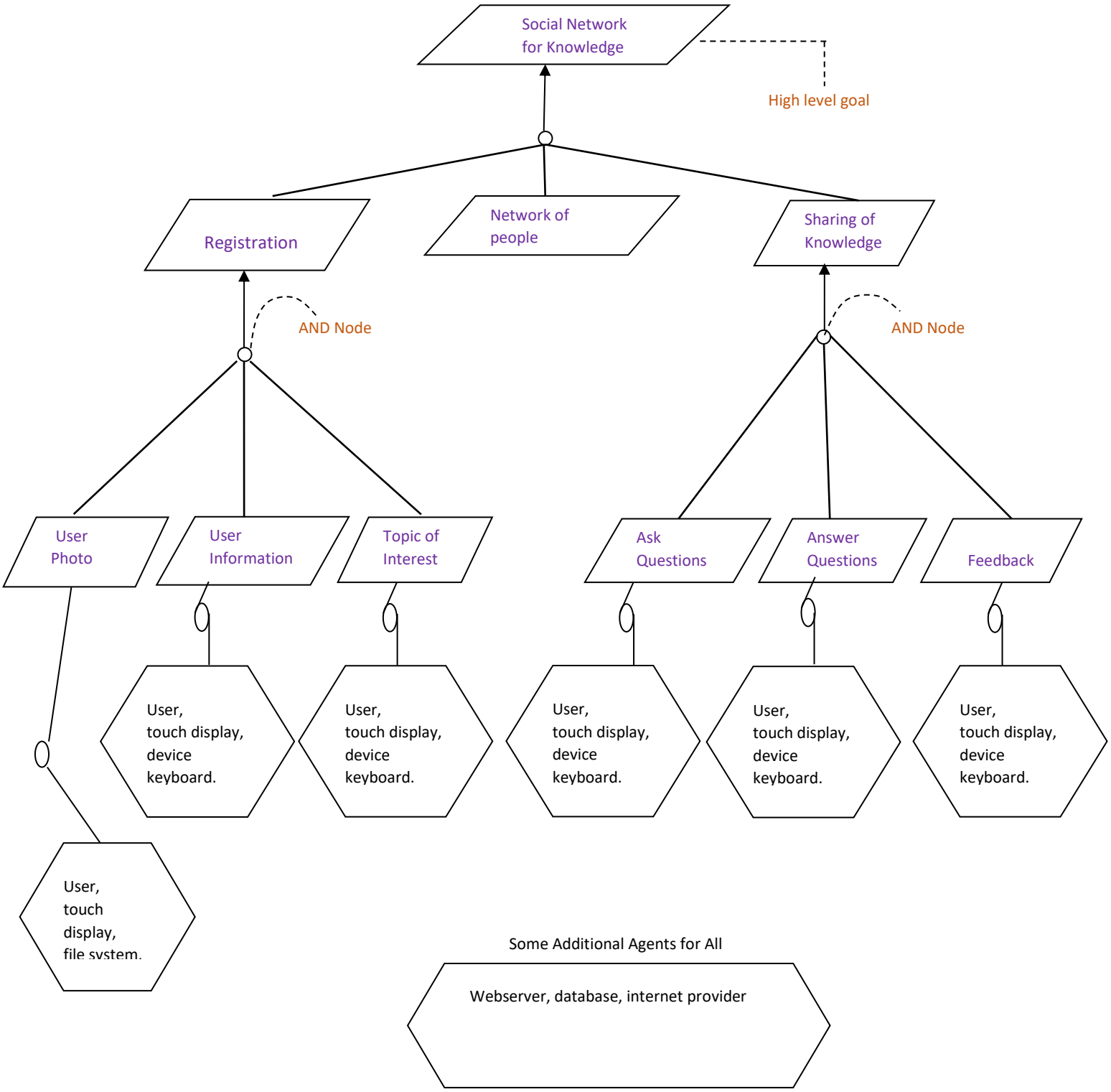
GOAL DIAGRAM

Application Quora



Data Dictionary:

NAME	Kind of Item	Description
User Photo	goal	User can upload a photo to their account.
User Information	goal	User can enter information about themselves.
Topics of Interest	goal	Users can follow particular topics they are interested in.
Ask Questions	goal	Users can ask questions regarding any topic.
Answer Questions	goal	Users can answer any question that has been asked.
Feedback	goal	Feedback can be given to previous answers.
Registration	goal	Users can create a profile.
Network of People	goal	User can create a social network of people with similar interests.
Sharing of Knowledge	goal	Knowledge shared through asking and answering questions.
Social network for knowledge	goal	Creating a social network where people can share and get to know topics in detail according to their interests.

EVALUATION:

QUALITY: The presented diagram is a quality starting point. It captures the essential focus of the desired product and begins to outline the goals required for success.

COMPLETENESS: Our goal diagram is not complete as it only captures the highest level goals of the software. There scope is not fully defined to a concrete set of requirements.
Example: Up-vote/down-vote/comment can be evolved from feedback.

USEFULLNESS: Our goal diagram is very useful as it serves the purpose of covering most important high level goals that need to be achieved for a successful launch.

