

Question-Answer Forum with
Recommendation System
Software Project Lab-3

17/12/2019

Submitted by

Tulshi Chandra Das-811

Supervised by

Md. Saeed Siddik

Assistant Professor, IIT, DU



QUESTION-ANSWER FORUM WITH RECOMMENDATION SYSTEM

SUBMITTED TO

SPL3 Committee

Institute of Information Technology

University of Dhaka

SUBMITTED BY

Tulshi Chandra Das

BSSE0811

LETTER OF TRANSMITTAL

17 December 2019

The Coordinators, SPL3

Institute of Information Technology

University of Dhaka.

Subject: Submission of term report on “Educational Question Answer Forum with Recommendation System”

Sir

I have prepared the report on my project “**Educational Question Answer Forum**”. This report includes the details of each steps I followed to collect the requirements.

The primary purpose of this report is to summarize my findings from the work that I completed.

Sincerely yours

Tulshi Chandra Das- BSSE0811

8th Semester, 8th Batch

Institute of Information Technology

University of Dhaka

LETTER OF ENDORSEMENT

17 December, 2019

To Whom It May Concern

Subject: Approval of the report

This letter is to clarify that all the information mentioned in this document is true. The project mentioned here have had successful involvement of Tulshi Chandra Das, BSSE 0811 from Institute of Information Technology, University of Dhaka.

I wish him all the best and hope that he will lead a successful career.

Project Supervisor

Md. Saeed Siddik

Assistant Professor

Institute of Information Technology

University of Dhaka

ACKNOWLEDGEMENT

I am highly indebted for getting such a tremendous opportunity to prepare the report on Question Answer Forum. I would like to thank my Supervisor, Md. Saeed Siddik, Assistant Professor, Institute of Information Technology, University of Dhaka, whole-heartedly, for giving me guidelines about how I can prepare this report and the coordinators of this course for giving guideline me. In completing this paper, I have collected data from some students and instructors of the university and college. I am thankful to all for the works cited.

ABSTRACT

The study is made for Educational Question Answer Forum. The scope of the study is to analyze on Question Answer Forum and design the SRS of this system. The object of this study is to develop an SRS (Software Requirements Specification and Analysis) and implementation overview of this project. This project is a web app with a recommendation system. The project has implemented the features of an educational Q/A forum. Here user can ask question on any study topic, give answer to a question, comment to question or answer and vote to question or answer. The recommendation system displays the recommended questions to the home page of the user after login. The recommendation system implements the Apriori algorithm for recommendation. Recommendation system works based the user's followed tags. The project implements cosine similarity algorithm to provide similar questions at the time of creating question. Cosine similarity algorithm also used to find related questions of user search. The main challenge to implement the project is to use the technologies properly.

Table of Contents

CHAPTER-01: INTRODUCTION.....	1
1.1 PURPOSE.....	1
1.2 INTENDED AUDIENCE	1
1.3 CONCLUSION	1
CHAPTER-02: INCEPTION.....	2
2.1 INTRODUCTION	2
2.1.1 LIST OF STAKEHOLDERS.....	2
2.1.2 MULTIPLE VIEWPOINTS	3
2.1.3 WORKING TOWARDS COLLABORATION	3
2.2 CONCLUSION.....	4
CHAPTER-03: ELICITATION.....	5
3.1 INTRODUCTION	5
3.2 ELICITING REQUIREMENTS.....	5
3.2.1 COLLABORATIVE REQUIREMENTS GATHERING.....	5
3.2.2 PROBLEM IN THE SCOPE	6
3.2.3 QUALITY FUNCTION DEPLOYMENT	6
3.2.4 USAGE SCENARIO	7
3.2.5 ELICITATION WORK PRODUCT.....	8
CHAPTER-04: SCENARIO BASED MODELING	10
4.1 INTRODUCTION	10

4.2 DEFINITION OF USE CASE	10
4.3 USE CASE DIAGRAM.....	11
4.3.1 LEVEL- 0 USE CASE DIAGRAM-QAF (question-answer forum).....	11
4.3.2 LEVEL -1 USECASE DIAGRAM-SUBSYSTEM of qaf	12
4.3.3 LEVEL- 1.1 USE CASE DIAGRAM- question management	13
4.3.4 LEVEL -1.2 USE CASE DIAGRAM- ANSWER MANAGEMAGEMENT.....	14
4.3.5 LEVEL -1.3 USE CASE DIAGRAM- COMMENT MANAGEMENT	15
4.3.6 LEVEL -1.4 USE CASE DIAGRAM -USER MANAGEMENT	16
4.4 Activity Diagrams	17
CHAPTER-05: DATA BASED MODELING OF QAF	24
5.1 INTRODUCTION	24
5.2 DATA OBJECTS.....	24
5.2.1 FINAL DATA OBJECT	24
5.3 SCHEMA DIAGRAM.....	25
CHAPTER–06 CLASS BASED MODELING	28
6.1 INTRODUCTION	28
6.2 IDENTIFYING ANALYSIS CLASS.....	28
6.2.1 CLASS CARDS.....	28
6.2.2 CLASS COLLABORATION DIAGRAM	32
CHAPTER–07 IMPLEMENTATION OVERVIEW	33
7.1 INTRODUCTION	33

7.2 TOOLS AND TECHNOLOGIES.....	33
7.2.1 BACKEND TECHNOLOGIES.....	33
7.2.2 FRONTEND TECHNOLOGIES.....	33
7.2.3 DATABASE	33
7.2.4 THIRD PARTY DEPENDENCY.....	33
7.2.5 OTHERS	33
7.3 TECHNICAL CHALLENGES.....	33
7. 4 ALGORITHMS	34
7.4.1 APRIORI ALGORITITHM.....	34
7.4.2 RECOMMEDATION SYSTEM USING APRIORI.....	39
7.4.3 COSINE SIMILARITY ALGORITHM	40
7.4.4 MEASURING SIMILARITY BETWEEN QUESTIONS	40
CHAPTER-08 USER MANUAL	41
AUTHENTICATION	41
HOME PAGE	43
RECOMMENDED QUESTIONS	44
DISCUSSION PAGE.....	44
ASKING A QUESTION.....	45
CHAPTER–09: CONCLUSION	50
References.....	51

TABLES OF FIGURES

Figure 1: level 0 use case diagram- QAF.....	11
Figure-2: Figure 2level 1 use case diagram - Subsystem	12
Figure-3: Figure 3level 1.1 use case diagram – Question management	13
Figure 4: level 1.2 use case diagram- Answer management.....	14
Figure 5: level 1.3 use case diagram- Comment management	15
Figure 6: level 1.4 use case diagram- user management	16
Figure 7: Level 1.1 Activity diagram – Create question.....	17
Figure 8: Level 1.1.1 Activity diagram – User Reaction.....	18
Figure 9: Activity diagram of Answer to question	19
Figure 10: Level 1.1.2 Activity diagram – User reaction to answer.....	20
Figure 11: Level 1.1.3 Activity diagram –User comment	21
Figure 12: Level 1.2 Activity diagram – User Reaction to Comment.....	22
Figure 13: Level 1.2.1 Activity diagram – User search.....	23
Figure 14: Class collaboration diagram of QAF.....	32
Figure 15: User Authentication using OAuth	41
Figure 16: Sing in using google	42
Figure 17:Home page.....	43
Figure 18: Recommended question list for user	44
Figure 19: discussion page.....	44
Figure 20: 1st step to create a question.....	45

Figure 21:Inserting tags related to questions	46
Figure 22: Inserting question title	46
Figure 23: Similar question related to asking questions	47
Figure 24: Giving question description.....	47
Figure 25: Overview of all provided data	48
Figure 26: Discussion page after asking question.....	48
Figure 27: Help page.....	49

LIST OF TABLES

Table 1: schema table of User data object	25
Table 2: schema table of Question data object	25
Table 3: schema table of Answer data object	26
Table 4: schema table of Comment data object	26
Table 5: schema table of Tag data object.....	27
Table 6: schema table of Recommendation system.....	27
Table 7: User.....	28
Table 8: Question.....	29
Table 9:Answer	29
Table 10: Comment.....	29
Table 11: UserActivity.....	30

Table 12: Recommendation	30
Table 13: Tag	31
Table 14: UserManager.....	31
Table 15: Item Set	34
Table 16: support count of items	35
Table 17: The frequent item set(L1)	35
Table 18: Generated candidates(C2).....	36
Table 19: Frequent item set(L2)	37
Table 20: Generated candidates(C3).....	37
Table 21: Frequent item set.....	38
Table 22: User tags for two users stored in database.....	39

CHAPTER-01: INTRODUCTION

This chapter includes description of the purpose and intended audience.

1.1 PURPOSE

This document briefly describes the Software Requirement Analysis of Question Answer Forum. It can be considered the précised version of the SRS document. The requirements contained in the SRS are independent, uniquely numbered and organized by topic. The SRS serves as an official means of communicating user requirements to the developer and provides a common reference point for both the developer team and the stakeholder community. The SRS will evolve over time as users and developers work together to validate, clarify and expand its contents.

1.2 INTENDED AUDIENCE

This SRS is intended for several audiences including the customers as well as the project designers, developers. The customer will use this SRS to verify that the developer team has created a product that the customer finds acceptable. The designers will use this SRS as a basis for creating the system's design. The designers will continually refer back to this SRS to ensure that the system they are designing will fulfill the customer's demands. The developers will use this SRS as a basis for developing the system's functionality. The developers will link the requirements defined in this SRS to the software they create to ensure that they have created a software that will fulfill all of the customer's documented requirements. When portions of the software are complete, the developer will run their tests on that software to ensure that the software fulfills the requirements documented in this SRS. The testers will again run their tests on the entire system when it is complete and ensure that all requirements documented in this SRS have been fulfilled.

1.3 CONCLUSION

This analysis of the audience helped to focus on the users who will be using my 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. The project inception establishes a basic understanding of the problem, the people who want a solution, the nature of the solution that is desired and the effectiveness of preliminary communication and collaborations between the other stakeholders and the software team.

The factors are 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, I consulted a teacher and about 90 students of Dhaka University of Bangladesh:

The identified stakeholders are as follows:

- Inquirer
- Responder

Inquirer: Inquirer is a person who ask question.

Responder: Responder is a person who reply 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, It is necessary to recognize the requirements from multiple points of view, as well as multiple views of requirements. Viewpoints 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: Users can create a question to this forum.
- Answering to question: Users can answer to a question

- Comment system: Users can comment to questions and answers.
- Rating system: Users can upvote or downvote to question/answer
- Question recommendation: After log in user can view the recommended question for him/her.
- Suggesting similar questions: When user will ask question, he/she will view the questions similar to the asking question.
- Search user: Any user can search for another user and can view profile.

CONFLICTING REQUIREMENTS

- Profile view: A user can view profile of other users

FINAL REQUIREMENTS

- Creating Question
- Answering to question
- Rating system
- Comment system
- Profile view
- Question recommendation
- Suggesting similar questions
- Search user

2.2 CONCLUSION

The Inception phase helped to establish a basic understanding about the Educational Question Answer Forum, 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 my stakeholders.

CHAPTER-03: ELICITATION

After discussing on the inception phase, I 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. I 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, I have surpassed these problems in an organized and systematic manner.

3.2 ELICITING REQUIREMENTS

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. I 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

The stakeholders are Inquirer and Responder. Many different approaches to collaborative requirements gathering have been proposed by the stakeholders. This problem has been solved by discussion 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.)
- The forum is only for educational purpose not other.

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, I 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 my 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: The total reputation/popularity point will be converted to silver, gold and platinum count. 1 silver badge = 50 reputation, 1 gold badge = 100 reputation and 1 platinum badge = 200 reputation.

3.2.4 USAGE SCENARIO

Question-Answer Forum is a web app which include the following scenarios:

- Question
- Answer
- Comment
- User

3.2.4.1 QUESTION MANAGEMENT

QUESTION MANAGEMENT

User will create a question going through 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 to 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. To submit user must have to be signed. If he/she is new user he/she signup using OAuth [1] service provider auth0 [2] authentication system.

RECOMMENDING QUESTION TO USERS

In this forum different user from different educational background 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. The question or answer is considered the comment source. User will have to gain minimum 50 reputation to comment. Any user can upvote/downvote to a comment. Any user can mark the comment as usable or as problematic. If the negative rating crosses the limit of 50 then comment will be blocked. User reputation also increase/decrease based on user reaction.

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. As a result, user will appear in the search of users more in front position. He/she will gain different badge (silver, gold, platinum) at different level of popularity. User will use the online authentication system to be authenticated(auth0). User will be identified by a unique id.

3.2.5 ELICITATION WORK PRODUCT

At first, it is necessary 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 the requirements for the Question Answer Forum.
- Making a bounded statement of scope for my 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 there is better understanding of how end users (and other actors) want to interact with a system, the 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.

