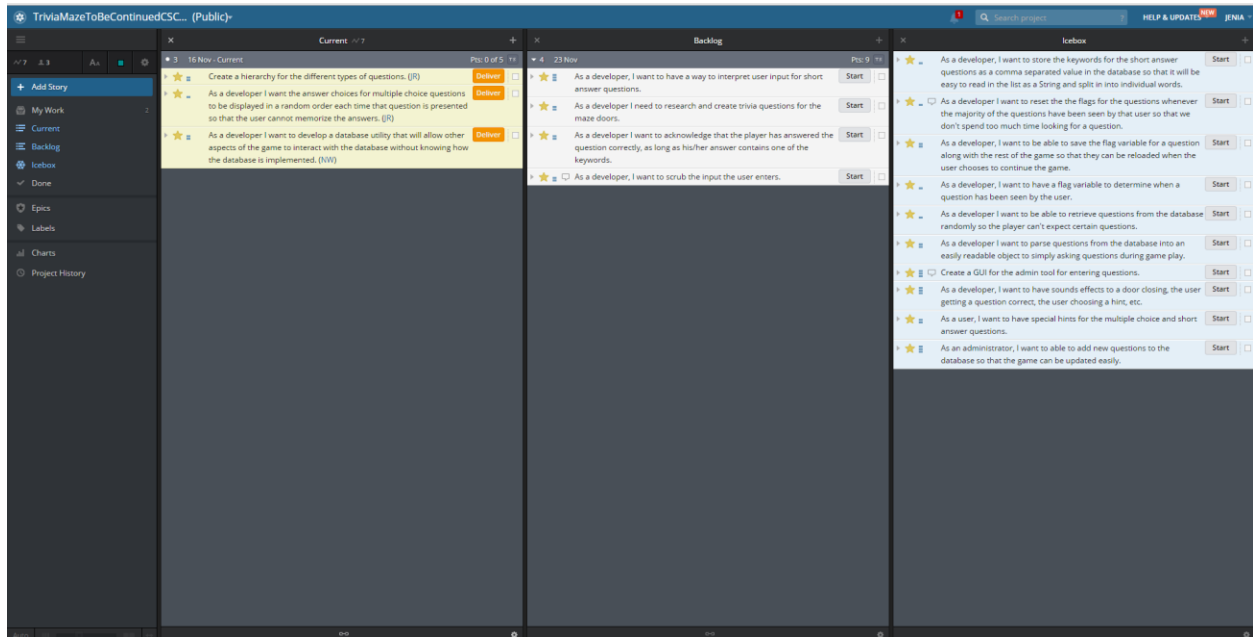# CSCD 350 Software Engineering
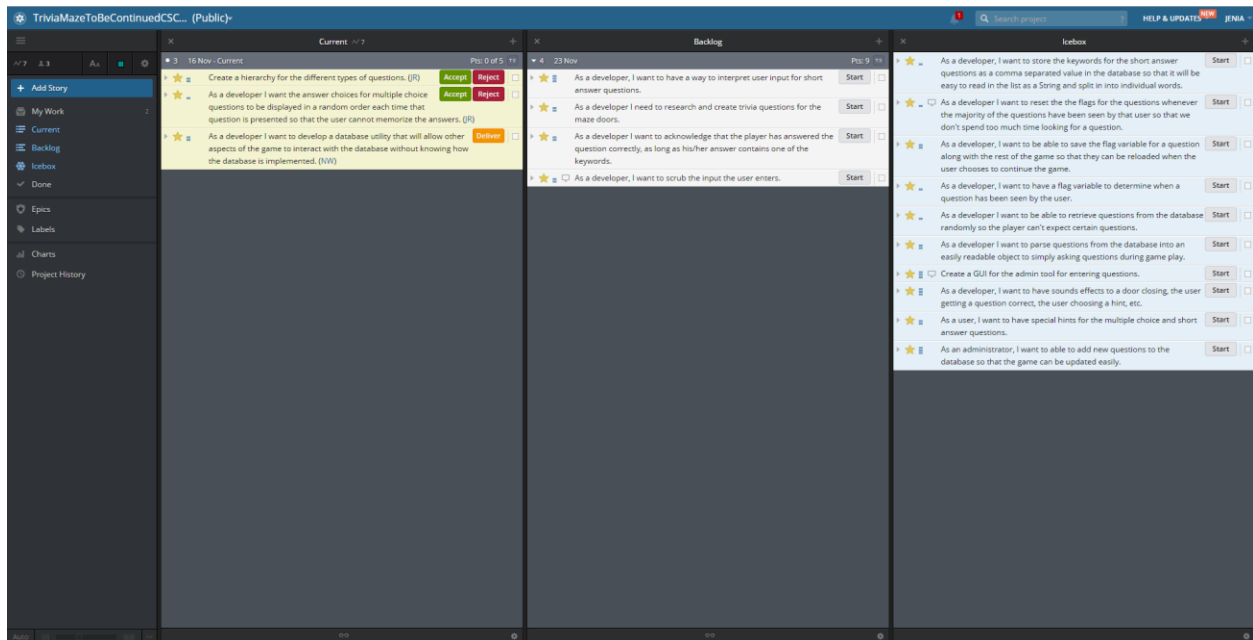
End of 2nd Iteration

