**Medical Education Application**

Developers: David Mattingly and Nathan Schmidt

Clients: Dr. Matt Broom and Dr. Chris Brownsworth

Supervisor: Dr. Jason Fritts

Deliverable #3

10/12/18

**Major Components for Finished Application:**

The application will first bring up a login screen, which will prompt the user to either login with a username and password, or create a new account. Creating a new account gives the choice of whether the user is a student or a teacher, and this choice determines how the app behaves. Since the questions will be officially made by doctors for medical school, users of the app will also be prompted to an end user legal agreement before creating an account. When teachers successfully log in to the app or web, they will have the option to create classes, post questions (which will be sorted into groups based on what topic the question is for), control privacy restrictions of the content, and receive feedback reports of the students’ performance. When students successfully log in to the app, they will be able to pull up a list of classes to register for (which the teacher will approve), filter questions by groups, highlight specific questions to review, and access reports for their overall scores and compare results.

Questions will be in a multiple choice format, and explanations will be given for each choice, regardless if it’s right or wrong. The teacher will have the option to add pictures or media to the questions as well as links to access more information related to the material. The questions will be peer reviewed, posted to private groups or all people using the app, and have the option to mark as a copyright violation. After answering the question, the students will be able to decide if questions will be posted either in blocks or over time. Overall, the app will be structured to be easy to use and able to access “teacher” or “student” functionality within the same app. The information and statistics from the app will be recorded for educational studies including demographic information if needed.

We will be developing the application for iOS devices, so we will be coding the application in Swift 3, using php and HTML for the web screens, and mySQL for the database of questions and user logins.

**Timeline for remaining goals:**

* Goals for Deliverable #4:
  + Test new features, such as rolling out automated questions over time
  + Connect database to Hopper
  + Test final features
  + Prepare for application rollout
  + Complete user guide for the application
  + Obtain final feedback from the client
  + Roll out app to a small pilot group
  + Analyze and fix any issues found by the pilot group
  + Finalize the app to be available to all users

**What we have designed so far:**

In addition to the app that we designed from Deliverable #2, we have added a website that the teachers use to enter in their class’s questions. We have decided to do this instead of implementing it in the app because this is both more user friendly to the teacher, and is easier to implement than the original method. Opening the website shows a login screen for the teacher. Successfully logging in shows two buttons for either creating a new class or opening existing classes, and logging out. Creating a new class shows the input for what the class is named, and then once it is created, it goes to another screen that shows the list of questions, plus an option to add a new questions. Creating a new question goes to the create question screen, which has inputs for the question, the 4 answer choices, the explanations for why they’re right or wrong, and media if there is any wanting to be added by the teacher.

The website contains a database that will connect to the application. Currently, the database is ran locally, however our goal is to connect the database to Hopper and have both the website and the application connected to it.

**Any known bugs / difficulties:**

Since the user database has not been created yet, some of the buttons do not function as they’re supposed to and some buttons don’t do anything when you click on them (specifically the “View Performance” button on the Home Screen and the “Highlight Question for Later” button on the question results screens). We also still have not figured out how to connect the application to a server.

**Code for website:**

demo.php:

<?php

define('DB\_NAME','forms1');

define('DB\_USER','root');

define('DB\_PASSWORD','');

define('DB\_HOST','localhost');

$link= mysqli\_connect(DB\_HOST, DB\_USER, DB\_PASSWORD);

if(!$link){

die('Could not connect:'.mysql\_error());

}

$db\_selected = mysqli\_select\_db($link,DB\_NAME);

if (!$db\_selected){

die('Can\'t use'.DB\_NAME.':'.mysqli\_error($link));

}

echo("Connected to the DB successfully");

$question= $\_POST['question'];

$answer1= $\_POST['answer1'];

$answer2= $\_POST['answer2'];

$answer3= $\_POST['answer3'];

$answer4= $\_POST['answer4'];

$correct= $\_POST['correct'];

$incorrect1= $\_POST['incorrect1'];

$incorrect2= $\_POST['incorrect2'];

$incorrect3= $\_POST['incorrect3'];

$media= $\_POST['media'];

$sql= "INSERT INTO demo (question,answer1,answer2,answer3,answer4,correct,incorrect1,incorrect2,incorrect3,media) VALUES ('$question','$answer1','$answer2','$answer3','$answer4','$correct','$incorrect1','$incorrect2','$incorrect3','$media')";

if (!mysqli\_query($link,$sql)){

die('Error: ' .mysqli\_error());

}

mysqli\_close($link);

?>

Demo-form.php:

<form action="demo.php" method="post" />

<p>Enter question:</p>

<p>Question: <input type="text" name="question" /></p>

<p>Answer 1: <input type="text" name="answer1" /></p>

<p>Answer 2: <input type="text" name="answer2" /></p>

<p>Answer 3: <input type="text" name="answer3" /></p>

<p>Answer 4: <input type="text" name="answer4" /></p>

<p>Correct: <input type="text" name="correct" /></p>

<p>Incorect 1: <input type="text" name="incorrect1" /></p>

<p>Incorect 2: <input type="text" name="incorrect2" /></p>

<p>Incorect 3: <input type="text" name="incorrect3" /></p>

<p>Media: <input type="text" name="media" /></p>

<input type="submit" value="Submit" />

<input type="submit" value="Previous Screen" />

</form>

**Requirements fulfilled:**

Requirement 1a.

Create a user application interface for IOS mobile devices.

Requirement 1b.

Have a login screen that will prompt the user for a username and password of an already enrolled user, or the option to create a new account for a new student or teacher.

Requirement 1c.

The new user screen should prompt the user for their email address, password, username, and whether they’re a student or teacher. - **database for users not created yet**

Requirement 2a.

Have a home screen for the student account with the ability to access currently registered classes, join a new class, view their performance, and a logout option.

Requirement 2b.

Have a home screen for the teacher account with the ability to create a new class, open a existing class, access the overall performance of the students, and logout. - **done in the website**

Requirement 3a.

The teacher account will have the create new class option, which will prompt the teacher to enter a class number and name. **- done in the website**

Requirement 3b.

In an created class, the teacher will be able to create a new group of questions to provide to the class, or access a group of questions that already exists. **- done in the new website**

Requirement 3c.

For every existing group of question created, the teacher will be able to edit current questions, and create a new question to be posted.  **- done in the new website**

Requirement 4a.

When the student has the option to select a class and open the groups of questions available or request to register sending the request to the teacher.

Requirement 4b.

When the student has access to the class it will select a group of questions to open the content available.

Requirement 4c.

For every group of questions, the student will have the option to prompt a new question, or review previously answered questions, with the option to highlight questions they want to focus on.

Requirement 5a.

The teacher creating a new question will type the question, create multiple answer choices for the students marking which are the right answers, add an image, multimedia, or links if desired, and provide an explanation for each answer. - **done in the new website**

Requirement 5b.

The question screen for the student will show the question, the multimedia, image or links added by the teacher, and the multiple possible answer for the student to choose from. When an answer is chosen, an explanation for whether the answer is right or wrong will be prompted.

**Requirements remaining:**

Requirement 5c.

The student will be given the choice to access the next new available question after answering a question, or be prompted to answer a new question in 24 hours.

Requirement 6a.

The overall performance screen for the teacher will show each of the students’ performance in a question group, how successful the students are with a specific question,

Requirement 6b.

The overall performance screen for the student will show the percentage of answering the questions right for all the different group of questions they have answered.