**DL836– Year 4 – Final Project**

**Feasibility Study and Requirements Document Guidelines**

***(Note: Discuss with your supervisor what is required for your project)***

The Feasibility Study & Requirements Document should have the following structure:

|  |  |
| --- | --- |
| 1. | Introduction  The introduction will describe the background to the project and the rationale for carrying out the project. |
| 2. | Requirements Analysis  2.1 Existing applications  A survey of similar applications may be carried out to determine the functionality provided by those applications, and possible features that may be included in the requirements.   * Screen shots * Descriptions * Advantages * Disadvantages   NOTE : Research from academic papers which may be done later may also influence the requirements.  2.2 User Profile  It is important to build a profile of the user which will help to understand the users’ requirements.  2.2.1 Interviews  Conduct interviews with potential users to determine their requirements. An overview of these interviews, the findings and decisions made due to these findings will be discussed here.  NOTE: Keep any proof of interviews such as notes, they may be included in the Appendices at the back of your final documentation if suitable.  2.2.2 Survey  You can create a questionnaire and use the results of the questionnaire as a basis for finding out requirements.  This section will provide an overview of these surveys and decisions made due to the results of the survey.  NOTE: Be specific with results, for example 90% of users wanted x in the application. Only 5% wanted y – therefore it has been decided to include x. Consider using images to represent your results – eg. Pie charts or bar charts.  NOTE: Once again keep proof of the surveys incase needed for the appendices in your final documentation  2.2.3 Personas  These are fictional characters to help the developer understand the users’ needs. They also help identify who the relevant users are.  2.3 Requirement Modelling  This section will define the requirements of the software application you are proposing.  NOTE: Below is a LinkedIn Learning course that may be useful when defining functional and non-functional requirements,  there's also a section on Use Cases and Use Case diagram.  [https://www.linkedin.com/learning/pro​gramming-foundations-object-oriented-design-3/defining-requirements?u=56744273](https://www.linkedin.com/learning/programming-foundations-object-oriented-design-3/defining-requirements?u=56744273)  Programming Foundations: Object-Oriented Design, by Baron and Olivia Stone - if the link above does not work, search for this title.  Watch the sections below (about 10-15 minutes)  2. Requirements (this covers Functional and Non-functional requirements)  3. Use Cases and User Stories  2.3.1 Functional Requirements  Create a numbered list of what the application should be able to do. Start with the most important feature.  If suitable for your project functional requirement/use case scenarios can also consist of outline wireframe diagrams of the user interface and an accompanying explanation of those diagrams.  Use Cases and Use Case Diagram  A use case is a description of how a person uses the system to accomplish a goal.  A use case diagram depicts all the use cases for a system and the Actors (users and other systems that interact with the systems) that are involved with each use case.  NOTE: Use Case Diagrams should be simple. They do not depict the order in which tasks are carried out. Watch the movie in the LinkedIn Learning course above for an example.  2.3.2 Non-Functional Requirement  For example - usability requirements, security and privacy requirements, and performance requirements.  NOTE : Once again – watch the movie in the link above. |
| 3. | System Model and System Requirements  A system model will be developed which identifies the principle system components for the application, and the interaction between these components.  This section will provide a diagram representing the system model and a discussion about each component.  The possible implementation platforms of the application will be identified and the advantages and disadvantages of these platforms discussed.  Finally conclude this section with your chosen implementation and the rational for this chosen implementation.    Fig. 1. Sample System Model |
| 4. | Feasibility Study  A feasibility study should be carried out to ascertain the possible risks associated with implementing the application.  The main issues that you may encounter are usually technical or project management issues. Identify these risks and discuss how you will manage and overcome these problems if they arise.  The project plan and test plan will help overcome these risks. |
| 5. | Project Plan  Usually the 4th year project will be approached using an agile methodology which is an iterative process. The initial iterations start off relatively simple, and gain in complexity as features are added.  Identify the steps/prototypes you plan for each iteration and map them out in a project plan – including dates and deliverables.  The stages of the development lifecycle are below, however development is not a linear process through these stages. Typically you will go through these stages a number of times; your project gaining complexity and new features during each iteration.   1. Research and analysis: background research in the problem domain and technical implementation platform; detailed analysis of functional requirements. 2. Outline design: including outline database, user interface and application logic design. 3. Detailed design: detailed specification of database, user interface and application logic. Class Diagram and sequence diagrams. 4. Implementation: implementation of system components, integration of system components. 5. Testing: unit testing, integration testing, system testing, and test plan. 6. Documentation write up   It is advisable that you start setting up the development environment and programming early in the project – consider working on the ‘theory’ (research, analysis, design and documentation) and the practical programming side of your project in parallel.  6 Test Plan  A description of how the application will be tested will be included.  The thought process involved in outlining a test plan will lead to a better understanding of application requirements.  Testing will include Unit, Integration, Regression, System and User Testing. A discussion of how you will approach these will be included in this section. |
|  |  |

NOTE : The quality of your requirements document is more important than the quantity. However, your document should be approximately 2,000 words in length (not including table of contents, references and any appendices). All external sources must be referenced properly using the ACM style (see below).

*You should discuss these guidelines including the interpretation and application of these guidelines, in the context of your project, with your project supervisor.*

**References**

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