Asking (Question and Answer Platform)

Software Requirements Specification

For the system

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 13/03/2017 | 1.0 | Initial software requirement specification | P.R.L. Fernando |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 5

1.1 Purpose 5

1.2 Scope 5

1.3 Definitions, Acronyms, and Abbreviations 5

1.4 References 6

1.5 Overview 6

2. Overall Description 6

2.1 Product perspective 6

2.2 Product functions 6

2.3 User characteristics 7

2.4 Constraints 7

2.5 Assumptions and dependencies 7

3. Specific Requirements 8

3.1 Functionality 8

3.1.1 Register with the system 8

3.1.2 Subscribe for modules 8

3.1.3 Unsubscribe modules 9

3.1.4 Search questions 9

3.1.5 View answers of a question 9

3.1.6 Submit questions to the system 10

3.1.7 Submit answers to questions 10

3.1.8 Rate submitted questions 10

3.1.9 Rate submitted answers 11

3.1.10 Submit answers for the questions 11

3.1.11 Rate submitted questions 11

3.1.12 Rate submitted answers 12

3.1.13 View report of questions submitted 12

3.1.14 Modify answers 12

3.1.15 Remove answers 13

3.1.16 Remove questions 13

3.1.17 Create lecturer accounts 13

3.1.18 Remove lecturer accounts 14

3.1.19 Blacklist student accounts 14

3.2 Usability 15

3.3 Reliability 15

3.4 Performance and Security 15

3.5 Supportability 16

3.6 Design Constraints 16

3.7 On-line User Documentation and Help System Requirements 16

3.8 Interfaces 17

3.8.1 User Interfaces 17

3.8.2 Hardware Interfaces 23

3.8.3 Software Interfaces 23

3.8.4 Communications Interfaces 24

3.9 Database Requirements 24

3.10 Licensing, Legal, Copyright, and Other Notices 24

3.11 Applicable Standards 24

4. Supporting Information 25

Software Requirements Specification

# Introduction

This section gives a scope description and overview of everything included in this SRS document. Also, the purpose for this document is described and a list of abbreviations and definitions is provided.

## Purpose

The purpose of this document is to give a detailed description of the requirements for the “Asking Question and Answer Platform” software. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system.

## Scope

The Asking is a Question and Answer Platform which specialize to fulfill the need of platform for undergraduate students submit their questions related to academic studies and exchange their knowledge among class mates. Also application allow lecturers to submit answers and to get an overall understanding of students by reviewing the questions submitted. The application should be available as a web service and students need to create an account in order to access the service. Lecturer accounts are created and managed through the administration.

Students can submit their questions through the platform by mentioning the module, title and appropriate tags which describe the question. Students can subscribe for modules in order to get notification about newly submitted questions. Lecturer accounts are assigned to modules by the administration. Students and lecturers can submit answers for questions. Rating system is used to prioritize questions and answers.

Further more lecturers can use this system to view a statistical view of questions submitted categorized according to modules and topics. This can use to get an overall understanding about their teaching process and to enhance it.

## Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| User | Someone who interacts with the mobile phone application |
| Administrator | System administrator who is given specific permission for managing and controlling the system |
| Student | An undergraduate student |
| Lecturer | A lecturer who teaches undergraduate students |
| Stakeholder | Any person who has interaction with the system who is not a developer. |
| CS-1,2,3…… | Development constraint 1,2,3….. |
| AS-1,2,3…… | Assumption 1,2,3….. |
| USE.R-1,2,3….. | Usability requirement 1,2,3….. |
| REL.R-1,2,3….. | Reliability requirement 1,2,3….. |
| PRF.R-1,2,3….. | Performance requirement 1,2,3….. |
| SUP-1,2,3….. | Supportability requirement 1,2,3….. |
| DES.C-1,2,3…... | Design constraint 1,2,3….. |
| SIR-1,2,3….. | Software interface requirement 1,2,3…… |
| CIR-1,2,3…… | Communication interface requirement 1,2,3….. |

## References

|  |  |
| --- | --- |
| [1] | IEEE Software Engineering Standards Committee, "IEEE Std 830-1998, IEEE Recommended," 20 October 1998. |

## Overview

The remainder of this document includes three chapters and appendixes. The second one provides an overview of the system functionality and system interaction with other systems. This chapter also introduces different types of stakeholders and their interaction with the system. Further, the chapter also mentions the system constraints and assumptions about the product.

The third chapter provides the requirements specification in detailed terms and a description of the different system interfaces. Different specification techniques are used in order to specify the requirements more precisely for different audiences.

The fourth chapter deals with the prioritization of the requirements. It includes a motivation for the chosen prioritization methods and discusses why other alternatives were not chosen.

