

Department of Informatics University of Leicester CO7201 Individual Project

**Preliminary Report**

**QUORA FOR COLLEGE STUDENT COMMUNITY**

## Prerana Gobbur

**Pg202@student.le.ac.uk**

**Pg202**

**Project Supervisor: Dr Severi Paula**

**Second Marker: Dr Zarbaf Monasadat**

**Word Count: 1114**

**Friday,1st July,2022**

### DECLARATION

#### All sentences or passages quoted in this report, or computer code of any form whatsoever used and/or submitted at any stages, which are taken from other people’s work have been specifically acknowledged by clear citation of the source, specifying author, work, date and page(s). Any part of my own written work, or software coding, which is substantially based upon other people’s work, is duly accompanied by clear citation of the source, specifying author, work, date and page(s). I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this module and the degree examination as a whole.

#### Name: Prerana Gobbur

#### Date: friday, 1st July, 2022

Contents

1. [Aims and Objectives 4](#_TOC_250006)
2. [Requirements 4](#_TOC_250005)
3. [Technical Specification 4](#_TOC_250004)
4. [Requirements Evaluation Plan 5](#_TOC_250003)
5. [Background Research and Reading list 5](#_TOC_250002)
6. [Time-plan and Risk Plan 6](#_TOC_250001)
7. [References 8](#_TOC_250000)

# Aims and Objectives

Many college students stuck in their questions and want reliable answers to their questions. Quora related websites swiftly rose to prominence as one of the most popular websites on the internet. These are the sites where students may post inquiries and receive responses. In addition, the programme allows users to respond to questions posed by other users, modify and update their responses over time, and arrange them. You will face several difficulties when creating a question-and-answer website. You may have challenges attracting users, in addition to certain technological concerns.

# Requirements

##### High Level Requirements:

The user will be able to ask a question and other users are free to answer the questions. In addition to this, events can be created, and question can have the time frame to answer. Leaderboard is provided based on the highest voted question among the university and highest voted answer. Questions can be searched with the help of tags and other key words. Ability to post the blogging articles.

##### Essential Requirements:

* 1. Efficient data fetching for displaying the relevant content, feed should be infinite scroll, so data at once cannot be retrieved through an API. Any methodology for spring boot should be implemented to get the data as per screen.
  2. Managing user sessions for 1hr and managing the sessions. Cron job will be executed from spring boot to manage the expired sessions that are left over without logout. User sessions are created based on the userid and the current time.
  3. Updating the content without refreshing, will be doing through infinite scroll feature.
  4. User can be posting a question and questions related to your university are only shown to a user when logged in.
  5. User can tag interests to a question, and you can search the questions through the tags also.
  6. User can comment to a question which is treated as an answer for the question, and highest liked comment will be considered as the best answer for the question.
  7. User can update, delete the comments added for a question.
  8. User can like/dislike a question, comment. These are basically the upvotes and downvotes. These might be helpful for displaying the best answer and fetching the best question.
  9. User cannot update the question once a comment has been posted, and user can delete a question or comment. It will be physical delete in the database.
  10. User will be searching the questions, through tags, time, and user interests.
  11. User can update his profile such as username and date of birth.
  12. Blog can be either liked or disliked, it doesn’t have comments because blog is related to own article writing and opinions of other users are not considered.

##### Recommended Requirements:

1. REST based search capabilities will be implementing either stored procedure for search or search through JAVA.
2. API to handle the large data for the blogging data as the blog data will be Long Text from SQL.
3. Events are the questions but having a time frame associated with it. And other users will be able to answer those questions within that time only.
4. User can update his interests which will affect the questions feed displayed to the user.

##### Optional Requirements:

1. User can share question or answer to different social media sites, link to the question will be shared through social media.
2. Users can get points and badges. Users with highest liked questions will be awarded points and stay top within a university.
3. Leaderboard is displaying the top user whose question has been highly upvoted/liked by other users within the university.

# Technical Specification

##### Programming Languages:

* 1. Java - 8
  2. Javascript - ES6
  3. HTML - 5
  4. CSS - 3
  5. Apache Maven - 3.6.3
  6. Sql

##### Frameworks:

1. Angular 2.x
2. Angular Material
3. SpringBoot - 2.x (Maven Based)

##### Operating System:

1. Windows

##### Cloud Technologies:

1. JBOSS for deployment testing

##### IDE:

1. STS
2. VS Code

##### Version Control:

1. GIT
2. SVN (University hosted)

# Requirements Evaluation Plan

##### Efficiency:

The application’s performance will be tested based on the amount of content it can handle. All the upvotes and downvotes should be handled by the user with ease. Database should be able to handle upvotes and downvotes of blog, posts, and event-based questions. Overall end user should have a smooth experience.

##### Database design:

As the content deals with questions, blog posting, likes, dislikes and comments, having an efficient database design would be very effective in terms of managing and retrieving the data. Handling the data should be very effective when data scales up and with minimal number of stored procedures the work needs to be done.

##### Testing:

The application can be tested by creating number of different university users, creating different type of questions. This can be heavy as most of the content is manual typing. Testing for such a huge number of users might be tough with having different universities under the plate. Having a responsive site which user can be able to use in both web and mobile can be handy.

# Background Research and Reading list

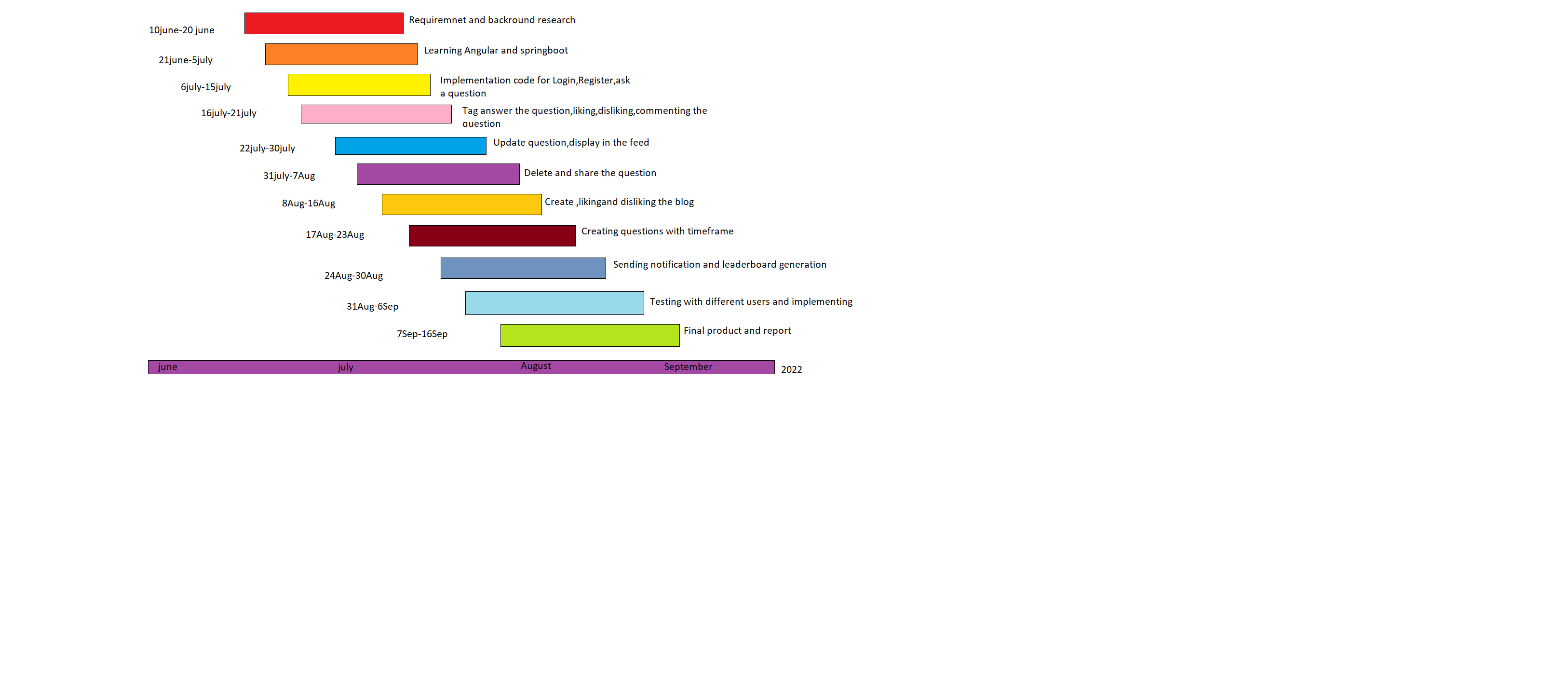
##### Background Research:

The process of creating an efficient database requires significant amount of effort to manage the content that we have. As it should handle both blog and questions under same design framework to avoid redundancy. Apart from these, maintaining the primary keys, having the database clear so that having an additional feature minimizes the change required on the tables. Designing the user interface so that the end user has the flexibility to work on the website on both desktop and mobile. So, creating the folder structure inside the angular framework can be key, having modules can be clear and can make a large difference in fetching and displaying the data. On the other hand, securely handling the backend server so as to create API end points which can handle large amounts of incoming and outgoing dataflow. Socket.io implementation for springboot also need to be tested before implementing. This provides app to be having the notification ability without refreshing.

##### Reading List:

1. Learn about Angular code structure designing patterns - <https://indepth.dev/posts/1232/designing-scalable-angular-applications>
2. Learn about Quora database design platform -<https://datascience.stackexchange.com/questions/113/when-a-relational-database-has-better-performance-than-a-no-relational>
3. Database design – [https://support.microsoft.com/en-us/office/database-design-basics-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5](%20https://support.microsoft.com/en-us/office/database-design-basics-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5)
4. Spring boot overview - <https://docs.spring.io/spring-boot/docs/current/reference/html/using.html>
5. Socket implementation - <https://spring.io/guides/gs/messaging-stomp-websocket/>

# Time-plan and Risk Plan



##### Breakdown of timeline:

|  |  |  |
| --- | --- | --- |
| **Task** | **Planned time taken** | **Explanation** |
| **Requirements and background research** | 10-June-22 – 20-June-22 | Setting requirements of the application and researching about the existing frameworks |
| **Learning Angular and spring boot** | 21-June-22 – 5-July-22 | Exploring angular and spring boot implementations |
| **Implementing Code for Login, register, ask a question**  **Milestone 1** | 6-July-22 – 15-July-22 | Code for login, register and ask a question. Tags will be given to the question while asking. |
| **Tag, answer the question, liking, disliking, commenting the question** | 16-July-22 – 21-July-22 | Code for tagging and other features for a question. Upvote and downvote for the question. |
| **Update question, display in the feed** | 22-July-22 - 30-July-22 | Displaying the feed, this feed comes along with the interests that user possess. For the new feed user has to refresh the page. |
| **Delete and share the question**  ***(Milestone 2)*** | 31-July-22 – 7-Aug-22 | Delete or share the question. Deleting the question doesn’t make the original delete in the db. |
| **Create, liking and disliking the blog** | 8-Aug-22 – 16-Aug-22 | Code for blog handling. This comes as an article writing where user writes a simple article, which the other users are prevented to comment. Other users only allowed to upvote or downvote the article. |
| **Creating questions with time frame (*Milestone 3)*** | 17-Aug-22 – 23-Aug-22 | Code for event handling. We are trying to implement the socket.io implementation on JAVA. Based on the complexity and integration capability it might change later. |
| **Sending notification and leaderboard generation** | 24-Aug-22 – 30-Aug-22 | Sockets implementation for notification and leaderboard generation |
| **Testing with different users and implementing** | 31-Aug-22 – 6-Sep-22 | Comparisons with different type of users. Having different data for testing such as user belonging to various colleges and different leaderboard positions. |

|  |  |  |
| --- | --- | --- |
| **Final Product and report** | 7-Sep-22 – 16-Sep-22 | Code refactor and finishing up the report |

##### Risk Plan:

**Unknown JBOSS costs*:*** In order to test the application’s efficiency, we need to deploy in the JBOSS, but having a professional JBOSS EAP might be costly.

**Multi-Environment and user Testing*:*** Web app should be effectively used under mobile and desktop. So, testing across the available browsers and mobile phones by creating different users can be good for product efficiency.

**Top answer issues*:*** There might be different answers for the questions, so displaying only one answer as per feed is mandatory. It can be decided by the highest voted comment/ answer. Apart from this technique there can be many other ways to decide the answer for a question, but the former technique is being is used.

# References

References that I have found so far have already been included in the reading list.