Contents

[Analysis 1](#_gjdgxs)

[Background to problem 1](#_30j0zll)

[Current Situation 1](#_1fob9te)

[Research 3](#_3znysh7)

[Identification of user needs 6](#_2et92p0)

[Components of this project 6](#_tyjcwt)

[Prototype 1 7](#_3dy6vkm)

[Data flow diagram 8](#_1t3h5sf)

[Design 8](#_4d34og8)

[Core Desirable and Supplementary 8](#_17dp8vu)

[Database design 8](#_3rdcrjn)

[Design 1 9](#_26in1rg)

[Design 2 9](#_lnxbz9)

[Design 3 10](#_35nkun2)

[Design 4 10](#_2jxsxqh)

[Prototype 2 10](#_1ksv4uv)

[References 12](#_44sinio)

# **Analysis**

## Background to problem

Thomas Whitham Sixth Form is a school in the north west that teaches a variety of subjects, including Physics. Simple Harmonic Motion is a topic taught at the end of Year 12, and two of the required practicals for Physics around that topic involve investigating SHM in a pendulum and a spring. Our Physics teacher, Mr C, is looking for new materials to use for teaching SHM to the Year 12s as well as revising the topic with the Year 13s.

However, a recent mock exam has shown that many students lack the essential understanding of the topic necessary for A level.

My aim is to create a program that accurately and realistically simulates the motion of a spring that follows simple harmonic motion, and also displays the relevant information that can be used as the simulation runs to gain a full understanding of how SHM works theoretically.

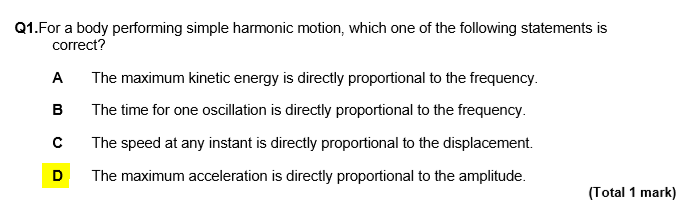
## Current Situation

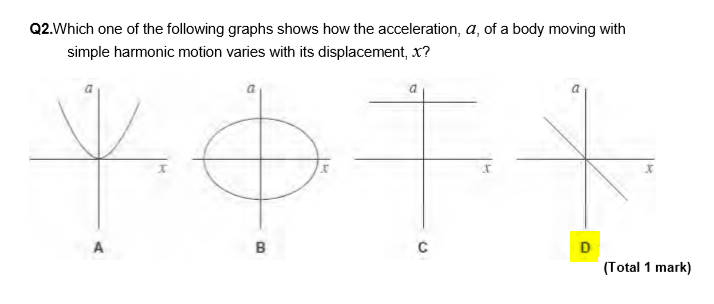
Simulations of the SHM spring already exist on the Internet, but tend to have issues:

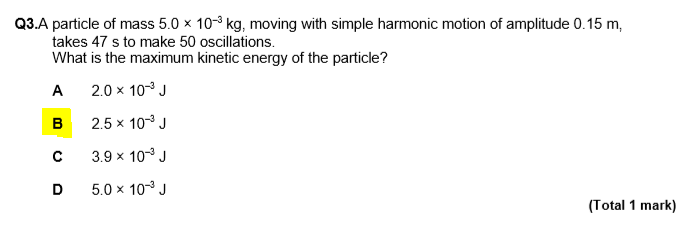
* They tend to be too complex for the level necessary at A level.
* Such simulations rely on having an Internet connection.
* Because they tend to be so packed with extras, simulations can take some time to initially load.

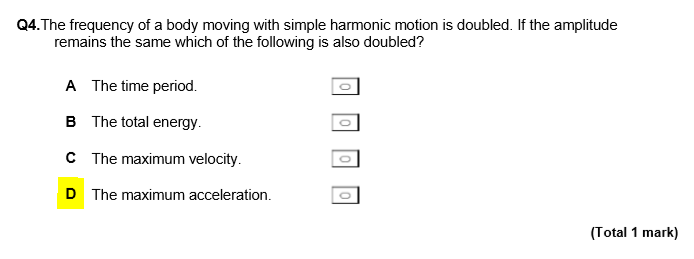
In order to test what knowledge my year 13 class lacks, I asked the 14 students in my class (discounting myself) to complete practice questions on the topic of SMH and complete survey questions about what they would like to see covered by my program.

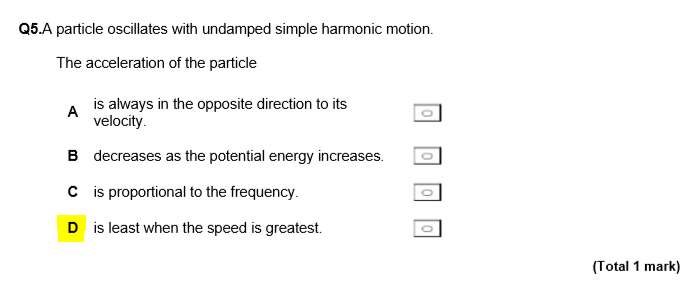
I took the practice questions from Physics and Maths Tutor online as the site is very reliable and frequently used as revision tool by myself and members of my class. In the following five questions, I have highlighted the correct answers.











(Physics and Maths Tutor, 2020)

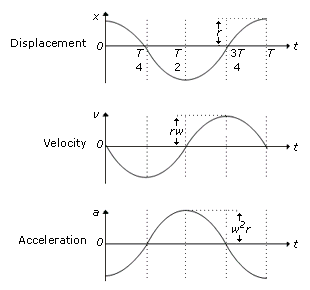
For the 14 others in my class, the students scored an average mark of 60%. The question answered most successfully was question 3, with 11 students answering correctly, as it has more of focus on working with given numerical data and finding kinetic energy, which is not exclusive to the SMH topic. The questions answered worst were questions 1 and 5, with only 6 students answering correctly.

This shows that the fundamental problem with my class’s knowledge is not in calculating data and using the correct formulae, but with the core knowledge associated with the topic.

## Research

Simple Harmonic Motion is a topic covered in A Level Physics at the beginning of Year 13. Most, if not all of the relevant understanding I would need for creating this simulation can be found in the textbooks for A Level Physics.

An object follows simple harmonic motion if it oscillates around an equilibrium position, moving repeatedly one way then in the opposite direction through its equilibrium position. Examples of oscillating objects could be:

* The springs in the suspension of a car

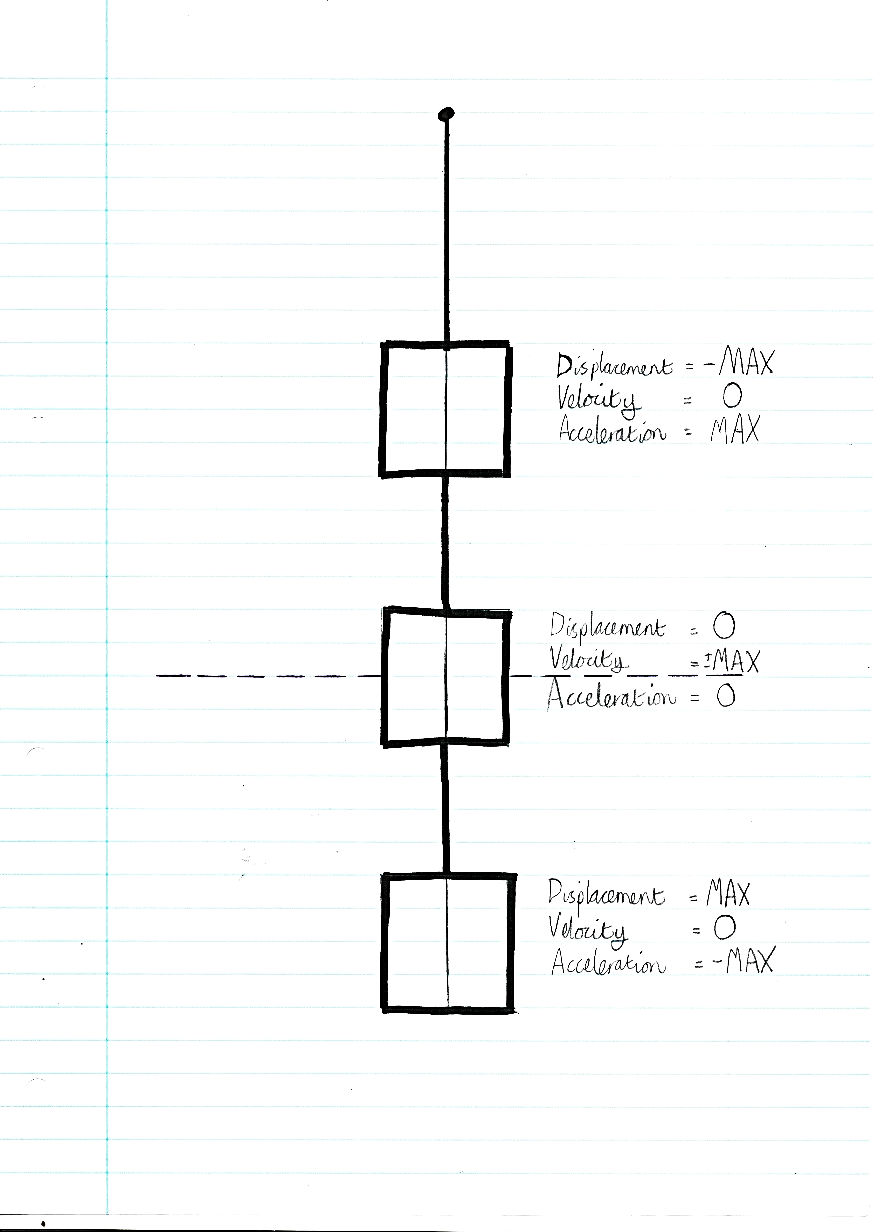
*Figure SEQ Figure \\* ARABIC 1*

* The electronic oscillator that regulates the internal clock of a computer
* A child on a swing.

(Breithaupt, 2008)

Figure 1 shows the three graphs of motion for SHM, showing how displacement, velocity and acceleration of the spring change over the same period of time.

(Oscillations and Wave Motion, 2009)

When the spring is at position 1 and, its displacement is at its maximum. As the weight is changing direction at this point, its velocity is 0, and its acceleration will be at its maximum due to this change in direction.

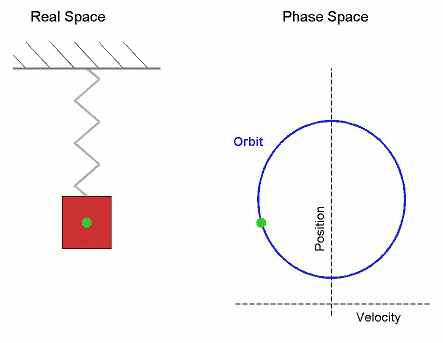
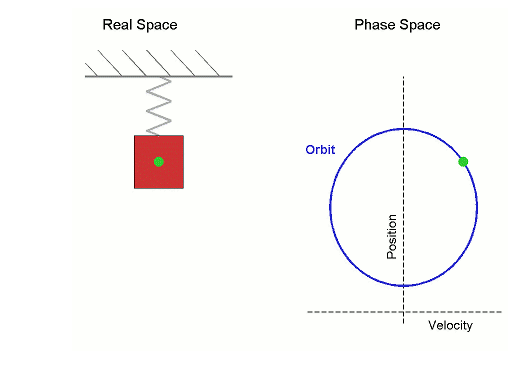
**1**

**2**

**3**

At position 2, the displacement is 0 as the weight is at the equilibrium point. The velocity is at its maximum so the acceleration is at 0 as the velocity is at its highest.

(Breithaupt, 2008)



*Figure SEQ Figure \\* ARABIC 2*

Figure 2 shows how an SMH oscillator can be modelled using circular motion. Simple harmonic motion is very similar to circular motion. If you look at an object going round in a circle side-on, it looks exactly like simple harmonic motion. For this reason angular velocity is one of the variables we use in modelling simple harmonic motion.

(Simple Harmonic Motion A level Physics, 2019)

The motion of a simple harmonic oscillator follows a number of different variables that would need to be accounted for in my simulation.

* **Angular velocity**, ω, measured in radians per. second. A SHM oscillator follows a pattern of movement very close to circular motion, angular velocity is used to help calculate the position of the oscillator.

For a mass on a spring…

Where:

k is the spring constant in meters,

m is the mass in kilogrammes.

Where:

T is the time period in seconds of the motion,

2is a full circle.

* **Displacement**, x, measured in meters. Displacement is the vertical or horizontal distance from the equilibrium point.

Where:

A is the Amplitude in meters, or the max. Displacement,

ω is the angular velocity,

t is time in seconds.

* **Time Period**, T, measured in seconds. The time period is the time taken for the for the oscillator to complete one full cycle.

Where:

2π is the full cycle in radians (constant)

k is the spring constant

m is the mass of the weight on the spring

(Simple Harmonic Motion A level Physics, 2019)

## Identification of user needs

For my end user, my goal is to create a SHM spring that:

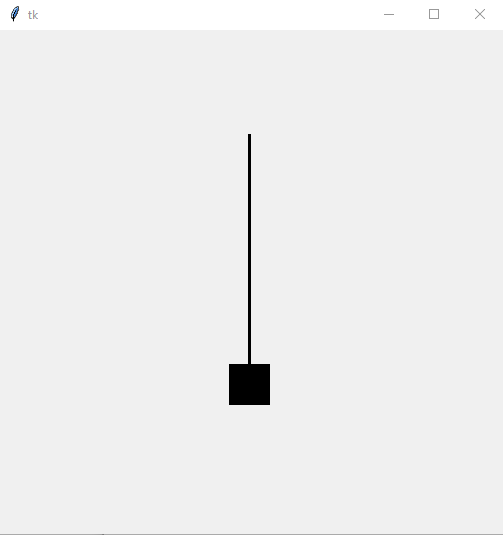
* Covers all the information necessary for A level Physics and displays it in an accessible way
* Can be accessed offline (so the program could be downloaded and run on school computers)
* Is visually appealing and engaging to look at and use.
* Has a prebuilt and modifiable database of custom setups for the spring.

