

---

# **Software Requirements Specification**

**for**

## **Organ Donation and Procurement Network Management System**

**Version 1.0.0**

**Prepared by**

**GOBBRURI SHIVA**

**B210566CS**

**gobhuri\_b210566cs@nitc.a  
c.in**

**BANOTH BALAJI**

**B210572CS**

**Banoth\_b210572cs@nitc.a  
c.in**

**Instructor: DR.K A ABDUL NASEER**

**Course: Database Management Systems**

**Date: 21/10/2023**

# Contents

<b>CONTENTS.....</b>	<b>II</b>
<b>1 INTRODUCTION.....</b>	<b>1</b>
1.1 DOCUMENT PURPOSE.....	1
1.2 PRODUCT SCOPE.....	1
1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW.....	1
1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS.....	2
1.5 DOCUMENT CONVENTIONS.....	3
1.6 REFERENCES AND ACKNOWLEDGMENTS.....	3
<b>2 OVERALL DESCRIPTION.....</b>	<b>4</b>
2.1 PRODUCT OVERVIEW.....	4
2.2 PRODUCT FUNCTIONALITY.....	8
2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS.....	9
2.4 ASSUMPTIONS AND DEPENDENCIES.....	9
<b>3 SPECIFIC REQUIREMENTS.....</b>	<b>10</b>
3.1 EXTERNAL INTERFACE REQUIREMENTS.....	10
3.2 FUNCTIONAL REQUIREMENTS.....	11
3.3 USE CASE MODEL.....	13
<b>4 OTHER NON-FUNCTIONAL REQUIREMENTS.....</b>	<b>19</b>
4.1 PERFORMANCE REQUIREMENTS.....	19
4.2 SAFETY AND SECURITY REQUIREMENTS.....	20
4.3 SOFTWARE QUALITY ATTRIBUTES.....	20
<b>APPENDIX B - GROUP LOG.....</b>	<b>21</b>

# 1 Introduction

The "Organ Donation and Procurement Network Management System" (ODPNMS) is a cutting-edge software solution designed to streamline and enhance the management and coordination of organ transplantation. It addresses the critical challenges in organ donation and procurement by maintaining a comprehensive database of donors and recipients, facilitating efficient organ allocation, reducing organ wastage, and providing valuable statistical data to support government regulations and ultimately save more lives.

## 1.1 Document Purpose:

The purpose of this document is to delineate the software requirements for our "Organ Donation and Procurement Network Management System" (ODPNMS) project. ODPNMS is designed to streamline organ transplantation processes, facilitate efficient organ allocation, and reduce wastage. This Software Requirements Specification (SRS) document outlines the essential functionalities, features, and constraints of ODPNMS, serving as a comprehensive guide to understanding the project's objectives, scope, and database design for optimizing organ donation and procurement within the healthcare ecosystem.

## 1.2 Product Scope:

ODPNMS is introduced as a database management system that maintains various data related to organ donation and procurement, including personal information, medical histories, insurance details, allergies, and current organ needs. This database is essential for the efficient functioning of the organ donation and procurement network, offering statistical insights into the availability and demand for organs.

The project's goal is to provide a solution that aids in finding donors, optimizes organ allocation, and offers valuable statistical data for government regulations. It includes basic statistics from 2017 and 2018, highlighting the scale and importance of organ transplantation.

## 1.3 Intended Audience and Document Overview:

- **Developers:** This document provides technical requirements and insights for system development.
- **Project Managers:** Understand the project scope, objectives, and functional requirements for effective project management.

- **Healthcare Professionals:** Gain insights into ODPNMS's role in healthcare and organ transplantation processes.
- **Educators:** Comprehend the user interactions and system structure for educational purposes.
- **Government Organizations:** Understand compliance with regulations and data collection for policy development.
- **Client:** Align project expectations with objectives and requirements.
- **Professor:** Evaluate the document's alignment with academic objectives.

The document is organized as follows:

- **Introduction:** Provides an overview of the project's significance.
- **Product Scope:** Defines project objectives and boundaries.
- **Specific Requirements:** Details external interfaces, functional requirements, use cases, and performance, safety, and security requirements.
- **Software Quality Attributes:** Discusses important quality characteristics for the project.
- **Other Requirements:** Addresses additional requirements not covered elsewhere.
- **Appendix A - Data Dictionary:** Lists variables, states, inputs, and outputs.
- **Appendix B - Group Log:** Records minutes and activities of the project group.

Suggested Reading Sequence:

- **All Readers:** Begin with the "Introduction" to understand the project's context and significance.
- **Developers:** Focus on "Specific Requirements" to grasp technical and functional requirements.
- **Project Managers:** Start with "Product Scope" for project objectives and "Specific Requirements" for functional details.
- **Healthcare Professionals:** Review "Specific Requirements" for insights into ODPNMS's healthcare role.
- **Educators:** Explore "Specific Requirements" and "Software Quality Attributes" to understand user interactions and system quality.
- **Government Organizations:** Refer to "Specific Requirements" for compliance and data collection.
- **Client:** Start with the "Introduction" and "Product Scope" to align expectations with objectives.
- **Professor:** Review the entire document to evaluate its alignment with academic objectives.

## 1.4 Definitions, Acronyms and Abbreviations

- **ODPNMS:** Organ Donation and Procurement Network Management System
- **SRS:** Software Requirements Specification
- **DBMS:** Database Management System
- **THO Act:** Transplantation of Human Organs Act
- **HTML:** Hypertext Markup Language
- **CSS:** Cascading Style Sheets
- **API:** Application Programming Interface
- **UI:** User Interface
- **UX:** User Experience
- **URL:** Uniform Resource Locator
- **HTTP:** Hypertext Transfer Protocol
- **GUI:** Graphical User Interface
- **CPU:** Central Processing Unit

- **RAM:** Random Access Memory
- **GPS:** Global Positioning System

## **1.5 Document Conventions**

This document adheres to the following formatting conventions:

Font: Arial  
Font Size: 11  
Text Spacing: Single spaced  
Margins: 1-inch margins  
Section and Subsection Titles: Follow the template

## **1.6 References and Acknowledgments**

*[www.organindia.org](http://www.organindia.org)*

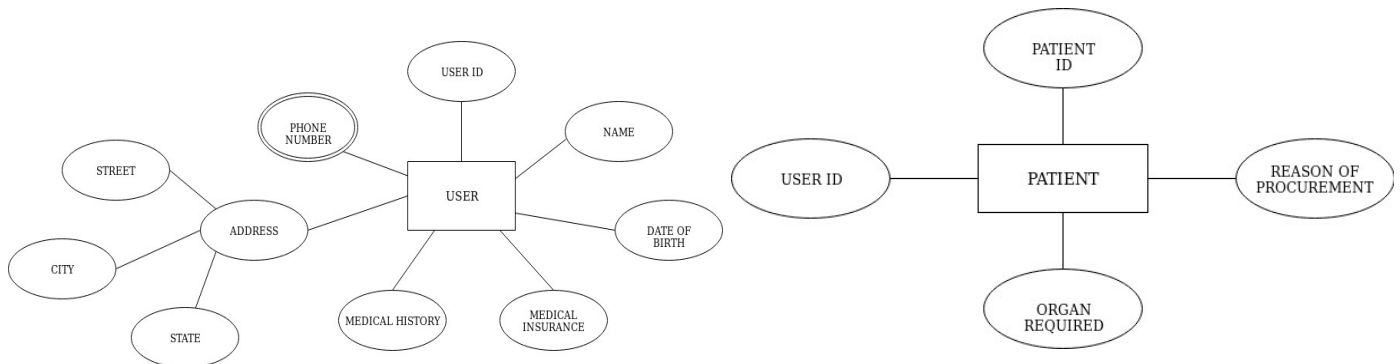
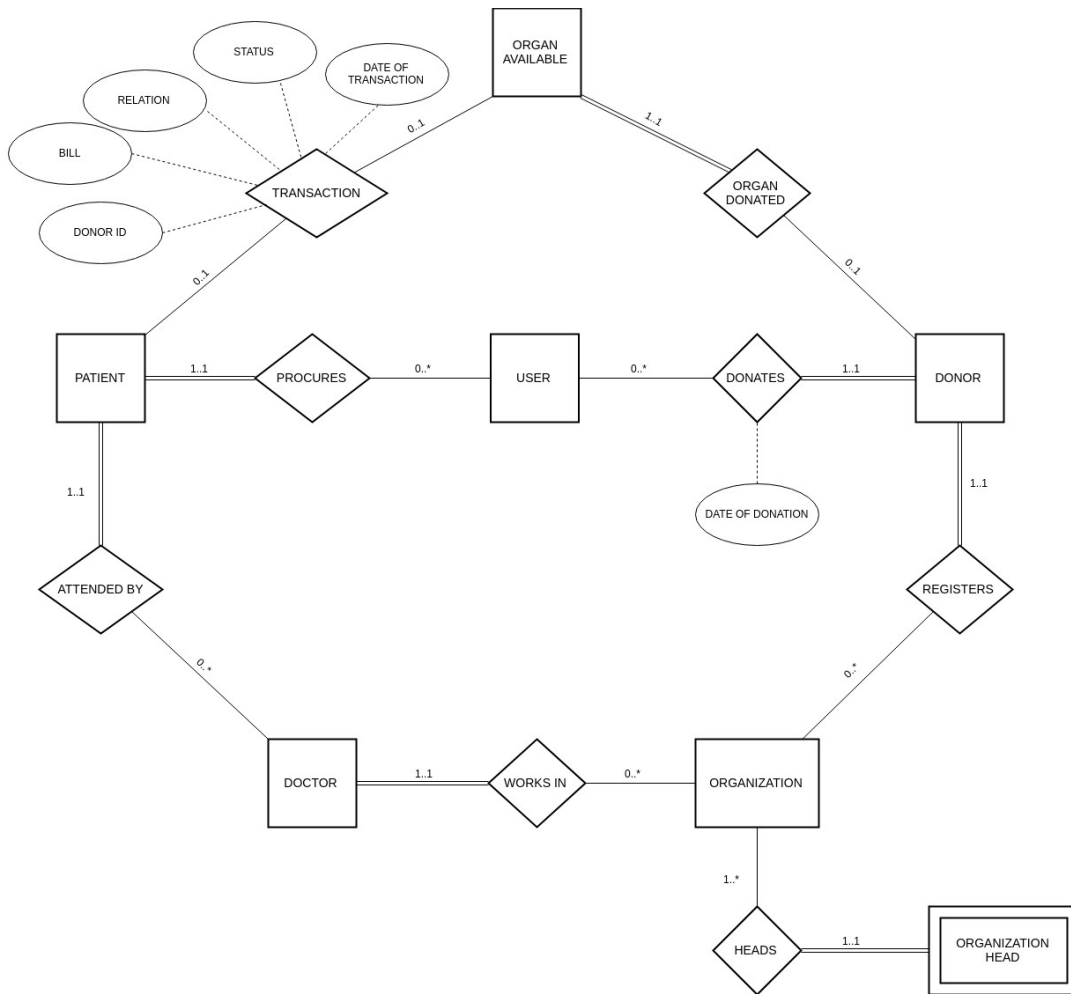
## 2 Overall Description

### 2.1 Product Overview

Basic Steps in Implementation :

- Every user has an account with can only be registered by a government certified hospital, which will keep all the information as defined in Problem Statement.
- Only Hospitals are eligible to request for a donation or procurement transaction.
- Government organizations will keep a watch on the pairing of donors and Patients and can approve a transplantation operation if all the rules are satisfied.
- Collecting Statistical Data through the history of Transplantation Transaction.

ODPNMS is developed to streamline and enhance the management and coordination of organ transplantation. It operates within the context of the healthcare ecosystem, specifically within government-certified hospitals and medical organizations. These institutions serve as the central entities responsible for managing user accounts, donor-patient pairings, and organ procurement transactions. The government organizations act as overseers, ensuring that transplantation operations adhere to predefined rules and regulations. Furthermore, ODPNMS collects statistical data on transplantation transactions, which is invaluable for regulatory purposes and optimizing organ allocation.







- Reason of Procurement
- User ID (foreign key)
- **Donor**
- Donor ID
- Organ Donated
- Reason of Donation
- User ID (foreign key)
- **Organ Available**
- Organ ID
- Organ Name
- Donor ID (foreign key)
- **Organization**
- Organization ID
- Organization Name
- Location
- Government-approved organization or not
- Phone Number (multi-valued)
- **Doctor**
- Doctor ID
- Doctor Name
- Phone Number (multi-valued)
- **Organization Head**
- Head Name
- Date of Joining
- Term Length

**Relationship Sets:**

- **Donates**
- Date of Donation
- The act of donation of an organ from a donor
- **Procures**
- The act of procuring an organ by the patient
- **Transaction**
- Date of Transaction
- Status (whether the surgery was successful or not)
- **Organ Donated**
- The organ donated by a donor, which is then stored in the Organ Available table.
- **Attended By**
- The transplantation performed by a doctor, involving procuring an organ from a donor and transplanting it to the patient by surgery.
- **Registers**
- Donor is registered in which organization.
- **Works In**
- The organization where the doctor works.
- **Headed By**
- The organization is headed by which person.

**Tables and their Functional Dependencies :-**

- **User Table**
- Attributes: User\_ID, Name, Date\_of\_Birth, Medical\_Insurance, Medical\_History, Street, City, State
- Functional Dependencies (FD):
- User\_ID → Name, Date\_of\_Birth, Medical\_Insurance, Medical\_History, Street, City, State
- **User\_phone\_no Table**

- Attributes: User\_ID, Phone\_no
- Functional Dependencies (FD):
- User\_ID → Phone\_no
- Note: User\_ID is a foreign key constraint.
- **Patient Table**
- Attributes: Patient\_ID, Organ\_Required, Reason\_of\_Procurement, Doctor\_ID, User\_ID
- Functional Dependencies (FD):
- Patient\_ID, Organ\_Required → Reason\_of\_Procurement, Doctor\_ID, User\_ID
- Notes: User\_ID and Doctor\_ID are foreign key constraints.
- **Donor Table**
- Attributes: Donor\_ID, Organ\_Donated, Reason\_of\_Donation, Organization\_ID, User\_ID
- Functional Dependencies (FD):
- Donor\_ID, Organ\_Donated → Reason\_of\_Donation, Organization\_ID, User\_ID
- Notes: User\_ID and Organization\_ID are foreign key constraints.
- **Organ\_Available Table**
- Attributes: Organ\_ID, Organ\_Name, Donor\_ID
- Functional Dependencies (FD):
- Organ\_ID → Organ\_Name, Donor\_ID
- Note: Donor\_ID is a foreign key constraint.
- **Transaction Table**
- Attributes: Patient\_ID, Organ\_ID, Donor\_ID, Date\_of\_Transaction, Status
- Functional Dependencies (FD):
- Patient\_ID, Organ\_ID → Donor\_ID, Date\_of\_Transaction, Status
- Notes: Patient\_ID and Donor\_ID are foreign key constraints.
- **Organization Table**
- Attributes: Organization\_ID, Organization\_Name, Location, Government\_Approved
- Functional Dependencies (FD):
- Organization\_ID → Organization\_Name, Location, Government-Approved
- **Organization\_phone\_no Table**
- Attributes: Organization\_ID, Phone\_no
- Functional Dependencies (FD):
- Organization\_ID → Phone\_no
- Note: Organization\_ID is a foreign key constraint.
- **Doctor Table**
- Attributes: Doctor\_ID, Doctor\_Name, Department\_Name, Organization\_ID
- Functional Dependencies (FD):
- Doctor\_ID → Doctor\_Name, Organization\_ID
- Note: Organization\_ID is a foreign key constraint.
- **Doctor\_phone\_no Table**
- Attributes: Doctor\_ID, Phone\_no
- Functional Dependencies (FD):
- Doctor\_ID → Phone\_no
- Note: Doctor\_ID is a foreign key constraint.
- **Organization\_Head Table**
- Attributes: Organization\_ID, Employee\_ID, Name, Date\_of\_Joining, Term\_Length
- Functional Dependencies (FD):
- Organization\_ID, Employee\_ID → Name, Date\_of\_Joining, Term-Length

## 2.2 Product Functionality

The "Organ Donation and Procurement Network Management System" (ODPNMS) is designed to perform several major functions to streamline organ transplantation and management processes. These functions include:

- **User Registration:** Allow government-certified hospitals to register and maintain user accounts with essential personal and medical information.
- **Donation and Procurement Request:** Enable hospitals to request organ donation and procurement transactions on behalf of patients.
- **Government Oversight:** Allow government organizations to monitor and approve transplantation operations, ensuring compliance with regulations.
- **Statistical Data Collection:** Collect and maintain historical data on organ transplantation transactions for decision-making and regulatory purposes.

## 2.3 Design and Implementation Constraints

- **Data Synchronization:** Any updates or additions to organ donation and procurement information, such as new donor or patient records, must be accurately and promptly synchronized with the MySQL database. This constraint ensures that the database reflects the most up-to-date and accurate information for real-time decision-making.
- **Authentication and Authorization:** The system must enforce strict authentication and authorization mechanisms to control access. Only users with valid credentials, such as hospital staff or authorized government personnel, will be allowed access to the system. This constraint is vital for data security and compliance.
- **Internet Connectivity:** ODPNMS relies on internet connectivity for real-time data exchange and access. Users, including hospitals and government organizations, need a stable internet connection to interact with the system. This constraint ensures that users can effectively use the system as long as they have internet access.
- **Hardware Limitations:** ODPNMS is designed to run on standard hardware configurations commonly found in healthcare facilities.
- **Integration Constraints:** The system may need to integrate with external systems or services, such as government databases for regulatory compliance. Developers will need to accommodate integration requirements and adhere to specified communication protocols.

## 2.4 Assumptions and Dependencies

### Assumptions:

- **Data Integrity :**Assumption that data related to organ donation and procurement transactions will be consistently and accurately stored in the database.
- **System Performance :**Assumption that the system meets specified requirements for efficient access to the database and overall performance.
- **User Language Proficiency :** Assumption that users are proficient in English, as the user interface is presented in English.

### Dependencies:

- **Hardware and Software Compliance :**The project's dependency on specific hardware and software that align with defined technical specifications and system requirements.
- **Administrative Oversight :**The project's dependency on qualified administrators who have a deep understanding of the system's operation and can manage any issues effectively.
- **Database Update :**The requirement for the database to be consistently updated with each organ donation and procurement transaction, ensuring data accuracy and integrity.

## 3 Specific Requirements

### 3.1 External Interface Requirements

#### 3.1.1 User Interfaces

- The login page serves as the primary entry point.
- Users, including government-certified hospitals, log in using their credentials (user ID and password).
- The interfaces feature a clean and organized design with user-friendly elements.
- Design considerations account for accessibility, including color-blindness.
- Intuitive instructions guide users through the various features.
- A consistent template ensures familiarity across different pages.
- Administrators have the ability to update the database with relevant data.
- Real-time updates are provided for transplantation statistics.
- The system ensures a seamless transition between different aspects of organ transplantation and management.

#### 3.1.2 Hardware Interfaces

- **Processor:** Dual-core processor (e.g., Intel Core i3 or equivalent)
- **Hard Disk:** 100GB of available storage space
- **RAM:** 4GB or more
- **Display:** Minimum 1366x768 resolution
- **Network Interface:** Ethernet or Wi-Fi for internet connectivity
- **Input Devices:** Keyboard and Mouse or equivalent input devices

### 3.1.3 Software Interfaces

SOFTWARE USED	DESCRIPTION
Operating system	<ul style="list-style-type: none"> <li>• <b>Linux (Ubuntu, CentOS, Debian, etc.):</b> Linux is a popular choice for web development due to its stability, security, and a wide range of development tools. Ubuntu, in particular, is known for its ease of use.</li> <li>• <b>Windows:</b> While less common, many developers use Windows for web development, especially if they are developing applications that will primarily run on Windows-based servers.</li> </ul>

- **MySQL (Version 8):** MySQL is utilized as the relational database management system to store and manage the project's data efficiently.
- **HTML 5:** HTML 5 is employed for structuring the content and creating the user interface of the web application.
- **Python:** Python serves as the primary programming language for the backend of the application, facilitating server-side logic and data processing.
- **Flask Framework:** The Flask framework is used to build and develop the web application on the server side. It aids in routing, handling requests, and managing the application's functionality.
- **CSS:** Cascading Style Sheets (CSS) are applied to enhance the visual appearance and layout of the web application, ensuring a user-friendly design.
- **Bootstrap:** Bootstrap is used to streamline and expedite the development of responsive, mobile-first web pages. It offers a range of pre-designed components and styles.
- **JavaScript:** JavaScript is employed to add interactivity and dynamic behavior to the application, improving the user experience and enabling real-time updates and data manipulation.

## 3.2 Functional Requirements

### Login Page Functions:

F1: The system shall display a login page that allows users to input their Username and Password.

F2: The system shall verify the entered Username and Password, granting access to the main page upon successful validation.

F3: The system shall deny access and display an error message if users enter invalid login credentials.

### Main Page Functions:

F4: The system shall present the main page after successful login, providing options for User, Search, Add, Update, Remove, and Statistics.

**Add Functionality:**

F5: The system shall allow users to add new users' information, including User ID, Name, Date of Birth, Medical Insurance, Medical History, and Address.

F6: The system shall enable users to add phone numbers to a user's profile by associating them with the User ID.

F7: The system shall permit users to add patient information, including Patient ID, required organ details, reason for procurement, and link them to a specific doctor and user.

F8: The system shall facilitate the addition of organizations by providing fields for Organization ID, Organization Name, Location, and Government Approval Status.

F9: The system shall allow users to add donor records, providing fields for Donor ID, donated organ details, reason for donation, and linking them to a specific organization and user.

F10: The system shall support the addition of phone numbers for organizations, associating them with the respective Organization ID.

F11: The system shall enable users to add organization heads by entering Head Name, Date of Joining, Term Length, and associating them with an organization.

F12: The system shall provide functionality to add doctor information, including Doctor ID, Doctor Name, Department Name, and linking them to an organization.

F13: The system shall allow users to add phone numbers for doctors, associating them with the specific Doctor ID.

F14: The system shall enable users to record transactions, including Patient ID, Organ ID, Donor ID, Date of Transaction, and a status indicator.

**Search Functionality:**

F15: The system shall permit users to search for specific users based on their User ID or other user details.

F16: The system shall allow users to search for patients using their Patient ID or specific patient details.

F17: The system shall provide search functionality for organs based on Organ ID or specific organ details.

F18: The system shall support searching for organizations based on Organization ID or specific organization details.

F19: The system shall allow users to search for organization heads based on their details.

F20: The system shall facilitate the search for doctor details based on Doctor ID or specific doctor information.

F21: The system shall allow users to search for transaction details based on transaction-specific information.

F22: The system shall enable users to search for log details based on specific log information.

**Update Functionality:**

F23: The system shall allow users to update user information, including Name, Date of Birth, Medical Insurance, Medical History, and Address.

F24: The system shall enable the update of doctor details, including Doctor Name, Department Name, and associated organization.

F25: The system shall support the modification of organization details, including Organization Name, Location, and Government Approval Status.

**Remove Functionality:**

F26: The system shall provide functionality to remove user accounts from the system.

F27: The system shall allow users to remove patient records from the system.

F28: The system shall permit users to remove donor records from the system.

F29: The system shall enable users to remove doctor records from the system.

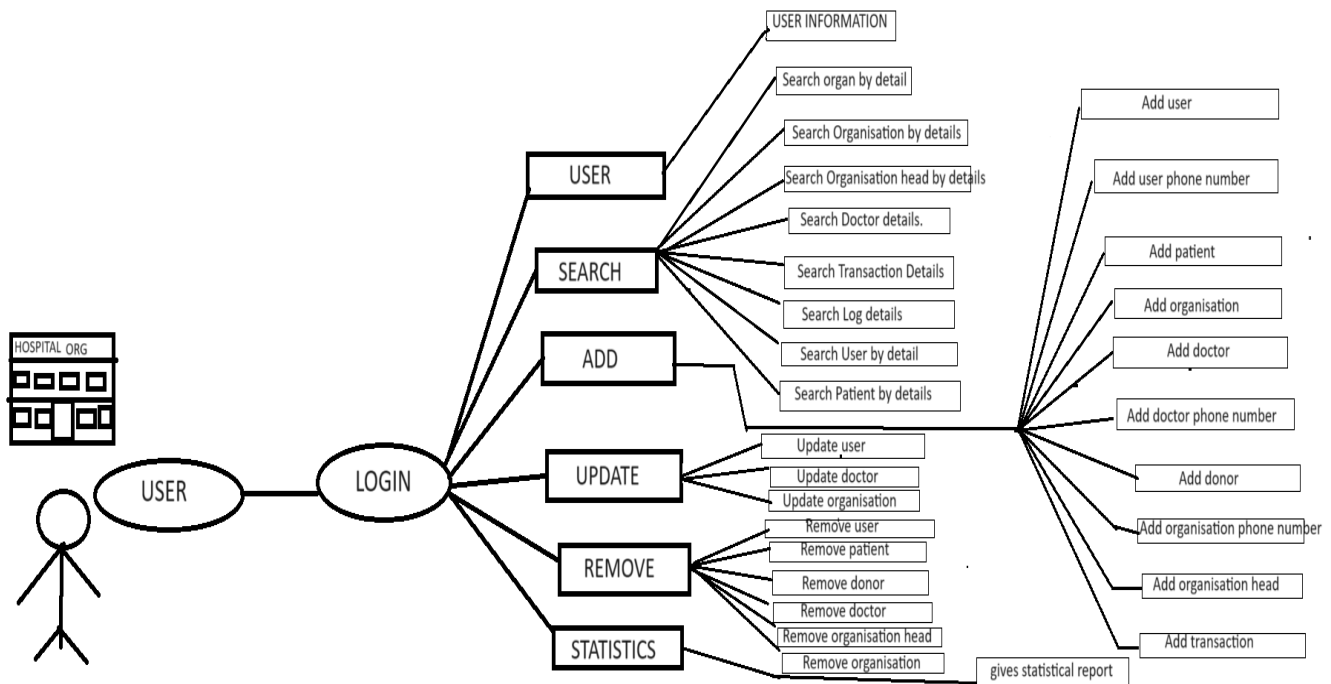
F30: The system shall allow users to remove organization records from the system.

F31: The system shall support the removal of organization head records from the system.

**Statistics:**

F32: The Statistics option shall provide users with statistical data and reports related to organ donation and procurement transactions.

### 3.3 Use Case Model



#### 3.3.1. Login – U1

**Author:** Gobburu Shiva

**Purpose:** The purpose of this use case is to allow a user to log in to the system using their username and password, granting them access to the main page upon successful validation.

**Requirements Traceability:** F1, F2, F3

**Priority:** High

**Preconditions:**

- The user has a valid username and password.
- The user is not already logged in.

**Post conditions:**

- The user is successfully logged in and granted access to the main page.
- If the login is unsuccessful, an error message is displayed.

**Actors:**

- User

**Flow of Events:**

**Basic Flow:**

- The user opens the application and navigates to the login page.
- The user enters their username and password.



- The system verifies the entered username and password.
- If the validation is successful, the system grants access to the main page (Use Case 4).
- If the validation fails, the system displays an error message and does not allow access.

**Alternative Flow:** None

**Exceptions:** None

**Includes:** None

**Notes/Issues:** None

### 3.3.2 Add User information :U2

**Author :** BANOTH BALAJI

**Purpose:** The purpose of this use case is to allow a user to add a new user's information, including User ID, Name, Date of Birth, Medical Insurance, Medical History, and Address.

**Requirements Traceability:** F4, F5

**Priority:** Medium

**Preconditions:**

- The user is logged in and has access to the main page (Use Case 1).

**Post conditions:**

- The user has successfully added a new user's information to the system.

**Actors:**

- User

**Flow of Events:**

**Basic Flow:**

- The user navigates to the "Add" option on the main page.
- The user selects the "Add User" function.
- The system presents a form for entering user information, including User ID, Name, Date of Birth, Medical Insurance, Medical History, and Address.
- The user fills in the required information.
- The user submits the form.
- The system adds the new user's information to the system.

**Alternative Flow:** None

**Exceptions:** None

**Includes:** None

**Notes/Issues:** None

### 3.3.3 Add User's Phone number : U3

**Author:** Gobburi Shiva

**Purpose:** The purpose of this use case is to allow a user to add phone numbers to a user's profile by associating them with the User ID.

**Requirements Traceability:** F6

**Priority:** Low

**Preconditions:**

- The user is logged in and has access to the main page (Use Case 1).

**Post conditions:**

- The user has successfully added phone numbers to a user's profile.

**Actors:**

- User

**Flow of Events:**

**Basic Flow:**



- The user navigates to the "Add" option on the main page.
- The user selects the "Add Phone Number to User" function.
- The system presents a form for entering phone number details and associating them with a User ID.
- The user fills in the required information.
- The user submits the form.
- The system associates the phone numbers with the User ID.

**Alternative Flow:** None

**Exceptions:** None

**Includes:** None

**Notes/Issues:** None

### 3.3.4 Search for Specific Users:U4

**Author:** Banoth Balaji

**Purpose:** The purpose of this use case is to allow a user to search for specific users based on their User ID or other user details.

**Requirements Traceability:** F15

**Priority:** Medium

**Preconditions:**

- The user is logged in and has access to the main page (Use Case 1).

**Post conditions:**

- The user has successfully found and retrieved user information based on the search criteria.

**Actors:**

- User

**Flow of Events:**

**Basic Flow:**

- The user navigates to the "Search" option on the main page.
- The user selects the "Search Users" function.
- The system presents a search form with options to enter User ID or other user details.
- The user enters the search criteria.
- The user initiates the search.
- The system retrieves and displays user information that matches the search criteria.

**Alternative Flow:** None

**Exceptions:** None

**Includes:** None

**Notes/Issues:** None

### 3.3.5 Search for Patients:U5

**Author:** Gobburu Shiva

**Purpose:** The purpose of this use case is to allow a user to search for patients using their Patient ID or specific patient details.

**Requirements Traceability:** F16

**Priority:** Medium

**Preconditions:**

- The user is logged in and has access to the main page (Use Case 1).

**Post conditions:**

- The user has successfully found and retrieved patient information based on the search criteria.

**Actors:**

- User

**Flow of Events:****Basic Flow:**

- The user navigates to the "Search" option on the main page.
- The user selects the "Search Patients" function.
- The system presents a search form with options to enter Patient ID or specific patient details.
- The user enters the search criteria.
- The user initiates the search.
- The system retrieves and displays patient information that matches the search criteria.

**Alternative Flow:** None**Exceptions:** None**Includes:** None**Notes/Issues:** None**3.3.6 Search for Organization Heads:U6****Author:** Banoth Balaji**Purpose:** The purpose of this use case is to allow users to search for organization heads based on their details.**Requirements Traceability:** F19**Priority:** Medium**Preconditions:**

- The user is logged in and has access to the main page (Use Case 1).

**Post conditions:**

- The user has successfully found and retrieved organization head information based on the search criteria.

**Actors:**

- User

**Flow of Events:****Basic Flow:**

- The user navigates to the "Search" option on the main page.
- The user selects the "Search Organization Heads" function.
- The system presents a search form with options to enter organization head details.
- The user enters the search criteria.
- The user initiates the search.
- The system retrieves and displays organization head information that matches the search criteria.

**Alternative Flow:** None**Exceptions:** None**Includes:** None**Notes/Issues:** None**3.3.7 Remove User accounts : U7****Author:** Gobburu Shiva**Purpose:** The purpose of this use case is to allow a user to remove user accounts from the system.**Requirements Traceability:** F26**Priority:** Medium**Preconditions:**

- The user is logged in and has access to the main page (Use Case 1).
- The user has appropriate permissions to remove user accounts.

**Post conditions:**

- The user account is successfully removed from the system.

**Actors:**

- User

**Flow of Events:****Basic Flow:**

- The user navigates to the "Remove" option on the main page.
- The user selects the "Remove User Account" function.
- The system presents a form or confirmation prompt for the user to confirm the removal.
- The user confirms the removal.
- The system removes the user account from the system.

**Alternative Flow:** None**Exceptions:** None**Includes:** None**Notes/Issues:** None**3.3.8 Remove Patient Record : U8****Author:** BANOTH BALAJI**Purpose:** The purpose of this use case is to allow a user to remove patient records from the system.**Requirements Traceability:** F27**Priority:** Medium**Preconditions:**

- The user is logged in and has access to the main page (Use Case 4).
- The user has appropriate permissions to remove patient records.

**Post conditions:**

- The patient records are successfully removed from the system.

**Actors:**

- User

**Flow of Events:****Basic Flow:**

- The user navigates to the "Remove" option on the main page.
- The user selects the "Remove Patient Record" function.
- The system presents a form or confirmation prompt for the user to confirm the removal.
- The user confirms the removal.
- The system removes the patient records from the system.

**Alternative Flow:** None**Exceptions:** None**Includes:** None**Notes/Issues:** None**3.3.9 Remove organisation records: U9****Author:** GOBBURI SHIVA**Purpose:** The purpose of this use case is to allow users to remove organization records from the system.**Requirements Traceability:** F30**Priority:** Medium**Preconditions:**

- The user is logged in and has access to the main page (Use Case 1).
- The user has appropriate permissions to remove organization records.

**Post conditions:**

- The organization records are successfully removed from the system.

**Actors:**

- User

**Flow of Events:****Basic Flow:**

- The user navigates to the "Remove" option on the main page.
- The user selects the "Remove Organization Record" function.
- The system presents a form or confirmation prompt for the user to confirm the removal.
- The user confirms the removal.
- The system removes the organization records from the system.

**Alternative Flow:** None**Exceptions:** None**Includes:** None**Notes/Issues:** None**3.3.10:Update user Information:U10****Author:** BANOTH BALAJI**Purpose:** The purpose of this use case is to allow a user to update user information, including Name, Date of Birth, Medical Insurance, Medical History, and Address.**Requirements Traceability:** F23**Priority:** Medium**Preconditions:**

- The user is logged in and has access to the main page (Use Case 1).
- The user has appropriate permissions to update user information.

**Post conditions:**

- The user's information is successfully updated in the system.

**Actors:**

- User

**Flow of Events:****Basic Flow:**

- The user navigates to the "Update" option on the main page.
- The user selects the "Update User Information" function.
- The system presents a form for the user to update their information, including Name, Date of Birth, Medical Insurance, Medical History, and Address.
- The user makes the desired updates.
- The user submits the form.
- The system updates the user's information in the system.

**Alternative Flow:** None**Exceptions:** None**Includes:** None**Notes/Issues:** None**3.3.11 Modification of organisation details:U11****Author:** GOBBURI SHIVA**Purpose:** The purpose of this use case is to support the modification of organization details, including Organization Name, Location, and Government Approval Status.**Requirements Traceability:** F25**Priority:** Medium**Preconditions:**

- The user is logged in and has access to the main page (Use Case 4).
- The user has appropriate permissions to update organization details.

**Post conditions:**

- The organization's details are successfully updated in the system.

**Actors:**

- User

**Flow of Events:****Basic Flow:**

- The user navigates to the "Update" option on the main page.
- The user selects the "Update Organization Details" function.
- The system presents a form for the user to update organization details, including Organization Name, Location, and Government Approval Status.
- The user makes the desired updates.
- The user submits the form.
- The system updates the organization's details in the system.

**Alternative Flow:** None**Exceptions:** None**Includes:** None**Notes/Issues:** None**3.3.12 Statistical data and reports:U12****Author:** BANOTH BALAJI

**Purpose:** The purpose of this use case is to provide users with statistical data and reports related to organ donation and procurement transactions.

**Requirements Traceability:** F32**Priority:** Medium**Preconditions:**

- The user is logged in and has access to the main page (Use Case 4).
- The user has appropriate permissions to access and view statistics.

**Post conditions:**

- The user has successfully accessed and viewed statistical data and reports.

**Actors:**

- User

**Flow of Events:****Basic Flow:**

- The user navigates to the "Statistics" option on the main page.
- The user selects the type of statistics or report they wish to view (e.g., Organ Donation Statistics, Transaction Reports).
- The system generates the requested statistical data or report.
- The system displays the statistical data or report to the user.

**Alternative Flow:** None**Exceptions:**

- If there is no statistical data available for the selected category, the system displays a message indicating that there are no statistics to show.

**Includes:** None**Notes/Issues:** None

## 4 Other Non-functional Requirements

### 4.1 Performance Requirements

**P1:** The system should be responsive, taking no more than 2 seconds to respond to user actions.

**P2:** The system should handle at least 100 users simultaneously without slowing down.

**P3:** Basic database operations, like searching, should be quick, executing within 1 second.  
**P4:** Organ transactions should be real-time, with a maximum 5-second delay.  
**P5:** Generating reports should take no more than 10 seconds.  
**P6:** Data backup and restore should be done during off-peak hours to avoid interruptions.  
**P7:** The system should be available 24/7, with at least 99.9% uptime.  
**P8:** Security checks should take no more than 1 second.  
**P9:** The system should easily grow by 20% annually while maintaining performance.  
These requirements ensure efficient, responsive, and available system performance

## 4.2 Safety and Security Requirements

S1: User data, including personal information, medical records, and transaction details, shall be encrypted during transmission to prevent unauthorized access. The system will employ secure encryption protocols (e.g., TLS) to ensure the confidentiality and integrity of data transferred between users' devices and the server.

S2: The system shall enforce strict access controls and user authentication mechanisms. User administrators, must log in with unique and secure credentials. Role-based access control will be applied to grant permissions only to authorized personnel based on their roles and responsibilities.

Rationale: S1 ensures that sensitive user data remains confidential and protected during transmission, reducing the risk of data breaches. S2 enhances the system's security by implementing user authentication and access controls, preventing unauthorized access and maintaining data integrity.

## 4.3 Software Quality Attributes

**Reliability:** The system will be available 99.9% of the time, with automated backup and error monitoring systems to ensure uninterrupted service.

**Maintainability:** The code will be well-documented and modular, and updates will be tested in a staging environment to facilitate efficient changes without disrupting service.

**Usability :** The user interface will be intuitive and user-friendly, with tooltips and guides for ease of use.

**Security :** Role-based access control, data encryption, and regular security audits will protect user data and maintain trust in the system.

These attributes prioritize dependability, adaptability, user satisfaction, and data security in the project.

**Appendix B - Group Log**

<i>Meeting day (date)</i>	<i>Timings</i>
<i>20/10/2023</i> <i>21/10/2023</i>	<i>10:00 am to 12:00 pm</i> <i>8:00 pm to 12:00 am</i>

<i>Name</i>	<i>Contribution</i>
<i>GOBBURI SHIVA</i>	<i>Document Purpose, Product Scope, Intended Audience, Document Overview, Definitions, Acronyms And Abbreviations NonFunctional Requirement, Software Interfaces, Functional Requirements ,Use case model,use cases1,3,5,7,9,11</i>
<i>BANOTH BALAJI</i>	<i>Product Overview, Product Functionality, Design And Implementation Constraints, Assumptions And Dependencies, External Interface Requirements – User, Hardware Use cases 2,4,6,8,10,12</i>