Turing Point

Educational Polling Software

Scott Hadley, JT Newsome, Tony Niven, Sean O’Connor

Senior Capstone Project: Design Document

Table of Contents

Product Summary

Classroom learning has been evolving for centuries and educators have been experimenting with ways to get students more actively engaged in the learning process. Research has shown that modern methods of classroom interaction — such as ‘interactive learning’ using electronic polling devices — have the potential to improve the quality of education a student receives [1]. However, current systems are failing to meet classroom needs by being too expensive and limited in features.

Currently, a majority of students carry their own mobile device and existing polling systems have been forced to recognize the dominance of these devices in a classroom setting. As a result, developing applications that are specifically created to work on modern smartphones. Our system will leverage the abundance of these devices to provide a more effective interactive learning experience, by providing more features and value than existing solutions.

Problems with existing systems

Existing polling systems charge for expensive hardware components, handheld ‘clicker’ devices, and annual subscription fees; including the Turning Technologies polling system used at the University of Utah. These systems are designed to extract maximal profit from students with little regard for the effective increase in their academic performance.

The expensive ‘clicker’ system mentioned above also provides almost no feedback other than acknowledgement that the a student provided an answer. This leaves students unsure about why their answer was incorrect, and without a way to review their responses in the future. Our system will be designed to address these problems by implementing a web application that students can access using their mobile devices, where they will be able to respond to polling questions and interact with our system using our intuitive user interfaces.

Instructors and students will use our system differently, therefore we will implement a web portal for Instructors and a separate portal for students. The two portals will be joined by a server/database backend, which will provide the ability to link these portals together in active sessions. Students will be able to review previous quizzes, their responses, and resources provided by their instructors to better understand material they answered during a quiz or poll. Instructors will be able to create questions or polls, set point values, and provide students with feedback. Additionally, Instructors will be able to use visualization and history features to better understand class performance in specific topics and time spans.

Summary

In order to facilitate better development, and to ensure our product meets the needs of existing and future students and instructors, we are actively obtaining feedback on existing interactive learning solutions and ideas for our system. By doing so, the development of our system will address the needs and concerns of the parties to whom the product is of the most value. By using the agile development methodology, and incorporating feedback regularly, we will create a system which is compelling for both instructor and student alike. Our system will become an integral learning tool rather than something needlessly appended to the learning process as a contractual obligation.

The resulting product will be useful, informative, intuitive —and most importantly, increase the effectiveness of interactive learning in classrooms due to our unique development approach and perspective.

Background and Technical Requirements

Existing Solutions

There are currently over 25 companies offering some form of polling solution, but a vast majority of these companies offer hardware-only solutions that require the purchase of Radio Frequency (RF) receivers and clicker devices, and some primarily focus on corporate clients, conventions, or events [2]. Some of these solutions are open-source, including LimeSurvey, which claims to be the most popular Open-Source polling solution. LimeSurvey has limited poll question types, no mobile support, no user management, and a complicated poll management interface. Other free polling solutions also have very few features, and poorly designed user interfaces [3].

Most of the available polling solutions do not disclose their pricing model on their websites, and instead offer phone numbers and/or contact forms to their sales department. This practice leads to the assumption that their prices are not affordable as the companies are not up front on their pricing. Some efforts have been made to contact several of these companies regarding their prices, and it has been discovered that some charge between $27-500 per clicker device, $200-700 per IR receiver base stations, and up to $1000 per day to have a company setup polling equipment for a conference or event [2].

Poll Everywhere is a popular web and mobile polling solution that has several of the features we plan to implement in our polling system, including users and roles, grading of responses, results exporting, and PowerPoint integration. Poll Everywhere offers four question types: multiple choice, open ended, Q&A, and clickable images. Poll Everywhere offers a variety of pricing plans for businesses and schools, ranging from free to $1999+ monthly for businesses, and free to $3000+ annually for schools. Many of their features are only available for paid service plans. Turning Technologies offers both a clicker hardware solution as well as a mobile platform, and offers features such as PowerPoint integration and basic grade management.

What sets our polling system apart from the currently available systems primarily is our focus on affordability, a rich variety of features, and our emphasis on educational learning. The hardware-based polling solutions are especially expensive and limited in features by nature due to users having limited response options, and the requirement to purchase dedicated hardware. Our primary competitors are Turning Technologies and Poll Everywhere since they are both market leaders and offer a software based solution. Our system will not require upfront fees from students or instructors, and it will also be less expensive than other polling solutions for educational institutions. The majority of costs associated with the use of our system will contribute directly to server maintenance, upkeep and development costs.

Our system will have more features than Turning Technologies, since their solution is mostly restricted to multiple-choice question types, and we plan to offer more variety. Poll Everywhere offers more question types, but our system will be different due to the integration of a multiple-question quiz feature, as well as adopting the competitor's best question types. Neither of these systems offer any form of feedback to students to aid in their understanding of the material, nor a resource to help students understand why they missed a particular question. Therefore, we plan to add this feature to our system to aid in the learning process.

Furthermore, these systems don’t provide adequate response tracking history for students, so students aren’t able to review their answers to previous polls. Therefore, we plan to implement this as a feature in our system. For instructors we will offer rich visualization features to track the student’s performance. These features are not offered by our competitors and we think they will help set us apart. We also plan to offer instructor grading management tools and the ability to export grades in various formats, which are features offered by top competitors. Lastly, we plan to make user-friendliness and ease-of-use a top priority because we feel that this will be a defining feature of our product that will set us apart from the competition.

Technologies Used

Instructor Portal

- JQuery
- JavaScript
- D3 JS - Data Visualization Library
- AJAX

Student Portal

- Javascript
- JQuery
- AJAX

Database

- MySQL on Amazon Web Services (AWS)

Server

- Hibernate - A powerful framework on Java that maps the database to Java entities, and allows persistence to/from the server backend.
- Jersey Server - A Java server that handles REST requests.
- Apache Web Server - A web server that serves our web portals.

Software Engineering Tools

- Atlassian JIRA - Bug tracking and project management.
- Confluence - Dev and user documentation.
- Crucible / Fisheye - A tool for code reviews and SVN repo browsing
- SVN

Testing Tools

- Selenium - A tool in Java to test and verify website client functionality.
- JUnit - A standard Java unit testing platform.

Testing Audience

We plan to collaborate with Davison Germaine of the University of Utah Scientific Computing Institute for in class testing. We have targeted development goals toward feedback received from our initial meeting in order to facilitate the instructor's needs in the classroom in the spring semester. With this sort of collaboration we plan to gain valuable feedback on the system development and features.

User Requirements

We aim to create a system that is managed by our group, so users will not need to install or configure any software. To that end, we anticipate no special requirements on the part of users other than having a modern, functional web browser on their desktop or mobile device. Their browser will need to be running javascript as it will handle the main interaction between their browser and our back-end system.

Requirements Analysis

System Architecture

(See Appendix C: Figure 1.0 - Proposed System Overview)

- **Database:** The database will store user information and saved poll data for use by the server and portals. In order to communicate with the database, the portals will have to communicate and make requests through the server. We plan to use a traditional relational database technology and will likely use MySQL for this purpose.
- **Server:** The server will be responsible for brokering information between the database, instructor portal and student portal. It will also be responsible for maintaining active polling sessions until they are finished and saved to the database. We anticipate the bulk of the server to be written in Java, with the database communication written using Hibernate, and using Jersey Web Server for handling the RESTful web requests.
- **Instructor Portal:** The Instructor's Portal will be a web-based portal that interacts with the web server backend, which allows instructors to create and edit polls and manage students. This portal will utilize standard web languages, such as HTML, Javascript, and CSS. Additionally the web portal will rely heavily on Javascript libraries such as JQuery. D3.js will be used to build charts and plots when instructors are viewing student's polling performance. AJAX will be used to dynamically fetch and replace portions of the page without the need to reload the entire page. This will be done heavily on on poll editing and launching page as there will be many actions that require content to be replaced (such as adding new poll questions, importing existing questions, and activating polls).
- **Student Portal:** The student portal will provide the student with the functionality to take polls and review previous poll results. The portal will interact with the server to retrieve polls from the database, and will sent poll responses to the server so that they can be logged in the database. The portal will be written in HTML, Javascript, CSS, and use Javascript libraries such as JQuery.

Personnel

- **Scott Hadley:** Scott will be responsible for creating the database schema and creating the queries that will be used to fetch and persist data to the database. Scott has created database schemas for previous classes, and has completed the database course. Additionally, Scott will also create java objects that marshal the database objects to Java objects that can be used directly on the server. Scott will also work closely with Tony on the server development once the database needs are met. Scott has experience working with Java web apps and Hibernate database mapping through his work experience at a web-based company.
- **JT Newsome:** JT will be responsible for the Instructor Portal during development. He has taken the Web Software Architecture and Database System classes so he is familiar with communicating with databases and using basic web languages as well as more specific ones such as JQuery and AJAX. He plans to use his academic and professional experience to learn and refine the development skills he needs in order to assist the team in completion of the Instructor Portal and project as a whole
- **Tony Niven:** Tony will be responsible for the creation of the server portion. The server will require asynchronous programming and the ability to communicate with both student and instructor portals as well as the database. Through professional work experience and school work, these tasks should be possible for this individual. Tony also has experience with D3.js and can help consult development of results visualization in the portals and database experience for help in the database.
- **Sean O'Connor:** Sean will be in charge with developing the student portal. He is familiar with Javascript, HTML, and JQuery through the Computer Graphics class he has taken. Sean will collaborate with JT, since the student portal and instructor portal shares some similar features.

System Features

Rank 1: Bare Essentials

- **Server**
    a. Session Hosting - Allows several students and one instructor to participate in polling sessions.
    b. Database Communication - The ability to interface with and communicate to the database.
    c. Multiple Class / Session Management - The ability to have multiple classroom and sessions open, serving different information to connected clients.

d. Multiple Questions Per Session - The ability to activate and manipulate several questions or question sets independently within a session.

- **Database**
  a. Schema - A basic schema to support rank 1 features.
  b. Server Database Communication - Between the server and clients.
- **Instructor Portal**
  a. Register/Login - The instructors should be able to register and login to the web application.
  b. Add Classes - Instructor can add a class to manage.
  c. Create/Edit Question - Create/Edit a question that will be used for later polling.
  d. Control (Start/Stop) Question - A question can be started or stopped.
  e. Create a lecture - A set of questions that are grouped together.
  f. Server Communication (asynchronous) - The instructor portal should be able to communicate with the server asynchronously.
  g. Basic Question Types - Multiple choice and numeric response questions requiring exact entry will be our initial question types.
- **Student Portal**
  a. Server Communication (asynchronous) - The student portal should be able to communicate with the server asynchronously.
  b. Login/Registration - The student should be able to register and login to the web application.
  c. Add Classes - The student can join a class.
  d. Take Quiz - The student can view and respond to a poll.

Rank 2: Planned Features

- **Server**
  a. Auto advance question - Make a service that optimizes turning off all questions but the given one.
  b. Authentication - Users will be authenticated and passwords will be encrypted before accessing secure information.
  c. Results Reporting - Query database to relay information stored within the respective portal from which it was requested.
- **Database**
  a. Additional Schemas - Schemas that allow for organizing lectures by week/subject.
  b. Add support to tag a poll question for later organization.
  c. Add support for additional question types.
  d. Add support for a question bank.

- **Instructor Portal**
    a. Auto Advance Question - Checkbox that makes it so when an instructor starts a new question all other questions in that lecture automatically stop.
    b. Deletions - Instructor can delete courses, lectures and questions.
    c. Import Previous Lecture Questions - Instructors can import a question from a previous lecture into the current lecture they are editing. Should load the data in the right pane then allow you to confirm import.
    d. Question Bank - Instructors will be able to create individual questions within a question bank, and import those questions into a lecture.
    e. Add Tags - Instructors will be able to tag questions for quick grouping in results, or later in the question bank (see rank 3).
    f. Assign Point Values - Instructors will be able to assign point values to individual questions.
    g. Question Grouping - Create hierarchal question groups to allow more fine grain control for instructors.
    h. Lecture Grouping - Instructors will be able to group lectures by week.
    i. Feedback Support - Instructors can add feedback to questions.
    j. View Raw Question Results - Score values for polls can be seen in manage results.
    k. Export Results - Export raw result data in a csv format.
    l. Results Visualizations - Add support for instructors to visualize classroom performance through charts and graphs.
    m. Advanced Question Type - Add support for Multiple Answer/Select All That Apply question type.
    n. Advanced Question Answering - Enhancement to numeric responses, allowing for an interval marked as correct, based on % difference or values.
    o. Manage Students - Instructor can view/add students in a class.
    p. Import List of Students - Instructors will be able to import a list of students from a file.
    q. Question Results Tab - A tab in the quiz play/edit page that you can switch to and see simple result statistics for that question.
    r. Answering Progress Status - Show how many unique students have answered the question.
- **Student Portal**
    a. View Raw Result - The student can view the score received for a given quiz.
    b. Improve View Result - When a student views a previous quiz the student will be able to see class statistics.
    c. Improvement to Take Quiz - Students can view what questions were already answered while polling. Create a sidebar that provides the student with easier navigation.

d. Get Feedback - When viewing previous quizzes the student will be able to view any feedback the instructor gave them for a specific question.
e. Organize previous quizzes - All quiz results will be organized by weeks.
f. Search by tags - Student can search for subject tags on the results page to view polls relevant to topics/subjects.