## Components of this project

* **Tkinter** - I plan on using the built-in python tkinter module to model the motion of the spring. As the appearance of the spring shouldn’t be particularly complex, the simplistic style of tkinter should be sufficient for modelling the motion of the spring. I will have the spring coloured red when the spring has negative displacement, and blue when positive.
* **Input values** - The user will be able to input:
  + **Initial displacement/Amplitude** in meters *(distance from the origin point which the weight will gravitate towards)*
  + **Mass of the weight** in kilogrammes
  + **Spring constant** in meters*(the original length of the spring when at rest and with no added weight on it)*
* **Output values** – The user will have displayed for them certain output values as the program runs to gain an understanding of how the movement of the spring can be modelled. These values would be:
  + **Displacement** in meters *(distance and direction from the origin point)*
  + **Velocity** in meters per second *(speed and direction the weight is travelling in)*
  + **Acceleration** in meters per second^2*(rate of change of velocity as the weight travels)*
  + **Period** in seconds *(time taken for one full cycle to occur)*

(Oscillation, n.d.)

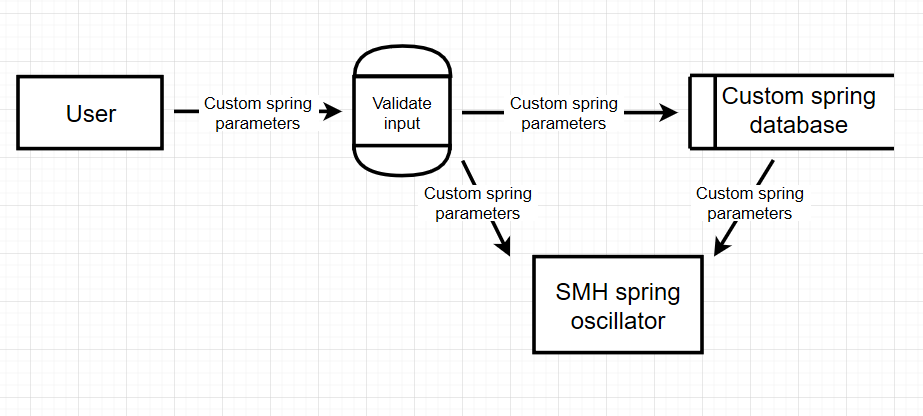
## Prototype 1https://lh3.googleusercontent.com/HwczQeEWA7579lPLaCgkrx_DL5tEaSdkkHeqfg1pYDvLbf8CPo-GBAxz3Smuglx3Zute5M0iQbf0ZsIT1UwjE8Zp66fvyvsn7SVVtGZZqtyOLLtaRkzUWpWiGF5JG4VkLU02X_1l



This is the first prototype for my spring. It follows an extremely basic movement. Whilst the weight does move up and down as I want it to, this prototype has many issues:

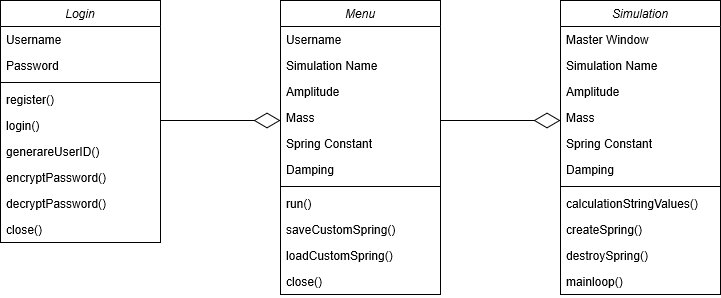
* The string moves with the weight and does not change length or have a fixed top point
* The weight does not change colour when it passes across the line of equilibrium
* The program has no input values so the user cannot customise the movement of the spring
* The program does not output the values for velocity, acceleration or displacement, as the program doesn’t actually calculate them.

## Data flow diagram



# **Design**

## Class Diagram



In my diagram, I have three tables to show the three main classes I will use in my project.

The **Login** class accepts the user’s username and password.

