QuizMaster – Web Quiz Application

Final Report

Submitted for the BSc (or MEng) in
Computer Science

April 2017

by

Matthew Burling

Word Count: 7,995

Table of Contents

1 Introduction

The *QuizMaster* application is a network oriented system which its primary aim is to aid and control the core functionality of quiz matches. It is known that quiz hosts and contestants have issues with managing scores and sometimes even revealing the answers which makes it harder for the host to control issues such as unfairness and cheating.

There are two common types of quiz systems. The first and most popular form is the pen and paper method, which requires trust in both the contestant or host to mark down and keep track of scores. The second is known as the "class quiz system"; this is limited to specific types of questions or answering styles and does not allow room for customization from the host. The features are simplistic because they are designed for a classroom environment.

The *QuizMaster* application will provide a stable and free flowing quiz games without any interruptions or cheating which can happen in previous systems such as pen and paper quiz management or be limited by certain question answering styles.

Unlike the pen and paper method previously mentioned above, the quiz master application will provide a stable yet free flowing game session. It will do this by eliminating interruptions like cheating or human caused errors. Questions and answers will also not be hindered by question or answer style making it more customisable.

1.1 Initial Brief

"This project will design and develop a system to manage the conduct of a team quiz, presenting questions and answers to the quizmaster and monitoring team responses and scoring. The system may be developed to include hardware to allow team members to signal their wish to answer, and/or be configured to operate in a network or Web environment. It is particularly suited to a student with interests in hardware, especially Computer Systems Engineering." (Rayner, 2016).

The initial brief states that it is looking for a system which is capable of conduction a team quiz with customizability on question and answering styles and the ability to monitor team responses. The defined system is also looking for a way to potentially include hardware to allow team members to signal their wish to answer. The ability to configure the program to work on a network or web environment for global play is also included.

2 Aim and Objectives

The primary aim of the project is to develop a quiz application which can be used for local and global quiz shows. It will be able to handle the core functionality of quizzes such as adding different question style types to the database. Question styles like, but are not limited to, multiple choice and open answering.

The project will be regarded as successful and complete if the following objectives have been met.

Objective 1 – Question and game storage

The storage system that is required for the *QuizMaster* application needs to be reachable and stable so players can connect and play from anywhere around the world. It will need to have as minimal lag and delay from the server end as possible to keep the games being played on a fair and even ground.

The database will need to have the ability to store the users of the system. This includes information such as public display names, role in the system, player's password and team status. Display names will be public and can be seen by other players. The user role will state if the player is an administrator of the system with full control, authorized host or a general participant. Passwords will allow the player to log on to the system. Teams will be chosen and their status will be assigned once indicated.

Objective 2 – Question decision tree

From the assigned rules and games that are added to the database; the *QuizMaster* application will need to decide on which certain structure and style set to use for what game is selected. The question decision tree will work quickly and accurately to randomize questions from the suitable set on the database for which round type is being played. For example, a multiple-choice question cannot be played if the game is *University Challenge*. However, if the game is *Who Wants to Be a Millionaire* multiple choice will need to be used.

Objective 3 – Web Application UI

The *QuizMaster* front-end user interface will manage the creation process of teams, users, quizzes and quiz structures. This will include the registration and login of a user as well as the creation and edition of quiz rules, structures, teams and quizzes. The application user interface will manage other functionalities such as the buzzer, questions and points.

The host panel will manage the teams, to override points, choice questions from available categories and control the scoreboard interface. They also have the option to choose manual or automated *QuizMaster* game play.

Lastly, the scoreboard front-end interface will be behind the contestants and aimed at the audience in a traditional quiz style setup. The style of board will be simplistic and easy to follow with the current active question, if it is being answered, which teams are playing against each other and the current scores on the display.

Objective 4 – Mobile Compatibility

Although regular computer based game play will be the most common platform that the application will be used on, it will also be able to handle game play on tablets, phones and other mobile devices. The transition from computer to mobile interface will have to be seamless, meaning that there will be little to no differences between both gaming platform applications.

Objective 5 – Project Reporting

One of the objectives of this project is to divide the research into three reports and one demonstration of said project. The three reports are included in the initial, interim and final reports. These are required to pass the module and must be finished before the deadline.

3. Background

3.1 Problem context

The focus of the *QuizMaster* project is to build a universal system which will be able to organize many different quizzes with their defined rules and create a way to easily play different game styles like *Who Wants to Be a Millionaire, University Challenge*, *100 to 1* and *Blockbusters* in an online venue. Giving it the actual ability to be played from anywhere around the world.

Another issue within the project is a technological difficulty with the storage of the questions and answers online to provide accurate timing with latency as well as be able for the system to choose what question is suitable for which quiz type.

The final issue with the project lies in the key factor of this project. Which is to define what the system will view as “a quiz” and how far the project will scale and flex with the wide variety of games like gameshows, pub quizzes, educational quizzes and many other types as stated previously. Uncommon styles of games would be theoretically out of scope for the project.

3.2 Comparison of Technologies

3.2.1 Database storage

3.2.1.1 Relational database

“A relational database is a collection of data items organized as a set of formally-described tables from which data can be accessed or reassembled in many different ways without having to reorganize the database tables.” (TechTarget, 2006).

The first advantage of a relational database is flexibility. This allows for the ability of different data tables to be linked, combined, extracted and manipulated easily and effectively without hassle. The second advantage of relational databases is the ability to execute queries such as insertion, updating, deleting, creating and execute searches using specific statements.

One of the disadvantages of a relational database is the physical storage consumption. When you have a large database with a lot of user data the size of the database itself will increase, which will take more space on the server. Another “major constraint and therefore disadvantage in the use of relational database system is machine performance. If the number of tables between which relationships to be established are large and the tables themselves effect the performance in responding to the SQL queries.” (Logix, n.d.)

3.2.1.2 Hierarchical Databases

“A hierarchical database consists of a collection of records that are connected to each other through links. A record is similar to a record in the network model. Each record is a collection of fields (attributes), each of which contains only one data value. A link is an association between precisely two records. Thus, a link here is similar to a link in the network model.” (Silberschatz, et al., 2010).

An advantage of using this type of database is that it is simple to follow on a physical diagram due to its structure and formatting. The other advantage if that its performance is better than a relational database. Due to accessing each node and only doing the calculations within only the nodes which are required. This alone makes it more efficient than going through every single record.

The first disadvantage and one of the reasons it is not commonly used is the hierarchical model is not as flexible to accommodate dynamic structures of data and, when set, the structures cannot be altered easily. The second major downfall of this structure is that if you delete the parent node, it results in deleting the children that are connected to said node this results in data loss.

3.2.2 Database technology

3.2.2.1 NoSQL Database technology

“NoSQL is a generic term used to refer to any data store that does not follow the traditional RDBMS model – specifically, the data is non-relational and it does not use SQL as the query language. It is used to refer to the databases that attempt to solve the problems of scalability and availability, against that of atomicity or consistency.” (Vaish, 2013).

What this means is that it’s not just a database technology it means that it uses different types of technologies merged into one umbrella like term, for example when creating a database using NoSQL technology it’s not a database it can be either a graph database, a Key-Value store, a Column store or a Document database.

NoSQL technology also allows you to change the data type on the fly so for example you can have numbers in a field and then you decide actually you don’t want numbers you want it as a string, you don’t have to change the type of data type it will just obey your type of data on the fly with having either structured, semi-structured, unstructured or polymorphic data types.

One of the advantages of NoSQL technology is that it allows for flexible data models "NoSQL databases have a very flexible data model, enabling them to work with any type of data; they don't comply with the rigid RDBMS data models." (Edward & Sabharwal, 2015). Another advantage of NoSQL technology is that it allows for easy manageability and administration with automatic repairing and simpler data models as there is no strict data model to follow like relational model.

One of the disadvantages of NoSQL technology is limited query capabilities "Since NoSQL databases are generally developed to meet the scaling requirement of the web-scale applications, they provide limited querying capabilities. A simple querying requirement may involve significant programming expertise." (Edward & Sabharwal, 2015). Another disadvantage of NoSQL technology is that it is a newer technology compared to RDBMS data models as the first RDBMS was released 25 years ago but some NoSQL database technologies have just made it out of pre-production stage and some of them are still in beta stages.

3.2.2.2 RDBMS Database technology

"RDBMS store the data into collection of tables, which might be related by common fields (database table columns). RDBMS also provide relational operators to manipulate the data stored into the database tables. Most RDBMS use SQL as database query language." (Kasbe, 2016).

What this means is that it stores all the database in a table and within the table there is columns to organize your data into as well as allowing using SQL database query language which allows to do searching like selecting data on certain conditions such as value is lower than a certain threshold.

One of the advantages of RDBMS is that integrity of the data can be enforced "Integrity of data means that data in database is always accurate, such that incorrect information cannot be stored in database. In order to maintain the integrity of data, some integrity constraints are enforced on the database. A DBMS should provide capabilities for defining and enforcing the constraints." (Thakur, n.d.). Another advantage of a DBMS is the ability of concurrency control as it helps provide concurrent access of data to multiple users which little to no issues.

One of the disadvantages of DBMS is that there is sometimes a common performance hit when the size of the database increases in size applications connected to the database and retrieving data from it can potentially slow down. Another disadvantage of DBMS is there is a higher impact of a failure "The centralization of resources increases the vulnerability of the system. Since all users and applications rely on the ~availability of the DBMS, the failure of any component can bring operations to a halt." (Thakur, n.d.).

3.2.3 Language technology

3.2.3.1 Scripting Language

"A scripting language is a programming language designed for integrating and communicating with other programming languages." (Techopedia, n.d.). A scripting language is a type of language which isn't directed at creating an executable but it is on the fly interpreted by a browser or in this case a server which then turns the code into information and visual for the user to show in a browser.

One of the advantages of scripting languages is that it allows complex tasks to be performed in small, simple steps as well as allows for the addition of dynamic and interactive activities like games and other complex calculations to be done interactively through a web page.

"The major disadvantage of scripting languages is that executable code can inadvertently be downloaded from a remote server to a web browser's machine, installed and run using the local browser's interpreter." (SQA, 2007).

3.2.3.2 Programming language

"A programming language is a computer language engineered to create a standard form of commands. These commands can be interpreted into a code understood by a machine." (Techopedia, n.d.).

One advantage of a program language is that the application can be optimized for target hardware, that is aimed to deploy after it is compiled. Only then will it be executed. Another advantage is that it is a fast execution when the application is running. That is, until it has done its programmed job or crashes if there is an issue or unfixable error by the compiled system.

However, the disadvantage of programming languages is that it requires a compiler. Unlike a scripting language, compiling in a programming language application can be slow and tedious if it fails due to lacking libraries or dependencies. Editing and deploying code is a much slower process than interpreters as it requires the code to be optimized and error checking, which is invalid if done in this format.

3.3 Quizzes

"A quiz is a form of game or mind sport in which the players (as individuals or in teams) attempt to answer questions correctly. In some countries, a quiz is also a brief assessment used in education and similar fields to measure growth in knowledge, abilities, and/or skills." (Quiz, n.d.).

Quizzes are mainly scored on points and, in the most common formats, the team or contestants who have the most points are the winner. However, in some instances, this isn't always the accurate. Some games require the winning team to go into a final bonus round which will result in prizes being won.

3.3.1 Quiz formats and structures

Due to the fact that quizzes are extremely diverse and that there are many different types of quizzes that exist, that means there are also many different formations on how a certain type of quizzes can be played. For example, a generic pub pop quiz is normally six to eight rounds and contains six to ten questions per round. However, another format is that "The Question Reader asks each team in turn a three-part question. One person must give the answers. The team has 30 seconds to commence their answers" (League, 2015). This style format uses thirty-two questions instead of six or ten, is split into two individual types of rounds and includes a team quiz round.

Another example would be the comparison of *University Challenge* to *Family Fortune*. In *University Challenge* "Questions begin in the form of starters for 10 points, and must be answered on the buzzer with no conferring. The team that answers a starter correctly then gets three linked bonus questions for 5 points each during which the team may confer, with the captain (who sits third from the left) giving their final answer." (Anon., n.d.). It is composed of two teams of four which represent their university in a tournament style competition. On the other hand, the family Fortune gameshow format has two different families, four members of each family in a team, that compete against each other. Both families need to guess the right result of surveys where certain people are asked specific questions relating to certain situations in the format of "100 people were asked the most popular item taken on holiday". The contestants are then waited with certain points which would over all equal to 100.

During the research of quiz formats and structures each quiz can be shown by the following stages. Is the quiz team based, how many rounds does the quiz take, how many people are in a team if it is team based, what type of questions are asked, is it a tournament based quiz or is they a winner at the end of the show?

3.3.2 Question styles

Knowing that quizzes are diverse and there are many different types of quizzes, the only way to narrow this down is by looking at what makes them unique. One of the ways to see how some quizzes are unique is by the questions that contestants get asked when behind the podiums. When comparing and analyzing both American and British gameshows overall, ranging from young children to adult quiz shows, there is a plethora of different question styles. From reaching twenty-three gameshows overall, below is a bar chart of the most common type of question styles there is in gameshows in both countries.

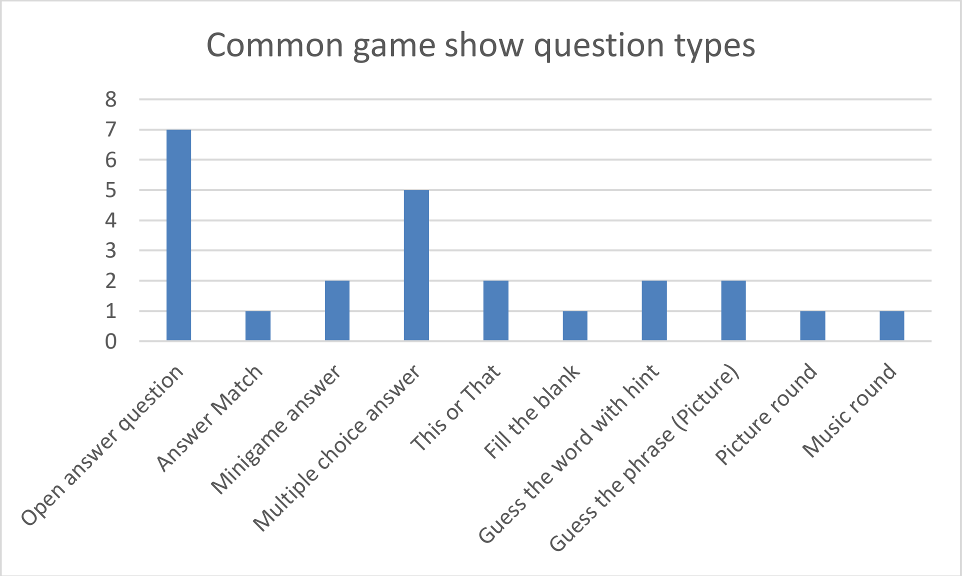
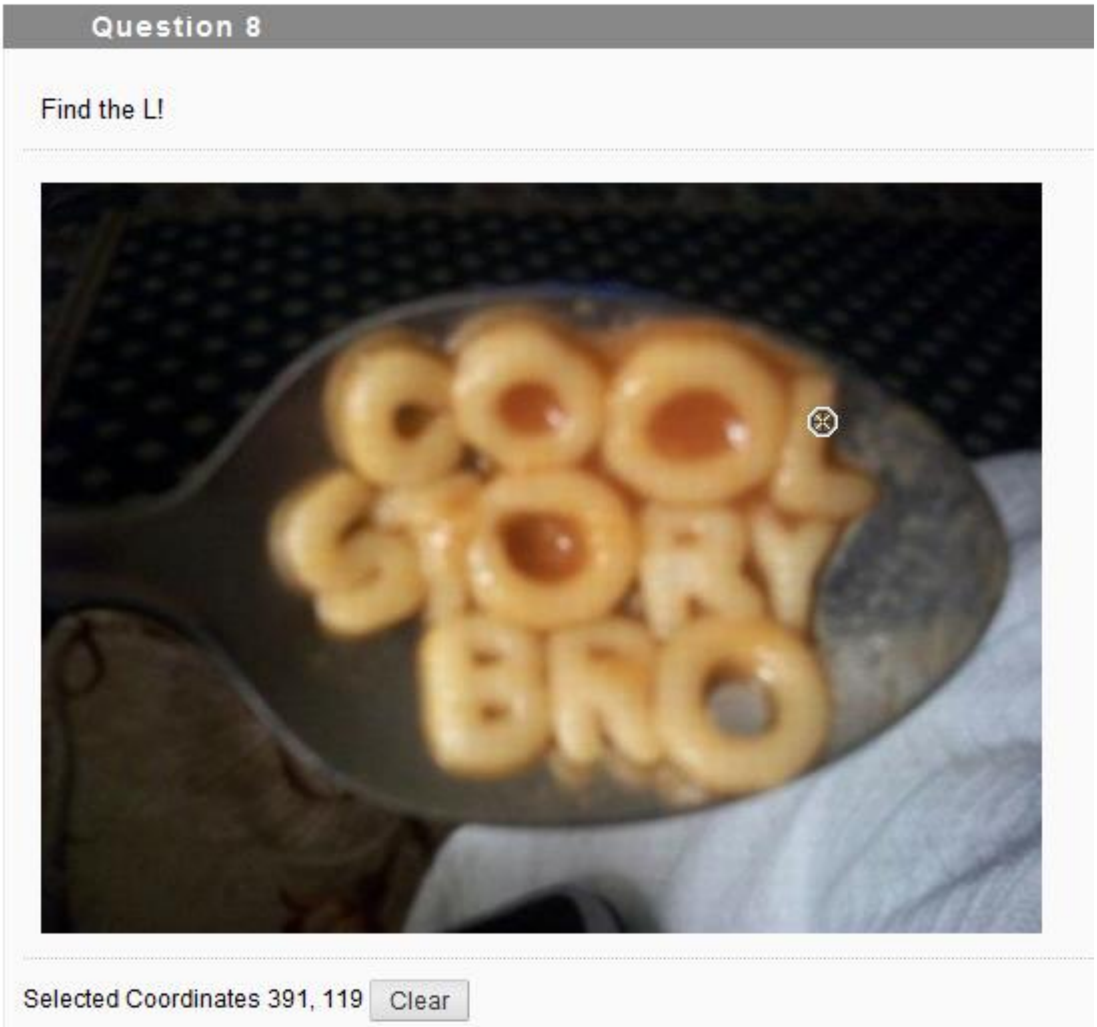


Figure 1. Graph showing common game show question types from analysis of gameshows

The table above shows from when analyzing game shows from watching them it shows that open answer questions and multiple choice questions was the most popular type of questions with game shows such as *Are You Smarter than a 5th Grader* or the UK equivalent *Are You Smarter than a 10-year-old, Family feud, Wheel of Fortune, Cash cab, University Challenge* or US equivalent *College Bowl* and the *Weakest Link* all follow an open answer question answering and asked style. For example, questions like “What is the capital of England?” is an example of open answer question. The next most popular question types are multiple choice questions/answers. Gameshows such as *Who Wants to Be a Millionaire, 1 vs 100, The Chase, Duel* and finally *Million Pound Drop* all included a multiple-choice question answering style. For example, questions such as “Which of the following is not a country, A) La la land, B) England, C) Japan or D) China?” is a multiple-choice question.

After watching many gameshow types; my research has concluded that the question style is not important compared to the structure of the game. Referring back to Section 3.3.1 of Quiz formats and structures it shows that *University Challenge* and *Wheel of Fortune* are majorly different but they follow the same type of game show questions styles. However, I also found that the two most common style types are multiple choice and open question. Which made it high priority for the *QuizMaster* application to have.

Further research of question styles, now knowing the most common gameshow types in the United States and England, shows which question types are potentially not seen in gameshows. Although, there are many ways of answering a simple question. For example, written by the Worcester Polytechnic Institute they introduce students to click on what is defined as a “Hot spot” they state that “students will have to click on the defined correct area of the image to get a correct answer.” (Institute, n.d.). The image in reference will be shown in Figure 2 below.



Question 8

Find the L!

Selected Coordinates 391, 119

Figure 2. Hot spot image from Worcester Polytechnic Institute (Institute, n.d.).

This question type is usually limited to schools and college institutes who are answering questions for their exams for example. But, these types of questions can also be implemented within an online quiz style of Where's Wally bonus round to throw off the competition, similar to the game show *Beat the Brain*. The minigame answer statistic is not commonly used within gameshows, but works better in an online quiz application.

In conclusion to this section, the most common types of question in gameshows are multiple choice and open answer question types. The main focus of the QuizMaster application will be to implement these types of questions without issues, then work slowly towards other types of questions which will be referenced with "minigame" type questions like *Hot Spot*, from Worcester Polytechnic Institute, *Beat the brain* and *Price is Right*.

3.4 Alternative Solutions

3.4.1 Kahoot

The first alternative solution is a quiz application called Kahoot, "Kahoot! is a free game-based learning platform that makes it fun to learn – any subject, in any language, on any device, for all ages!". Kahoot allows teachers to create a collection of questions on specific topics which can then be played in real-time to an unlimited number of players. Kahoot offers the one game type which is multiple choice answers between 2-4 answers and only one correct answer with an included time limit which can be set between 5 seconds to 2 minutes.

Kahoot's solution allows for any device to use their site from a desktop PC to a tablet/mobile device which solves the issue of accessibility and help bring the idea of Objective 4 in my aims and objectives that the project which I'm doing needs to be accessible to all devices and tailored towards all types of devices.

However, Kahoot doesn't do anything but the rules said above so it can't be split in-between teams and one team buzz in and then answer the question it only allows for simultaneous play.

