**FQA System**

**Development Report**

CSCI822

Group 2

Xinchi Wang

Rixin Nie

Kaijian Feng

Caixing Su

Akshay Lakhanpal

Table of Contents

System Request – FQA System 4

Software Development Plan 5

1. Introduction 5

1.1 Purpose 5

1.2 Scope 5

1.3 Definitions, Acronyms and Abbreviations 5

1.4 References 5

1.5 Overview 5

2. Project Overview 5

2.1 Project Purpose, Scope, and Objectives 5

2.2 Assumptions and Constraints 5

2.3 Project Deliverables 5

3. Project Organization 6

3.1 Organizational Structure 6

3.2 Roles and Responsibilities 6

4. Management Process 7

4.1 Project Plan 7

4.2 Iteration Plans 9

4.3 Project Monitoring and Control 9

Risk List 11

1. Introduction 11

1.1 Purpose 11

1.2 Scope 11

1.3 Definitions, Acronyms and Abbreviations 11

1.4 References 11

1.5 Overview 11

2. Risks 11

2.1 Project Risk: Hard Deadline 11

2.2 Project Risk: Team Roster Change 11

2.3 Project Risk: Manager’s non-presence 12

Use Case Diagram and Specification 13

Software Requirements Specification 21

1. Introduction 21

1.1 Purpose 21

1.2 Scope 21

1.3 Definitions, Acronyms and Abbreviations 21

1.4 References 21

1.5 Overview 21

2. Overall Description 21

2.1 Product Perspective 21

2.2 Product Functions 23

2.3 User Characteristics 23

2.4 Constraints 23

2.5 Assumptions and Dependencies 23

3. Specific Requirements 24

3.1 Functional Requirements 24

3.2 Non-Functional Requirements 34

Domain Model 36

Data Dictionary 37

Group Members 45

Group Meeting Reports 46

Working Diaries 51

FQA System Development Contribution Assessment 56

Version Controlling Records 57

System Request – FQA System

**Project sponsor:** University of Wollongong

**Business Need:** This project has been started to design a simple and efficient question and answer system which aim is to provide a platform for people to share their questions and answers with other in the internet.

**Business Requirements:**

Some people might meet many questions in daily life. They should need a platform to ask their question or help other people to solve their question. They also should need to share the experience of solving question with some friends or strangers. Basically, the FQA system should have these functions which are listed below:

* User can register as a member and login in to manage their account
* User can ask a question and submit it to the server
* User can answer some question
* User can vote the answer which they think it is useful
* Questioners decide which answer is the best answers
* Question and answer can be commented to help the questioner get a better solution
* Question and answer can be read by anyone

**Business Value:**

In fact, because the main purpose of this system is to design a system for sharing knowledge and experience and sometime the knowledge and the experience are invaluable and inestimable, so the business value of this question and answer system is very difficult and even impossible to estimate in the request analysis period.

**Stakeholder and Customer Descriptions:**

As our system is a question and answer system, so anyone could be the stakeholder and customer when they have a question or try to answer some questions. In particular, those people who are closely related to the question and answer are the most likely stakeholder such as teacher and student because teacher need to answer many questions which come from student and the student may meet some questions when they are studying new knowledge.

**Special Issues or Constraints:**

* The Marketing Department views this as a strategic system. This Internet system will add value to our current business model, and it also will serve as a proof of concepts for future Internet endeavors. For example, in the future, CD Selections may want to sell products directly over the Internet.
* The system should be in place for the holiday shopping season next year.

# 

# Introduction

## Purpose

The purpose of this document is to describe the plan of activities in development of FQA System. It also includes project overview and organization structure.

The intended audience of this document is the prospective software development team.

## Scope

This plan is only associated with FQA System, an online Q/A system using browser/server structure.

## Definitions, Acronyms and Abbreviations

FQA: Find Question and answer

## References

Rational Unified Process, Development Plan template (rup\_sdpln.dot)

FQA System Risk Management Plan

## Overview

The rest of this document is divided into three main sections:

* Project Overview is a brief introduction of the project.
* Project Organization describes the organization structure of the team and roles of each team member.
* Management Process describes the plans of development.

# Project Overview

## Project Purpose, Scope, and Objectives

This project is to build an online Q&A system in which users can ask and answer questions. The system should be a website and be deployed on Internet.

## Assumptions and Constraints

* The project must be finished by Week 13.
* We must use C++ or Java for implementation.

## Project Deliverables

The following deliverables will be produced during the project:

* Software Development Plan (this document)
* Risk List
* System Request
* Software Requirement Specification
* Use Case Specifications
* Use Case Realizations
* Domain Model
* Software Architecture Document
* Database Structure Document
* Build

# Project Organization

## Organizational Structure

|  |  |
| --- | --- |
| **Role** | **Names** |
| Project Manager | Xinchi Wang |
| Business Designer | Caixing Su |
| System Analyst | Caixing Su, Xinchi Wang |
| Designer | Rixin Nie, Kaijian Feng |
| Requirement Specifier | Caixing Su, Xinchi Wang, Akshay Lakhanpal |
| Implementer | Rixin Nie, Kaijian Feng |
| Test Designer | Kaijian Feng |
| Tester | Kaijian Feng, Akshay Lakhanpal |
| Tool Specialist | Kaijian Feng |
| Deployment Manager | Kaijian Feng |

## Roles and Responsibilities

Team members have volunteered for the following roles as defined by the Rational Unified Process.

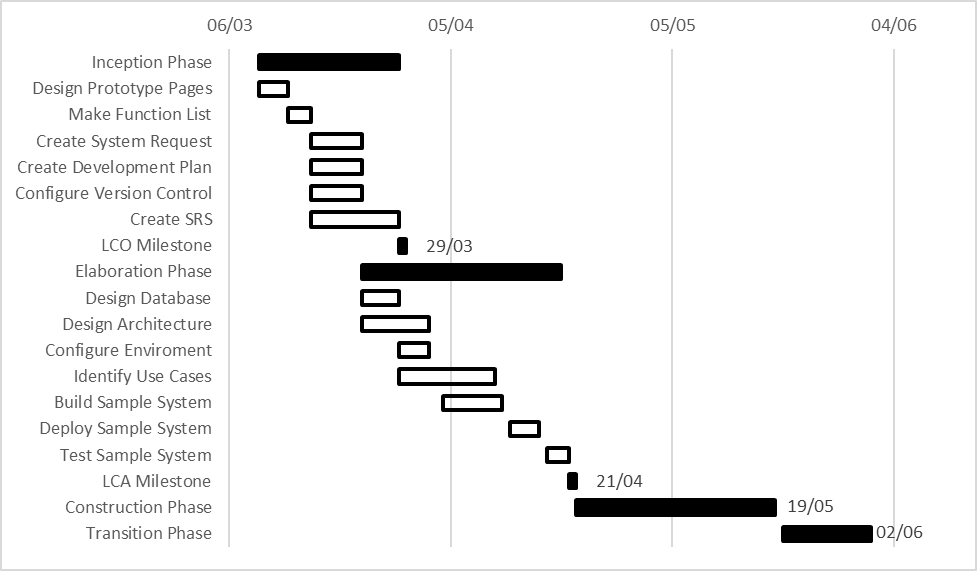
| **Role** | **Description** |
| --- | --- |
| Project Manager | Allocates resources, shapes priorities, coordinates interactions with the customers and users and generally tries to keep the project team focused on the right goal. The project manager establishes a set of practices to ensure the integrity and quality of project artifacts. |
| Business Designer | Details the specification of a part of the organization. The Business Designer specifies the workflow of business use cases in terms of business workers and business entities. It also distributes the behavior of these business workers and business entities-defining their responsibilities, operations, attributes, and relationships. |
| System Analyst | Leads and coordinates requirements elicitation and use-case modeling by outlining the system’s functionality and delimiting the system. |
| Requirements Specifier | Details the specification of a part of the system's functionality by describing the Requirements aspect of one or several use cases and other supporting software requirements. The requirements specifier may also be responsible for a use-case package, and maintains the integrity of that package. |
| Designer | Defines the responsibilities, operations, attributes, and relationships of one or several classes, and determines how they will be adjusted to the implementation environment. In addition, the designer role may have responsibility for one or more design packages, or design subsystems, including any classes owned by the packages or subsystems. |
| Implementer | Responsible for developing and testing components, in accordance with the project's adopted standards, for integration into larger subsystems. When test components, such as drivers or stubs, must be created to support testing, the Implementer is also responsible for developing and testing the test components and corresponding subsystems. |
| Tester | Responsible for the core activities of the test effort, which involves conducting the necessary tests and logging the outcomes of that testing. |
| Tool Specialist | Responsible for the supporting tools on the project. This includes selecting and acquiring tools. The tool specialist also configures and sets up the tools, and verifies that the tools work. |
| Test Designer | Responsible for the planning, design, implementation, and evaluation of testing, including generation of the test plan and test model, implementation of the test procedures, and evaluation of test coverage, test results, and effectiveness. |
| Deployment Manager | Responsible for planning the product's transition to the user community, ensuring those plans are enacted appropriately, managing issues and monitoring progress. |

# Management Process

## Project Plan

### Phase Plan

|  |  |  |  |
| --- | --- | --- | --- |
| **Task Name** | **START DATE** | **END DATE** | **DUARNATION (days)** |
| **Inception Phase** | 10/03/2016 | 29/03/2016 | 19 |
| Design Prototype Pages | 10/03/2016 | 14/03/2016 | 4 |
| Make Function List | 14/03/2016 | 17/03/2016 | 3 |
| Create System Request | 17/03/2016 | 24/03/2016 | 7 |
| Create Development Plan | 17/03/2016 | 24/03/2016 | 7 |
| Configure Version Control | 17/03/2016 | 24/03/2016 | 7 |
| Create SRS | 17/03/2016 | 29/03/2016 | 12 |
| **LCO Milestone** | 29/03/2016 | 29/03/2016 | - |
| **Elaboration Phase** | 24/03/2016 | 21/04/2016 | 27 |
| Design Database | 24/03/2016 | 29/03/2016 | 5 |
| Design Architecture | 24/03/2016 | 3/04/2016 | 9 |
| Configure Enviroment | 29/03/2016 | 3/04/2016 | 4 |
| Identify Use Cases | 29/03/2016 | 12/04/2016 | 13 |
| Build Sample System | 4/04/2016 | 12/04/2016 | 8 |
| Deploy Sample System | 13/04/2016 | 17/04/2016 | 4 |
| Test Sample System | 18/04/2016 | 21/04/2016 | 3 |
| **LCA Milestone** | 21/04/2016 | 21/04/2016 | - |
| **Construction Phase** | 21/04/2016 | 19/05/2016 | 28 |
| **Transition Phase** | 19/05/2016 | 2/06/2016 | 14 |



### Iteration Objectives

We choose one week as an iteration, each of which begins with a group meeting.