The Appendixes in the end of the document include the all results of the requirement prioritization and a release plan based on them

# Overall Description

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will also describe what type of stakeholders that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

## Product perspective

This system consists of web platform. The system will be used to submit questions and answers related to the academic studies of undergraduate students and will be used as a platform for lecturers to get an overall understanding of the student performance.

Since this is data centric if will need somewhere to store the data. For that, a database will be used. The client side will communicate with the web server and database, Client application uses the data base to retrieve, add and to modify data.

## Product functions

* Administration can create lecturer accounts
* Administration can remove lecturer accounts
* Administrator can blacklist student accounts
* Administrator can add modules
* Administrator can modify modules
* User can login to the system
* Student can register to the system
* Student can subscribe for modules
* Student can search questions
* Student can view questions and answers
* Student can submit questions
* Student can submit answers
* Student can rate questions
* Student can rate answers
* Lecturer can search questions
* Lecturer can view questions and answers
* Lecturer can submit answers
* Lecturer can rate questions
* Lecturer can modify answers
* Lecturer can delete questions and answers
* Lecturer can view a statistical report of answers submitted

## User characteristics

There are three types of users. There is student, lecturer and administrator. Each of them have their own requirements. Student must register with the system to submit question and answers whereas lecturer accounts are created by the administrator.

Student can only submit questions, submit answers and rate them. The student has to be able to search and browse questions, choose questions from that search and navigate to it. The student has to be able to rate questions and answers.

Lecturers will be also able to submit answers and rate questions and answers. In additions to these requirements lecturer will be able to has the ability to modify answers, remove answers, remove questions and to view a report of submitted question categorized by the module code.

The administrator managing the overall system so there is no incorrect information and inappropriate information within it. And also administrator will be able to has the ability to manage the lecturer accounts and student accounts.

## Constraints

**CS1:** Only a registered student can view, submit, rate the questions and answers.

**CS2:** Only the administrator has access to create and manage lecturer accounts.

**CS3:** Subscribe system is filtering notifications of newly submitted questions.

**CS4:** Only under graduate students can register in the system since the system is specialized for

undergraduates.

## Assumptions and dependencies

**AS-1:** 24/7 availability and operation is required from the system

**AS-2:** End user will have proper understanding of the purpose and functionality of Question and answer

platform

# Specific Requirements

This sections of the SRS contains all software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test the system satisfies those requirements.

## Functionality

This sections describes the functional requirements of the system of different user types in detail. Functional requirements are categorized with respect to the user.

**Student functional requirements**

### Register with the system

|  |  |
| --- | --- |
| Summary | The students can sign themselves up for the Asking with their email address |
| Basic Course of events | 1. View the login page 2. Click on sign up button 3. Enter email address as the user name 4. Provide password 5. Verify username and the password 6. Create new account for the student in Asking |
| Alternate paths | (None) |
| Exception paths | 5.a. If the email address is not valid or password do not match show an error message  5.b. Join on 3  5.a. If user already has existing account, inform to sign in instead.  5.b. Join on 1 |
| Extension points | (None) |
| Trigger(s) | Click on sign up button |
| Assumptions | (None) |
| Pre-conditions | Student has a valid email address |
| Post-conditions | Show the sign up confirmation page |

### Subscribe for modules

|  |  |
| --- | --- |
| Summary | The students can subscribe for modules in order to get notifications |
| Basic Course of events | 1. View the Add module page 2. Search for module 3. Navigate to module 4. Click on subscribe button 5. Subscribe for module |
| Alternate paths | (None) |
| Exception paths | 3.a. If the search result is empty  3.b. Join on 2  5.a. If the module is already subscribed  5.b. Join on 2 |
| Extension points | (None) |
| Trigger(s) | Choose Add module from homepage side bar |
| Assumptions | Modules must be available to subscribe |
| Pre-conditions | Student must have logged in  Student account is not blacklisted |
| Post-conditions | Show the subscription successful message. |

### Unsubscribe modules

|  |  |
| --- | --- |
| Summary | The students can unsubscribe modules in order to stop notifications |
| Basic Course of events | 1. View enrolled modules page 2. Select module 3. Navigate to module 4. Click on unsubscribe button 5. Unsubscribe module |
| Alternate paths | (None) |
| Exception paths | (None) |
| Extension points | (None) |
| Trigger(s) | Choose enrolled modules from homepage side bar |
| Assumptions | There is subscribed modules available |
| Pre-conditions | Student must logged in |
| Post-conditions | Show subscription successful message |

### Search questions

|  |  |
| --- | --- |
| Summary | The students can search previously submitted questions and answers |
| Basic Course of events | 1. View homepage 2. Enter keyword in the search bar 3. Click on the search button |
| Alternate paths | (None) |
| Exception paths | 3.a. If the search result is empty  3.b. Join on 2 |
| Extension points | 3.a. If single question is selected, show its answers (UC: 3.1.5) |
| Trigger(s) | Choose Add module from homepage side bar |
| Assumptions | (None) |
| Pre-conditions | User must in the page where the search bar is available  Student account is not blacklisted |
| Post-conditions | Show the search results with question ratings |

