LIFELINK

Blood & Organ Donation Platform

Software Requirements Specification

Document Version:	1.0
Date:	January 4, 2026
Prepared By:	Muhammad Abdal
	Muhammad Abdal
	Muhammad Abdal
Group No:	Group 27
Approved By:	Project Advisor / Committee
Project:	LifeLink Mobile Application
Client:	Final Year Project - Computer Science

Department of Computer Science
Forman Christian College (A Chartered University)
Lahore, Pakistan

Contents

List of Tables

List of Figures

1 Introduction and Background

1.1 Product (Problem Statement)

LifeLink is a Flutter-based mobile application designed to address Pakistan's critical shortage of timely blood and organ donations. Each year, thousands of patients die awaiting blood transfusions, organ transplants, and essential medicines due to fragmented donor databases, inefficient matching systems, and lack of real-time coordination between donors and recipients. The current system relies on unreliable social media appeals, word-of-mouth requests, and disconnected hospital registries that cannot provide immediate assistance during emergencies. LifeLink solves these problems by creating a unified, location-aware platform that connects willing donors with needy recipients in real-time, prioritizing emergency cases and ensuring life-saving resources reach those in need when every second counts.

1.2 Background

Pakistan's healthcare donation ecosystem faces systemic challenges that have persisted for decades. With only 10% of the country's blood needs met through organized donation systems, patients often depend on family members or paid donors. Organ donation rates remain among the lowest globally due to cultural misconceptions, lack of awareness, and inefficient registration systems. Simultaneously, millions of dollars worth of unused medicines expire annually in households while patients cannot afford them. Recent technological advancements in mobile computing, GPS services, and cloud infrastructure now enable the creation of responsive, scalable solutions. LifeLink builds upon these technologies, leveraging Firebase's real-time database, Google Maps API, and Flutter's cross-platform capabilities to bridge the gap between available donors and urgent medical needs across Pakistan.

1.3 Scope

1.3.1 In-Scope:

- Mobile application development for Android and iOS using Flutter framework
- Firebase backend integration (Authentication, Firestore, Cloud Storage, Cloud Functions)
- Real-time donor-recipient matching with location-based services
- Comprehensive blood donation management system with type compatibility checking
- Medicine donation hub featuring expiry tracking and prescription validation
- Organ pledge registration with digital documentation
- Emergency SOS system with priority queuing (First-Come-First-Serve)
- Push notification system using Firebase Cloud Messaging

- User profile management with donation history and recognition badges
- Multi-language support (English and Urdu)

1.3.2 Out-of-Scope:

- Web-based administrative dashboard
- Integration with government health databases or hospital management systems
- Physical delivery logistics or courier services
- Laboratory test verification or medical certification
- Payment processing or financial transactions
- International expansion beyond Pakistan
- Advanced analytics or machine learning predictions
- Integration with wearable devices or IoT sensors

1.4 Objective(s)/Aim(s)/Target(s)

1. **Create Centralized Donor Registry**: Develop a nationwide database of verified blood, organ, and medicine donors accessible to authorized healthcare providers
2. **Enable Real-Time Matching**: Implement intelligent algorithms to match donors with recipients based on blood type compatibility, location proximity, and urgency
3. **Reduce Response Time**: Cut emergency response time from hours to minutes through automated notifications and GPS-based donor location tracking
4. **Minimize Medical Wastage**: Create a redistribution system for unused medicines with expiry tracking and prescription validation
5. **Increase Organ Pledges**: Simplify organ donation registration with digital documentation and family consent management
6. **Build Donor Community**: Foster a sustainable donation ecosystem through recognition systems, badges, and community engagement features
7. **Ensure Data Privacy**: Implement robust security measures compliant with healthcare data protection standards
8. **Achieve Nationwide Coverage**: Launch in major Pakistani cities with scalability for nationwide expansion

1.5 Challenges

1. **Data Privacy and Security**: Protecting sensitive health information while maintaining system accessibility
2. **Real-Time Synchronization**: Managing concurrent user interactions and data updates across thousands of devices
3. **Location Accuracy**: Ensuring precise GPS tracking for accurate distance calculations in urban and rural areas
4. **Emergency System Reliability**: Maintaining 99.9% uptime for critical emergency features
5. **Medical Validation**: Verifying donor eligibility without physical medical examinations
6. **Cultural Sensitivities**: Addressing misconceptions about organ donation in conservative communities
7. **Infrastructure Limitations**: Ensuring functionality in areas with poor internet connectivity
8. **Regulatory Compliance**: Navigating Pakistan's healthcare regulations and data protection laws
9. **User Trust Building**: Establishing credibility for a platform handling life-critical requests
10. **Scalability Management**: Preparing for rapid user growth without compromising performance

1.6 Learning Outcomes

1. **Full-Stack Mobile Development**: Mastery of Flutter framework for cross-platform application development
2. **Firebase Ecosystem Integration**: Comprehensive understanding of Firebase services including Auth, Firestore, and Cloud Functions
3. **Real-Time System Architecture**: Experience designing and implementing WebSocket-based real-time communication systems
4. **Location-Based Service Implementation**: Expertise in GPS integration, geofencing, and distance calculation algorithms
5. **Healthcare Technology Standards**: Understanding of HIPAA-compliant data handling and medical software requirements
6. **State Management Patterns**: Advanced implementation of Provider pattern for efficient state management
7. **Emergency Response System Design**: Knowledge of high-availability system architecture and failover mechanisms

8. **Cross-Cultural UX Design**: Experience designing interfaces for diverse user demographics with varying technical literacy
9. **Performance Optimization**: Techniques for optimizing mobile applications for low-end devices and poor network conditions
10. **Ethical Software Development**: Understanding of ethical considerations in life-critical healthcare applications

1.7 Nature of End Product

LifeLink will be a production-ready mobile application available on Google Play Store and Apple App Store, featuring a clean, intuitive interface designed for users with varying technical expertise. The application will operate as a Minimum Viable Product (MVP) with core functionality including user registration, blood donation matching, medicine listing, organ pledge registration, and emergency SOS features. The backend will utilize Firebase's scalable infrastructure, ensuring reliable performance under increasing user loads. While initially focused on major urban centers, the architecture will support nationwide expansion. The application will serve as a proof-of-concept for digital healthcare coordination in Pakistan, with potential for integration with hospital systems and government health initiatives in future versions.

1.8 Completeness Criteria

The project will be considered complete when:

1. **Core Features Functional**: All high-priority features (blood donation, emergency SOS, user authentication) operate without critical bugs
2. **Performance Targets Met**: Application responds within 2 seconds for critical operations on mid-range Android devices
3. **Security Standards Achieved**: All user data encrypted in transit and at rest, with proper authentication and authorization controls
4. **Testing Requirements Satisfied**: 90% code coverage achieved, with all critical user paths tested on actual devices
5. **Documentation Complete**: Comprehensive technical documentation, user guides, and deployment instructions prepared
6. **App Store Ready**: Application meets all requirements for Google Play Store and Apple App Store submission
7. **Backend Scalability Verified**: Firebase infrastructure configured to handle at least 10,000 concurrent users
8. **Localization Implemented**: Full English and Urdu language support with proper right-to-left text rendering
9. **Emergency Reliability Proven**: SOS system tested with simulated emergency scenarios and response times documented

10. **User Acceptance Testing Passed**: Feedback from pilot user group indicates satisfaction with core functionality

