A2 - System Requirements Specification

for

Crowd Quizmaker

Version 1.0

Prepared by 0N3 N16H7 PR0J3C7

Suwat Inkaew 610610521
Kritsanaphong Tepweerakul 630610714
Kitpisan Tanngan 630610716
Chayanon Pitak 630610724
Nadtaphong Jandaboot 630610743
Woranut Kitchakan 630610760

https://github.com/ChayanonPitak/261361-Project/

Revision History

Name	Date	Reason for changes	Version
Chayanon	5 Jan 2023	Initial draft	0.1
Chayanon	14 Jan 2023	Complete Purpose and Scope for 1.0	0.2
Chayanon	15 Jan 2023	Complete Product Overview for 1.0 and Reference	0.3
Nadtaphong	15 Jan 2023	Complete Specification Requirement for 1.0	0.4
Kitpisan	15 Jan 2023	Verification on section 4.1-4.2	0.5
Suwat	15 Jan 2023	Verification on remaining section	0.6
Chayanon, Kritsanaphong, Nadtaphong	15 Jan 2023	Complete Appendix Section	1.0
Chayanon	6 Feb 2023	Specific requirements on Functions Requirements (3.4)	1.1

Table of Contents

1 Introduction

1.1 Purpose

Lecturer team of 261111 - Internet and Online Community, Chiang Mai University have created this course to create realize in technology, internet and online community and proper behaviour of using it. Originally, the course itself are using typical way of evaluate the understanding - assignments and exams which is one-way activities. The lecturer team have decided to create a new way of evaluate the understanding of the students by creating a quiz application that students also contribute the quiz creation from the knowledge they have learned from the course and try to answer and review the contributed quizzes. The application will be called Crowd Quizmaker.

1.2 Scope

Our software (Crowd Quizmaker) will be a cross-platform (Windows, MacOS, iOS, Android) web application that lecturer (or quiz admin) can create a quiz topic and students (or anyone - as quiz contributor) can contribute to a quiz - create, answer and review. Any users can be quizzes admin and/or quiz contributor and quizzes topic doesn't need to be academic purpose. Quizzes admin can specify what can quizzes contributor do.

1.3 Product Overview

1.3.1 Product Perspective

This product is a web application that can be accessed from any devices that have internet connection. The application will be a cross-platform (Windows, MacOS, iOS, Android) web application. The application is a quiz platform that any one can contribute to quizzes. The quizzes can be academic or non-academic.

1.3.2 Product Functions

The application consits of two part - Central system which provide the API, quizzes database storage and quizzes manager. And user client that allow any users to create quiz topic, manage quiz topic - moderate, manage, view and evaluate quiz content, contribute to quizzes - create, answer and review, and add support to Canvas LMS.

1.3.3 User Characteristics

The user classes of this systems will be:

1. Administrator: Lecturer team of 261111 - Internet and Online Community, Chiang Mai University or any one who have the permission to manage the system. Responsible for manage, support and operate whole systems - both central system and clients.

2. Course Lecturer: Any course lecturer who wish to use this system. Responsible for manage the quizzes topic, moderate quizzes content, evaluate quiz topic and/or Canvas LMS intregration.

3. Course student: Student from courses. Responsible for Canvas LMS connection and contribute to quizzes from courses - create answer and review.

4. Quiz Admin: Anyone who want to create quizzes topic. Responsible for create/manage the quizzes topic, moderate quizzes content and evaluate quizz topic.

5. Quiz Contributor: Anyone who want to contribute to quiz topic. Responsible for create, answer and review quizzes from quiz topics.

1.3.4 Limitations

Currently we do not have any limitations from the requirements of stakeholders, but it may including but not limited to hardware performance limitstions, storage limitations, internet connection limitations, specific course content, local regulations and etc. We will update this section if we have any exact limitations to update.

1.4 Definitions

- **Central system or server**: The API provider, quizzes database storage and quizzes manager. Only accessible by administrator.
- **Client**: The user interface that allow any users to interact with our systems - Manage and moderate quiz topics and contribute to quizzes.
- **Quiz**: A set of questions and answers that can be answered by users.
- **Quiz Topic**: A topic that contain quizzes.
- **Quiz Admin**: An user that can create/manage quiz topic, moderate quizzes content and evaluate quiz topic.
- **Quiz Contributor**: An user that can contribute to quiz topic - create, answer and review quizzes.
- **Course lecturer or lecturer**: A course lecturer which have same permissions as Quiz Admin.
- **Course student or student**: A student which have same permissions as Quiz Contributor.

