**HelpStack Mobile App**

Submitted by:   Team Helpstack

Submitted to:   Dr. Mehrdad Tirandazian

ICET Department,

School of Engineering Technology and Applied Science

Progress Campus, Block A

Centennial College

Discipline:   Software Engineering 

Due Date:    14.12.2018

# Declaration of Sole Authorship

We, Team Helpstack, confirm that this work submitted for assessment is our own and is expressed in our own words. Any uses made within it of the works of any other author, in any form (ideas, equations, figures, texts, tables, programs), are properly acknowledged at the point of use. A list of the references used is included.

Signed:

NehaSebastain, 300991285 (Health Informatics Technology)

Oreeba Badar, 300865033 (Software Engineering Technology)

Gurjit Singh, 300976262 (Health Informatics Technology)

Jasneet Kaur, 300976262 (Health Informatics Technology)

Amit Roy, 300973930 (Health Informatics Technology)

Amrinder Singh, 300984052 (Health Informatics Technology)

Date: 14.12.2018

# Abstract

Any registered members of this app can get help in every area of study from the users of the app. Hence, we can utilize the talents of bright minds and efficient faculties of the college. It also provides a platform for interaction among people in the college. The solution seekers post query for help in the app and all the registered members get notification and can provide help for the user through both chat support and face to face interaction.

**Table of Contents**

**Declaration of Sole Authorship 2**

**Abstract 3**

**List of Figures 6**

**List of Tables 7**

**1.0** **INTRODUCTION 8**

**2.0** **METHODOLOGY AND RESULTS 9**

**2.1** **Literature Review 9**

**2.2** **Proposed Solution 9**

**2.3** **User Role Modelling 10**

2.3.1 Brainstorm and Group [10](#_3rdcrjn)

2.3.2 Consolidated User Roles [11](#_lnxbz9)

2.3.3 Description of User Roles and Persona [12](#_1ksv4uv)

2.3.4 Additional Documentation [13](#_44sinio)

**2.4** **Release 1.0 14**

2.4.1  User Stories [14](#_z337ya)

2.4.2  Additional Documentation [18](#_qsh70q)

2.4.3  Release Plan 1.0 [19](#_3as4poj)

2.4.4  Iteration Plan (Release 1.0) [20](#_1pxezwc)

2.4.5   Additional Documentation [21](#_2p2csry)

2.4.6  Progress Monitoring [22](#_147n2zr)

2.4.7  Acceptance Tests for Release 1.0 [23](#_23ckvvd)

**2.5** **Release 2.0 25**

2.5.1  User Stories [25](#_1hmsyys)

2.5.2  Additional Documentation [26](#_41mghml)

2.5.3   Release Plan 2.0 [27](#_2grqrue)

2.5.4   Iteration Plan (Release 2.0) [28](#_vx1227)

2.5.5   Additional Documentation [29](#_3fwokq0)

2.5.6   Progress Monitoring [30](#_1v1yuxt)

2.5.7   Acceptance Tests for Release 2.0 [31](#_4f1mdlm)

**3.0** **CONCLUSIONS 32**

**4.0** **RECOMMENDATIONS 33**

**CREDITS, LICENSE, AND REFERENCES 34**

**Credits 34**

**License 34**

**References 34**

**APPENDIX A (DESIGN DOCUMENT) 35**

**APPENDIX B (TEST PLAN) 36**

**1.0** **Introduction 36**

1.0.1   Goals [36](#_111kx3o)

1.0.2   Assumptions [36](#_3l18frh)

1.0.3   Risks And Assets [36](#_206ipza)

Describe the elements that might positively influence testing on the project. [36](#_4k668n3)

**2.0** **Scope 36**

2.0.1   Features To Be Tested [36](#_1egqt2p)

2.0.2    Features Not To Be Tested [36](#_3ygebqi)

**3.0** **Testing Procedures 37**

3.0.1   Test Objectives [37](#_sqyw64)

3.0.2   Types Of Testing [37](#_3cqmetx)

3.0.3   Testing Tools [38](#_1rvwp1q)

**4.0** **Schedule and Deliverables 38**

**APPENDIX C (END-USER & ADMINISTRATOR MANUALS) 40**

# List of Figures

Figure 1: Organizing the user role cards on a table [1]. [10](#_26in1rg)

Figure 2: The consolidated role cards [1]. [11](#_35nkun2)

Figure 3: Example of a “consolidated” low-fidelity prototype. Note that each “individual” low-fidelity prototype is developed for each user role [1]. [15](#_3j2qqm3)

Figure 4: A story with notes providing additional detail [1]. [15](#_1y810tw)

Figure 5: The revised front of a story card with only the story and questions to be discussed [1]. [16](#_4i7ojhp)

Figure 6: Details that imply test cases are separated from the story itself. Here they are shown on the back of the story card [1]. [16](#_2xcytpi)

Figure 7: An example of a constraint story card [1]. 16

# List of Tables

Table 1: The Must-Have stories for Release x.y [1]. [17](#_3whwml4)

Table 2: The Should-Have stories for Release x.y [1]. [17](#_2bn6wsx)

Table 3: Disaggregated tasks per story [1]. [20](#_49x2ik5)

Table 4: Progress and changes for all Iterations for Release 1.0 [1]. [22](#_3o7alnk)

Table 5: Stories, acceptance tests, and contributors for Release 1.0 (Green=Accepted; Red=Rejected; Black=Not started). 24

# 1.0 INTRODUCTION

The lifestyle of post secondary education and the workloads that come with it can sometimes be overwhelming for students, with varying schedules and the potential lack of availability for extra help, it can be difficult for a student to complete some of the more challenging tasks presented from class. Complicated questions are a common occurrence with post secondary studies, and sometimes “Googling it” just doesn’t give the right information required or can be tough to understand. With a school filled with bright minds and educated faculty, why not utilize this resource in a freer flowing and centralized way. With a system or more specifically a mobile application in place, this centralized resource could become a reality allowing students and faculty to utilize each other as a knowledge source to overcome those tough questions/problems that arise. The application would allow you to meetup on campus and discuss the challenges you’re presented with to a fellow student or faculty member who may have a better understanding of the subject matter, which can greatly increase your odds of grasping the concept to solve the challenge you initially faced.

# 2.0 METHODOLOGY AND RESULTS

## 2.1 Literature Review

The fast-paced life always creates more opportunity to provide and get help with

monetary benefits and also to learn from it. The app focuses on getting a mutual benefit for the users. The lifestyle of students and the workloads are sometimes challenging, and they face a hard time in completing their work on time and “google it” does not always provides the right help. Our app “Help Stack” aims at providing a solution for this. The application would allow you to meetup on campus and discuss the challenges you’re presented with to a fellow student or faculty member who may have a better understanding of the subject matter, which can greatly increase your odds of grasping the

concept to solve the challenge you initially faced.

## 2.2 Proposed Solution

**Solution description:**

The app creates a platform to get help faster without booking an appointment and wait for the slot and hence both the seeker and provider gets benefit from it. It benefits the provider since provider can get monetary benefits or bonus points to earn a volunteering certificate.

**Business benefits**:

Due to the nature of the application and vision of free knowledge sharing, the profitable business benefits are limited to allow growth within the userbase of the application, which is the most important part for the success of the application. Some benefits could include:

1. Improving brand image by providing an excellent resource for students & faculty to utilize.

2. Collection of user details and identifying common trends of queries generated by students and faculty.

**Criteria by which the project is deemed** **successful:**

The users can get help more faster and users does not need to wait to fix an appointment and then get the solution, hence its 20% more faster than existing websites.

## 2.3 User Role Modelling

### 2.3.1 Brainstorm and Group

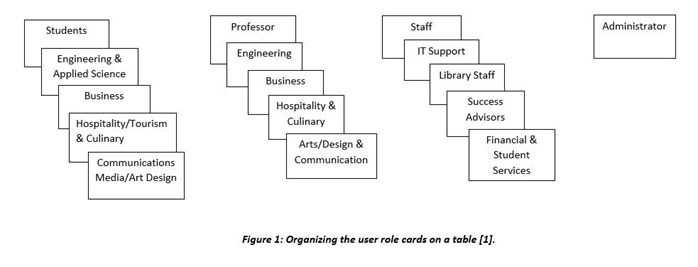
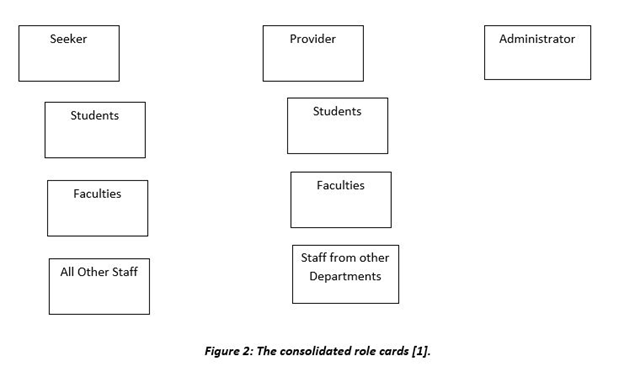


Figure 1: Organizing the user role cards on a table [1].

### 2.3.2 Consolidated User Roles



### 2.3.3 Description of User Roles and Personal

The students, faculties and staff all take the role of a seeker and a provider. In the consolidated role card, they all fall under the categories; provider and seeker.

We decided to keep the students, faculties and all other staff because these are the categories of the people who will be using this application. The administrator role was also kept, because the administrator is the only person responsible for the maintenance and support of the application.

**Description of User Roles:**

**Seeker**

- Able to register for the Help Stack application

o Able to sign in/out of the Help Stack application

o Able to retrieve forgotten password

- Able to post queries

o Can update queries

o Able to delete queries

o Search for queries using filters provides

- Able to receive notifications

- Able to communicate with help provider through a messaging thread

o option to email person providing the help

**Provider**

- Able to register for the Help Stack application

o Able to sign in/out of the Help Stack application

o Able to retrieve forgotten password

- Able to post queries

o Can update queries

o Able to delete queries

o Search for queries using filters provides

- Able to receive notifications

- Able to answer queries

- Able to communicate with help provider through a messaging thread

o option to email person providing the help

**Administrator**

- Able to sign in/out of the Help Stack application

- Able to create/read/update/delete users

- Able to maintain the application

- Able to support and help with the application

2.3.4 **Additional Documentation**

User Role video and Iteration-1 Video link:

<https://drive.google.com/drive/folders/1XPfPi7wRolcz2ruXYA32jbpPFjChasyU?usp=sharing>

Release 2 Video link:

<https://drive.google.com/open?id=1b7_BG-PlwnTkc4kft4g4Vq-rrytnQ48v>

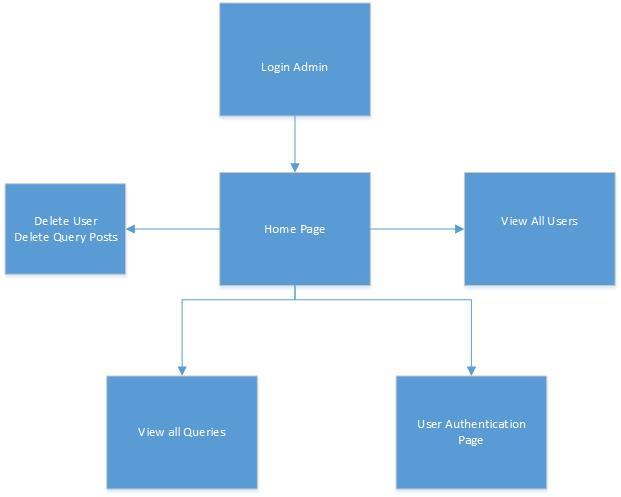
## 

## 2.4 Release 1.0

### 2.4.1  User Stories

**Story Cards for each User Role**

1. ***Administrator***

**

*Fig. 1.1 Low Fidelity Prototype for Administrator*

|  |
| --- |
| An administrator can login into the system.  ***Note****: Once logging in, administrator should be taken to into the Homepage* |

*Fig. 1.2 Administrator Story Card 1*

|  |
| --- |
| The Admin should be able to view/delete users  ***Note****: test if administrator can remove a user if necessary, then view the queries* |

*Fig. 1.3 Administrator Story Card 2*

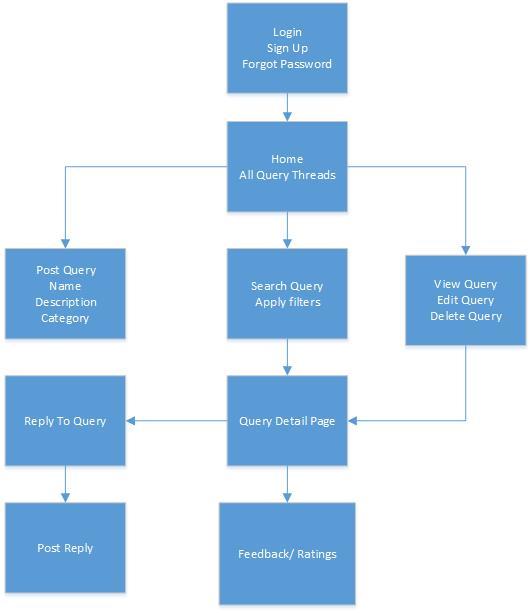
|  |
| --- |
| The Admin should also view/delete the queries in the system.  ***Note:*** *Test if admin can remove unnecessary queries* |

