**A Level computer Science**

Component 3

Physics Homework App

A black and blue logo

Description automatically generated

By: Konstantinos Papadopoulos

For: A. Issa

**Stoke on Trent Sixth Form College**

Table of Contents

[Chapter One: Analysis of the problem 3](#_Toc190263921)

[1.1 Problem Identification 4](#_Toc190263922)

[1.2 Possible Computational Methods 5](#_Toc190263923)

[1.3 Stakeholders analysis 6](#_Toc190263924)

[1.4 Research of solutions for similar problems 10](#_Toc190263925)

[1.5 Hardware and software requirements 11](#_Toc190263926)

[1.6 The requirements of the solution 12](#_Toc190263927)

[1.7 Features of the solution 13](#_Toc190263928)

[1.8 Success Criteria 14](#_Toc190263929)

[1.9 Limitations of the solution 15](#_Toc190263930)

[Chapter Two: Design 16](#_Toc190263931)

[2.1 Introduction 17](#_Toc190263932)

[2.2 Decomposition of the problem 18](#_Toc190263933)

[2.3 How All Solution Parts are Linked 19](#_Toc190263934)

[2.4 Database Design 20](#_Toc190263935)

[2.5 Design of other Parts of the Solution 26](#_Toc190263936)

[2.6 Stakeholders involvement 31](#_Toc190263937)

[Chapter Three: Iterative Development and Testing 32](#_Toc190263938)

[3.1 Introduction 33](#_Toc190263939)

[3.2 Stage 1: Building the Database 33](#_Toc190263940)

[3.2 Stage 2: Building The Log In 34](#_Toc190263941)

[3.2 Stage 3: Building The Validation 47](#_Toc190263942)

[3.2 Stage 4: Building The Admin View 53](#_Toc190263943)

[3.2 Stage 5: Building The Student View 54](#_Toc190263944)

[3.3 Final Review, Improvements and Corrective Actions 55](#_Toc190263945)

[Chapter Four: Evaluation 56](#_Toc190263946)

[4.1 Introduction 57](#_Toc190263947)

[4.2 Testing to inform evaluation 57](#_Toc190263948)

[4.3 Evaluation 58](#_Toc190263949)

# Chapter One: Analysis of the problem

## Problem Identification

While studying Physics at A-Level, we used multiple websites to do our homework. This was due to the fact that all of the different websites had their own flaws. The main flaw that I found is that no website that is available to us allows for both calculations and long answers to be entered as answers.

This is an example of a website allowing to answer a calculation question. As you can see in this example, you can only enter a value as an answer and then the website tells you if it is right or wrong. This is a big problem because, if you get the question wrong, you and your teacher will not be able to see where the problem in the calculation is and how to fix it.



As shown in this example, this particular website allows for a long answer to be entered as an answer. However, the limitation of this is that when you are marking your answer, it can only be marked as right or wrong. The solution to this problem is to allow the student to give their answer a mark, by checking their answer against the mark scheme, out of the number of marks that the teacher set. This will allow the student to see what they would score in the real exam.

## 1.2 Possible Computational Methods

## 1.3 Stakeholders analysis

1.3.1 Stakeholder Introduction

The demographic for my project is for teachers and students in the Physics A-Level. For my stakeholders I picked one Physics teacher and three Physics students. The teacher stakeholder is Mina and my student stakeholders are Ali, Jahin and Violet. I picked a teacher and three students for my stakeholders because I wanted to have many different perspectives where I can take ideas from.

I asked each group of stakeholders seven questions. I asked the teacher group the following questions:

1. What resources do you currently use to set physics homework?
2. What are some good features of the current resources that you use?
3. What are some bad features of the current resources that you use?
4. If a new homework website was made, what would you like to see included in it?
5. As a teacher, what are the biggest challenges when setting and marking homework?
6. What would make the website easier to navigate for you?
7. Anything you would like to add that wasn’t included in the questionnaire?

I asked the student group the following questions:

1. What resources do you currently use for your physics homework?
2. What are some good features of the current resources that you use?
3. What are some bad features of the current resources that you use?
4. If a new homework website was made, what would you like to see included in it?
5. What would make the website easier to navigate for you?
6. How can the website motivate you to do your homework?
7. Anything you would like to add that wasn’t included in the questionnaire?

1.3.2 Stakeholder Interviews

My teacher stakeholder interview was with Mina, an A-level Physics teacher. She answered my questions as follows:

What resources do you currently use to set physics homework?

“I use SENECA and Bromcom”

What are some good features of the current resources that you use?

“SENECA provides active learning which can be very beneficial for a student. Bromcom allows for the parents to easily check their kid’s progress, homework and behaviour at school.”

