The project is divided between the 3 members in the group, making sure that no one takes up too much work and also that everyone is given what they specialise in.

All 3 of the group members were responsible in the first week for researching and about the g-2 and finding exactly what needed to be coded. This also extended into the second week which included finding the appropriate equations needed to simulate this experiment.

On week 2 each member had their individual role and targets of what needed to be done.

For instance Lorna Waring was given the task of writing the introduction for the group report which include exactly how the g-2 works and also the reason why the experiment was created and what they hope to achieve. She also responsible for the code for the circle fitting of the electron after it has decayed which will be added to Zac’s code. She was also responsible for writing the abstract for this progress report.

Luke Jones was given the task of writing and testing the main prototype code in Java, which included so far writing a program for a muon going round the accelerator and also for the decayed electron. This code was made using scalers and circular coordinates and also code which found the end result and worked backwards. This code can be used the track the muon when it decays at its lifetime and then used to track the circular projection of the electron that came from the muon. The code also includes a class method for the 3 detectors which are present in the g-2 experiment which are represented by dots which can be given an error of 1cm to represent the detectors. Since this is a prototype it uses many shortcuts that the final code that Zac is making won’t include.

Zac Humphreys was set with the task of laying the foundational code for the main project, creating a few tools and visualisations for the project. This included creating four new data-types: *Vector, Rotation, Transform,* and *FourVector* and a set of tools that take either Vectors or double Arrays and either Add, Subtract, Scale, Dot product, or Cross Product the inputs depending on what method is being used which saves time having to code those maths functions individually and can be called as and when needed.

Zac Humphreys was also experimenting with C++ testing to see if he could model the same program and test to see if it ran faster in a lower-level language.   
He will continue building the infrastructure and working on the final code for the first iteration of the project whilst the prototype is still being completed.

The result will hopefully simulate the storage ring, the decay to positrons, and the positrons circular path through a set of detectors that then have their circular trajectory fitted with a circle-fit to determine where the decay of the muon was. Extensions could include variable momentum and position of the incoming muons, and a 3-body distribution in momentum of the positron decayed from the Muon.