*Fig. 1.4 Administrator Story Card 3*

|  |
| --- |
| The Admin should be able to authenticate the user registered.  ***Note****: Admin should grant access to new users* |

*Fig. 1.5 Administrator Story Card 4*

1. ***Provider/Seeker***

****** *Fig. 2.1 Low Fidelity Prototype for Provider/Seeker*

|  |
| --- |
| The Provider/Seeker should be able to login.  ***Note****: Test is the user is registered to the application* |

*Fig. 2.2 Provider/Seeker Story Card 1*

|  |
| --- |
| The Provider/Seeker should be able to post the Query.  ***Note****: Test if user can successfully post the query* |

*Fig. 2.3 Provider/Seeker Story Card 2*

|  |
| --- |
| The Provider/Seeker should be able to search for the Query, based on the filters provided  ***Note****: Test if the Seeker/Provider can search the query using the categories or tagged filters.* |

*Fig. 2.4 Provider/Seeker Story Card 3*

|  |
| --- |
| The Provider/Seeker should be able to view, edit and delete the query.  ***Note****: Test if the queries can be viewed, edited, and deleted by the seeker/provider.* |

*Fig. 2.5 Provider/Seeker Story Card 4*

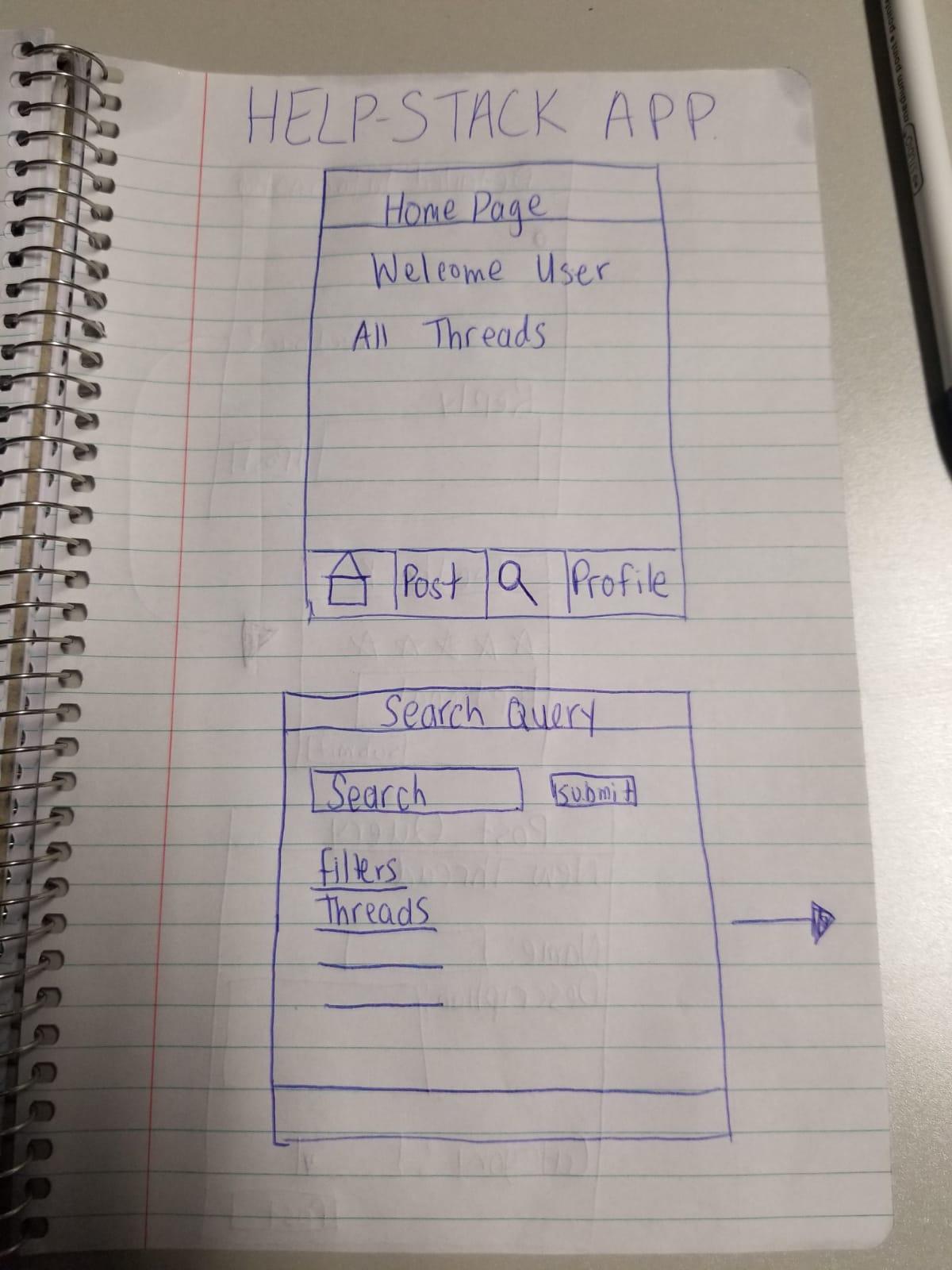
|  |
| --- |
| The Provider/Seeker should be able to reply to the posted query.  ***Note****: Test if the replies are posted for the queries, and if they can be viewed by the other users.* |

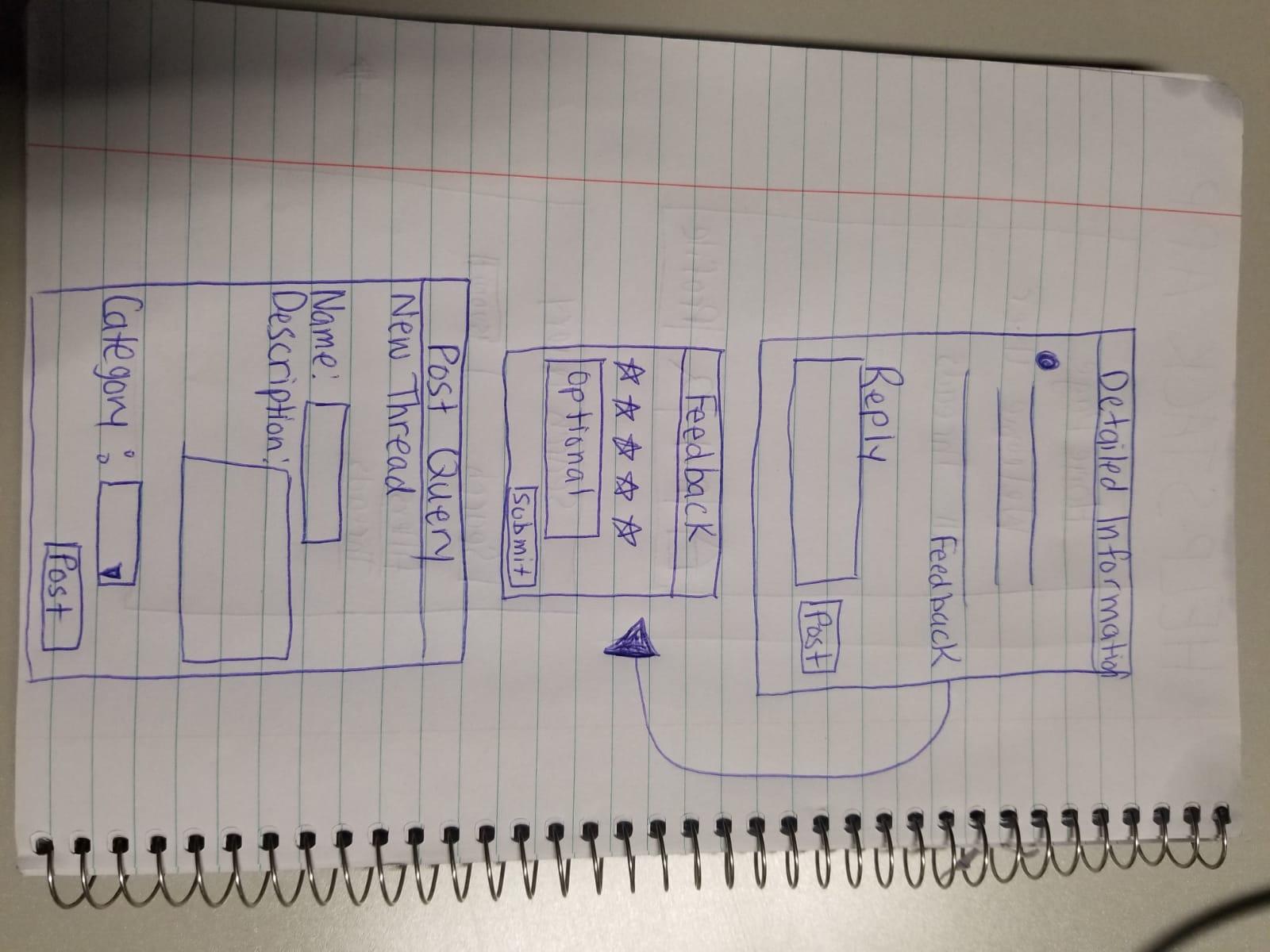
*Fig. 2.6 Provider/Seeker Story Card 5*

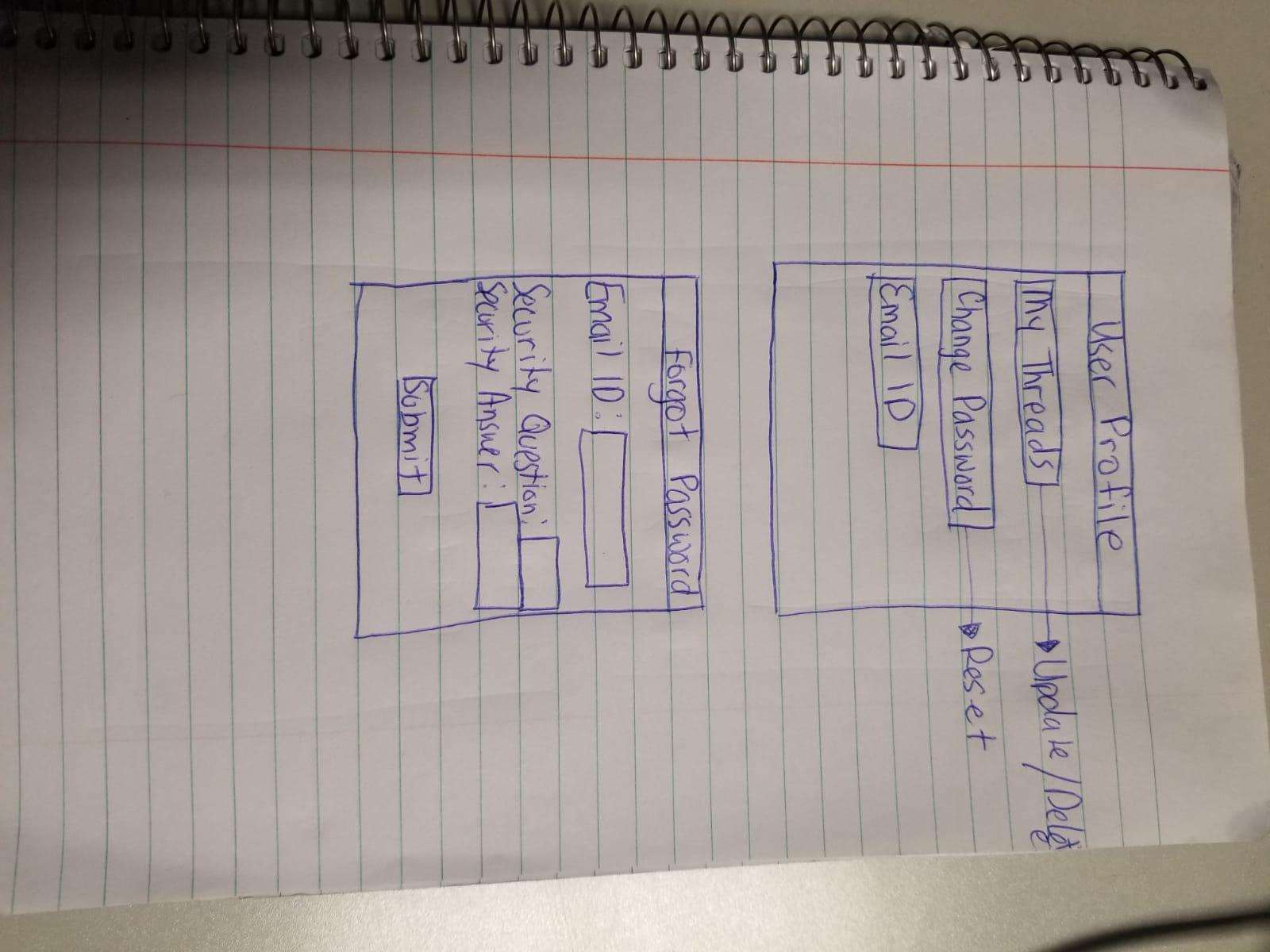
|  |
| --- |
| The Provider/Seeker should be able to post feedback for other users.  ***Note****: Test if the user ratings and feedback can be viewed by other users.* |

