## Grade-based Assessment Scheme

Each assessment element is awarded a grade according to the scale given below. The final grade is determined by how well the criteria have been met overall, and not the sum of the individual aspects of the work.

**Part 1 – Software requirements specification**

| **Class/**  **Grade**  **Assessment Criteria** | **Exceptional First**  (*includes all features of First Class with the additional characteristics listed below)* | **First**  **Low | Mid | High** | **Upper Second**  **Low | Mid | High** | **Lower Second**  **Low | Mid | High** | **Third**  **Low | Mid |High** | **Marginal Fail** | **Fail**  **Low | Mid** | **%** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Introduction | As for First plus:  Evidence of a user interview | As for Upper Second plus;  Significant consideration of each aspect of the problem is evidenced, and discussed  The Users’ technical expertise has been discussed.  All assumptions have been stated, and brief details on how the document would have to change if these were to change has been discussed.  Scope is bulletproof, there is no ambiguity in what is in scope. Constraints discussed in good detail. | The problem the system is to solve is explained is understood in detail.  The purpose of the document is outlined clearly, and consistent with the rest of the document.  The users of the system have been described and their needs discussed.  All major assumptions made in the rest of the document have been stated  The scope is well defined, and realistic for the project. Relevant hardware and software constraints identified and described.  The glossary serves as a useful aid in reading the rest of the document | The problem is analysed, but few aspects described in detail.  The purpose of the document is outlined, but maybe inconsistent with the rest of the document in some areas.  The main needs of the main stakeholder have been discussed.  Some major assumptions made in the rest of the document have been stated.  The scope is generally well defined but may be partially ambiguous or somewhat unrealistic | The brief has been understood, but not explained in any significant detail.  The purpose of the document is stated but may have been contradicted later  Users have been identified, but not described in much detail and their needs not discussed  Multiple assumptions that have been made throughout the rest of document are not explicitly stated in this section  Only very few obvious constraints have been mentioned.  Glossary is present, but terms may have been used in the rest of the document that are not present here. | The problem domain is not fully understood or explained.  The purpose of the document is stated incorrectly.  The main stakeholder has been identified but their characteristics have not been discussed.  Only very few assumptions stated when more have been made.  Few or no constraints have been mentioned, and scope is very ambiguous  Glossary not present | No discussion of problem at all.  Purpose of the document is not stated  No assumptions listed  No constraints mentioned and the scope is also not described. | 20% |
| Requirements  (Functional and Non-functional) | As for First plus:  Requirements are of a high professional standard (meets the characteristics of individual requirements stated in IEEE 29148-2018 5.2.5 | As Upper Second plus:  No functional aspect of the system is left ambiguous, and all the requirements fit the quality attribute described in upper second.  The requirements refer to the needs of the users outlined in the introduction.  Non-functional requirements addressing a range of system characteristics have been included and organised appropriately.  No requirement has been identified incorrectly | The requirements are detailed and organised in a way that is clear to read, and appropriate for the project. The requirements should also be prioritised using MoSCoW or similar.  Only very few functional requirements have been identified incorrectly and vice versa.  Full set of requirements covering the main stakeholders and their needs.  The majority of the requirements should be:  •Correct  •Unambiguous  •Complete  •Consistent  •Ranked for importance  •Verifiable  •Modifiable  •Not specify a particular design  •Traceable  Requirements are present that directly address user characteristics. | Requirements only concentrate on main stakeholder and may miss some key requirements needed to meet the goals of the system.  Some of the requirements may not meet the quality attributes outlined in upper second.  The requirements have been organised in a way that is not appropriate for the project/system.  More than a few functional requirements have been mislabelled as functional requirements, and vice versa. | List of requirements has some substantial holes in terms of coverage of usage and stakeholders. Requirements statements may be somewhat incoherent or too general.  Few requirements meet the quality attributes outlined in upper second.  No clear attempt to organise the requirements have been made.  Requirements list leaves the reader in some doubt as to whether it can satisfy the users’ needs. | Requirements not clearly differentiated into functional and non-functional  Requirements clearly will not satifisfy the stakeholders needs. | Very limited list of requirements that meet the quality attributes.  Requirements not organised or prioritised. | 25% |
| Interfaces | As for First plus: | As for Upper Second plus:  GUI standards have stated and discussed.  All necessary interfaces have been discussed, and it is made clear that these are not the final screen designs | The logical characteristics of each interface between the software and the users have been considered and described.  Interfaces between the system and any external systems have also been described.  A rough draft of sample screen images have been provided (using Balsamiq or similar).  Hardware and software interfaces outlined by an appropriate diagram.  Hardware interfaces could include supported device types etc. | A basic draft interface is present, but is not discussed in any significant detail and its purpose is not mentioned.  Some interfaces are missing, such as the any software and hardware interfaces that may be necessary. | Only the major user interfaces have been considered, and interfaces other than user interfaces may be absent entirely. | Only one main interface has been completed, indicating that the purpose of documenting the interfaces has not been understood. | No mock-up interfaces present or talked about in any significant detail | 10% |
| Use Case Modelling | As for First plus: | As for Upper Second plus:  Use case diagrams outlining all the use cases for the system. | Use case diagrams outlining the main use cases for the system. These should be done to the UML standard.  These should include all the actors mentioned in the introduction and address a problem the system is to solve.  At least one misuse case documented with comment on how it relates to (for example) security aspects of the system | Use case diagrams present, but may not fully conform to the UML specification.  1 Main use case may be missing, and not all actor’s needs satisfied.  Misuse case present but not explained | Only one use case present, and not all actors needs are present.  Little in the way of use-case specifications. | No use case diagram present at all, but description present. | No meaningful attempt at a use case | 15% |
| Project Plan | Full function point analysis has been performed on product backlog conforming to IFPUG 4.3.1 | As for upper second plus:  The benefits and potential drawbacks of the agile methodology chosen have been discussed.  Each members strengths have been briefly discussed, and their role assigned appropriately | The agile methodology that will be used has been clearly stated and explained. Details of the end of sprint meetings have been provided.  The role each member of the team will take on has been stated.  Evidence of an agile estimating technique such as function point analysis, T-shirt sizing, Story points or use case points.  A backlog has been provided. | An agile methodology has been described but the benefits not discussed.  The estimating of each task appears to have been done somewhat arbitrarily, but is still realistic. | A waterfall methodology has been described.  The development has been divided into tasks, but not subdivided fully. Estimating may feel inaccurate because of this. | Methodology not understood and explained incorrectly.  No estimating technique used, causing the estimating to be inaccurate. | No methodology named.  No evidence of estimating. | 15% |
| Individual Retrospective |  | Insightful analysis of the entire requirements elicitation process.  Clear, specific examples of how it may be approached differently next time | Detailed honest analysis of techniques used in the requirements gathering process that were effective.  Could include some techniques that could be used next time to gain requirements that are more detailed. | A more descriptive than analytical retrospective on the requirements gathering process.  Less attention given to what could have been done better next time. | Limited retrospective that covers only what went well in little detail. | Retrospective very limited and does not provide any useful information for next time. | No retrospective present | 10% |
| **Additional comments:** | | | | | | | |  |