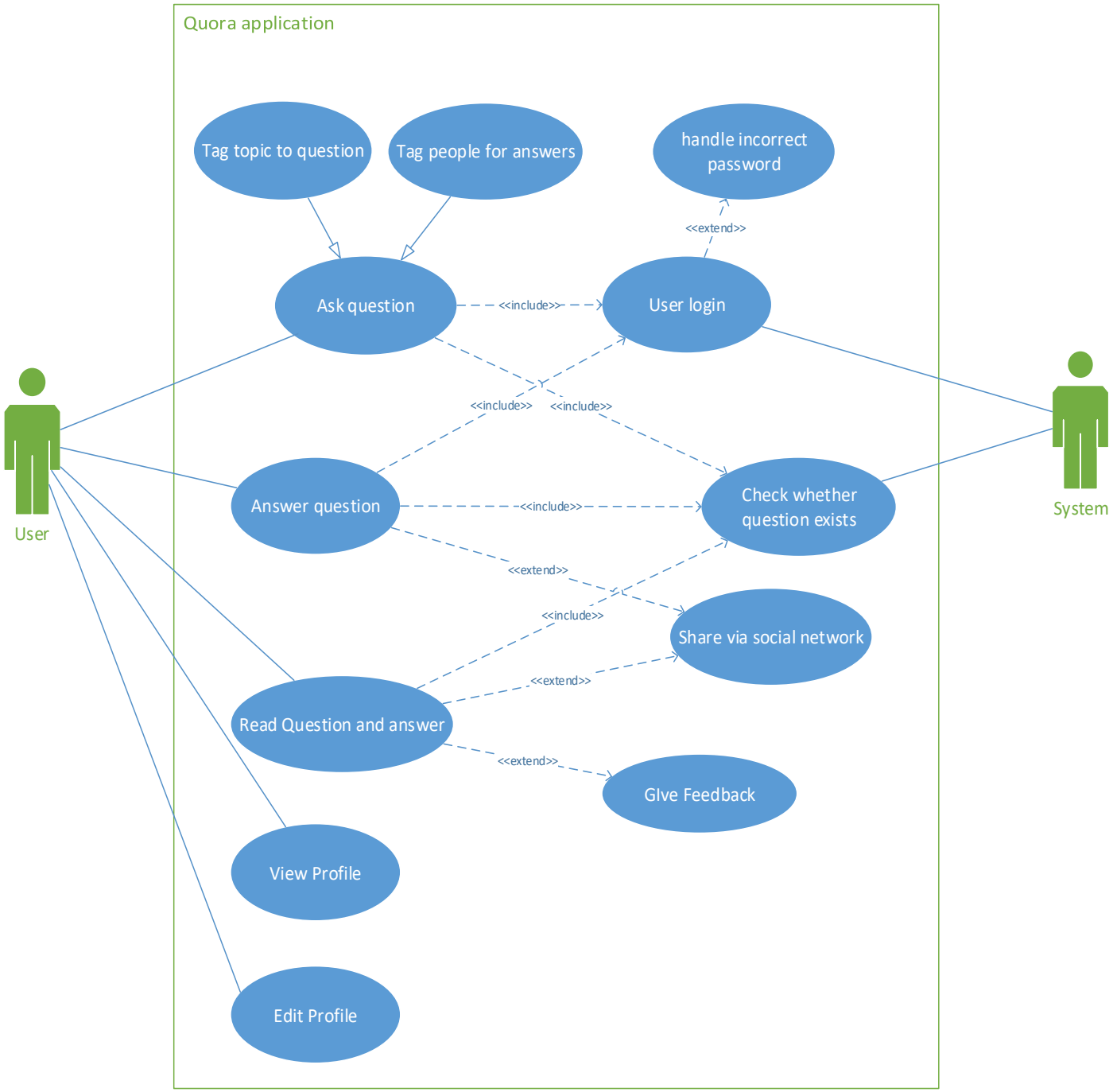
REVIEW:

Based on our review conducted on the goal diagram we modified the goal “Sharing of Knowledge” from its original form of “be able to ask questions, give answers and feedback.” We felt the original statement was too broad and that each aspect of it would serve well as a sub-goal of “Sharing of Knowledge”. In a similar fashion we had initially stated the ability to leave feedback as a 2nd level goal but felt it would serve better as a 3rd level goal with a 4th level being up-vote, down-vote and comment below it. We found through deliberation that it can be quite challenging to capture the core essence of a goal. Many of the goals we initially came up with

belonged lower in the tree, and it required a considerable amount of effort to not get caught up in the details and establish a clear goal hierarchy.

The application for this project is a question and answer application called ‘Quora’, where inquiries are asked, replied, altered and organized by its community of users. The goals are the following: The user can ask questions, answer questions, read questions and answers, and Registration.

