Software Requirements Specification

for

<Roh>

Version 1.0 approved

Prepared by:

< Mai Lahlouh, Dunia Abedalziz, Hajar Al-hajeh, Eman Alyat >

<Arab American University>

<6/2/2023 >

Table of Contents

Table of Contents ii

Revision History iii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Project Scope 2

1.4 References 2

1.5 Overview 2

2. Overall Description 3

2.1 Product Perspective 3

2.2 Product Functions 4

2.3 User Classes and Characteristics 5

2.4 Operating Environment 5

2.5 Design and Implementation Constraints 5

2.6 User Documentation 6

2.7 Assumptions and Dependencies 6

3. System Features (Functional Requirements) 6

3.1 Users can create an account. 6

3.1.1 Description 6

3.1.2 Stimulus/Response Sequences 6

3.1.3 Functional Requirements 7

3.2 Request for blood: 7

3.2.1 Description: 7

3.2.2 Stimulus/Response Sequences: 7

3.2.3 Functional Requirements 7

4. Data Requirements 8

4.1 Logical Data Model 8

4.2 Reports 9

5. External Interface Requirements 10

5.1 User Interfaces 10

5.2 Software Interfaces 13

6. Quality Attributes 14

6.1 Usability 14

6.2 Performance 14

6.3 Security 15

6.4 Safety 15

6.5 Maintainable: 15

6.6 Availability : 16

6.7 Scalability : 16

Appendix A: Glossary 16

Revision History

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

# Introduction

The software requirements specification is an official document defines what the system will do rather than how. It includes user and system requirements for all features to all stakeholders in the area of blood donation.

## Purpose

The SRS describe the functional and nonfunctional requirements for revision 1.0 of the Roh application. The purpose is to develop a software product, Web and Mobile applications. It is a Blood Donation App to help the donation process. This document will be used by Ministry of Health and Hospitals Managers to approve if it is applicable in our country or not. Marketing staff will use the SRS, since without marketing our application will not be popular and number of donors will not increase. Also used by developers and testers to help them understand the high priority requirements.

## Document Conventions

|  |  |
| --- | --- |
| **App** | Roh Application |
| **User** | Ministry of Health, Hospital, Blood Bank Managers, Donors. |
| **Stakeholder** | Blood Bank, Donors. |
| **Database** | Group of objects have specific proprieties and methods linked by relations. |
| **SRS** | Software Requirements Specification. A full, structured and complete document of what are the expected features of the system to do. |
| **GPS** | Global Positioning System, the fastest and most accurate way to determine the location. |
| **CSC** | Blood Cell Count, to discover if there is a disease such as Anemia |
| **HTML** | Hypertext Markup Language is used to structure a web page. |
| **MVC** | Model View Controller : design pattern that commonly used to implement user interfaces, data, and controlling logic. |
| **API** | application programming interface is a way for two or more computer programs to communicate with each other. |

## Project Scope

The product is a web or mobile app provides a fast communication between blood banks in hospital and donors. We are facing overhead time in spreading out the need of blood units. By our app we aim to reduce the gap between them to make a fast donation, especially in emergency events. To achieve it we send notifications to the most closet donors in the surrounded region. Also encourage people to donate.

## References

1. IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.
2. Wateen Application : <https://wateenapp.org/>
3. Interface to blood donation app : <https://dribbble.com/shots/7423676-Blood-donation-appconcept?fbclid=IwAR2UHWv5q5HodA8jZh26nRzAYrdhIjRUQoENR4WypYpoOSddQeBBb8XqqBQ>
4. Using article to understand how the operation will be done and restrictions : <https://www.mayoclinic.org/ar/tests-procedures/blood-donation/about/pac-20385144?fbclid=IwAR2rzZZRXgjkmIbGSZsgCe2Hxyh7A9D9nTuxvwroWHWz6HtEgJDrd27B36c>
5. Blood Bank SRS : <https://www.studocu.com/in/document/national-institute-of-technology-rourkela/software-engineering/online-blood-bank-manamgement-system-srs-report/6935838?fbclid=IwAR3OoXrJUyM4o0n0z5z1U8abrpNlxJHTAf89TqJWX4zm2y4iYgNthqe0sNY>

## Overview

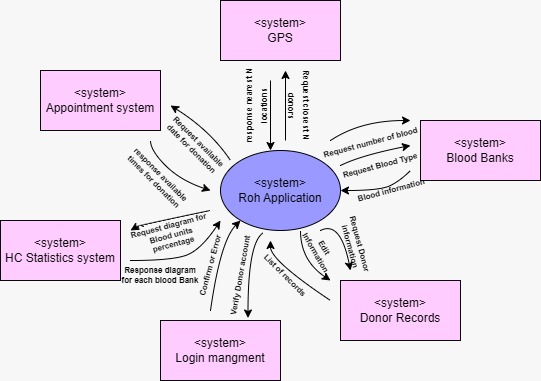
SRS document consists of eight chapters. We will write three chapters. The first is introduction, second is overall description and the third is system features. The second will talk about what are high level functionalities and constraints according to the requirements of stakeholders and environment of implementation. The third chapter will be useful for developers written in a detailed language and notations of functional and nonfunctional requirements. It defines the features needed to be in implementation.

# Overall Description

This section will give an overview of the system. It will be explained to show how the system will work and define the abstract functionality of it. It will describe all the stakeholders who will access to the system and authorization for any user.

## Product Perspective

Our product is a replacement for an application that are already exists in our country, but not as a cooperation with hospitals directly. Specially in emergency events. In our system that is consists of subsystems, we integrate with systems like GPS, appointment system, statistics system and hospital’s blood bank. They are interacted together to accomplish fast donation process.



## Product Functions

## 

|  |  |
| --- | --- |
| **FE-1** | Each of donors and hospitals need to login into the system |
| **FE-2** | The hospital can check the blood bank to select number of required units and blood type |
| **FE-3** | The hospital’s blood bank determine number of needed donors |
| **FE-4** | The hospital can check addresses and the profile of donors |
| **FE-5** | When required the hospital will send notifications to non-block donors |
| **FE-6** | The donor confirm or not the received message |
| **FE-7** | When donor confirm for blood donation message his information will be updated in hospital and the App database to be blocked until 2 months |
| **FE-8** | GPS return the most n of nearest locations. |

## User Classes and Characteristics

|  |  |
| --- | --- |
| **User Class** | **Description** |
| **Donors** | This class will contain username, password, address of the donors and his health information (blood type, CBC, diseases). Each donor receives notifications according to his location on GPS. Each donor can confirm or not, and give feedback to his experiment. |
| **Hospital’s blood bank** | This class will contain hospital’s blood bank. Each class can send the notifications in emergency events by determine the minimum number of blood units and blood type. Manage the date time of donations |
| **Guest** | Enter the website and view the blogs posted by the hospitals and read the feedback of donors. But receiving notifications just when register. |

## Operating Environment

|  |  |
| --- | --- |
| **OE-1:** | The Roh application shall operate on MACOS, Windows and Linux operating systems. |
| **OE-2:** | The Roh application shall operate on cloud server, database server, and web server |
| **OE-3:** | Web Browsers: Google Chrome, Microsoft Edge, Internet Explorer, Mozilla Firefox. By using the React framework to develop the static web page, and Laravel 9 for developing dynamic web page. |
| **OE-4:** | The Roh application shall operate on mobile operating systems (IOS and Android) by using Flutter framework |

## Design and Implementation Constraints

|  |  |
| --- | --- |
| **CO-1:** | The system design pattern is MVC. |
| **CO-2:** | The system shall use the current corporate standard Oracle database engine. |
| **CO-3:** | All HTML code shall conform to the HTML 5.0 standard. |
| **CO-4:** | All scripts shall be written in Dart. |
| **CO-5:** | The system will integrate with hospital’s database and GPS area measure using API. |

