Algebra Intelligence Design Document

by
Bill Ezekiel and Lori Benson

Table of Contents

1. Introduction

Algebra Intelligence is a website designed to test a user's knowledge in different topics of Algebra through a series of multiple choice quiz questions. The answers the users provide assist in identifying their strengths and weaknesses in the given Algebra topics. Each user has a skill level in each topic which increases or decreases based on their responses. The skill level is the deciding factor in the difficulty of each question.

The main goal of Algebra Intelligence is to address the strengths and weaknesses of a user and redirect them to another website for additional help when necessary. The website accomplishes this goal by generating a multiple choice quiz. Individual questions are generated using templates which provide the blueprints of the question. The skill levels range from one to five with one generating the easiest questions. Upon completion of a quiz, users will receive an overall grade and a grade in each category/topic. If a user does well, their skill level will increase. If a user does poorly, their skill level will decrease and they will receive a link redirecting them to additional help.

To address the intended age group of the users, the website is designed to provide the user with an easy to use graphical interface with simple point-and-click buttons. The home page buttons to direct the user to different pages of the website are easy to locate and understand**.** The user log in button redirects them to another web page that requires a username and password to access the Algebra Intelligence quiz**.** The start quiz button located in the center of the home page allows the user to start a quiz, but only if the user is logged into the system.

2. Backend Design

The following sections explain the backend design of Algebra Intelligence.

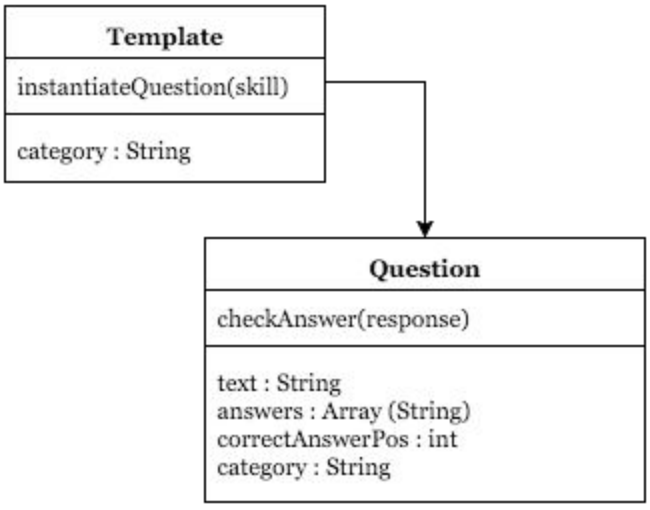
2.1 Languages

The backend of Algebra Intelligence was implemented using Node.JS, a runtime environment for server-side JavaScript. Most of the work is handled on the server side, while certain components, more specifically, the loading of HTML required on most pages, is handled on the client side. Anything related to databases is done through MongoDB, a NoSQL database system interpreting entries as JSON objects or documents. The design of the databases will be further explained in Section 2.3.

2.2 Objects and Modules

The following subsections will elaborate on the vital objects that Algebra Intelligence will be using. The Node.JS modules that will be used will also be mentioned.

2.2.1 Templates and Question



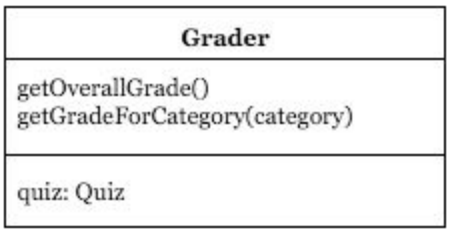
Since Algebra Intelligence does not contain a database of questions, the questions are automatically generated. The communication between Template and Question objects

accomplishes this goal. A Question object includes text, a category, an array of answers, and an integer representing the position in the answer array that is correct. Besides accessor and mutator functions, *checkAnswer* is the only function in a Question object. The *checkAnswer* function takes a response and compares it with the correct answer. The function will return a one if the correct answer is selected and then it is added to the score. Score will be explained in the next section.