4.3.1 LEVEL- 0 USE CASE DIAGRAM-QAF (QUESTION-ANSWER FORUM)

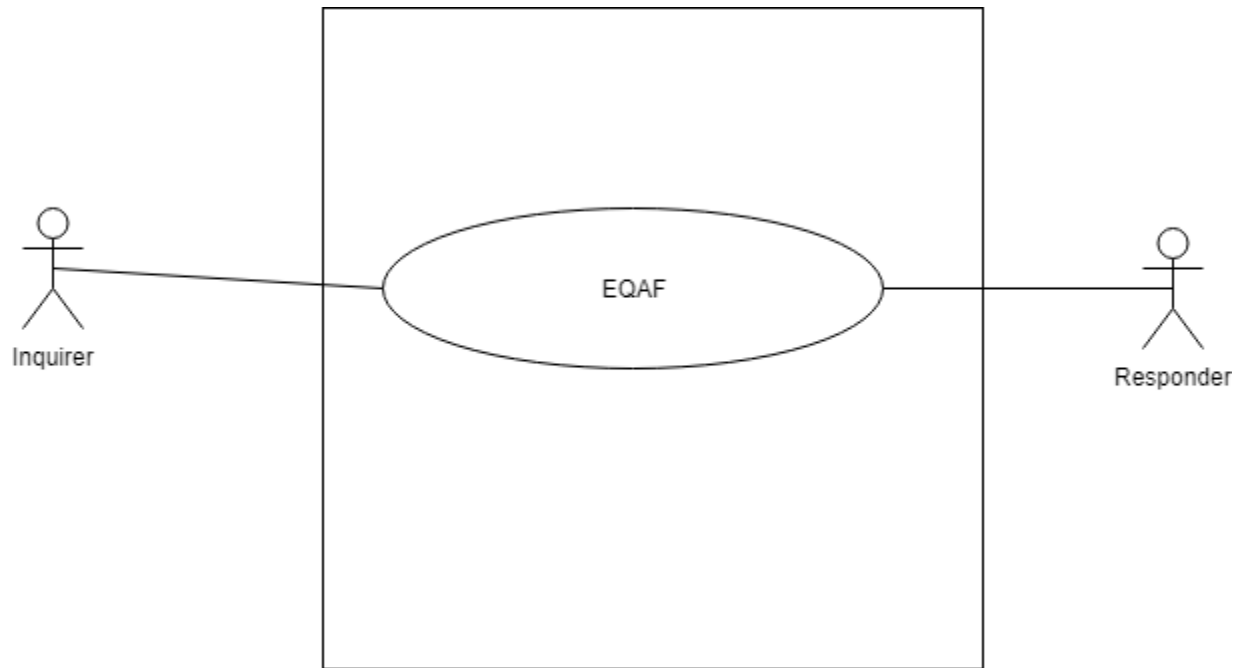


Figure 1: level 0 use case diagram- QAF

Name: Educational Question-Answer Forum

Primary actor: Inquirer, Responder

Secondary actor: N/A

DESCRIPTION OF USE CASE DIAGRAM LEVEL-0

After analyzing user story, two actors are 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 one the actors of the system:

- Inquirer
- Responder

4.3.2 LEVEL -1 USECASE DIAGRAM-SUBSYSTEM OF QAF

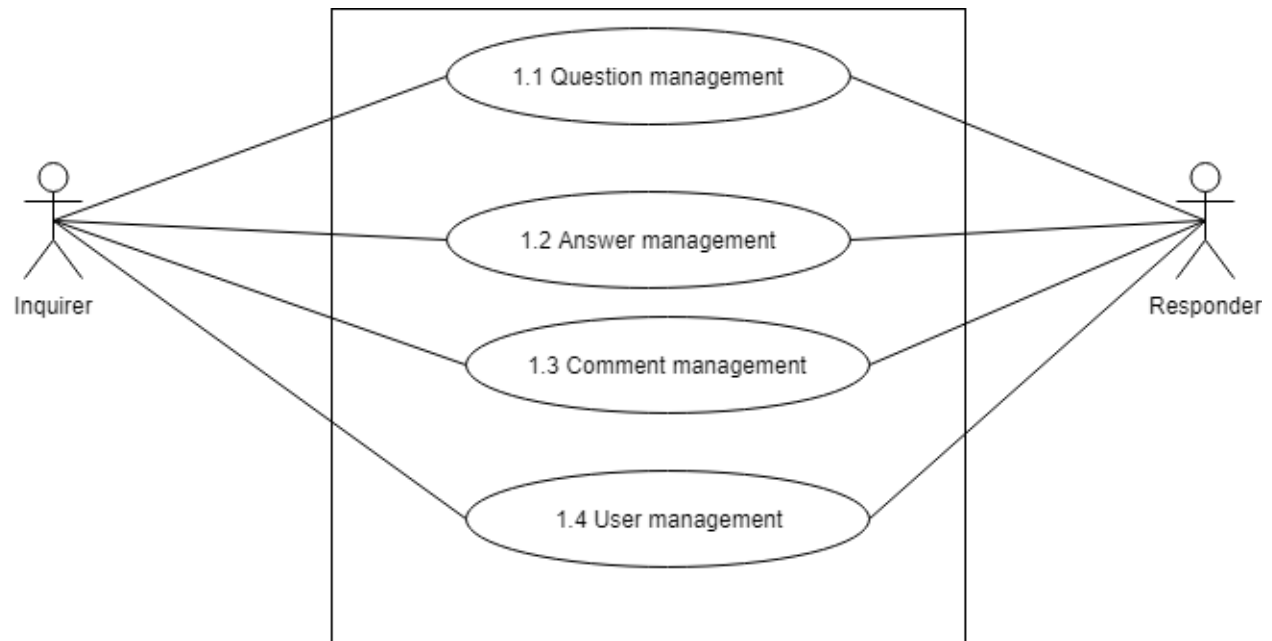


Figure-2: Figure 2level 1 use case diagram - Subsystem

Name: Subsystem of QAF

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

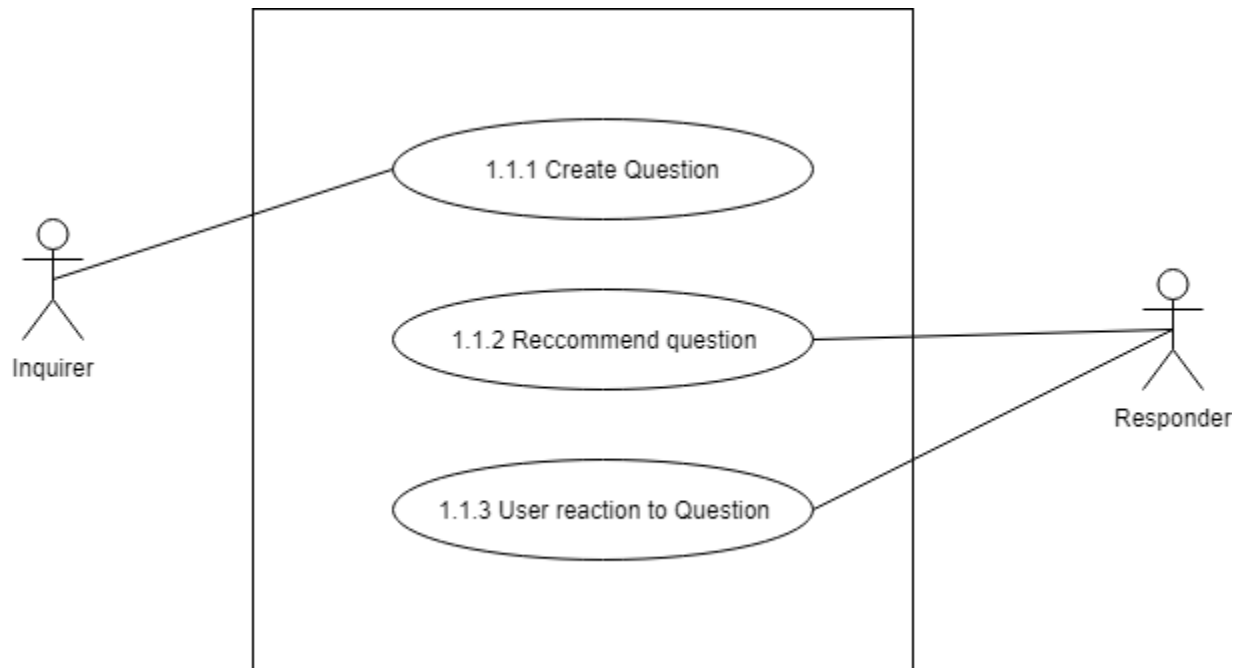


Figure-3: Figure 3level 1.1 use case diagram – Question management

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

4.3.4 LEVEL -1.2 USE CASE DIAGRAM- ANSWER MANAGEMENT

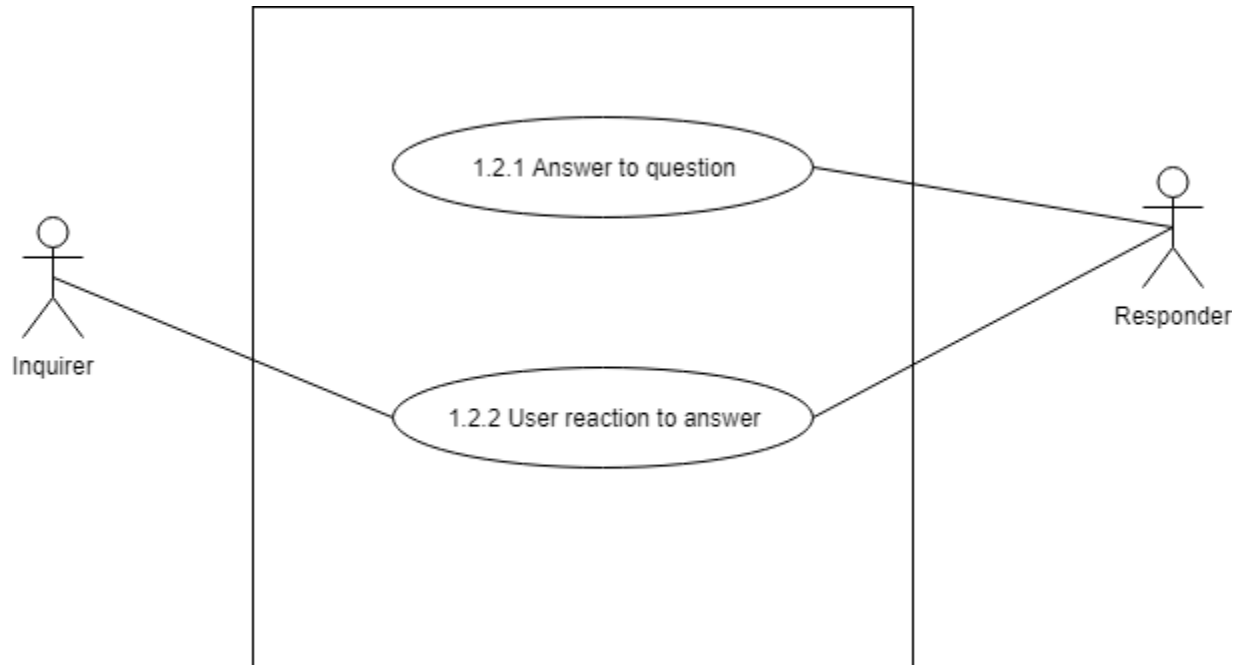


Figure 4: level 1.2 use case diagram- Answer management

Name: Answer management of QAF

Primary actor: Responder, Inquirer

Secondary actor: N/A

DESCRIPTION OF LEVEL- 1.2 USE CASE DIAGRAM-

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

- Answer to question
- User reaction to answer

4.3.5 LEVEL -1.3 USE CASE DIAGRAM- COMMENT MANAGEMENT

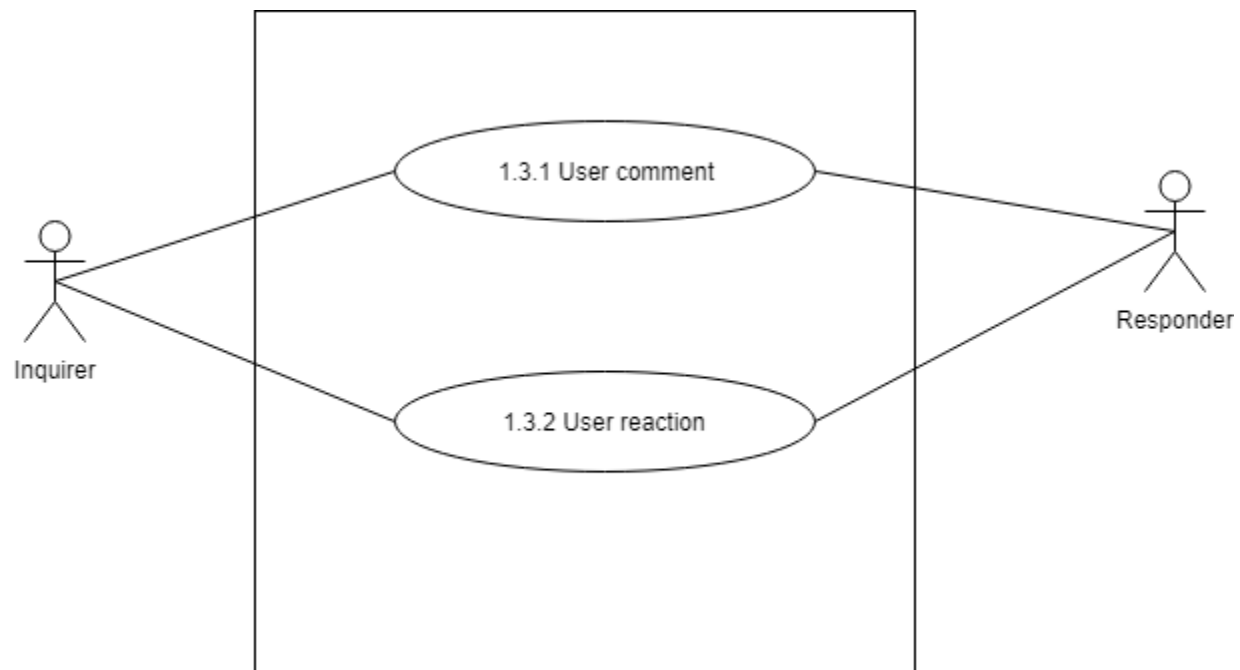


Figure 5: level 1.3 use case diagram- Comment management

Name: Comment management of QAF

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

4.3.6 LEVEL -1.4 USE CASE DIAGRAM -USER MANAGEMENT

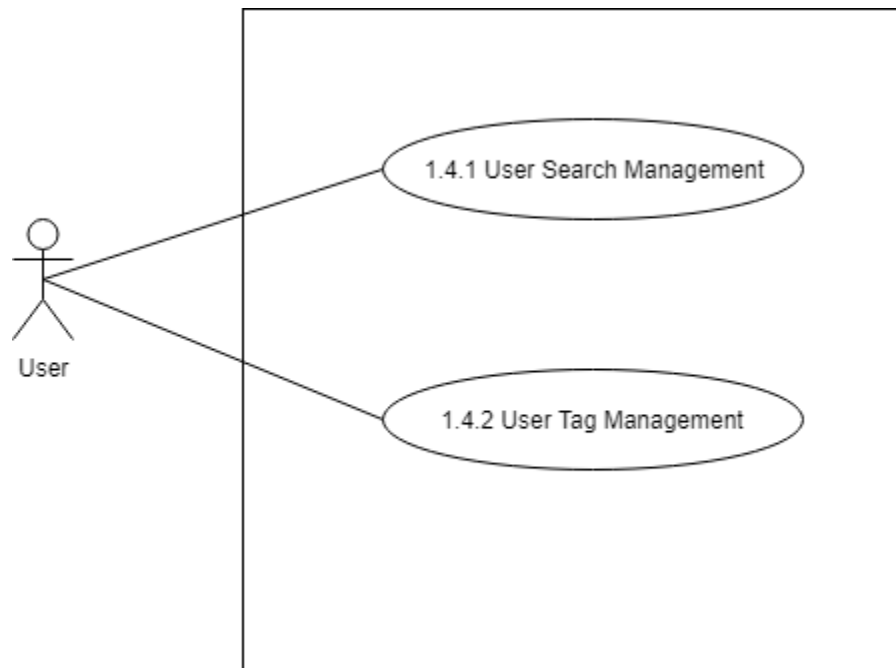


Figure 6: level 1.4 use case diagram- user management

Name: User Management of QAF

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

4.4 ACTIVITY DIAGRAMS

ACTIVITY DIAGRAM: CREATE QUESTION

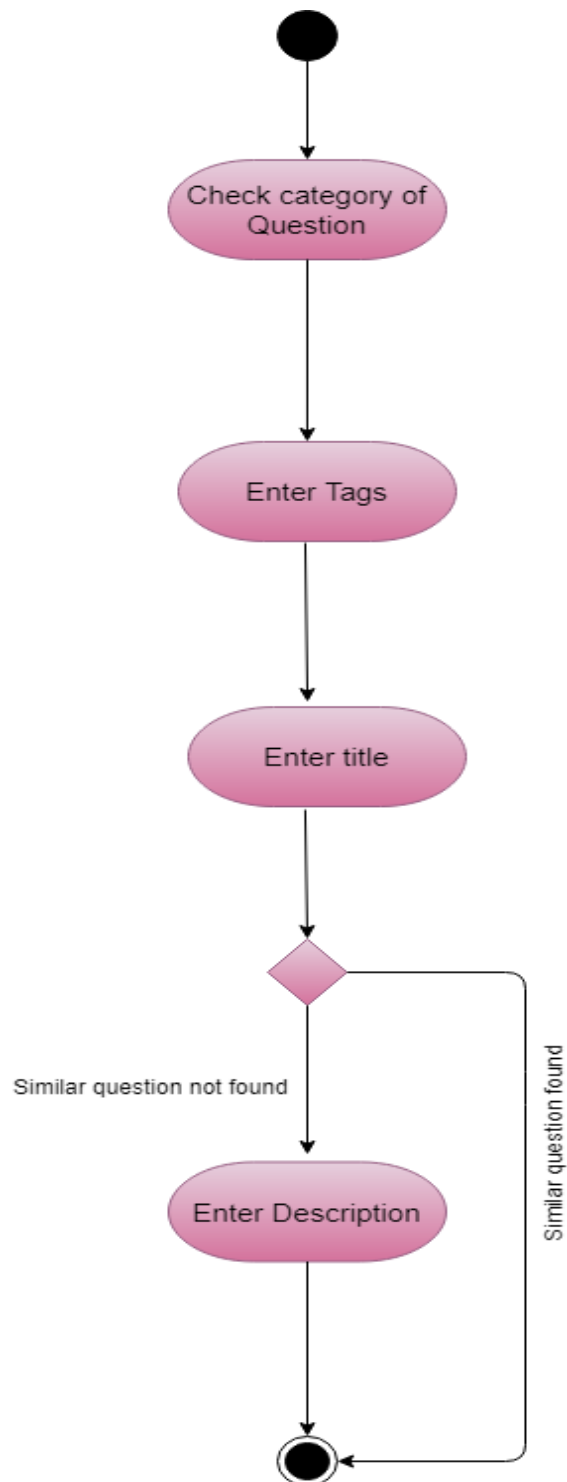


Figure 7: Level 1.1 Activity diagram – Create question

ACTIVITY DIAGRAM: USER REACTION

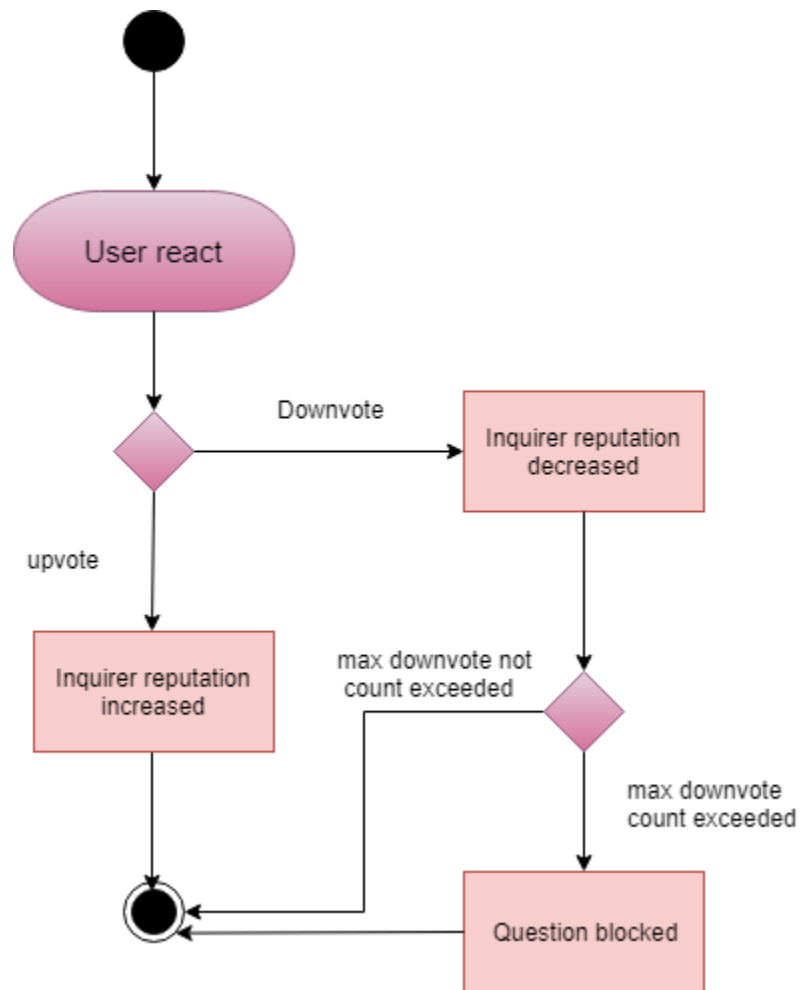


Figure 8: Level 1.1.1 Activity diagram – User Reaction

ACTIVITY DIAGRAM: ANSWER TO QUESTION

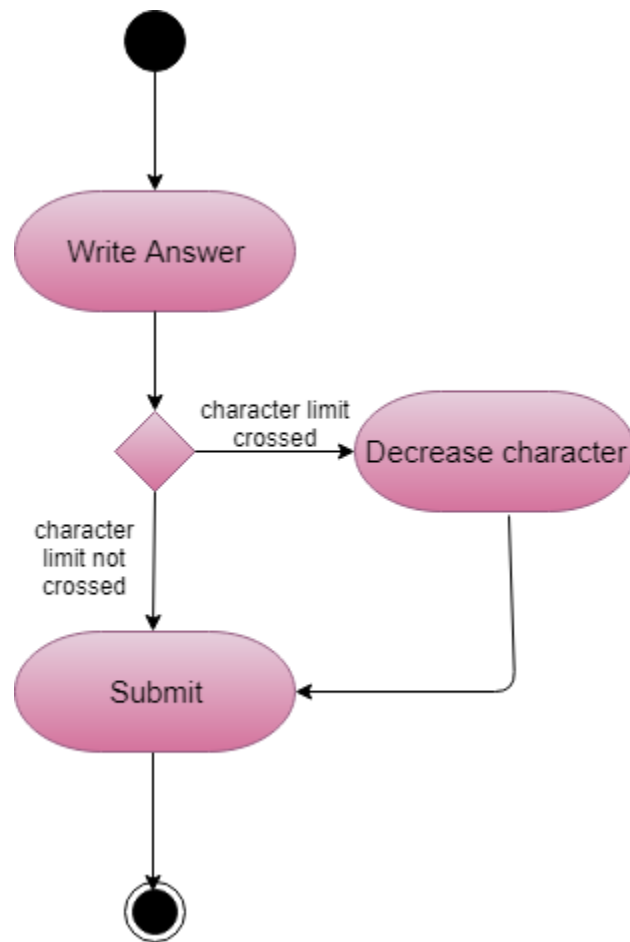


Figure 9: Activity diagram of Answer to question

ACTIVITY DIAGRAM: USER REACTION TO ANSWER

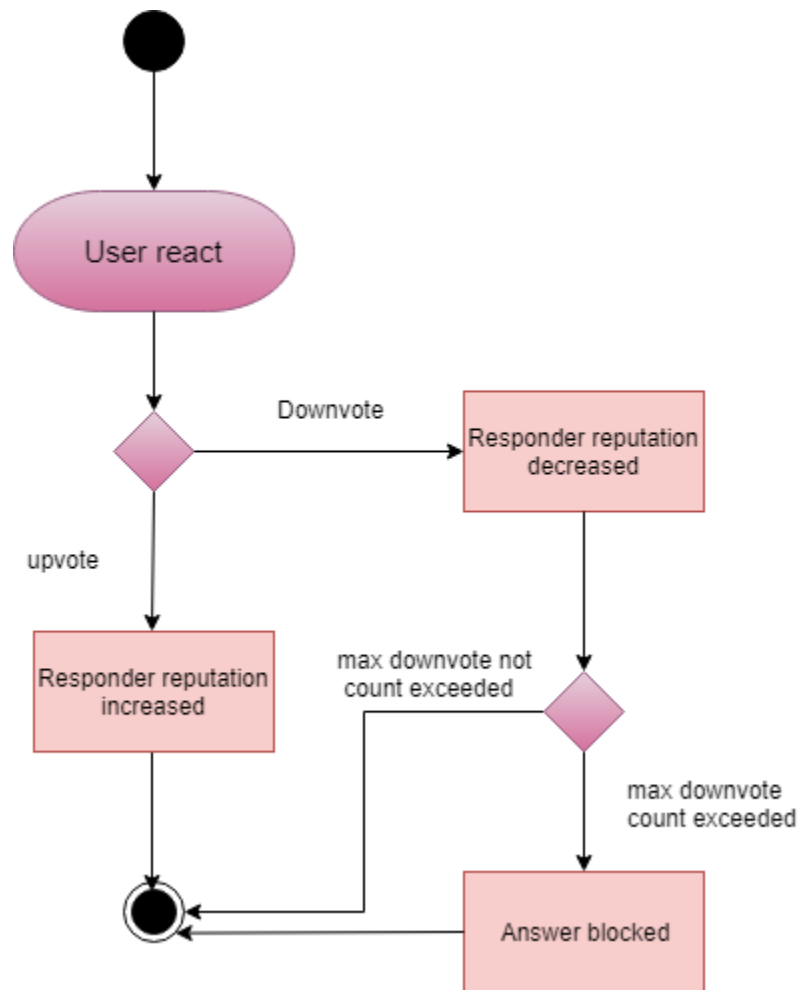


Figure 10: Level 1.1.2 Activity diagram – User reaction to answer.