Use Case Diagram



Use Case

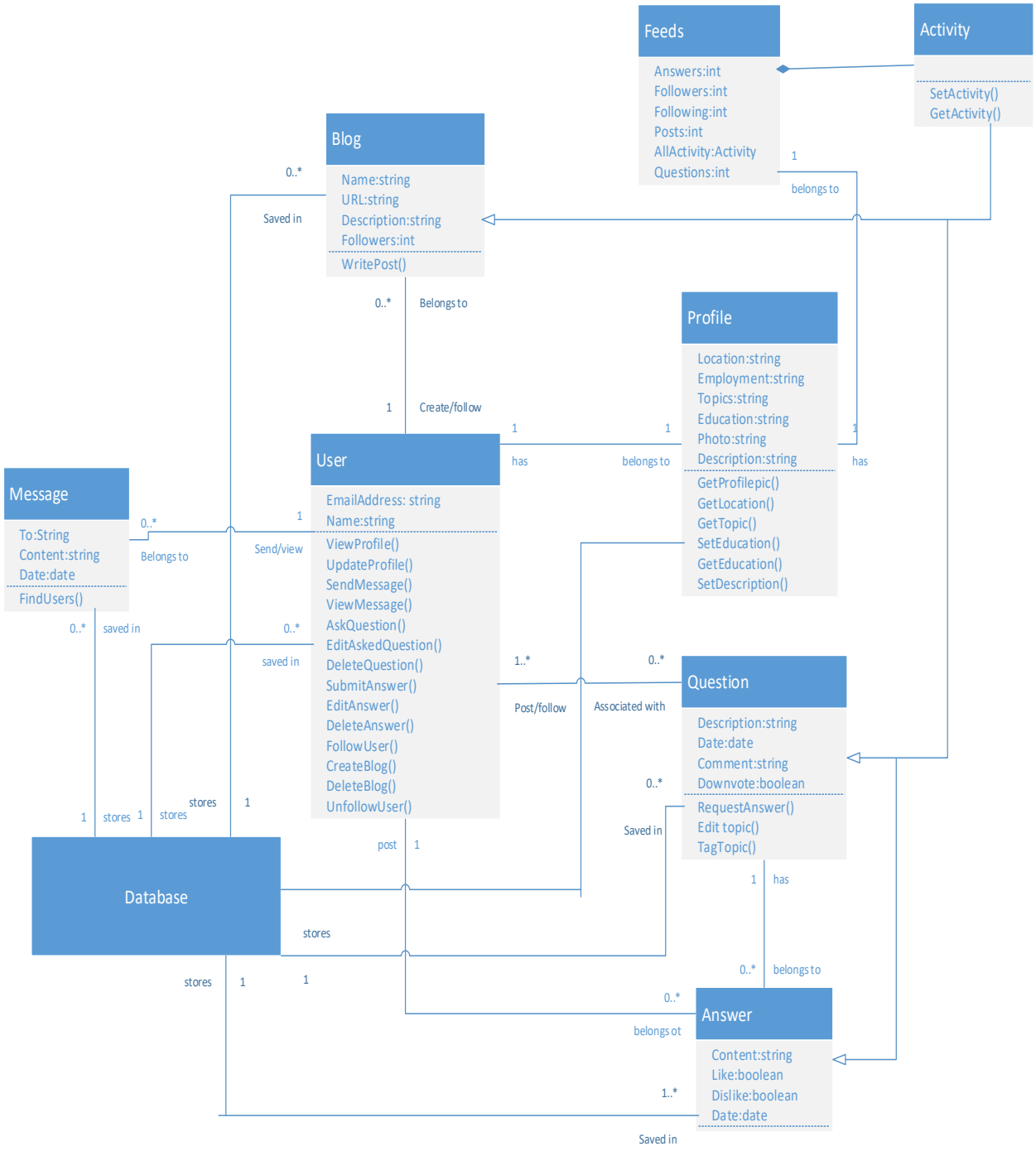
Use Case Name	Answer question
Actors	User, System
Author	Abhishek Mugalikar
Goal in context	Answer an already asked question.
Overview	User wants to answer a question.
Trigger	User trying to give answer to a question.
Preconditions	Users logs in successfully and there is a question to answer.
Post conditions	System records the answer and User should see the submitted answer on screen.
Typical course of events (Scenario): **Actor Action** **1.** User searches for question he wishes to answer. **3.** User selects the question. **4.** User starts to answer the question. **6.** User completes the answer. **7.** User submits his answer.	**System Response** **2.**System displays the question **5.** System keeps recording the answer and saves it after every word is entered. **8.** System records the answer.
Exceptions	1.At step 2: the question which user is interested in answering may not exist ,System will not display any question 2.At step 4: User can input any random text and submit the answer, System will recognize the random text entered and display a popup letting user know that the input was random
Frequency of use	Whenever the user knows the answer to a question.
Communication channel to actor	Via Smartphone and laptop/computer using internet

Use Case Name	Ask Question
Actors	User, System
Author	Abdul Hameed
Goal in context	To allow user to ask a question.
Overview	User searches for a question and if the question doesn't exist, user can ask the question
Trigger	User trying to ask a question.
Preconditions	User should have an account and logs in successfully.
Post conditions	The asked question should be displayed in the feed.
Typical course of events (Scenario): **Actor Action** **1.** User indicates question he wishes to ask. **3.** User submits the question. **4.** User tags a topic to question. **5.** User tags a list of people to the question.	**System Response** **2.** System validates and displays that question doesn't exist. **6.** System tags relevant topic and records the question. **7.** System sends notification to the tagged people. **8.** System displays the question asked.
Exceptions	At step 2, the system finds that the question that user wished to answer already exists.
Frequency of use	Whenever the user wants to ask a question.
Communication channel to actor	Via smartphone and laptop/computer using internet.

Use Case Name	User login

Actors	User, System
Author	Chinmaya Datta
Goal in context	To gain access to a quora account.
Overview	After successfully logging in, user can ask or answer any questions.
Trigger	User trying to login.
Preconditions	User should have an account to be able to login.
Postconditions	System should display home page of user.
Typical course of events (Scenario): **Actor Action** **1.** User opens Quora application. **2.** User enters username and password.	**System Response** **3.** System verifies login information and authorizes user.
Exceptions	Wrong username or password
Alternate Flow **Actor Action** **1.** User opens Quora application. **2.**User enters username and password **5.** User re-enters username and password.	**System Response** **3.** System verifies details and identifies username/password is wrong. **4.** Prompts user to enter details again. **6.** System verifies login information.
Frequency of use	Every time the application is used
Communication channel to actor	Via smartphone and laptop/computer using internet

