

Asknknow – a questioning and answering website

Course: SE505 Software Project Lab – II

Submitted by

<i>Samima Aktar</i> <i>Uddip Acharjee Shuvo</i>	<i>Roll: 0841</i> <i>Roll:0844</i>	<i>2014-2015</i> <i>2014-2015</i>
--	---	--

Document Version 1.0

Submitted to

SPL II Committee



Institute of Information Technology

University of Dhaka

[20-03-2018]

LETTER OF TRANSMITTAL

20th March, 2018

Md. Nurul Ahad Tawhid

Assistant Professor,

Institute of Information Technology,

University of Dhaka

Subject: Submission of Software Requirement Specification report of Software Project Lab-II

Sir,

With due respect, we are very glad to submit our Software Requirement Specification report of our project **Asknknow** – a questioning and answering website. Although this report may have many drawbacks, we have given our highest effort to fulfill all the requirements of our stakeholders and to prepare a detailed documentation that will help us in future.

Therefore, we hope that you will be kind enough to overlook our mistakes and acknowledge the our attempt that we have put in this document.

Sincerely,

Samima Aktar (0841)

Uddip Acharjee Shuvo (0844)

Institute of Information Technology,

University of Dhaka

Signature of Supervisor

ACKNOWLEDGEMENT

First of all I would like to thank the Almighty for enabling us to complete this project.

Then we would like to express our heartiest gratitude to all of those who helped us to accomplish the tiresome work of identifying the requirements for the project titled 'asknknow'. We would like to specially thank to Md. Nurul Ahad Touhid, our supervisor for helping us to gather requirement as well as keeping ourselves connected with the project stakeholders.

Finally, we are grateful to the Institute of Information Technology for giving us the opportunity to do a project of great caliber.

ABSTRACT

This document contains the software requirements and specifications for Software Project Lab-2. It contains a detailed depiction of the requirements of our project 'Asknknow'. It includes a scenario-based model, data-based model, class-based model and behavioral model. Using this document as a guide, we are describing the requirements, necessary diagrams, procedures, designs for database and working sequence of our project.

This document will help to make the software according to the demand of the stakeholders.

Table of Contents

CHAPTER 1: Introduction of SRS	1
1.1 Purpose	1
1.2 Intended Audience.....	1
1.3 Conclusion.....	1
CHAPTER 2: INCEPTION.....	2
2.1 Introduction.....	2
2.2 Establishing the groundwork.....	2
2.3 List of Stakeholders	2
2.3 Recognizing Multiple Viewpoints	3
2.4 Working Towards Collaboration.....	3
2.5 Requirement Questionnaire	4
2.6 Conclusion.....	4
CHAPTER 3: ELICITATION.....	5
3.1 Introduction.....	5
3.2 Eliciting Requirements	5
3.2.1 Collaborative Requirements Gathering:.....	5
3.2.2 Quality Function Deployment:	6
3.3 Usage Scenario	7
CHAPTER 4 : SCENARIO-BASED MODELLING	10
4.1 DEFINITION OF USE CASE	10
4.2 ACTORS	10
4.3 USE CASE DIAGRAMS	11
4.3.1 Level-0 Use Case Diagram-ANK.....	11
4.3.2 Level-1 Use Case Diagram-Subsystems.....	12
4.3.3 LEVEL- 1.1 Use Case Diagram-Authentication	13
4.3.4 LEVEL- 1.1.1 CREATING ACCOUNT	14
4.3.5 LEVEL 1.1.2 USE CASE DIAGRAM-ACCESSING ACCOUNT	15
4.3.6 LEVEL-1.2 USE CASE DIAGRAM-ACTIVITY MANAGEMENT	16
4.3.7 LEVEL-1.3 USE CASE DIAGRAM-QUESTION MANAGEMENT	17
4.3.8 LEVEL-1.4 USE CASE DIAGRAM-USER MANAGEMENT	18
4.4 ACTIVITY DIAGRAM OF ANK.....	19
4.5 SWIMLANE DIAGRAM OF ANK.....	25

CHAPTER 5: DATA-BASED MODELLING	31
5.1 Grammatical Parsing and Analysis	31
5.2 Potential Data Objects	32
5.3 Analysis of Potential Data Objects.....	33
5.4 Final Data Objects	33
5.5 Data Object Relationship	34
5.6 Entity Relation Diagram	34
5.7 Schema Tables.....	35
CHAPTER 6: CLASS BASED MODELLING	37
6.1 Grammatical Parsing and Analysis	37
6.1.1 Class Identification with General Classification	37
6.1.2. Class Identified With Selection Criteria.....	39
6.2 Preliminary Classes	40
6.3 Verb Identification.....	40
6.4 Attributes and Methods of Preliminary Classes.....	42
6.5 Analysis of Potential Classes.....	43
6.6 Final Classes	43
6.7 Attributes, Methods of Final Classes.....	43
6.7 Responsibilities and Collaborative Classes of Final Classes.....	44
6.8 Class Diagram	45
CHAPTER 7: FLOW-ORIENTED MODELING	47
7.1 Data flow diagram	47
7.1.1 Level 0 Data Flow Diagram	47
7.1.2 Level 1 Data Flow Diagram	47
7.1.3 Level 2 Data Flow Diagram	48
CHAPTER 8: BEHAVIORAL MODELING	49
8.1 Event Identification	49
8.2 State Transition	50
8.3 Sequence diagram	53

CHAPTER 1: Introduction of SRS

1.1 Purpose

This chapter aims to specify the purposes of this document and its intended audiences. This document is the Software Requirement Specification (SRS) for “Asknknow”- a questioning and answering website. It focuses on elaborately defining the functional and nonfunctional requirements of the system as well as provides an outline of the core design of the complete system. Specifically, this document is written to:

- Build a common reference point for all stakeholders, including but not limited to customers, developers, testers to reduce communication overhead.
- Provide an unambiguous and consistent high level overview of core design of the system for developers to implement.
- Serve as a basis for preparing a general schedule and future modifications in the system.

1.2 Intended Audience

This SRS is intended for several audiences including customers, designers, developers and testers.

- The customers will use this document to ensure their functional requirements has properly been met in the end product.
- The project manager will use this document to plan a schedule and estimate cost and delivery date.
- The designer will use the high level architecture outlined in this document to design the complete system to meet the customer requirements, which is also detailed in this document.
- The developers will periodically refer back to this document to ensure their implementation corresponds to the customer requirements.
- The testers will use this document to get a clear picture of functionality of individual components and whole system to test against.

1.3 Conclusion

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

CHAPTER 2: INCEPTION

In this chapter, the inception part of the software requirement analysis of our project will be discussed briefly.

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.

2.1 Introduction

Inception is the beginning phase of requirements engineering. It defines how does a software project get started and what is the scope and nature of the problem to be solved. The goal of the inception phase is to identify concurrence needs and conflict 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.

2.2 Establishing the groundwork

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

- Identifying all stakeholders.
- Recognizing multiple viewpoints
- Working towards collaboration.
- Asking requirement questions.

2.3 List of Stakeholders

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

We defined stakeholders as anybody or organization directly or indirectly associated with the end software. We identified following entities as our system's stakeholder:

- Registered users
- General users
- Website administrator
- Project Manager, Developer and tester

2.3 Recognizing Multiple Viewpoints

As many different stakeholders exist, the requirements of the system are explored from many different points of view.

User's viewpoint

- The system should have easy and user friendly way for accessing all the questions
- The questions should be categorized for finding them quickly
- Every registered user should be able to see their activities such as asked question list, answered question list.
- There should be voting system for every answers of a question so that it becomes easier to evaluate or assess the standard of the answer
- It should provide with the facility to sort out the experts of a specific subject.
- General user can see the questions and answers without registration.

Website Admin's viewpoint

- There should be easy and simple arrangement for administration activities.
- The system must provide with the notification system for the experts so that they can get notification when a question is asked of their domain area
- It must include with high security and strong authentication.

Developer's viewpoint

- Total development time should not exceed two months
- Top security and highly efficient features should not be expected

2.4 Working Towards Collaboration

Every stakeholders has their own requirements from their own point of view. We followed following steps to merge all the requirements. We-

- Identified the common and conflicting requirements
- Categorized the requirements
- Take priority points for each requirements from stakeholders
- Make final decision about the requirements

Our first step was to find the common grounds among multiple viewpoints. We singled out the conflicting ones to finalize our overall requirements.

Common requirements

- User friendly
- Efficient system
- Error free
- Strong authentication

Conflicting requirements

- Website administrator expects a highly secure system while for the development team it is not possible to provide complete security.
- Too many requirements to be fulfilled within a short time. So some requirements cannot be fulfilled such as secure communication, and highly efficient modules

Final requirements

- Notification to the experts of an asked question should be automated
- Voting system for each answer should be provided
- Categorized list of question should be given
- Admin can add new user and remove existing user

2.5 Requirement Questionnaire

Our questions were initially context free, which focused on the user and other stakeholders, the overall project goals and benefits. This created a basic understanding of the problem for us. After having a general concept of the system, we got into details. Our second set of questions were intended to provide us a deeper understanding of the problem and solution. And lastly, we judged the effectiveness of our communication through a set of meta questions.

2.6 Conclusion

The Inception phase helped us to establish basic understanding about the “Asknknow” (ANK) identify the stakeholders who will be benefited if this system becomes automated, define the nature of the system and the tasks done by the system, and establish a preliminary communication with our stakeholders.

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

CHAPTER 3: ELICITATION

3.1 Introduction

Requirements elicitation is also called requirements gathering. The purpose of this step is to combine elements of problem solving, elaboration, negotiation and specification. In requirements engineering requirements elicitation is the practice of collecting the requirements of a system from users, customers and other stakeholders. We have faced many problems like understanding the problems, problems of making questions for the stakeholders, problems of less communication with the stakeholders for time limitation, problems of volatility. Though it is not too easy to gather requirements within a very short time, we have surpassed these problems in an organized and systematic manner.

3.2 Eliciting Requirements

In requirements engineering, requirements elicitation is the practice of collecting the requirements of a system from users, customers and other stakeholders. It specifies the answer to the questions like what the objectives for the system or product are, what is to be accomplished, how the system or product fits into the needs of the business, and finally, how the system is to be used on a day-to-day basis. We have seen Question and Answer (Q&A) approach in the previous chapter where the inception phase of requirement engineering has been described. The main task of this phase is to combine the elements of problem solving, elaboration, negotiation and specification. We have finished the following tasks for eliciting requirements-

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

3.2.1 Collaborative Requirements Gathering:

We have met with many stakeholders in the Inception phase such as the students and teacher of different institution of various department. They were questioned about their requirements, expectations and expected structure of the program. These meetings created an indecisive state for us to elicit the requirements. To solve this problem we have met with the stakeholders (who are acting a vital rule in the whole process) again to elicit the requirements. We discussed to

define the scopes and extents to which we could develop the system. A slightly different scenario from these approaches has been found.

3.2.2 Quality Function Deployment:

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

Normal Requirements:

Normal Requirements are the objectives and goals that are stated for system during meetings with the customer. Normal requirements of ANK are:

1. Signup and login system of users
2. Asking or post question section in text or image
3. Comment or reply section only for the registered users.
4. Up vote and down vote system by the viewers for a specific question and the related answer of the users.
5. View count system for a particular question
6. Question list sorted based on time of posting, most voted, unanswered etc.
7. Question tag for specifying questions of particular domain
8. Searching section for question

Expected Requirements:

1. Providing personal account for user
2. Secured authentication system
3. System admin should be able to approve new user or delete users

Exciting Requirements:

Exciting Requirements are those features that go beyond the customer's expectations and prove to be very satisfying when present. Exciting requirements of ANK are:

1. Ranking system of the users

3.3 Usage Scenario

“Asknknow” -a questioning and answering website serves the following purposes:

1. Authentication:
2. Activity management:
3. Question management
4. User management
5. Notification

Authentication:

ANK has three types of users.

1. Admin
2. Registered user
3. General user

Any person can use this website as a general user. Then no authentication will be needed. But this time user will be restricted to access only specific area. To be a registered user one has to give the following input.

- user name
- Full name
- Designation
- email
- password

These data will be stored in database. Every time users access into the website for log in they have to provide the following data.

- User name
- password

These data will be verified every time the registered user log in. For user name, only alphanumeric characters and underscore (‘_’) are allowed. If a user tries to create an account with empty field or contain invalid input, the system shows an error message and allows him/her to try again.

To be an admin, approving from another admin is required.

Activity management:

This module is divided into three subsystem. They are as follows:

1. Search
2. Sort
3. Approve or delete

Under the “search” module searching are performed by different attributes. A user can search for questions, user and category.

“Sort” subsystem provides with the necessary functionalities to sort the questions according to most voted, highest answered, unanswered, date. Thus it ensures the quick and easy access to the expected query.

“Approve/delete” subsystem is reserved only for the administration purposes to accept or reject request for submitting questions and new admin.

Question management:

“Question management” is further divided into five more subsystems. They are as follows:

1. Question List
2. Category List
3. Ask
4. Answer
5. Upvote / Downvote

In the “Question List” subsystem, all the questions will be preserved and maintained. All the users can access the questions repository.

There will be predefined or user-defined category or “Tag” for any question in the “Category List”. Any user can access the category List. Registered user can make new Tag/s while submitting any question that belongs to completely new one. Tag/s refers to the keyword/s of that content. A question may have multiple tags. User can also edit tag/s. Registered and non-registered users can search question by using tag/s.

“Ask” subsystem will provide with the functionalities to ask a question only for the registered user. When a general user intends to ask a question, then s/he will be prompted with the opportunity for becoming a registered user by displaying the sign-up page. To ask a question a user have to provide with the following information.

- Title of the question
- Category of the question
- Question Description
- Designation of the questionnaire
- Designation of the target audience

After giving all the information, user will submit it to publish in the question repository. Then an approval from the admin panel is required. If the question is eligible for publishing, it will be published otherwise a notification message will be sent to the user.

“Answer” subsystem will provide with the functionalities to answer any question from the question list. Only the registered user can answer any question.

“Upvote/Downvote” subsystem will allow the users to appreciate or discourage the answer by casting vote. If s/he likes the answer given by other user, s/he can upvote the answer and vice versa. Only the registered users are accessible for this subsystem. Ranking system of the users will be provided based on upvote, downvote and contribution.

User management:

“User management” is the subsystem for the maintenance of all the user related data. It is divided into further subsystems as follows:

- User profile
- User activity history
- User specialized domain

All the data of the user account will be maintained under this “User profile” module. A registered user can access, modify and manipulate these data.

“User activity history” will store all the activities of any specific registered user account. A registered user can ask and answer. This module will maintain the list of all the asked questions, answered questions, and the most answered category of the registered user.

“User specialized domain” will save all the domain or area of which a registered user prefers to answers. S/he may have a good command over those area. So, when any other user ask a query of those domain, s/he will be notified about this question.

Notification:

In the “ANK” system there will be three types of notification. They are as follows:

When a new user account is created by a general user to be registered as an admin, a notification is sent to the existing admin panel for approval or rejection him/her as a new admin.

