Software Requirements Specification
for Quiz Maker

Prepared by

Daniel Garcia,

and Pashang Engineer

Version 2.7 approved

Last revised on 12/08/2020

Table of Contents

Revision History

Name	**Date**	**Changes**	**Version**
DG & MR	9/08/20	Created document, defined purpose and scope	1
DG	9/09/20	Additions to References, Product Perspective, System Interface, User Interface, Software Interface, User Characteristics	1.1
HB	9/10/20	Ensured consistent formatting and that all empty sections were replaced with "TBD"	1.2
HB	9/14/20	Standardized formatting styles and updated Table of Contents	1.3
PE, & DG	9/16/20	Completed Section 3, drafted sections 4 and 5	1.4
HB	9/29/20	Reviewed formatting, linked LibertyBikes_Bing Google Slides	1.5
HB, DG, PE	9/30/20	Detailed Portability section and converted necessary bulleted lists to numbered lists to improve referencing.	1.6
DG, PE, HB	10/01/20	Detailed Security section and made it sound more formal and added activity diagrams. Added UML Class Diagram for Quiz Maker	1.7
DG, PE	10/06/20	Added UML Activity Diagram for Creating a Quiz Updated Functional, Security in Section 3.6 Updated Scenarios in Section 3.6 Added Solution Oriented Requirements Updated Section 2.1.2	1.8
DG, PE, HB	10/08/20	Updated Sections 1.3 and 3.5.3 Updated Instructor Goals in section 3.6 Updated Functional Requirements	1.9
DG, HB	10/11/20	Updated Sections 1.1,1.2,1.3, and 2.1.2 Added UML Activity Diagram for Taking a Quiz	2.0
PE, DG	10/13/20	Updated sections 1.3 and 2.1.2	2.1

		Updated functions in 3.2 Updated Quality and Functional Requirements in sections 3.6	
PE, DG	10/17/20	Updated 1.3 Updated System Context in 3.1.1 Updated Functions in 3.2 Updated Logical Database Requirements in section 3.3 Updated Design Constraints in Section 3.4 Updated Security Requirements in Section 3.5 Updated Portability Requirements in Section 3.5.4 Updated Learner and Instructor Goals in Section 3.6 Updated Constraints and Solution Oriented Requirements in Section 3.6	2.2
DG	10/20/20	Updated Section 3.1.1 Updated Design Constraints in Section 3.4 Updated Scenarios and Constraints in Section 3.6	2.3
PE	10/22/20	Updated Section 3.1 Updated Scenarios to include insertion of CSV file	2.4
DG	11/5/20	Updated Functions in Sections 2.2 and 3.2	2.5
DG	12/2/20	Updated Appendix in Section 5 to include MIT License	2.6
DG, PE	12/08/20	Revised Sections 2.1.2,3.1, and 5.1	2.7

1. Introduction

1.1. Purpose

The purpose of Quiz Maker is to have students take the initiative in the learning process, while making sure the knowledge they learned is preserved after each semester and easily accessible to users in a given course. The intended audience of Quiz Maker is students, faculty and staff at SUNY Oswego.

1.2. Scope

Quiz Maker will allow students to create their own quizzes, take quizzes, and review quizzes. Quizzes and questions will be stored in a central database. This is to ensure that quizzes and questions are preserved once the semester ends. Storing quizzes and questions on a database will allow these items to be reused when a user creates a new quiz. Teachers on Quiz Maker will be able to add new courses. After creating a course, teachers will have the ability to manage their courses as well. Teachers will also have the ability to add and remove students from a given course. Users in the instructor role also have the ability to promote Quiz Maker will not check for intellectual integrity. It is up to the student to uphold that agreement when using Quiz Maker.

1.3. Definitions, Acronyms and Abbreviations

- **Administrator:** An person with the ability to control courses
- **Exam:** An instructor-generated quiz consisting of learner-generated questions.
- **GUI:** A graphical user interface
- **Instructor:** Any user that's a professor or a Teaching Assistant
- **Learner:** Any user that's been assigned to a Quiz Maker course roster by the course's instructor
- **Microservice:** An individual component of an application with its own database.
- **JWT:** JSON Web Token
- **Header:** The algorithm used to encode and decode the JWT
- **Payload:** User information that is stored in the JWT

- **Signature:** Data used to verify if the JWT has not been changed by the user
- **Course Database:** A back end structure used to store course information
- **Quiz Database:** A back end structure used to store quiz information

1.4. References

1. OpenLiberty/liberty-bikes: Real-time web based multiplayer game running on OpenLiberty
2. Building a simple RESTful Java microservice using JAX-RS and JSON-B
3. MicroProfile Rest Client and JSON-B make working with REST APIs easier
4. JSON Web Tokens - jwt.io
5. LibertyBikes_Bing Google Slides

1.5. Overview

Developers will look to section 3 for implementation while keeping the information in section 2 in mind.

2. The Overall Description

Quiz maker will run on a server created by the Open Liberty software that is provided by IBM. Quiz maker will involve learners, which consist of students, and instructors, who are professors and TAs. Instructors will be able to create and manage courses and instructors can be learners by taking other quizzes or courses. Learners can be added to an instructor's course and be able to create quizzes, take quizzes, review their quizzes and rate quizzes. Learners will be students by default and instructors will switch to a learner role when they're taking quizzes of other courses. Instructors will also be in an instructor role and students will be in a learner role in QuizMaker by default.