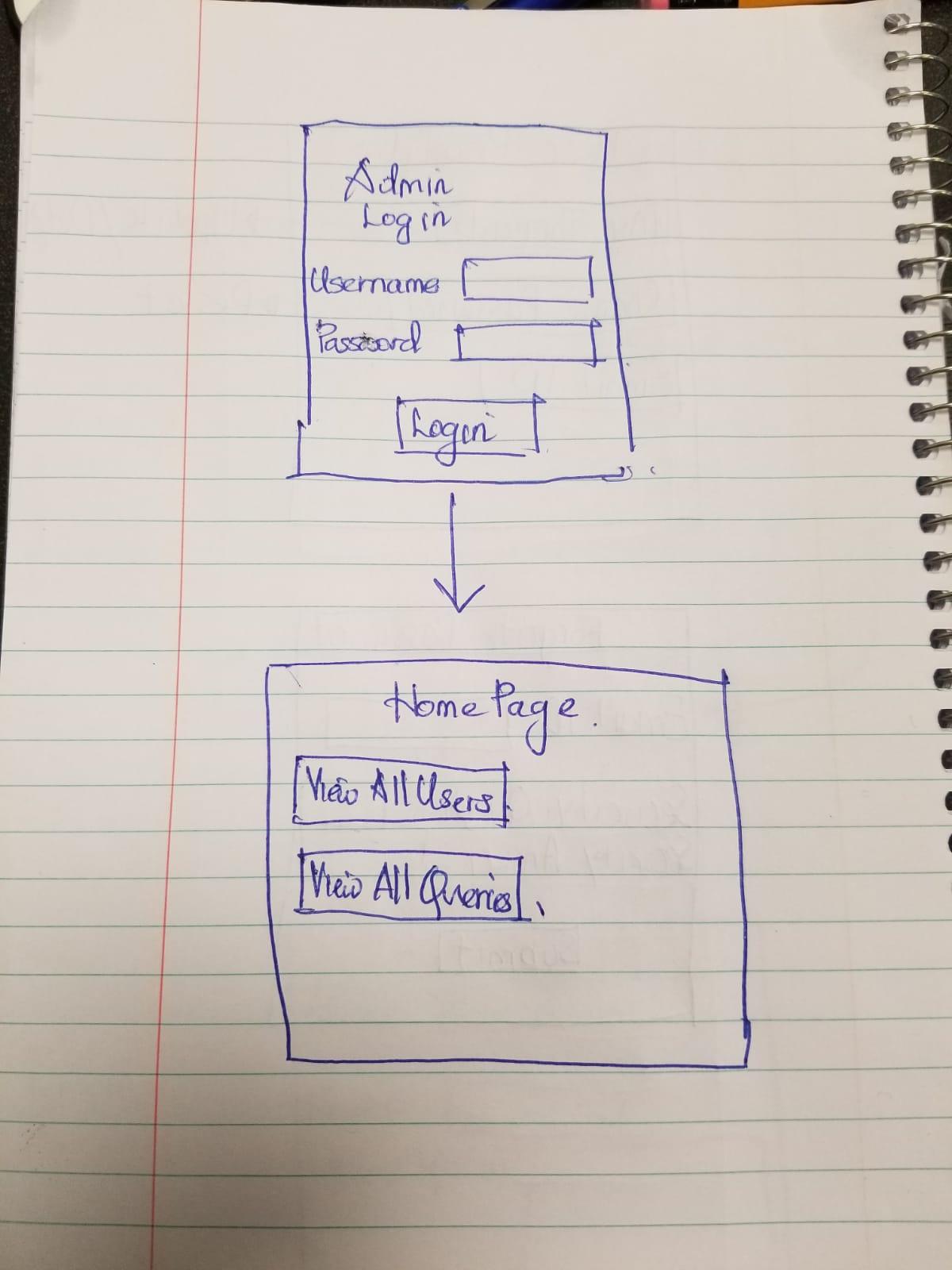
*Fig. 2.7 Provider/Seeker Story Card 6*

**Low Fidelity Prototype as an Image**

**

**

**

**

### 2.4.2  Additional Documentation

User Role video and Iteration-1 Video link:

<https://drive.google.com/drive/folders/1XPfPi7wRolcz2ruXYA32jbpPFjChasyU?usp=sharing>

Release 2 Video link:

<https://drive.google.com/open?id=1b7_BG-PlwnTkc4kft4g4Vq-rrytnQ48v>

2.4.3  Release Plan 1.0

**Must-Have stories for Release 1.0 [1]**

|  |  |
| --- | --- |
| **Story** | **Estimated Time (hours)** |
| The user can login | 1 |
| The user can register for an account | 1 |
| A user can search for a query | 3 |
| A user can post a query | 3 |
| A user can reset their password | 2 |
| A user can set their profile | 2 |
| A user can select the category and get filtered queries | 4 |
| A user can reply to posted queries | 3 |
| A user can communicate through email with the provider | 2 |
| A user can provide feedback | 3 |

**Should-Have Stories Release 1.0 [1]**

|  |  |
| --- | --- |
| **Story** | **Estimated Time (hours)** |
| A user should be able to view all threads | 4 |
| A user can customize their profile | 3 |
| A user will be given a two-factor authentication for registration | 4 |
| Admin can view all the users | 3 |
| Admin can remove users | 3 |

**Could-Have Stories Release 1.0 [1]**

|  |  |
| --- | --- |
| **Story** | **Estimated Time (hours)** |
| Admin can generate a statistical report | 4 |
| A user can customize their profile | 3 |
| A user can communicate through direct messaging with other users | 5 |

### Release Plan 1.0 Iteration Cycle

|  |  |
| --- | --- |
| **Iteration 1** | **Iteration 2** |
| User can signup | User can reset password |
| User can login by username and password | Communication module |
| Post query | Two factor authentications |
| Search Query | Edit and manage queries |
| Feedback | User can Customize their profile |
| Admin can view all the users | Filter query by selecting categories |
|  | User can search for queries by entering tags or keywords. |
|  | Admin can remove users |

### 2.4.4  Iteration Plan (Release 1.0)

|  |
| --- |
| **Iteration 1** |
| User can signup |
| User can login by username and password |
| Post query |
| Search Query |
| Feedback |
| Admin can view all the users |

Table 1: Disaggregated Task - User Sign-Up [1]

|  |  |  |  |
| --- | --- | --- | --- |
| **Task 1** | **Who** | **Estimate** | **Actual** |
| Code and Design Basic Signup Screen | Jasneet Kaur | 30 min | 30 min |
| Develop Tables For Signup | Neha Sebastian | 30 min | 30 min |
| Write and tune SQL to Query the database for Basic Sign up | Gurjit Singh | 1hr | 1hr |
| Develop API for Sign Up | Oreeba Badar | 1hr | 1hr |
| API for Sign up is tested | Amrinder Singh | 30 min | 30 min |
| API for Signup is Implemented in Android Studio | Gurjit Singh | 2 Hrs | 2 Hrs |

Table 2: Disaggregated Task - User can login by Username and Password [1]

|  |  |  |  |
| --- | --- | --- | --- |
| **Task 2** | **Who** | **Estimate** | **Actual** |
| Code and Design Basic Login Screen | Jasneet Kaur | 30 min | 30 min |
| Develop Tables for Login | Amrinder Singh | 30 min | 30 min |
| Write and tune SQL to Query the database for Basic Login Screen | Gurjit Singh | 2 Hrs | 2 Hrs |
| Develop API for Login | Neha Sebastian | 2 Hrs | 2 Hrs |
| API for Login is tested | Amit Roy | 1 Hr | 1 Hr |
| API for login is implemented in Android Studio | Oreeba Badar | 2 Hrs | 2 Hrs |

Table 3: Disaggregated Task - Post Query [1]

|  |  |  |  |
| --- | --- | --- | --- |
| **Task 3** | **Who** | **Estimate** | **Actual** |
| Code and design Post Query Screen | Jasneet Kaur | 30 min | 15 min |
| Create Tables for Post Query | Oreeba Badar | 30 min | 15 min |
| Write and tune SQL to Query the database for Post Query | Neha Sebastian | 2 Hrs | 2 Hrs |
| Develop API for Post Query | Amrinder Singh | 2 Hrs | 2 Hrs |
| API for Post Query is tested | Gurjit Singh | 2 Hrs | 2 Hrs |
| API for Post Query Is Implemented In android Studio | Jasneet Kaur | 2 Hrs | 2 Hrs |

Table 4: Disaggregated Task - Search Query [1]

|  |  |  |  |
| --- | --- | --- | --- |
| **Task 4** | **Who** | **Estimate** | **Actual** |
| Code and Design Search Query Screen | Oreeba Badar | 30 min | 15 min |
| Develop Tables for Searching the Query | Amit Roy | 30 min | 20 min |
| Write and tune SQL to Query the database for Basic Search | Amrinder Singh | 2 Hrs | 2 Hrs |
| Develop API for Search Query | GurjitSingh | 2 Hrs | 2 Hrs |
| API for Search Query is tested | Neha Sebastian | 1 Hrs | 1 Hrs |
| API for Search Query is implemented in Android Studio | Jasneet Kaur | 2 Hrs | 1 Hrs |

Table 5: Disaggregated Task - Feedback [1]

|  |  |  |  |
| --- | --- | --- | --- |
| **Task 5** | **Who** | **Estimate** | **Actual** |
| Code and Design Admin Page to see all the users | Neha sebastian | 30 min | 30 min |
| Develop Tables for Feedback | Amrinder Singh | 30 min | 15 min |
| Write and tune SQL to Query the database for Admin Page | Gurjit Singh | 2 Hrs | 1 Hr |
| Develop API for Feedback | Oreeba Badar | 2 Hrs | 1 Hr |
| API for feedback Is Tested | Jasneet Kaur | 1 Hr | 1 Hr |
| API for Feedback is implemented in Android Studio | Amit Roy | 1 Hr | 30 min |

Table 6: Disaggregated Task - Admin can view all the Users [1]

|  |  |  |  |
| --- | --- | --- | --- |
| **Task 6** | **Who** | **Estimate** | **Actual** |
| Design the admin Webpage | Jasneet Kaur | 2 Hr | 2Hr |
| Write and tune SQL to Query the database for Basic Search | Gurjit Singh | 1 Hr | 45 min |
| API for Admin is Tested | Neha Sebastian | 1Hr | 30 min |
| API for Admin is implemented in Android Studio | Oreeba Badar | 1Hr | 30min |

### 2.4.5 Additional Documentation

Videos of Iteration planning meeting:

User Role video and Iteration-1 Video link:

<https://drive.google.com/drive/folders/1XPfPi7wRolcz2ruXYA32jbpPFjChasyU?usp=sharing>

Release 2 Video link:

<https://drive.google.com/open?id=1b7_BG-PlwnTkc4kft4g4Vq-rrytnQ48v>

### 

### 

### 

### 

### 2.4.6  Progress Monitoring

|  |  |  |
| --- | --- | --- |
|  | **Iteration 1** | **Iteration 2** |
| Story points at the start of iteration | **120** | **40** |
| Completed during iteration | 90 | 40 |
| Changed estimates | 6 | 0 |
| Story points from new stories | 4 | 0 |
| **Story points at the end of iteration** | **40** | **0** |

### 

### 3.0 CONCLUSIONS

The app creates a platform to get help faster without booking an appointment and wait for the slot and hence both the seeker and provider gets benefit from it. It benefits the provider since provider can get monetary benefits or bonus points to earn a volunteering certificate.

# 4.0 RECOMMENDATIONS

Any registered members of this app can get help in every area of study from the users of the app. Hence, we can utilize the talents of bright minds and efficient faculties of the college. It also provides a platform for interaction among people in the college. The solution seekers post query for help in the app and all the registered members get notification and can provide help for the user through both chat support and face to face interaction.

# CREDITS, LICENSE, AND REFERENCES

## Credits

Neha Sebastian< nsebast6@my.centennialcollege.ca>

Jasneet Kaur < jkaur327@my.centennialcollege.ca>

Oreeba Badar < obadar@my.centennialcollege.ca>

Gurjit Singh< gsin1109@my.centennialcollege.ca>

Amit Roy< hgandh13@my.centennialcollege.ca>

Amrinder Singh < asin1111@my.centennialcollege.ca>

## License

The Software may be used only for Licensee’s internal business use. Licensee shall not, nor permit any third party to, reverse engineer,decompile, disassemble or otherwise reduce the Software to any other humanly perceivable form, and may not modify, adapt, rent, lease, loan or create derivative works based upon the Software or any part thereof, except as expressly permitted by this Agreement. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts.