Information model: Class diagram



Data Dictionary

Class Description and Entity Description

S .No	Entity	Description	Attributes	Data type	Description
1	User	Information about users of system	1.EmailAddress	string	Email address used for login
			2.Name	string	Name of user
2	Profile	Gives additional information about users	1.Location	string	Users location
			2.Employment	string	Users Employers name
			3.Topics	string	Users topic of interest.
			4.Education	string	Users highest education
			5.Photo	string	Users picture
			6.Description	string	Users general description
3	Question	Question asked by user for knowledge sharing	1.Description	string	Description will describe the question in detail
			2.Date	date	Date on which question was posted for knowledge sharing
			3.Comment	string	User can post comment related to question
			4.Downvote	boolean	Question will be shown to few people
4	Answer	Answers that will be posted by users to the related questions	1.Content	string	The answer that user provides for question
			2.Like	boolean	Count of likes for answer
			3.Dislike	boolean	Count of dislike for answer
			4.Date	date	Date on which answer was posted
5	Profile	Profile will have details of users	1.Location	string	Users can input his location like city, state country.
			2.Employment	string	Users current Employment
			3.Topics	string	Topics that user is interested in
			4.Education	string	Users highest degree
			5.Photo	string	User can upload a picture for his profile
			6.Description	string	A description about user
6	Blog	User can create and write blog .blogs can be topics of his interest.	1.Name	string	Name of blog
			2.URL	string	Unique string for URL of blog
			3.Description	int	Details of blog where in user can write in detail about blog
			4.Followers	int	Count of followers of blog
7	Message	Message that are sent and received by user	1.To	string	Name of person whom message is to be delivered

			2.Content	string	Content of message sent by one user to other
			3.Date	date	Date on which message was sent
8	Feeds	Feeds contain count of all activity that user performs and it also has all activity that displays all the activity.	1.Answers	int	Count of number of answers answered by user.
			2.Questions	int	Count of number of question asked by user
			3.Followers	int	Count of number of followers user has
			4.Following	int	Count of number of users user is following
			5.Posts	int	Count of posts submitted by user
			6.All activity	int	Lists all activity that a user has performed.
9	Activity	Lists all activates that user has performed. the activities include question posted, answer posted, blog details			

Methods

Class Name	**Method Name**	**Method Description**
User	1.ViewProfile()	Method is used by user to view profile details like education, employment
	2.UpdateProfile()	Method is used by user to update profile details like education, employment
	3.SendMessage()	This method is used by user to send message to other users
	4.View Message()	This method is used by user to view message that are sent by users
	5.AskQuestion()	This method is used by user to ask/post new question
	6.EditAskedquestion()	This method is used by user to edit question that he has posted
	7.DeleteQuestion()	This method is used by user to delete question which he has posted
	8.SubmitAnswer()	This method is used by user to submit a answer to question
	9.EditAnswer()	This method is used by user to Edit his answer to question
	10.DeleteAnswer()	This method is used by user to delete his answer to question
	11.FollowUser()	This method is used by user to follow other users.
	12.CreateBlog()	This method is used by user to create blog
	13.DeleteBlog()	This method is used by user to delete blog
	14.UnfollowUser()	This method is used by user to stop following other users.
Question	1.RequestAnswer()	If the question has been already asked and user wants to ask the same question again user can request answer.
	2.EditTopic()	User can edit the tagged topic relevant to question
	3.TagTopic()	User can tag topics that are relevant to question

Message	1.FindUsers()	This method is used to find the users whom a user wants to send message.
Blog	1.WritePost()	User can write post that has content relevant to blog.
Profile	1.GetProfilePic()	This is used to retrieve profile picture of user
	2.GetEducation()	This is used to retrieve education from user.
	3.GetLocation()	This is used to display the education of user
	4.GetTopic()	This is used to retrieve topics of interest for user
	5.SetTopic()	The user can select among multiple topics and set topics of his interest
	6.SetDescription()	The user can set a brief description about him.
Activity	1.SetActivity()	This sets the recent activity. Recent activity includes the all the activities like posting question ,answer and writing posts
	2.GetActivity()	This gets all the activities that the user performs.

Relationships

Class Name	**Type of Relationship**	**Comments**
Activity	Inherits from Blog, Question and Answer	Activity displays all recent actions that the user has performed so it inherits from Blog, Question and Answer class.
Profile Activity	There is a composition relationship between Profile and Activity Classes.	Activity class cannot exist without Profile class. Profile class has a section called activity which displays all activity that user has performed.

Evaluation:

Quality: The quality of use cases and information model can be labeled as good, since we have integrated all the primary functions of the application into them. By observing these two diagrams and the use case specifications we developed, one can easily understand how the application works and can identify that this application serves its purpose. We have done several reviews before fixing on the final versions of use cases and UML diagrams. Each review gave a clear perspective of the application and gave us new ideas on how to design the models such as the relationships between the entities and some inconsistencies in the UML diagram." Give Feedback" was a goal at top level in the goal diagram of previous document. We changed it in use case diagram, that goal extends 'Read Question and answer' use case bubble. Only after reading question and answer, the user can give feedback. This was used to improve the overall quality and make use case easy to understand