2.1. Product Perspective

Quiz Maker, using Eclipse Microprofile, will operate using a microservice architecture, where each component of Quiz Maker, such as the GUI, will become an individual application, with its own database. Quiz Maker requires a functional server, which is created using Open Liberty, to operate. Quiz Maker will rely on the capabilities of Open Liberty and Eclipse Microprofile. Therefore, they're both components of the system.

2.1.1. System Interfaces

To achieve this, the use of the REST API will be necessary to communicate with different microservices using HTTP calls. Also, the use of JAX-RS will be needed to assist in the deployment of web services. For authentication, the use of JSON-B will be needed.

2.1.2. User Interfaces

Learner

When the user successfully logs into Quiz Maker, they will go to the home page. The home page will consist of the color scheme of SUNY Oswego. On the home page, three of the most recent courses will be displayed. This section shall be named, "My Courses". The home page will also display 'My Recently Created Quizzes'.

After a learner clicks on a particular course, a list of top rated quizzes in that course will be displayed with a full list of quizzes for that particular course will be displayed below. Once we go to the Quizzes section from the navigation bar, we will see 'My Top Rated Quizzes' and in the list of quizzes section, there will be two sections 'My Created Quizzes' and 'Quizzes I Took'. In the 'My Created Quizzes' section, attributes like the name of the quiz, author, topic, number of quiz questions, date and rating will be visible to the user and in the 'Quizzes I Took' section, there will be the same attributes as specified in the 'My Created Quizzes' section.

While taking the quiz, the learner will be able to see a progress bar located at the top of the screen. This progress bar will fill as the user answers the quiz questions. While taking a quiz, a thumbs up icon along with a thumbs down icon. A learner can click on the thumbs up icon to upvote a question. A learner can also click the thumbs down icon to downvote a question. A flag icon shall be displayed for every question. Upon clicking on the flag icon, the learner shall have the ability to report a quiz with multiple checklist options like Spelling Mistake, Wrong Question and Wrong Answer and a text box below specifying 'If other please specify' which allows the user to mention their problem.

After the learner finishes taking the quiz, they will go to a post quiz screen. This screen will display a score representing the amount of questions the user answered correctly during the quiz. Additionally, the post quiz screen will display which questions were answered correctly and which questions were answered incorrectly and also a back to course button is displayed that takes you back to your course page.

Finally, learners while creating a quiz can review their quizzes after creating theirs and can even edit their quiz questions while on the review page. They can even go back to a quiz question while creating a quiz if they would like to.

Instructor

As an instructor your course page will be different from what a learner sees. The learner sees ‘Courses where I am a student’ whereas the instructor sees ‘Courses where I am a professor’ and a list of courses are displayed (3 courses). The user interface shall allow instructors to add courses or delete courses. The user interface shall allow instructors to add or drop students from a course. The user interface shall also allow instructors to import quizzes from previous quizzes created by the instructor or learner of their course.

The instructor will also be able to see a option called Create Course which allows the instructor to create a course using a csv file.

Both Learners and Instructors

The user interface will allow the learners and instructors to create or take a quiz. While creating quizzes both learners and instructors will see the home page 'My Courses' and ‘My Recently Created Quizzes’. Users will have the ability to switch to, “Dark Mode” easily as the icon will be visible on the navigation bar. Dark Mode shall consist of predominantly grey with around 85-90% opacity. For Learners, the top navigation bar will allow the user to see options like Home, Create, Quizzes and Courses whereas for instructors the top navigation bar will also have an added button called Create Course . The user interface will allow instructors or learners to report a quiz question.

2.1.3. Software Interfaces

The development of Quiz Maker will involve the use of the REST API, JAX-RS, which is done using Java EE, and JSON-B.

2.1.4. Memory Constraints

Since Quiz Maker will run on a web browser, the memory will be constrained by the user's system.

2.1.5. Operations

Learners will be able to create quizzes and take their own quizzes. Learners will also be able to upvote or downvote quizzes. Instructors will be able to edit their class roster, and manage their courses. Instructors will also be able to create tests from learner generated questions

2.2. Product Functions

- **authenticate()**: Quiz Maker will use JWT to authenticate a user
- **addLearner()**: The instructor will add a learner to their class roster
- **deleteLearner()**: The instructor will delete a learner from their class roster
- **filterTopic()**: List of quizzes will be filtered by a desired topic
- **createQuiz()**: A learner will be able to create a quiz
- **deleteQuiz()**: The instructor will delete a quiz created by learners
- **takeQuiz():** Allows a learner to take a quiz
- **reportQuestion():** Allows a learner to report a quiz question
- **addCourse()**: The instructor will add a course
- **deleteCourse()** The instructor will delete a course
- **normalize():** Determines the rating for a given quiz

2.3. User Characteristics

The intended users of Quiz Maker are SUNY Oswego students and faculty. A user with any amount of technical expertise can successfully use Quiz Maker.

2.4. Constraints

Quizzes on Quiz Maker must only contain multiple choice questions. Multiple choice questions shall consist of 5 possible answers. After a user submits their answer, Quiz Maker must mark the correct answer. Users must be allowed to upvote or downvote a quiz. Users will be able to report a quiz question. Instructors on Quiz Maker must be able to save their created quizzes. Quiz Maker must not have any intellectual integrity checks. Quiz Maker must use JSON web tokens for authentication. Quiz Maker must use the React library for front end development. Quiz Maker must use MongoDB for back end development. Quiz Maker shall also have an admin. This admin shall be allowed to elevate learners to instructors. Admins shall also be allowed to delete any course.

