Software Requirements Specifications Document

Project: Cardinal

SWE 3313 | Group #1

Andujar Brutus | Pamir Ahmad | Audrey Allen | El Arbi Belfarsi | Thomas Breitung

Ver. 1.0 **10/23/2021**

1. Introduction

1.1 Purpose

This SRS document is intended to inform interested parties of the Cardinal program's intended functions. Potential readers may be any person who would like to utilize the software after launch, namely blood bank resource managers, donors, and medical professionals. This SRS will inform potential users of what they can expect the program to accomplish once it is launched.

1.2 Scope

The software product in production has been given the name Cardinal; this product is intended to establish real-time resource tracking and communication between blood banks and the people who utilize their services. Donors will be able to view which bank is the best option for them, based on factors such as distance and need for resources. The blood banks themselves will use the program to catalog their resources, as well as keep track of the expiration dates of those resources. When the medical professionals using Cardinal request blood or plasma, they will be prioritized by order of expiration as to utilize as many resources as possible. This software will be used exclusively for blood banks; Cardinal is not planned to be applied to other medical facilities or means of resource management.

2. Specific Requirements

2.1 External Interfaces

Blood bank administators –

Inputs:
-Identification of bank
-Resource type (ex. blood, plasma, etc)
-Resource subtype (ex. blood/plasma type)
-Amount donated
-Donor
-Date of donation
-Expiration date of donated resources
Outputs:
-All variables listed above (utilized through a search function)

Donors –

Inputs:
-Location of donor
-Donor ID
-Resource to be donated
-Blood/plasma type
Outputs:
-Location of nearest blood bank
-Distance of blood bank from donor's location
-Estimated travel time
-Blood bank with greatest need
-IDs of listed bank(s)

Medical professionals –

Inputs:
-Location of medical facility
-ID of medical facility
-Resource(s) needed
-Blood/plasma type needed
-Amount needed
-Purpose of request
-Request ID (if modifying an established request)
Outputs:
-Location of blood bank that best fits request (prioritized by resource expiration)
-ID of listed bank(s)

2.2 Functional Requirements

The system shall provide real-time updates on the resources of registered facilities.

- The system shall provide a table of appropriate centers.
  -**Inputs**: maximum distance from user center, blood component requested, general data structure containing all centers
  -**Output**: table corresponding to input parameters
- The system shall sort the table using the expiration dates of resources
  -**Input**: an unsorted table
  -**Output**: a table sorted by expiration dates

The system shall provide a management system for reservations.

- The system shall process requests for batches of specific blood components.
  -**Inputs**: blood component, amount of blood, purpose of request, center of interest
  -**Output**: full request in priority queue sent to center
- The system shall allow modification of requests.
  -**Inputs**: request ID, updated information
  -**Output**: updated request in priority queue sent to center

The system shall account for a bank's total resources and level of need.

- The system shall exhibit the banks with low availability of resources.
  -**Inputs**: location of bank, level of need, blood component in question
  -**Output**: a table corresponding to the input parameters
- The system shall suggest centers convenient to donors.
  -**Inputs**: exhibition table, parameters provided by user
  -**Output**: suggested centers

2.3 Non-Functional Requirements

2.3.1 Performance

As Cardinal is a web-based application, the program should work on most devices with a supported browser installed. Unfortunately, at this current state we can only estimate the expected simultaneous users. To estimate the number of concurrent users we would need information on the hosting server we would likely use for the website, and the average time for a page request. With cardinal we will be able to set a max range in which a user can check for blood donations, and this is how we can limit the amount of information to be handled so one user cannot overload the system. We expect that about 90% of the clients request to be processed in less than 1 second. We also expect the connection of the server to be less than 1 second for loading the UI needed so the user can query blood donations.

2.3.2 Reliability

At launch we expect that the Cardinal system should have an approximated uptime of 85 to 90 percent. With efficient testing and appropriate usage conditions, the cardinal system should experience few critical failures. If there are to be any critical failures at all, it is estimated that this number should fall between 1 and 3 for the first month of launch with an ideal average downtime of 6 to 10 hours for maintenance. The average time between critical failures should be greater than 1.5 weeks as bugs are eliminated post-launch. As bugs and critical failures are expected to be discovered at a higher rate over the first few months or so, the engineers will be expected to dedicate extra time in maintaining the system. With these factors in mind, the availability of the system is expected to have an average uptime of 85 to 90 percent for the first month.

2.3.3 Availability

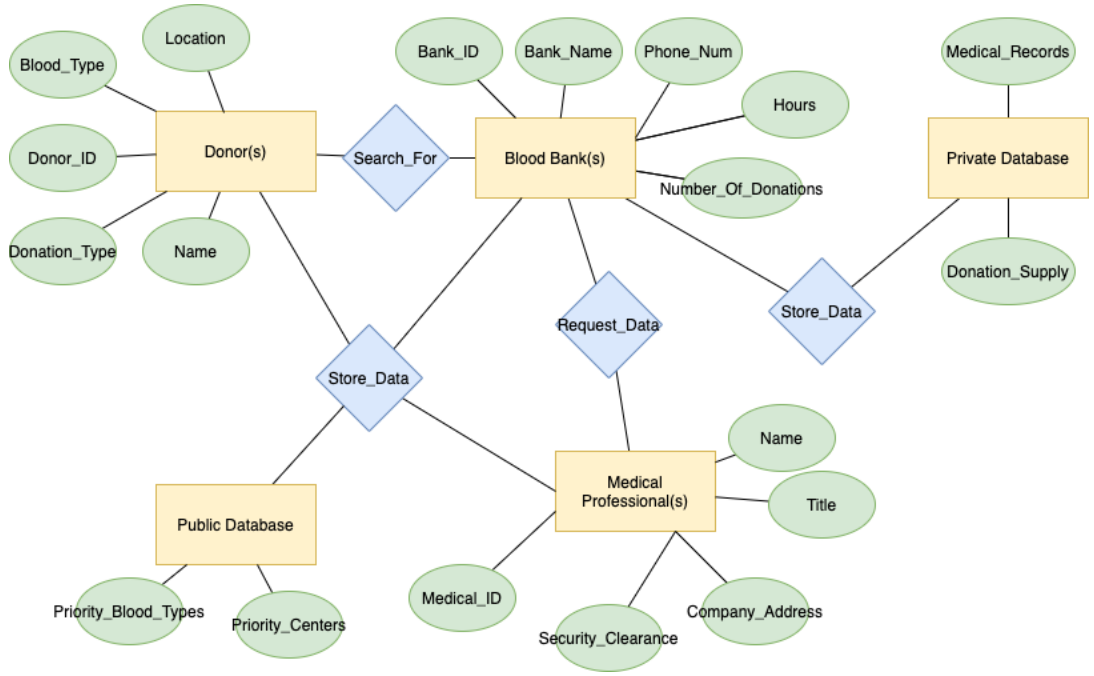
We expect the Cardinal system to have a high uptime of around 90% therefore it should be exceptionally available as it will be a web application that can be accessed from many different devices. With the proper checkpoint system, it should allow the user to relaunch with minimal data loss. As this is a web app that should be readily accessible, it will function around the clock until a scheduled maintenance frame. Proper testing will be set in place to ensure everything is up and running properly.

2.3.4 Security

In order to ensure security for our users, methods will be implemented to deter bad actors. Naturally, abstraction will be the most fundamental part. Users will be able to search and filter for various blood banks and nothing more. In addition to this, engineers will be proactive and do their best to find and fix potential security risks before breaches happen. To ensure that there are no unnecessary data breaches, data for medical professionals and blood banks will be stored on separate databases. This allows the data that is relevant to each to be accessed on a case-by-case basis. Medical professionals of varying titles will have varying access depending on the jurisdiction of their licensure. All of this will be done as well as:

- Specific classes and attributes will be set to private or protected
- Proper abstraction will be implemented
- There will be vigurous testing of the system

2.4 Database Requirements



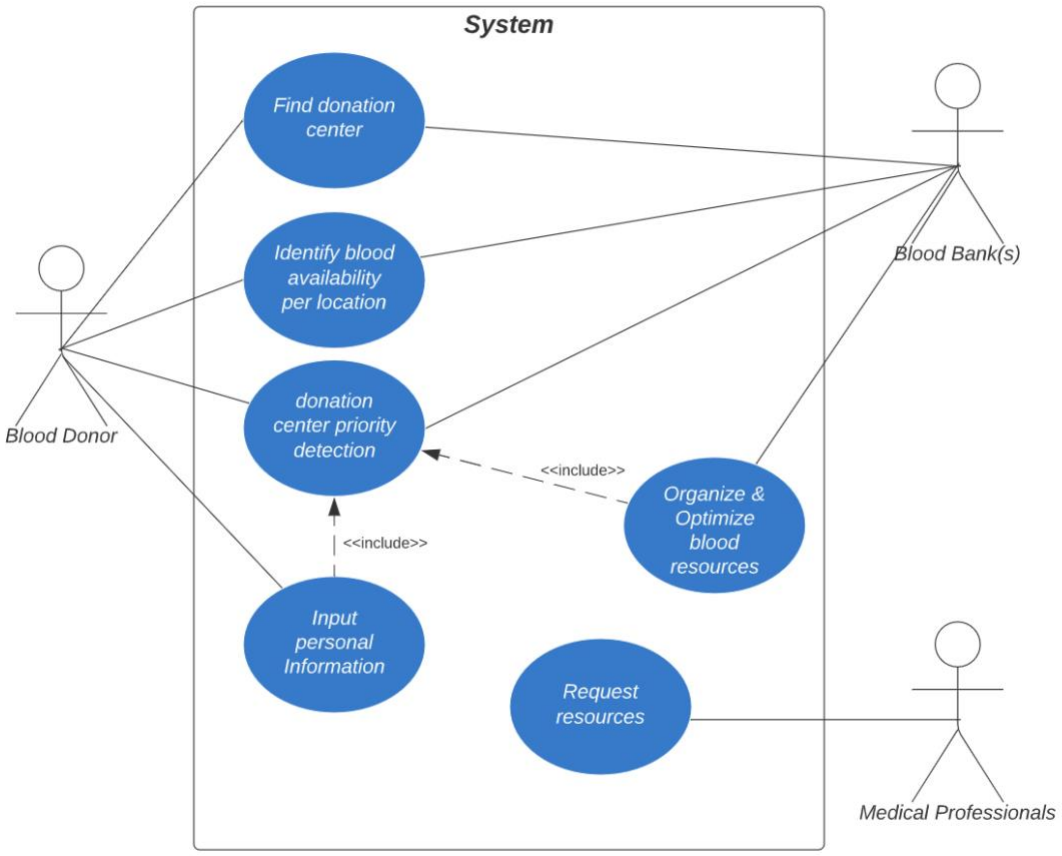
2.5 Design Constraints

Hardware limitations: after exploring similar software products, the software does seem to be able to run on machines with dual core processors and a 2GB RAM. Most of current computers used in health care facilities are above this requirement. These above average medical machines will make Cardinal accessible hardware-wise.

Software limitations: for the best user experience, we will require the OS to be Windows 7 and above to make sure our software runs on a system that receives periodic updates.

3. Use Case Models

3.1 Use Case Diagrams



3.2 Use Case Descriptions

Use Case Name	Find donation center
Use Case Description	will contain necessary information such as the nearest locations of donation centers, phone numbers, hours of operation, and blood volumes available

Pre-Condition	the user's current location or the zip code in which the user is looking to donate
Post-Condition	a map that dynamically updates to show the donation centers in a particular area
Scenarios	1) Enter current location 2) Location is validated 3) Location is queried against available centers 4) Display results on page

Use Case Name	Identify blood availability per location
Use Case Description	allows users and centers alike to know the blood availability of any participating center
Pre-Condition	The input needed for this activity is the data provided by a participating donation center on a backend-oriented database
Post-Condition	the correct data is displayed on an easy-to-read page
Scenarios	1) Choose blood type to query for 2) Information is validated 3) Query is compared to database 4) Database returns results

Use Case Name	Donation center priority detection
Use Case Description	features that will help individual users or organizations find the centers that require more donations of a specific type
Pre-Condition	The input required for this activity will be solely provided by the participating donation center's database inputs
Post-Condition	The output will be known to be functional once the information can be properly displayed on an easy-to-read page
Scenarios	1) Backend data is pulled 2) System compares blood supply data of one center vs another 3) Center with higher priority criteria is highlighted 4) Centers are sorted 5) Results are displayed

Use Case Name	Input personal information
Use Case Description	Common personal data is stored of each user
Pre-Condition	User's understand and enter personal data
Post-Condition	User information is stored and displayed
Scenarios	1) Enter personal information 2) Fields are validated 3) Valid fields are stored 4) Public data is displayed on profile Extensions: 2a) Invalid fields show error messages

Use Case Name	Organize and Optimize blood resources
Use Case Description	Organize a donation center's current resources
Pre-Condition	Any information the organizations deem crucial
Post-Condition	A webpage or data sets for participating centers to view and interact with
Scenarios	1) Blood center enters resource data 2) Resource data is validated 3) Resource data is stored 4) Data is sorted by importance 5) Public data is shared across secure network 6) Private data is stored behind private/encrypted database 7) Public data is displayed on webpage

Use Case Name	Request resources
Use Case Description	Request a donation center's current resources
Pre-Condition	Any information the organizations deem crucial
Post-Condition	A webpage or data sets for participating centers to view and interact with

Scenarios	1) Medical Professional enters resource data 2) Security clearance is validated 3) Resource data is validated 4) Resource data is retrieved 5) Data is sorted by importance 6) Public data is shared across secure network 7) Private data is retrieved from private/encrypted database 8) Data is displayed on secure webpage