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **Iteration** | **Description** | **Associated Milestone** |
| Inception | I1 | Designs prototype pages, makes a function list |  |
| I2 | Creates Development Plan, creates System Request, configures version control system, creates Software Requirement Specification |  |
| I3 | Finishes Software Requirement Specification | Lifecycle Objectives Milestone |
| Elaboration | E1 | Designs database and system architecture, Configures implementation environment |  |
| E2 | Identifies use cases, builds sample system |  |
| E3 | Deploys and tests sample system | Lifecycle Architecture Milestone,  Release 0.1 |
| Construction | C1 |  |  |
| C2 |  |  |
| C3 |  |  |
| C4 |  |  |
| Transition | T1 |  |  |
| T2 |  |  |

### Releases

Release 0.1(internal release) is a sample system with basic functions. User should be able to login, ask questions, answer questions, view questions and answers.

## Iteration Plans

|  |  |  |
| --- | --- | --- |
| **Iteration** | **Date** | **Tasks** |
| Inception 1 | 10/03/2016 - 14/03/2016 | Designs prototype pages |
| 15/03/2016 - 17/03/2016 | Makes a function list |
| Inception 2 | 17/03/2016 - 24/03/2016 | Creates Development Plan, creates System Request, configures version control system, creates Software Requirement Specification |
| Inception 3 | 24/03/2016 - 29/03/2016 | Finishes Software Requirement Specification |
| Elaboration 1 | 24/03/2016 - 07/04/2016 | Designs database and system architecture, Configures implementation environment |
| Elaboration 2 | 07/04/2016 - 12/04/2016 | Identifies use cases, builds a sample system |
| Elaboration 3 | 12/04/2016 - 17/04/2016 | Deploys the sample system |
| 17/04/2016 - 21/04/2016 | Tests the sample system |

## Project Monitoring and Control

### Schedule Control Plan

The project manager will maintain a summary schedule showing the expected date of each milestone. Every week, using the weekly team meeting, the project manager will reevaluate the progress of the project, to determine whether the project is on schedule.

If the project is not on schedule, the project manager will consult with team members to determine corrective action, which may result in updating the schedule and/or reducing the number of optional functions that the system will perform.

### Quality Control Plan

All deliverables are required to go through the appropriate review process. The review is required to ensure that each deliverable is of acceptable quality, using guidelines described in the Rational Unified Process [3] review guidelines and checklists.

### Reporting Plan

An assignment report will be generated on 13/04/2016. It will include System Request, Development Plan, Risk List, Software Requirement Specification, Use Case Specification and Meeting Reports.

# Introduction

## Purpose

The purpose of this document is to support the development of the FQA System by documenting potential risks, their magnitudes, and mitigation strategies.

## Scope

The Risk List is only associated with FQA System.

## Definitions, Acronyms and Abbreviations

FQA: Find Question and answer

## References

Rational Unified Process: Risk List template (rup\_rskpln.dot)

## Overview

The risks known at the publication date of this document are listed below, along with mitigation strategies for each risk.

# Risks

## Project Risk: Hard Deadline

### Risk Magnitude or Ranking

Severe

### Description

Two hard deadlines to submit assignments are built into this project. The assignments may not be finished by the deadlines.

### Impacts

An incomplete product, a bad grade.

### Indicators

Falling behind in the Software Development Plan.

### Mitigation Strategy

Every member keeps reporting working progress on meetings. Hard milestones are being established.

### Contingency Plan

If project falls too far behind, the developers will begin to spend more time developing FQA System, sacrificing some documentation.

## Project Risk: Team Roster Change

### Risk Magnitude

Average to Severe

### Description

Some team members may exit the team. New members may join the team.

### Impacts

Need to assign roles and work again. Hard to continue with left members' work. New members may be inexperienced.

### Indicators

Some members may abandon CSCI822. Newcomers of CSCI822 may join our group.

### Mitigation Strategy

Team members should tell others about his assigned work so the process is not interrupted. Manager should tell new members how to work in development.

### Contingency Plan

Assign roles and work again to new team members.

## Project Risk: Manager’s non-presence

### Risk Magnitude

Average

### Description

We are following RUP process and in this process Manager plays a key role in keeping all documentation and general direction for the project. So Manager role is a key, responsible role in the project management.

### Impacts

Waste of time if the other project members understand and dispense the responsibilities and another general loss of direction in project management.

### Indicators

No work or very little work accomplished, assigned people not performing because of no answerability in absence of the manager.

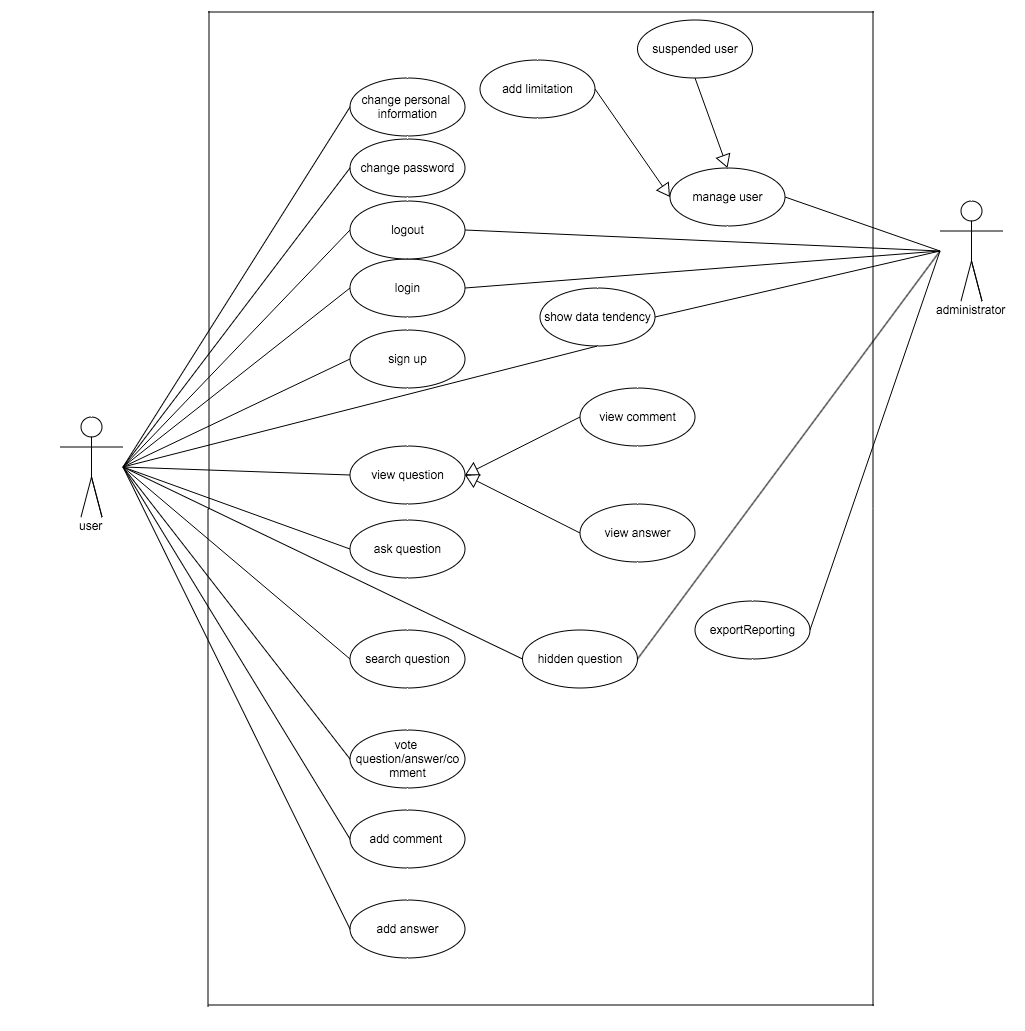
### Mitigation Strategy

There should be a proxy Manager who is always aware of the work done from the manager point of view, or more than one person as the manager.

### Contingency Plan

Somebody take the responsibility of manager and in a short notice at the cost of his/her work, does the work actually meant for the manager.

Use Case Diagram and Specification



|  |  |
| --- | --- |
| **Name: Sign Up** | **ID: PS\_US\_SU\_01** |
| **Stakeholders and goals:** User can sign up a new account | |
| **Description:** User can sign up a new account by providing some information | |
| **Trigger:** User click the button of sign up | |
| **Normal flow:**  1:User click sign up button in the top right corner。  2:Input their username  3:Check is their a account which has the same display name with the user’s input or not, if there is already someone named the display name, notice user, and check the length of display name, if it was not from 5 to 10, noticed user to change  4: Input password  5: If the length of password was not from 5 to 10, noticed user to change  6: Input confirm password, if it is not the same with password, noticed user  7: Input email  8: Check the format of email is right or not, if not, notice user.  9:Click sign up button  10: Check there are some notice or not, if there is some notice that will be forbid user to sign up  11: End | |
| **Sub-flows:** none | |
| **Alternative/Exceptional flows:**  none | |

|  |  |
| --- | --- |
| **Name: Login** | **ID: PS\_US\_LG\_01** |
| **Stakeholders and goals:** User can login according their username and password | |
| **Description:** User can login according their username and password | |
| **Trigger:** User click the button of login | |
| **Normal flow:**  1:User click login button in the top right corner。  2:Input their username/email and password  3:System will check that the username and the password is correct or not.  4:System will recognize the user is normal user or administrator  5:Redirect to different page according the user’s role  6:End | |
| **Sub-flows:** none | |
| **Alternative/Exceptional flows:**  none | |

|  |  |
| --- | --- |
| **Name: Log Out** | **ID: PS\_US\_LG\_02** |
| **Stakeholders and goals:** User can log out | |
| **Description:** User log out | |
| **Trigger:**  Log out | |
| **Normal flow:**  1:User click the button which show their display name in the top right corner。  2:Choose logout  3:End | |
| **Sub-flows:** change password | |
| **Alternative/Exceptional flows:**  none | |

|  |  |
| --- | --- |
| **Name: Manage User Profile** | **ID: PS\_US\_VP\_01** |
| **Stakeholders and goals:** User – to manage all profile about users’ | |
| **Description:** User through this case to manage their own profile | |
| **Trigger:** User want to manage their profile | |
| **Normal flow:**  1:User click the button which show their display name in the top right corner。  2:Choose management button  3:The system check user is illegal or not(if user do not do anything for 30 minutes, that user will be recognize as an illegal user, need login again)  4:User will choose which operate the want to do  5:End | |
| **Sub-flows:** change password | |
| **Alternative/Exceptional flows:**  none | |

|  |  |
| --- | --- |
| **Name: Change Password** | **ID:PS\_US\_CP\_02** |
| **Stakeholders and goals:** User can change their own password | |
| **Description:** User can change their own password | |
| **Trigger:** User click the button of update | |
| **Normal flow:**  1:User click update button。  2:Input new password  3: If the length of password was not from 5 to 10, noticed user to change  4: Click save button  5:Notice user this operation is successful or not  6: End | |
| **Sub-flows:** none | |
| **Alternative/Exceptional flows:**  none | |

|  |  |
| --- | --- |
| **Name: Question** | **ID:** **QS\_VQ\_01** |
| **Stakeholders and goals:** Anyone can scan the detail information for every question. | |
| **Description:** User can scan question | |
| **Trigger:** User click the title of each question | |
| **Normal flow:**  Consulate with its sub-flows. | |
| **Sub-flows:** ask question, search question, scan question detail information, add comment, add answer, Accept answer， sort answer, vote | |
| **Alternative/Exceptional flows:**  none | |

|  |  |
| --- | --- |
| **Name: View Question Detail Information** | **ID:QS\_QM\_VQA\_01** |
| **Stakeholders and goals:** User can scan the detail information for every question | |
| **Description:** User can scan question | |
| **Trigger:** User click the title of each question | |
| **Normal flow:**  1:Click question which do not have detail information  2:Redirect to question detail page  3:End | |
| **Sub-flows:** add answer, add comment | |
| **Alternative/Exceptional flows:**  none | |

|  |  |
| --- | --- |
| **Name: Add Comment** | **ID:** **QS\_CM\_AC\_01** |
| **Stakeholders and goals:** User can comment the question | |
| **Description:** User can comment the question | |
| **Trigger:** User click the button of add comment | |
| **Normal flow:**  1:Click add comment button  2:Show the input text to user to accept user’s input  3:Input comment  4:Click save button  5:Check user is legal or not  6:End | |
| **Sub-flows:** add answer, add comment | |
| **Alternative/Exceptional flows:**  none | |

|  |  |
| --- | --- |
| **Name: Add Answer** | **ID:QS\_AM\_ANA\_01** |
| **Stakeholders and goals:** User can answer the question | |
| **Description:** User can answer the question | |
| **Trigger:** User click the button of answer | |
| **Normal flow:**  1:Click answer button  2:Show the input text to user to accept user’s input  3:Input answer  4:Click save button  5:Check user is legal or not  6:End | |
| **Sub-flows:** | |
| **Alternative/Exceptional flows:**  none | |

|  |  |
| --- | --- |
| **Name: Accept answer** | **ID:** **QS\_AM\_AA\_01** |
| **Stakeholders and goals:** The question’s owner can choose which answer will be accepted. | |
| **Description:** The question’s owner can choose which answer will be accepted. | |
| **Trigger:** User click the button of accept | |
| **Normal flow:**  1:User click question’s title which belongs to the login user  2:Click the accept  3:Notice user successful or not  4: End | |
| **Sub-flows:** none | |
| **Alternative/Exceptional flows:** none | |

|  |  |
| --- | --- |
| **Name: Sort Answer** | **ID:** **QS\_AM\_SA\_01** |
| **Stakeholders and goals:** Sort the answer by creation time or vote number | |
| **Description:** Sort the answer by creation time or vote number | |
| **Trigger:** tick sort rules. | |
| **Normal flow:**  1:User click question  2:Click the sort rule, according to creation date is the default rule.  3: End | |
| **Sub-flows:** none | |
| **Alternative/Exceptional flows:**  none | |

|  |  |
| --- | --- |
| **Name: Show Comment** | **ID:** **QS\_CM\_VC\_01** |
| **Stakeholders and goals:** Show comments | |
| **Description:** Show comments | |
| **Trigger:** Click show more herf | |
| **Normal flow:**  1:User click question  2:Click ‘show more ’ button to see more comment.(there are 5 comments will be showed when show the question)  3: End | |
| **Sub-flows: the same as view question’s detail** | |
| **Alternative/Exceptional flows:**  none | |

|  |  |
| --- | --- |
| **Name: Vote** | **ID:QS\_CM\_VQA\_01** |
| **Stakeholders and goals:** vote the question and answer up or down | |
| **Description:** vote up or down | |
| **Trigger:** Click the button of vote up or vote down | |
| **Normal flow:**  1:User click question  2:Click up or down button  3: End | |
| **Sub-flows:** none | |
| **Alternative/Exceptional flows:**  none | |

|  |  |
| --- | --- |
| **Name: View the number of the Vote of Question and Answer** | **ID:** **QS\_CM\_VV\_01** |
| **Stakeholders and goals:** view times counting | |
| **Description:** view times counting | |
| **Trigger:** Click question’s title | |
| **Normal flow:**  1:User click question  2: End | |
| **Sub-flows:** none | |
| **Alternative/Exceptional flows:**  none | |

|  |  |
| --- | --- |
| **Name: Ask Question** | **ID: QS\_AM\_EA\_01** |
| **Stakeholders and goals:** User can scan ask a new question | |
| **Description:** User can ask a new question | |
| **Trigger:** User click the button ask question | |
| **Normal flow:**  1:User click ask question button  2:Check user is legal or not  3:Redirect to ask question page  4:Input question information  5:Click save button  6:Check user is legal or not  7:End | |
| **Sub-flows: no** | |
| **Alternative/Exceptional flows:**  none | |

|  |  |
| --- | --- |
| **Name: Search Question** | **ID:** **SS\_SQ\_01** |
| **Stakeholders and goals:** search question | |
| **Description:** search question | |
| **Trigger:** click the button of search question | |
| **Normal flow:**  1:User input what the want to know in index page  2:Click search button  3: End | |
| **Sub-flows:** sort searching question | |
| **Alternative/Exceptional flows:**  none | |

|  |  |
| --- | --- |
| **Name: Sort For Searching Question** | **ID:** **SS\_SR\_01** |
| **Stakeholders and goals: sort for** searching question | |
| **Description:** sort for searching question | |
| **Trigger:** Select sort rule in searching page | |
| **Normal flow:**  1:User input what the want to know in index page  2:Click search button  3:Choose sort rule  4: End | |
| **Sub-flows:** none | |
| **Alternative/Exceptional flows:**  none | |

|  |  |
| --- | --- |
| **Name: Manage User** | **ID:PS\_AMS\_MU\_01** |
| **Stakeholders and goals:** Administrator can manage user | |
| **Description:** Administrator can manage user | |
| **Trigger:** Administrator can manage user according the button of “manage user” | |
| **Normal flow:**  1: Administrator can manage user click manage user button  2:Show user’s important information (not personal information)  4:End | |
| **Sub-flows: add some limitation to user, suspend user** | |
| **Alternative/Exceptional flows:**  none | |

|  |  |
| --- | --- |
| **Name: Add some Limitation to User** | **ID: PS\_AMS\_AL\_01** |
| **Stakeholders and goals:** Administrator can add some extra limitation user | |
| **Description:** Administrator can add some extra limitation user | |
| **Trigger:** Administrator click the button of “as a administrator” | |
| **Normal flow:**  1: Administrator click the button of “as a administrator”  2:Check administrator is legal or not  3:Noitce operation is successful or not  4:End | |
| **Sub-flows: add some limitation to user, suspend user** | |
| **Alternative/Exceptional flows:**  none | |

|  |  |
| --- | --- |
| **Name: Suspended User** | **ID: PS\_AMS\_SU\_01** |
| **Stakeholders and goals:** Administrator can suspend user | |
| **Description:** Administrator can suspend user | |
| **Trigger:** Administrator click the button of “suspend” | |
| **Normal flow:**  1: Administrator click the button of “suspend”  2: Check administrator is legal or not  3: Notice operation is successful or not  4: End | |
| **Sub-flows: add some limitation to user, suspend user** | |
| **Alternative/Exceptional flows:**  none | |

|  |  |
| --- | --- |
| **Name: Operator** | **ID:PS\_AMS\_HD\_01** |
| **Stakeholders and goals:** Administrator can hide some illegal data | |
| **Description:** Administrator can hide some illegal data | |
| **Trigger:** Administrator click the button of “hidden” | |
| **Normal flow:**  1: Administrator click the button of “hidden”  2: Check administrator is legal or not  3: Notice operation is successful or not  4: End | |
| **Sub-flows: add some limitation to user, suspend user** | |
| **Alternative/Exceptional flows:**  none | |

|  |  |
| --- | --- |
| **Name: Show Data Tendency** | **ID: PS\_AS\_SDT\_01** |
| **Stakeholders and goals:** Show data tendency to administrator | |
| **Description:** Show data tendency to administrator | |
| **Trigger:** Show data tendency to administrator | |
| **Normal flow:**  1: Administrator log in  4:End | |
| **Sub-flows: add some limitation to user, suspend user** | |
| **Alternative/Exceptional flows:**  none | |

|  |  |
| --- | --- |
| **Name: Reporting System** | **ID: PS\_AS\_RT\_01** |
| **Stakeholders and goals:** Report data tendency | |
| **Description:** Report data tendency | |
| **Trigger:** Click the button export data | |
| **Normal flow:**  1: Administrator log in  2: Click the button of “export data”  3: Choose what need export  4: Click button “export”  5: End | |
| **Sub-flows: add some limitation to user, suspend user** | |
| **Alternative/Exceptional flows:**  none | |

# Introduction

## Purpose

The purpose of this document is to describe the specifications on the external behaviors of a question and answer system. It also documents nonfunctional requirements, design constraints and other factors necessary to provide a complete and comprehensive understanding of the To-Be system.

The intended audience of this document includes the prospective software development team and the potential users of the system.

## Scope

This software system is a system which is based on the interest and the browser/server structure and this system will be referred to as “FQA system” thorough this document.

The two main users of this FQA system are user and administrator. The purpose of administrator is to maintain smooth operations of the system and ensure the high quality of the questions and answers. Meanwhile, user uses this system mainly for asking their question and answering question of the other user. In order to meet the needs of user and administrator, the system is designed as five subsystems. Each of them is an independent but correlated subsystem.

* Reporting Subsystem generate various reports such as generate various reports such as the most interesting questions, weekly questions, monthly questions, the top users (in terms of reputations), etc.
* Question Subsystem allow user to manage questions, (correct) answers, their votes and associated comments.
* Search Subsystem page allow user use it to search their question and answer
* Profile Subsystem allow user to update their information and check some detail information of themselves
* Administration Subsystem allow administrator to administrate the system

## Definitions, Acronyms and Abbreviations

FQA: Find Question and answer

## References

* **Rational Unified Process**, SRS template (upedu\_srs.doc), COSC2151 Final Year Software

Engineering Project, RMIT International University Vietnam, 2004

## Overview

The rest of this document is divided into two main sections:

* The Overall Description (section 2) describes the general factors that affect the system and its requirements.
* The Specific Requirements (section 3) contains all software requirements that the system must meet in order to satisfy customer’s needs.

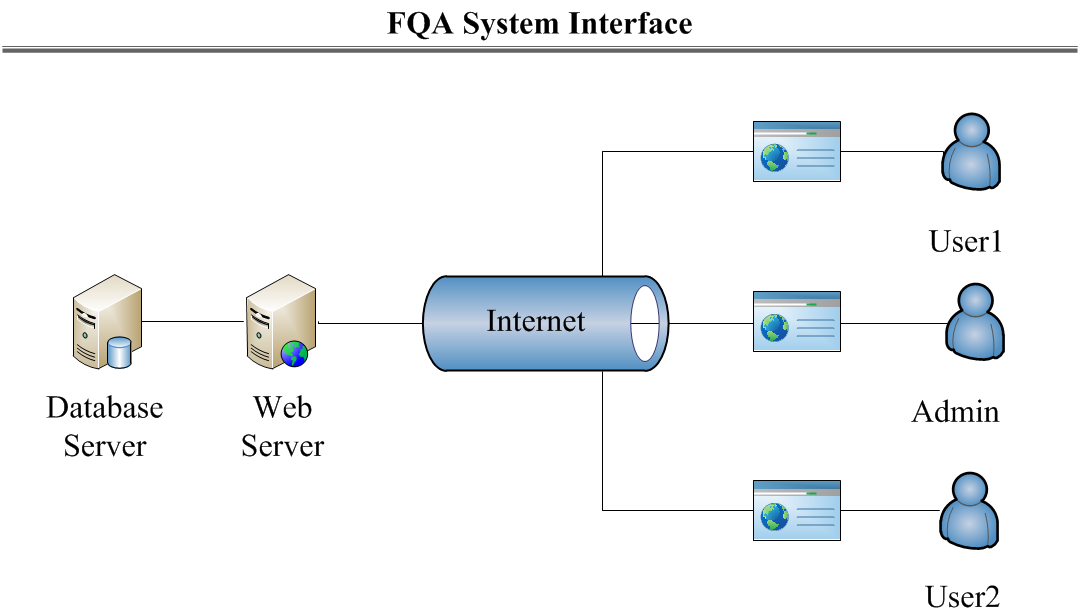
# Overall Description

## Product Perspective

Nowadays, there are many question and answer system exists in the internet and every of them have their own feature and function. However, many people have found that it is too difficult to match the right answer of their question effectively from those question and answer systems on the internet and sometime people need to spend many time on searching the answer.

Base on that need, it is necessary to building a question and answer system which help people to find the answers of their question easily and quickly. The goal of FAQ system is to save the money and time of people in spending on searching answers.

### System Interfaces

The FAQ system is a browser and server mode application that can be deployed on the Internet

### User Interfaces

The user interfaces can accessible through any web browser such as IE, Mozilla, FireFox, and Safari etc.

FQA system administrators also connect to the system via web browser like other users but administrators have their administrate center and they have higher access right to the system.

All users’ accesses to the Database, which is powered by MySQL server, must be performed indirectly through FQA system.

### Hardware Interfaces

All components must be able to execute on a personal computer.

### Software Interfaces

#### User Interface

* The user interact with the system through web browser
* The system supports both IE 5.5 or above and Firefox 7.0 or above

### Communication Interfaces

* The client machines must communicate with the Web Server over TCP/IP connection
* The Web Server and the Database Server are located on different servers

### Memory Constraints

* The client machine must be able to operate within 100MB minimum (including memory for browser)
* The Web Server and the Database Server must be able to operate within 256MB minimum

### Operations

FQA system should be easy for all users to use, e.g. no specific information or skills (except knowledge on how to access the Internet via Web browser) must be required to use the tool.

The Web Server installation and maintenance should be simple enough for a network administrator to perform and should not require any special technical skills from the administrator.

The Database Server should be able to import data from other external database systems. Backup and Recovery operations must be specified in case of network failure, database failure, out of power etc.

## Product Functions

The main function of the FQA system is to provide a platform for users to pose a question and answer their interested question.

For the users who have a question, FQA system provides the following main functions to meet the basic requirement of raising a question:

* Search a similar question. For example, some questions may exist and have a perfect answer so what they need to do is to search but not raise a question.
* Pose a question if their questions have not been posed.
* Discuss the answer with the user who has provided an answer for questioner. For instance, some users who would like to answer the question they may not understand the question fully or misunderstand the question and questioner can discuss with them through the way of comment.
* Adopt which answer is the best answer.

For the users who would like to answer the question, FQA system provides the following main functions for them to satisfy the needs of answering the question:

* Edit the answer before the questioner have adopt as the best answer. Some answers need to be edited serval times before they become a good or best answer.
* Comment the answer with questioner in order to solve the question perfectly.

Additionally, FQA system is controlled by administrators through the following main functions:

* Delete some questions, answers and comments which are illegal or disobey the system rules.
* Define the system configuration settings and maintain the system operations.

All functionalities of the system are built based on the basic needs of question and answer system so that FQA system can make all activities and communication carried out by questioner and answerer easier and more convenient.

## User Characteristics

The users of FQA system include system, user and moderator

* Administrators have strong knowledge on networks and web applications to be able to install and maintain FQA system. At the meanwhile, administrators are responsible for the quality of the question and maintain the daily operation of this system such as delete some question which were illegal etc.
* Moderator have some authority to guarantee the questions and answers
* User can raise a question and answer a question

## Constraints

The system should strictly obey and satisfy the following constraints:

* Authentication security: the system should enforce user authentication security
* Access control: the system must provide specific interface for administrator control the quality of the question
* Backup and recovery: the backup and recovery of all the system’s database must be easy to perform to prevent databases from corruption and loss risks
* The system must be developed using University of Wollongong facilities

## Assumptions and Dependencies

The following assumptions and dependencies for the system are stated:

* All potential users of E-Library system must have a unique username

# Specific Requirements

## Functional Requirements

### Profile Subsystem

This section captures functionalities that FQA system provides to administrators, moderators and users to create, view, edit user profiles.

#### User Side

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.1.1\_1** | **Requirement Type: Functional** | **Use Case:PS\_US\_SU\_01** |
| **Description:** The system should provide a User with a dialog box to sign up and set his/her Username, Password, E-mail address and Display name. | | |
| **Rationale:** A user wants to sign up. | | |
| **Source:** Unregistered User | | |
| **Fit Criterion:** A new user can be registered with profiles saved into database. | | |
| **Dependencies:** None | | |
| **Rank of importance:** Critical | | |
| **Supporting Materials:** None | | |
| **History:** Created by Xinchi Wang 19/03/2016 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.1.1\_2** | **Requirement Type: Functional** | **Use Case:PS\_US\_LG\_01** |
| **Description:** The system should provide a User with a dialog box to login to the website by entering his/her Username or E-mail and Password. | | |
| **Rationale:** A user wants to log in. | | |
| **Source:** Administrator, Users | | |
| **Fit Criterion:** An existing user can log in successfully | | |
| **Dependencies:** The user must exist in database. | | |
| **Rank of importance:** Critical | | |
| **Supporting Materials:** None | | |
| **History:** Created by Xinchi Wang 19/03/2016 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.1.1\_3** | **Requirement Type: Functional** | **Use Case:****PS\_US\_LG\_02** |
| **Description:** The system should provide a User with a dialog box to login out | | |
| **Rationale:** A user wants to log out. | | |
| **Source:** Administrator, Users | | |
| **Fit Criterion:** An existing user can log out successfully | | |
| **Dependencies:** The user must exist in database. | | |
| **Rank of importance:** Critical | | |
| **Supporting Materials:** None | | |
| **History:** Created by Caixing Su 09/04/2016 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.1.1\_4** | **Requirement Type: Functional** | **Use Case:PS\_US\_VP\_01** |
| **Description:** The system should provide a User with a page to view his/her own profile, including Username, Display name, Age, E-mail, Reputation, Last access date, Account creation date, Location, Number of up votes and down votes, Answers Submitted, Questions asked and Favorite tags. | | |
| **Rationale:** A User wants to view his/her profile. | | |
| **Source:** Administrator, Moderators, User | | |
| **Fit Criterion:** A user profile should be viewed successfully. | | |
| **Dependencies:** The to-be-viewed profile must exist. | | |
| **Rank of importance:** Critical | | |
| **Supporting Materials:** None | | |
| **History: Created** by Xinchi Wang 19/03/2016 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.1.1\_5** | **Requirement Type: Functional** | **Use Case:PS\_US\_CP\_01** |
| **Description:** The system should provide a user with a page to change his/her own profile, including Display name, Age, E-mail, Location and Favorite tags. | | |
| **Rationale:** A user wants to change his/her profile. | | |
| **Source:** Administrator, Moderator, User | | |
| **Fit Criterion:** A user profile should be changed successfully. | | |
| **Dependencies:** The to-be-changed profile must exist. | | |
| **Rank of importance:** Critical | | |
| **Supporting Materials:** None | | |
| **History:** Created by Xinchi Wang 19/03/2016 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.1.1\_6** | **Requirement Type: Functional** | **Use Case:PS\_US\_CP\_02** |
| **Description:** The system should provide a user with a page to change his/her own password of his/her own account | | |
| **Rationale:** A user wants to change his/her password of his/her account | | |
| **Source:** Administrator, Moderator, User | | |
| **Fit Criterion:** A user profile should be changed successfully. | | |
| **Dependencies:** The to-be-changed profile must exist. | | |
| **Rank of importance:** Critical | | |
| **Supporting Materials:** None | | |
| **History:** Created by Xinchi Wang 19/03/2016 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.1.1\_7** | **Requirement Type: Functional** | **Use Case:** |
| **Description:** The system should automatically send an notice e-mail to Users with enough reputation to become a Moderator | | |
| **Rationale:** A User gains enough reputation to become a Moderator | | |
| **Source:** User | | |
| **Fit Criterion:** The e-mail should be sent successfully | | |
| **Dependencies:** The reputation must be enough  The e-mail address must exist in profile | | |
| **Rank of importance:** Important | | |
| **Supporting Materials:** None | | |
| **History:** Created by Xinchi Wang 11/04/2016 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.1.1\_8** | **Requirement Type: Functional** | **Use Case:** |
| **Description:** The system should provide an Administrator with a page to view any User profile, including Username, Display name, Age, E-mail, Reputation, Last access date, Account creation date, Location, Number of up votes and down votes, Answers Submitted, Questions asked and Favorite tags. | | |
| **Rationale:** An administrator wants to view user profiles. | | |
| **Source:** Administrator | | |
| **Fit Criterion:** The profile should be viewed successfully. | | |
| **Dependencies:** The to-be-viewed profile must exist. | | |
| **Rank of importance:** Critical | | |
| **Supporting Materials:** None | | |
| **History:** Created by Xinchi Wang 01/04/2016 | | |

#### Administrator Side

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.1.2\_1** | **Requirement Type: Functional** | **Use Case:PS\_AS\_VUP\_01** |
| **Description:** The system should provide an Administrator with a page to view any User profile, including Username, Display name, Age, E-mail, Reputation, Last access date, Account creation date, Location, Number of up votes and down votes, Answers Submitted, Questions asked and Favorite tags. | | |
| **Rationale:** An administrator wants to view user profiles. | | |
| **Source:** Administrator | | |
| **Fit Criterion:** The profile should be viewed successfully. | | |
| **Dependencies:** The to-be-viewed profile must exist. | | |
| **Rank of importance:** Critical | | |
| **Supporting Materials:** None | | |
| **History:** Created by Xinchi Wang 01/04/2016 | | |

### Question Subsystem

This section captures functionalities that FQA system provides to administrators, moderators and users to manage questions, answers and comments.

#### Question Management

This section includes all functions which are related to creating, editing and viewing questions.

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.2.1\_1** | **Requirement Type: Functional** | **Use Case:QS\_QM\_AQ\_01** |
| **Description:** The system should provide a user with a page to ask a new question with Question title, Question contents and Tags. | | |
| **Rationale:** A user wants to ask a question. | | |
| **Source:** Moderator, User | | |
| **Fit Criterion:** A question should be added to the question list. | | |
| **Dependencies:** The new question should be added successfully. | | |
| **Rank of importance:** Critical | | |
| **Supporting Materials:** None | | |
| **History:** Created by Xinchi Wang 19/03/2016 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.2.1\_2** | **Requirement Type: Functional** | **Use Case:QS\_QM\_EQ\_01** |
| **Description:** The system should provide a user with a page to edit his/her own question, including Question contents and Tags. | | |
| **Rationale:** A user wants to edit his/her question. | | |
| **Source:** Moderator, User | | |
| **Fit Criterion:** The question should be edited successfully. | | |
| **Dependencies:** The to-be-edited question must exist. | | |
| **Rank of importance:** Critical | | |
| **Supporting Materials**: None | | |
| **History:** Created by Xinchi Wang 19/03/2016 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.2.1\_3** | **Requirement Type: Functional** | **Use Case:QS\_QM\_VQA\_01** |
| **Description:** The system should provide a user with a page to view a question and its answers. | | |
| **Rationale:** A user wants to read a question and its answers. | | |
| **Source:** Moderator, User | | |
| **Fit Criterion:** The question and its answers should be viewed successfully. | | |
| **Dependencies:** The to-be-viewed question must exist. | | |
| **Rank of importance:** Critical | | |
| **Supporting Materials:** None | | |
| **History:** Created by Xinchi Wang 19/03/2016 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.2.1\_4** | **Requirement Type: Functional** | **Use Case:QS\_QM\_RE\_01** |
| **Description:** The system should provide a user with a checkbox when the user submits the question to see whether the user wants to receive an alert e-mail. If so, the system should send an e-mail to the user’s e-mail address when the question is answered. | | |
| **Rationale:** A user wants to receive alert e-mail. | | |
| **Source:** Moderator, User | | |
| **Fit Criterion:** The alert e-mail should be sent successfully. | | |
| **Dependencies:** The question must be submitted. | | |
| **Rank of importance:** Important | | |
| **Supporting Materials:** None | | |
| **History:** Created by Xinchi Wang 01/04/2016 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.2.1\_5** | **Requirement Type: Functional** | **Use Case:QS\_VQ\_01** |
| **Description:** The system should provide a question list for anyone who wants to view some questions of the database. | | |
| **Rationale:** Anyone wants to view some question without sign up | | |
| **Source:** Anyone | | |
| **Fit Criterion:** None | | |
| **Dependencies:** None | | |
| **Rank of importance:** Important | | |
| **Supporting Materials:** None | | |
| **History:** Created by Caixing Su 09/04/2016 | | |

#### Answer Management

This section includes all functions which are related to creating and editing answers.

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.2.2\_1** | **Requirement Type: Functional** | **Use Case:QS\_AM\_ANA\_01** |
| **Description:** The system should provide a user with a page to add a new answer to a question. | | |
| **Rationale:** A user wants to answer a question. | | |
| **Source:** Moderator, User | | |
| **Fit Criterion:** The answer should be added successfully. | | |
| **Dependencies:** The to-be-answered question must exist. | | |
| **Rank of importance:** Critical | | |
| **Supporting Materials:** None | | |
| **History:** Created by Xinchi Wang 19/03/2016 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.2.2\_2** | **Requirement Type: Functional** | **Use Case:****QS\_AM\_EA\_01** |
| **Description:** The system should provide a user with a page to edit his/her answer. | | |
| **Rationale:** A user wants to edit an answer. | | |
| **Source:** Moderator, User | | |
| **Fit Criterion:** The answer should be edited successfully. | | |
| **Dependencies:** The to-be-edited answer must exist. | | |
| **Rank of importance:** Important | | |
| **Supporting Materials:** None | | |
| **History:** Created by Xinchi Wang 01/04/2016 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.2.2\_3** | **Requirement Type: Functional** | **Use Case:QS\_AM\_AA\_01** |
| **Description:** The system should provide a user with a button to accept an answer to his/her question. | | |
| **Rationale:** A user wants to accept an answer. | | |
| **Source:** Moderator, User | | |
| **Fit Criterion:** The answer should be accepted successfully. | | |
| **Dependencies:** The to-be-accepted answer must exist. | | |
| **Rank of importance:** Important | | |
| **Supporting Materials:** None | | |
| **History:** Created by Xinchi Wang 01/04/2016 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.2.2\_4** | **Requirement Type: Functional** | **Use Case:QS\_AM\_SA\_01** |
| **Description:** The system should provide a user with an option to sort answers by time or voting. | | |
| **Rationale:** A user wants to sort answers. | | |
| **Source:** Moderator, User | | |
| **Fit Criterion:** The answer should be sorted successfully. | | |
| **Dependencies:** The to-be-sorted answers must exist. | | |
| **Rank of importance:** Important | | |
| **Supporting Materials:** None | | |
| **History:** Created by Xinchi Wang 01/04/2016 | | |

#### Comment Management

This section includes all functions which are related to creating and viewing comments.

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.2.3\_1** | **Requirement Type: Functional** | **Use Case:QS\_CM\_AC\_01** |
| **Description:** The system should provide a user with a page to add comments to a question. | | |
| **Rationale:** A user wants to add comments to a question. | | |
| **Source:** Moderator, User | | |
| **Fit Criterion:** The comments should be added successfully. | | |
| **Dependencies:** None | | |
| **Rank of importance:** Critical | | |
| **Supporting Materials:** None | | |
| **History:** Created by Xinchi Wang 19/03/2016 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.2.3\_2** | **Requirement Type: Functional** | **Use Case:QS\_CM\_VC\_01** |
| **Description:** The system should provide a user with a page to view comments to a question. | | |
| **Rationale:** A user wants to view comments to a question. | | |
| **Source:** Moderator, User | | |
| **Fit Criterion:** The comments should be viewed successfully. | | |
| **Dependencies:** The to-be-viewed comments must exist. | | |
| **Rank of importance:** Critical | | |
| **Supporting Materials:** None | | |
| **History:** Created by Xinchi Wang 19/03/2016 | | |

#### Voting Management

This section includes all functions which are related to voting.

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.2.4\_1** | **Requirement Type: Functional** | **Use Case:****QS\_CM\_VV\_01** |
| **Description:** The system should provide a user with a section to view up/down votes to questions and answers. | | |
| **Rationale:** A user wants to view comments to questions or answers. | | |
| **Source:** Moderator, User | | |
| **Fit Criterion:** The comments should be viewed successfully. | | |
| **Dependencies:** The to-be-viewed comments must exist. | | |
| **Rank of importance: Critical** | | |
| **Supporting Materials:** None | | |
| **History:** Created by Xinchi Wang 19/03/2016 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.2.4\_2** | **Requirement Type: Functional** | **Use Case:QS\_CM\_VQA\_01** |
| **Description:** The system should provide a user with buttons to vote up/down to a question or an answer. | | |
| **Rationale:** A user wants to vote up/down to a question or an answer. | | |
| **Source:** Moderator, User | | |
| **Fit Criterion:** The vote should be added successfully. | | |
| **Dependencies:** The to-be-voted answer must exist. | | |
| **Rank of importance:** Critical | | |
| **Supporting Materials:** None | | |
| **History:** Created by Xinchi Wang 19/03/2016 | | |

### Search Subsystem

This section captures functionalities that FQA system provides to moderators and users to search questions by keywords or tags.

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.3\_1** | **Requirement Type: Functional** | **Use Case:SS\_SQ\_01** |
| **Description:** The system should provide a user with a function to search for questions by keywords or tags. Then the system should provide the user with a page to show search results. | | |
| **Rationale:** A user wants to search for questions. | | |
| **Source:** Moderator, User | | |
| **Fit Criterion:** The search results should be displayed successfully. | | |
| **Dependencies:** None | | |
| **Rank of importance:** Critical | | |
| **Supporting Materials:** None | | |
| **History:** Created by Xinchi Wang 01/04/2016 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.3\_2** | **Requirement Type: Functional** | **Use Case:SS\_SR\_01** |
| **Description:** The system should provide a user with a button to sort search results by time or voting. | | |
| **Rationale:** A user wants to sort search results. | | |
| **Source:** Moderator, User. | | |
| **Fit Criterion:** The search results should be sorted successfully. | | |
| **Dependencies:** The search results are generated. | | |
| **Rank of importance:** Important. | | |
| **Supporting Materials:** None. | | |
| **History:** Created by Xinchi Wang 01/04/2016 | | |

### Reporting Subsystem

This section captures functionalities that FQA system provides to moderators and administrators to get reports of questions and users.

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.4\_1** | **Requirement Type: Functional** | **Use Case: PS\_AS\_RT\_01** |
| **Description:** The system should be able to generate a report of the most answered and voted questions asked in both week and month. | | |
| **Rationale:** An Administrator wants a report of hot questions | | |
| **Source:** Administrator | | |
| **Fit Criterion:** The report is generated successfully | | |
| **Dependencies:** The questions data must exist | | |
| **Rank of importance:** Important | | |
| **Supporting Materials:** None. | | |
| **History:** Created by Akshay Lakhanpal 09/04/2016,  Edited by Xinchi Wang 11/04/2016 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.4\_2** | **Requirement Type: Functional** | **Use Case: PS\_AS\_RT\_01** |
| **Description:** The system should be able to generate a report of the reputation of users. | | |
| **Rationale:** An Administrator wants a report of users’ reputation | | |
| **Source:** Administrator | | |
| **Fit Criterion:** The report is generated successfully | | |
| **Dependencies:** The reputation data must exist | | |
| **Rank of importance:** Important | | |
| **Supporting Materials:** None. | | |
| **History:** Created by Akshay Lakhanpal 09/04/2016,  Edited by Xinchi Wang 11/04/2016 | | |

### Administration Subsystem

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.5\_1** | **Requirement Type: Functional** | **Use Case:PS\_AMS\_MU\_01** |
| **Description:** The system should provide an Administrator with a page to manage all the Users including to limit a User’s functions and to suspend a User. | | |
| **Rationale:** An Administrator wants to manage Users | | |
| **Source:** Administrator | | |
| **Fit Criterion:** The list of user is displayed | | |
| **Dependencies:** The users profiles must exist | | |
| **Rank of importance:** Important | | |
| **Supporting Materials:** None | | |
| **History:** Created by Caixing Su 10/04/2016  Edited by Xinchi Wang 11/04/2016 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.5\_2** | **Requirement Type: Functional** | **Use Case:****PS\_AMS\_AL\_01** |
| **Description:** The system should provide an Administrator with a dialog box to limit functions of a User. | | |
| **Rationale:** An Administrator wants to limit functions of a User | | |
| **Source:** Administrator | | |
| **Fit Criterion:** The User’s functions are successfully limited | | |
| **Dependencies:** The User’s profile must exist | | |
| **Rank of importance:** Important | | |
| **Supporting Materials:** None | | |
| **History:** Created by Caixing Su 10/04/2016  Edited by Xinchi Wang 11/04/2016 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.5\_3** | **Requirement Type: Functional** | **Use Case:****PS\_AMS\_HD\_01** |
| **Description:** The system should provide an Administrator or a Moderator with a page to hide answers or questions with illegal contents | | |
| **Rationale:** An administrator wants to hide a question or an answer | | |
| **Source:** Administrator, Moderator | | |
| **Fit Criterion:** The question or answer is successfully hidden | | |
| **Dependencies:** The to-be-hidden question or answer must exist | | |
| **Rank of importance:** Important | | |
| **Supporting Materials:** None | | |
| **History:** Created by Caixing Su 10/04/2016  Edited by Xinchi Wang 11/04/2016 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.5\_4** | **Requirement Type: Functional** | **Use Case:PS\_AS\_SDT\_01** |
| **Description:** The system should provide an administrator with a page to view the data tendency of the system. | | |
| **Rationale:** An administrator wants to view user profiles. | | |
| **Source:** Administrator | | |
| **Fit Criterion:** This function should only be used by administrator. | | |
| **Dependencies:** The role of the user must be administrator | | |
| **Rank of importance:** Critical | | |
| **Supporting Materials:** None | | |
| **History:** Created by Caixing Su 10/04/2016 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.5\_5** | **Requirement Type: Functional** | **Use Case:** |
| **Description:** System administrator should have ability to disable unnecessary services. User does not able to access anything without permission of admin. Report malicious or suspicious activity on systems and install patches. | | |
| **Rationale:** To maintain the system up-to-date. | | |
| **Source:** Administrator. | | |
| **Fit Criterion:** Strictly used by Administrator. | | |
| **Dependencies:** Reports from user. | | |
| **Rank of importance:** Critical. | | |
| **Supporting Materials:** None. | | |
| **History:** created by Akshay Lakhanpal 31/03/2016 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.5\_6** | **Requirement Type: Functional** | **Use Case:AS\_CS\_01** |
| **Description:** System administrator can view a statistical user-end data and compare among all the user’s their performance and inform them by system messages according to outcome. | | |
| **Rationale:** To provide Administrator data of user in statistical. | | |
| **Source:** System administrator. | | |
| **Fit** **Criterion:** The character of the user should be the system administrator and normal user should not be allowed to check the statistical data. | | |
| **Dependencies:** The character of the user are system administrator | | |
| **Rank of importance:** Critical. | | |
| **Supporting Materials:** None. | | |
| **History:** created by Akshay Lakhanpal 31/03/2016,changed by Caixing Su 09/04/2016 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement#:F3.1.5\_7** | **Requirement Type: Functional** | **Use Case:** |
| **Description:** System audit login. In addition to documenting which resources were accessed, audit log entries usually include destination and source addresses, a timestamp and user login information. | | |
| **Rationale:** Record of a user. | | |
| **Source:** Administrator. | | |
| **Fit Criterion:** Strictly used by Administrator. | | |
| **Dependencies:** User account detalis. | | |
| **Rank of importance:** Critical. | | |
| **Supporting Materials:** None. | | |
| **History:** created by Akshay Lakhanpal 31/03/2016 | | |

## Non-Functional Requirements

### Usability

|  |  |  |
| --- | --- | --- |
| **Requirement#:NF3.2.1\_1** | **Requirement Type: Usability** | **Use Case:#NONE** |
| **Description:** The system should maintain consistency in the layout of user interfaces | | |
| **Rationale:** To provide users with consistent user interfaces and layout | | |
| **Source:** Administrator, Moderator, User | | |
| **Fit Criterion:** All pages of the system follow the defined template and structure as specified in the layout of user interfaces | | |
| **Dependencies:** None | | |
| **Rank of importance:** Essential | | |
| **Supporting Materials:** None | | |
| **History:** Created by Caixing Su 29/03/2016 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement#:NF3.2.1\_2** | **Requirement Type: Usability** | **Use Case:#NONE** |
| **Description:** The system should not require users to have any specific knowledge or special technical skill to be able to use it. FQA system should be easy for a computer-literate user community with no additional training on the system so that users can use it after at most half of one day of exploring the system. | | |
| **Rationale:** Users want to use the system without having to take any training records on the system. | | |
| **Source:** Administrator, Moderator, User | | |
| **Fit Criterion:** The users can use FQA system after at most half of one day of exploring the system. | | |
| **Dependencies:** None | | |
| **Rank of importance:** Essential | | |
| **Supporting Materials:** None | | |
| **History:** Created by Caixing Su 29/03/2016 | | |

### Security

|  |  |  |
| --- | --- | --- |
| **Requirement#:NF3.2.2\_1** | **Requirement Type: Security** | **Use Case:#NONE** |
| **Description:** The system should automatically log out after thirty minutes of being idle and ask the user to re-log in. | | |
| **Rationale:** To control authorized usage of the system. | | |
| **Source:** Administrator, Moderator, User | | |
| **Fit Criterion:** The system should automatically log out after thirty minutes of being idle. | | |
| **Dependencies:** None | | |
| **Rank of importance:** Desirable | | |
| **Supporting Materials:** None | | |
| **History:** Created by Caixing Su 29/03/2016 | | |

### Reliability

|  |  |  |
| --- | --- | --- |
| **Requirement#:NF3.2.3\_1** | **Requirement Type: Reliability** | **Use Case:#NONE** |
| **Description:** The system should be able to recover from failure after no more than 24 hours. | | |
| **Rationale:** The system should recover from failure as soon as possible to support 24/7 services. | | |
| **Source:** Administrator, Moderator, User | | |
| **Fit Criterion:** The system should recover after no more than 24 hours. | | |
| **Dependencies:** None | | |
| **Rank of importance:** Essential | | |
| **Supporting Materials:** None | | |
| **History:** Created by Caixing Su 29/03/2016 | | |

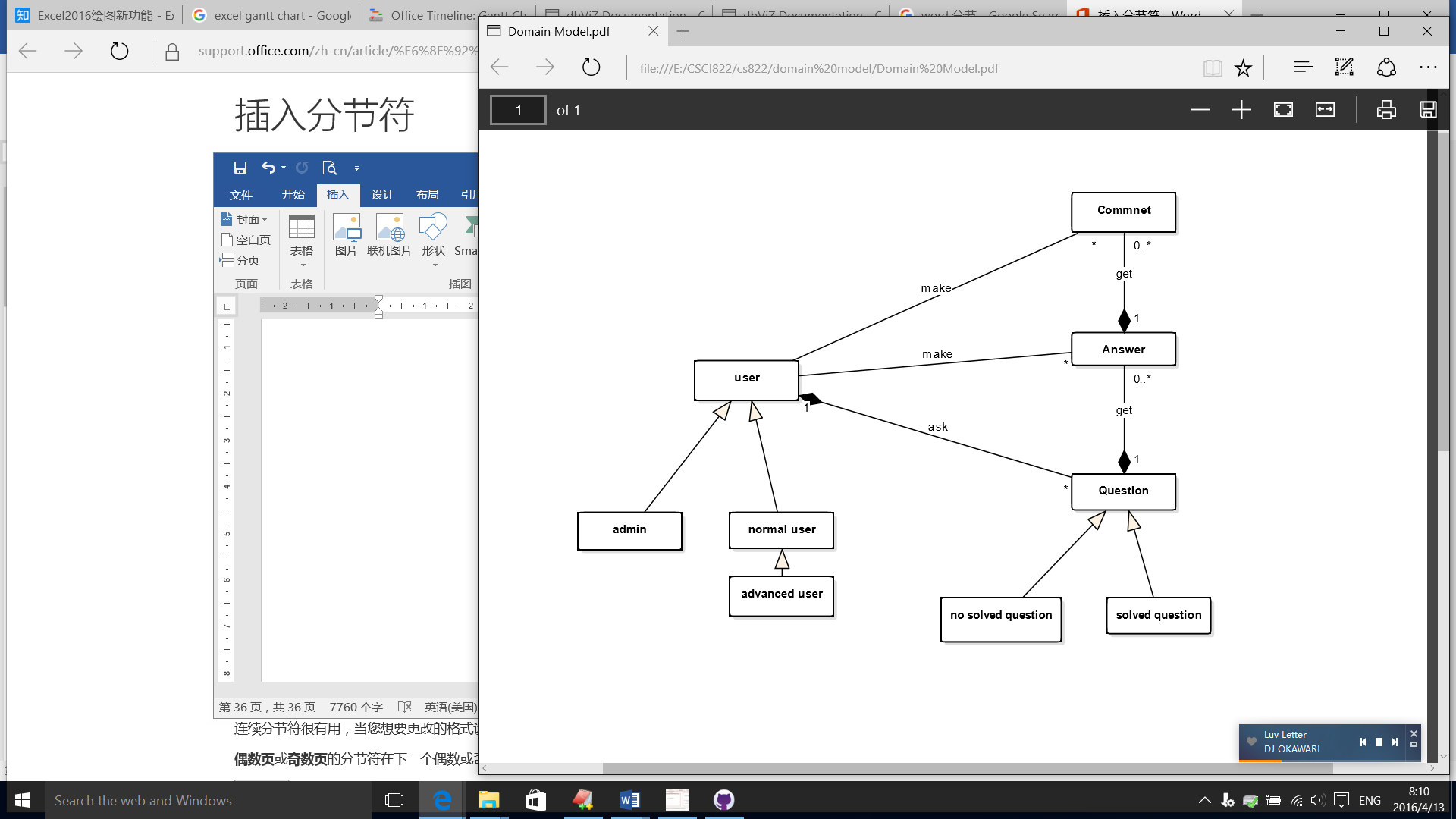
### Performance

|  |  |  |
| --- | --- | --- |
| **Requirement #: NF3.2.4\_1** | **Requirement Type:** **Performance** | **Use Case:#NONE** |
| **Description:** The system should be able to work 24 hours per day, seven days per week except during the maintenance time. | | |
| **Rationale:** All users want to use the system at anytime (24 hours per day, seven days per week). | | |
| **Source:** Administrator, Moderator, User | | |
| **Fit Criterion:** Any authorized user can use the system at anytime (24 hours per day, seven days  per week). | | |
| **Dependencies:** None | | |
| **Rank of importance:** Essential | | |
| **Supporting Materials:** None | | |
| **History:** Created by Caixing Su 29/03/2016 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #: NF3.2.4.\_2** | **Requirement Type: Performance** | **Use Case:#NONE** |
| **Description:** The system should response to any action of any user within 15 seconds. | | |
| **Rationale:** A user wants to receive response to his action quickly. | | |
| **Source:** Administrator, Moderator, User | | |
| **Fit Criterion:** Any response to any user’s action should be performed within 40 seconds. | | |
| **Dependencies:** None | | |
| **Rank of importance:** Essential | | |
| **Supporting Materials:** None | | |
| **History:** Created by Caixing Su 29/03/2016 | | |

|  |  |  |
| --- | --- | --- |
| **Requirement #: NF3.2.4\_4** | **Requirement Type: Performance** | **Use Case:#NONE** |
| **Description:** The system should display search results within 15 seconds. | | |
| **Rationale:** A user wants to receive results of his query quickly. | | |
| **Source:** Administrator, Moderator, User | | |
| **Fit Criterion:** Search results should be displayed within 40 seconds | | |
| **Dependencies:** None | | |
| **Rank of importance:** Essential | | |
| **Supporting Materials:** None | | |
| **History:** Created by Caixing Su 29/03/2016 | | |

Domain Model



Data Dictionary

Class name: User

Superclasses: none

Attributes:

-Id: this is an int number that can increase by itself, it is the symbol to recognize the object of user

-Reputation: every question has reputation, if the asker accepts one answer; the owner of that answer will get the reputation for that question.

-CreationDate: the account create date.

-DisplayName: like a nick name that to use to show how to call user in the website.

-LastAccessDate: the date of last time when account login.

-Location: location

-Views: the whole scan times for all question account asked.

-UpVotes: the while up votes for all question and answer which belong to this account.

-DownVotes: the while down votes for all question and answer which belong to this account.

-Age: age

－type: type, only administrator can change this field, 0 means normal user, 999 means administrator, 777 means advancer, administrator have the limitation to give extra limitation to other users.

Method:

+setId(id): to set the field of id;

+getId(): to get the field of id;

+setReputation (Reputation): to set the field of Reputation;

+getReputation (): to get the field of Reputation;

+setCreationDate (CreationDate): to set the field of CreationDate;

+getCreationDate (): to get the field of CreationDate;

+setDisplayName (DisplayName): to set the field of DisplayName;

+getDisplayName (): to get the field of DisplayName;

+setLastAccessDate (LastAccessDate): to set the field of LastAccessDate;

+getLastAccessDate(): to get the field of LastAccessDate;

+setLocation (location): to set the field of Location;

+getLocation(): to get the field of Location;

+setViews (Views): to set the field of Views;

+getViews (): to get the field of Views;

+setUpVotes (UpVotes): to set the field of UpVotes;

+getUpVotes (id): to get the field of UpVotes;

+setDownVotes (DownVotes): to set the field of DownVotes;

+getDownVotes (): to get the field of DownVotes;

+setAge (Age): to set the field of Age;

+getAge (): to get the field of Age;

+setType(Type): to set the field of Type;

+getType (): to get the field of Type;

+checkUserByDisplayName(displayname) to check is there a user’s display name named displayname or not. Return User, if there is not such a user named displaynam, return null.

+checkUserByEmail(email) to check email is already been used or not, return User, if that email are not exist in database, return null.

+checkUser() to check the user is legal or not. Legal will return true else return false. if user do not have any operation, it will be return false. Need login again.

+login(displayNameOrEmail, password) to check there is a user who display name equals displayNameOrEmail and password equals password or email equals displayNameOrEmail and password equals to password.

+cechDisplayName(displayname) check the displayname is legal or not, only user while sign up.

+checkEmail(email) check the email format is legal or not

+signup(User) insert a new user’s information into database.

+logout() log out.

+changePassword(password) change password

Class name: Question

Superclasses: none

Attributes:

-Id: a field to recognize questions

- AcceptedAnswerId: every question just have one accepted answer, this field will store the accepted answer’s id

- Score: how many reputations to answer this question will get

- ViewCount: view count

- Body: question’s content

- CreationDate: question’s create date

- OwnerUserId: question’s owner’s id

- Title: question’s title

- Targs: question’s targ

- AnswerCount: question’s answer counting

- CommentCount: question’s comment counting

Method:

+sets(Field): every attribute has their own set method to set that attribute value

+gets():every attribute has their own get method to get that attribute value

+addQuestion(Question):User can ask a question, according this method, it will create a record of question in database

+getQuestionsOrderByCreateDate(currentPage,limitNumber):get several questions from currentPage\*limitNumber to currentPage\*limitNumber+limitNumber.

+getQuestionByQuestionId (id):get one question according to id. It will make this question’s view count add 1 and the owner of this question’s view count add 1.

+searchQuestion(condition):search questions which the question title or body like condition.

+UpdateQuestion(question):update the data of this question.

+findQuestionByHot():find several lines of the hottest questions.

+findRandomQuestion():find several lines of the random questions.

+findRecommendQuestion(user):find several lines of the recommend questions. According user’s information.

+upVote():up vote this question. It will make this question’s up vote count add 1 and the owner of this question’s up vote count add 1.

+downVote():up vote this question, It will make this question’s down vote count add 1 and the owner of this question’s down vote count add 1.

Class name: Answer

Superclasses: none

Attributes:

-Id: a field to recognize answer

- ParentId: recognize which question this answer answered.

- ViewCount: view count

- Body: answer’s content

- CreationDate: answer’s create date

- OwnerUserId: answer’s owner’s id

- AnswerCount: answer’s answer counting

- CommentCount: answer’s comment counting

Method:

+sets(Field): every attribute has their own set method to set that attribute value

+gets():every attribute has their own get method to get that attribute value

+findAnswersByQuestionId(id): find all answers which belong to a question which id equals id.

+addAnswer (Answer): Add a answer for a question

Class name: Comment

Superclasses: none

Attributes:

-Id: a field to recognize comment

- postId: recognize which question or answer this comment belongs to.

- text: comment’s content

- CreationDate: answer’s create date

- OwnerUserId: answer’s owner’s id



Method:

+sets(Field): every attribute has their own set method to set that attribute value

+gets():every attribute has their own get method to get that attribute value

+addComment(Comment): Add a comment

+finCommentByPostId (postid): find all comments which postid equals to postid.

Group Members

|  |  |
| --- | --- |
| **Role** | **Names** |
| Project Manager | Xinchi Wang |
| Business Designer | Caixing Su |
| System Analyst | Caixing Su, Xinchi Wang |
| Designer | Rixin Nie, Kaijian Feng |
| Requirement Specifier | Caixing Su, Xinchi Wang, Akshay Lakhanpal |
| Implementer | Rixin Nie, Kaijian Feng |
| Test Designer | Kaijian Feng |
| Tester | Kaijian Feng, Akshay Lakhanpal |
| Tool Specialist | Kaijian Feng |
| Deployment Manager | Kaijian Feng |

|  |  |
| --- | --- |
| **Member** | **Artefact** |
| Xinchi Wang | Software Development Plan, Risk List, Software Requirement Specification (part) |
| Caixing Su | System Request, Software Requirement Specification (part) |
| Kaijian Feng | System Request (part), Version Control, Web Pages |
| Rixin Nie | Use Case, Domain Model, Data Dictionary, Database Model, Website Back-end Codes |
| Akshay Lakhanpal | Software Requirement Specification (part) |

Group Meeting Reports

**Week 2**

**Thursday 10th March, 2016**

**10:00 AM**

**At GS3, Library**

**Attendees: Yixiang Fan**

**Rui Zhang**

**Rixin Nie**

**Caixing Su**

**Kaijian Feng**

**Xinchi Wang**

**Item 1: New Members**

Xinchi Wang and Kaijian Feng temporarily joined the team

**Item 2: Prototype Page Design**

Each member showed one page

**Item 3: Tasks**

Change prototype pages to one style

**Week 3**

**Thursday 17th March, 2016**

**10:00 AM**

**At GS5, Library**

**Attendees: Xinchi Wang**

**Rixin Nie**

**Caixing Su**

**Kaijian Feng**

**Item 1: Team Roster Change**

Rui Zhang and Yixiang Fan left the team.

Xinchi Wang and Kaijian Feng formally joined the team

**Item 2: Deciding Software Processing Model**

Rational Unified Process(RUP)

**Item 3: Deciding Roles of Members**

Project Manager - Xinchi Wang

Business Designer - Caixing Su

System Analyst - Caixing Su, Xinchi Wang

Designer - Rixin Nie, Kaijian Feng

Implementer - Rixin Nie, Kaijian Feng

Test Designer/Tester - Kaijian Feng, Xinchi Wang

Tool Specialist – Kaijian Feng

Deployment Manager – Kaijian Feng

**Item 4: Functions List**

Made a list of functions of each web page

**Item 5: Tasks**

Sample Homepage – Rixin Nie

Development Plan – Xinchi Wang

System Request – Caixing Su

Software Requirement Specfication – Caixing Su, Xinchi Wang

**Week 4**

**Thursday 24th March, 2016**

**11:00 AM**

**At GS3, Library**

**Attendees: Xinchi Wang**

**Rixin Nie**

**Caixing Su**

**Kaijian Feng**

**Item 1: System Request**

Almost finished. Lack stakeholders part.

**Item 2: Development Plan**

Finished.

**Item 3: Software Requirement Specification**

Introduction and overall description finished.

Some critical functional requirement finished.

**Item 4: Version Controlling System**

Tried to use Dropbox to control version, but it can only keep record for 30 days.

So we decided to use Github.

**Item 5: Sample Homepage**

A prototype is made but is not satisfying. Kaijian suggests to use a frame.

**Item 6: Tasks**

Sample Homepage – Rixin Nie, Kaijian Feng

Development Plan – Xinchi Wang

System Request (Stakeholder part) – Kaijian Feng

Software Requirement Specfication – Caixing Su, Xinchi Wang

Data Dictionary & Database Design – Rixin Nie

Version Controlling System – Kaijian Feng

**Week 5**

**Tuesday 29th March, 2016**

**10:00 AM**

**At GS6, Library**

**Attendees: Xinchi Wang**

**Rixin Nie**

**Caixing Su**

**Kaijian Feng**

**Akshay Lakhanpal**

**Item 1: New Member**

Akshay Lakhanpal formally joined the team

**Item 2: System Request**

Finished

**Item 3: Development Plan**

Made some changes.

**Item 3: Software Requirement Specification**

More functional requirements finished.

Added functions about reputation and moderator after talking with customers

**Item 4: Version Controlling System**

Began to use Github to control version. Every member has an account.

**Item 5: Sample Homepage**

Decided to use Bootstrap as frame of the website. A sample page is finished.

**Item 6: Database Design**

Rixin finished designing and implementing the database using MySQL.

Imported test data to the database

**Item 7: Implementation Tool**

Html + JSP + MySQL

Struts2 + Hibernate + Spring

**Item 8: Tasks**

Sample Homepage – Rixin Nie, Kaijian Feng

Software Requirement Specification (non-functional requirements) – Caixing Su

Software Requirement Specification (functional requirements)-Xinchi Wang, Akshay Lakhanpal

Implementation Environment – Rixin Nie, Kaijian Feng

**Week 6**

**Thursday 7th April, 2016**

**10:00 AM**

**At 207, Library**

**Attendees: Xinchi Wang**

**Rixin Nie**

**Caixing Su**

**Kaijian Feng**

**(Akshay Lakhanpal didn't attend because he needed to visit visa office)**

**Item 1: Software Requirement Specification**

Non-functional requirements finished

**Item 2: Implementation Environment**

The environment is ready.

A sample system with basic functions is built.

**Item 3: Sample Pages**

Finished

**Item 4: Tasks**

Domain Model & Data Dictionary – Rixin Nie

Software Requirement Specification - Xinchi Wang, Akshay Lakhanpal

Use Case Diagram – Rixin Nie, Caixing Su

Working Diaries

**Xinchi Wang:**

**Week 2:**

Kaijian Feng and I joined group 2. We made and showed prototype webpage to other members and discussed it. We decided to unify our page style.

**Week 3:**

I was assigned the team manager. I began to create development plan and risk list. I borrowed Software Development for Small Teams: A RUP-Centric Approach from the library to guide development process.

We discussed functions on the meeting.

**Week 4:**

I finished development plan and risk list. Caixing Su and I began working on Software Requirement Specification.

**Week 5:**

Akshay Lakhanpal joined the team. Caixing Su, Akshay and I worked on Software Requirement Specification.

**Week 6:**

Software Requirement Specification is basicly finished. We associated this with user case diagram.

**Caixing Su:**

**Week1:**

I found some group members and finally form a team with four classmates. In the first meeting, we meet each other and we also introduce ourselves to each other include name, age, what subject we major and what we are good at etc. We also settled down the team role of each other based on our strength. In detail, Rui Zhang as a tester and Rixin Nie as a chief programmer and Yixiang Fan as a manager and I am the system analysts and designers.I am responsible for the document writing and specific requirement designing.

**Week2:**

In the meeting of this week, we study the stack overflow system thoroughly and we decide to write a simple function picture of each main webpage and show it to our client (teacher) in the next meeting with client. We allocate the main function of page to each other and I am responsible for writing the ‘ask a question’ pages. In the week, I finished the design of ‘ask a question’ page and some basic function requirement.

**Week3:**

In this week, I had showed the ‘ask a question’ basic function picture to the client and I mark down some functions requirement after I talked with the client. I also finished the writing of business case document.

**Week4:**

In this week, I start to write the beginning part of the FQA System Software Requirements Specification (SRS) document include the Introduction and overall description section. Besides, I also settle down some detail design of the system after discussed with my teammates. At the same time, I modify the business case document and add the ‘Stakeholder and Customer Descriptions’ section.

**Week5:**

In this week, I finished the brief design of non-functional requirement. I also discuss some details with them when I design these non-functional requirements. After I finished writing the non-functional requirement in the SRS, I share and sent it to my group member.

**Week6:**

In this week, I reformat the SRS document and update some requirements of the SRS document.

**Kaijian Feng:**

**Week1:**

I was looking for a team, and I found a member Xinchi Wang， we both need a team.

**Week2：**

This week, we two joined a team. In the meeting of this week, we show each webpage to others, and discussed it. And we decided to unify our style.

**Week3:**

In this week, at the meeting, I get my job (Implementer, Test Designer, Tool Specialist, Deployment Manager).Also I had showed the ‘main page’ basic function picture to the client and I mark down some functions requirement after I talked with the client. And then I began to build our development environment, we choose Dropbox as our working directory.

**Week4:**

In this week, we have finished a prototype, but I thought that is not suitable for our project. So I suggested to choose a frame to build our website, and at last I chose bootstrap as our frond-end frame. And because of Dropbox only save 30 days operate history, we change our project to Github.

**Week5:**

In this week, a new member Akshay Lakhanpal join us, and we introduced our project to him. We discussed Software Requirement, and we decide use jsp+mysql+java to build our project. According to requirement, I finished some simple webpages with Dreamweaver.

**Week6:**

In this week, our meeting discussed next week’s job, and I show the webpage to other member. I finished all the webpage, and I send them to Rix to complete JavaScript, database and java code in this webpage. And then I deployed our project to a webserver (use Ubuntu 14.04)

**Rixin Nie:**

**Week1:**

Found group members, Finally, we decide Rui Zhang, Yixiang Fan, Caixin Su and me 4 classmates to be a group. And decide to make a website with Java. We also decided everyone’s role, Yixiang Fan as a manager, Rui Zhang as a tester, Caixin Su as a system’s analysts, me as a coder. We discuss which function will have in our website and distribute our work before next group meeting: to draw different page out. I need to draw the administrator page.

**Week2：**

In this week, we have another two members join into our group, they are Xinchi Wang and Kaijian Feng。They also bring their page to discuss with us. We display each page have which function and discuss the detail about the pages.

**Week3:**

We decide our group members are 5 people include Yixiang Fan, Rui Zhang, Kaijian Feng, Caixin Su and me, but at the end of this week, our group members have changed, Former manager and tester Yixiang Fan and Rui Zhang decide to learn this course next autumn term.

**Week4:**

In this week, I had showed the “administrator page” to our tutor, and introduce the function in that page to our tutor. Xinchi Wang add into our group. In this week, we distribute roles again, Xinci Wang as manager, Business Designer is Caixin Su,

System Analyst is Caixin Su and Xinci Wang, Kaijian Feng and me as a designer and Implementer, Kaijian Feng and Xinci Wang as a tester too, tool specialist is Caixin Su and Kaijian Feng.

**Week5:**

In this week, We distribute some work, Kaijian Feng and me need to write HTML to write HTML pages and I finished the index.html, however, we decide do not write html by our self, we decide to use Bootstrap to build our web pages because it will be more easy and easy to control the page style. I also create the database and import the data into our database. Build the develop environment with strust2, hibernate, spring.

**Week6:**

In this week, I write a data dictionary and draw the use case, I also draw a part of domain model.

**Akshay Lakhanpal:**

**Week 4:**

I joined a group of four member and the manager introduce me with rest of the team member. And planned for a next meeting that’s on week 5.

**Week 5:**

The manager introduced me with basic of Github and how to operate it. I owned a responsibility of system administrator for functional requirements. Before next week of Monday I submitted the task to the manager.

**Week 6:**

The manager gave me responsibility of reporting subsystem regarding functional requirement. Before next week of Monday I submitted the task to the manager. After that Caixing Su did little changes in my work.

FQA System Development Contribution Assessment

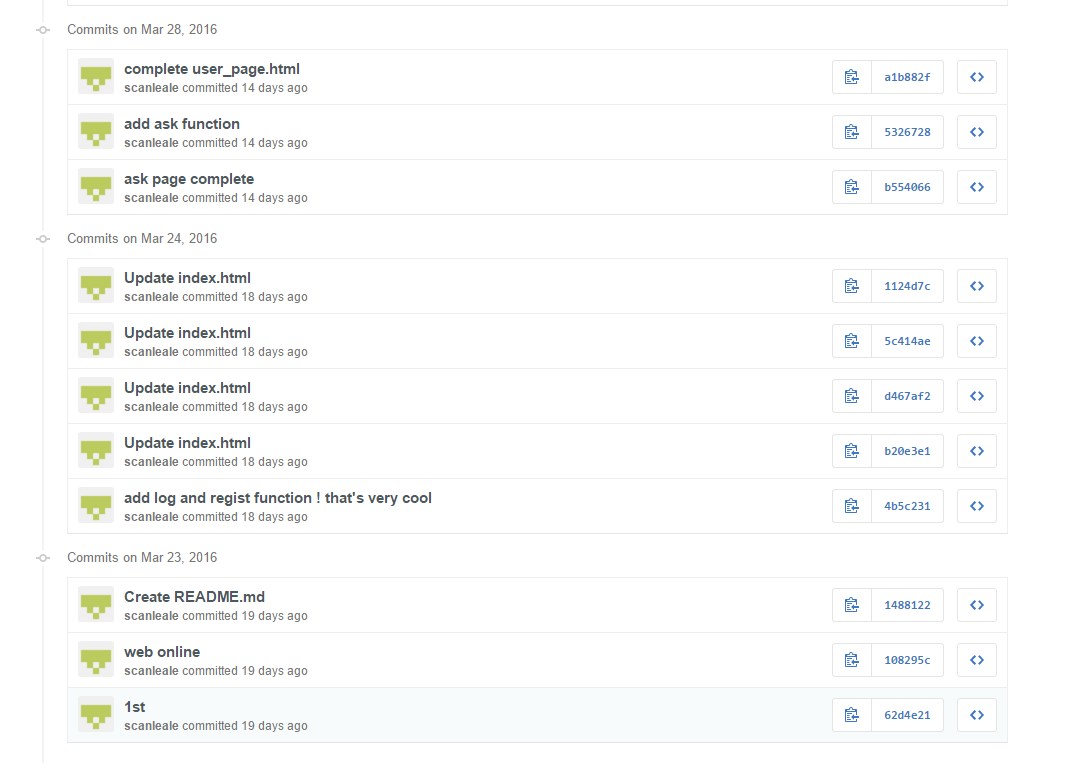
|  |  |
| --- | --- |
| **Name** | **Contribution** |
| Xinchi Wang | Contributed |
| Rixin Nie | Contributed |
| Caixing Su | Contributed |
| Kaijian Feng | Contributed |
| Akshay Lakhanpal | Some Contribution\* |

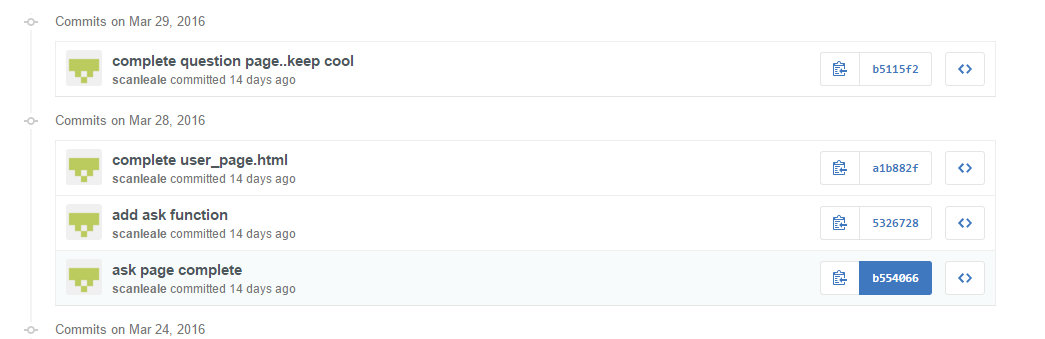
**\***Akshay joined the team in week 4.

Version Controlling Records

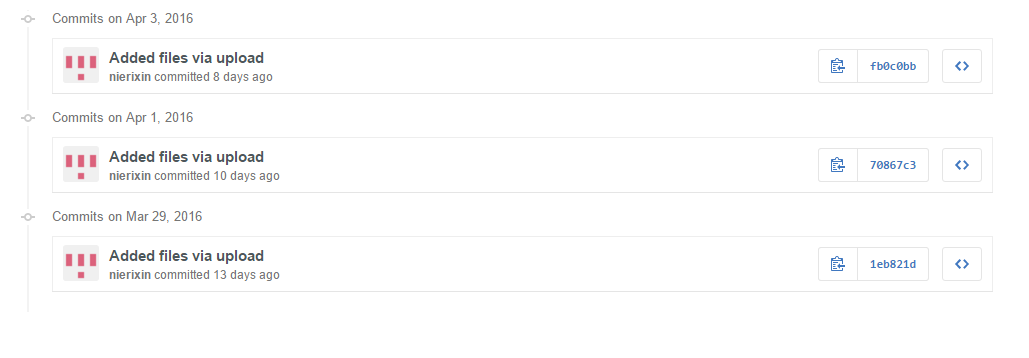
We use github as our version control tool. Some pictures can prove our effect.

1. This is web frond-end design period version control.





1. This is our background program version control.



We upload this code as a war file, so we can deploy it on server quickly.