Classroom Connect

Software Requirements Specification

Kyle Wright, Anthony Chan, Jacob Wyner, Joseph Cao

1. Introduction

Interactions between students and faculty in lecture halls can become unnecessarily difficult and one sided. Classroom Connect is an intuitive and reliable software solution which serves to bridge the current gap between students and faculty in the same space through real time polling.

Classroom Connect allows for direct communication between participants, where the main interactions are from the student to lecturer or from student to student. All users who are within the same virtual and physical space can see each others public interactions.

As of yet, the software will not be provided as a native iOS or Android application, but moreso as a mobile-friendly web service. As a web service, the most up to date version of the application is only available via an internet connection. In fact, the direct interactions are only possible when users are connected to the same virtual classroom instance.

2. User Requirements

Classroom connect provides users with real time polling analytics, bridging the student to lecturer communication gap a little bit more every usage. In order to provide such a service, several initial user functionalities and components are necessary:

- User Dashboard
- Mobile Friendly Interface
- Poll Instantiation
- Poll Search
- Poll Visualization
- Poll User Feedback

2.1 Software Interfaces

- JS libraries (JQuery, Charts.js, Node.js, Express.js)
    - https://jquery.com
    - http://www.chartjs.org

    - https://nodejs.org/en/
    - https://expressjs.com
- Bootstrap (HTML5, CSS3)
    - http://getbootstrap.com
- Java Database with SQL (if no Node.js Backend)
    - http://dev.mysql.com
- Github (Git)
    - https://github.com

2.2 User Interfaces

The system will interact with the user in one of two ways. Either the user will instantiate a lecture or the user will connect to a lecture that has already been created. If the user has instantiated a lecture that user will be able to both view the results to polls and clear the results so a fresh poll can be done. That user will also be able to view a live update chart of the student's understanding of the lecture.

If the user is connecting to the another user's lecture they will have the option of either responding to polls when prompted or updating their understanding of the lecture whenever they wish. This user will not have the ability to view either the results of the polls or the live stream of understanding

2.3 User Characteristics

Development of Classroom Connect is focused at catering towards students at or above higher education (BA, BS, MS, PHD, etc.). A certain level of maturity and responsibility must be held while using this application, as the team of Classroom Connect is not liable for harassment done to or created by users of the application. Users must be well versed in knowing how to interact with certain UI elements such as form fields, buttons, drop down fields, and more. In addition, desired users are ideally knowledgeable about identity theft and secure password protection.

2.4 Assumptions and dependencies

- Storage & Server Capabilities
- User Demographic Experience
- Internet Connection

3. System Requirements

These subsections contain all the software requirements at a level of detail sufficient enough to enable designers to design a system to satisfy those requirements, and tests to test that the system satisfies those requirements.

3.1 Functional Requirements

Functional Requirements describe what the service should do. Below are the necessary functions that Classroom Connect must implement to become a complete service:

- Student / Faculty Roles
  - Software should have two separate user interfaces with distinct user permissions depending on the role of the user.
- Sign In & Authentication
  - Software should authenticate the legitimacy of a user's enrollment in the class before allowing responses to be processed.
- Data Backtracking
  - Software should be capable of backing up data and providing such to user if user accidentally loses such on his / her own local system.
- Data Encryption
  - A reasonable effort should be made to ensure that data transferred by the software is stable and not easily intercepted with malicious intent.
- Data Visualization
  - A user signed into the faculty role of the software should be provided with a user interface that easily and quickly provides 'at-a-glance' live and initiated poll results.
- Poll Initiation
  - A user signed into the faculty role of the software should be provided with easily accessible means of initiating a custom poll. This will take the form of a multiple choice question.
- Push Notifications
  - A user signed into the faculty role of the software should be notified of a significant drop in student understanding of course material, without the faculty role initiating the results of such.

3.2 Non-Functional Requirements

Non-Functional Requirements

- Security
    - System is responsible for ensuring that data created, received, and deleted by users is not public or accessible by other users or non users without direct consent by the former.
- Scalability
    - System must be able to proficiently handle users numbering from 10 to 150+ users.
- Maintainability
    - System functions must be easily customized by users, and easily
- Backup
    - Users must be able to access previously created data, ad hoc.
- Documentation
    - Development of the software must be well laid out and explained, for future modification.
- Modularity
    - Instances of the web service must be easily created, and should not impede on other software modules created.
- Quality
    - Software is expected to completely and concisely deliver information on time, with reasonable implementation onto various visual interfaces
- Privacy
    - Data created, modified, stored, used, read, analyzed, must be secured, and only visible and granted to the users whose privacy depends on such.
- Accessibility
    - Software is available on all platforms via the web.
- Portability
    - [See above]; Software must be easily accessible on platforms other than desktops - tablets, mobile
- Price
    - Software is a free service, as it must not require users to pay. While donations will eventually be acceptable, users should not be limited from the service due to expenses.

3.2.1 Software Quality Attributes

List the quality attributes that are important for your system (e.g., reliability, security) and explain why they are important for your system and how they will be measured.

- Cross-Platform Portability

- Response Time
- User Interface & Experience