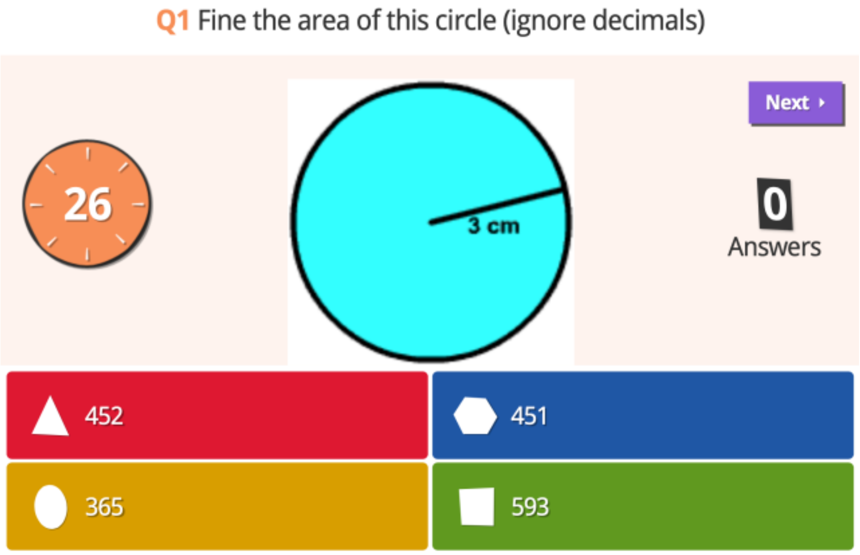

Q1 Fine the area of this circle (ignore decimals)

Figure 2. Screenshot of Kahoot

3.4.2 Quizizz

The second alternative solution is a classroom quiz application called Quizizz. "Quizizz allows you to conduct student-paced formative assessments in a fun and engaging way for students of all ages." (Quizizz, 2017). Quizizz allows teachers or students host their own quizzes to share and play with their peers, as well as be able to play with their own devices which included tablets, laptops and smartphones.

Quizizz's solutions allow to take any question from any public quiz so instead of writing your own questions as well as easily add images onto quizzes for picture type rounds. As well as some customization options "Teachers have multiple options to customize their quiz session to toggle the level of competition, speed and other factors." (Quizizz, 2017).

However, Quizizz doesn't clearly allow the ability for manual control with the questions and switching between multiple choice and text answers similar issue which Kahoot doesn't allow.

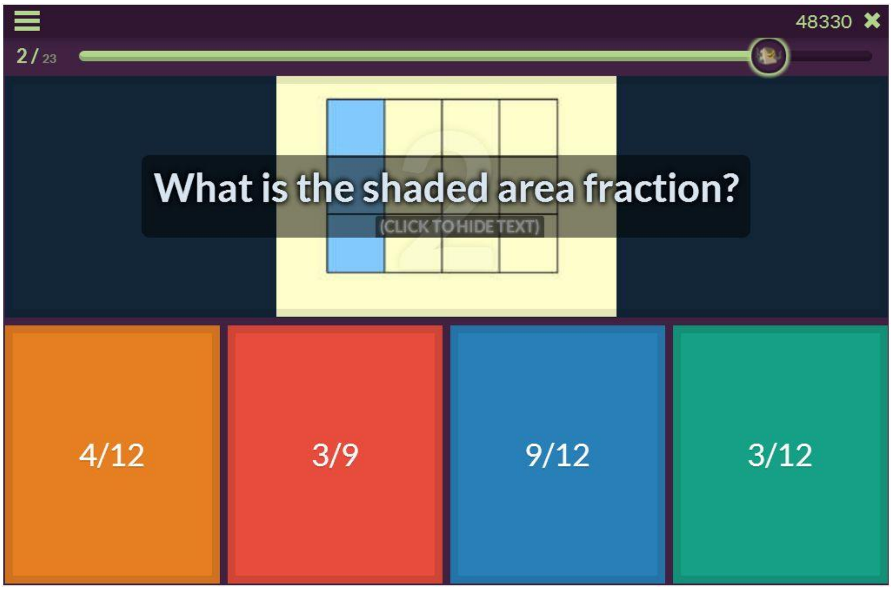


Figure 3. Screenshot of Quizizz

3.4.3 Desktop application

One of the ideas of a solution for this issue is to create a Desktop application which would be able to read the questions from a database online but require the “always online” model with no ability to have offline support, due to how the Quizmaster design was going with as the questions would be given from a database as planned.

The UI would show the quizzes waiting to be played and the user clicks on the lobby and press join to be a part of the lobby for the game, however this was not decided as one of the options to take due to their no easy portability which the quizmaster application required to be able to play anywhere on any device which supports a browser and have an internet connection.

4. Designs and Technical Development

The quizmaster application which is being developed is a system that enables a quiz host to either locally or globally host a quiz which allows teams and players alike compete in a quiz style setting with rounds and points with a scoreboard for spectators and a host panel for the host to control the quiz as well as finally having a way to add new questions into the panel all on the web. In order to complete this goal, the system needed to be drawn out and planned to ensure all requirements for the project was gathered and to minimize the large scope which the initial brief referenced in section 1.1 of the report can be planned and achieved.

4.1 System design and Implementation

4.1.1 Entity Relationship Diagram

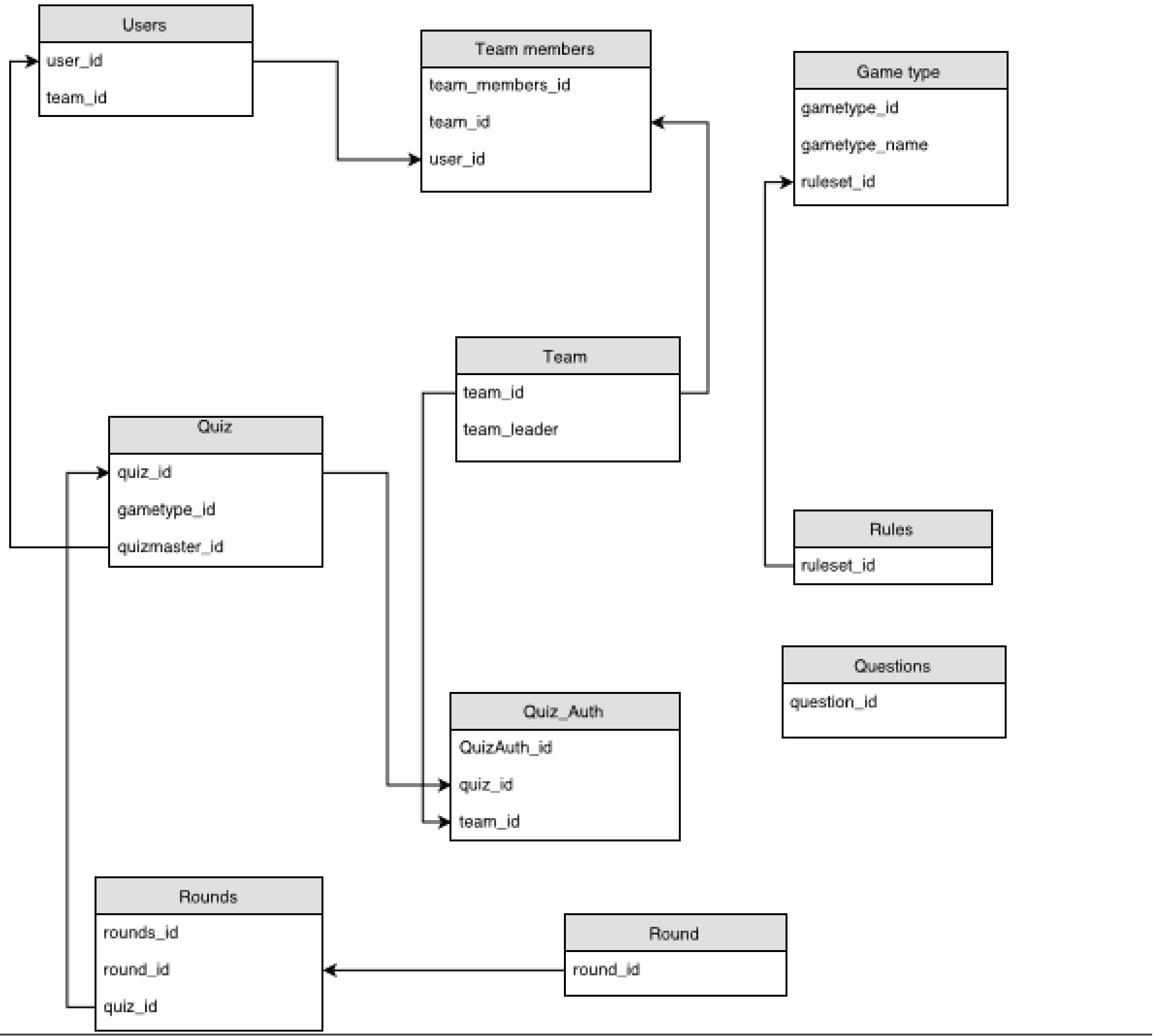


Figure 4. Simplified Entity Relationship Diagram for Quizmaster DB.

This diagram below shows a simplified diagram of Appendix B's Entity relationship diagram. This diagram is the database finalized plan which will run the Quizmaster system and the links between all the tables which would be in the relational database system, this database plan is also can be tailored for a non-relational database format which is what the new plan will be now following which would be discussed later on in this document.

The user table will be where all the user information data will be stored such as users display name what team they are part of and if they are a quizmaster or not. The team members table will be a list of all the team members which are related to a certain team. The game type table will be the table which defines the title of a game if it's University challenge or Who wants to be a millionaire etc. The rules table will define what all the rules are for the game which will be played. The question table will define all the questions including the correct answer and all the potential answer and what category of question they are. The quiz table will define the active quiz game and who is the quizmaster. The round/rounds table will define what is in a round and how many rounds we have and finally the "Quiz auth" table will define who is authorized to join the quiz.

4.1.2 Use Case Diagram

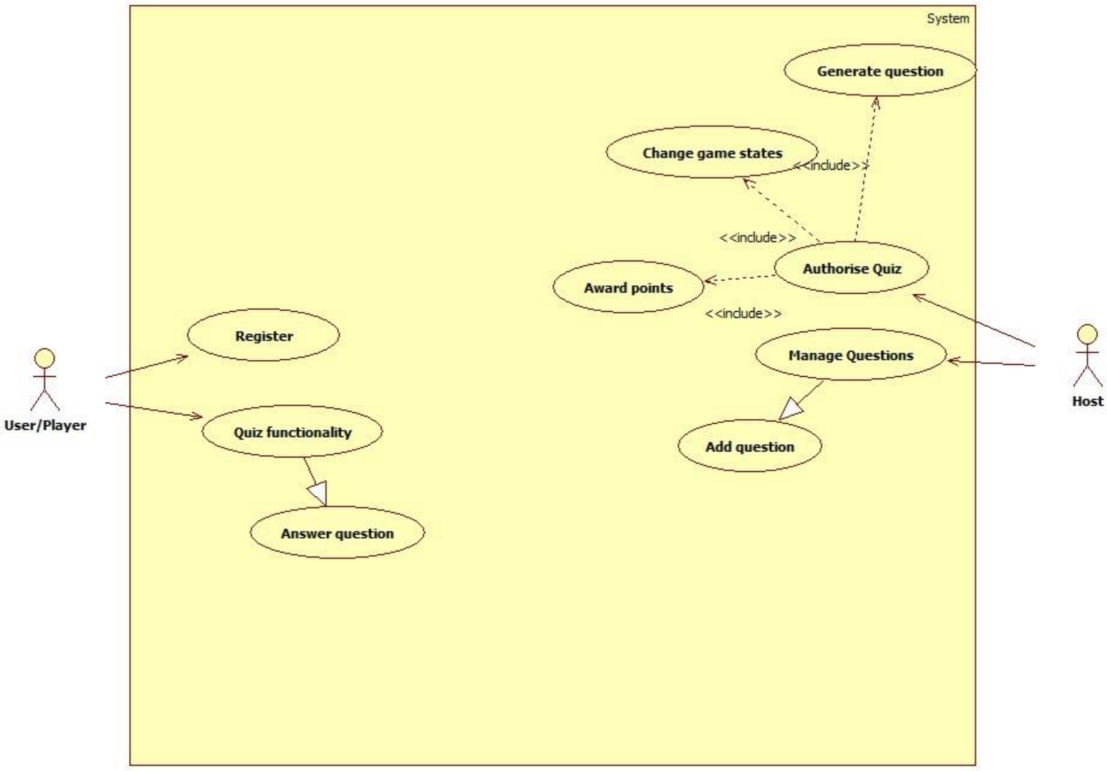


Figure 5. Refined Use Case Diagram which reflects better on how the quizmaster application looks

This sections covers a refined use case diagram which reflectst better on how the quizmaster application looks in it’s current state.