2.5. Assumptions and Dependencies

The current architecture of designating a learner or instructor is being made on the assumption that an administrator will have access privileges to designate or ascertain whether the learner or instructor are in their right roles or elevate access privileges to the said learner with instructor privileges after certain information is confirmed with the administrator.

3. Specific Requirements

3.1. External Interfaces

Home Page — Learner:

The home page will display 'My Courses' and 'My Recently Created Quizzes'.

Input: Learner clicks on a course listed on "My Courses"

Output: Learner is taken to the homepage of the selected course

Input: Learner clicks on "Create"

Output: Learner is taken to the "Create a Quiz" page

Input: Learner clicks on a quiz that they have created

Output: The learner is taken to the first question of the selected quiz

Create a Quiz Page:

Input: Learner clicks on a given course,

Output: Learner is taken to a page prompting them to add topics for the quiz

Input: Learner clicks on a topic

Output: Is taken to a page offering the learner a few more topics to base their quiz on.

Input: Learner clicks on "Go to Quiz Creation"

Output: Is taken to the Quiz page, with text fields for the question along with text fields for the 5 possible answers.

Creating a Quiz:

Input: Learner clicks on the "Add Question" button in the top left corner page

Output: The next question will be created. The learner will be taken to that created question

Input: Learner clicks on a back arrow.

Output: The learner will be taken to the previous question

Input: Learner clicks on "Add Topic"

Output: 5 additional topic tabs will display next to "Add Topic"

Input: Learner clicks on "Add Topic" a second time

Output: The list of additional topics will disappear

Input: Learner clicks on one of the additional topic tabs

Output: The selected topic will be added to the list of previously selected topics

Input: Learner clicks on "Review Quiz"

Output: The learner is taken to the 'Review Quiz' screen

Input: Learner clicks on "Delete Quiz"

Output: A pop will display, confirming the user if they want to delete the quiz

Input: Instructor clicks on 'Import Question'

Output: A pop up will display, showing a list of favorited quizzes

Input: Instructor clicks on a favorited quiz while importing a question

Output: The questions for the selected quiz will be displayed

Input: Instructor clicks on ‘Use Questions’ while importing a question

Output: The text for the imported question will be copied and pasted onto the current quiz question

Review Quiz Page:

Input: User clicks on ‘Publish Quiz’

Output: User will be taken back to the home page.

Input: Learner clicks on ‘Delete quiz’

Output: A pop will display, confirming the user if they want to delete the quiz

Input: User clicks on ‘Edit Question’

Output: Text in the question will be editable

Input: User clicks on ‘Cancel’ while editing a question

Output: Recent changes to a question will not be saved. The user will return to the original ‘Review Quiz’ screen.

Input: User clicks on ‘Save Changes’ while editing a question

Output: User will return to the original ‘Review Quiz’ screen. The recent changes to the quiz will appear..

Taking a Quiz:

Input: The learner clicks on one of the 5 possible answers

Output: Quiz Maker will mark the correct answer, along with the answer that the learner submitted

Input: Learner clicks on the arrow to the right of the screen

Output: Learner is taken to the next question

Input: Learner clicks on the arrow to the left of the screen

Output: The learner is taken to the previous question

Input: Learner clicks on the flag icon

Output: A pop-up appears, along with 4 possible tabs that the learner can click on, and a text box

Input: Learner clicks on “Submit”

Output: Text will appear, thanking the user for reporting the question

Input: Leaner clicks the “X” icon to the top right of the report question pop-up

Output: The report question pop-up disappears

Input: The learner clicks on the “X” icon on the top right of the Quiz page

Output: The learner is taken back to the course home page

Home Page — Instructor:

The home page will be ‘My Courses’ .

Input: Instructor clicks on a course listed on “My Courses”

Output: Instructor is taken to the homepage of the selected course.

Input: Instructor decides to “Create a Quiz” based on the selected course

Output: Instructor is taken to the “Create a Quiz” page,

Input: Instructor clicks on “Courses” on top of the home page

Output: Instructor is taken to the “Courses” webpage, listing all of the courses that the instructor is enrolled in

Input: Instructor clicks on “Quizzes” on the top of the page

Output: Instructor is taken to the “Quizzes” webpage

Input: Instructor clicks on “Upload Roster”

Output: Instructor is taken to the “Upload Roster” webpage

Upload Roster — Instructor:

Input: Instructor clicks on “Drop CSV file here or click to upload”

Output: A pop-up from the Instructor’s file app on their operating system will appear

Input: The instructor clicks on “Add Roster”

Output: A pop-up notifying the instructor that the course was created

3.1.1. System Context

- **Subject Facets: Information Quiz Maker needs to know to sufficiently operate**
    a. User login credentials
    b. Created questions
    c. Created answers
    d. Quiz ID
    e. Course ID
    f. Learners taking a given course
    g. Instructors teaching a given course
- **Usage Facets: Who interacts with Quiz Maker**
    a. SUNY Oswego students
    b. SUNY Oswego faculty
    c. Web browser
- **IT System Facets: Factors that influence how Quiz Maker is implemented**
    a. Open Liberty web server
    b. React API
    c. MongoDB
    d. JSON web tokens
    e. Google API
    f. Microservice architecture
- **Development Facets: Factors that will affect Quiz Maker during development**
    a. OpenLiberty web server
    b. Google API
    c. REST API
    d. JSON-B

e. React API

f. JAX-RS

g. JSON web tokens

h. Google authentication system

3.2. Functions

1. If the user enters their login credentials, Quiz Maker shall call the **authenticate()** function to check if those credentials are valid.

