

# ASE Project Increment4

## Group# 12: Project "Ask UMKC"

### Group Members:

- Sravan Kumar Appana
- Vikesh Padarathi
- Architha Mukka
- Manikanta Maddula

### Introduction:

Name of the application “Ask UMKC”.

The App is about questionnaires which can be posted by any user and any registered user can answer the question. The questions need not to be in specific area like finance, course and sports etc., it can be of any topic related to university activities like events, career, etc. Student who wants to join UMKC will have many doubts like ‘last date for paying fee!’, ‘where is a particular event?’, ‘Student associations ’etc. It is easy to drop a mail to the authorities asking for details but if answer is needed on weekends and on holidays, it’s difficult. But with this application answer can be found at any time.

### Objective:

An android application which help students of UMKC in clearing their doubts in all domains like fees, course, sports etc. Answers can be replied by other students or any other registered user.

### Features:

Users can login to app using their Google account or they can sign up for the account. After successful login, user navigates to a page with ask button and when slided left in home page, app displays various fields like ‘Academic’, ‘Nonacademic ‘, ‘Admissions’, ‘Career’, ‘update profile’, ‘My questions’ and FAQ. User can select a particular field to question and answer. When a particular field is selected, questions are displayed as a list in reverse chronological order. User can scroll for different questions. Clicking on a question will display question and multiple answers for it. User has a feasibility to answer in the same page or the user can use the ask button in home page for posting a question and answer. Home page consists of questions as a

list view, when clicked on a specific question where it navigates to a page with multiple answers for that question and a button where user can post a answer. There is a vote up option to decide appropriate answers for every question in the form of like and dislike buttons. Moreover, users can follow questions using follow button if needed.

### Existing Services or API:

User can sign up in to the application using their Google account. Mash up of Google sign in with the application is achieved with the following API:

<https://www.googleapis.com/auth/userinfo.profile>

Mongo Db is a service used for retrieving and updating data.

<https://mlab.com>

### Design of features:

#### Wireframes:-

#### Login Screen:

ASK UMKC

User Name

Password

LOGIN REGISTER

Login With Google

UMKC

## Register:

Ask UMKC

First Name
Last Name
Phone Number
Email Id
User Name
Password

Register

## Homepage:

Application Name

Question 1:  
Answer 1:

Follow

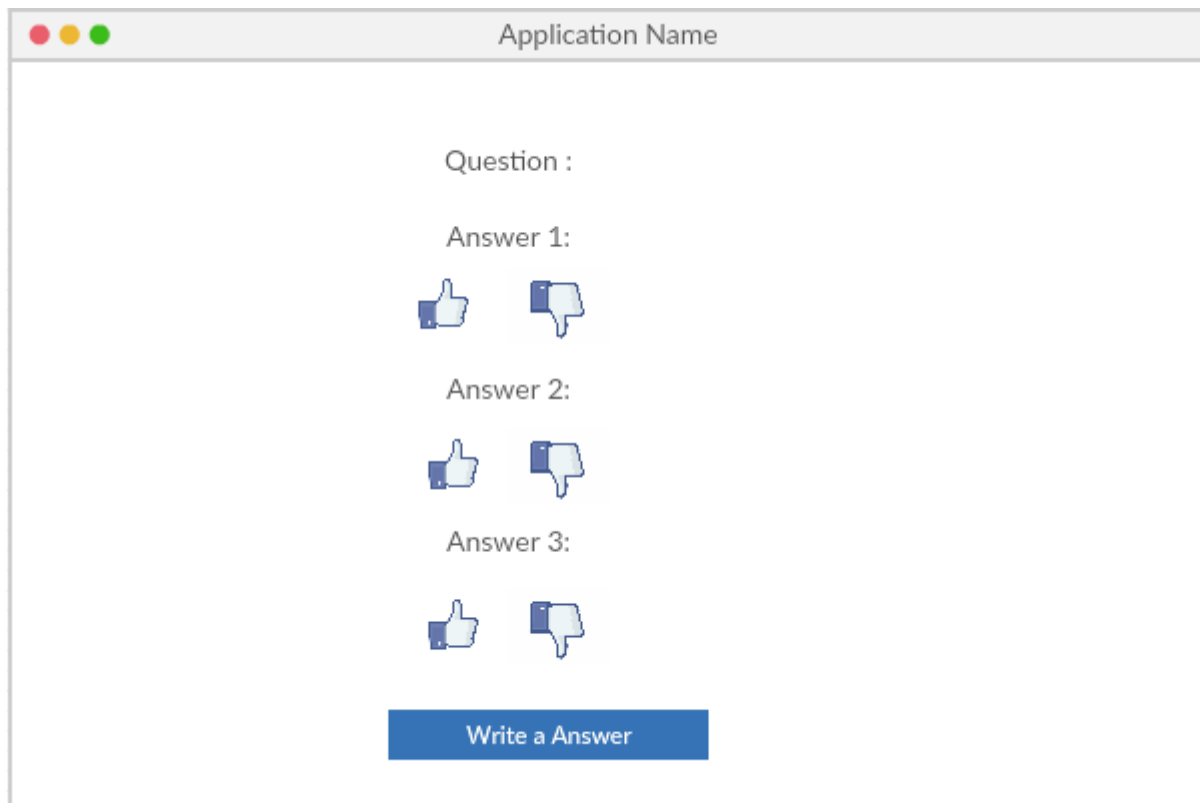
Question 2:  
Answer 2:

Follow

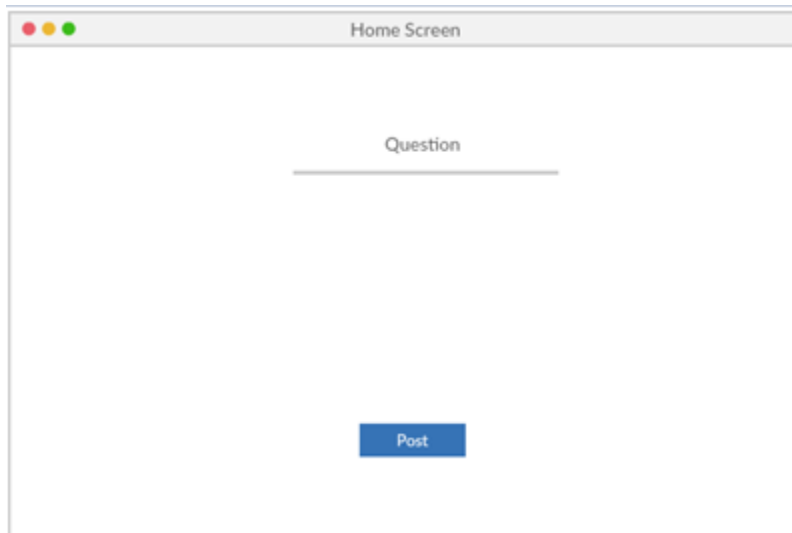
Question 3:  
Answer 3:

Follow

ask




**Ask question:**



## Categories:

Application Name



User Name

Academics

Non- Academics

Admissions

Career

Profile

My Questions

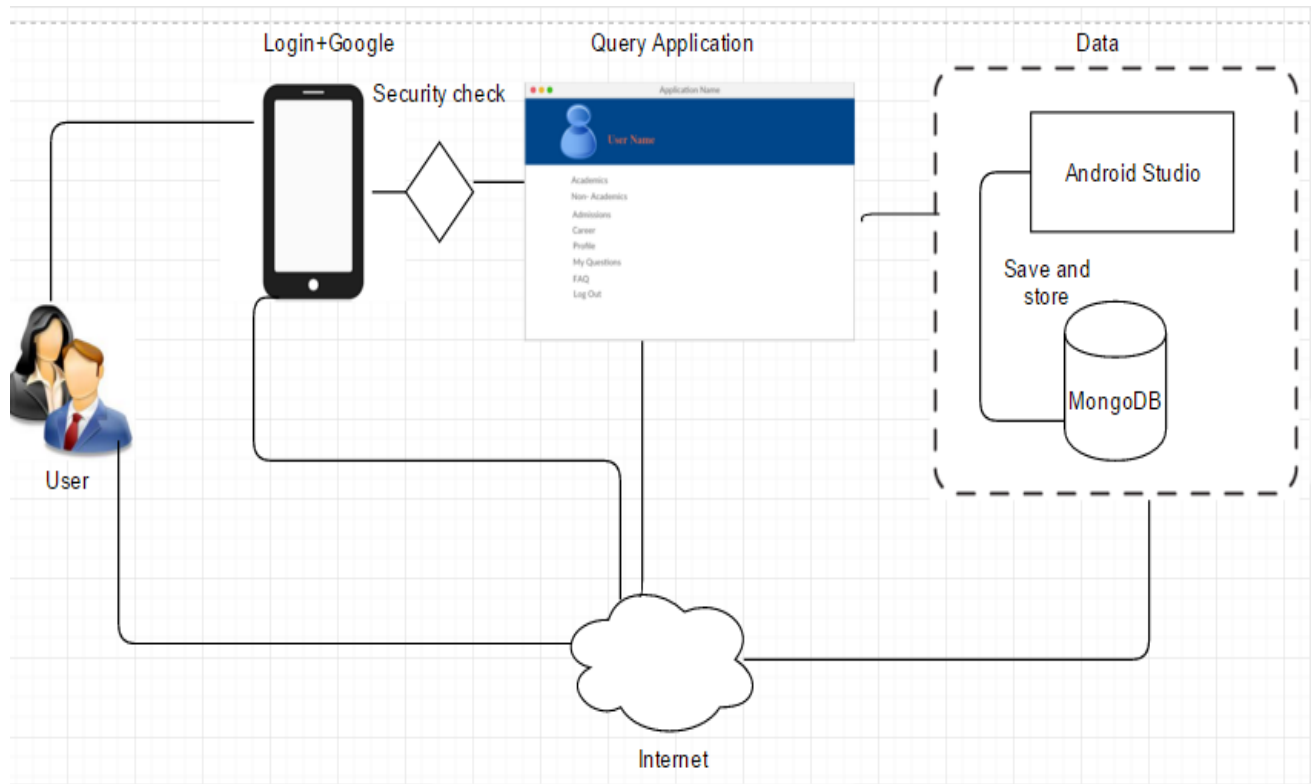
FAQ

Log Out

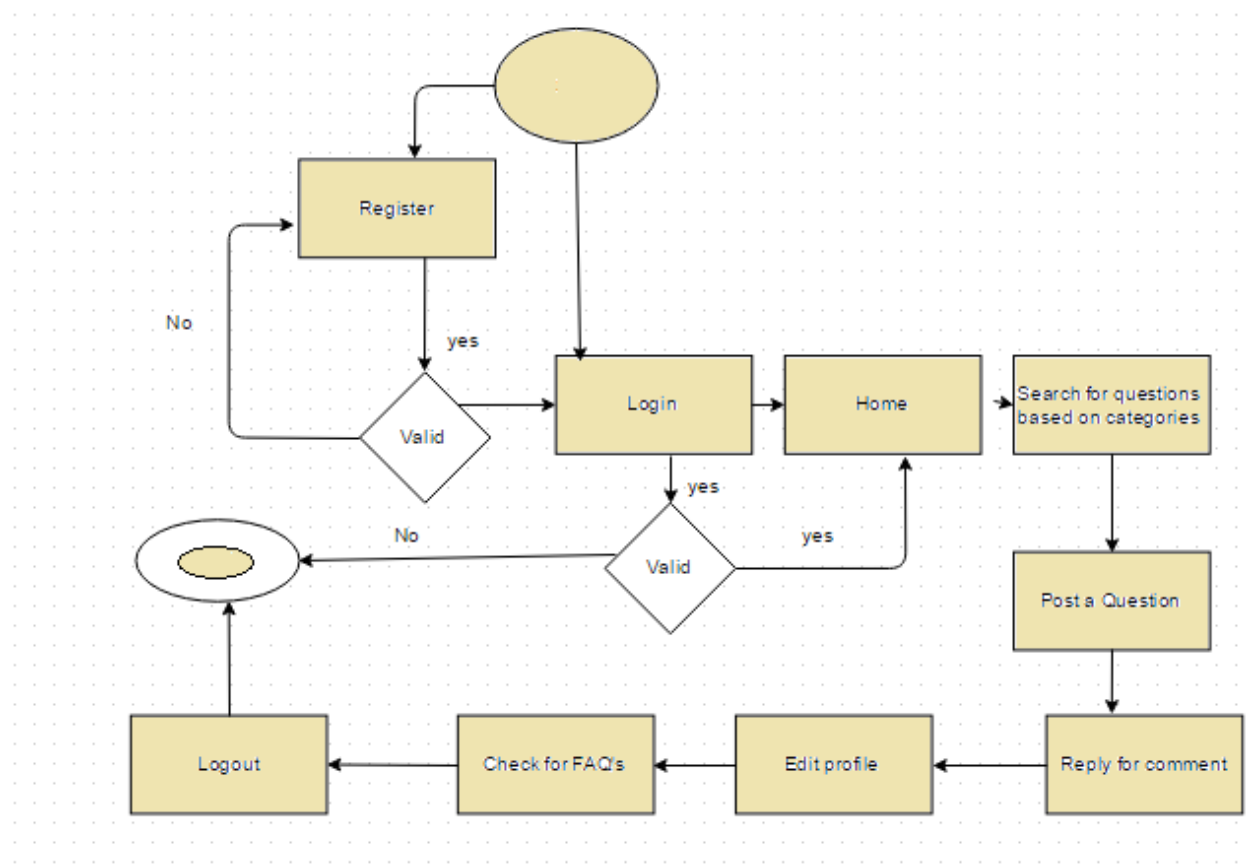
## Profile update:



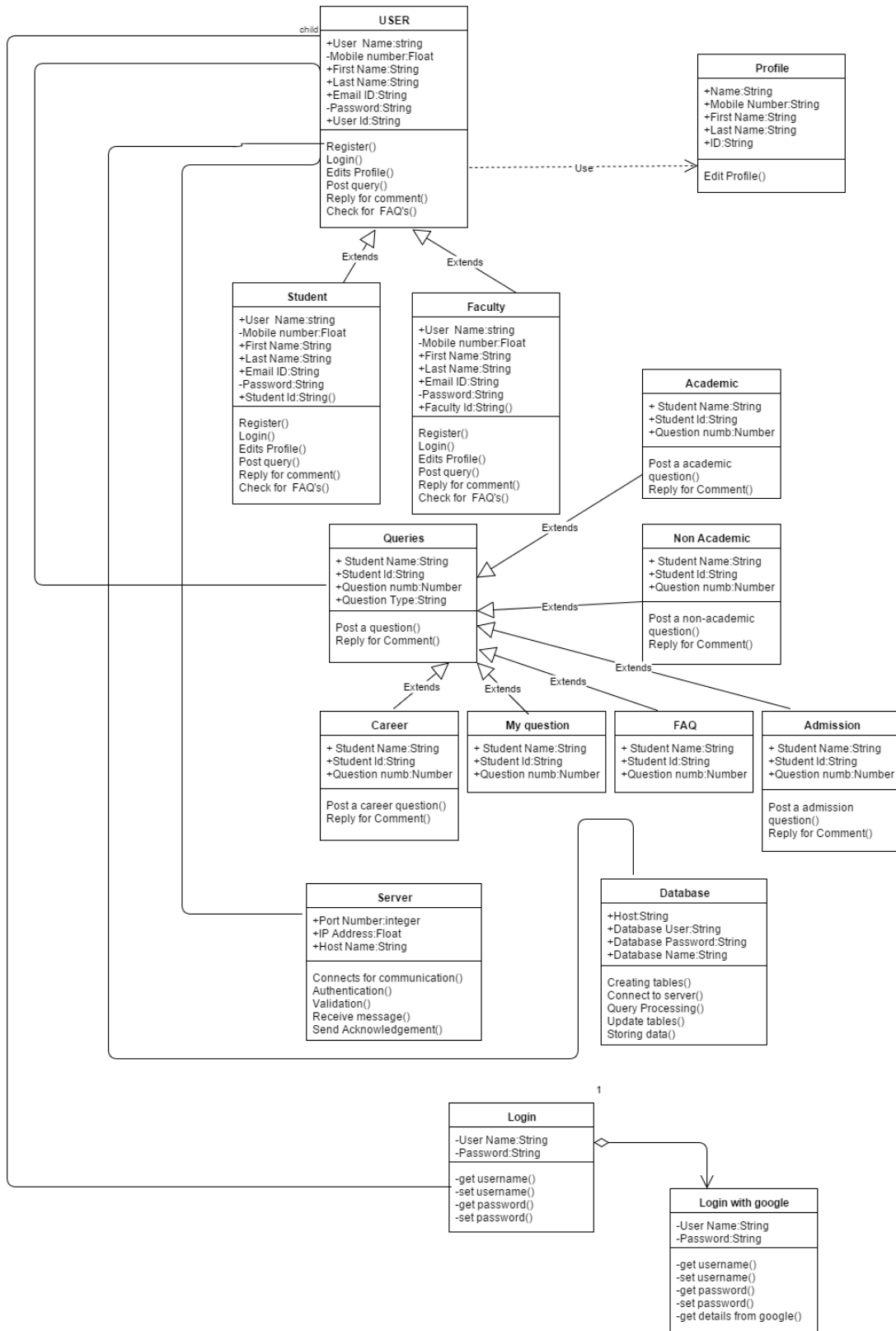
## Architecture Diagram:



## Activity Diagram:

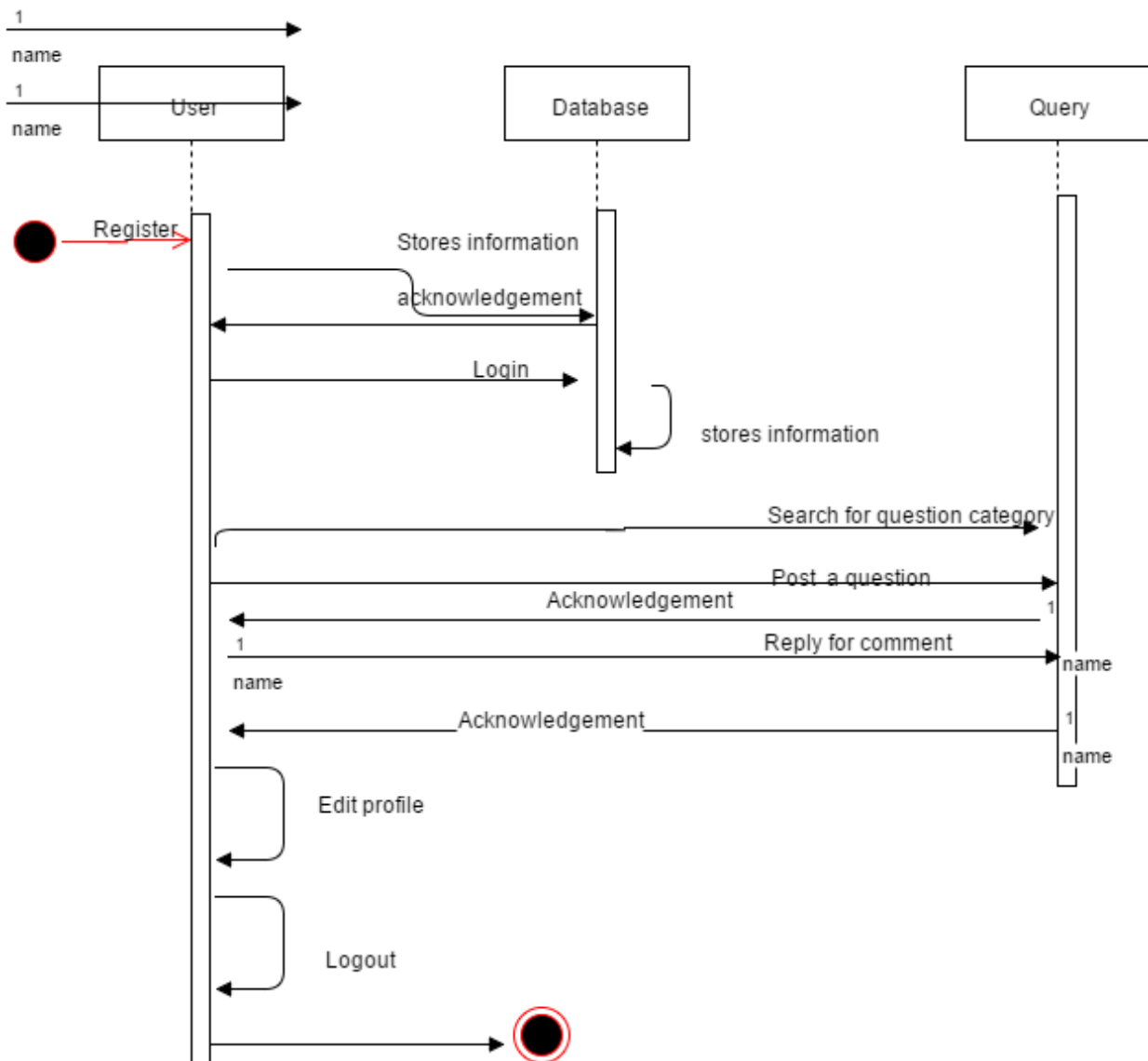


## Class Diagram:





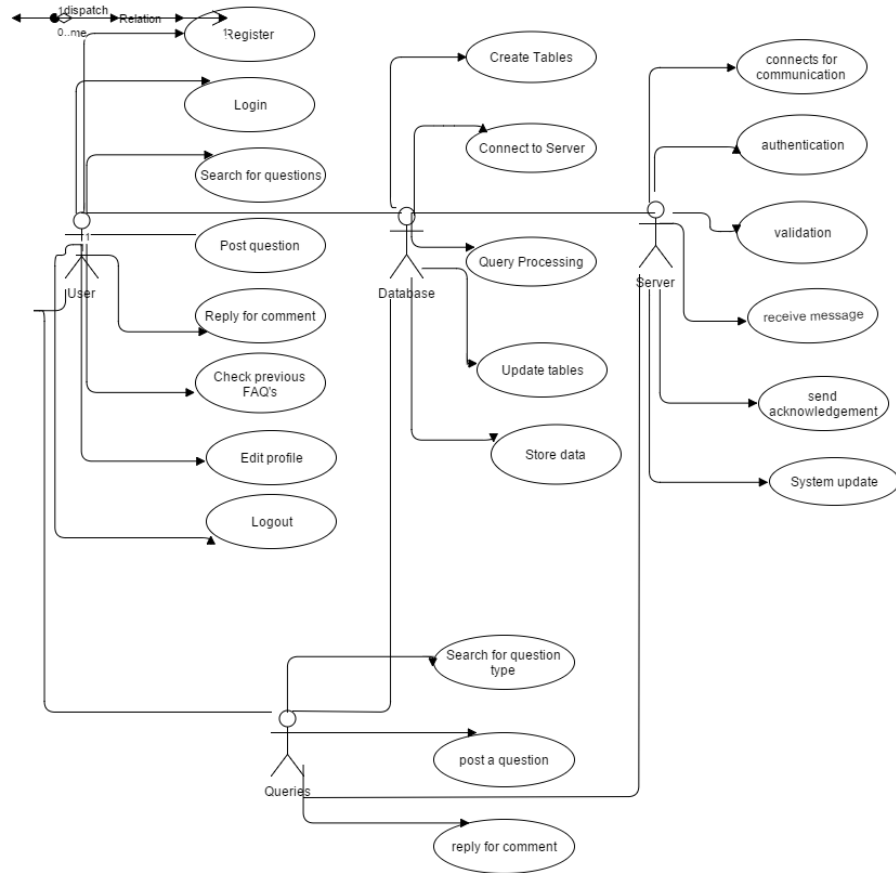
## Sequence Diagram:



## User Stories:

- As a user, I must login in to the account to access the application.
- As a user, I must register in to the account to access the application.
- As a user, I must sign up through Google account.
- As a user, I must post a question and answer and like, unlike and follow a answer specified to that question.
- As a user, I can select categories in the menu.
- As a user, I can search for questions required.
- As a user, I can update my profile.
- As a system, I must allow the user to register, login, sign up only if the credentials are correct.
- As a system, I must be able to access the database which shows the details of questions, answers and saves user profile.
- As a system, I must display question and answer in home screen page.
- As a system, I must allow to like, unlike and follow a answer.
- As a system, I must analyze the question and display multiple answers for that question.
- As a system, I must allow to select categories in the menu.
- As a system, I must allow the user to select categories in the menu.
- As a system, I must display search results.
- As a system, I must convert speech to text.

