The name of your project

CS39440 Major Project Report

Author: Your Name ([rhs24@aber.ac.uk](mailto:rhs24@aber.ac.uk))

Supervisor: Dr Edel Sherratt (eds@aber.ac.uk)

19th February 2019

Version 1.0 (Draft)

This report is submitted as partial fulfilment of a BSc degree in  
Computer Science (G400)

Department of Computer Science

Aberystwyth University

Aberystwyth

Ceredigion

SY23 3DB

Wales, UK

Declaration of originality

I confirm that:

* This submission is my own work, except where clearly indicated.
* I understand that there are severe penalties for Unacceptable Academic Practice, which can lead to loss of marks or even the withholding of a degree.
* I have read the regulations on Unacceptable Academic Practice from the University’s Academic Registry (AR) and the relevant sections of the current Student Handbook of the Department of Computer Science.
* In submitting this work, I understand and agree to abide by the University’s regulations governing these issues.

Name Rhodri Smith

Date 22/04/2020

Consent to share this work

By including my name below, I hereby agree to this project's report and technical work being made available to other students and academic staff of the Aberystwyth Computer Science Department.

Name Rhodri Smith

Date 22/04/2020

Acknowledgements

I am grateful to friend and family for mental and financial support, not just within this project but through the 3 years of university so far. I am also grateful to the tutors and

I’d like to thank…

Abstract

This document will contain the design and implementation of a web based game that interacts differently based on the type of user. The project is aimed at primary school students and is designed to allow a student to play a simple game, where a score is recorded and uploaded to a database. The students and their score will then be visible to a staff member, wherethey will be able to change the student’s difficulty for the game. A guest account was also necessary for a someone to be able to test the site. The website contains a login screen for the user and will require the user to either sing in or be a guest to visit other pages.

The project was designed with the basic web languages and within an IDE called Atom[7]. The languages used where front end languages and backend languages.

Contents

[1. Background, Analysis & Process 6](#_Toc39362978)

[1.1. Background 6](#_Toc39362979)

[1.1.1. Languages 6](#_Toc39362980)

[1.1.2. Other sites 6](#_Toc39362981)

[1.1.3. The Game 6](#_Toc39362982)

[1.1.4. The Table 6](#_Toc39362983)

[1.1.5. Security 7](#_Toc39362984)

[1.1.6. Database 7](#_Toc39362985)

[1.1.7. Design ideas 7](#_Toc39362986)

[1.2. Analysis 7](#_Toc39362987)

[1.2.1. Aims and objectives 7](#_Toc39362988)

[1.2.2. Issues and Approach 8](#_Toc39362989)

[1.3. Process 9](#_Toc39362990)

[2. Design 10](#_Toc39362991)

[2.1. Overall Architecture 10](#_Toc39362992)

[2.1.1. Introduction, what I wanted it too to look like too 10](#_Toc39362993)

[2.1.2. UML and other diagrams 10](#_Toc39362994)

[2.2. Detailed Design 10](#_Toc39362995)

[2.2.1. Login page 10](#_Toc39362996)

[2.2.2. Game Page 11](#_Toc39362997)

[2.2.3. Staff and Admin Page 11](#_Toc39362998)

[2.2.4. Even More Detail 11](#_Toc39362999)

[2.3. User Interface Design 11](#_Toc39363000)

[2.4. Other Relevant Sections 12](#_Toc39363001)

[3. Implementation 13](#_Toc39363002)

[4. Testing 14](#_Toc39363003)

[4.1. Overall Approach to Testing 14](#_Toc39363004)

[4.2. Automated Testing 14](#_Toc39363005)

[4.2.1. Unit Tests 14](#_Toc39363006)

[4.2.2. User Interface Testing 14](#_Toc39363007)

[4.2.3. Stress Testing 14](#_Toc39363008)

[4.2.4. Other Types of Testing 14](#_Toc39363009)

[4.3. Integration Testing 14](#_Toc39363010)

[4.4. User Testing 14](#_Toc39363011)

[5. Critical Evaluation 15](#_Toc39363012)

[6. Annotated Bibliography 16](#_Toc39363013)

[7. Appendices 17](#_Toc39363014)

[A. Third-Party Code and Libraries 18](#_Toc39363015)

[B. Ethics Submission 19](#_Toc39363016)

[C. Code Samples 19](#_Toc39363017)

# Background, Analysis & Process

## Background

### Languages

Researching the different languages was a major section of research, without this the project would not have been able to be completed. This was mainly looking at the way the languages should be set out, e.g. should the languages be in separate files. The research also took into consideration how the files should be set out, e.g. for the database connection it should not be stored within the server, another would be to have a main CSS file that would store data for items that are on every page, e.g. the footer. Node.js was another language that was investigate briefly for the project

### Other sites

Some of the research that was done before the web application was started was to look at similar websites and try to look at the pros and cons of the websites. This meant that the project would be able to use the advantages of those websites while not having their disadvantages. An example of some pages, where that they did not store a user’s score for long term viewing while others forced the users to login to view the pages.