1.1. If the user enters invalid credentials, then the user will return to the login screen.

1.2. If the user enters valid login credentials, then the user shall proceed to the home screen.

2. If an instructor uploads a CSV file containing a list of learners, **addLearner()** will be called.

2.1. If **addLearner()** is called, the uploaded CSV file shall be converted to a JSON format.

2.2. If a learner is successfully added to a course from the result of the **addLearner()** function, then **addLearner()** shall notify the instructor that the learner is now in the course.

2.3. If **addStudent()** is unable to add a learner to a course, then **addLearner()** shall notify the instructor that the specified student is unable to be found.

3. If an instructor requests to delete a learner from a course, **deleteLearner()** shall be called.

3.1. If **deleteLearner()** is called, selected learners shall be removed from the course roster.

3.2. If a learner is successfully removed from a course from the result of the **deleteLearner()** function, then **deleteLearner()** shall notify the instructor of the successful removal.

3.3. If the **deleteLearner()** function is unable to remove a learner from a course, then Quiz Maker shall notify the instructor that the learner was unable to be removed.

4. If a learner clicks on "Create Quiz", then Quiz Maker shall call the **createQuiz()** function.

4.1. After a user creates a quiz, then **createQuiz()** shall create an ID of the created quiz shall be created.

4.2. The ID of the created quiz shall be stored in the Quiz Database.

5. If an instructor requests to add a course, then Quiz Maker shall call the **addCourse()** function.

5.1. If **addCourse()** is called, then the created course shall be added to the Course Database.

6. If an instructor requests to delete a course, then Quiz Maker shall call the **deleteCourse()** function.

6.1. If **deleteCourse()** is called, then the requested course shall be removed from the Course Database.

6.2. If **deleteCourse()** is unable to remove the requested course from the Course Database, then the instructor shall be notified of the unsuccessful removal.

7. If an instructor requests to delete a quiz, then Quiz Maker shall call the **deleteQuiz()** function.

8. If a learner or instructor upvotes a quiz question, then **normalize()** shall be called

8.1. If a learner or instructor downvotes a quiz question, then **normalize()** shall be called

8.2. If **normalize()** is called, a quiz's rating shall be calculated by first subtracting the total of downvotes they have received from the total upvotes they have received.

9. While in a particular course, if a learner selects a quiz to take, than the **takeQuiz()** function shall be called.

9.1. If the **takeQuiz()** function is called, the learner shall be taken to the first question of the selected quiz

10. While taking a quiz, if a learner clicks on the flag icon, the **reportQuestion()** function shall be called.

10.1. If the **reportQuestion()** function is called, than a report containing a learner's feedback for a given question shall be sent to the instructor of the course

3.3. Logical Database Requirements

1. Created quizzes from a learner shall be stored in the Quiz Database

   1.1. Created quizzes shall be stored in the form of an ID

2. Created courses from an instructor will be stored in the Course Database

2.1. Courses shall be stored in the form of an ID

3. Learners will be stored in the User Database
4. Instructors will be stored in the User Database
5. The database must store an instructor's created exams

3.4. Design Constraints

1. Quizzes must consist of only multiple choice questions.
2. Multiple choice questions shall only have 5 possible answers.
3. Quiz Maker must use JSON web token for authentication.
4. Quiz Maker must run on an Open Liberty Server.
5. Quiz Maker shall use the JavaScript React Library for the GUI.
6. Quiz Maker must not check for intellectual integrity.
7. Quiz Maker shall use the Google API for the login process
8. Users must login using their Oswego gmail account
9. Instructors must be allowed to save quizzes they have created.
10. While a learner is taking a quiz, one question shall only be displayed at a time.
11. While a learner is taking a quiz, after answering one question, instant feedback will be given to the learner on which answer is right
12. If the learner submits a wrong answer, then instant feedback shall also be given to the learner on which answer is wrong

3.4.1. Standards Compliance

Quiz Maker must comply with fundamental web accessibility law.

3.5 Software System Attributes

3.5.1. Reliability

Given the functionality of OpenLiberty, this web app is expected to function without fail.

3.5.2. Availability

Quiz Maker shall be able to run at any given time, even when a microservice is down

3.5.3. Security

1. While attempting to login, users must provide an @oswego.edu email.
2. If a user provides a valid @oswego.edu email and corresponding password, the user will go to the home page.
3. If a user provides a non–@oswego.edu email, the user will go back back to the login screen.
4. Administrators shall exclusively have the ability to promote learners to instructors.
5. Users shall have the ability to delete only their own quizzes and quiz questions.
6. Instructors shall exclusively have the ability to delete quizzes created by learners in a given course.
7. Instructors shall exclusively have the ability to delete quizzes created by learners from a given course.
8. Instructors can only modify quizzes within the classes they are assigned to.
9. Quiz Maker shall use google authentication using JSON Web tokens
10. If a user enters the URL of a given page without successfully logging in, then Quiz Maker shall take that user to the login page.