The register use case from the user actor allows the user put a team name or their own name at the start of the quiz to register themselves/their team for the active game, waiting to start.

The authorise quiz use case from the host actor in the diagram above allows the host to control all the aspects of authorising the quiz from generating the questions for the contesents/players to be able to answer the questions. The host also change the game states from either question/answer stage to intermission stage.

The manage question use case from the host actor allows the host to have the ability to add questions to the question bank. Some of the types of fields which the host is allowed to add is what category/topic is the question for example “Geography” or “50s Music”, the ability to add the question and answer, how many points are available for the question and what type of question is it either multiple choice, text based/ open answer etc.

The quiz functionality use caser from the user/player actor allows the user to be able to handle all the functionality for example one of the features of quiz functionality is being able to answer the question either via buzzing your answer in on a text based answer round or being able to press a button to buzz your answer in multiple choice round.

The refined user case diagram looks a lot smaller than the initial proposed use case diagram as shown in Appendix E this is due to the case of time constraints which caused some of the initial proposed features not being able to be completed.

4.1.3 Database Development

A database has been developed to store the question bank which will be used for the quizmaster quiz games. For the reasons in the Database Technology (Section 3.2.2) a NoSQL database has been chosen for the technology of choice within the system. Unfortunately, due to scope creep not all elements as disclosed in Appendix B ERD Database Diagram was implemented but was researched and this section will discuss all the sections which would be included in a full version of the quizmaster application.

4.1.3.1 Question Table

The question table was first created to store all the information which would be stored within a question so for example the Question category would be stored to determine for future expansion of game structures with first "choose a category" then these types of questions show up within the correct category rather than showing up at random.

The next field to be stored is the quiz question which is the question shown to the contestants of the quiz. The answer aspect which is the next section is to show the host if the answer is correct or not and they manually approve the answer of a player. However, the planned functionality of this field was to be able to automate the correct answer field to remove the chance of error from the host in question.

The next field is the score weight field which would allow for the diversity of the point system for the contestants and not all be neutral answering of only 1 point questions as in most quizzes each question has a different point depending on the round, but unfortunately for now due to the game having no game structure database the points are tied to the question.

The final field is what type of question will it be which is an important field as it determines what type of question style the question will need to be answered so for example, multiple choice questions or open answer type questions.

4.1.3.2 Users Table

The users table was designed to be able to store their name, user rank of either if they are allowed to be a host, what team they are part of if they are part of a team which could be loaded when they log into the proposed User hub which would have been an extended out of scope.

4.1.3.3 Round Table

The round table is a crucial part of the quiz structure as it would be able to handle how each round is played when linking towards each type of quiz as each quiz has a different set of rounds.

4.1.3.4 Quiz Table

The quiz table is also just as crucial as the round table as it will hold what type of quiz you are playing by name, how it is laid out and following which type of game structure and how many rounds are available for the quiz, max points and other elements which make up a quiz show.

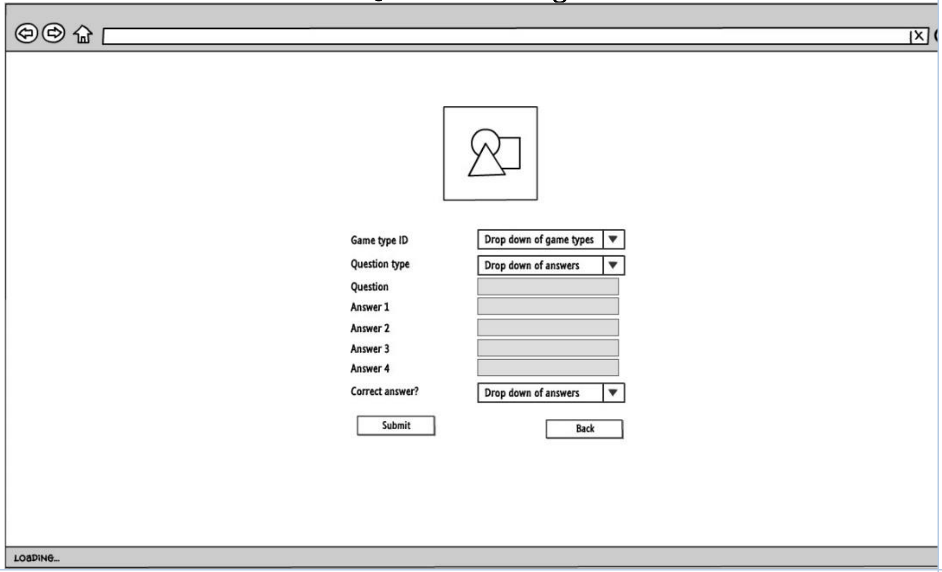
4.2 UI Design

The UI design choices which I have chosen from working out are optimal for the "three-click rule" in some places. The three click rule is an unofficial web design rule concerning design where if you can't get to the information or the page you want within three clicks then it's poor design. The reason why I disagree against this rule and I have broken this rule in my design is because I feel the design should be intuitive and if it takes you more clicks to get to the information but the UI is simplistic and easy to follow and easy on the eye for the user then the more clicks are worth it.

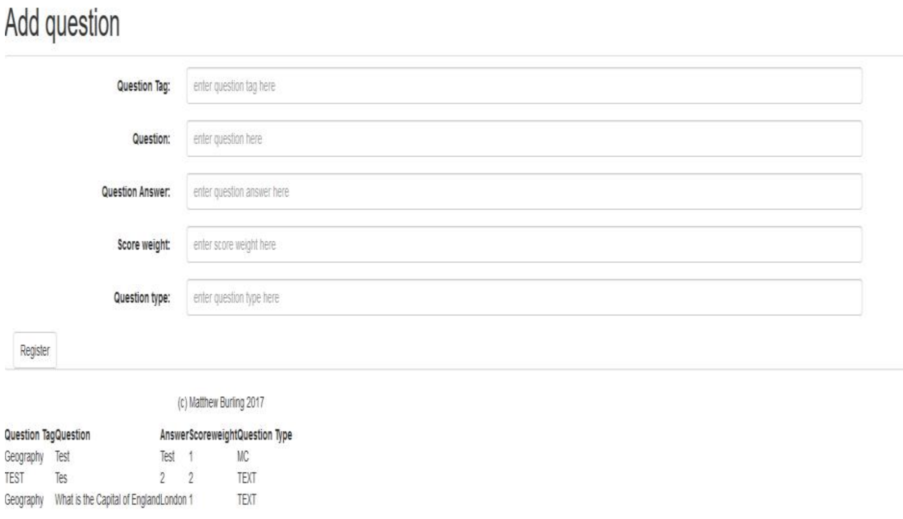
The table below will show the user interface sketch/wireframe design which was created previously in the design stage and it will also show the implementation where relevant

<table>
<tr><th>UI Design Sketch/Wireframe Design</th><th>Implementation</th></tr>
<tr><td>

Question adding



</td><td>



Add question

Question Tag: enter question tag here

Question: enter question here

Question Answer: enter question answer here

Score weight: enter score weight here

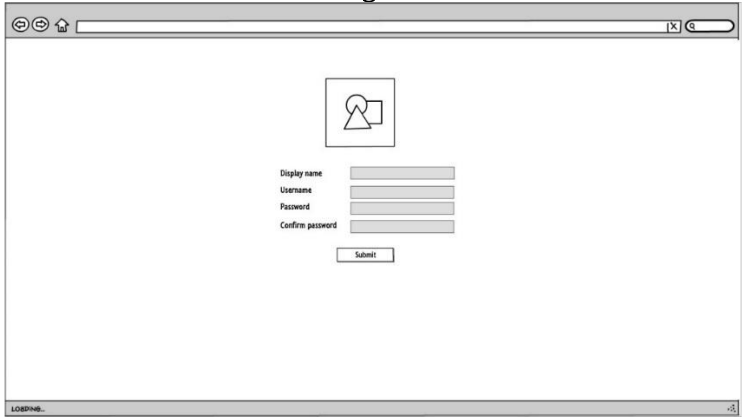
Question type: enter question type here

(c) Matthew Burling 2017

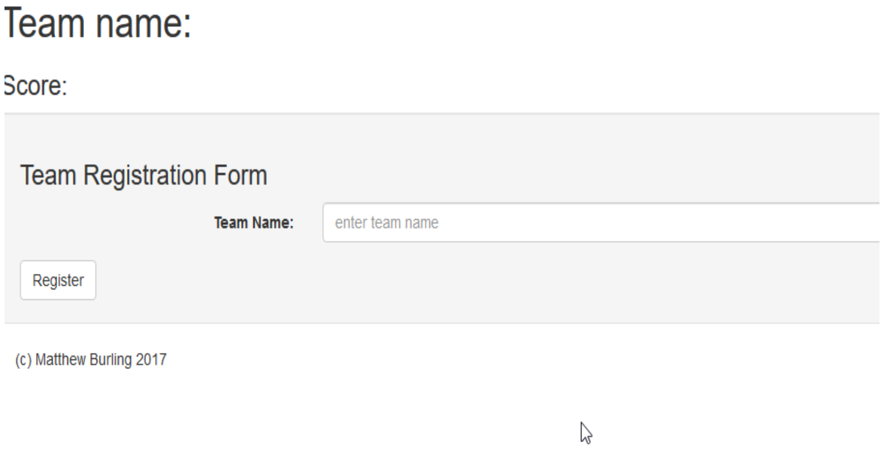
<table>
<tr><th>Question Tag</th><th>Question</th><th>Answer</th><th>Scoreweight</th><th>Question Type</th></tr>
<tr><td>Geography</td><td>Test</td><td>Test</td><td>1</td><td>MC</td></tr>
<tr><td>TEST</td><td>Tes</td><td>2</td><td>2</td><td>TEXT</td></tr>
<tr><td>Geography</td><td>What is the Capital of England</td><td>London</td><td>1</td><td>TEXT</td></tr>
</table>