## User Documentation

|  |  |
| --- | --- |
| **UD -1:** | In first time user access the app, The system shall give alert to enter his information. |
| **UD -2:** | The system shall provide a tutorial when a notification it receives and how to accept or reject this message. |
| **UD -3:** | The system shall provide a training how the hospital can send notification to the donors according to donor’s profile and their location. |

## Assumptions and Dependencies

|  |  |
| --- | --- |
| **AS -1:** | The application runs 24/7, then the user should be able to access the services all time without interruption. |
| **AS -2:** | The application can adapt to changing demands, such as an increase in the number of donors, without a significant decrease in performance. |
| **DE -1:** | The application should be flexible with changes in user’s location. |
| **DE -2:** | The application should make updates in donor’s information according to hospital database. |

# System Features (Functional Requirements)

## Users can create an account.

### Description

Each of donor and hospital can register to create account. It is a high priority feature.

### Stimulus/Response Sequences

Stimulus: The user sends a request to create account.

Response: the system shows two choices to select. If the selection is a hospital the next interface will be different from the donor choice.

Stimulus: Each interface requests to enter information about the two users. The donor enters his health informatics and allow for connecting with GPS. Also, the hospital connects with its blood bank database.

Response: Two different pages show to each user with different authorization.

### Functional Requirements

1. The system shall provide two different register forms.
2. The system shall validate the entered data.
3. The system shall insert their records in to the database.

### Detailed Use cases description:

|  |  |
| --- | --- |
| **Use Case ID:** | **UC-1** |
| Use case Name: | create an account. |
| Actors: | Donors and Hospital’s blood bank. |
| Description: | Any user with certain conditions can create an account and be a donor.  Any hospital’s blood bank can create an account. |
| Preconditions: | 1. The user does not have an account. 2. Users with correct validation can be a donor. 3. The hospital must be in Palestine to register in our application. |
| Postconditions: | 1. Each of hospital are login on the application and redirect to specific interface. 2. The donors can receive notifications. 3. The patients’ records integrated with hospital database. 4. The hospital can send notification. 5. The hospital can manage patients’ records. |
| Normal Course: | 1. display the home page for the application. 2. when the user chooses to register as a donor, he will be redirected to answer some questions. 3. If his answers apply to the criteria he can register by his identity and a password. 4. When the hospital’s blood bank wants to register it must register with identical number match to its number in ministry of health. 5. If registration is approved the blood bank can access the donors’ database. |
| Alternative course: | There is no alternative courses. |
| Exceptions: | 1. If the donor’s information is not as required, a message to retry registration later is shown. 2. If the blood bank identity does not match with ministry of health a message is shown to confirm identity. |
| Frequency of Use: | Approximately 1000 users, average of one usage per hour. |
| Special Requirements: | donor shall be able to change his address. |
| Assumptions: | Internet Connection: Assumption is made that the user has a stable internet connection to access the register page and submit their information. |
| Notes and Issues: | *The registration 24/7, then the user should be able to register at any time.* |

## Request for blood:

### Description:

When the hospital needs more blood units. It sends notification for available donors.

### Stimulus/Response Sequences:

Stimulus: Check the number of blood bank units is needed.

Response: Return the numbers for each blood group.

Stimulus: select the number of needed donor.

Response: The system sends this number to GPS to return the nearest donors for sending

message for them.

Stimulus: when time limit is out and the number of confirming not as required then the

blood bank sends another request for other donors.

Response: the system returns the previous for farther locations.

