**Software Requirement Specification (SRS) for Organ Donation and Transplant Matching System**

**1. Introduction**

**1.1 Purpose**

The purpose of the Organ Donation & Transplant Matching System is to streamline and enhance the efficiency of organ donation and transplantation. This system simplifies user registration, ensures secure data management, and optimizes the recipient matching process. It provides a centralized platform for hospitals to coordinate transplants, track real-time organ availability, and facilitate seamless donor-recipient matching based on medical compatibility and urgency.

**1.2 Scope**

* Secure authentication and role-based access.
* Matching of donors and recipients based on medical compatibility.
* Real-time tracking of organ availability and transportation.
* Secure documentation management for legal and medical records.
* Coordination among hospitals for transplant scheduling.
* Automated notifications and updates for transplant progress.
* Reporting and analytics for transplant trends and success rates.

**1.3 Definitions, Acronyms, and Abbreviations**

* **AI**: Artificial Intelligence
* **SQLite**: Lightweight database system
* **API**: Application Programming Interface

**1.4 References**

* Organ Transplant Regulations & Guidelines
* Medical Standards for Organ Compatibility
* Django Official Documentation

**2. Overall Description**

**2.1 Application Perspective**

The system will be a web-based application, ensuring that hospitals and medical professionals can manage the organ donation and transplant process efficiently. The architecture will integrate:

* **Backend:** Django Framework (Python)
* **Frontend:** HTML, CSS, JavaScript
* **Database:** SQLite

**2.2 Application Functions**

* Donor and recipient registration
* Organ availability tracking
* Recipient-donor matching based on compatibility
* Hospital coordination for organ transfer and transplantation
* Secure legal documentation storage and access
* Emergency transplant request handling
* Automated notifications and alerts
* Report generation for donation trends and analytics

**2.3 User Classes and Characteristics**

* Donors are individuals willing to donate organs.
* Recipients are patients registered in the system awaiting organ transplants.
* Hospitals are medical institutions responsible for organ transplantation and patient care.
* Administrators are system managers overseeing data security, user access, and overall system operations.

**2.4 Operating Environment**

**• Web Browser (**Chrome, Firefox, Edge) **•** Mobile apps (Android, iOS) for user interaction. **•** Cloud-based system for secure data storage and access.

**3. System Features**

**3.1 User Management**

* User registration and authentication
* Profile management to update personal and medical details.

**3.2 Donor Management Module**

* Collects donor details, including organ type, blood type, and availability.
* Allows donors to update their medical history and availability status.

**3.3 Recipient Management Module**

* Stores recipient details such as required organ type and compatibility factors.
* Matches recipients with donors based on medical criteria.

**3.4 Transplant Matching Module**

* Matches donors with recipients based on compatibility factors such as blood type and urgency.
* Tracks the status of transplant requests (Pending, Approved, Completed).

**3.5 Organ Tracking Module**

* Monitors the real-time movement of donated organs.
* Tracks organ status (Available, In Transit, Delivered).

**3.6 Hospital Management Module**

* Manages hospitals participating in organ transplants.
* Facilitates hospital coordination for organ transportation and transplant scheduling.

**3.7 Legal Documentation Management**

* Secure storage and retrieval of donor consent forms and recipient approval documents.
* Ensures compliance with legal and ethical guidelines.

**3.8 Report and Analytics Module**

* Generates donor-recipient match reports and transplant success statistics.
* Provides insights for healthcare professionals and policymakers.

**4. External Interface Requirements**

**4.1 User Interfaces**

* Web Application UI for donor, recipient, and hospital management.
* Mobile Application UI for real-time organ tracking and status updates.

**4.2 Hardware Interfaces**

* Biometric Authentication Devices for secure access to donor and recipient records.
* Cloud Server Integration for storing and retrieving transplant-related data.

**4.3 Software Interfaces**

* Django API for backend organ donation management.
* Hospital Database APIs for retrieving donor, recipient, and transplant records.
* Notification API for sending real-time alerts to users regarding transplant status.

**5. System Requirements**

**5.1 Software Requirements**

* Development IDE (VS Code, PyCharm, etc.)
* Django (Backend Framework) for managing organ donation and transplantation.
* Python for backend processing and system logic.
* HTML, CSS, JavaScript for front-end UI.
* SQLite as the database for storing donor and recipient information.

**5.2 Hardware Requirements**

* Processor: Intel i5 or higher.
* RAM: Minimum 4GB.

**6. Performance Requirements**

* Real-time donor-recipient matching within 5 seconds.
* Maximum server downtime of 2% annually.

**7. Security Requirements**

* User authentication with secure login credentials.
* Data encryption for protecting donor and recipient information.
* Backup and recovery system to prevent data loss.