### View answers of a question

|  |  |
| --- | --- |
| Summary | Once student searches and finds a matching question, he can click on it and see submitted answers. |
| Basic Course of events | 1. Select (Click on) specific question from the search result 2. System request submitted answers for the selected questions 3. System shows question summary and answers to the user |
| Alternate paths | (None) |
| Exception paths | (None) |
| Extension points | 3.a. Student can add answers to the questions |
| Trigger(s) | Select specific question from the search result |
| Assumptions | (None) |
| Pre-conditions | Search result must be available |
| Post-conditions | Show answers with question summary |

### Submit questions to the system

|  |  |
| --- | --- |
| Summary | The students can submit questions |
| Basic Course of events | 1. View the submit question page 2. Enter question title and description 3. Select question module and question tags 4. Click submit button 5. View submitted question |
| Alternate paths | (None) |
| Exception paths | 4.a. If the submission is not successful, notify the user of the situation, and ask to retry.  4.b. Join on 3 |
| Extension points | (None) |
| Trigger(s) | Student account is not blacklisted  Choose Add question from homepage side bar |
| Assumptions | (None) |
| Pre-conditions | Student must have logged in  Student account is not blacklisted |
| Post-conditions | Show the submitted question |

### Submit answers to questions

|  |  |
| --- | --- |
| Summary | The students can submit answers for the question |
| Basic Course of events | 1. Select (Click on) question 2. Click add new answer button 3. Add answer inside the text field 4. Click on submit button 5. Show updated question with answers |
| Alternate paths | (None) |
| Exception paths | 4.a. If the submission is not successful, notify the user of the situation, and ask to retry.  4.b. Join on 4 |
| Extension points | (None) |
| Trigger(s) | Choose question from main page notifications or from the search result |
| Assumptions | There must be questions available |
| Pre-conditions | Student must be logged in  Student account is not blacklisted |
| Post-conditions | Show the updated question with answers |

### Rate submitted questions

|  |  |
| --- | --- |
| Summary | The students can rate questions to change its priority |
| Basic Course of events | 1. Choose (Click on) question 2. Click on rate up or rate down button |
| Alternate paths | (None) |
| Exception paths | (None) |
| Extension points | 1.a. Rate answers (UC 3.1.9) |
| Trigger(s) | Choose question from main page notifications or from the search result |
| Assumptions | There must be questions available |
| Pre-conditions | Student account is not blacklisted |
| Post-conditions | Show the question list with updated order |

### Rate submitted answers

|  |  |
| --- | --- |
| Summary | The students can rate answers to change its priority |
| Basic Course of events | 1. Choose (Click on) question 2. Click on rate up or rate down button |
| Alternate paths | (None) |
| Exception paths | (None) |
| Extension points | (None) |
| Trigger(s) | Choose answer from the submitted answers |
| Assumptions | There must be answers available |
| Pre-conditions | Student account is not blacklisted |
| Post-conditions | Show the answer list with updated order |

**Lecturer functional requirements**

### Submit answers for the questions

|  |  |
| --- | --- |
| Summary | The lecturer can submit answers for the question |
| Basic Course of events | 1. Select (Click on) question 2. Click add new answer button 3. Add answer inside the text field 4. Click on submit button 5. Show updated question with answers |
| Alternate paths | (None) |
| Exception paths | 4.a. If the submission is not successful, notify the user of the situation, and ask to retry.  4.b. Join on 4 |
| Extension points | (None) |
| Trigger(s) | Choose question from main page notifications or from the search result |
| Assumptions | There must be questions available |
| Pre-conditions | Student must be logged in  Student account is not blacklisted |
| Post-conditions | Show the updated question with answers |

### Rate submitted questions

|  |  |
| --- | --- |
| Summary | The lecturer can rate questions to change its priority |
| Basic Course of events | 1. Choose (Click on) question 2. Click on rate up or rate down button |
| Alternate paths | (None) |
| Exception paths | (None) |
| Extension points | 1.a. Rate answers (UC 3.1.9) |
| Trigger(s) | Choose question from main page notifications or from the search result |
| Assumptions | There must be questions available |
| Pre-conditions | Lecturer must logged in |
| Post-conditions | Show the subscription successful message. |

### Rate submitted answers

|  |  |
| --- | --- |
| Summary | The lecturer can rate answers to change its priority |
| Basic Course of events | 1. Choose (Click on) question 2. Click on rate up or rate down button |
| Alternate paths | (None) |
| Exception paths | (None) |
| Extension points | (None) |
| Trigger(s) | Choose answer from the submitted answers |
| Assumptions | There must be answers available |
| Pre-conditions | Lecturer must logged in |
| Post-conditions | Show the subscription successful message. |

