Marking Guide

Part A – Project Management (worth 25% of total course mark)

There are 3 key components to this:

1. Project Plan and Gantt Chart (40 marks)
2. Software Design Document (50 marks)
3. Software Version Control (10 marks)

Marked out of 100, then times by 0.25 to get total mark. Unless otherwise specified, all group members should receive same mark.

Project Plan and Gantt Chart (40 marks) W4 Fri

In the **Project Plan.docx**

1. Introduction (5 marks)

Should contain an overview of the project (from a project management/component perspective) and mention the scope and outline of the project management document

2. Work Breakdown Structure (WBS) (10 marks)

Should be a breakdown of all the different activities involved in completing the project. For Part A this should contain all of the work involved including preparing the project plan and software design document, as well as all related preparatory/organisation work. For Part B this should include all of the required implementation, testing and reporting activities. This can be somewhat high level for Part B, but should still contain some reasonable assumptions. This should be presented as some form of diagram/hierarchy.

3. Activity Definition & Estimation (10 marks)

For each item in the WBS, the item should be explained in detail and include a time estimate that is reasonable.

4. Gantt chart (15 marks)

All of the items in the Activity definition should be listed in the Gantt chart with the relevant estimates and scheduling. The students should have also tracked the actual start time and time taken. Also need to submit the **Gantt chart.xlsx**

Software Design Document (50 marks)

In the **Software Design Document.docx**

1. System Vision (10 marks)

Should include a background on the dataset, software overview and potential benefits of the software.

2. Requirements (10 marks)

There are 2 types of requirements to consider:

- User Requirements: How a user will interact with the program. What do they need to do ?
- Software Requirements: What functionality will the software provide (think functional requirements)
- Use Cases: These Use Cases should show the blending of user and software requirements by identifying use cases and how the user will interact with the product. Any diagramming format is acceptable, but the diagrams should clearly display the sequence of events and interactions between the user and the software. Expecting about ~5 use cases (1 for each of the functions) and a few accompanying Use Case Diagrams.

3. Software Components and Software Design (15 marks)

- Software Design: Flow chart / block diagram (5 marks)
- Software Components: Functions, Classes/Data Structures, Algorithms (10 marks)

There should be a listing of the main functions (I would expect at least a loadData function and some kind of display function), the main classes/data structures used and potentially a description of any algorithms that might be used for data analysis. Each component should have the relevant information (name, type, details).

4. User Interface Design (15 marks)

Structural design should almost be like a flowchart/hierarchy chart showing the *structure* of the interface. What screens/components and how do they interact.

Wireframes/mock ups of the interface. Should have clearly labelled interface components. No hand drawing (must be digital design). Different screens/menus/options should have their own wireframes. No colour/graphics required – just position/size of components and component layout.

Additional design information can be included here too.

Software Version Control (10 marks)

Git_log.txt : A Git repository is correctly used for all contributions to the project. A Git log is attached and shows regular commits and pushes from all group members.