INITIAL REPORT

ON

ORGAN BANK

Submitted By:

Dona Jince

MAC23MCA-2024

Faculty Guide:

Prof. Nisha Markose

Project Coordinator:

Prof. Sonia Abraham

ABOUT THE PROJECT

The Organ Bank is a comprehensive digital solution developed as a mobile application to automate and streamline the process of organ donation, matching, and transplantation. This project eliminates the inefficiencies of traditional manual systems by providing an integrated, real-time platform that ensures reliability, transparency, and efficiency in managing organ donations and transplants.

The existing system for organ donation and transplantation often involves manual processes, leading to delays, inefficiencies, and missed opportunities for life-saving procedures. While some organizations maintain static websites for registration or informational purposes, these systems fail to provide dynamic solutions for managing donor and recipient data, organ matching, and hospital coordination. Such limitations result in underutilized resources, lack of transparency, and delayed decision-making, which can cost lives.

The Organ Bank addresses these shortcomings by introducing a dynamic and real-time platform designed to meet the needs of all stakeholders, including administrators, donors, recipients, and hospitals. Key features include automated organ matching based on medical criteria, real-time notifications for compatibility testing schedules, transplant dates, and a feedback mechanism to ensure continuous improvement and accountability.

The system uses Flutter for developing a responsive and intuitive user interface across both mobile and web platforms, ensuring seamless user experiences. Node.js powers the backend, efficiently handling server-side logic, organ matching algorithms, and business processes, while MongoDB serves as the database for secure, scalable, and real-time data storage and management. The development environment integrates Visual Studio Code for the mobile app.

By utilizing these modern technologies, the Organ Bank provides a robust and efficient solution that automates processes, reduces manual errors, and accelerates decision-making. Its real-time features and enhanced communication build trust among stakeholders while significantly improving the success rate of organ transplants.

This innovative solution not only addresses critical challenges in organ donation and transplantation but also contributes significantly to the noble cause of saving lives, making it a vital tool for modern healthcare management.

ACTORS AND THEIR ROLES

The Organ Bank involves multiple actors, each playing a significant role in ensuring the efficient and transparent functioning of the system. These actors include Admin, Donor, Recipient, and Hospital.

1. Admin

- Manages donor, recipient, and hospital registrations.
- Monitors the system for new registrations and approves/rejects them.
- Oversees the automated organ matching system and reviews potential matches.
- Approves organ matches and adds them to the matched organs table.
- Notifies donors, recipients, and hospitals about approved matches.
- Tracks organ availability and generates detailed reports.
- Manages system data and ensures proper functionality.
- Monitors transplantation results and oversees feedback submissions.
- Removes matches from the system if compatibility testing fails.

2. Donor

- Registers with the system, providing consent for organ donation.
- Updates personal details and organ donation preferences.
- Receives notifications from the admin and hospital regarding matches, compatibility tests, and transplantation schedules.
- Provides feedback about the system and hospital services.

3. Recipient

- Registers with the system, requesting organ transplants.
- Updates personal details and organ requirements.
- Receives notifications about organ matches, compatibility tests, and transplantation schedules.
- Tracks organ match statuses and compatibility test results.
- Provides feedback about the system and hospital services.

4. Hospital

- Registers with the system and manages hospital-related data.
- Conducts compatibility tests for matched donors and recipients.
- Schedules and carries out transplantation procedures.
- Notifies donors and recipients about test and transplantation schedules.
- Updates the system with test results and transplantation outcomes.
- Provides feedback on the organ bank system.

DESCRIPTION OF MODULES

1. User Management

- Donors, Recipients, and Hospitals can register themselves by providing necessary details such as personal information, contact details, and medical records (for donors and recipients).
- Hospitals provide their registration details, including license numbers and address.
- Admin reviews the registration details.
- Approves or rejects registrations based on validity and authenticity of the provided information.
- Sends approval/rejection notifications to the respective users.
- Donors and Recipients can update their profiles after approval.
- Hospitals can edit their details, such as contact information and services provided.

2. Organ Matching Module

- Automatically matches registered donors and recipients based on compatibility factors such as blood group, organ type, and age.
- Adds successful matches to the Matched Organs Table, awaiting admin approval.
- Admin approves or rejects the matches, and notifications are sent to the donor, recipient, and hospital.

3. Compatibility Testing Module

- Hospitals schedule onsite compatibility tests for approved matches.
- Updates the system with test results (pass/fail).
- If compatibility tests fail:
    - The corresponding match is removed from the Matched Organs Table.
    - Notifications are sent to the donor, recipient, and admin.

4. Feedback and Complaint Management Module

- **Feedback**
    - Donors, Recipients, and Hospitals can submit feedback on their experiences, system usability, and hospital services.
    - Feedback is visible to the Admin for system improvement.

- **Complaints**
    - Donors and Recipients can raise complaints about hospital services, compatibility testing, or delays in processes.
    - Hospitals can report any issues related to the system or donor/recipient cooperation.
    - Admin tracks complaints and resolves them promptly, with updates visible to the complainant.

5. Report Management Module

- Admin generates reports on:
    - Registered Donors, Recipients, and Hospitals.
    - Organ availability (filtered by type, donor status, and location).
    - Compatibility test results (success/failure rates).
    - Transplantation outcomes and success rates.
    - Feedback and complaint trends.

