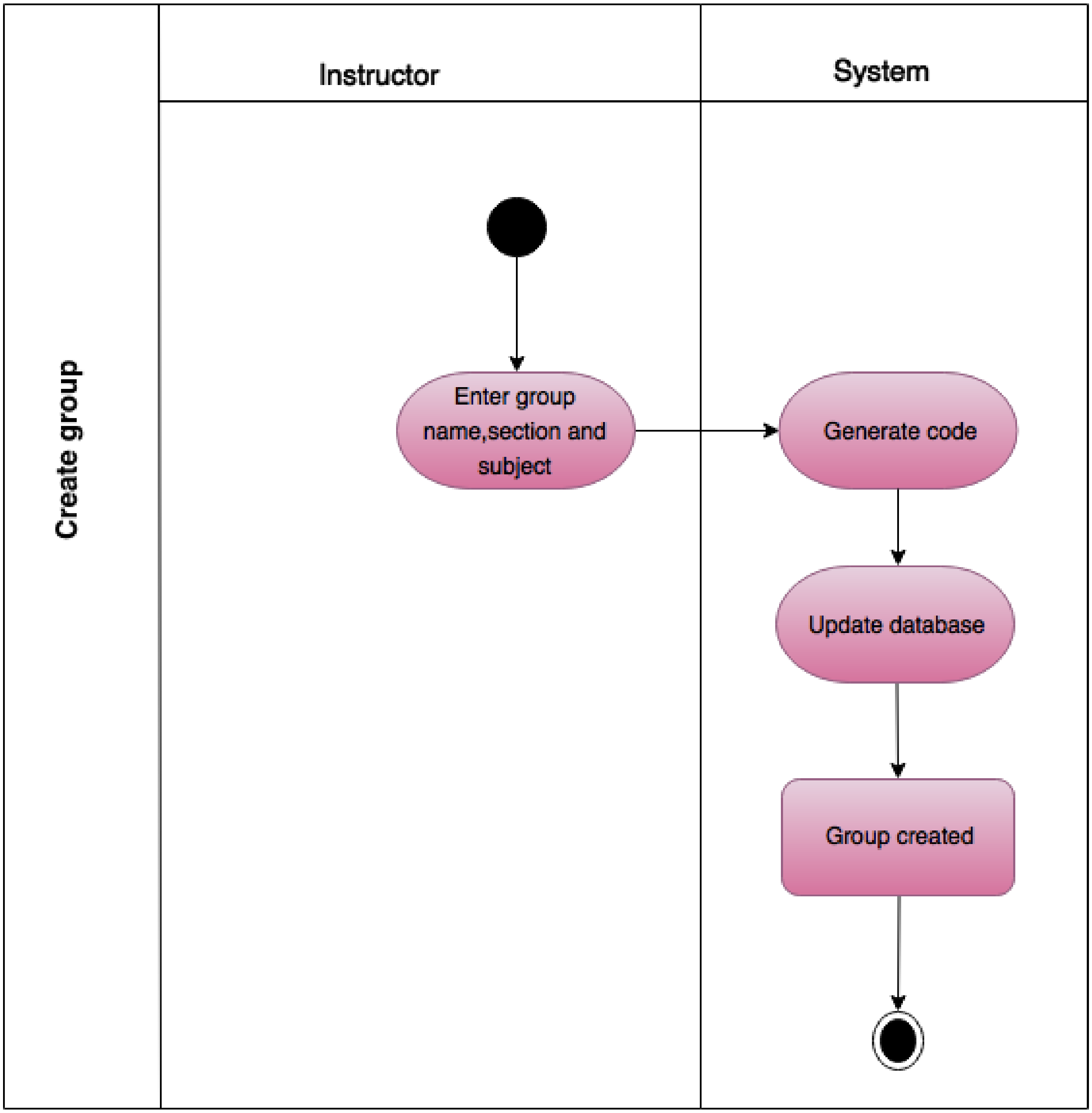


Figure 40: Level 1.3.1 Swim lane diagram – Create group

SWIM LANE DIAGRAM – 3.2: JOIN GROUP

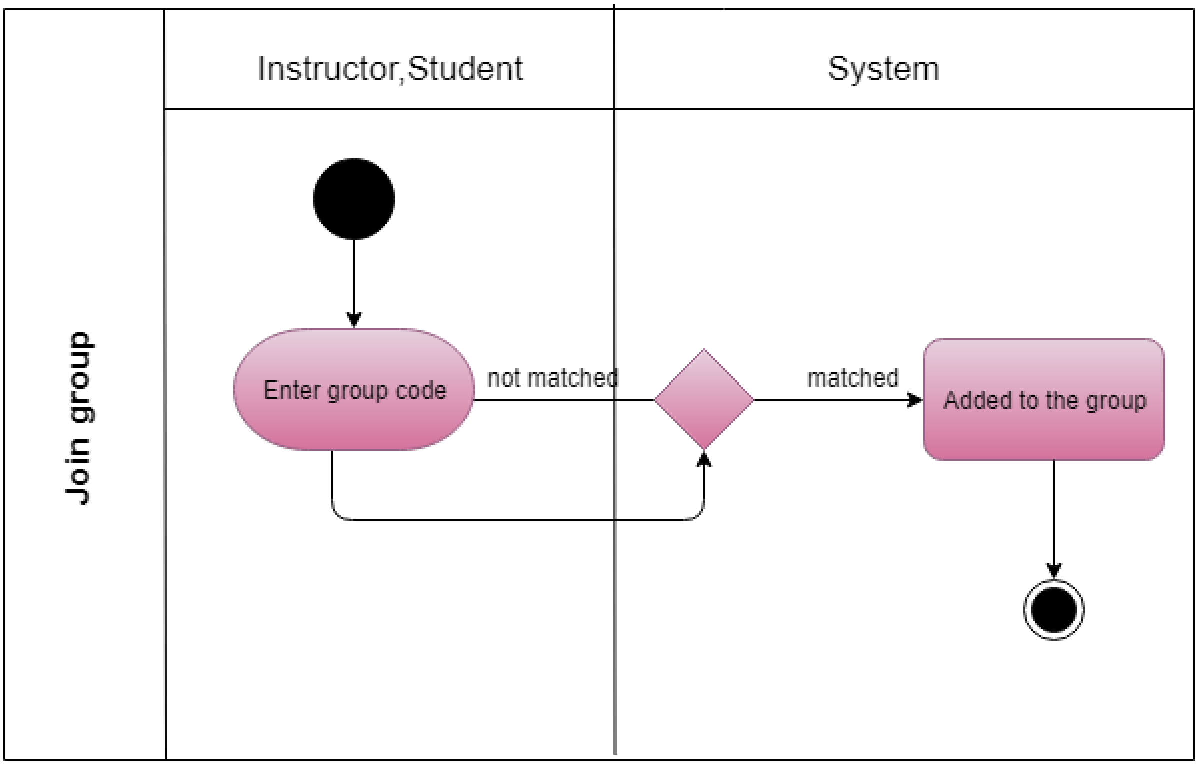


Figure 41: Level 1.3.2 Swim lane diagram – Join group

SWIM LANE DIAGRAM – 3.3: UPDATE GROUP

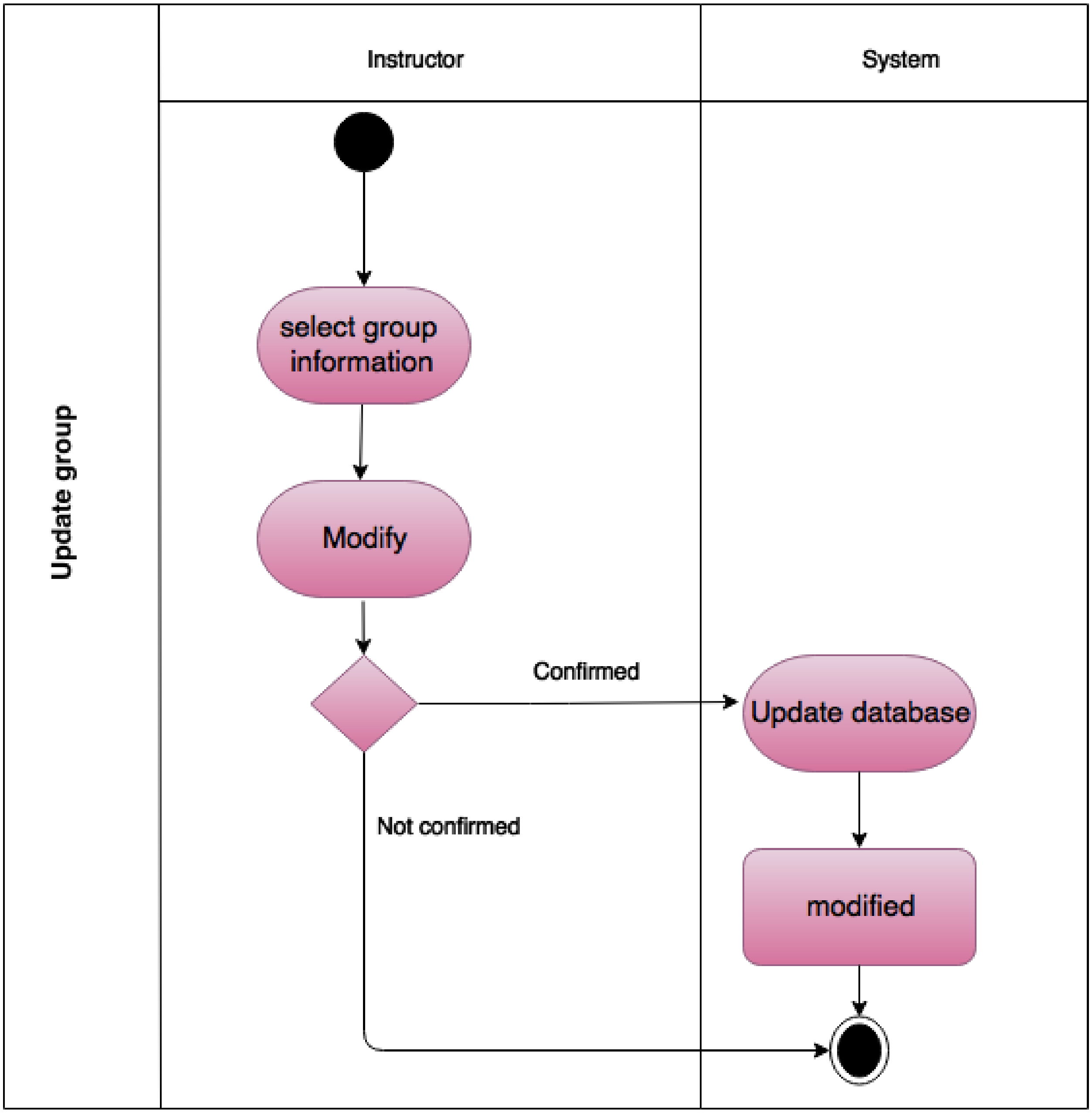


Figure 42: Level 1.3.3 Swim lane diagram – Update group

SWIM LANE DIAGRAM – 3.4: REMOVE GROUP

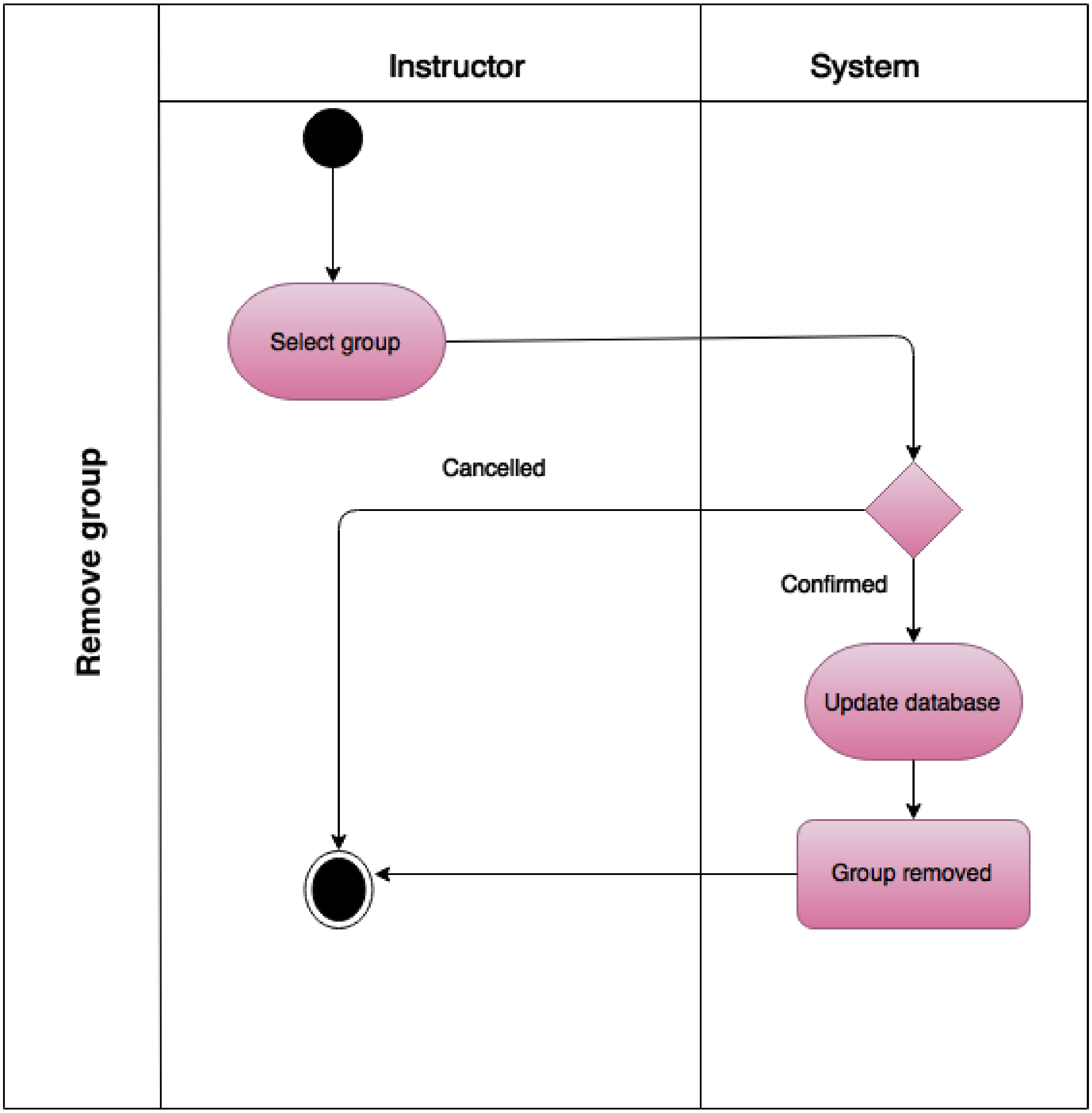


Figure 43: Level 1.3.4 Swim lane diagram – Remove group

SWIM LANE DIAGRAM – 3.5: POST

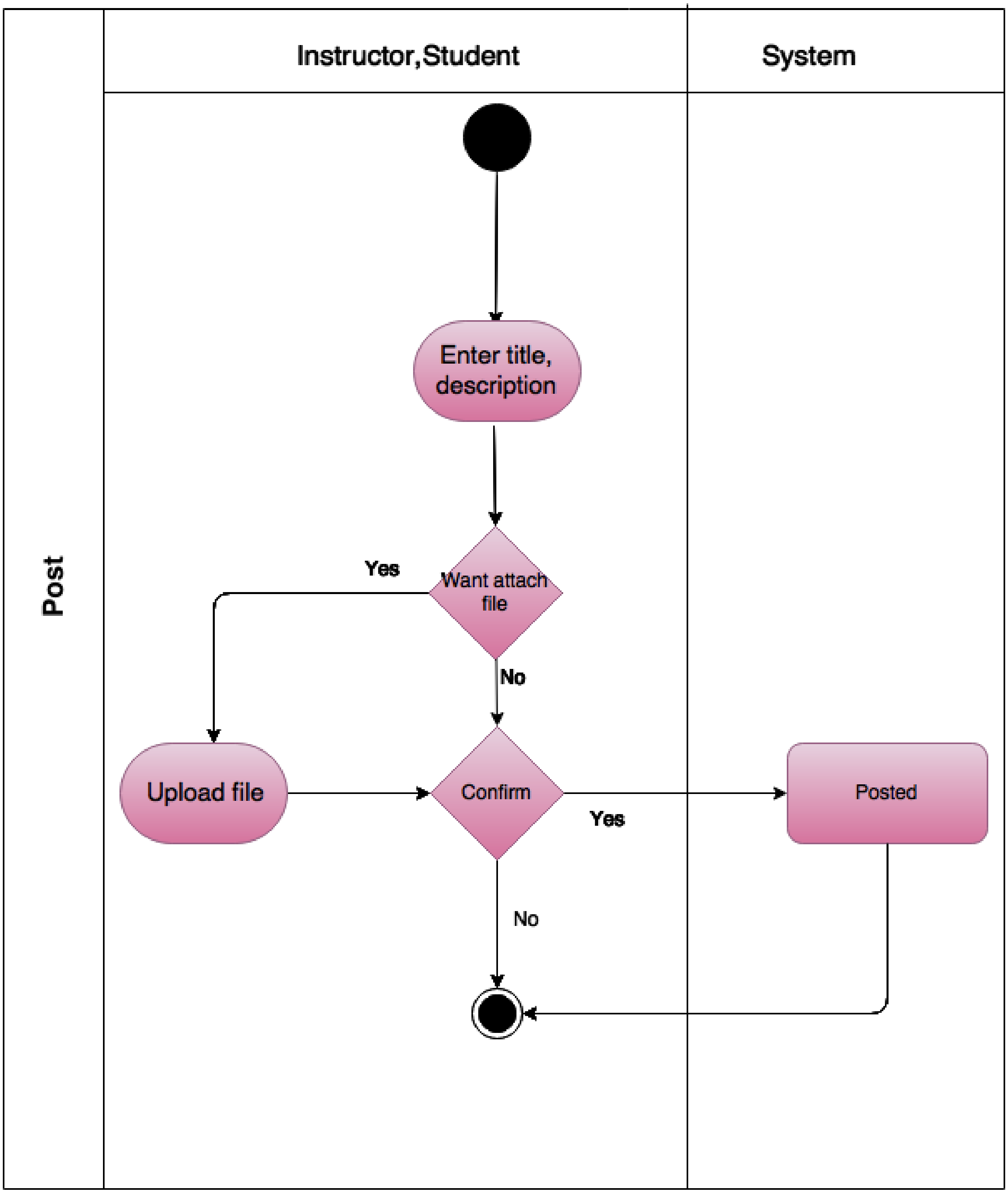


Figure 44: Level 1.3.5 Swim lane diagram – Post

CHAPTER 5: DATA BASED MODELING OF ASM

This chapter describes the Scenario-Based Model for the Assignment Management System.

5.1 INTRODUCTION

Sometimes software requirements include the necessity to create, extend or interact with a database or complex data structures need to be constructed and manipulated. The software team chooses to create data models as a 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 Modelling

Serial	Noun	S/P	Attributes
1	story	p	
2	Requirement	P	
3	Collection	P	
4	Instructor	S	11,12,13,14,15

5	Student	S	11,12,13,14,15
6	Project	P	
7	Application	P	
8	System	P	
9	User	S	11,12,13,14,15
10	Option	P	
11	First name	S	
12	Last name	S	
13	Email	S	
14	Phone number	S	
15	Password	S	
16	Account	P	
17	Data	P	
18	Validity	P	
19	Character	P	
20	Number	P	
21	Regex	P	
22	Sign up	S	
23	Sign in	S	
24	Database	S	

25	Account recovery	S	
26	Link	P	
27	Webpage	P	
28	Sign out	S	
29	Process	P	
30	Group	S	31,32,33,34
31	Group name	S	
32	Section	S	
33	Subject	S	
34	Code	S	