### View report of questions submitted

|  |  |
| --- | --- |
| Summary | The lecturer can view a report of submitted question with percentages to get an overall understanding |
| Basic Course of events | 1. View the Overall report page 2. Select module 3. Navigate to module 4. View report |
| Alternate paths | (None) |
| Exception paths | (None) |
| Extension points | 3.a. Search questions and answers (UC: 3.1.5) |
| Trigger(s) | Choose view report from the homepage side bar |
| Assumptions | There are question available |
| Pre-conditions | Lecturer must logged in |
| Post-conditions | Show the overall report of submitted question categorized by modules |

### Modify answers

|  |  |
| --- | --- |
| Summary | The lecturer can modify answers to enhance it |
| Basic Course of events | 1. Choose (Click on) question 2. Click on change answer button 3. Update answer 4. Click on submit button |
| Alternate paths | (None) |
| Exception paths | (None) |
| Extension points | (None) |
| Trigger(s) | Choose answer from the submitted answers |
| Assumptions | There must be answers available |
| Pre-conditions | Lecture must logged in |
| Post-conditions | Show the updated answer |

### Remove answers

|  |  |
| --- | --- |
| Summary | The lecturer can remove unnecessary answers |
| Basic Course of events | 1. Choose (Click on) answer 2. Click on remove answer button 3. Click on submit button |
| Alternate paths | (None) |
| Exception paths | (None) |
| Extension points | (None) |
| Trigger(s) | Choose answer from the submitted answers |
| Assumptions | There must be unnecessary answers available |
| Pre-conditions | Lecture must logged in |
| Post-conditions | Show the updated question with answers |

### Remove questions

|  |  |
| --- | --- |
| Summary | The lecturer can remove unnecessary questions |
| Basic Course of events | 1. Choose (Click on) question 2. Click on remove question button 3. Click on submit button |
| Alternate paths | (None) |
| Exception paths | (None) |
| Extension points | (None) |
| Trigger(s) | Choose question from main page notifications or from the search result |
| Assumptions | There must be unnecessary questions available |
| Pre-conditions | Lecture must logged in |
| Post-conditions | Show the updated question list |

**Administrator functional requirements**

### Create lecturer accounts

|  |  |
| --- | --- |
| Summary | The administrator can create lecturer accounts. |
| Basic Course of events | 1. View account management page 2. Click create new lecturer account 3. Enter lecture id and relevant details 4. Assign modules 5. Click on create new account button |
| Alternate paths | (None) |
| Exception paths | 5.a. If there is error in account creation, Notify user and  5.b. Join on 3 |
| Extension points | (None) |
| Trigger(s) | Choose account management from main menu side bar |
| Assumptions | Lecture has an id |
| Pre-conditions | Administrator must logged in |
| Post-conditions | Show the account successfully created message |

### Remove lecturer accounts

|  |  |
| --- | --- |
| Summary | The administrator can remove lecturer accounts. |
| Basic Course of events | 1. View account management page 2. Click lecturer accounts drop down 3. Select lecturer account 4. Click on delete account button 5. Click on submit button |
| Alternate paths | (None) |
| Exception paths | 5.a. If there is error in account deletion, Notify user and  5.b. Join on 4 |
| Extension points | (None) |
| Trigger(s) | Choose account management from main menu side bar |
| Assumptions | Lecturer accounts are available |
| Pre-conditions | Administrator must logged in |
| Post-conditions | Show the account successfully deleted message |

### Blacklist student accounts

|  |  |
| --- | --- |
| Summary | The administrator can remove lecturer accounts. |
| Basic Course of events | 1. View account management page 2. Click student accounts drop down 3. Select student account 4. Click on blacklist account button 5. Click on submit button |
| Alternate paths | (None) |
| Exception paths | 5.a. If there is error in account modification, Notify user and  5.b. Join on 4 |
| Extension points | (None) |
| Trigger(s) | Choose account management from main menu side bar |
| Assumptions | Student accounts are available |
| Pre-conditions | Administrator must logged in |
| Post-conditions | Show the account successfully blacklisted message |

## Usability

This sections includes all those requirements that affect usability of the system.

**USE.R-1:** User learning

A normal user should be able to use the system at a productive rate after a maximum time of 1 minutes. A power user should be capable of using the system at a fully productive rate after 5minutes or less time of learning.

**USE.R-2:** System GUIs

The user interface elements that will be used for the system implementation should be compatible with look and feel requirements in order to ensure that the user shall be able to easily understand what they mean and the user tends to use them without refusing due to other reasons.

## Reliability

This sections includes all those requirements that affect reliability of the system.