What are some bad features of the current resources that you use?

“Both websites have too many steps when setting a homework”

If a new homework website was made, what would you like to see included in it?

“I’d like to see a feature allowing the teacher to check a student’s progress”

As a teacher, what are the biggest challenges when setting and marking homework?

“Time. Setting a homework can be very time consuming because it involves a lot of steps”

What would make the website easier to navigate for you?

“Limiting the number of clicks required to reach important pages and actions. Also, a clean and uncluttered design”

Anything you would like to add that wasn’t included in the questionnaire?

“No”

My first student stakeholder interview was with Ali. He currently does Computer Science, Maths and Physics at A-Level. His responses were as follows:

What resources do you currently use for your physics homework?

“I currently use Isaac Physics and Carousel Learning for my homework”

What are some good features of the current resources that you use?

“I really like that there is a menu that allows you to review new and completed assignments. I also really like that there are hint options for when I am struggling with a question”

What are some bad features of the current resources that you use?

“One thing I dislike about Carousel Learning is that if I accidently go to the next question, I can’t go back to the previous question. I also really dislike that I can’t do theory questions on Isaac and calculations on Carousel which makes it really inconvenient to do my homework. Another thing I dislike about my current resources is that I don’t get notifications when new homework is set.”

If a new homework website was made, what would you like to see included in it?

“I would really like to see a feature that show how much progress I am making on my homework and how much progress I am making overall.”

What would make the website easier to navigate for you?

“My current resources have a bunch of useless stuff on the screen which makes it hard to navigate so I think a clean UI and design would make the website easier to navigate for me.”

How can the website motivate you to do your homework?

“A point reward system or a praise system when I get a question right and also a class leaderboard to see where I am compared to my classmates”

Anything you would like to add that wasn’t included in the questionnaire?

“No”

My next student stakeholder interview was with Jahin. He is studying Physics, Computer Science and Further Maths. His responses were as follows:

What resources do you currently use for your physics homework?

“I use past papers, Isaac Physics and Carousel Learning”

What are some good features of the current resources that you use?

“They let me work at my own pace, there’s no timers that rush me and stress me out”

What are some bad features of the current resources that you use?

“Some resources don’t have hints on how to solve questions I am stuck on”

If a new homework website was made, what would you like to see included in it?

“An option for hints for when I am stuck on a hard question”

What would make the website easier to navigate for you?

“A simple UI with no over-the-top styling”

How can the website motivate you to do your homework?

“Maybe a leaderboard comparing you against classmates”

Anything you would like to add that wasn’t included in the questionnaire?

“I would like to see that the app has a low bandwidth usage because the Wi-Fi in some places is really bad”

My final student interview was with Violet. She is studying Physics, Maths and Geography. Her interview went as follows:

What resources do you currently use for your physics homework?

“I use Physics and Maths tutor, questions from the textbooks and slides from the teachers”

What are some good features of the current resources that you use?

“They are easily accessible and have a wide range of topics”

What are some bad features of the current resources that you use?

“They have bad layouts which makes it hard to find the resources I need sometimes”

If a new homework website was made, what would you like to see included in it?

“I would like to see a main screen with all important features included in that screen”

What would make the website easier to navigate for you?

“A better layout with no clutter”

How can the website motivate you to do your homework?

“Some kind of praise system would be great”

Anything you would like to add that wasn’t included in the questionnaire?

“No”

1.3.3 Stakeholder Conclusion

From the interviews, it is clear that my stakeholders prefer a clean and uncluttered UI. I will try my best to incorporate this into my solution. I will do this by removing unnecessary buttons and have a main window with all of the important actions.

My teacher stakeholder said that there are too many steps when setting a homework which can be very time consuming. {yap}

My stakeholders also pointed out to me that they prefer to work at their own pace. Due to this, I will not include any timers when doing assignments {maybe some more}.

## 1.4 Research of solutions for similar problems

1.4.1 S**imilar existing solutions**

Seneca Learning

How does it work?

A screenshot of a video game

Description automatically generatedOne similar solution to my problem that I found is a website called Seneca Learning. This website has pre-made modules for many subjects and exam boards. The layout for homework does not allow students to do calculations. When completing a homework on Seneca Learning, the answer is given to you as an explanation and you are then given a question to answer (usually a complete the sentence question). This is website would be really good for students at the GCSE level but for A-Level students it would not be useful.

What components and features can be borrowed?

Seneca learning has many features that are very useful for a student to have. It has a class system that allows for teachers to create their own classes and then set homework for all the students in those classes. This feature will be borrowed because it will make it easy for teachers to set homework for their classes and it would allow teachers to group their students in their respective classes.