6. Transplantation Management Module

- Hospitals conduct transplantation procedures after successful compatibility tests.
- Updates the system with transplantation outcomes (success or failure).
- Notifications are sent to Admin, Donors, and Recipients about the procedure status.

Business Rules

- Registrations for Donors, Recipients, and Hospitals must be reviewed and approved by the Admin before granting access to the system.
- Donors must provide explicit consent for organ donation during registration.
- Consent can be revoked at any time before a match is approved by the Admin.
- Recipients must provide valid medical records during registration to justify their request for an organ.
- Priority will be given based on medical urgency, compatibility, and waiting time.
- Matches must be reviewed and approved by the Admin before proceeding
- to compatibility tests. Only verified hospitals can perform compatibility tests and transplantation procedures.
- Admin must review all registrations (Donors, Recipients, Hospitals) within 24–48 hours of submission.

- Complaints must be addressed by the Admin promptly and updates provided to the complainant.

TECHNOLOGY/FRAMEWORK

Frontend Development

- Flutter: Used for creating the user interface for both the mobile app and the web-based system. Ensures cross-platform compatibility, providing a consistent and seamless experience across Android, and web devices.Offers interactive and user-friendly interfaces for donors, recipients, hospitals, and the admin. Built using the Dart programming language for high performance and a smooth user experience.

Backend Development

- Node.js:Powers the backend by providing a fast, scalable, and event-driven environment. Manages server-side logic, including user authentication, organ matching, notifications, and admin approval workflows. Ensures efficient communication between the frontend and database via RESTful APIs.

Database Management

- MongoDB: A NoSQL database used for storing all system data in a flexible, JSON-like format.Manages critical data, and ensures data consistency and supports complex relationships within the system.

PROJECT FEASIBILITY

A project feasibility study is essential to evaluate the technical, economic, and operational viability of the proposed Organ Bank. The study ensures that the project can succeed within the defined scope, resources, and timeline while identifying potential challenges.

1. Technical Feasibility

The Organ Bank is technically feasible due to its use of robust and widely adopted technologies:

- **Frontend**:
    - Built using **Flutter**, a cross-platform framework delivering a seamless experience across Android, iOS, and web platforms.
    - **Dart**, the programming language used in Flutter, ensures smooth animations, responsive UI, and efficient application performance.

- **Backend**:
    - Developed using **Node.js**, a fast, scalable, and event-driven environment ideal for handling server-side logic, organ matching algorithms, and notifications.
    - RESTful APIs ensure efficient communication between the frontend and database.
- **Database**:
    - **MongoDB**, a NoSQL database, manages unstructured and semi-structured data like donor and recipient profiles, organ availability, and transaction logs.
    - MongoDB's flexible schema supports scalability, fast data retrieval, and ensures data consistency.

This technology stack provides a modern, reliable, and efficient solution that is widely supported. It ensures compatibility with various devices, making the system accessible to all users, including donors, recipients, hospitals, and administrators.

2. Economic Feasibility

The Organ Bank is cost-effective due to the following:

- **Open-source Technologies**: The use of **Flutter**, **Node.js**, and **MongoDB** eliminates licensing fees, significantly reducing development costs.
- **Availability of Skilled Developers**: These technologies are widely adopted, making it easy to find skilled professionals at reasonable rates.
- **Automation Benefits**: The organ-matching system, notification module, and reporting features reduce manual administrative costs and time investment.

While initial development, deployment, and maintenance will require investment,the system's ability to reduce operational costs, optimize resource utilization, and improve decision-making provides strong economic justification.

3. Operational Feasibility

The Organ Bank streamlines operations and builds trust through:

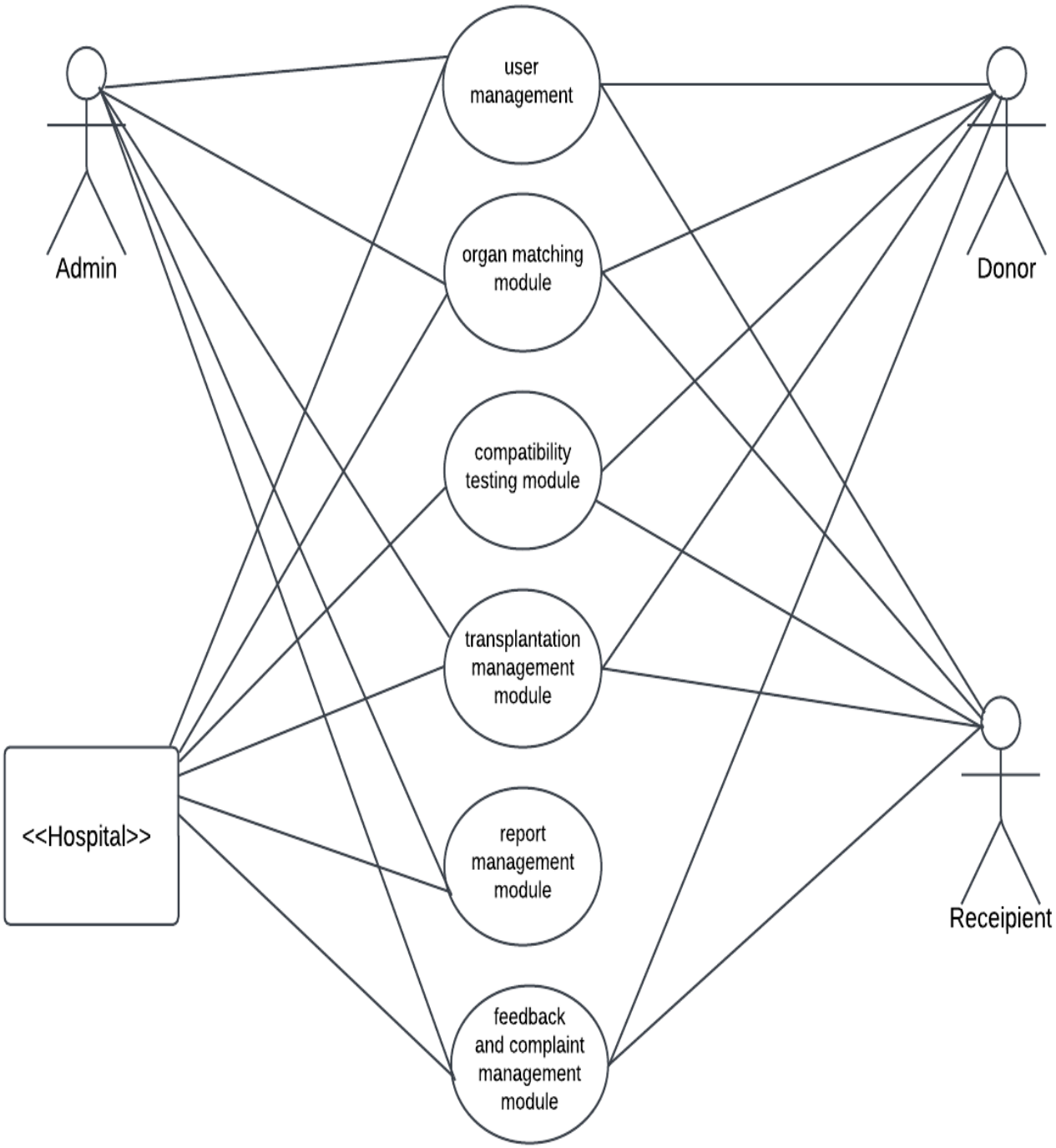
- **Frontend**:
    - **Flutter** delivers an intuitive and responsive user interface for all stakeholders.
    - The cross-platform nature ensures a consistent experience across mobile and web platforms.

- **Backend**:
    - **Node.js** efficiently handles backend processes, including user authentication, organ matching, and notifications.
    - The **MongoDB** database ensures fast and reliable data storage and retrieval, supporting complex queries and maintaining data integrity.
- **Automation**:
    - Automates organ matching, notifications, compatibility testing management, and data logging, significantly reducing manual effort and minimizing errors.
- **Trust and Transparency**:
    - The notification and feedback systems enhance transparency and communication, building trust among donors, recipients, hospitals, and administrators.

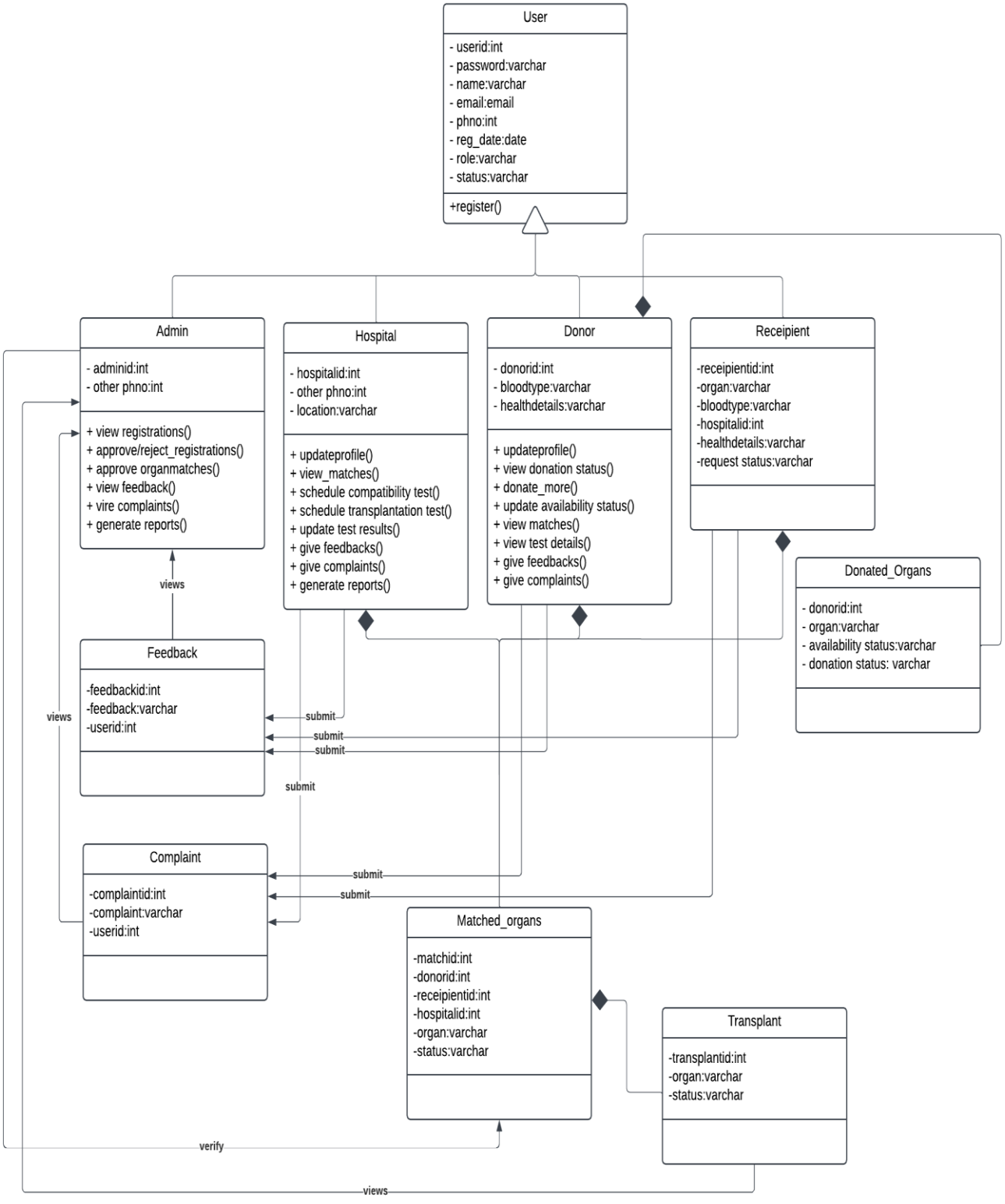
By automating critical processes and reducing manual interventions, the system optimizes workflows and enhances operational efficiency. Regular updates, skilled developers, and strong security measures will ensure long-term success and scalability.

UML Diagrams

1. UseCase Diagram

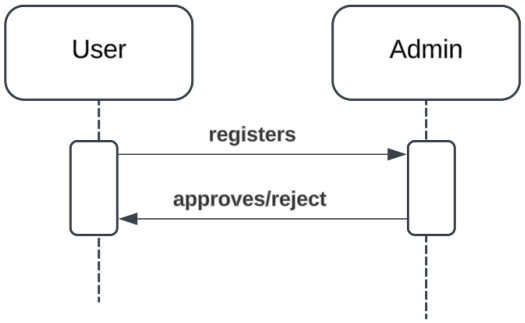


2. Class Diagram

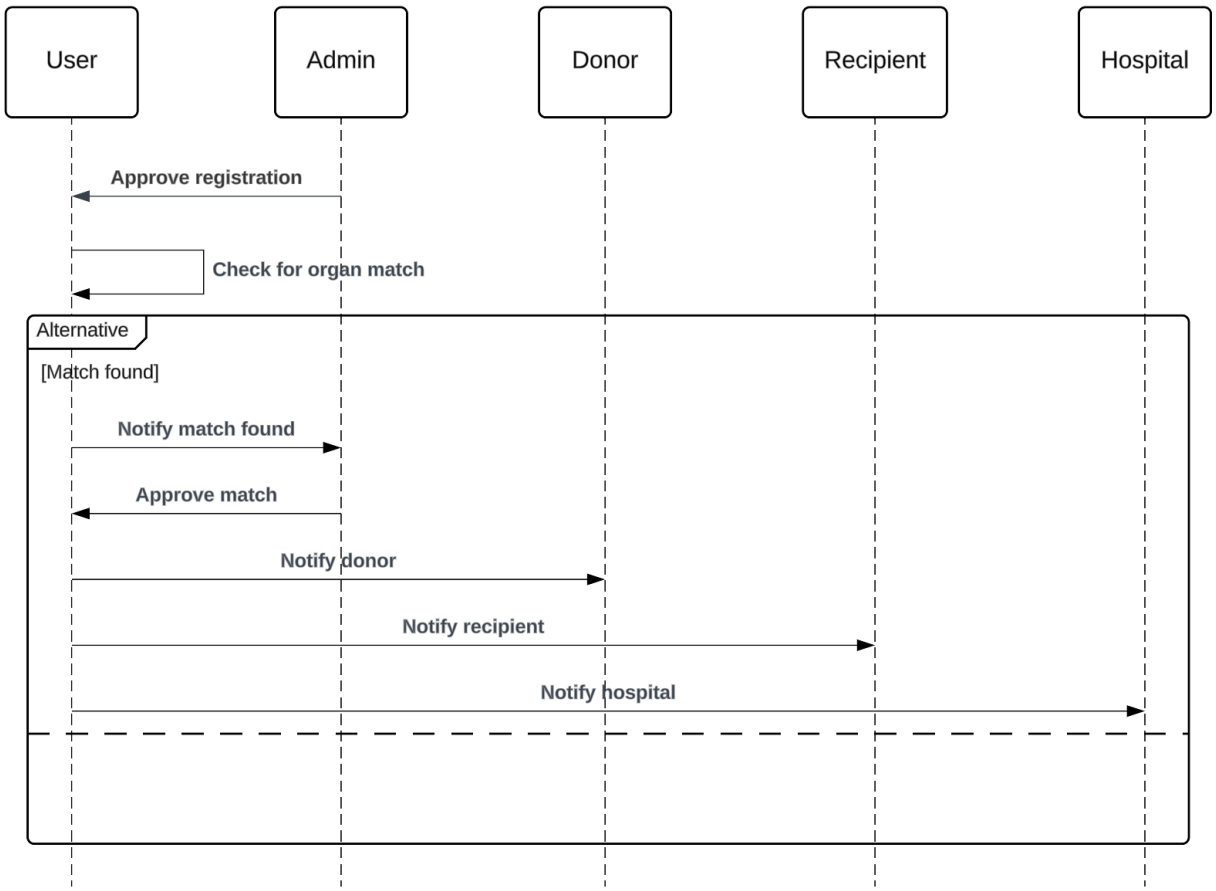


3. Sequence Diagram

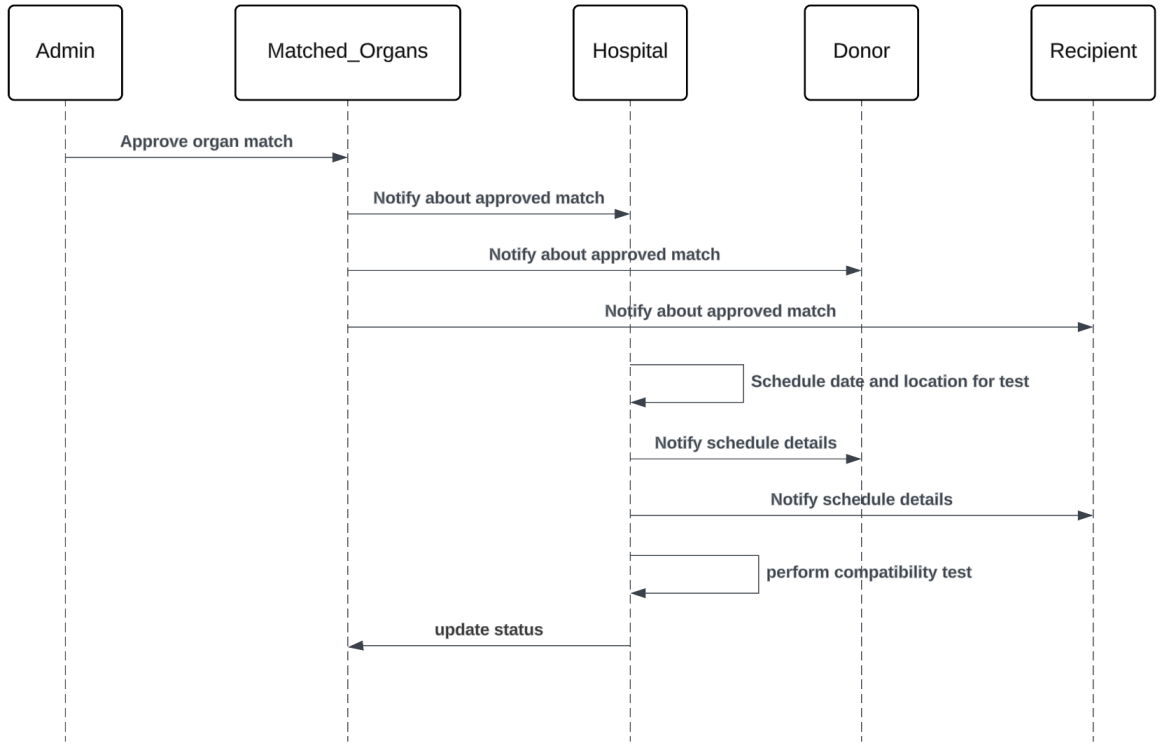
3.1 User Registration



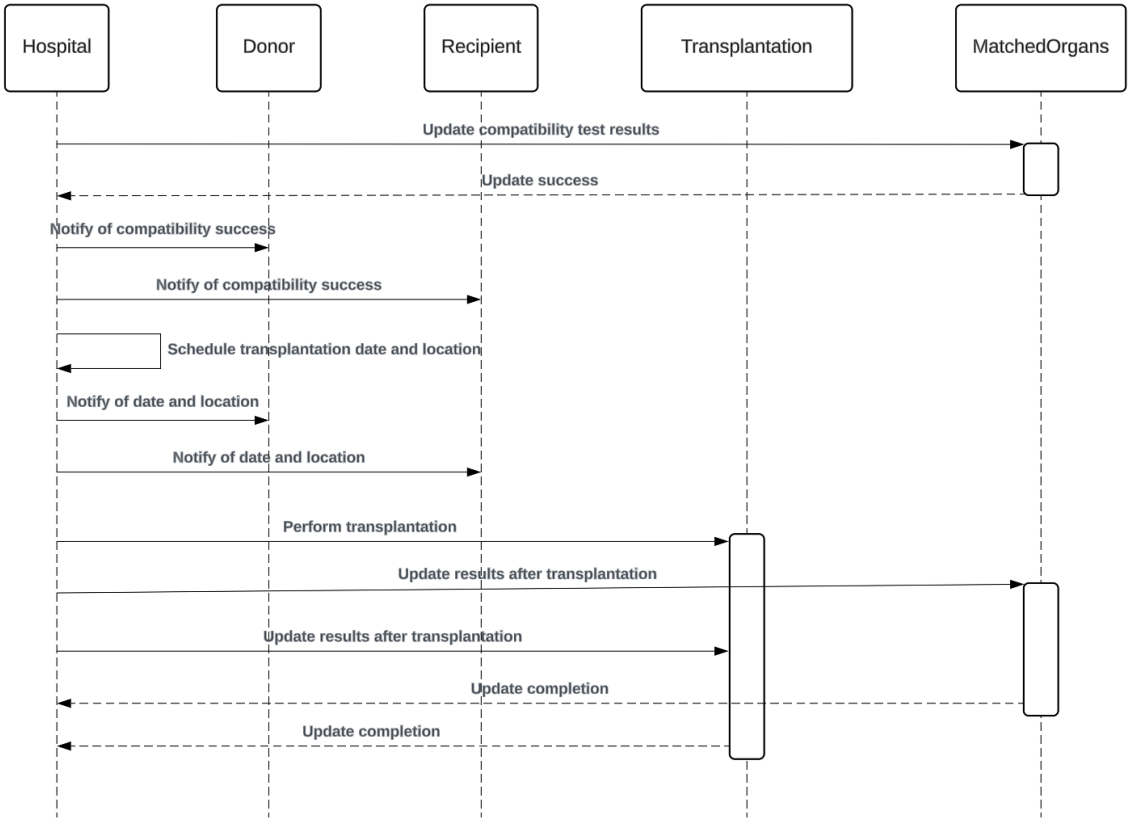
3.2 Organ Matching



3.3 Compatibility Test

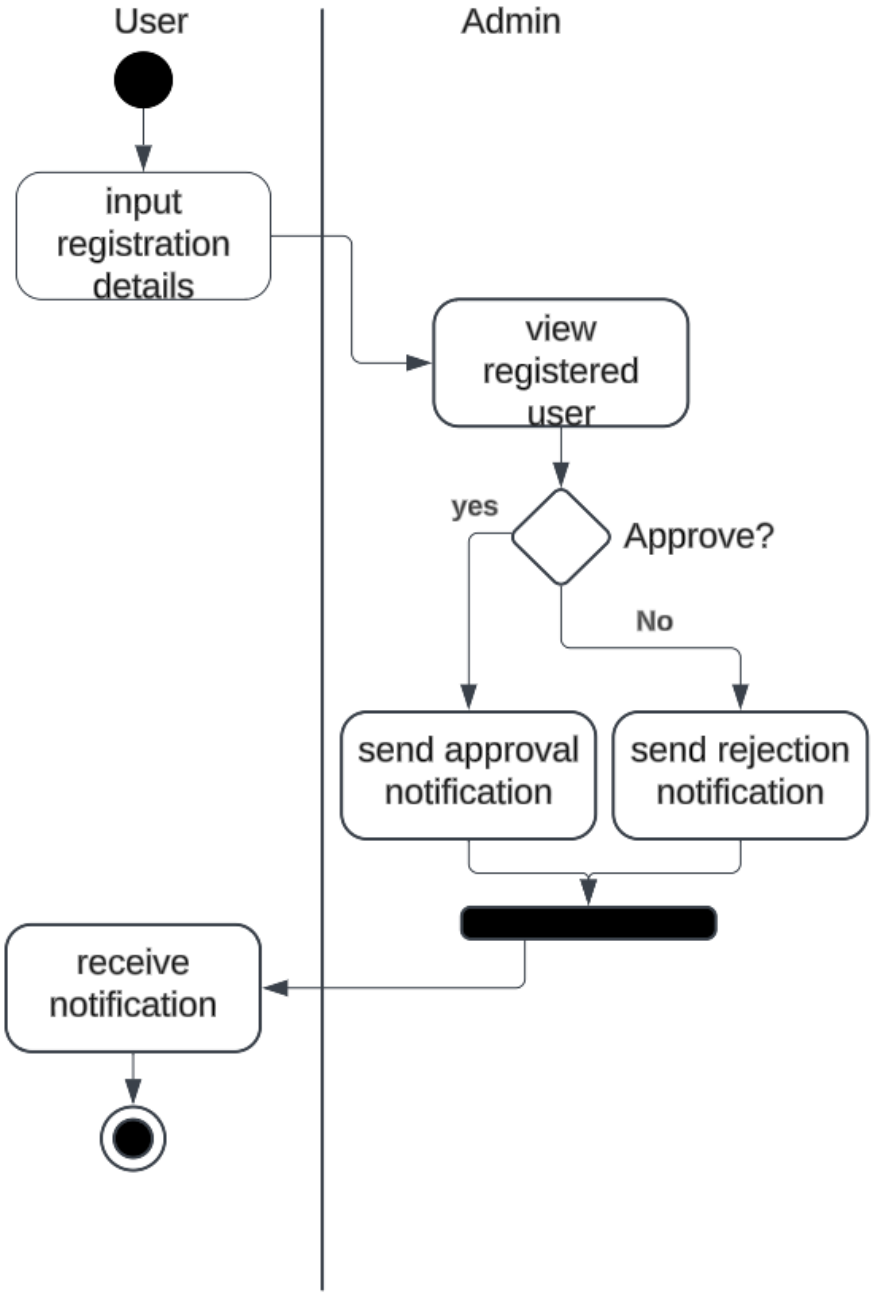


3.4 Transplantation Module

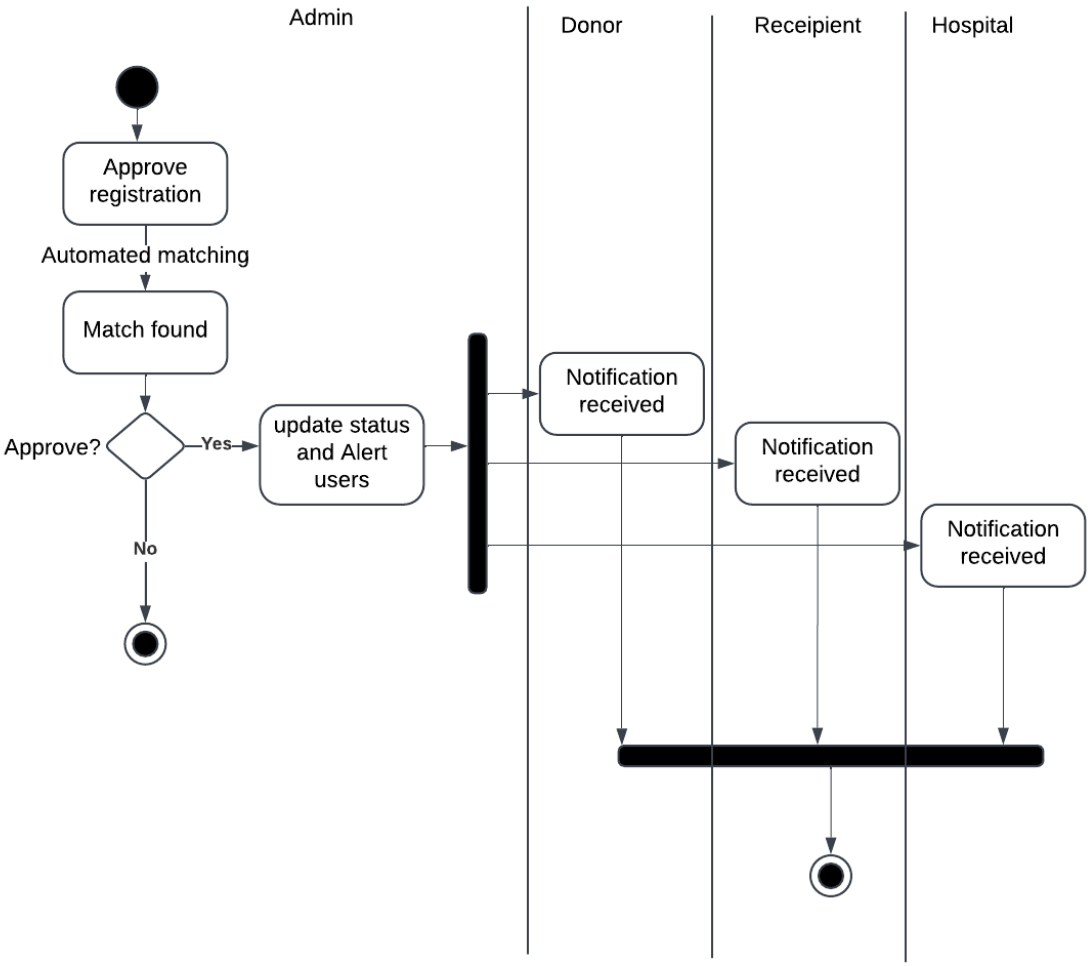


4. Activity Diagram

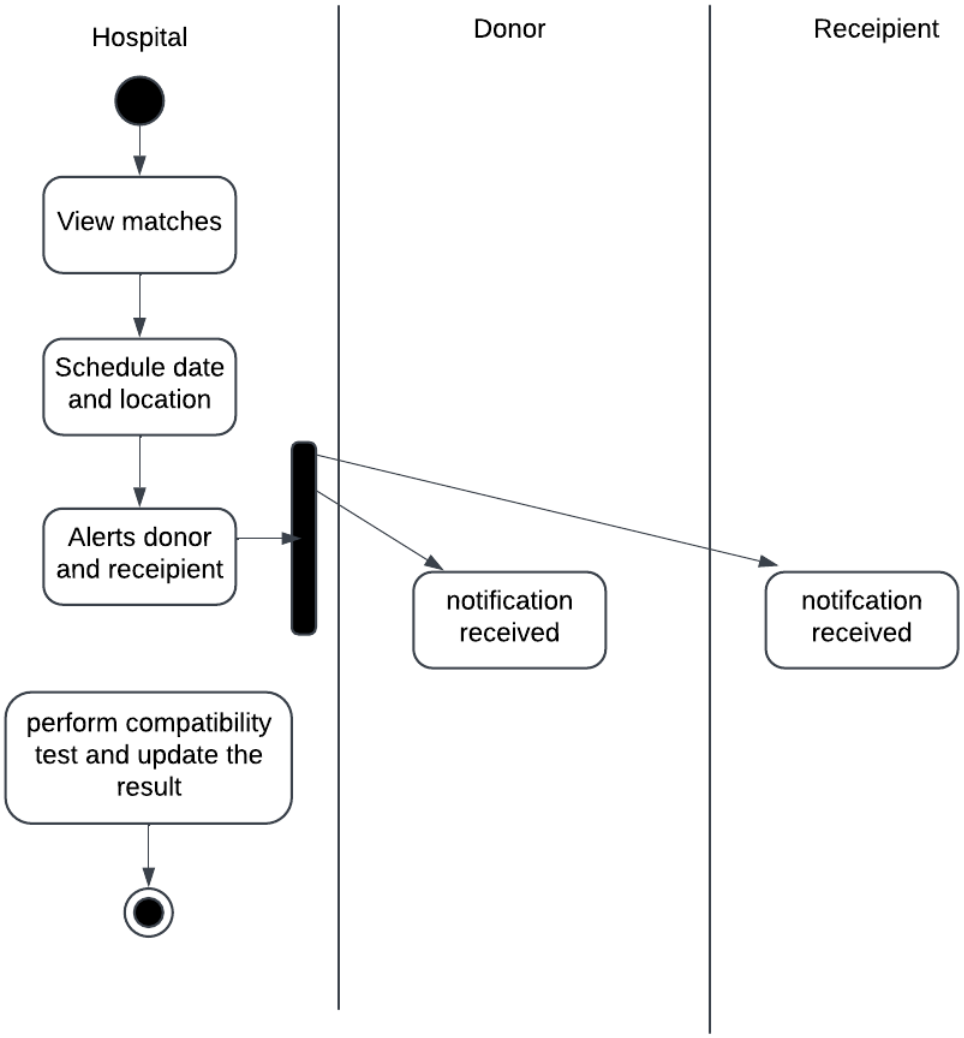
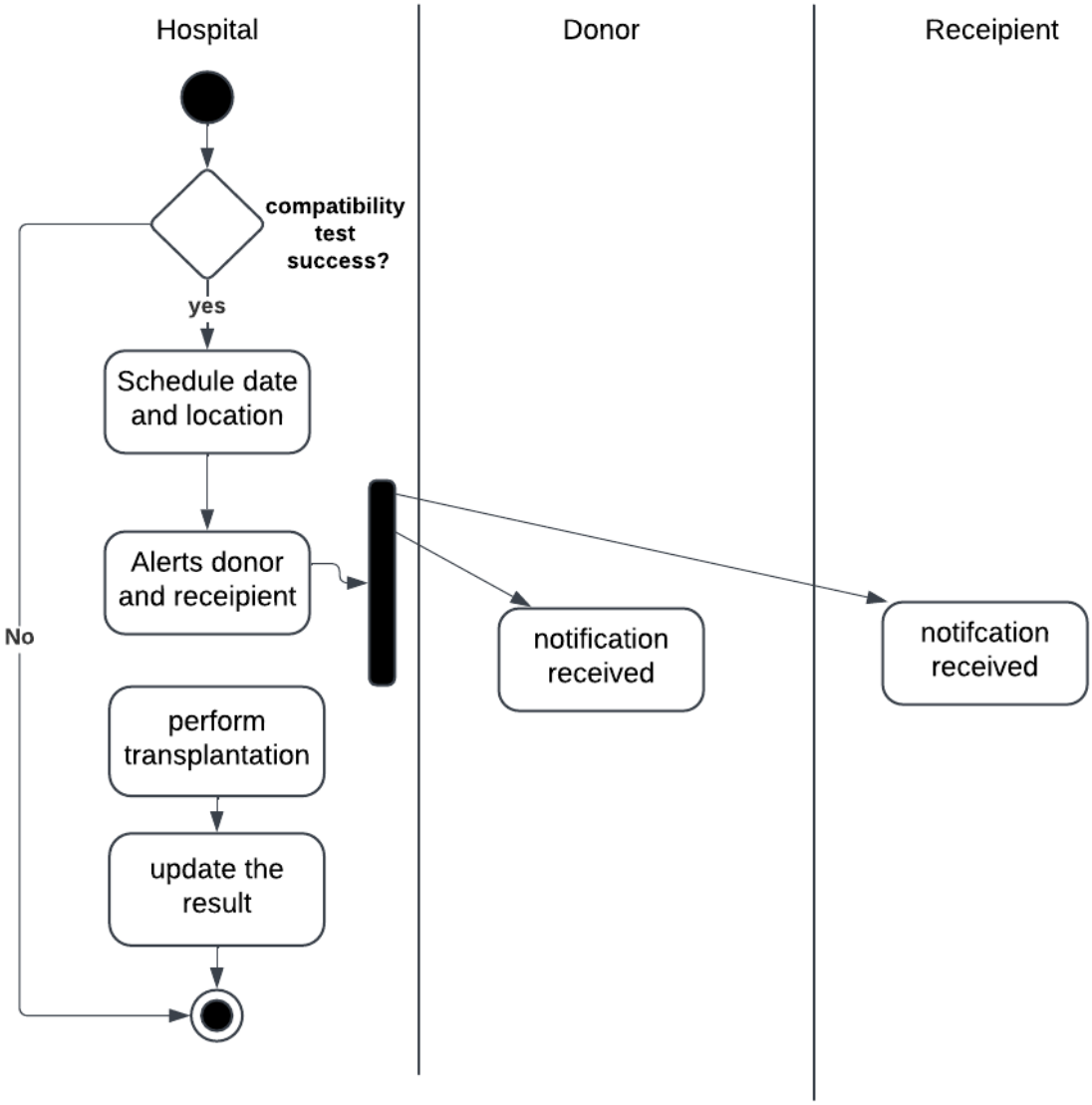
4.1 User Registration



4.2 Organ Matching



4.3 Compatibility Test



4.4 Transplantation Module