**Part 2 – System and Report**

| **Class/**  **Grade**  **Assessment Criteria** | **Exceptional First**  (*includes all features of First Class with the additional characteristics listed below)* | **First**  **Low | Mid | High** | **Upper Second**  **Low | Mid | High** | **Lower Second**  **Low | Mid | High** | **Third**  **Low | Mid |High** | **Marginal Fail** | **Fail**  **Low | Mid** | **%** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Code/ implementation/Demonstration/design | As for First plus:  Code conforms to a code quality standard. | As for Upper Second plus;  Implementation wholly successful. Includes strategic decisions regarding error handling etc.  Design and implementation show elegant simplicity.  Extensive and excellent use cases chosen with pertinent secondary use cases. | Design reasonable but may miss some aspects of classes.  Each member of the team is able to explain different aspects of the code base and the patterns used, with assistance from their diagrams.  Implementation complete but may miss some use cases.  A couple of the tests are evidenced able to be run | Use cases may not cover appropriate functionality.  Implementation partially covers function points, but lacks coherence, and may not have a clear or matching design.  During the demo the team is unable to fully explain their design patterns | Implementation successfully meets a few of the requirements.  Does not match design as documented | Poor choice of use cases.  Implementation at best achieves 1 or 2 minor functions.  Design patterns not used at all | Implementation does not work at all | 50% |
| Design documentation | Design documentation written to a high professional standard. | Documentation that leaves very little ambiguous  The diagram is well laid out and shows skill in draughtsmanship and presentation of the idea.  The design pattern used is explained in detail with a brief description of a possible alternative pattern and why it was not used. | Architecture diagram, Deployment diagram, process diagram and structure diagram all completed to a good standard.  Some unnecessary detail may be present, but the key elements are focused on.  Each diagram’s purpose is explained to an adequate standard.  The design pattern used in the design process is explained showing a good understanding of the design pattern used | All diagrams attempted, but may have been completed to a basic standard **or** there is an excessive amount of unnecessary detail present.  Naming of objects may not follow any particular convention  An appropriate design pattern has been chosen and explained but why it has been chosen is not clear. | All diagrams present, but missing large aspects of the system on.  Diagram may have been mislabelled or deviate significantly from the UML standard.  Poor layout with difficult to follow dependency flow.  Design pattern explained in very little detail, why it was chosen is not discussed. | Not all diagrams present or diagrams significantly lacking.  Design patterns chosen not appropriate for project, and not understood. | No diagram worthy of credit provided.  No explanation attempted. | 20% |
| User help document |  | As for Upper Second plus;  Evidence of testing of the documentation to confirm it is useful. | A thoughtful and helpful document that could aid users with roles outlined in the first submission. | The user help document is mostly clear but may be ambiguous in some areas. | User documentation very basic | User documentation very unclear or largely incomplete | User documentation not present or entirely unhelpful | 10% |
| Acceptance test plan | As for first plus;  Evidence of test automation.  Evidence of test driven development.  Evidence of code path coverage. | As for Upper Second plus;  Evidence of thinking of positive test path and negative test path planning.  Evidence of unit test and system test levels. | Comprehensive test plan, which evaluates the functionality and performance of most of the application.  Documents which tests pass and which do not. | The test plan covers most of the functional requirements.  Strong evidence of test execution, but no summary of what functionality passes and fails. | The test plan covers a few of the functional requirements.  Weak evidence of test execution.  No attempt to summarize | Some attempt at a test plan.  Some attempt at test execution | Not a coherent test plan or test plan absent.  Test execution evidence is absent or hard to understand. | 10% |
| Individually written evaluation and code review |  | As for Upper second plus;  Code review written to a professional standard.  Quality attributes all discussed critically in detail. | How the system meets quality attributes in ISO 9126 is discussed. Evidence and examples are given to support this.  Each quality attribute is mentioned.  Potential improvements have been detailed in the weakest areas. | Most of the quality attributes in ISO 9126 are mentioned. Some aspects of the system given may be a loose fit for some of the attributes.  Some aspects of the system that could be improved are mentioned, including why these would be useful | Only some of the attributes in ISO 9126 are mentioned, or not described in much detail at all. | No quality attribute in ISO 9126 is mentioned. Evaluation limited. | No aspect of the system meaningfully evaluated. | 10% |
| **Additional comments:** | | | | | | | |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  | |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |