**Software Requirement Specifications**

**[Help.com]**

**Version: [1.0]**

|  |  |
| --- | --- |
| Project Code: |  |
| Supervisor: | Ms. Zain Noreen |
| Co Supervisor |  |
|  |  |
| Project Team: | Awesh Kumar (21K-4526)  Danish Nanjiani (21K-3959)  Anil Kumar (20K-0444) |
| Submission Date: | 12/12/2024 |

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Name of Person** | **Date** | **Description of change** |
| 1.0 | Awesh Kumar | November 5, 2022 | Document Created |
| 1.0 | Dainsh Nanjiani | November 7, 2022 | Added External Interface  Requirement |
| 1.0 | Anil | November 12, 2022 | Added Introduction |
| 1.0 | Awesh Kumar | November 15, 2022 | Added Functional Requirements |
| 1.0 | Anil Kumar | November 18, 2022 | Added Use Cases |
| 1.0 | Dainsh Nanjiani | November 21, 2022 | Overall Project Description |
| 1.0 | Awesh Kumar | November 27, 2022 | Added Non Functional Requirements |

Distribution List

[Following table will contain list of people whom the document will be distributed after every sign-off]

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | |
| Ms. Zain Noreen | | Supervisor |
|  | |  |
|  | |  |

Document Sign-Off

|  |  |  |
| --- | --- | --- |
| **Version** | **Sign-off Authority** | **Sign-off Date** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Table of Contents

[1.](#_heading=h.30j0zll) Introduction 5

[1.1.](#_heading=h.3znysh7) Purpose of Document 5

[1.2.](#_heading=h.2et92p0) Intended Audience 5  
1.3 Abbreviations ………………………………………………………………………………………...5

[1.4.](#_heading=h.tyjcwt) Document Convention 5

[2.](#_heading=h.3dy6vkm) Overall System Description 6

[2.1.](#_heading=h.1t3h5sf) Project Background 6

[2.2.](#_heading=h.4d34og8) Project Scope 6

[2.3.](#_heading=h.2s8eyo1) Not in Scope 6

[2.4.](#_heading=h.17dp8vu) Project Objectives 6

[2.5.](#_heading=h.3rdcrjn) Stakeholders 7

[2.6.](#_heading=h.26in1rg) Operating Environment 7

[2.7.](#_heading=h.35nkun2) System Constraints 7

[2.8.](#_heading=h.1ksv4uv) Assumptions & Dependencies 8

[3.](#_heading=h.2jxsxqh) External Interface Requirements 9

[3.1.](#_heading=h.z337ya) Hardware Interface 9

[3.2.](#_heading=h.3j2qqm3) Software Interface 9

[3.3.](#_heading=h.1y810tw) Communications Interfaces 9

[4.](#_heading=h.4i7ojhp) Functional Requirements 10

[4.1.](#_heading=h.2xcytpi) Functional Hierarchy 10

[4.2.](#_heading=h.1ci93xb) Use Cases 11

[4.2.1.](#_heading=h.3whwml4) Admin 11

4.2.2. Donor. 16

4.2.3. Volunteer 20

4.2.4. Needy One 23

[5.](#_heading=h.2bn6wsx) Non-functional Requirements 26

[5.1.](#_heading=h.qsh70q) Performance Requirements 26

[5.2.](#_heading=h.3as4poj) Safety Requirements 26

[5.3.](#_heading=h.1pxezwc) Security Requirements 26

[5.4.](#_heading=h.49x2ik5) user Documentation 27

[6.](#_heading=h.147n2zr) References 28

[7.](#_heading=h.3o7alnk) Appendices 28

**1.Introduction**

**1.1 Purpose of Document**

The purpose of this Software Requirements Specification (SRS) is to serve as a comprehensive guideline for the design, development, and deployment of **Help.com**. This document aims to:

* Define the functional, non-functional, and technical requirements for the platform.
* Provide a unified understanding for developers, stakeholders, and testers.
* Ensure alignment with the project's objectives of creating a robust, user-friendly social welfare platform.

By leveraging this SRS, all involved parties can work collaboratively to ensure the successful implementation of a system that empowers communities through efficient management of donations and volunteer activities.

**1.2 Intended Audience**

* **Stakeholders**: Supervisors, donors, volunteers, organizations.
* **Developers**: Developers working on the MERN stack implementation.
* **Testers**: Quality assurance teams.
* **End Users**: Individuals, volunteers, and organizations.