Questions are generated through the Template object. One of the templates used is based on finding roots of quadratic equations. Quadratic equations are usually represented in the following form: $\underline{a}x^2 + \underline{b}x + \underline{c} = 0$, where a, b, and c are numbers. The Quadratic Root template also follows this generic form having values for a, b, and c that need to be filled in. A call to the *instantiateQuestion* function is done in order to fill in the values. When given a skill level, the template will generate different numbers for the question. Higher skill levels will equate to larger numbers or the introduction of decimals. The idea behind the template is to create a question by filling in these blanks. Returning to the quadratic root example, randomly generating the values for a, b, and c could result in a 1, -5, and 6. The three values are then plugged in allowing us to create the text for the question as follows: "What are the roots of x^2 -5x + 6", As far as the answer choices for these question, they are also calculated at this time based on the given values for this template. By using templates for many types of questions, the need for a database of predetermined questions is eliminated. To keep things simple, Algebra Intelligence currently only has four implemented templates: evaluate expressions addition, evaluate expressions subtraction, quadratic roots, and linear equations.

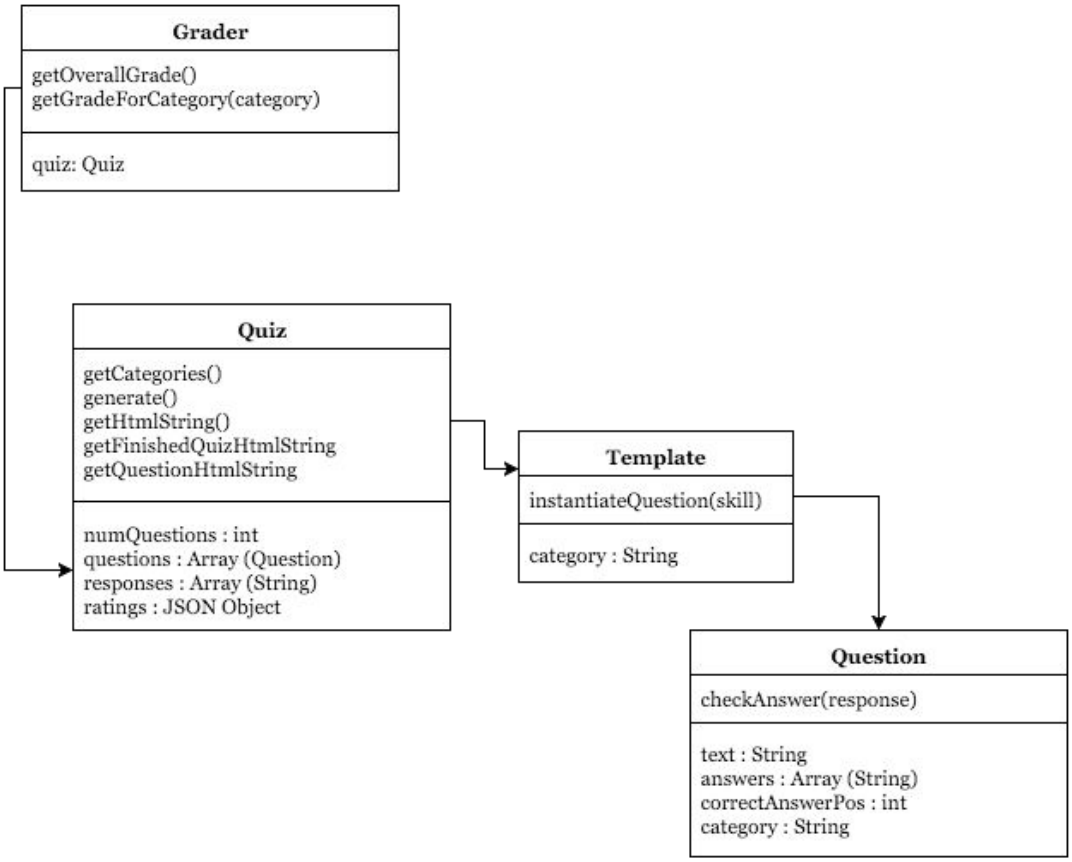
It is important to note, the Templates are loaded into the JavaScript environment by retrieving their source code from a templates database, but rather from the template database. In doing so, this ensures only required templates are loaded and not all of them.

2.2.2 Grader



The Grader is responsible for grading a completed quiz. Not only does it return a grade for the quiz overall, but it also grades a quiz based on each individual Algebra category by calling the *getGradeForCategory* function. The Grader is highly significant because grades are what causes skill levels to increase or decrease.

2.2.3 Quiz



The diagram above shows the interaction between the four main objects of Algebra Intelligence: Quiz, Template, Question, and Grader. The Quiz holds the questions, number of questions, a user's current rating, and a user's response. The Quiz object makes the call to the templates database and loads only the required template. The category for each question is selected first, narrowing down the choice of which templates will be chosen. Using the list of categories, corresponding templates are pulled from the database and turned into objects using the *eval* function in JavaScript. A Quiz object also includes a *getCategories* function which retrieves all the categories needed for the Quiz. The Grader uses this when it gets the grades based on the individual categories. Finally, the *HTMLString* functions are used to generate the HTML Strings that displays the ten question quiz in the browser.

2.2.4 Node.JS Modules

Node.JS has many module exports which allow use of different objects, similar to packages in other languages. Algebra Intelligence implements the following modules detailed below.

2.2.4.1 Body-Parser

Node.JS body parsing middleware developed by user dougwilson. Used for POST requests in Node. This module was developed by Douglas Wilson

2.2.4.2 Cookie - Parser

Used for cookies in Node.JS. This module was developed by Douglas Wilson.

2.2.4.3 Crypto

Allows use of cryptography algorithms in Node.JS. Used in Algebra Intelligence to hash passwords. This module was developed by Irakli Gozalishvili.

2.2.4.4 Express & Debug

Allows use to start the Node.JS app. This module was developed by Douglas Wilson.

2.2.4.5 EJS

Allows the rendering of embedded JavaScript files. This module was developed by user Matthew Eernisse.

2.2.4.6 Mongodb

Allows connections to MongoDB databases in Node.JS. This module was developed by user Christian Amor Kvalheim.

2.2.4.7 Uniq

Takes an array of values and turns it into an array containing unique values. This module was developed by Mikola Lysenko.

2.3 Database Design

Algebra Intelligence requires two different types of databases contained within MongoDB.

2.3.1 User Database

The user database for Algebra Intelligence is implemented in the MongoDB application. The example below defines the categories for each column along with an example of the type of input.

_id	username	pass	Quadratic Roots	Linear Equations
~~	“tim”	~~~~	1	5

Fields

- **_id** : An id used in by default for all MongoDB entries.
- **username** : string representing username
- **pass** : string representing a hashed version of the user's password
- **Quadratic Roots / Linear Equation**: These are the names of categories that store a number representing the user's skill level in that category. There is one field like these for each category.

2.3.2 Template Database

Algebra Intelligence's template database, like the user database, is also implemented in MongoDB. An example of how the database is defined can be viewed below.

_id	category	source
~~	"Quadratic Roots"	"function Quadr…"

Fields

- **_id** : An id used in by default for all MongoDB entries.
- **category** : string representing the template category.
- **source** : string representing the source code of the template. The template is made useable using the *eval* function.

3. Front-end Design

The following sections give a detailed description of the Front-end design of the website.

3.1 Languages

The languages used to design Algebra Intelligence's interface were HTML and

CSS. Embedded JavaScript files (.ejs) were used to allow the rendering of pages that include data from the server. They mostly follow the same format as HTML, and were simple to use. Bootstrap and CSS were used together to give the site a cleaner design.

3.2 Pages

The Algebra Intelligence website consists of seven pages and a brief description of each can be found in the following sections.

3.2.1 Home Page

The layout of the home page is designed with buttons along the top to provide the user an easy to understand graphical user interface. Each button redirects the user to a different page within the website. In the center of the page, there is a “Start a Quiz Now!” button that allows the user to begin a quiz.

3.2.2 About Page

After clicking the “About” button on the Home Page, the user is redirected to the “About” page which gives a brief description of the purpose of the Algebra Intelligence website.

3.2.3 Quiz Page

After clicking the “Start Quiz” button on the Home page, the user is redirected to the Algebra quiz page where a ten question multiple choice quiz is displayed. The user must be logged into the website in order to be able to start a quiz. In the instance the user is not logged into the website, they are redirected to the “Login” page first, then they are able to gain access to the Algebra quiz.

3.2.4 Quiz Results Page

The quiz results page is displayed once the submit button has been pressed and shows the ten question quiz with the correct answers. The answers chosen by the users are highlighted in green if they are correct, highlighted in red if incorrect, and highlighted in blue if an answer was not selected. The overall grade and grade for each present category are also given. If a user scores above 80% in a category, their rank in that category increases. If a user scores 50% or under in a category, their rank in that category decreases, and a link to an external source is given.

3.2.5 Ratings Page

The ratings page displays the user's skill levels in each category in tabular form.

3.2.6 Login Page

The login page lets users login so that they may use Algebra Intelligence's features. User's are notified of invalid credentials.

3.2.7 Signup Page

The "signup" page creates a new user and stores the username and password in the user database