One feature that I find to be a bad feature, is the ability for a student to create their own account. I think this is bad because, when students are signing up, they can choose if they are a student, teacher or parent. Having access to this feature means that a person with malicious intentions can create multiple teacher accounts and cause a big problem in a school system. Due to this, for my solution, I will only allow teacher to create student accounts and then having the system send the student an email with their details.

Physics & Maths Tutor

How does it work?

This solution does not help my problem but it is a good example to use. How this solution works is that it allows a student to choose a subject, exam board and topic and then do past papers and questions on that topic. It doesn’t allow for any answers to be entered as it only gives the student access to pdf files which they can print out and write on.

What components and features can be borrowed?

I think the only feature that I can borrow from this website is the ability to choose exam boards. When incorporating this feature in my solution, I will allow the teacher to pick what exam board an assignment would go under when setting a homework assignment. I will be borrowing this feature because, if the students know what exam board a question is from, they can see how an answer is structured in that specific exam board and they can see what structure they would need in the exam.

## 1.5 Hardware and software requirements

My solution will require a computer with the standard I/O devices (keyboard, mouse and monitor) so the user can interact with the application. My solution is a python based application. However, the user will not need to have python installed as the application will be compiled into an executable file. An internet connection will also be required so the application can access the database. This will be needed so the user can log in and see their assignments. The following requirements are the minimum requirements needed for the application to run. Most computers and laptops will already meet the requirements as python executable files are not very demanding when it comes to hardware and software requirements.

Hardware:  
- Processor: Any multicore processor. This will allow for faster computations and more efficient multitasking.

- RAM: At least 1GB as the application has a GUI and it will run slow with less RAM.

- Storage: At least 100MB to store the executable file and any temporary files.

- I/O devices: Keyboard, mouse and monitor. This will allow the user to use and interact with the application.

Software :

- Operating System: Windows 7 or above, macOS or a Linux based system. These are the current operating systems supported by python.

- Other software will not be needed as all the libraries used will be already bundled with the executable file.

## 1.6 The requirements of the solution

## 1.7 Features of the solution

## 1.8 Success Criteria

## 1.9 Limitations of the solution

# Chapter Two: Design

## 2.1 Introduction

## 2.2 Decomposition of the problem

2.2.1 Decomposition Diagram

2.2.2 Flow Charts / Data Flow Diagram

2.2.3 Input Process Output

## 2.3 How All Solution Parts are Linked

2.3.1 State Diagram of the different forms/parts

2.3.2 How different functions /classes are connected

## 2.4 Database Design

### 

Normalisation

### 

Data Dictionary For Each Table

Table name: admin\_acc

This table will store all the admin and teacher accounts. More rows can’t be added to this table unless the database is accessed directly in the database website.

It will store the following information:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Data Type | Length | Rules | Comments | Example |
| id | Integer | One digit and above | N/A | Primary key. Generated by the database | 13 |
| email | String | N/A | Must be in the form: example@domain.com | Stores the user’s email | “JohnSmith@hotmail.com” |
| password | String | N/A | N/A | Stores the user’s hashed password | 1d460455f292dc60ed53ad  bea5365bfa27262c25e724  a6eecec799f203a69d7a |
| name | String | More than 5 characters | Must have a space in the middle | Stores the user’s name | “John Smith” |

Table name: main\_acc

This table stores all the student accounts. This table can be changed by any admin or teacher when creating an account.

It stores the following information:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Data Type | Length | Rules | Comments | Example |
| Id | Integer | One digit and above | N/A | Primary key. Generated by the database. | 32 |
| name | String | More than 5 characters | Must have a space in the middle | Stores the user’s name | “Rachel Smith” |
| password | String | N/A | N/A | Stores the user’s hashed password | 1d460455f292dc60ed5  3adbea5365bfa27262c  25e724a6eecec799f20  3a69d7a |
| email | String | N/A | Must be in the form: [example@domain.com](mailto:example@domain.com) | Stores the user’s email | “RachelSmith@gmail.com” |
| teacher\_id | Integer | One digit or above | N/A | Foreign key. Links with the id column in the admin\_acc table | 21 |

Table name: stud\_classes

This table stores all the class names and which teacher own them. This table is updated when a class is created or deleted.

It stores this information:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Data Type | Length | Rules | Comments | Example |
| Id | Integer | One digit and above | N/A | Primary key. Generated by the database. | 54 |
| class\_names | String | More than one character | N/A | Stores the name of the class | “Physics Class 7E” |
| teacher\_id | Integer | One digit or above | N/A | Foreign key. Links with the id column in the admin\_acc table | 12 |

Table name: assignments

This table stores the information of every assignment. This table is needed to allow for many-to-many table relationships to happen. It also makes looking for an assignment easier.

