

**Software Requirements Specification**

**for**

**<Project>**

**Version 1.0 approved**

**Prepared by <author>**

**<organization>**

**<date created>**

***Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.***

***Software Requirements Specification for <Project>*** ***Page ii***

**Table of Contents**

|  |  |  |  |
| --- | --- | --- | --- |
| **Table of Contents ..........................................................................................................................** | | | **ii** |
| **Revision History ...........................................................................................................................** | | | **iii** |
| **1.** | **Introduction ..............................................................................................................................** | | **1** |
|  | 1.1 | Purpose ........................................................................................................................................ | 1 |
|  | 1.2 | Document Conventions ............................................................................................................... | 1 |
|  | 1.3 | Intended Audience and Reading Suggestions ............................................................................. | 1 |
|  | 1.4 | Product Scope .............................................................................................................................. | 1 |
|  | 1.5 | References ................................................................................................................................... | 1 |
| **2.** | **Overall Description ..................................................................................................................** | | **2** |
|  | 2.1 | Product Perspective ..................................................................................................................... | 2 |
|  | 2.2 | Product Functions ........................................................................................................................ | 2 |
|  | 2.3 | User Classes and Characteristics ................................................................................................. | 2 |
|  | 2.4 | Operating Environment ............................................................................................................... | 2 |
|  | 2.5 | Design and Implementation Constraints ..................................................................................... | 2 |
|  | 2.6 | User Documentation .................................................................................................................... | 2 |
|  | 2.7 | Assumptions and Dependencies .................................................................................................. | 3 |
| **3.** | **External Interface Requirements ...........................................................................................** | | **3** |
|  | 3.1 | User Interfaces ............................................................................................................................. | 3 |
|  | 3.2 | Hardware Interfaces .................................................................................................................... | 3 |
|  | 3.3 | Software Interfaces ...................................................................................................................... | 3 |
|  | 3.4 | Communications Interfaces ......................................................................................................... | 3 |
| **4.** | **Domain Model ..........................................................................................................................** | | **4** |
| **5. System Features (Use Cases) ...................................................................................................** | | | **4** |
|  | 5.1 | Use Case 1 ................................................................................................................................... | 4 |
|  | 5.1.1 | Name: ...................................................................................................................................... | 4 |
|  | 5.1.2 | Goal: ........................................................................................................................................ | 4 |
|  | 5.1.3 | Input: ....................................................................................................................................... | 4 |
|  | 5.1.4 | Output: .................................................................................................................................... | 4 |
|  | 5.1.5 | Main Scenario: ........................................................................................................................ | 4 |
|  | 5.1.6 | Pre-condition: .......................................................................................................................... | 4 |
|  | 5.1.7 | Steps: ....................................................................................................................................... | 4 |
|  | 5.1.8 | Post-condition ......................................................................................................................... | 4 |
|  | 5.1.9 | Exceptional Scenario 1 ........................................................................................................... | 4 |
|  | 5.1.10 Example................................................................................................................................... | | 4 |
|  | 5.2 | Use Case 2 (and so on) ................................................................................................................ | 5 |
| **6.** | **Other Nonfunctional Requirements .......................................................................................** | | **5** |
|  | 6.1 | Performance Requirements ......................................................................................................... | 5 |
|  | 6.2 | Safety Requirements .................................................................................................................... | 5 |
|  | 6.3 | Security Requirements ................................................................................................................ | 5 |
|  | 6.4 | Software Quality Attributes ........................................................................................................ | 5 |
| **7.** | **Other Requirements ................................................................................................................** | | **5** |
| **Appendix A: Glossary....................................................................................................................** | | | **6** |
| **Appendix B: Analysis Models .......................................................................................................** | | | **6** |
| **Appendix C: To Be Determined List ............................................................................................** | | | **6** |