Rank 3: Bells and Whistles

- **Server**
  a. Authorized Web API - Authenticated parties can make requests of the server in order to extract certain data.
  b. Traffic Optimization - Server will have functional caching to allow for rapid reuse of received data.
- **Instructor Portal**
  a. ‘Results Explorer’ - The results explorer will allow instructors to formulate customized queries and derive their own data from existing poll data.
  b. Question Bank Refinement - Filter and sort questions in the question bank. Allow question groups to be visible as groups within the question bank.
  c. Timed Polling - Set a time limit for automatic closing of a particular question or question group after it is opened.
  d. Advanced Question Types:
     - Matching
     - Essay
     - Multiple Fill-In-The-Blank
  e. Cache Questions - Cache question details so that they do not need to be re-fetched from the server.
  f. Cache Results - Cache question results so that they do not need to be re-fetched from the server.
  g. Essay Question Grading View - Allow instructors to easily grade essay format questions
- **Student Portal**
  a. View remaining time for quizzes - When a student takes a quiz, the student will be able to see the available time he has left before the quiz closes.
  b. Cache Questions - Cache question details so that they do not need to be re-fetched from the server.
  c. Cache Results - Cache question results so that they do not need to be re-fetched from the server.
  d. Advanced Question types:
     - Matching
     - Essay

        - Multiple Fill-In-The-Blank

Software Engineering Techniques

- **Development Methodology**
    a. We plan to start by prototyping in a long agile cycle with feedback on features as they mature.
        - When our system is in working order we will transition to a shorter agile cycle to better incorporate feedback.

    b. We plan to use Google Style guides for the languages we plan to use.
        - We also plan on using industry standard best practices for web development such as proper hierarchy and designing for accessibility where possible.
- **Version Control**
    a. We have setup an SVN repository on our team server provided by the CADE lab. This SVN repository will be used to store our code revisions, to reference code for code reviews, and to host our code for building our project in Jenkins.

    b. We have selected SVN because our entire team is familiar with the basic operations of storing and fetching code using SVN. Additionally, we know that SVN will work with our other chosen tools (Jenkins and Crucible).

    c. We plan to commit our code to the repository as frequently as possible, so long as the commit does not break the project build. We will have a separate repository for scratch code for non-critical code versioning.
- **Bug / Development Tracking**
    a. We will be using Jira to track bugs, and to track the tasks assigned to individuals.
- **Testing tools and Methods**
    a. Methods
        - Mix of manual developer testing and automated testing in addition to classroom trials when possible.

    b. **Tools**
        - Using Selenium Web Framework for testing of web portals.
            - Selenium has bindings for a variety of languages including the language in which we plan to develop our server software (Java).
        - Maven or a similar framework for unit testing on Java code.
            - Maven is also directly supported by Jenkins and can be used to run Selenium tests.
        - Jira will be used to track bugs and problems as they are found.
- **Documentation Tools**
    a. We will use Confluence — a wiki style system that integrates directly with Jira. We plan to use this system for storing development documentation and procedures.
- **Communication Tools**
    a. We are currently using Google groups for collaborative efforts. We also use Skype regularly for online meetings.

    b. Jira has the ability to email automatically when updates to certain issues are performed, we plan to exploit this feature.

- **Team Meetings**
  a. We’ll have regular in-person standup meetings on Monday, Wednesday, and Friday that last between one to two hours.
  b. We’ll use Skype for online team meetings as needed.

Timeline

	Scott Hadley	JT Newsome	Tony Niven	Sean O’ Connor
Alpha	-	-	-	-
Week 1	Research / perform server client encryption and user authentication	Instructor: Delete courses, lectures and question (client and server) Instructor: Auto Advance Question (client and server)	Server: Question Results services and results Instructor: Basic result histogram for instructor when a question ends. Server: Lecture Results Services	Student: HTML for student result page Student: know which questions been answered on quiz Instructor: Lecture results Display (Textual statistical data)
Week 2	Instructor Portal - Create student management page.	Instructor: Question Results Tab Instructor: Student Answering Progress Status	Instructor: Results Export CSV Instructor: Results Export Other Types	Student: Get result after quiz end

Week 3	Update database schema	Instructor: - Feedback - Point Values - Tags	Instructor: Overall Results services	Instructor: Overall Instructor / Student: Results display (Textual statistical data) view feedback
Week 4	Update database server communication	Instructor: Advanced Question Types (Multiple Answer, e.g. select all that apply)	Clients: Basic Results Visualization improvements Help implement advanced question types	Student: Support more question type, (multiple answer)
Week 5	1) Update database server communication - Including question bank 2) Bugs / Testing	Instructor: Import Previous Lecture Questions Instructor: Question Bank	Server: Question Bank services for instructor portal -remove question from bank -Getting questions in bank	Bugs/Testing Server: Question bank -Add question to bank
Beta	-	-	-	-
Week 6	Setup testing / selenium framework	Instructor: Question Grouping	Server: Grouping services -add to group -remove from group Help with UI dev on grouping	Student: Tag feature for student's result
Week 7	Refine testing for clients and servers to detect bugs	Server: Lecture Grouping	Instructor: Results Explorer -Page and initial view	Student: Question Grouping for student
Week 8	Instructor sets valid ranges for fill-in-blank questions (client and server updates).	Instructor: Lecture Grouping, by Week (Client-side)	Instructor: Results Explorer -topical browsing -line charts over time	Student: Lecture grouping for student

Week 9	Instructor portal question types: - Matching - Essay	Instructor: Multiple Fill-In-Blank question type	Student: Answer Multiple fill in the blank question	Student: Answer essay questions
Week 10	Server-side poll timer support	Instructor: Essay Question Grading View	Student: Answer matching question	Student: cache question details
Production	-	-	-	-
Week 11	Instructor Caches: - questions - results	Instructor: Question Bank Refinement	Instructor: Results Explorer -Customizable queries	Student: cache results
Week 12	Testing / Debugging / Documentation Instructor cache results	Instructor: Timer for Quizzes	Server: WebAPI stems -results grab Server: WebAPI refinements -remote query support Traffic Optimization -Server caching	Testing/ Debugging /Documentation Student: Timer for quizzes on student side
Week 13	Testing / Debugging / Documentation	Testing/ Debugging/ User Documentation	Testing/ Debugging/ User Documentation	Testing/ Debugging/ Documentation
Week 14	Testing / Debugging / Documentation / Presentation Prep	Testing/ Prepare for Final Prof. Evaluation and Public Demo	Testing/ Debugging/ User Documentation	Testing /Debugging/ Documentation/ Presentation
Week 15	Public Demo	Public Demo	Public Demo	Public Demo

Appendix A: UI Sketches

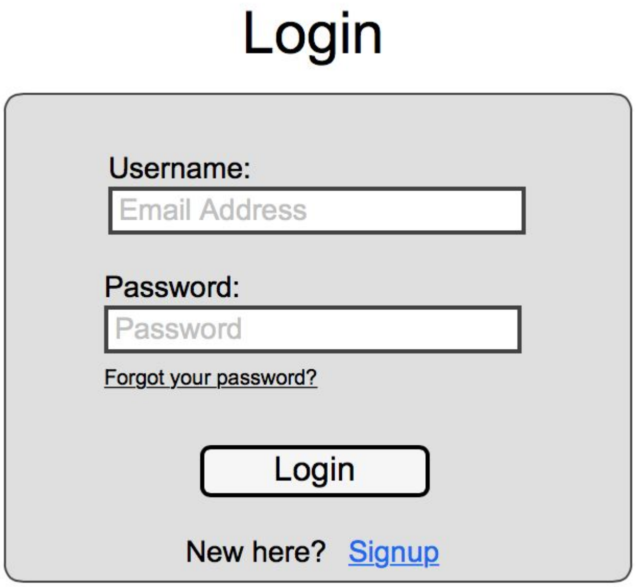
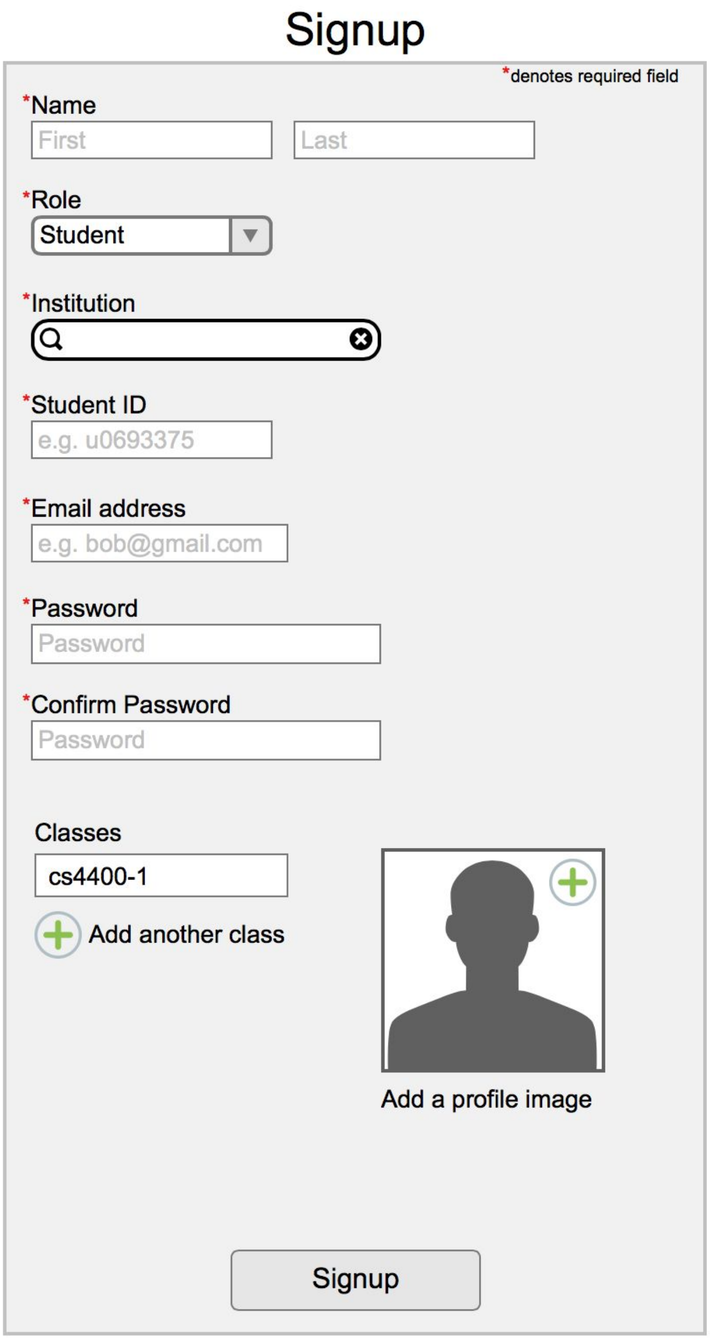


