

ASE Project Increment2

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

Application Name

Question :

Answer 1:

Answer 2:

Answer 3:

Write a Answer

Ask question:


Home Screen

Question

Post

Categories:

Application Name



User Name

Academics

Non- Academics

Admissions

Career

Profile

My Questions

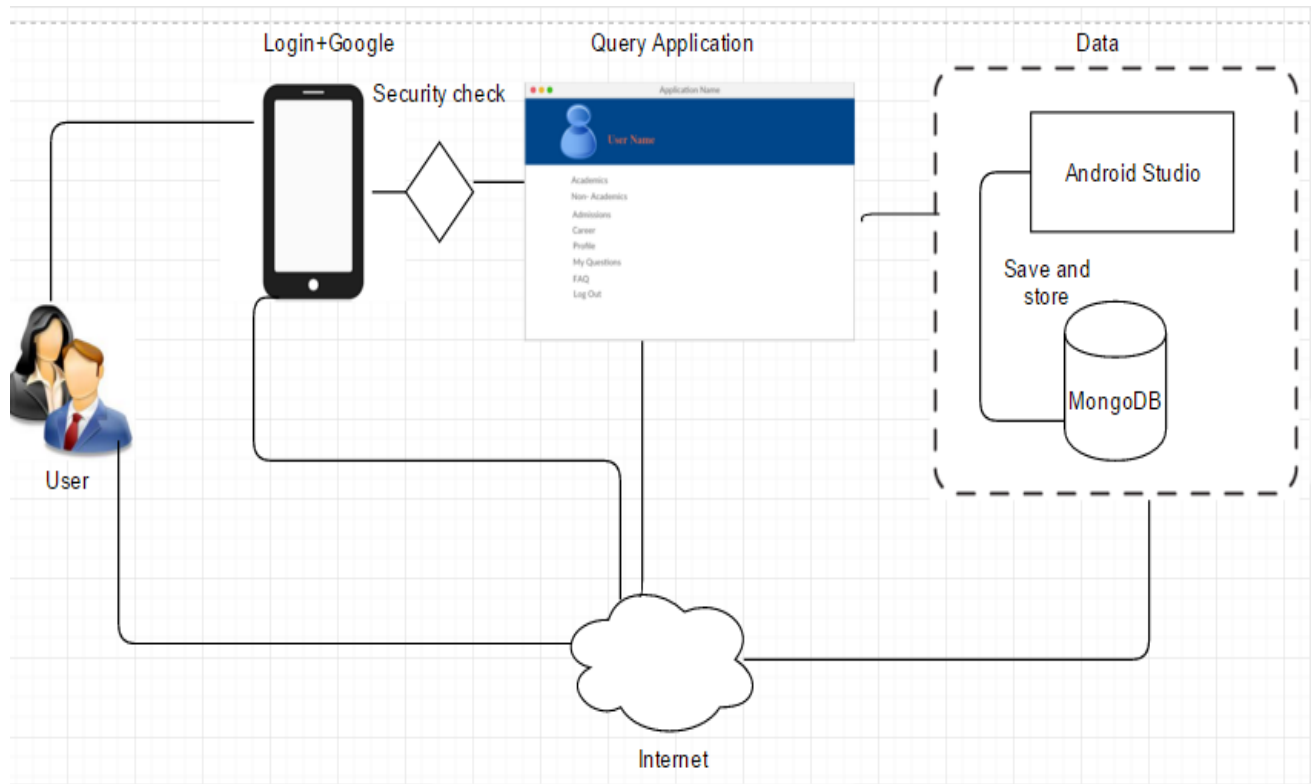
FAQ

Log Out

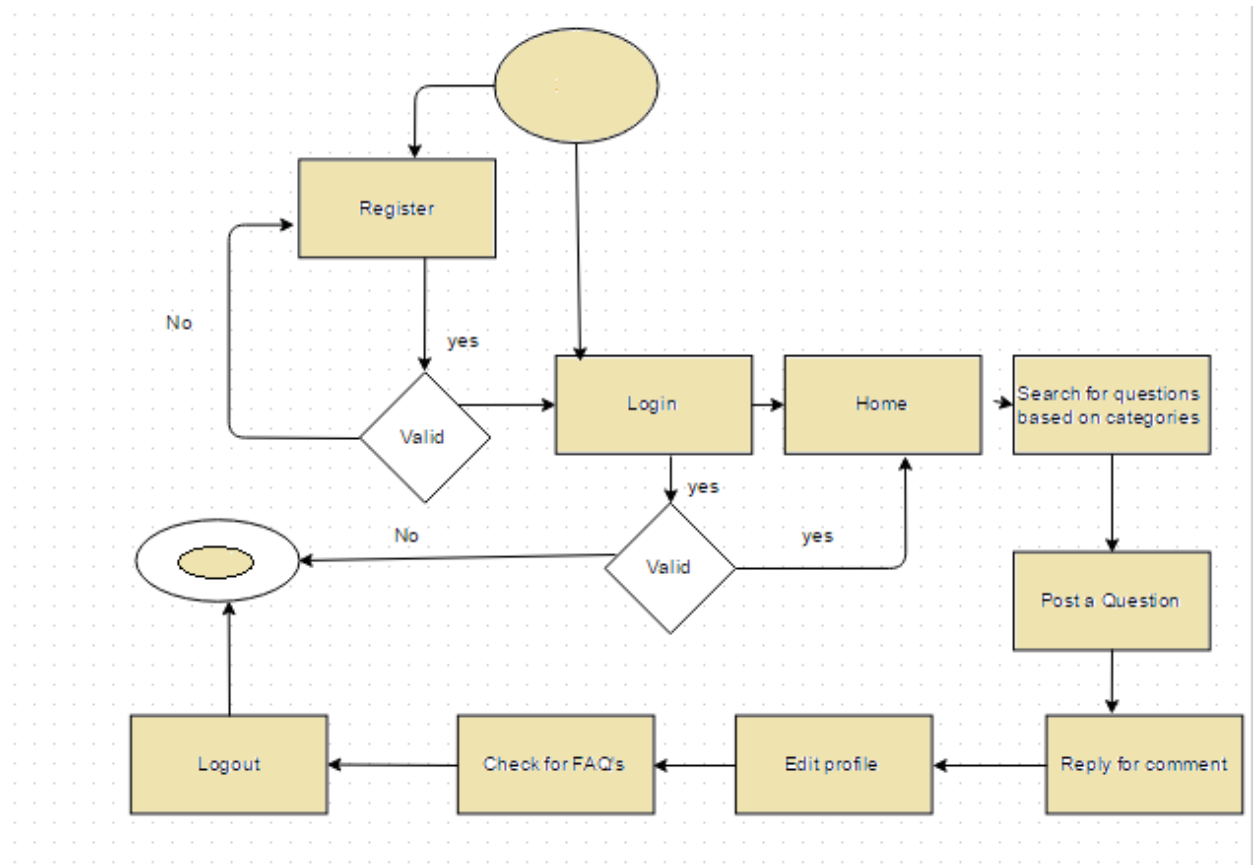
Profile update:



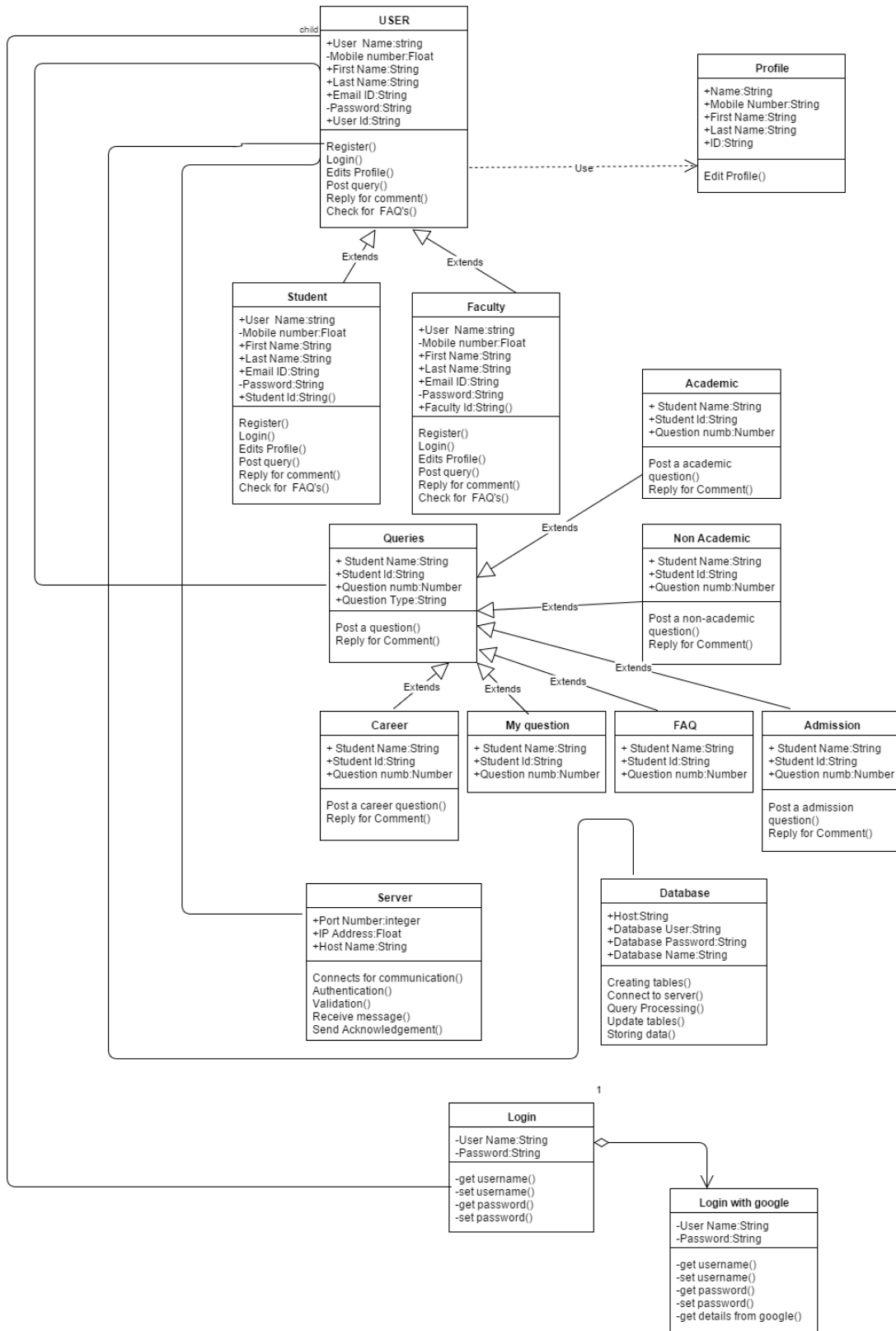
Architecture Diagram:



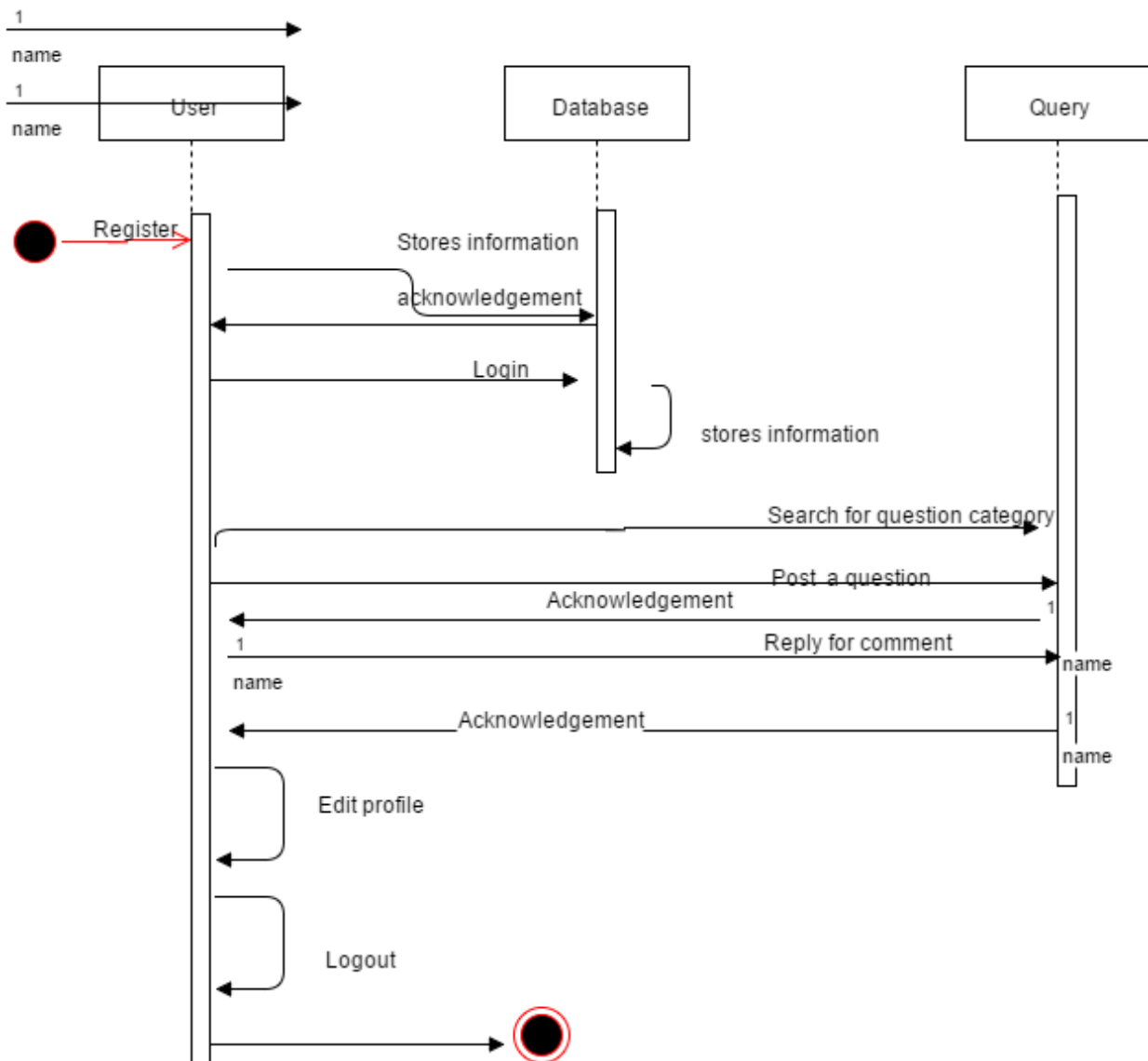
Activity Diagram:



Class Diagram:



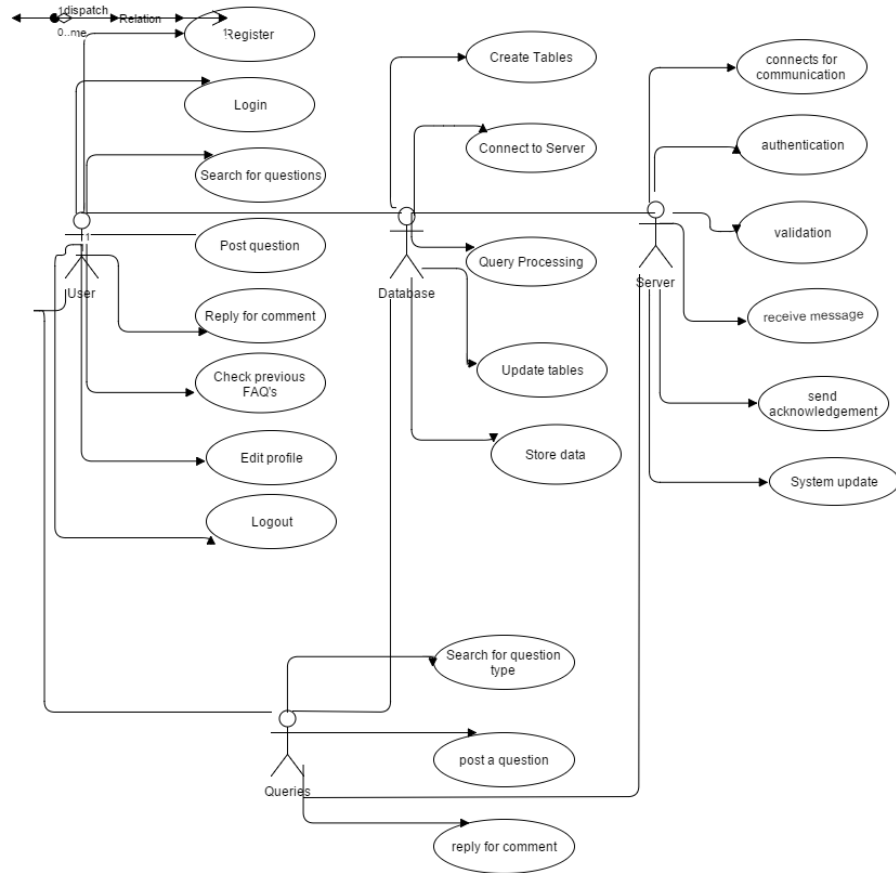
Sequence Diagram:



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 or unlike a answer specified to that question.
- 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 and unlike a answer.
- As a system, I must analyze the question and display multiple answers for that question.

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.

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.

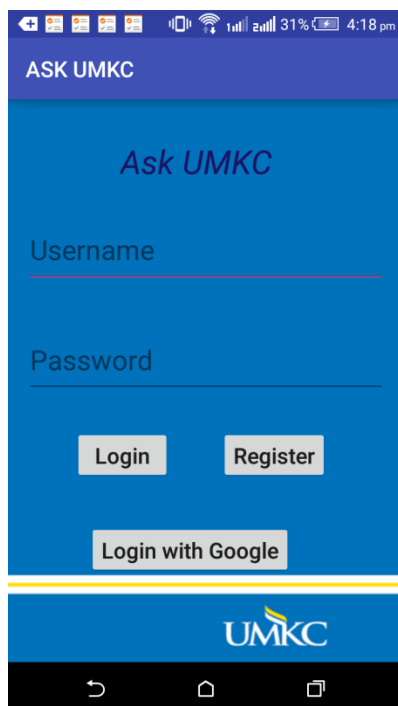
Mobile client Implementation:-

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.

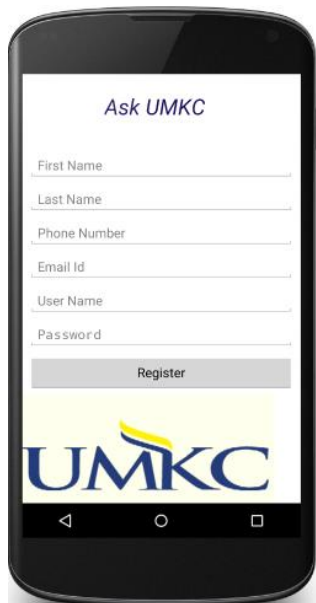
Deployment:

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:



Ask UMKC

First Name

Last Name

Phone Number

Email Id

User Name

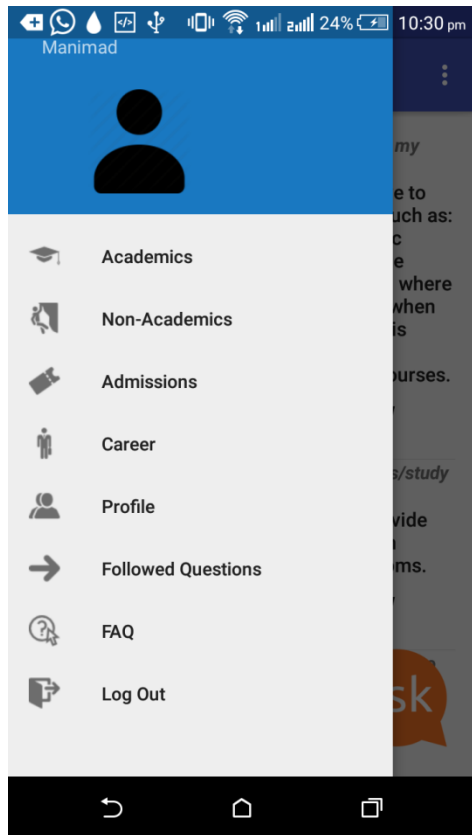
Password

Register

UMKC

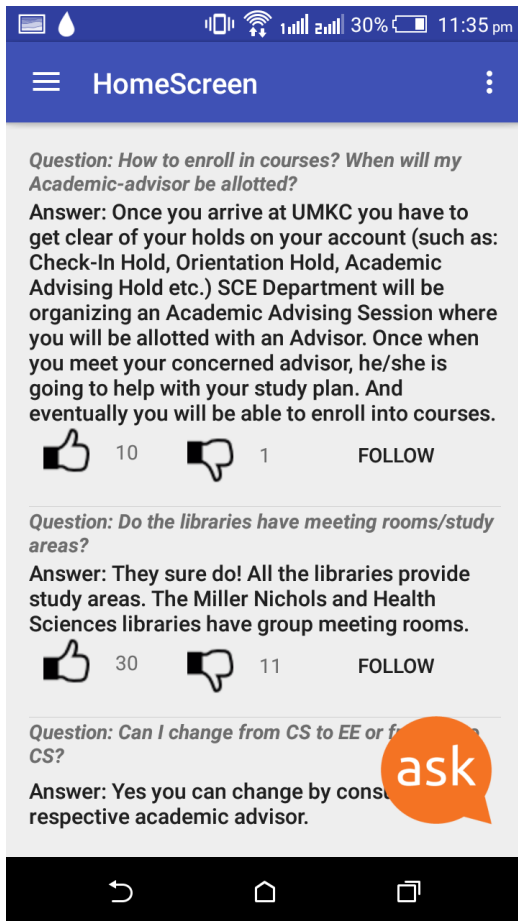
In order to register, user should fill a form containing basic information like First Name, Last Name, Phone Number, Email Id, User Name, and Password. All the fields are mandatory to become a registered user. After successful registration, user will be redirected to login page and user have to login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

