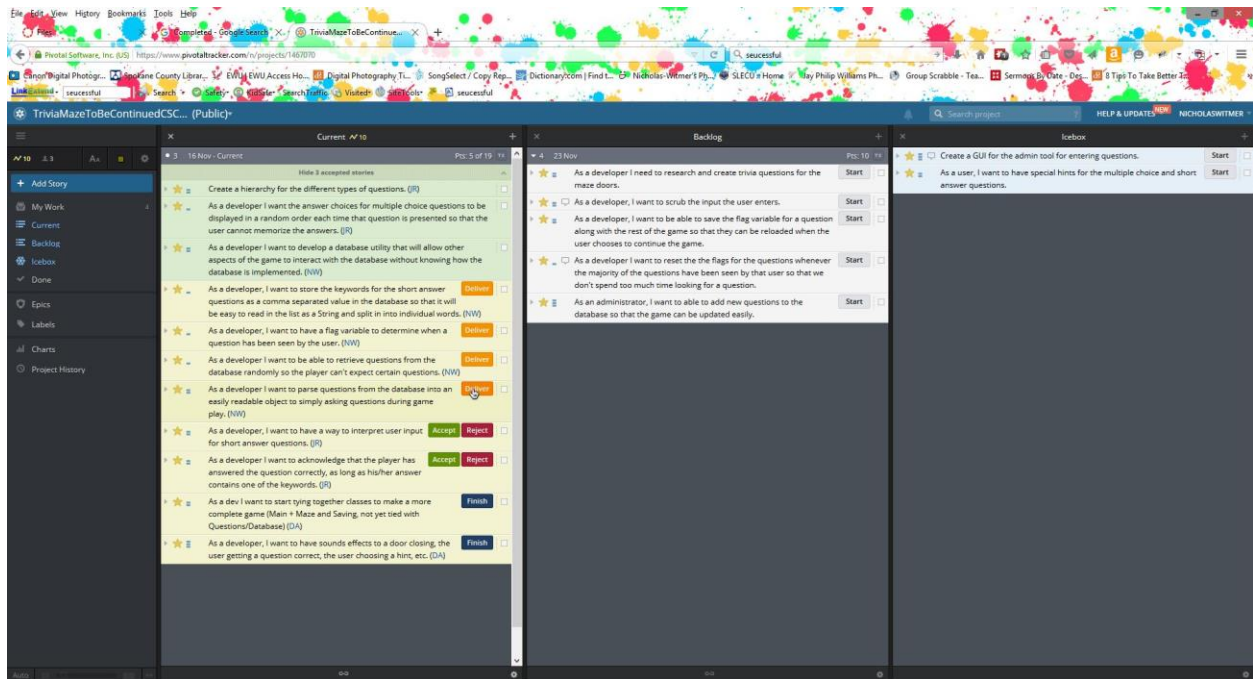
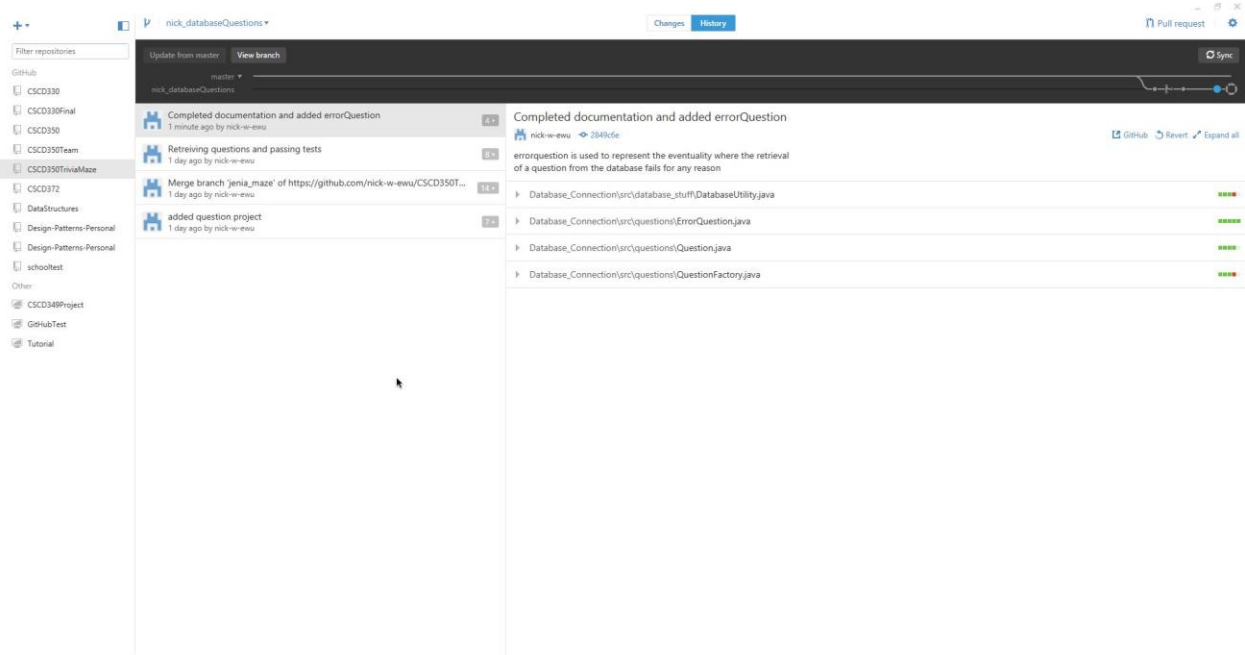