ACTIVITY DIAGRAM: USER COMMENT

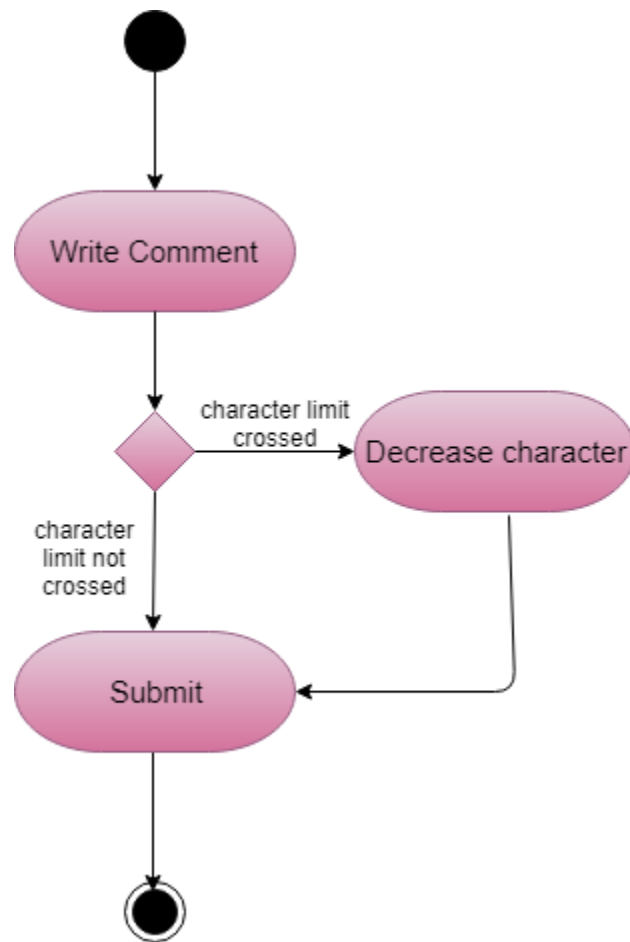


Figure 11: Level 1.1.3 Activity diagram –User comment

ACTIVITY DIAGRAM: USER REACTION TO COMMENT

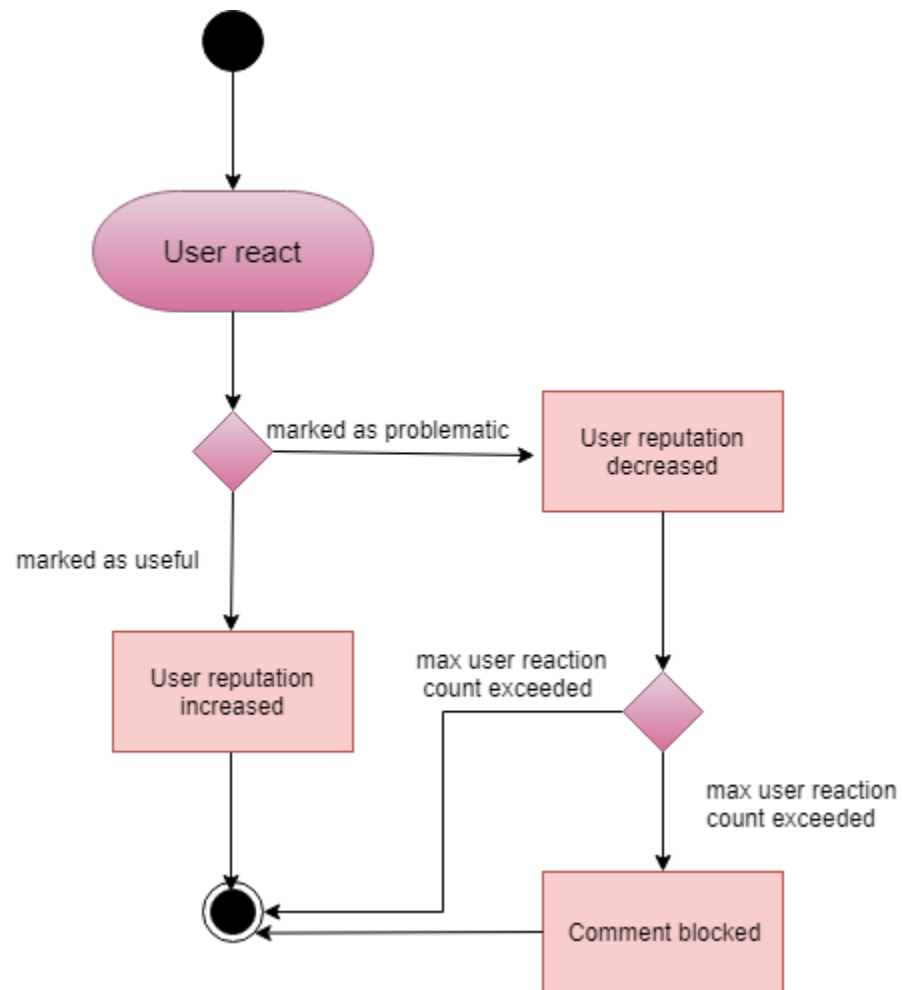


Figure 12: Level 1.2 Activity diagram – User Reaction to Comment

ACTIVITY DIAGRAM: USER SEARCH

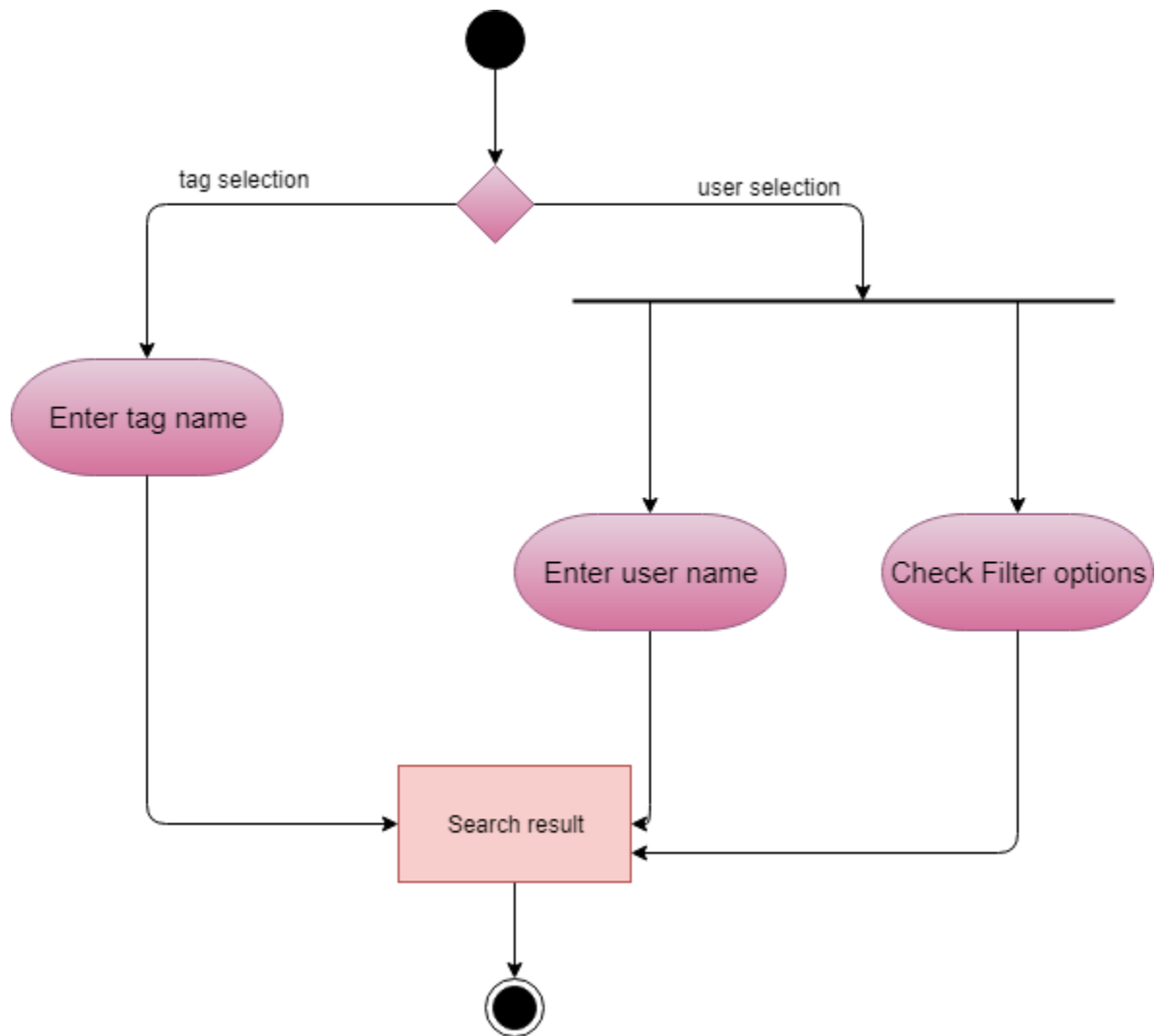


Figure 13: Level 1.2.1 Activity diagram – User search

CHAPTER-05: DATA BASED MODELING OF QAF

This chapter describes the Scenario-Based Model for the Educational Question Answer Forum.

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 FINAL DATA OBJECT

- 1 User: userId, tags, reputation, location
- 2 Question: _fileName, description, title, time, reaction, userId, questionId, tags
- 3 Answer: _fileName, description, time, reaction, userId, questionId, answerId
- 4 Comment: description, time, reaction, userId, source
- 5 Tag: description, tagname, tagId
- 6 Notification: description, time, userId, status, notificationId

5.3 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 1: schema table of User data object

User		
Attributes	Type	Size
Id	VARCHAR	24
<u>userId</u>	VARCHAR	40
tags	VARCHAR	80
reputation	NUMBER	40
location	VARCHAR	50

Table 2: schema table of Question data object

Question		
Attributes	Type	Size
<u>Id</u>	VARCHAR	40
userId	VARCHAR	24
tags	VARCHAR	80
description	VARCHAR	50
title	VARCHAR	30

datetime	VARCHAR	15
reaction	NUMBER	80
isAccepted	Boolean	1

Table 3: schema table of Answer data object

Answer		
Attributes	Type	Size
<u>answerId</u>	VARCHAR	40
questionId	VARCHAR	40
description	VARCHAR	40
datetime	VARCHAR	50
ratings	VARCHAR	30

Table 4: schema table of Comment data object

Comment		
Attributes	Type	Size
<u>Id</u>	VARCHAR	40
userId	VARCHAR	40

targetId	VARCHAR	40
datetime	VARCHAR	80
target	VARCHAR	1
ratings	VARCHAR	50
text	VARCHAR	30

Table 5: schema table of Tag data object

Tag		
Attributes	Type	Size
<u>tag Id</u>	VARCHAR	40
description	VARCHAR	80
name	VARCHAR	40
Users	NUMBER	32

Table 6: schema table of Recommendation system

Recommendation		
Attributes	Type	Size
<u>Id</u>	VARCHAR	40
recommendations	Array: VARCHAR	40

CHAPTER-06 CLASS BASED MODELING

This chapter describes the class-based model for Educational Question-Answer Forum.