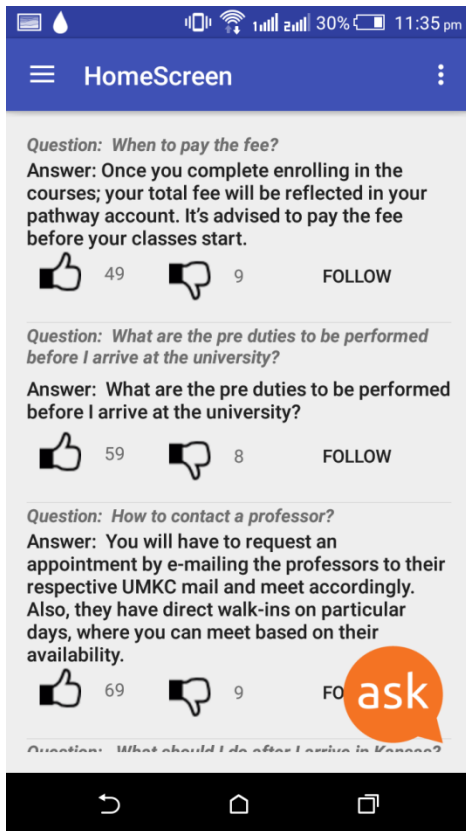
Home Page:

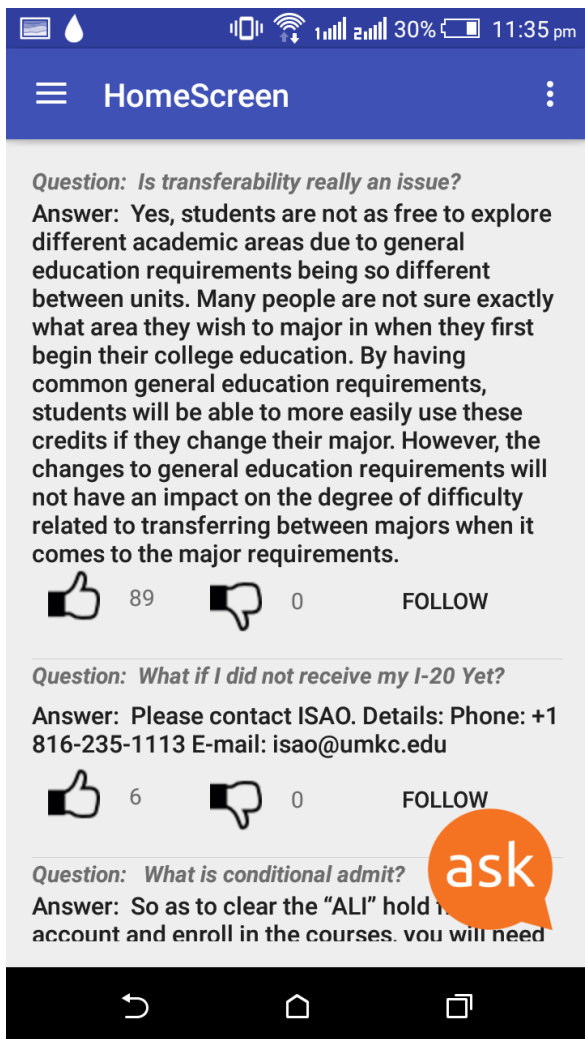


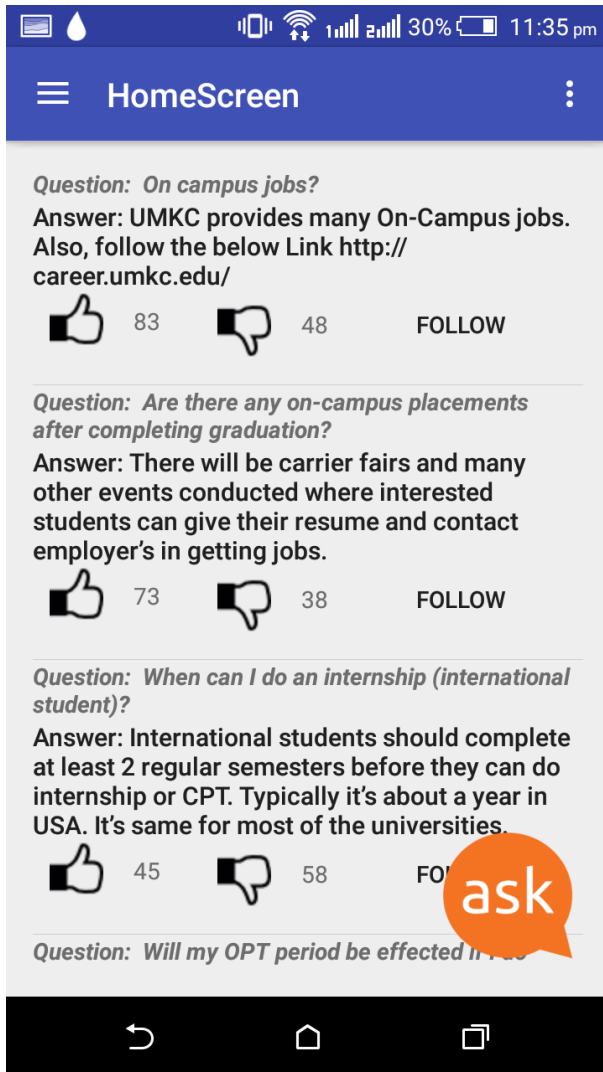
Categories (Menu)

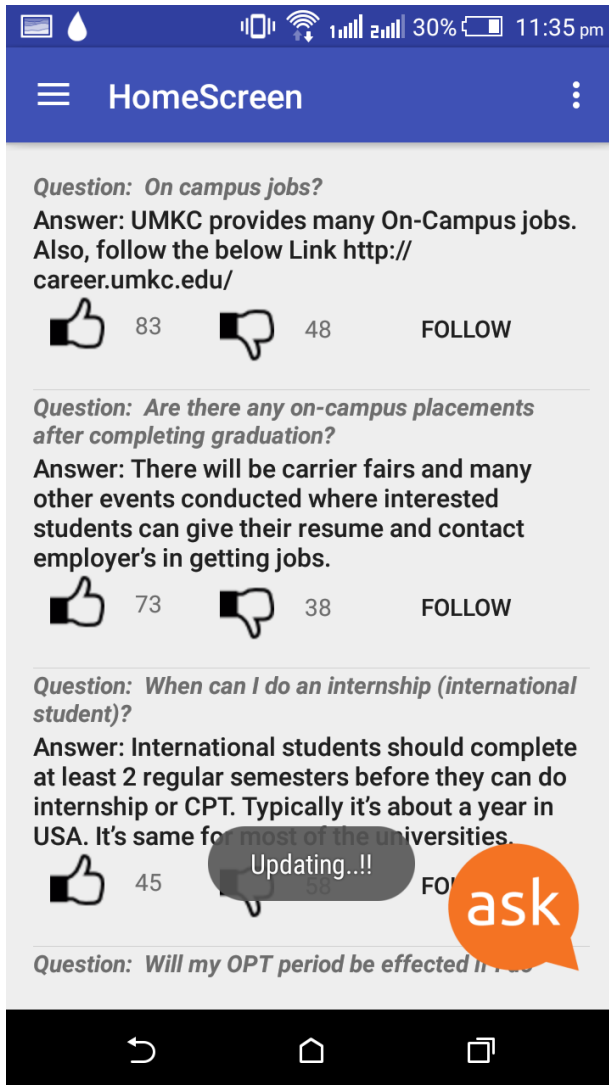
After successful login user navigates to home page where user can view all questions in a list view. User can scroll down to several questions. Questions are presented in reverse chronological order. Home will have menu option as shown in below image. Questions are divided into several categories and 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.





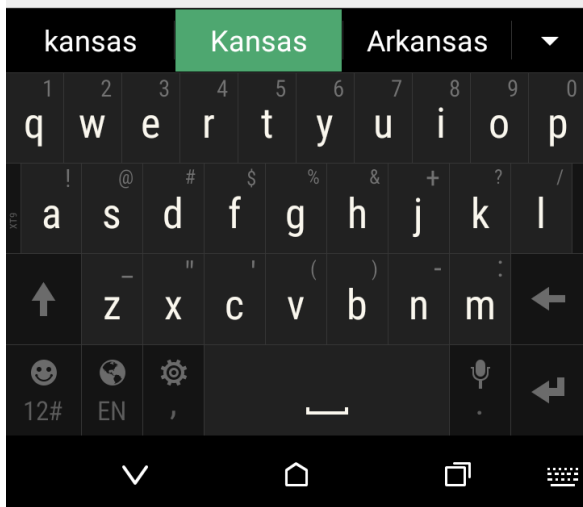






climate conditions in kansas

POST



Question: How to enroll in courses? When will my Academic-advisor be allotted?

Answer: Once you arrive at UMKC you have to get clear of your holds on your account (such as: Check-In Hold, Orientation Hold, Academic Advising Hold etc.) SCE Department will be organizing an Academic Advising Session where you will be allotted with an Advisor. Once when you meet your concerned advisor, he/she is going to help with your study plan. And eventually you will be able to enroll into courses.



11



1

FOLLOW

Question: Do the libraries have meeting rooms/study areas?

Answer: They sure do! All the libraries provide study areas. The Miller Nichols and Health Sciences libraries have group meeting rooms.



30



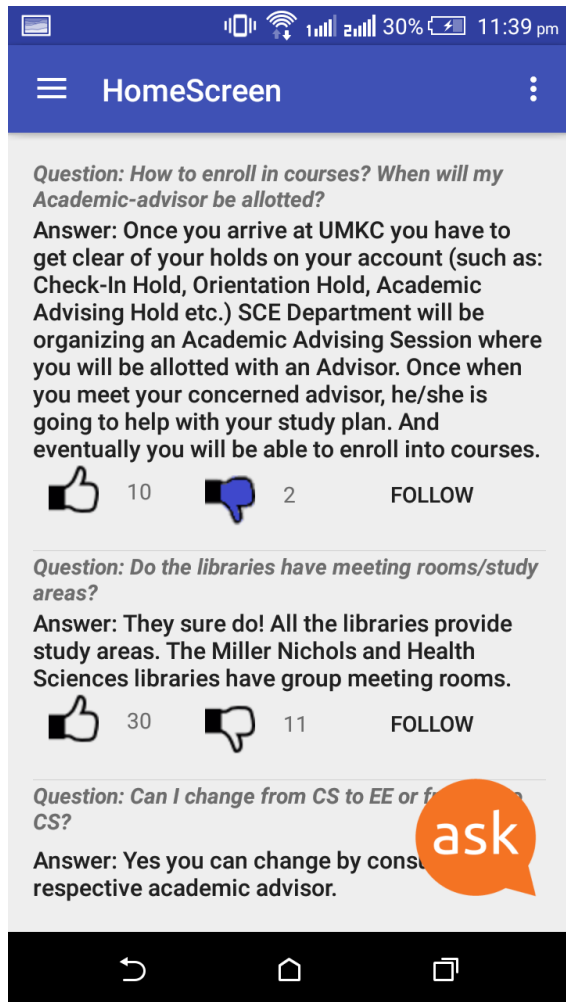
11

FOLLOW

Question: Can I change from CS to EE or from EE to CS?

Answer: Yes you can change by consulting your respective academic advisor.

ask

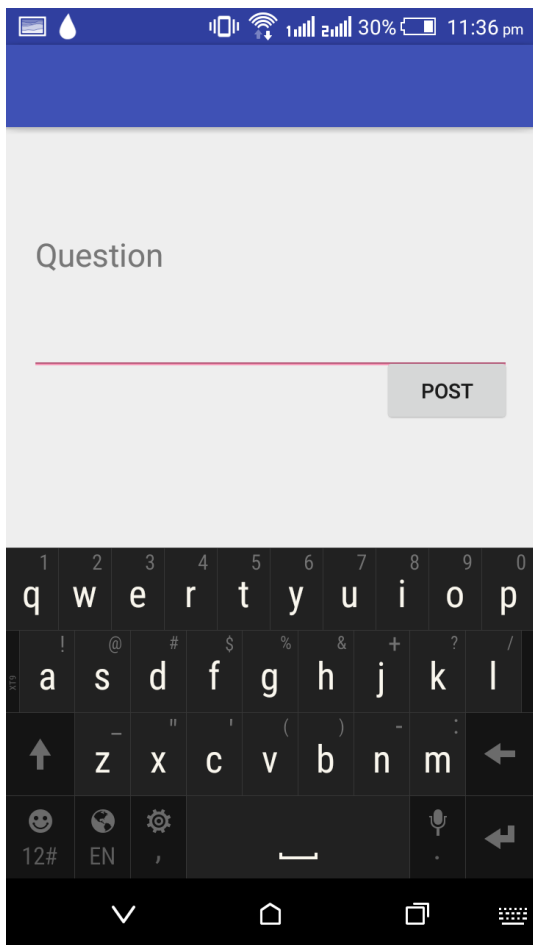


Home page consists of questions as a list view, when clicked on a specific question where it navigates to a page with multiple answers for that question and a button where user can post a answer. There is a vote up option to decide appropriate answers for every question in the form of like and dislike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

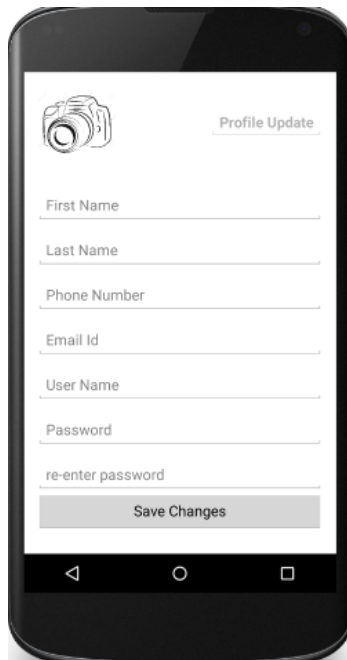
Ask question:

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.



Profile Update:



The image shows a mobile application interface for updating a user's profile. At the top left is a camera icon for profile picture selection. To its right is the title 'Profile Update'. Below the title are seven input fields: 'First Name', 'Last Name', 'Phone Number', 'Email Id', 'User Name', 'Password', and 're-enter password'. A 'Save Changes' button is located at the bottom of the form. The entire interface is displayed within a black smartphone frame with standard Android navigation icons at the bottom.