Figure 1.0 - Login Screen



Signup

*denotes required field

*Name

First

Last

*Role

Student ▼

*Institution

*Student ID

e.g. u0693375

*Email address

e.g. bob@gmail.com

*Password

Password

*Confirm Password

Password

Classes

cs4400-1

Figure 1.1 - Registration Screen

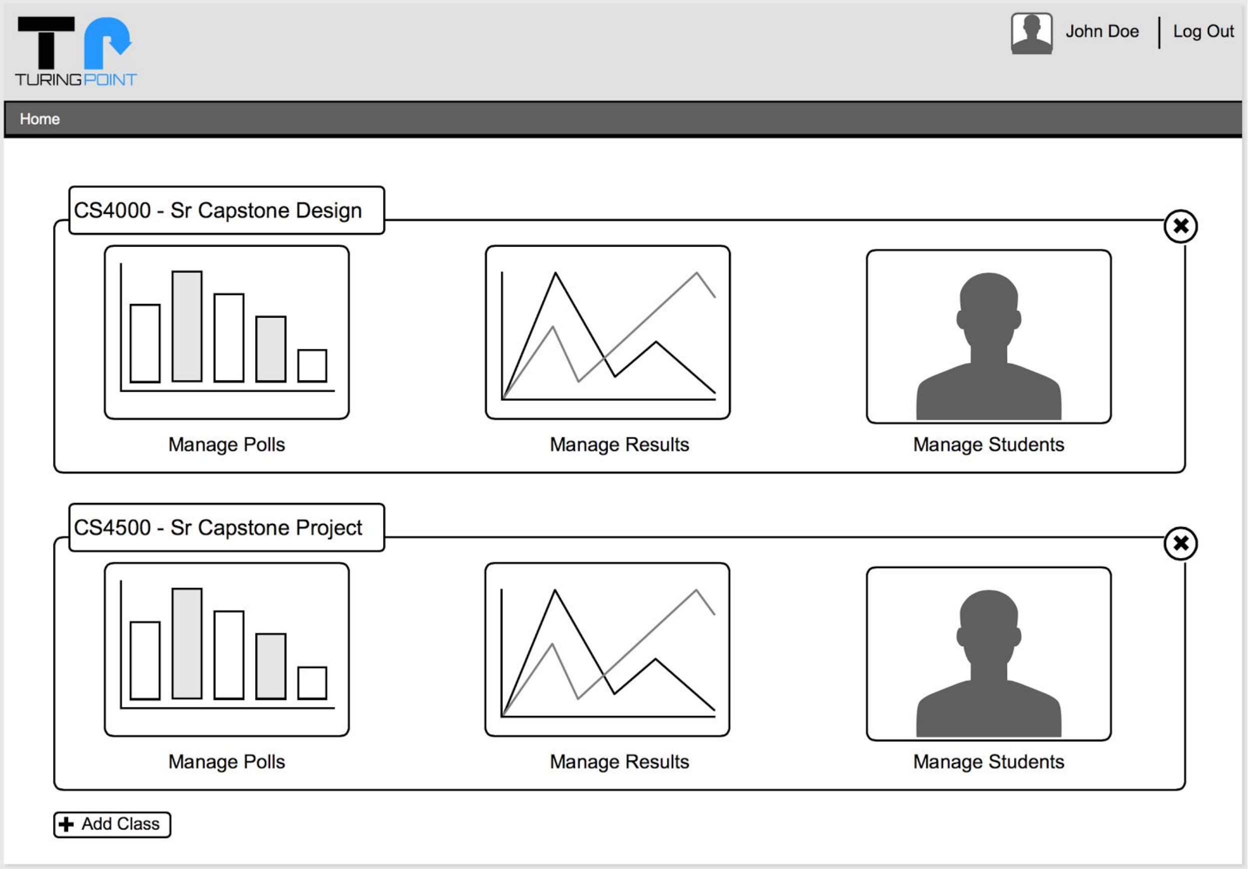


Figure 2.0 - Instructor Portal Home

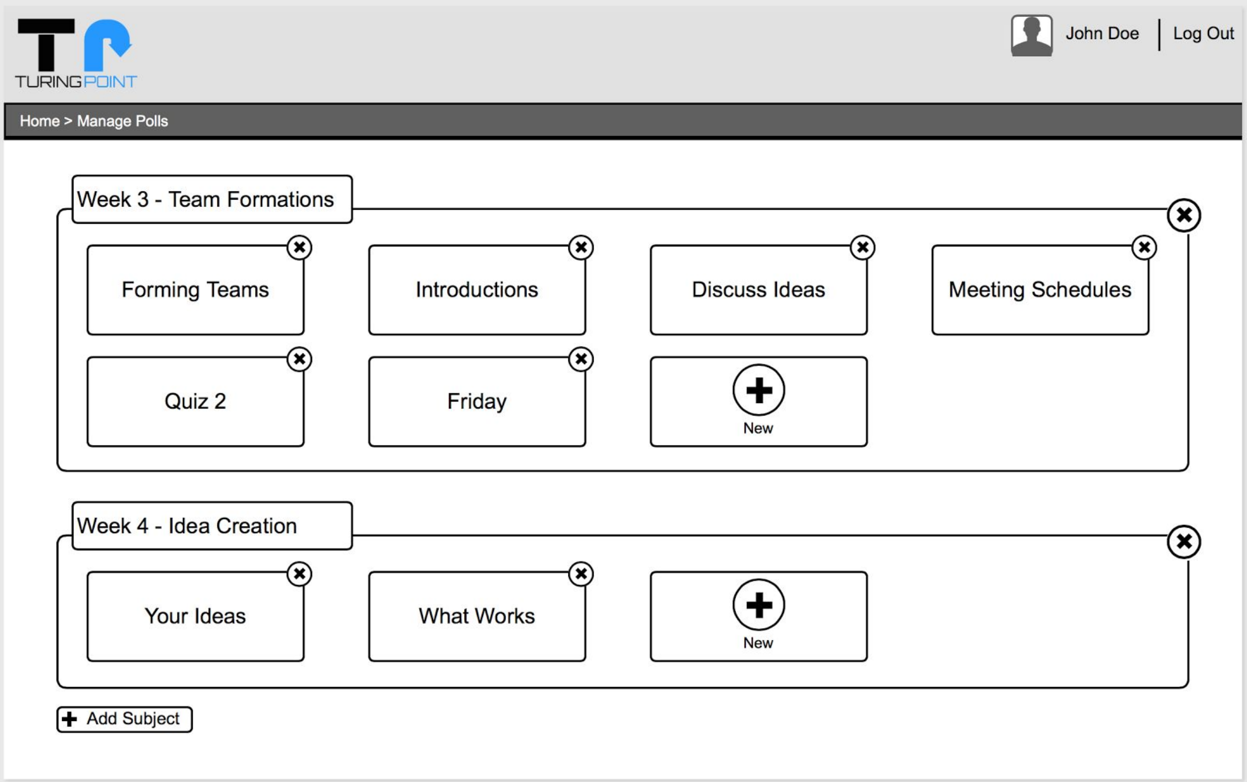
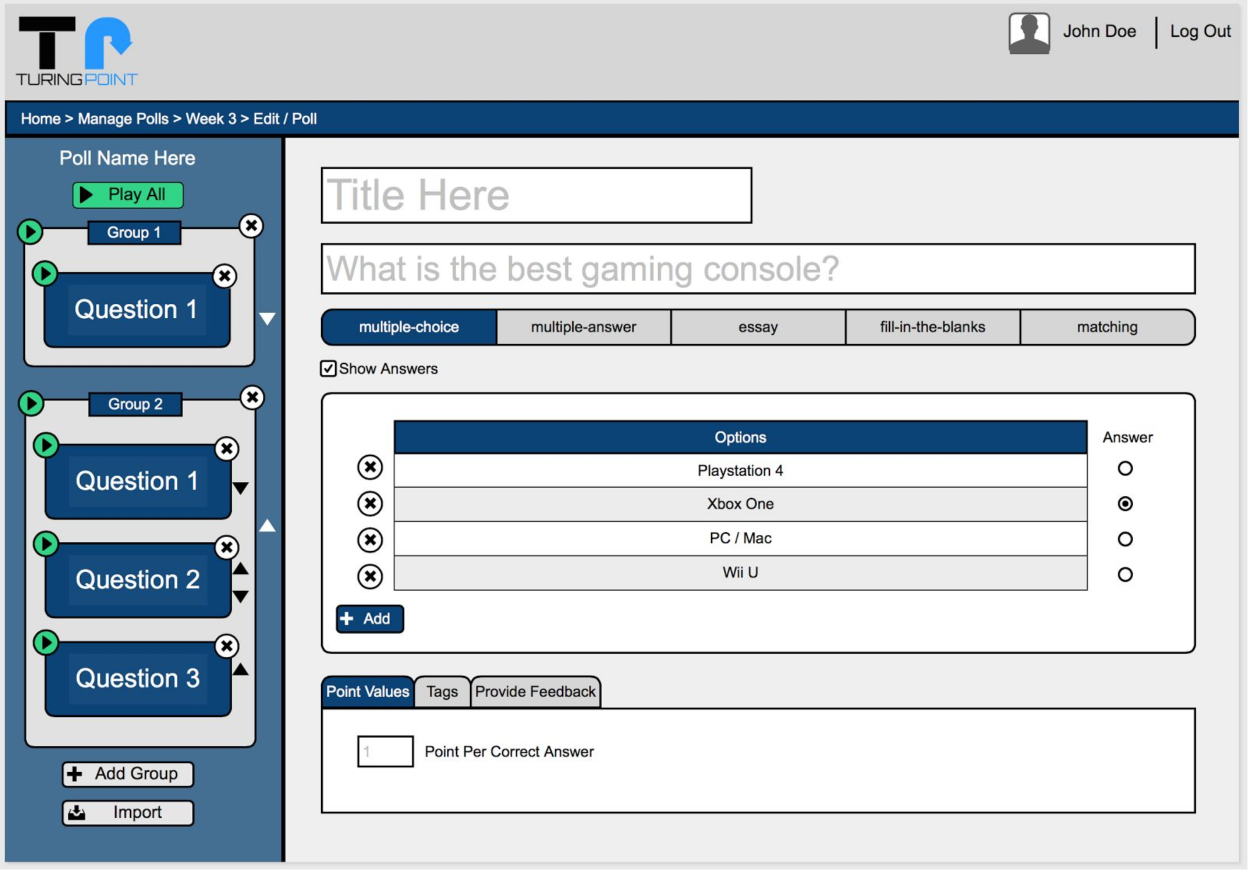


Figure 2.1 - Manage Polls



Title Here

What is the best gaming console?

