

CST 338 Final: Spec Sheet & UML

Quiz App

1. Spec Sheet

HomeActivity: class creates the welcome screen seen at start of android app

QuestionLibrary: Holds the Questions and Answers. Also determines the logic for the three choices.

MainActivity: class holds the main. Here is where it assigns the choices to the individual buttons. The Score logic is held here. And the Questions are updated here as well.

Phase 1: Home Activity Class.

- Does not hold any variables. Has one method called onCreate() which is a void. It does not do anything besides present a welcome screen xml folder for about 4 seconds. Then it transfers to the MainActivity class to run that logic's onCreate().

What to Test: Make sure that screen holds the XML that you want. Also that it moves to the main screen after about 4 seconds or so.

Phase 2: Question Library Class.

Private Member Data:

private String nQuestions []- An array that holds the questions.

private String nChoices[][] - A double 3x3 array. Each Array holds three choices for the buttons, dependent on question.

private String nCorrectAnswers[]: holds correct answers for each choice that can be made.

Public Methods:

QuestionLibrary(): helps to initialize the class in the main.

String getQuestion(int): returns the Question in the nQuestion array.

String getChoice1(int): returns the first choice in a nChoices Array.

String getChoice2(int): returns the second choice in a nChoices Array.

String getChoice3(int): returns the third choice in a nChoices Array.

String getCorrectAnswer(int): returns the correct answer in the nChoices array.

int getLength(): returns the length of the nQuestions array. This will help in keeping track of the amount of questions there are in Main.

What to Test: This class will be referenced in Main. So Testing will be done there.

Phase 3: MainActivity class.

Private Member Data:

TextView mScoreView: controls the score view.

TextView mQuestionView: controls the question view.

Button mButtonChoice1: first choice will be set here to user.

Button mButtonChoice2: second choice will be set here to user.

Button mButtonChoice3: third choice will be set here to user.

Button mQuit: A quit button if the user wants to move on to next question.

String mAnswer: holds the right answer. Will be used to check for right answer.

int mScore: holds the score and will be used to update the Score View.

int mQuestionNumber: Keeps track of what question number it is.

Public Methods:

void onCreate(Bundle): majority of game logic in here. Action Bar will be displayed on top to make the Quiz Android icon.

Buttons will be linked to choices. There will be an onClickListener for each button. If it is pressed, the game checks for whether it is the right or wrong answer. If it is right it will update the score by 1. If it is wrong it will not update the score. Either way it will move on to the next question.

Toast is an android class that allows the user to see if they got the right or wrong answer on the bottom. It also will notify the game ending. Then it will start from the beginning. This is a simple way to study, flashcard style.

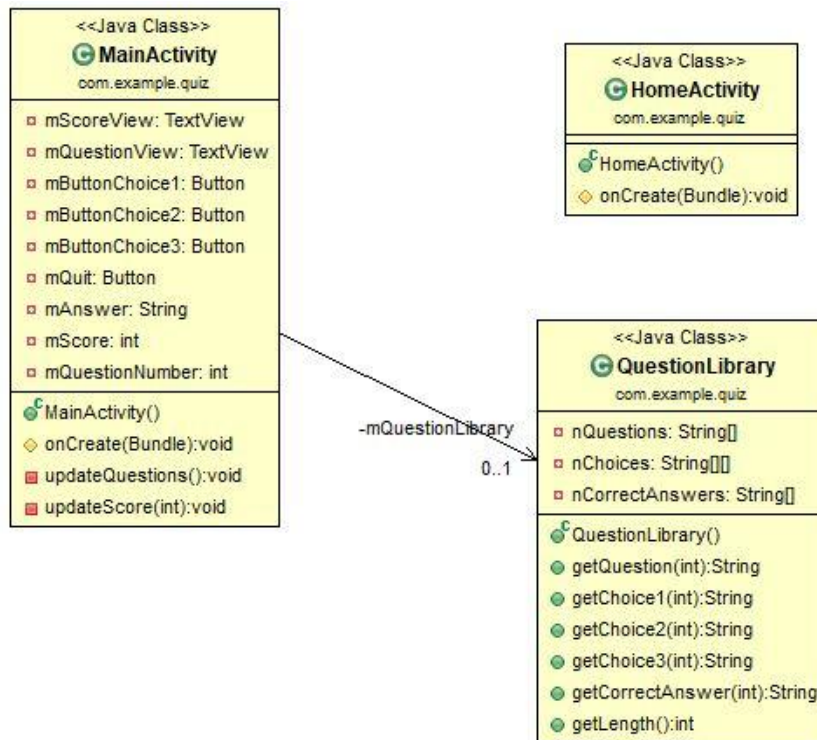
void updateQuestions(): Program will need to update questions after a successful or unsuccessful question. Referenced from Question Library. Also updates correct Answer for Question.

void updateScore(int): will add the score if the user gets answer right. Does nothing otherwise.

What to Test: Test the Buttons to make sure that the choices successfully show up. Test to see if the question will move on after each successful for failed answer. Check if the score will update successfully if right answer is shown or wrong answer.

Check the Toast method to see if it shows "correct" and "failed" outputs below.

UML



OutPut