3.5.4. Portability

1. Users shall have the ability to clone source code from Quiz Maker's main GitHub repository.
2. Users shall have the ability to download a ZIP file of the source code from Quiz Maker's main GitHub repository.
3. Quiz Maker must be able to run on any server.
4. Quiz Maker must run on the most updated version of Maven and Java.
5. Quiz Maker will be a web application, therefore it must operate on any operating system with a web browser.
6. Quiz Maker shall use portable languages such as React (Javascript) and Java

3.6. Requirements for a Technical Actor

Learners:

- Goals:
  a. Sign in with only a @oswego.edu gmail account
  b. Create quizzes
  c. Modify their own quizzes
  d. Take Quizzes
  e. Upvote a quiz question
  f. Downvote a quiz question
  g. Report a quiz question
- Scenarios:
  a. Creating a Quiz:
     - Learner opens Quiz Maker, enters valid login credentials and successfully logs into Quiz Maker. After logging in successfully, the learner views the list of their courses, learner clicks on a course, goes to a home page of selected course, clicks on “Create a Quiz”, the learner clicks on relevant quiz topics, clicks on “Create Questions”, the learner then goes to the “Create Quiz" page. After this, the learner then types in a question, the learner types possible answers for that question. The learner then marks the correct answer for that question. The learner repeats this process depending on how many questions a learner wants to have on a quiz. After finishing creating the quiz, the learner clicks on “Finish Creating Quiz”. Then the learner names the quiz. The learner clicks on “Create Quiz” and goes back to their course home page.
  b. Taking a Quiz:

- Learner opens Quiz Maker, enters valid login credentials and successfully logs into Quiz Maker. The learner then views the list of their courses, clicks on a desired course. The learner is then taken to the home page of the selected course, and the learner sees “My Top Rated Quizzes” at the top. Below “My Top Rated Quizzes” is a section called “Quizzes I Made” and “Quizzes I Took”.The learner then clicks on a quiz and is taken to the first question. The learner then selects one answer. If their selected answer is correct, the learner whose selected answer will be displayed as the correct answer. The learner then moves on to the next question. The process repeats until the learner answers the final question.

c. Reporting a Quiz Question:

- Learner opens Quiz Maker, enters valid login credentials and successfully logs into Quiz Maker. The learner then views the list of their courses, clicks on a desired course. The learner is then taken to the home page of the selected course, and the learner sees “My Top Rated Quizzes” at the top. Below “My Top Rated Quizzes” is a section called “Quizzes I Made” and “Quizzes I Took”. The learner then clicks on a quiz and is taken to the first question of that quiz. While attempting to answer the first question, the learner spots a mistake in the question and clicks on the “flag” icon. A pop-up is then displayed, giving the learner a variety of possible reporting options to choose from. The learner selects one of the options, then clicks “Submit”. After clicking on submit, an email is sent to the creator of the quiz, which contains details of the report. .

- Solution-Oriented Requirements:
    1. While creating a quiz, learners shall type in questions
    2. While creating a quiz, learners must type 5 multiple choice answers per question
    3. While creating a quiz, learners must set one answer as the correct answer per question

4. While taking a quiz, learners shall select one multiple choice answer per question
    - 4.1. If the answer that the learner selected was correct, then the selected answer will be displayed as the correct answer.
    - 4.2. If the learner selects the wrong answer, then the correct answer will be displayed.
    - 4.3. If the learner selects the wrong answer, then the selected answer will be displayed as the wrong answer.
5. While upvoting a quiz question, learners shall have the ability to click on the “thumbs up” icon.
    - 5.1. The “thumbs up” icon shall represent upvoting a given quiz question
6. While downvoting a quiz question, learners shall have the ability to click on the “thumbs down” icon.
    - 6.1. The “thumbs down” icon shall represent downvoting a given quiz question
7. To report a quiz, learners shall have the ability to click on the “flag” icon.
8. While reporting a quiz question, learners shall have the ability to select from the following options:
    - 8.1. Spelling Mistake
    - 8.2. Wrong Question
    - 8.3. Wrong Answer
    - 8.4. Other
9. While reporting a quiz question, if the learner decides the available options aren’t relevant to their issue, the learner shall have the ability to type the issue with a given question.
10. After a learner reports a quiz question, an email shall be sent to the creator of the quiz.
    - 10.1. The email shall say that a learner has found a problem with the quiz.
    - 10.2. The email shall also contain the question that the learner reported
    - 10.3. While reporting a quiz, if a learner selected “Other” to type in a response, the email shall contain the learner’s response.