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 CLASS CARDS

After identifying my final classes, I have generated following class cards

Table 7: User

User	
Attributes	Methods
UserId, Reputation, Location, Tags, Answers, Questions, Comments	N/A

Responsibilities	Collaborative class
N/A	Tag, Comment, Answer, Question, Recommendation

Table 8: Question

Question	
Attributes	Methods
Description, Image, Tags, User, Reactions, Comments, Time	N/A
Responsibilities	Collaborative class
N/A	Answer, User, Tag, Comment

Table 9: Answer

Answer	
Attributes	Methods
Description, Image, Tags, User, Reactions, Comments, Time	N/A
Responsibilities	Collaborative class
N/A	Question, User, Tag, Comment

Table 10: Comment

Comment

Attributes	Methods
Reaction, User, Description	N/A
Responsibilities	Collaborative class
N/A	Question, Answer, User

Table 11: UserActivity

UserActivity	
Attributes	Methods
N/A	Answer (), ask (), comment (), vote ()
Responsibilities	Collaborative class
<ul style="list-style-type: none"> ➤ Save the answer information ➤ Save the question information ➤ Save the comment information ➤ Send the answer/question formation 	Answer, Question, Comment,

Table 12: Recommendation

Recommendation	
Attributes	Methods
N/A	Recommend(), findSimilarQuestions()
Responsibilities	Collaborative class
<ul style="list-style-type: none"> ➤ Recommend question to users ➤ Find similar question 	Question, Answer

Table 13: Tag

Tag	
Attributes	Methods
Name, Description, Users	N/A
Responsibilities	Collaborative class
N/A	N/A

Table 14: UserManager

UserManager	
Attributes	Methods
N/A	Get(), Set(), Search(), Filter()
Responsibilities	User
<ul style="list-style-type: none"> ➤ Send the user list based on reputation ➤ Save new user ➤ Search a user ➤ Filter users 	N/A

6.2.2 CLASS COLLABORATION DIAGRAM

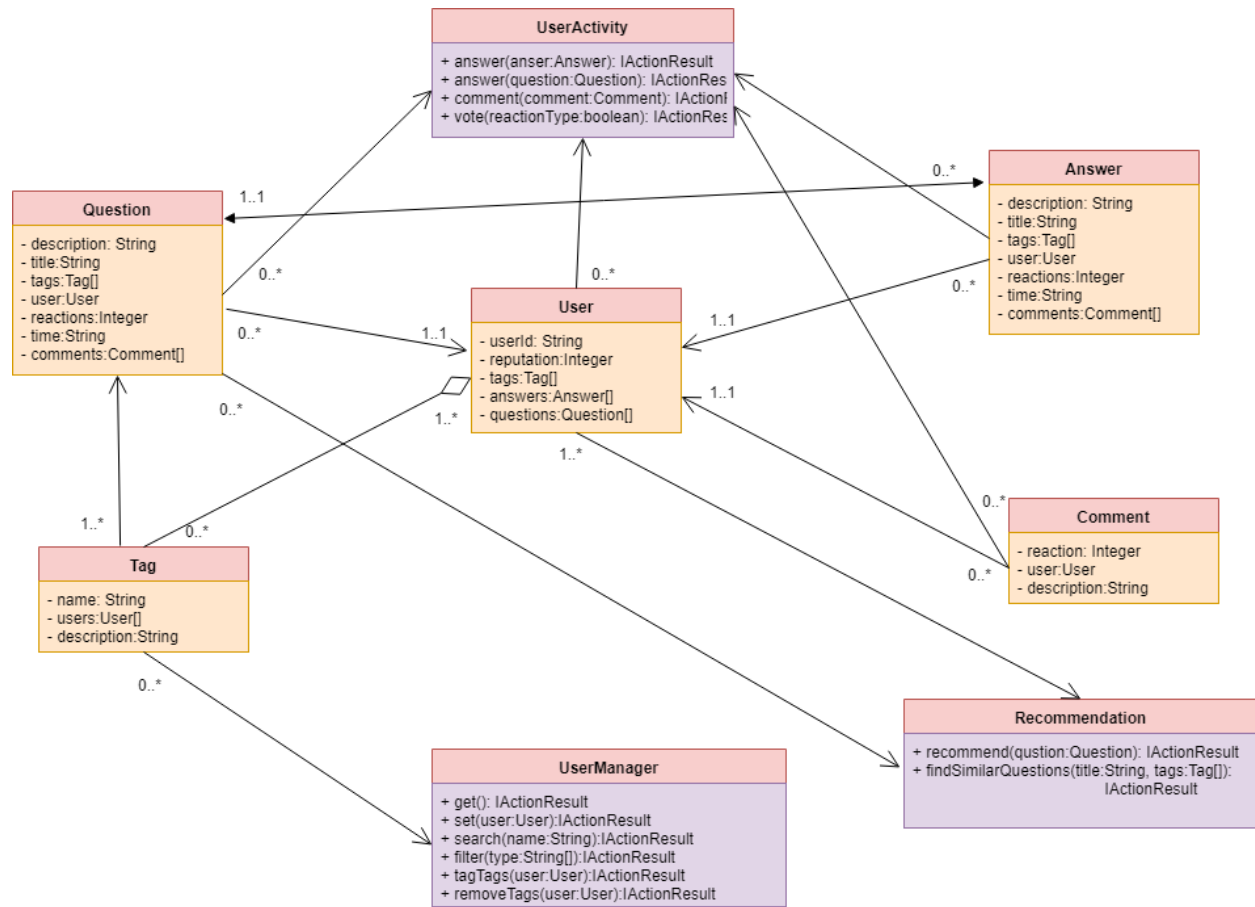


Figure 14: Class collaboration diagram of QAF

The class collaboration diagram describes the relationship between the classes. From this diagram hierarchical structure of classes and size of a project can be found.

CHAPTER-07 IMPLEMENTATION OVERVIEW

7.1 INTRODUCTION

In this project there exist technical challenges. To get solution to these challenges I need to implement some algorithms.

7.2 TOOLS AND TECHNOLOGIES

In this part describe the tools and technologies in this application.

7.2.1 BACKEND TECHNOLOGIES

Framework: ASP Dotnet Core 2.2

Programming language: C#

7.2.2 FRONTEND TECHNOLOGIES

Framework: ReactJS

Programming language: TypeScript

7.2.3 DATABASE

MongoDB Database

7.2.4 THIRD PARTY DEPENDENCY

Auth0: Auth0 service has been used to implement the OAuth2.0(Open Authentication) [1]

7.2.5 OTHERS

HTML, CSS3, Font Awesome 4

7.3 TECHNICAL CHALLENGES

The technical challenges to implement the application are below:

1. Question recommendation to appropriate users
2. Similar Question suggestion to user while creating question
3. New to C# programming, dotnet framework, ReactJS, Mongo DB, OAuth2.0.

7.4 ALGORITHMS

1. Apriori: To recommend/send asked question to appropriate users [3]
2. Cosine Similarity measurement: To get similar question while user creating question [4]

7.4.1 APRIORI ALGORITITHM

Apriori is an algorithm for frequent item set mining and association rule learning over relational databases. [5] It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database.

Consider the following dataset and we will find frequent itemsets and generate association rules for them.

Table 15: Item Set

TID	ITEM
T1	I1,I2,I5
T2	I2,I4
T3	I2,I3
T4	I1,I2,I3
T5	I1,I3
T6	I2,I3
T7	I1,I3

T8	I1,I2,I3,I5
T9	I1,I2,I3

Let minimum support count is 2 minimum confidence is 60%.

Step-1: K=1

(I) Create a table containing support count of each item present in dataset – Called **C1(candidate set)**

Table 16: support count of items

Itemset	Support count
I1	6
I2	7
I3	6
I4	2
I5	2

(II) compare candidate set item's support count with minimum support count(here min_support=2 if support_count of candidate set items is less than min_support then remove those items). This gives us itemset L1.

Table 17: The frequent item set(L1)

Itemset	Support count
I1	6
I2	7

I3	6
I4	2
I5	2

Step-2: K=2

- Generate candidate set C2 using L1 (this is called join step). Condition of joining L_{k-1} and L_{k-1} is that it should have (K-2) elements in common.
- Check all subsets of an itemset are frequent or not and if not frequent remove that itemset.(Example subset of {I1, I2} are {I1}, {I2} they are frequent.Check for each itemset)
- Now find support count of these itemsets by searching in dataset.

Table 18: Generated candidates(C2)

Itemset	Support count
I1,I2	4
I1,I3	4
I1,I4	1
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2
I3,I4	0
I3,I5	1
I4,I5	0

(II) compare candidate (C2) support count with minimum support count(here min_support=2 if support_count of candidate set item is less than min_support then remove those items) this gives us itemset L2.

Table 19: Frequent item set(L2)

Itemset	Support count
I1,I2	4
I1,I3	4
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2

Step-3:

- Generate candidate set C3 using L2 (join step). Condition of joining L_{k-1} and L_{k-1} is that it should have (K-2) elements in common. So here, for L2, first element should match. So itemset generated by joining L2 is {I1, I2, I3} {I1, I2, I5} {I1, I3, I5} {I2, I3, I4} {I2, I4, I5} {I2, I3, I5}
- Check if all subsets of these itemsets are frequent or not and if not, then remove that itemset.(Here subset of {I1, I2, I3} are {I1, I2}, {I2, I3}, {I1, I3} which are frequent. For {I2, I3, I4}, subset {I3, I4} is not frequent so remove it. Similarly check for every itemset)
- find support count of these remaining itemset by searching in dataset.

Table 20: Generated candidates(C3)

Itemset	Support count
I1,I2,I3	2
I1,I2,I5	2

(II) Compare candidate (C3) support count with minimum support count(here min_support=2 if support_count of candidate set item is less than min_support then remove those items) this gives us itemset L3.

Table 21: Frequent item set

Itemset	Support count
I1,I2,I3	2
I1,I2,I5	2

Step-4:

- Generate candidate set C4 using L3 (join step). Condition of joining L_{k-1} and L_{k-1} ($K=4$) is that, they should have ($K-2$) elements in common. So here, for L3, first 2 elements (items) should match.
- Check all subsets of these itemsets are frequent or not (Here itemset formed by joining L3 is {I1, I2, I3, I5} so its subset contains {I1, I3, I5}, which is not frequent). So no itemset in C4
- We stop here because no frequent itemsets are found further

Thus, we have discovered all the frequent item-sets. Now generation of strong association rule comes into picture. For that we need to calculate confidence of each rule.

Confidence

A confidence of 60% means that 60% of the customers, who purchased milk and bread also bought butter.

$$\text{Confidence}(A \rightarrow B) = \frac{\text{Support_count}(A \cup B)}{\text{Support_count}(A)}$$

So here, by taking an example of any frequent itemset, we will show the rule generation.

Itemset {I1, I2, I3} //from L3

SO rules can be

$$[I1 \wedge I2] \Rightarrow [I3] \text{ //confidence} = \frac{\text{sup}(I1 \wedge I2 \wedge I3)}{\text{sup}(I1 \wedge I2)} = \frac{2}{4} * 100 = 50\%$$

$[I1 \wedge I3] \Rightarrow [I2]$ //confidence = $\frac{\text{sup}(I1 \wedge I2 \wedge I3)}{\text{sup}(I1 \wedge I3)} = \frac{2}{4} * 100 = 50\%$

$[I2 \wedge I3] \Rightarrow [I1]$ //confidence = $\frac{\text{sup}(I1 \wedge I2 \wedge I3)}{\text{sup}(I2 \wedge I3)} = \frac{2}{4} * 100 = 50\%$

$[I1] \Rightarrow [I2 \wedge I3]$ //confidence = $\frac{\text{sup}(I1 \wedge I2 \wedge I3)}{\text{sup}(I1)} = \frac{2}{6} * 100 = 33\%$

$[I2] \Rightarrow [I1 \wedge I3]$ //confidence = $\frac{\text{sup}(I1 \wedge I2 \wedge I3)}{\text{sup}(I2)} = \frac{2}{7} * 100 = 28\%$

$[I3] \Rightarrow [I1 \wedge I2]$ //confidence = $\frac{\text{sup}(I1 \wedge I2 \wedge I3)}{\text{sup}(I3)} = \frac{2}{6} * 100 = 33\%$

So if minimum confidence is 50%, then first 3 rules can be considered as strong association rules.

7.4.2 RECOMMEDATION SYSTEM USING APRIORI

User follows different tags. The data are stored in database as follows:

Table 22: User tags for two users stored in database

User Id	Tags
5d8358193e2c7c281009e8c8	javascript, html, css
5db0c90db02c403c404ba8ce	গণিত, তথ্য_প্রযুক্তি, বিজ্ঞান, পদার্থ, বাংলা, ইংরেজী, রসায়ন

The data above are used as transactions in Apriori algorithm. Based on all data from database Apriori algorithm generates association rules. Two example of association rules as follows:

$\text{html} \wedge \text{css} \Rightarrow \text{javascript}$

$\text{গণিত} \wedge \text{পদার্থ} \Rightarrow \text{রসায়ন}$

The meaning of the rule $\text{html} \wedge \text{css} \Rightarrow \text{javascript}$ above is that “if a user follows html and css he/she will be also recommended questions associated with javascript”. The generated association rules are stored on database.

