Technical Design Document

Are you a Science Nerd Quiz App

Version: 1.0

8/17/2017

Author: Korrenti Mayweather

**1. Introduction**

This document is a Techinical Design Document for the “Are you a science nerd?” science quiz app. It provides guidance and template material intended to assist those involved in the development of the app.

**2. Architecture Design**

**Start Page**

Name: Start Page

Type: Web Page

Description: This is the initial home page of our app to be presented to the user. The start page will be rendered to the page with three science vector images at the top of the page. Under these images, a left justified title will be shown to the user followed by a description of the app below the title. Below the app description will be a button centered on the page with the label “Start Quiz”.

Operations:

Name: renderStartPage(), handleStartButtonClicked()

Arguments: None

Actions: Calls renderQuestionPage()

Arguments: questionNum

**Question Page**

Name: Question Page

Type: Web Page

Description: This is the page that displays the quiz questions and answers to the user. This page will have the same three vector images at the top. Below these images will be text that shows the user what question number they are on out of the total number of questions (i.e. “Question 1 of 10”). Below this will be the question text justified left. Below the question will be an unordered list of four answers listed vertically and centered on the page. Each answer will have a border around it with the text centered inside. A “Next” button will be centered at the bottom of the page. When an answer is selected the box that contains the answer will turn green or red to illustrate whether the answer is correct or incorrect respectively. The words correct and incorrect will also appear at the bottom of the answers list depending on whether the answer was correct or not. The users selections will be stored in a cookie variable that will persist throughout the quiz. The user will not be able to change their answer once selected. When the user reaches the last question, the net button will be replaced with a “Results” button. Clicking this button will render the “Results” page.

Attributes: Text

Resources: None

Operations:

Name: renderQuestionPage()

Arguments: questionNum

Name: handleNextButtonClicked()

Arguments: questionNum

Actions: calls renderQuestionPage()

Name: handleResultsButtonClicked()

Arguments: results

**Results Page**

Name: Results Page

Type: Web Page

Description: This is the page that will show the users quiz results. Three vector icons will be shown at the top of the page. Below this there will be text that tells the user how many questions they got right out of the total number of questions. Depending on the number of questions the user answers right, the user’s feedback will vary:

10/10 Questions Answered correctly shows the user “Congratulations, you are a science Nerd”

7-9/10 Questions Answered correctly shows the user “Way to go, you are a science buff. Try again for nerd status!”

6/10 Questions Answered correctly shows the user ”Way to go, you know some science but you’re no nerd”

5 or less / 10 questions answered correctly shows the user “Sorry, you’re no science nerd”

The user will be able to click a “Try Again” button to restart the Quiz.

When the user clicks the “Try Again button, the Start page will again be rendered.

Attributes: Text

Resources: None

Operations:

Name: renderResultsPage()

Arguments: results

Name: restart()

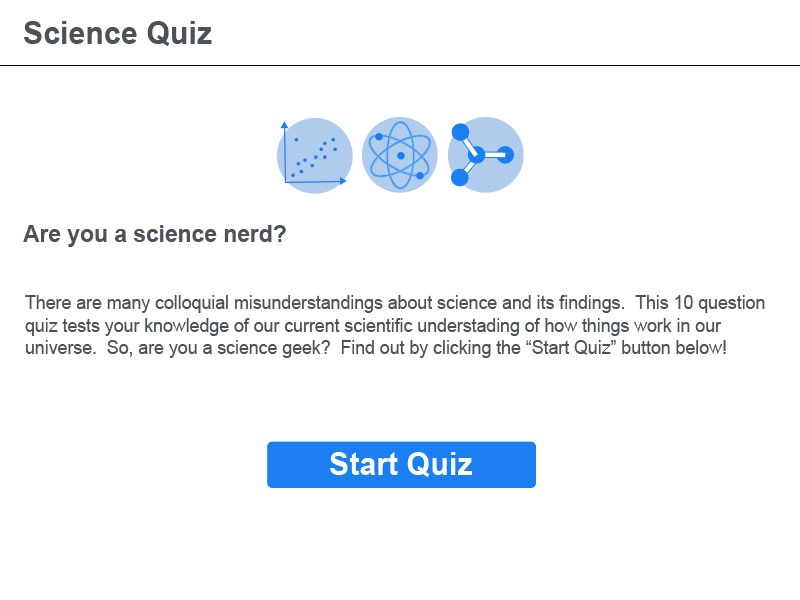
Arguments: None

**3. Database Design**

Our questions, answers, and correct answers are stored in anarray named “questions”. This global variable will allow us to easily display pertinent information to the user. Information will be pulled from “questions” and rendered in the DOM. Our user’s results will be stored in a variable named “results”. The “results” variable will be updated each time a user gets a question right. The total number of correct answers will then be rendered to the user after all questions have been answered.