10.4. While reporting a quiz, if a learner did not select “Other”, then the email shall contain the choice that the learner selected

- Functional Requirements:
  1. Learners shall be able to search for a quiz in the database
  2. Learners should be notified of a searched quiz doesn’t exist in the database
  3. Learners shall be able to filter through the list of quizzes
  4. Learners shall be able to see a list of quizzes they created
- Quality Requirements:
  1. Learners shall have the ability to take a quiz once it’s created
  2. Learners shall be able to create a quiz without Quiz Maker closing unexpectedly
  3. Learners shall be able to take a quiz without Quiz Maker unexpectedly
  4. Learners shall be able to switch from Light Mode to Dark Mode
  5. Learners shall be able to switch from Dark Mode to Light Mode

- Constraints:
  - a. A correct answer must be marked once a learner submits their answer to a given question
  - b. A learner must be allowed to report a quiz question
  - c. A learner must not be allowed to leave comments on quizzes

Instructors:

- Goals:
  - a. Sign in with only an @oswego.edu gmail account
  - b. Create quizzes

c. Take quizzes
d. Upvote a quiz
e. Downvote a quiz
f. Modify quizzes
g. Modify quiz questions
h. Report a quiz question
i. Import Multiple Quiz Questions to create quizzes
j. Delete quizzes made by learners
k. Delete quiz questions made by learners
l. Provide feedback about why the instructor chose to delete a quiz or quiz question
m. Create a class roster
n. Modify the class roster
o. Save quizzes they have created from learner-generated quiz questions for future use
p. Enable quizzes within their class roster, including their own, to be made accessible to the public
q. Enable quizzes within their class roster, including their own, to be made private from the public
r. Create a copy of an existing quiz

- Scenarios:

  a. Creating a Course

     - The instructor first opens Quiz Maker, enters valid login credentials and successfully logs into Quiz Maker. After logging in successfully, the instructor clicks on “My Courses”. The instructor then adds the course roster by uploading a CSV file containing a list of students and their SUNY Oswego email accounts. After that, the instructor names the course. After naming the course, the instructor clicks on “Create Course”.

  b. Creating a Course Roster

        - The instructor first opens Quiz Maker, enters valid login credentials and successfully logs into Quiz Maker. After logging in successfully, the instructor views the list of their courses, and the instructor clicks on a course that is ongoing. Then the instructor is taken to the home page for the selected course, then the instructor clicks on “Add/Remove Student”. After that, the instructor is taken to the “Add/Remove Student” web page. While on that page, the instructor sees a dotted text box that says ‘Drop your Class roster CSV file or click to upload’. The instructor clicks on class roster text that is highlighted and then uploads a CSV file containing a list of students. The CSV file is read and each student is added to the course roster list. The instructor is finally notified that the course roster is successfully created. .

- Creating a Quiz:
    a. The instructor first opens Quiz Maker, enters valid login credentials and successfully logs into Quiz Maker. After logging in successfully, the instructor views the list of their courses, instructor clicks on a course, goes to a home page of selected course, clicks on “Create a Quiz”, the instructor clicks on relevant quiz topics, clicks on “Create Questions", the instructor then goes to the “Create Quiz" page. After this, the instructor then types in a question, the instructor types possible answers for that question. The instructor then marks the correct answer for that question. The instructor repeats this process depending on how many questions an instructor wants to have on a quiz. The instructor wants to also import quiz questions from previous quizzes they created. The instructor clicks import quiz and imports a list of quiz questions they favorited. After creating the quiz, the instructor clicks on " Publish Quiz”.

- Deleting a Course Roster
    a. The instructor first opens Quiz Maker, enters valid login credentials and successfully logs into Quiz Maker. After logging in successfully, the instructor

views courses with the title “Courses where I’m an Instructor”. The instructor clicks on one of those courses and goes to the homepage of the selected course. The instructor then goes to the “Roster” page, which contains a list of learners currently in the course. To select a learner to remove from the course. the instructor clicks on the learner to be removed from the course. That learner will be highlighted in red. The Instructor clicks on ‘Remove Student’ and the learner is deleted from the course..

b. The instructor first opens Quiz Maker, enters valid login credentials and successfully logs into Quiz Maker. After logging in successfully, the instructor views courses with the title “Courses where I’m an Instructor”. The instructor clicks on one of those courses and goes to the homepage of the selected course. The instructor then sees a Manage Students section , which contains a list of learners currently in the course. To remove multiple students in the course, the Instructor clicks on all of the students that the Instructor wants removed from the course. All of the selected students will be highlighted in red. The Instructor then clicks on ‘Remove Students’, and all of the selected students will then be removed from the course.

- Solution-Oriented Requirements:
  1. While creating a quiz, instructors shall have the exclusive ability to clone previously created quizzes.
  2. While creating a quiz, instructors shall have the ability to type questions
  3. While creating a quiz, instructors shall have the ability to type in 5 multiple choice answers
  4. While creating a quiz, instructors must set one correct answer per question
  5. Exams shall be created learner-generated quiz questions
  6. Instructors shall add learners to a course by uploading a CSV file containing a list of the course’s students

7. Instructors will have to ability to star quizzes in order to save them to a database

- Functional Requirements:
    1. Instructors shall be notified if they are attempting to save a quiz that already exists in the database
    2. Instructors must be able to save quizzes to the Quiz database
    3. Instructors must be able to delete quizzes from the Quiz database
    4. While attempting to add learners to a given course, instructors must be able to upload a CSV file containing a list of learners
    5. Instructors shall be notified they attempt to delete a quiz that doesn't exist in the database

- Quality Requirements:
    1. Instructors shall be able to create a quiz without exiting the quiz unexpectedly
    2. Instructors shall be able to add the appropriate number of learners to a given course
    3. Instructors shall be able to smoothly delete a quiz
    4. Instructors shall be able to effectively add learners from a course roster
    5. Instructors shall be able to effectively delete learners from a course roster
    6. Instructors shall have the ability to switch from Light Mode to Dark Mode
    7. Instructors shall have the ability to switch from Dark Mode to Light Mode