## References

[1] Cohn, Mike. 2004. *User Stories Applied: For Agile Software Development*, Addison-Wesley Professional.

<https://firebase.google.com/docs/>

<https://www.tutorialspoint.com/android/android_sending_sms.htm>

<https://stackoverflow.com/>

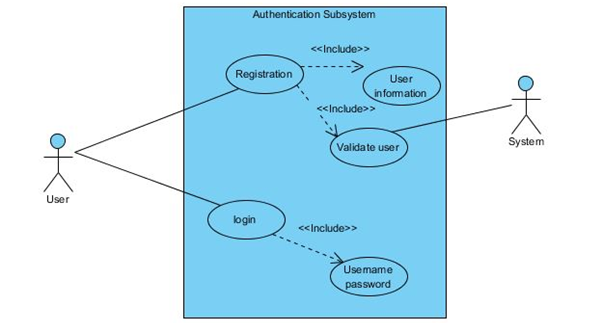
<https://developer.android.com/docs/>

<https://docs.angularjs.org/guide>

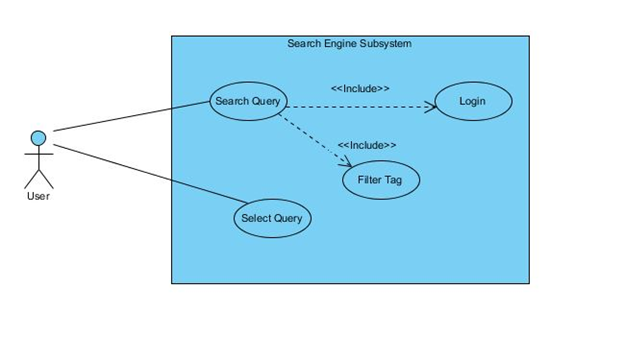
# APPENDIX A (DESIGN DOCUMENT)

Use cases:

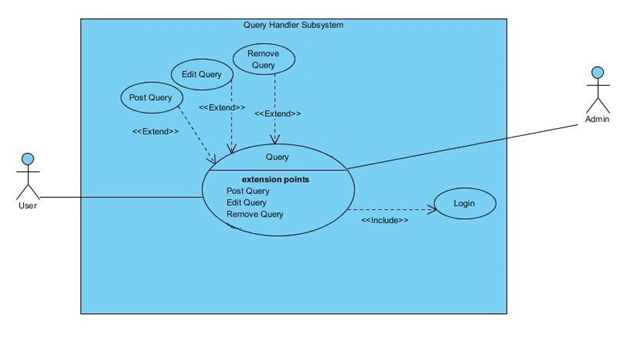
1.Authentication:



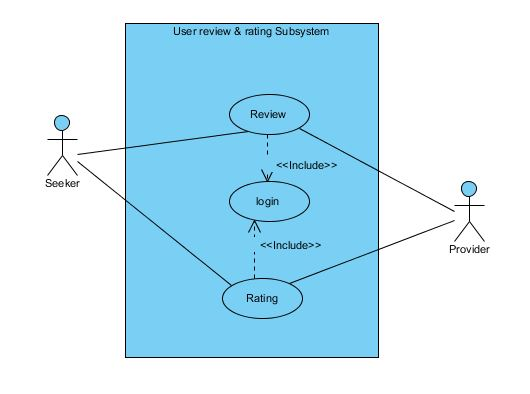
2. Search Query



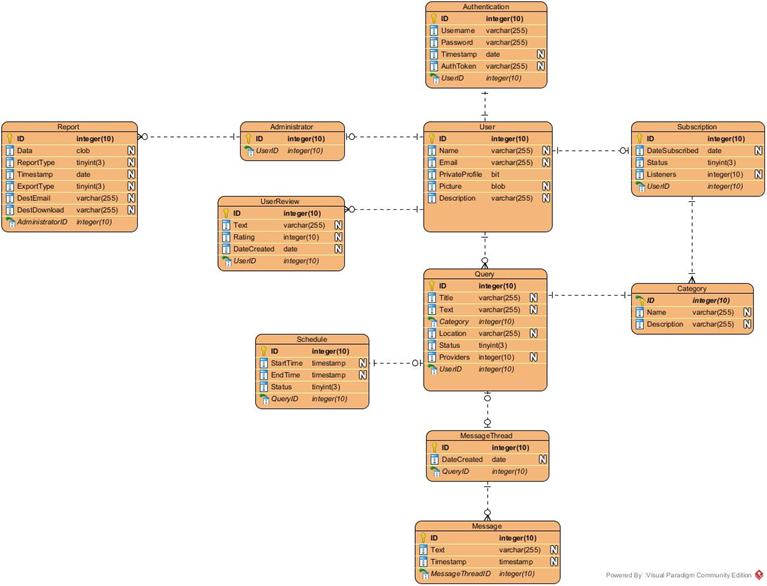
3.Query management:



4.User rating and comments:

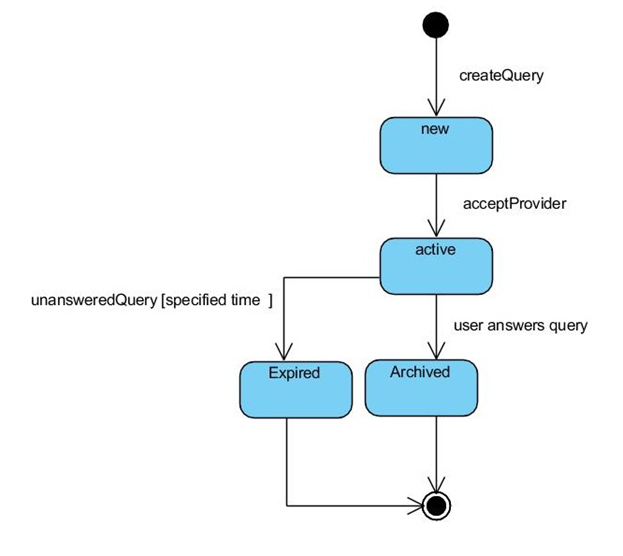


Entity Relationship diagram:

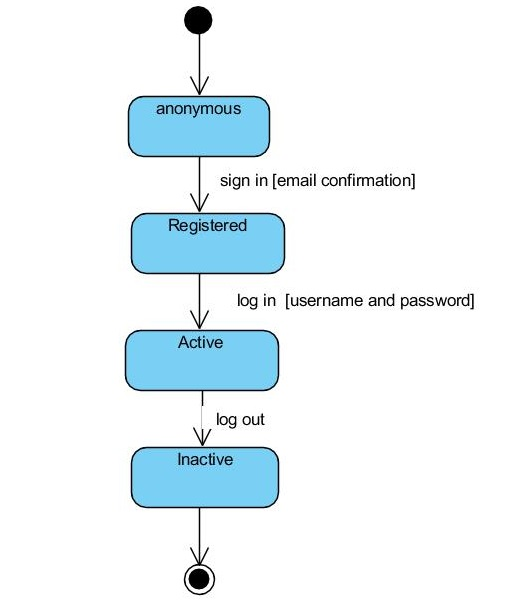


State machine diagrams:

1.Query State:



2.User State:



# APPENDIX B (TEST PLAN)

## 1.0 Introduction

### 1.0.1   Goals

Summarize the testing goals for project.

The goal for the testing HelpStack app is to deliver flawless and ready to use app for enthusiast users, so the app can fulfill the user needs.

### 1.0.2   Assumptions

It is assumed that students and faculty both are the users for this app. they both can be a provider or seeker.

### 1.0.3   Risks And Assets

Risks:

The realtime cloud application firebase is used in the application, all configurations must be checked at the frontend and backend. Wrong configuration may affect the functioning of the application.

Assets:

The backend cloud application on firebase has console view that might help is the data is being saved or not. Moreover, firebase also allows to make efficient code using inbuilt libraries.

## 2.0 Scope

During the testing following things will be tested:

1. User Interface
2. API code
3. Backend Services

### 2.0.1   Features To Be Tested

Main features to be tested in the app are:

1. Registration confirmation by email
2. Search Query by filters
3. Post Queries regarding similar Queries.
4. Security Questions

### 2.0.2    Features Not To Be Tested

None

## 3.0 Testing Procedures

Describe the testing procedures that project will use. This includes the test lifecycle, types of testing, test objectives, and test criteria.

### 3.0.1   Test Objectives

Key Objectives for app testings:

1. Platform will be use
2. Emulators software are required
3. Mobile devices are required.
4. Compatible operating system for the app.

### 

### 

### 

### 

### 

### 

### 3.0.2   Types Of Testing

Describe the types of testing that the project will use.

3.0.2.1   Unit Testing

#### Able to register for the Help Stack application

#### o Able to sign in/out of the Help Stack application

#### o Able to retrieve forgotten password

#### - Able to post queries

#### o Can update queries

#### o Able to delete queries

#### o Search for queries using filters provides

#### - Able to receive notifications

#### - Able to communicate with help provider through a messaging thread

#### o option to email person providing the help

#### 3.0.2.2   Integration Testing

● Verify access to database

● Verify cloud connectivity of web and mobile application

● Verify correct retrieval of update of database data.

· Verify proper privileges

· Verify correct retrieval of update of database fields

#### 3.0.2.3   Acceptance Testing

Admin can view all the users

· Admin can edit and delete users

· User can add posts

· users can view and edit queries

· A user can search from a query using keywords

#### 3.0.2.4   Stress Testing

The minimum amount of memory that need is 5 Mb disk space.

#### 3.0.2.5   Performance Testing

Refer to the functional requirements that specify acceptable performance.

1. Loading time after login.
2. Posting Queries to the database.
3. Response time.

### 3.0.3   Testing Tools

Describe the tools that you will use for testing

1. Android studio
2. Firebase console
3. Selenium for testing Admin Panel

## 

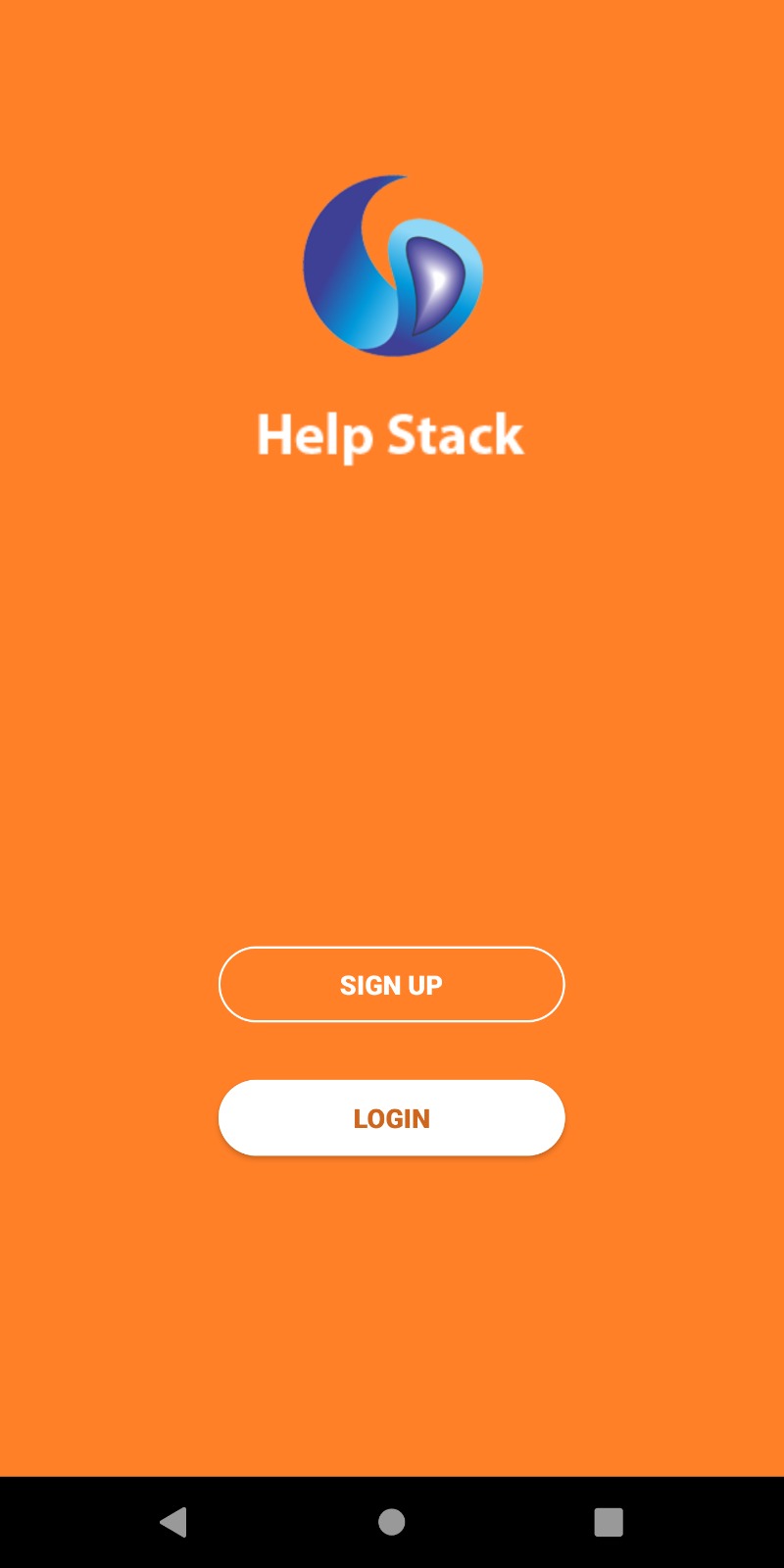
## 4.0 Schedule and Deliverables

|  |  |
| --- | --- |
| Testing | Date |
| Integration Testing | 10.12.2018 |
| Unit Testing | 8.12.2018 |
| Performance Testing | 9.12.2018 |
| Stress Testing | 12.12.2018 |

