



# Quiz Manager Submission: **Design Documentation**

Prepared by: Linna Trieu, Associate Software Engineer at Paddle

Qualification: Level 4 Software Development Apprenticeship with Makers Academy

Prepared for: BCS, Digital Industries Apprenticeship

Date: September 2020

# DESIGN DOCUMENT

1.1. Project Plan

1.2. Requirements Analysis

1.3. User Interface Design & Wireframes (see file)

1.4. Database Design & ERD (see file)

1.5. User Stories

# PROJECT PLANNING

## Approach

I decided to plan my project immediately after reading the project brief, before getting into any project-specific activities. To plan I took two approaches:

1. Deliverable and activity focused - From the project brief, I went through a ‘post-it’ note exercise listing all the deliverables required. Following this, I went through each deliverable identifying the high level activities and tasks that would be required in order to meet that deliverable. This exercise articulated exactly what I would need to produce over the course of the project time. Refer to Figure 1 for a photo of the planning session.
2. Deadline and rough timings - With the deliverables and activity breakdown, I have roughly estimated the time/effort required per activity. As this is relatively low level, I feel the estimates will be more accurate and give me more confidence in the plan. The activities breakdown should also serve as a helpful checklist of activities I can use to structure my day. Refer to Figure 2 for a screenshot of the Project Plan excel doc.
3. Day to day granular tasks - I will plan my day using a Kanban board - Trello. This will list out granular tasks for me to complete.

Based on this, I have my high level plan with an activities breakdown and rough time estimations for how long it should take.

I am breaking down my project time into daily sprints. Each sprint will be planned according to the user requirements and have a sprint objective based on what user stories will be aimed at. Matching my sprints to the requirements will make sure I am covering all the user requirements, if not more.

## Project management methodology

To plan the project, I will use Agile project management methodology. I decided to opt for an agile approach, instead of Waterfall project management, as Agile is considered a superior approach for software development projects and I have the most experience with Agile methodologies on a day-to-day basis in my job as a Junior Software Engineer at Paddle.

I will lean on agile fundamentals to take an iterative approach to my project, focused on continuous releases and user value.

