**PART A**

Planning  
  
Find our designated group and gather together.  
Assign roles and split up into groups of 2 to take on specific roles, according to our skills.  
Gather a detailed understanding of the task.  
Interview the client.  
Research the difference between functional and non-functional requirements, then into moscow.  
Risk analysis, calculating the likelihood and seriousness.  
Research the different software for a gantt chart and which is the best one.  
Design an estimated Gantt Chart.  
Set-up dates for particular milestones/deliverables.

Task1

Set up github project

Conduct research into use case diagrams, this includes what each attribute must contain e.g name, role.   
Conduct research into the project specification and what is required from the group to ensure max marks.  
UML Diagrams.  
Use Cases.  
Refer back to specification again.  
Create a prototype and show it to the client to receive feedback, to further improve before the deadline.  
Create the actual system through programming language of your choice (say we used python because we felt all members of the group were more experienced in this, due to previous modules.)  
Debugging the program, to make sure that its full proof.  
Testing the entire system for errors and against the requirements and specification.  
Finalise and produce all documentation.

**USE CASE MODEL**

1. Include use case diagram and relationships between them – identify actors, professor etc.
2. For each use case identify the basic flow of events, each group member do one use case each
3. Because we had a team of 6 we needed a detailed specification of two use cases – one that included assigning new students to tutors and one that displayed a list of tutees for a particular tutor.

**USE CASE SPECIFICATION**

1. Tested to ensure the template (background looks the same throughout)
2. Interactions between actors and other attributes are clear.
3. Where the flows start and end must be clear.
4. The flow is an advantage for at least one of the actors.
5. Alternatives flows are justified and described and identified clearly.
6. Pre and post conditions are clearly described.

Task2

Classes

Clear class names.  
Reasonable class attributes.  
appropriate methods for classes.

Relationships

Diff types of class relationships – 1. Associations. 2. Aggregation. 3. Composition. 4. Generalisation.  
Multiplication of relationships and clearly identified.

The class diagram appears to sufficiently support the major requirements in use cases.

Produce a first draft use case diagram for hand in on Feb 10th

Do same stuff for the Class diagram before completing a first draft for the hand in on Feb 10th