

Quiz Generator

Authors: Nivi Hariharan, Katie Zutter
#CS400 #2019Spring

Overview

This document outlines the design for the Quiz Generator project. The detailed project description and requirements can be found on Canvas. You are allowed to design this project in any way that makes sense to you; this document is a general outline that may help guide you.

Design

Interfaces

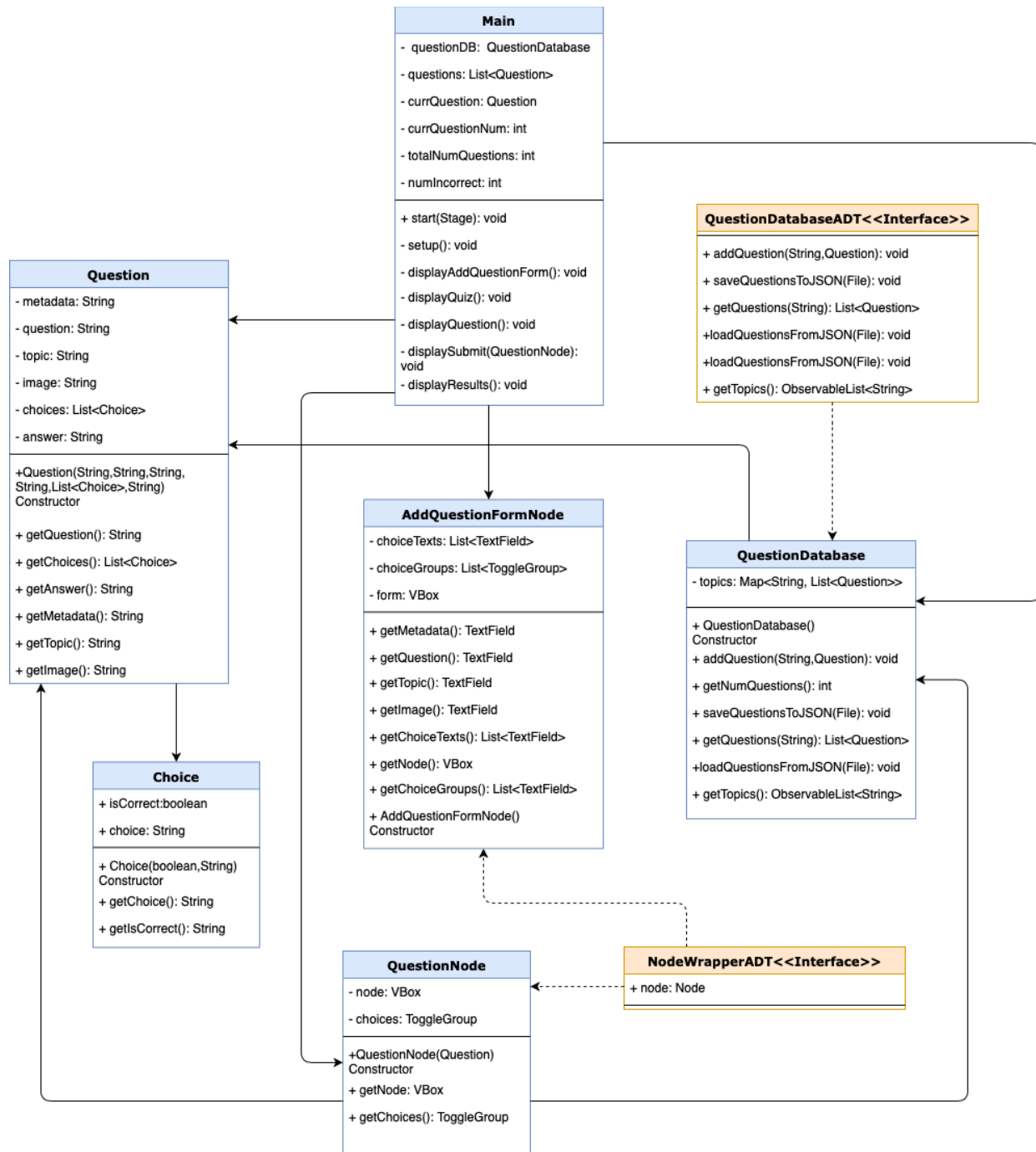
Interface	Description
QuestionDatabaseADT	Contains all questions loaded from .json files and added manually through Add Question Form.
NodeWrapperADT	An interface for classes which wrap around the JavaFX Node class. The intent is to abstract out classes which can internally contain complex JavaFX entities, but could be represented externally by a single Node object, allowing the node to be contained inside other JavaFX containers without exposing any internal details.