### Functional Requirements

1-*The system shall receive the number of donors and send this number to GPS system.*

*2-The system shall send notification for returned locations from GPS (by default distance is 50Km).*

*3-If the number of confirmed donors are not as required the system shall resend request to GPS*

*for farther distance to complete unit blood number.*

*4-The system shall update the blood bank database when the donor coming and donating.*

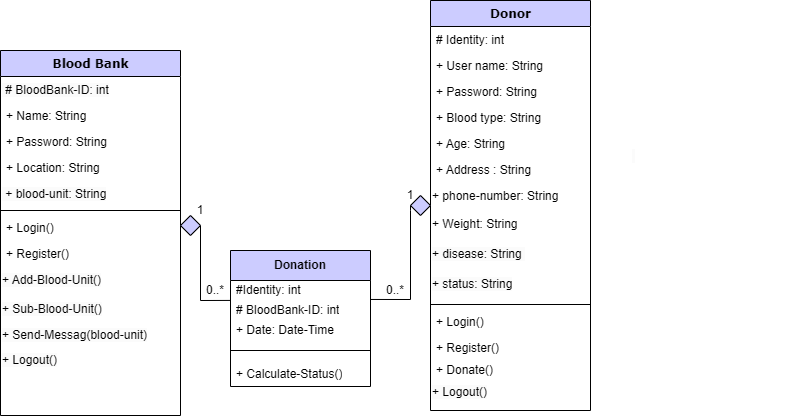
*5-The system shall update the donor database to block him for 60 days.*

### Detailed Use cases description:

|  |  |
| --- | --- |
| **Use Case ID:** | **UC-2** |
| Use case Name: | Request for blood |
| Actors: | Donors, Hospital’s blood bank and GPS. |
| Description: | When the hospital’s blood bank need blood units it sends notifications to all available donors in the database. |
| Preconditions: | 1. Check the blood stuck. 2. If the number of blood units less than minimum number of units. |
| Postconditions: | 1. Send the number of required donors to GPS. 2. GPS specify the number of nearest donors and their locations. |
| Normal Course: | 1. The hospital sends notification to the returned donors. 2. The donors receive these notifications. 3. The donors can either approve or reject this message. |
| Alternative course: | There is no alternative courses. |
| Exceptions: | If the GPS is not active, sending messages depends on address not the actual locations. |
| Frequency of Use: | Approximately 1000 notifications at the same time. |
| Special Requirements: | In emergency events the application sends notifications for the most regular donors and donors who live in highly populated areas. |
| Assumptions: | If the number of approving of notifications less than the required another request is send to GPS. |
| Notes and Issues: | *Sending messages 24/7, then the user should be able to approve at any time.* |

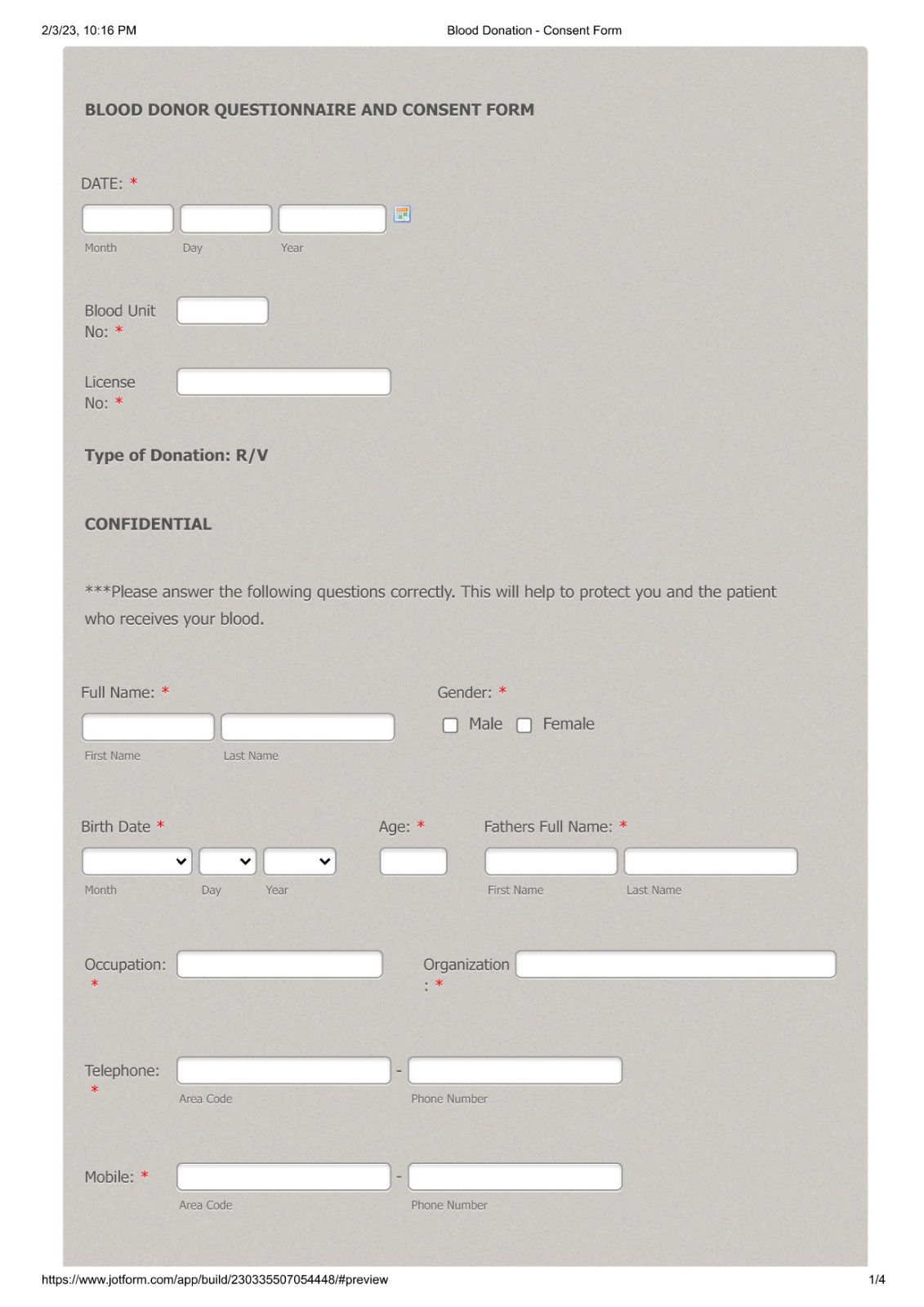
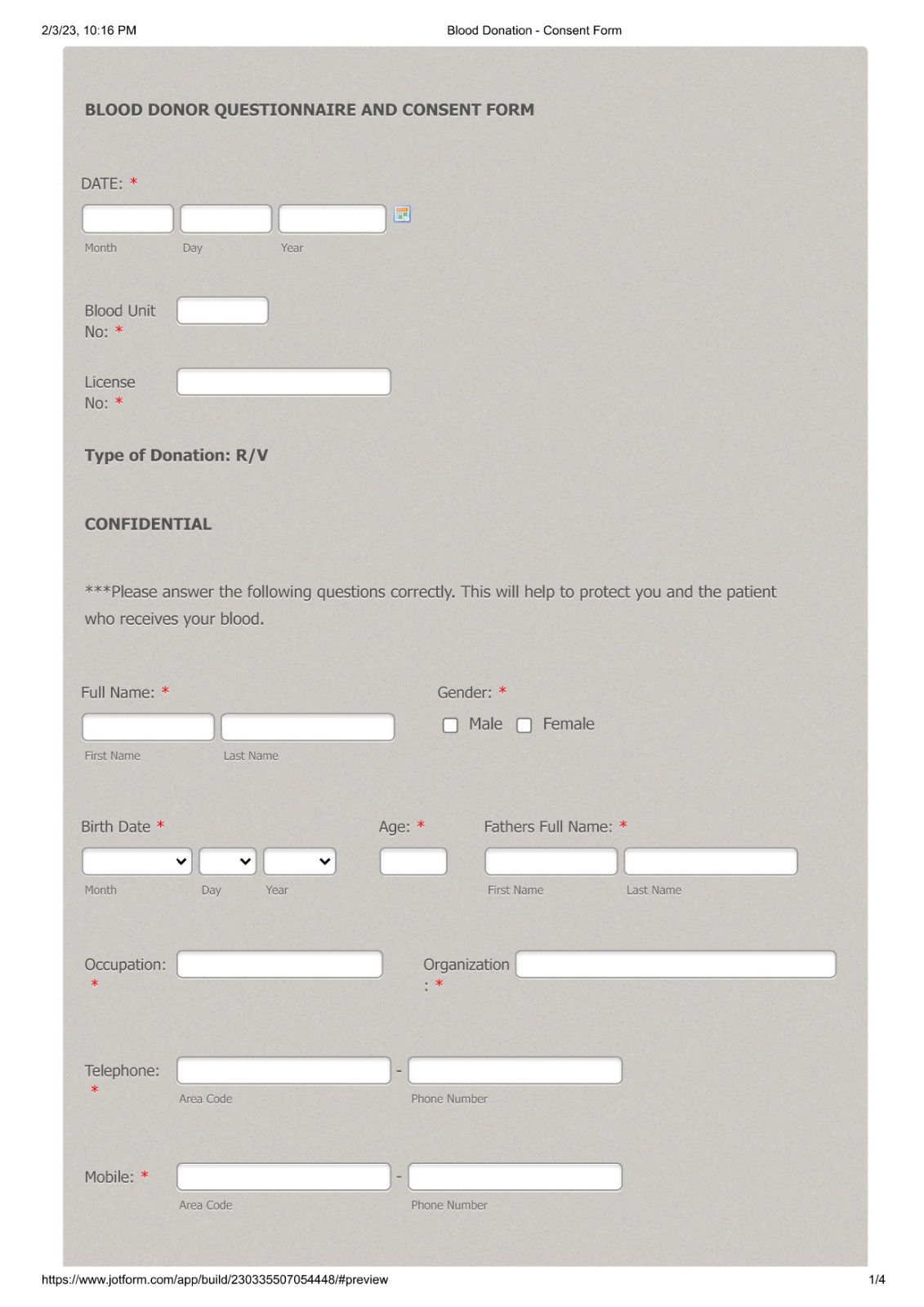
# Data Requirements

