Project Proposal

for

BoardCast

Version 0.1a

Prepared by TheTeamFromNTU

University of Waterloo

CS 446, ECE 452, CS 646

19/01/2018

Table of Contents

Table of Contents 1

Revision History 1

1. Introduction 2

1.1 Team Members 2

1.2 Purpose 2

1.3 Functional Properties 3

1.4 Use Case Scenario 4

1.5 Non-Functional Properties 4

2. Low Fidelity Prototype 6

2.1 User Interfaces 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
| Elgin Lee | 13/01/2018 | Initial Inception | 0.1 |
| Elgin Lee | 19/01/2018 | Addition of Lo-Fi Prototypes | 0.1a |

# 

# Introduction

## Team Members

|  |  |
| --- | --- |
| **Name** | **Quest ID** |
| Elgin Lee Wei Sheng | 20753953 |
| Norman Yeo Guo Kuan | 20755357 |
| Lee Kian Lon | 20755364 |
| Harold Lim Jie Yu | 20755365 |

## Purpose

The proposed system is BoardCast, a mobile application which serves to aid the evolution of smart classrooms using a peer-to-peer infrastructure to reduce the cost of setup and complexity of deployment.

BoardCast will include features such as real-time polling, screen sharing, attendance taking, results reporting and two-way feedback. The precise details of the features are documented in the sections below. These are key features that teachers and students would find useful and engaging. Presently, most Universities still use a clicker system where an external clicker needs to be rented and students would have an additional device to bring to class. The setup of the clicker system also requires a server which adds unnecessary complexity and leads to resistance to the adoption of the clicker system. The clicker system is also not dynamic, in the sense that the lecturer is unable to project images or feedback back to the clickers. High schools or lower may not even want to invest in such a large and cumbersome system as they do not have the economy of scale. Though, these kind of smart classroom features would greatly benefit classrooms at any level of education.

BoardCast is developed as a mobile application as almost all students and teachers would have their phones with them. Even the students in high school or elementary school now bring their phones to class. This provides us with a device that has enough computation power not only replace the clicker system but enhance it with new features such as two-way feedback and screen sharing.

## Functional Properties

The following list offers a brief outline and description of the essential features and functionalities of the BoardCast system.

Core Features

1. Hosting and Connection
   * The teacher would create a session which will be broadcasted over the Wi-Fi network

* Students can see the new session and join it
* If the teacher had set a passcode, the student would have to enter the passcode to join

1. Poll Creator
   * Before the lesson, the teacher would be able to create a poll using a suite of tools which allows flexibility to add innovative question formats
   * The fields are easily editable through an intuitive user interface
   * The poll is then saved into the teachers vault only to be retrieved when the teacher creates the session and selects the poll
2. Polling
   * The teacher would be able to view the number of students connected to the session and then physically verify the number of students in class
   * The teacher can start the poll at any time and control the flow of the poll by setting a timer for the question, going to the next question and closing the poll
   * The students would select an option among those available
   * After each question, the results of the poll would be reflected on the teachers and student’s device. This allows the students to review his mistakes. This can be turned off if the teacher wishes to conduct a quiz instead of a lesson
3. Results Export
   * After the poll, teachers can export the class response and results to a PDF file
   * After the poll, students can export their responses and the actual answer to a PDF file for review later. Extra comments added by the lecturer to aid understanding would also appear in the PDF. This export function adds an extra layer of security to ensure if the results were not captured on the lecturer side, the student has evidence to prove his participation
4. Screen share
   * The lecturer can share his screen with the rest of the class, this would be useful when a projector is not available to be used
   * The lecturer can also select a student to share with the rest of the class, the student would be able to draw on his mobile device and this answer would be broadcasted to the rest of the class
5. OAuth Login
   * The user would be able to link accounts of other services (such as DropBox and Google) to store his previous polls, results and poll templates

## Use Case Scenario

**Scenario 1: Teacher wishes to conduct a quiz for the class**

The teacher would create a poll before the start of class using a suite of interactive tools. During class, he would create a session with a passcode. Students would join the session and key in the passcode for authentication. The teacher would physically tally the number of students present with the number connected to the application. The teacher can start the session and conduct the quiz, students would answer accordingly. At the end of the quiz, the teacher would receive a report of each student’s answers, a per-question graphical analysis would also be drawn for the teacher. The student receives a report on his answers and the correct answer, with comments from the teacher for self-review. The teacher benefits as the quiz is automatically marked and statistics for each question are generated automatically allowing the teacher to focus on weak areas of all students. The student benefits as he can review his answers and learn from his mistakes.

**Scenario 2: Student wishes to answer a question which requires diagrams**

The student raises his hand, the lecturer selects the student and authorizes the students broadcast. The student is then able to draw and type on his screen which is broadcasted to the rest of the class. This benefits other students as they are now able to see the answer with a diagram. The lecturer can then correct the students answer which is also broadcasted to the entire class.

