Software Requirements Specification

for

<Project>

Version 1.0 approved

Prepared by <author>

<organization>

<date created>

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1.   **Introduction**

1.1   Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>

1.2   Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

1.3   Intended Audience and Reading Suggestions

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

1.4   Product Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

1.5   References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

2.   **Overall Description**

2.1   Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify

interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

2.2   Product Functions

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 4, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

2.3   User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

2.4   Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

2.5   Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

2.6   User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

2.7   Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the

project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

3.   **External Interface Requirements**

3.1   User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

3.2   Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

3.3   Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

3.4   Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

4.   **Domain Model**

<Sometimes, this section is optional. However, it may be important to have it since domain model may give more useful as well>

5.   **System Features (Use Cases)**

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of

these, whatever makes the most logical sense for your product. You should Use-case diagram>

5.1 Use Case 1

5.1.1 *Name:* System Administrator

5.1.2 *Goal:* To administrate authorisation process of entire users.
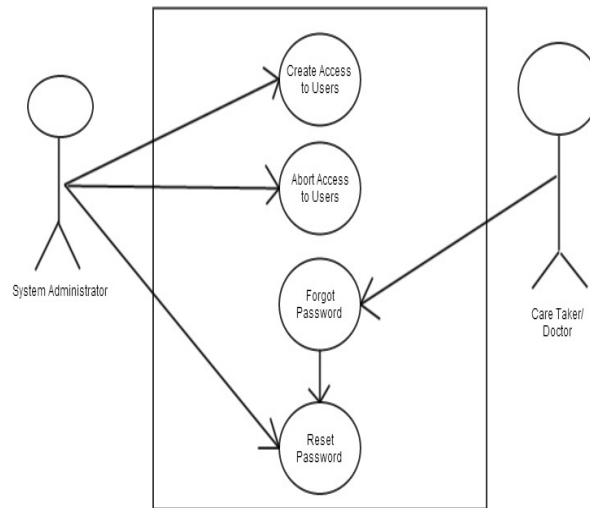
5.1.3 *Input:* **Create Access to Users**

1. A set of User Name and Password belonging to particular user will insert in to Data Base.

Abort Access to Users
1. A set of User Name and Password belonging to particular user will get deleted from Data Base.

5.1.4 *Output:* **Create Access to Users**

1. Authentication is enabled for the particular user.

Abort Access to Users
1. Authentication is disabled for the particular user.

5.1.5 *Main Scenario:* To provide access to all the users and also to reset password when the user forgot password. He should pop up error message when authentication fails.

5.1.6 *Pre-condition:* **Create Access to Users :** No pre-conditions.

Abort Access to Users :

1. User must already have an access to system.

5.1.7 *Steps:*

5.1.7.1 Step1: Create access to particular user by inserting user name and password in to table.

5.1.7.2 .Step2: If authentication is successful it should redirect in to particular home page. Else it should pop up an error message.

*5.1.7.3 Step 3:*If the user is no longer in the organisation then he should deactivate particular account.

5.1.8 *Post-condition:* none.

5.1.9 *Exceptional Scenario 1:* none
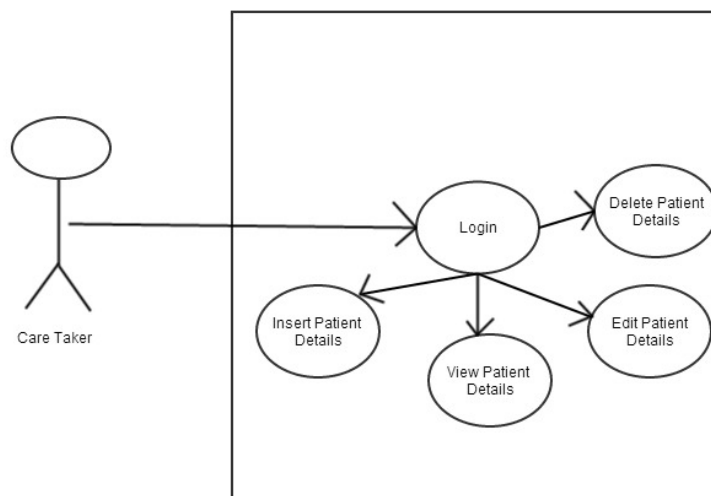
5.1.10 *Example:*

Suppose we want to give access to Care taker name "Venkat" then the system administrator will insert Venkat as user name and password like "subbu" in to data base.

If "Venkat" is no longer in organisation then he will delete his record from the data base.

5.2 Use Case 2

5.2.1 *Name:* Care Taker

5.2.2 *Goal:* To Insert,view,edit and delete patient records.

5.2.3 *Input:*

1. Login :
      Care Taker will enter username and password.
2. Insert Patient Details:
      Care Taker will enter details of patient such as name,id and 12 readings of that particular
timestamp.
3. View Patient Details:
      Care Taker will enter patient Id of the patient to view particular patient details.
4. Edit Patient Details:
      Care Taker will enter patient Id of the patient to edit particular patient details.
5. Delete Patient Details:
      Care Taker will enter patient Id of the patient to be deleted.

5.2.4 *Output:*

1. Login:
      If authentication fails it will pop up error message. If authentication suceeds he will redirect to
particular page.
2. Insert Patient Details:
      After inserting it will display message as " Patient items inserted successfully".
3. View Patient Details:
      It will display the details of particular patient.If there is no record of that particular patientId then it
should display "No Records found".
4. Edit Patient Details:
      It will display the details of particular patient to edit and after editting it will insert back to database.
If there is no record of that particular patientId then it should display "No Records found".

5. Delete Patient Details:
      After deleting patient details it will display message "Details of patient Id are deleted successfully".
If there is no record of that particular patientId then it should display "No Records found".

5.2.5 *Main Scenario:*
      Care Taker will login and he can insert, view, edit and delete patient records.

5.2.6 *Pre Conditions:* No.

5.2.7 *Steps*
5.2.7.1 *Step1:*
      Care Taker will log in in to system.
5.2.7.2 *Step2:*
      If he wants to insert data then he will insert details of particular patient.
**5.2.7.3** *Step3:*
      If he wants to view data then he will view details of particular patient.

5.2.7.4 *Step4:*

        If he wants to edit data then he will edit details of particular patient.

5.2.7.5 *Step5:*

        If he wants to delete data then he will delete details of particular patient.

5.2.8 *Pre Condtions*: None

5.2.9 *Exceptional Scenario 1:* None

5.3   Use Case 3

5.3.1 *Name:* Doctor

5.3.2 *Goal:* To view details of patient records and generate reports.
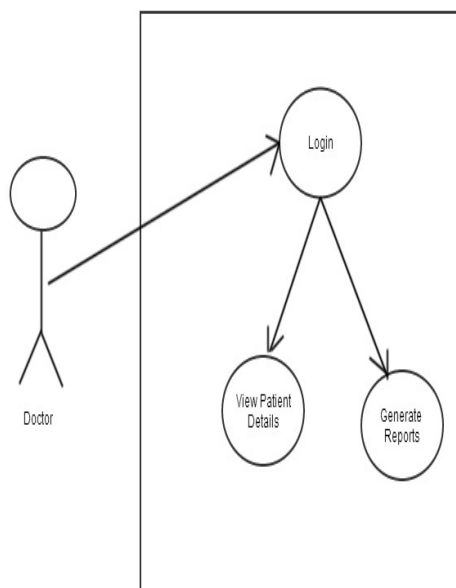
5.3.3 *Input:*

1. Login :

        Doctor will enter username and password.

2. View Patient Details:

        Care Taker will enter patient Id of the patient to view particular patient details.

3.Generate Reports:

        Doctor will generate reports based on readings.

5.3.4 *Output:*

1. Login:

        If authentication fails it will pop up error message. If authentication suceeds he will redirect to particular page.

2. View Patient Details:

        It will display the details of particular patient.If there is no record of that particular patientId then it should display "No Records found".

3. Generate Reports:

      A graphical representation of the condition of patient is drawn from the readings.

5.3.5 *Main Scenario:*

Doctor will login and he can view patient records and generate reports .

5.3.6 *Pre Conditions:* No.

5.3.7 *Steps*

5.3.7.1 *Step1:*

      Doctor will log in in to system.

5.3.7.2 *Step2:*

      He can view the details of the patient.

5.3.7.3 *Step3:*

      He can generate reports based on the readings.

5.3.8 *Pre Condtions*: None

5.3.9 *Exceptional Scenario 1:* None

6.   **Other Nonfunctional Requirements**

6.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

6.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

6.3 Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

6.4   Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

7.   Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>