## **Objectives**

From my analysis and research into the problem, I have a good idea about what my client and potential users want, as well as the limitations of current solutions and the improvements they need.

1. Include a total of 3 simulations – Determination of Young’s Modulus, Stationary Waves on a String and Interference effects of a double slit.
   1. The application should solely make use of the PyGame library, as it allows for graphics rendering
   2. Simulations need to be interactive and have parameters that can be changed.
   3. There needs to be additional information about the simulation.
2. The simulation should be as close to the practical setup that is being used at ICHS, which I have found by talking to my client.
   1. The graphics of my simulation will be very minimalistic to make it easy to understand the apparatus that is being used.
   2. To aid in being able to set up the practical properly, the user will be able to view the name of the apparatus by hovering over them.
3. The simulation needs to be accurate to how it would work in the real world.
   1. Relevant physics equations, such as the harmonic motion of a wave, young’s modulus equation, wave speed equation etc. will be used in order to make a realistic simulation
4. The user should be able to change different parameters of the simulations and the simulation should react to those changes.
   1. There will be a fixed range of values that the parameters can be – these values are continuous and can be changed using a slider.
   2. For the Stationary Wave on a String practical the parameters that can be changed are: Length of string, Tension, Frequency of the vibrator and the Linear Density of the string.
   3. For the Interference Effect of a Double Slit the parameters are: Wavelength of the monochromatic light source, distance between double slit and screen, slit width and the slit separation.
   4. For Determining Young’s Modulus, the parameters are: Tension and diameter of the wire.
5. There will be a pause/play button for the simulation that the user can click on to see the simulation at different points of the practical.
   1. The button will be placed on the top left, and will stop the simulation at the point in time when it was clicked.
6. The user should be able to navigate between simulations easily.
   1. I plan to link the simulations to each other using a circular doubly-linked list structure, which includes two-way connections so that the user can traverse between simulations. As a result, the user can easily go to the next or previous simulation from the current one.
   2. All buttons should have an indication of being clicked – i.e. change its colour or change in size.
7. Include a section with additional information about the practical.
   1. The information must be concise and relevant to the practical.
   2. Needs to contain information about the theory behind the practical, how to improve it, how to obtain data accurately and health and safety notes.
8. Graphs shown in the additional information section will be changing in real-time according to the simulation
   1. User should be able to add readings to the graph based on what the simulation is showing and clear the graph to add new readings.
   2. The section will have target results that the user will try to meet – this will be done my comparing the data on the graph to the targets.
9. The section which includes additional information needs to only be accessed when needed by the user.
   1. This will be done by making a collapsible tab on the side of the screen, that will be triggered by clicking on the related button. This way the user won’t have a section of text distracting them from the simulation the entire time.
   2. Buttons should take up a small part of the screen and the pop-out window should be slim in size so as not to cover the majority of the screen.
   3. Depending on the amount of text, the collapsible window will be scrollable so that more text can fit into the same space.
10. The program should be free so that it can be used by the teachers in school and by the students to help them learn.
    1. This will be ensured by using non-copyright and free graphics for my simulation.
11. The program needs to be intuitive and easy to use
    1. The buttons will have icons to illustrate what they do.
    2. There will be a tutorial overlay when the program is opened which gives a brief rundown of all the features available