35	Assignment	S	36,39,40,41,42
36	Format	S	
37	Time	P	
38	Assignment submission	S	37,42,67
39	Assignment title	S	
40	Assignment description	S	
41	Deadline	S	
42	Files	S	
43	Message	S	17, 37,67
44	Comment	S	17,37,67

45	Posts	S	37,47,67
46	Announcement	P	
47	Topic	P	
49	Resubmission	S	37,42,67
50	Folder	P	
51	GroupWise	P	
52	Course wise	P	
53	Error message	P	
54	Late submission	S	37,42,67
55	Plagiarism	S	
56	MOSS	P	
57	Software	P	
58	Result	P	
59	Similarity	P	
60	Evaluation	P	
61	Marks	S	
62	Mark distribution	S	
63	Authentication	S	
64	Assignment management	S	
65	Group management	S	

66	Communication	S	
67	Date	S	

5.2.2 POTENTIAL DATA OBJECTS

- **User:**11-15
- **Student:**11-15
- **Instructor:**11-15
- **Group:**31-34
- **Assignment:** 36,39-42
- **Assignment submission:**37,42,67
- **Message:**37,67,68
- **Comment:**37,67,69
- **Posts:**37,47,67
- **Resubmission:**37,42,67
- **Late submission:**37,42,67

5.2.3 ANALYSIS FOR FINAL DATA OBJECT

- Instructor and student are all users of AMS and thus common attributes stored as data object User.
- Group stores group information such as group name, subject, and section code.
- Assignment stores assignment information such as assignment description, format, deadline, and files.
- Assignment submission, late submission, and resubmission are all submission of AMS and thus common attributes stored as data object Submission and these are under submit action.
- Message stores information sender, receiver, date, time and message description.
- Comment stores information such as comment description, date and time.
- Posts are the assignment posted in the group by the instructor.