**1.3 Abbreviations**

* **MERN**: MongoDB, Express.js, React.js, Node.js.
* **API**: Application Programming Interface.
* **UI**: User Interface.

**1.4 Document Convention**

**Font:** Arial

**Font-size:**

Heading: 20px

Subheading: 14px

Description: 12px

**2.Overall System Description**

**2.1 Project Background**

Communities often face challenges in organizing charitable actions like donations and volunteering due to scattered resources and lack of centralized platforms. **Help.com** addresses this gap by providing an innovative social welfare platform to streamline these efforts. Leveraging modern web technologies, it connects donors, recipients, and volunteers efficiently. The platform offers features such as location-based donation matching, automated notifications, and a volunteer management system. With integrated tools like Google Maps API and an AI-powered chatbot, **Help.com** ensures ease of use and accessibility for all users. By simplifying processes and promoting transparency, this platform empowers communities to collaborate for the greater good and achieve impactful results.

**2.2 Project Scope**

**Help.com** aims to transform how communities contribute and receive help by offering a centralized platform for donations, volunteering, and resource management. Volunteers can create detailed profiles, post or request donations, and sign up for volunteer opportunities with ease. Through its advanced features like AI-powered assistance and location-based logistics, the platform ensures a seamless user experience. By bridging the gap between donors, volunteers, and those in need, **Help.com** fosters collaboration and promotes impactful community engagement, making social welfare accessible and efficient for everyone.

**2.3 Not In Scope**

Not applicable

**2.4 Project Objectives**

* Establish a platform that connects individuals in need with donors and volunteers, fostering a community of mutual support.
* Streamline donation logistics to ensure resources are delivered efficiently to those seeking help.
* Provide an intuitive, user-friendly interface powered by AI tools to make the platform accessible to all.
* Highlight the positive impact of community contributions through transparency and success stories.

**2.5 Stakeholders**

* **Individuals**: Donors and recipients.
* **Organizations**: Managing volunteers and campaigns.
* **Administrators**: Overseeing platform operations.
* **Supervisors**: Monitoring project progress.

**2.6 Operating Environment**

* Hardware: Desktop, laptop, and mobile devices.
* Software: Google Chrome, Mozilla Firefox, React.js, Node.js.
* Network: High-speed internet connection.

**2.7 System Constraints**

**Software Constraints**  
The software is built on Node.js (server-side JavaScript) framework with Express.js and uses MongoDB as a database. For deployment, it requires a cloud service to host and ensure accessibility on web browsers and mobile devices.

**Hardware Constraints**  
The application will function on PCs, laptops, and smartphones with a stable internet connection and minimum specifications of 2GB RAM and a 1.5GHz processor.

**Cultural Constraints**  
The application supports the English language, making it accessible to a broad audience familiar with this medium.

**Legal Constraints**  
The software can only be accessed by registered volunteers to comply with privacy and data protection regulations.

**Environmental Constraints**  
The platform is web-based, eliminating the need for software installation and ensuring cross-device compatibility.

**User Constraints**  
The platform is designed for donors, volunteers, and organizations, each having tailored functionalities based on their roles.

**Off-the-Shelf Components**  
The platform does not utilize any off-the-shelf components, ensuring custom-built solutions for all features.

**2.8 Assumptions & Dependencies**

* Availability of Google Maps API and third-party integrations.
* Users have basic digital literacy and access to compatible devices.
* Reliable hosting on scalable cloud infrastructure.

**3.External Interface Requirements**

**3.1 Hardware Interfaces**

* **Devices:** PCs, laptops, and smartphones.
* Minimum Specifications: 2GB RAM, 1.5GHz processor.

**3.2 Software Interfaces**

* **Frontend**: React.js for dynamic UI.
* **Backend**: Node.js and Express.js for server-side operations.
* **Database**: MongoDB for data management.
* **APIs**: Google Maps for location services and Dialogflow for chatbot functionality.

**3.3 Communications Interfaces**

* Secure communication via HTTPS.
* Email and SMS notifications for real-time updates.
* WebSocket protocols for live data transmission.

**4.Functinal Requirements**

**4.1 Functional Hierarchy**

**1. User Management**

* Register User
* Authenticate User
* Manage User Profiles (Volunteer, Admin)

**2. Donation Management**

* Post Donations (Categorize, Item Details)
* View Donations (For Admins, Donors, and Needy)
* Match Donations with Requests
* Schedule Pickup or Drop-off for Donations

**3. Help Request Management**

* Create Help Request (Item/Service Description, Urgency, Location)
* View and Track Help Requests (By Needy Users)
* Match Help Requests with Available Donations

**4. Volunteer Management**

* Post Volunteer Opportunities (Admins/Organizations)
* Search and Sign Up for Volunteer Opportunities (Volunteers)
* Track Volunteer Activities and Attendance

**5. Logistics and Notifications**

* Schedule Pickup for Donations
* Locate Drop-off Points (Integration with Google Maps API)
* Send Real-Time Notifications (Donation/Request Status Updates, Volunteer Reminders)

**6. AI Assistance**

* Provide Navigation and FAQ Support via AI Chatbot
* Redirect Complex Queries to Admins

**4.2 Use Cases**

**4.2.1 Admin**

**A diagram of a company

Description automatically generated**

Table 1: Login Use Case Description

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case Name:** | Login |
| **Use Case Description:** To allow the admin to securely log in to the website and access administrative functionalities. | |
| **Primary Actor:** Admin | **Other Actors:** Database |
| **Relationship:**  **Includes:** Verify |  |
| **Stack Holders:** Admin |  |
| **Flow of Events:**   * The admin navigates to the Help.com login page. * The admin enters their username and password. * The system verifies the credentials against the stored records in the database. * Upon successful verification, the admin is granted access to the admin dashboard. * If the credentials are incorrect, an error message is displayed, and the admin can retry or reset their password. | |
| **Post Conditions:** The admin gains access to the system and can perform administrative tasks.  The login attempt is logged in the system for security and audit purposes. | |

Table 2: Contact Use Case Description

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case Name:** | Contact |
| **Use Case Description:** To allow the admin to initiate and manage contact with donors, volunteers, or needy individuals. | |
| **Primary Actor:** Admin | **Other Actors:** Database |
| **Relationship:**  **Includes:**   * Contact Donor * Contact Volunteer * Contact Needy ones |  |
| **Stack Holders:** Admin |  |
| **Flow of Events:**  The admin logs into the system and navigates to the "Contact" section.  The admin selects the type of user to contact:  **Donor:**   * The admin selects a donor from the list. * The system retrieves the donor's contact details (e.g., email, phone number). * The admin composes and sends a message or notification.   **Volunteer:**   * The admin selects a volunteer from the list. * The system retrieves the volunteer's contact details. * The admin sends a task assignment or general communication.   **Needy Individual:**   * The admin selects a needy individual from the list. * The system retrieves their contact details. * The admin sends updates or offers support based on their request.   The system logs the communication for record-keeping purposes. | |

Table 3: Manage Items Use Case Description

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case Name:** | Manage Items |
| **Use Case Description:** To enable the admin to add, update, view, or delete items available for donation on the platform. | |
| **Primary Actor:** Admin | **Other Actors:** Database |
| **Relationship:**  **Includes:** Verify |  |
| **Stack Holders:** Admin |  |
| **Flow of Events:**  The admin logs into the Help.com system and navigates to the "Manage Items" section.  The admin can perform the following actions:  **Add Items:**   * The admin clicks on "Add Item." * The admin provides details such as item name, quantity, description, and category. * The system verifies and saves the new item in the database.   **Update Items:**   * The admin selects an existing item from the list. * The admin modifies the item details and submits the changes. * The system updates the item record in the database.   **Delete Items:**   * The admin selects an item and clicks "Delete." * The system removes the item from the database after confirmation.   **View Items:**   * The admin views a list of all items along with their details such as name, description, quantity, and status.   The system logs all actions performed by the admin for audit purposes. | |
| **Post Conditions:** The items database is updated based on the actions performed by the admin.  Donors and needy individuals see the updated item list in real time. | |

Table 4: Mange Donor Use Case Description

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case Name:** | Mange Donor |
| **Use Case Description:** To enable the admin to view and manage donor details, including their donation history. | |
| **Primary Actor:** Admin | **Other Actors:** Database |
| **Relationship:**  **Includes:** Verify |  |
| **Stack Holders:** Admin |  |
| **Flow of Events:**  The admin logs into the Help.com system and navigates to the "Manage Donor" section.  The admin selects a donor from the list of registered donors.  The admin can perform the following actions:  **View Donor Details:**   * The system displays donor details such as name, email, phone number, and registration date.   **View Donation History:**   * The admin clicks on "Donation History." * The system fetches and displays a list of all donations made by the donor, including details such as donation type (money/items), date, amount/quantity, and status. | |
| **Post Conditions:** The donor’s records, including donation history, are updated and maintained accurately in the system. | |

Table 5: Manage Needy Ones Use Case Description

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case Name:** | Manage Needy Ones |
| **Use Case Description:** To enable the admin to view and manage details of needy individuals registered on the platform. | |
| **Primary Actor:** Admin | **Other Actors:** Database |
| **Relationship:**  **Includes:** Verify |  |
| **Stack Holders:** Admin |  |
| **Flow of Events:**  The admin click on the Needy ones Menu.  **View Needy Individual Details:**   * The system displays details such as name, contact information, address, and registration date. * See the Urgent Need of any Needy and assign the Volunteer for it.   The system logs all actions performed by the admin for accountability and audit purposes. | |

Table 6: Mange Volunteer Use Case Description

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case Name:** | Manage Volunteer |
| **Use Case Description:** To enable the admin to view, update, and manage details of volunteers registered on the platform. | |
| **Primary Actor:** Admin | **Other Actors:** Database |
| **Relationship:**  **Includes:** Verify |  |
| **Stack Holders:** Admin |  |
| **Flow of Events:**  The admin logs into the Website and navigates to the "Manage Volunteer" section.  The admin selects a volunteer from the list of registered users.  The admin can perform the following actions:  **View Volunteer Details:**   * The system displays details such as name, email, phone number, assigned tasks, and registration date.   **Update Volunteer Information:**   * The admin clicks "Edit Details." * The admin updates information such as contact details or status (active/inactive). * The system saves the updated information in the database.   **Assign Tasks to Volunteer:**   * The admin selects "Assign Task." * The admin provides task details (e.g., description, deadline). * The system assigns the task to the volunteer and notifies them.   **Remove Volunteer:**   * The admin selects "Remove Volunteer." * The system prompts for confirmation and, upon approval, removes the volunteer’s record from the database.   The system logs all actions performed by the admin for record-keeping and audit purposes. | |
| **Post Conditions:** Volunteer details, roles, and tasks are updated in the system.  Removed volunteers are no longer part of the platform. | |

**4.2.2 Donar**

A person standing next to a diagram

Description automatically generated

Table 1: Form Submission Use Case Description

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case Name:** | Form Submission |
| **Use Case Description:** To add items for Donation | |
| **Primary Actor:** Donar | **Other Actors:** Database |
| **Relationship:**  **Includes:** Verify |  |
| **Stack Holders:** Donar |  |
| **Flow of Events:**   * The donor logs into the website and navigates to the donation section. * The donor clicks on the "Add Donation" button. * The donor fills out the donation form, including details such as item name, quantity, and any additional description. * The system assigns a unique ID to the donation item automatically. * The donation details are submitted and verified by the database. | |
| **Post Conditions:** The donor can view the details of their donation and track its status on the website. | |

Table 2: View Donation History Use Case Description

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case Name:** | View Donation History |
| **Use Case Description:** To allow donors to view their past donation activities and statuses on the | |
| **Primary Actor:** Donar | **Other Actors:** Database |
| **Relationship:**  **Includes:** Verify |  |
| **Stack Holders:** Donar |  |
| **Flow of Events:**   * The donor logs into their account. * The donor navigates to the "Donation History" section. * The system fetches donation records from the database linked to the donor's account. * The donor views a list of all their previous donations, including details such as item name, date, quantity, and status. | |
| **Post Conditions:** The donor can access a detailed history of their donations and track the impact of their contributions. | |

Table 3: View Items Use Case Description

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case Name:** | View Items |
| **Use Case Description:** To View Items for Donations | |
| **Primary Actor:** Donar | **Other Actors:** Database |
| **Relationship:**  **Includes:** Verify |  |
| **Stack Holders:** Donar |  |
| **Flow of Events:**   * The donor enter into the website. * The donor navigates to the Help Center section * The donor click on View Items Button | |

Table 4: Donate Use Case Description

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case Name:** | Donate |
| **Use Case Description:** To verify whether a user is registered as a volunteer. | |
| **Primary Actor:** Donar | **Other Actors:** Database |
| **Relationship:**  **Includes:** Verify |  |
| **Stack Holders:** Donar |  |
| **Flow of Events:**  The donor navigates to the "Donate" section.  The donor selects one of the following options:  **Donate Money:**   * The donor enters the desired amount. * The system redirects the donor to a secure payment gateway. * The payment is processed and confirmed.   **Donate Items:**   * The donor clicks "Add Items." * The donor fills out item details (e.g., name, quantity, description). * The system verifies and saves the item details. | |
| **Post Conditions:** The donor can view and track their donations (money and items) in the donation history.  The donation is recorded and updated in the database. | |

Table 5: Volunteer Status Verification Use Case Description

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case Name:** | Volunteer Status Verification |
| **Use Case Description:** To allow donors to view their past donation activities and statuses on the | |
| **Primary Actor:** Donar | **Other Actors:** Database |
| **Relationship:**  **Includes:** Verify |  |
| **Stack Holders:** Donar |  |
| **Flow of Events:**   * The Donor navigates to the "Volunteer Management" section. * The Donor searches for a specific user by entering their details (e.g., name, email, or ID). * The system fetches volunteer records from the database. * The system displays whether the user is a registered volunteer, along with additional details such as their status (active/inactive) and assigned tasks (if any). | |
| **Post Conditions:** The admin can confirm the user's volunteer status and manage their roles accordingly. | |

**4.2.3 Volunteer**

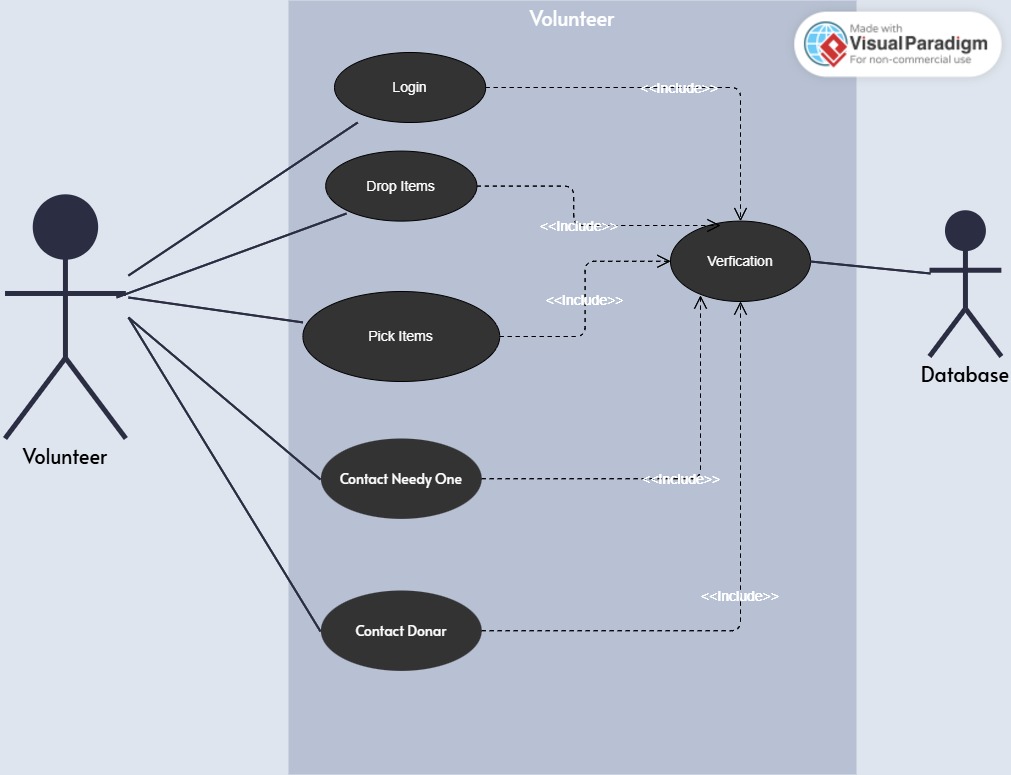


Table 1: Login Use Case Description

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case Name:** | Login |
| **Use Case Description:** To allow volunteers to securely log in to the website and access their assigned tasks and communication tools. | |
| **Primary Actor:** Volunteer | **Other Actors:** Database |
| **Relationship:**  **Includes:** Verify |  |
| **Stack Holders:** Volunteer |  |
| **Flow of Events:**   * The volunteer navigates to the Help.com login page. * The volunteer enters their username and password. * The system verifies the credentials against the stored records in the database. * If the credentials are incorrect, an error message is displayed, and the volunteer can retry or reset their password. | |
| **Post Conditions:** The volunteer gains access to the system and can view their assigned tasks and communication tools.  The login attempt is logged in the system for security and audit purposes. | |

Table 2: Drop Items Use Case Description

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case Name:** | Drop Items |
| **Use Case Description:** To allow volunteers to drop off physical items at designated locations for distribution to needy individuals. | |
| **Primary Actor:** Volunteer | **Other Actors:** Database |
| **Relationship:**  **Includes:** Verify |  |
| **Stack Holders:** Volunteer |  |
| **Flow of Events:**  The volunteer logs into the system and navigates to the "Drop Items" section.  The user selects the items they intend to drop off from their donation history.  The system provides a list of designated drop-off locations.  The user selects a drop-off location and confirms the drop-off details.  Upon arrival at the drop-off location, the items are verified by the admin or an automated system (e.g., QR code scanning).  The system updates the status of the items to "Dropped Off" and logs the transaction in the database.  A confirmation notification is sent to the user. | |
| **Post Conditions:** The system records the successful drop-off of items.  The items are now available for further processing and distribution to needy individuals. | |

Table 3: Pick Items Use Case Description

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case Name:** | Pick Items |
| **Use Case Description:** To allow volunteers to pick up items from designated locations for delivery to needy individuals. | |
| **Primary Actor:** Volunteer | **Other Actors:** Database |
| **Relationship:**  **Includes:** Verify |  |
| **Stack Holders:** Volunteer |  |
| **Flow of Events:**   * The volunteer logs into the system and navigates to the "Pick Items" section. * The system displays a list of items available for pickup, along with their designated pickup locations. * The volunteer selects the items they will pick up and confirms the assignment. * The system provides details of the pickup location and instructions. * Upon arrival at the pickup location, the volunteer verifies the items (e.g., through a QR code or confirmation from an admin). * The system updates the status of the items to "Picked Up" and logs the transaction in the database. * A notification is sent to the needy individual indicating that their items are on the way. | |
| **Post Conditions:** The system records the successful pickup of items.  The items are now marked as "In Transit" for delivery to needy individuals. | |

Table 4: Contact Needy One Items Use Case Description

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case Name:** | Contact Needy One |
| **Use Case Description:** To enable the volunteer to contact needy individuals for coordination regarding their requests or needs. | |
| **Primary Actor:** Volunteer | **Other Actors:** Database |
| **Relationship:**  **Includes:** Verify |  |
| **Stack Holders:** Volunteer |  |
| **Flow of Events:**  The volunteer logs into the system and navigates to the "Contact Needy One" section.  The system displays a list of needy individuals along with their basic information (e.g., name, request type).  The admin or volunteer selects a needy individual to contact.  The system retrieves the contact information (e.g., phone number, email, or chat interface).  The admin or volunteer initiates contact through the selected method.   * If using email, the system provides a template for structured communication. * If using chat, the system redirects to the integrated chat module.   The system logs the communication for future reference and accountability. | |
| **Post Conditions:** The needy individual is contacted for clarification or coordination regarding their request.  The communication details are logged for record-keeping purposes. | |

Table 5: Contact Donor Use Case Description

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case Name:** | Contact Donor |
| **Use Case Description:** To enable the admin or volunteer to contact donors for coordination regarding their donations or to provide updates. | |
| **Primary Actor:** Volunteer | **Other Actors:** Database |
| **Relationship:**  **Includes:** Verify |  |
| **Stack Holders:** Volunteer |  |
| **Flow of Events:**  The volunteer logs into the system and navigates to the "Contact Donor" section.  The system displays a list of donors along with their donation history and contact information.  The admin or volunteer selects a donor to contact.  The system retrieves the donor’s contact information (e.g., phone number, email, or chat interface).  The admin or volunteer initiates contact through the selected method.   * If using email, the system provides a template for structured communication. * If using chat, the system redirects to the integrated chat module.   The system logs the communication for tracking and accountability. | |
| **Post Conditions:** The donor is contacted for donation-related coordination or updates.  The communication is logged in the system for record-keeping. | |

**4.2.4 Needy One**

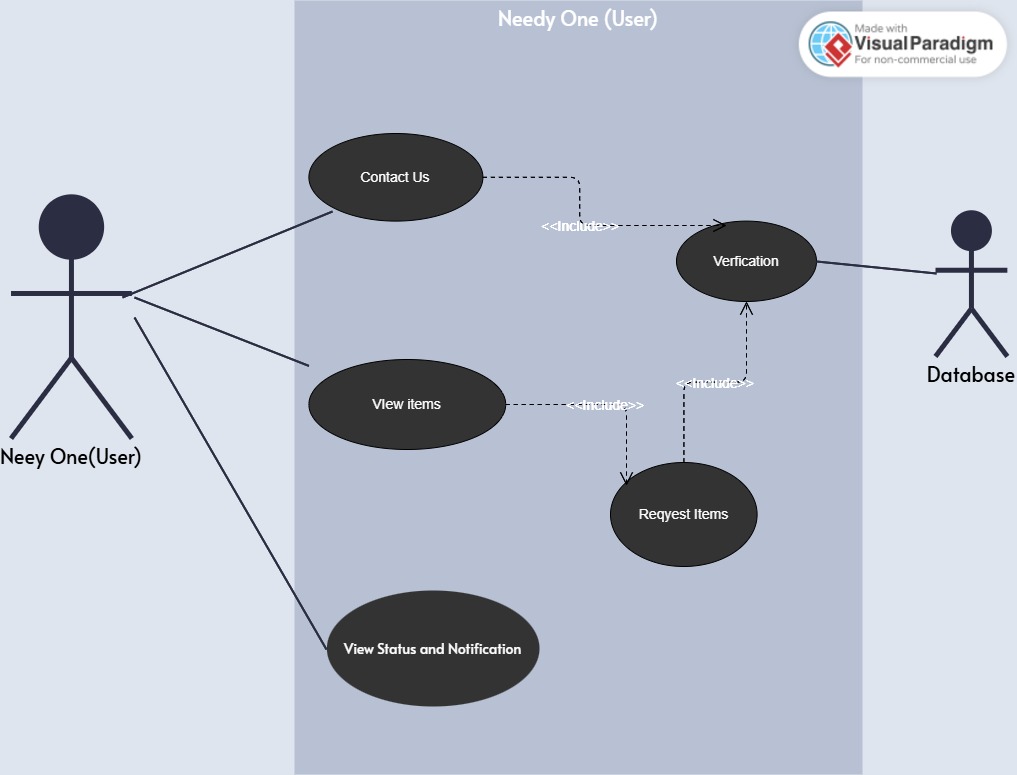


Table 1: Contact Us Use Case Description

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case Name:** | Contact Us |
| **Use Case Description:** To allow needy individuals to contact the website for assistance or to request volunteer support. | |
| **Primary Actor:** Needy One | **Other Actors:** Database |
| **Relationship:**  **Includes:** Verify |  |
| **Stack Holders:** Needy One |  |
| **Flow of Events:**  The needy individual navigates to the "Contact Us" section on the website.  They fill out a form providing their details (e.g., name, contact info, and type of assistance required).  The system validates the submitted information.  The request is logged in the system and forwarded to the admin for review.  The admin assigns a volunteer to handle the request and notifies the needy individual about the assigned volunteer.  The volunteer contacts the needy individual to provide the requested assistance. | |
| **Post Conditions:** The needy individual’s request for assistance is submitted and logged in the system.  A volunteer is assigned to assist, and the communication process is initiated. | |

Table 2: View Items Use Case Description

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case Name:** | View Items |
| **Use Case Description:** To allow needy individuals to view the available items for assistance or donation. | |
| **Primary Actor:** Needy One | **Other Actors:** Database |
| **Relationship:**  **Includes:** Verify |  |
| **Stack Holders:** Needy One |  |
| **Flow of Events:**  The needy individual logs into the system and navigates to the "View Items" section.  The system displays a list of available items categorized (e.g., food, clothes, medicines).  The user can search for specific items using keywords or filters (e.g., category, availability, or location).  The system retrieves and displays the filtered results based on the search criteria.  The needy individual selects an item to view its details, including the description, availability, and pickup/drop-off process. | |
| **Post Conditions:** The needy individual can browse and view the details of available items.  The system logs the interaction for future analytics and tracking. | |

Table 3: Status and Notification Use Case Description

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case Name:** | Status and Notification |
| **Use Case Description:** To allow needy individuals to view the status of their requests and notifications related to updates or assistance. | |
| **Primary Actor:** Needy One | **Other Actors:** Database |
| **Relationship:**  **Includes:** Verify |  |
| **Stack Holders:** Needy One |  |
| **Flow of Events:**  The user navigates to the "Status and Notifications" section.  The system retrieves the current status of their requests (e.g., pending, approved, or in-progress).  The system also fetches and displays notifications (e.g., updates on assigned volunteers, pickup schedules, or additional instructions).  The user reviews the displayed statuses and notifications. | |
| **Post Conditions:** The needy individual is informed about the current status of their requests and any related updates.  The system logs the interaction for tracking and analytics. | |

**5.Non-functional Requirements**

**5.1 Performance Requirements**

The system's performance is critical to ensuring a seamless user experience. Key performance characteristics include:

* **Speed**: The platform must respond to user interactions within 2 seconds under normal conditions.
* **Concurrency**: Support for at least 100 concurrent users accessing the platform without significant performance degradation.
* **Capacity**: Efficient handling of large volumes of data, including user profiles, donation details, and volunteer activities.
* **Reliability**: The system should maintain an uptime of 99.9% to ensure continuous availability for users.
* **Precision**: Accurate matching of donations and requests based on predefined algorithms to meet user needs effectively.

**5.2 Safety Requirements**

* **Data Protection**: Regular automated backups must be performed to prevent data loss in case of system failures or cyberattacks.
* **Error Handling**: Robust error handling mechanisms must be in place to address unexpected issues and prevent harm to user data or functionality.
* **Access Control**: Unauthorized access must be strictly prevented through secure login and role-based authentication protocols.
* **Compliance**: The platform must adhere to relevant data privacy regulations, such as GDPR or local equivalents, to ensure legal compliance.
* **Prevention of Harmful Actions**: Safeguards should be implemented to prevent malicious activities, such as spamming, hacking, or exploitation of system vulnerabilities.
* Regular data backups and robust error handling mechanisms.

**5.3 Security Requirements**

* Role-Based Access Control: Access to different functionalities must be restricted based on user roles, such as donors, volunteers, and administrators.
* Data Encryption: Sensitive data such as user credentials and payment information must be encrypted both in transit and at rest to prevent unauthorized access.
* Regular Security Audits: Conduct periodic security audits to identify and fix vulnerabilities.
* Compliance: Adhere to industry-standard security regulations, including GDPR, to ensure the protection of user data and maintain trust.
* Role-based access control with encrypted data storage.

**5.4 User Documentation**

* User Manuals: Comprehensive manuals will be provided to guide users through platform functionalities.
* FAQs: A dedicated FAQ section will address common user queries and concerns.
* Tutorial Videos: Step-by-step video tutorials will assist users in understanding key operations like creating accounts, posting donations, and signing up for volunteer activities.
* Context-Sensitive Help: Integrated tooltips and help sections within the platform to provide on-the-spot assistance.
* Support Portal: An online support portal will offer detailed articles and user forums for additional help.
* Comprehensive user guides, FAQs, and tutorials for first-time users.

**6. References**

1. "Building Scalable Applications with MERN Stack," Developer Resources, 2024.
2. "Designing Social Welfare Systems Using AI Chatbots," AI Research Journal, Vol. 15, 2023.
3. GoogleMapsAPIDocumentation: https://developers.google.com/maps/documentation
4. Dialogflow Documentation: https://cloud.google.com/dialogflow/docs
5. "Social Welfare Initiatives: Technology and Impact," Social Development Reports, 2023.
6. Steele, J., and Adams, D., Full Stack React, Node, and MongoDB: Learn to Build Modern Web Applications, Addison-Wesley, Boston, 2019.
7. [https://www.gofundme.com](<https://www.gofundme.com>

**7.Appendcies**