## Use Case :



## Testing:

### 1. Login/register:

S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	User should be able to login/New-user should be able to register	User would be able to enter the E-mail and password and click login to enter to application	User should be able to login	User is logged-in.
2.	User should be able to login/New-user should be able to register	New-User should click the Sing-up button to register, to navigate to registration page	New-users are directed to register page.	New-Users are able to view the registration page.
3.	User should be able to login/New-user should be able to register	User should enter E-Mail and Password and are validated for the users.	User details are authenticated.	User is able to login only when accurate details are submitted
4.	User should be able to login/New-user should be able to register	When user enters wrong details an error would pop up.	User credential are to be validated and if wrong a pop-up would be display	User would be able to view a message if incorrect details are entered.

### 2. Home Screen:

S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	Registered User should be able to ask a question or search for answers	User enters question and answer to post.	Users should be able to enter question and answer for posting	Users should be able to enter question and answer for posting.
2.	Registered User should be able to ask a question or search for answers	Users can select categories based on the doubt.	Users can view the categories	Users are able to view categories.
3.	Registered User should be able to ask a question or search for answers	User can navigate to the respective category page and check for question or post.	Users can view multiple answers for a question.	Users can view multiple answers for a question.
4.	Registered User should be able to ask a question or search for answers	User enters information	User should search using search bar	User should search using search bar
5.	Registered User should be able to ask a question or search for answers	User should navigate to ask a question page	Users when uses speech to text ,user speech words should be converted to text	Users when uses speech to text ,user speech words should be converted to text

**Implementation:****Server side Implementation:-**

We implemented the whole application in android studio.

The source code is written in java and used bootstrap to implement GUI of our application. The database is maintained by Mongo DB.

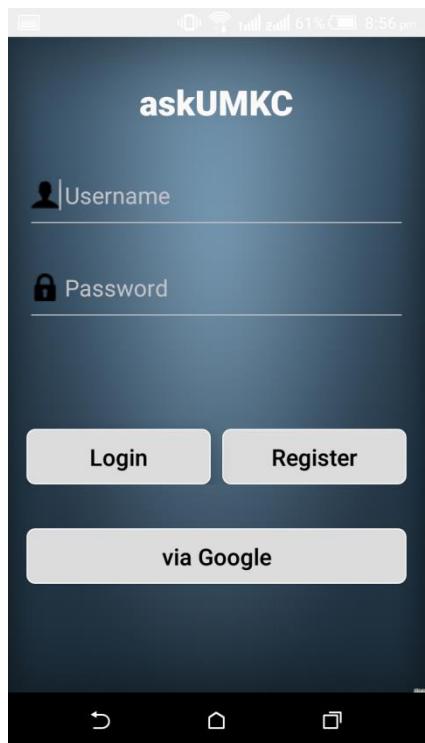
**Mobile client Implementation:-**

Mobile client application enable users to access the applications on their smart phones unlike web applications which need PC to access the application.

We are implementing our project in mobile client using android studio.

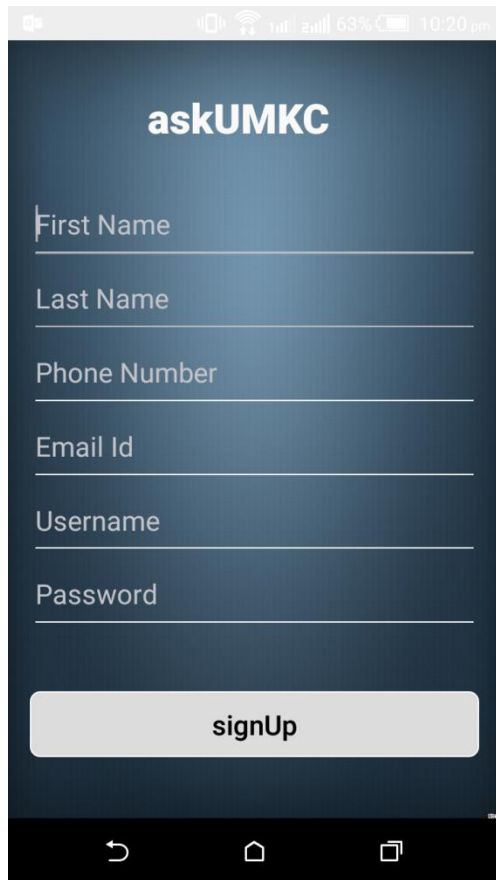
**Deployment:**

Login Screen:



User can login using Google account or with the credentials of registered app account. New users can register for the app using register button.

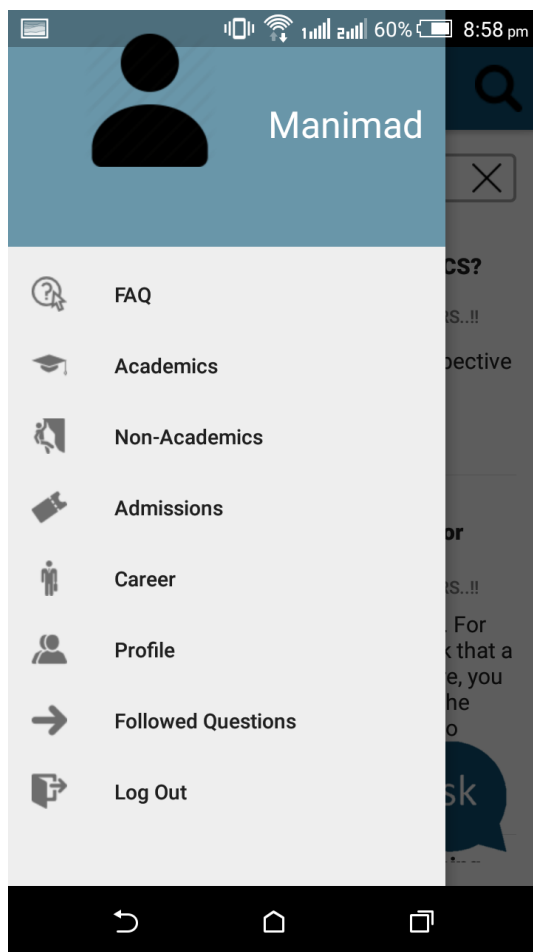
### Register:



The image shows a mobile application interface for registration. At the top, the status bar displays icons for signal, Wi-Fi, and battery (63%), along with the time 10:20 pm. The app's header is dark blue with the text "askUMKC" in white. Below the header, there are six text input fields stacked vertically, each with a light gray label: "First Name", "Last Name", "Phone Number", "Email Id", "Username", and "Password". Each field has a thin white underline. At the bottom of the form is a light gray rectangular button with the text "signUp" in black. The bottom of the screen shows a black navigation bar with three white icons: a back arrow, a home icon, and a recent apps icon.