5.2.4 FINAL DATA OBJECT

1 User: user_Id, first name, last name, email, password, phone number

2 Instructor: user_Id, first name, last name, email, password, phone number

3 Student: user_Id, first name, last name, email, password, phone number

4 Group: group_Id, group name, section, subject, code

5 Assignment: assignment_Id, format, title, description, file

6 Message: time, date, data,

7 Comment: time, date, data,

5.3 DATA OBJECT RELATIONS

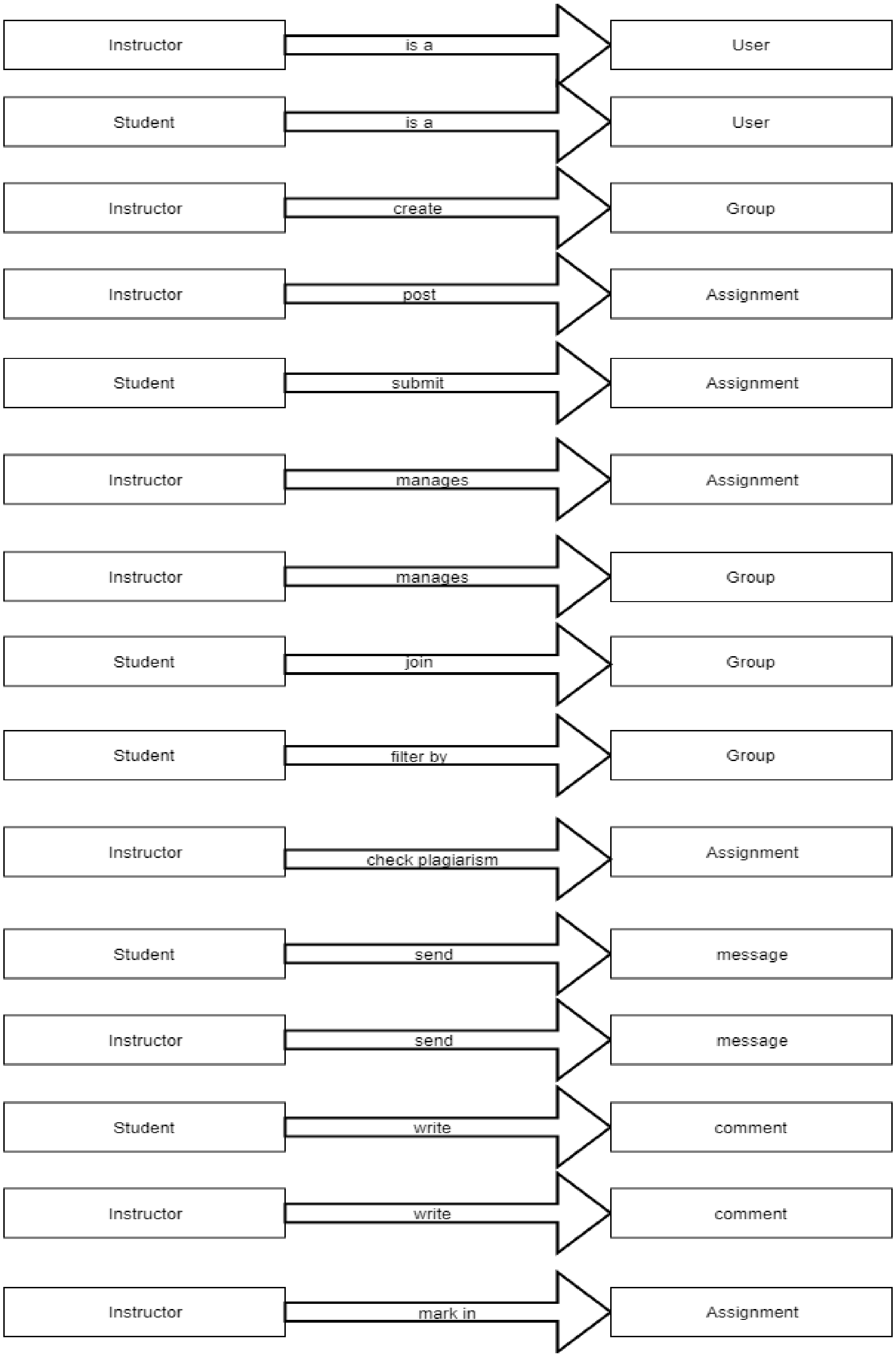


Figure 45: Relationships between data objects

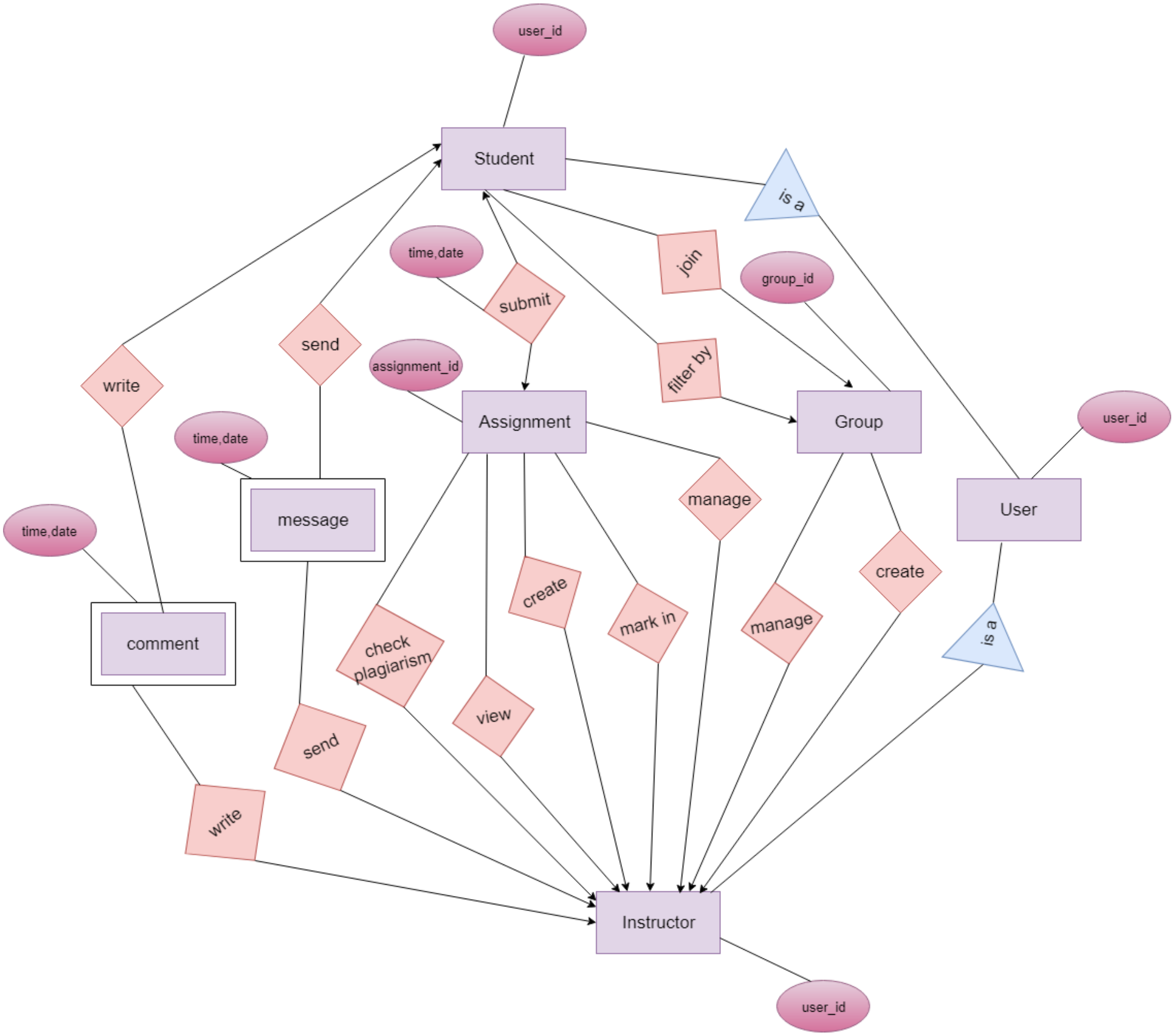


Figure 46: Entity Relationship of Assignment Management System

5.5 SCHEMA DIAGRAM

A schema is a structure behind data organization. In a schema diagram, all database table are designated with unique columns and special features, e.g., primary key, foreign keys.

Table 2: schema table of User data object

User		
Attributes	**Type**	**Size**
user_Id	VARCHAR	40
first_name	VARCHAR	80
last_name	VARCHAR	40
email	VARCHAR	50
password	VARCHAR	30
phone_number	VARCHAR	15

Table 3: schema table of Instructor data object

Instructor		
Attributes	**Type**	**Size**
user_Id	VARCHAR	40
first name	VARCHAR	80
last name	VARCHAR	40
email	VARCHAR	50
password	VARCHAR	30
phone number	VARCHAR	15

Table 4: schema table of Student data object

Student		
Attributes	**Type**	**Size**
user_Id	VARCHAR	40
first_name	VARCHAR	80
last_name	VARCHAR	40
email	VARCHAR	50
password	VARCHAR	30
phone number	VARCHAR	15
group_Id	VARCHAR	40

Table 5: schema table of Group data object

Group		
Attributes	**Type**	**Size**
group_Id	VARCHAR	40
group name	VARCHAR	80
section	VARCHAR	40
subject	VARCHAR	50
code	VARCHAR	30

Table 6: schema table of Assignment data object

Assignment		
Attributes	**Type**	**Size**
assignment_Id	VARCHAR	40
format	VARCHAR	80
title	VARCHAR	40
phone_number	VARCHAR	50
description	VARCHAR	500
file	BLOB	1
creation date	DATE	30

Table 7: schema table of Message data object

Message		
Attributes	**Type**	**Size**
user_id	VARCHAR	40
time	VARCHAR	40
date	DATE	80
data	VARCHAR	40

Table 8: schema table of comment data object

Comment		
Attributes	**Type**	**Size**
user_id	VARCHAR	40
time	VARCHAR	40
date	DATE	80
data	VARCHAR	40

Table 9: schema table of Assignment submission data object

Assignment Submission		
Attributes	**Type**	**Size**
user_Id	VARCHAR	40
assignment_Id	VARCHAR	40
time	VARCHAR	80
date	DATE	40
Files	BLOB	1

Table 10: schema table of Marks data object

Marks		
Attributes	**Type**	**Size**
user_Id	VARCHAR	40

assignment_Id	VARCHAR	40
marks	NUMBER	80

CHAPTER – 6 CLASS BASED MODELING

This chapter describes the class-based model for Assignment Management System.

6.1 INTRODUCTION

Class-based methods for requirements modeling use common concepts of object-oriented programming to craft an impression of an application that can be understood by nontechnical stakeholders. As the requirements model is refined and expanded, it evolves into a specification that can be used by software engineers in the creation of the software design. Class-based modeling represents:

1. The objects the system will manipulate
2. The operations (methods or services) that will be applied for effective manipulation
3. The relationships between the objects
4. The collaboration that occurs between the classes

6.2 IDENTIFYING ANALYSIS CLASS

Classes are identified by underlining each noun or noun phrase and plotting it into a simple table. If the class (noun) is required to implement a solution, then it becomes a part of the solution space. Otherwise, if the noun is used only to describe a solution, it is regarded as a part of the problem space. Once all the nouns have been isolated, General classification and Selection is done.

6.2.1 GENERAL CLASSIFICATION

Nouns belonging to the solution space should exhibit any of the following criteria to be considered as a class. The 7 general characteristics are stated below:

1. *External entities:* Other systems, devices, people that produce or consume information to be used by a computer-based system
2. *Things*: Reports, displays, letters, signals that are a part of the information domain for the problem.

3. *Events*: Actions or transfers (a property transfer or the completion of a series of robot movements) that occur within the context of system operation.
4. *Roles*: Responsibilities played by the people who interact with the system.
5. *Organizational units:* Divisions, groups, teams that are relevant to an application.
6. *Places:* Platform that establishes the context of the problem and overall function of the system.
7. *Structures*: Something that defines a class of objects or related classes of objects.

Table 11: Noun with general classification

Serial Number	Noun	General classification
1	User	4,5,7
2	Instructor	4,5,7
3	Student	4,5,7
4	First name	
5	Last name	
6	Email	
7	Phone number	
8	Password	
9	Sign up	3,5
10	Sign in	3
11	Database	1
12	Account recovery	3
13	Group	5,7
14	Group name	
15	Section	
16	Subject	
17	Code	
18	Assignment	2,7
19	Format	
20	Time	

21	Assignment submission	3
22	Assignment title	
23	Assignment description	
24	Deadline	
25	File	2
26	Message	3
27	Comment	3
28	Post	3
29	Resubmission	3
30	Late submission	3
31	Plagiarism	3
32	Marks	
33	Mark distribution	3
34	Authentication	3,5
35	Assignment management	5,7
36	Group management	5,7
37	Date	
38	Message description	
39	Comment description	
40	Sign out	3

6.2.2 SELECTION CRITERIA

Classes that fulfilled at least 3 characteristics of general classification are again reconsidered by six Selection Criteria. The six characteristics for the selection criteria are:

1. *Retained information:* The potential class will be useful during analysis only if information about it must be remembered so that the system can function.
2. *Needed services:* The potential class must have a set of identifiable operations that can change the value of its attributes in some way.

3. *Multiple attributes:* During requirement analysis, the focus should be on "major" information; a class with a single attribute may, in fact, be useful during design, but is probably better represented as an attribute of another class during the analysis activity.
4. *Common attributes:* A set of attributes can be defined for the potential class and these attributes apply to all instances of the class.
5. *Common operations:* A set of operations can be defined for the potential class and these operations apply to all instances of the class.
6. *Essential requirements:* External entities that appear in the problem space and produce or consume information essential to the operation of any solution for the system will almost always be defined as classes in the requirements model.

To be considered a legitimate class for inclusion in the requirements model, a potential object should satisfy all (or almost all) of these characteristics. The decision for inclusion of potential classes in the analysis model is somewhat subjective, and later evaluation may cause an object to be discarded or reinstated.