When a registered user ask a question submitting it to publish, a notification is sent to the admin to consider the question whether it is eligible to publish.

When a registered user submit a question of a specific domain, notification will be sent to all the other registered users who are expert at that domain.

CHAPTER 4 : SCENARIO-BASED MODELLING

4.1 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.

4.2 ACTORS

The first step in writing a Use Case is to define that set of “actors” that will be involved in the story. Actors (such as admin, operator) 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 DIAGRAMS

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

4.3.1 Level-0 Use Case Diagram-ANK

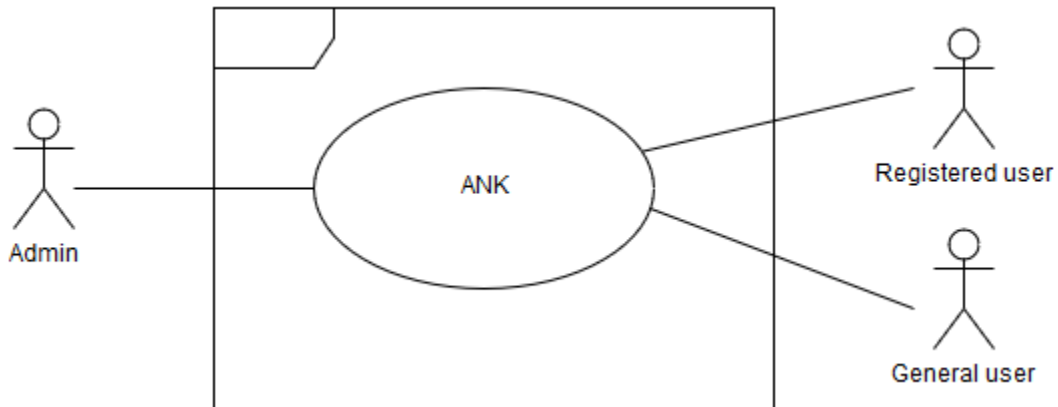


Figure: Level 0 Use Case diagram-ANK

Description of Use Case Diagram Level 0:

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

4.3.2 Level-1 Use Case Diagram-Subsystems

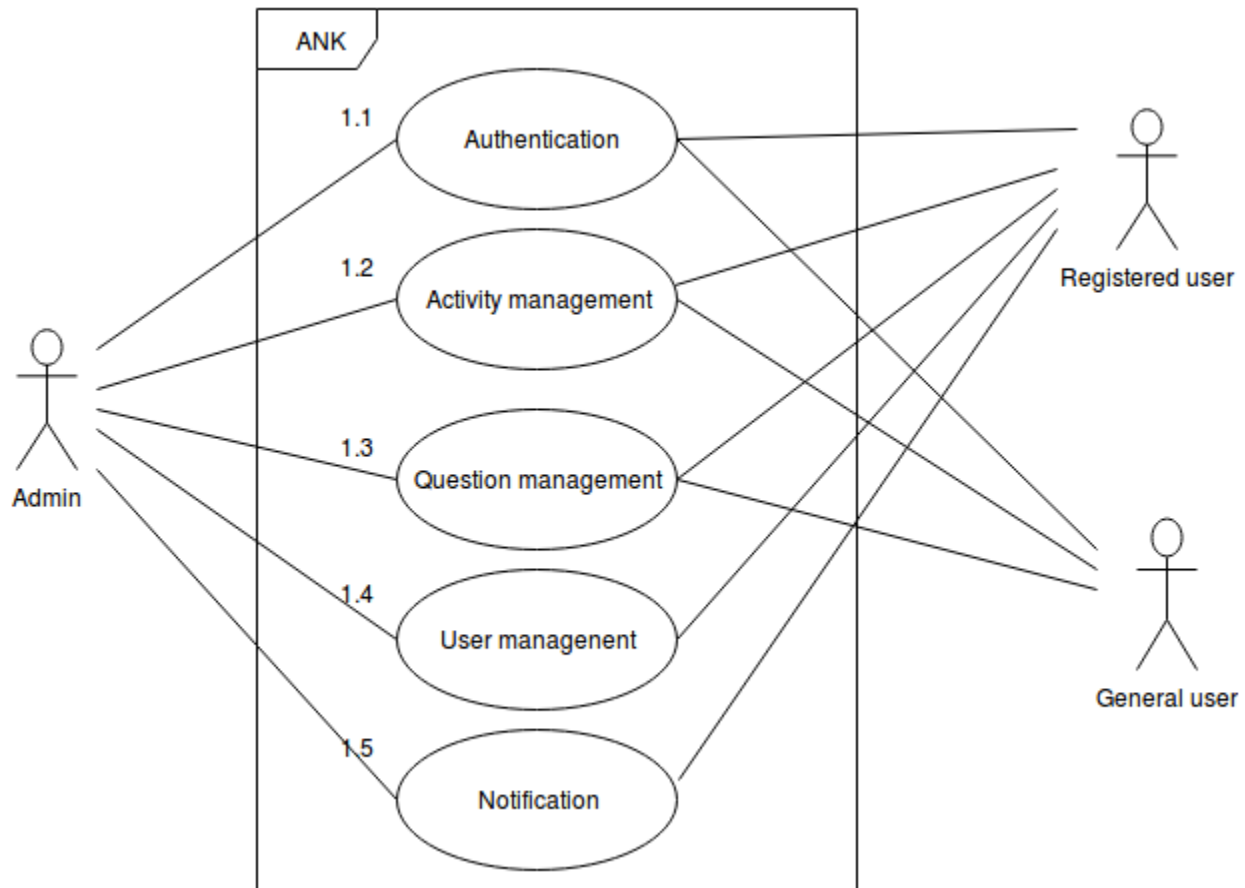


Figure 2: Level 1 Use Case diagram-Subsystems

Description of Use Case Diagram Level 1:

There are five subsystems in the "Asknknow" (ANK). They are:

1. Authentication
2. Activity management
3. Question management
4. User management
5. Notification

The first four subsystems are further decomposed in level 1.1, 1.2, 1.3, 1.4 respectively. The description of notification system is given below.

In the "ANK" system there will be three types of notification. They are as follows:

When a new user account is created by a general user to be registered as an admin, a notification is sent to the existing admin panel for approval or rejection him/her as a new admin.

When a registered user ask a question submitting it to publish, a notification is sent to the admin to consider the question whether it is eligible to publish.

When a registered user submit a question of a specific domain, notification will be sent to all the other registered users who are expert at that domain.

4.3.3 LEVEL- 1.1 Use Case Diagram-Authentication

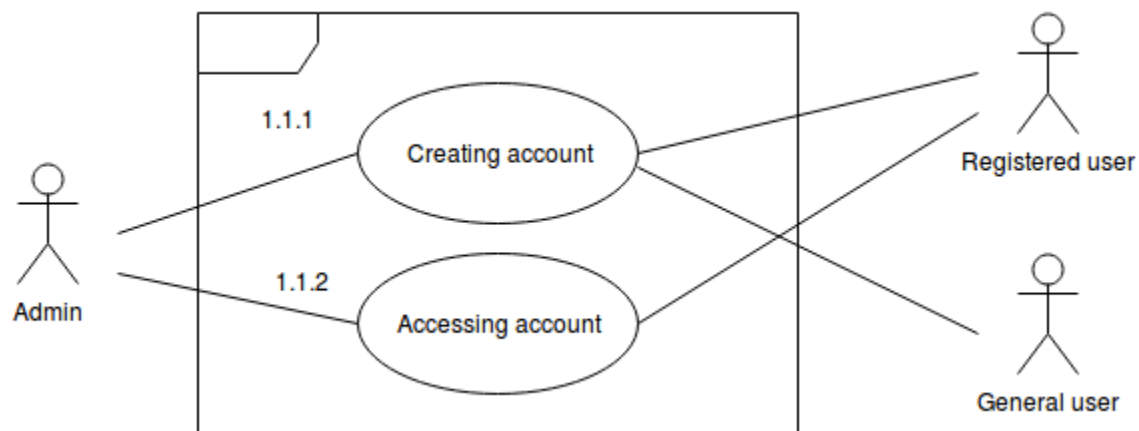


Figure 3: Level 1.1 Use Case diagram-Authentication

Description of level-1.1 use case diagram:

Authentication is a process in which credentials provided are compared to those on files in a database of authorized user's information. The authentication subsystem can be divided into two parts. They are as follows:

1. Creating Account
2. Accessing Account

4.3.4 LEVEL- 1.1.1 CREATING ACCOUNT

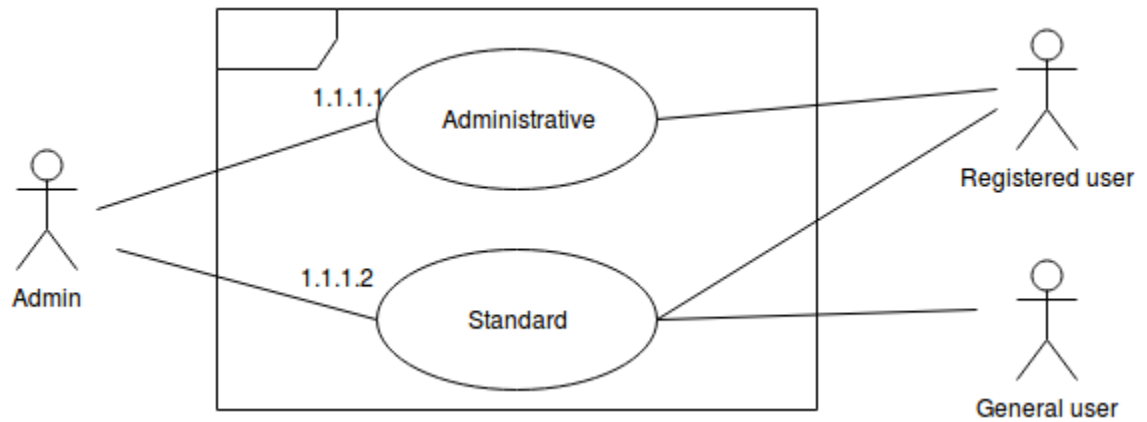


Figure 4: Level 1.1.1 Use Case diagram- Creating account

Description of level-1.1.1 use case diagram:

1. Primary Actors: Admin, Registered User
2. Secondary Actors: General User

4.3.5 LEVEL 1.1.2 USE CASE DIAGRAM-ACCESSING ACCOUNT

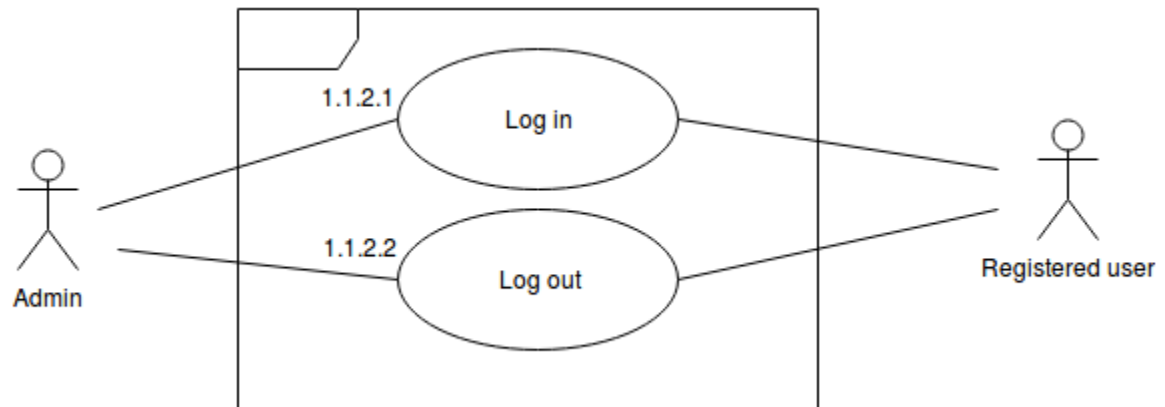


Figure 5: Level 1.1.2 Use Case diagram- Creating account

Description of level-1.1.2 use case diagram:

1. Primary Actors: Admin, Registered User
2. Secondary Actors: None

4.3.6 LEVEL-1.2 USE CASE DIAGRAM-ACTIVITY MANAGEMENT

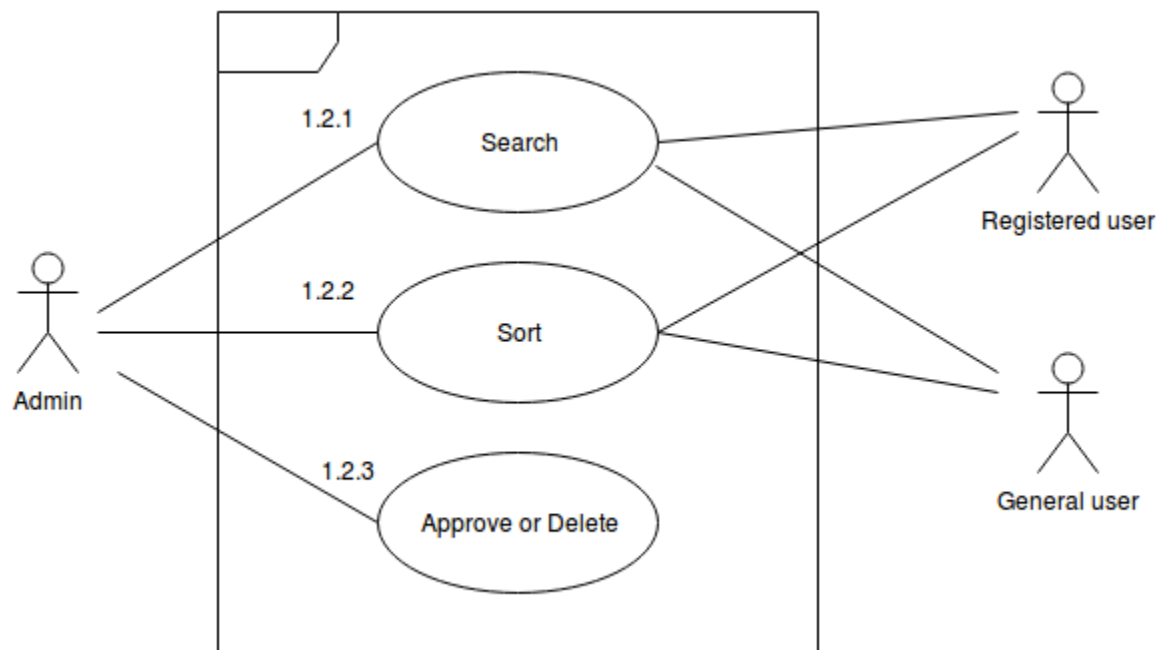


Figure 6: Level 1.2 Use Case diagram- Activity Management

Description of level-1.2 use case diagram:

1. Primary Actors: Admin, Registered User
2. Secondary Actors: General User

4.3.7 LEVEL-1.3 USE CASE DIAGRAM-QUESTION MANAGEMENT

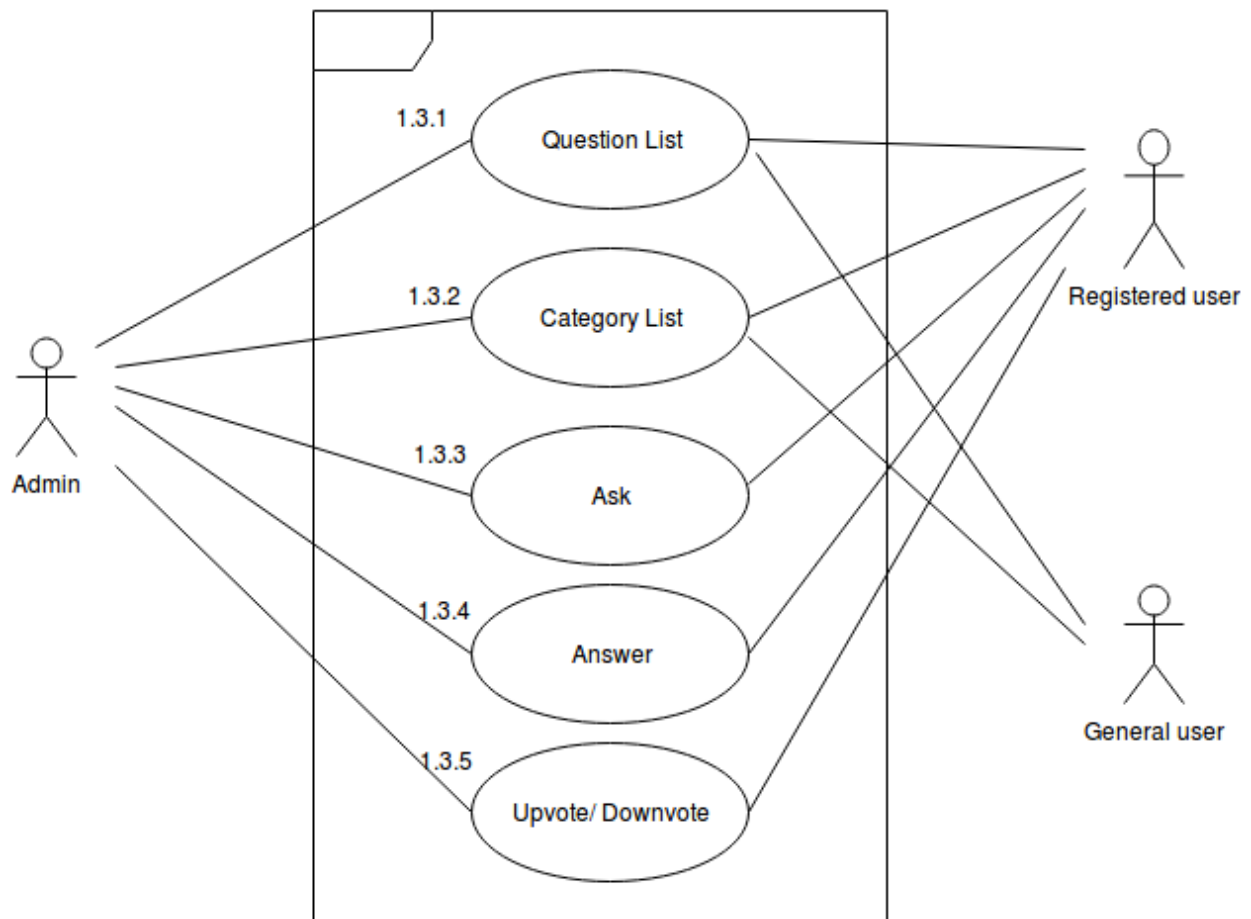


Figure 7: Level 1.3 Use Case diagram- Question Management

Description of level-1.3 use case diagram:

1. Primary Actors: Admin, Registered User
2. Secondary Actors: General User

4.3.8 LEVEL-1.4 USE CASE DIAGRAM-USER MANAGEMENT

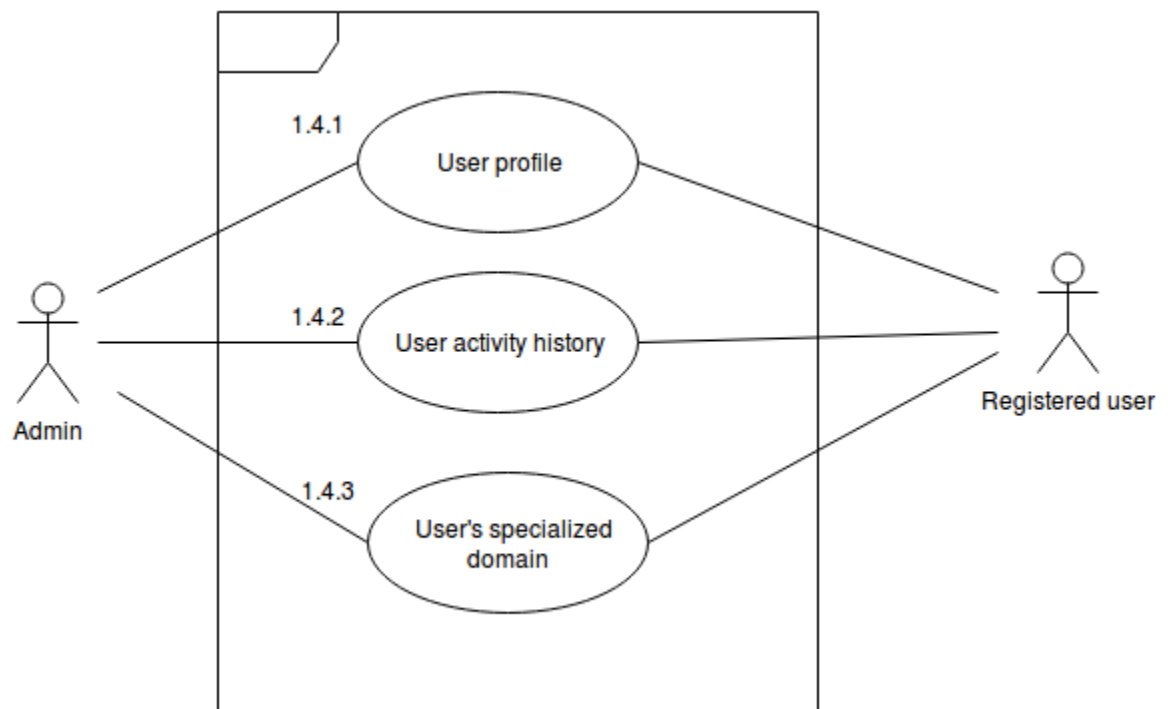


Figure 8: Level 1.4 Use Case diagram- User Management

4.4 ACTIVITY DIAGRAM OF ANK

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.

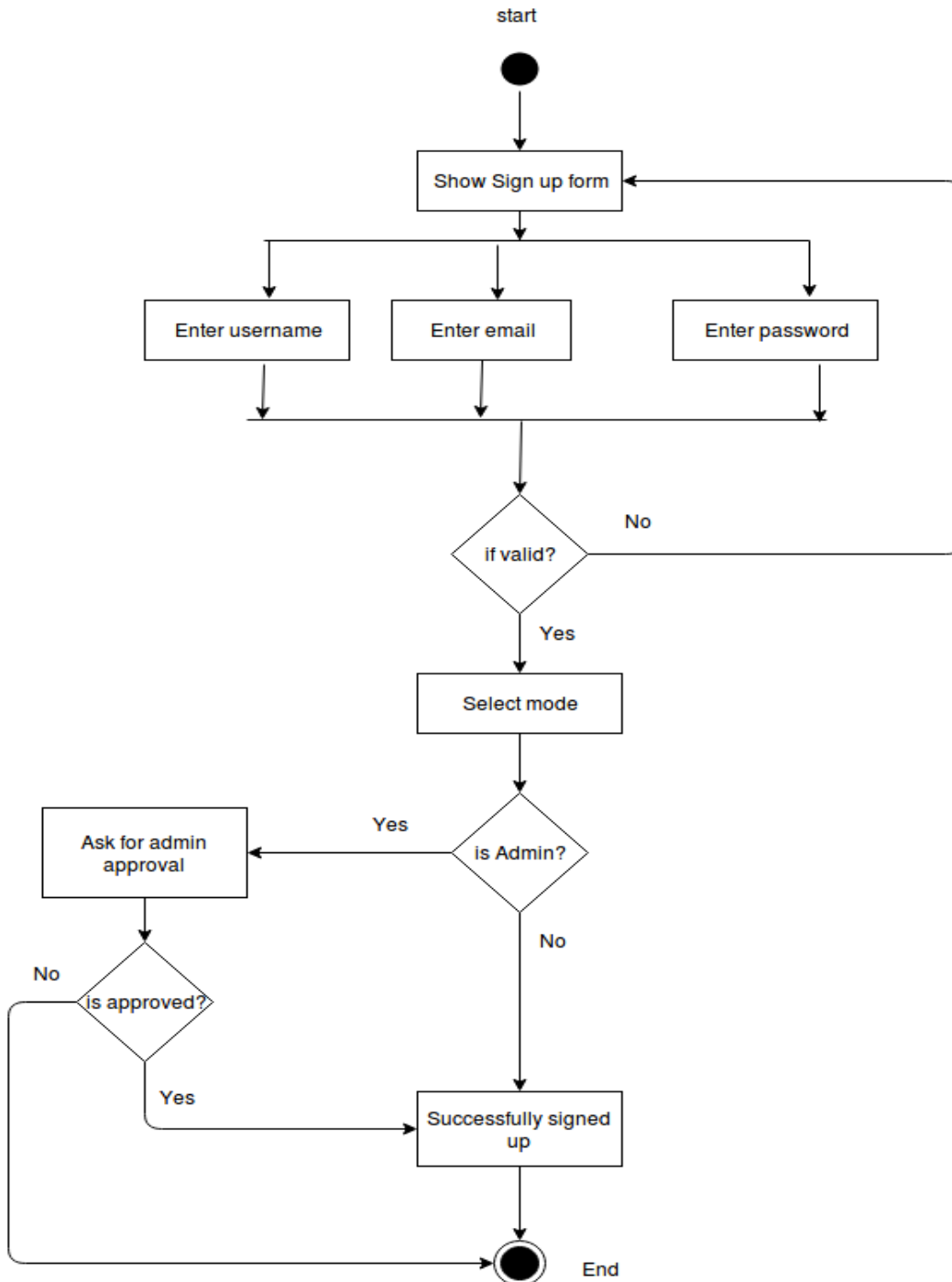


Figure 9: Activity Diagram- Creating Account

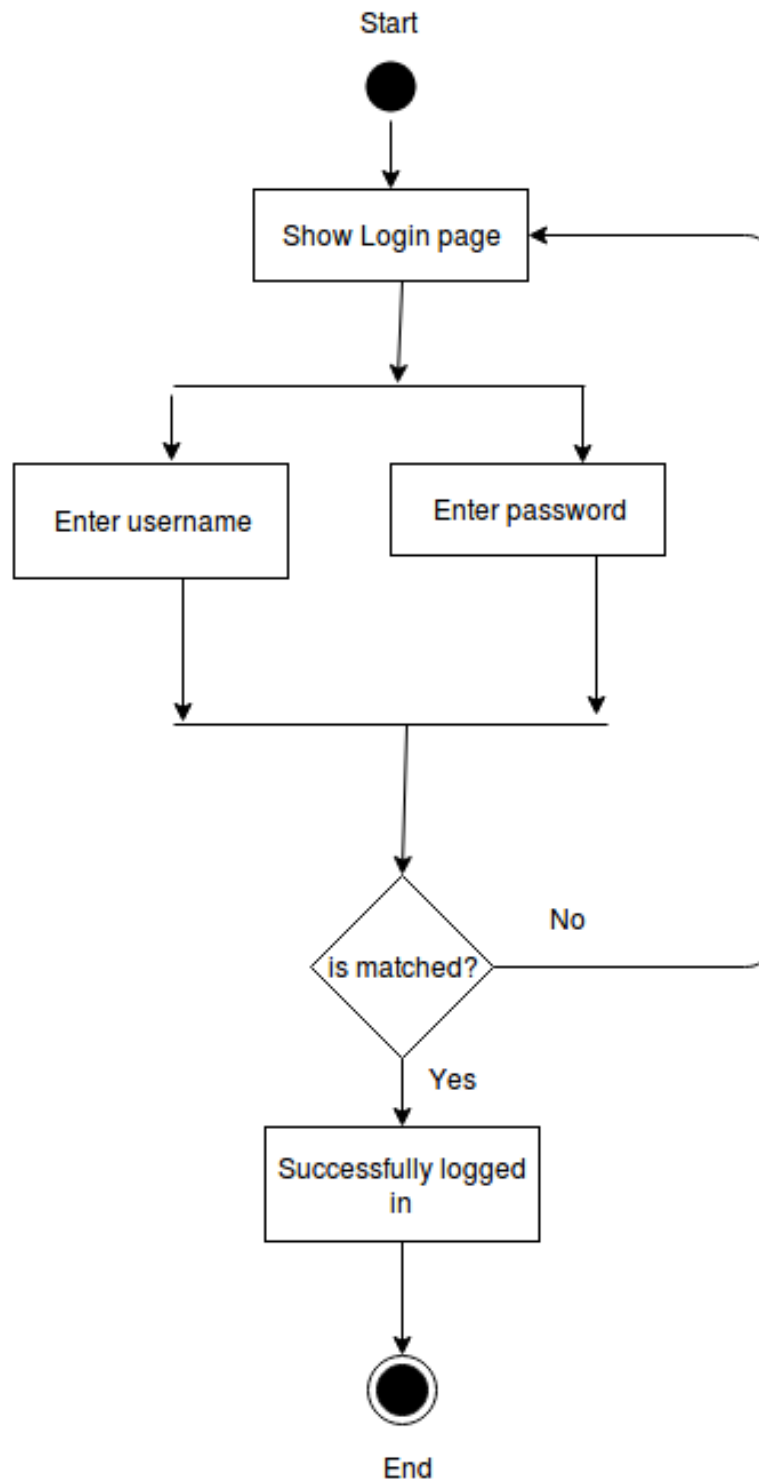


Figure 10: Activity Diagram – Accessing Account

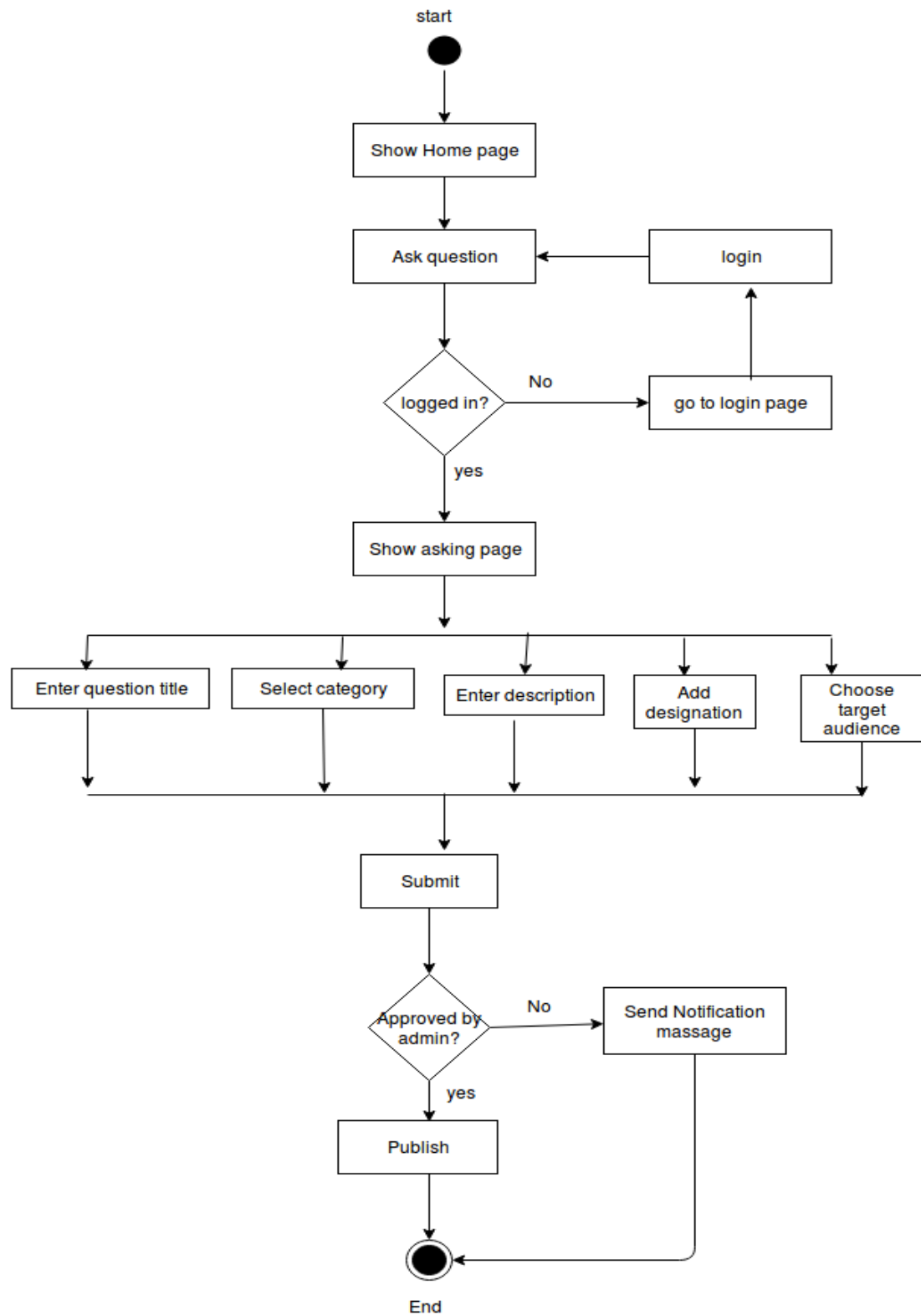


Figure 11: Activity Diagram- Ask

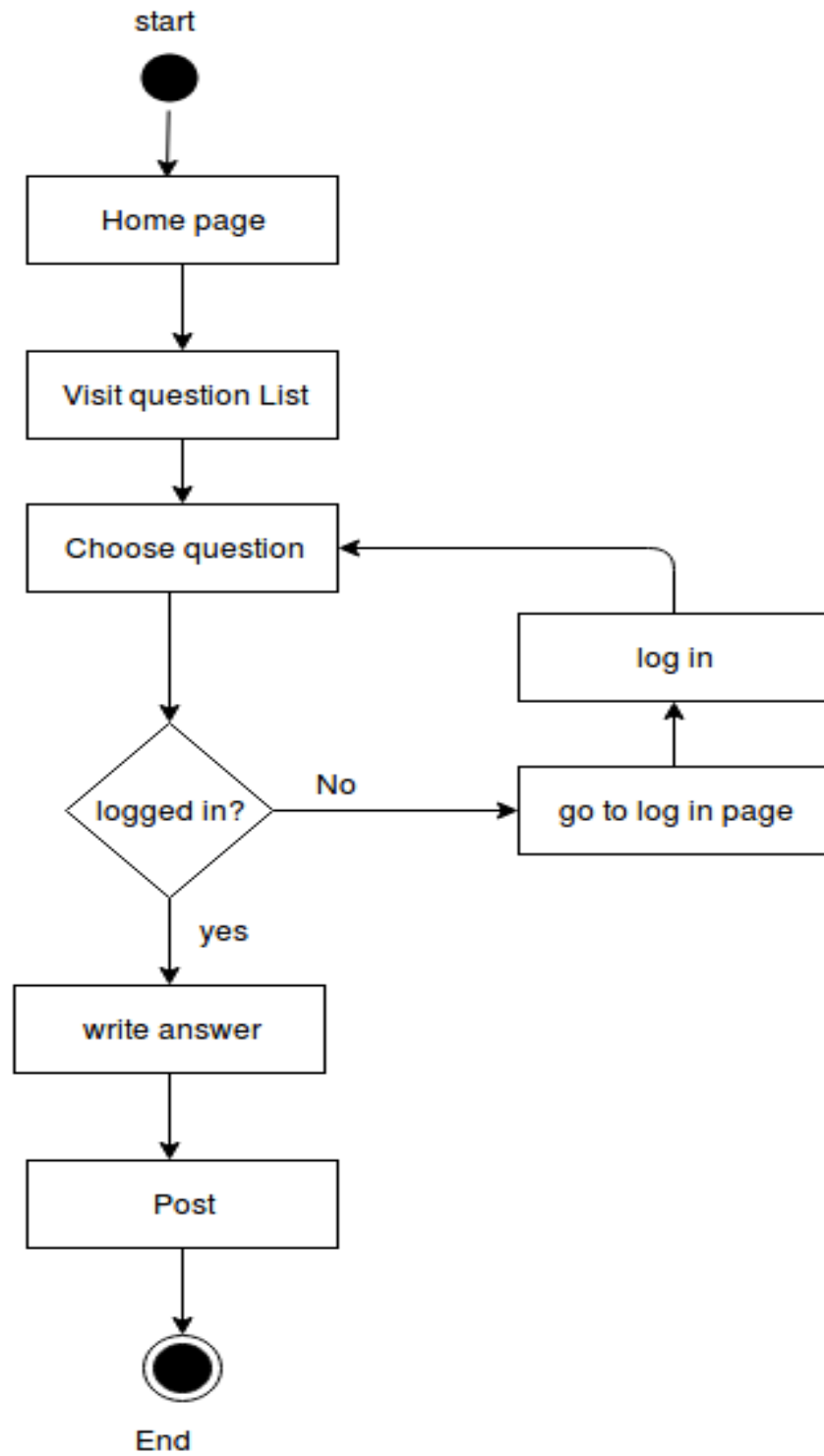


Figure 12: Activity Diagram- Answer

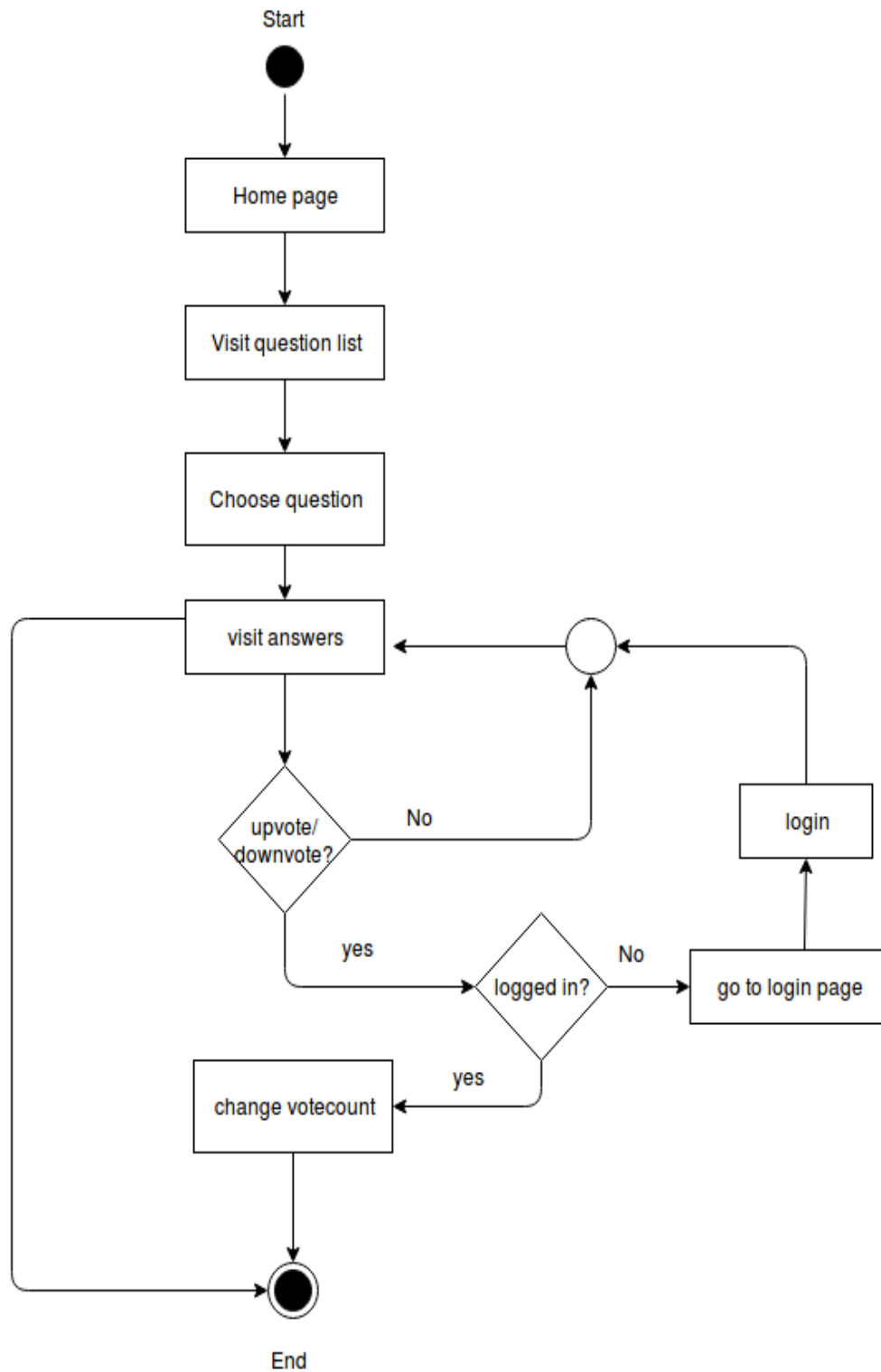


Figure 13: Activity Diagram- Downvote

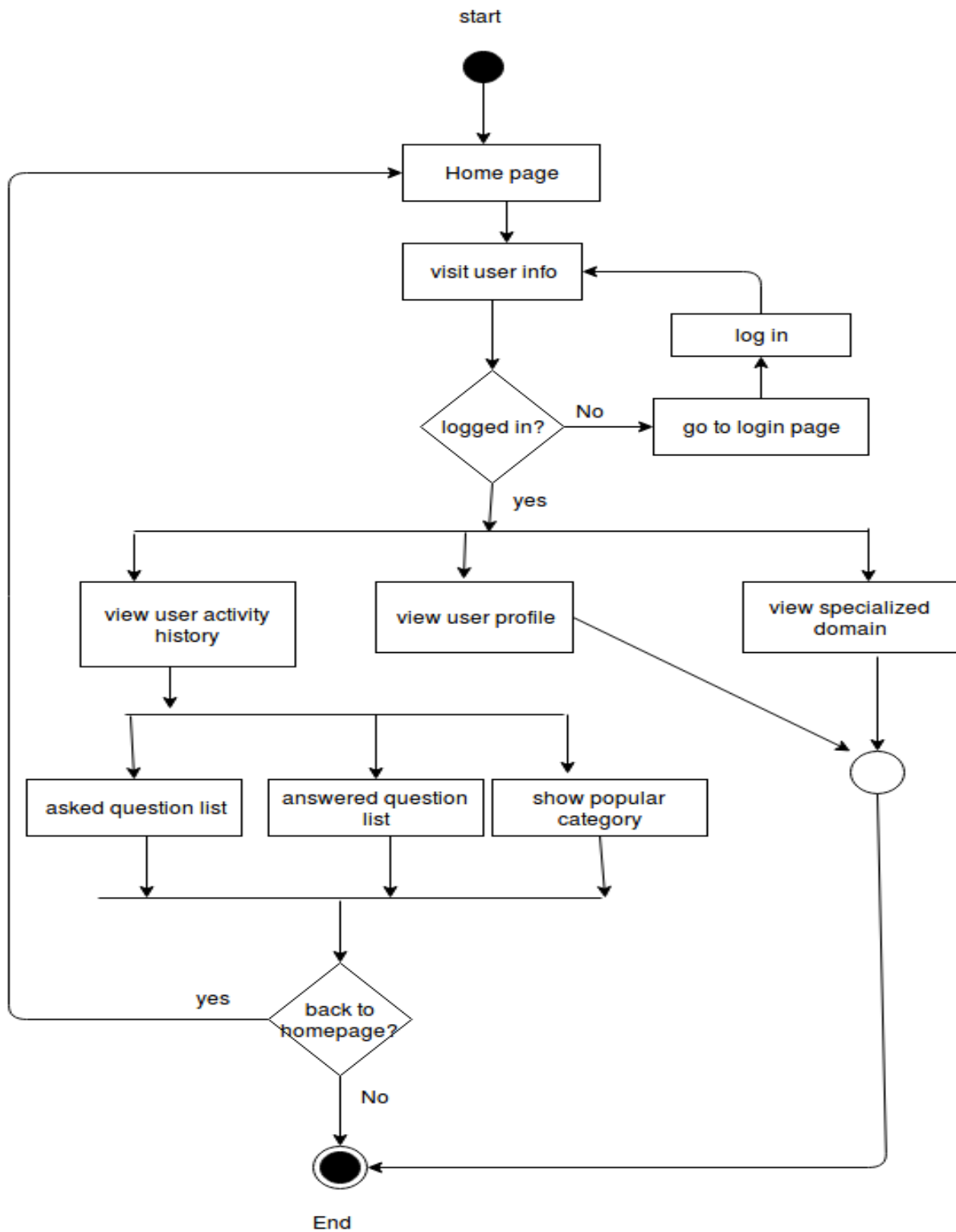


Figure 14: Activity Diagram- User management

4.5 SWIMLANE DIAGRAM OF ANK

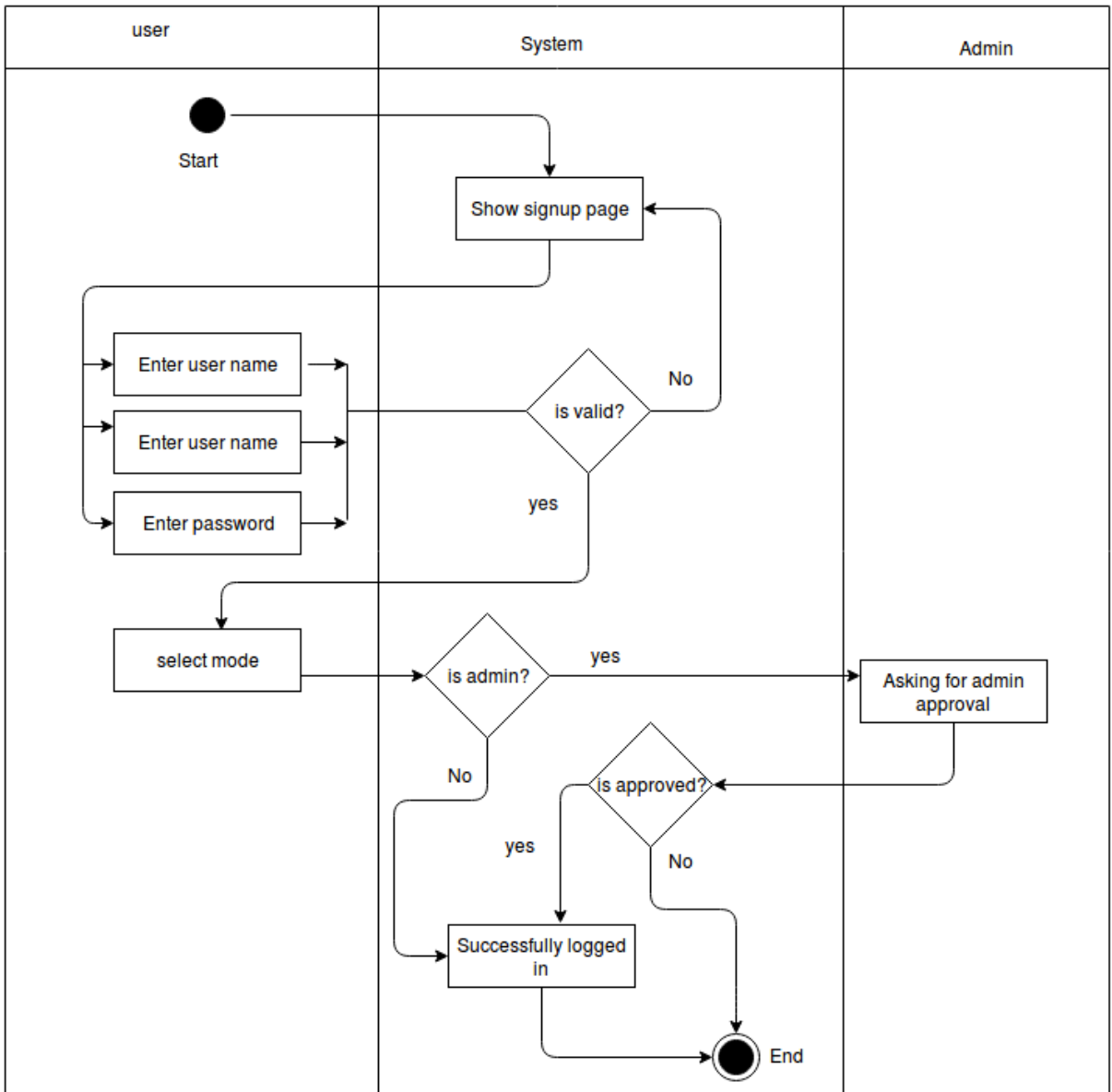


Figure 15: Swimlane Diagram- Creating Account

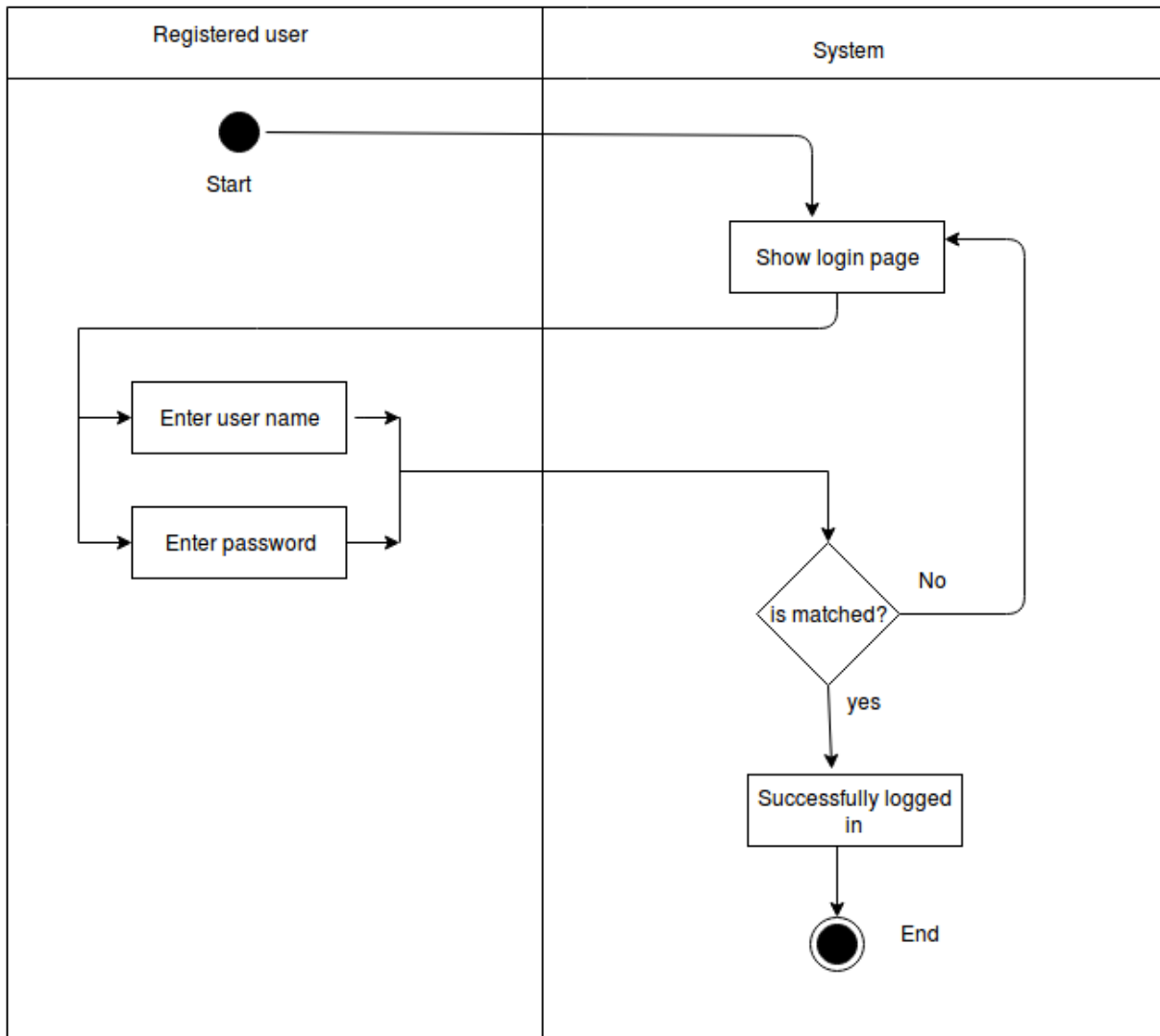


Figure 16: Swimlane Diagram- Accessing Account

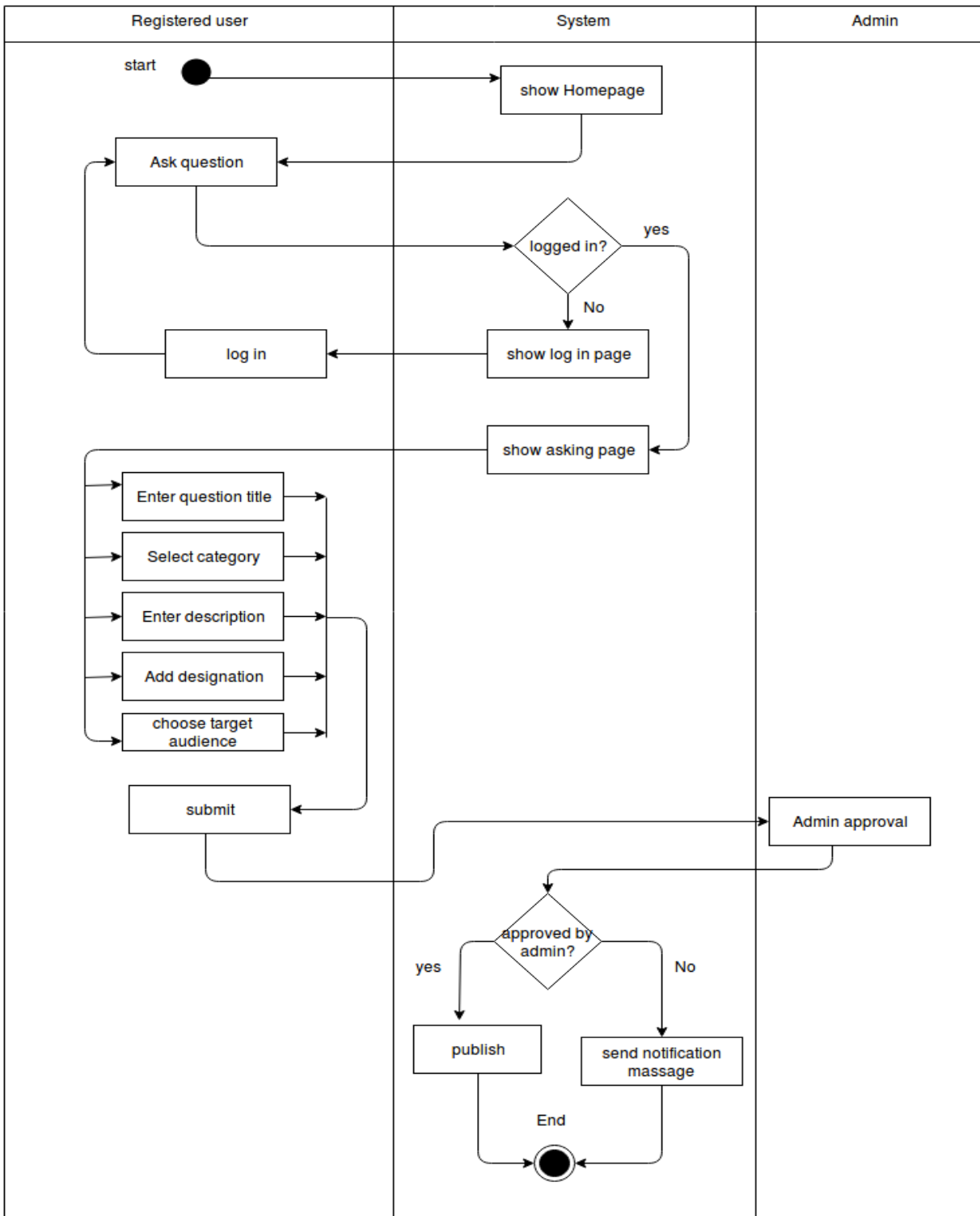


Figure 17: Swimlane Diagram- Ask

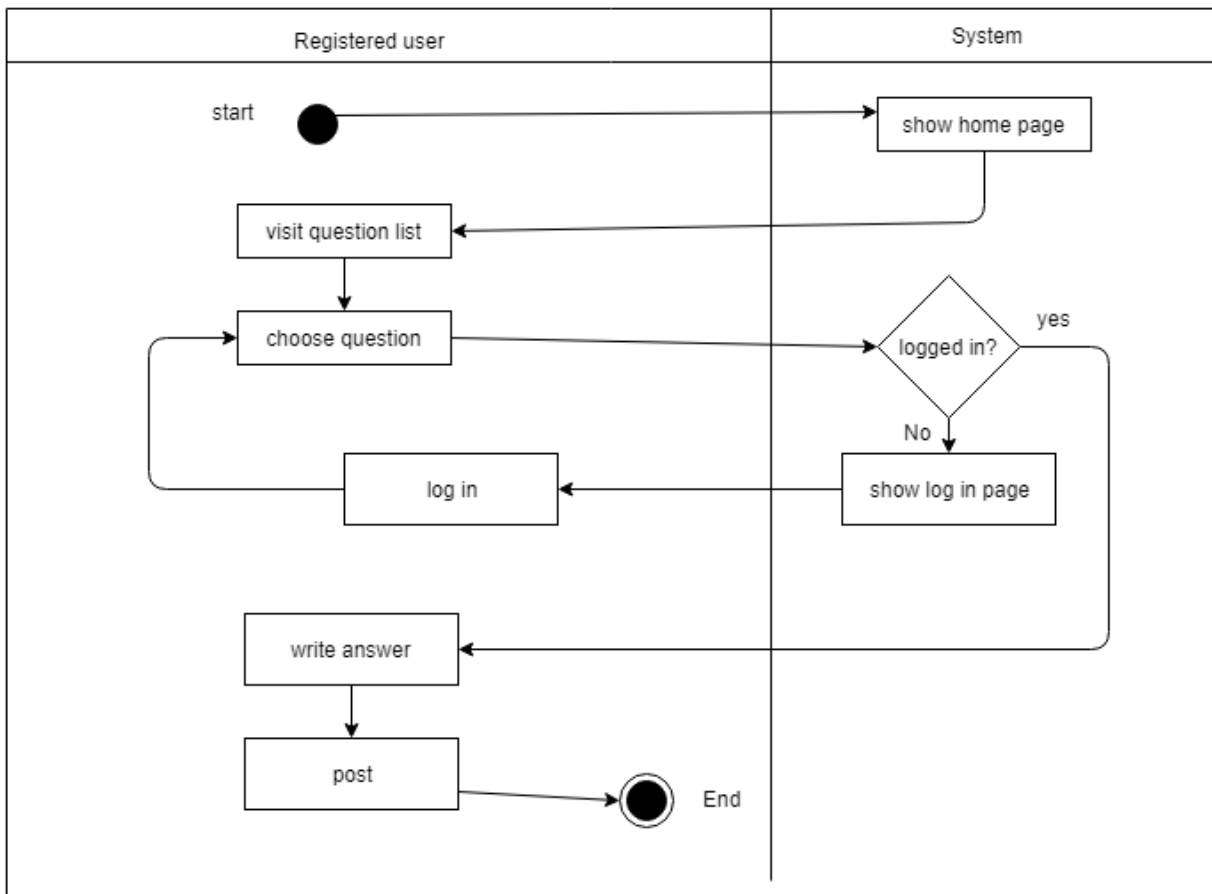


Figure 18: Swimlane Diagram- Answer

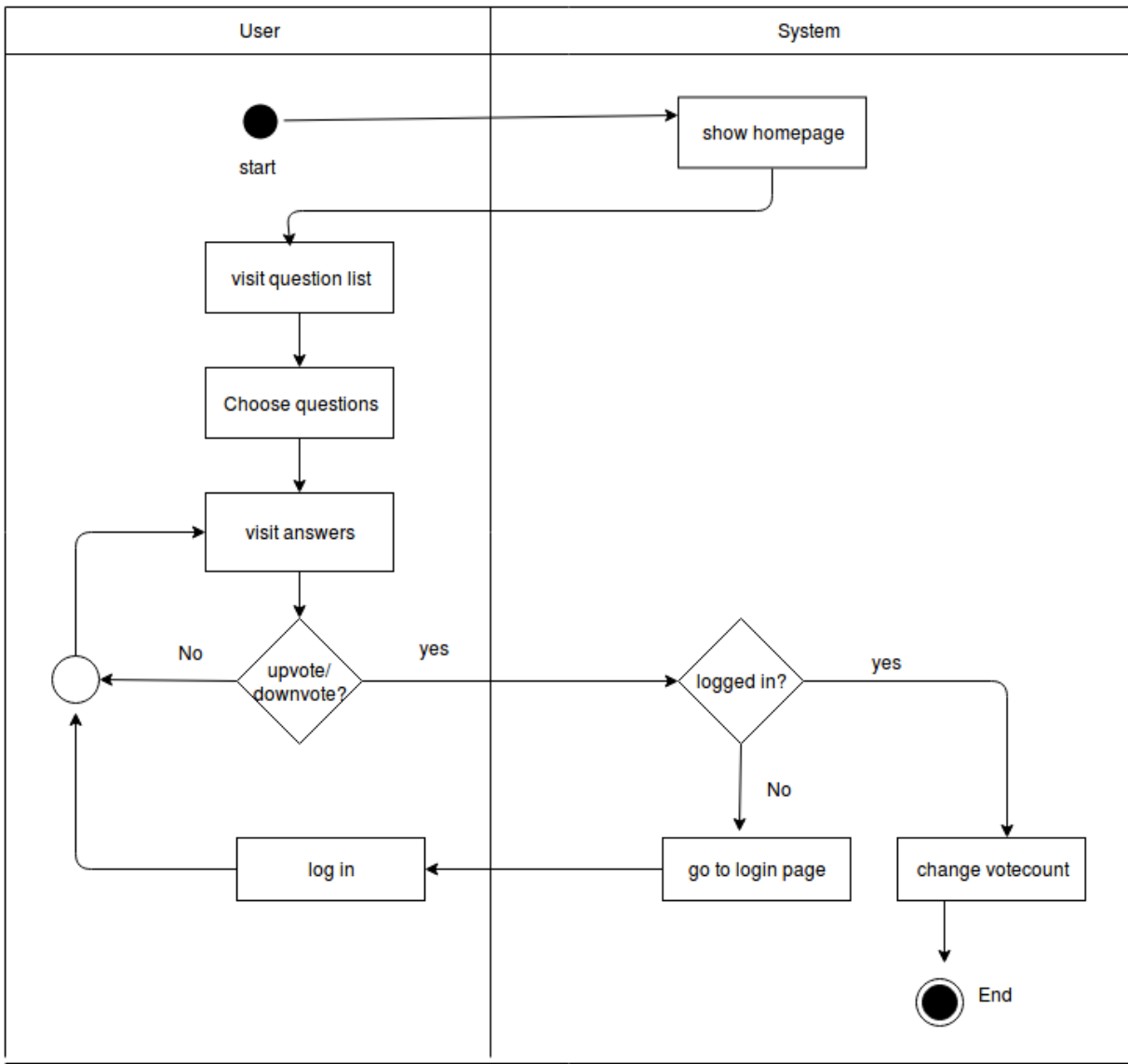


Figure 18: Swimlane Diagram- Upvote and Downvote

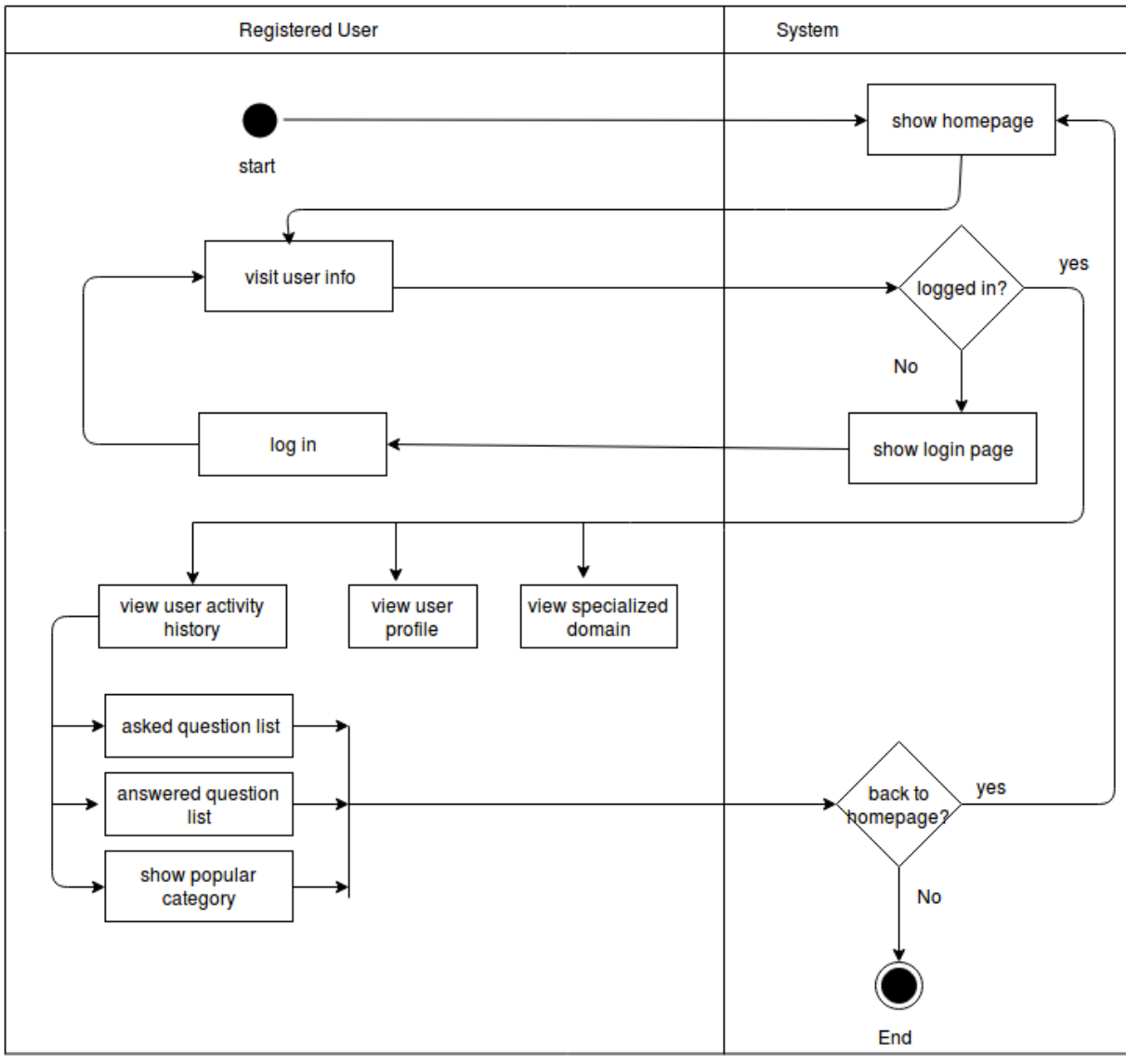


Figure 19: Swimlane Diagram- user management

CHAPTER 5: DATA-BASED MODELLING

5.1 Grammatical Parsing and Analysis

SL No.	Nouns	P/S	Attributes
1	Website	P	
2	Platform	P	
3	Registration	P	
4	Authentication	P	
5	Application	P	
6	User	S	7-13,29
7	Username	S	
8	Designation	S	
9	date of birth	S	
10	Address	S	
11	phone number	S	
12	email_id	S	
13	Password	S	
14	System	P	
15	Question	S	17,18,21-26,29,30
16	Answer	S	18,21,22,25,29,,31
17	Tag	S	

18	Content	S	
19	Comment	S	18,21,22,25,29,30
20	Reply	S	18,21,22,25,29,31
21	Upvote	S	