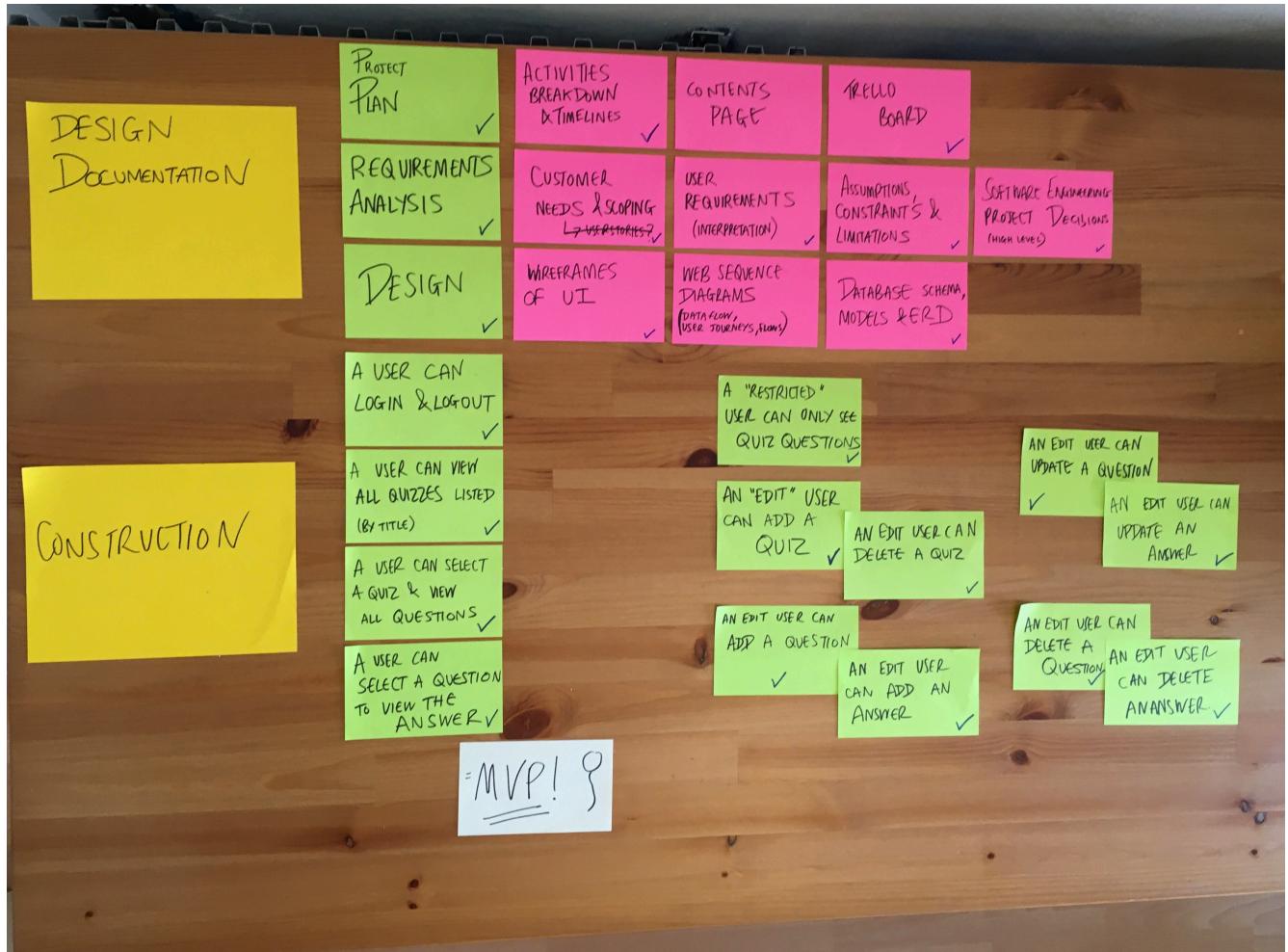
## Ways of Working

- Daily sprints, with a start and end of day review going through the Kanban board and the user requirements.
- Retrospective at the end of each sprint; what went well and what didn't go well.

## Sprint Planning Process

- Please refer to Figure 3 for an example screenshot of my Trello board, used in the sprint planning process.
- In terms of Sprint Planning, I broke down all the higher level activities listed from the Excel planning doc into granular tasks and activities.
- I have created these into Kanban (Trello) tickets, and tickets are grouped based upon on what User Story they belong to.
- The tickets will be added into the '*Backlog*' column by default.
- On a daily basis, my plan is to define the goals/objectives for the current sprint (e.g. 'a user can login and logout'). And based on this I will plan what Trello tickets can be pulled into the current sprint.
- This means physically moving the tickets from the '*Backlog*' column into the '*To Do (in current sprint)*' column.
- In accordance with Agile Kanban practices; as I progress through the work, I will physically move the ticket along with me from left-to-right of the board. Moving a ticket from '*To Do*' into '*In Progress*' when I am working on the ticket; '*Ready for Testing*' when the ticket has been developed and is ready to be manually tested; and finally '*Done*' when the ticket has passed manual testing and is considered finished.

**Figure 1. Photo of white boarding & 'post-it' planning session - this initial session broke down the project into distinct deliverables**



## LINNA TRIEU: DESIGN DOCUMENT

---

**Figure 2. Screenshot of the Project Plan & Activities document**

- Please refer in full to the Excel at file ref '1.1.project\_plan\_activities'