It follows this structure:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Data Type | Length | Rules | Comments | Example |
| assign\_id | Integer | One digit and above | N/A | Primary key. Generated automatically by the database | 31 |
| title | String | More than 2 characters | N/A | Stores the title of the assignment | “The Solar System” |
| class\_id | Integer | One digit and above | N/A | Foreign key. Links to the stud\_classes table. Stores the class which the assignment is assigned to . | 54 |
| due\_date | Date | N/A | Must be in the format yyyy-mm-dd | Stores the due date of the assignment. Chosen by the teacher | 2024-09-14 |
| teacher\_id | Integer | One digit and above | N/A | Foreign key. Links to the admin\_acc table. Stores which teacher set the assignment | 13 |
| title\_id | String | N/A | Must be in the format a12345678 | Created automatically when creating an assignment. Id is unique to each assignment | "a99217410" |

Table name: submissions

This table is used to store the student’s homework submissions.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Data Type | Length | Rules | Comments | Example |
| subm\_id | Integer | One digit and above | N/A | Primary key. Generated automatically when the assignment is submitted | 14 |
| assignment\_id | Integer | One digit and above | N/A | Foreign key. Links to the assignments table. | 54 |
| student\_id | Integer | One digit and above | N/A | Foreign key. Links to the main\_acc table. Stores which student submitted the assignment | 21 |
| submission\_date | Date | N/A | Must be in the format yyyy-mm-dd | Stores when the assignment was submitted. | 2025-12-13 |
| mark | Integer | One digit and above | N/A | Stores the mark that the student achieved | 13 |

When a teacher creates an assignment, a table is created into the database that stores all the questions and answers. The structure of that table is as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Data Type | Length | Rules | Comments | Example |
| questionnum | Integer | One digit and above | N/A | Primary key. Generated in a sequential order. | 3 |
| assignment\_id | Integer | One digit and above | N/A | Foreign key. Links with the assignments table. | 4 |
| question | String | More than 5 characters | N/A | Stores the question | “What is the mass of the Sun?” |
| answer | String | More than one character | N/A | Stores the answer | “17 kilograms” |
| marks | Integer | One digit and above | N/A | Stores the marks the question is worth | 2 |
| question\_type | String | N/A | The teacher can select one of two options when creating the assignment (Standard answer/Calculation) | Stores the type of question. The UI when answering the question changes depending on the type. | “Standard answer” |

When a class is created, a table is made that stores all the students that are in that class. It follows this structure:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Data Type | Length | Rules | Comments | Example |
| student\_id | Integer | One digit and above | N/A | Primary key. Generated by the database when the class is created | 23 |
| student\_name | String | More than 5 characters | Must have at least one space | Stores the student’s name | “Laurence Stone” |

{YAP ABOUT THE DATABASE}

Entity Relationship Diagram

A computer screen shot of a diagram

Description automatically generated

SQL Pseudocode

## 2.5 Design of other Parts of the Solution

2.5.1 Log In

### A screenshot of a computer screen Description automatically generated

UI

I am using Tkinter for the UI. {YAP ABOUT THE LOG IN UI}

A screenshot of a computer error

Description automatically generatedA screenshot of a login screen

Description automatically generated

Validation rules

Key Variables/Data Structures /Classes

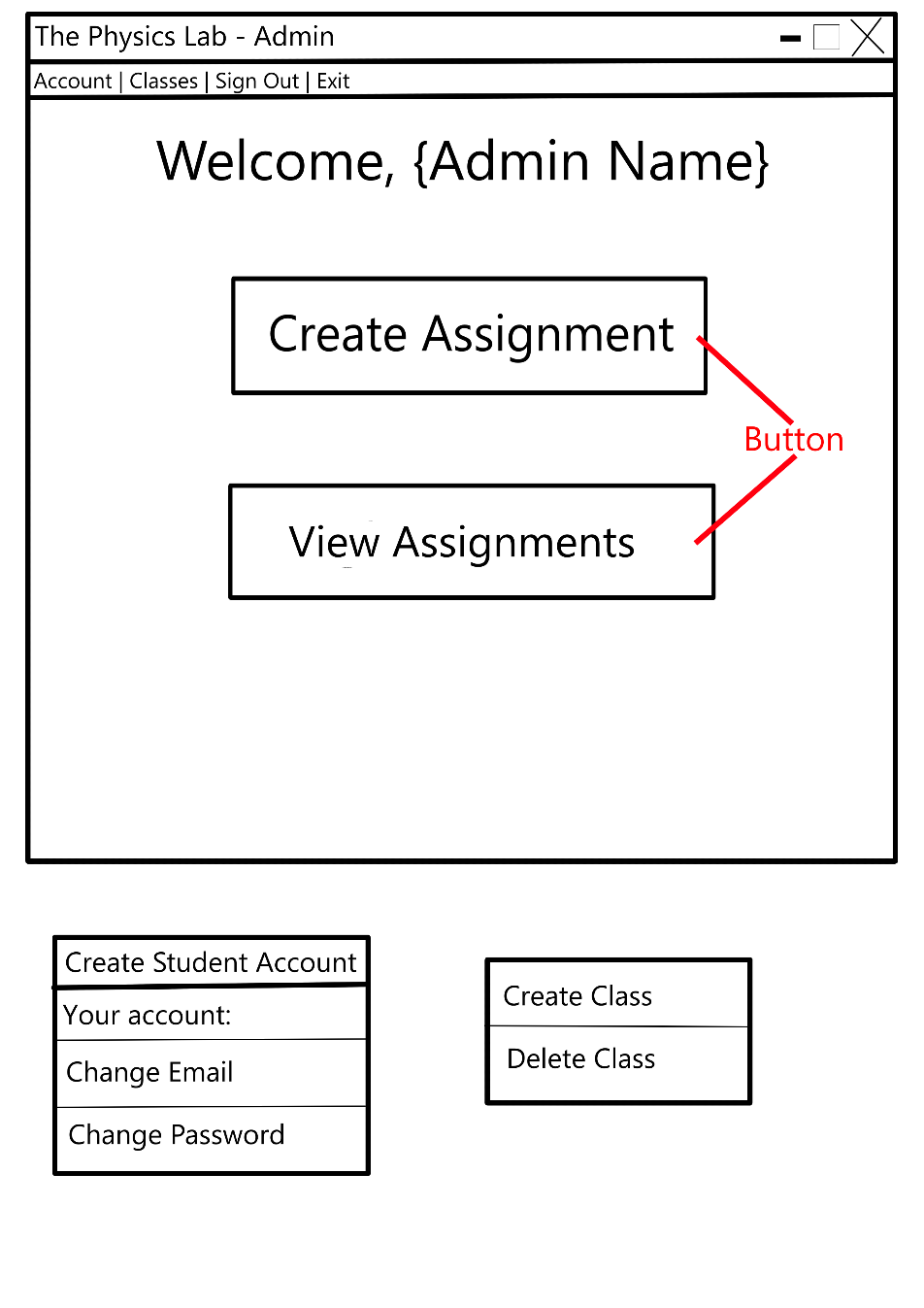
Algorithms and PseudoCode

Test Plan for PART ONE ….