**REL.R-1:** Availability

95% of the time, system should be available. (Expected down time of the system 18x24 + 6 hours per year

**REL.R-1:** Mean time between failures

Mean time of failures should be 24 hours.

**REL.R-1:** Mean time to repair

System should not be allowed more than 24 hours to be out of operation after if has failed.

**REL.R-1:** Accuracy

More than 98% of the times, the system gives the correct results on a search.

More than 98% of answers are according to the question.

## Performance and Security

**PRF.R-1:** Response time

95% of the search results shall be responded in less than 5 seconds.

**PRF.R-1:** Capacity

Support at least 1000 concurrent access at any given instance.

**PRF.R-1:** Degradation

Shall handle expected and unexpected errors to prevent loss of information.

## Supportability

This section indicates requirements that will enhance the supportability or maintainability of the system being built, including coding standards, naming conventions, class libraries, maintenance access, and maintenance utilities.

**SUP-1:** Coding standards

Entire coding should be done in a standard way so that it helps extendibility and maintainability. Always comments should be used to describe what the code does.

**SUP-1:** Naming conventions

Naming conventions should be done in a meaningful way where the developers can easily understand the variables and methods which will improve the maintainability.

**SUP-1:** Class libraries

Before using any class library all the limitations should be understood so that after sometime of using the library it won’t affect the systems functionality.

**SUP-1:** Application testability

Test environment should test all the functions of the system in order to make sure the requirements are correctly implemented.

## Design Constraints

This section indicated any design constraints on the system being built.

**DES.C-1:** Standards development tools

Web site should be built in a way that desktop computers, tablets and phones must be able to handle the GUIs very conveniently.

**DES.C-1:** Cloud database

Since the database cannot be saved inside a phone there must be a server to connect to the database. To perform this cloud architecture is used. In this case user, should first switch on data to connect to the server and not very applicable for time critical. Hence cloud computing is partially use for this system with slight modifications which will preserve the time critical functionality.

## On-line User Documentation and Help System Requirements

After registering to the system there should be a user guide about the functionalities of the components and where to use and what will be the output. This is essential to make the system user friendly and reduce the learning curve.

## Interfaces

This section defines the interfaces that must be supported by the application. Including user interfaces, software interfaces and communication interfaces.

### User Interfaces

This section describes the logical characteristics of each interface between the software product and the users. This includes sample screen images, screen layout constraints, standard buttons and functions that will appear screens

#### Interface I: Sign up page

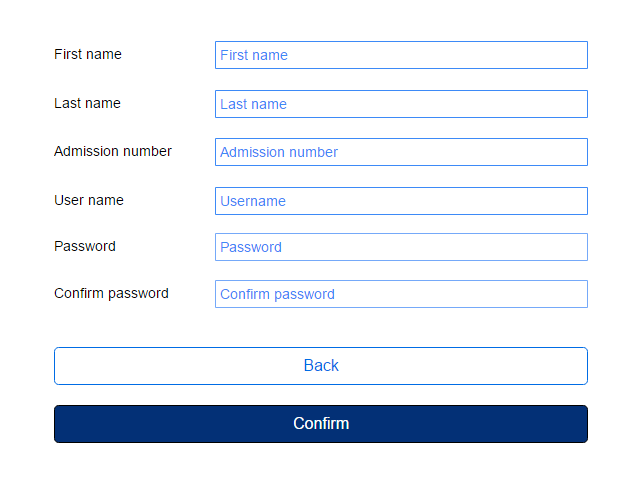


Figure 3- 1 Sign up page

Students can sign up in the system to get the full functionality of the system including submitting, rating the questions and answers. Student can visit the sign up page by selecting the sign up button in the sign in page (Figure 3-2 Sign in page). This directs the user to “Sign Up” page.

When signing up, the user is required to enter the valid email address and password. After validating entered information. If the student successfully signed up, he/she should see the login page (Figure 3-1 Sign in page). If the entered data is invalid, an error message will be shown to the user in the space above the Confirm button

#### Interface II: Sign in page

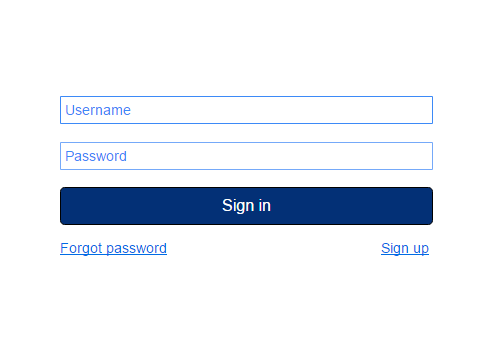


Figure 3- 2 Sign in page

A user will redirect to the sign in page when accessing the web. If the user is not registered, the user has to select on Sign Up button to create a new account. Then the user will be redirected to Sign Up (Figure 3-1 Sign Up page) for registration. Otherwise, the user can enter his or her username and password, and click on Sign in to go to the Home page.