7.4.3 COSINE SIMILARITY ALGORITHM

Cosine similarity calculates similarity by measuring the cosine of angle between two vectors. Mathematically speaking, Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. The cosine of 0° is 1, and it is less than 1 for any angle in the interval $(0, \pi]$ radians.

$$\text{Similarity} = \cos(\Theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{k=1}^n A_i B_i}{\sqrt{\sum_{k=1}^n A_i^2} \sqrt{\sum_{k=1}^n B_i^2}}$$

7.4.4 MEASURING SIMILARITY BETWEEN QUESTIONS

The cosine similarity algorithm in question similarity as follows:

```
function getSimilarity(question1, question2):
    xList ← tokenize(question1)
    yList ← tokenize(question2)
    vect1 ← [ ]
    vect2 ← [ ]
    allwords ← union(xList, yList)
    for each word in allwords:
        if xList contains word:
            vect1.add(1)
        else vect1.add(0)
        if yList contains word:
            vect2.add(1)
        else vect2.add(0)
    sum ← 0
    i ← 0
    while i < allwords.size:
        sum ← sum + (vect1[i] * vect2[i])
        i ← i+1
    cosine ← sum / (rms(vect1) * rms(vect2))
    return cosine

function rms(list[ ]):
    rms ← 0
    for each value in list:
        rms ← rms + value*value
    return sqrt(rms)
```

CHAPTER-08 USER MANUAL

AUTHENTICATION

OAuth is used of user authentication.

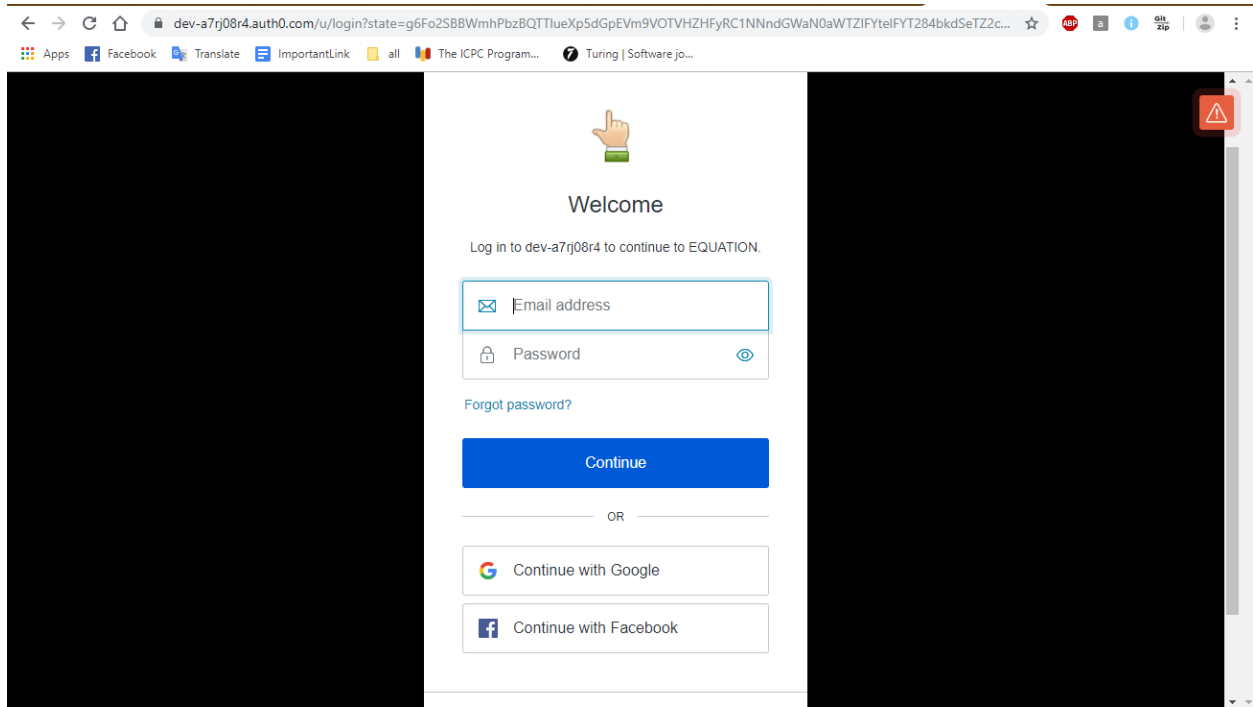


Figure 15: User Authentication using OAuth

User can log in with Gmail or Facebook. The auth0 authentication service has been used here. Dev-a7rj08r4 refers the auth0 API id.

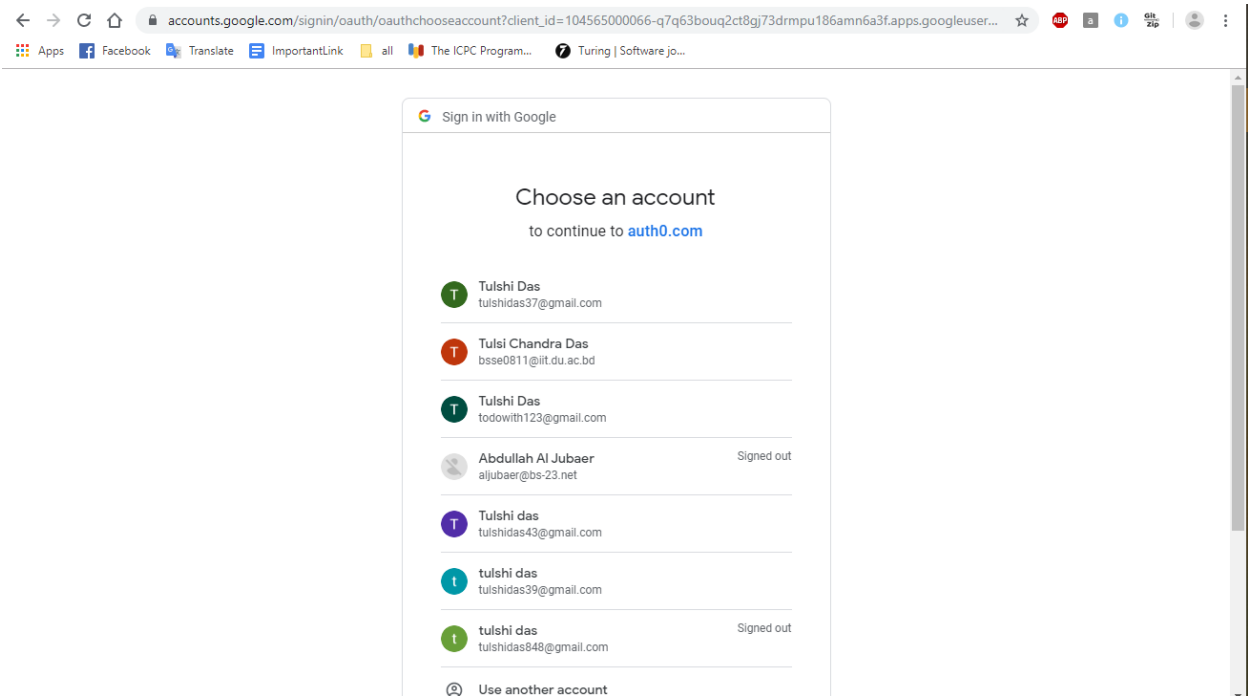


Figure 16: Sing in using google

HOME PAGE

User will see the recommended questions after login in home page.

User can click on a question and go to the discussion page.

