**Recommended Practice for Software Requirements Specifications**

**(Based on IEEE Standard 830)**

(Will)

1. **Introduction**: This section contains an overview of the complete document.
   1. **Purpose** What is the purpose of this document, and for whom is it written?
   2. **Scope** An identification of the product to be developed, what does it do (and what does it not do), why is the product being developed (including a precise description of its benefits, goals and objectives)?
   3. **Definitions, acronyms and abbreviations** This subsection contains definitions of all the terms, acronyms and abbreviations used in the document. Special attention should be paid to the clarification of terms and concepts from the domain of application. (As Needed)
   4. **References** References to all documents that are referred to in the remainder of the requirements specification. (TBD)
   5. **Overview** This subsection contains an outline of the remainder of the document.
2. **General** **Description** This section contains a description of matters that concern the overall product and its requirements. It provides a perspective for understanding the specific requirements from section 3 of this document. (Katie)
   1. **Product** **perspective** This product is independent and stands alone. It will be its own executable file independent of other hospital software and software resources. Hardware used for this product will be the existing hardware already used by the hospital and its staff.
   2. **Product** **functions** There are 3 major functions for our product: Admitting, Testing/Diagnostics, and Billing/Discharge. Admitting will deal with taking in data about the patient and entering into the system. Testing and Diagnostics will serve to order tests, view results, and a section to enter a diagnosis. Billing/Discharge will generate a document of discharge instructions for the patient given by the doctor. It will also generate a bill compiled of all tests performed and medications given.
   3. **User** **characteristics** The users for this product will be general ER staff, nurses, doctors, and billing employees. There will be no special training, experience, or technical expertise required to use our product. The only requirement should be a general walkthrough with client and final product.
   4. **General** **Constraints** The only constraints for this product will be concerning access to different parts of the software and what sensitive information is being stored. Some users should not have access to read or write in certain areas. This is due to sensitive information about the patient and giving proper access to those that need it.
   5. **Assumptions** **and** **dependencies** This does not concern constraints on the system to be developed, but things which may influence the requirements specification once they change. As an example, we may think of the availability of certain supporting software, such as some given operating system or numeric library. If that operating system or library turns out not to be available, the requirements specification will have to be adapted accordingly. (TBD)
3. **Specific** **Requirements** This section contains all the details which are relevant for the design phase to follow. The ordering given here is just one way to present the specific requirements in a logical way. Specific requirements should be such that one may objectively determine whether they are fulfilled or not.

**(Nurses and Physicians Will)**

**(ER Staff and Financial Katie)**

ER Staff

* 1. **Functional** **Requirements** In this subsection, a description is given of how the transportation of inputs to outputs is achieved. The description given for each class of functions, and sometimes for each individual function. To a certain extent, this description can be seen as a solution to the user. This component of the requirement specification is the main starting point for the design phase.
     1. **Admitting and entering Patient Information**
        1. **Introduction** This function will allow ER staff to admit patients and enter in all required information. This will be done through a combination of textboxes, combo boxes, selection boxes etc.
        2. **Inputs** Required inputs will be: Name, address, phone number, prior medical history, time and date, and admission status. Address should be in standard format along with phone number. Prior medical history will just be general text entered by user. Time and Date will automatically be added upon submission. Admission status will be entered/updated by user.
        3. **Processing** Before submitting data to the database we will check for all required lengths and formats. Functions to count length of strings will be performed as well as checking that required information was entered. If there is anything incorrect it will prompt the user to correct it before attempting to submit again.
        4. **Outputs** The output for this function will be the submitting/entering the given data into the database.
     2. **Generating Bill**
        1. **Introduction** This function will generate a bill document based on all tests ordered, prescriptions/medications given, and total nights stayed. These will all have set fees that will be listed on the itemized bill.
        2. **Inputs** There will be no actual direct input for the required information. This information will be pulled from the database and formatted into a document.
        3. **Processing** After pulling data we will need to ensure calculations for the total bill are done correctly based on the data. We will also need to ensure that the total is properly formatted.
        4. **Outputs** The output for this will be a generated document for the ER staff to print and give to patient. This document will clearly state what the charge is and how much it was. It will also list the total amount due to be paid.
     3. **Discharge Papers**
        1. **Introduction** We will need to generate discharge papers using instructions and medications given by the doctor/nurse. These will be given to the patient upon discharge with their bill.
        2. **Inputs** There will be no direct inputs and information will be requested from the data base. There should only be discharge instructions, a diagnosis from doctor, and prescriptions if given.
        3. **Processing** There should be no need to process or check data pulled from the database as no changes to the data will be made. We will only need to request the information from the data base and then format it into a discharge document.
        4. **Outputs** The output for this function is the discharge document with instructions, diagnosis, a prescription information. This will be provided to the patient.
  2. **External** **interface** **requirements**
     1. **User** **interfaces** A description of the characteristics of the user interfaces, such as screen layout, function keys, help functions. In order to support testing, verifiable requirements regarding learning time for the system functions should be included either here or in some subsection of 3.5 (Attributes).
     2. **Hardware** **interfaces** A description of the logical characteristics of hardware interfaces, such as interface protocols, or screen-oriented versus line-oriented terminal control. (Basic Work Computer, system already in place)
     3. **Software** **interfaces** There will be no required software for this product. It will be able to run all computers in the ER department with the already existing operating systems.
     4. **Communications** **interfaces** An example is a communication protocol for LANs. (N/A or TBD)
  3. **Performance** **requirements** Performance requirements encompass both static and dynamic requirements. Static requirements concern, amongst others, the number of terminals to be connected and the number of users that can be handled concurrently. Dynamic requirements concern the operational performance of the system: how frequently will certain functions be called for and how fast should the system’s reaction be. It is important that these requirements be stated in measurable terms. (TBD)
  4. **Design** **constraints** Design constraints may result from such things as the prescribed use of certain standards or hardware. (Ask about it, possibly comparable to max number of people admitted to ER)
     1. **Standards** **compliance** Which existing standards or regulations must be followed, and what requirements result from these. For example, certain report formats or audit procedures may be prescribed.
     2. **Hardware** **limitations** A description of the characteristics of the hardware environment, in as far as they lead to software requirements. An example of this might be the amount of memory available.
  5. **Attributes** In this section, particular attention is paid to quality aspects. These requirements must be measurable and verifiable. They must be stated in objective terms. The subsections below by no means comprise a complete list of such attributes.
     1. **Availability** Factors that guarantee a certain level of availability, such as restart procedures. In this subsection we may also enlist requirements regarding fault tolerance (with respect to both hardware failures and software failures).
     2. **Security** **Requirements** regarding unauthorized access and other forms of misuse. Certain cryptographic techniques may be prescribed, and we may put constraints on the communication between different parts of the system. (Ask)
     3. **Maintainability** Requirements to guarantee a certain level of maintainability of the system, such as a maximum allowable coupling between components. (Ask)
  6. **Other** **requirements** A description of requirements that are specific to certain software, and which have not been discussed yet. (TBD)