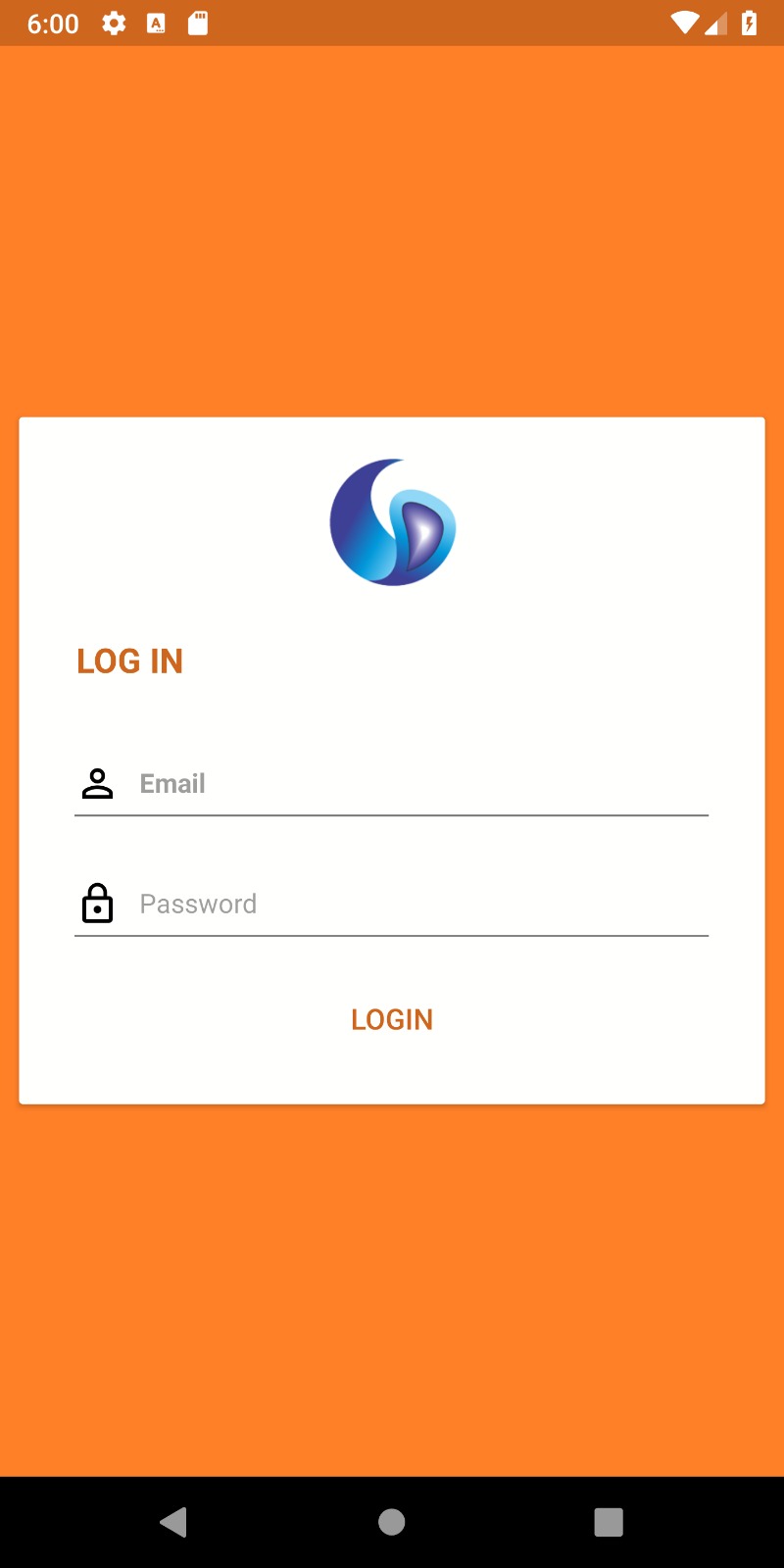
# APPENDIX C (END-USER & ADMINISTRATOR MANUALS)

The software can be installed in any android phone and below are the screenshots for a user to use the app:

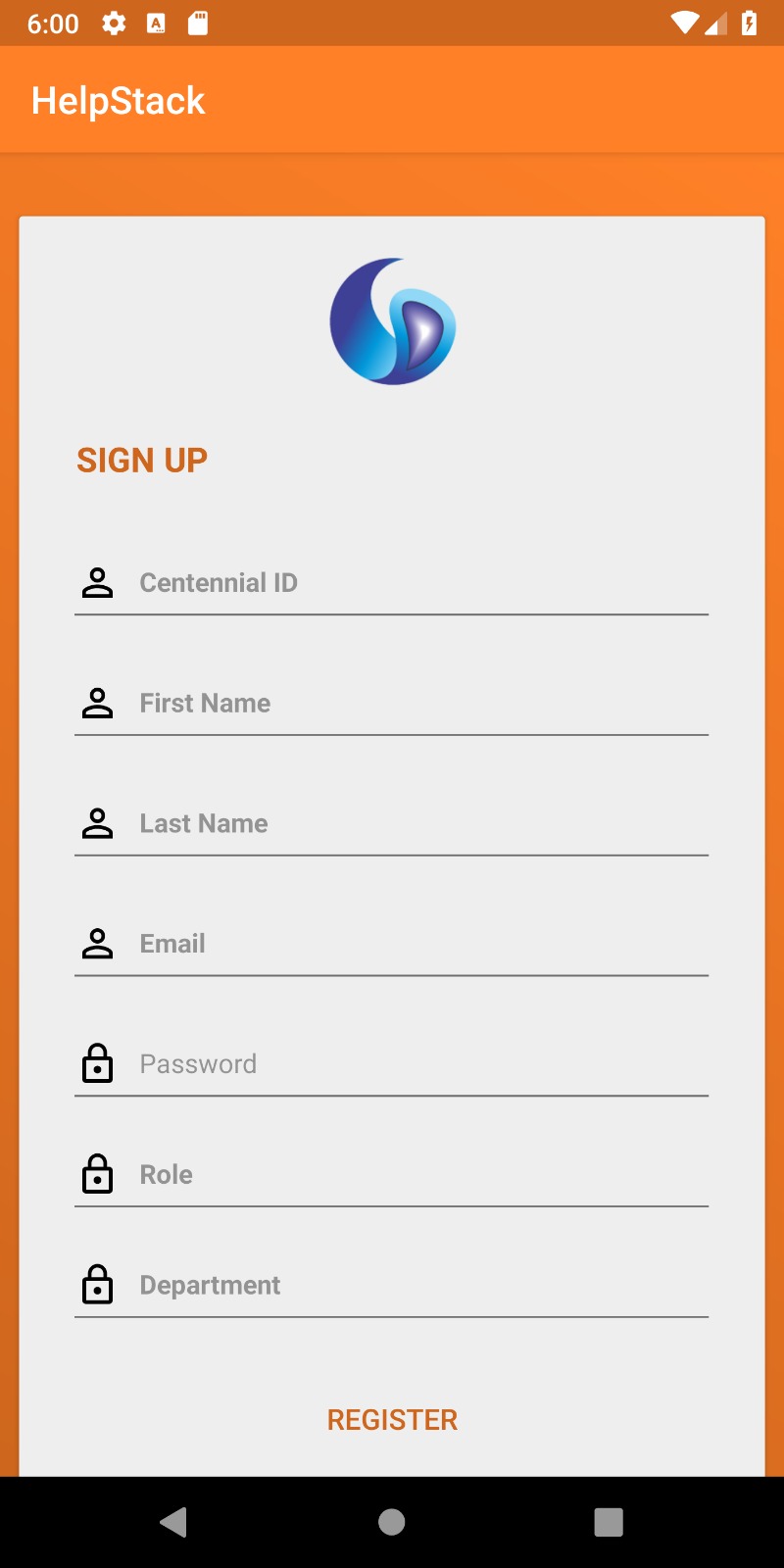
1. Front screen for User to Login or Sign up:



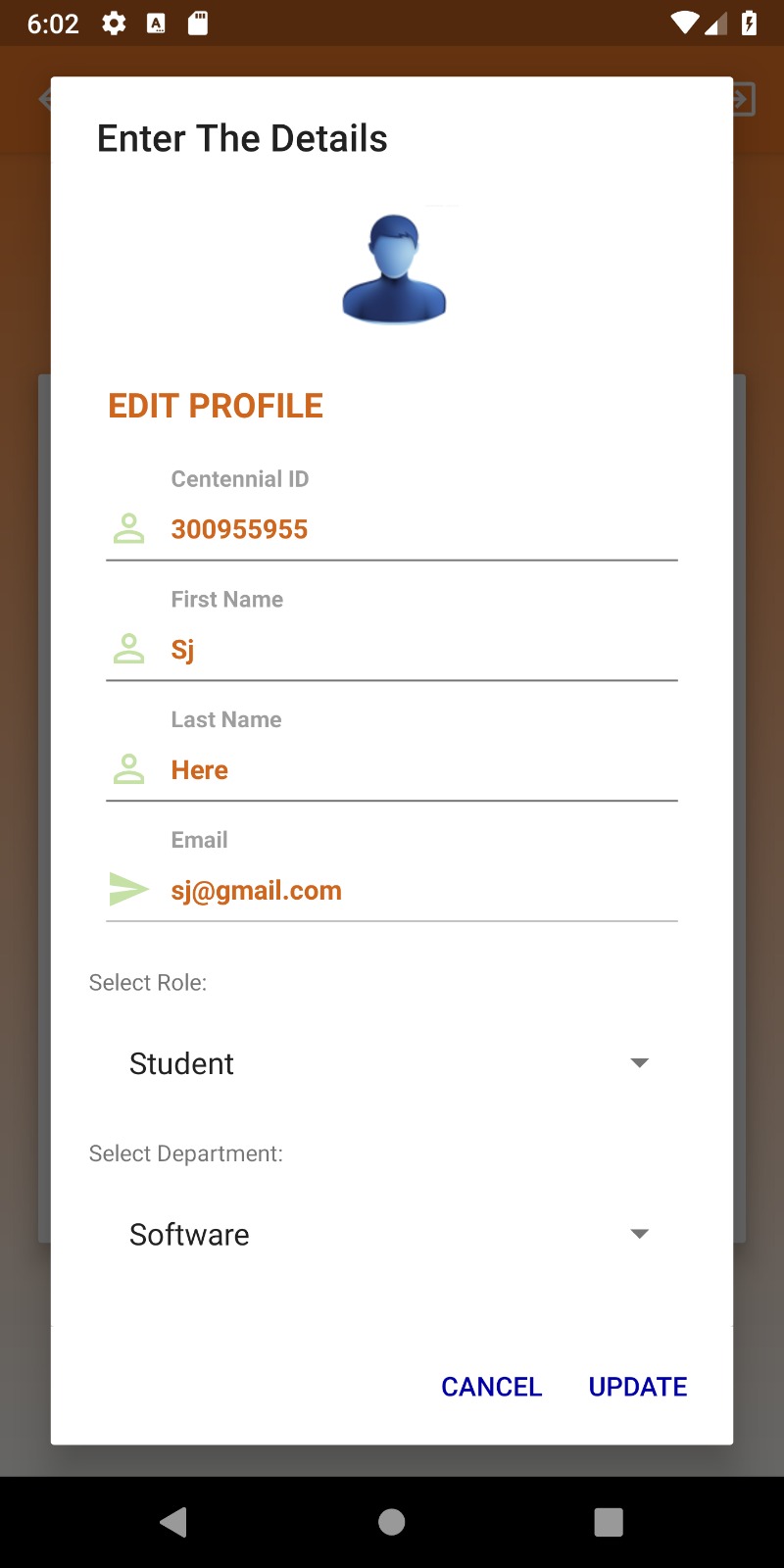
2. User Login screen:

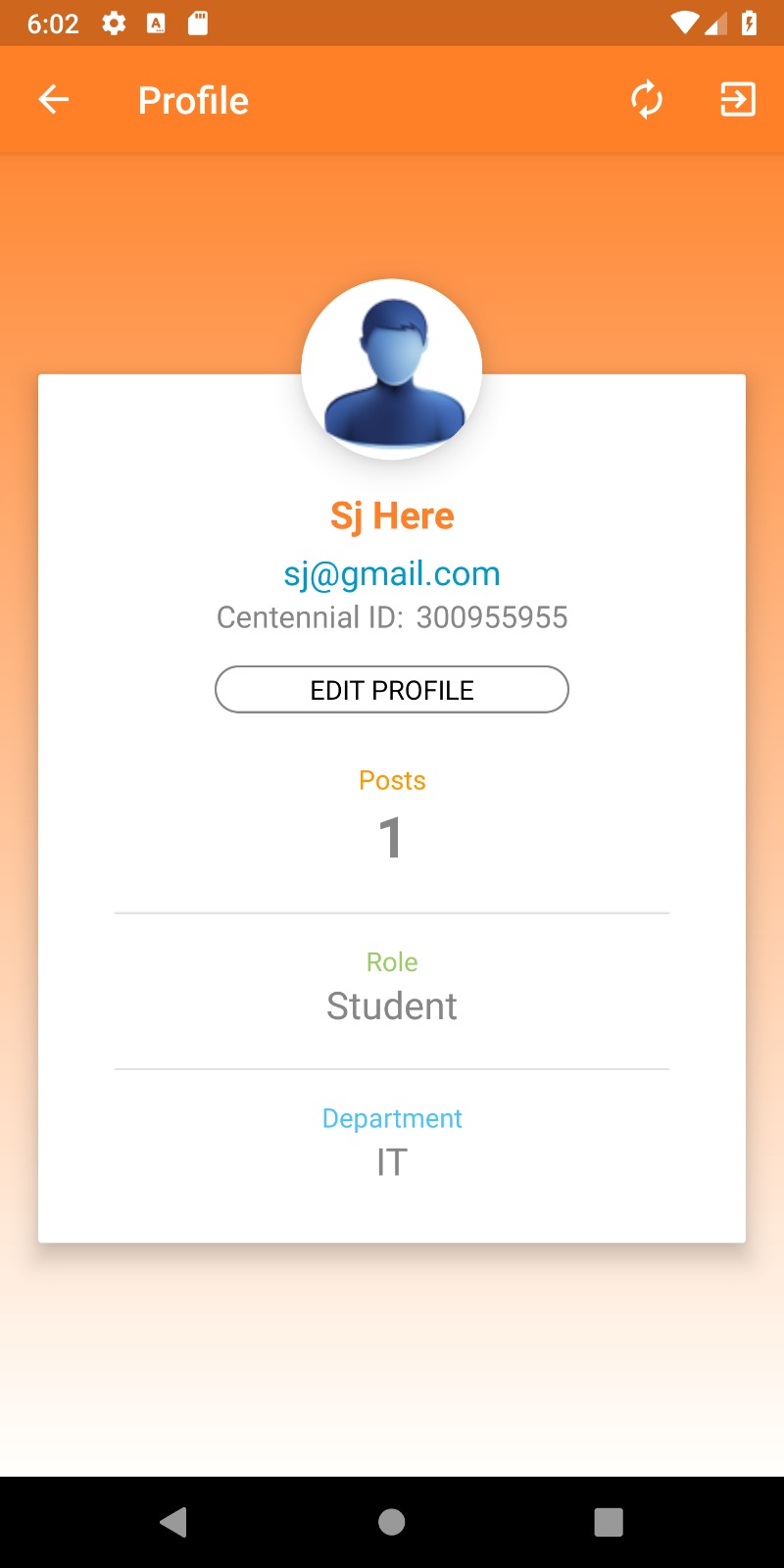


3. User Registration Screen:

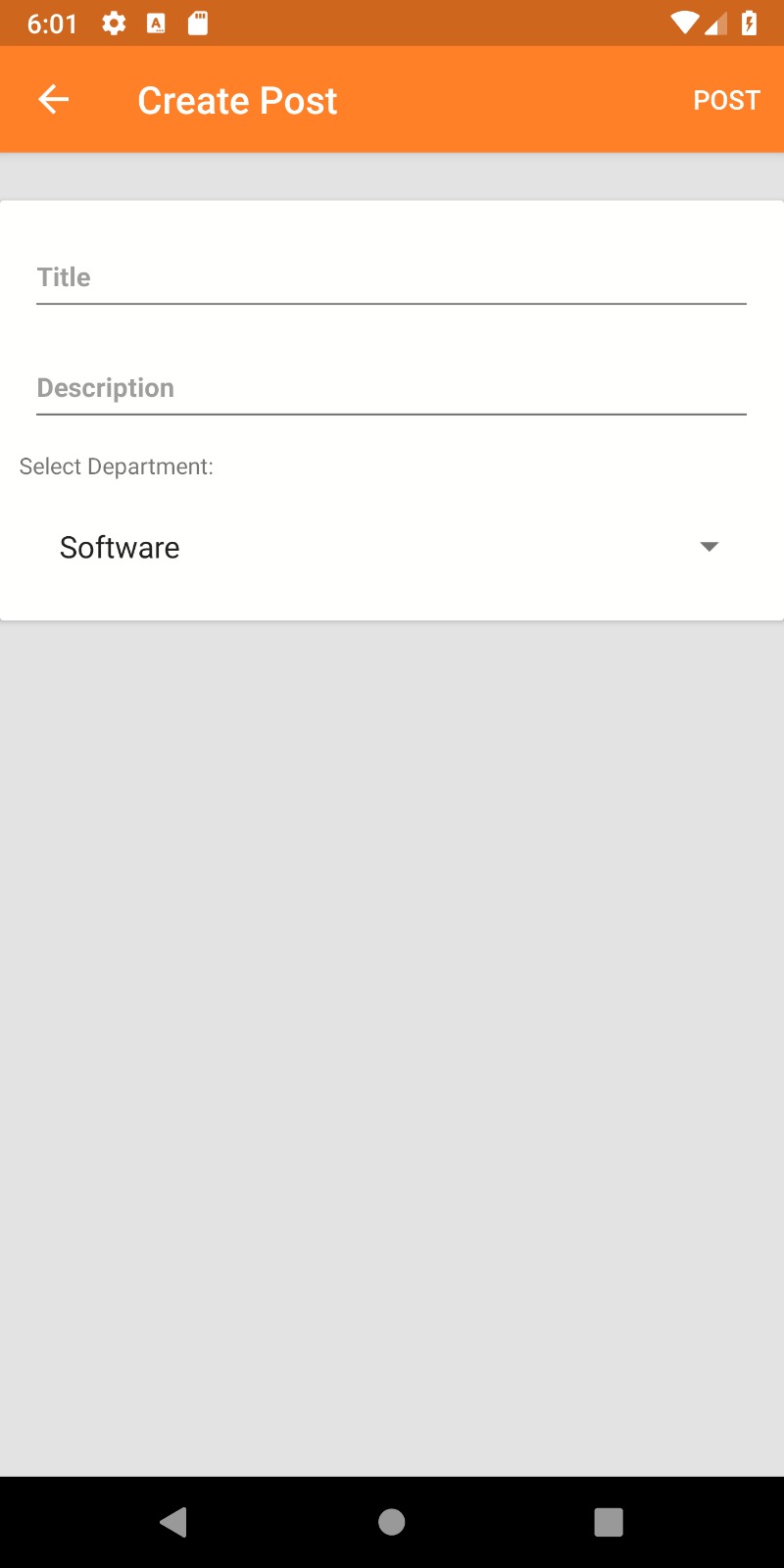


3. Edit Profile screen:

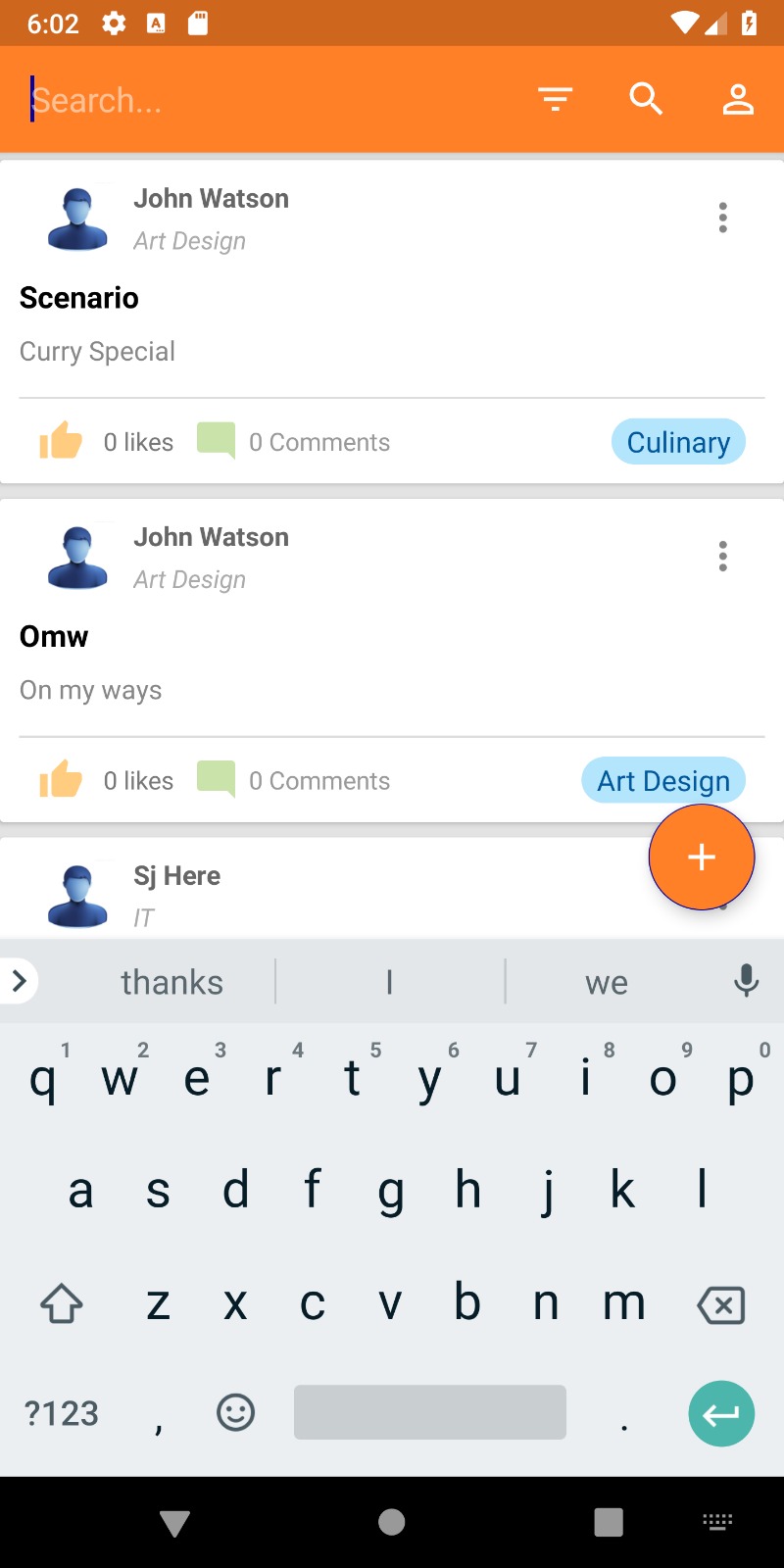


4. User can view his profile screen:

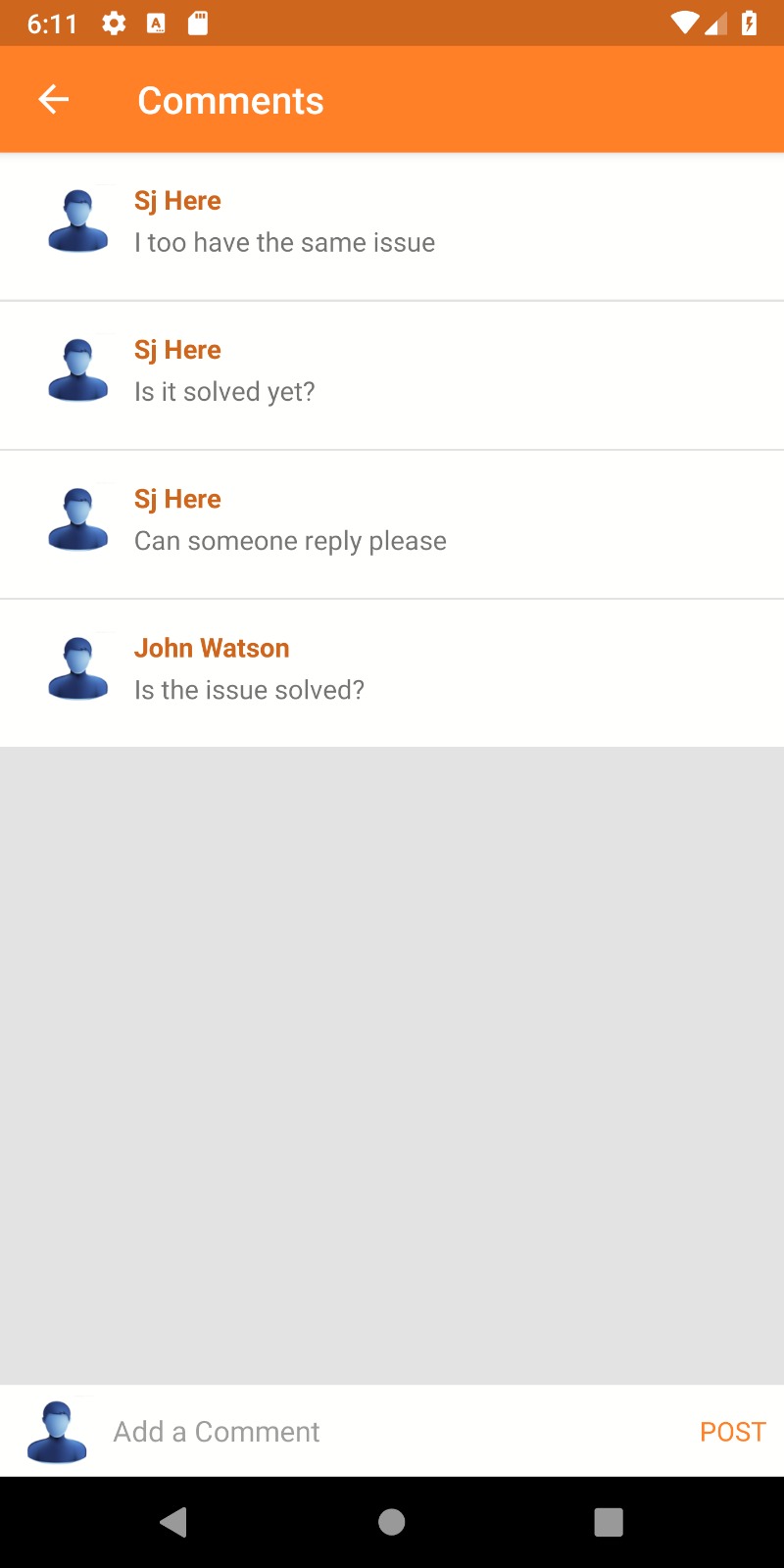
