Software Requirements Specification

for

Central Blood Bank Management System

Version 1.0

Prepared by

Group : 14

Group Name: Dead Inside

Nibir Baruah	19807545	nibir@iitk.ac.in
Harsh Patel	190363	harshp@iitk.ac.in
Aditya Prakash	190066	prakasha@iitk.ac.in
Kartik Jhanwar	190418	kartikjh@iitk.ac.in
Ayush Singh	190217	ayusgh@iitk.ac.in
Nikhil Mehta	190549	nikhilme@iitk.ac.in
Amit Kumar Singh	190117	amitsgh@iitk.ac.in
Nishima Panwar	190562	niship@iitk.ac.in
Adhiraj Sinha	190053	adhirajs@iitk.ac.in

Course: CS253

Mentor TA: *Ashutosh*

Date: 27-01-2023

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Nibir Baruah Aditya Prakash Amit Kumar Singh Harsh Patel Anuj Chaudhary	Initial Draft	27/01/2023

Version	Primary Author(s)	Description of Version	Date Completed
	Ayush Singh		
	Manish Mayank		
	Kartik Jhanwar		
	Nikhil Mehta		
	Nishima Panwar Adhiraj Sinha		

1. Introduction

1.1. Product Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals.
TO DO: 1-2 paragraphs describing the scope of the product. Make sure to describe the benefits associated with the product.>
Our app is a central blood bank database management system. The app allows blood banks from various parts of the country to register themselves and be part of a centralized blood database. Once a blood bank registers itself, any user visiting the web app may view the available blood units of different blood groups and blood components of that blood bank. The blood bank can update the available units using the app interface. The app also supports adding blood camp events by individual blood camps. These events can then be viewed by any visiting user of the webpage. The app display would show various camp details like start and end dates, start and end time for each day, location and organizer.

The purpose of our app is to provide a centralized blood database online. By doing this, anyone who is need of a particular blood component of a blood group can look up if there are available blood units in any of the blood banks nearby her/his area. Additionally, a person who is willing to donate blood can look up in the app if there are any blood camp events happening around his/her neighborhood or in fact directly go to a nearby blood bank to donate blood. Our aim was also to provide the blood banks with a common interact to interact with potential donors and consumers. The app also provides them with a very trivial way to update their blood data.

1.2. Intended Audience and Document Overview

<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.>
The users of this document are the document writers, developers, testers, project managers, clients, and Blood Bank tech officials. This document will be used by software developers who are accountable for its creation. The testers and users will use this document to see if the software satisfies the specifications mentioned in the document. Clients and managers who are in charge of software management will also study this text.

The project is primarily for managing a central common Blood Bank, and it can be used by any individual or general public. Blood bank data updating is restricted to the Blood bank officials. The project will be implemented under the guidance of the course instructor, teaching assistants. All the people can be benefited from this endeavor.

***Document Overview**: This section describes the overview of the Software Requirement Specifications for **'Central Blood Bank Management System'**.*

***Section 1**: This section explains the product scope, intended audience of the document, conventions, and abbreviations. Product scope will be critical for the end-users and the marketing staff.*

Section 2*: The product overview,* ***high-level descriptions of the product functionality, restrictions****, limits, assumptions, and dependencies are all described in this section. This information will be critical for the developers and the marketing staff. The user manual writer needs to go through this section.*

Section 3: This section explains the **system's requirements** in detail. Hardware and requirements, user interfaces, use case diagrams, and functional requirements are included in this section. Details in this section are important for the developers.

Section 4: **Non-functional requirements** are discussed in this section. This encompasses the quality, performance, safety, and security aspects of our software. This section will be critical for the developers, testers, and the manager.

All developers, testers, and the product manager are required to read the complete document. On the other hand, marketing staff should go through sections 1, 2.1, and 2.2. The user manual writer needs to go through sections 2.1, 2.2, 2.4, and 3.1.

1.3. Definitions, Acronyms and Abbreviations

<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.
TO DO: Please provide a list of all abbreviations and acronyms used in this document sorted in alphabetical order.>
Blood Components - Red Blood cells, Plasma, Platelets, Granulocytes, Cryo-precipitated AHF.

1.4. Document Conventions

<In general this document follows the IEEE formatting requirements. Use Arial font size 11, or 12 throughout the document for text. Use italics for comments. Document text should be single spaced and maintain the 1" margins found in this template. For Section and Subsection titles please follow the template.
TO DO: Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. Sometimes, it is useful to divide this section to several sections, e.g., Formatting Conventions, Naming Conventions, etc.>

This document largely follows the **IEEE template for System Requirement Specification Documents.** Document text is singly paced and the 1'' margins.
Formatting Conventions - We used Arial font size 11, or 12 throughout the document for text, and bold weight to indicate and emphasize important text in the document.

1.5. References and Acknowledgments

*<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. >

Acknowledgement - We would like to express our heartfelt gratitude to **Dr. Indranil Saha**, our Course instructor, for encouraging and allowing us to work on the project “A Central Blood Bank Management System” as our software development project and assisting us whenever needed.
*It is also our pleasure to extend our heartfelt gratitude to **Mr. Ashutosh**, our Project TA, for his wise counsel and constructive criticism during the term of our project.*

2. Overall Description

2.1. Product Overview

<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. In this part, make sure to include a simple diagram that shows the major components of the overall system, subsystem interconnections, and external interface. In this section it is crucial that you will be creative and provide as much information as possible.
TO DO: Provide at least one paragraph describing product perspective. Provide a general diagram that will illustrate how your product interacts with the environment and in what context it is being used. This is not a formal diagram, but rather something that is used to illustrate the product at a high level.>

The entire ecosystem consists of both offline and online components. The collection of blood will be manual through blood donation camp. The donor can either register on the blood bank website on his own or can visit the blood donation camp and the responsible authority at the camp can do the registration for the donor. An online database is maintained with all the information about the donors. Once the blood is collected it is stored in a safe place. An online blood inventory database is maintained as well for the blood units collected. From here a small sample of the blood units are sent to the testing unit. Here the blood samples are tested to determine whether they are fit to be used or not. A report is made by the lab technician and is sent back to the blood bank. Based on the reports received the blood bank inventory is updated and some blood samples might need to be discarded as they are not fit for use. Hospitals place orders from this blood bank. Once the hospital makes the payment the order is delivered by the blood bank staff. Blood donation camp information can also be looked up on the website.

2.2. Product Functionality

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary is needed here. These can be at the level given in the project description.>
TO DO:
1. Provide a bulleted list of all the major functions of the system

- Blood banks can register themselves on the website and be a part of a centralized blood donation system.
- Anyone who is looking for blood can look up the website to see if there are available units of the required type and component in the nearby blood camps.
- If a person wishes to donate blood, he can look up in the website if there are any blood donation camps happening or he can look up the nearest blood bank to go to for donating blood.
- A donor can register himself on the website so that he can updates about possible blood donation events on his dashboard.
- Blood banks can register blood donation camps events on the website so that all the registered donors can get notified about the event.

2.3. Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: 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). You can be creative here to some degree.>

1. As the system is about providing services online, we should make sure the power supply, internet, internet bandwidth, and communication with the central server are always functioning.

2.4. 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.

TO DO: Provide a short list of some major assumptions that might significantly affect your design.>

1. A client should have a modern browser *which supports JavaScript and HTML5.*
2. Each blood bank will have someone to manage their account, otherwise many users will be misguided.

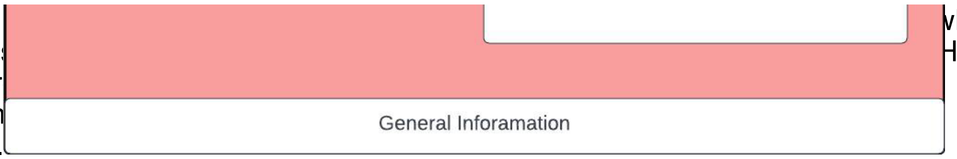
3. Specific Requirements

2.4. External Interface Requirements

2.4.1. User Interfaces

In this part, we have tried to comprehend all the pages which any user will see and how the interactions work between them. Some tentative wireframes are provided below with their explanation.

Public Page:



Every vhich has various
button Header and footer
may r screen are self-
explan is defined
below.

Donate Blood:

When a user clicks on the Donate blood button (present in the header as well as main screen), he/she is redirected to another public page which has information about all the upcoming blood camps and can be filtered by state and city. City can only be selected once state is selected.

Find Blood:

When a user clicks on the Find Blood button, he/she is redirected to a page where information about the blood banks and available quantity is provided. This can be filtered out using state, city, blood type etc. Blood bank search bar can be added but that will not be our priority initially.

Login Screen:

A blood bank user can login through this screen. This can be modified to add more login options such as Login with Google etc.

A blood bank can create an account which gives them the privileges to update their data of availab d donation camps.

Add Camp:

A signed in user can create a camp which will be shown to the public in Donate Blood screen. We have not yet decided exactly which data we will be taking, but it will contain information of city, location, contact person etc.

Update Blood Data:

This screen is similar to where a blood bank user can update the available unit of each component and this data will be updated for public in the find blood tab.

2.4.2. Hardware Interfaces

a) Server Side:

The web application will be hosted on a web server which is listening on the web standard port, port 8080.

b) Client Side:

There are no specific requirements on the client side.

2.4.3. Software Interfaces

Client:

A client should have a modern browser *which supports JavaScript and HTML5.* We are using *HTML, CSS* and *JS* for the frontend and python Django framework for the backend.

Server:

Django, being a web framework, needs a web server in order to operate.

2.5. Functional Requirements

< Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform. This section is the direct continuation of section 2.2 where you have specified the general functional requirements. Here, you should list in detail the different product functions.

Blood banks can register themselves on the website and be a part of a centralized blood donation system. Anyone who is looking for blood can look up the website to see if there are available units of the required type and component in the nearby blood camps. If a person wishes to donate blood, he can look up on the website if there are any blood donation camps happening or he can look up the nearest blood bank to go to for donating blood. A donor can register himself on the website so that he can updates about possible blood donation events on his dashboard. Blood banks can register blood donation camps events on the website so that all the registered donors can get notified about the event.

2.5.1. F1: The system shall …

2.5.2. <Functional Requirement or Feature #2>

…

2.6. Use Case Model

TO DO: Provide a use case diagram that will encapsulate the entire system and all actors.

2.6.1. Use Case #1 (use case name and unique identifier – e.g. U1)

TO DO: Provide a specification for each use case diagram

Author **– Identify team member who wrote this use case**

Purpose - What is the basic objective of the use-case. What is it trying to achieve?

Requirements Traceability – Identify all requirements traced to this use case

Priority - What is the priority. Low, Medium, High. Importance of this use case being completed and functioning properly when system is deployed

Preconditions - Any condition that must be satisfied before the use case begins

Post conditions - The conditions that will be satisfied after the use case successfully completes

Actors – Actors (human, system, devices, etc.) that trigger the use case to execute or provide input to the use case

Exceptions - Exceptions that may happen during the execution of the use case

Includes (other use case IDs)

Notes/Issues - Any relevant notes or issues that need to be resolved

2.6.2. Use Case #2

…

Other Non-functional Requirements

2.7. Performance Requirements

- Regular virus protection must be done.
- Database must be archived regularly.

2.8. Safety and Security Requirements

- Care must be taken to minimize and prevent human error. Users must be asked to confirm once again before approving actions that alter the data stored in the database such as adding or deleting entries.
- Access to data in the database is restricted and the data is secured through multiple layers of protection. **Each user is required to enter a unique userID and password pair to access the software.** The relevant features are made available based on the level of authority of the logged-in user for maximum security, each member has to protect her/his password secretively.
- If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, **the recovery method restores a past copy of the database** that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed-up log, up to the time of failure.

2.9. Software Quality Attributes

- **Reliability**: The databases are archived regularly and can be recreated.
- **Adaptability**: The ability to easily add or remove features in the portal without affecting the data and the database structure will be achieved using proper and flexible integration between frontend and backend software.
- **Availability;** The system should be available 24 x 7. In the case of data corruption, the data should be retrived from the application saved be the administrator.
- **Performance** The system is interactive and it is a one-stop solution for blood donors, blood banks and consumers.

3. Other Requirements

<This section is ***Optional.*** *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 – Data Dictionary

<Data dictionary is used to track all the different variables, states and functional requirements that you described in your document. Make sure to include the complete list of all constants, state variables (and their possible states), inputs and outputs in a table. In the table, include the description of these items as well as all related operations and requirements.>

Had a total of 2 online zoom meetings, where we discussed the division of work. Regularly coordinated over WhatsApp group conversations. -> till 27/01/2023

Appendix B - Group Log

<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist in determining the effort put forth to produce this document>