22	Downvote	S	
23	view-count	S	
24	time of posting	S	
25	Vote	P	
26	answer status	S	
27	ranking system	P	
28	Contribution	P	
29	user_id	S	
30	question_id	S	
31	comment_id	S	
32	reply_id	S	

Table 1 : Noun parsing table

5.2 Potential Data Objects

1. **User:** u_id, username, designation, date of birth, address, phone number, email_id, password
2. **Question:** question_id, u_id, content, tag, upvote, downvote, view-count, time of posting, answer status
3. **Answer:** comment_id, u_id, question_id, content, reply, upvote, downvote
4. **Comment:** comment_id, u_id, question_id, content, reply, upvote, downvote
5. **Reply:** reply_id, u_id, comment_id, content, upvote, downvote, vote

5.3 Analysis of Potential Data Objects

1. **Answer** and **Comment** have the same attributes so they can be merged to create **Comment** object.
1. **Question**, **Comment** and **Reply** object have attributes: upvote, downvote and vote. The all three of them have same meaning, so they can be merged. We will refer these as 'vote'.

5.4 Final Data Objects

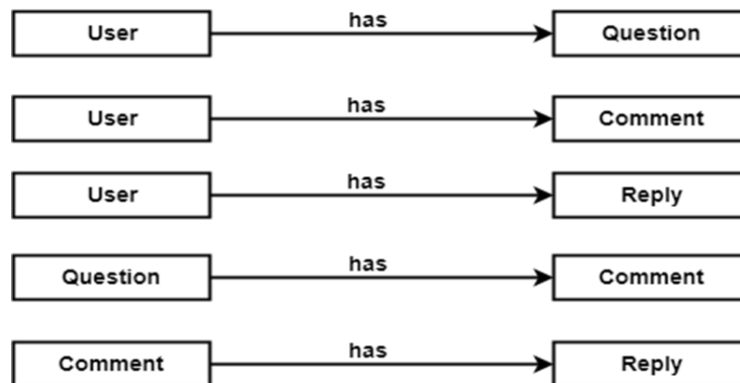
The attributes of the final data objects are shown-

No.	Entity	Attributes
1	User	u_id, username, designation, date of birth, address, phone number, email_id, password
2	Question	question_id, u_id, content, tag, view-count, time of posting, answer status
3	Comment	comment_id, u_id, question_id, content
4	Reply	reply_id, u_id, comment_id, content

Table 2 : Final data object tableTable 3

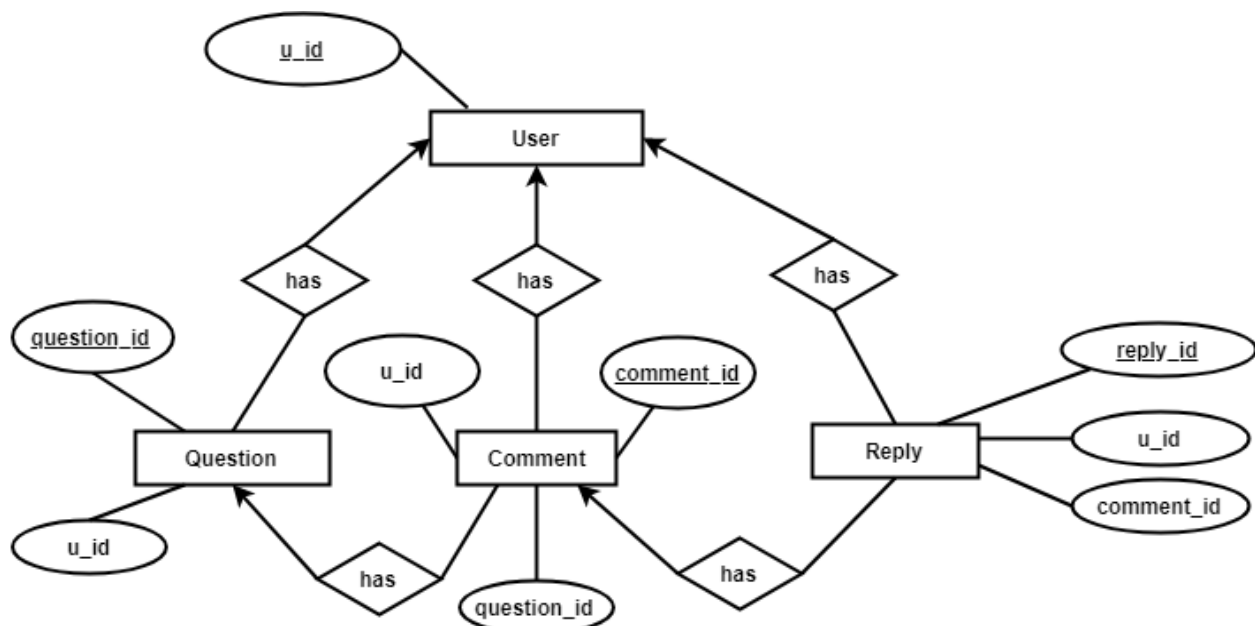
5.5 Data Object Relationship