2.5.2 Admin View

Form Design and Layout

UI



Justification of Validation rules

Key Variables/Data Structures /Classes

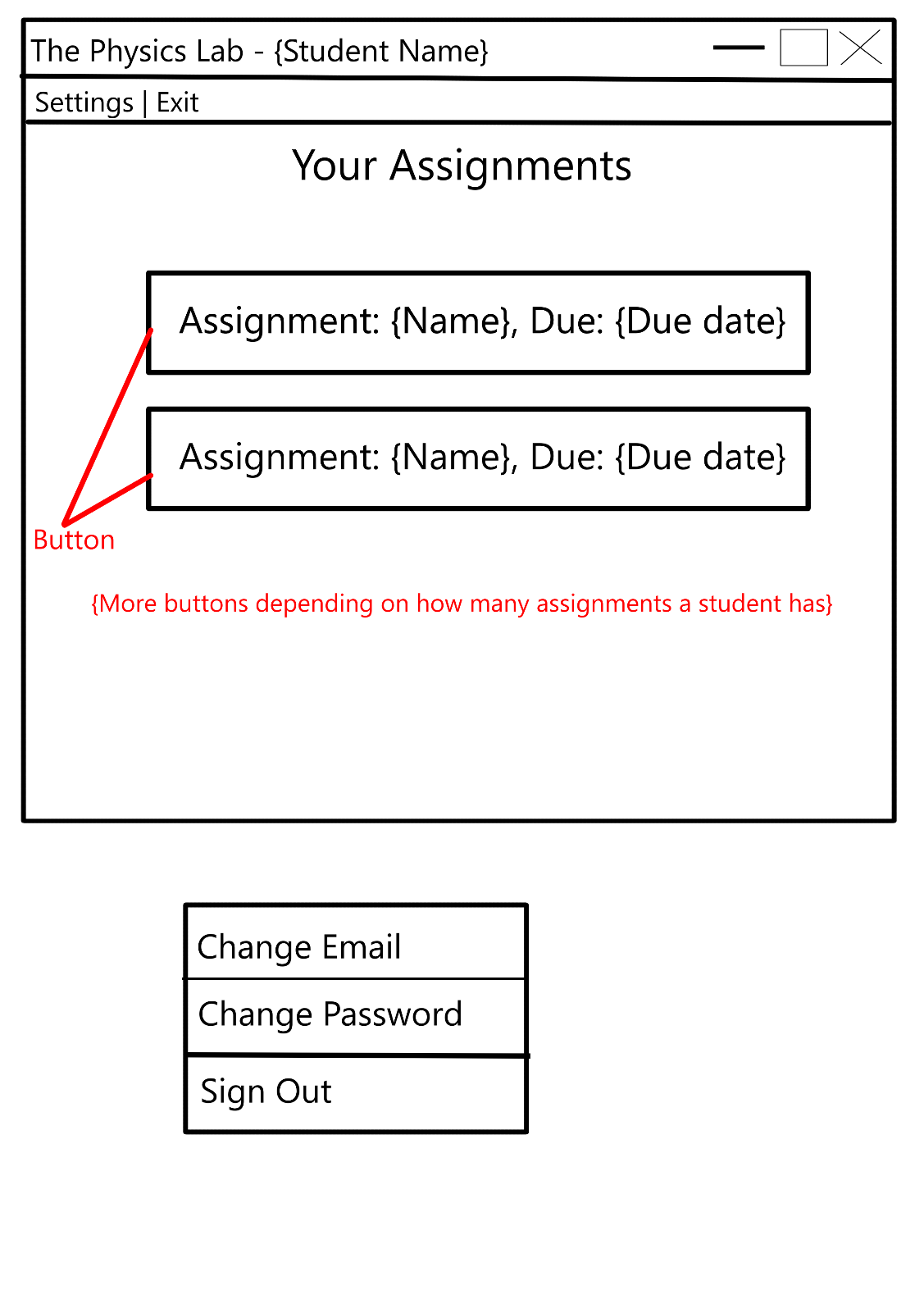
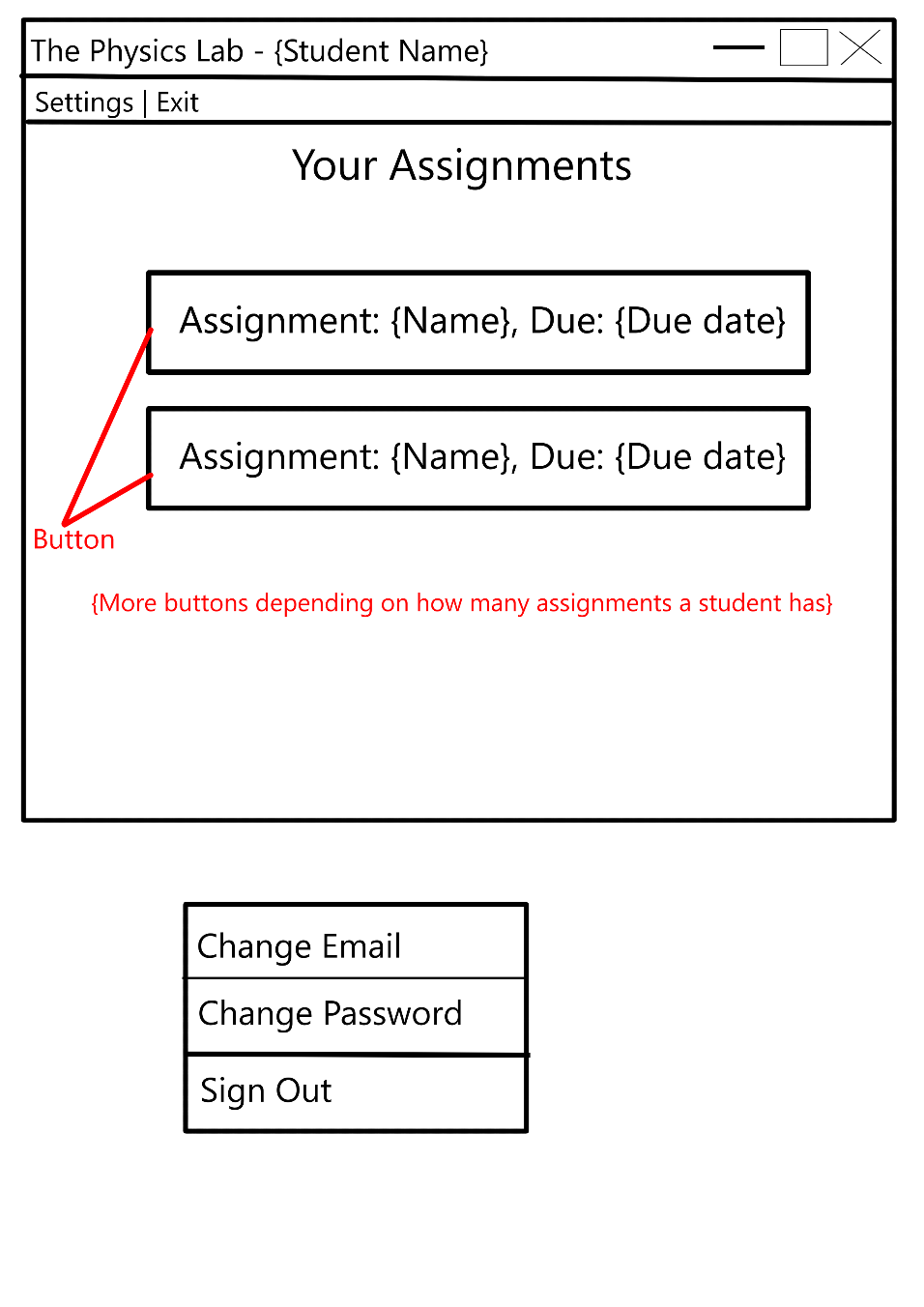
Algorithms and PseudoCode

