**Educational Question-Answer Forum for Bangladesh**

Software Project Lab-3

Submitted by

Tulshi Chandra Das-811

Supervised by

Md. Saeed Siddik

Lecturer, IIT, DU



3/20/2018

#### question answer forum for bangladesh

##### SUBMITTED TO

SPL3 Committee

Institute of Information Technology

University of Dhaka

##### SUBMITTED BY

Tulshi Chandra Das

BSSE0811

##### LETTER OF TRANSMITTAL

20 March 2018

The Coordinators, SPL2

Institute of Information Technology

University of Dhaka.

**Subject: Submission of term report on “Question Answer Forum For Bangladesh”**

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 Supervisor

Md. Saeed Siddik

4rd Year, 8th Semester, 8th Batch Lecturer, IIT, DU

Institute of Information Technology

University of Dhaka

##### ACKNOWLEDGEMENT

We are highly indebted for getting such a tremendous opportunity to prepare the report on Question Answer Forum for Bangladesh. We would like to thank our Supervisor, Md. Saeed Siddik, 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 analyze on Question Answer Forum for Bangladesh and design the SRS of this system. The object of this study is to develop an SRS (Software Requirements Specification and Analysis) of this project

## Table of Contents

[CHAPTER-01: INTRODUCTION](#_Toc509285770) 1

[1.1 PURPOSE 1](#_Toc509285771)

[1.2 INTENDED AUDIENCE 1](#_Toc509285772)

[1.3 CONCLUSION 2](#_Toc509285773)

[CHAPTER -02: INCEPTION OF AMS 2](#_Toc509285774)

[2.1 INTRODUCTION 2](#_Toc509285775)

[2.1.1 LIST OF STAKEHOLDERS 2](#_Toc509285776)

[2.1.2 MULTIPLE VIEWPOINTS 3](#_Toc509285777)

[2.1.3 WORKING TOWARDS COLLABORATION 4](#_Toc509285778)

[2.1.4 REQUIREMENTS QUESTIONNAIRE 6](#_Toc509285779)

[2.2 CONCLUSION 6](#_Toc509285780)

[CHAPTER-03: ELICITATION OF AMS 7](#_Toc509285781)

[3.1 INTRODUCTION 7](#_Toc509285782)

[3.2 ELICITING REQUIREMENTS 7](#_Toc509285783)

[3.2.1 COLLABORATIVE REQUIREMENTS GATHERING 7](#_Toc509285784)

[3.2.2 PROBLEM IN THE SCOPE 8](#_Toc509285785)

[3.2.3 QUALITY FUNCTION DEPLOYMENT 8](#_Toc509285786)

[3.2.4 USAGE SCENARIO 9](#_Toc509285787)

[3.2.5 ELICITATION WORK PRODUCT 12](#_Toc509285788)

[CHAPTER-04: SCENARIO BASED MODELING OF AMS 13](#_Toc509285789)

[4.1 INTRODUCTION 13](#_Toc509285790)

[4.2 DEFINITION OF USE CASE 13](#_Toc509285791)

[4.3 USE CASE DIAGRAM 14](#_Toc509285792)

[4.3.1 LEVEL- 0 USE CASE DIAGRAM-AMS 14](#_Toc509285793)

[4.3.2 LEVEL -1 USECASE DIAGRAM-SUBSYSTEM 15](#_Toc509285794)

[4.3.3 LEVEL- 1.1 USE CASE DIAGRAM- AUTHENTICATION 16](#_Toc509285795)

[4.3.4 LEVEL -1.2 USE CASE DIAGRAM- ASSIGNMENT MANAGEMENT 18](#_Toc509285796)

[4.3.5 LEVEL -1.2.1 USE CASE DIAGRAM- POST ASSIGNMENT 19](#_Toc509285797)

[4.3.6 LEVEL -1.3 USE DIAGRAM- GROUP MANAGEMENT 24](#_Toc509285798)

[DESCRIPTION OF LEVEL -1.3 USE CASE DIAGRAM 24](#_Toc509285799)

[4.4 Activity Diagrams 27](#_Toc509285800)

[4.4 swim lane Diagrams 46](#_Toc509285801)

[CHAPTER 5: DATA BASED MODELING OF ASM 65](#_Toc509285802)

[5.1 INTRODUCTION 65](#_Toc509285803)

[5.2 DATA OBJECTS 65](#_Toc509285804)

[5.2.1 NOUN IDENTIFICATION 65](#_Toc509285805)

[5.2.2 POTENTIAL DATA OBJECTS 69](#_Toc509285806)

[5.2.3 ANALYSIS FOR FINAL DATA OBJECT 69](#_Toc509285807)

[5.2.4 FINAL DATA OBJECT 69](#_Toc509285808)

[5.3 DATA OBJECT RELATIONS 71](#_Toc509285809)

[5.5 SCHEMA DIAGRAM 73](#_Toc509285810)

[CHAPTER – 6 CLASS BASED MODELING 78](#_Toc509285811)

[6.1 INTRODUCTION 78](#_Toc509285812)

[6.2 IDENTIFYING ANALYSIS CLASS 78](#_Toc509285813)

[6.2.1 GENERAL CLASSIFICATION 78](#_Toc509285814)

[6.2.2 Selection Criteria 80](#_Toc509285815)

[6.2.3 ASSOCIATE NOUN WITH VERB 82](#_Toc509285816)

[6.2.4 ATTRIBUTE SELECTION 83](#_Toc509285817)

[6.2.5 METHOD IDENTIFICATION 85](#_Toc509285818)

[6.2.6 CLASS CARDS 87](#_Toc509285819)

[6.2.7 cLASS COLLABORATION DIAGRAM 93](#_Toc509285820)

[CHAPTER – 7: FLOW ORIENTED MODEL 94](#_Toc509285821)

[7.1 Introduction 94](#_Toc509285822)

[7.2 Data flow diagram (dfd) 94](#_Toc509285823)

[Level - 0 Data flow diagram 94](#_Toc509285824)

[level 1 Data flow diagram 95](#_Toc509285825)

[level 1.1.1 data flow diagram 96](#_Toc509285826)

[level 1.1.2 Data flow diagram 96](#_Toc509285827)

[level 1.1.3 data flow diagram 97](#_Toc509285828)

[level 1.2.1 Data flow diagram 97](#_Toc509285829)

[level 1.2.2 Data flow Diagram 98](#_Toc509285830)

[level 1.2.3 data flow diagram 98](#_Toc509285831)

[level 1.2.4 data flow diagram 98](#_Toc509285832)

[level 1.2.5 data flow diagram 99](#_Toc509285833)

[level 1.2.6 data flow diagram 99](#_Toc509285834)

[level 1.2.7 data flow diagram 99](#_Toc509285835)

[level 1.2.8 Data flow diagram 100](#_Toc509285836)

[level 1.3.1 DATA flow diagram 100](#_Toc509285837)

[level 1.3.2 data flow Diagram 100](#_Toc509285838)

[level 1.3.3 data flow Diagram 101](#_Toc509285839)

[level 1.3.4 data flow diagram 101](#_Toc509285840)

[level 1.3.5 data flow diagram 101](#_Toc509285841)

[CHAPTER - 8: BEHAVIOURAL MODEL OF AMS 102](#_Toc509285842)

[8.1 STATE TRANSACTION 102](#_Toc509285843)

[8.1.1 EVENT IDENTIFICATION 102](#_Toc509285844)

[8.1.2 State Transaction 109](#_Toc509285845)

[8.1.3 sequence diagram 112](#_Toc509285846)

[CHAPTER – 9: CONCLUSION 114](#_Toc509285847)

##### Content of figures

[Figure 1: level 0 use case diagram- AMS 14](#_Toc509283024)

[Figure-2: Figure 2 level 1 use case diagram - Subsystem 15](#_Toc509283025)

[Figure-3: Figure 3level 1.1 use case diagram – Authentication 16](#_Toc509283026)

[Figure 4: level 1.2 use case diagram- Assignment management 18](#_Toc509283027)

[Figure 5: level 1.2.1 use case diagram- Post assignment 19](#_Toc509283028)

[.Figure 6: level 1.3 use case diagram- Group management 24](#_Toc509283029)

[Figure 7: Level 1.1 Activity diagram – Authentication. 27](#_Toc509283030)

[Figure 8: Level 1.1.1 Activity diagram – Sign up 28](#_Toc509283031)

[Figure 9: Level 1.1.2 Activity diagram – Sign in. 29](#_Toc509283032)

[Figure 10: Level 1.1.3 Activity diagram – Account recovery 30](#_Toc509283033)

[Figure 11: Level 1.2 Activity diagram – Assignment management 31](#_Toc509283034)

[Figure 12: Level 1.2.1 Activity diagram – Assignment post 32](#_Toc509283035)

[Figure 13: Level 1.2.2 Activity diagram – Assignment submission 33](#_Toc509283036)

[Figure 14: Level 1.2.3 Activity diagram – Comment. 34](#_Toc509283037)

[Figure 15: Level 1.2.4 Activity diagram – Assignment resubmission. 35](#_Toc509283038)

[Figure 16: Level 1.2.5 Activity diagram – Filter assignment 36](#_Toc509283039)

[Figure 17: Level 1.2.6 Activity diagram – Plagiarism check 37](#_Toc509283040)

[Figure 18: Level 1.2.7 Activity diagram –Notification 38](#_Toc509283041)

[Figure 19: Level 1.2.8 Activity diagram – Mark distribution. 39](#_Toc509283042)

[Figure 20: Level 1.3 Activity diagram – Group management. 40](#_Toc509283043)

[Figure 21: Level 1.3.1 Activity diagram – Create group 41](#_Toc509283044)

[Figure 22: Level 1.3.2 Activity diagram – Join group 42](#_Toc509283045)

[Figure 23: Level 1.3.3 Activity diagram – Update group. 43](#_Toc509283046)

[Figure 24: Level 1.3.4 Activity diagram – Remove group. 44](#_Toc509283047)

[Figure 25: Level 1.3.5 Activity diagram – Post. 45](#_Toc509283048)

[Figure 26: Level 1.1 Swim lane diagram – Authentication. 46](#_Toc509283049)

[Figure 27: Level 1.1.1 Swim lane diagram – Sign up. 47](#_Toc509283050)

[Figure 28:Figure 28: Level 1.1.2 Swim lane diagram – Sign in. 48](#_Toc509283051)

[Figure 29: Level 1.1.3 Swim lane diagram – Account recovery 49](#_Toc509283052)

[Figure 30: Level 1.2 Swim lane diagram – Assignment management 50](#_Toc509283053)

[Figure 31: Level 1.2.1 Swim lane diagram – Assignment post. 51](#_Toc509283054)

[Figure 32: Level 1.2.2 Swim lane diagram – Assignment submission. 52](#_Toc509283055)

[Figure 33: Level 1.2.3 Swim lane diagram – Comment. 53](#_Toc509283056)

[Figure 34: Level 1.2.4 Swim lane diagram – Assignment resubmission. 54](#_Toc509283057)

[Figure 35: Level 1.2.5 Swim lane diagram – Filter assignment. 55](#_Toc509283058)

[Figure 36: Level 1.2.6 Swim lane diagram – Sign in. 56](#_Toc509283059)

[Figure 37: Level 1.2.7 Swim lane diagram – Notification. 57](#_Toc509283060)

[Figure 38: Level 1.2.8 Swim lane diagram – Mark distribution 58](#_Toc509283061)

[Figure 39: Level 1.3 Swim lane diagram – Group management 59](#_Toc509283062)