If the user is a student, then the user will go to the student home page (Figure 3-3 Student home page) or if the user is a lecturer, then the user will go to the lecturer home page (Figure 3-4 Lecturer home page).

If the user has forgotten the password, the Forgot Password button should be clicked. Then a message will be sent to the relevant person regarding the situation and the user will be sent a recovery password which can be used to sign in.

#### Interface III: Student home page

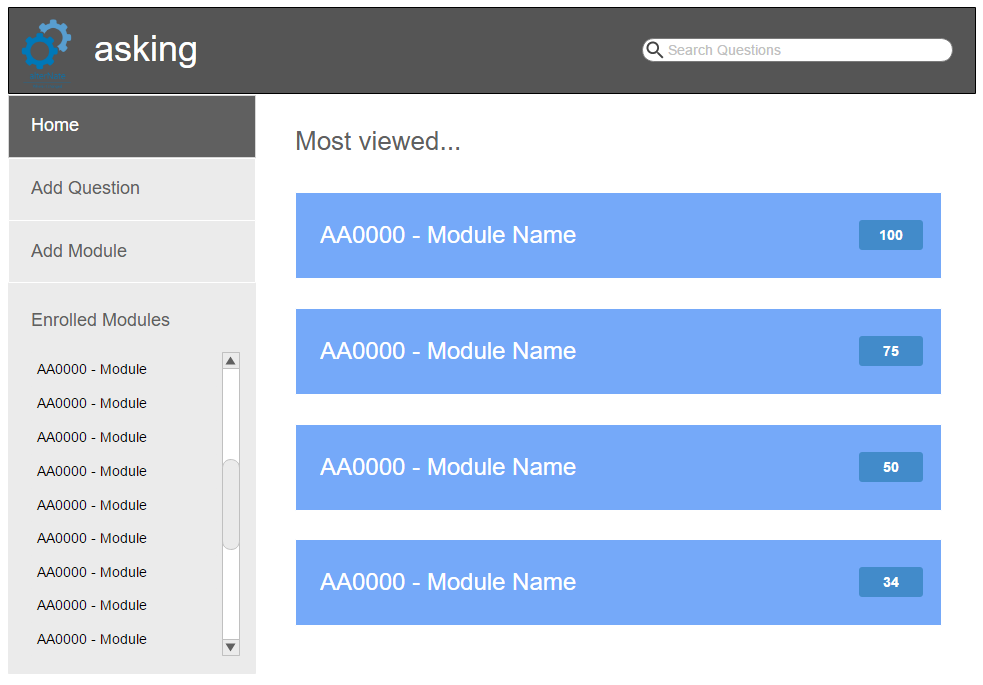


Figure 3- 3 Student home page

Once the student logged on the student is presented with the student home page. It shows the modules with recently submitted answers in a descending order. In order to search question student, type a keyword in the search bar. By clicking on the featured module student will go to the questions list page of that module (Figure 3-4 Questions list page).

Student can add new question by selecting the Add new question then the student will go to the add new question page (Figure 3-5 Add new question page).

Student can subscribe for new module by selecting the Add new module then the student will go to the Subscribe new module page (Figure 3-6 Subscribe new module page).

Student can navigate to subscribed modules to view the questions submitted on that module. By selecting the module student will go to the questions list page of that module (Figure 3-4 Questions list page).

