**Common Details**

Screens are referred to as such , but they don’t actually mean another html file, this functionality is part of the jQuery Mobile CSS file which skips to different parts of a single html file.

All submission of data will be accomplished with a form submit button sending JSON data via ajax.

All passwords are held with something like MD5 encryption to scramble and unscramble a string.

**Login Screen**

**Description**

Email – text entry

Password – text entry

Login – button

Register – button

Forgot Password? – button

**Functionality**

Utilizing client side persistent memory if a username / password have been successfully entered they will be pre-populated into their respective fields. Otherwise they will await input with placeholder text denoting the type of input.

The Login button submits the form data and if successful then the server returns code to allow the submit to complete and move to the Quiz Registration Screen while updating the persistent memory with the users email and password details.

If unsuccessful the submit will display a notification for either field not matching basic requirements (e.g. an email without an ‘@’ symbol) or that the user was not identified in the online database and they should complete a registration.

The Registration button is a straight link to the Registration Screen.

The Forgot Password? Button is a straight link to the Forgot Password Screen

**Forgot Password Screen**

**Description**

Email – text entry

Submit – button

**Functionality**

If the email meets basic requirements as in the Login button above it will be sent to the server. If it successfully matches an email account the password for that account is unencrypted and sent to the email associated with the account. A notification to this effect is displayed and a code is sent back from the server allowing return to the Login Screen.

If the Email is not found in the database then a notification of this fact is displayed and urges the user to register, the Register Screen is then brought up.

**Registration Screen**

**Description**

Email – text entry

Password – text entry

Repeat Password – text entry

Male – check box

Female – check box

Birthday – date selector

Receive Email Notifications? – check box

Login – button

**Functionality**

Only three text entries are mandatory: Email, Password and Repeat Password. These are checked for basic requirements and submitted null values for those items not completed.

The Login button submits the form data and if successful then the server returns code to allow the submit to complete and move to the Quiz Registration Screen while updating the persistent memory with the users email and password details.

If unsuccessful the submit will display a notification for either the field not matching basic requirements (including Password and Repeat Password not matching) or that the user already existed in the database and then move them to the Login Screen.

**Quiz Registration Screen**

**Description**

Quiz Code – text entry

Join Quiz – button

**Functionality**

The input into the text entry is sent to the database if it matches then the user is created as a participant of the quiz, and the app moves to the Quiz Wait Screen and waits for the quiz to start. If it does not match a notification pops up informing the user of this fact and waits for another input.

**Quiz Question Screen**

**Description**

Current Score – html text

Question Category – html text

Question – html text

Timer Bar – slider bar controlled by javascript (30 seconds)

Four multiple choice Answers - buttons

**Functionality**

When the status indicates an “Open Question” the current score, category, question text, and multiple choice answers are displayed on this screen. When displayed the timer bar immediately starts counting down from 30 seconds to 0.

Each answer button is a submit button submitting it’s corresponding answer as the users choice and points are awarded for right answers as a factor of the time left they had to answer the question. For example if I choose the right answer in 5 seconds I would get 2500 points 100 each for the remaining 25 seconds left.

If the wrong answer is selected then no points are awarded.

Either way the resulting updated score is submitted to the online database and the app returns to the quiz wait screen described below.

**Quiz Wait Screen (also Intermission and Finish Game Screen)**

**Description**

There are four elements that can appear in the main display area of this screen:

Advertisement – image

Correct Answer was… – html text

Current Score – html text

Leader Board – table text

Sign Out – button

**Functionality**

This screen is approached from the Quiz Registration Screen, or Quiz Question Screen. Once here the app uses javascript to poll the online database every second for the current question. Each time it pulls down the question details (or in the case of before a quiz starts null values and a status) and checks the status. The resulting action is determined by the status:

Before Start – Before the quiz master has loaded up the first question

Advertisements

First Question – Loaded a question and the details haven't been released to the room

Advertisements

Closed Question – Loaded a question and the details haven't been released to the room

Advertisements, Correct Answer was…, Current Score

Open Question – Question released to the room

Move to the Quiz Question Screen

Intermission – Pause during the game

Advertisements, Leader Board

Quiz Finish – Quiz is finished

Advertisements, Sign Out button

The Sign Out button signs the user out of the quiz and closes the app.