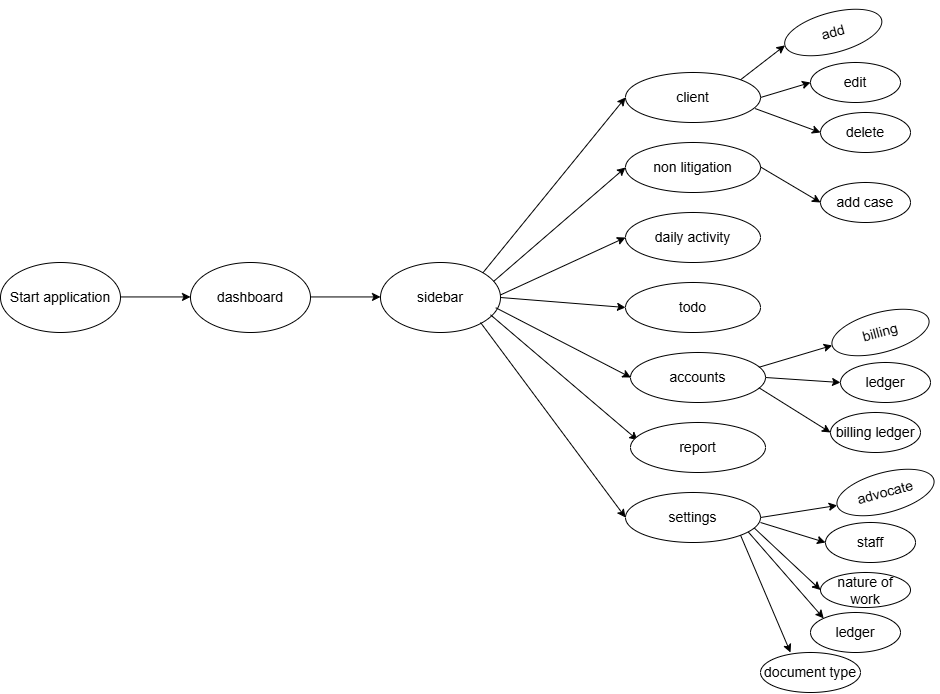
**8. Future Enhancements**

1. AI-based predictive matching to improve organ compatibility and optimize recipient selection.
2. Mobile app integration for remote user registration, status updates, and real-time tracking.

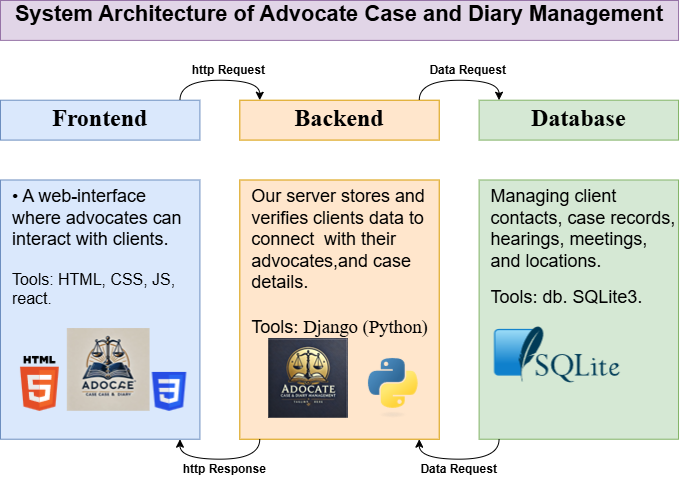
**9. Conclusion**

The Organ Donation & Transplant Matching System enhances efficiency in donor-recipient matching, organ tracking, and hospital coordination. It ensures secure documentation management, real-time updates, and automated notifications.