☑ Show Answers

Options	Answer
Playstation 4	○
Xbox One	●
PC / Mac	○
Wii U	○

1 Point Per Correct Answer

Figure 2.2 - Edit / Play Polls

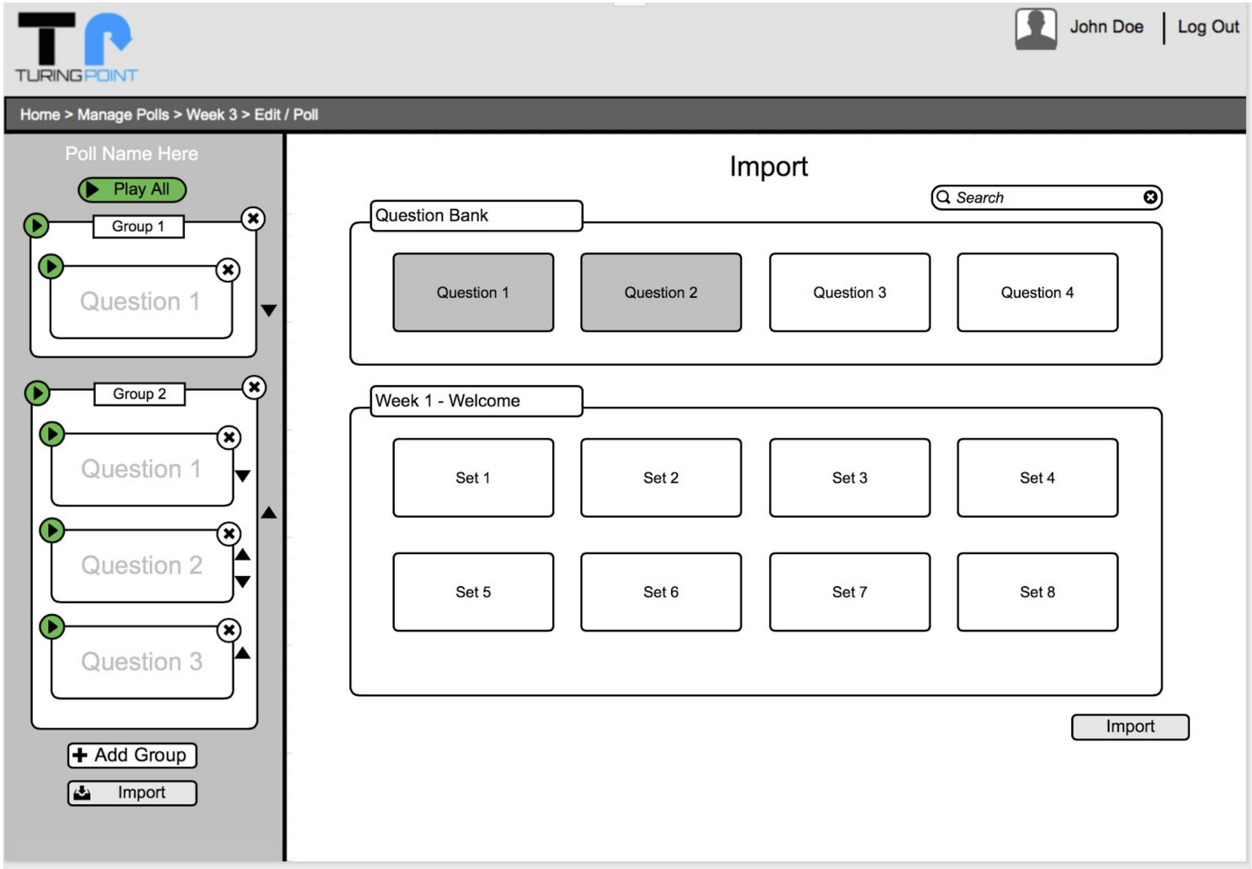
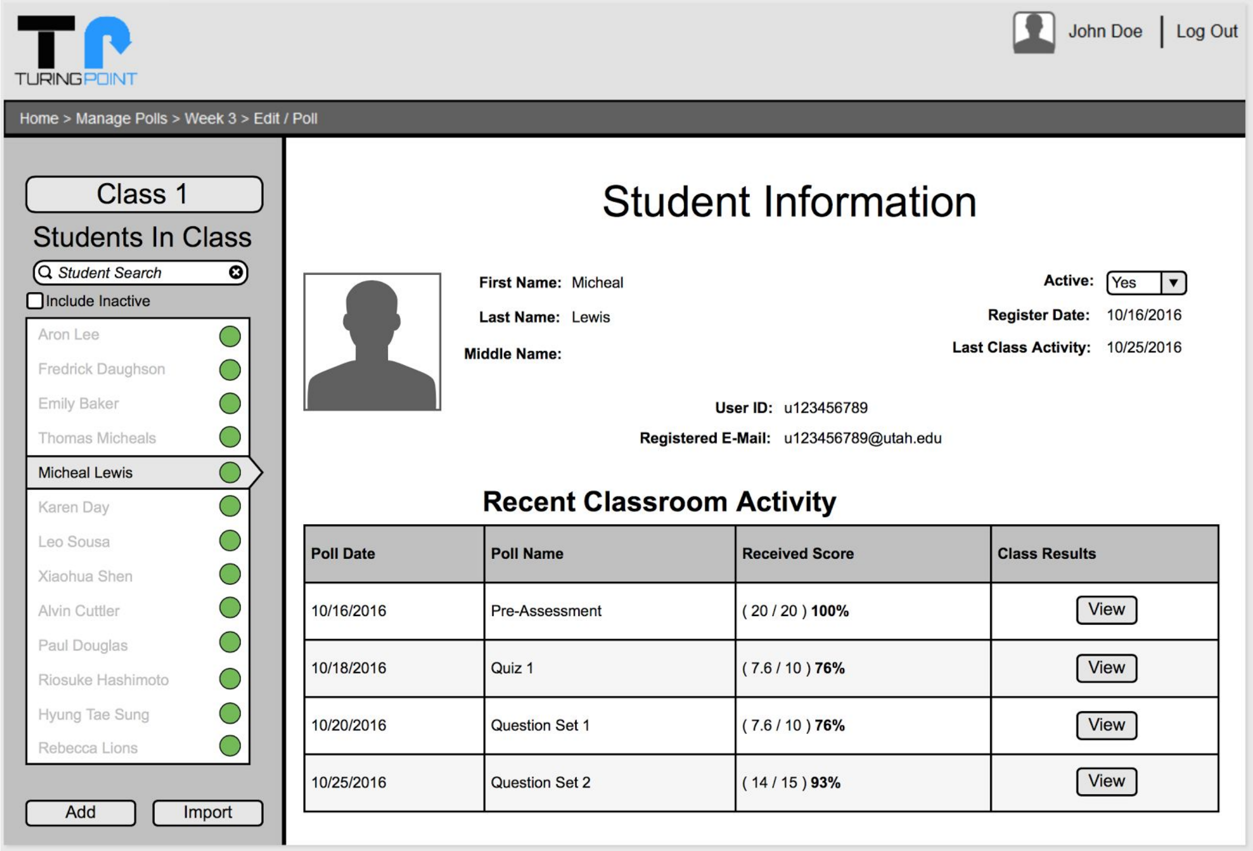


Figure 2.3 - Import Questions



Students In Class

□ Include Inactive

- Aron Lee 🟩
- Fredrick Daughson 🟩
- Emily Baker 🟩
- Thomas Micheals 🟩
- Micheal Lewis 🟩
- Karen Day 🟩
- Leo Sousa 🟩
- Xiaohua Shen 🟩
- Alvin Cuttler 🟩
- Paul Douglas 🟩
- Riosuke Hashimoto 🟩
- Hyung Tae Sung 🟩
- Rebecca Lions 🟩

Student Information

First Name: Micheal

Last Name: Lewis

Middle Name:

Active: Yes ▼

Register Date: 10/16/2016

Last Class Activity: 10/25/2016

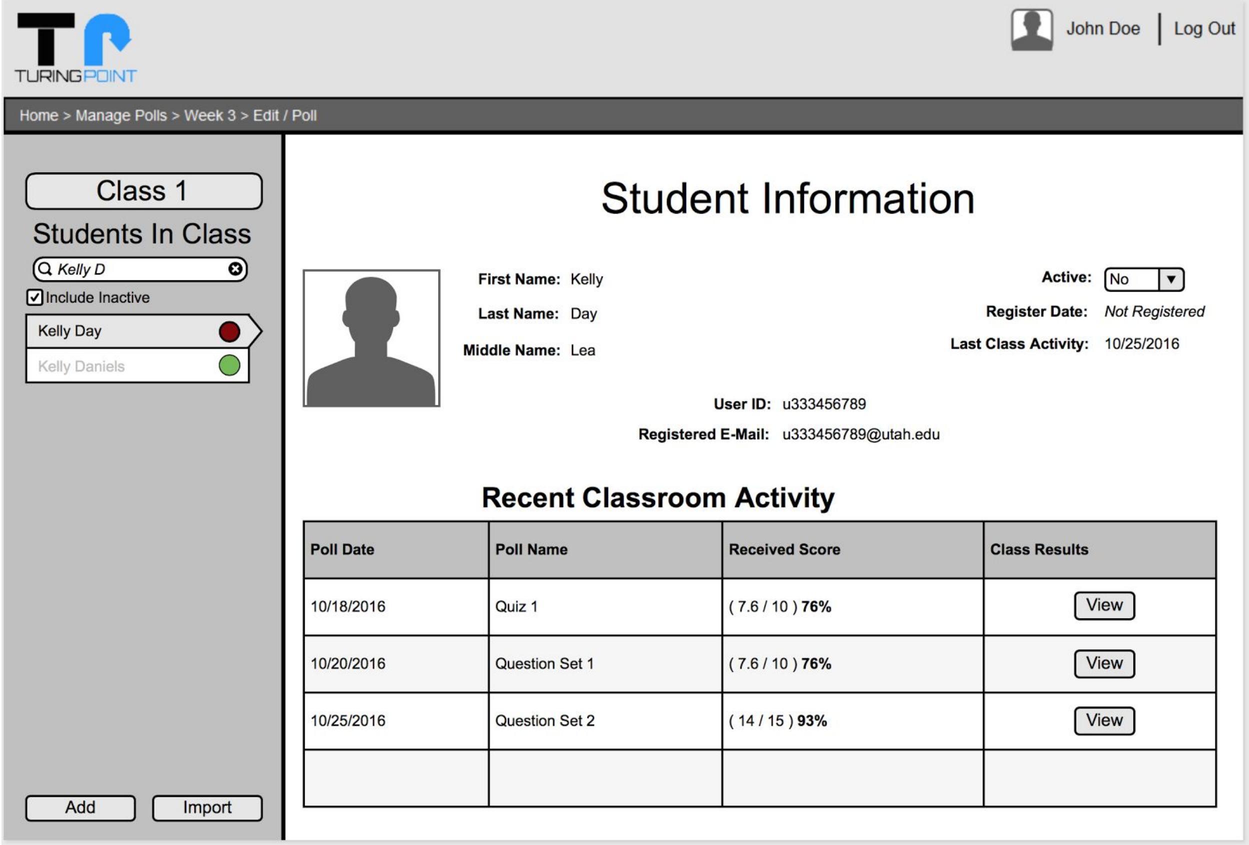
User ID: u123456789

Registered E-Mail: u123456789@utah.edu

Recent Classroom Activity

Poll Date	Poll Name	Received Score	Class Results
10/16/2016	Pre-Assessment	(20 / 20) **100%**	View
10/18/2016	Quiz 1	(7.6 / 10) **76%**	View
10/20/2016	Question Set 1	(7.6 / 10) **76%**	View
10/25/2016	Question Set 2	(14 / 15) **93%**	View

Figure 2.4 - Student Management



Student Information

First Name: Kelly
Last Name: Day
Middle Name: Lea

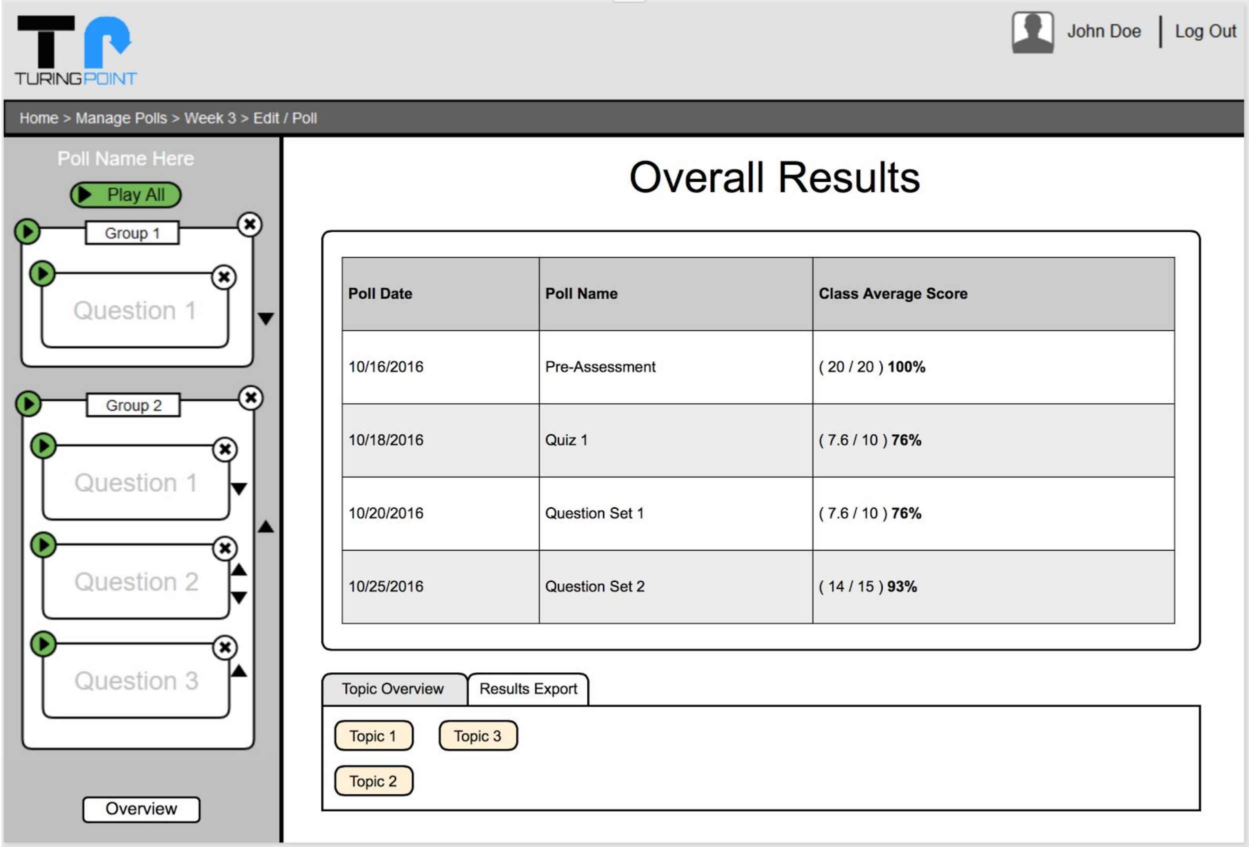
Active: No ▼
Register Date: *Not Registered*
Last Class Activity: 10/25/2016

User ID: u333456789
Registered E-Mail: u333456789@utah.edu

Recent Classroom Activity

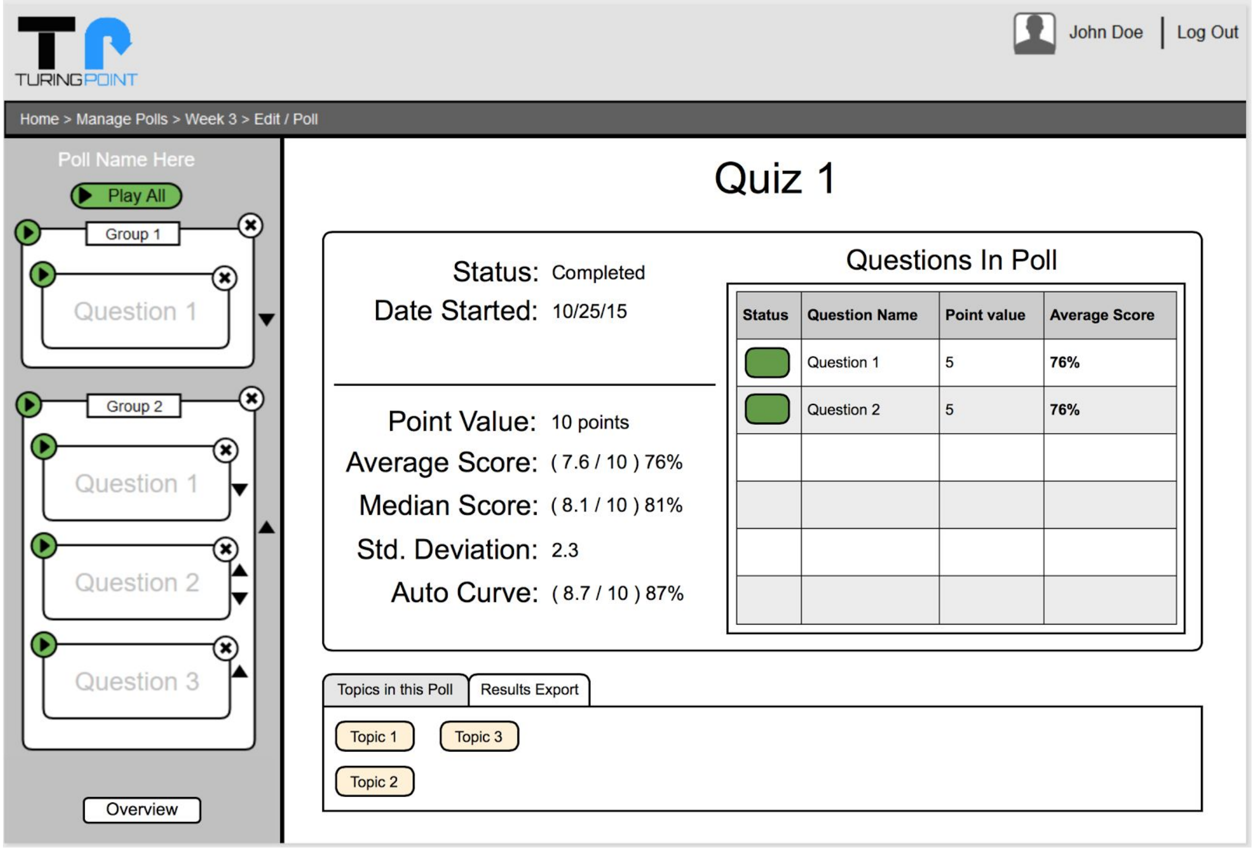
Poll Date	Poll Name	Received Score	Class Results
10/18/2016	Quiz 1	(7.6 / 10) **76%**	View
10/20/2016	Question Set 1	(7.6 / 10) **76%**	View
10/25/2016	Question Set 2	(14 / 15) **93%**	View

Figure 2.4.1 - Student Management - Inactive Students



Poll Date	Poll Name	Class Average Score
10/16/2016	Pre-Assessment	(20 / 20) **100%**
10/18/2016	Quiz 1	(7.6 / 10) **76%**
10/20/2016	Question Set 1	(7.6 / 10) **76%**
10/25/2016	Question Set 2	(14 / 15) **93%**

Figure 2.5 - Mange Results



Quiz 1

Status: Completed

Date Started: 10/25/15

Point Value: 10 points

Average Score: (7.6 / 10) 76%

Median Score: (8.1 / 10) 81%

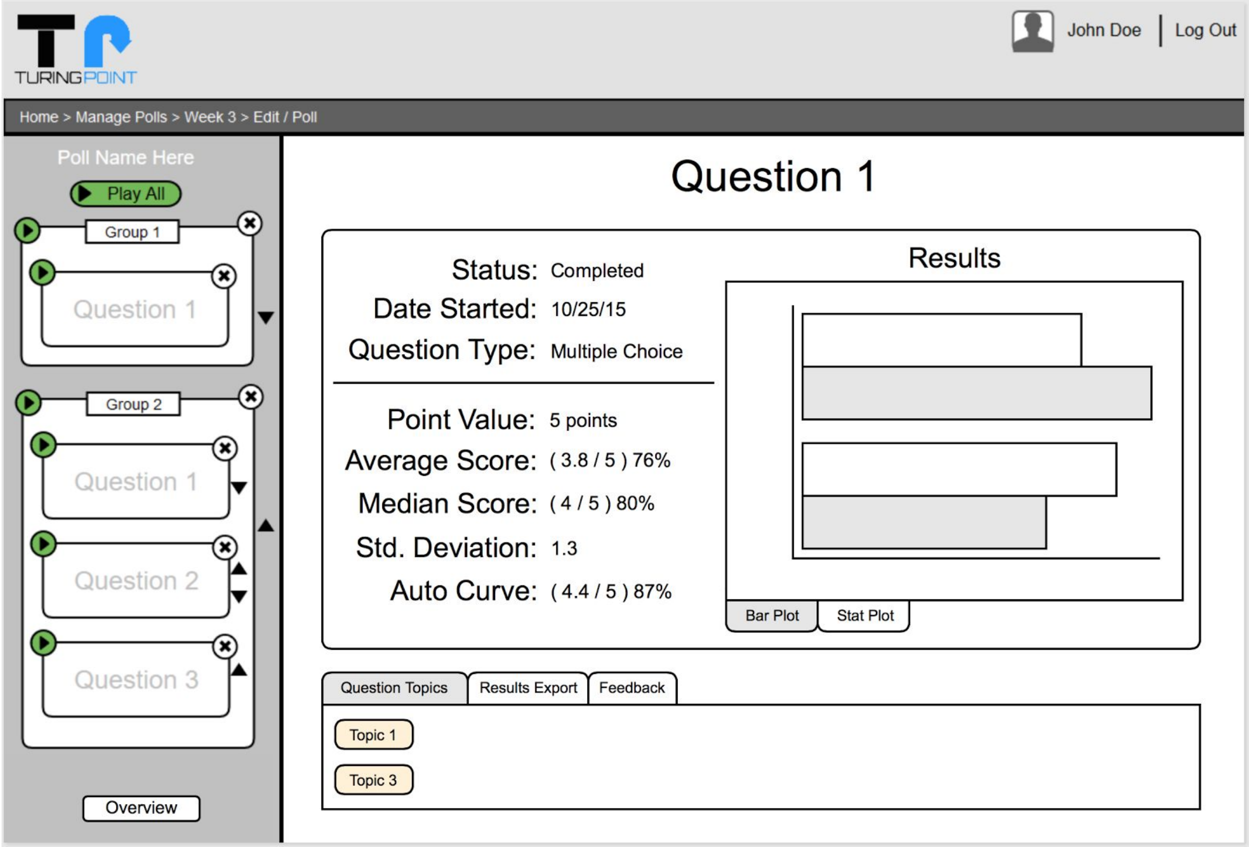
Std. Deviation: 2.3

Auto Curve: (8.7 / 10) 87%

Questions In Poll

Status	Question Name	Point value	Average Score
🟩	Question 1	5	**76%**
🟩	Question 2	5	**76%**