- Constraints:
    a. Instructors must be allowed to save quizzes they created
    b. Instructors must allowed to review quizzes they created
    c. Instructors must be allowed to delete learner generated quizzes

d. Instructors must be allowed to delete learner generated questions
e. Instructors must be able to be given a choice to provide feedback as to why a quiz or a quiz question was deleted
f. Instructors must be allowed to create topics for a given course

4. Diagrams:

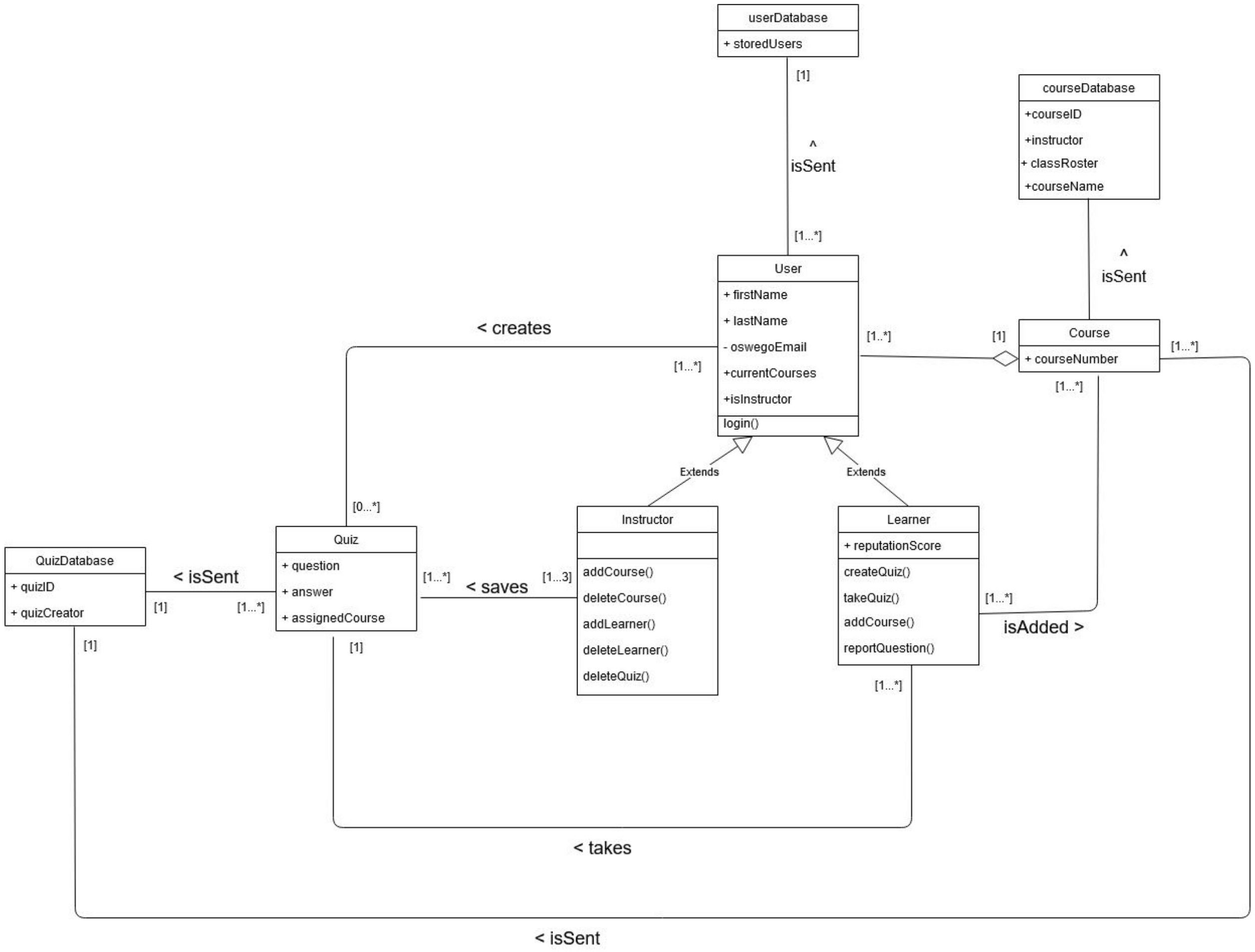

Figure 1: UML Class Diagram displaying relationships between the different classes of Quiz Maker