Profile Update

First Name

Last Name

Phone Number

Email Id

User Name

Password

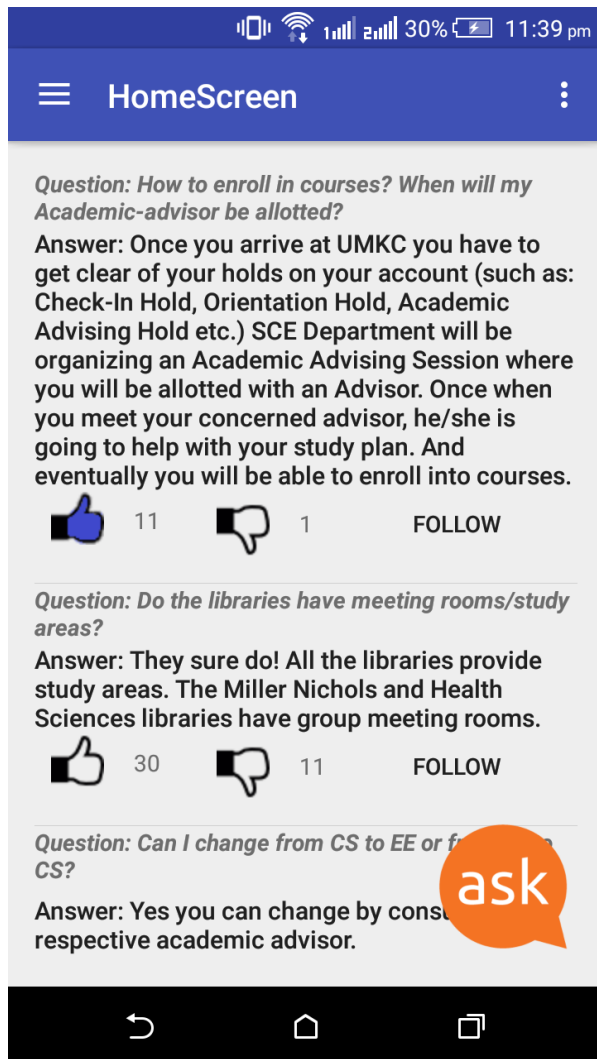
re-enter password

Save Changes

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

FAQ:

When user checks for FAQ in category, it navigates to the above screen with frequently asked questions.



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 Padarathi
- Architha Mukka
- Manikanta Maddula

Issue1: Worked on parsing Mongo DB data in to list views in the home screens and also worked on up vote and down vote counter changes in the home screen. Worked on UI issues in the home screen. Working on pushing data to Mongo DB, worked on camera option and image updating in profile update page and converting the image data to bit map data. Moreover, worked on menu options and fixing bugs. He worked for about 60 hours on this issue.

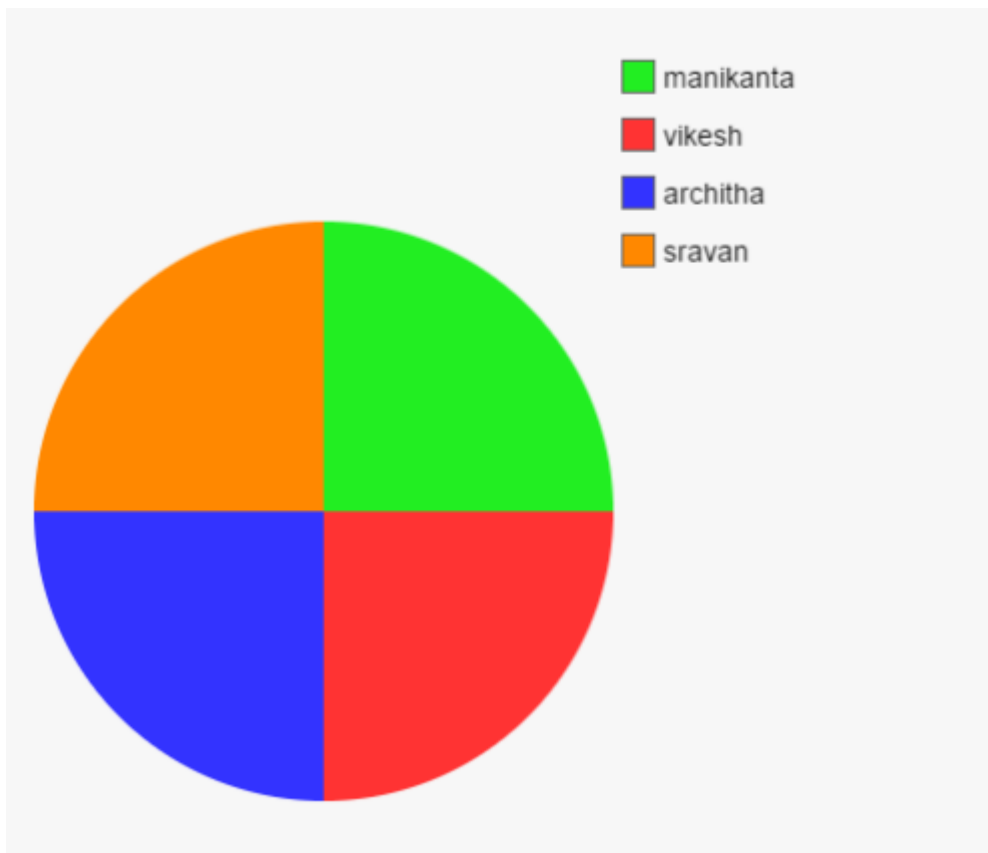
Issue2: Worked on answer list view screen which contains a question and multiple answers for it in a list view with up vote and down vote button and a button for posting a answer and working on updating data to Mongo DB. Collected images for up vote and down vote button. Wireframes for answer list view screen. She worked for about 30 hours on this issue.

Issue3: 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. Created sample Mongo DB document for question collection, collected few questions in list view and answer screen. Worked on answer button bug in answer list screen button overlapping up vote and down vote button. He worked for about 25 hours on this issue.

Issue 4: The data entered by user should be created in DB(mongo DB). Vikesh worked on this critical issue for many hours and pushing data to db worked. This critical issue is solved after trying several methods. Async task is used to avoid slow working of application due to heavy queries. He worked on this for about 35 hours on this.

Issues/Concerns:

1. Delete/Update operations on documents present in Mongo lab.
2. Assigning followed questions of a user needs to be tested for other use cases, need to see for a work around for all scenarios.
3. Need to work on auto incrementing question ids globally for adding new questions to Mongo DB.
4. Still need to present the questions in reverse chronological order from Mongo DB.
5. Data of users related to particular questions and answers may need to be presented on screens.



Bibliography:

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