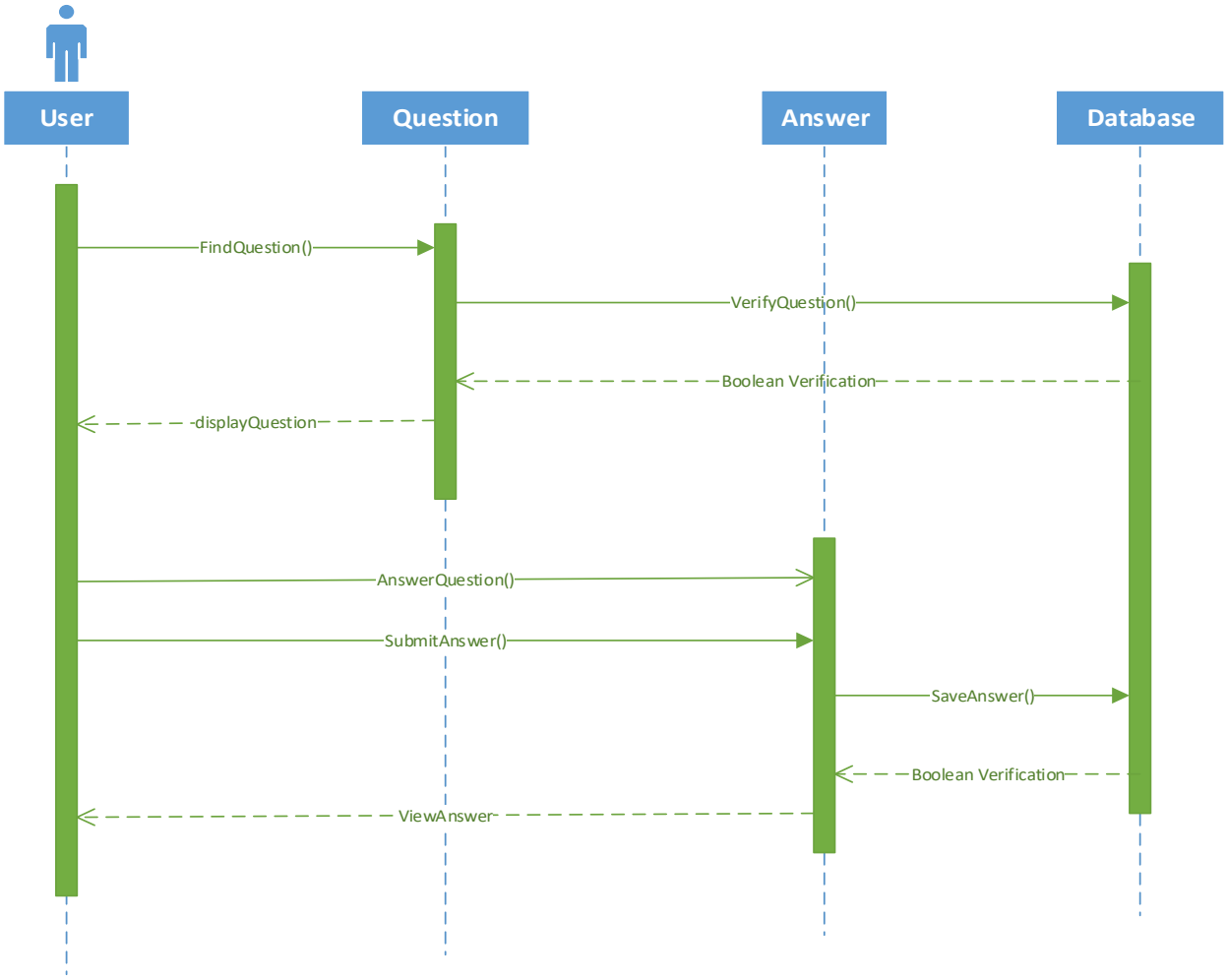
Completeness: We cannot say that our use cases and UML diagram are complete. We have included everything we knew and also did some research before completing them. The diagram does have potential to be expanded. There is also scope for adding many new functionalities and requirements in the future. Class diagram does not cover all methods, there is scope to add new methods. For example the user class can have method to Like (), Dislike (), SubscribetoBlog (), UnSubscribeBlog().New classes could also be added like Topic class which can have information related to topics of interest for user. More use case bubble could also be added for "creation of blog", "Delete Question".

Usefulness: Our use cases and information model are very useful because we have developed them after identifying and studying all the goals and the true purpose of the application. Taking this information into consideration, we have designed our use case diagram with all the actions done by user and the system, and the UML class diagram with all the required entities, their attributes, behavior and relationship amongst them. Our information model also consists of a data dictionary which contains explanation for all the terms we have used in our UML diagram, such as the entity names, attributes and relationships etc. The use cases and information model we developed helps in understanding the functions and purpose of the application with ease. Working in a team, members decided what functionality was most important and necessary to be mentioned in the use cases to enhance overall comprehension, how the use case should be organized, how all actors worked together to meet the goal.

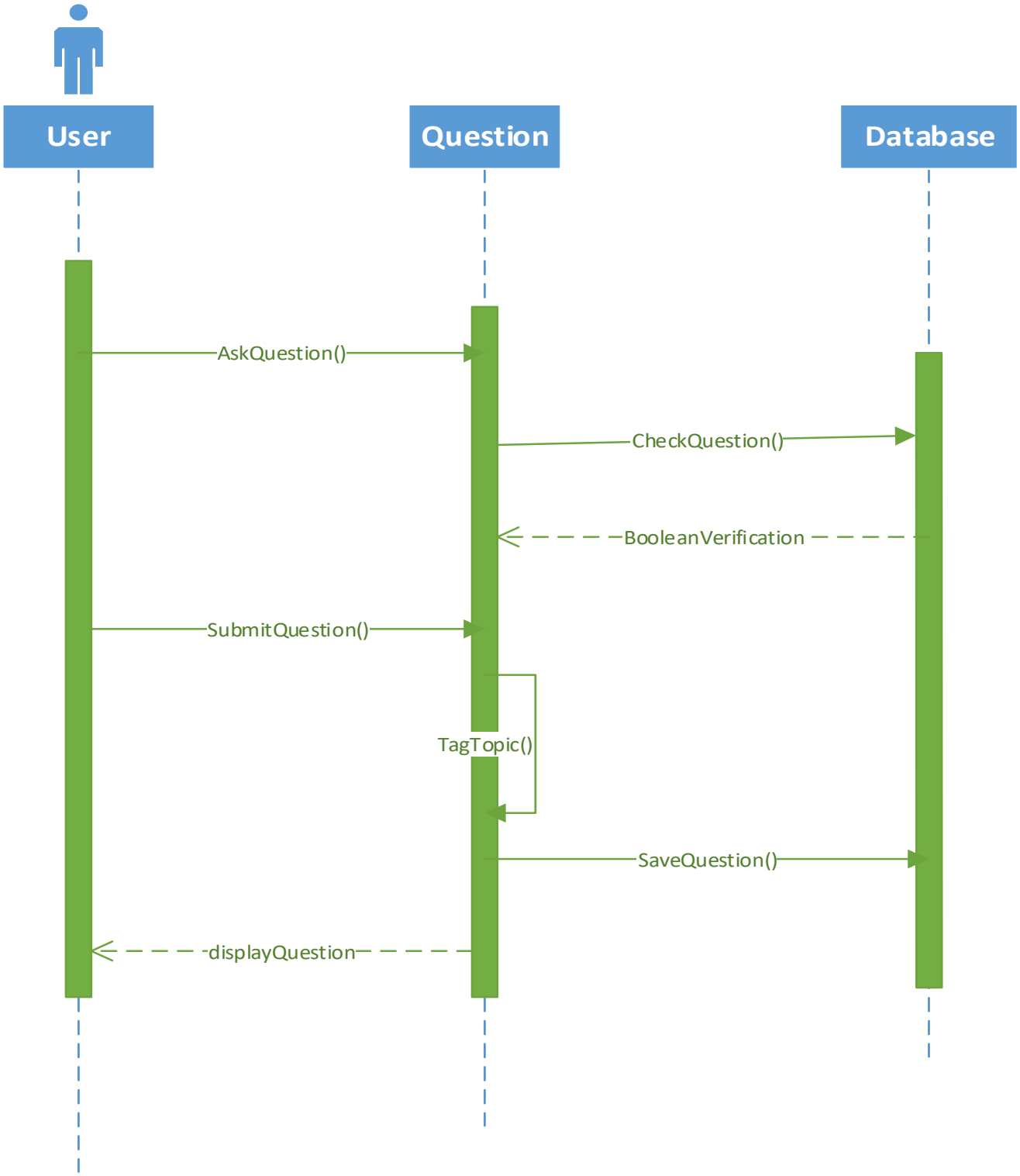
Sequence Diagram:-

For all Sequence Diagram we have used User Class as a Class and as an actor. User class has been mentioned in Class diagram.

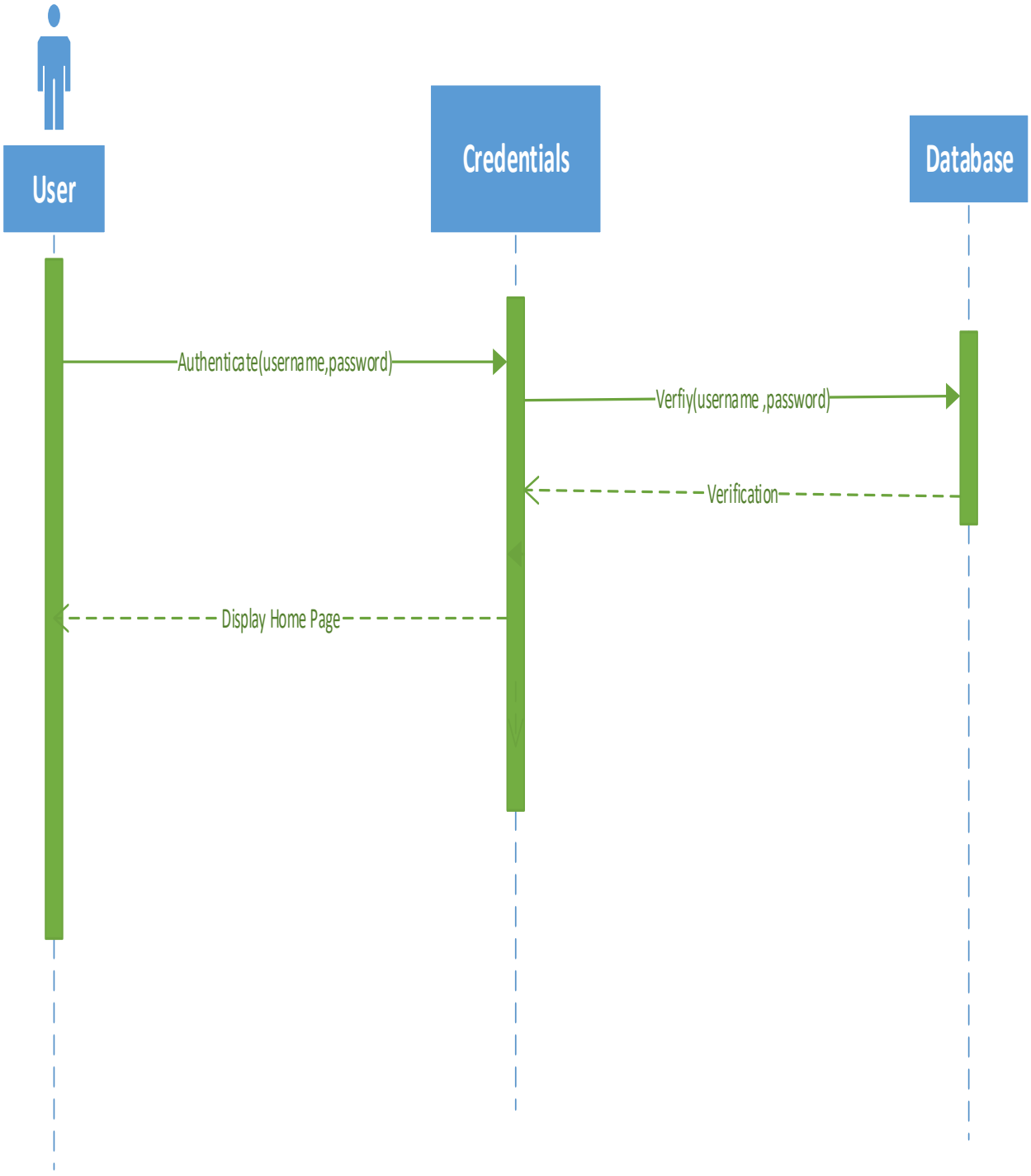
Answer Question



Ask Question Sequence Diagram



User Login Sequence Diagram



Requirements:-

1) The user has to create his account and profile.

2) The user reads the questions and answers that are present in the website; he can answer the questions that are posted on the website by other users.

3) The user can search the questions that are already answered by other users and he can also post the question for which he needs an answer.

4) He can share the questions and answers via social network.

5) The user tags people that are interested in the answer and tag the topic of the questions.

6) User should be able to give feedback to existing answer.

For User Login Sequence Diagram we changed the Exercise B class Diagram to add new Class "Credentials". The newly added class details are

S .No	Entity	Description	Attributes	Data type	Description
1	Credentials	Log in information about users of system	1.username	string	Username for login
			2.password	string	Password of user

S .No	Entity	Methods	Description
1	Credentials	1.Authenticate(username,password)	Method is used to authenticate user with username and password
2	Credentials	2.Verfiy(username,password)	Method is used to verify whether the username and password is present in database.

Relationship with existing classes

S .No	Relationship	Description
1	User — 1 has / belongs to 1 — Credentials	User can have only one credentials and credentials belong to only one user. there is one to one relationship between the two
2	Credentials — 0..* saved in / stores 1 — Database	Credentials are saved in database.

Methods added in Class Diagram

While creating Sequence diagram we added methods in Existing Classes Below are the method added to existing Class diagram.

Class	Method	Description
Question	1.FindQuestion	Method is used to find Question.
	2.ViewAnswer	Method is used to view answer.
	3.SubmitQuestion	Method is used to Submit question for saving it into database.
	4.Verfiy Question	Method is used to check whether question exists in database.
	5.SaveQuestion	Method is used to save question in database
Answer	1.Save Answer	Method is used to save answer in database

Traceability Table

Traceable Item	Kind of Item	Item traced from	Item Traced to
Ask Question	Use case	Requirement 3, User [Actor]	UML ask Question Sequence Diagram
Answer Question	Use case	Requirement 2, User [Actor]	UML Answer Question Sequence Diagram
Read Question and Answer	Use case	Requirements 2, User [Actor]	UML ask Question Sequence Diagram
View Profile	Use case	Requirement 1, User [Actor]	
Edit Profile	Use case	Requirement 1, User [Actor]	
User Login	Use case	User [Actor],Requirement 1	UML User Login Sequence Diagram
Give Feedback	Use case	User [Actor],Requirement 6	
Check Whether Question Exists	Use case	Requirement 2, User [System]	Ask Question use case and Answer question UML Sequence Diagram
Share via Social network	Use case	Requirement 4 , User [Actor]	
Tag Topic to Question	Use case	Requirement 5	UML Sequence diagram Ask Question
Tag People for answers	Use case	Requirement 5	UML Sequence Diagram
Handle incorrect passwords	Use case	Requirement 1	
User	Class Entity	User Login Use Case, Create Account	UML Sequence Diagram
Credentials	Class Entity	User Login Use Case, Create Account	UML User Login Sequence Diagram

username	Attribute-Credentials class	User Login Use Case	UML Class Diagram Credentials Class
password	Attribute-Credentials class	User Login Use Case	UML Class Diagram Credentials Class
FindQuestion()	Method of Question Class	UML class diagram, Answer Question Use Case.	Design, Source Code, UML Sequence Diagram Ask Question, Answer Question
AskQuestion()	Method of User Class	UML class diagram, Ask Question Use Case, Requirements	UML Sequence diagram Ask Question.
ViewProfile()	Method of User Class	UML diagram User Class, View Profile Use Case	User[Actor]
ViewAnswer()	Method of Question Class	View Question and Answer Use case	UML Sequence Diagram
UpdateProfile()	Method of User Class	Edit Profile Use case, UML Diagram user class	User[Actor]
SendMessage()	Method of User Class	UML Class Diagram User Class, Send Message Use Case	
Authenticate	Method of Credentials Class	UML Class Diagram Credentials Class	User[Actor],UML Sequence Diagram
View Message()	Method of User Class	UML Class Diagram User Class, Message Class, View Message Use Case	
EditAskedQuestion()	Method of User Class	UML Class Diagram User Class, Change Question Use Case	
DeleteQuestion()	Method of User Class	Change Question use Case, Delete Question Use case	
SubmitAnswer()	Method of User Class	User[Actor], Answer Question use case	UML Answer Question Sequence Diagram
EditAnswer()	Method of User Class	Edit Answer Use Case, User[Actor]	
DeleteAnswer()	Method of User Class	Delete Answer Use Case User[Actor]	
FollowUser()	Method of User Class	UML Class Diagram	
CreateBlog()	Method of User Class	UML Class Diagram	
DeleteBlog()	Method of User Class	UML Class Diagram	`
UnfollowUser()	Method of User Class	UML Class Diagram	
EmailAddress	Attribute	User Class	User[Actor]
Name	Attribute	User Class	User[Actor]
Question	Class Entity	Ask Question Use Case, User[Actor],Answer Question Use case	UML Sequence Diagram Ask Question ,Design