## Logical Data Model



## Reports

Every time the donor donates, our application makes a report for documentation.



# External Interface Requirements

## User Interfaces

|  |  |
| --- | --- |
| **UI-1:** | Home Page for the project. |
| **UI-2:** | To allow the user to login as a donor or hospital blood bank. |
| **UI-2.1:** | To allow the user to register as a donor or hospital blood bank if they haven’t an account. |
| **UI-3:** | For donor, to review percentage for each blood bank in the app, and blogs. |
| **UI-3.1** | To review form for appropriate date for donation. |
| **UI-3.2** | Interface for approving the message. |
| **UI-4** | Interface to choose the number of needed blood unit and which type with a button to request send . |
| **UI-4.1** | To review the list of all patient for managing their certain information. |
| **UI-5:** | The system shall provide a help link from each displayed HTML page to explain how to use that page. |

# 

# 

# 

## 

## 

## Software Interfaces

|  |  |
| --- | --- |
| **SI-1:** | **Blood Bank database** |
| **SI-1.1:** | The Roh app request number and type of blood unit. |
| **SI-2:** | **GPS system** |
| **SI-2.1:** | To request the nearest number of donors. |
| **SI-2.2:** | To check the distance between any hospital and donor. |
| **SI-2.3:** | To determine the number of donor activating the GPS. |
| **SI-3:** | **Appointment system** |
| **SI-3.1:** | To check the list of available date in blood bank. |
| **SI-3.2:** | To allow the donor to set an appropriate date. |
| **SI-4:** | **Database Interface** |
| **SI-4.1:** | To add any donor which want to join with the database. |
| **SI-4.2:** | To allow donor modify his health informatics. |
| **SI-4.3** | To allow hospital to modify donors profile. |
| **SI-4.4:** | The Roh app can modify the state of donor automatically when he confirm donation process. |
| **SI-5:** | **HC statistics system** |
| **SI-5.1:** | To show a diagram for donation percentage in a certain period. |