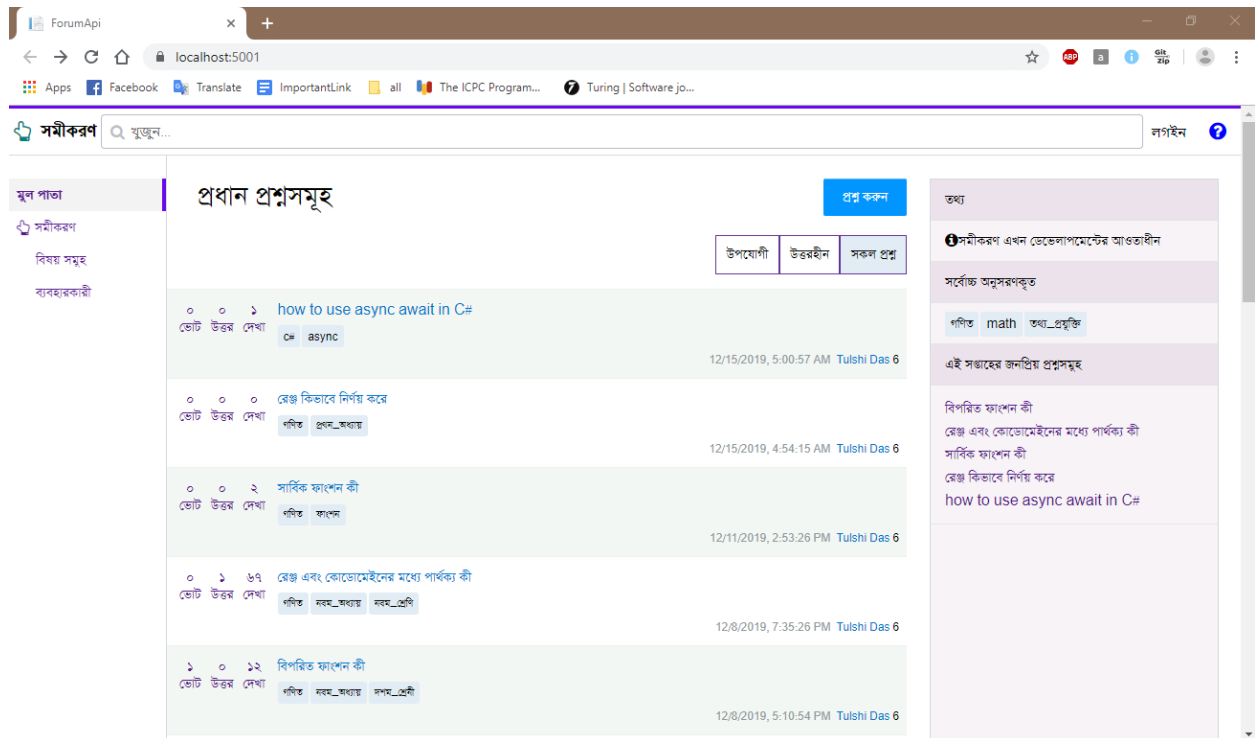


Figure 17: Home page

RECOMMENDED QUESTIONS

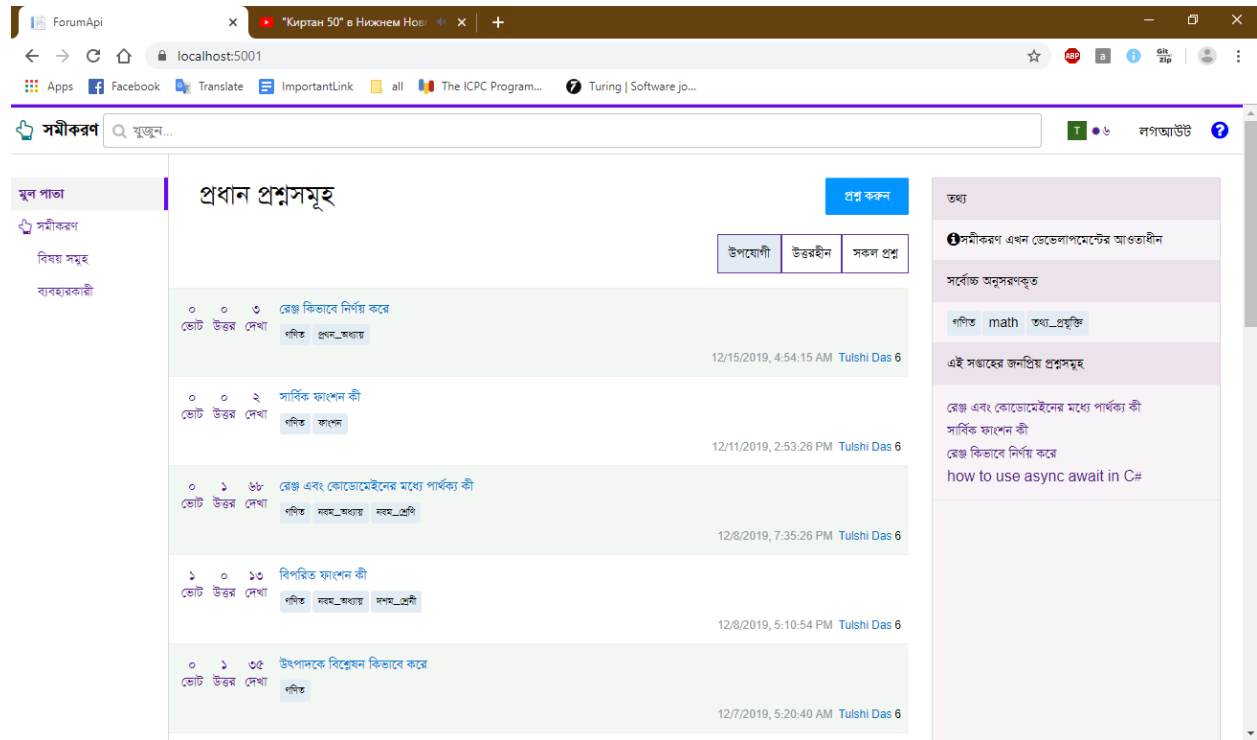


Figure 18: Recommended question list for user

DISCUSSION PAGE

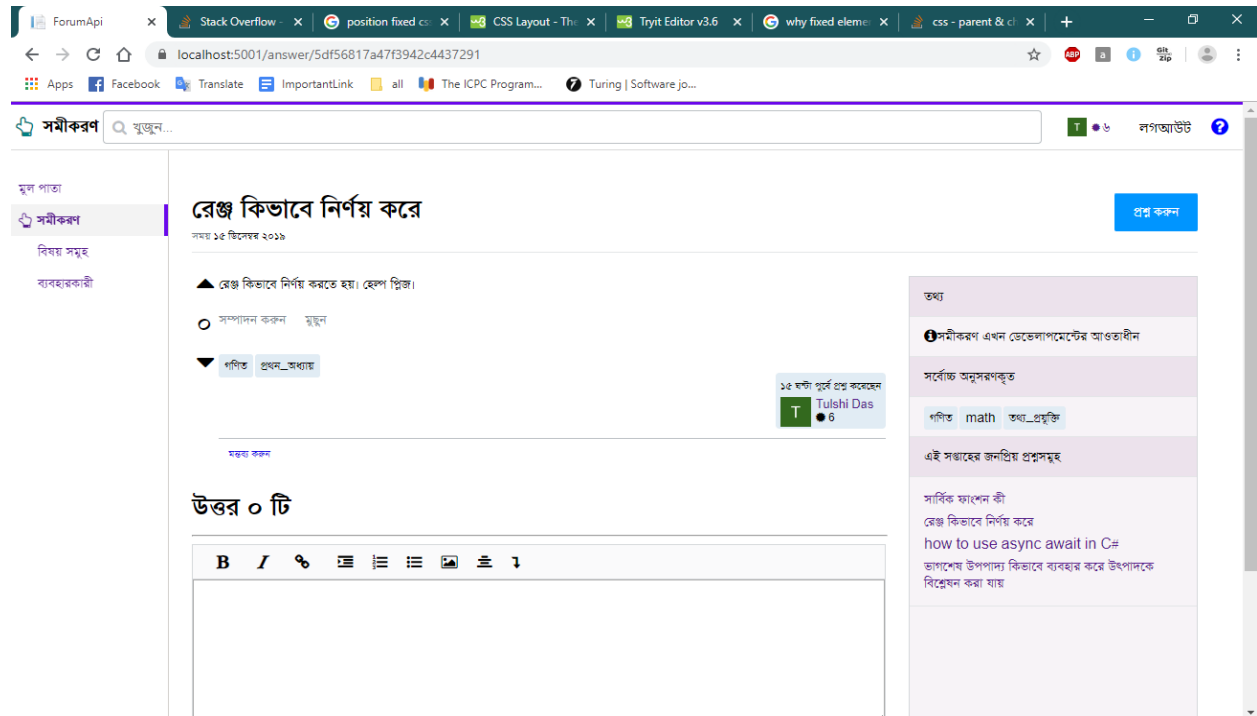


Figure 19: discussion page

Here user can answer to question, comment and give react.