5. User can Post query :



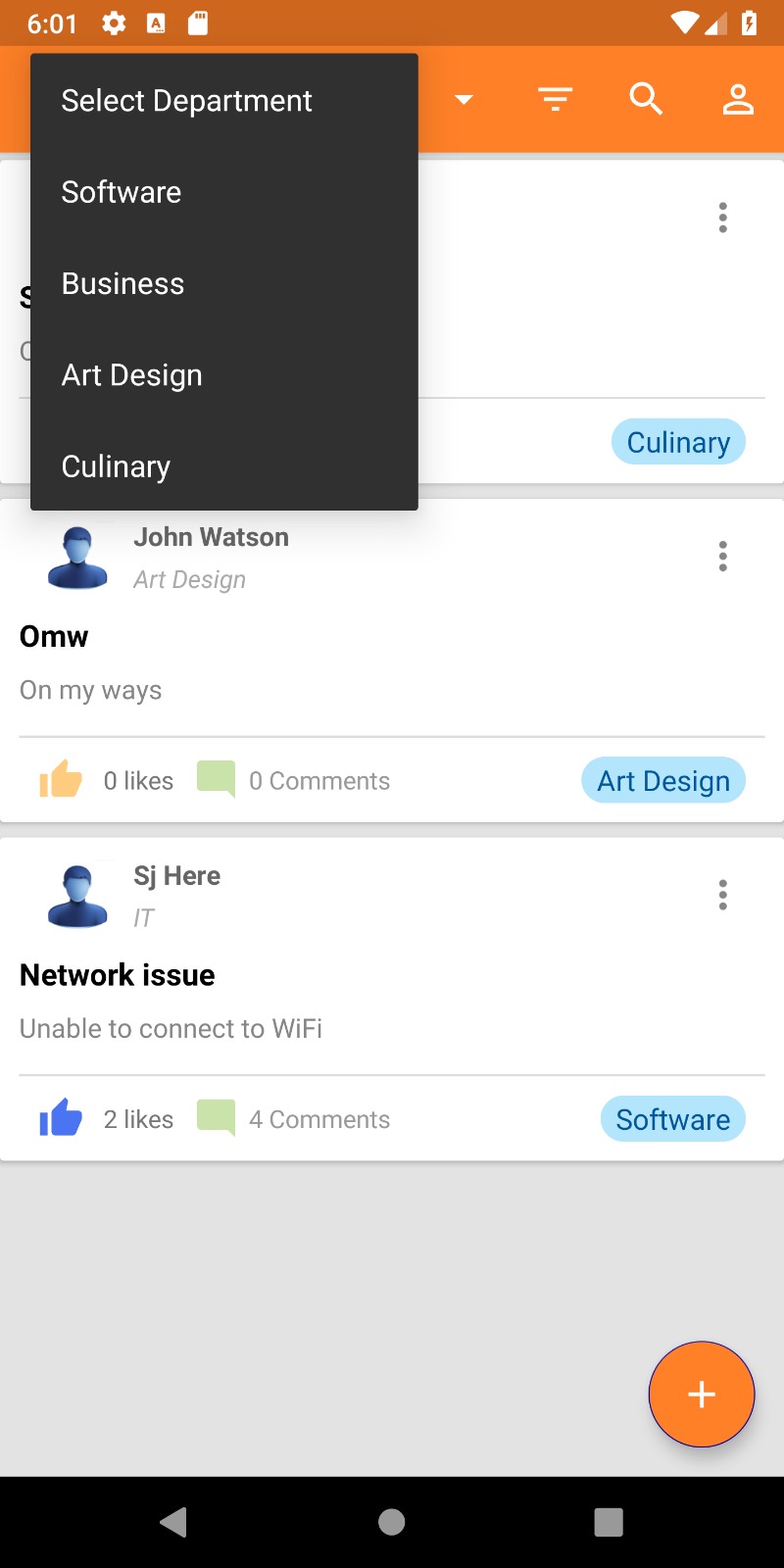
6. View all queries screen:



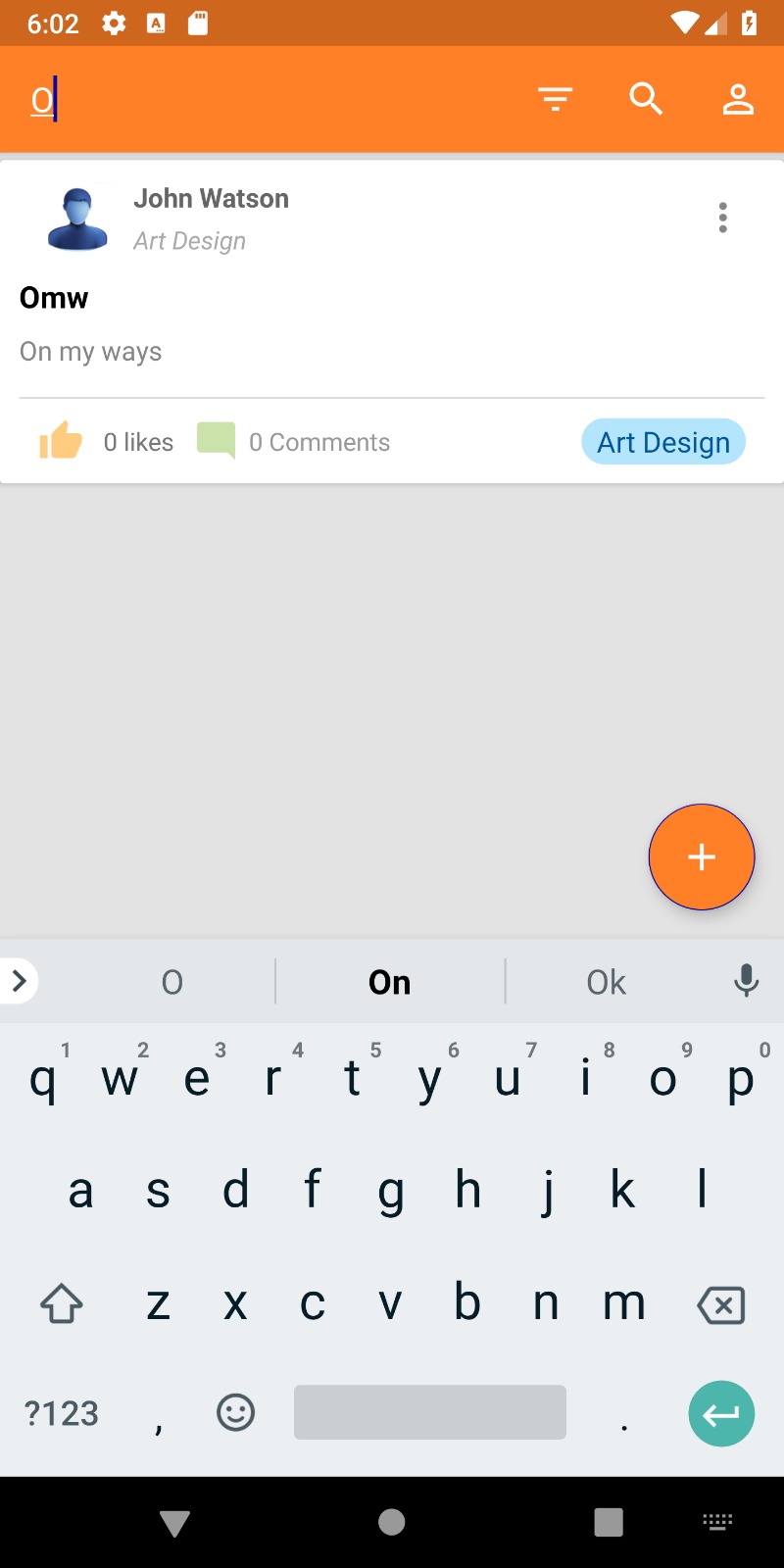
7. comments on a particular posted query screen:



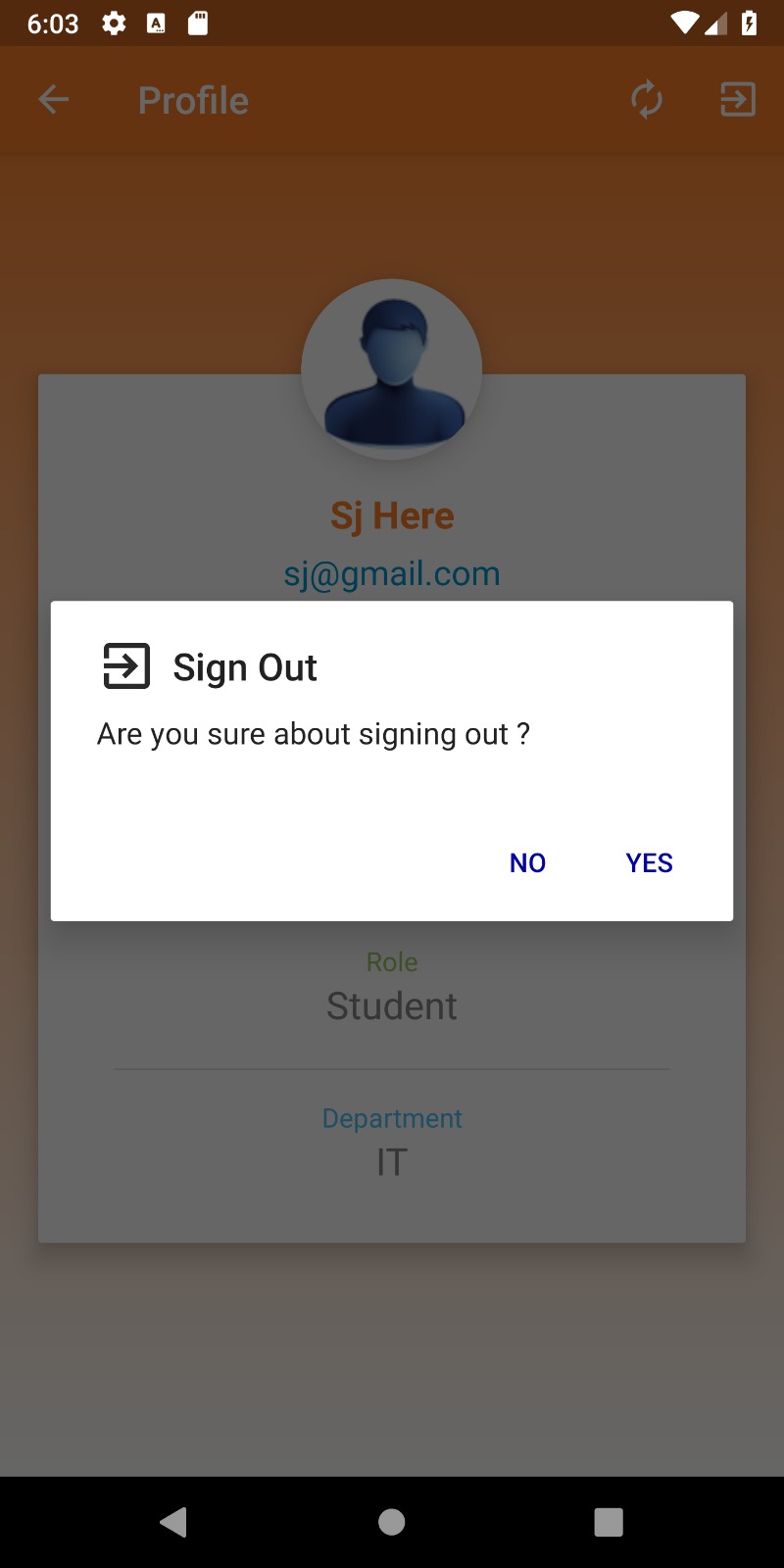
8. Search by Department Screen:



9. Search by Tag Filter:

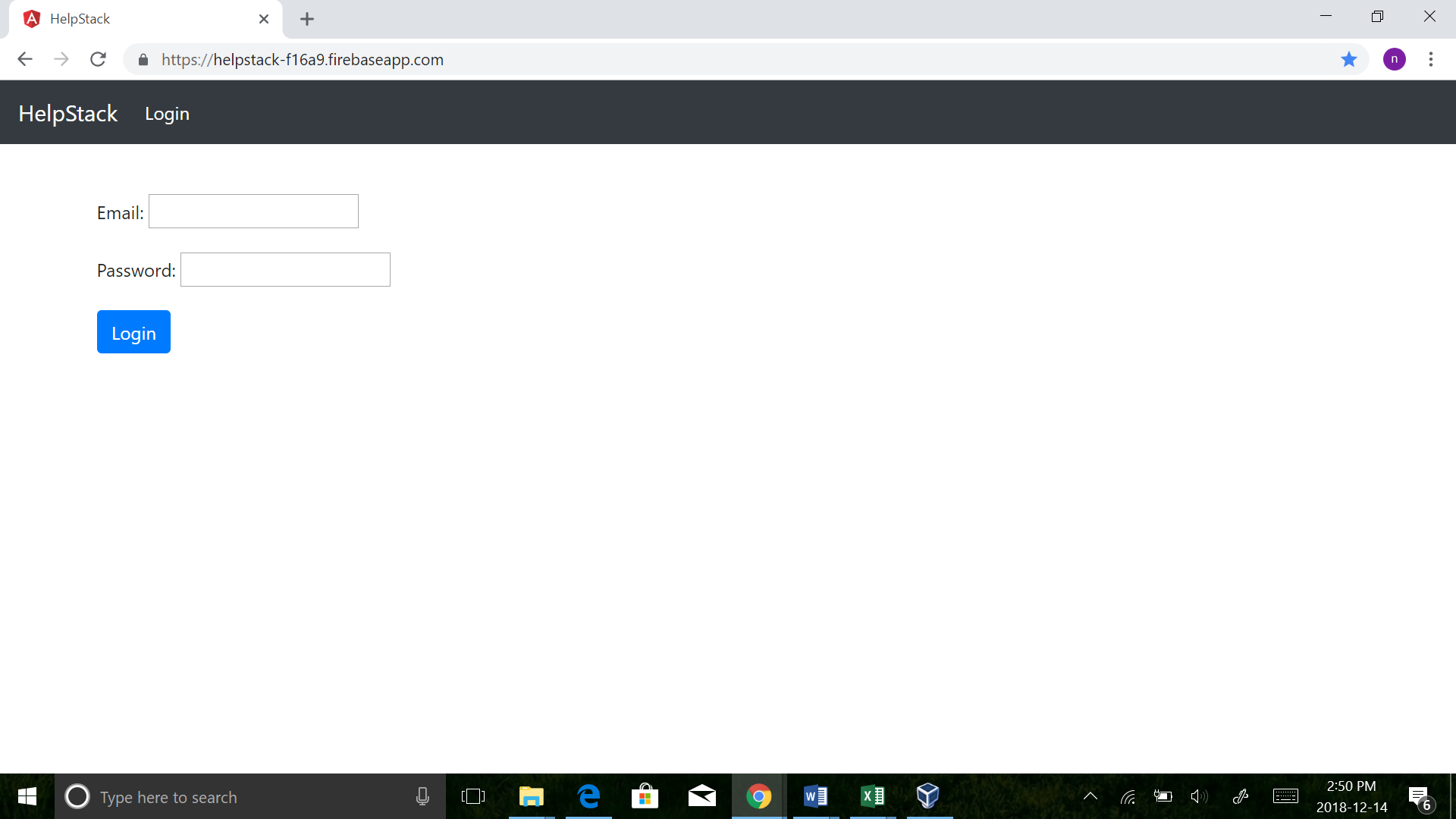


10. Sign out Screen:

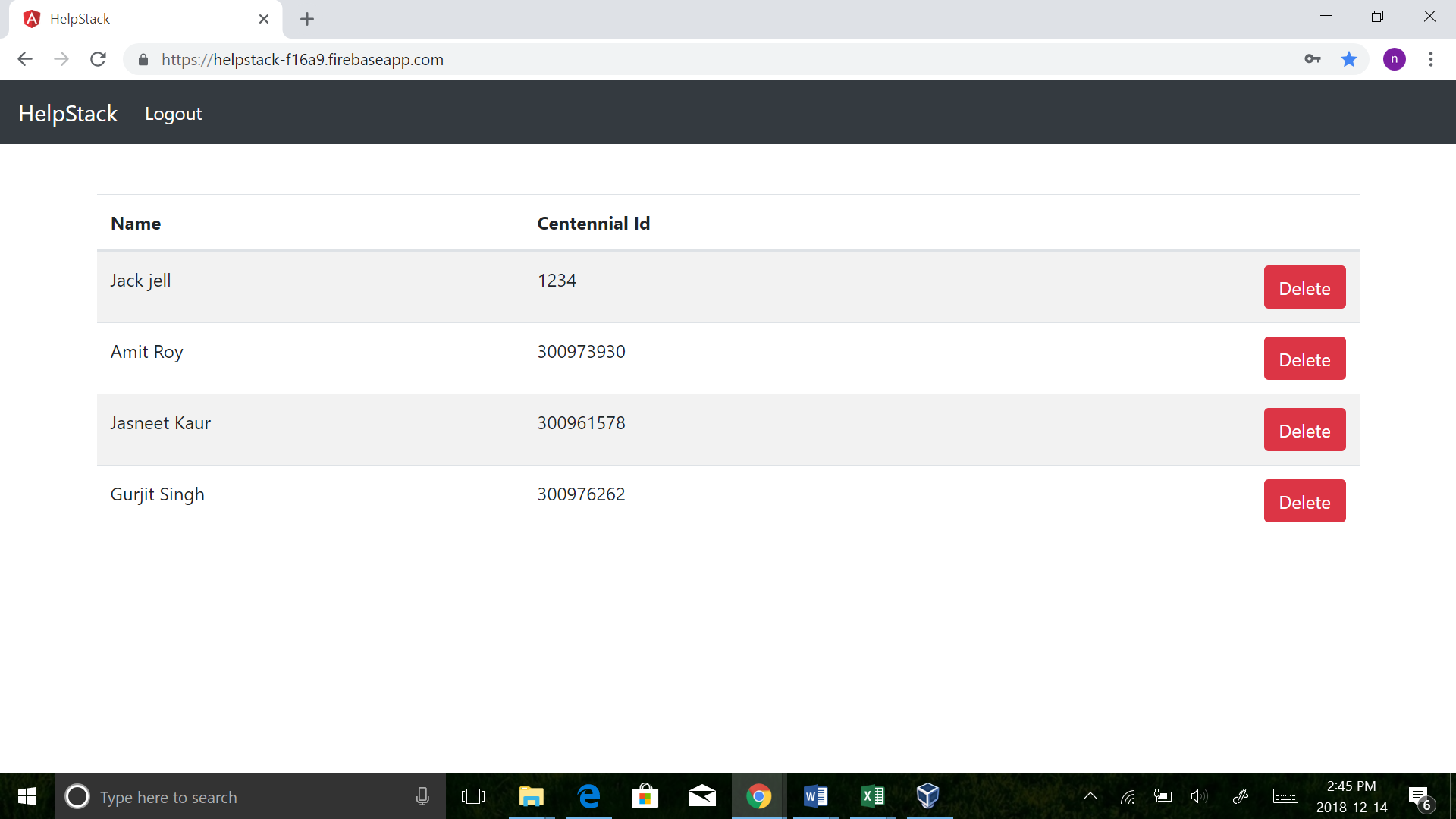


Admin page:

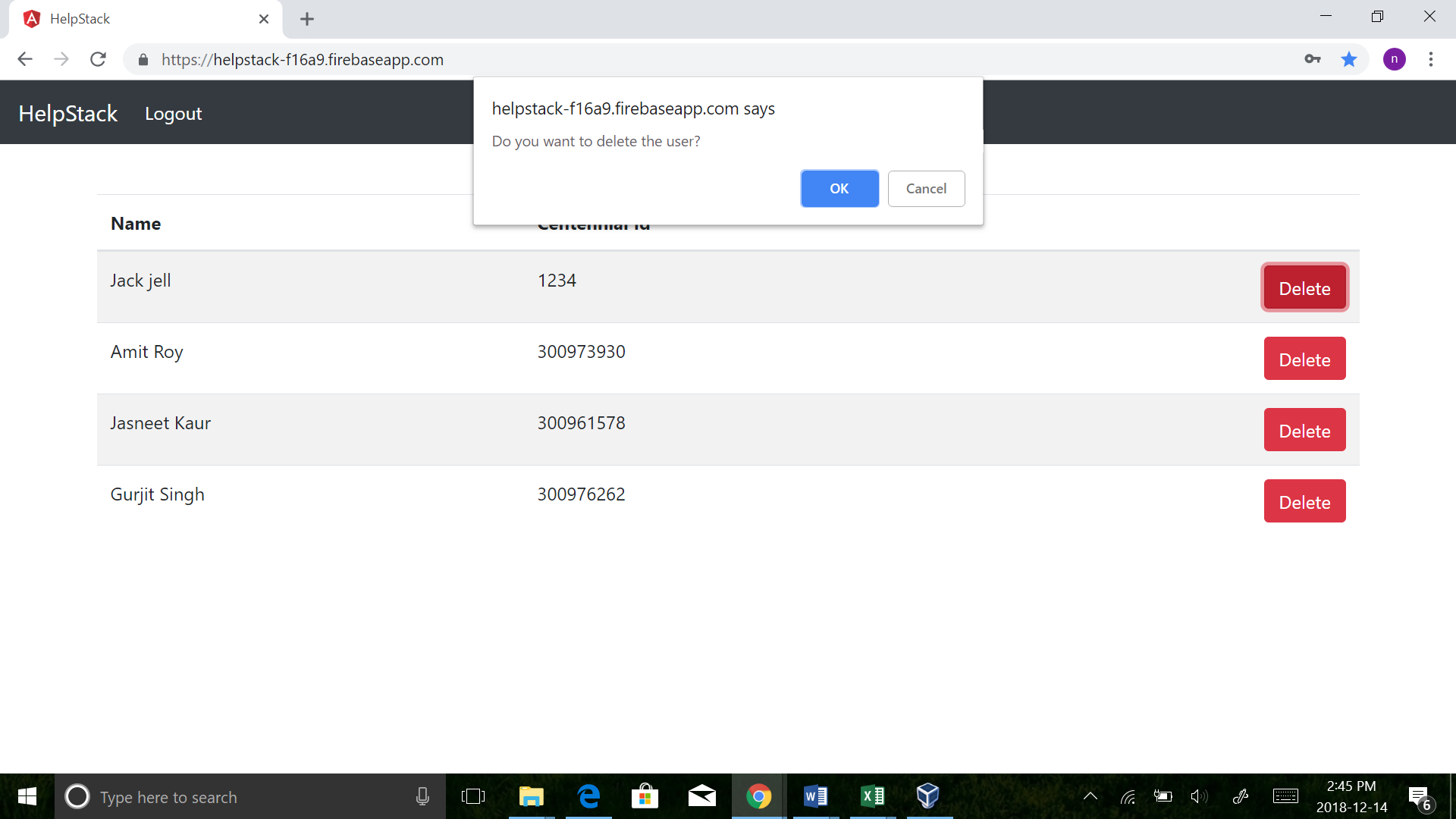
1.Admin can login with username and password



2.Admin can view all the users:



3.Admin can delete the users:



4.User deleted by admin:

