<FQA System>

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 19/03/2016 | 1.0 | First SRS | Xinchi Wang  Caixing Su |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Definitions, Acronyms and Abbreviations 3

1.4 References 3

1.5 Overview 3

2. Overall Description 3

2.1 Product Perspective 3

2.1.1 System Interfaces 3

2.1.2 User Interfaces 4

2.1.3 Hardware Interfaces 5

2.1.4 Software Interfaces 6

2.1.4.1 User Interfaces 7

2.1.5 Comunication Interface 6

2.1.6 Memory Constraints 7

2.1.7 Operations 8

2.2 Product Functions 4

2.3 User Characteristics 5

2.4 Constraints 6

2.5 Assumptions and Dependences 7

3. Specific Requirements 3

3.1 Functionality 3

3.1.1 <Functional Requirement One> 3

3.2 Usability 3

3.2.1 <Usability Requirement One> 3

3.3 Reliability 3

3.3.1 <Reliability Requirement One> 3

3.4 Performance 3

3.4.1 <Performance Requirement One> 3

3.5 Supportability 3

3.5.1 <Supportability Requirement One> 3

3.6 Design Constraints 3

3.6.1 <Design Constraint One> 3

3.7 Online User Documentation and Help System Requirements 3

3.8 Purchased Components 3

3.9 Interfaces 3

3.9.1 User Interfaces 3

3.9.2 Hardware Interfaces 3

3.9.3 Software Interfaces 3

3.9.4 Communications Interfaces 3

3.10 Licensing Requirements 3

3.11 Legal, Copyright and Other Notices 3

3.12 Applicable Standards 3

4. Supporting Information 3

# 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 shows some questions which user may interest in and other useful links
* 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
* Administrator Subsystem allow administrator to administrate the system

## Definitions, Acronyms and Abbreviations

Temporarily unavailable

## 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.

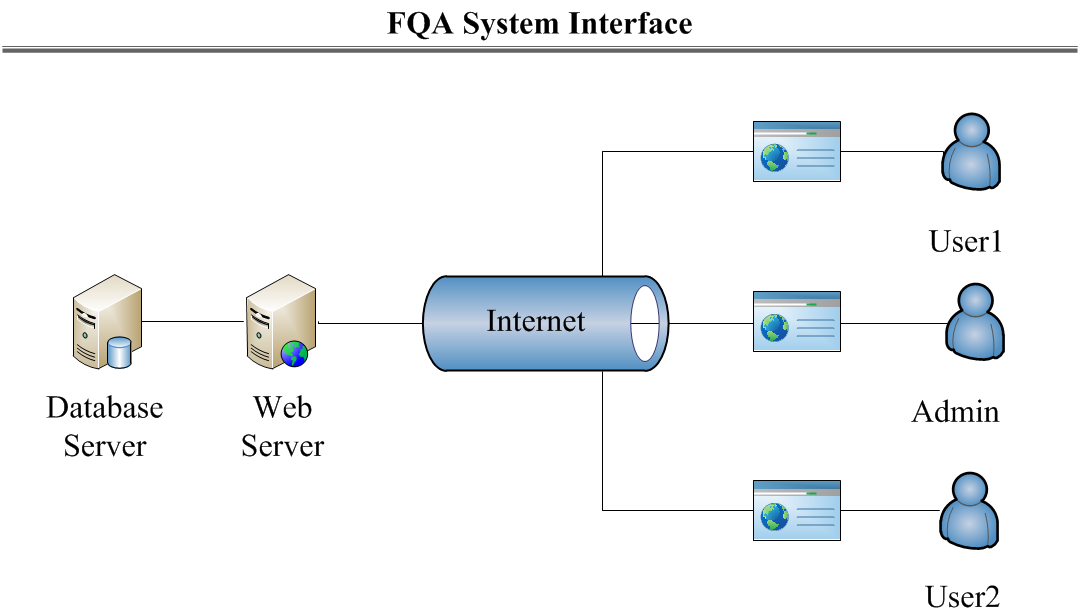
# Overall Description

## 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

## User Characteristics

The users of FQA system include system administrator and user.

* 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.

## 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

[This section of the **SRS** should contain all the software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements.   When using use-case modeling, these requirements are captured in the Use-Cases and the applicable supplementary specifications. If use-case modeling is not used, the outline for supplementary specifications may be inserted directly into this section, as shown below.]