In order to register, user should fill a form containing basic information like First Name, Last Name, Phone Number, Email Id, User Name, and Password. All the fields are mandatory to become a registered user. After successful registration, user will be redirected to login page and user have to login in order to view home page.

### Categories (Menu)

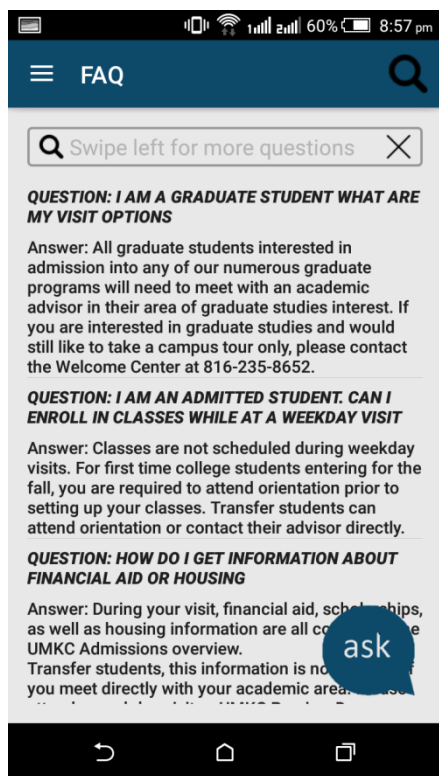


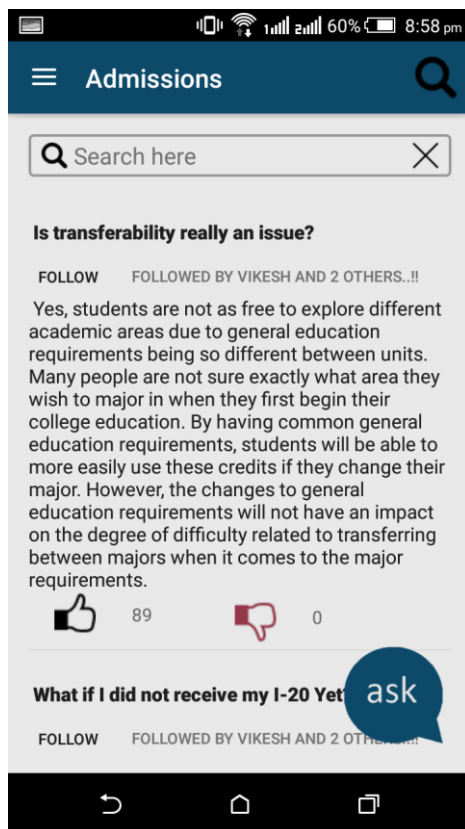
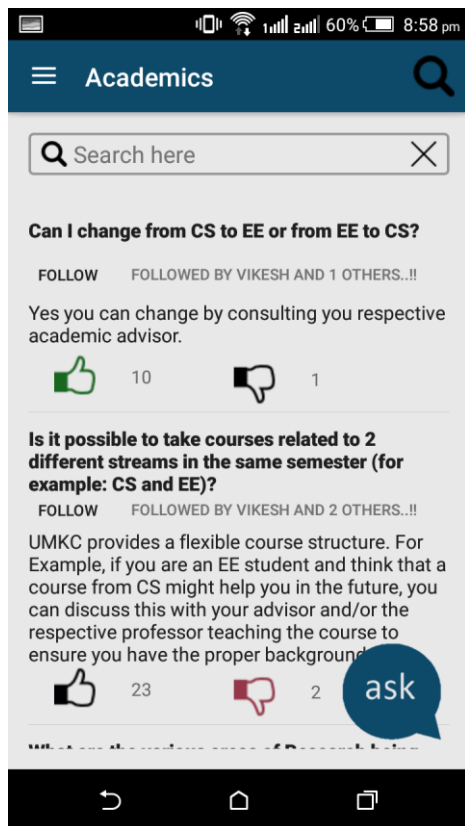
After successful login user navigates to home page where user can view all questions in a list view. User can scroll down to several questions. Questions are presented in reverse chronological order. Home will have menu option as shown in below image. Questions are divided into several categories and presented in menu. Menu will have options to 'update profile', 'My questions', 'FAQ'. Selecting a question launches a screen where user can see

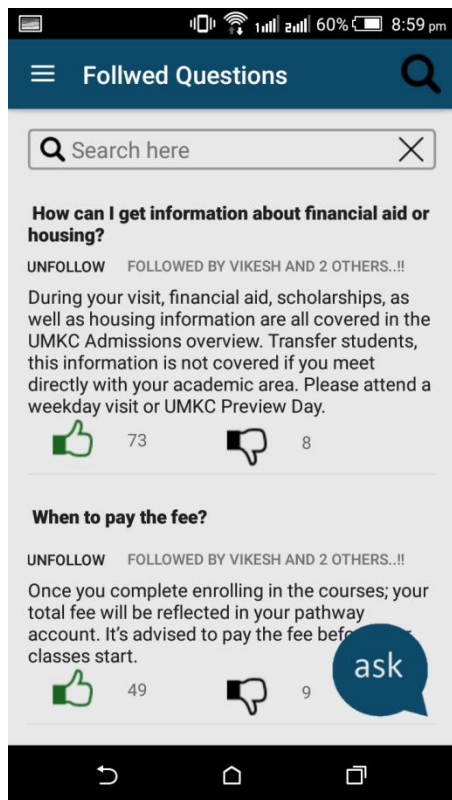
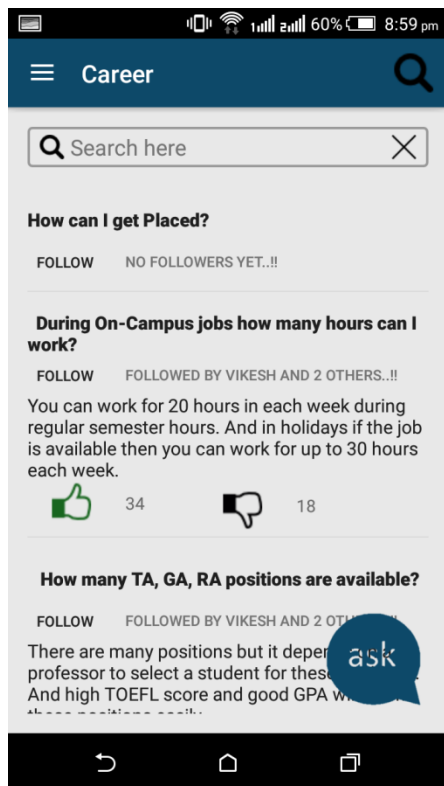