[Figure 40: Level 1.3.1 Swim lane diagram – Create group 60](#_Toc509283063)

[Figure 41: Level 1.3.2 Swim lane diagram – Join group 61](#_Toc509283064)

[Figure 42: Level 1.3.3 Swim lane diagram – Update group 62](#_Toc509283065)

[Figure 43: Level 1.3.4 Swim lane diagram – Remove group 63](#_Toc509283066)

[Figure 44: Level 1.3.5 Swim lane diagram – Post 64](#_Toc509283067)

[Figure 45: Relationships between data objects 71](#_Toc509283068)

[Figure 46: Entity Relationship of Assignment Management System 72](#_Toc509283069)

[Figure 47: Class collaboration diagram of AMS 93](#_Toc509283070)

[Figure 48: level – 0 DFD of AMS 94](#_Toc509283071)

[Figure 49: level 1 DFD of AMS 95](#_Toc509283072)

[Figure 50: level 1.1.1 DFD of AMS 96](#_Toc509283073)

[Figure 51: level 1.1.2 DFD of AMS 96](#_Toc509283074)

[Figure 52: level 1.1.3 DFD of AMS 97](#_Toc509283075)

[Figure 53: level 1.2.1 DFD of AMS 97](#_Toc509283076)

[Figure 54: level 1.2.2 DFD of AMS 98](#_Toc509283077)

[Figure 55:1.2.3 DFD of AMS 98](#_Toc509283078)

[Figure 56: level 1.2.4 DFD of AMS 99](#_Toc509283079)

[Figure 57:1.2.5 DFD of AMS 99](#_Toc509283080)

[Figure 58: level 1.2.6 DFD of AMS 99](#_Toc509283081)

[Figure 59: level 1.2.7 DFD of AMS 99](#_Toc509283082)

[Figure 60: level 1.2.8 DFD of AMS 100](#_Toc509283083)

[Figure 61: level 1.3.1 DFD of AMS 100](#_Toc509283084)

[Figure 62: level 1.3.2 DFD of AMS 100](#_Toc509283085)

[Figure 63: level 1.3.3 DFD of AMS 101](#_Toc509283086)

[Figure 64: level 1.3.4 DFD of AMS 101](#_Toc509283087)

[Figure 65: level 1.3.5 DFD of AMS 101](#_Toc509283088)

[Figure 66: state transaction diagram – Authentication 109](#_Toc509283089)

[Figure 67: state transaction diagram – User 110](#_Toc509283090)

[Figure 68: state transaction diagram – Instructor 110](#_Toc509283091)

[Figure 69: state transaction diagram – Student 111](#_Toc509283092)

[Figure 70: state transaction diagram- System 111](#_Toc509283093)

[Figure 71: state transaction diagram- Database 112](#_Toc509283094)

[Figure 72: Sequence diagram 113](#_Toc509283095)

##### List of Tables

[Table 1: Noun Identification for Data Modelling 65](#_Toc509284816)

[Table 2: schema table of User data object 73](#_Toc509284817)

[Table 3: schema table of Instructor data object 73](#_Toc509284818)

[Table 4: schema table of Student data object 74](#_Toc509284819)

[Table 5: schema table of Group data object 74](#_Toc509284820)

[Table 6: schema table of Assignment data object 75](#_Toc509284821)

[Table 7: schema table of Message data object 75](#_Toc509284822)

[Table 8: schema table of comment data object 76](#_Toc509284823)

[Table 9: schema table of Assignment submission data object 76](#_Toc509284824)

[Table 10: schema table of Marks data object 76](#_Toc509284825)

[Table 11: Noun with general classification 79](#_Toc509284826)

[Table 12: selection criteria of nouns 81](#_Toc509284827)

[Table 13: Associate noun and verb identification 82](#_Toc509284828)

[Table 14: class and attributes selection 83](#_Toc509284829)

[Table 15: Method identification 85](#_Toc509284830)

[Table 16: User 87](#_Toc509284831)

[Table 17: Instructor 87](#_Toc509284832)

[Table 18: Student 88](#_Toc509284833)

[Table 19: Authentication 89](#_Toc509284834)

[Table 20: System 90](#_Toc509284835)

[Table 21: Database 91](#_Toc509284836)

[Table 22: Assignment 91](#_Toc509284837)

[Table 23: Group 92](#_Toc509284838)

[Table 24: Event identification 102](#_Toc509284839)

# CHAPTER-01: INTRODUCTION

This chapter is a part of our software requirement specification for the project “Question Answer Forum for Bangladesh”. In this chapter, we focus on the intended audience for this project.

## 1.1 PURPOSE

This document briefly describes the Software Requirement Analysis of Question Answer Forum for Bangladesh. 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.

## 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.

## 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

## 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 some universities of Bangladesh and asked them the following questions:

We identified following stakeholders for our assignment system:

* Inquirer
* Responder

**Inquirer:** Inquirer is a person who ask question to this forum.

**Responder:** Responder is a person who answer to a question

### 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:

#### user VIEWPOINTS

* Creating Question
* Answer to question
* Voting/Downvoting to question/answer
* Comment System
* Profile view
* Search user
* Rating system

### 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

* Creating Question
* Answering to question
* Commenting to question
* Rating system

#### CONFLICTING REQUIREMENTS

* Profile view

#### FINAL REQUIREMENTS

* Creating Question
* Answer to question
* Voting/Downvoting to question/answer
* Comment System
* Profile view
* Search user
* Message to teacher

### 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

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 Inquirer and Responder. 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.

### 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 the Question-Answer Forum.

**What will not be done**:

* In the forum user cannot upload any document file (docx, pdf etc.)
* Plagiarism checking will be performed between the submitted assignments only.

**What will be done:**

* In this forum image can be uploaded
* User can give answer of maximum 5000 character.

### 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-

* Answering to question
* Creating a question
* Commenting to answer or a question
* Upvote/Downvote to question/answer

#### 3.2.3.2 EXPECTED REQUIREMENTS

* Question/Answer modification deletion
* User-friendly
* Efficient question recommendation system
* Efficient question similarity check

#### 3.2.3.3 EXCITING REQUIREMENTS

* Badge popularity system in user rating

### 3.2.4 USAGE SCENARIO

Question-Answer Forum for Bangladesh is an automated system for the following purposes:

* Question
* Answer
* Comment
* Reputation (Rating)

#### 3.2.4.1 Question management

##### CREATING QUESTION

User will create a question going throw a set of defined steps. First user will provide the category of his/her question. Then user will give the title of the question. Then he/she will provide the tags related the topics of question. In the next step the system will show some similar question related to his asking question. If user not satisfied user will go to next step and give description and can upload image. At last user will submit question.

##### RECCOMMENDING QUESTION TO USERS

In this forum different user from different will create question on different study topics. So, all user will not be recommended to all types of question. The system will use user profile data and activity to recommend questions.

#### 3.2.4.2 Answer management

Any user can answer to any question. User can upload image with his/her answer. User will be able to write maximum 5000-character description to an answer.

#### 3.2.4.3 Comment management

Any user can comment to an answer/question. User will have to gain minimum 50 reputation to comment. Any user can upvote/downvote to a comment.

#### 3.2.4.4 user management

Managing user and proofing their activity is one of the major parts of this project. This system will store the activities like answers, question, popularity, following tags, answer count to corresponding to tags, question count corresponding to tags. Popularity is the exciting thing for users to be motivated active in question-answer discussion. User will gain more popularity based on upvote to his/her question or answer or comment. He/she will gain different badge (silver, gold, platinum) at different level of popularity. As a result, user will appear in the search of users more in front position.

### 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

This chapter describes the Scenario-Based Model for the project.

## 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.

## 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.

### 4.3.1LEVEL- 0 USE CASE DIAGRAM-EQAF(Educational question-answer forum)

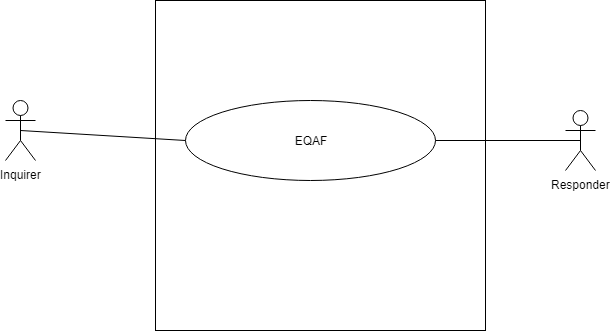


Figure : level 0 use case diagram- EAAF

Name: Educational Question-Answer Forum

Primary actor: Inquirer, Responder

Secondary actor: N/A

##### 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 “Educational Question-Answer Forum”:

* Inquirer
* Responder

### 4.3.2 LEVEL -1 USECASE DIAGRAM-SUBSYSTEM of eqaf

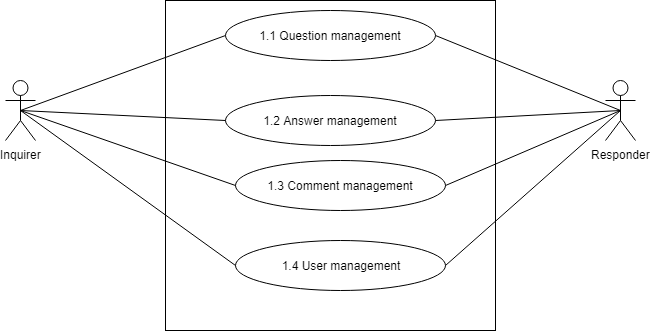


Figure-2: Figure level 1 use case diagram - Subsystem

Name: Subsystem of EQAF

Primary actor: Inquirer, Responder

Secondary actor: N/A

There are 4 subsystems in the Educational Question Answer Forum. They are-

* Question Management
* Answer Management
* Comment Management
* User Management

### 4.3.3 LEVEL- 1.1 USE CASE DIAGRAM- question management

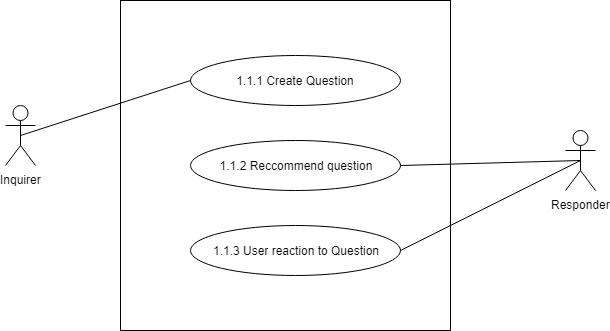


Figure-3: Figure level 1.1 use case diagram – Question management

Name: Question management

Primary actor: Inquirer, Responder

Secondary actor: N/A

#### DESCRIPTION OF LEVEL- 1.1 USE CASE DIAGRAM-

Creating question refers the process of creating question. User will create a question following some defined steps. After creating question system will recommend the asked question to appropriate user. Users can react by giving upvote or downvote to the question. If downvote count cross a specific limit question will be blocked. The subsystems are:

* Create question
* Recommend question
* User reaction to question

#### 1.1.1 CREATE QUESTION

* Primary actor: Inquirer
* Secondary actor: N/A

##### INQUIRER ACTION/REPLY

* Action: Inquirer will ask the question.
* Reply: System will recommend the question to appropriate users.

#### 1.1.2 RECOMMEND QUESTION

* Primary actor: N/A
* Secondary actor: Responder

##### RESPONDER ACTION/REPLY

* Action: Responder will view the question.
* Reply: Responder can respond to the question

#### 1.1.3 USER REACTION TO QUESTION

* Primary actor: Users
* Secondary actor: N/A

##### RESPONDER ACTION/REPLY

* Action: Responder will upvote/downvote to question
* Reply: Inquirer will be notified of it

### 4.3.4 LEVEL -1.2 USE CASE DIAGRAM- ANSWER MANAGEMAGEMEnt

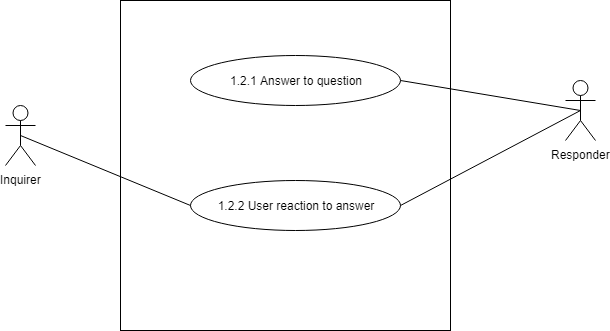


Figure : level 1.2 use case diagram- Answer management

Name: Answer management of EQAF

Primary actor: Responder, Inquirer

Secondary actor: N/A

#### DESCRIPTION OF LEVEL- 1.2 USE CASE DIAGRAM-

There are 2 subsystems in Assignment subsystem. These are-

* Answer to question
* User reaction to answer

#### 1.2.1 ANSWER TO QUESTION

* Primary actor: Responder
* Secondary actor: N/A

##### RESPONDER ACTION/REPLY

* Action: Responder answer to question.
* Reply: Answer will be view by other users.

#### 1.2.2 USER REACTION TO ANSWER

* Primary actor: Responder, Inquirer
* Secondary actor: N/A

##### INQUIRER ACTION/REPLY

* Action: Inquirer can vote/downvote and identify as accepted answer
* Reply: Reputation of responder will increase/decrease.

##### RESPONDER REACTION

* Action: Other Responders can vote/downvote to answer
* Reply: Reputation of responder will increase/decrease.

### 4.3.5 LEVEL -1.3 USE CASE DIAGRAM- COMMENT MANAGEMENT

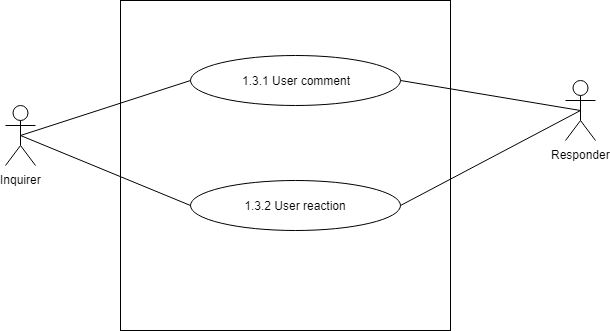


Figure : level 1.3 use case diagram- Comment management

Name: Comment management of EQAF

Primary actor: Inquirer, Responder

Secondary actor: N/A

#### DESCRIPTION OF LEVEL -1.3 USE CASE DIAGRAM

There are 2 subsystems in comment management subsystem. These are-

* User comment
* User reaction

#### 1.3.1 user comment

* Primary actor: Inquirer, Responder
* Secondary actor: N/A

##### INQUIRER/RESPONDER ACTION/REPLY

* Action: A user can comment to others answer/question
* Reply: Comment will be seen by other

#### 1.3.2 user reaction

* Primary actor: Inquirer, Responder
* Secondary actor: N/A

##### INQUIRER/RESPONDER ACTION/REPLY

* Action: User can mark the comment as useful
* Reply: Reputation of commenter will change
* Action: User can mark the comment as problematic
* Reply: Reputation of commenter will change and comment will be block after crossing a limit of flag

### 4.3.6 LEVEL -1.4 USE CASE DIAGRAM -USER MANAGEMENT

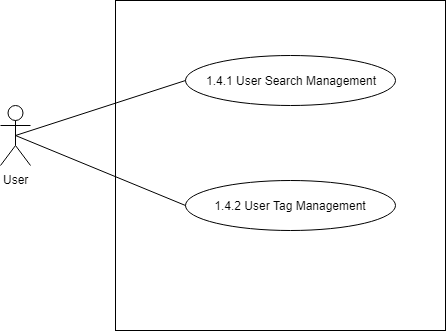


Figure : level 1.4 use case diagram- user management

Name: User Management of EQAF

Primary actor: User

Secondary actor: N/A

#### DESCRIPTION OF LEVEL- 1.4 USE CASE DIAGRAM

There are 2 subsystems in comment management subsystem. These are-

* User search management
* User tag management

#### 1.4.1 USER SEARCH MANAGEMENT

* Primary actor: User
* Secondary actor: N/A

##### USER ACTION/REPLY

* Action: A user can search or filter users by reputation, tag, location.
* Reply: Search result

## 4.4 Activity Diagrams

#### ACTIVITY DIAGRAM: CREATE QUESTION

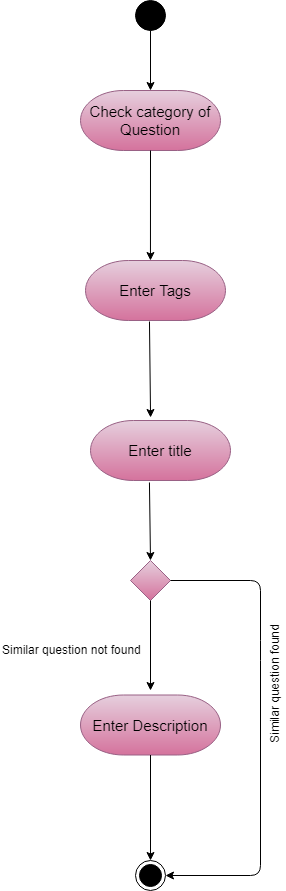


Figure : Level 1.1 Activity diagram – Create question

#### ACTIVITY DIAGRAM: USER REACTION

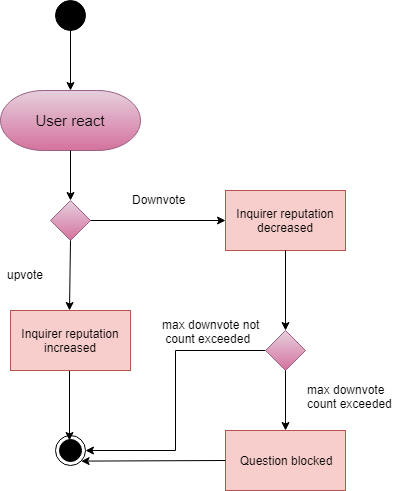


Figure : Level 1.1.1 Activity diagram – User Reaction

#### ACTIVITY DIAGRAM: ANSWER TO QUESTION

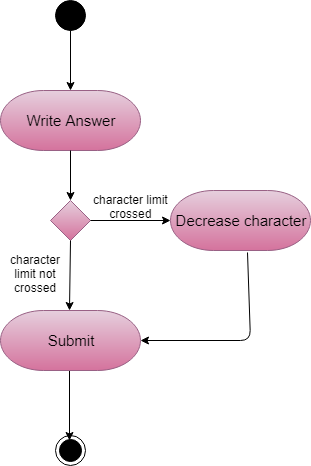


Figure : Activity diagram of Answer to question

#### ACTIVITY DIAGRAM: USER REACTION TO ANSWER

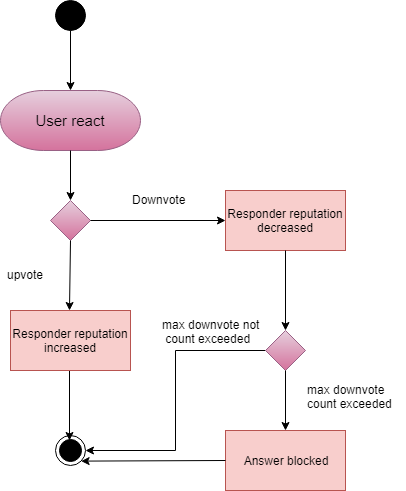


Figure : Level 1.1.2 Activity diagram – User reaction to answer.

#### ACTIVITY DIAGRAM: USER COMMENT

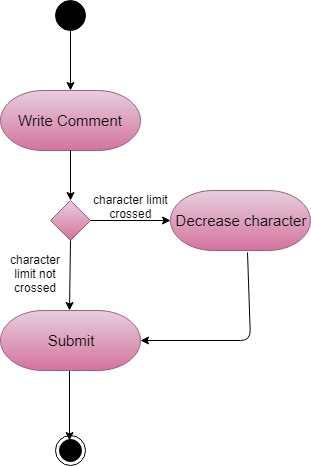


Figure : Level 1.1.3 Activity diagram –User comment

#### ACTIVITY DIAGRAM: USER REACTION TO COMMENT

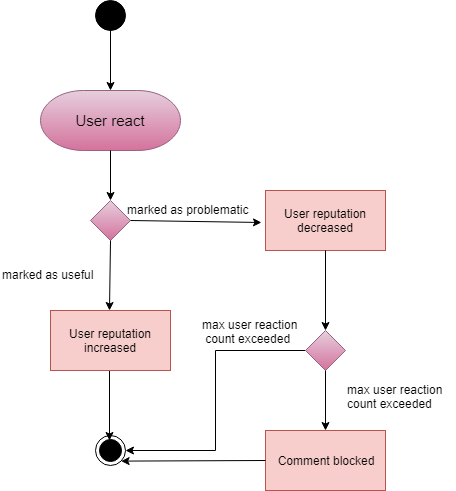


Figure : Level 1.2 Activity diagram – User Reaction to Comment

#### ACTIVITY DIAGRAM: USER SEARCH

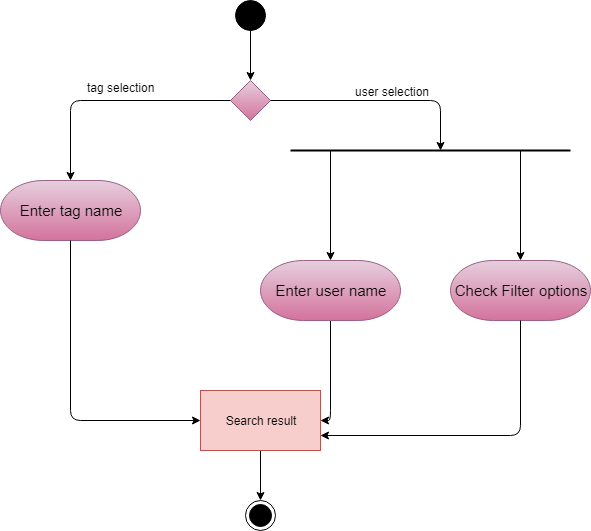


Figure : Level 1.2.1 Activity diagram – User search

## 4.4 swim lane Diagrams

#### swim lane DIAGRAM – 1: AUTHENTICATION

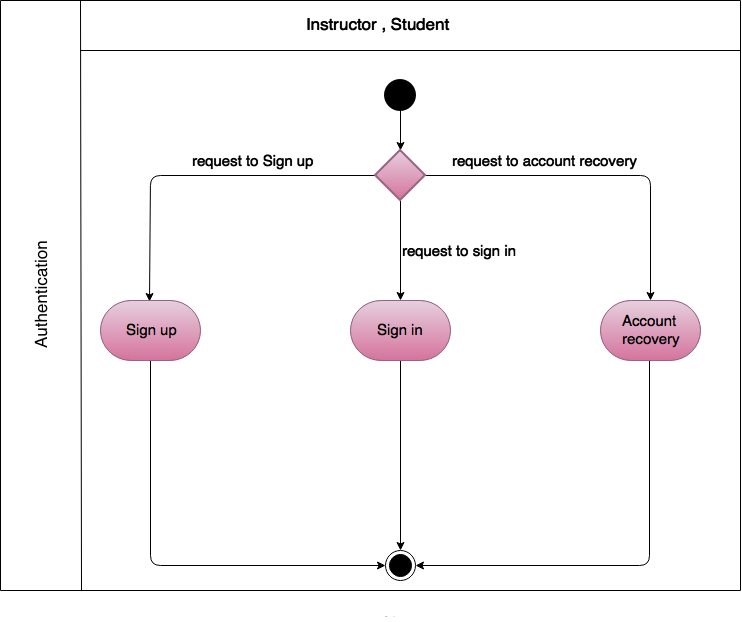


Figure : Level 1.1 Swim lane diagram – Authentication.

#### swim lane DIAGRAM – 1.1: SIGN UP

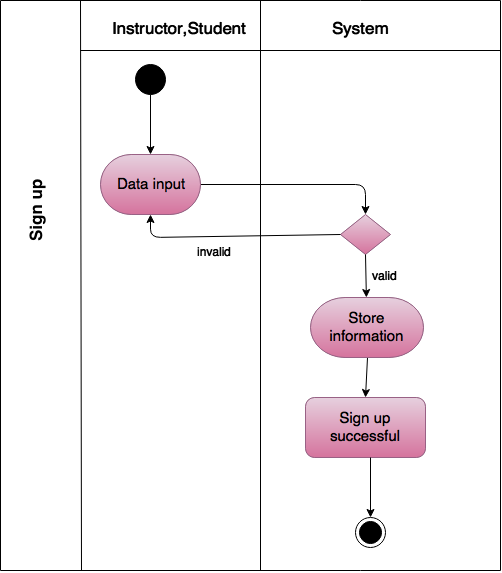


Figure : Level 1.1.1 Swim lane diagram – Sign up.

#### swim lane DIAGRAM – 1.1: SIGN in

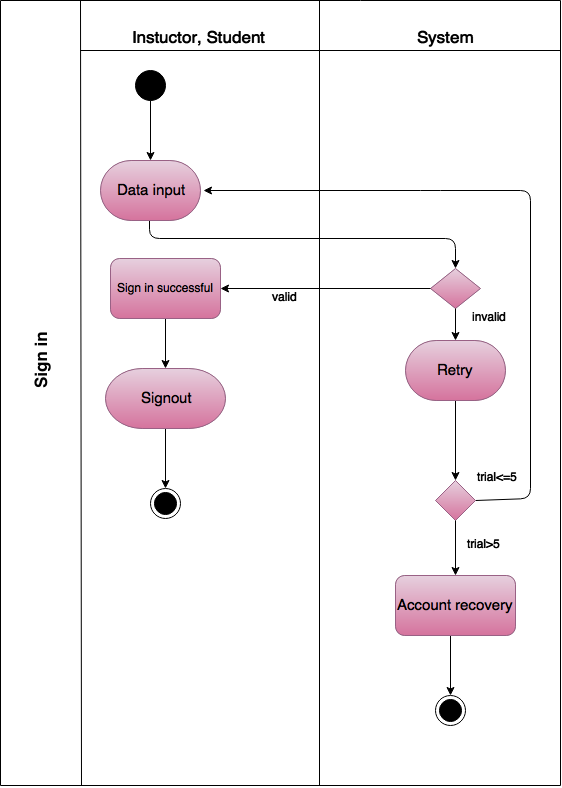


Figure – 28:Figure : Level 1.1.2 Swim lane diagram – Sign in.

#### swim lane DIAGRAM – 1.1: account recovery

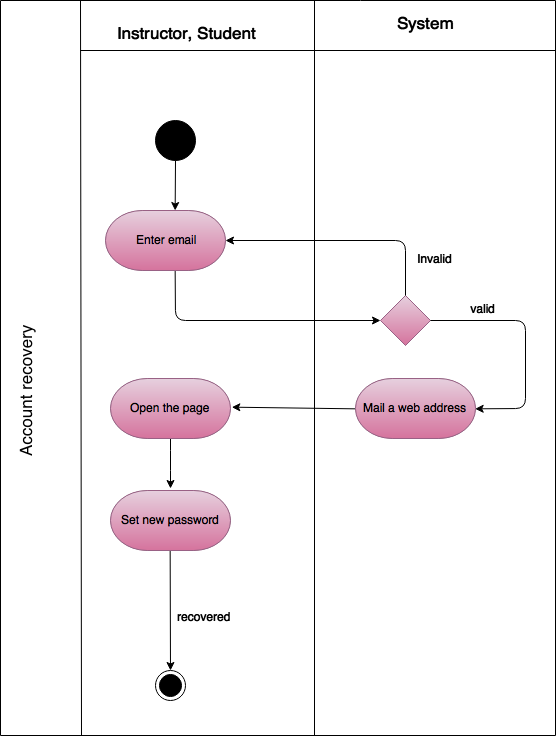


Figure : Level 1.1.3 Swim lane diagram – Account recovery

#### swim lane DIAGRAM – 2: Assignment management

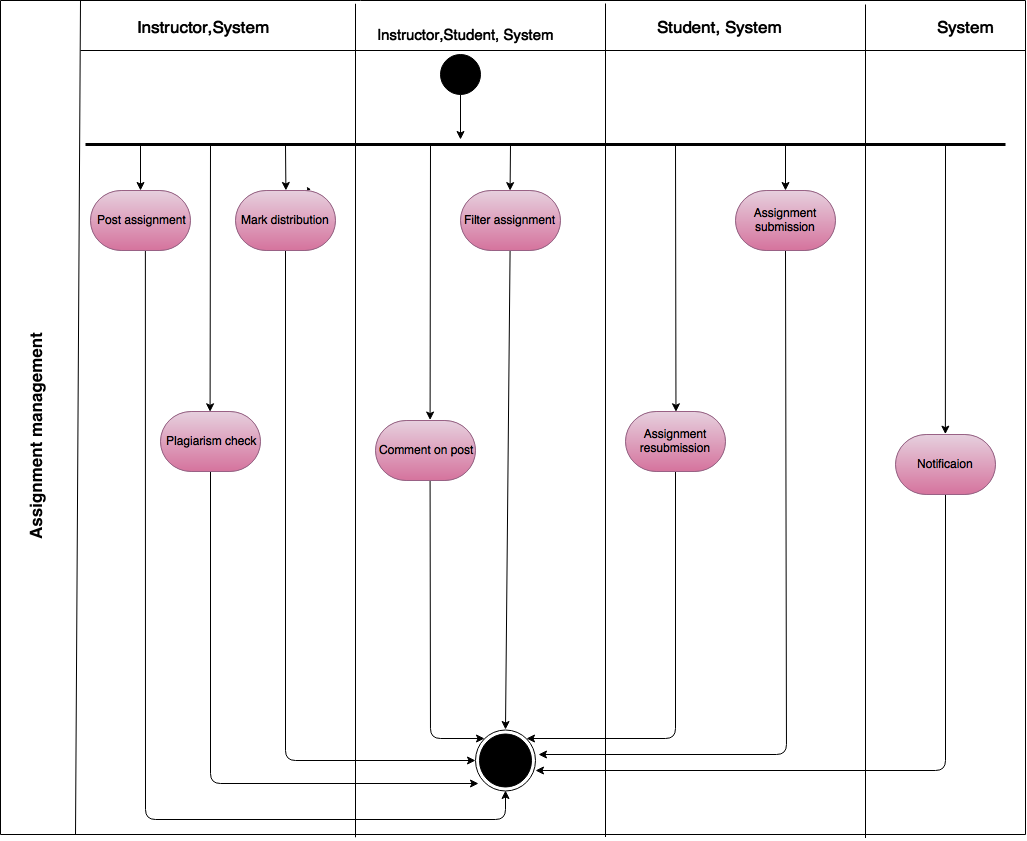


Figure : Level 1.2 Swim lane diagram – Assignment management

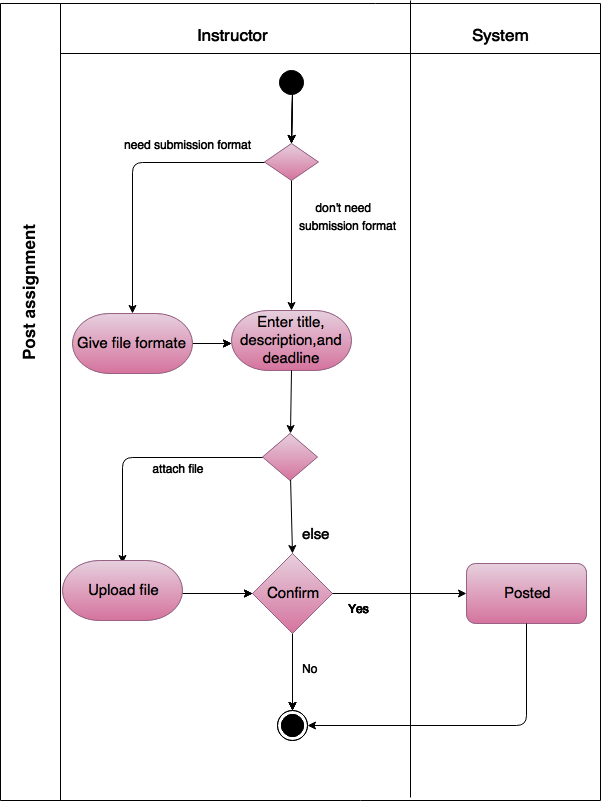


Figure : Level 1.2.1 Swim lane diagram – Assignment post.

#### swim lane DIAGRAM – 2.2: Assignment submission

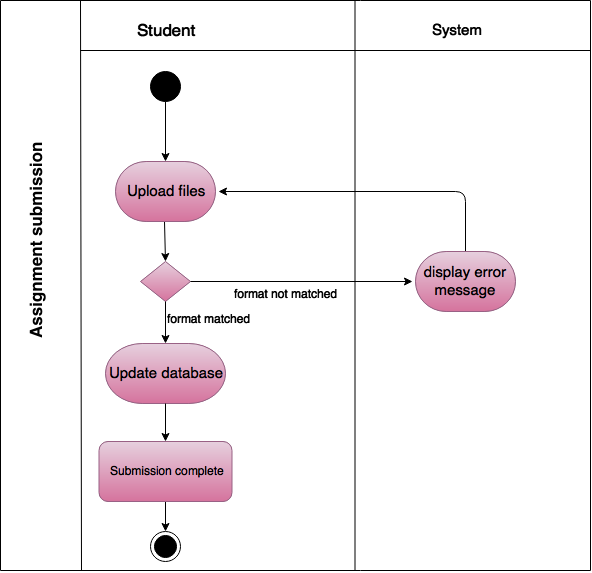


Figure : Level 1.2.2 Swim lane diagram – Assignment submission.

#### swim lane DIAGRAM – 2.3: comment

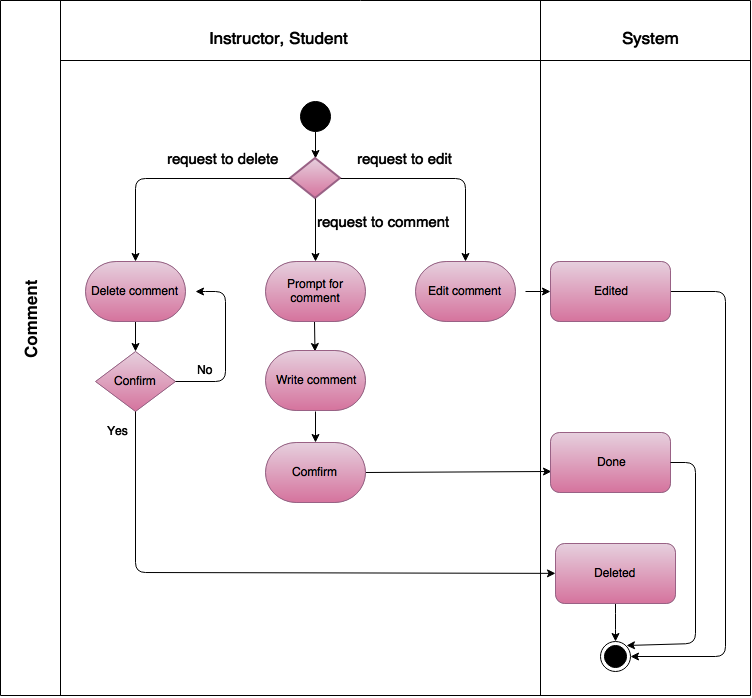


Figure : Level 1.2.3 Swim lane diagram – Comment.

#### swim lane DIAGRAM – 2.4: Assignment resubmission

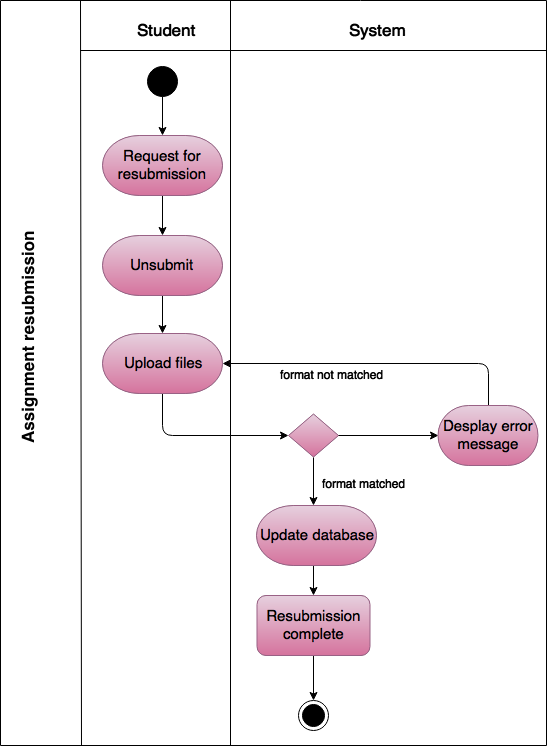


Figure : Level 1.2.4 Swim lane diagram – Assignment resubmission.

#### swim lane DIAGRAM – 2.5: Filter assignment

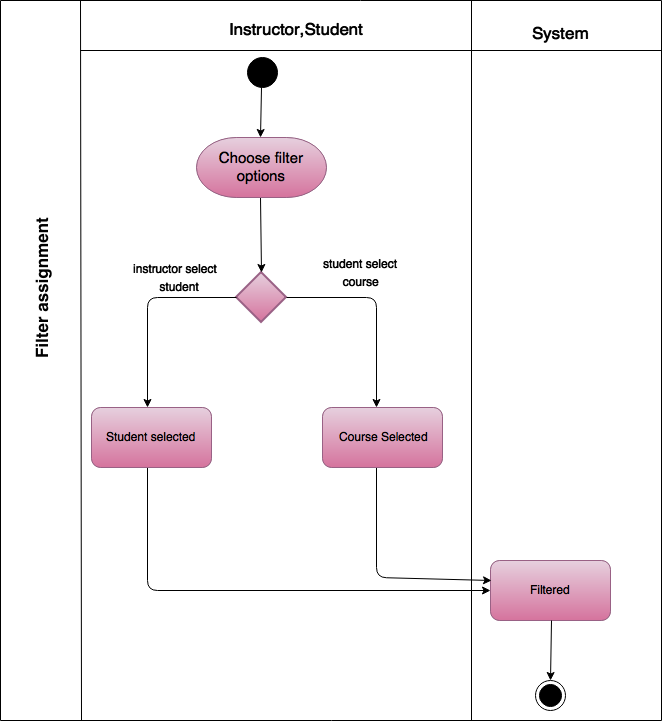


Figure : Level 1.2.5 Swim lane diagram – Filter assignment.

#### swim lane DIAGRAM – 2.6: Plagiarism check

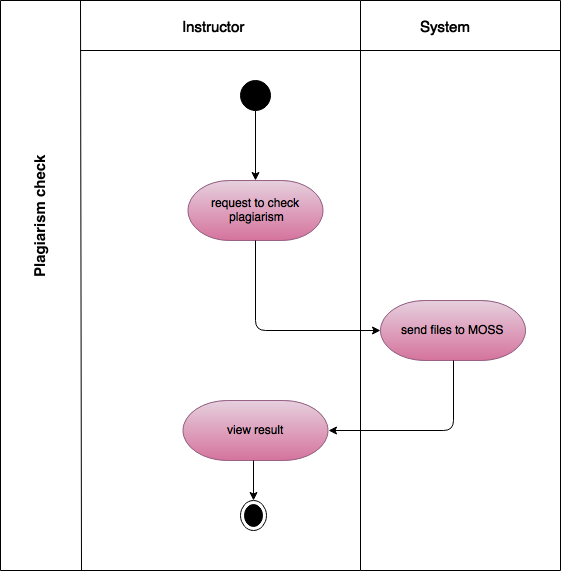


Figure : Level 1.2.6 Swim lane diagram – Sign in.

#### swim lane DIAGRAM – 2.7: notification

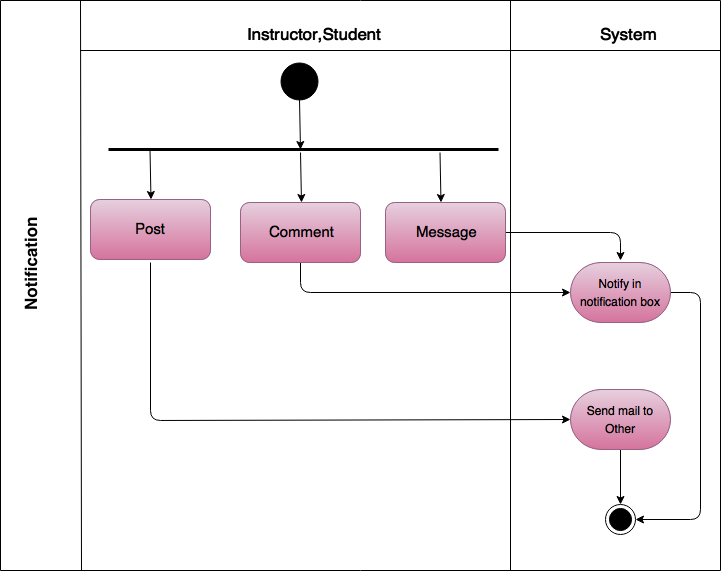


Figure : Level 1.2.7 Swim lane diagram – Notification.