The relationship among the data objects are shown-



5.6 Entity Relation Diagram

An entity-relationship diagram (ER Diagram) is a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities. We have shown the relationship among the entities through this model.



5.7 Schema Tables

1. User		
Attributes	Types	Size
<u>u_id</u>	NUMBER	10
username	VARCHAR2	30
designation	VARCHAR2	15
date of birth	DATETIME	15
address	VARCHAR2	90
phone number	VARCHAR2	15
email_id	VARCHAR2	15
password	VARCHAR2	15

2. Question		
Attributes	Types	Size
question_id	NUMBER	10
u_id	NUMBER	10
content	VARCHAR2	5000
tag	VARCHAR2	30
view-count	NUMBER	10
upvote	NUMBER	10
downvote	NUMBER	10
time of posting	DATETIME	10
answer status	BOOLEAN	10

3. Comment		
Attributes	Types	Size

comment_id	NUMBER	10
u_id	NUMBER	10
question_id	NUMBER	10
content	VARCHAR2	5000
upvote	NUMBER	10
downvote	NUMBER	10

4. Reply		
Attributes	Types	Size
reply_id	NUMBER	10
u_id	NUMBER	10
comment_id	NUMBER	10
content	VARCHAR2	5000
upvote	NUMBER	10
downvote	NUMBER	10

CHAPTER 6: CLASS BASED MODELLING

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

6.1 Grammatical Parsing and Analysis

To identify our analysis class, firstly, we grammatically parsed all the nouns and then categorized them according to general classification and selection criteria. We identified potential class by identifying the nouns from the scenery. Then we compared those with the following criteria whether they matched or not. We noted down the number of the fulfilled criteria at the right column.