</td></tr>
<tr><td>

User registration



</td><td>



Team name:

Score:

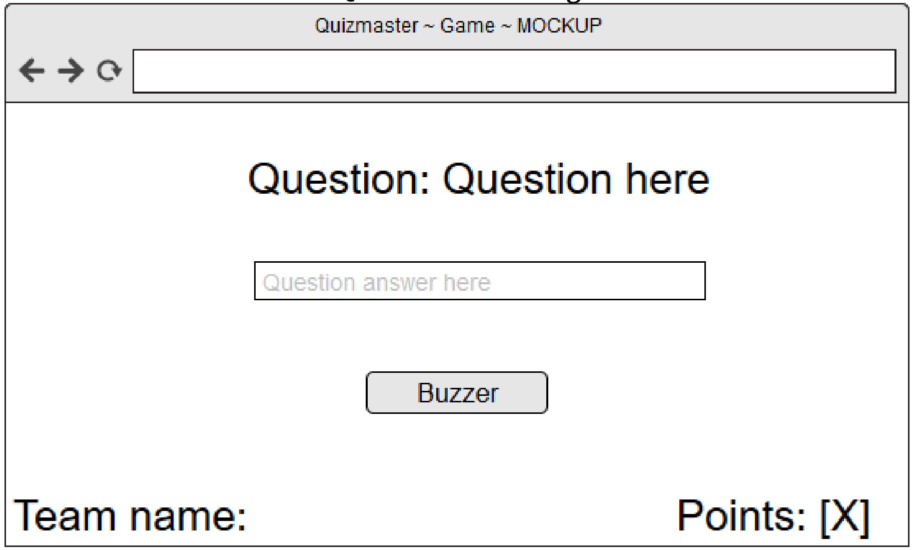
Team Registration Form

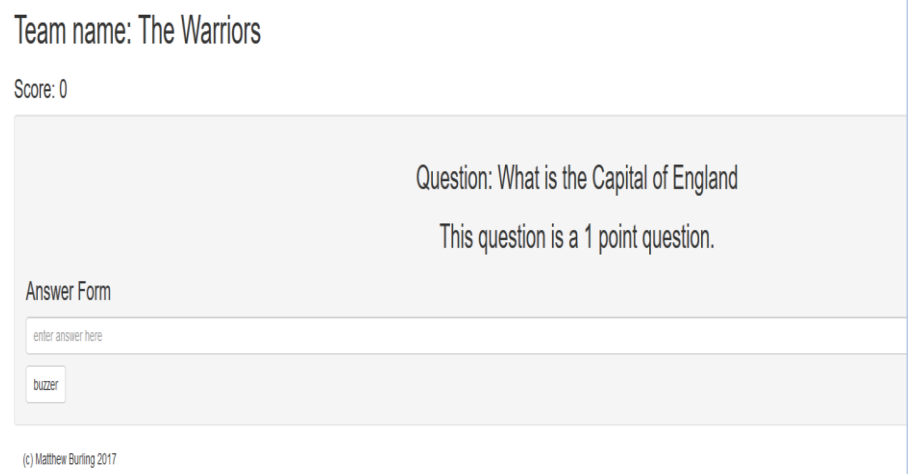
Team Name: enter team name

(c) Matthew Burling 2017

</td></tr>
</table>

Question answering





Team name: The Warriors

Score: 0

Question: What is the Capital of England

This question is a 1 point question.

Answer Form

enter answer here

buzzer

(c) Matthew Burling 2017

Table 1 – UI designs of the different sketches with implementation counter part

The screenshots above show some of the UI mockups compared to their implementation counter parts. The implementation counter parts aren't exactly as the UI mockups nor are they aesthetically pleasing to the user this is due to time constraints and seeing that the UI wasn't the most important part of the project it was the functionality as UI can change at any time but the functionality was the main focus throughout the entire project.

4.3 Question Storage

The quizmaster application stores all the questions which can/will be used for all the quizzes on the system. The types of questions it can currently store is text based answering questions like university challenge has and multiple choice questions like Who Wants to be A Millionaire has with either A, B, C or D answers. This allows the quizmaster application to be able to play games like university challenge as well as have who wants to be a millionaire type questions as well with multiple choice answering.

The quizmaster question storage does have some features which aren't fully implemented but stored for later use, for example the ability to add a question category either "Geography" or "80s music" for example. This allows the host when implemented the ability to give the contestant a choice of category question, similar to how Million Pound Drop or 1 vs 100 plays where the host gives the contestant either 2 categories to choose from then a question from the category is generated.

Another part is the score weight storage which has some implementation but not based on much logic. This storage was for having the ability to make 10-point question,50-point questions and showing the point system majorly however if this database was to be expanded this element would be tied to the round which we are on and not the question.

5. Evaluation

The quizmaster application project has been a bumpy and only semi successful when comparing it to the original objectives which stated in the report above which where to allow a database storage for questions and game storage, have a question decision tree, to create a web based application user interface, have mobile compatibility and finally report the project within three reports.

5.1 Project Achievements

Objective five was also successful as both the initial report and interim report have been handed in as of writing this report and the final report will be handed in before deadline day to confirm that this objective is fully concluded as successful.

5.1.1 Project Achievements related to Question and Game Storage

The project achievements related to question and game storage was partially successful with only partial support of this objective as only the question storage element was completed which allowed storage of either a text based answer or multiple choice questions to be created and stored for the contestants to play. The only reason this objective was semi successful was due to time constraints and due to change of design midway through the project and the vision for the project was altered due to the vast scope of the project caused issues to be able to narrow it down to be something seen as successful and maintainable.

The question and game storage also when creating this objective wasn't known to me of how many question types was within quizzes and how long each element would take to implement with a smooth transition between each question type so early on in the project it was decided to delay the game storage which would control the round structure and type of games such as "University challenge", “Blockbusters” and others and focus on getting the question structure of each type of question so then this feature could have been added at a later date.

5.1.2 Project Achievements related to Question decision tree

The project achievements related to the question decision tree unfortunately wasn’t as developed as expected for the project as due to uncontrollable issues with technology as well as scope creep played a major part in the unexpected poor achievements in this subject, however with that fact in mind they only question decision within the project is to decide whether to give out a multiple choice or text based answer and the user interface for the contestants/players will change based on which question they are asked to answer.

The plans for this objective and the aims for achieving was to be able to add a way to save a type of game for example Blockbusters and allow the host panel and contestant game interface to tweak based on the rules so for example in blockbusters to show a blockbusters board and allow the contestants play a game of blockbusters when the host selecting the game type and the questions following the blockbusters structure.

5.1.3 Project Achievements related to Web Application UI

The project achievements related to the web application user interface succeeded to a satisfactory/good status as the web application allow the addition of questions within the question bank, it allows the game functionality with swapping between different game interfaces with room to expand for more question types. This also included the scoreboard UI which is very simplistic looking which shows the “hot seat” player which is the player the host has highlighted out for various reasons, the question and the points which each contestant’s points shown to the audience. Behind the scenes as well there is a manual host controlling interface which generates the question and pops up no questions left in the question bank if all questions have been played. Finally, it includes a keyboard bound buzzer as well as a UI interface buzzer to confirm your answer to the host.

Some of the things which could be changed as will be visible in Appendix D is that the UI looks still in alpha stage from the visual aspect which needs to be improved hence the reason why it’s achieved under a satisfactory/good state as it’s not perfect or as what is needed to be fully seen as a successful pass of objective.

5.1.4 Project Achievements related to Mobile compatibility

The project achievements related to the mobile compatibility was a success due to the fact of how the web application user interface was programmed it was created from the ground up to also support mobile devices such as tablets and mobile phones as this was a crucial element of the project, as the system was aimed to hit as many devices as possible as long as they are using a sufficient browser which supports the JavaScript scripting language.

But when running tests from a couple of devices it showed that all elements of the application from entering questions into the question bank to playing the quiz worked flawlessly minor some issues with screen size limitations and how they impact on the viewer and user.

5.2 Further Work

The other thing which would benefit the project is the allowance of a user hub which was in the works originally and wasn’t in the initial plan and was demo in extreme simplicity but then put on hold due to focusing on the core quiz application structure as well as graphical upgrades of the entire project to mention a few core improvements to advance the project closer to release in the future.

Within the quizmaster application there is two known bugs with the application. First bug is that sometimes the question will not load and sometimes it will first time which is a strange bug and not been diagnosed at the time as well as the buzzer will be active throughout the entire program even if the quiz hasn't "officially" started.

5.2.1 Further work related to Question and Game storage

The further work which could be seen beneficial to the project regarding the question and game structure which would be to allow different type of questions such as video questions from YouTube to guess the certain song in multiple choice format, picture questions similar to a where's wally where you click a position on the picture and then answer it similar to the example discussed in Section 3.3.2 with finding the letter within the soup.

The other further work related to game storage would be allowing the ability to store different types of quiz games like round structures with set rounds for multiple choice set rounds for picture and so on and so forth with a max round count, max point structure and team allowance.

5.2.2 Further work related to Question decision tree

The further work which could be seen as beneficial to the project regarding the question decision tree and would require miraculous improvement which would be following the question decision tree based on an actual game structure as currently it only sees if the question is being played or not, if it hasn't then choose a random question from the pool which is insufficient and doesn't rely based on any game structure on whether it's a picture round, multiple choice round or anything like that it just acts like any round can have any question which in real quizzes this isn't the case and this would allow the quizmaster application be on the steps of simulating games like *Blockbusters, Family Fortunes and Who Wants To Be a Millionaire.*

5.2.3 Further work related to Web application UI

The further work which could be seen as beneficial to the project regarding the web application UI would be to give it more design as currently it looks really flat and not visually pleasing to the user, as it is just in black and white with no vibrant color so designing the UI and cleaning up the elements such as buttons would improve the user interface design of the quiz master application as well as add different elements such as adding quiz games, editing questions and deleting questions which would miraculously improve the usability of the application without feeling features are lackluster which it currently seems like as of writing this report.

5.2.4 Further work related to out of scope objectives

The further work which could be seen as beneficial to the project regarding out of scope objectives, some of these objectives is the objective called User Hub.

The user hub was supposed to be a way for teams and players to register and login to a container of quizzes, teams and player's a like to find active quiz games or quiz games which are scheduled, teams looking for players and players looking for teams. This would have been the place where teams would meet before the quizmaster game application which was succeeded in certain aspects would be as the game application would be a locked down to only the teams which are registered. The other aspect of the user hub would be a ranking system so then players and teams could find who the best team was in a certain game show so for example Team Dinosaur was number one at Blockbusters on the quizmaster service in "ranked games".

The other out of scope objective was an automated quizmaster so then teams would be able to practice against other teams in unofficial games before full quiz shows or if two teams needed to play with no quiz host available it would be possible to play any games at any time with the automated host answering the questions and follow the quiz structure automatically with a report logging for if the automated system fails.

5.3 Discontinued and refined discussion

This section discusses about the discontinued and refined items within the Task list which can be seen in Appendix A.

No	What was refined/discontinued	Why was it refined/discontinued
1	Front end system/Creating front end UI	The reason why it was discontinued is due to the fact this element was a part of “scope creep” where there was seen as a non-important objective until it was seen as it could be beneficial to the project but in the end, it caused delay in so many aspects to the project. The other element would accompany
2	Registration system / Creating the registration UI	The reason why this was altered is due to time constraints where this would be a part of the user hub or as known in the table as the front-end system.
3	Create the DB Schema	The reason why this was refined was due to the fact that the technology of the database technology from PostgreSQL to NoSQL (MongoDB)

Table 2. The table which discusses what was refined/discontinued and the reasons why.

5.4 Personal Reflection

Some of the features were implemented for the project so it could be argued that it was a partial success but not as much as would be liked to be completed and didn’t reach my personal goal with the project, however if some of the objectives which wasn’t focused on mainly was focused on more and was completed the project in my opinion would be in a happier state.

This has been the second type of project of this type of size has been undertaken and honestly with how this project has gone with the planning and the tasks which needed to be completed was of right caliber but with personal issues causing a conflict within the project it made the project harder than it should have been physically and mentally this time around than the previous time.

During this project, overall lessons was made in time planning and realizing when to actually alter the time plan as last report it was stated that no need for a time plan change was required however if the time plan change may have been included there could have been potentially more success in the project.

6. Conclusion

Overall the project to create a web quiz application which would be able to handle a quiz type game was a partial success as it was able to do so but not with enough rich features to make the project stand out from others.

Most of the objectives have been tackled and have been at least has the scrape of the subject with only two seen as fully finished and achieved, with the others having being dabbled into and have some sort of progress even though it may not seem as a lot of progress.

A user can in the projects current state play rounds of a quiz game with a manual host generating the questions for each contestant, the host also can add more questions into the question bank with two type of options either text based answering or multiple choice answering types.

Some of the reason of not being able to implement all of the features which would have made the project unique is time scale and personal issues handling mid-way to three quarters of the project which crippled development unfortunately but overall lessons were learned of how to manage my health as well as time when it comes to long projects which didn't happen in my previous project.

If a project was to be taken again under similar conditions, a lot more planning would be involved and not underestimating the project timescale and if there is a feeling to change the timescale then changing the timescale may be needed to be done and not try and "push forward anyway" which was the mentality from the interim report.

Appendix A: Initial Task List

#	Task Name	Description	Duration (days)
1	**Preliminary work**	To gather the needed information before the project can be started	10
1.1	Requirements gathering	To gather the requirements needed for the system and do the research for the project needed	4
1.2	Initial system design	To cover the initial system design by creating use case diagrams and activity diagrams to organize the operations of the system	2
1.2.1	Use case diagrams	To draw the use case diagrams which will be needed for the entire system	2
1.2.2	Activity diagrams	To draw the activity diagram which will be needed to show the workings of the system	2
1.2.3	Preliminary work completed	Milestone for preliminary work	0
2	**Database structure**	To plan the full database structure using a database design diagram to show where the primary keys are in the project and the database holding structure	7
2.1	Gather entities and their relations	To gather the entities of the database for all the tables as well as the primary keys and foreign keys	2
2.2	Design database ERD	To design and draw of the database ERD diagram to visualize the database relations as well as the entities within	2
2.3	Create DB schema	To create the database schema and make sure there is no PostgreSQL errors when creating the SQL query	2
2.3 (refined)	Create database document	To create the database document via MongoDB and make sure there is no errors with the storage of the data.	2
2.4	Testing database	To test the database required for the system with dummy data provided if the data is stored correctly then the test has succeed otherwise if this test fails revert to the planning stage.	2
2.5	Database test passed	Milestone for the database structure	0
3	UI Design	To plan the UI design of the application including the scoreboard mechanism for the spectators, the user interface for the players as well as the host.	20
3.1	Design all UI needed	To design all the UI which will be needed for the system this will include: Scoreboard UI, Player UI,. Host UI and Registration UI	4
3.2	Research alternative technologies	To research the alternative technologies and method which could be used for the project as well as list there disadvantages and advantages	4
3.3	Create the player UI	To create the player UI that will tailor towards the official planned UI design or close as possible to it	3
3.4	Creating the host UI	To create the host UI that will tailor towards the official planned UI design or close as possible to it	3

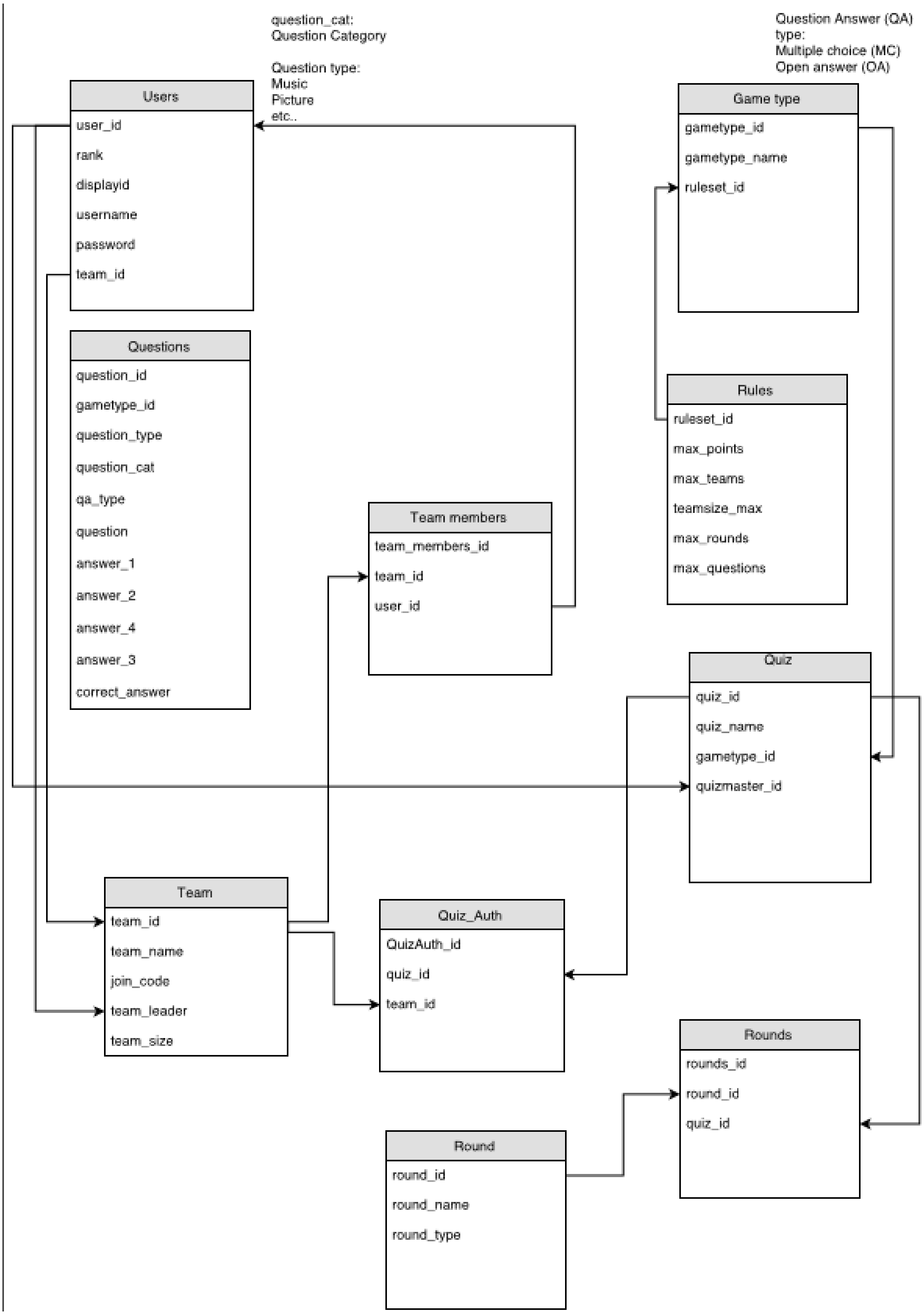
		This will include the settings UI and some basic functionality of the question selection system	
3.5	Creating scoreboard UI	To create the scoreboard UI that will tailor towards the official planned design or close as possible to it	3
3.6	Creating the registration UI	To create the registration UI needed for the players to register for the quiz	3
3.6 (refinedv2)	Creating the front end UI	To create the front end ui which will cover the registration and login of users as well as cover other management of editing the questions and other functionality.	7
3.6	Creating the front end UI	To create a front end UI to access the quiz application via simple entering of name.	7
3.7	UI Design complete	Milestone for UI Design	0
4	**Website functionality**	To create the website functionality needed for the quiz to operate	2 month
4.1	**Quiz question selection system**	To create the quiz question selection system using the research of quiz formats to do so	11
4.1.1	Analyze format options	To analyze the options of the quiz formats and questions which will be followed for the system and to plan how to implement as many as possible within the system	5
4.1.2	Question decision tree	To create a question decision tree to help design the question selection system	3
4.1.3	Design the question selection system	To design the question selection system, which is designed to choose an appropriate question, based on the settings based on the rule structure which has been chosen.	3
4.1.4	Implement question selection system	To implement the quiz question selection system which should run automatically.	3
4.1.5	Quiz question selection system complete	Milestone for quiz question selection system	0
4.2	**Buzzer system**	To create the buzzer system so the players can buzz in to answer their question	5
4.2.1	Design the buzzer	To design the buzzer and to work out how the buzzer will be operated by the user	1
4.2.2	Implement the buzzer system	To implement the buzzer functionality to be able to show up on the scoreboard as well as to the host who has buzzed in if using a manual host otherwise just to allow to user who buzzed in first to insert their answer.	2
4.2.3	Buzzer system complete	Milestone for buzzer system	0
4.3	**Registration system**	To create the registration system so the players can register up the players/teams	13
4.3.1	Design the registration system	To design the registration system so the players can register individually or in teams depending on the game setting	5

4.3.2	Implement the registration system	To implement the design of the registration system	4
4.3.3	Test the registration system	To test the registration system and to make sure that the teams can register based on the mode which is selected	1
4.3.4	Registration system complete	Milestone for registration system	0
4.3 (Revised)	**Front end system**	To create the front end system so the players can register and login into the system and access the pre created games or prepare to join the "game"	13
4.3.1	Design the front end system	To design the front end system so the players can register and login as well as access all the games which are offered on the system	5
4.3.2	Implement the front end system	To implement the design of the front end system	4
4.3.3	Test the front end system	To test the front end system and make sure that all players can register and login as well as access the games which are needed	1
4.3.4	Front end system complete		0
4.4	**Player interaction**	To create the player interaction to be able to buzz in for the questions as well as to answer the questions on the screen	4
4.4.1	Design the player interaction	To design the player interaction to work out which is needed as well as how it will function. Shown via flow diagram	2
4.4.2	Implement the player interaction	To implement the player interaction following the flow diagram	3
4.4.3	Player interaction complete	Milestone for player interaction	0
4.5	**Host interaction**	To create the host interaction to be able to tie with the quiz question selection system and to be able to select the new question and submit it to the players as well as create the type of quiz on manual overriding.	4
4.5.1	Design the host interaction	To design the system which the host will be using	2
4.5.2	Implement the host interaction	To implement the design of the system which the host will be using	3
4.5.3	Host interaction complete	Milestone for host interaction	0
5	**Testing of project**	To test the project so far and to make sure that it reaches all the tasks which are discussed and no issues have occurred.	10
6	**Project reporting**	To reach all the report deliverables including the demonstration which is required for the module	43

6.1	Initial report	Write the initial report deliverable which is required for the module.	8
6.2	Interim report	Write the interim report deliverable which is required for the module.	15
6.3	Final report	Write the final report deliverable which is required for the module.	15
6.4	Demonstration	Perform the demonstration deliverable	5

Key:
Item Completed
Item Altered. See (refined)
Item in progress
Item Not completed
Item Discontinued

Appendix B: Quizmaster full ERD Database diagram



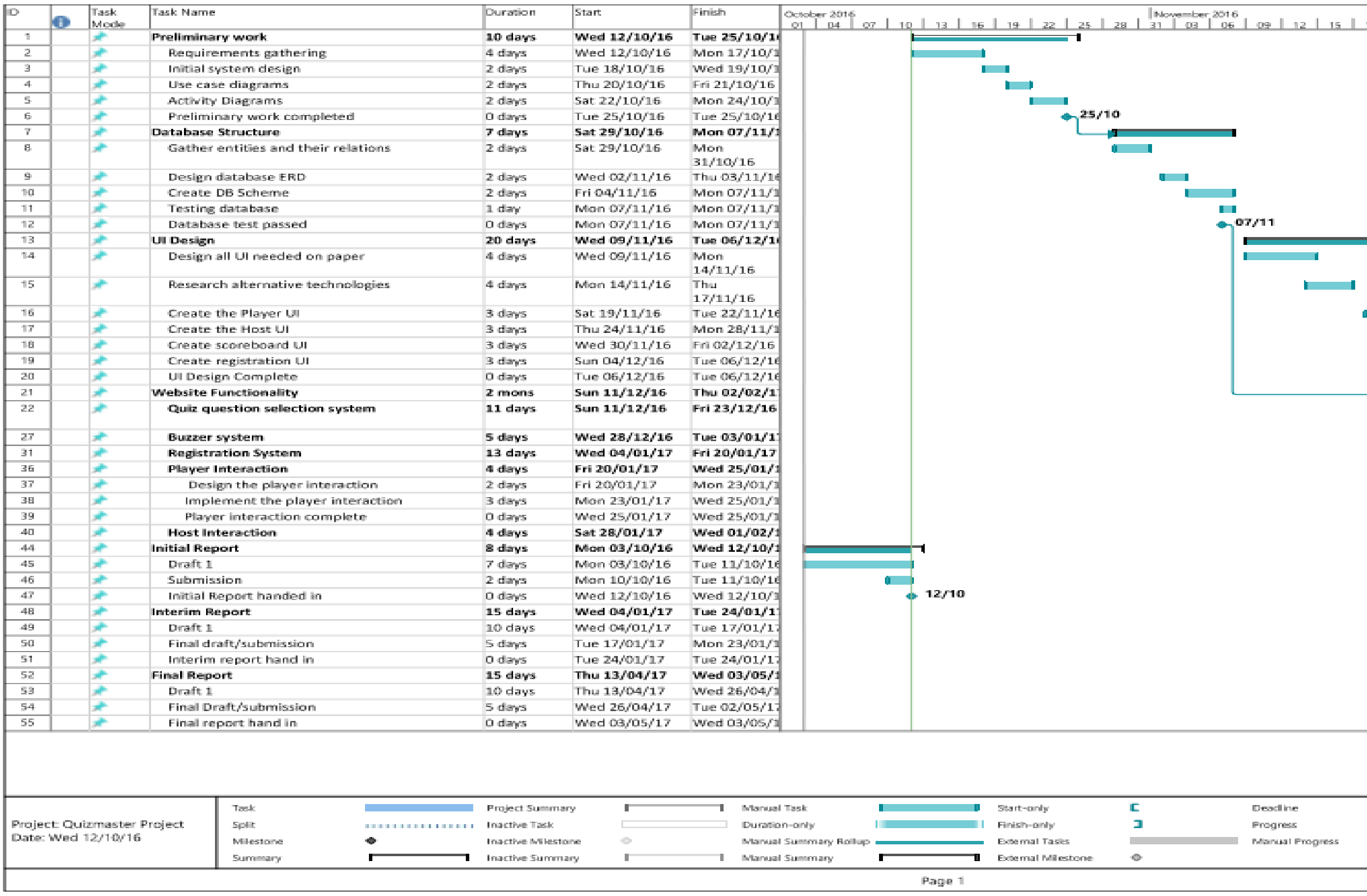
question_cat:
Question Category

Question type:
Music
Picture
etc..

Question Answer (QA)
type:
Multiple choice (MC)
Open answer (OA)

Appendix C: Initial time plan

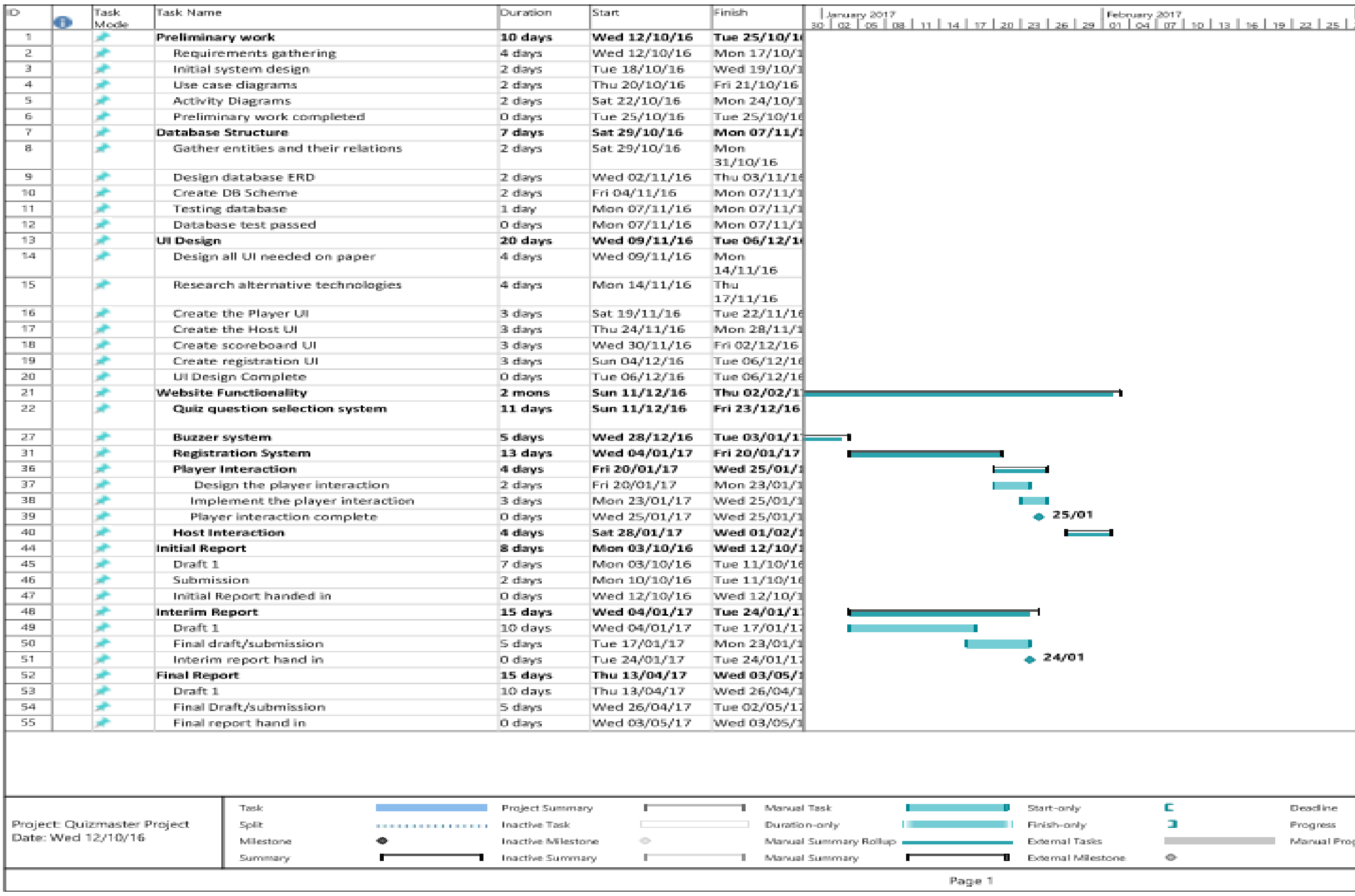
This is the time plan for months October 2016 – December 2016



ID		Task Mode	Task Name	Duration	Start	Finish
1			**Preliminary work**	**10 days**	**Wed 12/10/16**	**Tue 25/10/1**
2			Requirements gathering	4 days	Wed 12/10/16	Mon 17/10/1
3			Initial system design	2 days	Tue 18/10/16	Wed 19/10/1
4			Use case diagrams	2 days	Thu 20/10/16	Fri 21/10/16
5			Activity Diagrams	2 days	Sat 22/10/16	Mon 24/10/1
6			Preliminary work completed	0 days	Tue 25/10/16	Tue 25/10/1
7			**Database Structure**	**7 days**	**Sat 29/10/16**	**Mon 07/11/1**
8			Gather entities and their relations	2 days	Sat 29/10/16	Mon 31/10/16
9			Design database ERD	2 days	Wed 02/11/16	Thu 03/11/1
10			Create DB Scheme	2 days	Fri 04/11/16	Mon 07/11/1
11			Testing database	1 day	Mon 07/11/16	Mon 07/11/1
12			Database test passed	0 days	Mon 07/11/16	Mon 07/11/1
13			**UI Design**	**20 days**	**Wed 09/11/16**	**Tue 06/12/1**
14			Design all UI needed on paper	4 days	Wed 09/11/16	Mon 14/11/16
15			Research alternative technologies	4 days	Mon 14/11/16	Thu 17/11/16
16			Create the Player UI	3 days	Sat 19/11/16	Tue 22/11/1
17			Create the Host UI	3 days	Thu 24/11/16	Mon 28/11/1
18			Create scoreboard UI	3 days	Wed 30/11/16	Fri 02/12/16
19			Create registration UI	3 days	Sun 04/12/16	Tue 06/12/1
20			UI Design Complete	0 days	Tue 06/12/16	Tue 06/12/1
21			**Website Functionality**	**2 mons**	**Sun 11/12/16**	**Thu 02/02/1**
22			**Quiz question selection system**	**11 days**	**Sun 11/12/16**	**Fri 23/12/16**
27			**Buzzer system**	**5 days**	**Wed 28/12/16**	**Tue 03/01/1**
31			**Registration System**	**13 days**	**Wed 04/01/17**	**Fri 20/01/17**
36			**Player Interaction**	**4 days**	**Fri 20/01/17**	**Wed 25/01/1**
37			Design the player interaction	2 days	Fri 20/01/17	Mon 23/01/1
38			Implement the player interaction	3 days	Mon 23/01/17	Wed 25/01/1
39			Player interaction complete	0 days	Wed 25/01/17	Wed 25/01/1
40			**Host Interaction**	**4 days**	**Sat 28/01/17**	**Wed 01/02/1**
44			**Initial Report**	**8 days**	**Mon 03/10/16**	**Wed 12/10/1**
45			Draft 1	7 days	Mon 03/10/16	Tue 11/10/1
46			Submission	2 days	Mon 10/10/16	Tue 11/10/1
47			Initial Report handed in	0 days	Wed 12/10/16	Wed 12/10/1
48			**Interim Report**	**15 days**	**Wed 04/01/17**	**Tue 24/01/1**
49			Draft 1	10 days	Wed 04/01/17	Tue 17/01/1
50			Final draft/submission	5 days	Tue 17/01/17	Mon 23/01/1
51			Interim report hand in	0 days	Tue 24/01/17	Tue 24/01/1
52			**Final Report**	**15 days**	**Thu 13/04/17**	**Wed 03/05/1**
53			Draft 1	10 days	Thu 13/04/17	Wed 26/04/1
54			Final Draft/submission	5 days	Wed 26/04/17	Tue 02/05/1
55			Final report hand in	0 days	Wed 03/05/17	Wed 03/05/1

Project: Quizmaster Project
Date: Wed 12/10/16

Page 1



ID		Task Mode	Task Name	Duration	Start	Finish
1			**Preliminary work**	**10 days**	**Wed 12/10/16**	**Tue 25/10/1**
2			Requirements gathering	4 days	Wed 12/10/16	Mon 17/10/1
3			Initial system design	2 days	Tue 18/10/16	Wed 19/10/1
4			Use case diagrams	2 days	Thu 20/10/16	Fri 21/10/16
5			Activity Diagrams	2 days	Sat 22/10/16	Mon 24/10/1
6			Preliminary work completed	0 days	Tue 25/10/16	Tue 25/10/1
7			**Database Structure**	**7 days**	**Sat 29/10/16**	**Mon 07/11/1**
8			Gather entities and their relations	2 days	Sat 29/10/16	Mon 31/10/16
9			Design database ERD	2 days	Wed 02/11/16	Thu 03/11/1
10			Create DB Scheme	2 days	Fri 04/11/16	Mon 07/11/1
11			Testing database	1 day	Mon 07/11/16	Mon 07/11/1
12			Database test passed	0 days	Mon 07/11/16	Mon 07/11/1
13			**UI Design**	**20 days**	**Wed 09/11/16**	**Tue 06/12/1**
14			Design all UI needed on paper	4 days	Wed 09/11/16	Mon 14/11/16
15			Research alternative technologies	4 days	Mon 14/11/16	Thu 17/11/16
16			Create the Player UI	3 days	Sat 19/11/16	Tue 22/11/1
17			Create the Host UI	3 days	Thu 24/11/16	Mon 28/11/1
18			Create scoreboard UI	3 days	Wed 30/11/16	Fri 02/12/16
19			Create registration UI	3 days	Sun 04/12/16	Tue 06/12/1
20			UI Design Complete	0 days	Tue 06/12/16	Tue 06/12/1
21			**Website Functionality**	**2 mons**	**Sun 11/12/16**	**Thu 02/02/1**
22			**Quiz question selection system**	**11 days**	**Sun 11/12/16**	**Fri 23/12/16**
27			**Buzzer system**	**5 days**	**Wed 28/12/16**	**Tue 03/01/1**
31			**Registration System**	**13 days**	**Wed 04/01/17**	**Fri 20/01/17**
36			**Player Interaction**	**4 days**	**Fri 20/01/17**	**Wed 25/01/1**
37			Design the player interaction	2 days	Fri 20/01/17	Mon 23/01/1
38			Implement the player interaction	3 days	Mon 23/01/17	Wed 25/01/1
39			Player interaction complete	0 days	Wed 25/01/17	Wed 25/01/1
40			**Host Interaction**	**4 days**	**Sat 28/01/17**	**Wed 01/02/1**
44			**Initial Report**	**8 days**	**Mon 03/10/16**	**Wed 12/10/1**
45			Draft 1	7 days	Mon 03/10/16	Tue 11/10/1
46			Submission	2 days	Mon 10/10/16	Tue 11/10/1
47			Initial Report handed in	0 days	Wed 12/10/16	Wed 12/10/1
48			**Interim Report**	**15 days**	**Wed 04/01/17**	**Tue 24/01/1**
49			Draft 1	10 days	Wed 04/01/17	Tue 17/01/1
50			Final draft/submission	5 days	Tue 17/01/17	Mon 23/01/1
51			Interim report hand in	0 days	Tue 24/01/17	Tue 24/01/1
52			**Final Report**	**15 days**	**Thu 13/04/17**	**Wed 03/05/1**
53			Draft 1	10 days	Thu 13/04/17	Wed 26/04/1
54			Final Draft/submission	5 days	Wed 26/04/17	Tue 02/05/1
55			Final report hand in	0 days	Wed 03/05/17	Wed 03/05/1

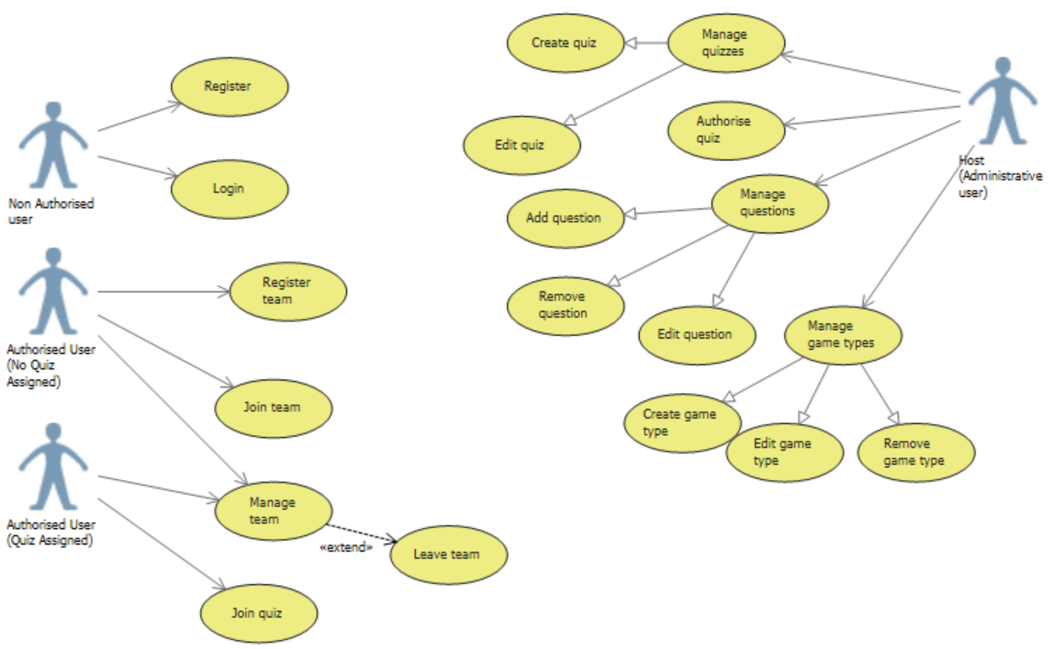
Project: Quizmaster Project
Date: Wed 12/10/16

Task		Project Summary		Manual Task		Start-only		Deadline
Split		Inactive Task		Duration-only		Finish-only		Progress
Milestone	●	Inactive Milestone		Manual Summary Rollup		External Tasks		Manual Prog
Summary		Inactive Summary		Manual Summary		External Milestone		

Page 1

This is the time plan for the months January 2017 – May 2017

Appendix D: Initial Use Case Diagram



References

AdaTheDev, 2011. *Pros and cons of mongodb.* [Online]
Available at: http://stackoverflow.com/questions/5244437/pros-and-cons-of-mongodb
[Accessed 17 January 2017].

Anon., n.d. *University Challenge rules.* [Online]
Available at: http://www.blanchflower.org/uc/show_rules.html
[Accessed 17 January 2017].

Edward, S. G. & Sabharwal, N., 2015. *Practical MongoDB - Architecting, Developing and Administering MongoDB.* s.l.:Apress.

Institute, W. P., n.d. *Online Quiz Question Types.* [Online]
Available at:
https://web.wpi.edu/academics/ATC/Collaboratory/HowTo/MyWPI/Bb9/questiontypes.html
[Accessed 24 April 2017].

Kasbe, T., 2016. *DBMS Concepts - A Practical approach.* 1st ed. s.l.:Educreation Publsihing.

League, D. P. Q., 2015. *Quiz Rules.* [Online]
Available at: https://derbyshirepubquizleague.wordpress.com/quiz-rules/
[Accessed 11 October 2016].

Logix, M., n.d. *Advantages and Disadvantages of using Relational Databases.* [Online]
Available at: http://maxlogix.blogspot.co.uk/2009/09/advantages-and-disadvantages-of-using.html
[Accessed 17 January 2017].

MongoDB, 2011. *What is MongoDB? | MongoDB.* [Online]
Available at: https://www.youtube.com/watch?v=CvIr-2lMLsk
[Accessed 17 January 2017].

Quiz, B., n.d. *What is a quiz.* [Online]
Available at: http://barcelonaquiz.com/our-quizzes/history/
[Accessed 11 October 2016].

Quizizz, 2017. *What is Quizizz.* [Online]
Available at: https://quizizz.zendesk.com/hc/en-us/articles/203610052-What-is-Quizizz-
[Accessed 21 April 2017].

Rayner, J., 2016. *Project Catalogue.* [Online]
Available at: https://canvas.hull.ac.uk/courses/4022
[Accessed 11 October 2016].

Silberschatz, A., Korth, H. F. & Sudarshan, S., 2010. *Database System Concepts.* 6th ed. s.l.:s.n.

SQA, 2007. *Advantages and Disadvantages of Scripting Languages.* [Online]
Available at: http://www.sqa.org.uk/e-learning/ClientSide01CD/page_22.htm
[Accessed 17 January 2017].

Techopedia, n.d. *What is a programming language.* [Online]
Available at: https://www.techopedia.com/definition/24815/programming-language
[Accessed 17 January 2017].

Techopedia, n.d. *What is a scripting language.* [Online]
Available at: https://www.techopedia.com/definition/3873/scripting-language
[Accessed 17 January 2017].

Tezer, O., n.d. *SQLite vs MySQL vs PostgreSQL a comparison of relatyional database management systems.* [Online]
Available at: https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems
[Accessed 17 January 2017].

Thakur, D., n.d. *What is DBMS? Advantages and Disadvantages of DBMS.* [Online]
Available at: http://ecomputernotes.com/fundamental/what-is-a-database/advantages-and-disadvantages-of-dbms
[Accessed 21 April 2017].

Tutorial, P., n.d. *What is PostgreSQL.* [Online]
Available at: http://www.postgresqltutorial.com/what-is-postgresql/
[Accessed 17 January 2017].

Vaish, G., 2013. *Getting Started with Nosql.* s.l.:Packt Publishing Ltd.