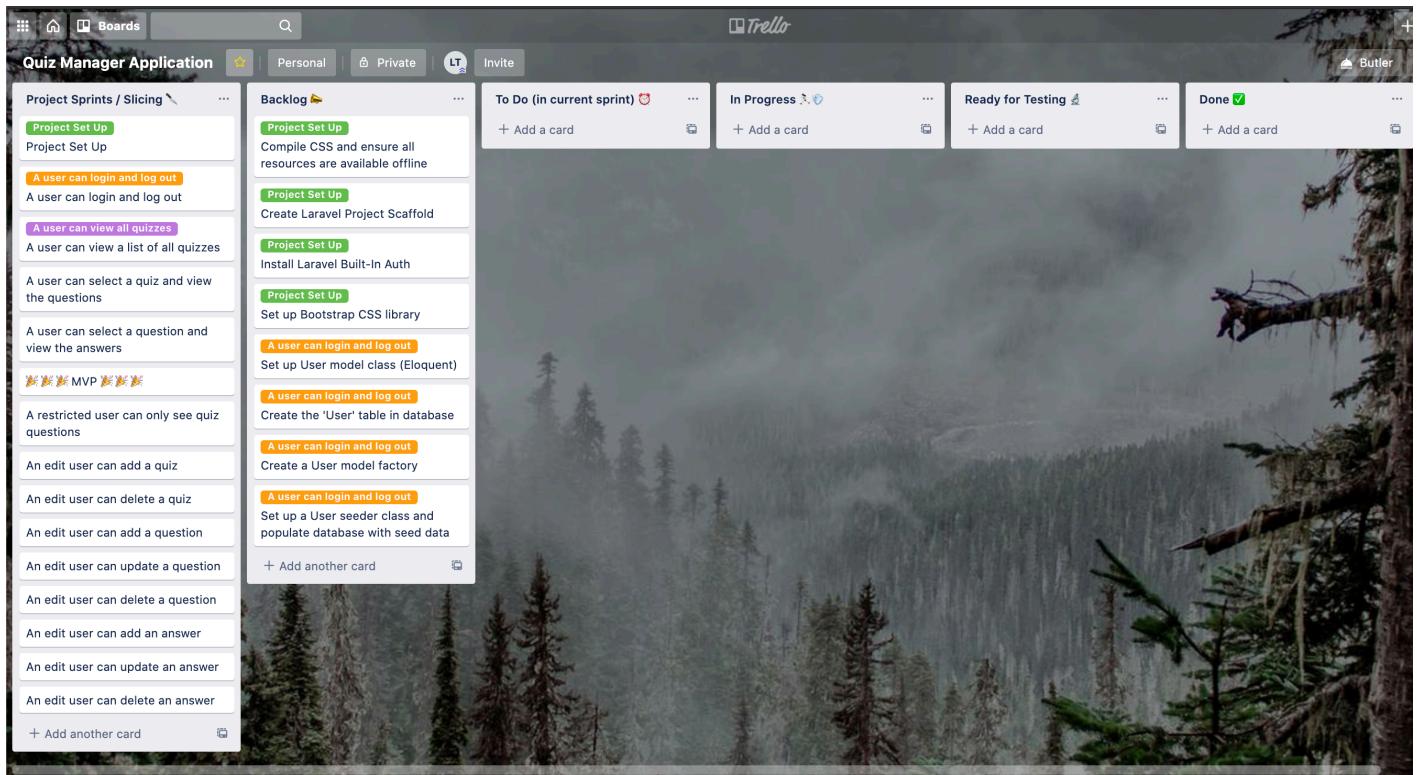
The screenshot shows a Microsoft Excel spreadsheet titled "1.1.project\_plan\_activities". The spreadsheet is organized into several sections representing different phases of the project:

- Phase 1: Design (Sprint #1)** (Rows 1-8):
  - Deliverable:** 1. Design
  - Activity name:** Quiz Manager Project Planning, Business Requirement & Assumptions assessment, UI Design and Wireframes, Database Design and ERD, User Stories analysis, Establish Project Ways of Working.
  - Date:** Tues 8th Sept
  - Estimated (hr):** 8.00
- Phase 2: Construction (Sprint #2)** (Rows 9-16):
  - Deliverable:** 2. Construction
  - Activity name:** Daily sprint planning, Daily product backlog grooming in Trello tickets, Project Infrastructure (Create application scaffold structure - laravel, auth, bootstrap, Set up database and configure with application).
  - Date:** Wed 9th Sept
  - Estimated (hr):** 8.00
- Phase 3: MVP (Sprint #3)** (Rows 17-39):
  - Deliverable:** MVP
  - Activity name:** A user can login and log out (Create users database table, model, factory, seeder, controller classes, Add automated tests), A user can view the list of quizzes (Create quiz database table, model, factory, seeder, controller classes, Create the blade webpage, http route, controller logic), A user can select a quiz and view the questions (Create question database table, model, factory, seeder, controller classes, Create the blade webpage, http route, controller method, Handle foreign key logic, Add automated tests), A user can view answers to questions (Update view, handle javascript logic to show answers on click).
  - Date:** Thurs 10th Sept
  - Estimated (hr):** 8.50
- Phase 4: Testing (Sprint #4)** (Rows 40-52):
  - Deliverable:** 3. Testing
  - Activity name:** A restricted user can only see quiz questions, An edit user can add a quiz, An edit user can delete a quiz.
  - Date:** Fri 11th Sept
  - Estimated (hr):** 8.50
- Phase 5: Documentation (Sprint #5)** (Rows 53-64):
  - Deliverable:** 4. Documentation
  - Activity name:** Daily sprint planning, Daily product backlog grooming in Trello tickets, An edit user can add a question and its answers, An edit user can delete a question and answer.
  - Date:** Mon 14th Sept
  - Estimated (hr):** 8.50
- Phase 6: Final Sprint (Sprint #6)** (Rows 65-66):
  - Deliverable:** 4. Documentation
  - Activity name:** Sprint #7 - FINAL SPRINT, Daily sprint planning.
  - Date:** Tue 15th Sept
  - Estimated (hr):** 8.50
- Phase 7: Testing (Sprint #7)** (Row 67):
  - Deliverable:** 3. Testing
  - Activity name:** Testing documentation.
  - Date:** Wed 16th Sept
  - Estimated (hr):** 4.00
- Phase 8: Documentation (Sprint #8)** (Row 68):
  - Deliverable:** 4. Documentation
  - Activity name:** Daily sprint planning.
  - Date:** Wed 16th Sept
  - Estimated (hr):** 0.25

At the bottom of the spreadsheet, there are navigation buttons for "Project Plan" and a "+" sign, along with other standard Excel ribbon tabs like Home, Insert, Draw, Page Layout, Formulas, Data, Review, View, Tell me, and a status bar indicating "AutoSave OFF".

**Figure 3. Daily Sprint Planning and Backlog Grooming with Trello board/tickets**

- For more detail, please refer to the Development Diary in section 2, as I documented my daily sprint planning sessions.



# REQUIREMENTS ANALYSIS

## Overview of the Product

- As a software engineer for Webbiskools Ltd ('Webbiskools'), this engineering project will design, build and test the 'Quiz Manager' website.
- Webbiskools Ltd are a software company that provide online educational solutions to our clients.
- Typical customers of Webbiskools, and users of Quiz Manager, are likely to be government clients and large commercial companies, such as universities and the training departments of large corporations.
- The Quiz Manager website will enable the creation and management of quizzes, consisting of multiple-choice questions.

## High Level Customer Needs

The Quiz Manager Website needs to enable the creation and management of quizzes. Based on the business requirements passed on by Webbiskools Ltd:

### User Interface

- The branding of the website needs to be largely generic and encapsulated. This is because my customers will likely re-brand Quiz Manager in their organisation's style, colour schemes and logos, etc. So it must be straightforward for my clients to apply their own bespoke styles.

### User & permissions

- The website will need authentication features to allow for 'known' users. This means the website needs sign in/sign out functionality.
- The website will need a user permission level system, to determine what features of the website are available to what groups (restricted, view, edit levels).
- A user's permission level will determine how much a given user can access in terms of the Quiz Manager's features (detailed further in the functional requirements).

## Quizzes & Questions

- The website will contain a set of quizzes with multiple-choice questions.
- Quiz Manager will include the following functionality:
  - View a list of all quizzes available on Quiz Manager
  - Select a quiz to view all questions in the quiz
  - Select a question to view the multiple-choice options and answer
  - Create a new quiz
  - Delete an existing quiz
  - Add, edit or delete questions in a quiz
  - Add, edit or delete answers to any question

## Out of Scope

- User registration functionality - a set list of users will be pre-configured for this project. There will be no functionality for users to register, reset their password or change their permission level with this version of the website.
- Completing/taking quizzes - there will be a separate website to allow students to sit quizzes with an accompanying mark and grade. This is out of scope for Quiz Manager.