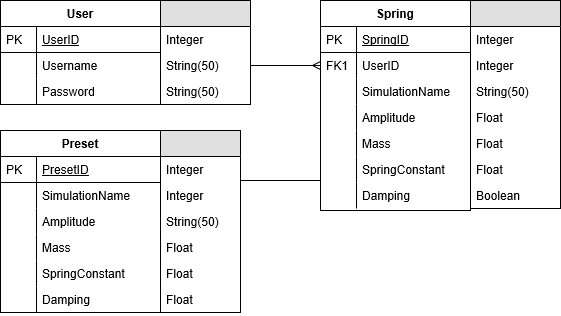
* The function close() closes the program without saving anything input by the user.
* The register() function checks that the input username is not already in the Login database. If it isn’t, the generateUserID() function assigns the user an ID, encryptPassword() encrypts the user’s password for storage and saves the UserID, the username and the encrypted password to the database. If the username the user attempts to register is already in the database, the function flags up an error message and stops the username from being registered.
* The login() function checks that the input username is already in the database. If it is, the function checks that the user’s password is correct using the decryptPassword() function, and if it is, the user successfully logs into the system. If the input username isn’t in the Login database, the function flags up an error message and asks the user to input a valid username or register a new account.

The **Menu** class inherits the username of the user as the username is needed to access the user’s custom springs.

* The function close() closes the program without saving anything input by the user.
* The function run() takes whatever values have been input or loaded into the Menu class and passes them onto the Simulation class
* The function saveCustomSpring() calls the values input by the user. One user can have a maximum of 8 values saved to the Spring database at one time, including, initially, the preset spring value. If the user has fewer than 8 slots in use in the database, the input data is saved in the next available slot. If the user has all 8 slots in use, every value saved for that user is shifted in step back in their list, so the first value in the list is erased, and the input data is saved in the final slot. After the input data is saved, the user input in the window is cleared.
* The function loadCustomSpring() is called when the user selects a value from the database via the drop down list. When this happens, the data is called into the program to be run by the user.

The **Simulation** class inherits the spring input values from the Menu class. The username, however, is not inherited

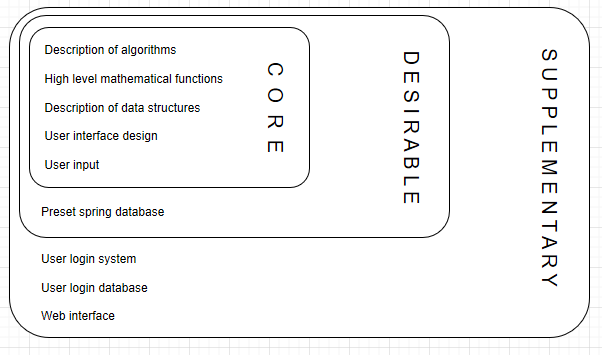
## Database design



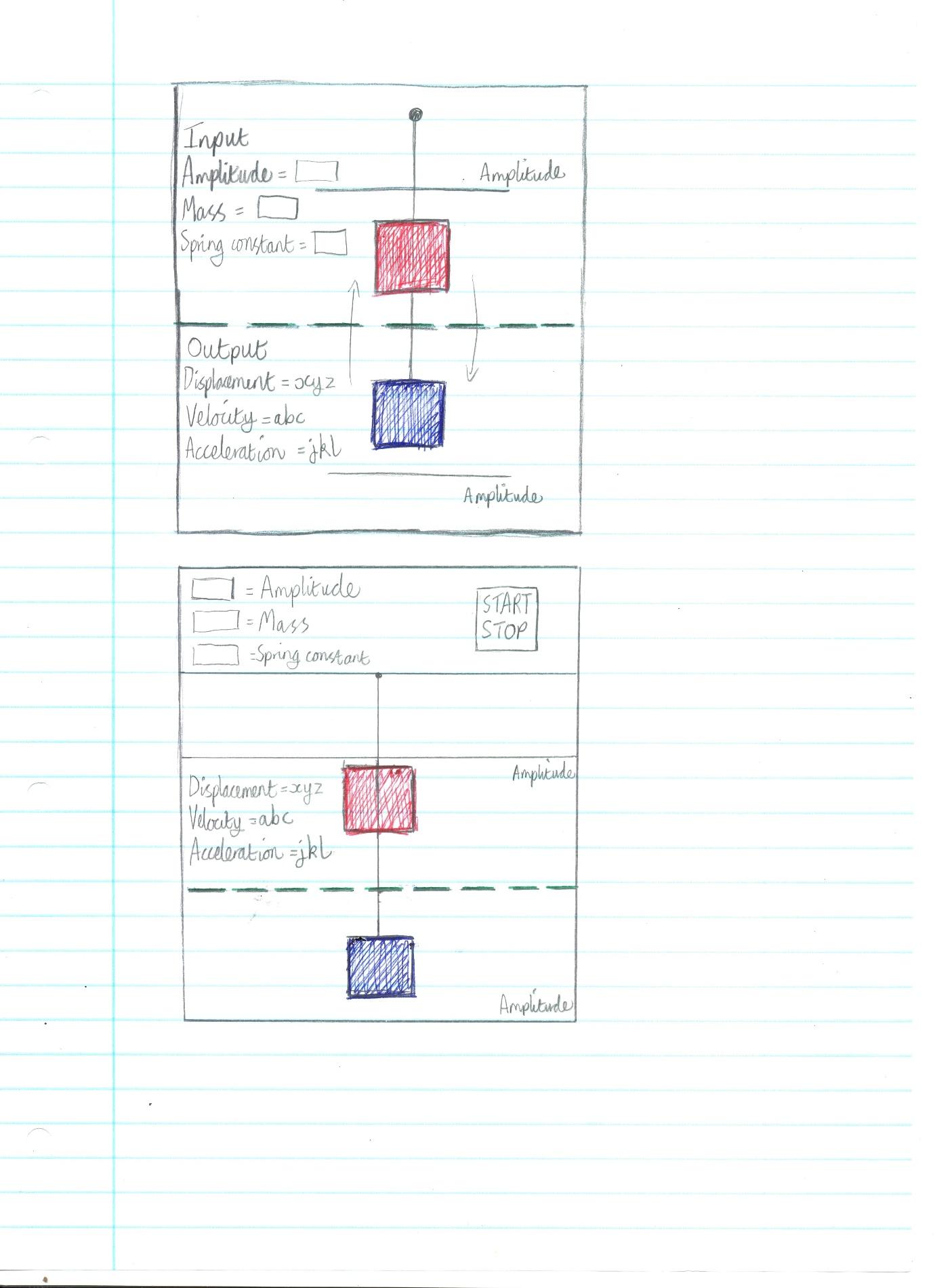
|  |  |
| --- | --- |
| **Column Name** | **Value** |
| ID (Primary Key) | Integer |
| Simulation Name | String |
| Amplitude | Float |
| Mass | Float |
| Spring Constant | Float |

ID (Primary Key) is generated by hashing the string given for Simulation Name.

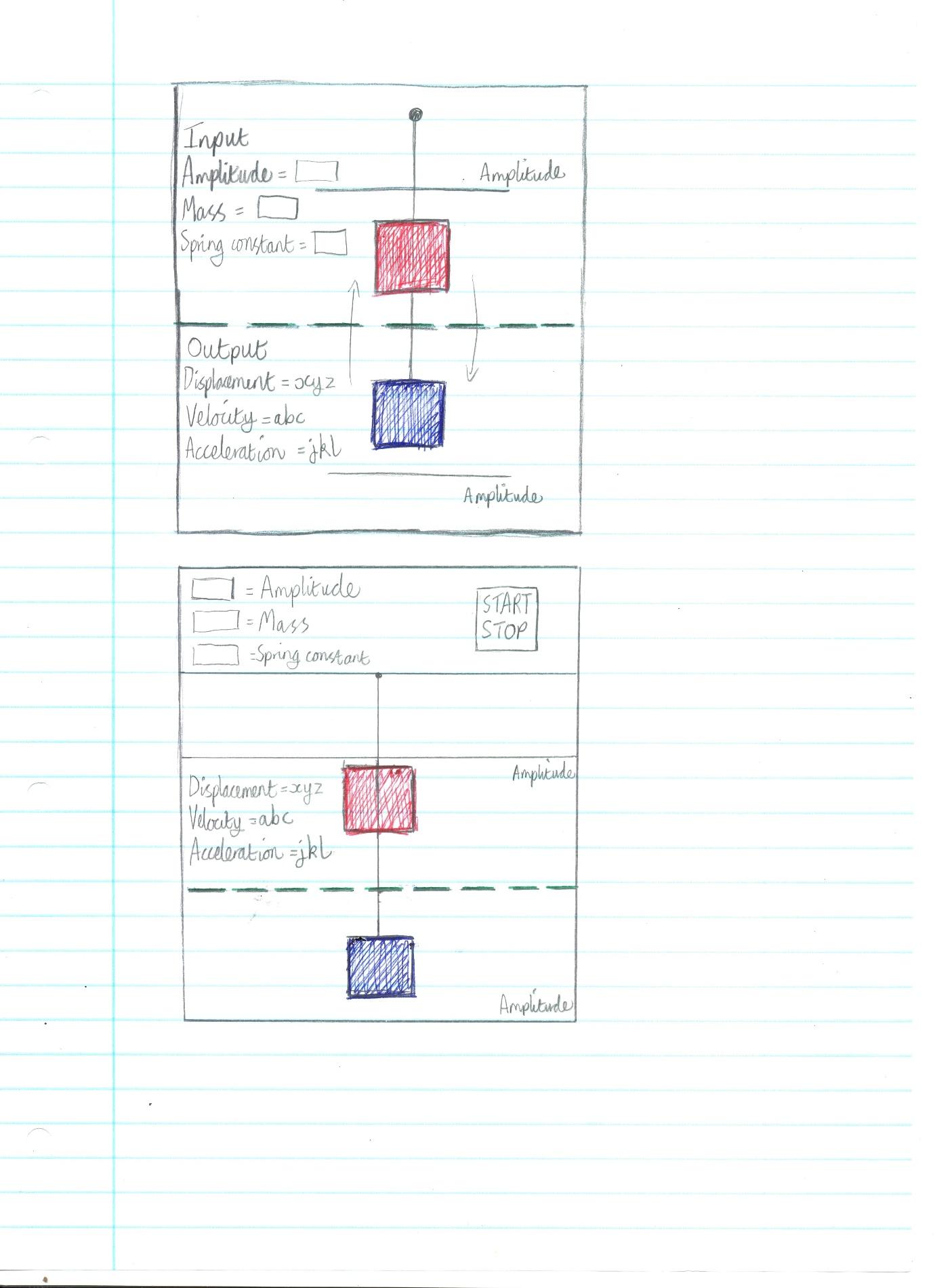
## Core Desirable and Supplementary



## Design 1



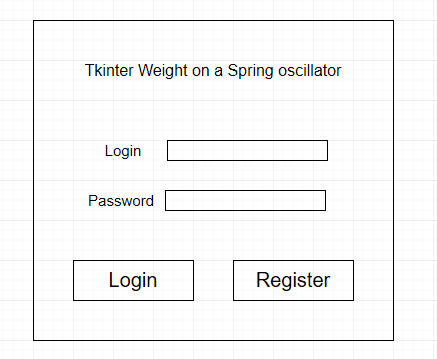
## Design 2



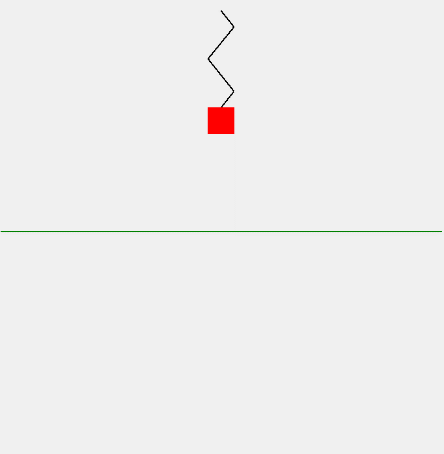
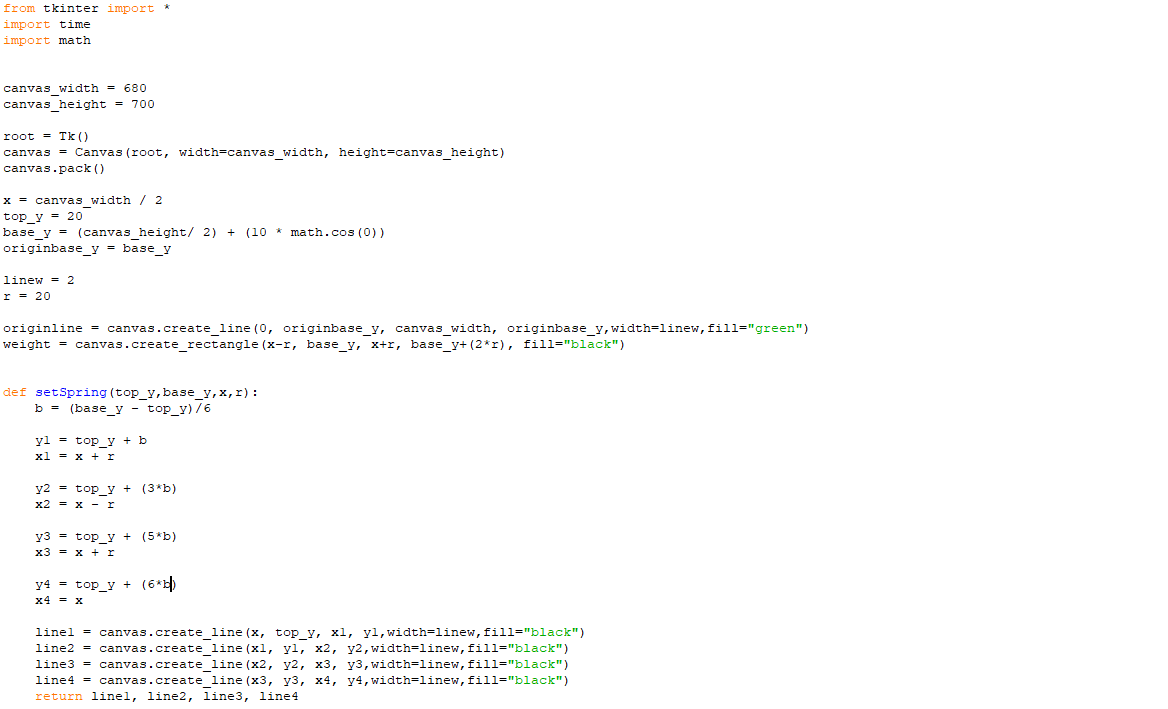
## Design 3H:\computing\COURSEWORK\scan0007.jpgH:\computing\COURSEWORK\scan0007.jpg

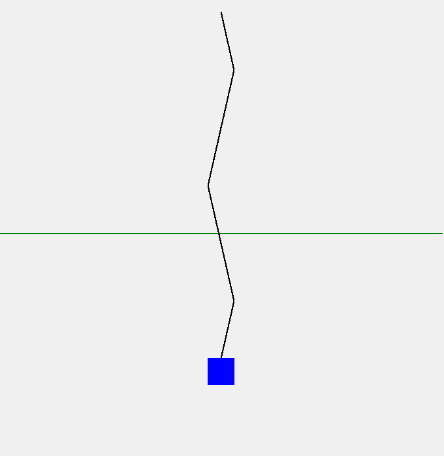
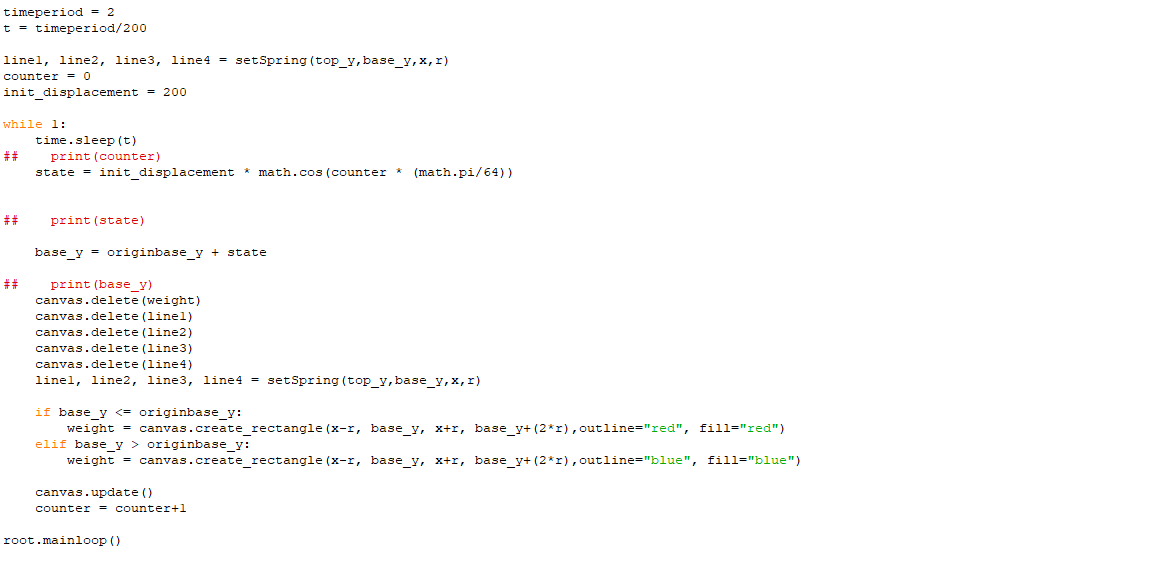
“Amplitude”, “Mass” and “Spring Constant” use sliders set between preset values. This is to keep the simulation within the limits of the model.

“Preset values” references a database and calls values for amplitude, mass and spring constant



## Prototype 2





# References

Breithaupt, J. (2008). *AQA Physics A2.* Cheltenham: Nelson Thornes.

*Oscillation*. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Oscillation

*Oscillations and Wave Motion*. (2009). Retrieved from Penguin Physics: https://penguinphysic.wordpress.com/tag/wave-motion/

Reeves, B. (2015). Computer Science. London: Hodder Education.

*Simple Harmonic Motion A level Physics*. (2019). Retrieved from Wikibooks: https://en.wikibooks.org/wiki/A-level\_Physics\_(Advancing\_Physics)/Simple\_Harmonic\_Motion