Classes

Class	Description
Main	The main application class responsible for creation of the question database and displaying the graphical user interface.
QuestionDatabase	Contains all questions (of type Question) from .json files and added manually by user.
Question	Represents a single question, with members for metadata, question, topic, image, choices (a list of type Choice), and

	answer.
Choice	Contains one of the several choices in a question.
QuestionNode	JavaFX elements for displaying a single Question to the user.
AddQuestionFormNode	JavaFX elements for displaying a form to add a new question to the database. User is required to input all information required to create a new Question object and add it to the database.

Class Diagram



User Interface

Primary GUI

Title (Label)

Title of your application.

Total Number of Questions in Database (Label)

Shows the user how many total questions are in the database. This counter should update whenever new questions are added to the database, either by importing from a new .json file or from adding a new question through the form.

Load Data (Button)

Loads data from a .json file and adds each question to the database. This may result in duplicate questions being added to the database; you may allow this behavior.

Recommended: Use FileChooser to allow the user to select the .json file from the file system.

Add Question (Button)

Opens a new Stage and Scene and displays an AddQuestionFormNode in which user can input all necessary fields to create a new Question object and add it to the database. (See *Add Question Form* section.)

Save to File (Button)

Saves all questions in the question database to a .json file. Important: You must create JSONObject and JSONArray in the same manner as outlined in the provided .json files; i.e., if you input a .json file and then immediately save to file, the input .json file and the output .json file should be the same. (Whitespace doesn't matter as long as the structure is identical.)

Selectable Topics List (ListView<String>)

Displays all topics present in the questions database. The user may select one or multiple topics from which to retrieve questions from the database. This topics list must update when new questions are added to the questions database, either by importing a new .json file or by adding a question manually via the add question form.

Number of Questions in Quiz (TextField)

Allows the user to input how many questions will be in the generated quiz. If this number is more than the available questions for the selected topic, generate a quiz with as many questions as are available. Remember to include user input validation.

Generate Quiz (Button)

Generates the quiz and displays the first question to the user as a `QuestionNode`. (See *Display Question* section.)

Add Question Form

Metadata (TextField)

Metadata of the new question.

Question (TextField)

Question text of the new question.

Topic (TextField)

Topic of the new question.

Image Filename (TextField)

Image filename of the new question. Important: this image file must live in your JavaFX project directory.

Choices (List<ToggleGroup>)

All choices for the new question. You may limit the number of choices allowed to 5.

Choice (ToggleGroup)

A single choice.

True or False (RadioButtons)

Whether the individual choice is a correct one or not. You are not expected to handle multiple correct responses.

Choice (TextField)

Text of the choice.

Choice (ToggleGroup)

A single choice.

True or False (RadioButtons)

Whether the individual choice is a correct one or not. You are not expected to handle multiple correct responses.

Choice (TextField)

Text of the choice.

... (repeat for number of choices user inputs)

Submit (Button)

Creates a list of Choice objects based on user input and creates a new Question object with all necessary fields and list of choices. Adds the new question to the database. Alerts user if any fields are missing or incorrect. Your program should not crash on invalid user input.

Display Question

Question Number / Total Questions in Quiz (Label)

Displays the current question number and the total number of questions in the quiz.

Question (Label)

Displays the question text.

Image (ImageView)

Displays the image in a 200 x 200 pixel frame. Not every question will have an image; you may either always display the frame even when it's not filled with an image or only display the frame when the question has an attached image.

Choices (ToggleGroup)

Group of choices that the user may select from. You are not expected to handle selection of multiple choices.

Choice (RadioButton)

A single choice, as a selectable radio button with text.

Choice (RadioButton)

A single choice, as a selectable radio button with text.

... (repeat for number of choices in question)

Submit (Button)

Submits the answer for the current question and displays whether they got the correct answer. On the last question, displays the results of the quiz. (See *Display Results* section.)

Correct or Incorrect

Displays to the user whether the question they have just answered was answered correctly or incorrectly.

Display Results

Number of Correct Answers (Label)

After the quiz is completed, displays the number of correct answers given by the user.

Total Number of Questions Answered (Label)

After the quiz is completed, displays the number of questions in the quiz.

Score (Label)

After the quiz is completed, displays the user's score as a percentage.

Exit Quiz Generator

Save to File (Button)

Upon exiting the program, allows the user to save the question database to a .json file. (See *Save to File* section.)

Exit Without Saving (Button)

Upon exiting the program, allows the user to exit without saving the question database to a file.