Table 12: selection criteria of nouns

Serial number	Noun	Selection criteria
1	User	1,2,3,4,5
2	Instructor	1,2,3,4,5
3	Student	1,2,3,4,5
4	Sign up	5
5	Sign in	5
6	Account recovery	5
7	Sign out	5
8	Database	6

9	Group	3,4,5
10	Authentication	3,4,5
11	Assignment	3,4,5

6.2.3 ASSOCIATE NOUN WITH VERB

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

Table 13: Associate noun and verb identification

No	Class name	Nouns	Verbs
1	User	firstname, lastname, username, email, phone number, password	register, recover account, sign in, sign out
2	Instructor	firstname, lastname, username, email, phone number, password	Create group, register, sign in, recover account, sign out, Post assignment, give format, fill assignment, attach files, post announcement, search assignment, download assignment, view assignment, filter assignment, check plagiarism, distribute mark, comment, message, update group, remove group.
3	Student	firstname, lastname, username, email, phone number, password	Register, sign in, sign out, account recovery, join group, submit assignment, view assignment, download assignment, filter assignment, comment, message, cancel submission, resubmit assignment,

4	Authentication	N/A	Data entry, validity check, checking running process.
5	Database	N/A	Store/provide information
6	Assignment	Assignment id, title, description, file	N/A
7	Group	Group id, subject, code, section	N/A

6.2.4 ATTRIBUTE SELECTION

Table 14: class and attributes selection

No	Class	Attributes
1	User	firstName lastName userName Password Email Phone number
2	Instructor	firstName lastName userName Password Email Phone number

		Type
3	Student	firstName lastName userName Password Email Phone number Type
4	Authentication	Name Password Email Phone number
5	Database	DBname Password tableName url
6	Assignment	Assignment id Assignment title Assignment description Deadline
7	Group	Group id

		Group code Group name Section Subject

6.2.5 METHOD IDENTIFICATION

Table 15: Method identification

No	Class	Methods

1	Authentication	signIn()
		signUp()
		accountRecovery()

2	User	sendMessage()
		receiveMessage()
		logout()
		createGroup()
		filter()
		post()

3	Instructor	distributeMarks()
		checkPlagiarism()
		createAssignment()
		removeGroup()
		updateGroup()
		searchAssignment()

4	Student	submitAssignment()
		resubmitAssignment()
		joinGroup()

5	Assignment	toString()
6	Group	toString()

7	System	takeInput()
		validateInput()
		verifyInput()
		generateId()
		generateCode()
		checkingFormatAndDeadline()
		storeAssignment()

8	Database	insert()
		view()
		update()

		remove()
		retrieve()

6.2.6 CLASS CARDS

After identifying our final classes, we have generated following class cards

Table 16: User

User	
Attributes	Methods

firstName
lastName
userName
email
password
phoneNumber
	senndMessage()
	receiveMessage()
	signOut()
	createGroup()
	comment()
	filter()
	post()

Responsibilities	Collaborative class
➢ Sending message to users ➢ Receiving message from users ➢ Creating group ➢ Commenting on the post ➢ Filtering assignment ➢ Making post ➢ Sign out from system	System, Database, Group

Table 17: Instructor

Instructor	
Attributes	Methods

firstName
lastName
userName
email
password
phoneNumber
type
	distributeMarks()
	checkPlagiarism()
	createAssignment()
	removeGroup()
	updateGroup()
	searchAssignment()

Responsibilities	Collaborative class
➢ Creating assignment ➢ Distributing mark ➢ Checking plagiarism ➢ Removing group ➢ Updating group ➢ Searching assignment	System, Database, Assignment

Table 18: Student

Student	
Attributes	Methods

firstName
lastName
userName
email
	submitAssignment()
	resubmitAssignment()
	joinGroup()

password phoneNumber type	
Responsibilities	Collaborative class
➢ Submitting assignment ➢ Resubmitting mark ➢ Joining to group	System, Database

Table 19: Authentication

Authentication	
Attributes	Methods

firstName
lastName
userName
email
password
phoneNumber
type
	signIn()
	signUp()
	accountRecovery()

Responsibilities	Collaborative class

➢ Registration to the system ➢ Log in to the system ➢ Recovery of user account	System, Database, Instructor, Student

Table 20: System

System	
Attributes	Methods

id
code
	takeInput()
	validateInput()
	verifyInput()
	generateId()
	generateCode()
	checkingFormatAndDeadline()
	storeAssignment()

Responsibilities	Collaborative class
➢ Taking input ➢ Validating Input ➢ Generating id and class code ➢ Checking format of assignment and deadline ➢ Storing assignment	Database,User, Instructor, Student

Table 21: Database

Database	
Attributes	Methods

DB_name
Password
table_Name
url
	insert()
	view()
	update()
	remove()
	retrieve()

Responsibilities	Collaborative class
➢ Storage system information ➢ Manipulation of stored information	N/A

Table 22: Assignment

Assignment	
Attributes	Methods
Assignment id Assignment title Assignment description Deadline	toString()
Responsibilities	Collaborative class
N/A	N/A

Table 23: Group

Group	
Attributes	Methods
Group id Group code Group name Section Subject	toString()
Responsibilities	Collaborative class
N/A	N/A

6.2.7 CLASS COLLABORATION DIAGRAM

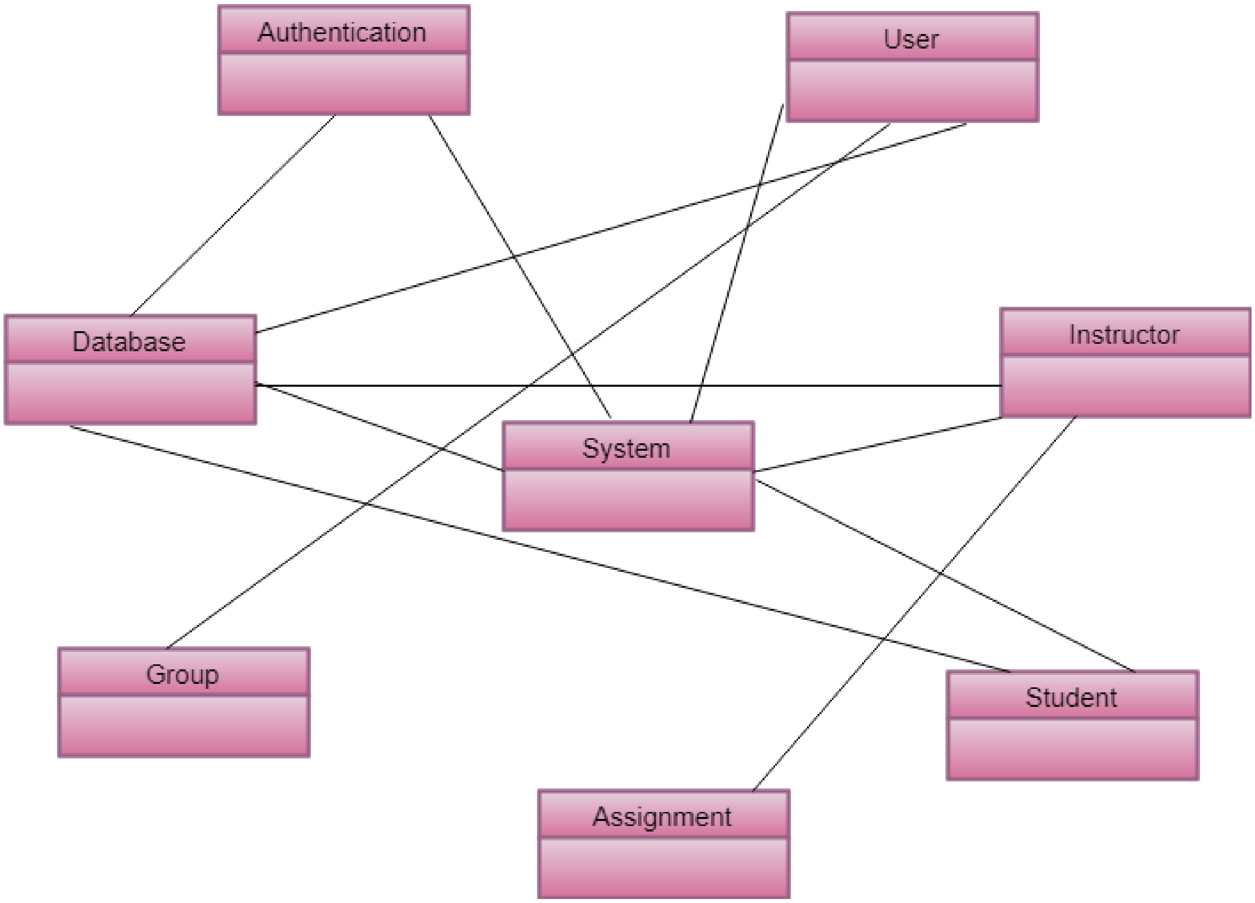


Figure 47: Class collaboration diagram of AMS

CHAPTER – 7: FLOW ORIENTED MODEL

This flow-oriented model described how data flow in a system.

7.1 INTRODUCTION

Although flow-oriented modeling is perceived as an outdated technique by some software engineers, it continues to be one of the most widely used requirement analysis in use today. Although the data flow diagram (DFD) and related diagrams and information are not a formal part of UML, they can be used to complement UML diagrams and provide additional insight into system requirements and flow.

7.2 DATA FLOW DIAGRAM (DFD)

The DFD takes an input process – output view of a system.