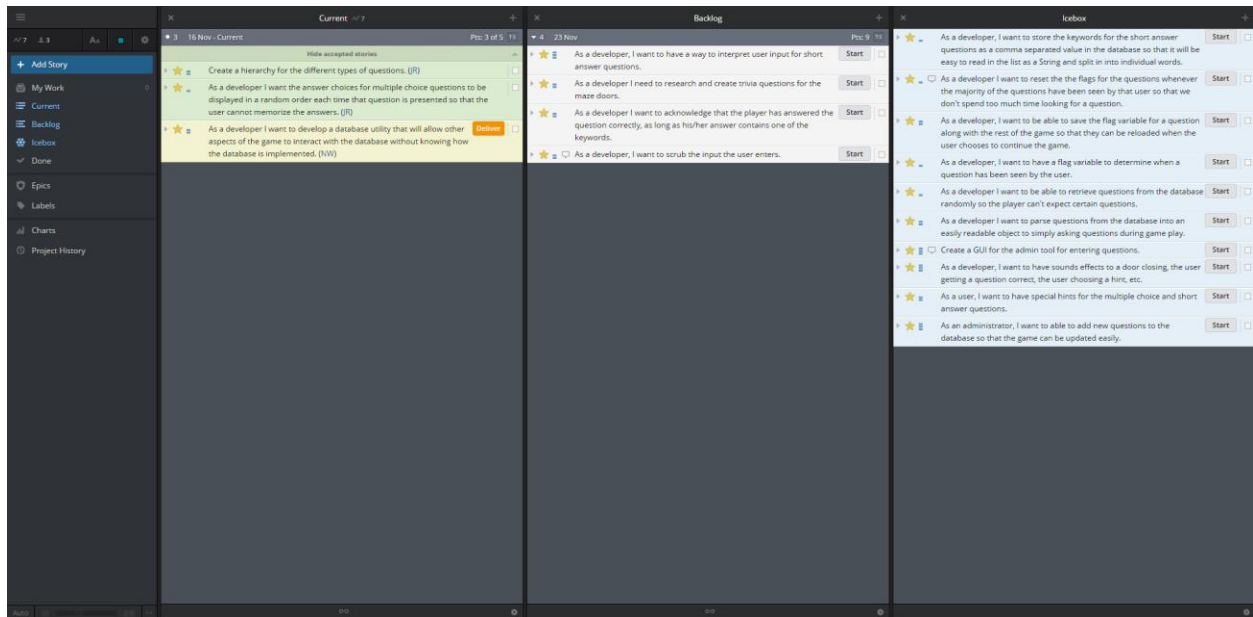Jenia's work for Iteration 2:

Finishing stories:

Delivering user stories:

Stories accepted:

Committing the changes to GitHub:
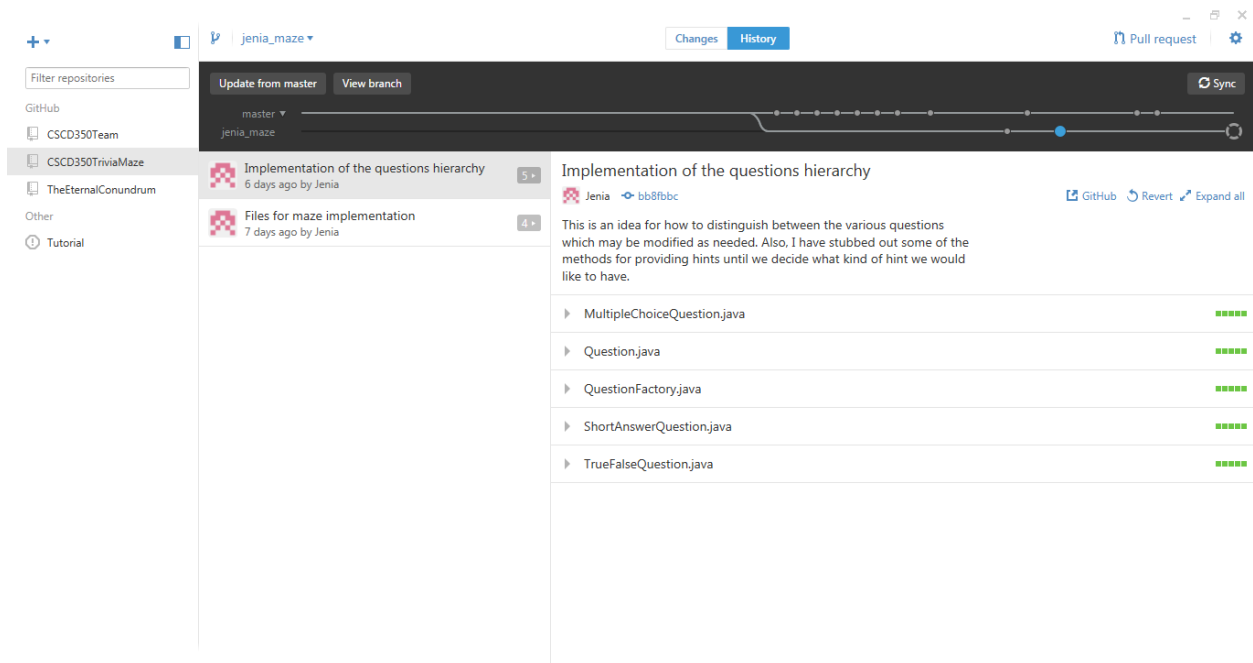
Code Snippets:


```java
14 +
15 + public class MultipleChoiceQuestion extends Question
16 + {
17 +     private String[] choices; // a special object for an answer choice may be used here
18 +
19 +     public MultipleChoiceQuestion()
20 +     {
21 +         super(null, null);
22 +         super.setPattern("((?i)[abcd])");
23 +     }
24 +
25 +     public MultipleChoiceQuestion(String question, String correctAnswer, String[] choices)
26 +     {
27 +         super(question, correctAnswer);
28 +         this.choices = choices;
29 +         super.setPattern("((?i)[abcd])");
30 +     }
31 +
32 +     public String[] getChoices()
33 +     {
34 +         return choices;
35 +     }
36 +
37 +     public void setChoices(String[] choices)
38 +     {
39 +         this.choices = choices;
40 +     }
```



```java
 5 + /**
 6 +  * Question.java
 7 +  * Author: Jenia Rousseva
 8 +  * Revision: N/A
 9 +  * Date: 11/09/2015
10 +  * This file provides the implementation of an abstract class for a generic
11 +  * question. All questions must have a question itself and a correct answer.
12 +  * The class also contains an abstract method called hint.
13 +  * Reference:
14 +  *
15 +  * https://docs.oracle.com/javase/tutorial/java/IandI/abstract.html
16 +  */
17 +
18 + public abstract class Question
19 + {
20 +     private String question;
21 +     private String correctAnswer;
22 +     private String pattern;  // a regex pattern for input validation
23 +
24 +     public Question(String question, String correctAnswer)
25 +     {
26 +         this.question = question;
27 +         this.correctAnswer = correctAnswer;
28 +         this.pattern = "(.*?)"; //matches anything
29 +     }
30 +
31 +     public String getQuestion()
32 +     {
```