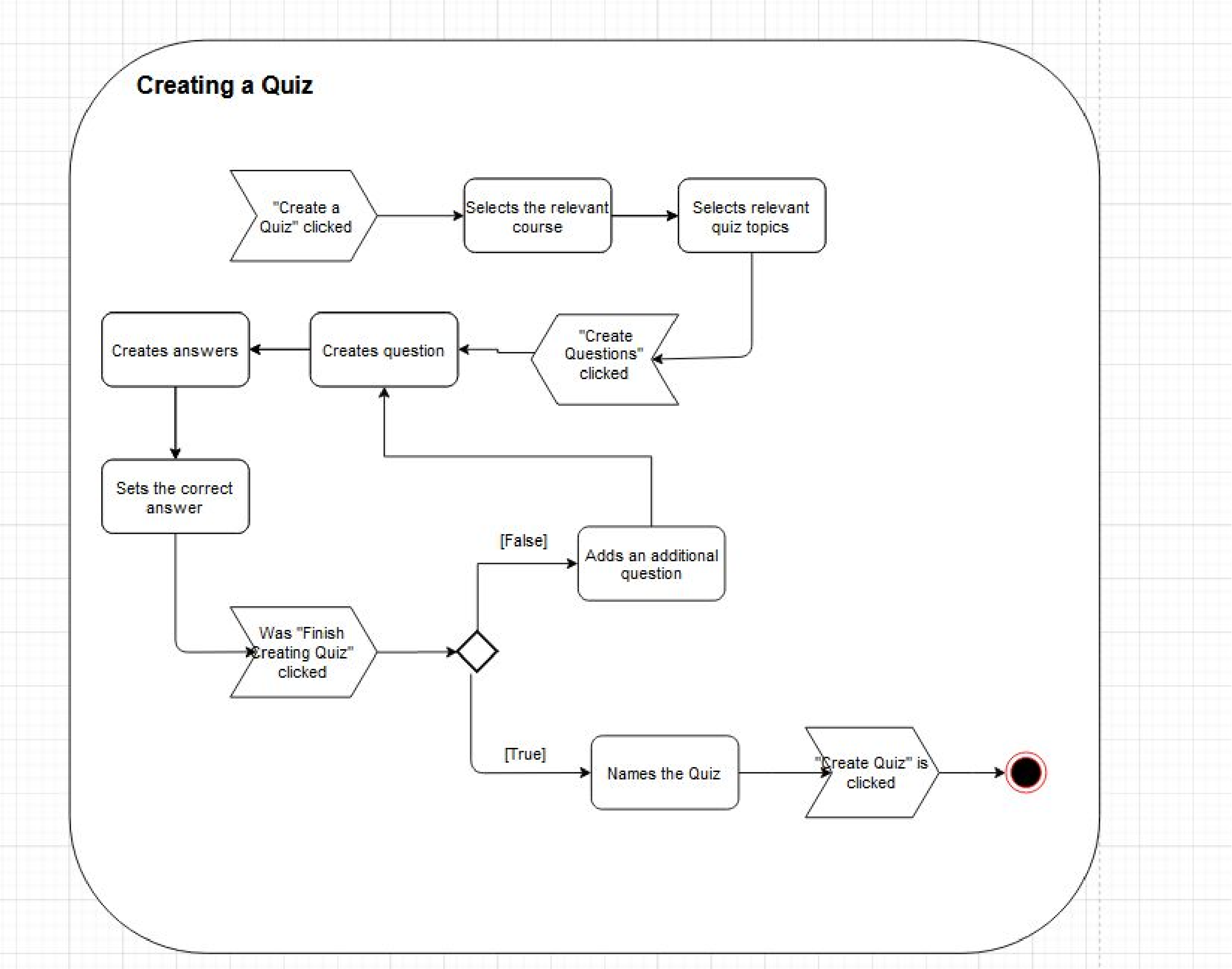

Figure 2: UML Activity Diagram displaying the process of creating a quiz

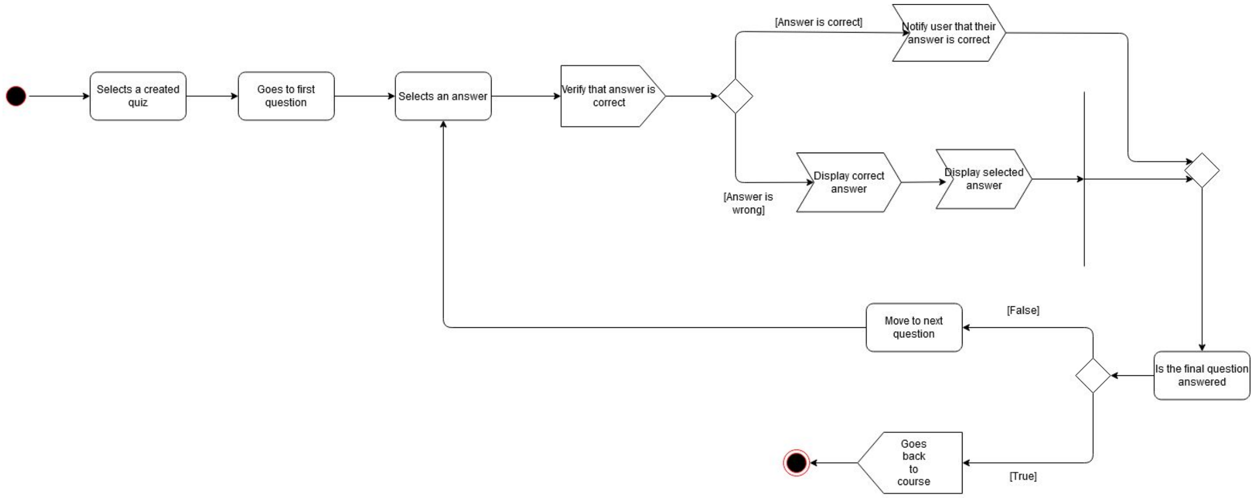

Figure 3: UML Activity Diagram displaying the process of taking a quiz

5. Appendix

5.1 Acknowledgements:

The Requirements Team would like to thank and acknowledge Halle Bistrol’s earlier contributions to the Requirements Specification of Quiz Maker.