6.1.1 Class Identification with General Classification

1. External entities
2. Things
3. Occurrence or events
4. Roles
5. Organizational unit
6. Places
7. Structure

SL No.	Potential Class	P/S	General Classification (GC)
1	website	P	

2	platform	P	
3	registration	S	2,3,4
4	authentication	S	2,3,4
5	application	P	
6	user	S	2,4,5
7	username	S	2
8	designation	S	2
9	date of birth	S	2
10	address	S	2
11	phone number	S	2
12	email_id	S	2
13	password	S	2
14	system	S	2,4,5
15	question	S	2,3,5,7
16	answer	S	2,3,5,7
17	tag	S	2
18	content	S	2
19	comment	S	2,3,5,7
20	reply	S	2,3,5,7
21	upvote	S	3
22	downvote	S	3
23	view-count	S	2
24	time of posting	S	2
25	vote	S	3
26	answer status	S	2

27	ranking system	P	
28	contribution	P	
29	user_id	S	2
30	question_id	S	2
31	comment_id	S	2
32	reply_id	S	2

Table 4 : General Criteria table

6.1.2. Class Identified With Selection Criteria

The nouns having two or more than two were selected from the general classification list. After that step, we compared them with the following criteria list. Those are-

1. Retained information
2. Needed services
3. Multiple attributes
4. Common attributes
5. Common operations
6. Essential requirements

SL No	Potential Class	Special Classification (SC)	
		Accepted	Rejected
1	Registration	2,6	1,3,4,5
2	Authentication	2,6	1,3,4,5
3	User	1-6	
4	Question	1-6	

5	Answer	1-6	
6	Comment	1-6	
7	Reply	1-6	
8	System	2,6	1,3,4,5

Table 5 : Selection criteria table

6.2 Preliminary Classes

From above table, we have taken the nouns who passed three or more accepted criteria. So there are fourteen candidate classes who are selected primarily. Those are-

1. User
2. Registration
3. Authentication
4. Question
5. Answer
6. Comment
7. Reply
8. System

6.3 Verb Identification

We have identified the verbs from our scenery to find out the necessary methods for the classes. In the following table, “P” stands for problem space and “S” stands for solution space. The possibility of verbs in the solution space have higher possibilities to become methods of the class.

SL No	Verbs	Remarks
1	Sign up	S
2	Provide info	S
3	Check username	S

4	Create account	S
5	Log in	S
6	Perform task	S
7	Log out	S
8	Ask Question	S
10	Edit Question	S
11	Delete Question	S
12	Add Tag	S
13	Edit Tag	S
14	See	P
15	Add Comment	S
16	Edit Comment	S
17	Delete Comment	S
18	Upvote	S
19	Downvote	S
20	Add Reply	S
21	Edit Reply	S
22	Delete Reply	S
23	Search Question	S
24	Sort	S

Table 6 : Verb parsing table

6.4 Attributes and Methods of Preliminary Classes

Analyzing the above table, we have categorized the verbs and convert them into method names. We put them to their respective classes and showed them in the following table-

SL No	Preliminary Class	Attributes	methods()
1	Registration	u-id + username + designation + date_of_birth + address + phone_number + email_id + password	checkIfUserNameAvailable()+ takeInput()+ signUp()+ profileCreation()
2	Authentication	username+ password	takeInput()+ checkCredentials() + logIn()
3	Question	question_id+ u_id+ content+ tag+ view_count+ time_of_posting+ answer_status+ vote	askQuestion()+ editQuestion()+ deleteQuestion()+ addTag()+ editTag()+ deleteTag()+ manageQuestionVote()+ saveQuestion()
4	Answer	comment_id+ u_id+ question_id+ content +time_of_commenting+ vote	addAnswer()+ editAnswer()+ deleteAnswer()+ +manageAnswerVote()+ saveAnswer()
5	User	username+ password	performOperation()+ terminateAccount() + logout()
6	Comment	comment_id+ u_id+ question_id+ content +time_of_comment+ vote	addComment()+ editComment()+ deleteComment()+ +manageCommentVote()+ saveComment()
7	Reply	reply_id+ comment_id+ u_id+ content +time_of_replying + vote	addReply()+ editReply()+ deleteReply()+ +manageReplyVote()+ saveReply()
8	System	question_id	searchQuestion()+ rankingQuestion()+ calculateViewCount()+ calculateTimeOfPsoting()+ calculateVote()+ identifyAnswerStatus()

Table 7 : Attribute and method table

6.5 Analysis of Potential Classes

1. **Answer** and **Comment** have same attributes and methods since we are making it a common class name **Comment**.
2. **Registration** class can be merged with **Authentication** class to remove complexity.

6.6 Final Classes

From above analysis, our final classes are:

1. Authentication
2. Question
3. Comment
4. Reply
5. System
6. User

6.7 Attributes, Methods of Final Classes

Attributes, methods, responsibilities and collaborative classes of Authentication class are given below-

SL No	Final Class	Attributes	methods()
1	Authentication	u-id + username + designation + date_of_birth + address + phone_number + email_id + password	checkIfUserNameAvailable()+ takeInput()+ signUp()+ profileCreation()+ checkCredentials() + logIn()
2	Question	question_id+ u_id+ content+ tag+ view_count+ time_of_posting+ answer_status+ vote	askQuestion()+ editQuestion()+ deleteQuestion()+ addTag()+ editTag()+ deleteTag()+ manageQuestionVote()+ saveQuestion()
3	Comment	comment_id+ u_id+ question_id+ content +time_of_commenting+ vote	addComment()+ editComment()+ deleteComment()+ +manageCommentVote()+ saveComment()

4	Reply	reply_id+ comment_id+ u_id+ content +time_of_replying+ vote	addReply()+ editReply()+ deleteReply()+ +manageReplyVote()+ saveReply()
5	User	username+ password	performOperation()+ terminateAccount() +logout()
6	System	question_id	searchQuestion()+ rankingQuestion()+ calculateViewCount()+ calculateTimeOfPsoting()+ calculateVote()+ identifyAnswerStatus()

6.7 Responsibilities and Collaborative Classes of Final Classes

1. Authentication	
Responsibilities	Collaborative Classes
Getting registered Getting authenticated	User User

2. Question	
Responsibilities	Collaborative Classes
Updating question Updating tag Updating vote	User System User, System

3. Comment	
Responsibilities	Collaboration

Updating comment Updating vote	User, Question User, System
-----------------------------------	--------------------------------

4. Reply	
Responsibilities	Collaboration
Updating reply Updating vote	User, Comment User, System

5. System	
Responsibilities	Collaboration
Searching question Providing ranking	System, Question System, Question

6. User	
Responsibilities	Collaboration
Performing operation Terminating account Logging out	Question, Comment, Reply System Authentication

6.8 Class Diagram

Class diagram is a diagram where dynamics of object interaction and collaboration are represented through UML diagrams and their networks. Here composition, association and inheritance of the classes are shown in the diagram. The notations are-

*

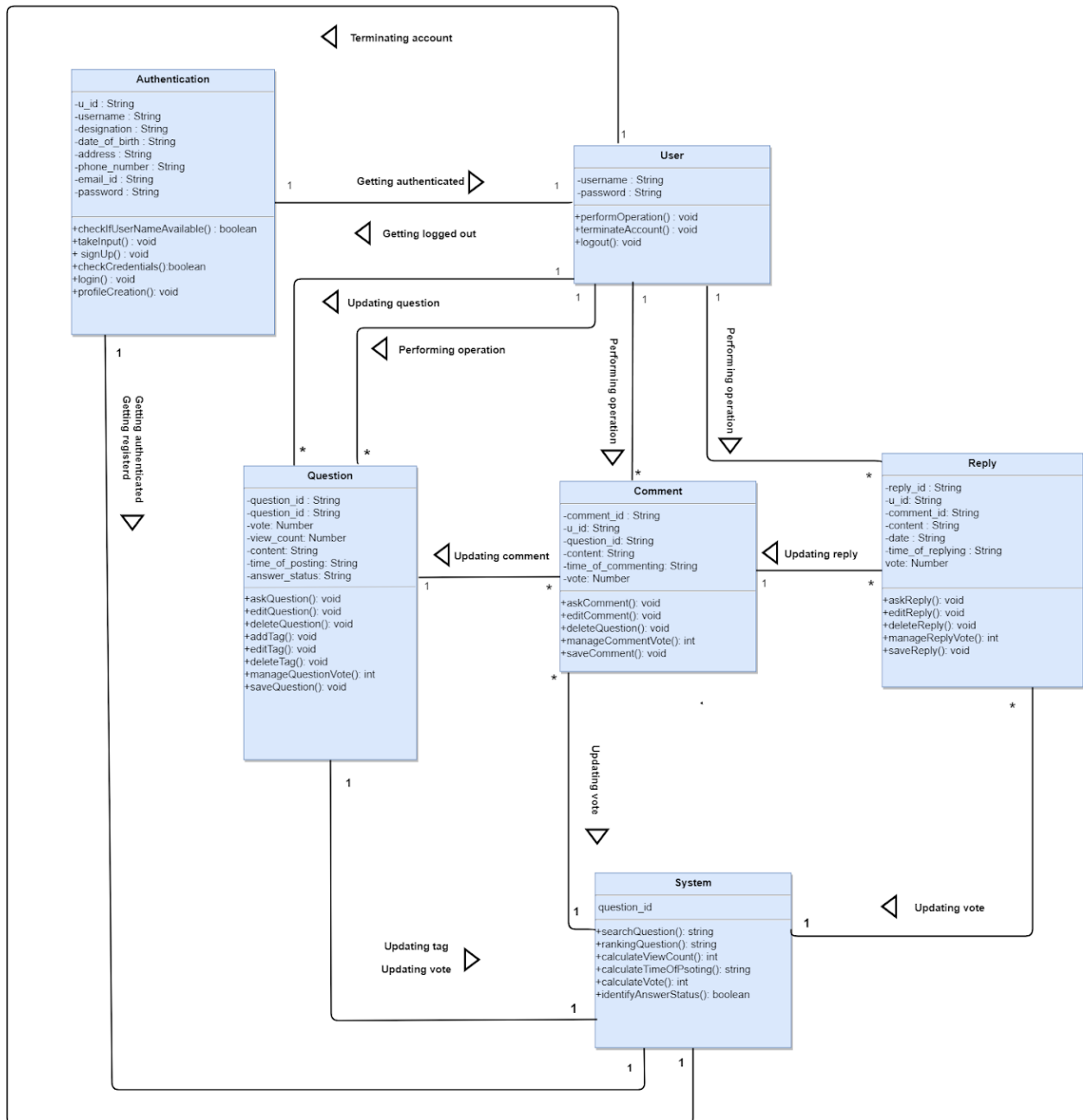
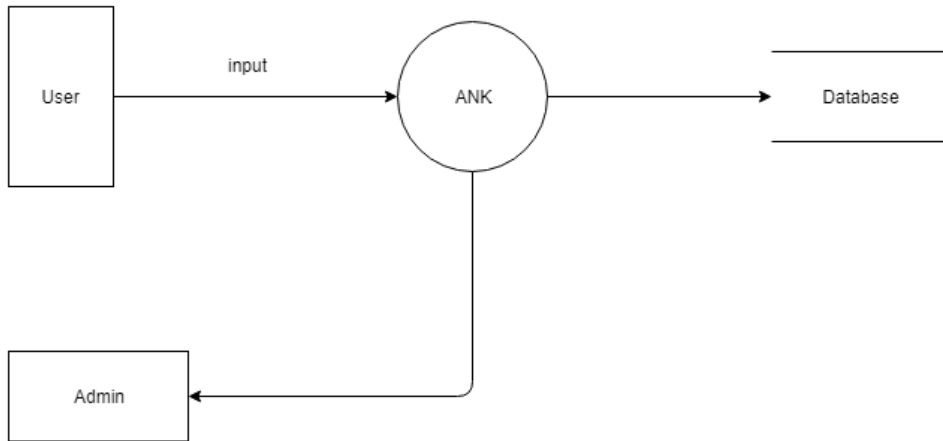