# DETAIL SPECIFIC REQUIREMENTS

*Key for the development process. This analysis details the specific requirements for building the product.*

## System Features

*Functional Requirements - features that are required for a system to function.*

### User and permissions

- Visitors to the Quiz Manager website will either be ‘known’ or ‘anonymous’ users.
- Quiz Manager will only be available for known users.
  - We identify known users as those who can login to the website and therefore use the website’s functionality/features.
  - Anonymous users do not have a user login, and therefore they cannot access Quiz Manager content.
- Our known users will have a user login (such as an email address and password) and a user permission level.
- There will be three user permission levels which will determine how users can use the Quiz Manager website:
  - Edit
  - View
  - Restricted

### Viewing and editing

- Once logged in, a user of any kind is able to see a full list of quizzes available.
- Restricted users must be able to select any quiz from the list, and view the questions in the quiz.
- View users can select and view the quiz as above. They also need to be able to view the answers associated with any give question.
- Edit users have the access of ‘restricted’ and ‘view’ users. In addition, they will be able to:
  - Create a new quiz or delete any existing quizzes.
  - Add or delete any questions or answers

- Edit the text of any question or answers
- Select a quiz from the available

## **User Interface**

*External interface - any specific requirements of the UI.*

- Quiz Manager must provide a graphical user interface.
- End users will interact with Quiz Manager solely through the user interface. End users will have no access to the back end systems, or the database, etc.
- The user interface will interact with the database to store and persist Quiz Manager data, via the back end technology stack.

## **Nonfunctional Requirements**

*Include performance, safety, security, quality.*

- The website will be designed and built to production standards.
- The website should adhere as far as possible to industry best practices. In particular the following programming best practices and standards.
  - Object Orientated Programming
  - SOLID principles
  - Front end - styles encapsulation and reusable components.
  - Responsive design - the application can be used on any device, e.g. browser or mobile devices, etc

# ASSUMPTIONS & DEPENDENCIES

## Introduction

This is a list of assumption and dependencies I have noted whilst going through the project brief and the user requirements.

This is a critical document, as it lists out the factors that could impact my ability to fulfil the project requirements outlined. I am making assumptions that could turn out to be incorrect.

Typically, on a project my priority would be to go through this list of Assumptions in detail with the user/business in order to verify the assumptions and clear up any misinterpretations or ambiguities in the Project Brief. This would ensure I have the same understanding as the stakeholders/end users of the business requirements and objective.

As this is obviously not possible in this scenario, so for this project where there are ambiguities in the project brief, I have made a decision and assumption on what I consider to be an unclear user requirement. These assumptions will be carried forward in the project construction and implementation.

## User and permissions

- What constitutes a ‘known’ user to the website?
  - Assumption: A user that is signed up with the website - a registered user. In this version of the project, this constitutes any one of the users in the pre-configured user data.
- Can anonymous users access the website at all?
  - Assumption: No, anonymous users cannot access the website. Guest users will only be able to see a landing page or login page. If they try to access any specific endpoints, e.g., “/quiz”, they will be re-directed to the login page.
- What can Restricted users see on an individual quiz? Are the multiple-choice options considered as part of the question, or part of the answer?
  - This is relatively vague and it is not explicitly stated in the brief.
  - Assumption: I will make the assumption that the multiple-choice options are part of the answer.
  - Therefore, restricted users will not see the multiple-choice options. They can only view the question titles.

- View and Edit users will be able to see the multiple-choice options by default when they navigate to an individual quiz.
- Assumption: There is a small number of Edit users on the website, therefore the probability of users editing at the same time is negligible.

## Quizzes

- How many Questions are there in a Quiz?
  - This is ambiguous in the brief, there was nothing stated.
  - Assumption: I will assume there is no minimum or maximum limit on the amount of questions per quiz.
  - A quiz can have zero questions, as a user might create a placeholder quiz to return to and add questions later. Likewise, a single quiz may have many questions, and there will be no explicit limit for the end user.
- Does every Question have the same amount of multiple-choice options?
  - The brief mentioned 3-5 multiple choice options per question.
  - Assumption: I have assumed there is a minimum of 3 options and a maximum of 5 options, but each question can differ in exactly how many options they have.
  - If an option is deleted and there are only 3 options, a user must add another option, and is alerted by the application. Likewise if a user tries to add a new option, and there are 5 options already, the user must be alerted and delete an existing option.
- Can a single question be in multiple quizzes? Or can it only belong to one quiz?
  - Assumption: No - each question belongs to a single quiz.
- Questions are multiple-choice. Should there be only one correct answer per question; or can a single question have multiple answers e.g. select the two correct options.
  - Assumption: There can only be one correct answer from the multiple-choice options per question.
- Is the answer revealed by question or all answers are revealed for the full Quiz?
  - Assumption: As this is a Quiz Manager (ie. not for the students taking a quiz itself), I can imagine the functionality of having to find the answers by opening individual question-by-question to be fairly laborious for the educational providers (e.g. teachers, for example).
  - As this is quite vague in the brief, I will make the assumption that the answers for a given quiz will be revealed for all questions in the entire quiz, simultaneously.