#### swim lane DIAGRAM – 2.8: Mark Distribution

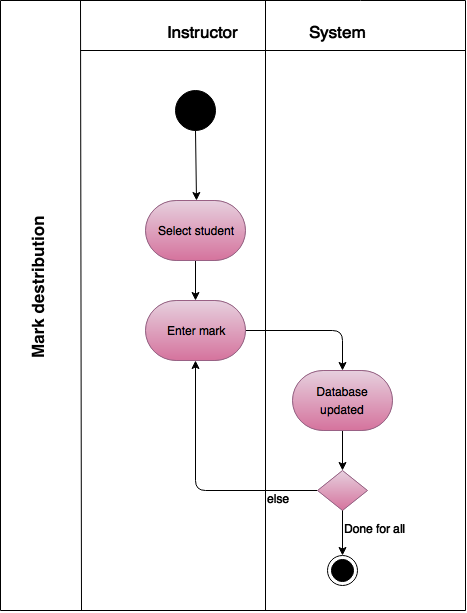


Figure : Level 1.2.8 Swim lane diagram – Mark distribution

#### swim lane DIAGRAM – 3: Group management

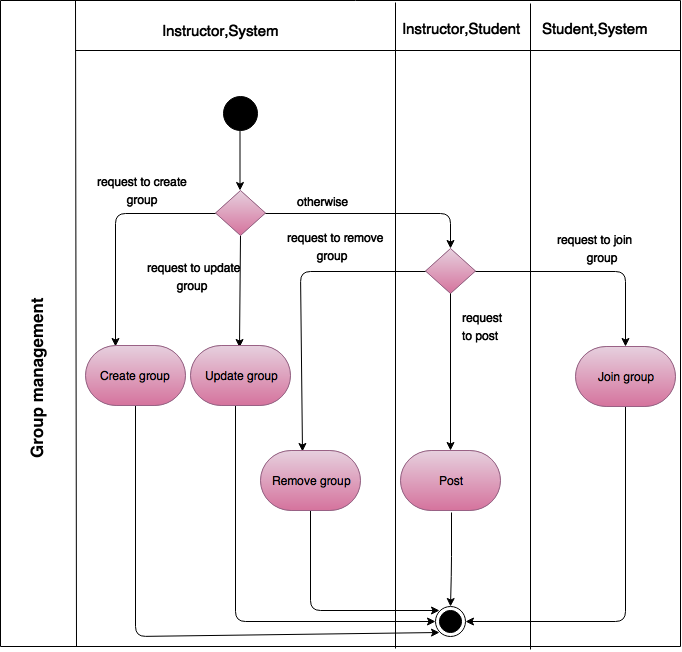


Figure : Level 1.3 Swim lane diagram – Group management

#### swim lane DIAGRAM – 3.1: create Group

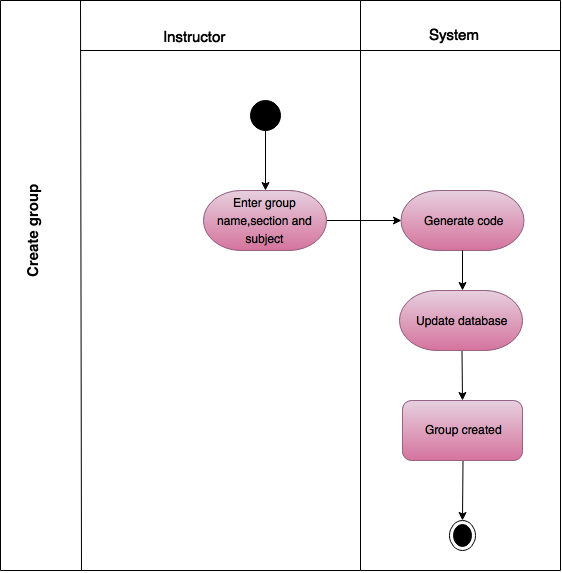


Figure : Level 1.3.1 Swim lane diagram – Create group

#### swim lane DIAGRAM – 3.2: join Group

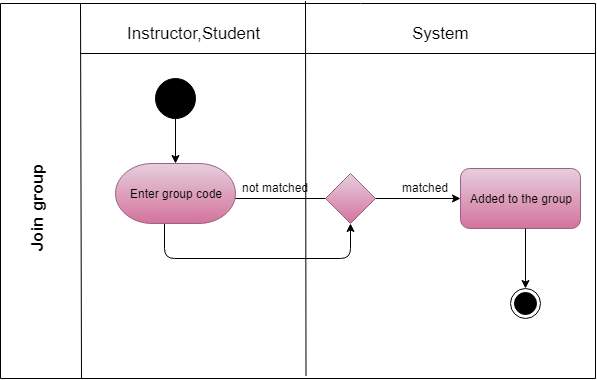


Figure : Level 1.3.2 Swim lane diagram – Join group

#### swim lane DIAGRAM – 3.3: update Group

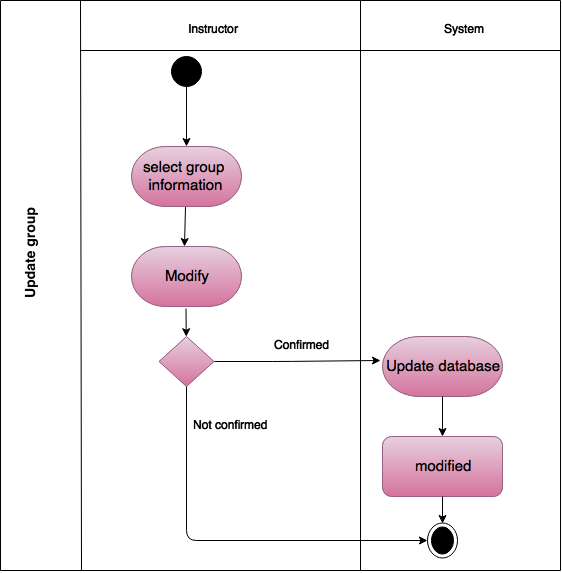


Figure : Level 1.3.3 Swim lane diagram – Update group

#### swim lane DIAGRAM – 3.4: remove Group

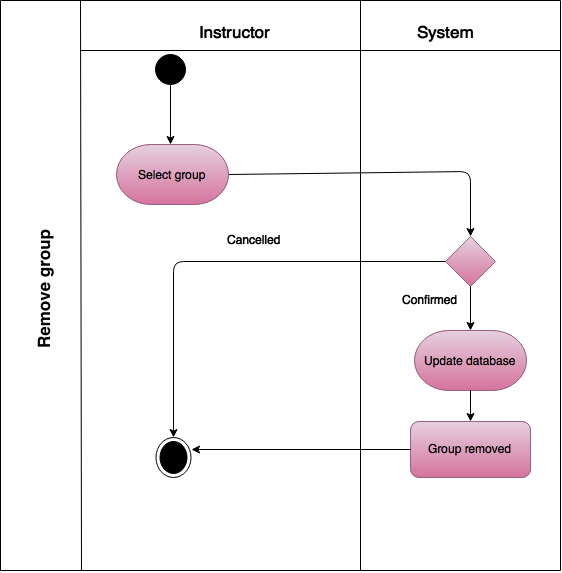


Figure : Level 1.3.4 Swim lane diagram – Remove group

#### swim lane DIAGRAM – 3.5: post

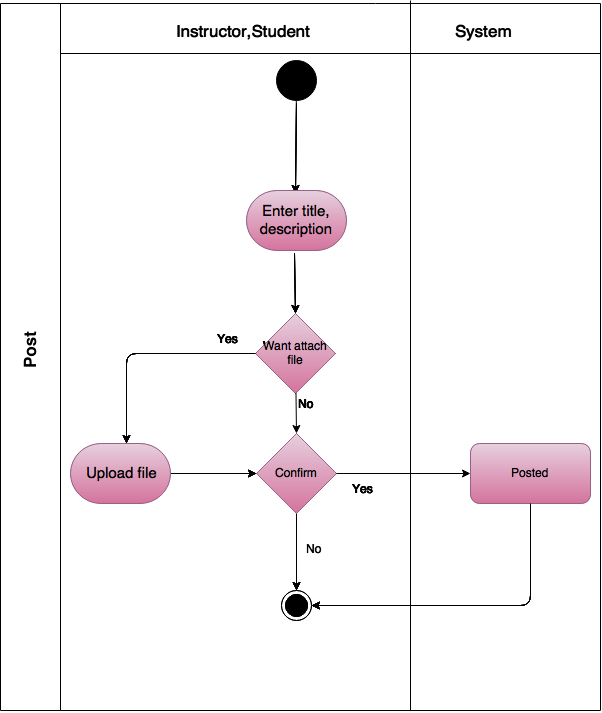


Figure : Level 1.3.5 Swim lane diagram – Post

# 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 : 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

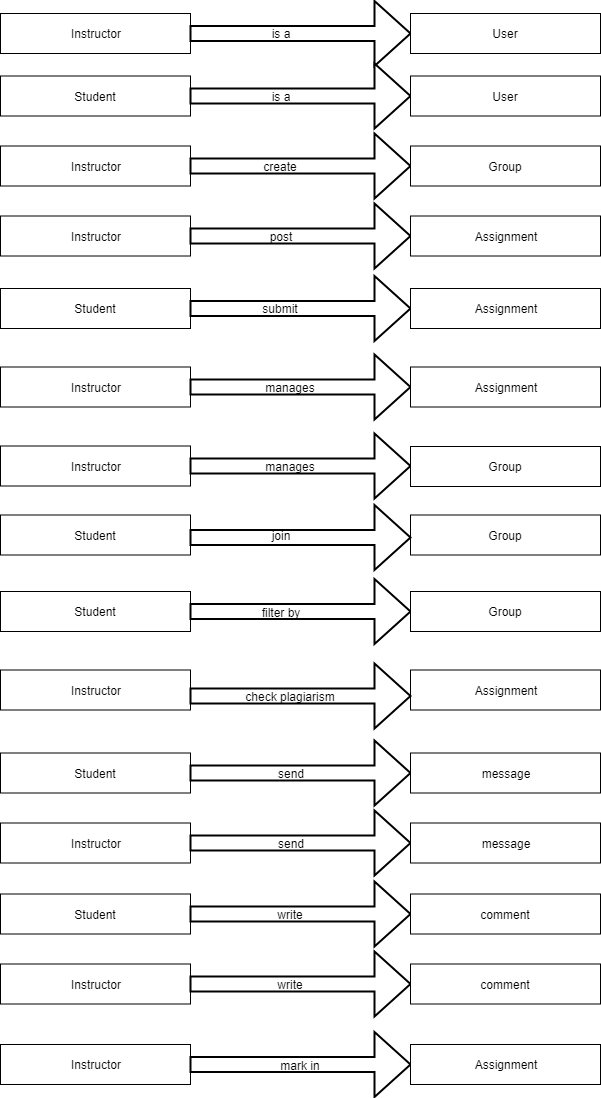


Figure : Relationships between data objects

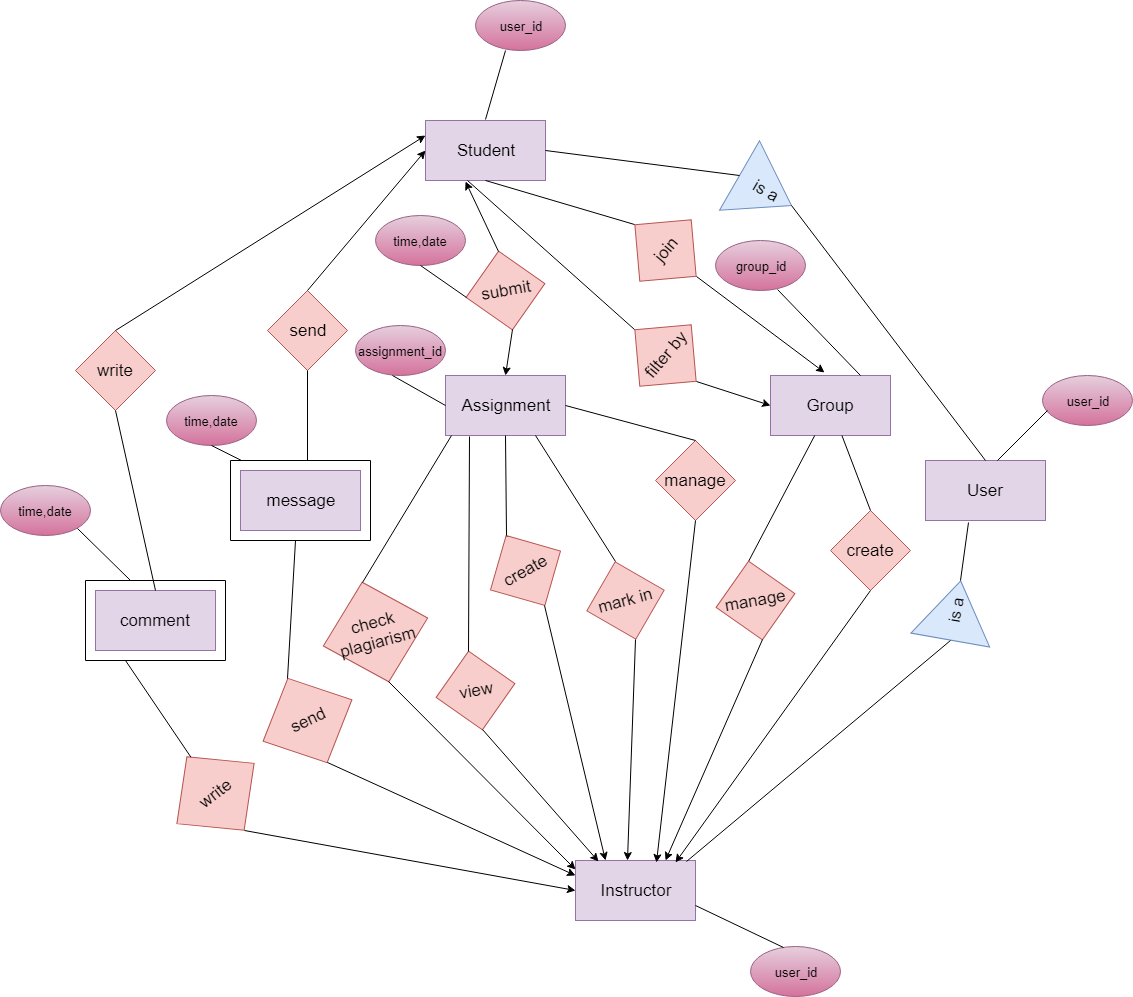


Figure : Entity Relationship of Assignment Management System

## 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 : 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 : 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 : 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 : 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 : 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 : schema table of Message data object

|  |  |  |
| --- | --- | --- |
| Message | | |
| Attributes | **Type** | **Size** |
| **user\_id** | VARCHAR | 40 |
| **time** | VARCHAR | 40 |
| **date** | DATE | 80 |
| **data** | VARCHAR | 40 |

Table : schema table of comment data object

|  |  |  |
| --- | --- | --- |
| Comment | | |
| Attributes | **Type** | **Size** |
| **user\_id** | VARCHAR | 40 |
| **time** | VARCHAR | 40 |
| **date** | DATE | 80 |
| **data** | VARCHAR | 40 |

Table : 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 : 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 Educational Question-Answer Forum for Bangladesh.

## 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 : Noun with general classification

|  |  |  |
| --- | --- | --- |
| Serial Number | Noun | General classification |
| 1 | User | 4,5,7 |
| 2 | Inquirer | 4,5,7 |
| 3 | Responder | 4,5,7 |
| 4 | Question | 3 |
| 5 | Answer | 3 |
| 6 | Forum |  |
| 7 | Reputation | 2 |
| 8 | Comment | 3 |
| 9 | File | 2 |
| 10 | Tags | 2 |
| 11 | Notification | 3 |
| 12 | Description | 2 |
| 13 | Title | 2 |
| 14 | Time | 2 |
| 15 | Recommendation | 3 |
| 16 | Reaction | 3 |
| 17 | User activity | 3,5 |

### 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 : selection criteria of nouns

|  |  |  |
| --- | --- | --- |
| Serial number | Noun | Selection criteria |
| 1 | User | 1,2,3,4,5 |
| 2 | Answer | 3 |
| 3 | Question | 3 |
| 4 | Comment | 3 |
| 5 | Tags | 3 |
| 6 | Recommendation | 2 |
| 7 | UserActivity | 2,5 |

### 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 : Associate noun and verb identification

|  |  |  |  |
| --- | --- | --- | --- |
| No | Class name | Nouns | Verbs |
| 1 | User | UserId, Reputation, Location, Tags | N/A |
| 2 | Answer | Description, Image, Tags, User, Vote count, Comments, Time | N/A |
| 3 | Question | Title, Description, Image, Tags, User, Vote count, Comments, Time | N/A |
| 4 | Comment | Reaction, User, Description | N/A |
| 5 | Tag | Name, Description, UserCount | N/A |
| 6 | UserActivity | N/A | Answer, ask, comment, vote |
| 7 | Recommendation | Question, users | Recommend |

### 6.2.4 ATTRIBUTE SELECTION

Table : class and attributes selection

|  |  |  |
| --- | --- | --- |
| No | Class | Attributes |
| 1 | User | UserId, Reputation, Location, Tags |
| 2 | Answer | Description, Image, Tags, User, Vote count, Comments, Time |
| 3 | Question | Description, Image, Tags, User, Vote count, Comments, Time |
| 4 | Comment | Reaction, User, Description |
| 5 | UserActivity | N/A |
| 6 | Tag | Name, Description, UserCount |
| 7 | Recommendation | Question, users |

### 6.2.5 METHOD IDENTIFICATION

Table : 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 : 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 : 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 : Student

|  |  |
| --- | --- |
| Student | |
| Attributes | Methods |
| firstName  lastName  userName  email  password  phoneNumber  type | submitAssignment()  resubmitAssignment()  joinGroup() |
| Responsibilities | Collaborative class |
| * Submitting assignment * Resubmitting mark * Joining to group | System, Database |

Table : 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 : 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 : 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 : Assignment

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

Table : 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

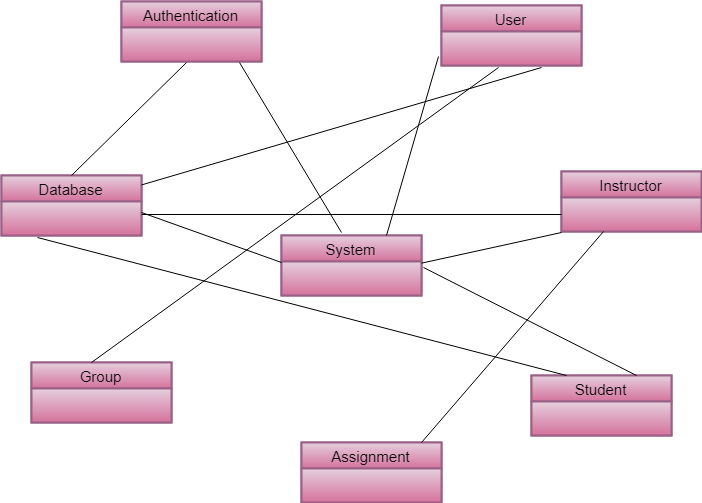


Figure : Class collaboration diagram of AMS

# 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

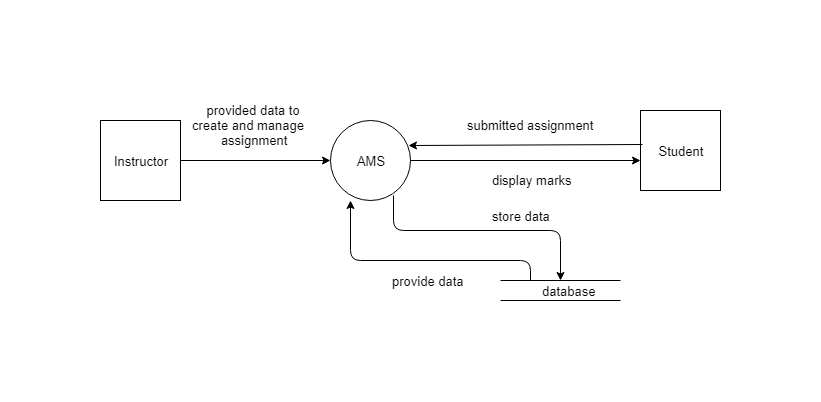


Figure : level – 0 DFD of AMS

### level 1 Data flow diagram

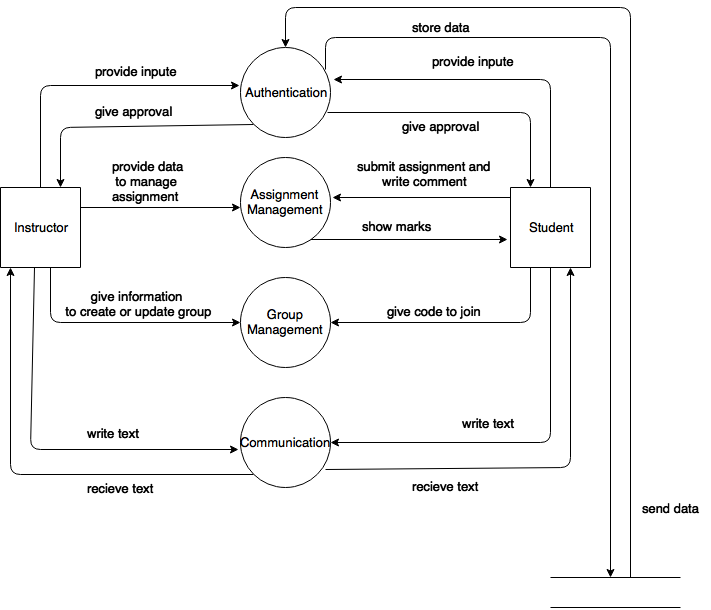


Figure : level 1 DFD of AMS

### level 1.1.1 data flow diagram

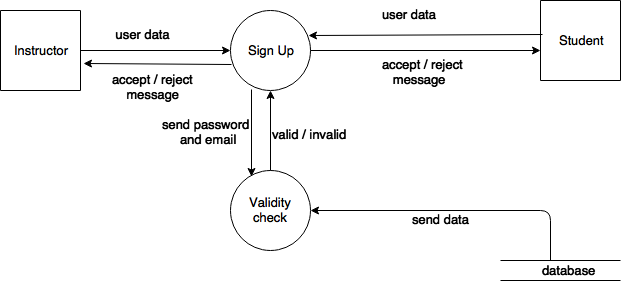


Figure : level 1.1.1 DFD of AMS

### level 1.1.2 Data flow diagram

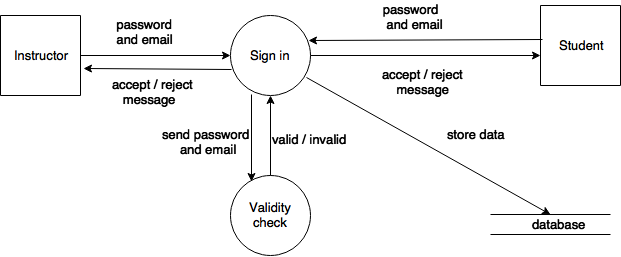


Figure : level 1.1.2 DFD of AMS

### level 1.1.3 data flow diagram

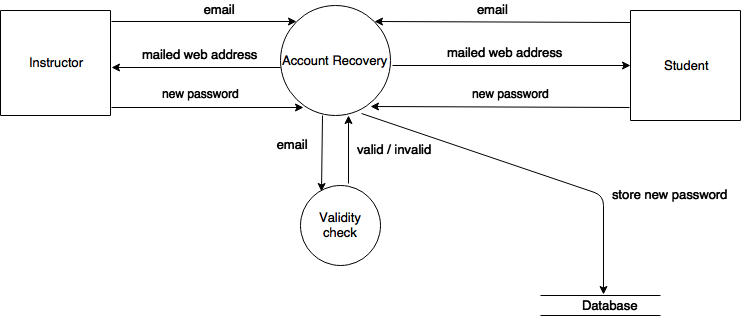


Figure : level 1.1.3 DFD of AMS

### level 1.2.1 Data flow diagram

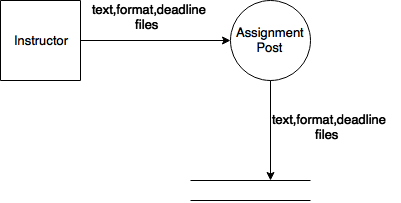


Figure : level 1.2.1 DFD of AMS

### level 1.2.2 Data flow Diagram

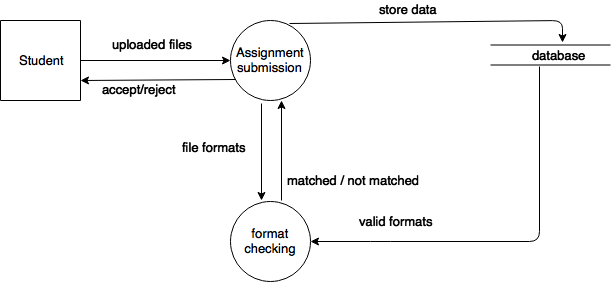


Figure : level 1.2.2 DFD of AMS

### level 1.2.3 data flow diagram

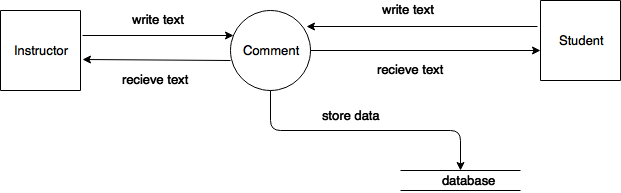


Figure :1.2.3 DFD of AMS

### level 1.2.4 data flow diagram



Figure : level 1.2.4 DFD of AMS

### level 1.2.5 data flow diagram

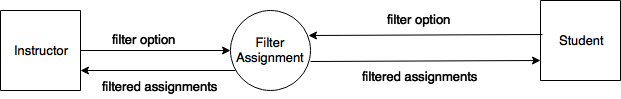


Figure :1.2.5 DFD of AMS

### level 1.2.6 data flow diagram

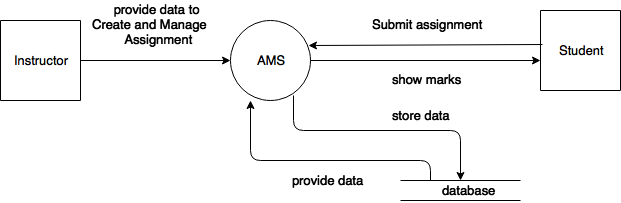


Figure : level 1.2.6 DFD of AMS

### level 1.2.7 data flow diagram



Figure : level 1.2.7 DFD of AMS

### level 1.2.8 Data flow diagram

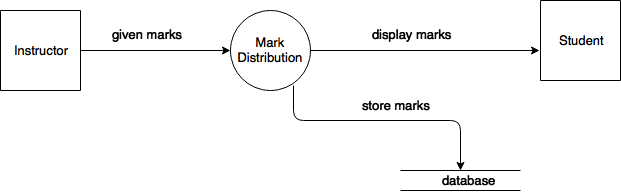


Figure : level 1.2.8 DFD of AMS

### level 1.3.1 DATA flow diagram



Figure : level 1.3.1 DFD of AMS

### level 1.3.2 data flow Diagram

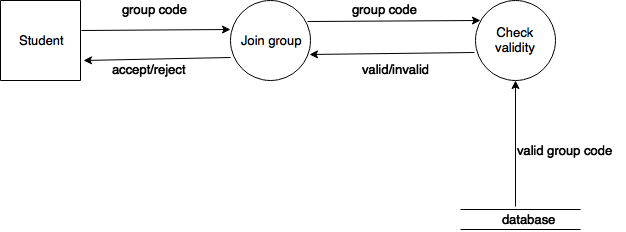


Figure : level 1.3.2 DFD of AMS

### level 1.3.3 data flow Diagram



Figure : level 1.3.3 DFD of AMS

### level 1.3.4 data flow diagram

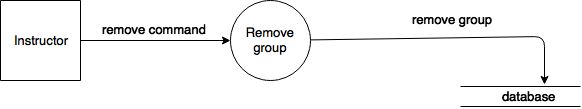


Figure : level 1.3.4 DFD of AMS

### level 1.3.5 data flow diagram



Figure : 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 : 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