Filter repositories

GitHub
- CSCD350Team
- CSCD350TriviaMaze
- TheEternalConundrum

Other
- Tutorial

Update from master View branch   Sync

master ▾
jenia_maze

Implementation of the questions hierarchy
6 days ago by Jenia   5 ▸

Files for maze implementation
7 days ago by Jenia   4 ▸

Implementation of the questions hierarchy
Jenia ⊸ bb8fbbc   GitHub ↺ Revert ⤢ Collapse all

```
11 +
12 + public class QuestionFactory
13 + {
14 +     public QuestionFactory() {}
15 +
16 +     public Question createQuestion(String type) //may change the type of the parameter to
               int or char
17 +     {
18 +         Question question = null;
19 +
20 +         if (type.equals("shortAnswer"))
21 +         {
22 +             question = new ShortAnswerQuestion();
23 +         }
24 +         else if (type.equals("multipleChoice"))
25 +         {
26 +             question = new MultipleChoiceQuestion();
27 +         }
28 +         else if (type.equals("trueFalse"))
29 +         {
30 +             question = new TrueFalseQuestion();
31 +         }
32 +         return question;
33 +     }//end createQuestion
34 +
35 + }//end QuestionFactory
```

Implementation of the questions hierarchy
Jenia ⊸ bb8fbbc   GitHub ↺ Revert ⤢ Collapse all

```
 4 +  * ShortAnswerQuestion.java
 5 +  * Author: Jenia Rousseva
 6 +  * Revision: N/A
 7 +  * Date: 11/09/2015
 8 +  * This file provides the implementation of a short answer question object.
 9 +  * These types of questions differ from other types in that a set of keyword(s)
10 +  * will be used to match the user's answer to the correct answer.
11 +  * Reference:
12 +  *
13 +  * https://docs.moodle.org/22/en/Regular_Expression_Short-Answer_question_type
14 +  * https://docs.moodle.org/29/en/Short_answer_analysis
15 +  */
16 +
17 + import java.util.Scanner;
18 +
19 + public class ShortAnswerQuestion extends Question
20 + {
21 +     String [] keywords; // acceptable answer choices; this can be changed to only one
               keyword
22 +     String hint; // may use a special object
23 +
24 +     public ShortAnswerQuestion()
25 +     {
26 +         super(null, null);
27 +     }
28 +
29 +     public ShortAnswerQuestion(String question, String correctAnswer, String[] keywords)
```

Filter repositories

GitHub

📄 CSCD350Team

📄 CSCD350TriviaMaze

📄 TheEternalConundrum

Other

ⓘ Tutorial

Update from master View branch

⟳ Sync

master ▾
jenia_maze

Implementation of the questions hierarchy
6 days ago by Jenia   5 ▸

Files for maze implementation
7 days ago by Jenia   4 ▸

Implementation of the questions hierarchy

👾 Jenia �þ bb8fbbc   ↗ GitHub ↺ Revert ⤢ Collapse all

```
15  +
16  + public class TrueFalseQuestion extends Question
17  + {
18  +     public TrueFalseQuestion()
19  +     {
20  +         super(null, null);
21  +         super.setPattern("((?i)[tf])");
22  +     }
23  +
24  +     public TrueFalseQuestion(String question, String correctAnswer)
25  +     {
26  +         super(question, correctAnswer);
27  +         super.setPattern("((?i)[tf])");
28  +     }
29  +
30  +     @Override
31  +     public void printQuestion()
32  +     {
33  +         super.printQuestion();
34  +         System.out.println("\n\nChoose T or F:\n");
35  +     }
36  +
37  +     @Override
38  +     public void hint() // cannot have hints here
39  +     {
40  +         // TODO Auto-generated method stub
41  +     }
42  +
```

🕐 **10** commits   ⑂ **4** branches   ◇ **0** releases   👥 **3** contributors

Your recently pushed branches:

⑂ **davidBranch** (less than a minute ago)   ⇅ **Compare & pull request**

⇅   Branch: **davidBranch** ▾   **CSCD350TriviaMaze** / +   ☰

This branch is 2 commits ahead, 4 commits behind master.   ⑂ Pull request   🗋 Compare

👥 **David2Walker** Added Trivial Util + save feature		Latest commit 535dd63 27 seconds ago
📁 Iteration 1	Nick	9 days ago
🗎 Item.java	Added Trivial Util + save feature	26 seconds ago
🗎 Player.java	Added Trivial Util + save feature	26 seconds ago
🗎 README.md	Revert "Revert "Initial commit""	9 days ago
🗎 TriviaMain.java	Added Trivial Util + save feature	26 seconds ago
🗎 triviaUtil.java	Added Trivial Util + save feature	26 seconds ago

☰ 🔍 Search project ?

✕ Current ∿ 12 +

≡
∿ 12 ⌖ 3 Aᴬ ■ ⚙

+ Add Story

🔘 My Work   0
☰ Current
☰ Backlog
❄ Icebox
✓ Done

🛡 Epics
🏷 Labels

📊 Charts
🕐 Project History

● 2 9 Nov - Current   Pts: 3 of 10 T$

Hide accepted stories   ⌃

▸ ⭐ ≡ 💬 As a user I want to be able to save my game so that I can continue later (DA)   ☐

▸ ⭐ ▪ Create a hierarchy for the different types of questions. (JR)   **Finish** ☐

▸ ⭐ ▪ As a developer I want the answer choices for multiple choice questions to be displayed in a random order each time that question is presented so that the user cannot memorize the answers. (JR)   **Finish** ☐

▸ ⭐ ▪ As a developer I want to develop a database utility that will allow other aspects of the game to interact with the database without knowing how the database is implemented. (NW)   **Deliver** ☐

▸ ⭐ ▪ As a developer I need to research and create trivia questions for the maze doors.   **Start** ☐

✕ Backlog +

✕ Icebox +

▸ ⭐ ▪ As a developer I want to be able to retrieve questions from the database randomly so the player can't expect certain questions.   **Start** ☐

▸ ⭐ ▪ As a developer I want to parse questions from the database into an easily readable object to simply asking questions during game play.   **Start** ☐

▸ ⭐ ▪ Create a GUI for the admin tool for entering questions.   **Start** ☐

▸ ⭐ ▪ As a developer, I want to have a way to interpret user input for short answer questions.   **Start** ☐