- Questions must be indexed, so if it is deleted this may cause questions to be re-indexed
  - I found this requirement to be vague in the brief.
  - Not 100% how to implement this, though I imagine that when a user deletes, say, question 3, from a quiz, the entire quiz should be re-ordered in that what was formerly question 4 will now become the new question 3 and so on. Likewise with the multiple-choice answers.
  - It depends on the importance of this requirement - if this is critical for end-users, I imagine each question in the database could have a data field of 'position'. I could then use this to set and get an exact position of the question in an exact quiz. This could be over-engineered though and it would be database intensive.
  - Perhaps something a bit more cleaner on the front-end like using a HTML ordered lists might work. Something to investigate during implementation.

## Viewing and Editing

- When an edit users deletes a quiz, do they delete all the questions and answers that are associated with the quiz?
  - Assumption: Yes - a quiz and all its associated questions and answers will be deleted from the application and the database.
- If a user creates a new Quiz, do they have to also add all the new questions at the same time?
  - Not stated in brief.
  - Assumption: assume not. I think it makes sense to allow a user to create a quiz (almost as a placeholder) - with them returning to the application to add questions/answers separately at their own pace.
- If a user creates a new Question, do they have to add all multiple-choice options and the answer?
  - Not stated in brief.
  - Assumption: Yes - if a user wants to create a new question, they need to submit all the multiple choice options and its answer (at least 3 options). I think it makes sense to require a user to submit a full question, and there will also be the functionality to edit the question later.
- Can a user submit a single question?
  - Assumption - No, no-one can submit a single question.
  - The question must be associated with a specified quiz, and users can add the question to the quiz.

# PROJECT CONSTRAINTS AND LIMITATIONS

## Project Management Constraints

- Limited amount of time to design, build and test a full stack database driven website. Many additional features are nice-to-have given the time constraints, and focused on producing an MVP (minimum viable product) in the time available.
- Creating the project alone from start-to-finish - I am used to working in a team or pair. In the project planning phase, it is useful to bounce ideas and challenge assumptions/opinions regarding the business requirements, designs and features. Also in the code construction and testing, typically in industry, I would have my code peer-reviewed and subject to a pull request process.
- No client to validate and discuss my assumptions with. Typically I would go through the business requirements and validate/challenge the client's brief in order to confirm my understanding of the project. As this is not possible in this project, there is a constraint that the user requirements are based on my interpretation of the brief and subject to assumptions.

## Limitations / Technology Stack

- Webbiskool Ltd Customers must have the required technology stack in order to run the project code source files and use a local server to run the website.
- Using OOP - there are other software development paradigms available (such as Functional Programming). But I have the most experience with OOP.
- This version of the Quiz Manager will use CSS library, Bootstrap and JavaScript library, jQuery. Due to time constraints, the use of Bootstrap will accelerate front end development of the UI. Additionally the style will likely be overwritten with my client's style libraries. As future versions of the UI and CSS will be customised further, having bespoke personalised stylesheets will not be a critical functional requirement of Quiz Manager.
- For this project, I did not use a Continuous Integration or Continuous Deployment (CI/CD) tool. Typically CI/CD tools are employed to add automated additional validation checks on the. As my project lacked this, there could be some errors that were not picked up in manual tasks (e.g. running test suite and a linter before every commit, etc.)

- The encouraged best practice at work would be to execute the automated test suite on every Git commit, via a CI/CD tool, such as ‘Jenkins’ or ‘TravisCI’. As this program had to be available offline, a CI/CD tool was not used for this project.

# SOFTWARE ENGINEERING PROJECT DECISIONS

High-level technical & coding choices with justification

## Type of Website Needed

- Based of the User/Business Requirements Analysis, a website needs to be built which can manage quizzes. The website must be a database-driven website, with a graphical user interface.

## Engineering architectural patterns and paradigms

- I decided to proceed with OOP as a programming paradigm as its concepts value clean and easy to maintain code. Additionally this is the programming paradigm I have the most experience with in building applications.
- As the brief explicitly states that this is a database driven website, I will build the website using an MVC pattern (Model View Controller) as an architectural pattern.
- I decided to use MVC as the architectural pattern because it is good for full stack websites that require database and front end user interface interaction. The MVC pattern has also evolved from OOP concepts such as Separation of Concerns.

## Languages and Technology Stack

- I will predominantly work with PHP in this project.
- I will use the popular PHP framework - Laravel. Because this framework is based on the MVC architectural pattern, and I have experience with Laravel from my day to day job.
- I will also use HTML, CSS, JavaScript to predominantly to support user interface and implement front end design. Includes Twitter's popular CSS framework, Bootstrap and jQuery to provide best user experience with the time available over the 5-days/project.
- For data management, I will work with relational database in this project. Quiz Manager will use mySQL driver, as I have the most experience with this driver.
- The application will not be deployed. I am aiming to run at least the MVP Website on a local server.

## Ways of Working

- Limited by time constraints in project, with only so many days to complete a full project software development lifecycle (SDLC).
- I will work in agile way as this delivers value to business in quickest and most efficient way possible.
- Will work in sprints, and break each sprint & slice down into tickets.
- I will estimate tickets to ensure I am measuring progress, and project management for time constraints/deadlines.
- **Minimum Viable Product** - For the Quiz Manager MVP, the first iteration will focus on getting a 'viewing-only' version of the website; as suggested by the project brief.
- Once I have produced the website's MVP, my next iteration will involve adding editing functionality to the website.

# USER STORIES

## Purpose

Immediately after examining the Project Brief and creating the Business Requirements & Assumptions doc, I decided to articulate the User Stories for the Quiz Manager application.

A user story is a general explanation of a product requirement, written from the perspective of the end user in non-technical terms.

As a result, I find writing user stories in the planning phase of a project extremely useful for reference to throughout the project and integrated in the workflow. For me, the User Stories articulate how a product feature or piece of work will provide value to the end user. This will keep Quiz Manager project grounded and focused on the user requirements.

## Structure

My user stories will be integrated into my project workflow as high-level requirements and typically expressed in three parts: “persona + need + purpose”.

## User Stories

1. As a user of any kind

I want to log into the website

So that I am considered an authenticated user and can access Quiz Manager's features

2. As a user of any kind

I want all content on each page to be responsive

So that I can view the content on every page of Quiz Manager on any device

3. As a user of any kind

I want to be able to scroll pages up and down

So that I can see all content on any page

4. As a user of any kind

I want to log out of the website

So that my authenticated session is terminated and I am redirected to a login page

5. As a user of any kind

I want to log into the website

So that I can view a list of the quizzes available

6. As a Restricted user

I want to select a quiz from the quiz list

So that I can view a list of its questions

7. As an View user

I want to select a quiz from the quiz list

So that I can view a list of its questions and associated answers

8. As an Edit user

I want to select a quiz from the quiz list

So that I can view and edit a list of its questions and associated answers

9. As an Edit user

I want to create a quiz

So that it appears in the quiz list and it is saved in the database

10. As an Edit user

I want to delete a quiz

So that the quiz and its associated question and answers are removed from the website and database

11. As an Edit user

I want to create a question and its associated answers

So that they appear in the quiz and are saved in the database.

12. As an Edit user

I want to delete a question

So that the question and its associated answers are removed from the website and database.

13. As an Edit user

I want to edit a question

So that I can change the text of the question on the website and database

14. As an Edit user

I want to edit an answer

So that I can change the text of the answer on the website and database

15. As an Edit user

I want to add an answer to a question

So that it is added to the question on the website and saved in the database

16. As an Edit user

I want to delete an answer to a question

So that it is removed from the question on the website and database