2 References

ISO/IEC/IEEE 29148:2018
https://ieeexplore.ieee.org/document/8559686
Systems and software engineering – System and software requirements engineering – System requirements specification

CPE111 - Internet and Online Community facebook pages
https://www.facebook.com/CPE111/

Canvas Learning Management System
https://www.instructure.com/en-au/canvas/

3 Specific Requirements

3.1 External Interface

3.1.1 System Interfaces

The application will retrieve the registration information for the required course from "canvas.cmu.ac.th" in the form of JSON. This will provide the registration information for the application.

3.1.2 User Interfaces

The user interface will be simple to understand for all the user that uses this application. We will design a new user interface for our application using our experience and knowledge that doesn't consist of any standard. The reason is to eliminate the difficulty of the user's understanding of our application interface.

3.1.3 Hardware Interfaces

No extra hardware interfaces are needed.

3.1.4 Software Interfaces

The application will use the GraphQL API from Canvas.

3.1.5 Communications Interfaces

The application will use the HTTP protocol to communicate with the client. So, an internet connection is required.

3.2 Functions

The system shall

1. **Let user create quiz topic** which contains topic name, tags(optional), contributors(optional), format(multiple choices and/or text area), and distribution(question only, answer only or both).
2. **Let quiz topic admin manage quiz topic** on topic name, contributors, format, and distribution and delete whole quiz topic.
3. **Let quiz topic admin moderate quizzes content** on contributors' question, answer and review.
4. **Let contrubutor create question to the quiz topic** based on quiz topic distribution and format. Contributor shall define the question, tag(depends on admin), answer(s) and correct answer(s).
5. **Let contrubutor attempt to answer and review the question to the quiz topic** based on quiz topic distribution. System shall random the question based on distribution settings. User shall answer the question and review the question - rate, comment and/or report. (only quiz topic admin can see the review)
6. **Evaluate the user's quiz attempt** which visualize the result of the quiz attempt. The result contains correct answer count on that attempt.
7. **Let quiz topic admin evaluate the quiz topic** which visualize the result of the quiz attempt. The result contains correct answer count on questions and questions' review.

3.3 Usability Requirements

The system shall be able to operate normally without any errors or server issues. Also, it shall be simple to use for every user.

3.4 Performance Requirements

- **Respond Time**: The system shall have a response time of fewer than 10 seconds for topic-related requests, except for adding contributors shall not have a responce time greater than 30 seconds.
- **Throughput**: The system shall be able to handle 500 simultaneous users and shall be able to handle at least 100 requests per second.
- **Memory usage**: The system shall use no more than 1 GB of memory under normal conditions.
- **CPU usage**: The system shall require atleast 4 core of CPU under normal conditions.
- **Storage**: The system shall be able to store up to atleast 1 TB of data.
- **Avalibility**: The system shall always be available to the user all the time.

3.5 Logical Database Requirements

1. The database shall be able to store user information and quiz data.
2. The database shall always be able to provide data for the server most of the time.
3. The database shall be able to ensure that each role of user can access their interface by their role.

3.6 Design Constraints

- **Time**: This application must be complete within 2-3 weeks.
- **Resource**: This application project team has 6 members.
- **Scalability**: The system must be design to support a minimum of 500 simultaneous users.
- **Deployment**: The system must be deployable to server-based and client-based infrastructure.

3.7 Software System Attributes

- **Reliability**: The system shall be able to perform an intended function without unexpected errors.
- **Maintainability**: The system shall be easy to maintain and update.
- **Portability**: The system shall be able to run on multiple operating system, including Window, Mac, Android and iOS.

3.8 Supporting Information

1. The application shall be able to run on every modern browsers.
2. The application shall not send any personal data of the user to the internet or an unknown source.

4 Verification

4.1 External Interface

4.1.1 System Interfaces

- **Fetch from canvas**: After retrieving the course information, we will check the number of people on the course that are the same number people of information we have retrieved.
- **Import manually**: After imported manually, we will check that people are imported are really in this course.

