Proposed Requirements
Software Design
SUNY Oswego
Spring 2017
Dr. Tenbergen and Dr. Harris

Introduction:

Purpose:

This document shall specify the requirements for the OpenClicker system that shall be created by the Spring 2017 Software Design class. This document shall enable the participants in that class to read and clearly understand the requirements for the system.

Scope:

The OpenClicker system will enable that Instructors to ask their Students questions during class and the students to answer said questions. It shall display the results of this interaction to the Instructor, it can also display this to the Students at the Instructor's discretion. This system shall offer a free, mobile alternative to other similar clicker systems that require Students and Instructors to pay for additional devices.

Definitions:

A **confidence scale** is a slider (1-10) indicating a student's confidence with their answer.

A **quiz review** will display to the user their overall score and each question in the quiz. For each question, the correct answer will be indicated, along with the answer chosen by the student. This list shall appear as a *consecutive list of questions.* This **quiz review function** can be turned on or off.

An **individual quiz** is a quiz which each student takes privately.

A **group quiz** is a quiz which a group of students (one or more) take simultaneously.

Private questions will be seen only by the sending user and the instructor.

Public questions will be posted for all students to see, as well as the instructor. (*Other users can respond to public questions with comments?*)

Overview:

The rest of the following document shall outline requirements in more specific detail for the above outlined system. The requirements are split up into two main sections, which are divided further by requirements pertaining to various functionalities of the system. The two main sections are features that can not under any circumstances be left out of the system, the other is requirements to additional features that would improve the system but are not necessary.

BARE MINIMUM FEATURES

High Usability:

The system shall be a web based application.

The interface shall resize cleanly for common screen sizes.

The interface shall be simple to use.

The interface shall be intuitive.

The interface shall be aesthetically appealing.

The system shall provide a link to a help page to further explain its functions to confused users.

The interface shall be readable.

The system shall be able to interface with blackboard. *(If we are able.)*

While Students are connecting to a course, a “connecting…” message shall be displayed that they are connecting to the course.

Legality

The name for the system must not violate any licensing laws.

The contents of the system must not violate any licensing or privacy laws.

Login:

Users shall be given feedback after successful login.

A hash system shall be used for login security.

Users shall log in by a username and a password.

Upon login the User shall be taken to a dashboard relevant to their user type in the system.

Dashboard:

Student View:

Student shall be able to see an active question and submit an answer to it.

Course notes shall be visible to all Students in the Course.

A student’s answer shall remain anonymous from other students.

Instructor View:

Instructors shall be able to add course notes to their Course(s).

A student’s answer shall be visible to the Instructor who owns the Course(s).

Instructors shall have control over their owned Courses.

The instructor may set permissions for what abilities Students have during a quiz. For example, returning to previous questions.

Instructors shall be able to input answers for their Students.

Instructors shall be able to regulate which users are able to join that class/course/session.

The instructor shall be able to see what students are logged in to a given Course.

Instructors shall be able to end a session.

Instructors shall be able to start a session.

Questions:

Instructors shall be able to start a question for their course.

Instructors shall be able to end questions.

Instructors shall be able to add time to an active question’s timer for their course.

Instructors shall be able to remove time to an active question’s timer for their course.

Students shall be able to answer an active question by pressing a multiple choice answer button on their interface.

If a student submits an answer, but still has access to the question, the student may resubmit a new answer.

A personal answer history shall be visible to users.

A selected but not submitted answer shall be indicated to the Student. *(Might belong under results.)*

Results:

Real-Time Graphs (as people answer the graph populates)

○ The system shall be able to display real-time histograms which populate in as users answer a question.

○ Answer statistics shall be displayable using histograms.

Upon successful answer submission, user shall receive feedback to indicate their answer was properly submitted.

The time remaining for a given question shall be displayed during the course of the question.

System shall detect strange behavior that may indicate academic dishonesty (*ex: frequent logging in/out of multiple accounts during a question*).

The system shall measure attendance for each question.

Attendance information shall be available to the instructor of the course for a Quiz.

Upon error (*such as connection loss*), an informative message shall be displayed.

When possible, error messages should offer a choice to the user to attempt to fix the error automatically (*reconnect button*).

CRUD Courses:

The system will have a feature to download a class list as a csv file.

The system will have a feature to upload class list as a csv file.

An instructor shall be able to create a course.

Open Enrollment via Link:

- ○ Students will be able to click a link provider by the Instructor to join a course without the Instructor manually adding Students.

Enforce enrollment through csv file:

- ○ Instructors will have a way to add students through a csv file.

POSSIBLE ADDITIONAL FEATURES

Multiple Types of Questions

Created questions shall have one of several types of answers:

- ○ Questions with one correct answer.
- ○ Questions with more than one correct answer.
- ○ Extended response questions (*Text fields*).
- ○ Upload a file.
- ○ Slider Responses.

Admin. Role:

Admins will have full control over the system.

Admins shall be able to assign courses to instructors (*Admin vs Instructor?*).

Confidence Scale:

There shall be an optional *confidence scale* for questions.

Instructors shall determine whether a *confidence scale* is present for a given question in their quiz.

Group Quizzes:

Course class can be split into teams for trivia game type quizzes. (*Do we want this? It sounds like it will require a bit more than it sounds like at first.*)

` During a *group quiz* a counter shall display the number of user answers that have been received out of the total number of users that can submit an answer. (*ex:5/13 shows that 5 users have submitted an answer, out of 13 users that are able to submit an answer*).

API:

The system shall have an API.

The API will allow users to use parts of the system's capabilities separately from the system.

Exportable data:

System shall detect strange behavior that may indicate academic dishonesty (*ex: frequent logging in/out of multiple accounts during a quiz*).

The system shall measure attendance for each quiz.

Attendance information shall be available to the instructor of the course for a quiz.

Ask the Instructor:

Users shall be able to send *private questions* or *public questions* to the instructor.