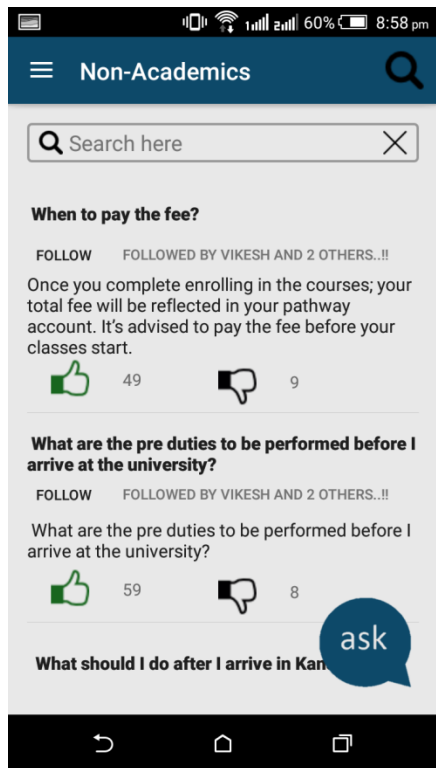
question and answers given to them. User can add an answer here in same page. Here user will have option to up vote an answer.

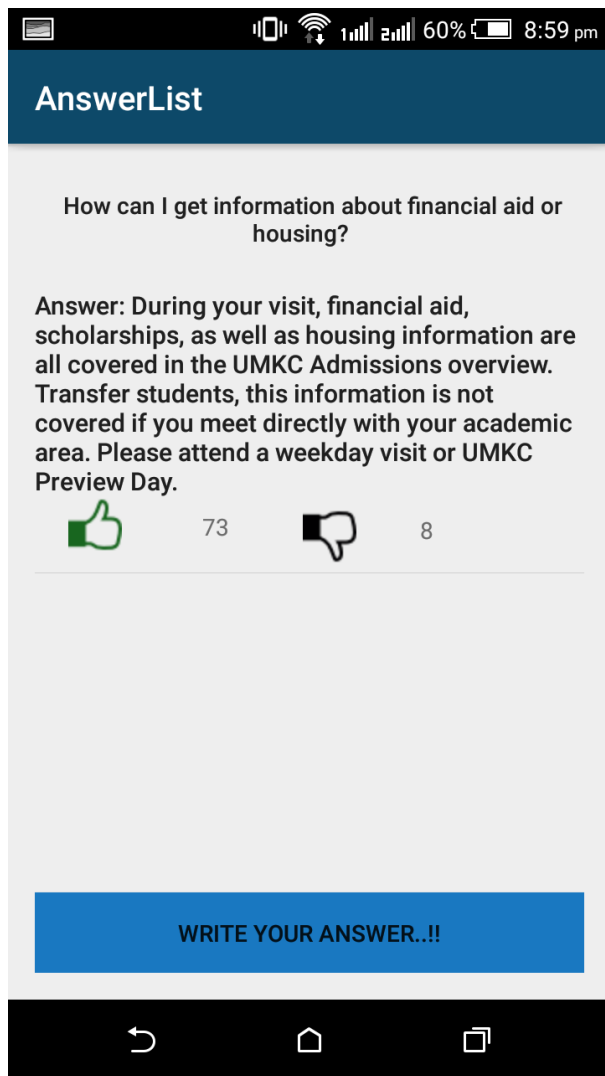
### Home Page:





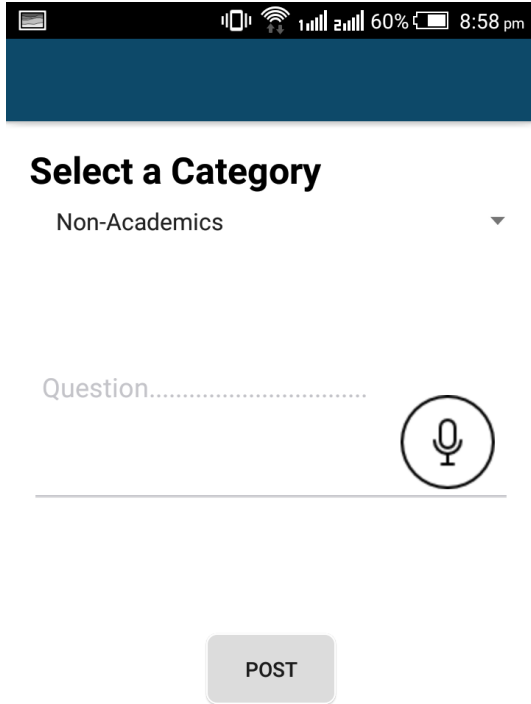






Home page consists of FAQ questions as a list view, when clicked on a specific question it navigates to a page with multiple answers for that question and a button where user can post a answer. There is a vote up option to decide appropriate answers for every question in the form of like and dislike buttons. Moreover, users can follow questions using follow button if needed. Users can search for questions using search bar in the home screen.