Figure 2.6 - Results Management - Question Set



Question 1

Status: Completed

Date Started: 10/25/15

Question Type: Multiple Choice

Point Value: 5 points

Average Score: (3.8 / 5) 76%

Median Score: (4 / 5) 80%

Std. Deviation: 1.3

Auto Curve: (4.4 / 5) 87%

Figure 2.6.1 - Results Management - Individual Question

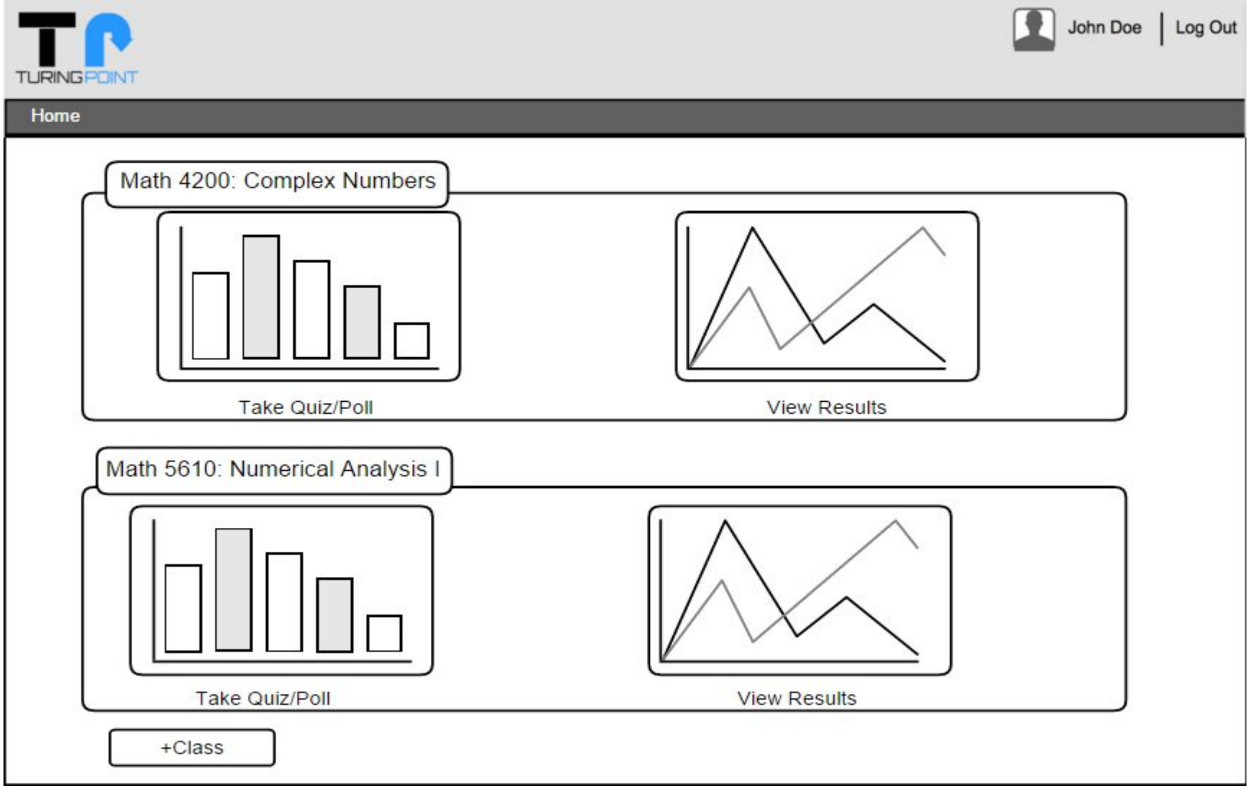


Figure 3.0 - Student Portal home

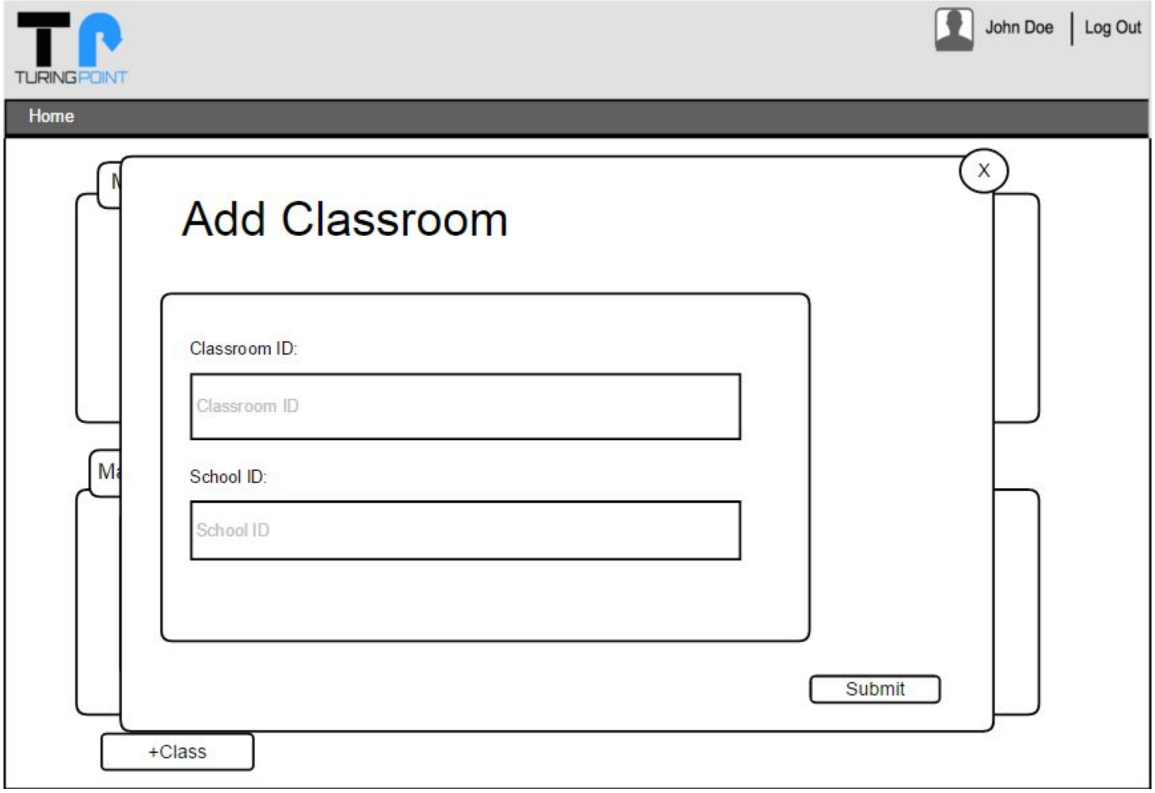
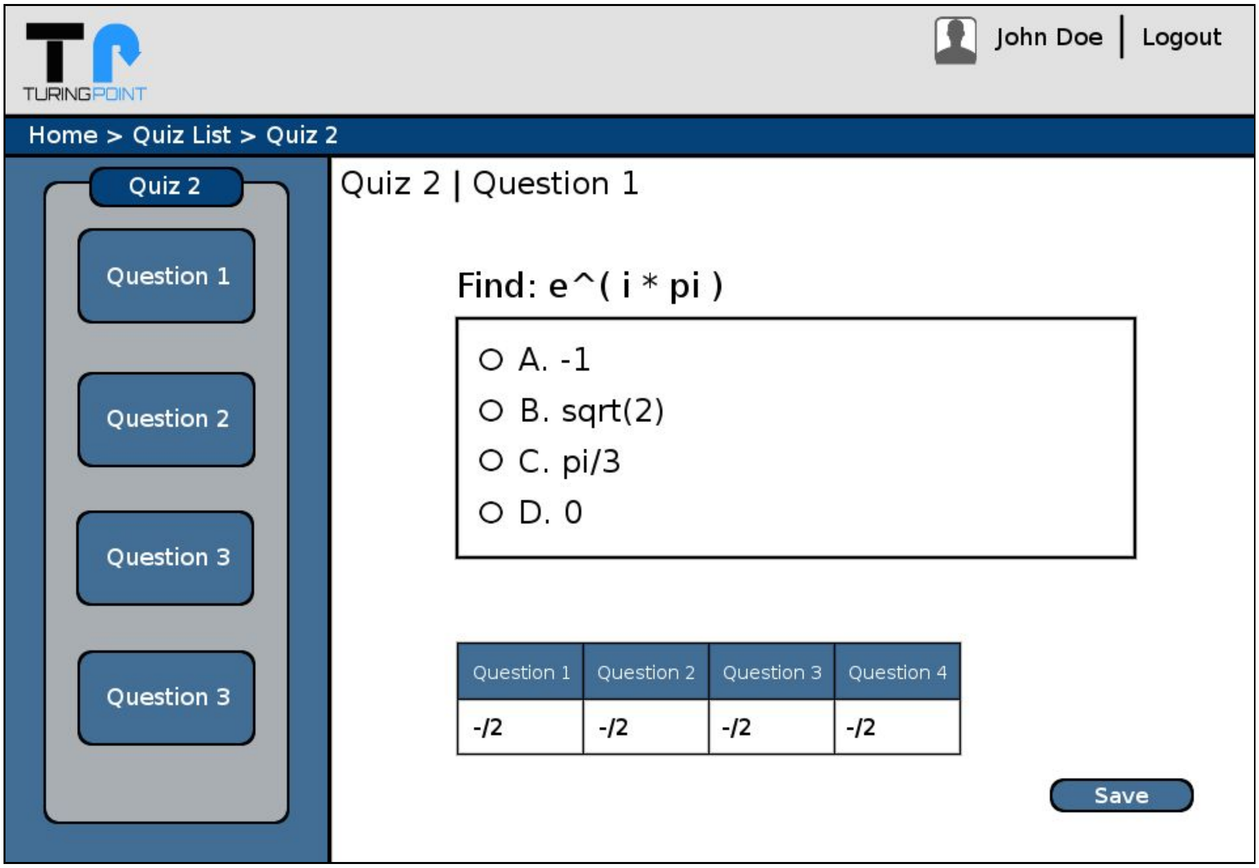


Figure 3.1 - Add Classroom



Find: e^(i * pi)

- ○ A. -1
- ○ B. sqrt(2)
- ○ C. pi/3
- ○ D. 0

Question 1	Question 2	Question 3	Question 4
-/2	-/2	-/2	-/2

Figure 3.2 - Take Quiz

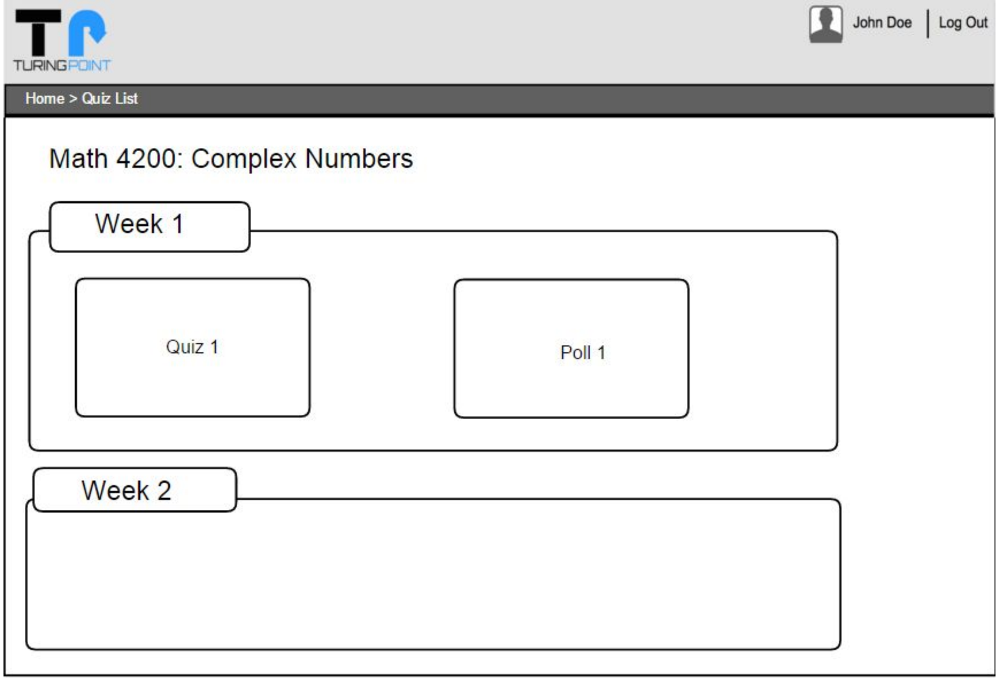
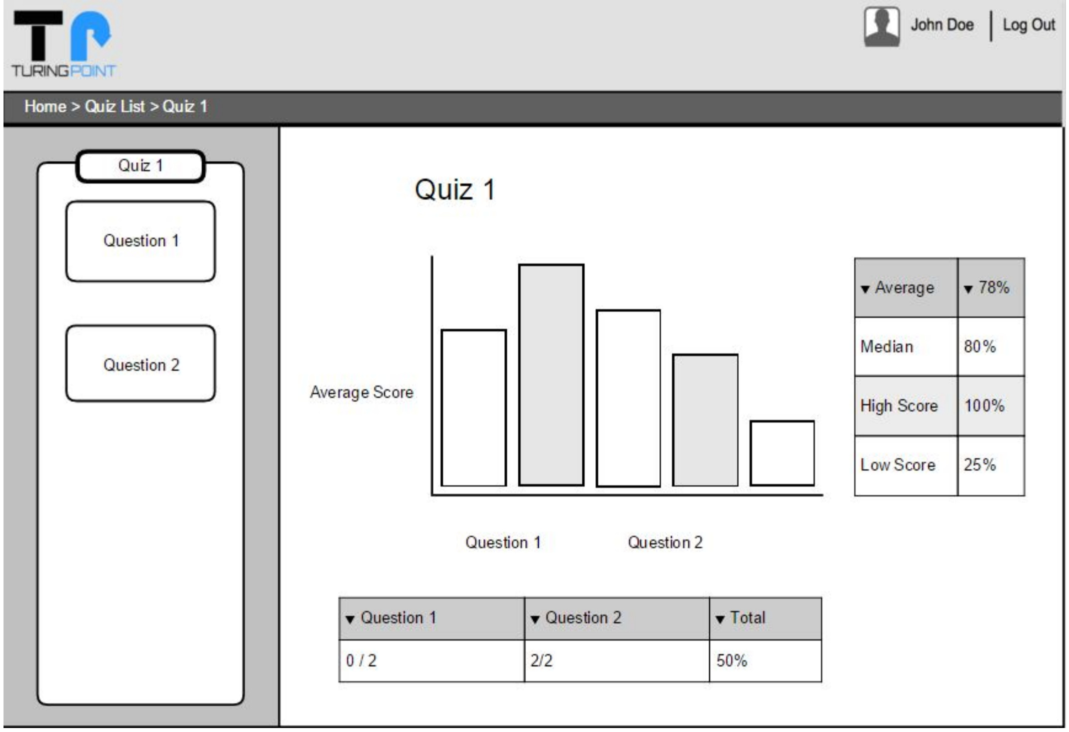


