**Analysis (10) Done Feedback/ Improvements**

|  |  |  |
| --- | --- | --- |
| Identified stakeholders (audience) and described them, explaining how they’ll use the solution and why it fits needs |  |  |
| Described the features of the problem that make a computational approach suitable |  |  |
| Researched **in depth** existing solutions to similar problems & identified your approach based on these |  |  |
| Identified a list of user requirements, explaining your choices |  |  |
| Identified & explained within the user reqs what the **essential** features of your solution are |  |  |
| Identified the hardware and software requirements of the solution |  |  |
| Identified & explained and limitations of the proposed solution |  |  |
| Identified & justified measureable success criteria |  |  |

**Design (15)**  **Done Feedback/ Improvements**

|  |  |  |
| --- | --- | --- |
| Produced a **structure diagram / class diagram** to show the modular nature of the solution |  |  |
| **Development Plan** –   * shown how the problem is broken down into smaller problems & j**ustified** the processes/ steps you have chosen * Identified the **test data** to be used during the development phase * Identified further data to be used in post-development BETA testing to make sure user reqs met and it is robust |  |  |
| **Diagrams** - produced a set of appropriate diagrams to show the structure of the solution e.g.  UML diagrams (Class, object, activity, communication) |  |  |
| **Algorithms** - Produced appropriate and accurate algorithms (pseudo-code and/ or flowcharts) to **fully** describe the solution   * Describe each sub-problem/ sub-routine * Show how all algorithms fit together (system flowchart) * Show how the algorithms have been **tested** using trace tables/ dry runs |  |  |
| **Interfaces** –   * Explain and justify the design of and user interfaces * Described & **justified** the **usability** features to be included |  |  |
| **Data Dictionary** –   * Identified and **justified** the key variables, data structures, classes (where appropriate) * Explained any necessary validation |  |  |

**Development (25)** **Done Feedback/ Improvements**

|  |  |  |
| --- | --- | --- |
| **Iterations –** provide evidence of following each stage of your development plan, showing how each stage was coded and tested - **explain what you did and why**. |  |  |
| **Prototypes** – save prototypes made at each stage that show well commented code.  Show test tables and x-ref to vids & screenshots. |  |  |
| **Modular code**  They are looking for:   * Well commented code * Meaningful variable names * Efficient use of sub-routines |  |  |
| **Validation**  You need to show where the validation is in the program and where you have tested it to prove it works as expected (valid/ invalid/ extreme) |  |  |
| **Reviews**  Have you summarised at the end of each section and explained how it was tested?  Have you explained and changes/ updates required and any modifications to the design that have occurred due to testing? |  |  |

**Evaluation (20** [5 + 15]**)** **Done Feedback/ Improvements**

|  |  |  |
| --- | --- | --- |
| **Post development testing BETA**  Show you have tested:   * the solution as a whole * that the system works as designed * that it is robust and will not break easily  (Prove you have tried to break it!)   Cross-ref all of the tests to the success criteria/ user reqs to show how well the solution meets them. |  |  |
| **Usability testing**  Show how you tested each feature identified in the Design phase to make sure they meet stakeholder needs |  |  |
| **Evaluation**  How well does the solution match the reqs?  What changes were made to the design during development?  Which criteria were **unmet/ partially met** and how could these be achieved in future development?  What additional features might be added and how might you implement those (overview)? |  |  |
| **Maintenance**  What maintenance might the program need/ what limitations are there in the current solution?  How might the program be modified to meet any additional requirements or changing requirements?  Have you included any maintenance features in the program that could be used? |  |  |