# Detailed Software Requirements Specification for *Helping Unity*

HPLH-MAIN-DSRS-01

# TABLE OF CONTENTS

## Document Revisions

| Date | Version | Description | Author |
|---|---|---|---|
| 28/02/2025 | 1.0 | First SRS document | M.M.Naja |
| | | | |
| | | | |

## Document Approval

Quality Software Corporation and Helping Unity have reviewed this document and hereby agree that the contents herein are accurate. Any changes to this document must be communicated in writing and signed off by both parties.

| Signature | Signature |
|---|---|
| Date: 28/02/2025 | Date: 28/02/2025 |
| Name: M.M.Naja | Name: Asantha Yagika |
| Customer: Quality Software Corporation | Helping Unity Inc. |

# Introduction

## *1.1 Purpose*

The purpose of this web application is to create a bridge between donors and individuals in need. It serves as a secure and reliable platform for various types of donations, ensuring that contributions reach the intended recipients.

## *1.2 Summary*

The platform covers a wide range of donations, including used items, monetary contributions, services, blood, and even human organs. While facilitating donations, it places significant emphasis on privacy protection for recipients, allowing them to maintain their dignity.

## *1.3 Company Overview*

Helping Unity Inc is a reputable software company made by our team, which is dedicated to producing the best quality software products. *We are specialized in Cloude, Web, Mobile, and IOT products. We are more empathetic toward AI based software applications.*

## *1.4 Project Overview*

The web application aims to establish a trustworthy environment. Donors can confidently contribute, knowing that their donations will not be misused. For recipients, the platform ensures confidentiality by sharing personal information only with relevant parties, preserving their dignity.

## *1.5 Scope*

The scope of the project includes:

- Creating a user-friendly web application for donations.
- Allowing donors to contribute items, funds, services, blood, and organs.
- Ensuring recipient privacy and dignity.
- Integrating with payment gateways for monetary donations.

### *1.6 Assumptions*

- Users have internet access.
- Donors and recipients are genuine and have legitimate intentions.

### *1.7 Definitions, Acronyms and Terminology*

- **Donors:** Individuals or organizations contributing to the platform.
- **Recipients:** Needy individuals seeking assistance.
- **Platform:** The web application connecting donors and recipients.

### *1.8 References (Optional)*

- Client's project description.

## 2 Project Scope and Impact

This website is committed to creating a reliable, trustworthy, and secure environment for the doners and the people who require help.

### *2.1 Scope Inclusions*

The following functionalities are within the scope of the "Helping Unity" web application:

1. **Donor Registration and Authentication:**
    - Donors can create accounts, log in securely, and manage their profiles.
    - Authentication mechanisms ensure the validity of donors.
    - Provide help.
2. **Recipient Registration and Request Submission:**
    - Needy individuals can register as recipients.
    - Recipients can submit requests for various types of assistance (items, funds, services, blood, organs).
    - Request for help
3. **Donation Tracking and Reporting:**
    - The system tracks donations, including their type (e.g., monetary, item, blood, organs).
    - Reporting features allow donors and administrators to view donation history.
4. **Trustee Registration and Authentication:**
    - Trustees can create accounts, log in securely, and manage their profiles.

- Authentication mechanisms ensure the legitimacy of trustees.
- Endorse recipient's requirement is genuine or not.

5. **Manage Donors, Recipients, and Trustee**
   - Block or suspend donors, recipients, and trustees.

### *2.2 Scope Exclusions*

We, as the project team, are responsible for providing the required images, domain name, and hosting. Our team will not provide these services externally. If needed, our team can offer expertise and suggestions in these areas.

### *2.3 Impact on other systems*

The system will not impact any existing systems already owned by the Helping Unity company. Furthermore, the system is self-sufficient, any other system will not impact on Helping Unity application.

#### 2.3.1 Affected by other systems.

The system is self-sufficient, any other system will not impact on Helping Unity application.

#### 2.3.2 Effects on Other Systems

The system will not impact any existing systems.

## 3 Functional Requirements

#### 3.1.1 Function 1

Figma Link: https://www.figma.com/design/MET6MyxfjjhjSiA7dAr6Ol/Helping-Unity?node-id=2-2&t=MvujdDQvezzY4c5D-1

**Navigation Bar**

**Functionality:** The navigation bar provides easy access to different sections of the platform, ensuring users can navigate efficiently.

**Process Flow:**

1. **User Interaction:** User clicks on a navigation tab (Donate, Fundraise, Payment, About, or Search).
2. **Action:** The system highlights the selected tab and loads the corresponding page.
3. **Feedback:** Visual feedback (highlight) indicates the active tab.

**Screen Layout:**

- **Top Horizontal Bar:** Contains navigation tabs and the logo.
- **Active Tab Highlight:** Clearly indicates which section is currently active.
- **Search Bar:** Positioned on the right, allows users to search the platform.

### Donation Page

**Functionality:** Allows users to view and select from different types of donations they can make.

**Process Flow:**

1. **Page Load:** User navigates to the donation page.
2. **Display Options:** The system displays four categories (Used/New Items, Money Donations, Human Organs, Blood Donations).
3. **User Interaction:** User clicks on a "Donate Now" button under a category.
4. **Action:** The system redirects a detailed donation form for the selected category.

**Screen Layout:**

- **Donation Categories:** Four rectangular sections with images and "Donate Now" buttons.
- **Pagination Controls:** At the bottom, allowing navigation between pages.

### User Profile Page

**Functionality:** Allows users to view and update their profile information.

**Process Flow:**

1. **Page Load:** User navigates to the profile page.
2. **Display Information:** The system displays the user's profile picture, username, email, and description.
3. **Edit Interaction:** User clicks the "Update" button.
4. **Action:** The system allows the user to edit their information and save changes.

**Screen Layout:**

- **Profile Icon:** Placeholder for the user's profile picture.
- **User Details:** Fields displaying username, email, and description.
- **Update Button:** Positioned for easy access to initiate edits.

## Payment Methods

**Functionality:** Enables users to view and manage their payment options and transaction history.

**Process Flow:**

1. **Page Load:** User navigates to the payment page.
2. **Display Payment Options:** The system displays available payment methods (VISA, MasterCard etc.).
3. **User Interaction:** User selects a payment method for donating.
4. **Transaction History:** Users can view their past transactions.

**Screen Layout:**

- **Payment Icons:** Visually represented payment methods.
- **Transaction List:** Detailed view of past transactions.

## System Administration

**Functionality:** Allows administrators to manage and administer the system, including database maintenance.

**Process Flow:**

1. **Admin Access:** Administrator logs into the system with appropriate credentials.
2. **Dashboard Display:** The system displays an administration dashboard with various management options.
3. **Database Maintenance:** Administrator can add, update, or delete entries in system tables and parameters.
4. **Remote Management:** Administrators can perform tasks remotely through a management console.
5. **Feedback:** System provides success or error messages based on actions performed.

**Screen Layout:**

- **Admin Dashboard:** Central hub with links to different administrative functions.
- **Management Console:** Interface for performing database maintenance tasks.
- **Feedback Messages:** Informative messages about the status of administrative tasks.

### 3.1.2 Data Archival and Retention

Data Retention Policy

**Purpose:** To define how long different types of data are retained within the system before being archived or deleted.

**Policy Details:**

- **User Data:**
  - **Active Users:** Retained indefinitely until the user requests deletion.
  - **Inactive Users:** User accounts with no activity for more than 2 years will be marked inactive and retained for an additional year before being archived.
  - **Deleted Accounts:** User data from deleted accounts will be retained for 6 months post-deletion to allow for recovery in case of accidental deletion, then permanently deleted.

- **Donation Records:**
  - **Transaction Data:** Retained for 7 years to comply with financial regulations.
  - **Beneficiary Data:** Retained for 5 years after the last transaction.
- **Payment Information:**
  - **Payment Methods:** Stored securely and retained for 5 years after the last transaction or update.
  - **Transaction Logs:** Retained for 7 years to comply with audit requirements.
- **System Logs:**
  - **Access Logs:** Retained for 1 year.
  - **Error Logs:** Retained for 1 year unless flagged for review, in which case retained until resolved.

**Data Archival Policy**

**Purpose:** To move infrequently accessed data to a secure archival storage to optimize system performance while ensuring data is still available if needed.

**Policy Details:**

- **Archival Triggers:**
  - User accounts marked as inactive for over 3 years.
  - Transaction records older than 7 years.
  - System logs older than 1 year.
- **Archival Process:**
  - Data identified for archival will be moved to a secure, read-only storage.
  - Archived data will be compressed and encrypted for security.
  - Metadata about archived data will be maintained to facilitate easy retrieval if necessary.
- **Access to Archived Data:**
  - Only system administrators with appropriate permissions can access archived data.
  - Access to archived data will require an additional layer of authentication.

## Data Deletion Policy

**Purpose:** To ensure data is securely and permanently deleted after the retention period ends.

**Policy Details:**

- **Deletion Process:**
  - Data identified for deletion will be securely erased using industry-standard methods.
  - Deletion logs will be maintained for audit purposes.
- **User-Initiated Deletion:**
  - Users can request deletion of their data at any time. Upon request, data will be deleted within 30 days, except where retention is required by law.

## Compliance and Audit Requirements

**Purpose:** To ensure the platform complies with legal, regulatory, and organizational policies.

**Policy Details:**

- **Regular Audits:**
  - Annual audits to ensure compliance with data retention and archival policies.
  - Audit logs will be retained for 5 years.
- **Legal Holds:**
  - In case of legal inquiries or investigations, data relevant to the inquiry will be retained until the legal hold is lifted, regardless of the standard retention period.
- **Data Protection Regulations:**
  - Compliance with GDPR, CCPA, and other relevant data protection regulations.

### 3.1.3 User Profiles, Roles and Privileges

- **Role Differentiation:**
  - The system should differentiate between donor, recipient, Trustee, and Admin roles. Each role has distinct permissions and responsibilities specific to the role.
- **Privileges Assignment:**
  - Assign appropriate privileges to users based on their roles. For example:
    - Donors can view recipient profiles and submit donations.
    - Recipients can manage their requests and view relevant information.

### 3.1.4 Reporting Requirements

- **Donation Reports:**
  - Generate reports summarizing donation activity. Examples include:
    - Total funds raised over a specific period.
    - Items donated (quantity and type).
- **Real-Time Updates:**
  - Provide real-time updates to donors and recipients regarding their contributions and requests.

# 4 Non-Functional Requirements

### 4.1.1 Performance and Load Requirements

| | |
|---|---|
| Availability | The application should available 24/7 with 99.9 uptime |
| Reliability | The system should recover from failures within 5 minutes. |
| Performance | The system should handle 100 concurrent users with an average response time of less than 15 seconds. |
| Scalability | The system must be able to scale horizontally to support a 100% increase in user base (Double the user base) without performance reduction. |
| Usability | The system should achieve a usability score of 90% in user testing. |
| Security | The system must encrypt all sensitive data |
| Maintainability | The system should allow updates and maintenance with minimal downtime, ideally not exceeding 30 minutes. |
| Localization | The system should support multiple languages and regional formats. |
| Interoperability | The system should be able to integrate with third-party payment gateways. |
| Accessibility | The system should support accessibility for elders and differently abled people. |
| Auditability | The system should log all important user actions and changes for audit purposes. |

### 4.1.2 Compatibility Requirements

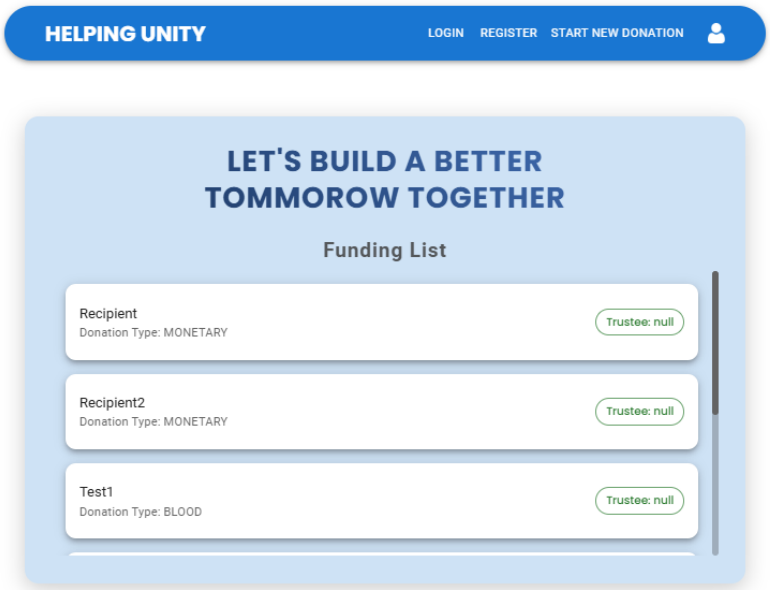
| | |
|---|---|
| HTML Versions to be supported | HTML5 |
| Browser Versions to be supported | Google Chrome, Mozilla Firefox, Microsoft Edge, Safari (latest versions) |
| Database Versions to be supported | MySQL 8.0 and later |
| Communication Protocol | HTTP/HTTPS |
| Platform Version to be supported | Windows 10 and later, macOS 11 and later, Linux |
| Any other external systems or standards | OAuth 2.0 for authentication, RESTful APIs for external service integration |

### 4.1.3 External Interface Requirements

- **Payment Gateways Integration:**
  - Integrate with payment gateways (e.g., PayPal, Stripe) to handle monetary donations securely. Use APIs provided by these services.
- **Secure Communication Channels:**
  - All **communication** between the application and external services (including APIs) should use HTTPS to encrypt data in transit.

#### 4.1.3.1 User Interfaces

- **Home (Landing)**




***Top Navigation Bar***

- **Items:**
  - **Donate**
  - **Fundraise**
  - **Payment**
  - **Search**
  - **About**

- **Center Logo:** A logo with the text "Helping Unity" underneath.

**Main Content Area**

- **Two Main Sections:**
  - **Requirement**
    - **Subtext:** "Join Us in"
    - **Button:** "Raising Funds Today!"
  - **Your Donations**
    - **Subtext:** "Make a Difference"
    - **Button:** "Donate Now!"

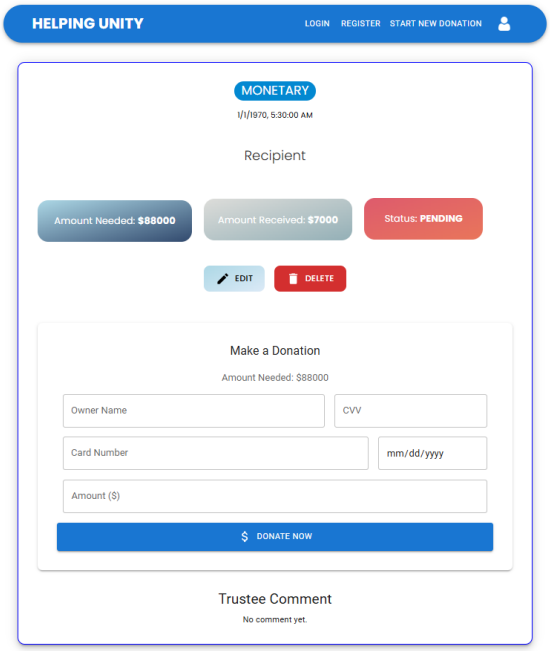
**Design Specifications:**

1. **Top Navigation Bar:**
   - Should be a horizontal bar at the top of the page.
   - Items should be evenly spaced.
   - The logo should be centrally aligned within the navigation bar.
2. **Main Content Area:**
   - Contains two main rectangular sections side by side.
   - Both sections have a heading, a subtext, and a button.
   - Sections should be equal in size and have round corners.
   - Buttons should have a distinct border and background.

**Special User Interface Requirements:**

- **Responsiveness:** The interface should be responsive and adapt to different screen sizes, ensuring readability and accessibility on mobile devices.
- **Accessibility:** Ensure the interface is accessible to users with disabilities, including keyboard navigation, screen reader compatibility, and high-contrast color schemes.
- **Consistency:** Maintain consistent styling across all elements, including fonts, colors, and button styles.
- **Interactivity:**
  - Buttons should have hover effects to indicate they are clickable.
  - Navigation items should provide visual feedback when selected.

- **Donate**




**Donation Options Section:**

- **Four Donation Categories:**
  - **Used/New Items**
    - **Image:** Picture of items in a donation box.
    - **Button:** "Donate Now"
  - **Money Donations**
    - **Image:** Picture of a jar labeled "Donate" with coins inside.
    - **Button:** "Donate Now"
  - **Human Organs**
    - **Image:** Hands holding a heart-shaped sign that says, "Organ donors save lives".
    - **Button:** "Donate Now"
  - **Blood Donations**
    - **Image:** Heart and blood drop graphic with the text "Donate Blood".
    - **Button:** "Donate Now"

**Pagination:**

- **Pagination Controls:**
  - **Page Number 1:** Highlighted to indicate the current page.
  - **Page Number 2:** Indicating a second page.
  - **Next >>:** Button to navigate to the next set of donation categories.

**Design Specifications:**

1. **Top Navigation Bar:**
   - Should be a horizontal bar at the top of the page.
   - Items should be evenly spaced.
   - The logo should be centrally aligned within the navigation bar.
   - The "Donate" section should be highlighted to indicate the current active page.
2. **Donation Options Section:**
   - Four rectangular sections, each representing a different donation category.
   - Each section includes an image, a title, and a "Donate Now" button.
   - Sections should be evenly spaced and have rounded corners.
   - Buttons should have a distinct border and background.
3. **Pagination:**
   - Pagination controls should be located at the bottom of the donation options section.
   - Highlight the current page number.
   - Include "Next >>" to navigate to the next page of options.

**Special User Interface Requirements:**

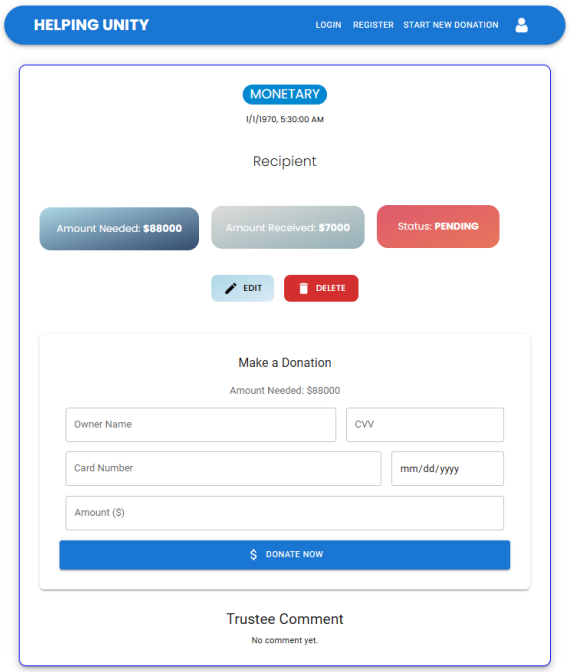
- **Responsiveness:** The interface should be responsive and adapt to different screen sizes, ensuring readability and accessibility on mobile devices.
- **Accessibility:** Ensure the interface is accessible to users with disabilities, including keyboard navigation, screen reader compatibility, and high-contrast color schemes.
- **Consistency:** Maintain consistent styling across all elements, including fonts, colors, and button styles.
- **Interactivity:**
  - Buttons should have hover effects to indicate they are clickable.
  - Navigation items should provide visual feedback when selected.

- Images should be clear and relevant to the donation category.

**Additional Considerations:**

- **Logo Integration:** The logo should be integrated in a way that does not interfere with navigation items but is still prominent.
- **Pagination Functionality:** Ensure pagination is functional and provides a seamless transition between pages.

- **Payment**



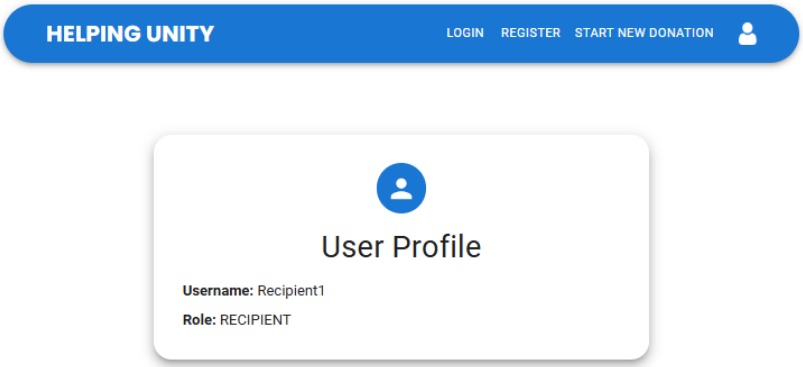

The UI screen appears to be a donation page. The donation page consists of the following elements:

- **Header:** The header contains the charity's name and navigation links.
- **Donation amount:** A field where users can enter their donation amount. There is a dropdown menu to select the currency.
- **Payment methods:** The page shows logos for two payment methods, VISA and PayPal. It's likely that there is also a radio button selection for each payment method.
- **Donate button:** A button that allows users to submit their donation.

Here are some special UI requirements:

- **Mobile-friendly:** The donation page should be responsive and adjusted to different screen sizes, including mobile devices.
- **Security:** The donation page should use secure encryption to protect user's financial information.
- **Clear calls to action:** The donation button should be clear and easy to find.

- **About**




**Top Navigation Bar**

- **Items:**
  - **Donate**
  - **Fundraise**
  - **Payment**
  - **Search**
  - **About** (highlighted as the current active section)
- **Center Logo:** A logo with the text "Helping Unity" underneath.

**User Profile Section**

- **Profile Icon:**
  - A placeholder image/icon representing the user's profile picture.
- **User Details:**
  - **Username:** Placeholder text "Username"

    - **Email:** Placeholder text "Email"
    - **Description:** Placeholder text "Description about you"
- **Update Button:**
    - Located towards the bottom right of the user details section.
    - A button labeled "Update" for updating the user's profile information.

**Design Specifications:**

1. **Top Navigation Bar:**
    - Should be a horizontal bar at the top of the page.
    - Items should be evenly spaced.
    - The logo should be centrally aligned within the navigation bar.
    - The "About" section should be highlighted to indicate the current active page.
2. **User Profile Section:**
    - The profile icon should be circular and large enough to be noticeable.
    - User details should be aligned to the right of the profile icon.
    - The "Update" button should be distinctly styled to be easily recognizable as clickable.

**Special User Interface Requirements:**

- **Responsiveness:** The interface should be responsive and adapt to different screen sizes, ensuring readability and accessibility on mobile devices.
- **Accessibility:** Ensure the interface is accessible to users with disabilities, including keyboard navigation, screen reader compatibility, and high-contrast color schemes.
- **Consistency:** Maintain consistent styling across all elements, including fonts, colors, and button styles.
- **Interactivity:**
    - Buttons should have hover effects to indicate they are clickable.
    - Navigation items should provide visual feedback when selected.
- **Profile Picture:**
    - Consider allowing users to upload or change their profile picture.

#### 4.1.3.2 Hardware Interfaces

The hardware interfaces required for the web application designed to facilitate donations and help provision. These interfaces define the necessary hardware components and their interactions to ensure smooth operation and optimal performance of the application.

**Web Server**

- Processor: Minimum Quad-core CPU (e.g., Intel Xeon E3-1220 v6 or equivalent)
- Memory: Minimum 16 GB RAM
- Storage: SSD with at least 500 GB capacity
- Network: Gigabit Ethernet

**Database Server**

- Processor: Minimum Quad-core CPU (e.g., Intel Xeon E5-2620 v4 or equivalent)
- Memory: Minimum 32 GB RAM
- Storage: SSD with at least 1 TB capacity, with RAID 10 for redundancy
- Network: Gigabit Ethernet

**Backup Storage**

Requirement: The web application must have a reliable backup storage solution.

- NAS (Network-Attached Storage) with at least 2 TB capacity and support for automated scheduled backups
- Network: Gigabit Ethernet

**Security Appliances**

- Firewall: Hardware firewall capable of handling at least 10,000 concurrent connections (e.g., Cisco ASA 5500-X Series)
- Intrusion Detection System (IDS) / Intrusion Prevention System (IPS): Capable of real-time traffic monitoring and threat prevention

**User Devices**

**Specifications:**

- **Desktop Computers**:
  - Minimum requirements include a dual-core CPU, 4 GB RAM, and a modern web browser (Chrome, Firefox, Edge, Safari).
- **Mobile Devices**:
  - Compatibility with iOS 13+ and Android 8+ operating systems.
- **Tablets**:
  - Compatibility with popular tablet devices (e.g., iPad, Samsung Galaxy Tab).

#### 4.1.3.3 Software Interfaces

**Payment Gateway Integration**

- **Interface**:
  - API: RESTful API provided by payment gateways such as Stripe, PayPal, etc.
  - Authentication: OAuth 2.0 for secure API access.
  - Endpoints:
    - `/v1/charges` for processing payments.
    - `/v1/customers` for managing customer information.
  - Data Format: JSON

**Email Service Integration**

- **Interface**:
  - API: RESTful API provided by email services like SendGrid or Mailgun.
  - Authentication: API key for secure access.
  - Endpoints:
    - `/v3/mail/send` for sending emails.
  - Data Format: JSON

**Authentication Service**

- **Interface**:

- API: RESTful API provided by authentication services like Auth0 or Firebase Authentication.
- Authentication: OAuth 2.0/OpenID Connect for secure authentication and authorization.
- Endpoints:
  - `/oauth/token` for obtaining access tokens.
  - `/userinfo` for retrieving user profile information.
- Data Format: JSON

**Social Media Integration**

- **Interface**:
  - API: RESTful APIs provided by social media platforms (e.g., Facebook Graph API, Twitter API).
  - Authentication: OAuth 2.0 for secure API access.
  - Endpoints:
    - `/me/feed` for posting to the user's timeline (Facebook).
    - `/statuses/update.json` for posting a tweet (Twitter).
  - Data Format: JSON

**Analytics Service**

- **Interface**:
  - API: JavaScript API provided by analytics services (e.g., Google Analytics, Mixpanel).
  - Authentication: API key for secure access.
  - Endpoints:
    - `trackEvent` for logging custom events.
    - `identifyUser` for tracking user interactions.
  - Data Format: JSON

**Geolocation Service**

- **Interface**:
  - API: RESTful API provided by geolocation services (e.g., Google Maps API, Mapbox).
  - Authentication: API key for secure access.
  - Endpoints:
    - `/geocode` for converting addresses to geographic coordinates.

        - `/reverseGeocode` for converting geographic coordinates to addresses.
    - Data Format: JSON

#### 4.1.3.4 Communications Interfaces

**Local Area Network (LAN)**

- **Protocol**: TCP/IP
- **Ports**:
    - HTTP: 80
    - HTTPS: 443
    - Database: Specific to the database being used (e.g., MySQL: 3306, PostgreSQL: 5432)
- **Data Format**: JSON, XML

**Wide Area Network (WAN)**

- **Protocol**: TCP/IP
- **Ports**:
    - HTTPS: 443 (for secure communications with external services)
- **Data Format**: JSON

**Remote API Communication**

- **Protocol**: HTTPS
- **Ports**: 443
- **Data Format**: JSON
- **Security**: OAuth 2.0 for secure API access

**WebSocket Communication**

- **Protocol**: WebSocket (ws) or Secure WebSocket (wss)
- **Ports**:
    - WebSocket: 80
    - Secure WebSocket: 443
- **Data Format**: JSON

**Email Communication**

- **Protocol**: SMTP for sending emails, IMAP/POP3 for receiving emails

- **Ports**:
  - SMTP: 587 (submission), 465 (secure SMTP)
  - IMAP: 143 (standard), 993 (secure IMAP)
  - POP3: 110 (standard), 995 (secure POP3)
- **Data Format**: MIME (Multipurpose Internet Mail Extensions)

### 4.1.4 Security and Authentication requirements

#### 4.1.4.1 Data Storage Security

- **Database Security**:
  - The database must be secured against unauthorized access and potential threats.
  - All sensitive data must be encrypted both at rest and in transit.
  - Set up secure backup and recovery mechanisms to ensure data availability and integrity.
  - Ensure data integrity to prevent unauthorized alterations.
  - Comply with relevant data protection regulations and standards.

#### 4.1.4.2 Data Communication Security

- **Secure Protocols**:
  - All data communication must use secure protocols such as Transport Layer Security (TLS), HTTPS, and Secure WebSocket's to protect data in transit.
  - Data must be encrypted during transmission to ensure confidentiality and integrity.
  - Use secure methods for data transmission, such as HTTP POST methods.
  - Validate all incoming and outgoing data to prevent data leakage and unauthorized access.
  - Ensure secure authentication and authorization for data communication.
  - Establish secure communication with external APIs and services using API Keys and Rate Limiting.

### 4.1.5 Quality Assurance Requirements

#### 4.1.5.1 QA Test Scope

**Integration Testing**

Integration testing focuses on verifying the interaction between integrated units or components to ensure they work together as expected. This includes testing interactions between the web application and external services, such as payment gateways, email services, and authentication providers.

- **Tools**: Postman (for API testing) and manual testing.

**System Testing**

System testing involves testing the complete and integrated application to verify that it meets the specified requirements. This includes:

- Functional testing: Ensuring that all features and functionalities work correctly.
- Usability testing: Evaluating the user-friendliness and overall user experience.
- Performance testing: Assessing the application's responsiveness and resource usage.
- **Tools**: Selenium (for automated testing) and manual testing.

**Regression Testing**

Regression testing involves re-running previously completed tests after changes have been made to the application. The goal is to ensure that the changes have not introduced new defects. This includes re-running unit tests, integration tests, and system tests.

- **Tools**: Selenium (for automated regression testing).

**User Acceptance Testing (UAT)**

UAT involves testing the application with end-users to verify that it meets their needs and expectations. This includes assessing the application's functionality, usability, and overall user experience from the perspective of the end-user.

- **Tools**: User feedback forms, surveys, and manual testing.

### Performance Testing

Performance testing evaluates the application's performance under various conditions, such as load testing, stress testing, and scalability testing. It includes measuring response times, throughput, and resource usage.

- **Tools**: JMeter (for load testing).

### Security Testing

Security testing identifies and mitigates potential security vulnerabilities in the application. Common vulnerabilities tested include SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).

- **Tools**: OWASP ZAP (for security testing).

#### 4.1.5.2 QA Environment

This section describes the software, hardware, tools, and environmental requirements for the quality assurance (QA) process. Establishing a well-defined QA environment ensures consistent, reliable, and effective testing of the web application.

### Software Requirements

### Operating Systems

- Windows: Version 10 or higher
- MacOS: Version 10.14 (Mojave) or higher
- Linux: Ubuntu 20.04 LTS

### Browsers

- Google Chrome: Latest stable version
- Mozilla Firefox: Latest stable version
- Microsoft Edge: Latest stable version
- Safari: Latest stable version

### Development Tools

- Integrated Development Environment (IDE): Visual Studio Code and IntelliJ IDEA
- Version Control System: Git, with repositories hosted on GitHub

### Testing Tools

- Unit Testing: JUnit
- Integration Testing: Postman
- System Testing: Selenium
- Regression Testing: Selenium
- Performance Testing: JMeter
- Security Testing: OWASP ZAP
- Compatibility Testing: BrowserStack
- CI/CD Tools: Jenkins or GitLab CI
- Databases: MySQL or Firebase

### Hardware Requirements

### QA Servers

- Processor: Intel Xeon processor
- Memory: Minimum 16 GB RAM
- Storage: SSDs with at least 500 GB capacity
- Network: High-speed Ethernet connection

### Test Devices

- Desktops/Laptops: Various configurations to cover different user environments
- Mobile Devices: Range of smartphones and tablets running iOS and Android, covering various versions and screen sizes

### Network Configuration

- Test Network: Isolated network environment to simulate different network conditions and ensure security.
- VPN: For secure access to internal test environments

### Environmental Requirements

### Test Environments

- Development Environment: Mirrors the development setup for initial testing and debugging
- Staging Environment: A close replica of the production environment for final testing before deployment

- Production Environment: The live environment where the application is deployed for end-users

### Data Management

- Test Data: Use of anonymized production data or synthetic data for testing purposes
- Database Snapshots: Regular snapshots of the test database to restore the environment to a known state

### Security Measures

- Access Control: Strict access control policies to ensure that only authorized personnel can access the QA environments
- Encryption: Encryption of data at rest and in transit within the QA environments

#### 4.1.5.3 QA Data

To begin testing, we need to consider the following key data entities:

1. **Outside Data Sources**:
    - Identify any external data sources that the application interacts with (e.g., APIs, third-party services, legacy systems).
    - Ensure that test data from these sources is available for integration testing.
2. **Populated Databases**:
    - Set up test databases with representative data.
    - Populate the databases with sample records to cover various scenarios (e.g., user profiles, transactions, product data).
3. **Data Generation Tools**:
    - Utilize tools or scripts to generate synthetic data for load testing, stress testing, and performance testing.
    - Consider tools that can create realistic data distributions and simulate user behavior.

### 4.1.6 Development Requirements

#### 4.1.6.1 Development Environment

For development, we require the following:

- **Software**:
    - Operating Systems: Windows 10 or higher or MacOS 10.14 (Mojave) or higher or Ubuntu 20.04 LTS (Linux).
    - Browsers: Google Chrome, Mozilla Firefox, Microsoft Edge, Safari (latest stable versions).
    - IDEs: Visual Studio Code or IntelliJ IDEA.
    - Version Control: Git with repositories hosted on GitHub.
    - Databases: MySQL or Firebase.

- **Hardware**:
    - Development servers with Intel Xeon processors, at least 16 GB RAM, and SSD storage.
    - Test devices (desktops/laptops and mobile devices) to cover various user environments.

- **Network Configuration**:
    - Isolated test network environment for security and network testing.
    - VPN for secure access to internal test environments.

#### 4.1.6.2 Development Data

Key data entities required for development include:

1. **Outside Data Sources**:

- Understand the data structures and formats from external APIs or services.
- Use mock data or sandbox environments during development.

2. **Populated Databases**:

- Develop against a populated database (either real or synthetic data) to ensure accurate functionality.
- Consider using database snapshots for restoring to a known state during development.

#### 4.1.6.3 Coding Standards

Coding standards and naming conventions are essential for consistency and maintainability. Refer to any existing coding standards documentation or establish project-specific guidelines. Some common practices include:

- Consistent indentation and formatting.
- Descriptive variable, function, class, and package names.
- Proper commenting and documentation.

#### 4.1.6.4 Implementation Packaging Requirements

Consider any special packaging requirements for application modules. For instance:

- In Java, adhere to package naming conventions
  - (e.g., com.bridginggaps.helpinghands.module).
- Group related functionality into appropriate packages for clarity and organization.

### 4.1.7 Deployment Requirements

#### 4.1.7.1 Installation Packaging Requirements

Consider any special packaging requirements for application modules. For instance:

- In Java, adhere to package naming conventions
  - (e.g., com.bridginggaps.helpinghands.module).
- Group related functionality into appropriate packages for clarity and organization.

#### 4.1.7.2 Deployment Requirements

Installation Packaging Requirements

This subsection describes the packaging requirements for installing the web application, including the necessary files, scripts, and documentation.

**Requirement**: The installation package must have a well-defined structure to ensure all necessary components are included and organized.

**Components**:

- Application Code: All source code files and compiled binaries.
- Configuration Files: Environment-specific configuration files (e.g., config.json, .env).
- Database Scripts: SQL scripts for database schema creation and data migration.
- Static Assets: Images, CSS files, JavaScript files, and other static assets.
- Documentation: User manuals, installation guides, and API documentation.

- Scripts: Deployment scripts, build scripts, and initialization scripts.

### Deployment Overview

### Server Locations

Outline the various locations where servers will be deployed. Consider factors such as geographical distribution, redundancy, and scalability. Common server locations include:

**Cloud Services**:

1. Deploying on cloud platform Azure or Google Cloud.

### 4.1.7.3 Documentation requirements

### User Access

Describe how different types of users will access the system:

1. **End-Users**:
    - Access via web browsers or mobile apps.
    - Supported browsers Chrome, Firefox, Edge, Safari and OS versions (e.g., Windows, macOS, iOS, Android).
2. **Administrators**:
    - Access to management interfaces
    - Specify any additional security measures (e.g., VPN access).

### Version Requirements

Consider whether multiple versions of the application will run on the same machine. If so, address any version compatibility requirements.

### 4.1.8 Special Documentation Requirements

### *Legal Disclaimers*

### Disclaimer of Liability

The information provided by this software/application is for general informational purposes only. While we strive to keep the information up to date and accurate, we make

no representations or warranties of any kind, express or implied, about the completeness, accuracy, reliability, suitability, or availability with respect to the software or the information contained within it. Any reliance you place on such information is strictly at your own risk.

### No Professional Advice

This software does not provide professional advice. If you require specific advice related to legal, financial, medical, or any other professional field, you should consult a qualified professional.

### Use at Your Own Risk

By using this software, you agree that we shall not be liable for any direct, indirect, incidental, consequential, or exemplary damages resulting from your use of the software or any information provided within it.

### *Warranties*

Limited Warranty

We provide a warranty for three months for the customer to use and test the system and come up with any errors or omissions. Thereafter, we will provide support for the application for a nominated fee and support to include any additional features required by the customer case by case with agreed fee for each assignment.

### *Copyright Notices*



### *Patent Notices*

No specific patents apply to this software.

### *Trademark Compliance*

The names and logos used in this software are trademarks or registered trademarks of their respective owners. Their use does not imply any affiliation with or endorsement by the trademark owners.

### *Logo Usage Guidelines*

### Helping Unity Logo

### *Logo Usage Guidelines for Helping Hands*

1. **Logo Overview**:
    - Our logo features a circular emblem with vibrant colors.
    - It depicts two hands in orange and blue cradling a red heart with three green leaves on each side.
    - The background consists of stylized sun rays in yellow and orange hues.

2. **Color Usage**:
    - Use the exact colors as shown in the logo:
        - Orange (#FFA500)
        - Blue (#0000FF)
        - Red (#FF0000)
        - Green (#008000)
        - Yellow (#FFFF00)
    - Maintain color consistency across all applications.

3. **Clear Space**:
    - Ensure sufficient clear space around the logo to maintain visibility.
    - No other elements (text, graphics) should encroach upon this space.

4. **Minimum Size**:
    - Maintain legibility by using the logo at a minimum size of 1 inch (2.54 cm) in diameter.

5. **Background Usage**:
    - Use the logo against light backgrounds to ensure contrast.
    - Avoid placing it on busy or cluttered backgrounds.

6. **Prohibited Modifications**:
    - Do not alter the logo's proportions or distort its shape.
    - Do not add effects (shadows, gradients) that deviate from the original design.
7. **Placement**:
    - Place the logo prominently on marketing materials, websites, and official documents.
    - Use it consistently across all communication channels.

8. **Logo Variations**:
    - We have a full-color version for most applications.
    - For monochromatic or grayscale contexts, use a black-and-white version.

9. **Contact for Approval**:
    - Any deviations from these guidelines require approval from our branding team.
    - Contact [Brand Manager Name] at [Email Address] for logo usage inquiries.

### 4.1.9 Applicable Standards

We are adhering to the following standerds.

- Legal and regulatory standards (e.g., FDA, UCC)
- Communications standards (e.g., TCP/IP, ISDN)
- Platform compliance standards (e.g., Windows, UNIX)
- Quality and safety standards (e.g., UL, ISO, CMM)

### 4.1.10 On-line User Documentation and Help System Requirements

### Online User Documentation

1. **Comprehensive Guides:**
    - Detailed step-by-step guides covering all functionalities of the platform, including donation processes, fundraising setup, payment history, and account management.
2. **Search Functionality:**
    - A powerful search engine that allows users to quickly find relevant information by typing keywords or phrases.
3. **FAQs Section:**
    - A frequently asked questions (FAQs) section addressing common user inquiries and issues.
4. **Glossary:**
    - A glossary of terms used within the platform to help users understand specific terminology.
5. **Multimedia Support:**
    - Integration of images, diagrams, and videos within the documentation to provide visual aids and tutorials.
6. **User Feedback:**
    - A feedback system allowing users to rate the helpfulness of documentation and suggest improvements.

## Help System

1. **Context-Sensitive Help:**
    - Contextual help icons or links available throughout the platform, providing users with relevant information based on the current screen or task.
2. **Interactive Tutorials:**
    - Interactive tutorials guiding users through key tasks and features of the platform, especially during their initial use.
3. **Live Chat Support:**
    - Integration of a live chat system enabling users to get real-time assistance from support staff.
4. **Help About Notices:**
    - Help About notices providing information about the platform, version number, contact information for support, and links to the user documentation.
5. **Troubleshooting Guide:**
    - A detailed troubleshooting guide to help users resolve common issues they may encounter while using the platform.

### Accessibility

1. **Language Support:**
    - Online documentation and help content available in multiple languages to cater to a diverse user base.
2. **Screen Reader Compatibility:**
    - Ensuring that all documentation and help systems are compatible with screen readers to support visually impaired users.
3. **Responsive Design:**
    - All help and documentation pages should be responsive and accessible on various devices, including desktops, tablets, and smartphones.

## 4.1.11 Usability Requirements

### User Interface Design

1. **Consistency:**
    - Maintain a consistent look and feel throughout the platform, using standard colors, fonts, and design elements.
2. **Intuitive Navigation:**
    - Ensure that the platform's navigation is intuitive, allowing users to find information and complete tasks with minimal effort.
3. **Responsive Design:**
    - Implement a responsive design that adapts to different screen sizes and devices, providing a consistent experience across desktops, tablets, and smartphones.
4. **Widget Sets:**
    - Utilize modern, accessible widget sets for forms, buttons, and interactive elements to enhance usability and accessibility.

### Usability Testing

1. **Browser Compatibility:**
    - Test the platform on all major browsers, including Chrome, Firefox, Safari, Edge, and Opera, to ensure compatibility and performance.
2. **Cross-Platform Testing:**

- Conduct testing on different operating systems (Windows, macOS, Linux) and devices (iOS, Android) to ensure consistent functionality and appearance.

3. **User Feedback:**
   - Incorporate user feedback into the development process through usability testing sessions, surveys, and feedback forms.
4. **Accessibility Testing:**
   - Perform accessibility testing to ensure compliance with WCAG (Web Content Accessibility Guidelines) standards, making the platform usable for individuals with disabilities.

### Portability Requirements

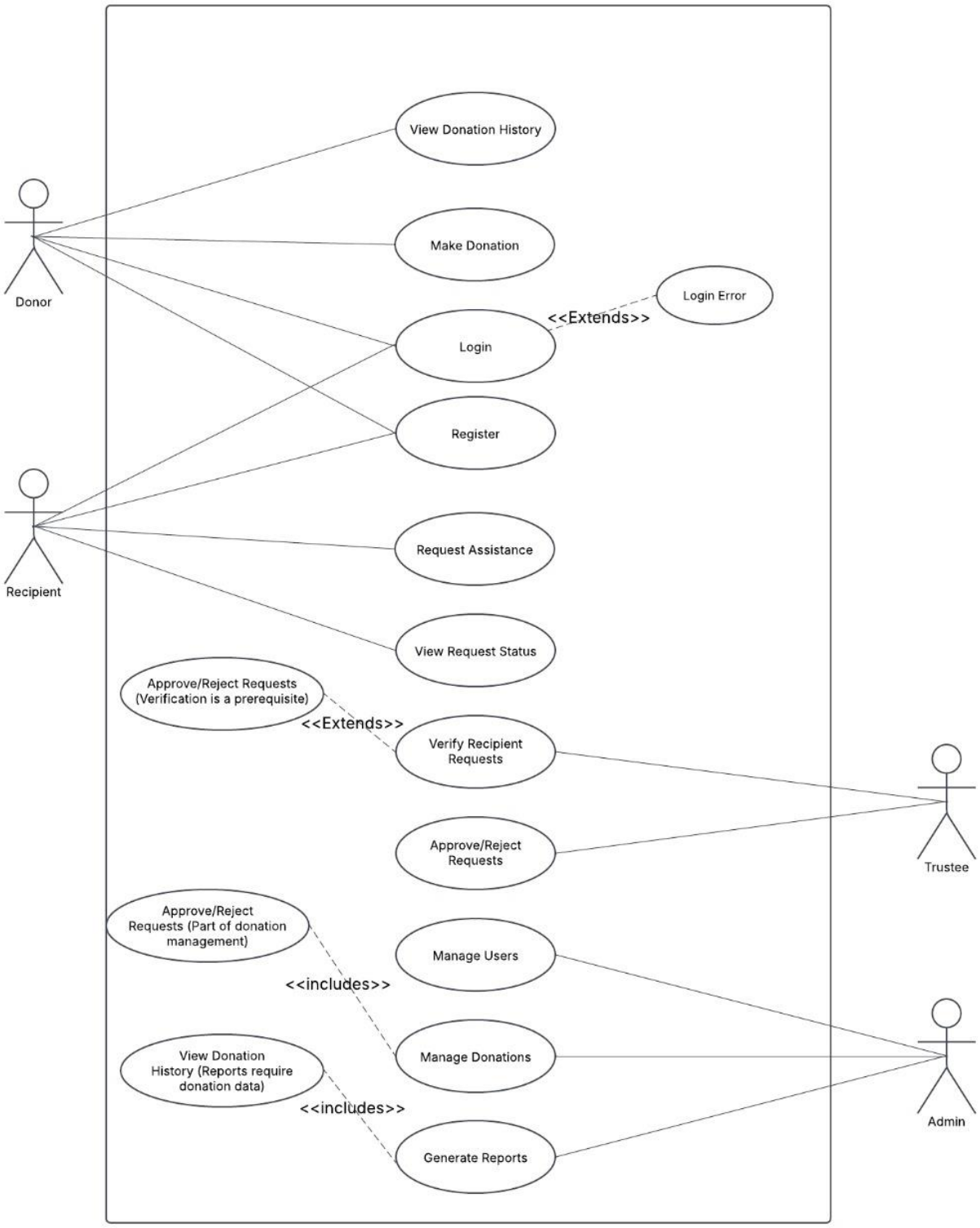
1. **Browser Portability:**
   - Ensure that the user interface is portable across various web browsers without requiring additional plugins or extensions.
2. **Device Portability:**
   - Optimize the platform for both desktop and mobile devices, ensuring that all features are accessible and functional regardless of the device used.
3. **Scalability:**
   - Design the user interface to be scalable, accommodating future expansions and new features without significant redesigns.

# 5 Future Requirements (Optional)

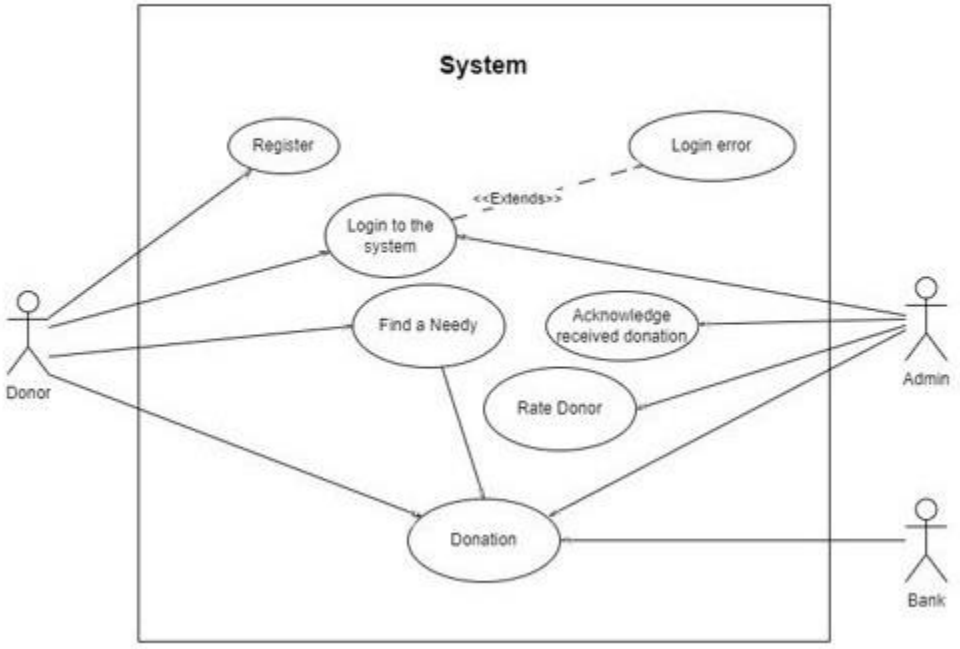
# 6 Appendix

## Use Case Diagrams

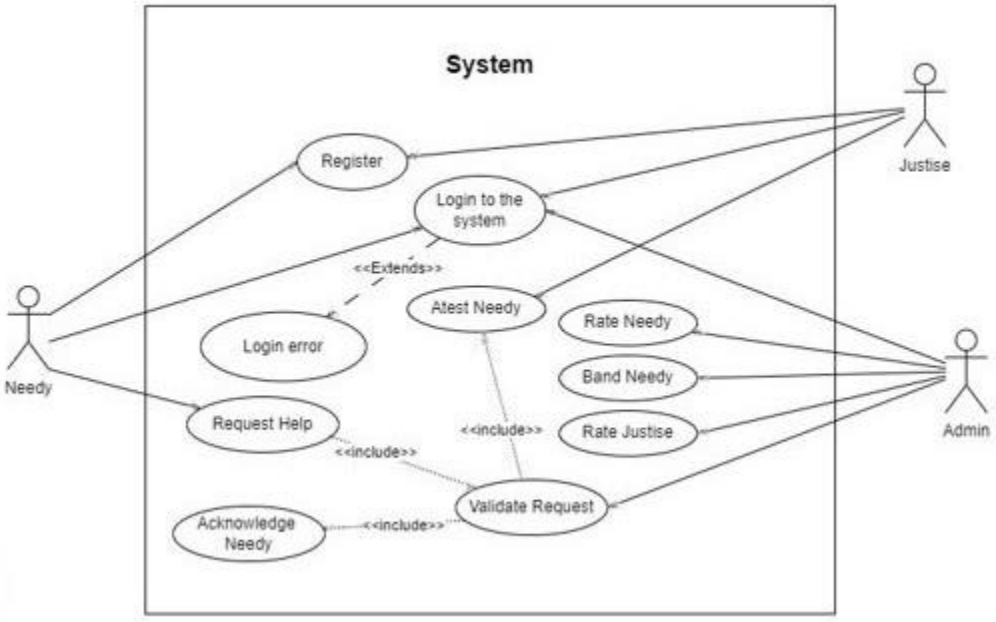
# Main Use Case



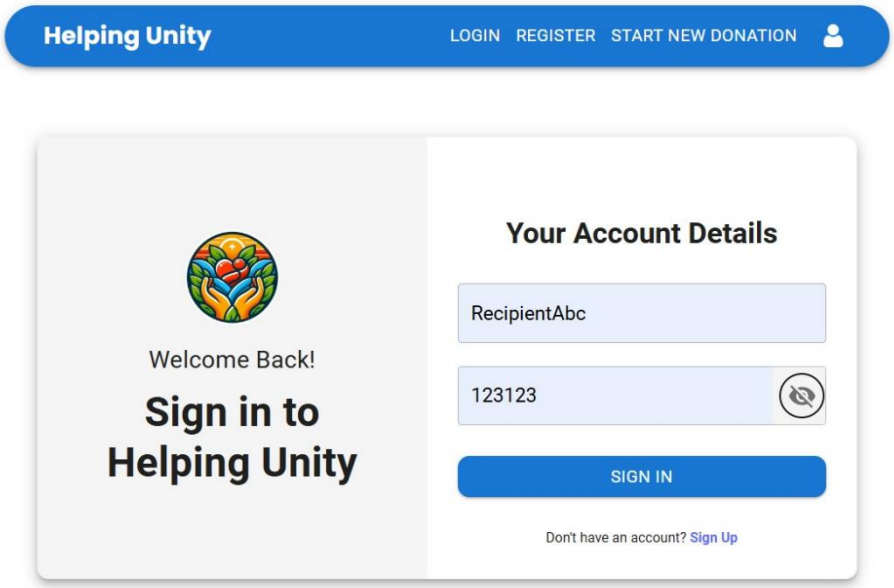
***Use Case of Donating Through the Platform***



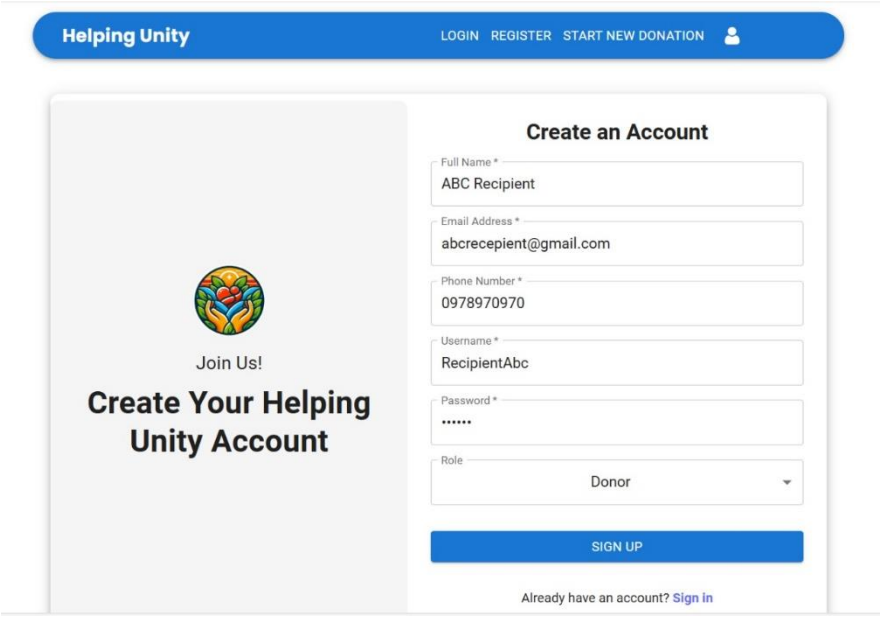

***Use Case of Requesting Needs from the Platform***



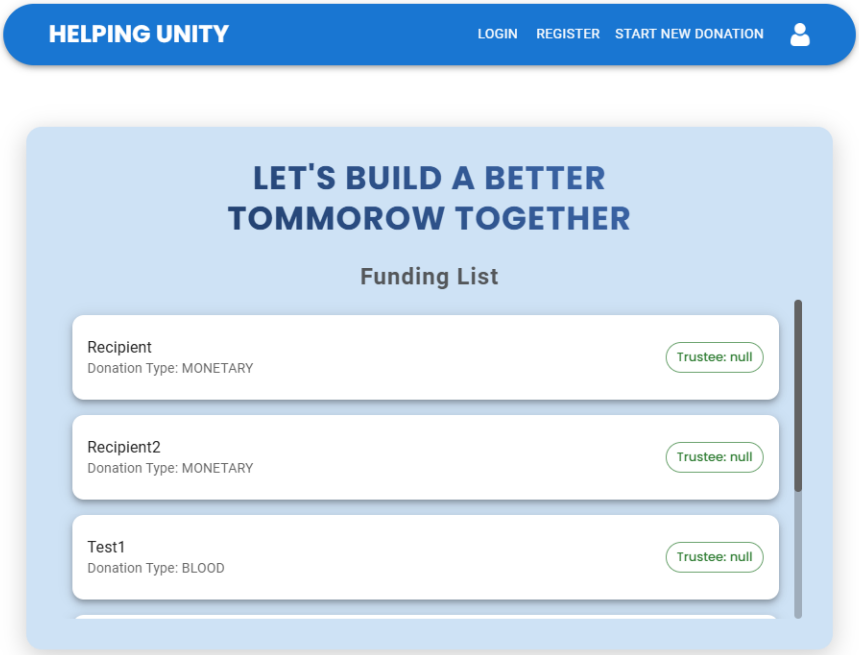
## User Interfaces



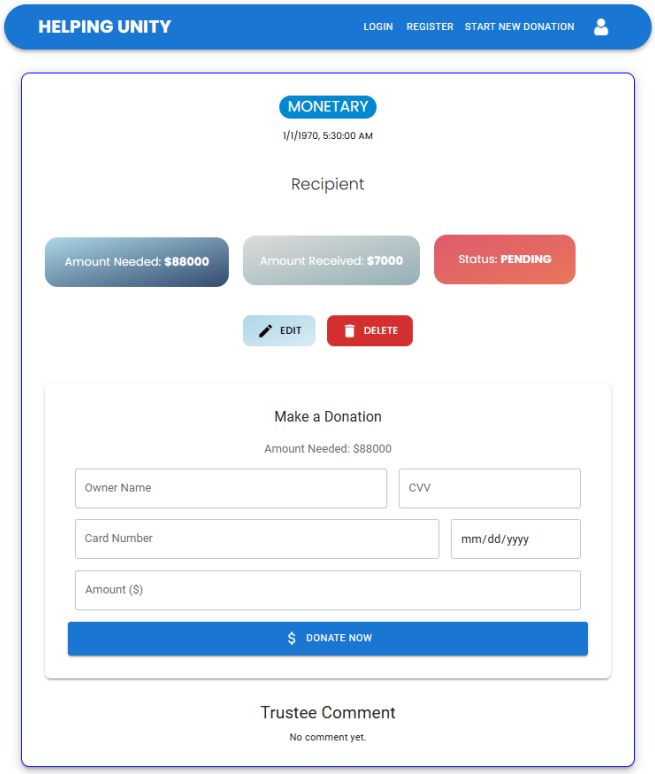

*Sign in Page*



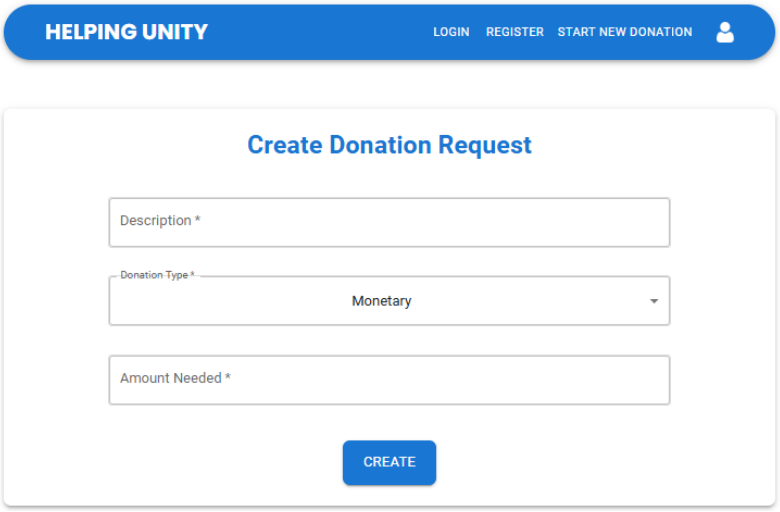

*Sign up Page*



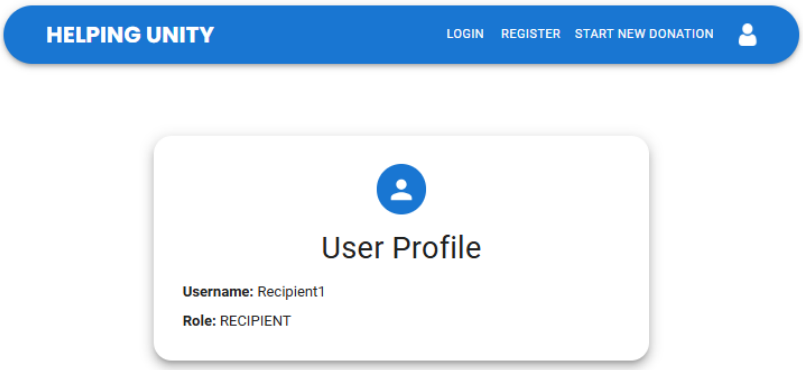

*Funding Page*




*Donate Page*




*Donation Request Page*




*User Profile Page*