1.9 Business Goals

1. **Life-Saving Impact**: Reduce mortality rates by 30% for patients awaiting blood transfusions in pilot cities within the first year
2. **Donor Base Growth**: Register 50,000 active donors across Pakistan within 18 months of launch
3. **Medical Waste Reduction**: Redirect 1 million PKR worth of unused medicines from expiration to patients in need annually
4. **Organ Donation Increase**: Increase registered organ donors by 200% in participating cities through simplified digital registration
5. **Healthcare Partnership**: Establish collaborations with at least 10 major hospitals for verified donor programs
6. **Financial Sustainability**: Develop premium features for healthcare institutions to ensure long-term financial viability
7. **National Recognition**: Become the government-recommended donation platform for healthcare emergencies
8. **International Standards Compliance**: Achieve certification for healthcare data security and privacy protection
9. **Community Engagement**: Build an active community of regular donors through recognition programs and social features
10. **Research Contribution**: Provide anonymized data for public health research on donation patterns and healthcare needs

1.10 Literature Review

Existing Solutions Analysis:

1. **Blood Donation Apps in Pakistan**: Current solutions like "Blood Donors Pakistan" and "Blood Bank" offer basic directory services but lack real-time matching, location tracking, and emergency features. Most operate as simple contact lists rather than interactive platforms.
2. **International Models**: Applications like "Blood Donor" by American Red Cross and "OrganDonor.gov" by U.S. Department of Health provide sophisticated features but are tailored to Western healthcare infrastructures with established regulatory frameworks unavailable in Pakistan.
3. **Academic Research**: Studies published in Journal of Medical Systems (2022) demonstrate that location-based donor matching can reduce emergency response times by 65%. Research in Developing World Bioethics (2023) highlights cultural factors unique to South Asia that must inform organ donation platform design.

4. **Technology Infrastructure**: Firebase case studies show successful deployment of similar real-time applications in developing countries, with particular success in Indonesia's "DarahKita" blood donation platform which achieved 40,000 active users in 18 months.

5. **Medical Guidelines**: WHO's 2023 recommendations emphasize digital tracking systems for blood safety, while Pakistan Medical Association guidelines outline specific requirements for donor verification and medical data handling.

6. **Privacy Regulations**: While Pakistan lacks comprehensive data protection laws, international standards like GDPR Article 9 provide frameworks for handling special category data including health information.

Innovation Points for LifeLink:

- **Integrated Multi-Donation Platform**: First Pakistani application combining blood, organ, and medicine donation in a single ecosystem
- **Emergency-First Design**: Priority queuing system for critical cases with broadcast notifications to maximize response likelihood
- **Cultural Adaptation**: Interface and messaging designed specifically for Pakistani users, considering regional languages and cultural sensitivities
- **Offline Capability**: Basic functionality maintained during network outages, critical for areas with unreliable connectivity
- **Verified Donor Network**: Partnership model with hospitals for donor verification, addressing trust issues in existing systems
- **Community Building Features**: Social recognition systems tailored to local values of community service and altruism

1.11 Document Conventions

This Software Requirements Specification (SRS) follows consistent formatting conventions for clarity and readability:

1. **Headings**: All section headings use bold title case (e.g., **1.1 Product (Problem Statement)**)

2. **Subheadings**: Second-level headings use italicized sentence case (e.g., *In-scope features*)

3. **Technical Terms**: Code elements, database fields, and API endpoints appear in `monospaced font`

4. **Requirements Identification**: Functional requirements are numbered as **REQ-001**, **REQ-002**, etc.

5. **Use Case Notation**: Use cases follow **UC-001**, **UC-002** format

6. **Priority Indicators**: Requirements include priority labels: **(High)**, **(Medium)**, **(Low)**

7. **Cross-References**: Internal document references appear as [Section 2.3]
8. **External References**: Citations follow [Author, Year] format with full details in References section
9. **User Roles**: Generic user types appear in lowercase (e.g., donor, recipient, administrator)
10. **Status Indicators**: Project status uses color coding: Completed In Progress Pending
11. **Diagram Notation**: All diagrams use UML 2.5 standards with consistent notation
12. **Document Versioning**: Each section includes last updated date in footer for revision tracking

2 Overall Description

2.1 Product Features

LifeLink is organized into five core modules, each addressing specific aspects of the donation ecosystem:

1. User Management Module

- **User Registration & Authentication**: Secure sign-up with email/phone verification, social login options
- **Profile Management**: Comprehensive health profiles including blood type, medical history, donation preferences
- **Privacy Controls**: Granular settings for data visibility and notification preferences
- **Account Recovery**: Multi-factor authentication and secure password reset mechanisms

2. Blood Donation System

- **Smart Matching Algorithm**: Real-time matching based on blood type compatibility (A+, B-, O+, etc.), location proximity, and donor availability
- **Request Management**: Creation, tracking, and management of blood donation requests with status updates
- **Emergency Priority System**: FCFS (First-Come-First-Serve) queuing for critical cases with broadcast notifications
- **Donation Scheduling**: Calendar integration for appointment setting and reminder notifications
- **Post-Donation Tracking**: Follow-up system for donor health monitoring and recipient outcome tracking

3. Medicine Donation Hub

- **Inventory Management**: Catalog system for listing unused medicines with automatic expiry tracking
- **Prescription Validation**: Verification system for prescription-only medications
- **Smart Matching**: Connection between medicine donors and verified healthcare providers or patients
- **Expiry Alert System**: Automated notifications for approaching expiry dates
- **Quality Assurance**: Basic guidelines for medicine storage and condition verification

4. Organ Pledge Registry

- **Digital Pledge System**: Electronic organ donation pledge with legal documentation
- **Family Consent Management**: Integrated system for family notification and consent tracking
- **Educational Resources**: Information about organ donation process, benefits, and myths
- **Hospital Coordination**: Pre-registration with transplant centers and coordination systems
- **Pledge Modification**: Flexible system for updating or withdrawing pledges

5. Emergency Response Module

- **One-Tap SOS**: Immediate emergency activation with location sharing
- **Broadcast Notification**: Simultaneous alert to all nearby donors and hospitals
- **Real-Time Tracking**: Live tracking of responder locations and estimated arrival times
- **Priority Override**: Automatic prioritization over regular requests
- **Emergency Contact Integration**: Automatic notification to user's emergency contacts

2.2 User Classes and Characteristics

Primary User Classes:

1. Blood/Organ/Medicine Donors

- **Demographics**: Ages 18-65, varying technical proficiency, primarily urban with smartphone access
- **Frequency**: Regular donors (monthly), occasional donors (quarterly), one-time donors

- **Motivation**: Altruism, social recognition, religious/cultural fulfillment, personal experience
- **Key Needs**: Easy request response, flexible scheduling, safety assurance, recognition
- **Technical Proficiency**: Basic to moderate smartphone usage skills
- **Special Requirements**: Medical eligibility verification, privacy controls, scheduling flexibility

2. Recipients/Patients

- **Demographics**: All ages, often in stressful medical situations, varying technical proficiency
- **Frequency**: As-needed basis, often urgent or emergency situations
- **Motivation**: Immediate medical need, life-saving requirement, financial constraints
- **Key Needs**: Fast matching, reliable donors, clear communication, trust verification
- **Technical Proficiency**: Often low during medical emergencies, may rely on family assistance
- **Special Requirements**: Simple interface, quick onboarding, emergency prioritization

User Priority Ranking:

1. **Emergency Requesters** (Highest priority - life-critical)
2. **Recipients/Patients** (High priority - medical need)
3. **Donors** (High priority - resource providers)
4. **Healthcare Providers** (Medium priority - institutional users)
5. **Administrators** (Medium priority - system maintenance)
6. **Other Users** (Low priority - secondary roles)

2.3 Operating Environment

Hardware Requirements:

- **Mobile Devices**: Android smartphones (version 8.0+) or iOS devices (version 12.0+)
- **Minimum Specifications**:
  - RAM: 2GB minimum, 3GB recommended
  - Storage: 100MB free space for app installation and cache
  - Processor: Quad-core 1.5GHz minimum

  - Screen Size: 4.5 inches minimum, responsive design for all sizes
  - Camera: Required for document scanning and verification
  - GPS: Required for location services with assisted-GPS support
  - Internet Connectivity: Wi-Fi or mobile data (3G minimum, 4G recommended)

Software Environment:

- **Operating Systems**:
  - Android 8.0 (Oreo) and above
  - iOS 12.0 and above
- **Development Framework**: Flutter 3.0+ with Dart programming language
- **Backend Services**: Firebase ecosystem
  - Firebase Authentication (email, phone, Google sign-in)
  - Cloud Firestore (NoSQL database)
  - Firebase Storage (media files)
  - Cloud Functions (serverless backend)
  - Firebase Cloud Messaging (push notifications)
  - Firebase Analytics & Crashlytics
- **External APIs**:
  - Google Maps API for location services
  - SMS Gateway for OTP verification
  - Email Service (SMTP) for notifications

2.4 Design and Implementation Constraints

Technical Constraints:

1. **Cross-Platform Consistency**: Must maintain identical functionality and user experience on both Android and iOS platforms
2. **Firebase Dependency**: Core application functionality depends on Google Firebase services availability and pricing tiers
3. **GPS Requirement**: Location services are mandatory for donor-recipient matching; app functionality is limited without GPS access
4. **Real-Time Data Synchronization**: Requires constant WebSocket connections which may impact battery life on mobile devices
5. **Image Size Limitations**: Maximum 5MB per upload for document scanning and verification
6. **API Rate Limits**: Google Maps API has daily request quotas; application must implement efficient usage patterns

7. **Device Fragmentation**: Must support wide range of Android device manufacturers with varying hardware capabilities
8. **Network Dependency**: Critical features require active internet connection; offline functionality is limited

Development Constraints:

1. **Single Developer Limitation**: All design, development, testing, and deployment handled by one individual
2. **Timeframe**: 6-month development cycle for Minimum Viable Product (MVP)
3. **Budget Limitations**: Dependence on free tiers of Firebase and Google services with strict quota management
4. **Testing Device Availability**: Limited physical device testing, reliance on emulators for most testing scenarios
5. **Medical Regulation Navigation**: Compliance with Pakistan's evolving healthcare technology regulations
6. **Documentation Requirements**: Comprehensive documentation needed for future maintenance and potential handover
7. **Third-Party Library Risks**: Dependence on community-maintained Flutter packages with varying update cycles

2.5 Assumptions and Dependencies

Critical Assumptions:

User Behavior Assumptions:

1. **Smartphone Accessibility**: Target users have access to Android smartphones with basic functionality
2. **Internet Availability**: Users have at least intermittent internet access for app synchronization
3. **Technical Literacy**: Users possess basic smartphone operation skills (installing apps, basic navigation)
4. **Language Proficiency**: Primary users understand either English or Urdu sufficiently for app interaction
5. **Medical Honesty**: Users provide accurate health information when registering as donors
6. **Emergency Responsiveness**: Users will respond to emergency notifications when available and able
7. **Family Involvement**: Organ donation decisions involve family consultation in Pakistani culture

Key Dependencies:
Technical Dependencies:

1. **Firebase Services**: Complete dependency on Google Firebase for authentication, database, storage, and hosting
2. **Google Maps API**: Critical dependency for location services, distance calculation, and mapping features
3. **Mobile Operating Systems**: Dependency on Android and iOS platform updates and compatibility
4. **Flutter Framework**: Dependency on Flutter's continued development and Dart language evolution
5. **Third-Party Libraries**: Dependencies on community packages for specific functionalities (camera, image processing, etc.)

Risk Mitigation Strategies:
For Technical Dependencies:

1. **Alternative Services Identified**: Backup plans for critical services (alternative mapping solutions, etc.)
2. **Graceful Degradation**: Features degrade gracefully when dependencies are unavailable
3. **Regular Monitoring**: Continuous monitoring of service health and performance
4. **Cost Management**: Budget planning for potential service tier upgrades

3 Specific Requirements

3.1 Functional Requirements

Table 1: Functional Requirements

ID	Description	Priority	Status
REQ-001	User Registration with email/phone verification	High	Pending
REQ-002	Blood type compatibility checking algorithm	High	Pending
REQ-003	Real-time donor-recipient matching	High	Pending
REQ-004	GPS-based location tracking	High	Pending
REQ-005	One-tap emergency SOS system	High	Pending
REQ-006	Medicine donation with expiry tracking	Medium	Pending
REQ-007	Organ pledge registration system	Medium	Pending
REQ-008	Push notification system	High	Pending
REQ-009	Donation history tracking	Low	Pending
REQ-010	Multi-language support (English/Urdu)	Medium	Pending

3.2 Non-functional Requirements

Table 2: Non-functional Requirements

Category	Requirement
Performance	Application must respond within 2 seconds for critical operations
Availability	Emergency features must maintain 99.9% uptime
Security	All health data must be encrypted both in transit and at rest
Usability	Interface must be usable by people with basic smartphone literacy
Scalability	System must support up to 100,000 concurrent users
Reliability	Critical functions must have 95% success rate in testing

3.3 Use Case Descriptions

Table 3: Use Case UC-001: Register as Donor

Attribute	**Description**
Use Case ID	UC-001
Use Case Name	Register as Donor
Actor	New User
Preconditions	User has installed app and has valid email/phone
Postconditions	User profile created and donor status active
Basic Flow	1. User opens app and selects "Register" 2. Enters personal information and health details 3. Verifies email/phone via OTP 4. Sets donation preferences 5. Completes registration
Alternative Flow	If verification fails, system allows resend OTP
Priority	High

Table 4: Use Case UC-002: Request Blood Donation

Attribute	**Description**
Use Case ID	UC-002
Use Case Name	Request Blood Donation
Actor	Recipient/Patient
Preconditions	User is logged in and has verified profile
Postconditions	Blood request created and donors notified
Basic Flow	1. User selects "Request Blood" 2. Enters patient details and blood type needed 3. Sets urgency level and location 4. Submits request 5. System matches with donors and sends notifications
Alternative Flow	If emergency, triggers SOS protocol
Priority	High

4 Dynamic Behavior

4.1 Sequence Diagrams

Blood Donation Request Flow:

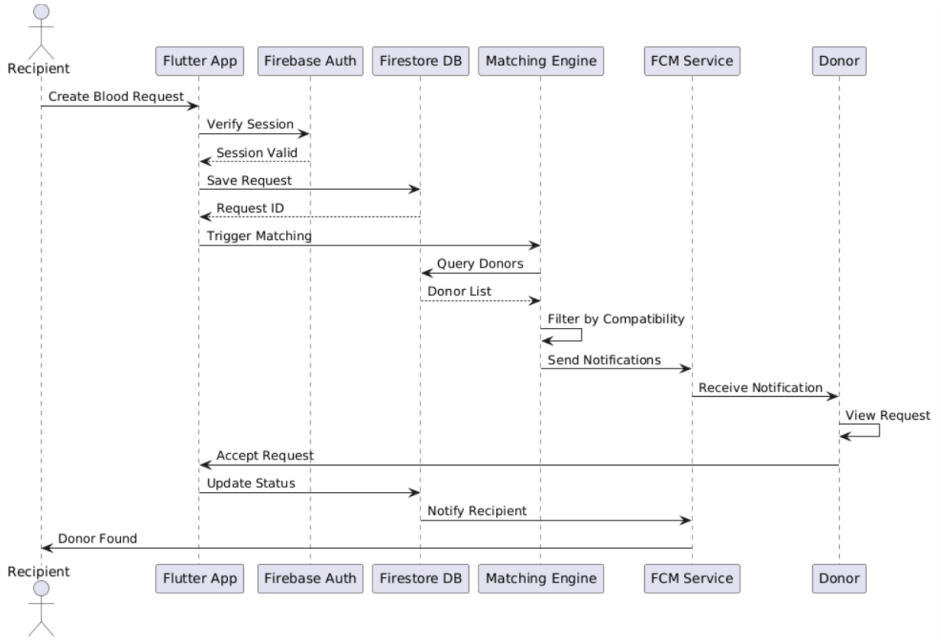

Figure 1: Sequence Diagram: Blood Donation Request Flow

User Authentication Process:

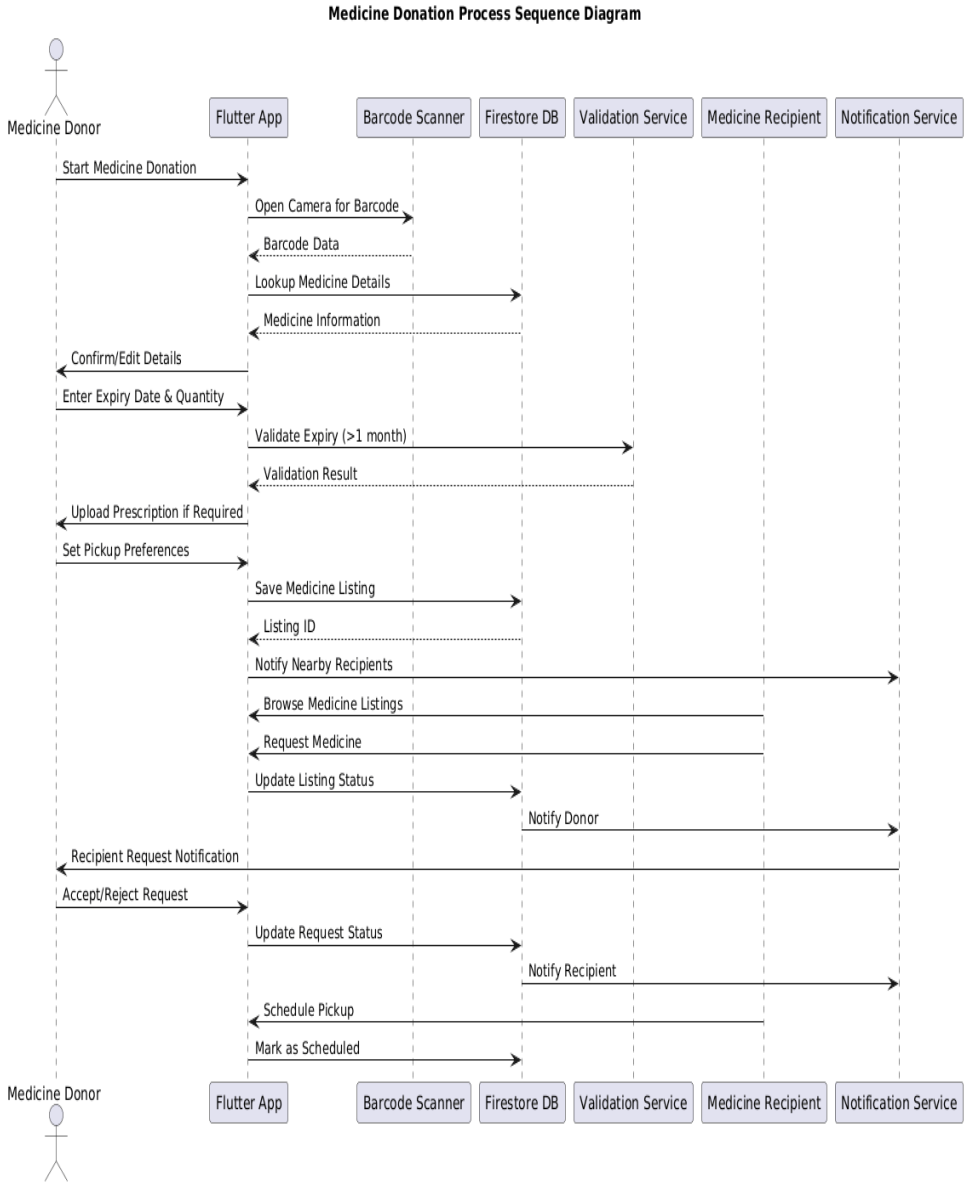

Figure 2: User Authentication Process

Location Determination Process:

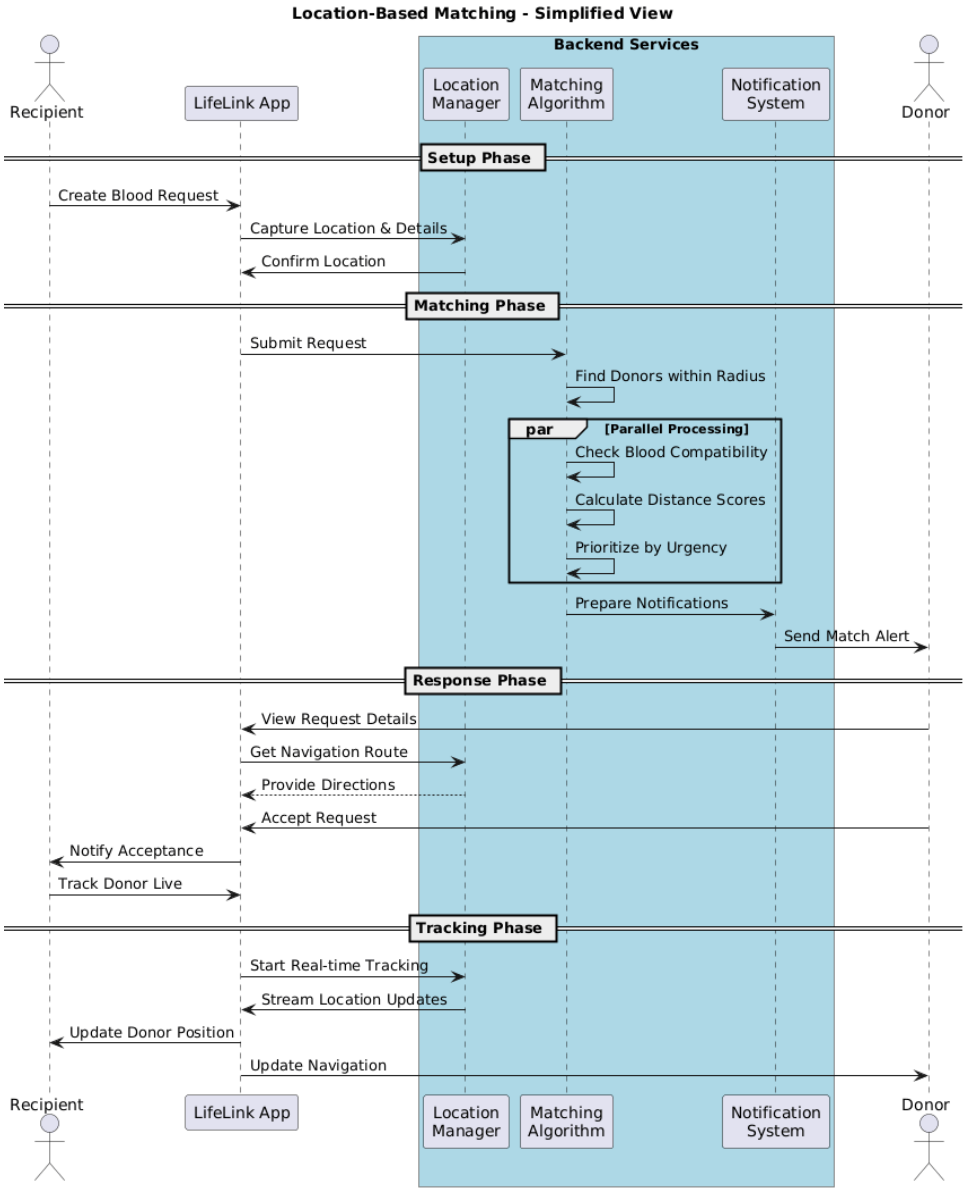

Figure 3: Location Determination Process

Medicine Donation Process:

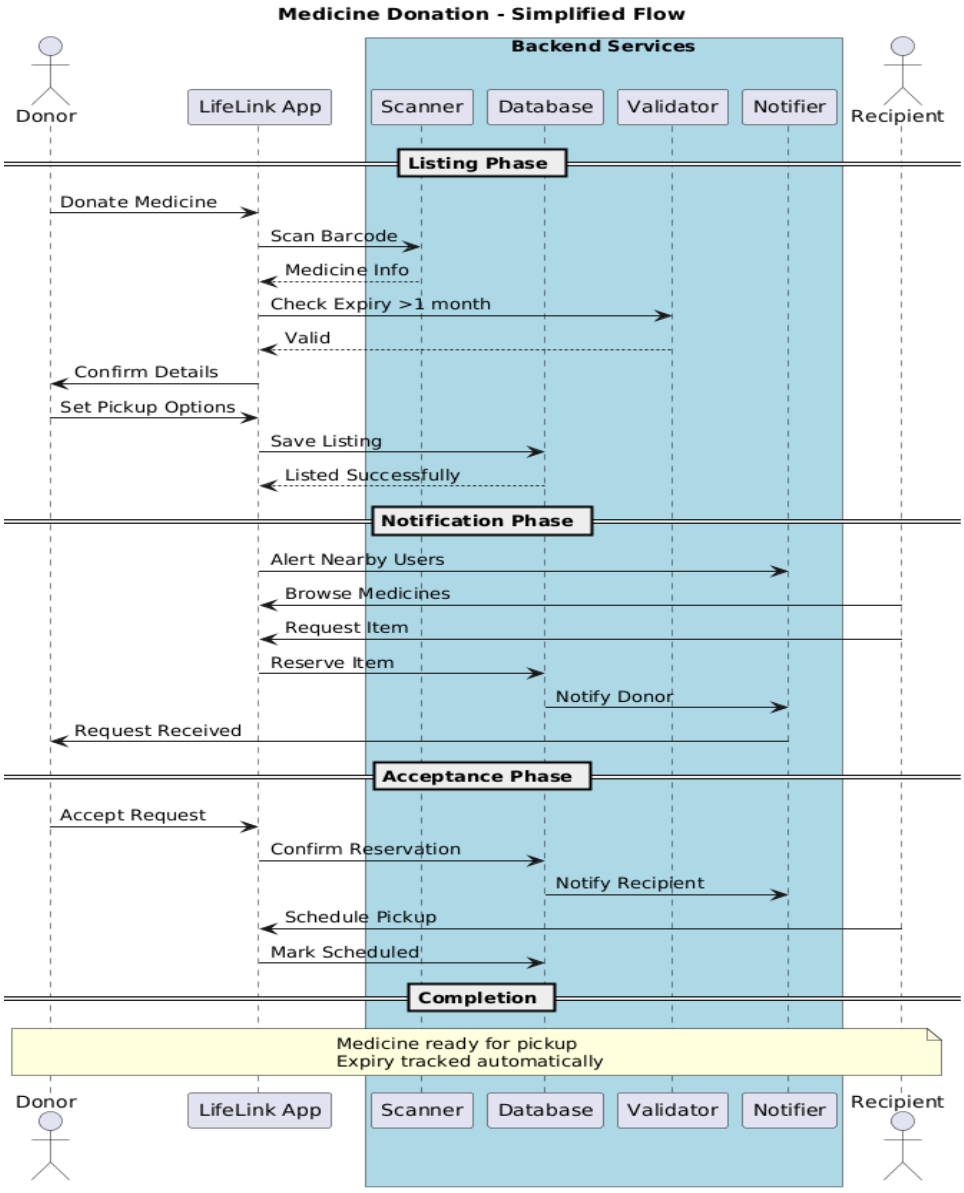

Figure 4: Medicine Donation Process

Organ Donation Process:

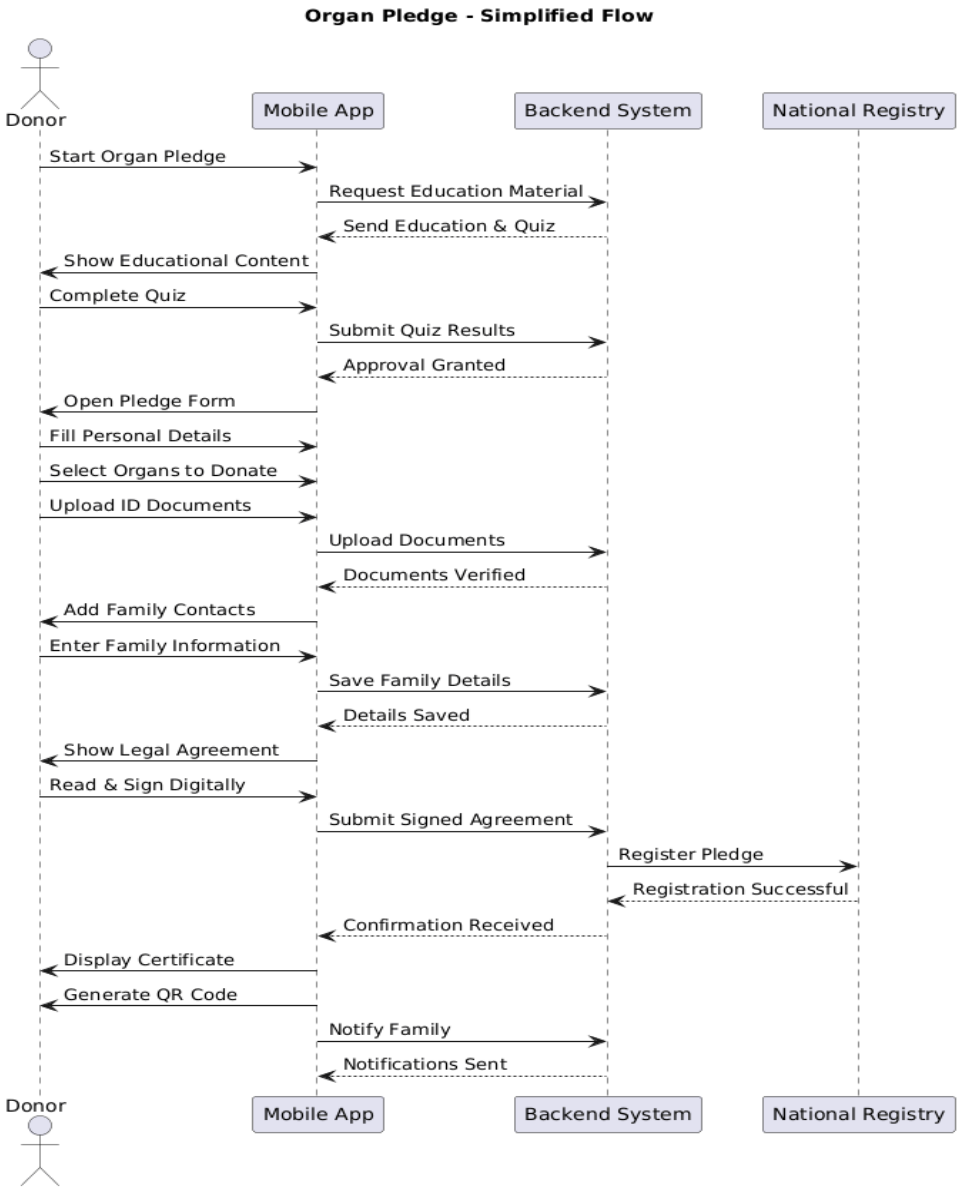

Figure 5: Organ Donation Process

4.2 State Diagrams

Blood Request State Machine:

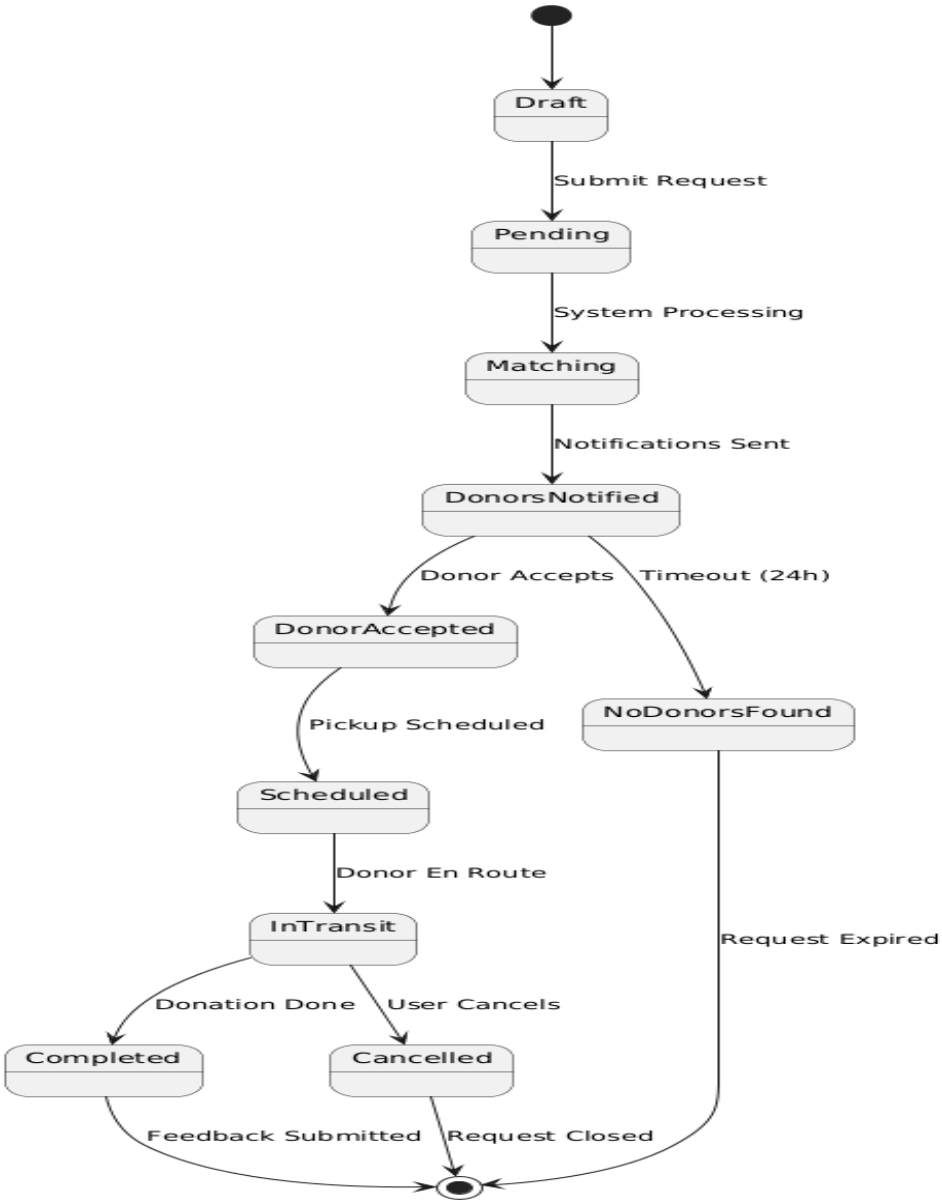

Figure 6: Blood Donation State

User Authentication State Machine:

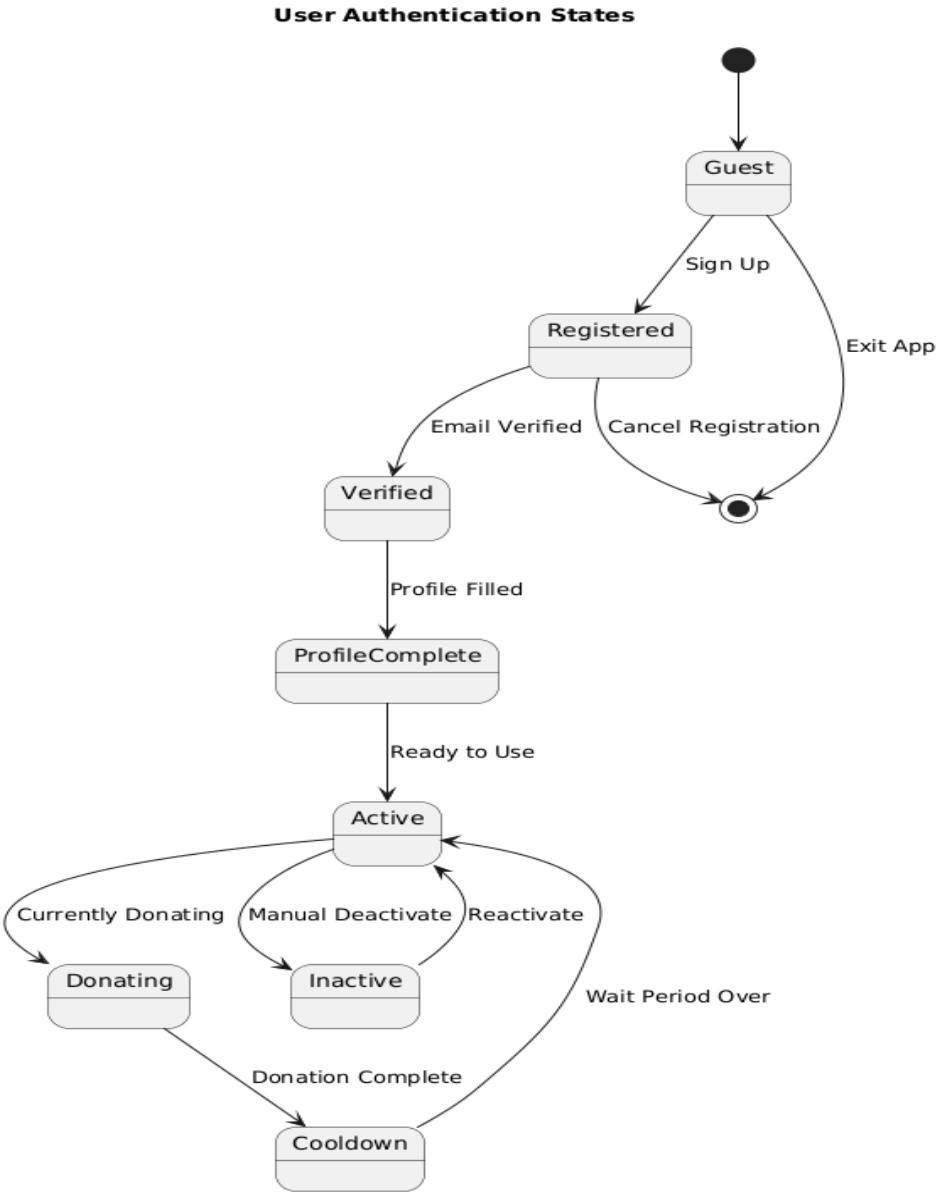

Figure 7: User Authentication State

Medicine Donation State Machine:

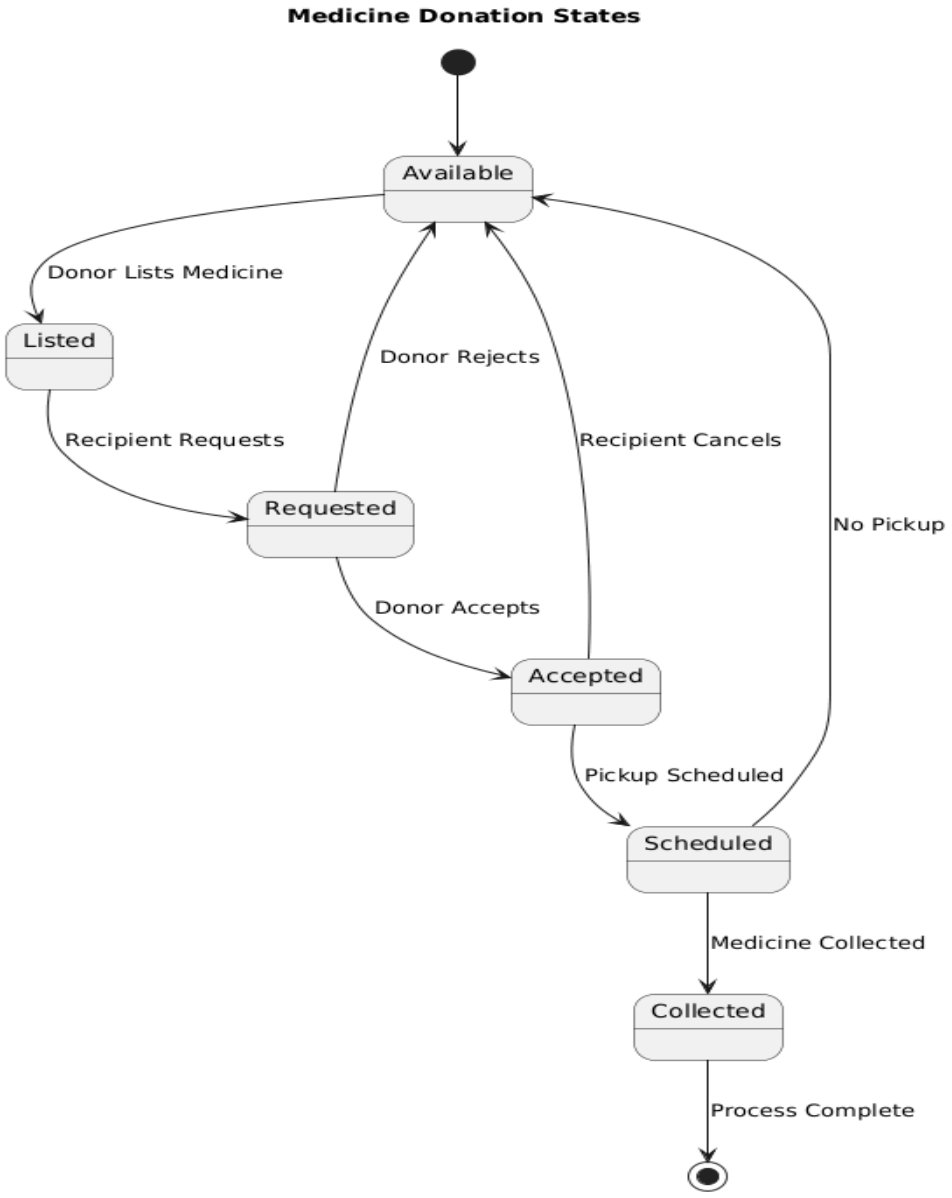

Figure 8: Medicine Donation State

Organ Pledge State Machine:

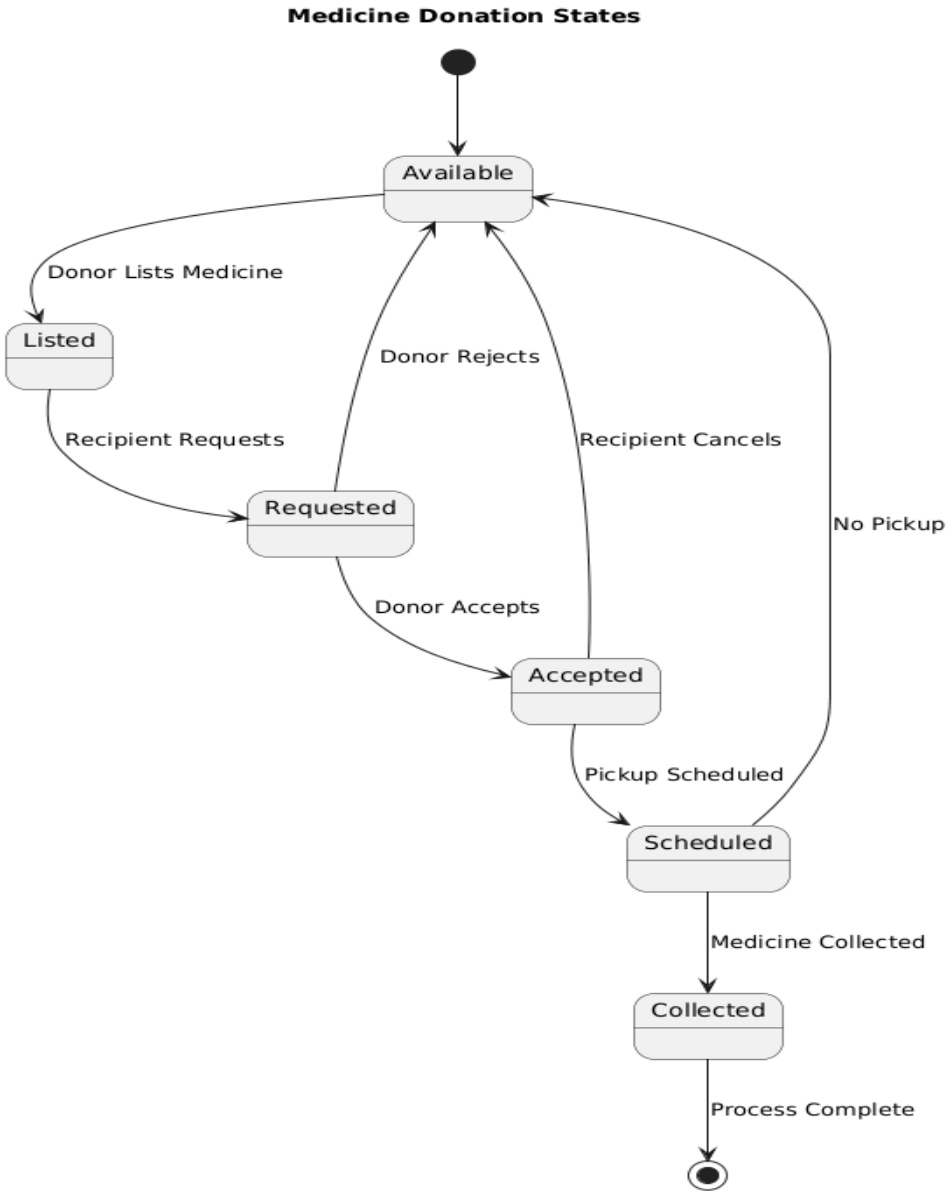

Figure 9: Organ Pledge State

4.3 Activity Diagrams

Complete Donation Workflow:

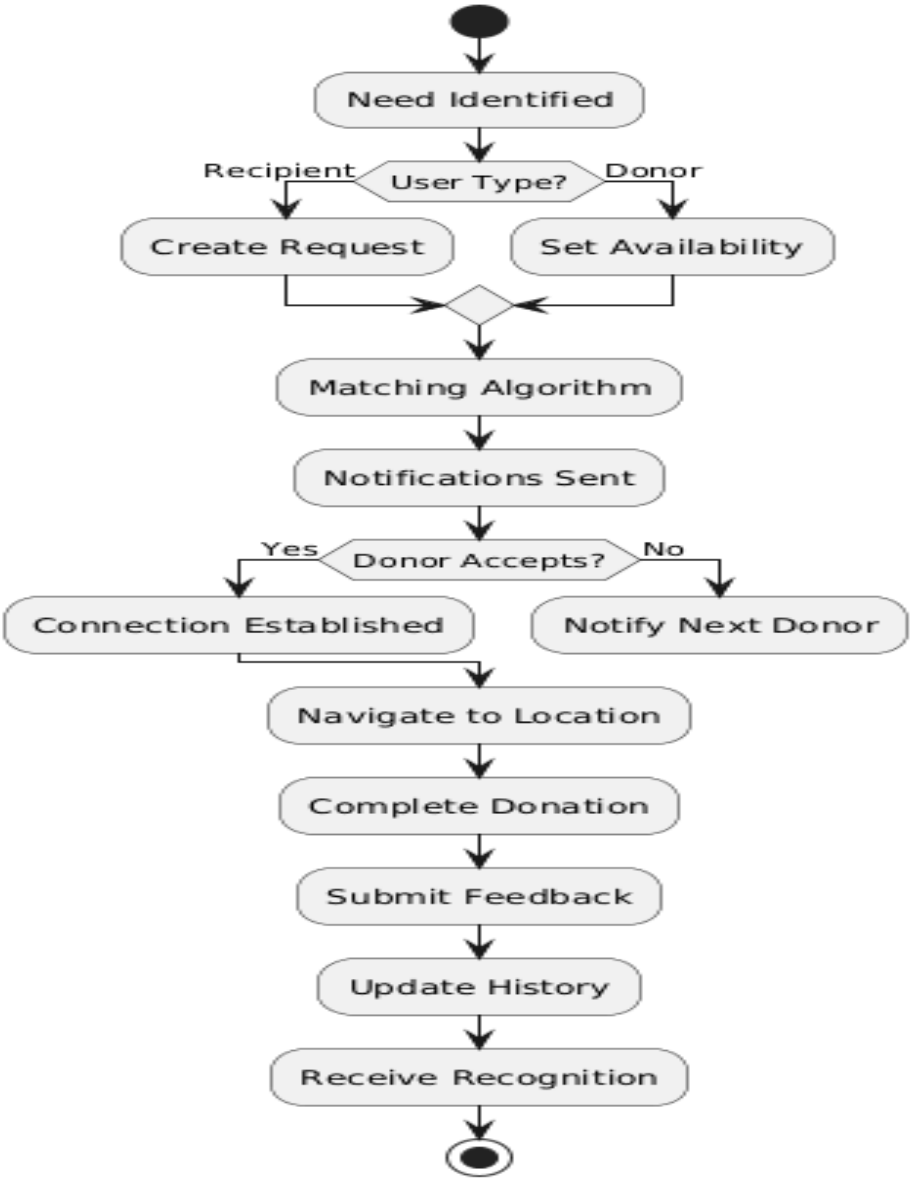

Figure 10: Organ Pledge State

5 Other Requirements

5.1 Database Schema

Listing 1: Firestore Collections Structure

```
// Users Collection
users/{userId} {
  uid: string,
  email: string,
  phone: string,
  bloodType: string,
  location: geopoint,
  medicalHistory: array,
```

```
  donorStatus: string,
  createdAt: timestamp
}

// Blood Requests Collection
blood_requests/{requestId} {
  patientName: string,
  bloodTypeNeeded: string,
  unitsRequired: number,
  hospital: string,
  urgency: string,
  location: geopoint,
  status: string,
  createdAt: timestamp
}

// Medicine Listings Collection
medicine_listings/{medicineId} {
  medicineName: string,
  expiryDate: timestamp,
  quantity: number,
  donorId: string,
  status: string,
  listedAt: timestamp
}

// Organ Pledges Collection
organ_pledges/{pledgeId} {
  donorId: string,
  organsPledged: array,
  legalDocument: string,
  status: string,
  pledgeDate: timestamp
}
```

5.2 API Endpoints

Table 5: API Endpoints

Method	Endpoint	Description
POST	/api/auth/register	User registration
POST	/api/auth/login	User authentication
POST	/api/blood/request	Create blood request
GET	/api/blood/requests	Get nearby requests
PUT	/api/blood/request/id	Update request status
POST	/api/medicine/donate	List medicine for donation
POST	/api/organ/pledge	Register organ pledge
POST	/api/emergency/sos	Activate emergency SOS
GET	/api/user/profile	Get user profile
PUT	/api/user/profile	Update user profile

5.3 Error Handling

Table 6: Error Codes and Handling

Error Code	Type	Handling Strategy
ERR-001	Authentication Error	Prompt re-login, clear local cache
ERR-002	Network Error	Show offline mode, retry automatically
ERR-003	Location Error	Use last known location, prompt GPS enable
ERR-004	Database Error	Cache data locally, sync when available
ERR-005	Validation Error	Show specific field errors, guide correction
ERR-006	Emergency Error	Switch to SMS/phone fallback
ERR-007	Permission Error	Guide to settings, provide alternatives

6 References

1. Google Flutter Documentation. (2024). *Flutter Framework Guide*. Available at: https://flutter.dev/docs

2. Firebase Documentation. (2024). *Firebase Services Guide*. Available at: https://firebase.google.co

3. World Health Organization. (2023). *Guidelines for Blood Safety*. WHO Press.

4. Pakistan Medical Association. (2023). *Healthcare Technology Standards*. PMA Publications.

5. Android Developers. (2024). *Android Development Guidelines*. Available at: https://developer.android.com

Appendix A: Glossary

Table 7: Glossary of Terms

Term	Definition
SRS	Software Requirements Specification - document describing software functionality
MVP	Minimum Viable Product - initial version with core features only
FCFS	First Come First Serve - emergency request priority system
GPS	Global Positioning System - location tracking technology
FCM	Firebase Cloud Messaging - push notification service
OTP	One-Time Password - secure authentication method
API	Application Programming Interface
UI/UX	User Interface / User Experience
NoSQL	Non-relational database (Firestore)
Geofencing	Virtual perimeter for real-world geographic area

Appendix B: Data Models

Listing 2: User Model

```
class UserModel {
  String uid;
  String email;
  String phone;
  String fullName;
  String bloodType;
  DateTime dateOfBirth;
  String gender;
  GeoPoint location;
  List<String> medicalHistory;
  String donorStatus;
  DateTime lastDonation;
  int totalDonations;
  DateTime createdAt;
}
```

Listing 3: Blood Request Model

```
class BloodRequestModel {
  String requestId;
  String patientName;
  String bloodTypeNeeded;
  int unitsRequired;
  String hospital;
  String urgency;
  GeoPoint location;
```

```
  String status;
  String donorId;
  DateTime createdAt;
  DateTime expiresAt;
}
```