## Non-Functional Properties

1. Efficiency

The application should respond to user requests instantaneously as any delay would make the application feel slow and degrade the user experience. The application must record the responses of students instantaneously and update the results on the lecturer’s device at real time. The overall user experience determines if the user would reuse the application or go for a competitor’s product, as such, this is a key requirement.

1. Reliability

The application should be able to recover in the case of intermittent connections or dropped connections to the host device. The application must also consistently inform the student if he is connected to the teacher. The application must be able to generate reports for both teachers and students to ensure non-repudiation. This is essential as user trust in the application is very fragile and an error whereby the results are not recorded would lead to user mistrust. This would determine if the user would reuse the application or not use it anymore.

1. Security

The application should use a certain level of encryption for communication between the teacher and student. This ensures that other students are unable to eavesdrop on answers with a packet sniffer over the wireless medium. The encryption also ensures that the answers are not altered by others and not affected by transmission error.

1. Ease of Use

The user must be able to learn the application without a tutorial or guide. The interface must be intuitive enough such that the user is able to perform his intended actions without getting lost or confused. This is a crucial requirement as users usually decide whether they should continue using the application by the initial look and feel of the application. If the user finds the application too confusing, the user would probably uninstall the application after a sole use.

# Low Fidelity Prototype

## User Interfaces

### Login

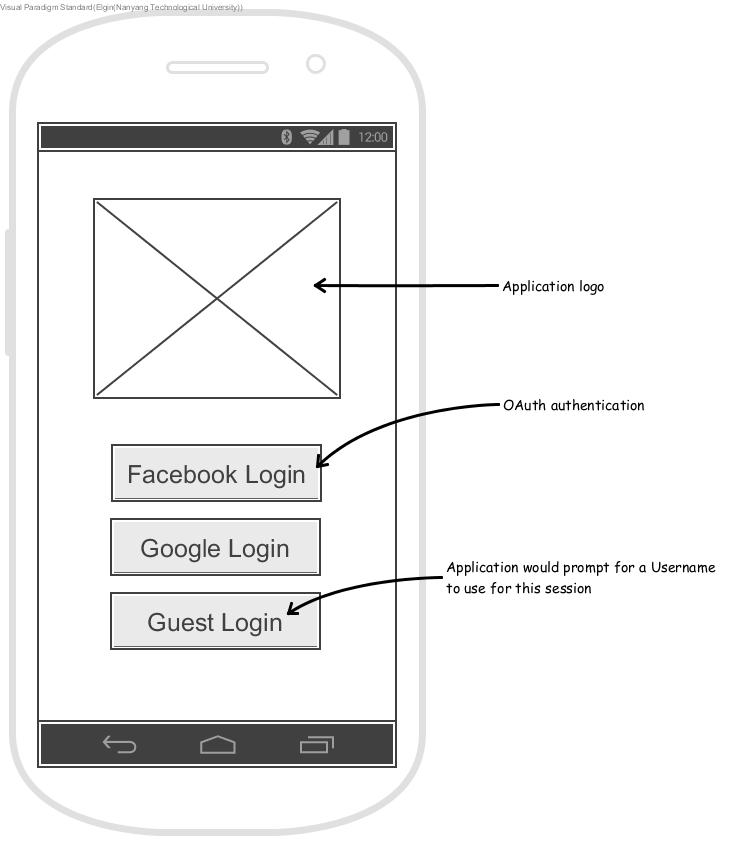


Figure 1 - User can choose to load his profile from other services

### Main Menu

Figure 2 - Simplistic main menu to increase intuitiveness

### Poll Creation

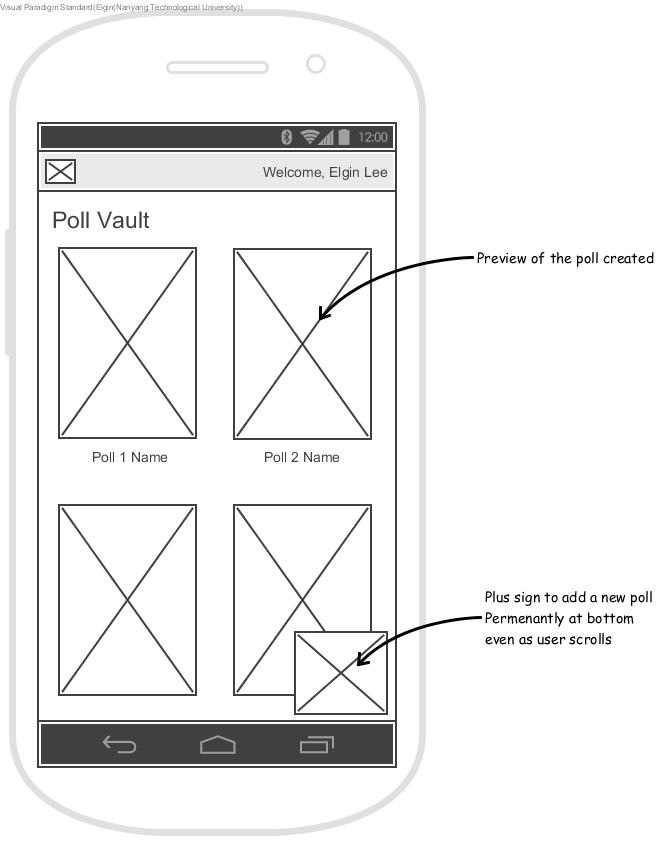


Figure 3 - User can view his created polls

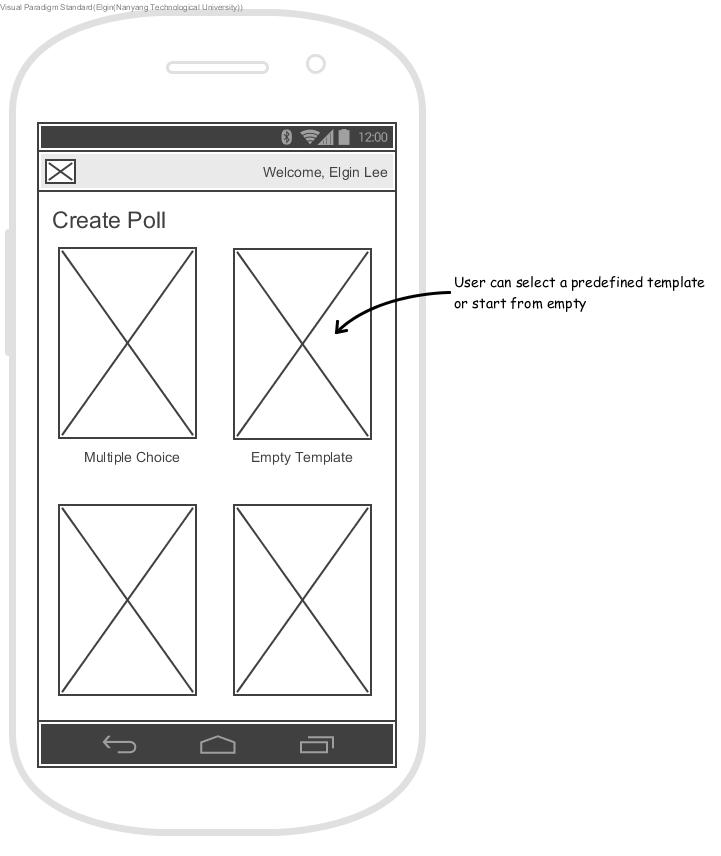


Figure 4 - User chooses a template to create his poll

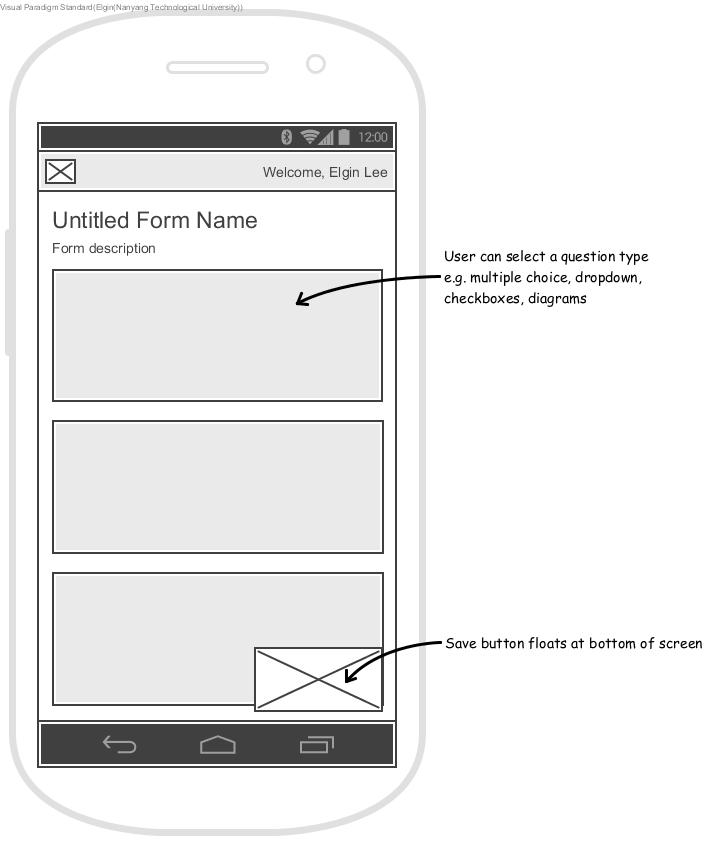
****

Figure 5 - Interactive tools to aid in poll creation

### Join Poll

****

Figure 6 - From main menu, user taps join poll

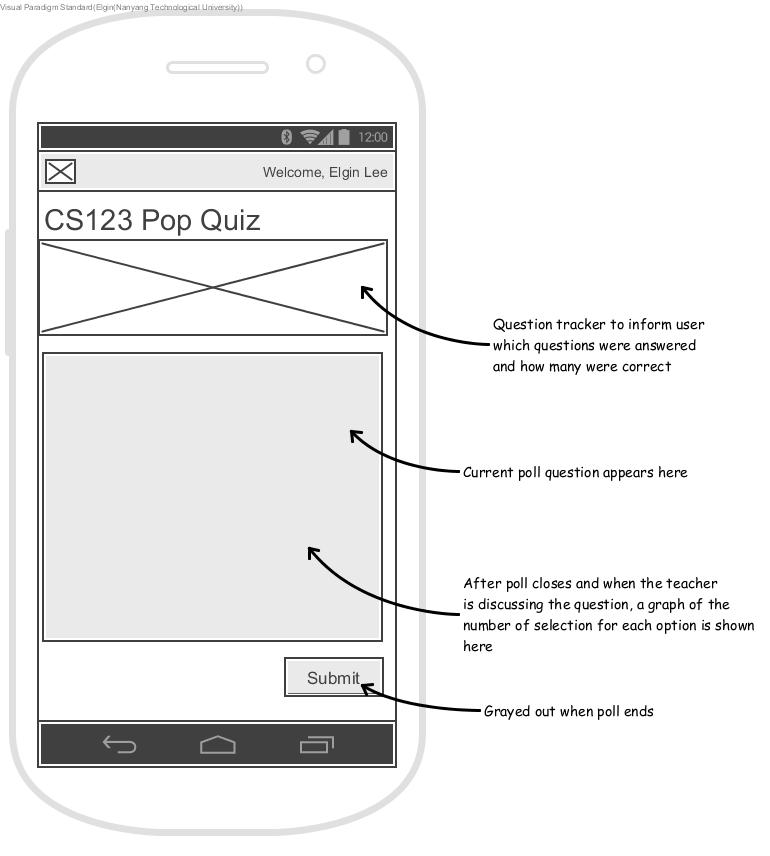
****

Figure 7 – Participant view during active poll

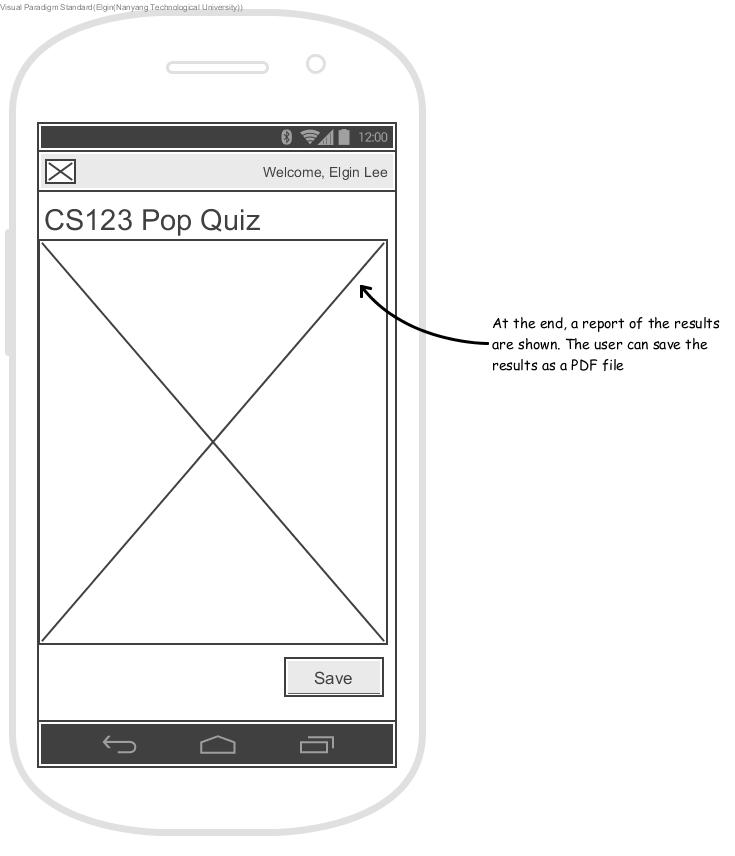
****

Figure 8 - Participant view at end of poll

### Hosting a Poll

Figure 9 - Teacher view during active poll