**REQUIREMENTS DOCUMENT**

**(Suggested Full Proposal Template by DrM)**

Your requirements document shall be a textual document (usually of 2-4 pages) that includes the following information. (The following is taken from the text, "Software Engineering', by Ian Sommerville, Addison-Wesley, 1995.)

**Introduction:** This shall describe the need for the system, its functions, and explain how it will work with other systems. Explain how the system fits into the overall business or strategic objectives of the organization commissioning the software. What is the system and why should someone buy or use it?

**Management issues:** This is also a good time to begin considering some project management issues. You have already chosen a team and a team leader. It's probably too early to work out a detailed plan for who's doing what, or a detailed schedule with milestones, etc. However, do include the following:

* Sketch of a schedule with milestones (not detailed).
* Resource inventory and wish list. What technical skills do the members of the group have that are relevant to the project? What technical skills or knowledge do you plan to acquire during your work on the project (if any)? What domain knowledge do you have? Contacts with other domain experts?

**Glossary:** This should define the technical terms used in the document. No assumptions should be made about the experience or expertise of the reader.

**Functional requirements definition:** The services provided for the user should be described in this section. This description may use natural language, diagrams or other notations that are understandable by customers. In this class you are asked to abstract scenarios into use cases for the major system functions. This should not be detailed. We are not doing the software design yet; that comes later.

*DrM. Suggests the CS members of the team "interview" the BIO members of the team in the presence of the "customer-ie me" to learn the details of the workflow that is typically done now, or that they anticipate doing, before we design the software during this course that will "do it better" but this part of the proposal should be readable by everyone.*

**Requirements specification:** This should describe the functional requirements in more detail. Give a detailed list of all of the tasks the system will perform for the user, *without committing to how those tasks will be accomplished*. This section is directed at the developers; it should be readable by buyers/users who are willing to use the glossary, and should record the requirements of the users, but be detailed enough to act as a roadmap for the developers.

*DrM. Suggests the CS members of the teams use this section to "translate" the practical goals and procedures carried out by the BIO members into a more precise and logical description of the project here, with a "precise yet flexible plan" for the approaching the BIO goals using state-of-the-art informatics tools and paradigms.*

**System models:** This should set out one or more system models showing the relationships between the system components and the system and its environment. These might include *high level* object models, data flow models and semantic data models.

*DrM. Suggests a flow chart or diagram or some tables here to quickly illustrate the planned scope of the project and its parts and how they relate to each other.*

**Constraints:** This section covers the constraints imposed on the software and relates them to the functional requirements. In the real world, these might include hardware, software, other software components to be integrated, deadlines, budget, response time, etc. For purposes of this class, at a minimum, the constraints include:

* The deadlines given on the syllabus;
* The hardware that you plan to use (this is up to you, but must be a system that will make it possible for you to demo your system during the final week of class. In general, you should give the minimal and optimal configurations for the system here);
* The software that you plan to use (programming language(s) and any other requirements such as Netscape, databases, etc.); and
* Any other constraints specific to a particular project

**System Evolution:** This should describe the fundamental assumptions on which the system is based and anticipated changes due to hardware evolution, changing user needs, and so on.

The document shall also contain the following as chapters or appendices when needed.

* **Hardware:** If the system is to be implemented on special hardware, this hardware and its interfaces should be described. If off-the-shelf hardware is to be used, the minimal and optimal configurations for the system should be defined.
* **Database requirements:** The logical organization of the data used by the system and its interrelationships should be described. Data modeling techniques such as entity-relational modeling may be used to describe the database requirements (high level design only at this time).
* **Index:** You are going to be referring back to this document -what will make it easier for you to find the information you want? You should start with a table of contents. For a longer document, you might want a normal alphabetic index, and even for this document, you may find an index of functions to be helpful.

**Grading:** You will be graded on both the content and on the readability of your document.