# Software Project – year 2

**Requirements Document Guidelines**

The Requirements Document should have the following structure:

|  |  |
| --- | --- |
| 1. | **Introduction**  The introduction should describe the background to the project and the rationale for carrying out the project. The background of the project should include a description of context in which the application will be used.  This may include a description of the business of the client (if there is one) and a description of any existing applications (or non-computerised systems) that the proposed application will replace, integrate or compete with). The rationale for the project should outline the benefits of implementing the proposed application with respect to this background. |
| 2. | **Requirements Analysis**  This section should define the requirements of the software application you are proposing to develop for your Software Project. These requirements need to be described at a number of different levels of abstraction.  **2.1 User requirements**  This should be done for each potential user of the application, and should be illustrated using appropriate UML diagrams such as Use Case diagrams. According to Sommerville (2007, p. 118):  User requirements are statements, in a natural language plus diagrams, of what services the system is expected to provide and the constraints under which it must operate.  **2.2 Functional requirements**  These requirements should be of sufficient detail to represent a contract between a client and a software developer. Sommervile (2007, p. 118) describes the system requirements as follows:  System requirements set out the system's functions, services and operational constraints in detail. The system requirements document (sometimes called a functional specification) should be precise. It should define exactly what is to be implemented. It may be part of the contract between the system buyer and the software developers.  The functional requirements should describe, in outline, the principle elements of the user interface, and the functions to be performed in response to user |
|  | interaction with the principal user interface components.  This description should consist of outline wireframe diagrams of the user interface and an accompanying explanation of those diagrams and the functionality provided.  The relationship between these diagrams should be described in a storyboard. In addition, the functional requirements should outline the major entities that will comprise the system data model.  **2.3 Non-Functional requirements.**  Thirdly, any non-functional requirements should be described. Non-functional requirements include usability requirements, security and privacy requirements, and performance requirements. Given the scope of the Software Project project, non-functional requirements are often of lesser importance. However, this is not always the case.  In general, requirements analysis and specification should start by interviewing the client(s) (if there are any) to determine their requirements for the application.  A survey of similar applications should also be carried out to determine the functionality provided by those applications, and possible features that may be included in the user requirements. Given the scope of the Software Project project, it may not be possible to implement all of the (possible) user requirements. In this case, a subset of those requirements should be identified, and a further analysis of those requirements should be carried out. This should result in a detailed description of the associated functional requirements. |

# References

Sommerville, I. (2007). *Software Engineering* (Eighth ed.). Harlow, England: Pearson Education.