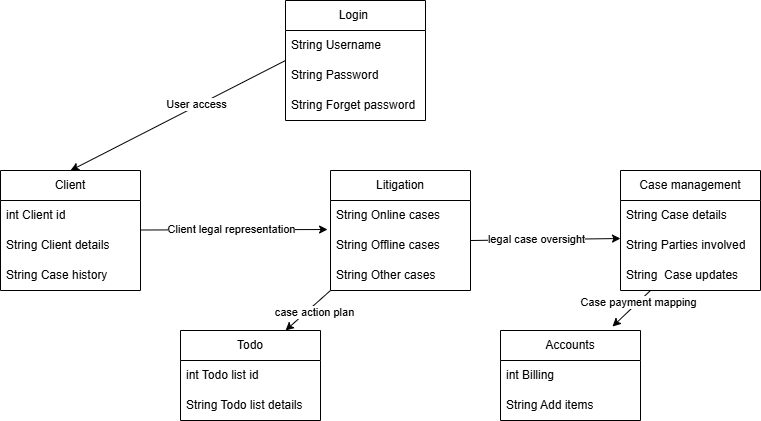
**Use Case Diagram:**

****

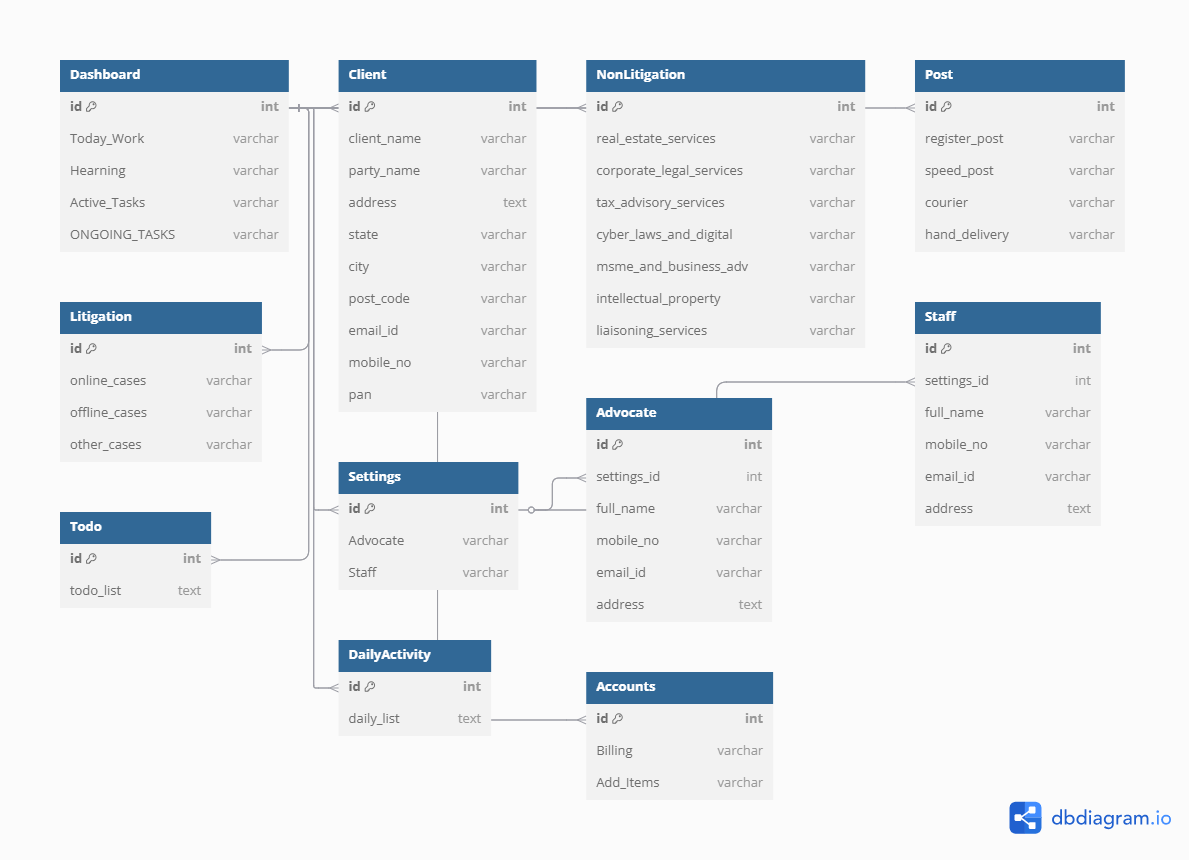
**System Design:**

****

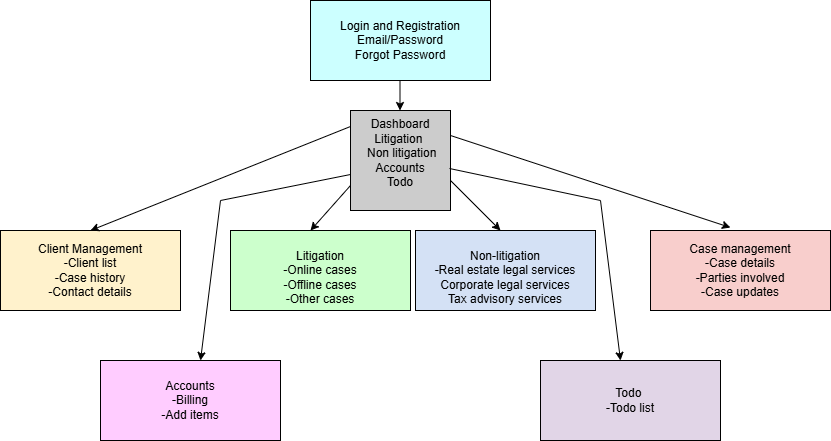
**Database Diagram:**

****

**ER diagram**

****

**User interface design**

****