

Iteration: 1 (23/02 - 01/03)

Alex Taylor - amt22@aber.ac.uk

March 2, 2017

Version 0.6 (Draft)

Contents

1	Story: Admins can create quizzes via the website	2
1.1	Analysis - Breakdown of Tasks	2
2	Story: Admins can log into the backend	3
2.1	Analysis - Breakdown of Tasks	3
2.2	Design	3
2.3	Implementation	3
2.4	Testing	3
2.5	Extras	3
3	Story: They are presented a list of their quizzes	4
3.1	Analysis - Breakdown of Tasks	4
3.2	Design	4
3.3	Implementation	4
3.4	Testing	4
3.5	Extras	4
4	Non-Story Work	5
4.1	Database Work	5
4.2	Seed Data	5
4.3	Layout Changes	5

1 Story: Admins can create quizzes via the website

1.1 Analysis - Breakdown of Tasks

This story is quite large and should be broken down into several substories:

- (3) Admins can log into a backend
- (2) They are presented a list of their quizzes
- (5) They can create a new quiz in the backend
- (3) They can edit an existing quiz they own

These have been added to the story list document.

2 Story: Admins can log into the backend

2.1 Analysis - Breakdown of Tasks

- Create users table
- Add login page
- Add register page

2.2 Design

Design was limited due to the automated builder. However it added some view files and a default HomeController for a basic homepage.

2.3 Implementation

Implementing this was far easier than originally anticipated, Laravel comes with the needed tables out of the box and has a command to run that sets up simple auth for users: *php artisan make:auth*

2.4 Testing

A simple test to make sure that when you try to visit /home, it is redirected to /login. Most testing here will be left because it is auto generated and feels like a waste to test something wirtten by the original framework developers.

3 Story: They are presented a list of their quizzes

3.1 Analysis - Breakdown of Tasks

- Make the homepage the QuizController rather than the HomeController
- Check and get the user who is logged in
- Display a list of quizzes for that user

3.2 Design

The HomeController was removed in this period of work and the QuizController used in its place.

3.3 Implementation

Quite an easy amount of work, changing the controller was a simple change to the routes and then updating the quiz view file to use the same layout as the original home views. To get the user, a helper function is provided: *auth()->user()* which gets the user object. Obtaining the id from this is simple and then using an Eloquent ORM call it is easy to find all the quizzes owned by that user.

3.4 Testing

Testing withheld until DB testing sorted out

4 Non-Story Work

4.1 Database Work

Some initial work that is needed for almost all the stories is having a working database set up to store the users, quizzes, questions etc. Seeing as the amount of work to setup all the tables and their relationships would not take long, it was decided that this could be done all at once at the start of the iteration.

While creating these tables it was possible to create the controllers needed within the application at the same time using: *php artisan make:model *name* -mc*

4.2 Seed Data

Because of the amount of changes to the database that were being made, the tables were repeatedly wiped and seeding some data was needed. To do this some seeders were generated with *php artisan make: seed *name**. These were created under `database/seeds/` and simply required creating new objects of the desired Model and adding the various fields as parameters of the objects. These objects are then saved to the database. These seeders can be run when a migration is called such that the data is replaced as soon as its lost.

4.3 Layout Changes

The initial *make:auth* command created a default home page with a menu bar and some basic styling. This styling was created with Bootstrap and looks quite nice so the basic styling has been kept. This layout was modified somewhat to add some menu options that persist across pages. This layout is then used by all the backend pages created by extending it.