4.1.2 User Interfaces

We will be using "User Acceptance Testing" in user interface by taking users to use the interface to check the simplest of the website.

4.1.3 Software Interfaces

We will test by checking whether the information is correct or not using "System Testing".

4.1.4 Communications Interfaces

We won't test because HTTP protocal are standard system and already Verificate.

4.2 Functions

We will be using "User Acceptance Testing" by finding the tester to test the function that is working properly or not.

4.3 Usability Requirements

Tester will test the app and send feedback about user experience.

4.4 Performance Requirements

- **Respond Time**: Tester will test by reckon system's response time.
- **Throughput**: Tester will test by sending requests to system.
- **Memory usage**: Tester will test by performance monitoring while running application.
- **CPU usage**: Tester will test by performance monitoring while running application.
- **Storage**: Tester will test by performance monitoring while running application.
- **Avalibility**: Tester will test by uptime monitoring.

4.5 Logical Database Requirements

1. Tester will test by send requests about user information and quiz data.
2. Tester will test by uptime monitoring about data.
3. Tester will test by access each rolls and test.

4.6 Design Constraints

- **Time**: Verification by scrum meeting, github Kanbanboard to check progress on implementation.
- **Resource**: Have a verification on skills and progress done by each members.
- **Scalability**: Verification by sending requests to system.
- **Deployment**: Verification by show connection between server and client.

4.7 Software System Attributes

- **Reliability**: Verification by test at most as possibe expected errors.
- **Maintainability**: Verification by change some part and it will done by few steps and don't have any causes to another parts.
- **Portability**: Verification by testing application on multiple operating system.

4.8 Supporting Information

1. Verification by testing application on multiple modern browsers.
2. Verification by penetration test.

5 Appendix

5.1 Acronyms and Abbreviations

- **API**: Application Programming Interface
- **LMS**: Learning Management System
- **GB**: Gigabyte
- **HTTP**: Hypertext Transfer Protocol
- **TB**: Terabyte

5.2 Project Plan

At the first sprint (about three weeks - Until 16th Feburary 2023), our product should be able to authenticate users, create quizz topics, manage and moderate quiz topics on question format, content moderation, quiz distribution. do some quiz contributions - create, answer questions, Evaluate quiz topics on some content - correctness statistics.

At the second sprint (about three weeks - Until 16th Feburary 2023) additional from thr first sprint, our product should be able to moderate quiz topics on subtopics, tag, contribution period, complete quiz contribution and distribution setup, do additional quiz contributions - review and report. Evaluate quiz topics on additional content - correctness statistics on each subtopics, tags. Integration with Canvas LMS.

Only include until 1.0

5.3 A1 - responsibility

- **Suwat Inkaew 610610521** (17.5%) - Verification section(17.5%)
- **Kritsanaphong Tepweerakul 630610714** (17.5%) - Consult with stakeholder(s) about project (Prof. Sakgasit Ramingwong / 23 Dec 2022), Provide supporting information for SRS(10%), Appendix section (Project Plan)
- **Kitpisan Tanngan 630610716** (12.5%) - Verification section(12.5%)
- **Chayanon Pitak 630610724** (22.5%) - Docuent setup, Consult with stakeholder(s) about project (Prof. Sakgasit Ramingwong / 23 Dec 2022), Miscellinious consult with stakeholder(s), Provide supporting information for SRS(5%), Introduction section, Appendix section (Project Plan)
- **Nadtaphong Jandaboot 630610743** (25%) - Specific Requirements section, Appendix section (Acronyms and Abbreviations)
- **Woranut Kitchakan 630610760** (10%) - Provide supporting information for SRS (10%)

5.4 A2 - Responsibility percentage calculation

Activities that not directly contribute to the document

- Consult with stakeholder(s) 10%(Divided) each sessions
- Miscellinious consult with stakeholder(s) 0%
- Provide supporting information for SRS (not to be confused with the supporting information section) 25%(Divided)

Activities that is directly contribute to the document

- Document setup 0%
- Introduction section 10%
- Reference section 0%
- Specific Requirements section 30%(Divided)
- Verification section 30%
- Appendix section (Acronyms and Abbreviations) 0%(Divided)
- Appendix section (Project Plan) 5%(Divided)