|  |  |
| --- | --- |
| **Intro** |  |
| The problem the system is to solve is explained is understood |  |
| 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 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 |  |
| Significant consideration of each aspect of the problem is evidence, 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 be |  |
| Scope is bulletproof, there is no ambiguity in what is in scope. Constraints discussed in good detail. |  |
| As for First plus:  Evidence of a user interview |  |
| **Requirements** |  |
| The requirements are detailed and organised in a way that is clear to read, and appropriate for the project. |  |
| Requirements don’t only concentrate on main stakeholder and cover all key requirements needed to meet the goals of the system. |  |
| The requirements should also be prioritised using MoSCoW or similar. |  |
| Full set of requirements covering the main stakeholders and their needs. |  |
| 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 |  |
| Requirements are present that directly address user characteristics. |  |
| **Interfaces** |  |
| 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. |  |
| 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 |  |
| **Use Case Modelling** |  |
| 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 outlining all the use cases for the system. |  |
| **Project Planning** |  |
| The agile methodology that will be used has been clearly stated and explained |  |
| The benefits and potential drawbacks of the agile methodology chosen have been discussed. |  |
| Details of the end of sprint meetings have been provided |  |
| The role each member of the team will take on has been stated. |  |
| Each members strengths have been briefly discussed |  |
| 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. |  |