# Quality Attributes

## Usability

|  |  |
| --- | --- |
| **US-1:** | All buttons or links should have suitable name close to its function. |
| **US-2:** | For the first use the app alert in boxes for using the app. |
| **US-3:** | When the user create account a placeholder show example for the input data. |
| **US-4:** | The notification message can be accepted outside Roh app even though if he using another applications. |
| **US-5** | The application shown in Arabic or English languages. |
| **US-6** | A user-friendly interface that allows users to easily find and access the information they need. |
| **US-7** | Presenting information in a clear and organized way. |

## Performance:

|  |  |
| --- | --- |
| **PE-1:** | The system shall send notification to 1000 donor in the same time response. |
| **PE-2:** | The system shall move between mobile pages rapidly, not more one second. |
| **PE-3:** | The system delays are less. |
| **PE-4:** | The system shall perform tasks efficiently since we use less number of instructions for given tasks and optimized query to develop the system. |

## 

## Security

|  |  |
| --- | --- |
| **SE-1:** | The password for donor and hospital must be strong and encrypted. |
| **SE-2:** | Just the registered hospital can access health and location information for the donors. |
| **SE-3:** | The system shall prevent any donor to access another account. |
| **SE-4:** | The hospital only can edit the number of blood units in its own blood bank. |
| **SE-5** | The system must have robust user authentication mechanisms, such as multi-factor authentication, to ensure that only authorized users can access sensitive information. |
| **SE-6:** | The system use SSL in all transaction and the delay involved are less. |

## Safety

|  |  |
| --- | --- |
| **SA-1:** | The backup strategies are used to save information, so if the system damage, the data will not be expired. |
| **SA-2:** | The system must have redundant components, such as backup power supplies and redundant servers, to ensure high availability and minimize the risk of system failure. |
| **SA-3** | Providing mechanisms to prevent errors from occurring and always have a backup plan. |

## Maintainable:

|  |  |
| --- | --- |
| **MA-1:** | Following a standard design pattern (MVC). |
| **MA-2:** | Follow the clean code rules use human readable and sensible names. |

## Availability:

The system must be operational and available for use 24/7 to ensure that blood supplies can be quickly and efficiently distributed when needed.

## Scalability:

|  |  |
| --- | --- |
| **SC-1:** | Following a standard design pattern (MVC). |
| **SC-2:** | Keep monitoring the system Plan for the expected growth. |
| **SC-3:** | An API-based architecture allows different components of the system to communicate and exchange data, improving the scalability and flexibility of the system. |

Appendix A: Glossary

|  |  |
| --- | --- |
| App | Roh Application |
| User | Ministry of Health, Hospital, Blood Bank Managers, Donors. |
| Stakeholder | Blood Bank, Donors. |
| Database | Group of objects have specific proprieties and methods linked by relations. |
| SRS | Software Requirements Specification. A full, structured and complete document of what are the expected features of the system to do. |
| GPS | Global Positioning System, the fastest and most accurate way to determine the location. |
| CSC | Blood Cell Count, to discover if there is a disease such as Anemia |
| HTML | Hypertext Markup Language is used to structure a web page. |
| MVC | Model View Controller : design pattern that commonly used to implement user interfaces, data, and controlling logic. |
| API | application programming interface (API) is a way for two or more computer programs to communicate with each other. |
| SSL | 1. Secure Sockets Layer, a computing protocol that ensures the security of data sent via the internet by using encryption. |

<https://docs.google.com/forms/d/e/1FAIpQLSf3hYCf--4CoaxE5Sh1TYQr9WCiq5ljyzLzGP-0D8jT0OgMyQ/viewform>

## 

## 