### The Game

Creating the game was a large area of research, how and what sort of game was. During this phase of the project a few different areas to do this was investigated. One area was creating a canvas game within JavaScript. Another method was to use a framework that uses a canvas and WebGL. This would have been phaser [1], which is free to use software that works within the browser. The final way that was investigated was to use just basic HTML to create basic games. Using just html would lower the amount of animation that could be done within the game, however.

The canvas element that could have been used for the game would have allowed a dynamic 2d shapes. It is not designed specifically around creating games however, it is possible to make them within the canvas.

Research was also looked into how the questions and answered would be saved on the page, would they be stored on the database and pulled into an array for the JavaScript to run, or would the page randomly create the questions. Storing the questions would require a table to store them all and a way to upload new questions. Randomly making them would not require this, but would require the creation of random questions and answers that are appropriate, e.g. incorrect answers need to be realistic to the real answer.

A multi canvas approach was looked into for this project, since detection of the canvas itself is easily possible and could allow for some sort of visually stimulating affect for when the user clicks on an answer. Issues again with this however is scalability for different resolutions

### The Table

Creating the table was another section that was investigated. During this time DataTables [2] was one table system that was looked into. DataTables [2] allow a more advanced way to create and interact with tables, this includes a search feature and a feature that added more than one page that a user can view. Data Tables [2] is a free to use plugin that uses jQuery JavaScript

### Security

Security was another item that needed to be researched, this is to stop anything happening that should not. The research was done into looking not only security of user’s data, but also into the security of the website, e.g. can a user just break the website by entering the wrong data or entering data where they should not. The security is paramount to the website not being taken down or data being stolen. A section of research within security investigated the login area of the project and investigated PHP commands that would do thing such as hashing passwords. Hashing passwords is a secure way to store the passwords. It changes the passwords from plane text and changes it into a bunch of random text. This means that if someone gets access into the database, the passwords are still security for the users.

### Database

Research into the database that was storing the information was a crucial part for the project, as it would need to put into detail how the data would be stored, what level of normalisation would be used and the formats that the data would be stored as, e.g. text or int. The type of database was also investigated and the two main types that where investigated was a PostgreSQL and MySQL. These are both popular database management systems and both are free to use (MySQL under the terms of the ‘GNU General Public License [3]’). PostgreSQL is more advanced with its datatypes, allowing support of arrays a user defined types while MySQL only allows standard types. Some of the detail that was taken into consideration was from a website: PostgreSQLTutorial.com [4]. This website took into consideration the pros and cons of PostgreSQL and MySQL. The database research also required investigation into how a group of users (school) data would be stored. This meant looking into should they have an entire database to themselves or only a table.

Looking at using the database long term, using command line was an option however, research was done into using other methods one method was the phpMyAdmin [5] tool. This tool allows for easy maintenance of use of the database without having to have the knowledge of the command line commands. It also allows a user to control the website from anywhere, whereas to do this otherwise a user would need to use something such as TeamViewer [6] to control the database off site. TeamViewer lets a user to connect to a device and control it from anywhere, e.g. at home.

### Design ideas

Before starting the project, a few design ideas where looked into, like how the user would get from one screen to another. This would range from the colours the website will have, to how the user would navigate the website. Would the user be able to navigate the website themselves or would they only be allowed onto certain pages that they are taken too?

## Analysis

### Aims and objectives

For the project, there were a few main aims and objectives that needed to be completed to fulfil the project. Other smaller aims where also made, however if these did not get completed, the main areas of the project would still be completed, and the project would run as needed. These aims where to have a game that would upload a user’s (student) score to a database, an area that another user (teacher) can view the score. The teacher would also need to be able to increase the difficulty for the students game. A login screen was also necessary for the project and an admin section. There would also need to be an option for a user to connect to the website through a guest account that would not save any details

One of the main aims and objectives that where created was to create a simple game that would track a user’s score and then upload it to a database. This Game would need to be a basic game that’s aimed towards young people that are primary school age.

Another main objective was to complete a login screen for the user. This would allow the user to login only if they could. The page would have to take 4 different types of accounts to 2 different places. The first being a guest account and the student account, that needed to the game page of the website. Other 2 accounts are the

### Issues and Approach

#### Languages

During the time researching the project, a few issues came to light. One being that php and JavaScript did not work very well together, or in some areas not at all. This being an issue when needing to pass variables such as difficulty to the PHP functions, e.g. changing the user’s difficulty. To get around this issue, the website used forms and the submit button, as the php could read the data that was entered into the forms and work through it that way. Another way this could have been done would be through cookies and changing data within it and then getting the PHP to read that file for the needed data.

#### Multi-Page Site

Another issue that came up was how would the website remember the users details once they go onto the next page. While researching, 2 ways came up more common that not on how to do this. Cookies and sessions, cookies storing the information locally and sessions storing them on the server side. For the project, sessions where used as no long-term data needed to be stored. The data that was passed from one page to another was just the user’s username and table name (school), if the user is an admin, another session is sent that tells the website that they are an admin.

#### The Game

When creating the games for the website, research was done into what type of game should be created and how. Within the project, the HTML, non-canvas approach was taken, this was partly to do with the webpage being able to change size based on monitor resolution. This approach was take as it would be easily reusable and would be easy for the user to play/understand. The main issue with a canvas approach, is that because the canvas is not designed for game or really even dynamically changing items, e.g. items moving on the canvas. This does not mean its not possible however it is mainly designed to allow more graphically enhanced items to be created. Another issue with it is that to detect clicks, the coordinates of the clickable items must be imputed into the code. This meant that a non canvas approach felt a lot more appropriate for the needed game and design.

#### The Table

The approach for the table that was used within the project was to use the DataTables [2] as it gives a more advanced while still being easy to use for the user and developer. DataTables [2] uses jQuery and JavaScript which is easy to work with and allows events such as onclick events for when the user clicks on specific rows or columns. DataTables[2] also gives a good professional look to the table, while also giving it some ease of life features, e.g. a user can choose how many items they can see at once on the table. This apporch allowed a much more professional feel to the website while being mostly simple to use.

#### Security

Security is another issue that needed to be investigated and taken into consideration when creating the website. Website require a lot of security to stop users from either stealing information or breaking the website.

#### Add New Students

The Website will not allow new students to be created through the site. This is partly to do with it not feeling appropriate for a new user to be added, as they would need to be added through the staff members section, which should not be able to access more than the users details and allow them to change the difficulty. The admin is also notable to add a user to the database through the website, however if they had access to the database, they would be able to add a user from there. A school would need to register their students and staff members when singing up to the website. A school will then need to contact the company to be able to access the database and add a student though the database. This was partly to do with the ease of adding someone through the phpMyAdmin [5]. Which also lowers the risk of errors getting into the database, e.g. duplicates. This does not mean however, the users details cannot be changed, data such as the users name and certain other details can be changed through the website however, so an error such as a misspelling exists, this can be changed by one of the admins within the website

## Process

# Design

## Overall Architecture

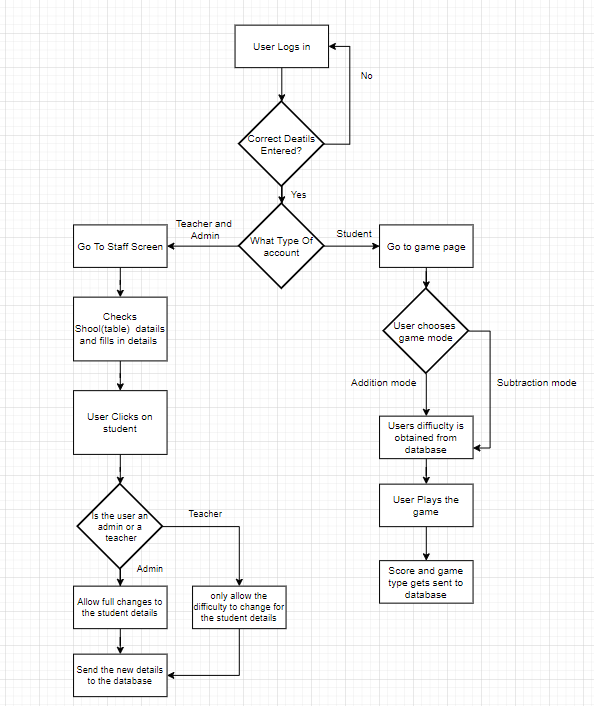
### Introduction, what I wanted it too to look like too

### UML and other diagrams

The diagram bellow shows the basic flow for the website, from what the user does to what the website also does backend. This is just a basic diagram to easily show the reader the basics of the website. As the reader can see the user will start at the top on the login screen. Once the user logs in, the system will then check if the password is correct, if the password is correct, the website will then check what account type the user is.

If the user is an admin or staff member, the diagram would go down the left branch. The left side shows the user being taken to the staff screen. This system then checks the detail of that user to fill in the table with appropriate details, e.g. students. The next box shows what happens when a user clicks on one of the table rows. This shows what a user can do based on what account type they are, if they are a staff account, all they can change is the difficulty of their students, however, if they are an admin account, they can change all the details that are shown on the table. The last box on this side shows the data that the user has entered, changing the clicked-on item in the database, not just the table. This is just a basic insight on how the staff and admin pages work, this is not a detailed section for the reader to view.

The right-hand side of the diagram shows where the user would be taken if they where the student or guest profiles. The first box within this branch of the diagram shows the user being taken to the game page. This is the page that contains the game that the user will be playing. The user can then choose the game mode that they would like to play the game on, this is a diamond as there are multiple options that change how the game can run, however still all lead to the same outcome for the rest of it. The next box then shows the system getting the users difficulty, this gets the difficulty from the online database and is not something that the user sets when starting the game. The next box shows the user playing the game. The users score will change based on the amount of correct answers they get e.g. a correct answer gives +1 to their score. The score will always start at 0. The next box and last box show’s the users score being sent to the database. This then would show up on the staff pages table as a new score. This is also a very basic flow for the website.



## Detailed Design

### Login page

The login page’s design is designed to look simply, allowing people to easily go onto the site without any issues. The design makes it, so the user only has to enter their details in and are then taken to the correct place. The user must enter their details into 3 boxes, one being the school box, this would be the table name within the code. The second two input boxes are for the username and password of the users, this gets used within the database to search for the user. The third box would only need to be filled in by and admin, and would just be the school name that they wish to connect too

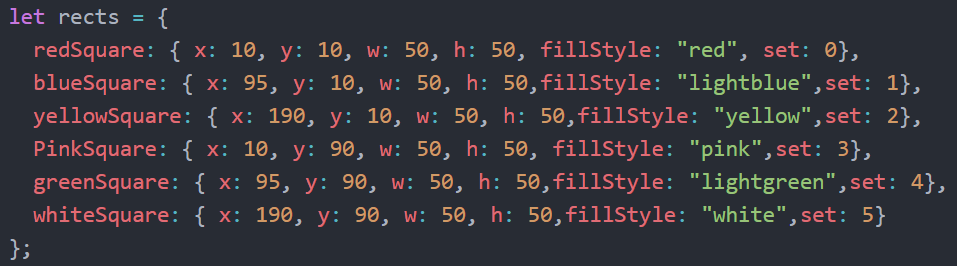
The page uses HTML forms and PHP to complete the login and take the user to the correct page. The pages that the user will be sent to will be dependent on the type of user, a student and guest will be taken to the game page, while the admin and the staff accounts will be taken to the staff page. This page must be the first page that a

### Game Page

The Game screen was also created in a way that the user can easily use and get through. It works by requiring the user to click on the ‘click me’ button and then after that the user just clicks on one of the coloured boxes to answer. The game will then automatically submit the score for the user. The game for the game page will be in the centre of the screen, with the buttons that the user can click on being just under them. The question will come up above the boxes and the answers will show up in the boxes themselves. Making it easier for the user to know what box to click on to give their answer. The text within the boxes and the boxes will be reasonably sized for the user to see the answer.

The users score is sent to the database through a form. The user will not see this form as never shows up on screen and the form is automatically filled in and submitted. This was done so that someone cannot accidently not submit their score and close the page.

The game works by giving the player a completely random question, this is not a question that is stored anywhere. This lowers the chance of a use getting a repeat question and does not require a user to put every possible question they want into a database. The game is based off users clicking on 6 coloured squares, this square is not coded onto the HTML itself, they are instead added after through JavaScript on the load of the page. The data for the squares are stored within an array in the JavaScript. This array stores the coordinates for the squares, the hight, width and colour of the squares. This allows the user to easily add more or remove the number of squares when needed. This could also be used to change the number of squares based on difficulty; however, the project does no incorporate this at the time of this report.