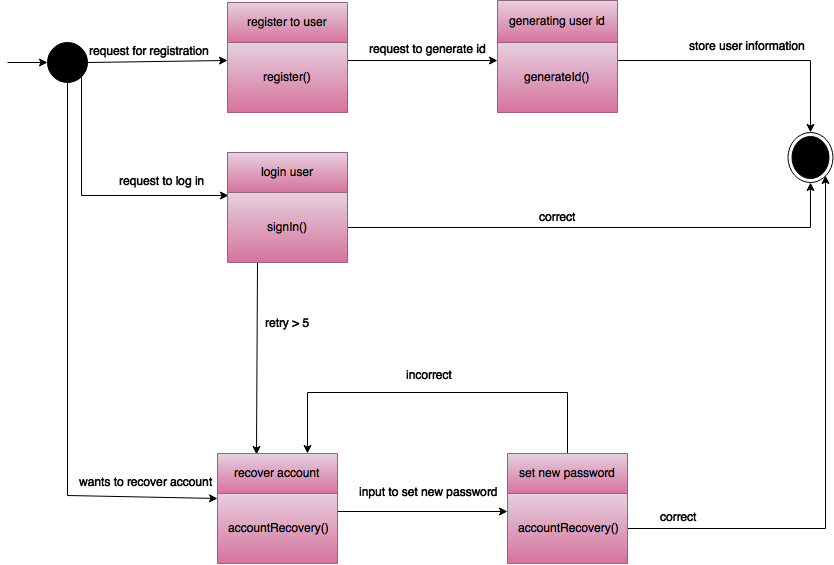


Figure : state transaction diagram – Authentication

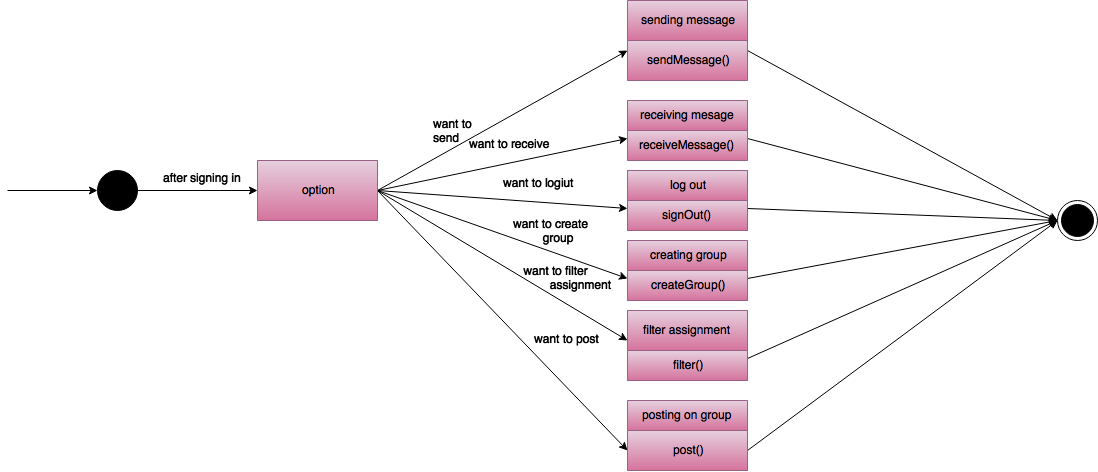


Figure : state transaction diagram – User



Figure : state transaction diagram – Instructor

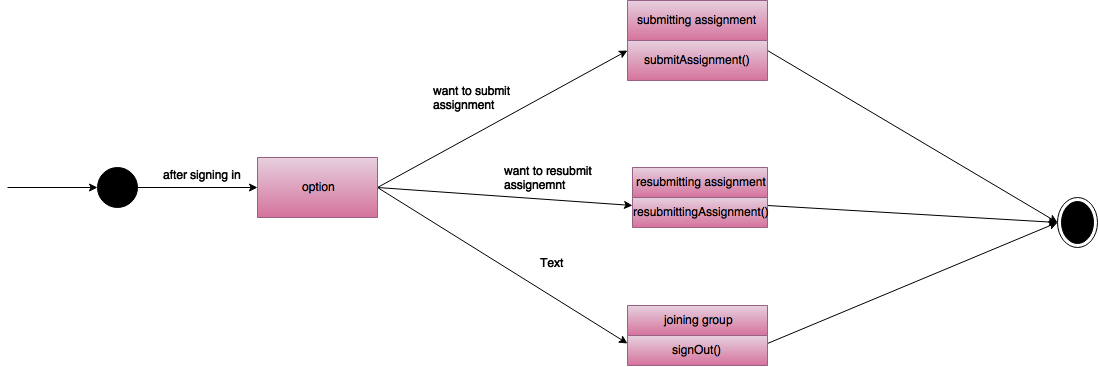


Figure : state transaction diagram – Student



Figure : state transaction diagram- System

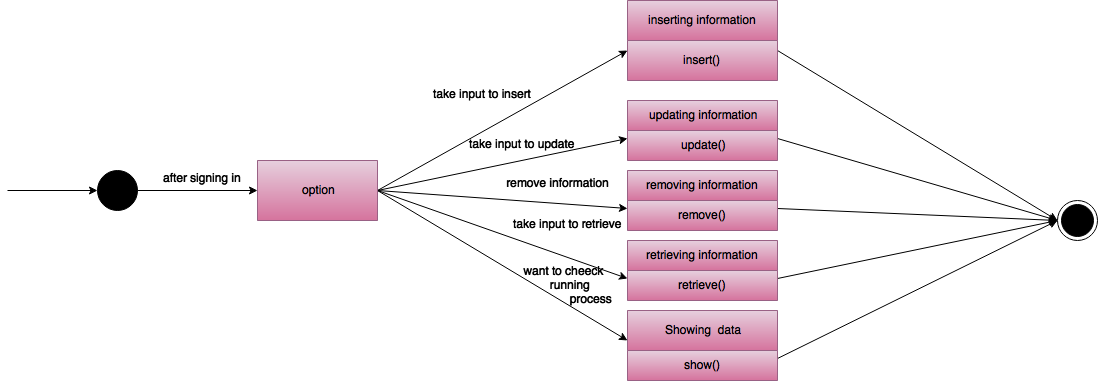


Figure : state transaction diagram- Database

### 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.

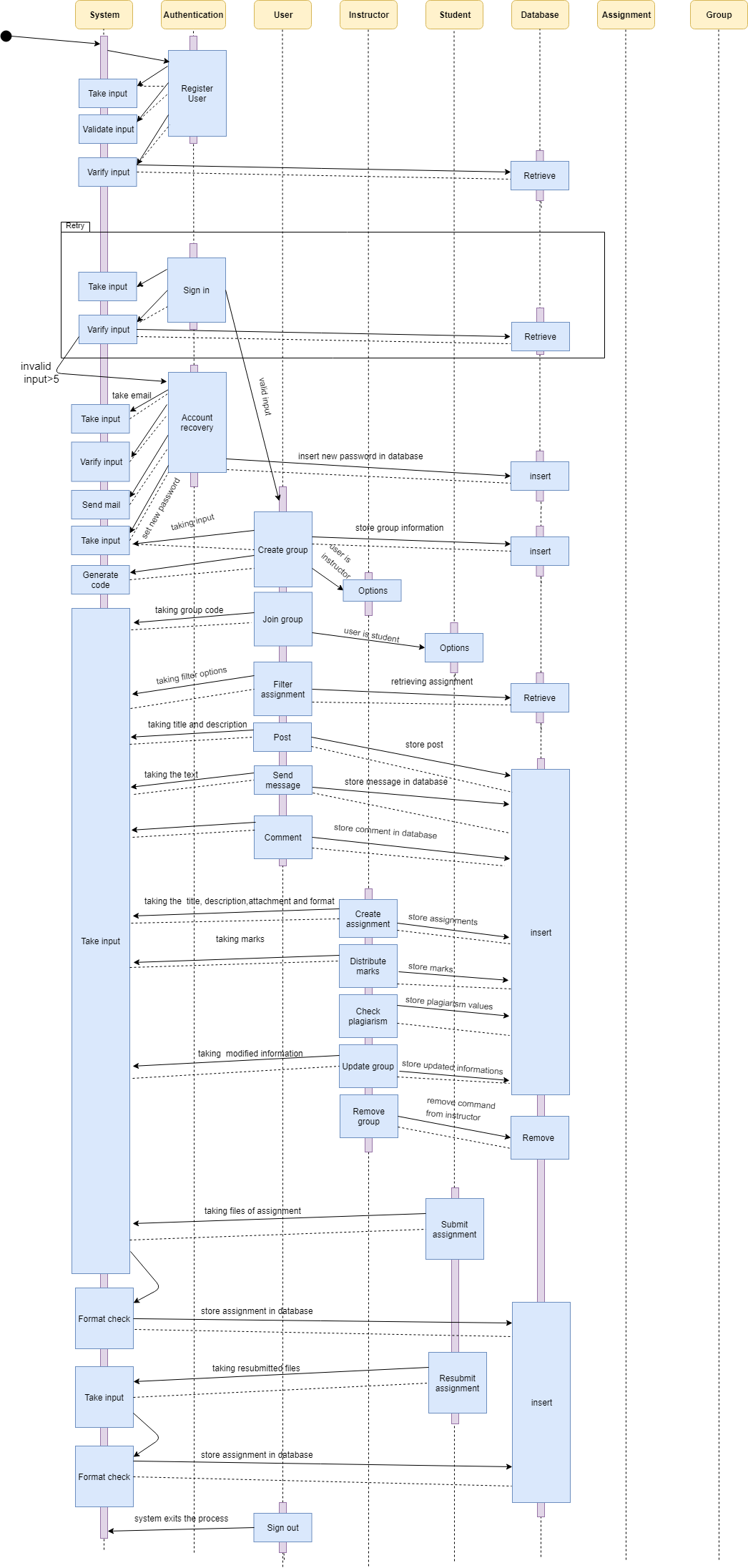


Figure : Sequence diagram

# 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.