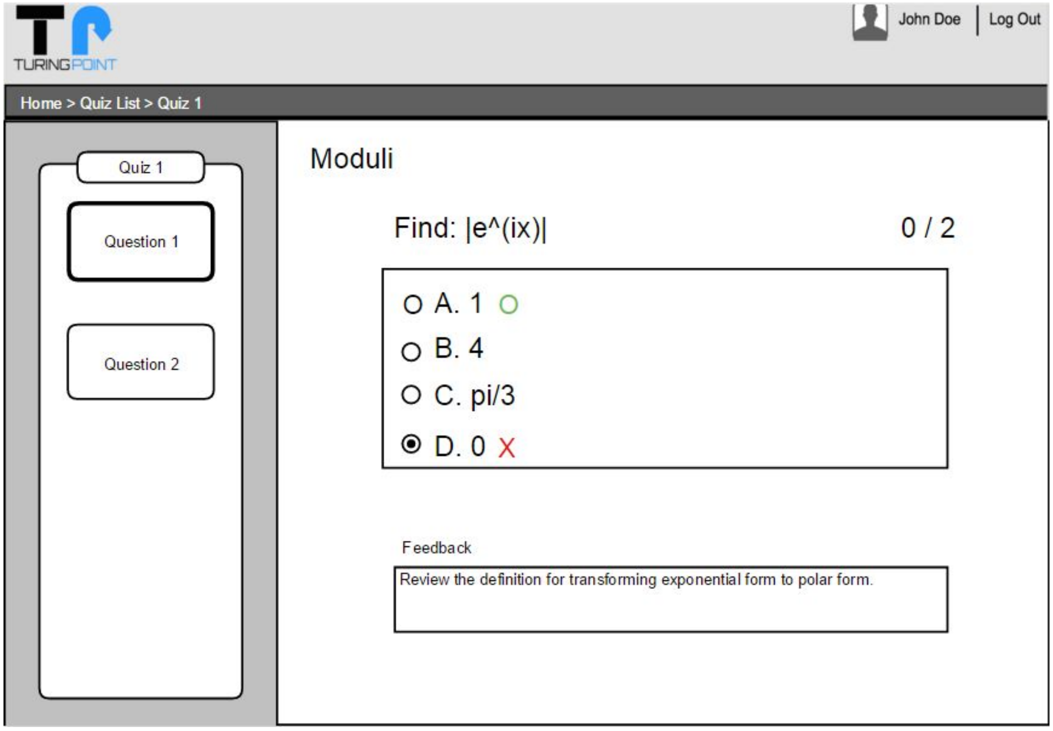
Figure 3.3 - View Previous Quiz



▾ Average	▾ 78%
Median	80%
High Score	100%
Low Score	25%

▾ Question 1	▾ Question 2	▾ Total
0 / 2	2/2	50%

Figure 3.4 - Quiz Result Overview



Moduli

Find: |e^(ix)| 0 / 2

○ A. 1 ○

○ B. 4

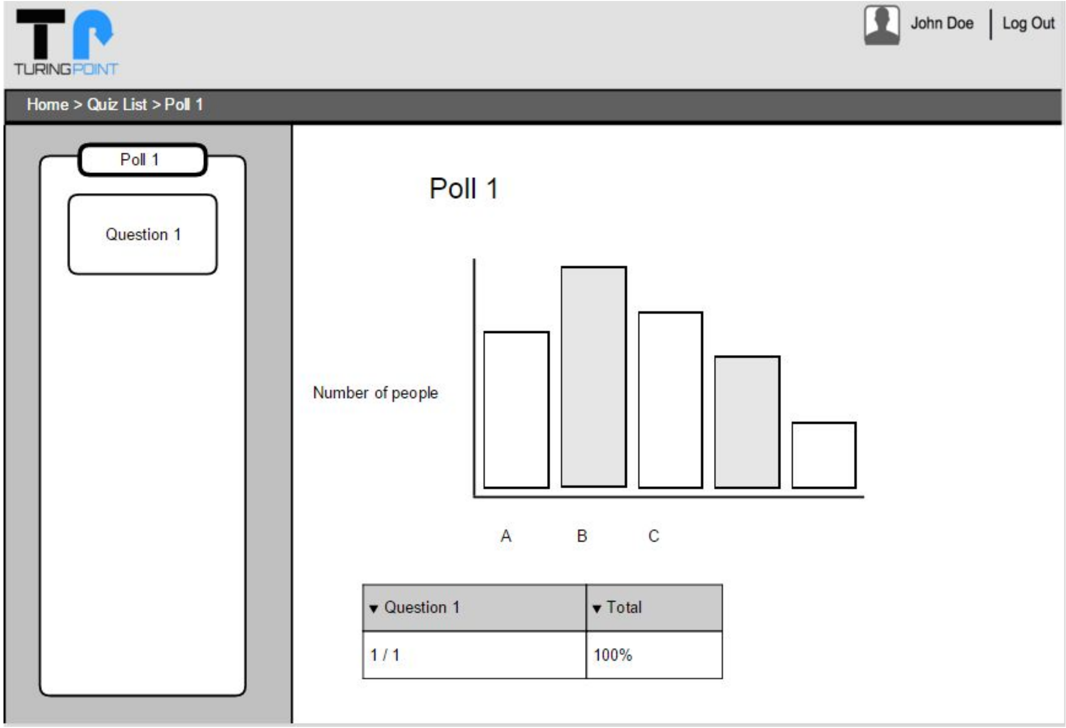
○ C. pi/3

● D. 0 X

Feedback

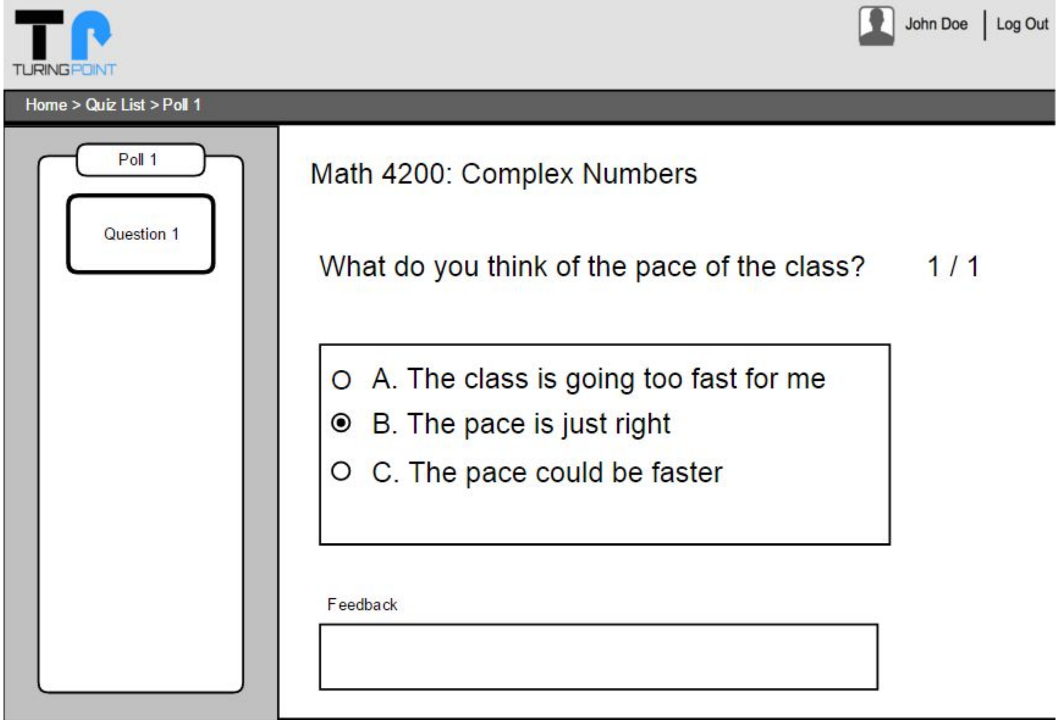
Review the definition for transforming exponential form to polar form.

Figure 3.4.1 - Quiz Question Review



▾ Question 1	▾ Total
1 / 1	100%

Figure 3.5 - Poll Result Overview



Math 4200: Complex Numbers

What do you think of the pace of the class? 1 / 1

○ A. The class is going too fast for me
● B. The pace is just right
○ C. The pace could be faster

Feedback

Figure 3.5.1 - Poll Question Review

Appendix B: Use Cases

Instructor and Student Use Cases

1. **Login**
   Description: The user logs into the system, this is the entry point for the application.
   Precondition: The user has already created an account for the system. The user is presented with the Login Screen (**Figure 1.0**).
   a. User will enter their email address.
   b. User will enter their password.
   c. User clicks login.
      i. Successful logins bring the user to the instructor portal.
      ii. Unsuccessful logins return the user to the login screen with an error message.
2. **Create a poll**
   Description: User wants to create a new poll set.
   Precondition: The user is logged in and on the Instructor Homepage (**Figure 2.0**).
   a. User clicks the “Manage Polls” button.

b. User is presented with the Manage Polls page (**Figure 2.1**).
c. User optionally clicks “Add Subject” button to create a new subject.
d. Clicks the “New” button within a subject.
e. Enters a poll name in a popup dialog.
f. Clicks “Create Poll” button within the dialog.
g. Is presented with the Edit / Poll page (**Figure 2.2**) where they begin creating questions.

Instructor Use Cases

1. **Create a question**
   Description: User wants to create a poll question.
   Precondition: The user is logged in, a poll set has already been created, and the user is on the Edit / Play Polls page (**Figure 2.2**).
   a. User clicks the “Add Group” button to create a new group of questions.
   b. There will be one question by default per group.
   c. User fills out the title, selects a question type, fill in the question details, and question answer(s).
   d. User can optionally add tags, change the point values for correct answers or add a feedback response displayed after the question is answered by students.
2. **Change/Assign point value for question**
   Description: To better facilitate the instructor control over grading of polls, instructors can assign differing point values on a per question basis or retain the default value.
   Precondition: The user is logged in , has created a question following the “Create a question” use case, and is on the Edit / Play polls Screen (**Figure 2.2**).
   a. Selects the 'point value' tab.
   b. Enters a point value into the number box.
3. **Change question type**
   Description: To allow more effective polling, the system supports multiple question types. The user can select between them freely at the time of question creation.
   Precondition: The user is logged in, has created a poll and question, and is on the Edit / Play polls Screen (**Figure 2.2**).
   a. User selects the question type from the button bank in the middle of the screen.
   b. User fills out the question type details for the new question type.
4. **Start/stop question-set**
   Description: The user can use the same interface in which they edit or modify and create questions to play them.
   Precondition: The user is logged in with a valid account and has created a class in the system, after having selected a subject from the list of available subjects. They are viewing Edit / Play polls Screen (**Figure 2.2**).
   a. Instructor may play an individual question by clicking the green play button on that question.

   b. Instructor may play an entire set of questions by clicking the green play button on the set.
   c. Instructor may also play all questions in the selected group of questions by clicking 'Play All' near the top of the list.
   d. Instructor stops any question or set by clicking the toggled play button.

5. **Create Account**
   Description: A user wants to register and use our system.
   Preconditions: User is on the login screen (**Figure 1.0**).
   a. Click the ‘Signup’ button from the login or home page.
   b. Is directed to the Registration Screen (**Figure 1.1**).
   c. Enters required information:
      i. Name
      ii. Role (Student or Instructor)
      iii. Institution/University/College
      iv. Student ID
      v. Email Address
      vi. Password
   d. Enters optional info, if desired:
      i. Add a class
      ii. Add a profile photo
   e. Clicks ‘Signup’ button to submit their information and redirects them appropriately.
      i. If successful, stores their info and redirects them to the student or instructor home page/dashboard.
      ii. If unsuccessful, prompts the user to correct/modify necessary fields:
         1. Email already in use
         2. Invalid email
         3. Student ID already in use
         4. Invalid student ID
         5. Did not complete required fields
         6. Passwords did not match
         7. Password does not meet length/charset requirements
6. **Navigate questions in a question-set**
   Description: Instructor wants to go to a different question on a question set.
   Preconditions: Instructor is on the Edit/Play page (**Figure 2.2**) for a poll that has question-sets with questions.
   a. Instructor clicks on the question they want to view or edit.
   b. The view pane updates to show relevant info for that question.
7. **Add/remove question to existing question-set**
   Description: Instructor wants to add/remove a question in a question-set.
   Preconditions: Instructor is logged in and is currently on the Edit/Play page (**Figure 2.2**) for a poll with an existing question-set.

    a. To add a question the instructor can click the “Add” button within a question set, on the navigation pane.
    b. A new question is populated that they can modify.
    c. To remove a question they can click the “X” button on the right of the question that they want to remove (also on the navigation pane).

8. **Recall existing poll**
   Description: Instructor wants to view/edit a poll that they already created.
   Preconditions: Instructor has created a poll and is on the Instructor Portal Home Page (Figure 2.0).
    a. Instructor clicks “Manage Polls” button.
    b. Instructor locates the desired subject, and then clicks the desired poll to view/edit.
    c. Instructor is taken to the view/edit page for that poll (**Figure 2.2**).
9. **Re-arrange questions in question-set**
   Description: Instructor wants to change the position of a question in a particular question-set.
   Preconditions: Instructor is logged in and is on the Edit/Play page (**Figure 2.2**).
    a. The instructor can change the order of questions by clicking the “up” or “down” arrows located next to the question or question group that they wish to change the order of.
    b. Alternatively, they can drag questions or question set up or down to change their order.
10. **Add/Remove a saved question-set**
   Description: Instructor wants to add or remove an existing question-set.
   Preconditions: Instructor is logged in and on the Edit/Play page (**Figure 2.2**) for a grouping that contains the question-set they wish to remove.
    a. The instructor can remove a question-set by clicking the “X” button to the right of the question-set that they wish to remove.
    b. To add a question-set they can click the “Add Group” button near the bottom of the navigation pane.
11. **View poll results (current)**
   Description: The instructor views the results of a poll that was just conducted.
   Preconditions: The instructor has created questions and is on the question editing page (**Figure 2.2**).
    a. The instructor clicks “play all”, or clicks an individual question they wish to be active (i.e. available for polling).
    b. The students participate in the poll.
    c. The instructor clicks the “stop all” button (a toggle of the “play all” button), or stops an individual active question.
    d. The poll results are displayed in the main windows of the poll edit page.
12. **View poll results (Existing)**
   Description: The instructor views the results of poll(s) that were previously conducted.
   Preconditions: Is logged in and on the Instructor’s homepage (**Figure 2.0**).

a. From the Instructor’s Homepage clicks “Manage Results”.
b. Instructor is navigated to the Manage Results page (**Figure 2.5**).
c. Instructor clicks on desired poll question on left navigation.
d. Statistics about the question’s results are displayed in the main window.

13. Edit question content

Description: The instructor edits a question that was previously created.
Preconditions: Is logged in and on the Instructor’s homepage (**Figure 2.0**).

a. Clicks “Manage Polls” button on desired class.
b. Clicks the desired poll set to modify.
c. Clicks the desired question to modify.
d. Edits the desired field(s).
e. Clicks save.

14. Visualize performance data from a question-set

Description: The instructor views visual results from question sets.
Preconditions: Is logged in and on the “Manage Results” page (**Figure 2.5**).

a. The instructor clicks the desired poll set from left navigation.
b. Graphical charts are presented on main window.
c. Instructor clicks various chart types.

15. Export question-set results

Description: The instructor exports question results sets.
Preconditions: Is logged in and on the “Manage Results” page (**Figure 2.5**).

a. The instructor clicks the desired poll set from left navigation.
b. Clicks “Results Export” tab.
c. Choses export format, clicks “Export all”.

16. Create Course

Description: The instructor creates a new course to conduct polling upon.
Preconditions: Is logged in and on the Instructor’s Homepage (**Figure 2.0**).

a. The instructor clicks the “Add Class” button.
b. A form covers the screen.
c. Fills out the course name.
d. Fills out an optional course description.
e. Clicks “Add Class” button within form.

17. Deletes Course

Description: The instructor deletes a course and all its results.
Preconditions: Is logged in and on the Instructor’s Homepage (**Figure 2.0**).

a. The instructor clicks the “X” button on the course desired to be deleted.
b. Clicks “Yes” on confirmation popup.

18. Add Student to Classroom

Description: The instructor adds a student to a classroom.
Preconditions: Is logged in and on the Instructor’s Homepage (**Figure 2.0**).

a. Clicks “Manage Students” on desired class.
b. Clicks “Add” on student management page(**Figure 2.4**).
c. Fills out student’s information on main page.

    d. Clicks “Save”.

19. **Switch between classrooms**

    Description: The instructor switch and manage a different classroom.

    Preconditions: Is logged in.

    a. Clicks “Home” link in header navigation to go to homepage.
    b. Clicks desired link on new classroom from homepage.

20. **Reuse existing question**

    Description: The instructor adds a question from a previous poll, or the question bank, to the current poll set.

    Preconditions: Is logged in and on the Poll Editing page (**Figure 2.2**).

    a. Clicks “Import” button.
    b. Selects desired questions from lists (see **Figure 2.3**).
    c. Clicks “Import” button within the list of questions.
    d. The imported questions appear in the list of questions.

21. **Add question to question bank**

    Description: The Instructor created a question and does not want to use it immediately, but they will use it at a future date. Therefore, they will save the question to the question bank.

    Precondition: The instructor has created a question and is on the question management screen (**Figure 2.2**).

    a. Give the questions some tags by clicking on the “tag” tab.
    b. After entering some tab, click on the export button.

22. **Move a question-set from an existing poll to a new poll**

    Description: Instructor wants to move a question-set they already made to a poll that they’re working on.

    Preconditions: Instructor has created a question-set with questions and is now on the Instructor Portal Home page (**Figure 2.0**).

    a. Clicks “Manage Polls”.
    b. Drags and drops a question-set into the poll that they want.

23. **View poll results**

    Description: Instructor wants to view existing poll results to see how students are performing.

    Preconditions: Instructor navigates to “Manage Results” page (**Figure 2.5**) for a particular class.

    a. Instructor selects a subject they wish to review results for.
        i. Instructor is presented with an overview of all results from the selected subject.
        ii. OR Instructor selects question set to view overall results for question set and specific statistics about it.
        iii. OR Instructor selects questions individually to view results for a single question and specific statistics about it.

24. **Student management**

    Description: The instructor wants to add, remove or view info about a student.

Precondition: They've navigated to the Student Management page (**Figure 2.4**).

   a. To add a single student they can click the "Add" button and follow the prompt.
   b. To add, say a whole class list of students, the instructor can click the "Import" button.
   c. To view info about a student, the instructor finds the student they're interested in by manually looking through the list or using the search filter.
      i. To include inactive students in the search the instructor can check the "Include Inactive" checkbox (see **Figure 2.4.1**).
   d. Instructor clicks on the student.
   e. View pane updates to show student info and recent activity.

25. **Results management**

Description: Instructor wants to review the results for a particular quiz or quiz question.

Precondition: Currently on the Results Management page (**Figure 2.6**).

   a. If the instructor wants to see the results for the entire quiz then they can click the "Overview" button.
   b. The view pane will update to show the results for the quiz (see **Figure 2.6**).
      i. The instructor can see tags for the quiz and export results.
   c. To view the results for a particular question the instructor simply clicks on the desired question.
   d. The view pane will update to show the results for the question (see **Figure 2.6.1**).
      i. From here the instructor can see question tags, feedback and export results.

26. **Prepare for midterm**

Description: Instructor wants to enhance classroom learning by utilizing Turing Point software. He/she has a midterm planned next week and wants to gauge the students' readiness during the next lecture.

Preconditions: Has a computer or mobile device with an internet browser.

   a. The instructor registers for an account (Instructor case #5) from our website home page.
   b. Instructor logs onto the system (case #1).
   c. Instructor creates a new course (Instructor case #16) to save their polls/quizzes in.
   d. Instructor creates a new poll (case #2) with questions that are similar to those that might be on the midterm.
   e. Instructor adds all of the students in his class to the course he/she just created (Instructor case #18) or ask them to each individually add the class.
   f. During their lecture the instructor can "play" the questions they want students to answer and "stop" them when they're finished, at any time (Instructor case #4).
   g. The instructor can then review the results (Instructor cases #24, #15) with the class or by themselves after lecture to identify if students are struggling with any topic(s).

h. The instructor can create a new poll (case #2), edit the previous one (Instructor case #13) or import the same question to a new lecture (Instructor case #20) to test if the students are ready, repeating the above process.

i. Instructor can administer the midterm knowing that students should be readily prepared.

Student Use Cases

1. **Join a classroom**

   Description: The student wish to join a classroom.

   Precondition: The student is logged in.

   a. The student will click on the “add classroom” button on the student homepage (**Figure 3.0**).

   b. The student will then be asked to input the class ID (see **Figure 3.1**).

   c. After clicking the submit button, the class will be added.

2. **Take a quiz**

   Description: The professor has started a quiz, which the user wish to take a part of.

   Precondition: The user has joined a classroom and is at the student homepage (**Figure 3.0**).

   a. The user will click on the quiz button to open the available quiz.

      i. If quiz is available, the user will be taken directly to the quiz.

      ii. If the quiz is not available, the user will be notified that there is no open quiz session.

   b. The user will see a quiz question and will answer the question depending on its format (see **Figure 3.2**).

      i. If the question is multiple choice, the student will select a choice.

      ii. If the question is essay question, the student will answer by typing in the text box.

   c. The student will click on the save button to save the progress of the quiz.

   d. The student will be notified that their progress has been saved.

   e. The quiz will end when the proctor/professor closes the quiz. The student can check their quiz result through use case 5.

      i. The student may leave the quiz early, by leaving the page through the menu shortcut.

3. **Review old session**

   Description: The student wish to review a previous quiz session.

   Precondition: There exist a previous quiz session, and the student is at the student homepage (**Figure 3.0**).

   a. The student will click on “view Result” for a selected classroom. This will let the student view all quizzes that the student have taken for that specific class.

   b. The student will see the quizzes organized by week and when the quiz was opened/closed. The student will select the desired quiz they wish to review (see **Figure 3.3**).
   c. The student will first see the overview of the quiz. This overview will show the student's and class's performance on the quiz. (see **Figures 3.4** and **3.5**).
   d. If the student wish to see a question in specific, they can select the question from the left side of the screen (see **Figures 3.4.1** and **3.5.1**).
   e. Selecting a question, the student will be able to see their final submitted answers, and the correct answers. The student can also view any feedback that the professor inserted for the question.

4. **View feedback on questions**
   Description: The user wishes to see the feedback a professor have given for a question.
   Precondition: The student is reviewing an old session (see **Figures 3.4** and **3.5**).
   a. The student will select a question.
   b. At the bottom of the question screen, the student will see any available feedback the professor has given for the specific question (see **Figures 3.4.1** and **3.5.1**).
5. **Switch between classes**
   Description: The user wish to change from a current classroom to another classroom.
   Precondition: The student is logged in and active in a classroom. "Active in a classroom" is defined that the student is taking a quiz or viewing previous quiz results.
   a. The user will click on the home button on the breadcrumb menu to return to the student homepage.
   b. The user will scroll down and select another class that is listed (**Figure 3.0**).

Appendix C: Diagrams and Figures

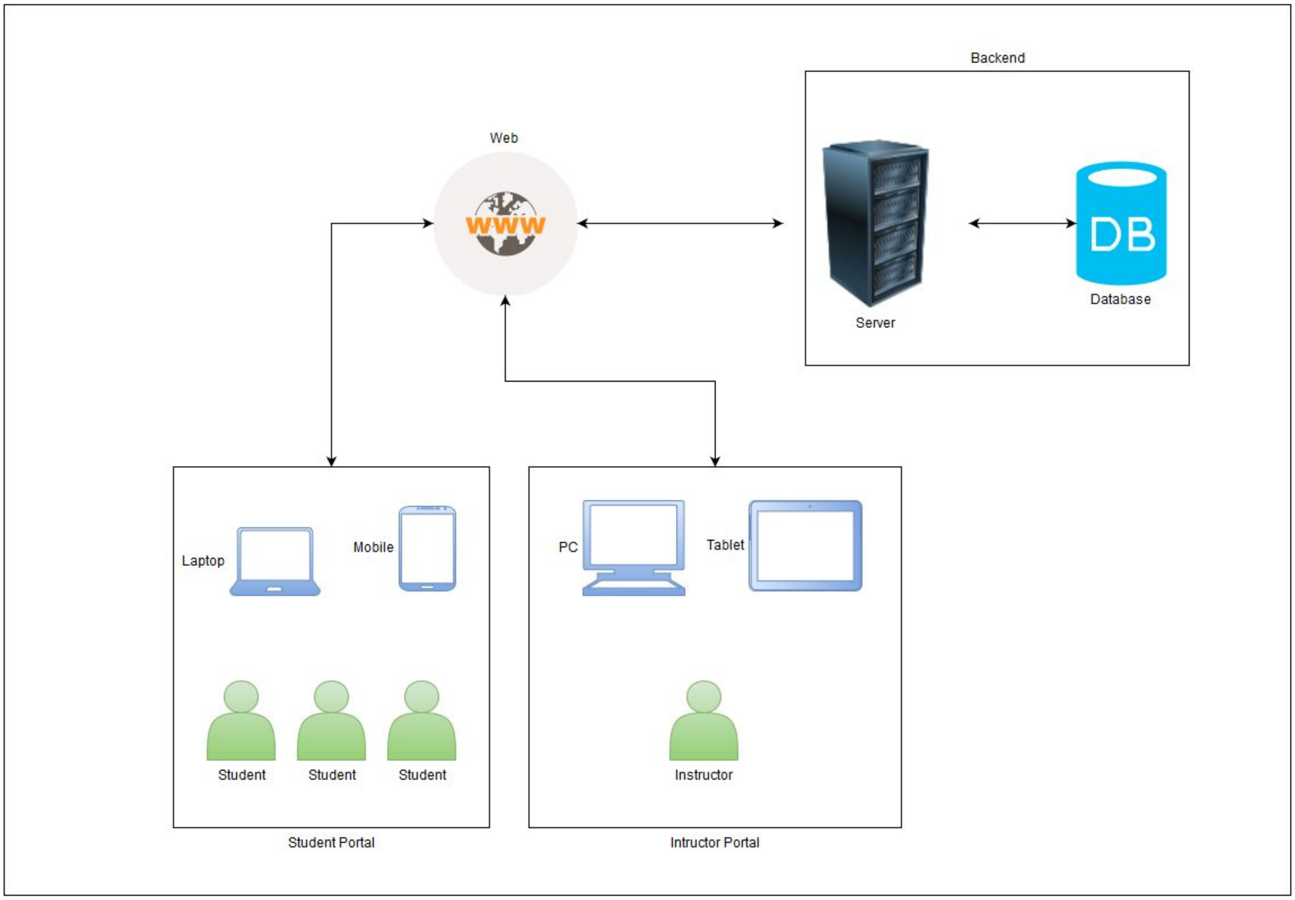


Figure 1.0- Proposed System Overview

References

[1] Mayer, Richard E., Andrew Stull, Krista Deleeuw, Kevin Almeroth, Bruce Bimber, Dorothy Chun, Monica Bulger, Julie Campbell, Allan Knight, and Hangjin Zhang. "Clickers in College Classrooms: Fostering Learning with Questioning Methods in Large Lecture Classes." *Contemporary Educational Psychology*: 51-57. Print.

[2] https://www.polleverywhere.com/ars-comparison

[2] http://myprogrammingblog.com/2011/09/16/web-poll-script-overview-top-5/