ASKING A QUESTION

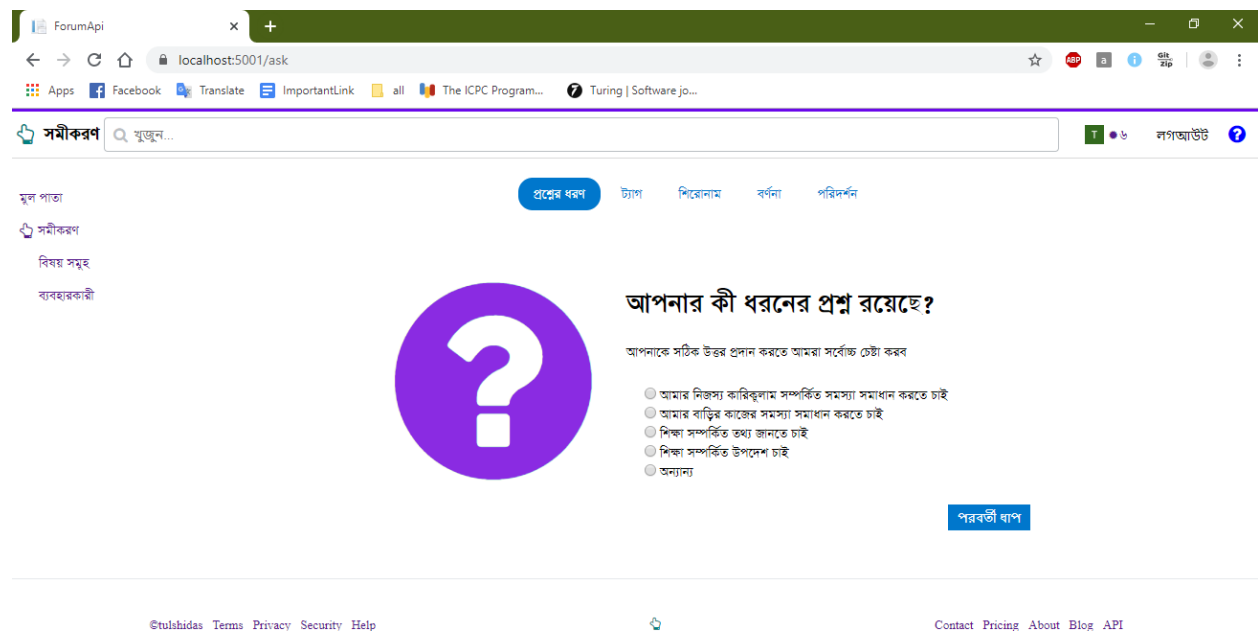


Figure 20: 1st step to create a question

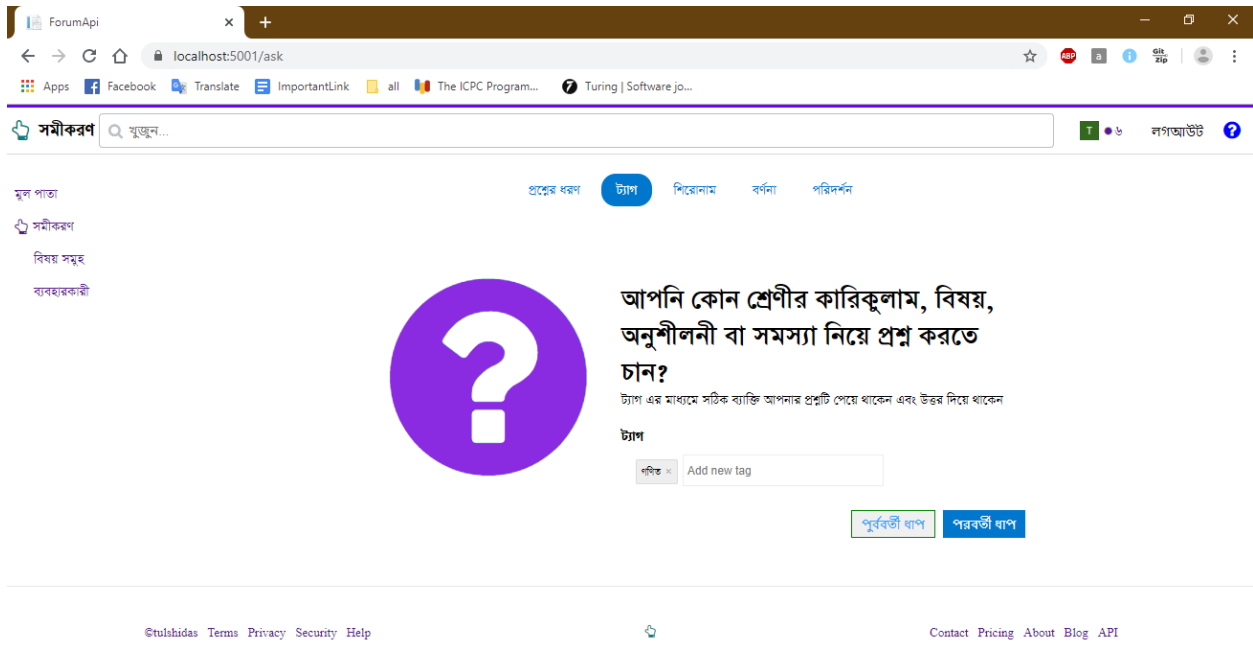


Figure 21: Inserting tags related to questions

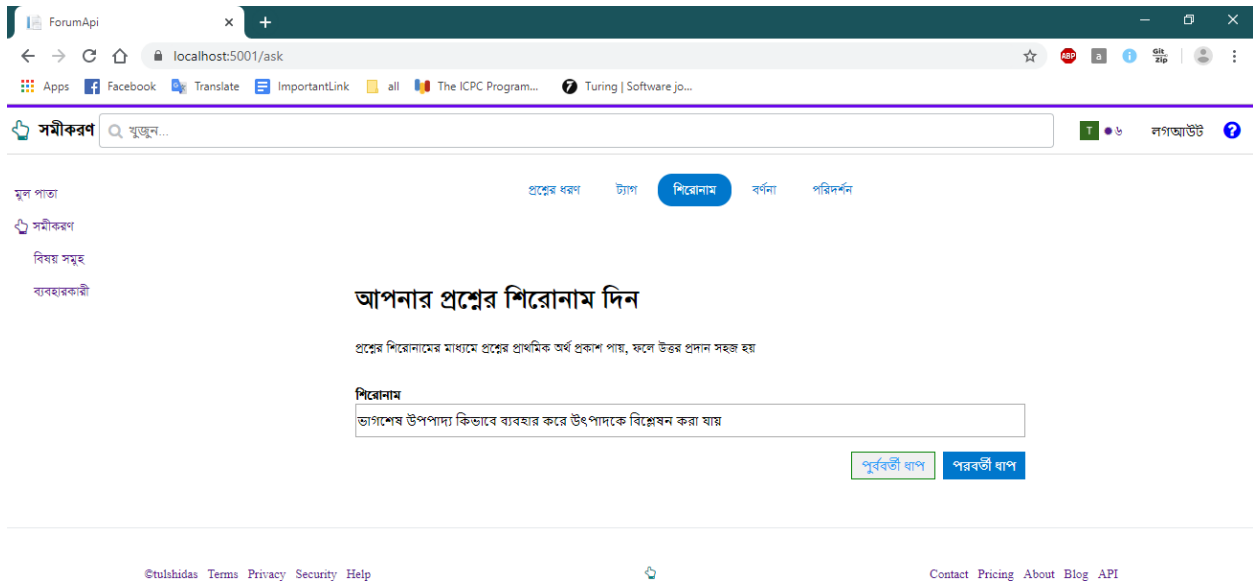


Figure 22: Inserting question title

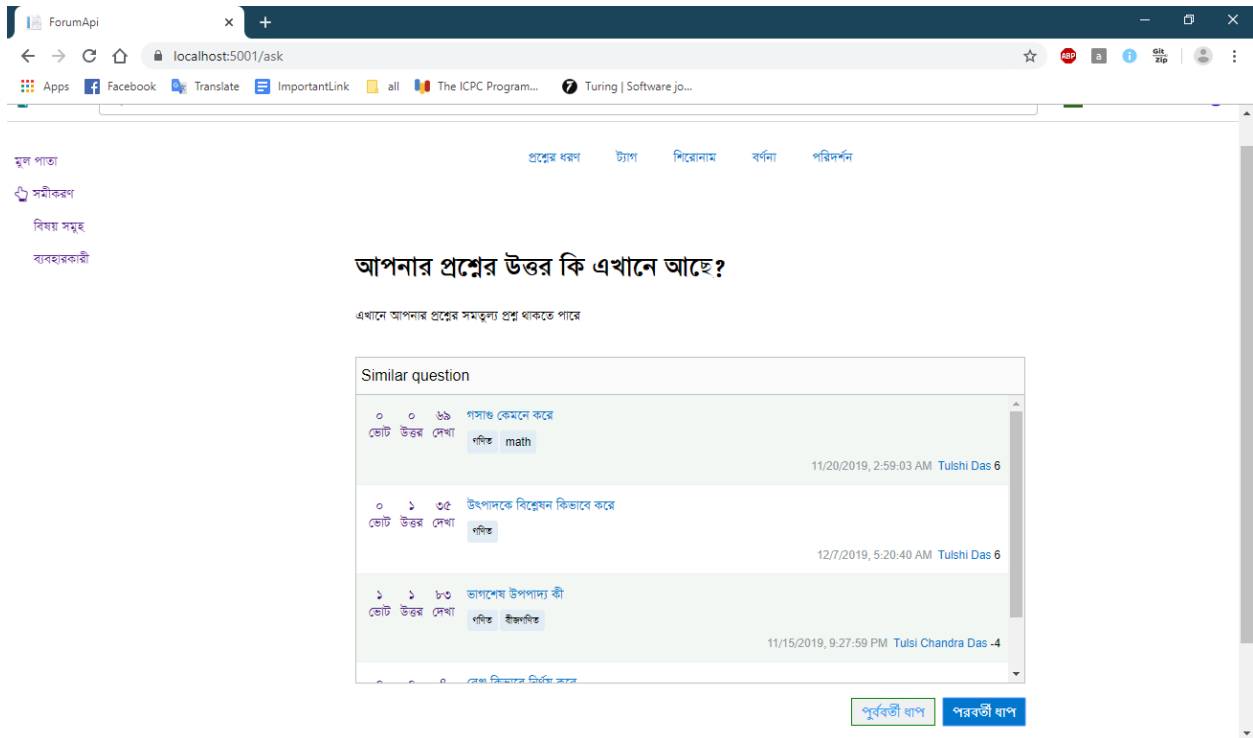


Figure 23: Similar question related to asking questions

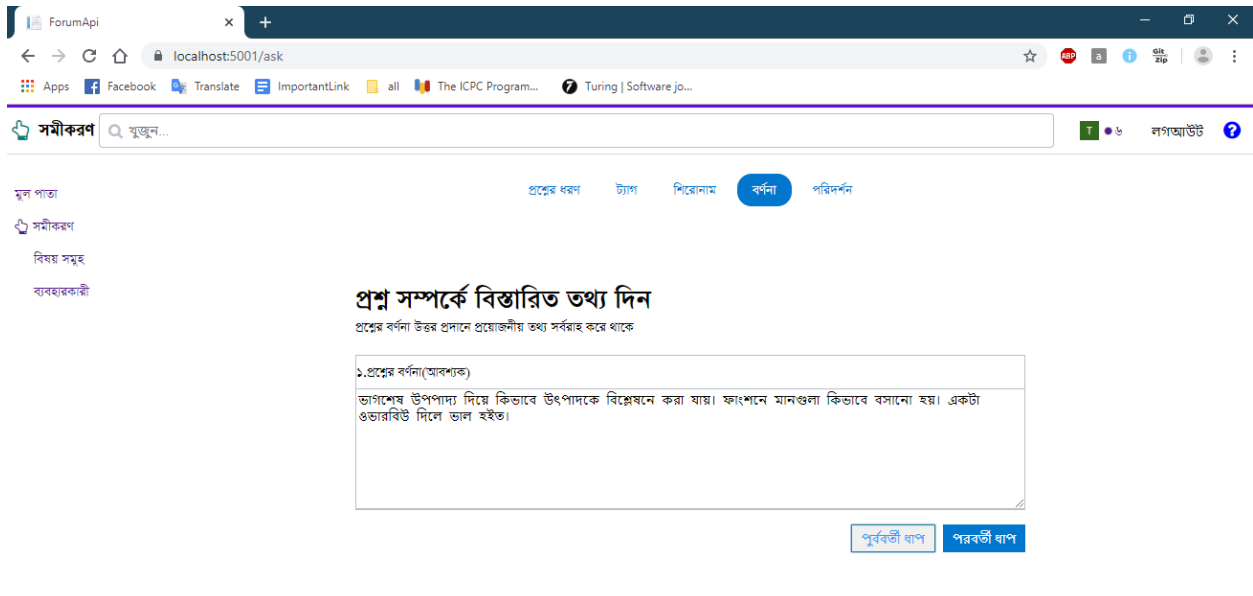


Figure 24: Giving question description

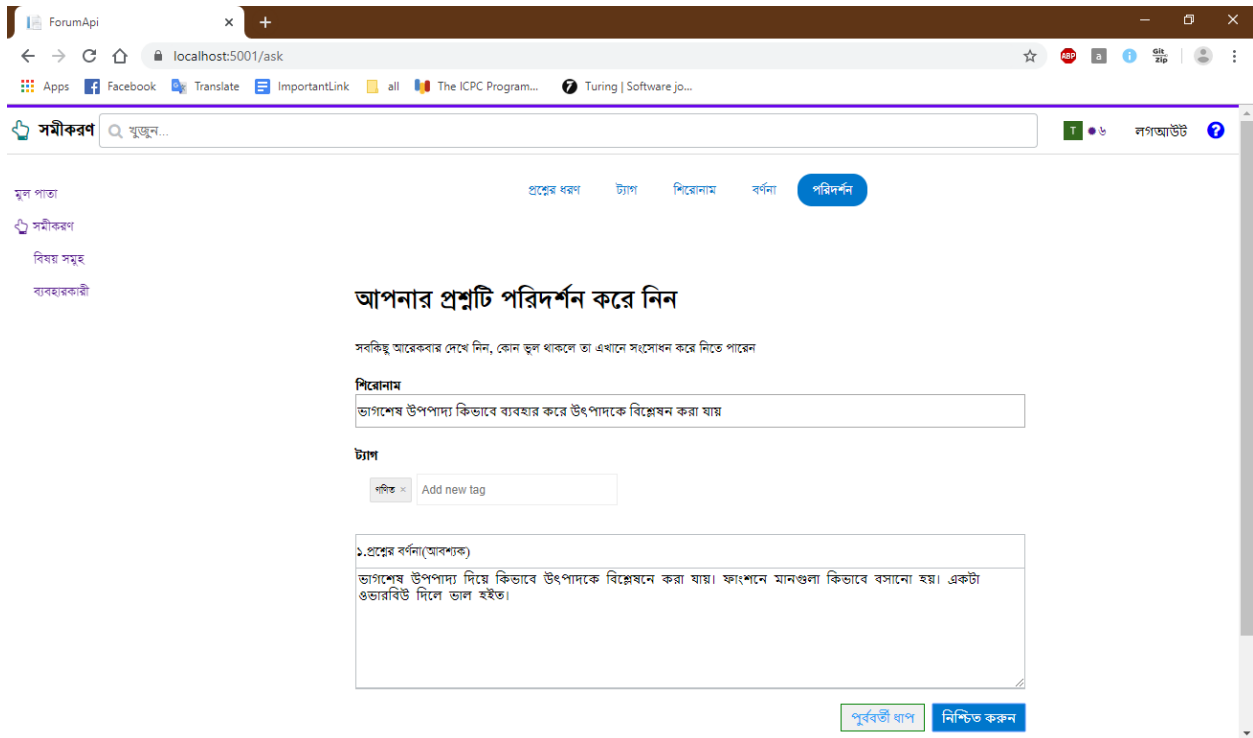


Figure 25: Overview of all provided data

If user click in “নিশ্চিত করুন” then question will be posted and discussion page will be displayed.

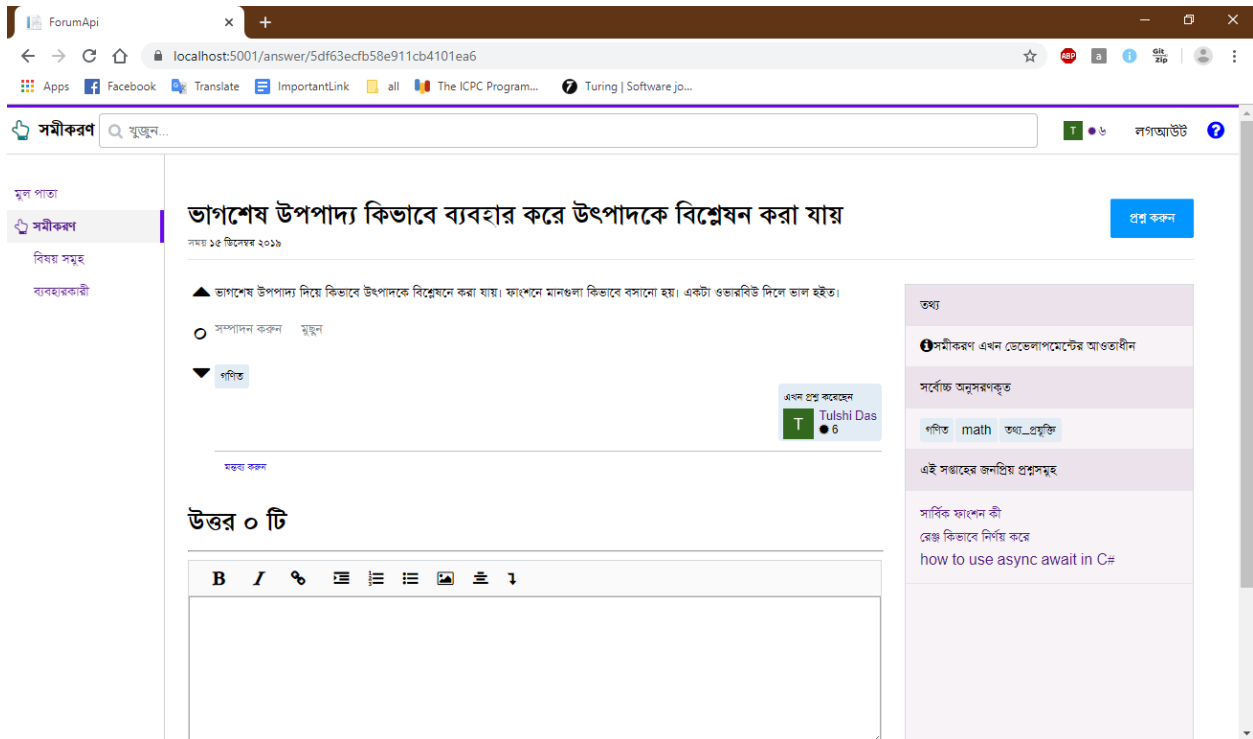


Figure 26: Discussion page after asking question

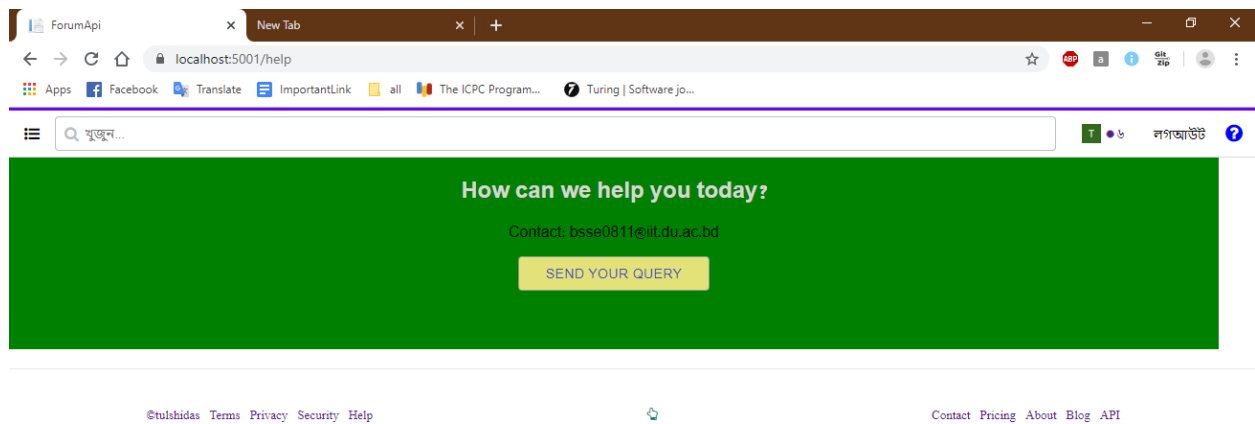


Figure 27: Help page

CHAPTER-09: CONCLUSION

I am pleased to submit the final report on Question Answer Forum. From this, the readers will get a clear and easy view of the overall system. This document can be used effectively to maintain the software development cycle and user manual will give a précised usage scenario of the application. I tried my best to remove all dependencies and present the concept of the application and implementation details. I believe that the reader will find it in order.

References

- [1] IETF, "RFC6749 - The OAuth 2.0 Authorization Framework," IETF, October 2012.
- [2] Auth0® Inc, "Auth0," <https://auth0.com/>.
- [3] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proceedings of the 20th International Conference on Very Large Data Bases*, September 1994..
- [4] A. Kumar, "Modern Information Retrieval: A Brief Overview," in *IEEE*, 2001.
- [5] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proceedings of the 20th International Conference on Very Large Data Bases, VLDB*, Santiago, Chile, September 1994..