Test Plan for PART TWO

2.5.3 Student View

Form Design and Layout

UI



Validation rules

Key Variables/Data Structures /Classes

Algorithms and PseudoCode

Test Plan for PART THREE

2.5.4

Form Design and Layout

Validation rules

Algorithms and PseudoCode

Key Variables/Data Structures /Classes

Test Plan for PART FOUR

## 2.6 Stakeholders involvement

Testing plan to inform evaluation

# Chapter Three: Iterative Development and Testing

## 3.1 Introduction

## 3.2 Stage 1: Building the Database

Explain the section of code (e.g. SQLITE3 Code)

Show it (e.g. Screen printouts)

Test it (e.g. test the databases)

Comment on the result

Show changes/correction and further testing

Feedback from Stakeholder (if appropriate)

Review

## 3.2 Stage 2: Building The Log In

Explain the section of code (e.g. SQLITE3 Code)

Show it (e.g. Screen printouts)

logInMenu.py





Test it (e.g. test the databases)

Login window UI test

|  |  |  |  |
| --- | --- | --- | --- |
| Test No. | Test Description | Outcome | Evidence |
| 1.1 | Test that the window opens | The window opens successfully | A computer screen shot of a computer code  AI-generated content may be incorrect.  (GIF 1 in evidence) |
| 1.2 | Show password button toggles the password visibility | Button changes the visibility of the password | A screenshot of a computer  AI-generated content may be incorrect.  (GIF 2 in evidence) |
| 1.3 | Exit button closes the window | Exit button closes the window successfully | A screen shot of a computer program  AI-generated content may be incorrect.  (GIF 3 in evidence) |
| 1.4 | Empty fields produce an error message | Error message created when both fields are empty | A screenshot of a computer  AI-generated content may be incorrect.  (GIF 4 in evidence) |
| 1.5 | Empty password field produces an error message | Error message is created successfully when the password field is empty | A screenshot of a computer  AI-generated content may be incorrect.  (GIF 5 in evidence) |
| 1.6 | Empty username field produces an error message | Error message is created successfully when the username field is empty | A screenshot of a computer  AI-generated content may be incorrect.(GIF 6 in evidence) |
| 1.7 | Wrong login information entered produces an error message | Wrong login information entered produces an error message successfully | A screenshot of a computer  AI-generated content may be incorrect.(GIF 7 in evidence) |
| 1.8 | Correct student information entered creates student view | Student view created successfully when correct student information is entered | A screenshot of a computer  AI-generated content may be incorrect.  (GIF 13 in evidence) |
| 1.9 | Correct admin information entered creates the admin/teacher view | The admin view is created successfully when the correct information is entered | A screenshot of a computer  AI-generated content may be incorrect.  (GIF 14 in evidence) |

Feedback

All of the tests were passed successfully so there is no need for any corrections. I showed the UI to my stakeholders and they were also pleased with the result. However, they brought up the idea of two-factor authentication. {yap more}

Here is the altered logIn.py code:









This code imports the sendEmaiOTP from the module processWindows.py. {yap}



Two-factor authentication UI test

|  |  |  |  |
| --- | --- | --- | --- |
| Test No. | Test Description | Outcome | Evidence |
| 1.1 | 2FA window opens when correct information is entered | 2FA window gets created successfully when the correct login information is entered | A screenshot of a computer  AI-generated content may be incorrect.  (GIF 8 in evidence) |
| 1.2 | Resend code button sends the same code | When the resend button is clicked, the same code is sent to the user | A screenshot of a computer  AI-generated content may be incorrect.  (GIF 9 in evidence) |
| 1.3 | Invalid 2FA code produces an error message | An invalid 2FA code successfully produces an error message | A screenshot of a computer  AI-generated content may be incorrect.  (GIF 10 in evidence) |
| 1.4 | Valid 2FA code logs the user in to the student view | When a student enters a valid 2FA code, they are logged in to the student view | (GIF 11 in evidence) |
| 1.5 | Valid 2FA code opens admin view when entered by an admin/teacher | The admin view opens when an admin/teacher enters a valid 2FA code | A screenshot of a computer  AI-generated content may be incorrect.  (GIF 12 in evidence) |

Comment on the result

Show changes/correction and further testing

Feedback from Stakeholder (if appropriate)

Review

## 3.2 Stage 3: Building The Validation

Explain the section of code (e.g. SQLITE3 Code)

This section of the code will have all the validation I will need for my project.

Show it (e.g. Screen printouts)

isValid.py:



Test it (e.g. test the databases)

To test my validation, I will be using the unittest library. {yap}

isValid\_testing.py:



Email validation and verification testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test No. | Description | Test data | Expected Outcome | Actual Outcome |
| 1.1 | Attempt to validate and verify a normal email | b32908@sfc.potteries.ac.uk | “True True”  “True True” | A computer screen with white text  Description automatically generated |
| 1.2 | Attempt to validate and verify an invalid email | b32908@uk | “False False” “False False” | A screen shot of a computer  Description automatically generated |
| 1.3 | Attempt to validate and verify a valid but not real email | b32908@uk.com | “True True”  “False False” | A screenshot of a computer error  Description automatically generated |

The verification on test 1.3 failed. The expected outcome should be false as the email address doesn’t exist but the email verification returns True. Upon further inspection of my code and the Kickbox API documentation, I discovered that the problem had to do with the API key. When I set up the API key, I configured it incorrectly and it simulated every test that I tried. This meant that all the results I was given were fake. To fix this, I created a new API key with the correct configuration. This change only affects the subroutine verifyEmail().



The tests for the new code are outlined below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test No. | Description | Test data | Expected Outcome | Actual Outcome |
| 2.1 | Attempt to validate and verify a normal email | b32908@sfc.potteries.ac.uk | “True True”  “True True” | A screenshot of a computer  Description automatically generated |
| 2.2 | Attempt to validate and verify an invalid email | b32908@uk | “False False” “False False” | A computer screen with red text  Description automatically generated |
| 2.3 | Attempt to validate and verify a valid but not real email | b32908@uk.com | “True True”  “False False” | A black screen with white text  Description automatically generated |
|  |  |  |  |  |

{change to english}

Tested password. Requirements: 8-20 characters, upper and lowercase characters, symbols and numbers

Password validation testing:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test No. | Description | Test data | Expected Outcome | Actual Outcome |
| 1.1 | Attempt to validate a valid password | passwA!or33345 | “True True” | A screen shot of a computer  Description automatically generated |
| 1.2 | Attempt to validate a password with no uppercase letters | password1! | “False False” |  |
| 1.3 | Attempt to validate a password with exactly 8 characters | Pass1!rd | “True True” | A black screen with white text  Description automatically generated |
| 1.4 | Attempts to validate a password with less than 8 characters | Pass1!r | “False False” | A screen shot of a computer  Description automatically generated |
| 1.5 | Attempts to validate a password with exactly 20 characters | PasswordLengthTest1! | “True True” | A black screen with white text  Description automatically generated |
| 1.6 | Attempts to validate a password with more than 20 characters | PasswordLengthTest1!! | “False False” | A computer screen shot of a test  Description automatically generated |
| 1.7 | Attempt to validate a password with no lowercase letters | PASSWORD1! | “False False” | A screen shot of a computer  Description automatically generated |
| 1.8 | Attempt to validate a password with no symbols | Password1 | “False False” |  |
| 1.9 | Attempt to validate a password with no numbers | Password! | “False False” | A black screen with white text  Description automatically generated |

Comment on the result

Show changes/correction and further testing

Feedback from Stakeholder (if appropriate)

Review

## 3.2 Stage 4: Building The Admin View

Explain the section of code (e.g. SQLITE3 Code)

Show it (e.g. Screen printouts)

Test it (e.g. test the databases)

Comment on the result

Show changes/correction and further testing

Feedback from Stakeholder (if appropriate)

Review

## 3.2 Stage 5: Building The Student View

## 3.3 Final Review, Improvements and Corrective Actions

# Chapter Four: Evaluation

Evaluation

## 4.1 Introduction

## 4.2 Testing to inform evaluation

## 4.3 Evaluation

Evaluating usability features

Evaluating Robustness

Limitations and Maintenance