LEVEL - 0 DATA FLOW DIAGRAM

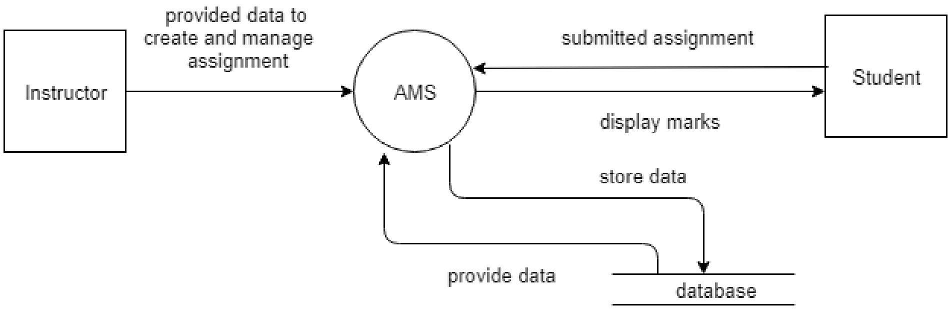


Figure 48: level – 0 DFD of AMS

LEVEL 1 DATA FLOW DIAGRAM

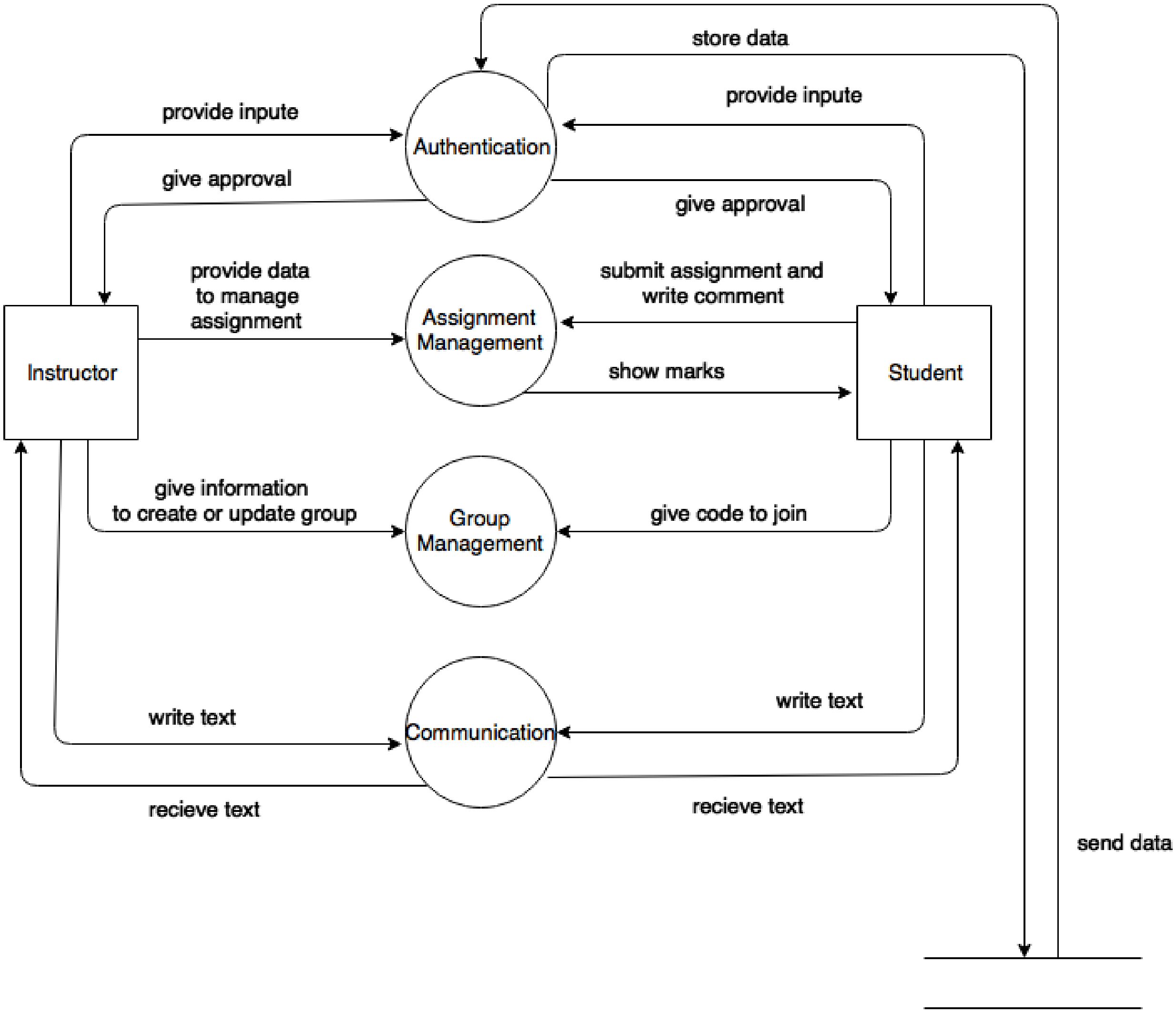


Figure 49: level 1 DFD of AMS

LEVEL 1.1.1 DATA FLOW DIAGRAM

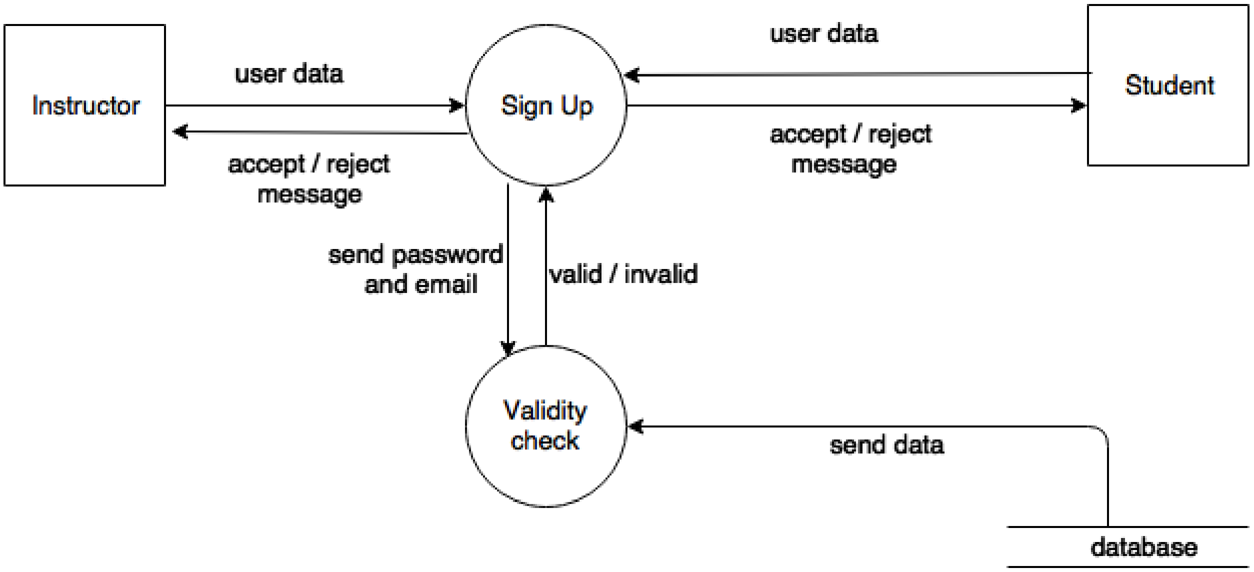


Figure 50: level 1.1.1 DFD of AMS

LEVEL 1.1.2 DATA FLOW DIAGRAM

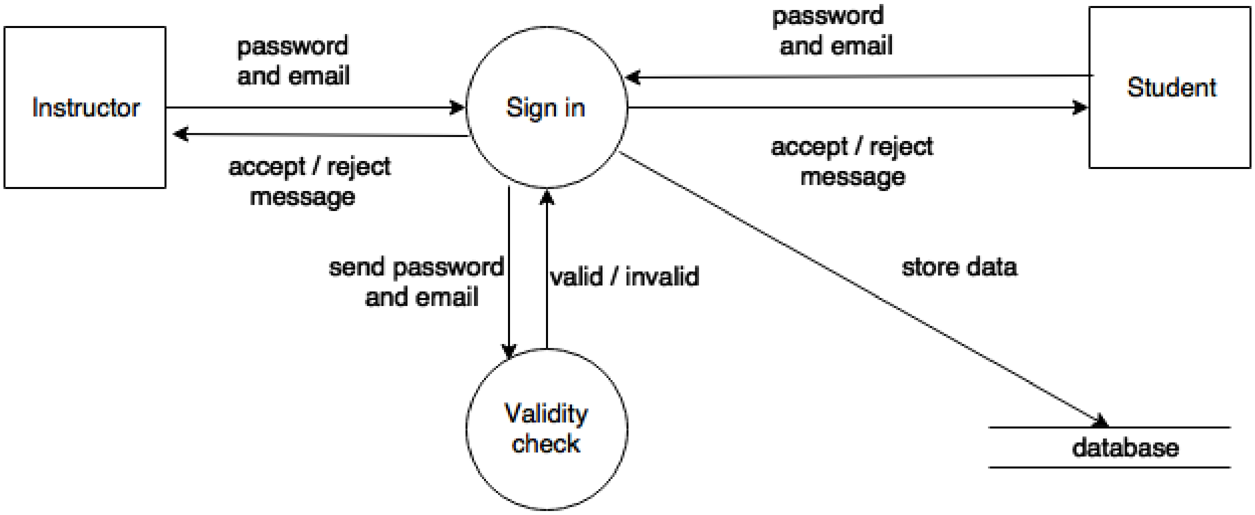


Figure 51: level 1.1.2 DFD of AMS

LEVEL 1.1.3 DATA FLOW DIAGRAM

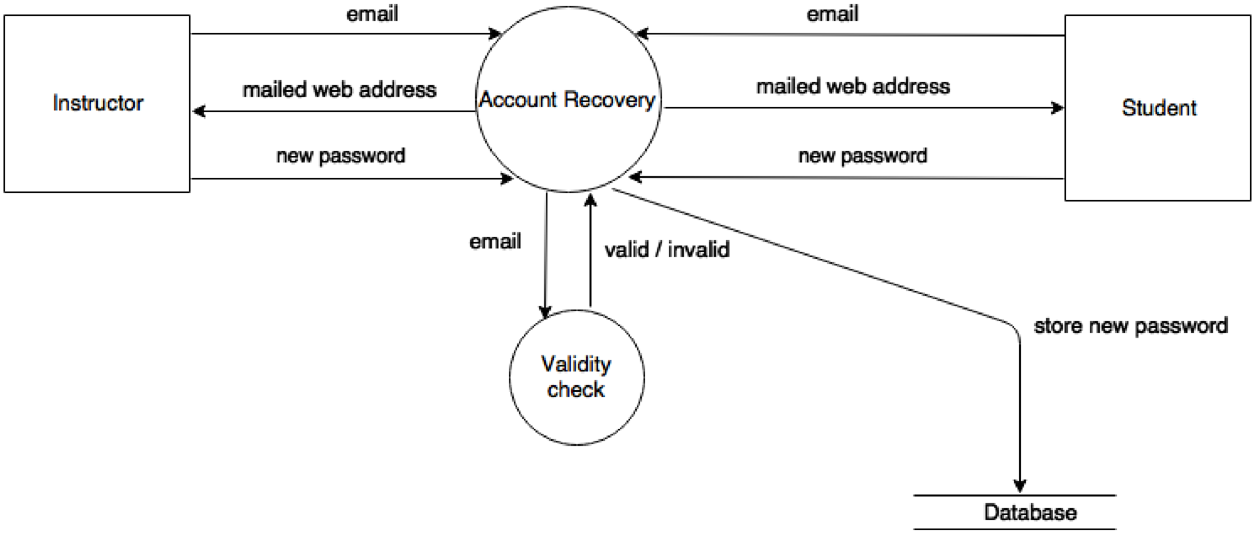


Figure 52: level 1.1.3 DFD of AMS

LEVEL 1.2.1 DATA FLOW DIAGRAM

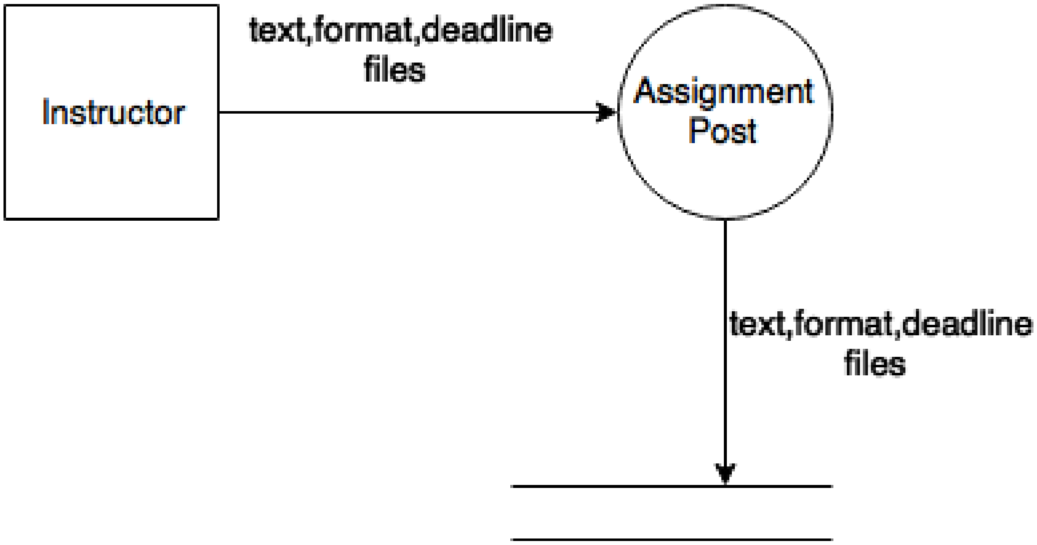


Figure 53: level 1.2.1 DFD of AMS

LEVEL 1.2.2 DATA FLOW DIAGRAM

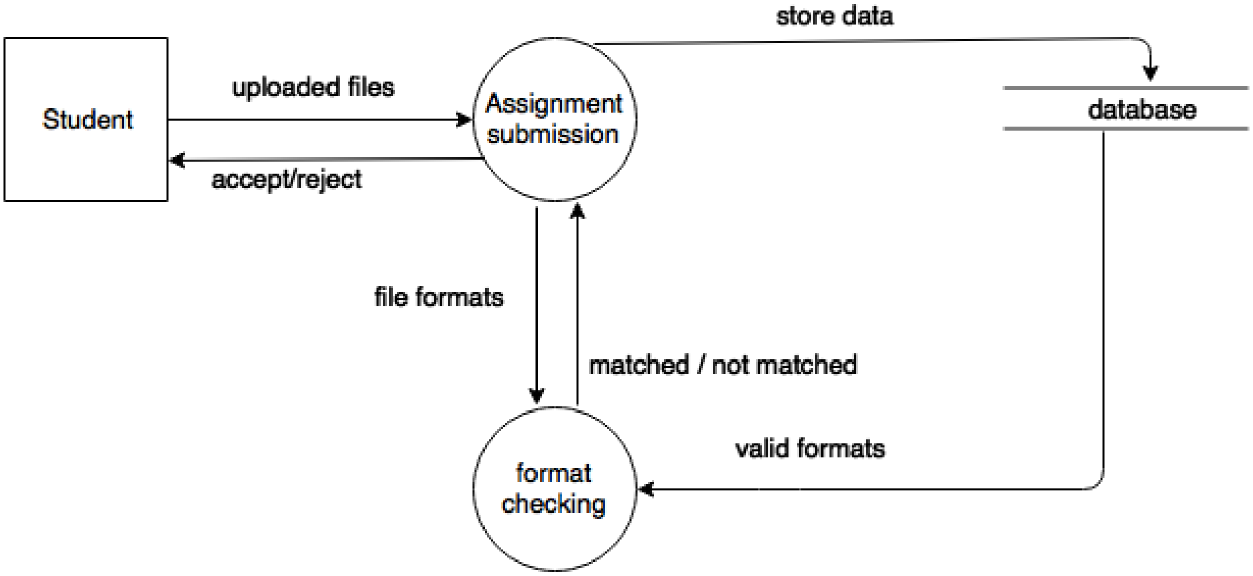


Figure 54: level 1.2.2 DFD of AMS

LEVEL 1.2.3 DATA FLOW DIAGRAM

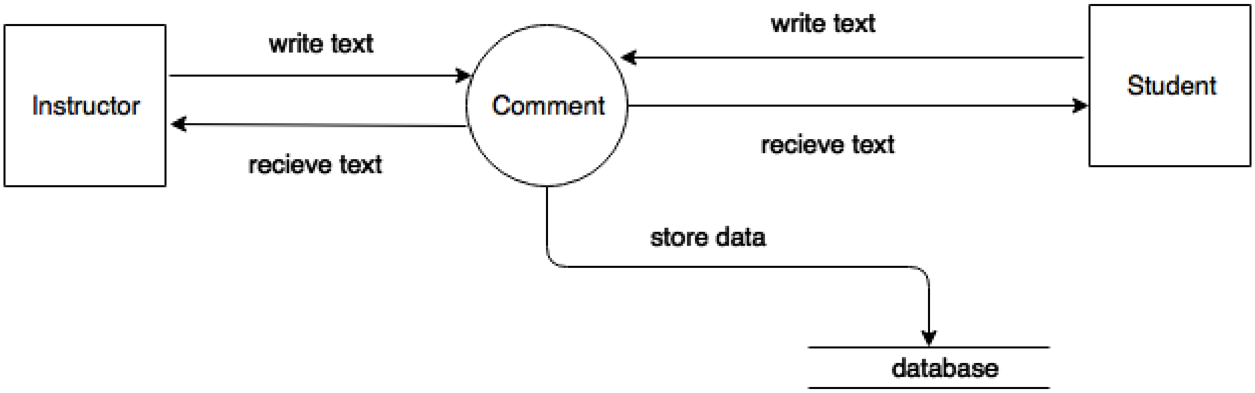


Figure 55:1.2.3 DFD of AMS

LEVEL 1.2.4 DATA FLOW DIAGRAM

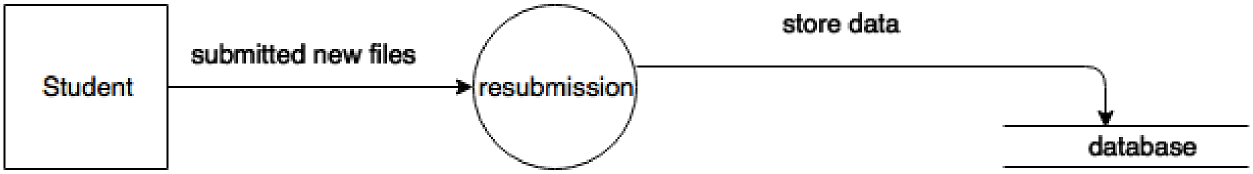


Figure 56: level 1.2.4 DFD of AMS

LEVEL 1.2.5 DATA FLOW DIAGRAM

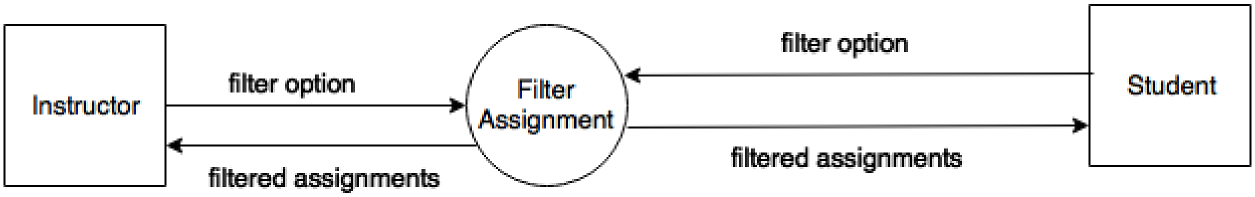


Figure 57:1.2.5 DFD of AMS

LEVEL 1.2.6 DATA FLOW DIAGRAM

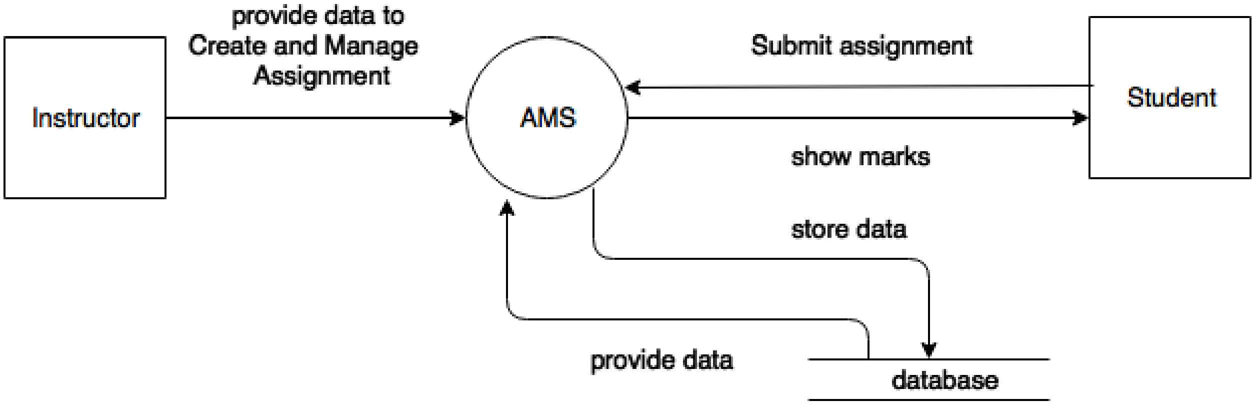


Figure 58: level 1.2.6 DFD of AMS

LEVEL 1.2.7 DATA FLOW DIAGRAM

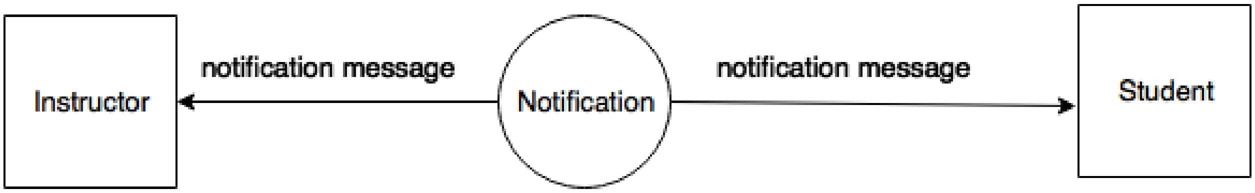


Figure 59: level 1.2.7 DFD of AMS

LEVEL 1.2.8 DATA FLOW DIAGRAM

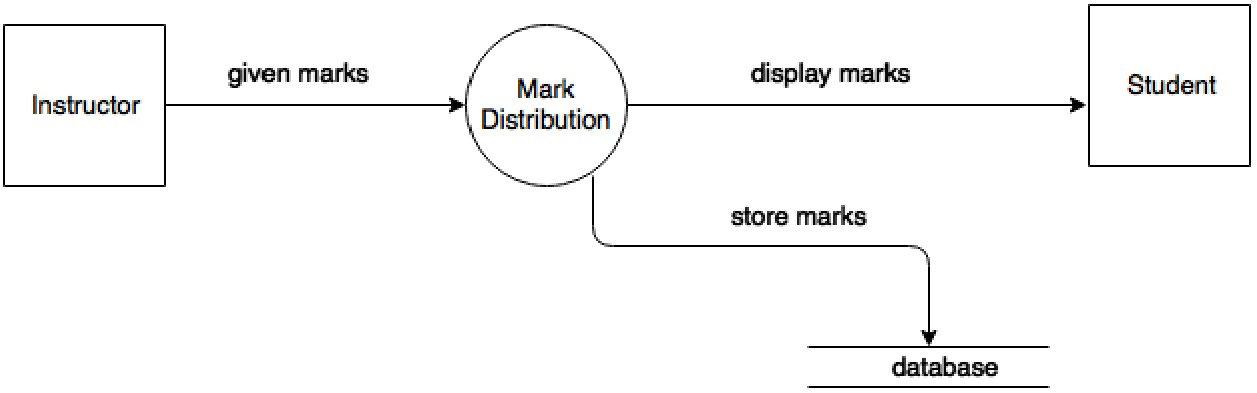


Figure 60: level 1.2.8 DFD of AMS

LEVEL 1.3.1 DATA FLOW DIAGRAM

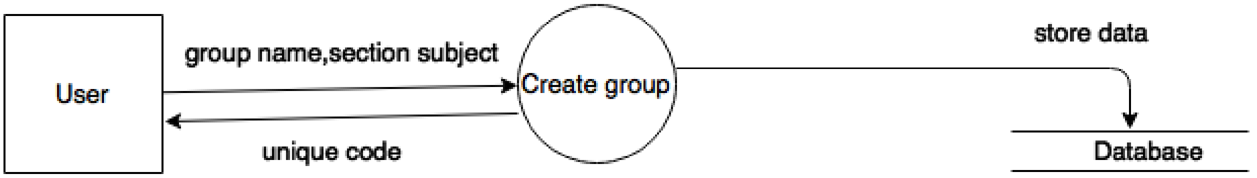


Figure 61: level 1.3.1 DFD of AMS

LEVEL 1.3.2 DATA FLOW DIAGRAM

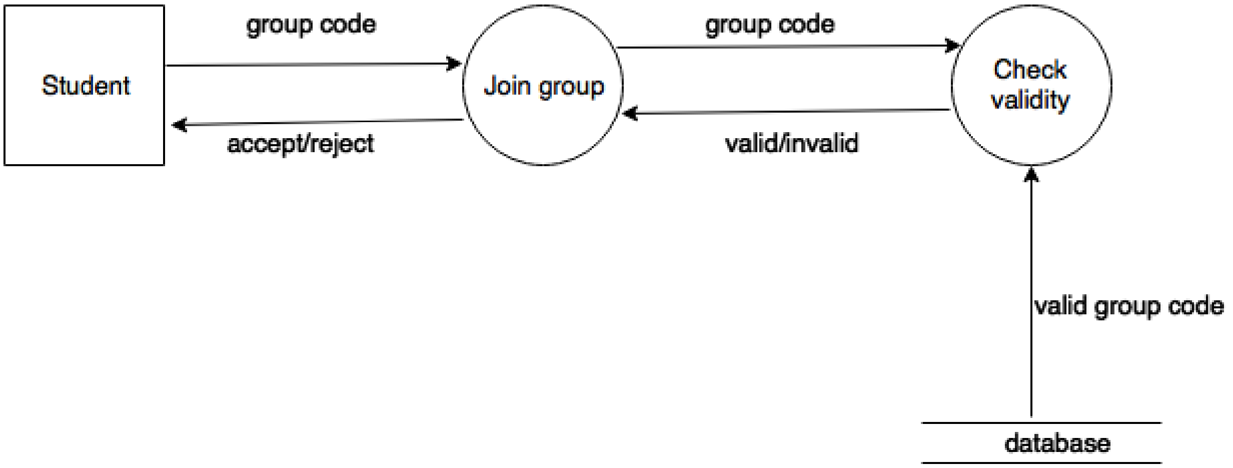


Figure 62: level 1.3.2 DFD of AMS

LEVEL 1.3.3 DATA FLOW DIAGRAM

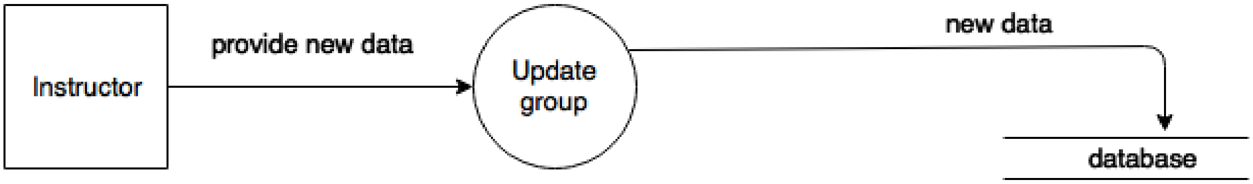


Figure 63: level 1.3.3 DFD of AMS

LEVEL 1.3.4 DATA FLOW DIAGRAM

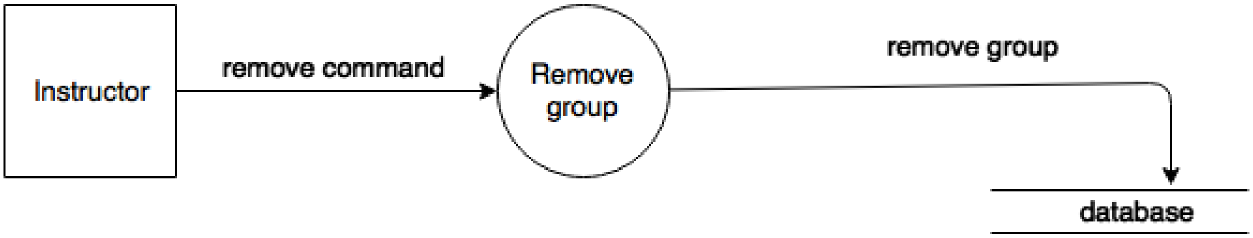


Figure 64: level 1.3.4 DFD of AMS

LEVEL 1.3.5 DATA FLOW DIAGRAM

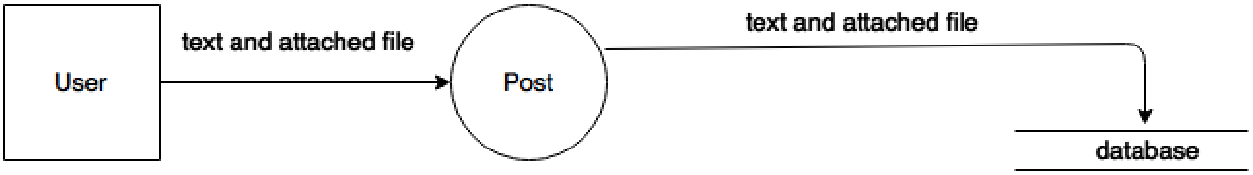


Figure 65: level 1.3.5 DFD of AMS

CHAPTER - 8: BEHAVIOURAL MODEL OF AMS

The behavioral model indicates how software responds to external event. This chapter describes the way AMS interacts.

8.1 STATE TRANSACTION

In the context of behavioral modeling to different characterization of state must be considered and these are:

- The state of each class as the system performs its function.
- The state of the system observed from the outside as the system perform its function.

8.1.1 EVENT IDENTIFICATION

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

Table 24: Event identification

Serial No	Event	Primary object	Collaborator	Method
1	Register to user	Authentication		Authentication: register()

2	Verifying and validating input to register	Authentication	System	Authentication:
				register()
				System:
				validateInput()
				verifyInput()

3	Generating user id	Authentication	System,
			Database
				Authentication:
				register()
				System:
				generatingUserId()
				Database:
				insert()

4	Log in user	Authentication	System,
			Database
				Authentication:
				signIn()
				System:
				verifyInput()
				Database:
				retrieve()

5	Recover account	Authentication	System,
			Database
				Authentication:
				accountrecovery()
				System:
				verifyInput()
				Database:
				retrieve()

6	Sending link	Authentication		Authentication: accountRecovery()

7	Set new password	authentication	System, Database	Authentication:
				accountRecovery()
				System:
				takeInput()
				validateInput()
				Database:
				insert()

8	Attempt to logout	User	System	User:
				signOut()
				System:
				checkRunningProcess()

9	Posts on assignment	Instructor	System	Instructor:
				createAssignment()
				System:
				takeInput()

10	Give format	Instructor		Instructor: createAssignment()

11	Send mail for assignment	Instructor	System	Instructor:
				createAssignment()
				System:
				sendMail()

12	Comment on assignment	User	System,
			Database
				User:
				comment()
				System:
				takeInput()
				Database:
				Insert()

13	Post on the group	User	System,
			Database
				User:
				post()
				System:
				takeInput()
				Database:
				Insert()

14	Search assignment	Instructor	Database	Instructor:
				searchAssignment()
				Database:
				show()

15	Submit assignment	Student	Database, System	Student: submitAssignment() System: checkFormatandDeadline() Database:

				insert()

16	Filter assignment	User	Database	User:
				filter()
				Database:
				retrieve()

17	Resubmit assignment	Student	Database, System	Student:
				resubmitAssignment()
				System:
				checkFormatandDeadline()
				Database:
				remove()
				insert()

18	Check plagiarism	Instructor	Database	Instructor:
				checkPlagiarism()
				Database:
				retrieve()

19	Distribute mark	Instructor	System, Database	Instructor: distributeMark() System: takeInput() Database:

				insert()

20	Create group	Instructor	System,
			Database
				Instructor:
				createGroup()
				System:
				takeInput()
				generateCode()
				Database:
				insert()

21	Join group	Student	System,
			Database
				Student:
				joinGroup()
				System:
				takeInput()
				verifyInput()
				Database:
				insert()
				retrieve()

22	Remove group	Instructor	Database	Instructor:
				removeGroup()
				Database:
				remove()

23	Update group information	Instructor	System, Database	Instructor:
				upadateGroup()
				System:
				takeInput()
				validateInput()
				Database:
				insert()