## Functionality

[This section describes the functional requirements of the system for those requirements which are expressed in the natural language style. For many applications, this may constitute the bulk of the **SRS** Package and thought should be given to the organization of this section. This section is typically organized by feature, but alternative organization methods may also be appropriate, for example, organization by user or organization by subsystem. Functional requirements may include feature sets, capabilities, and security.

Where application development tools, such as requirements tools, modeling tools, etc., are employed to capture the functionality, this section document will refer to the availability of that data, indicating the location and name of the tool that is used to capture the data.]

### <Functional Requirement One>

[The requirement description.]

## Usability

[This section should include all of those requirements that affect usability. For example,

• specify the required training time for a normal users and a power user to become productive at particular operations

• specify measurable task times for typical tasks or base the new system’s usability requirements on other systems that the users know and like

• specify requirement to conform to common usability standards, such as IBM’s CUA standards Microsoft’s GUI standards]

### <Usability Requirement One>

[The requirement description goes here.]

## Reliability

[Requirements for reliability of the system should be specified here. Some suggestions follow:

• Availability—specify the percentage of time available ( xx.xx%), hours of use, maintenance access, degraded mode operations, etc.

• Mean Time Between Failures (MTBF) — this is usually specified in hours, but it could also be specified in terms of days, months or years.

• Mean Time To Repair (MTTR)—how long is the system allowed to be out of operation after it has failed?

• Accuracy—specify precision (resolution) and accuracy (by some known standard) that is required in the system’s output.

• Maximum Bugs or Defect Rate—usually expressed in terms of bugs per thousand of lines of code (bugs/KLOC) or bugs per function-point( bugs/function-point).

• Bugs or Defect Rate—categorized in terms of minor, significant, and critical bugs: the requirement(s) must define what is meant by a “critical” bug; for example, complete loss of data or a complete inability to use certain parts of the system’s functionality.]

### <Reliability Requirement One>

[The requirement description.]

## Performance

[The system’s performance characteristics should be outlined in this section. Include specific response times. Where applicable, reference related Use Cases by name.

• response time for a transaction (average, maximum)

• throughput, for example, transactions per second

• capacity, for example, the number of customers or transactions the system can accommodate

• degradation modes (what is the acceptable mode of operation when the system has been degraded in some manner)

• resource utilization, such as memory, disk, communications, etc.

### <Performance Requirement One>

[The requirement description goes here.]

## Supportability

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

### <Supportability Requirement One>

[The requirement description goes here.]

## Design Constraints

[This section should indicate any design constraints on the system being built. Design constraints represent design decisions that have been mandated and must be adhered to. Examples include software languages, software process requirements, prescribed use of developmental tools, architectural and design constraints, purchased components, class libraries, etc.]

### <Design Constraint One>

[The requirement description goes here.]

## On-line User Documentation and Help System Requirements

[Describes the requirements, if any, for on-line user documentation, help systems, help about notices, etc.]

## Purchased Components

[This section describes any purchased components to be used with the system, any applicable licensing or usage restrictions, and any associated compatibility and interoperability or interface standards.]

## Interfaces

[This section defines the interfaces that must be supported by the application. It should contain adequate specificity, protocols, ports and logical addresses, etc. so that the software can be developed and verified against the interface requirements.]

### User Interfaces

[Describe the user interfaces that are to be implemented by the software.]

### Hardware Interfaces

[This section defines any hardware interfaces that are to be supported by the software, including logical structure, physical addresses, expected behavior, etc. ]

### Software Interfaces

[This section describes software interfaces to other components of the software system. These may be purchased components, components reused from another application or components being developed for subsystems outside of the scope of this **SRS** but with which this software application must interact.]

### Communications Interfaces

[Describe any communications interfaces to other systems or devices such as local area networks, remote serial devices, etc.]

## Licensing Requirements

[Defines any licensing enforcement requirements or other usage restriction requirements that are to be exhibited by the software.]

## Legal, Copyright, and Other Notices

[This section describes any necessary legal disclaimers, warranties, copyright notices, patent notice, wordmark, trademark, or logo compliance issues for the software.]

## Applicable Standards

[This section describes by reference any applicable standard and the specific sections of any such standards which apply to the system being described. For example, this could include legal, quality and regulatory standards, industry standards for usability, interoperability, internationalization, operating system compliance, etc.]

# Supporting Information

[The supporting information makes the **SRS** easier to use. It includes:

• Table of contents

• Index

• Appendices

These may include use-case storyboards or user-interface prototypes. When appendices are included, the **SRS** should explicitly state whether or not the appendices are to be considered part of the requirements.]