Software Requirements Specification

For

Online Blood Bank System

Prepared by:

16th January, 2016

Table of Contents

Table of Contents

1. Introduction 3

1.1 Purpose 3

1.2 Intended Audience and Reading Suggestions 3

1.3 Product Scope 3

1.4 References 4

2. Overall Description 4

2.1 Product Perspective 4

2.2 Product Functions 6

2.3 User Classes and Characteristics 9

2.4 Operating Environment 9

2.5 User Documentation 9

2.6 Assumptions and Dependencie

3. System Features 10

3.1 System Feature 1 10

3.2 System Feature 2 (and so on)……………………………………………………………………………………………………11

3.3 Functional Requirements…………………………………………………………………………………………………….……11

4. Other Nonfunctional Requirements 12

4.1 Performance Requirements 12

4.2 Security Requirements 12

4.3 Software Quality Attributes 13

5. Other Requirements 13

Appendix A: Glossary 13

Appendix B: Analysis Models 14

# Introduction

## Purpose

The purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals. This document describes the project's target audience and its user interface, hardware and software requirements. It defines how our client, team and audience see the product and its functionality.

## Intended Audience and Reading Suggestions

This document is intended for, such as developers who follows the guidelines according to use case diagrams and develop the modules, project managers use this SRS for checking project milestones and deadlines, users to verify the functional requirements and approve the project according to mentioned criteria, testers for testing purpose such as alpha testing, integration testing, etc. So these are the audience for the srs and in this document product scope, purpose, project description and all the references are mentioned which is useful for all the audience. Sequence of reading the document begins with purpose and product scope, then references from where we take some knowledge, then overall description of project, then interface requirements, then system features and non-functional requirements and then mentioning the customer of our project.

## Product Scope

The online blood bank system is a management system. This project is made to manage all the registered blood banks under one unit. Our project is intended to help the people in case of emergency where any registered user can view information about the availability of various blood groups in various blood banks.

The information for particular blood banks will be managed online.

1. It will be accessed only by internet.
2. It automatically processes the blood information.
3. It is a browser based system.

## References

1. <http://en.wikipedia.org/wiki/SoftwareEngg>.

2. <http://ieee.org>

3. <http://www.slideshare.net/udaschand940/software-requirementspecification-srs2>

# Overall Description

## Product Perspective

The online blood bank system software is the replacement of existing websites which have some limitations such as:

1. Information about individual bank.
2. Not secured.
3. Not much user friendly.

In this Project we can collect an information about the donor, receiver and camps that are related to donating, receiving the blood. Through this website any person who is interested in donating the blood can register himself and in the same way if any user wants to add a camp, can host it. Moreover if any user wants to make request for blood online, he can also take the help of this site. Admin is the main authority who can do addition, deletion, and modification if required.

This project has been divided into 3 main modules:

1. User module
2. Admin module
3. Banker module

**1. User**

The new user can register itself to site by filling the registration form. The user can register as a donor if he wants to donate blood. If user wants to receive blood, he can register as a receiver also. By registering as receiver the user can take the needed blood from the bank where it is available. If any user wants to organise any camp he can organise or host a new camp.

**2. Admin**

Admin manages all the data of users. Admin can view the registered users, view donors, view receivers. Admin also inserts and manages the blood banks. All the blood details can also be viewed.

Admin can also insert and manage camps. Admin can also view the feedbacks.

**3. Banker**

By login with bank userid , banker can view blood which is available in that particular bank. Banker can also add new blood in his bank. Banker can also view the information of person who has received blood from his bank.

In addition to the three modules, a **Central database server** and a **Backup database server** will also be used in order to read/write data onto the repository. The central database server will periodically update the Backup database server so that in case of server failure it can restore the data by retrieving the records stored in the Backup database server’s tables.

## Product Functions

## It must be able to perform Identification of each user by using login id and password.

## User: whenever user login is identified then it must direct them to their corresponding pages such as Register as Donor, Register as Receiver, Insert new camp etc.

* Banker: whenever banker login is identified it will direct them to the corresponding pages such as View Blood amount, Add new blood amount, View received blood information etc.
* Admin: whenever admin login is identified it will direct them to their corresponding pages such as Insert and manage camp, Insert and manage bank, View Users etc.

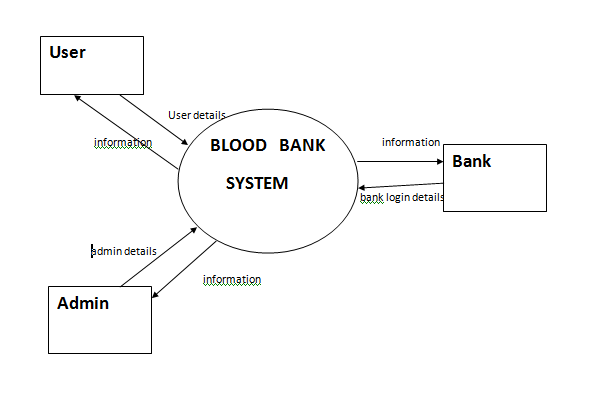
1. The banker must be able to view and update information about his bank.
2. The users must be able to register and view all the information about different blood banks.
3. The admin must be able to update, delete and insert all the new information regarding the blood banks.

**DATA FLOW DIAGRAMS**

**DFD level 0**

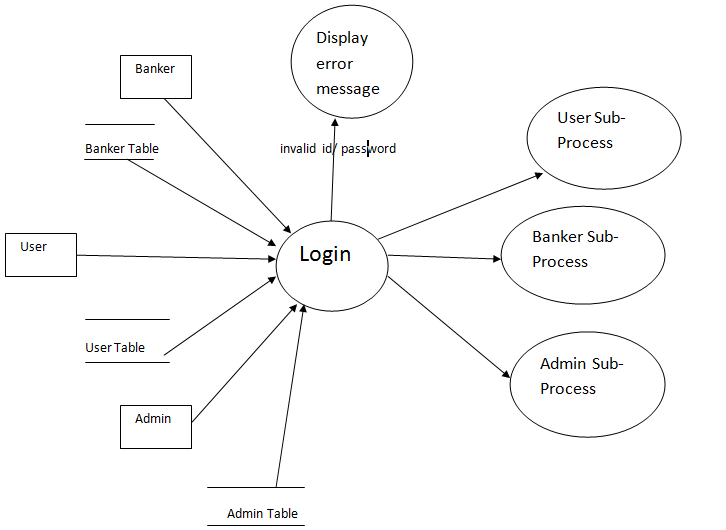
There are three types of users of the system:

* User
* Banker
* Administrator



Each type of user accesses the system in a different manner. The interactions with the system of the users are described in next level DFDs.

**DFD level 1**



This level of DFDs details the abstract 0-level DFD by showing the major sub processes in the system. One process that every user has to go through is the Login process. Every type of user has to login before the system can be accessed. The Login process works in the following manner:

* The different types of users provide their login ID/password and role to the process. The different roles are: User, Banker, and Administrator.
* The process verifies the credentials provided by checking it against the various tables stored for the users.
* If the credentials are verification, then the user is redirected to his/her homepage. The subsequent interactions of the users are handled by their specific sub processes as shown in the figure.
* If the verification fails, an error message is displayed. A user cannot use the system before logging in. So, the user has to use the Login process again to be able to use the system.

## 2.3 User Classes and Characteristics

In our system we have only three user classes which are as:

1. Admin

2. User

3. Banker

From above listed category of user we can see that all are from technical background or we can say that they have some knowledge of computer uses. Each user will use the product for different purpose. Thus while designing the software one can assume that each user type has the following characteristics:

* The user is computer-literate and has little or no difficulty in using this product.

## 2.4 Operating Environment

* Front end – php, java script, html,css
* Operating System – Windows XP/Vista/7/8
* Server –Wamp Server
* Back end – mySQL
* Ram – 512 mb
* Hard Disk – 50 gb

## 2.5 User Documentation

## This software provides security. The login form prevents the system from being misused by unauthorized users. Only an authorized operator will be granted rights to modify as per requirements. This software is also reliable and fault tolerant. The system developed is designed to handle invalid inputs. Since reliability is major area of concern the system has a backup to avoid data loss. The user should know the programming language very well that is used to develop software.

## 2.6 Assumptions and Dependencies and constraints

The following list presents the constraints, assumptions, dependencies or guidelines that are imposed upon implementation of the online blood bank system software:

* We assume that the Admin insert all the banks information based on the correct values obtained from banks. Users with administrator access should be careful in deleting or modifying any information knowingly or unknowingly which will lead to inconsistency of the database.
* The product must have a user friendly interface that is simple enough for all types of users to understand.
* Response time for loading the software and for processing a request should be no longer than five seconds.
* A general knowledge of basic computer skills is required to use the product.
* The central database server and backup database servers should be updated regularly. This updating and replication of data from central database server to the backup database server can introduce additional latency in the working of the system.

# 3. System Features

**3.1 Login**

**3.1.1 Description and Priority**

The login form is used by all the users, admin as well as bankers. This module has the highest priority when compared to all the other modules. This model allows the user to enter his username and password in order to make use of the software.

**3.1.2 Stimulus/Response Sequences**

This module has text boxes where the user can enter his username name and password. If the necessary information is not provided or if invalid inputs are given by the user then the system will pop a message box.

**3.1.3 Functional Requirements**

Only authorized users are allowed to login. The authorized users are the administrator, general users and the bankers. If invalid user name or password is given the system should inform the user. If unauthorized users try to access then it should not allow the user to work on the   
system.   
  
**3.2 Registration for donating and receiving blood**

**3.2.1 Description and Priority**

All users will register themselves within the time period and this feature has higher priority because this is the most essential feature of the product.

**3.2.2 Stimulus/Response Sequences**

Once the user registers himself/herself , one of the sub-flows is executed.

• “Register for Donating blood” sub flow  
• “Register for Receiving blood” sub flow

• “Organise camps” sub flow

**3.2.3 Functional Requirements**

In “Register for Donating blood”, once the user provides the requested information, the user will be able to donate the blood and an appropriate message is displayed.

In “Register for Receiving blood”, once the user selected this option the user will be forwarded to the ‘receiver registration’ page where he can fill the required details and submit the request for further processing.

In “Organise camps”, once the user selected this option, the user’s request will be forwarded to the ‘organize camp’ page and appropriate message is displayed.

**3.3 Data Entry module**

**3.3.1Description and Priority**

This module is used by admin who is responsible for entering the details of bank and camp. The module requests that the admin specify the function he/she would like to perform (either add a bank/camp, update a bank/camp, or delete a bank/camp details).

**3.3.2 Stimulus/Response Sequences**

Once the admin provides the requested information, one of the two sub-flows is executed.

* bank update(add a bank, delete a bank, update a bank)
* camp update(add a camp, delete a camp, update a camp)

In “Add a bank/camp” Once the admin provides the requested information, the bank/camp is added to the system and an appropriate message is displayed. In “Update a bank/camp” Once the admin updates the necessary information, the system updates the bank/camp record with the updated information In “Delete a bank/camp” Once the admin deletes the record, the system prompts the user to confirm the deletion of the student.

# Other Nonfunctional Requirements

## 4.1Performance Requirements

The system should be available 365/24/7 days, the performance should not be degraded with the increase in number of users.

## Security Requirements

The security of each user is provided with login id and password.

## Software Quality Attributes

* **Consistent uptime**
* This system will be able to stay up and running at least 98% of the time. Any downtime would be due to maintenance or upgrades. This downtime also includes any potential failures/crashes.
* **Load and concurrency**
* The system must be able to serve at least two thousand users concurrently without crashing.
* **Familiar Interface**
* The new system will have an interface that shares some of the feel of the old system so that users who are familiar with the old system will not have trouble adjusting to the new system.

# Other Requirements

Appendix A: Glossary

**Acronyms and Abbreviations:**

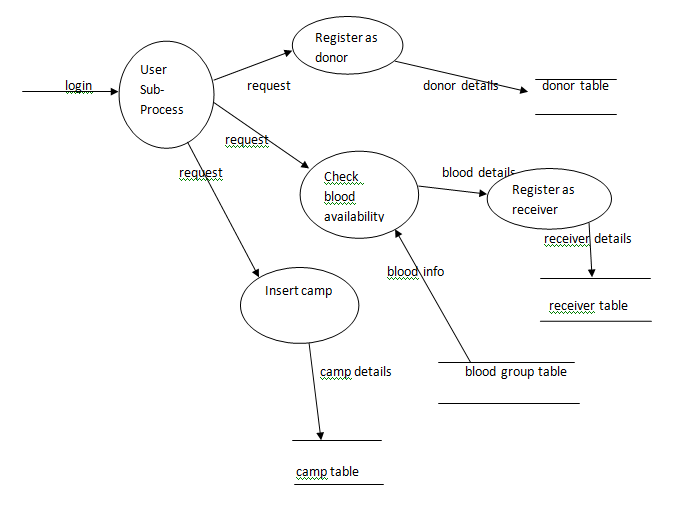
Admin – Administration

Info – Information

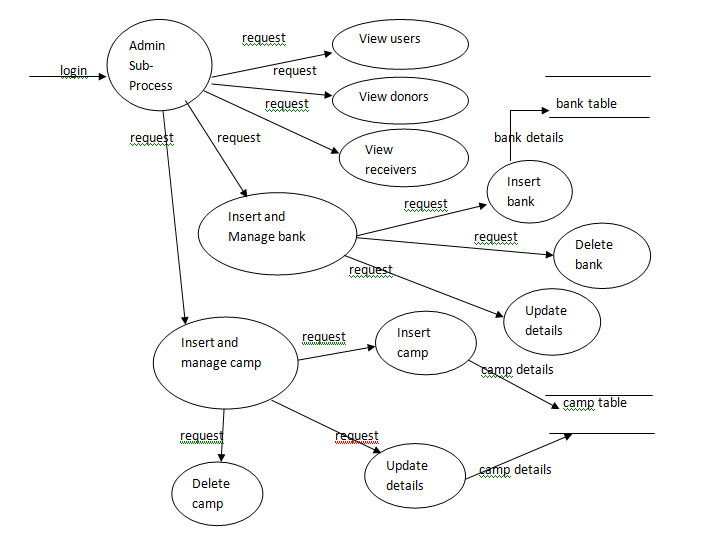
Appendix B: Analysis Models

**Data Flow Diagrams**

**DFD 1.1**



**DFD 1.2**



**DFD 1.3**