24	Retry to log in	Authentication		Authentication: Retry()

25	Sending message	User	System, Database	User:
				sendMessage()
				System:
				takeInput()
				Database:
				insert()

26	Receiving message	User	Database	User:
				receiveMessage()
				Database:
				retrieve()

8.1.2 STATE TRANSACTION

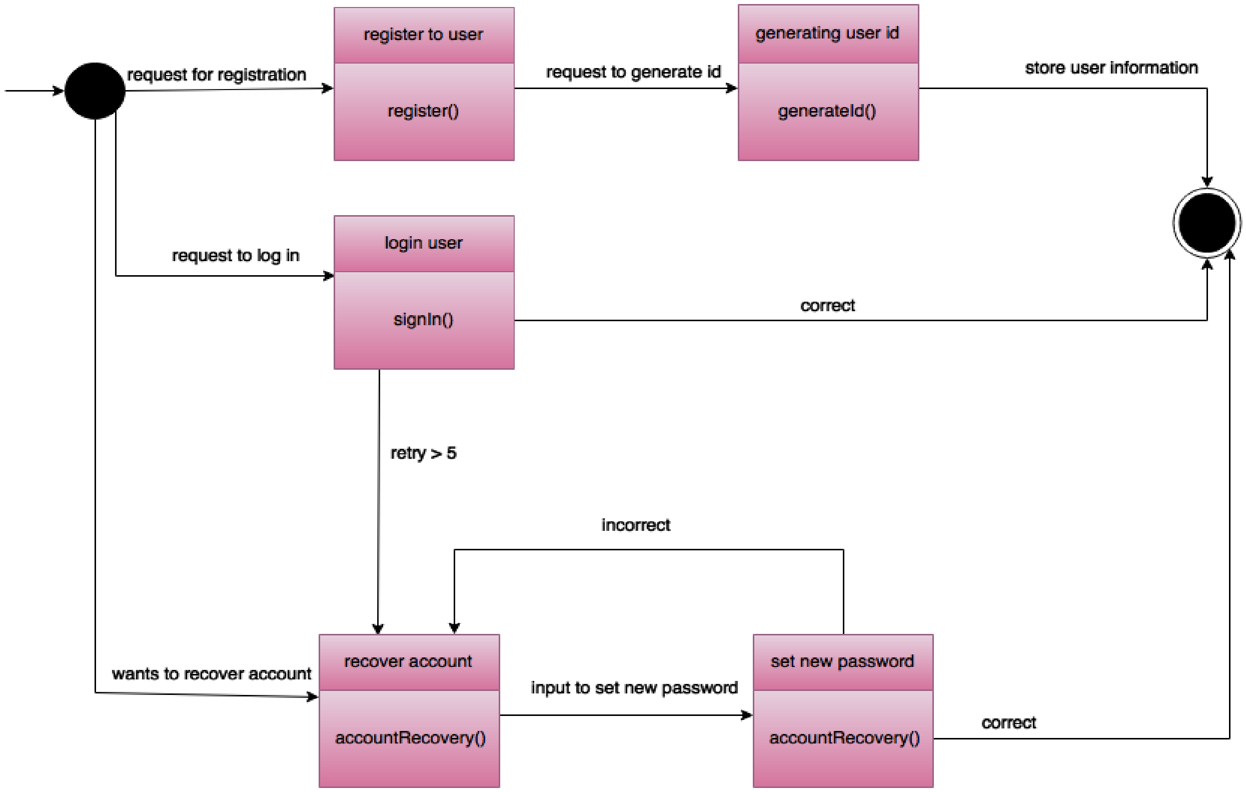


Figure 66: state transaction diagram – Authentication

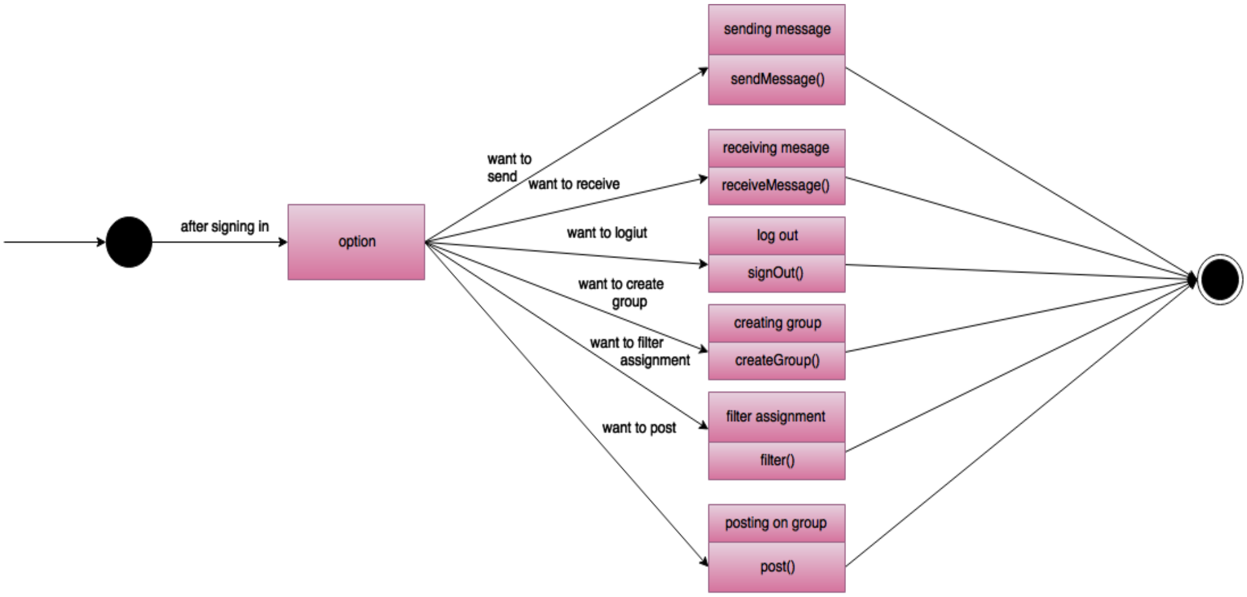


Figure 67: state transaction diagram – User

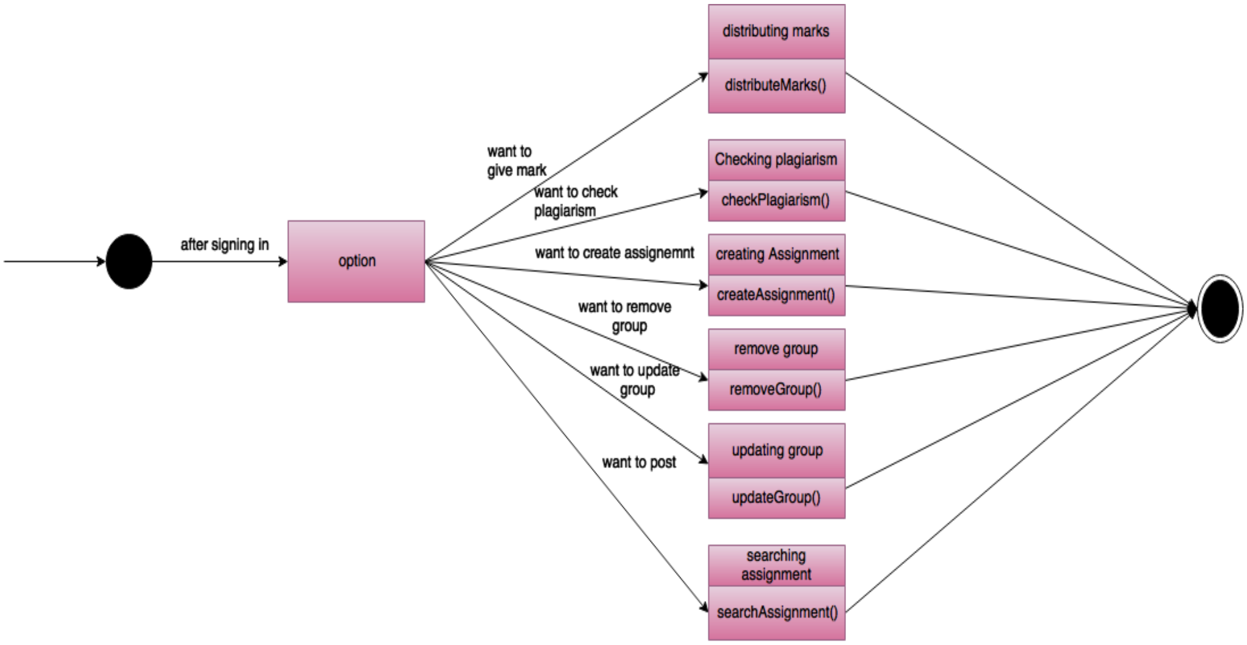


Figure 68: state transaction diagram – Instructor

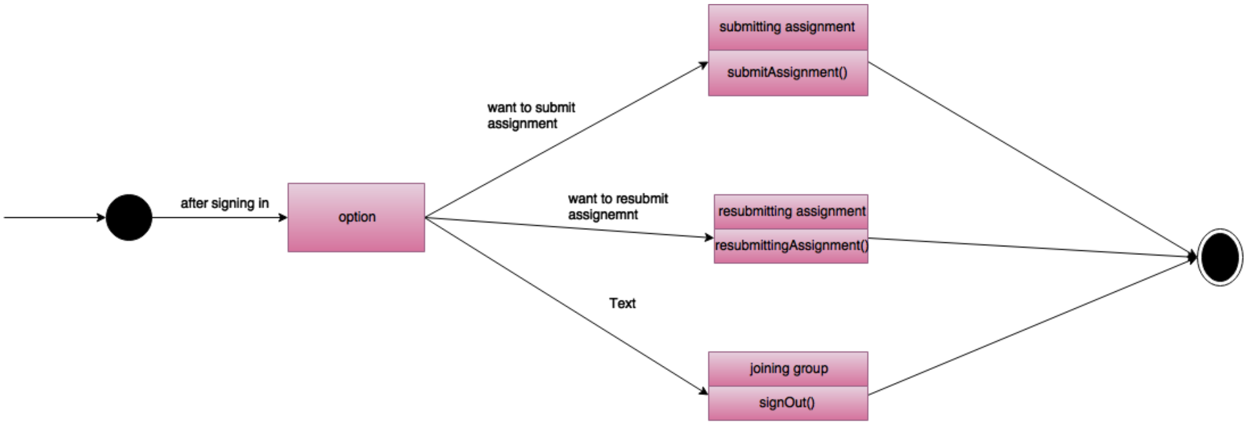


Figure 69: state transaction diagram – Student

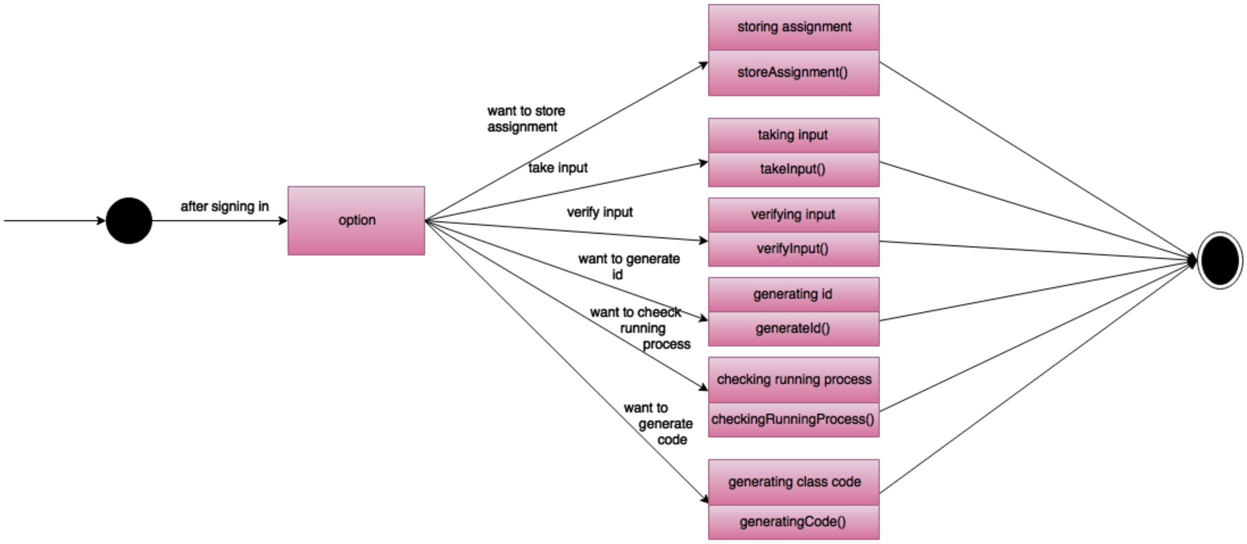


Figure 70: state transaction diagram- System

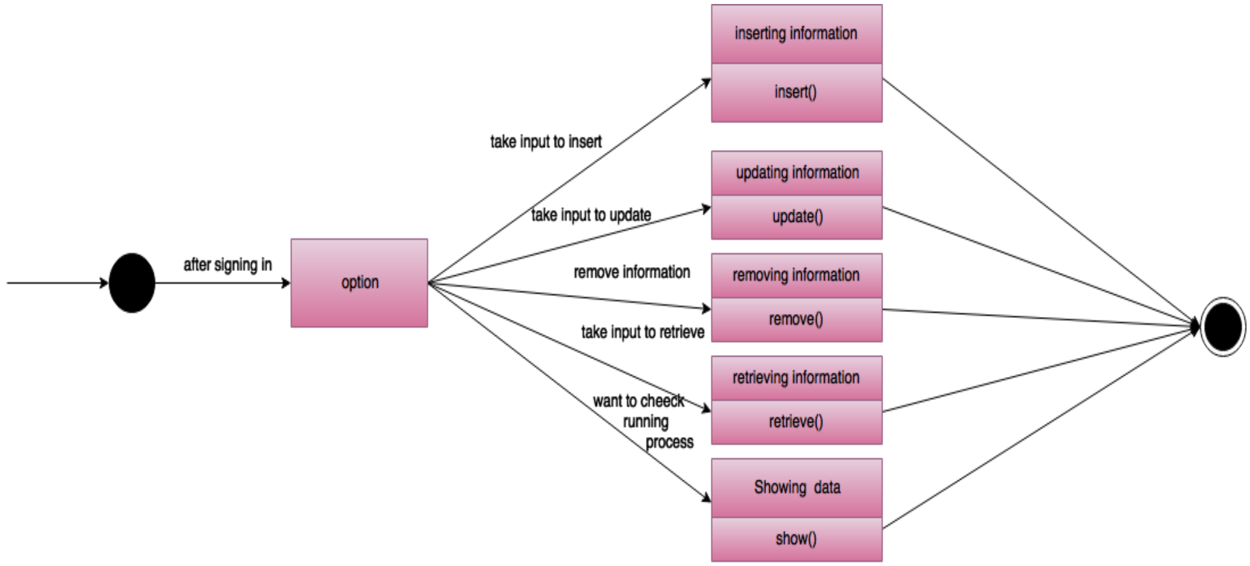


Figure 71: state transaction diagram- Database

8.1.3 SEQUENCE DIAGRAM

The second type of behavioral representation, called a sequence diagram of UML, indicates how events cause transaction from subject to subject.

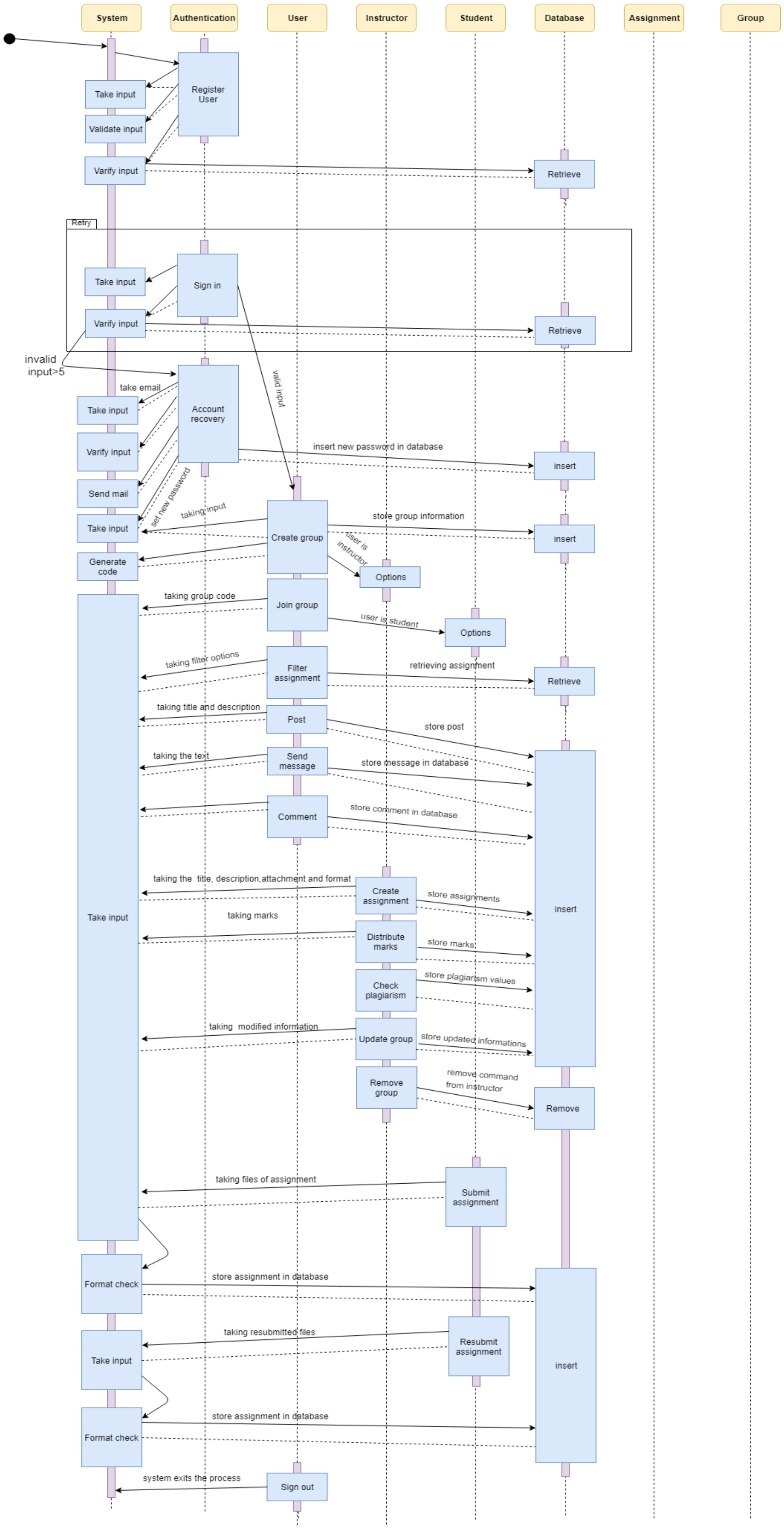


Figure 72: Sequence diagram

CHAPTER – 9: CONCLUSION

We are pleased to submit the final SRS report on Assignment Management System. From this, the readers will get a clear and easy view of the overall system. 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. Hopefully, this document can also help our junior BSSE batch students. 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.