#### Question list (Search result) page

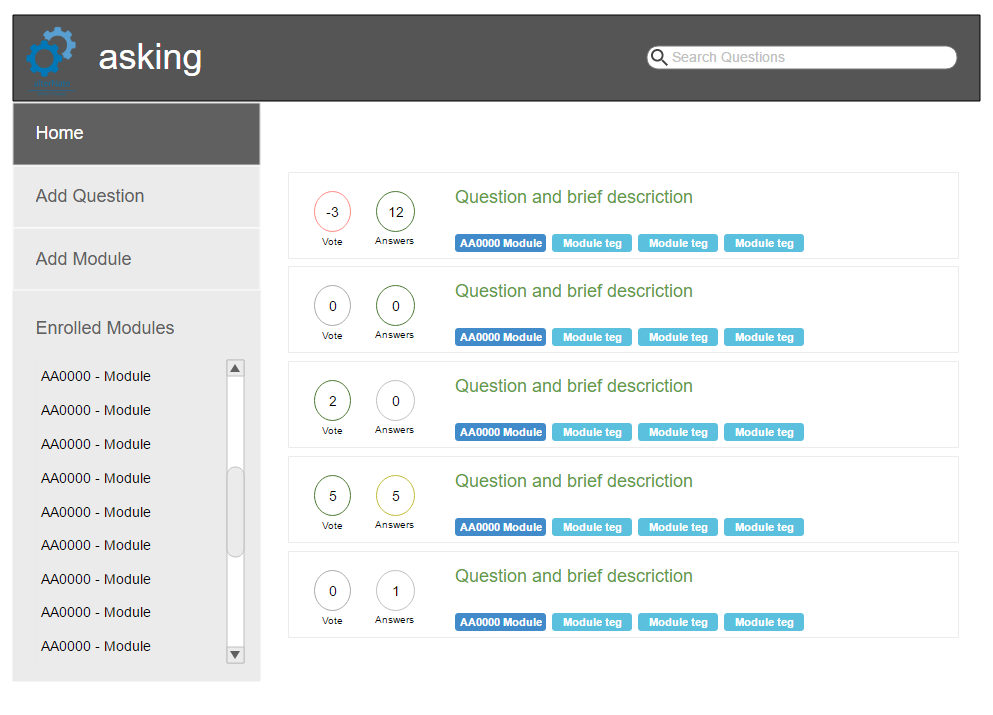


Figure 3- 4 Question list page

Once the search operation completed or the student navigated to a module a questions list will display regarding the search criteria. Student can select any question to view associated answers with the question description (Figure 3-5 Question description with answers page)

Overall ratings and total submitted answers for any given question will display in front of it to enhance the user experience (See Figure 3-4 Question list page)

#### Add new question page

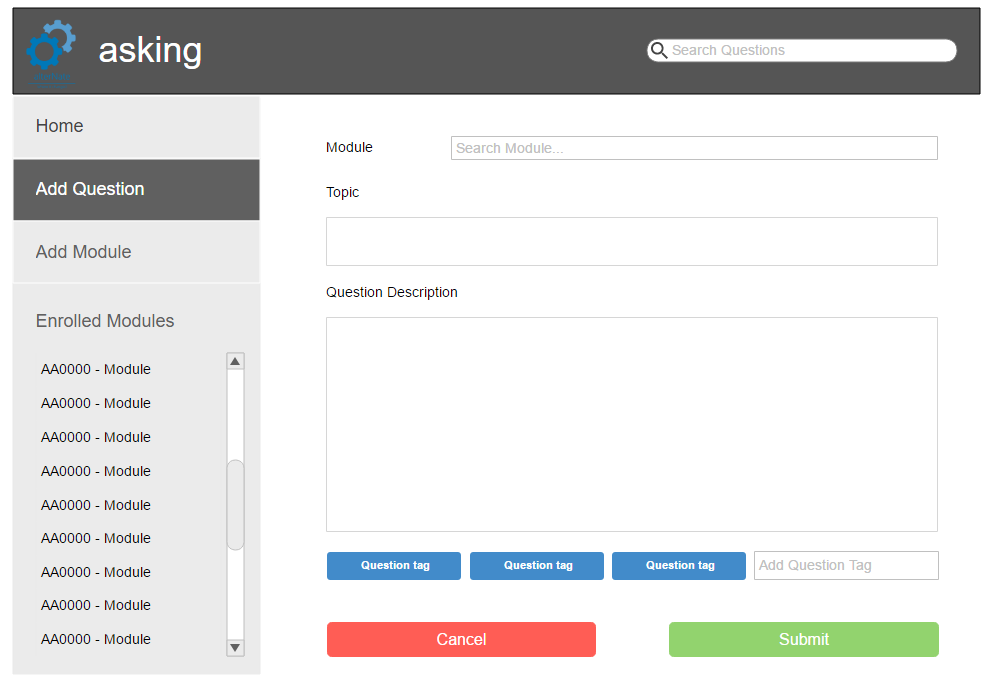


Figure 3- 5 Add new question page

Once the student selects add new question from the side bar then the student will navigate to add new question page.

Student can search for a module with the question is associated with. If the student is not enrolled for that module, he will notify that and ask for enrollment in order to get the notifications. Student can add question tags to specify the module titles which the question is associated with. Then the student can submit the question by click on the submit button.

#### Subscribe new module page

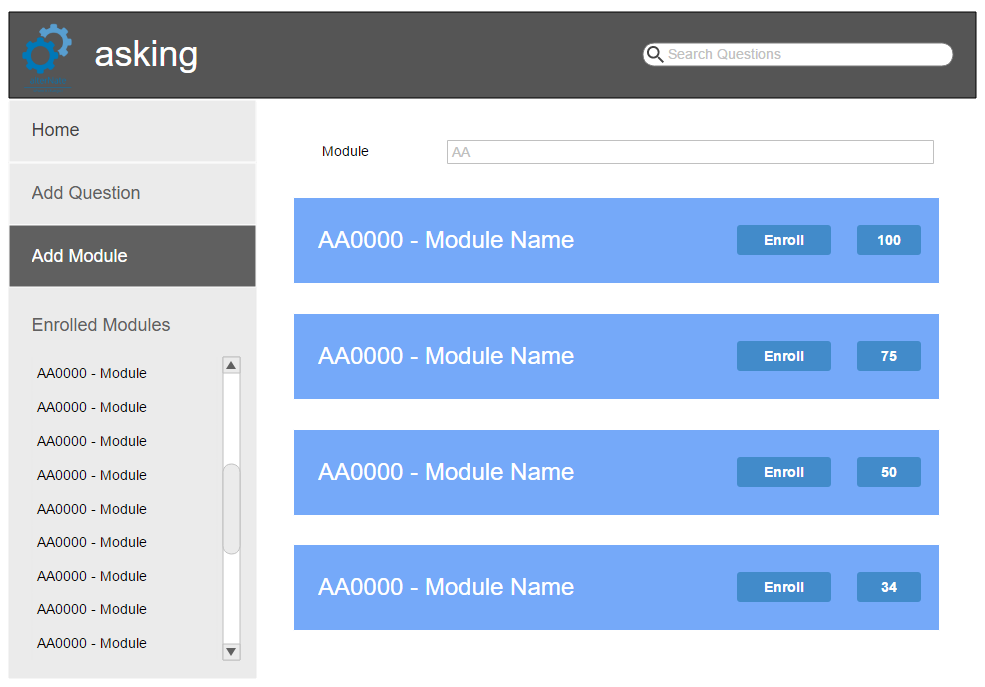


Figure 3- 6 Subscribe new module page

Once the student selects add new module from the side bar then the student will navigate to subscribe new module page. Student can search for a module and then the search result is displayed. Student can enroll for modules by clicking on the enroll button associated with each module.

Student can select any module to view the questions submitted within it then the student will go to Questions list page (Figure 3-4 Question list page)

### Hardware Interfaces

Since the web site not have any designated hardware, it does not have any direct hardware interfaces. The database server is managed by the underlying operating system on the web server.

### Software Interfaces

This section describes the logical characteristics of each interface between the software components. This includes input message and output message and format of the message.

**Between client application and server**

**SIR-1:** Login user

Input message: Parameter (Username, Password)

Output message: User type, User access level, Name, Index number

Format: JSON object array type

**SIR-2:** Register new user

Input message: Parameter (Email, Password)

Output message: New account added

Format: ASCII

**SIR-3:** Search a question

Input message: Search parameters (Keyword, Tags, Module code)

Output message: Question array

Format: JSON object array type

**SIR-4:** View question with answers

Input message: Search parameters (Question id)

Output message: Answers array

Format: JSON object array type

**SIR-5:** Submit a question

Input message: Parameters (Question title, Question description, Tags, Module code)

Output message: New question added message

Format: ASCII

**SIR-6:** Submit an answer

Input message: Parameters (Question id, Answer)

Output message: New answer added message

Format: ASCII

**SIR-7:** Rate a question

Input message: Parameters (Question id, Rate)

Output message: Rating increased/ decreased

Format: ASCII

**SIR-8:** Rate an answer

Input message: Parameters (Question id, Answer id, Rate)

Output message: Rating increased/ decreased

Format: ASCII

**SIR-9:** Modify a question

Input message: Parameters (Question id, Modified question)

Output message: Question modified

Format: ASCII

**SIR-10:** Modify an answer

Input message: Parameters (Question id, Answer id, Modified answer)

Output message: Answer modified

Format: ASCII

**SIR-11:** Remove a question

Input message: Parameters (Question id)

Output message: Question removed

Format: ASCII

**SIR-12:** Remove an answer

Input message: Parameters (Question id, Answer id)

Output message: Answer removed

Format: ASCII

**SIR-13:** Remove a user account

Input message: Parameters (Username, reason)

Output message: Account removed

Format: ASCII

### Communications Interfaces

This section describes the logical characteristics of each interface that use to communicate.

**CIR-1:** Reset password using email

When the user clicks on forgot password link then a link to reset the password is sent to the email.

## Database Requirements

Questions and answers are frequently accessing information in the system. The database should be designed in a way that the accessing data from the database does not hinder the performance of the application. And the increasing of questions and answers should not hinder the performance of the application.

## Licensing, Legal, Copyright, and Other Notices

Developers must aware of legal disclaimers, copyright notices, trademark, or logos that are bundled with third party libraries and packages.

## Applicable Standards

(None)

# Supporting Information

Appendix A: Table of figures

[Figure 3- 1 Sign up page 17](#_Toc477374380)

[Figure 3- 2 Sign in page 18](#_Toc477374381)

[Figure 3- 3 Student home page 19](#_Toc477374382)

[Figure 3- 4 Question list page 20](#_Toc477374383)

[Figure 3- 5 Add new question page 21](#_Toc477374384)

[Figure 3- 6 Subscribe new module page 22](#_Toc477374385)