

Department of Informatics University of Leicester CO7201 Individual Project

**Preliminary Report**

**Indexing and Querying Documents using MapReduce**

## Ashwath Vijayan [asv9@student.le.ac.uk](mailto:asv9@student.le.ac.uk) asv9

**Project Supervisor: Dr Thomas Erlebach Second Marker: Dr Nicole Yap**

**Word Count: 1440 Thursday, 4th March, 2021**

### 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: Ashwath Vijayan

#### Date: Thursday, 4th March, 2021

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. Strong database design.
  2. Efficient data fetching for displaying the relevant content.
  3. Providing one comment along with the question as per api is concerned so as to minimize the api traffic.
  4. Managing user sessions for 1hr and managing the sessions.
  5. Updating the content without refreshing.

##### Recommended Requirements:

1. Ability to update questions before the comments.
2. REST based search capabilities.
3. Api to handle the large data for the blogging data.

##### Optional Requirements:

1. There should be an Add comment option at the bottom of every answer.
2. User can share question or answer to different social media sites.
3. Users can get points and badges.

# 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. 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 db design would be very effective in terms of managing and retrieving the data.

##### Testing:

The application can be tested by creating number of different university users, creating different type of questions. This can be heavy as the most of the content is manual typing.

# 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.

##### 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>

# Time-plan and Risk Plan

##### Breakdown of timeline:

|  |  |  |
| --- | --- | --- |
| **Task** | **Planned time taken** | **Explanation** |
| **Requirements and background research** | 25-Feb-21 - 4-Mar-21 | Setting requirements of the application and researching about the existing frameworks |
| **Learning Angular and springboot** | 27-Feb-21 - 5-Mar-21 | Exploring angular and springboot implementaions |
| **Implementing Code for Login, register, ask a question**  **Milestone 1** | 5-Mar-21 - 9-Mar-21 | Code for login, register and ask a question |
| **Tag, answer the question, liking, disliking, commenting the question** | 9-Mar-21 - 16-Mar-21 | Code for tagging and other features for a question |
| **Update question, display in the feed** | 16-Mar-21 - 21-Mar-21 | Displaying the feed |
| **Delete and share the question**  ***(Milestone 2)*** | 21-Mar-21 - 26-Mar-21 | Delete or share the question |
| **Create, liking and dislking the blog** | 27-Mar-21 - 3-Apr-21 | Code for blog handling |
| **Creating questions with time frame *(Milestone 3)*** | 4-Apr-21 - 9-Apr-21 | Code for event handling |
| **Sending notification and leaderboard generation** | 9-Apr-21 - 14-Apr-21 | Sockets implementation for notification and leaderboard generation |
| **Testing with different users and implementing** | 14-Apr-21 - 21-Apr-21 | Comparisons with different type of users |

|  |  |  |
| --- | --- | --- |
| **Final Product and report** | 21-Apr-21 - 30-Apr-21 | 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.