***Software Requirements Specification for <Project>*** ***Page iii***

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

***Software Requirements Specification for <Project>*** ***Page 1***

**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.>*

***Software Requirements Specification for <Project>*** ***Page 2***

**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.>*

***Software Requirements Specification for <Project>*** ***Page 3***

**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.>*

***Software Requirements Specification for <Project>*** ***Page 4***

**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**

*<Don’t really say “Use case 1.” State the feature name in just a few words.>*

***5.1.1 Name:***

***5.1.2 Goal:***

***5.1.3 Input:***

***5.1.4 Output:***

***5.1.5 Main Scenario:***

***5.1.6 Pre-condition:***

***5.1.7 Steps:***

*5.1.7.1 Step1:*

*5.1.7.2 .*

*5.1.7.3 Step n:*

***5.1.8 Post-condition***

***5.1.9 Exceptional Scenario 1***

***5.1.10 Example***

***Software Requirements Specification for <Project>*** ***Page 5***

**5.2 Use Case 2 (and so on)**

**6. Other Nonfunctional Requirements**

**6.1 Performance Requirements**

* Due to the significant and pressing nature of many electrocardiogram results, the EKG cloud service should be available to all users at all times. That is, the cloud service should be accessible at all hours of the day, every day of the week, and every week of the year.
* EKG data input stream will be limited by the user’s internet connection/speed. Once the data is in the file storage, however, the maximum response time for accessing the data files should be 5 seconds in order to better facilitate rapid responses from the medical team(s).
* The file storage will be replicated over multiple independent cloud servers to decrease the risk of failure. If a failure occurs on the initial file storage server, the database will point to a different server in order to keep the data transaction seamless.
* Uptime and downtime for the system will be continuously evaluated so that necessary maintenance operations will occurs during times that have the least probability for negatively impacting users.

**6.2 Safety Requirements**

* The SQL database will simply store pointers to files in the Amazon S3 file storage. This will increase speed of database requests and insure that the database is not capable of modifying or interfering with the EKG data.
* EKG data will not be allowed to be deleted, and will instead be archived on the user side.
* The system will maintain log files which will allow data recovery and allow system admins to recognize points of failure in order to communicate inefficiencies with the development team.
* Database restore points will be created frequently in order to allow the database to rollback to a previous state.

**6.3 Security Requirements**

* Users (patients and/or medical staff) will not have direct access to the Amazon RDS database or the Amazon S3 file storage. Only authorized members of the software development team will have access to these.
* Users will have unique IDs, and their passwords will be required to be at least 8 characters long, case sensitive, and include letters, numbers, and symbols.
* Accounts for medical staff will be generated and confirmed by the clinics/hospitals/etc. using the software in order to insure that only authorized medical staff are able to access the software. Additionally, the respective medical facilities will be responsible for determining which nurses or attendants belong to a specific doctor’s team.
* Doctors will be required to verify patients as users before patients are allowed to submit data. Likewise, patients will be required to confirm that they are sending data to the correct doctor. This is an extra precaution to insure that patient data is not erroneously sent to unintended users.
* Doctors will confirm the privileges allowed for each member of their team.
* Database encryption will be employed as an additional precaution against data interception.

**6.4 Software Quality Attributes**

* Adaptability: This is a measure of the software’s ability to adapt to changes in its environment. By using our database to point to files, we allow more efficient options to be easily interchanged in the future.
* Availability: This is a measure of the system’s ability to be accessed when needed. Scheduling maintenance during calculated downtimes allows the software’s unavailability to be limited and impact the fewest users.
* Portability: This is a measure of the software’s ability to be used within multiple environments. By providing a web portal, we allow the software to largely be impartial in regards to operation on user devices.
* Reliability: This is a measure of the system’s ability to be error-free over a given period of time. The replication of our file storage allows for quick recovery options if a file storage server fails.
* Usability: This is a measure of the software’s ability to be efficiently and effectively used by users. The high availability, portability, and reliability of our system combined with a simple user interface helps to insure a positive user experience that is characterized by ease of use.

**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.>*

***Software Requirements Specification for <Project>*** ***Page 6***

**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.>*