Since the squares are not created within the HTML and the JavaScript must create them instead, a function was required that would allow the creation and put them into the correct place. A for each loop was also required to run, as it would run for the amount of items in the array, and set the details of the squares to be the necessary details. The function was called DivMaker, since it was creating the squares for the game. The ‘currentDiv variable gets the ‘JavaGame’ divider and the code bellow inserts it so that the squares are all put into the ‘JavaGame’ division tag.

A switch() statement is used for the chosen game mode to be used, within the switch() statement there are if statements for the difficulty the user has been given, this changes a variable called ‘randomNumberDifficulty’, this will change from 10, if the difficulty is 1, too 100 if the difficulty is 2. This then would change the questions from being single digit questions, e.g. 1+2, to double digits, e.g. 25+33. There is a break for this loop that runs when the user get to question 10. This statement changes the question section to show the users answer instead of a new score and removes all of the squares on the website.



Since the game makes random question, a random number generator was needed, this was the Math.random (), this creates a random integer, however, the issue with this is that it contains numbers after the decimal point. Which is why within the code, the Math.floor(Math.random() \* randomNumberDifficulty); was used. This created a random number, which is then multiplied by the randomNumberDifficulty variable. The Math.floor() then forces the number to be a full number and not have a decimal point behind it.

### Staff and Admin Page

The staff and admin page contain a table within the centre, this table contains all the students that are within the table. The staff member or admin can edit the details within the table by clicking on the user’s row. By clicking on a student within the table, a form pops up and the user can change the details of the student they clicked on. If they are not an admin, they can only change the user’s difficulty. The admins can change all the details that are shown within the table. The admin can also remove any users that are on the table.

The user changes the datils through a form that then sends the users details into a php function that then changes the database. The remove button also uses php to remove the user from the database. Both the ‘submit’ and ‘remove’ buttons are submit buttons for the form, they both also use a hidden column within the table. The hidden column is the users number, this is something that a staff member would not necessarily need to see and could just lead to confusion, this cannot also be changed and is used to change the users details or remove the user from the database table.

A php function is also used to fill in the table, it does this by searching through the database and then uses the ‘echo’ command to place the text into the html code. The echo command places the <tr> and <td> tags into the page for the number of items there are in the database, it does this with a while loop. These tags are the basic HTML table tags that can be used anytime within a table within HTML. It puts the students onto the table through a while loop that runs for the amount of students that are returned from the database and with the correct details for that student.

### Security

Each page, other than the login, must have had the user to login to access them. If does this by checking if a session is running. Which is done by checking for a session that stores the user’s username if it is empty. It will take the user back to the login page using the header() function, this function allows the developer to send people to other URLs. This header is partly needed as without it, pages such as the staff page would not show anything as it uses the user’s details that they logged in with to get the details for the table.

## User Interface Design

## Other Relevant Sections

# Implementation

### 

# Testing

## Overall Approach to Testing

## Automated Testing

### Unit Tests

### User Interface Testing

### Stress Testing

### Other Types of Testing

## Integration Testing

## User Testing

# Critical Evaluation

# Annotated Bibliography

[1] photonstorm.com, P., 2020. Phaser - A Fast, Fun And Free Open Source HTML5 Game Framework. [online] Phaser.io. Available at: <https://phaser.io/> [Accessed 28 April 2020].

[2] Datatables.net. 2020. Datatables | Table Plug-In For Jquery. [online] Available at: <https://datatables.net/> [Accessed 28 April 2020].

[3] Gnu.org. 2020. The GNU Operating System And The Free Software Movement. [online] Available at: <https://www.gnu.org/> [Accessed 29 April 2020].

[4] Postgresqltutorial.com. 2020. PostgresqlVs. Mysql: A Comprehensive Comparison. [online] Available at: <https://www.postgresqltutorial.com/postgresql-vs-mysql/> [Accessed 30 April 2020].

[5] contributors, p., 2020. Phpmyadmin. [online] phpMyAdmin. Available at: <https://www.phpmyadmin.net/> [Accessed 30 April 2020].

[6] TeamViewer. 2020. Teamviewer – Remote Support, Remote Access, Service Desk, Online Collaboration And Meetings. [online] Available at: <https://www.teamviewer.com/en/> [Accessed 30 April 2020].

[7] Atom. 2020. A Hackable Text Editor For The 21St Century. [online] Available at: <https://atom.io/> [Accessed 3 May 2020].

# Appendices

* 1. Third-Party Code and Libraries
  2. Ethics Submission
  3. Code Samples