VerfiyQuestion()	Method of Question Class	UML Class Diagram, Ask Question, Answer Question Use Case	UML Answer Question Sequence Diagram
SaveQuestion()	Method of Question Class	UML Class Diagram, Ask Question, Answer Question Use Case	UML Ask Question Sequence Diagram
SubmitQuestion()	Action	UML Class diagram, Ask Question use Case	UML Ask Question Sequence Diagram , Design
RequestAnswer()	Method of Question Class	Answer Question Use Case	
EditTopic()	Method of Question Class	Tag topic to Question Use case	
TagTopic()	Method of Question Class	Ask Question Use Case	UML Ask Question Sequence Diagram
Description	Attribute -Question Class	Ask Question Use Case	
Date	Attribute - Question Class	Question Class	
Comment	Attribute -Question Class	Question Class	
Downvote	Attribute - Question Class	Question Class	
Answer	Class, Actor in Sequence Diagram	Requirements, Use case- Answer Question, View Question and Answer	UML Sequence Diagram Answer Question and Ask Question
SaveAnswer()	Method	UML Class Diagram	UML Sequence Diagram Answer Question
Answer	Attribute - Answer Class	Answer Question Use case, Answer Class	UML Sequence Diagram Answer Question, Database
Like	Attribute - Answer Class	View Question Answer Use Case	User[Actor]
Dislike	Attribute - Answer Class	View Question Answer Use Case	User[Actor]
Date	Attribute - Answer Class	View Question Answer Use Case	
Profile	Class	Edit Profile ,View Profile ,Delete Profile Use Case, Create User Use Case	
Location	Attribute - Profile Class	View Profile, Edit Profile Use Case	User[Actor]
Employment	Attribute - Profile Class	View Profile, Edit Profile Use Case	User[Actor],Database
Topic	Attribute - Profile Class	View Profile, Edit Profile Use Case	User[Actor], Database
Education	Attribute - Profile Class	View Profile, Edit Profile Use Case	User[Actor], Database

Photo	Attribute - Profile Class	View Profile, Edit Profile Use Case	User[Actor], Database
Description	Attribute - Profile Class	View Profile, Edit Profile Use Case	User[Actor]
GetProfilePic()	Method –Profile Class	View Profile, Edit Profile Use Case	User[Actor]
GetLocation()	Method –Profile Class	View Profile, Edit Profile Use Case	User[Actor]
GetTopic()	Method –Profile Class	View Profile, Edit Profile Use Case	User[System]
SetEducation()	Method –Profile Class	View Profile, Edit Profile Use Case	User[Actor]
GetEducation()	Method –Profile Class	View Profile, Edit Profile Use Case	User[Actor]
SetDescription()	Method –Profile Class	View Profile, Edit Profile Use Case	User[Actor]
Blog	Class	Blog related use Case, create Blog, Delete blog	
Name	Attribute-Blog Class	Create Blog Use case, Write Blog Use case, Blog Class	
URL	Attribute-Blog Class	Create Blog Use case, Blog Class	
Description	Attribute-Blog Class	Create Blog Use case, Blog Class	
WritePost()	Method-Blog Class	Write Blog Use case, Blog Class	
Message	Class	UML Use case Diagram	
To	Attribute-Message Class	View Message, Send Message Use case	User[Actor]
Content	Attribute-Message Class	View Message, Send Message Use case	User[Actor]
Date	Attribute-Message Class	View Message, Send Message Use case	
FindUsers()	Method Message Class	Send Message ,View Message Use case	
Database	Actor	Requirements	UML Class diagram
Feeds	Class	View Profile, View Activity Use case	
Answers	Attribute-Feeds Class	View Profile, View Activity Use case, ,User[Actor]	
Followers	Attribute-Feeds Class	View Profile, View Activity Use case, ,User[Actor]	
Following	Attribute-Feeds Class	View Profile, View Activity Use case, ,User[Actor]	
Posts	Attribute-Feeds Class	View Profile, View Activity Use case, ,User[Actor]	
AllActivity	Attribute-Feeds Class	View Activity Use case, ,User[Actor]	

Questions	Attribute-Feeds Class	View Profile, View Activity Use case, ,User[Actor]	
Activity	Class	View Profile, View Activity Use case, ,User[Actor]	
SetActivity()	Method Activity Class	View Profile Use Case, Edit Profile Use Case, UML Class Diagram, ,User[Actor]	
GetActivity()	Method Activity Class	View Profile Use Case	User[System]
Boolean Verfication	Message	UML Ask Question Sequence Diagram	
displayQuestion	Message	UML Ask Question Sequence Diagram	
Display home page	Message	UML Sequence Diagram User Log In	
View Answer	Message	UML Ask Question Sequence Diagram	
Verification	Message	UML Sequence Diagram User Log In	