Figure 3: class diagram

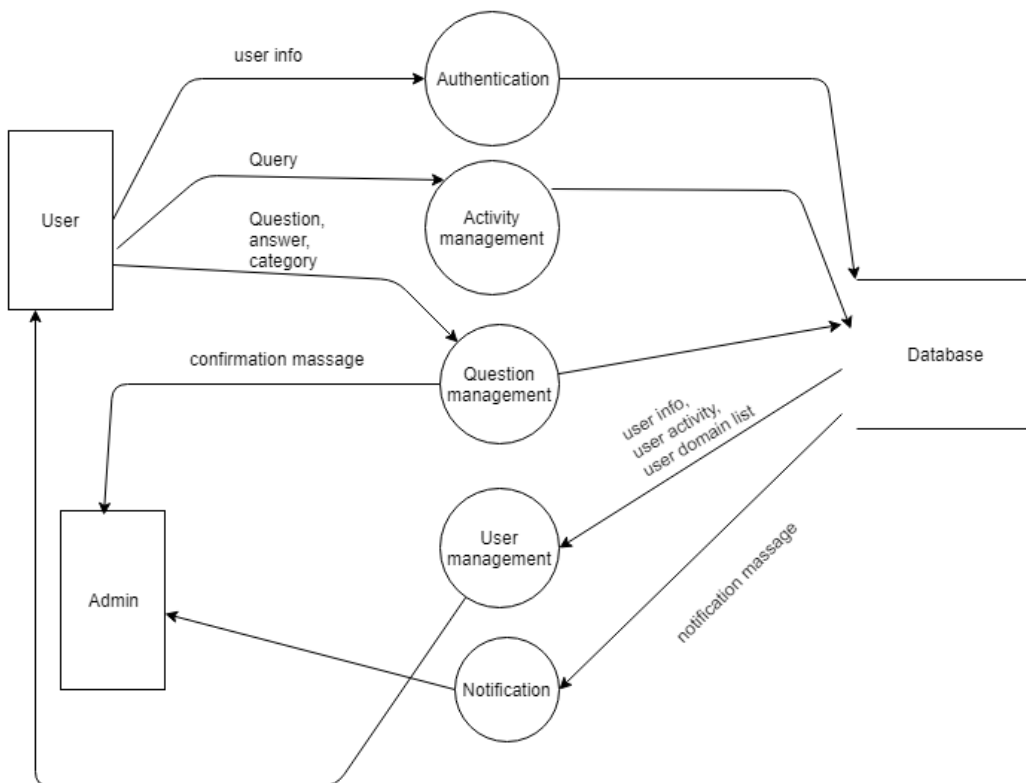
CHAPTER 7: FLOW-ORIENTED MODELING

7.1 Data flow diagram

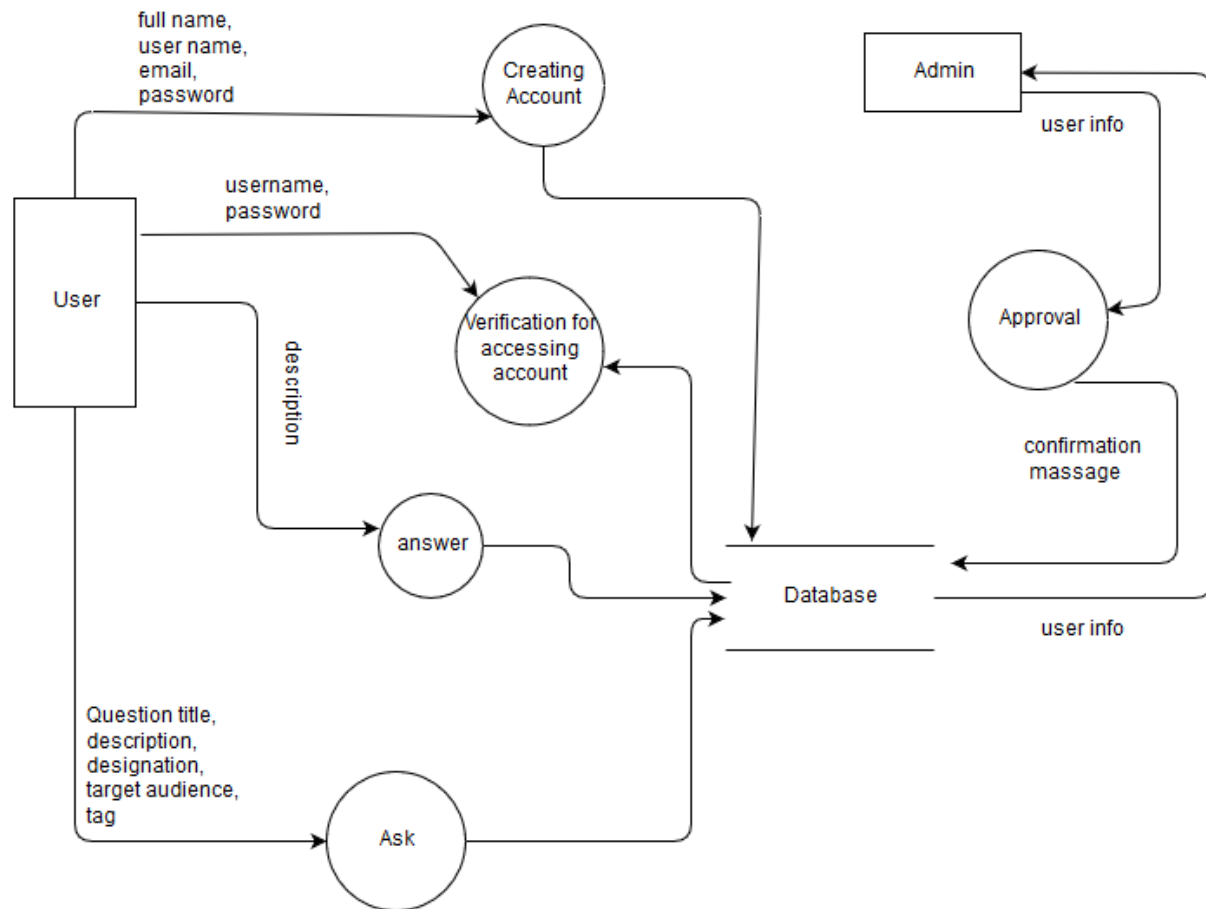
7.1.1 Level 0 Data Flow Diagram



7.1.2 Level 1 Data Flow Diagram



7.1.3 Level 2 Data Flow Diagram



CHAPTER 8: BEHAVIORAL MODELING

The behavioral model indicates how software will respond to external events. Two different behavioral representations are discussed in this chapter. The first indicates how individual class changes state based on external events and the second shows the behavior of the software as a function of time.

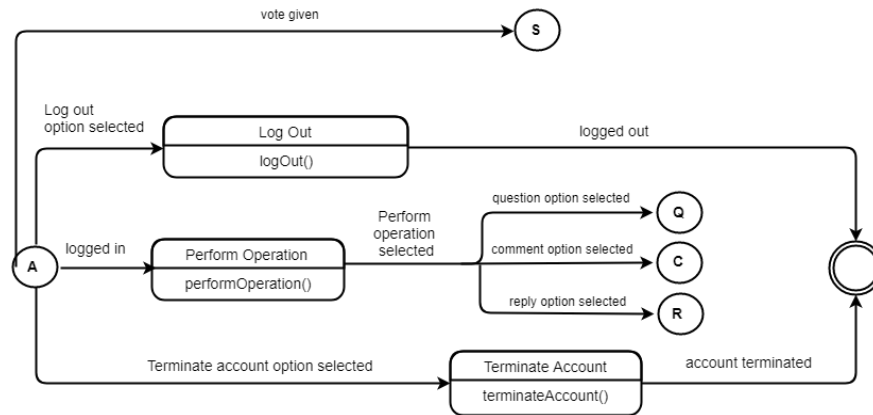
8.1 Event Identification

SL No	Event	Initiator	Collaborator
1.	Sign Up option selected	Authentication	
2.	Log in option selected	Authentication	
3.	Information for account is given	Authentication	
4.	Sending mail notification	Authentication	System
5.	User name and password given	Authentication	
6.	Username and password matched	Authentication	
7.	Username and password mismatched	Authentication	
8.	Username available	Authentication	
9.	Username not available	Authentication	
10..	Logged in	Authentication	User
11.	Logging out	User	
12.	Log out option selected	User	
13.	Perform operation selected	User	Question, Comment, Reply
14.	Terminate account option selected	User	
15.	Account Terminated	User	

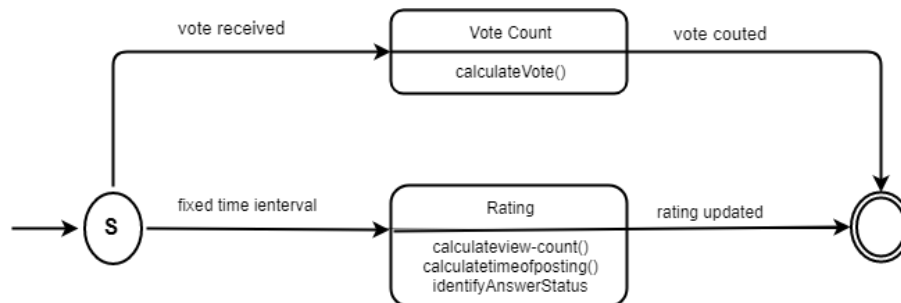
16.	vote given	User	System
17.	vote received	System	User
18.	Rating updated	System	Question
19.	vote counted	System	Question, Comment, Reply
20.	question added	Question	
21.	question edited	Question	
22.	question deleted	Question	
23.	tag updated	Question	System
24.	question saved	Question	System
25.	comment added	Comment	
26.	comment edited	Comment	
27.	comment deleted	Comment	
28.	comment saved	Comment	
29.	reply added	Reply	
30.	reply edited	Reply	
31.	reply deleted	Reply	
32.	reply saved	Reply	

8.2 State Transition

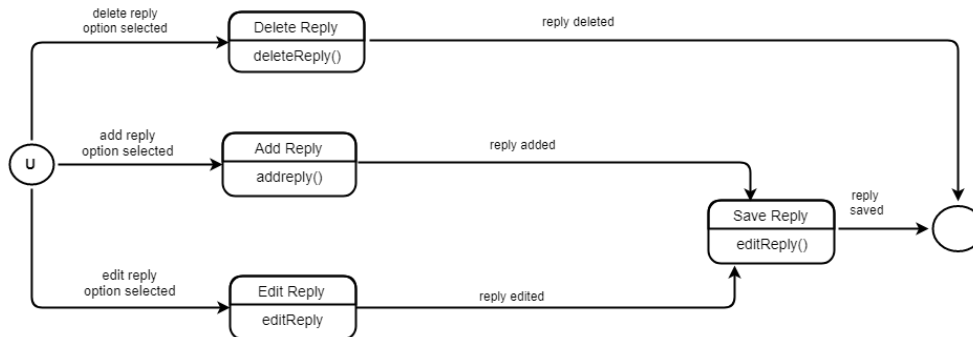
1. User:



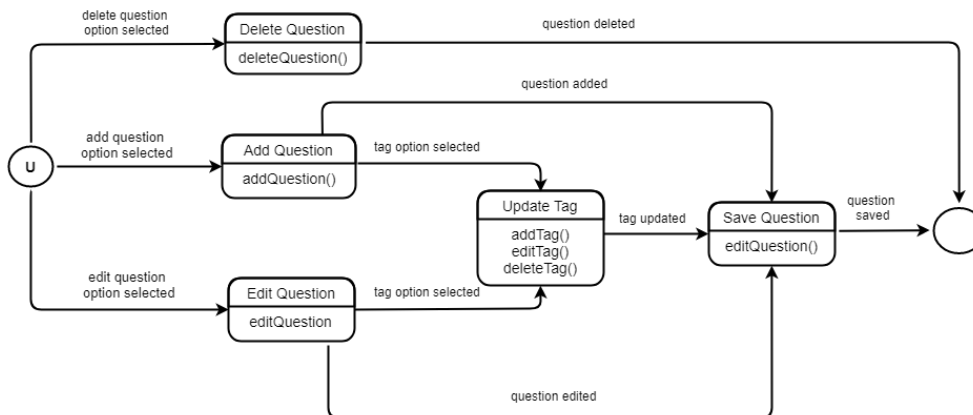
2. System



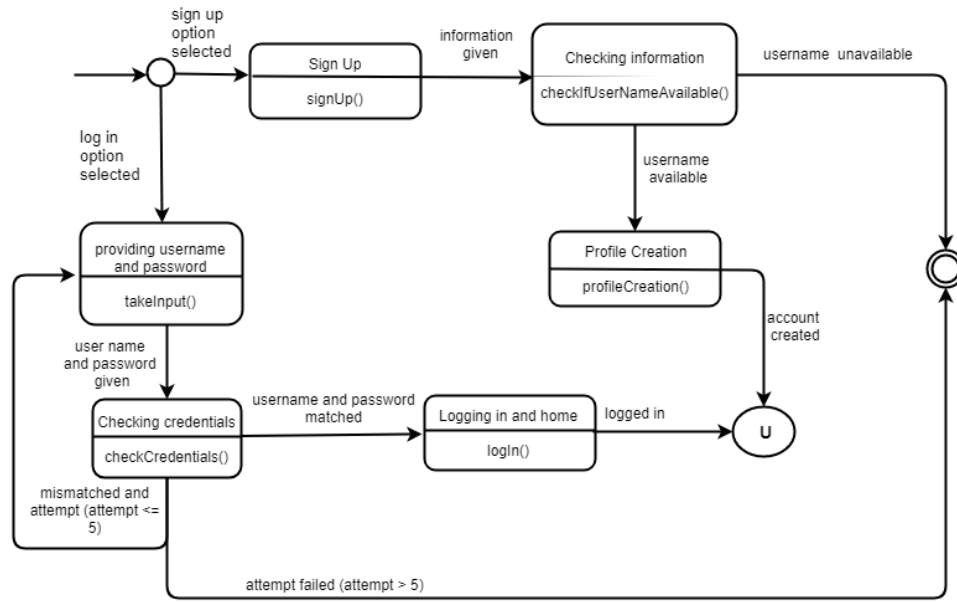
3. Reply



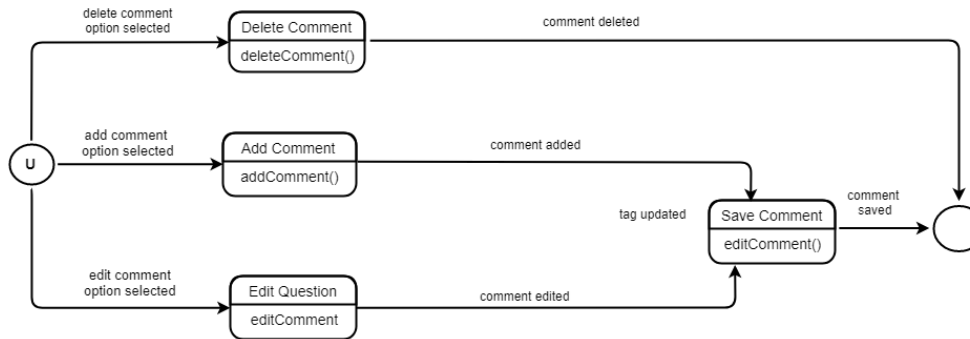
4. Question



5. Authentication

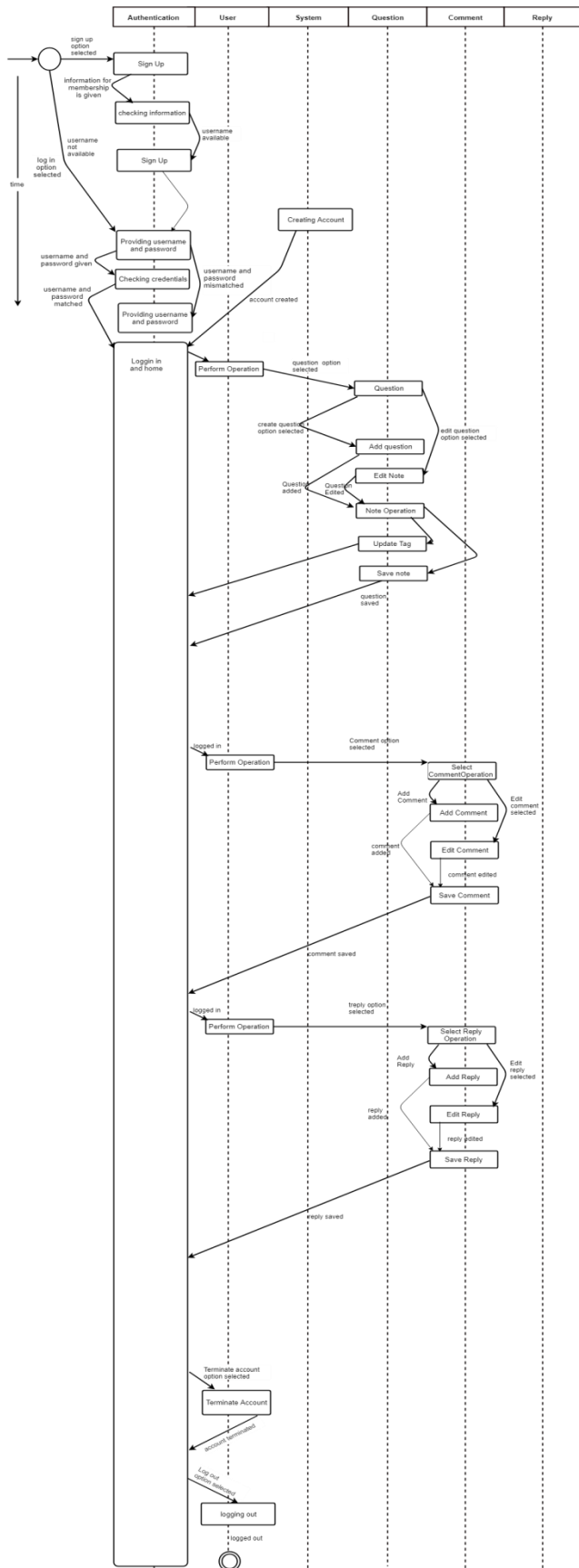


6. Comment



8.3 Sequence diagram

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence



decision taken