**4. Graphical User Interface**

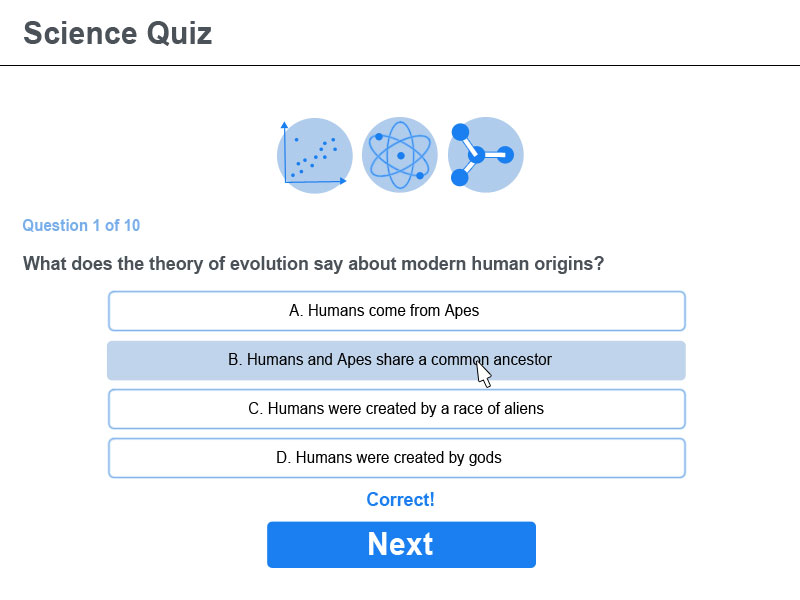
* Provide, in an organized way, the pictures of all the forms in the graphical user interface with a reference to the functional requirement it implements. You may use html to present the graphical user interfaces.
* For each form in the graphical user interface, provide:
  + The names of the controls and fields on that form,
  + The names of the events, methods, or procedures that cause that form to be displayed, and
  + The names of the events, methods, or procedures triggered by each control.



**Figure 1 Start Page**

Functions: renderStartPage(), handleStartButtonClicked(), handleResultsButtonClicked()

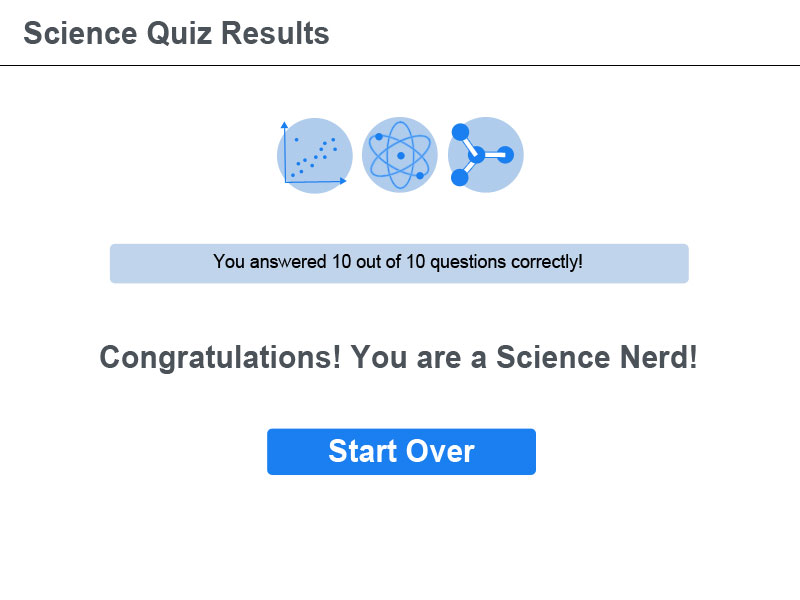
Button: “Start Quiz” button



**Figure 2 Question Page**

Functions: renderQuestionPage(), handleNextButtonClicked()

Button: “Next” Button, “Results” button



**Figure 3 Results Page**

Functions: renderResultsPage(), Restart()

Buttons: “Restart” button

**5. Design process**

Mockups were created in Adobe Photoshop using a blue and light blue color scheme. All vector elements were created in Adobe Photoshop. Vectors are in jpeg format and also follow the blue/light blue color scheme. Answers will turn Green if answered correctly and Red if they are answered incorrectly. HTML and CSS will be used to implement Mockup design.