Pivotal Tracker Nick's stories finished



GitHub branch, created a new one to resolve some issues caused by the power outage and trying to continue working on the project



Retrieval of questions from the database in the DatabaseUtility

```
DatabaseUtility.java | Question.java | ErrorQuestion.java | TrueFalseQuestion.java | QuestionFactory.java
219 public Question retrieveQuestion(String type)
220 {
221     String sql = "";
222     Question toReturn = null;
223     QuestionFactory factory = new QuestionFactory();
224     switch(type.toLowerCase())
225     {
226         case("truefalse"):
227             sql = "select * from truefalse where answered = 0 ORDER BY RANDOM() LIMIT 1;";
228             break;
229         case("multiplechoice"):
230             sql = "select * from multiplechoice where answered = 0 ORDER BY RANDOM() LIMIT 1;";
231             break;
232         case("shortanswer"):
233             sql = "select * from shortanswer where answered = 0 ORDER BY RANDOM() LIMIT 1;";
234             break;
235         default:
236             throw new IllegalArgumentException("You have provided an invalid question type");
237     }
238     try
239     {
240         this.stmt = conn.prepareStatement(sql);
241         ResultSet results = stmt.executeQuery();
242         switch(type.toLowerCase())
243         {
244             case("truefalse"):
245                 toReturn = processTrueFalse(results, factory);
246                 break;
247             case("multiplechoice"):
248                 toReturn = processMultipleChoice(results, factory);
249                 break;
250             case("shortanswer"):
251                 toReturn = processShortAnswer(results, factory);
252                 break;
253         }
254         return toReturn;
255     }
256     catch (SQLException e)
257     {
258         return factory.createQuestion(ERROR);
259     }
260     finally
261     {
262         try
263         {
264             this.stmt.close();
265         }
266         catch (SQLException e1)
267         {
268         }
269     }
270 }
```

Processing of ResultSets retrieved from the database in above function

```
DatabaseUtility.java | Question.java | ErrorQuestion.java | TrueFalseQuestion.java | QuestionFactory.java
282 private Question processTrueFalse(ResultSet results, QuestionFactory factory)
283 {
284     try
285     {
286         int id = results.getInt("question_id");
287         String question = results.getString("question");
288         String answer = results.getString("answer");
289         Question newQuestion = factory.createQuestion(TRUEFALSE);
290         newQuestion.setCorrectAnswer(answer);
291         newQuestion.setQuestion(question);
292         return newQuestion;
293     }
294     catch (SQLException e)
295     {
296         return factory.createQuestion(ERROR);
297     }
298 }
299
300 /*
301  * Processes the ResultSet for a shortanswer question retrieved from the database
302  * Parameters:
303  * ResultSet results - the result set returned from the select query in retrieveQuestion()
304  * QuestionFactory factory - a QuestionFactory to create the question to be returned
305  * Returns:
306  * Question - returns a Question containing the data retrieved from the database if successful and returns an ErrorQuestion if the retrieval was unsuccessful
307  */
308
309 private Question processShortAnswer(ResultSet results, QuestionFactory factory)
310 {
311     try
312     {
313         int id = results.getInt("question_id");
314         String question = results.getString("question");
315         String answer = results.getString("answer");
316         String words = results.getString("keywords");
317         String[] keywords = words.split(",");
318
319         Question newQuestion = factory.createQuestion(SHORTANSWER);
320         newQuestion.setCorrectAnswer(answer);
321         newQuestion.setChoices(keywords);
322         newQuestion.setQuestion(question);
323         return newQuestion;
324     }
325     catch (SQLException e)
326     {
327         return factory.createQuestion(ERROR);
328     }
329 }
330
331 /*
332  */
333 }
```

Testing for question retrieval

```
DatabaseTests.java
23 @Test
24 public void testInsertTrueFalse()
25 {
26     assertTrue(test.insertQuestion("Will it rain today", "t"));
27     assertFalse(test.insertQuestion(null, "t"));
28 }
29
30 @Test
31 public void testInsertShortAnswer()
32 {
33     assertTrue(test.insertQuestion("Will it rain today", "Maybe, it could", "maybe,probably"));
34     assertFalse(test.insertQuestion("What is my name", null, null));
35 }
36
37 @Test
38 public void testInsertMultipleChoice()
39 {
40     assertTrue(test.insertQuestion("How many children are in Mary Poppins", "2", "55", "2", "7", "25"));
41     assertFalse(test.insertQuestion("What is my name", "3", "Nick", "Fish", null, "Nothing"));
42 }
43
44 @Test
45 public void testInsertHint()
46 {
47     assertTrue(test.insertHint(id++, "shortanswer", "its possible"));
48     assertFalse(test.insertHint(0, null, null));
49 }
50
51 //Will fail on first run if the database is empty
52 @Test
53 public void testretrieveQuestions()
54 {
55     Question q1 = test.retrieveQuestion("shortanswer");
56     Question q2 = test.retrieveQuestion("truefalse");
57     Question q3 = test.retrieveQuestion("multiplechoice");
58     String[] choices = {"maybe", "probably"};
59     assertEquals(choices, q1.getChoices());
60     assertEquals("Will it rain today", q1.getQuestion());
61     assertEquals("Maybe, it could", q1.getCorrectAnswer());
62
63     assertEquals("t", q2.getCorrectAnswer());
64     assertEquals("Will it rain today", q2.getQuestion());
65
66     String[] options = {"55", "2", "7", "25"};
67     assertEquals("How many children are in Mary Poppins", q3.getQuestion());
68     assertEquals("2", q3.getCorrectAnswer());
69     assertEquals(options, q3.getChoices());
70 }
71
72
73
74
```