

Iteration: 2 (02/03 - 08/03)

Alex Taylor - amt22@aber.ac.uk

March 18, 2017

Version 0.5 (Draft)

Contents

| | | |
|----------|---|----------|
| 1 | Story: They can create a new quiz in the backend | 2 |
| 1.1 | Analysis - Breakdown of Tasks | 2 |
| 1.2 | Design | 2 |
| 1.3 | Implementation | 2 |
| 1.4 | Testing | 2 |
| 2 | Story: They can edit an existing quiz they own | 3 |
| 2.1 | Analysis - Breakdown of Tasks | 3 |
| 2.2 | Design | 3 |
| 2.3 | Implementation | 3 |
| 2.4 | Testing | 3 |
| 3 | Non-Story Work | 4 |
| 3.1 | Seeding | 4 |
| 3.2 | Quiz Description | 4 |

1 Story: They can create a new quiz in the back-end

1.1 Analysis - Breakdown of Tasks

- Need quiz/new form to add name
- Add validation
- This then redirects to quiz.show for this new quiz
- This page needs a button that can add a question

1.2 Design

The design is split into two parts, design for the quiz/new page and a design for the question/new page: (insert images for these, maybe some form of activity diagram?)

1.3 Implementation

Added create page which has a form, added validation to it
Could do validation with javascript.

1.4 Testing

Dusk tests:

1. Test to create a new quiz
2. Test the HTML validation for creating a new quiz
3. Test backend validation of quizzes
4. Test a quiz contains questions that have all been added before hand
5. Test to add a question to a quiz
6. Test to make sure the question page contains the relevent information

A problem encountered in this set of tests is the speed of testing. Running these tests takes about a minute. The reason for this is that the tests use DatabaseMigrations rather than DatabaseTransactions, meaning that the database is migrated for every test. Using transactions would reduce the time taken by only doing a migration at the start and then using a transaction for each test. Alternatively having a pre migrated test database on which you run transactions would work too, but this would mean any time you change your database, you would have to reember to migrate the test database.

After attempting to use the transactions it appears as though they are not usable within Dusk. This is because Dusk is running in another process and migrations is the only choice (site this with <https://github.com/laravel/dusk/issues/110>)

2 Story: They can edit an existing quiz they own

2.1 Analysis - Breakdown of Tasks

- There is an edit button on quizzes that links to quiz/edit
- You can edit the name of the quiz
- You can remove questions
- You can add questions
- You can edit the content of a question

2.2 Design

There was not much design for this stage as it only really adds an edit page which should look the same as the pages in the above story

2.3 Implementation

2.4 Testing

Dusk Tests:

1. Test the deletion of a quiz
2. Test the deletion of a question on a quiz
3. Test the edit page of a quiz
4. Test the edit page of a question

3 Non-Story Work

3.1 Seeding

Some more data needed to be seeded for this section of work, the question data. This involved creating some new Model Factories and using them in the seeders correctly. One issue was trying to create many questions for individual quizzes, but this was overcome using some very basic looping and calling the model factories in the right places.

3.2 Quiz Description

It was decided that quizzes should probably have descriptions attached to them, in case the lecturer needs reminding of what it is in 6 months time. This involved creating a new migration and simply adding a column to the table.