## Ask question:



The image shows a mobile application interface for asking a question. At the top is a status bar with icons for signal, Wi-Fi, battery (60%), and time (8:58 pm). Below this is a dark blue header bar. The main content area has a title "Select a Category" in bold black text. Underneath is a dropdown menu currently showing "Non-Academics" with a small downward arrow. Below the dropdown is a text input field with the placeholder text "Question.....". To the right of the text field is a circular icon containing a microphone, indicating a voice recording feature. A horizontal line separates the input area from a light gray button labeled "POST" at the bottom.

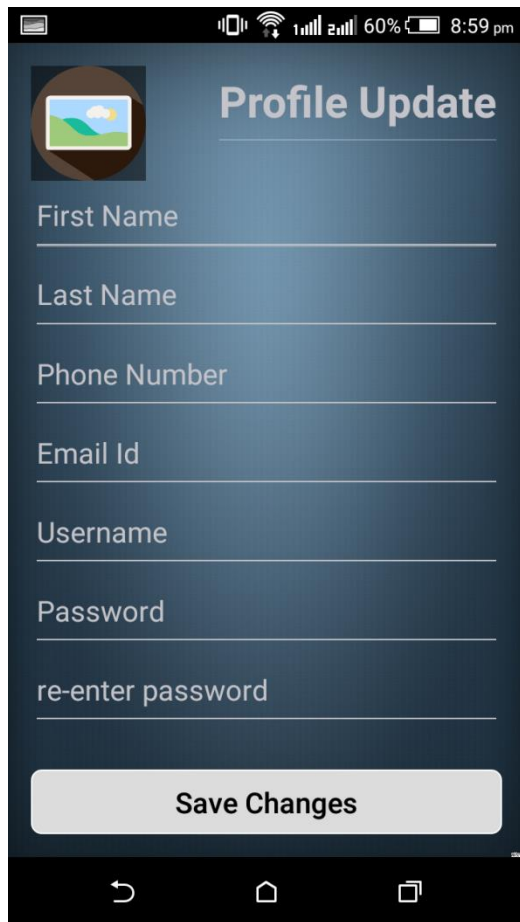


User can ask a question by touching 'Ask' button on home page.

After touching 'Ask' button in home page, user will be redirected to ask question screen. Here user will have different views to add question, add category.

In ask question page user have a facility to convert text to speech.

## Profile Update:



The screenshot shows a mobile application interface for updating a user profile. At the top, there is a status bar with icons for signal, Wi-Fi, and battery (60%), along with the time 8:59 pm. Below the status bar is a header section with a circular profile picture placeholder on the left and the title "Profile Update" on the right. The main form area contains several input fields: "First Name", "Last Name", "Phone Number", "Email Id", "Username", "Password", and "re-enter password". Each field has a horizontal line for text entry. At the bottom of the form is a large, light gray button labeled "Save Changes". The very bottom of the screen shows a standard Android navigation bar with back, home, and recent apps icons.

User can update various details of his profile like photo, password etc as shown in below screens.

**System requirements:**

Operating System: Windows /Linux/Mac OS X

RAM: 2 GB RAM (takes lot of time to run, so 4 GB recommended)

Hard Disk: 400 MB hard disk

Data Base: Mongo DB (can alter depending upon future enhancements)

Servers: Amazon Server (can alter depending upon availability of free online sources)

Software's: Java Development kit (JDK) and Android SDK

Screen resolution: 1280px \* 800px minimum

Additional: 1 GB space is required for android SDK, images and cache data

**Project Management:**

project timeline: 03-11-2016

members:

- Sravan Kumar Appana
- Vikesh Padartha
- Architha Mukka
- Manikanta Maddula

Issue1: Sravan worked on implementing search bar in home screen. So, that users can search for any question. Sravan and Vikesh together worked in implementing email notification. He worked for about 50 hours.



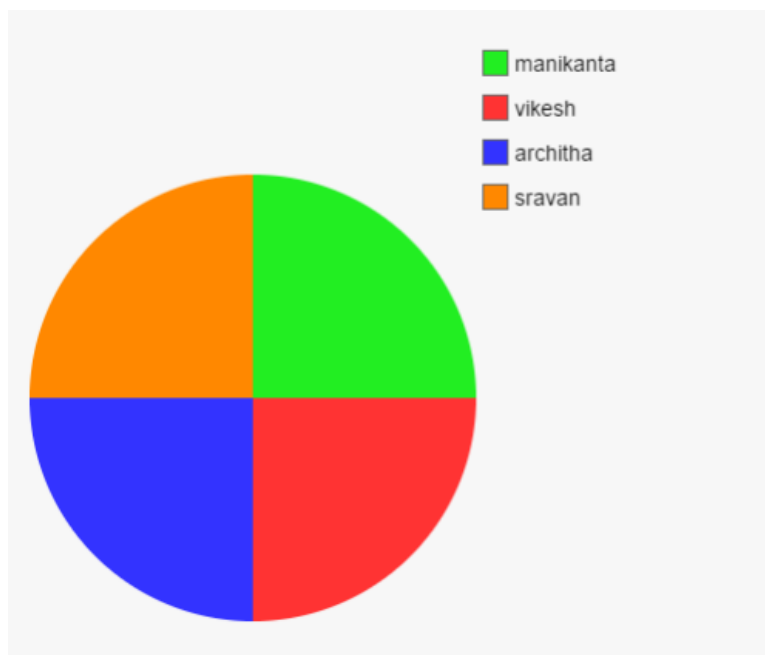
Issue2: Architha worked on UI part and changes to before screens. Worked on auto incrementing question ids globally for adding new questions to Mongo DB and fixed bugs in the application. She worked for about 40 hours.

Issue3: Vikesh worked on implementing speech to text conversion. Vikesh and Sravan together implemented email notification. Vikesh worked to Delete/Update operations on documents present in Mongo lab. Vikesh worked for about 50 hours.

Issue 4: Login is one of the basic operations that will be performed by any user. Our app allows to login by creating a profile or using Gmail account. Manikanta worked on integrating Gmail login, normal login, and validating a user while logging in. Worked to present questions in reverse chronological order from Mongo DB and fixed some bugs. He worked for about 45 hours.

Issues/Concerns:

1. Service Calls.
2. To improve performance degradation.



**Bibliography:**

- 1) <https://www.quora.com>
- 2) <http://www.umkc.edu/>