**Feasibility Study**

**asking**

**(A Question and answer platform)**

**17, February 2017**

P.R.L. Fernando

140161F

**Table of Contents**

1. Introduction

1 Overview of the Project

1.2 Objectives of the Project

1.3 The Need for the Project

1.4 Overview of Existing Systems and Technologies

1.5 Scope of the Project

1.6 Deliverables

2. Feasibility Study

2.1 Financial Feasibility

2.2 Technical Feasibility

2.3 Resource and Time Feasibility

2.4 Risk Feasibility

2.5 Social/Legal Feasibility

3. Considerations

4. References

1. Introduction
   1. Overview of the Project

**Question and answer platform for undergraduate students to use as an alternate solution to solve their problems related to academic studies and to exchange their and enable lecturers to get an overall understanding about their teachings by reviewing submitted problems.**

Web based system to submit and review question and offer statistical data representation to get an overall progress of students and teaching. Students can submit question by mentioning the relevant module and relevant topic and fellow students and lecturers can answer the submitted questions. Answers provided by lecturers and students can be used to gain and improve knowledge. Lecturers can use this to get a progress of students in a particular module.

* 1. Objectives of the Project
* Design and implement a platform to submit questions and answers
* Provide students an alternate method to improve their knowledge
* Provide a way to exchange knowledge among students.
* Analyse the questions submitted and generate a statistical report to get an overall progress.
  1. The Need for the Project
* Provide an alternative way to solve academic related problems
* Fellow students can exchange their knowledge
* Lectures can use the system to get the overall student progress
* Learning and teaching process will be much more effective.
* Categorizations of questions and answers make it easy to search and browse.
* Questions categorized according to modules and topics make it easy to search and browse
* Answers are formatted and structured according to modules.
* Well structured interface persuade student to participate
  1. Overview of Existing Systems and Technologies

|  |  |  |
| --- | --- | --- |
| Existing system | Existing system features | Introducing features |
| Moodle [5] | Forum based  Can start a forum thread  Can comment, rate | Centralized view  User friendly interface |
| Stack Exchange [4] | Can add questions  Comment, rate questions  Add, comment, rate answers  Guest user access  Privilege according to points | Categorized view  Notification system is updated  Question and answers are well structured and formatted |

Technologies planning to use

Node.js [1] with Meteor.js [2] for server side implementation

MongoDB [3] as the database management system

Angular.js [6] as the frontend development framework

Bootstrap [7] for the frontend development.

Git [8] as the version controlling system.

* 1. Scope of the Project

System has tree types of users Student, Lecturer, Administration. Students need to register using their admission number before submitting questions. And lecture accounts are created by the administration.

|  |  |  |  |
| --- | --- | --- | --- |
| User | Student | Lecturer | Administrator |
| Functions | * Raise and answer questions. * Rate questions and answers. * Subscribe and unsubscribe for modules. * Search and browse previously submitted questions and answers * Search questions and answers by category * Enable notifications about newly submitted questions. | * Answer questions * Rate questions * Remove questions and answers * View overall progress of teaching | * Add or remove lecturer accounts * Blacklist student account |

* 1. Deliverables.
* A web based software system
* Mobile application (Android)

2. Feasibility Study

* 1. Financial Feasibility

Since the software is targeted undergraduate student and the simplified and categorized interface influence universities to use the system. Hence there are more potential users.

All frameworks, libraries and tools used for development are free, hence no additional expenses are expected for resources during development stage. Only development effort and is accounted, hence the software is affordable to implement.

* 1. Technical Feasibility

The platforms and frameworks used for this project are new, hence extra time is needed to accommodate the learning curve. However, many tutorials and documentations were found that would simplify the learning process.

Meteor [2] is a relatively new full-stack framework for web application development. Official documentation is well organized and there are many discussions in stackoverflow.com which can help in most problems faced during development. Additional time could be allocated to familiarize with these technologies.

2.3 Resource and Time Feasibility

Meteor [2] framework support development on localhost. There is vast amount of libraries available for node.js. A webserver supporting Meteor [2] is required to host ‘asking’ at the end of system development. Currently identified options are Meteor’s own webservers, and Heroku [8]. Both provide free and commercial web hosting; hence resource requirements are feasible.

Since real time synchronization and all the fundamental features required for a real time application are already implemented eases the process of development, hence reducing the time needed for implementation of ‘asking’. Hence, the project can develop within the time constraints.

2.4 Risk Feasibility

Risk of productivity issues – Use a project schedule can help to achieve the target deadline and to manage project. Allocating time initially for researching reusable components for the project can help mitigate this risk further, by reducing required development time.

Risk of losing project data – Using a version controlling system (Git [7]) and committing changes and synchronizing with a remote repository.

Risk of unavoidable situations – Upon unavoidable occurrences such as the deadlines can be changed and extra work need to be done to recover the lost time. A strategic planning with extra time allocated for each task would be feasible.

2.5 Social/Legal Feasibility

At present, the society trends towards searching and using online services instead of traditional means to solve their questions related to academic studies. Hence there is a vast user base for ‘asking’.

‘Asking’ does not violate any social or legal barriers in terms of its core functionality.

Users held the full responsibility for questions and answers submitted to the system. Users who is responsible for unappropriated usage will be banned from system and user will agree to above terms at the registration process. Given the above conditions, ‘asking’ is socially and legally feasible.

1. Considerations

**Performance**

Searching and viewing question and answers should not take more than 3s for processing (Network overhead may affect the time). Submitting questions and answers should be synchronized across all active users viewing the same content in real time to avoid confusions.

**Security**

Users can view question and answers without sign in. Users must register with an email and password to be able to submit questions and answers. Passwords should be stored securely using encryption. Lecture accounts are more privileged than student hence those users can register with the system through the administration.

**Ease of use**

User Interface should be made minimalistic with most recent activities grouped together. Search results should be categorized and organized according to votes

4. References

[1] Node.js Foundation. “Node.js”. Internet: <https://nodejs.org/en/>. [Feb. 8, 2017].

[2] Meteor developers. “Introducing Meteor”. Internet: [http://docs.meteor.com/#/full/](http://docs.meteor.com/%23/full/). [Feb. 8, 2017].

[3] MongoDB, Inc. “The MongoDB 3.4”. Internet: <https://docs.mongodb.com/manual/>. [Feb. 8, 2017].

[4] Stack Exchange Network. “What is Stack Overflow”. Internet: <http://stackoverflow.com/tour>. [Feb. 8, 2017].

[5] Moodle. “Moodle Docs”. Internet: <https://docs.moodle.org/32/en/Features>. [Feb. 8, 2017].

[6] “Angular.js: Developers guide”. Internet: <https://docs.angularjs.org/guide/introduction>. [Feb. 8, 2017].

[7] “About Git”. Internet: <https://git-scm.com/about>. [Feb. 8, 2017].

[8] Salesforce.com. “Cloud Application Platform | Heroku”. Internet: <https://www.heroku.com/home>. [Feb. 8, 2017].