▸ ⭐ ▪ 💬 As a developer, I want to scrub the input the user enters.   **Start** ☐

▸ ⭐ ▪ As a developer, I want to have sounds effects to a door closing, the user getting a question correct, the user choosing a hint, etc.   **Start** ☐
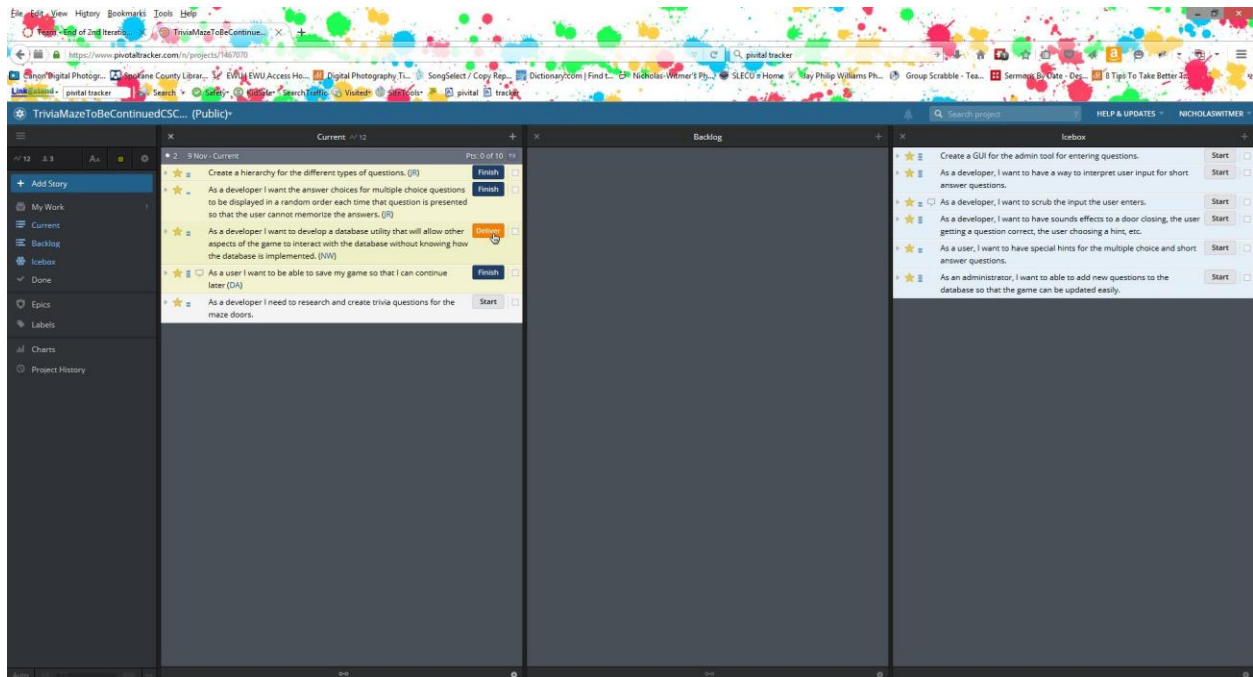
▸ ⭐ ▪ As a user, I want to have special hints for the multiple choice and short answer questions.   **Start** ☐
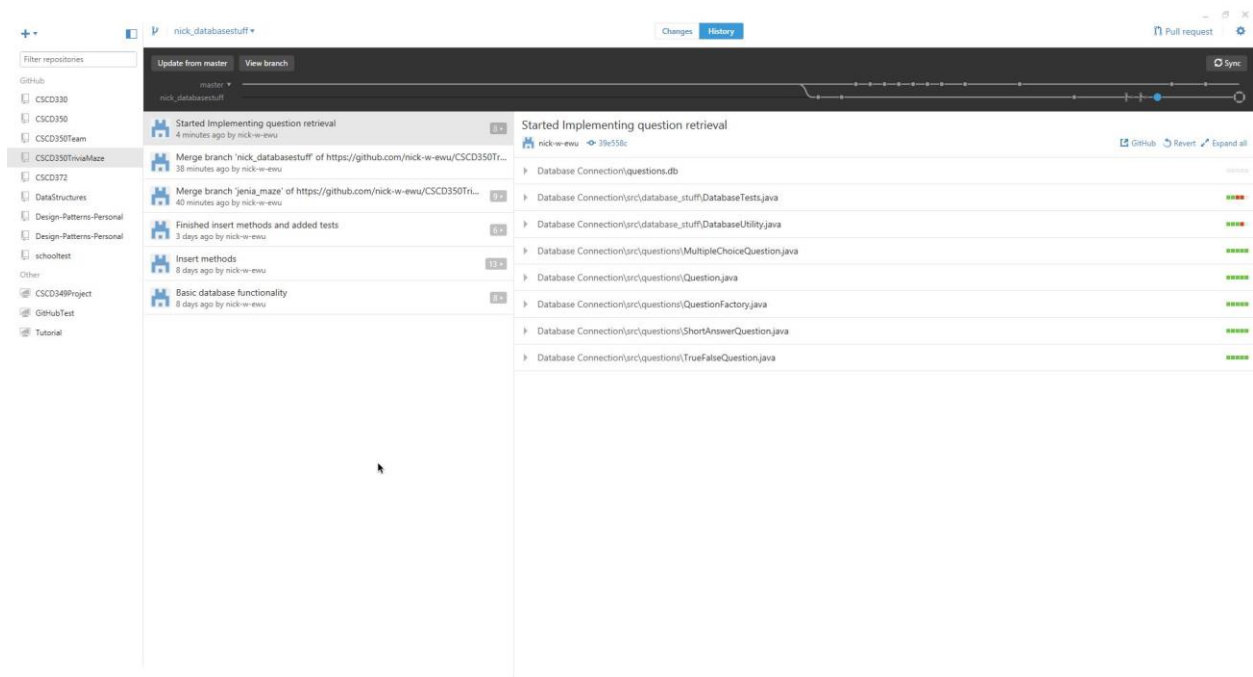
▸ ⭐ ▪ As an administrator, I want to able to add new questions to the database so that the game can be updated easily.   **Start** ☐
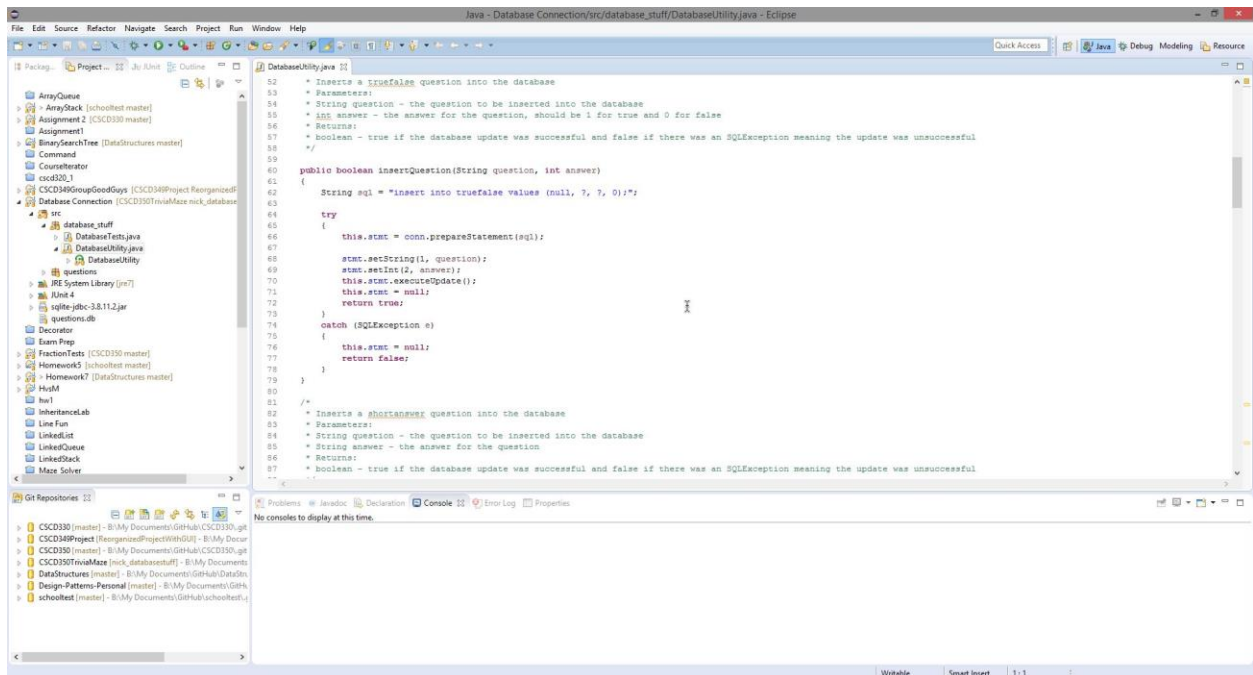
Pivotal Tracker finished user stories:

Nick's GitHub Branch

Insert functionality for in the DatabaseUtility


```java
 52    * Inserts a truefalse question into the database
 53    * Parameters:
 54    * String question - the question to be inserted into the database
 55    * int answer - the answer for the question, should be 1 for true and 0 for false
 56    * Returns:
 57    * boolean - true if the database update was successful and false if there was an SQLException meaning the update was unsuccessful
 58    */
 59
 60    public boolean insertQuestion(String question, int answer)
 61    {
 62        String sql = "insert into truefalse values (null, ?, ?, 0);";
 63
 64        try
 65        {
 66            this.stmt = conn.prepareStatement(sql);
 67
 68            stmt.setString(1, question);
 69            stmt.setInt(2, answer);
 70            this.stmt.executeUpdate();
 71            this.stmt = null;
 72            return true;
 73        }
 74        catch (SQLException e)
 75        {
 76            this.stmt = null;
 77            return false;
 78        }
 79    }
 80
 81    /*
 82    * Inserts a shortanswer question into the database
 83    * Parameters:
 84    * String question - the question to be inserted into the database
 85    * String answer - the answer for the question
 86    * Returns:
 87    * boolean - true if the database update was successful and false if there was an SQLException meaning the update was unsuccessful
```

Tests for insert methods


```java
  7
  8 public class DatabaseTests
  9 {
 10
 11    DatabaseUtility test;
 12
 13    @Before
 14    public void setUp()
 15    {
 16        test = new DatabaseUtility();
 17    }
 18
 19    @Test
 20    public void testInsertTrueFalse()
 21    {
 22        assertTrue(test.insertQuestion("Will it rain today", 1));
 23        assertFalse(test.insertQuestion(null, 0));
 24    }
 25
 26    @Test
 27    public void testInsertShortAnswer()
 28    {
 29        assertTrue(test.insertQuestion("Will it rain today", "Maybe, it could", "maybe, probably"));
 30        assertFalse(test.insertQuestion("What is my name", null, null));
 31    }
 32
 33    @Test
 34    public void testInsertMultipleChoice()
 35    {
 36        assertTrue(test.insertQuestion("How many children are in Mary Poppins", 2, "55", "2", "7", "25"));
 37        assertFalse(test.insertQuestion("What is my name", 3, "Nick", "Fish", null, "Nothing"));
 38    }
 39
 40    @Test
 41    public void testInsertHint()
 42    {
```