HopeFull

IS - Group 20

Interim Report

Table of Content

Introduction

Domain Description

The Hopefull Platform is a purpose-driven web application designed to facilitate processes of donations and volunteering while empowering communities in need. By Bridging the gap between donors, volunteers, and recipients, the platform creates an environment that supports charitable activities.

It also provides a marketplace (Hopefull MarketPlace) for differently abled individuals, enabling them to sell their goods(e.g.- Crafts, Artwork) and gain financial independence.

Current System and Limitation

Currently, there are very few platforms in Sri Lanka to incorporate donation activities but there is a lack of a system that provides a centralized, engaging platform for these charitable activities. And also most current systems do not offer programmes that will enhance user engagement.

And also these platforms do not offer a platform such as HopfullMarketPlace to ensure the financial independence of differently abled people.

Goal

Our goal is to create an effective, transparent, and engaging platform that connects donors and volunteers with recipients in need and also it provides a great platform for differently abled individuals to sell their goods to generate income.

Objectives

1. Facilitate monetary and non-monetary donations.
2. Support and promote volunteering opportunities.
3. Enable people with disabilities to sell goods via a marketplace.
4. Provide gamification features like leaderboards and badges.
5. Ensure efficient logistics and inventory management.
6. Promote transparency and security in all transactions.

Assumptions

1. This whole operation is managed through a charitable organization
2. The system admin, Authentication Moderator, Regional Moderator, Hopefull marketplace Sellers, and Delivery officers are the internal stakeholders or the users.
3. Donors/ Volunteers and Recipients are external stakeholders or users
4. Internet access is available to all users
5. Recipients have the capability to provide the proper documentation for requests
6. The marketplace place managed externally(the nonmonetary logistics and storage and marketplace are handled separately).

Requirements

Stakeholders/ Actors

The Hopefull platform involves several stakeholders, each with specific roles and responsibilities:

1. Users
   - Donors
   - Volunteers
2. Recipients (Organizations or individuals in need)
3. System Admin
4. Authentication Moderator
5. Regional Officer
6. Hopefull Market Place Sellers
7. Delivery Officer

Functional And Non-Functional Requirements

Functional Requirements

1. Unregistered users
   - View an overview of the platform and what it offers
   - Register if interested

2. Donors or Volunteers
   - Register and manage profiles.
   - Browse and search donation requests and volunteer opportunities.
   - Make monetary and non-monetary donations.
   - Access feedback and impact reports.
   - Participate in gamification features like leaderboards and badges.
   - Purchase goods from the Hopefull Market Place.

3. Recipients
   - Register with approval from the Authentication Moderator.
   - Create and manage requests for donations or volunteers.
   - Track request statuses.
   - Post-volunteering opportunities.
   - Provide feedback and impact assessments.

4. Regional Officer
    - Plan, coordinate, and manage logistics for donations.
    - Track shipments.
    - Update inventory in real-time.
    - Generate e-receipts for dropped-off items.

5. Marketplace Sellers
    - Manage profiles, product listings, and inventory.
    - Communicate with buyers.
    - Handle orders.

6. Delivery Officers
    - Coordinate and handle pick-up and delivery of donations.
    - Ensure safe item handling during transit.
    - Provide shipment tracking and gather feedback.

7. System Admin
    - Manage user profiles.
    - Handle technical issues and reports.
    - Ban accounts if necessary.

Non - Functional Requirements

1. Security
    - Secure payment gateways for monetary donations
    - Support anonymous donations

2. Transparency
    - Provide real-time updates on inventory and donation statuses.
    - Generate e-receipts for tracking accountability
3. User Engagement
    - Implement gamification features like badges and leaderboards
    - Maintain active community forums

4. Scalability
    - Support the platform's growth to include more users, donations, and volunteer opportunities.

5. Usability
    - Ensure the platform is user-friendly, with intuitive navigation and clear instructions.

In-Scope and Out-Scope

In-Scope

- All the above-mentioned functional requirements are in the Scope
- Secure and transparent processes for donations and marketplace transactions.
- Gamification elements for user engagement
- Inventory and logistic management
- Monetary and non-monetary donations
- Volunteer activity coordination
- MarketPlace operations
- Online donations

Out-Scope

- The platform does not support or manage offline donation activities
- The platform does not cover logistic services in areas without designated drop-off points
- Donation of perishable items like food is not facilitated.
- The platform is a web application, it is not optimized as a mobile app.

Constraints and Limitations

1. Logistical Constraints
    - Limited coverage in specific regions due to lack of drop-off points
    - Dependencies on delivery officers and regional officers for effective logistics

2. Technology constraints
    - The system is restricted to web-based interactions and lacks mobile application functionality.

- Recipients should have access to the necessary devices to make requests or make accounts.

3. Content Moderation
    - Recipients should wait for their requests to be approved and verified.

Proposed System's Architecture

Components and Their Functionalities

1. Frontend (Presentation Layer)

Functionality: The front end is the user-facing part of the HopeFull application. It interacts directly with the end-users and allows them to perform various activities such as:

- Donors can view and contribute donations (monetary or goods).
- Recipients can request donations and track progress.
- Volunteers can sign up and manage volunteer tasks.
- Access and interact with the HopeFull Marketplace for products.
- Manage user profiles and switch between roles (Donor, Volunteer, Recipient).

2. Backend (Application/Business Logic Layer)

Functionality: The backend is responsible for the core logic of the application. It processes requests from the frontend and handles:

- User management and authentication (sign-up, login, profile management).
- Donation processing (both monetary and goods donations).
- Recipient request handling (creating, viewing, and processing donation requests).
- Volunteer task management and coordination.
- Donation logistics, including delivery tracking and inventory management.
- Marketplace interactions (order placement, inventory updates, etc.).

3. Database (Data Layer)

Functionality: The database stores all the data required by the system, ensuring it remains persistent and consistent. The main responsibilities include:

- Storing user data (Donors, Volunteers, Recipients, Admins, etc.).

        - Storing donation requests, statuses, and transaction details.
        - Maintaining product listings, inventory data, and order information for the HopeFull Marketplace.
        - Storing feedback, impact reports, and analytics related to donations and volunteer activities.

Component Interactions

- **Frontend ↔ Backend**
    - The front end communicates with the backend through HTTP requests (REST API calls). The frontend sends user requests such as donating, creating requests, signing up, and more, to the backend, which then processes the logic and returns the necessary responses (e.g., confirmation, data, and error messages).
    - This interaction enables real-time updates on the user interface based on the backend processing.

- **Backend ↔ Database**
    - The backend communicates directly with the database to fetch, update, or delete data as needed. This could involve retrieving donor and recipient data, processing transactions, storing request details, or updating the inventory and order statuses in the database.
    - All database queries are executed in response to the actions performed through the backend, ensuring that the front end has access to real-time data.

- **Frontend ↔ Database (Indirect)**
    - The front end does not interact directly with the database but relies on the back end to fetch or manipulate data in the database. The front end only receives the processed data from the backend to display to the user.
    - The database acts as a persistent store for all data, while the backend manages access to this data.

Architecture Model

We are currently implementing a 3-tier Architecture, which suits our team dynamics and the development phase of the HopeFull application. The three layers are:

1. Presentation Layer (Frontend) – Responsible for handling user interaction.
2. Application Layer (Backend) – Contains the logic that drives the system, processes requests, and interacts with the database.
3. Data Layer (Database) – Manages and stores all the data used by the application.

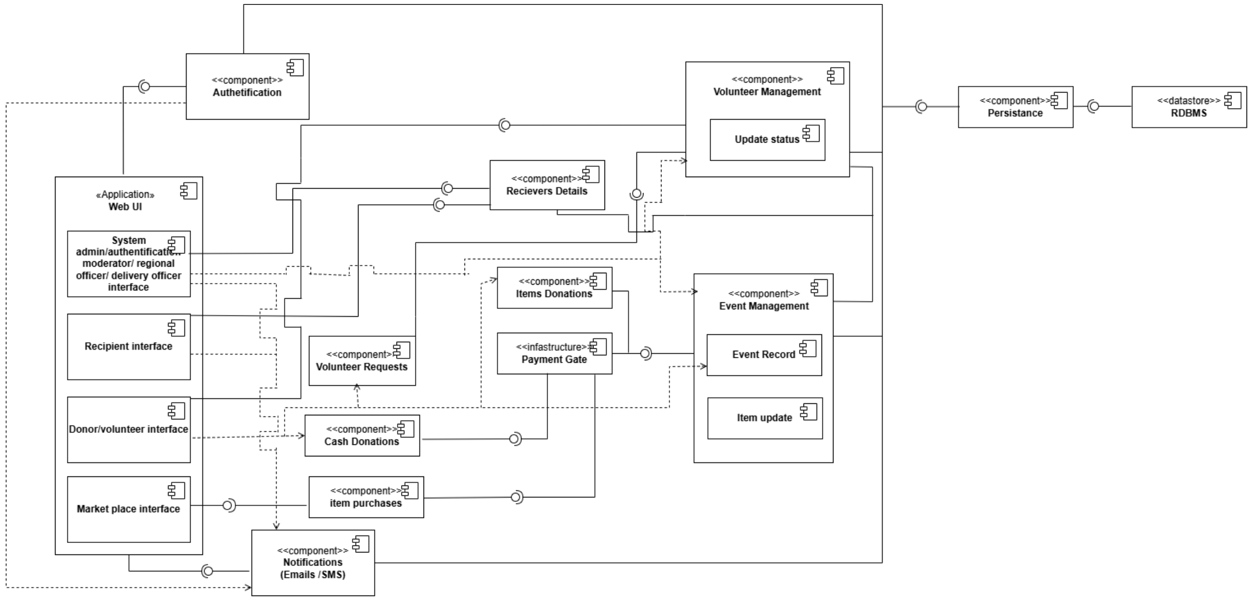
This architecture is chosen for its simplicity and modularity, allowing for each tier to evolve independently. This makes it easier to update or replace individual layers without affecting the overall system, and it fits well with our current team structure.

In the future, as the project scales and grows in complexity, we are planning to transition to an MVC (Model-View-Controller) Architecture. The MVC architecture will help us separate the concerns of data management (Model), user interface (View), and request processing (Controller), which will lead to better maintainability and scalability.

- Model - Will manage the data and application logic (e.g., handling donation requests, and user data).
- View - Will be responsible for rendering the UI and presenting the data to the user.
- Controller - Will manage user input and act as an intermediary between the Model and the View, processing user actions and updating the View accordingly.

The MVC architecture will allow for a more structured approach to development, making it easier to manage growing functionality and improving the quality of our codebase as we move towards future development stages.

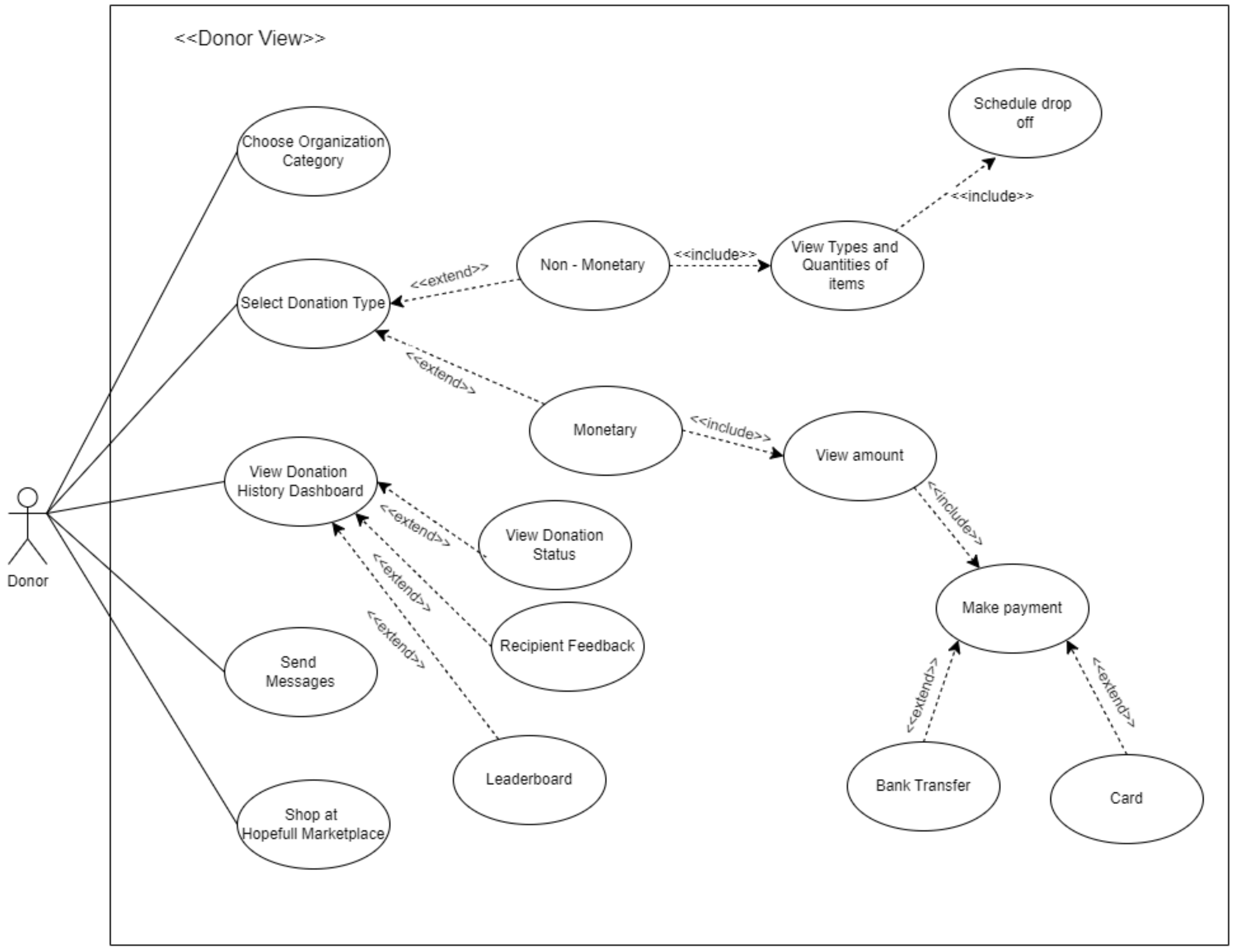
This 3-tier Architecture is currently the most suitable option for our team's workflow, but transitioning to MVC will ensure better scalability and maintainability as the project expands.



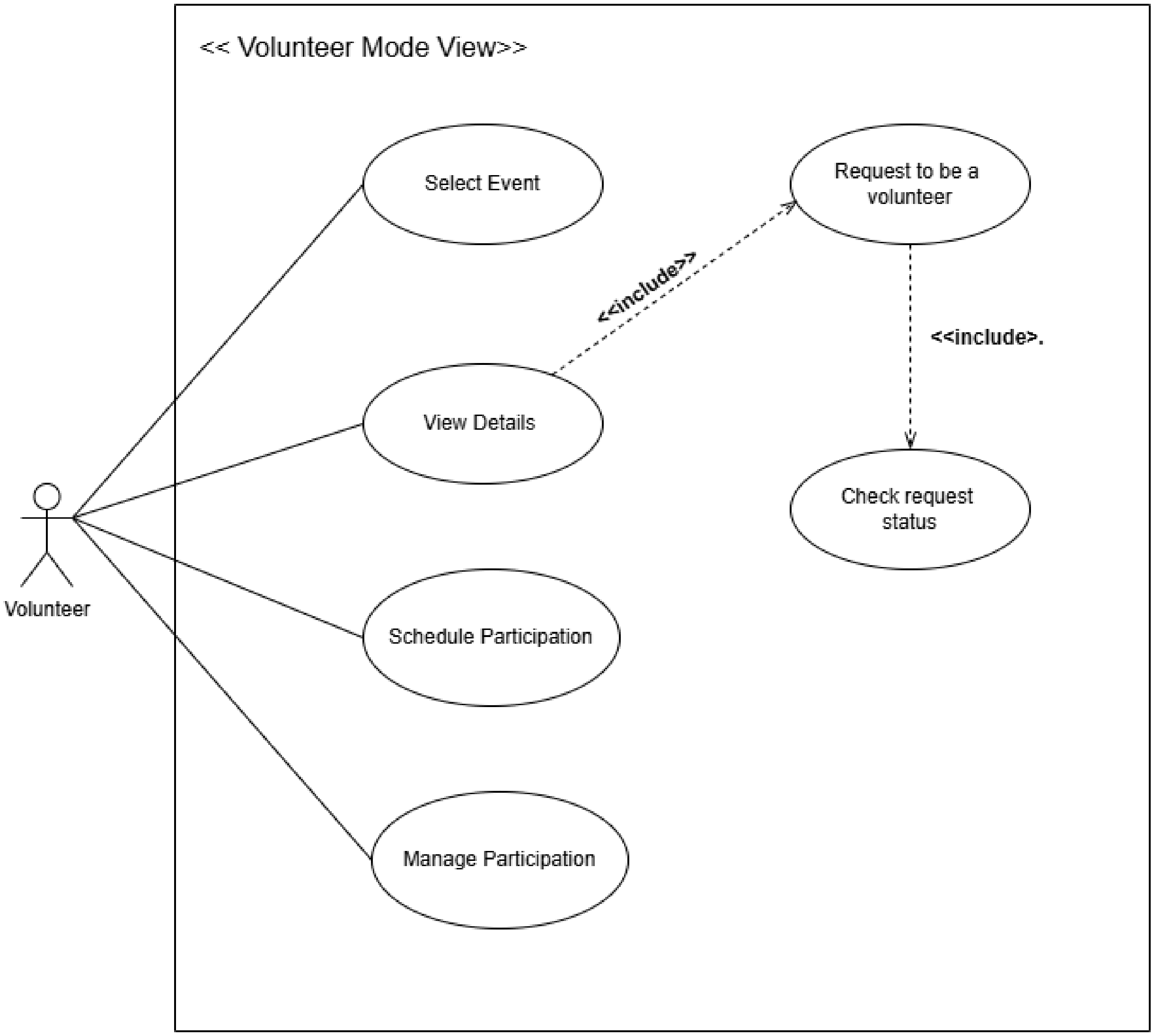
System Design Diagrams

Use Case Diagrams

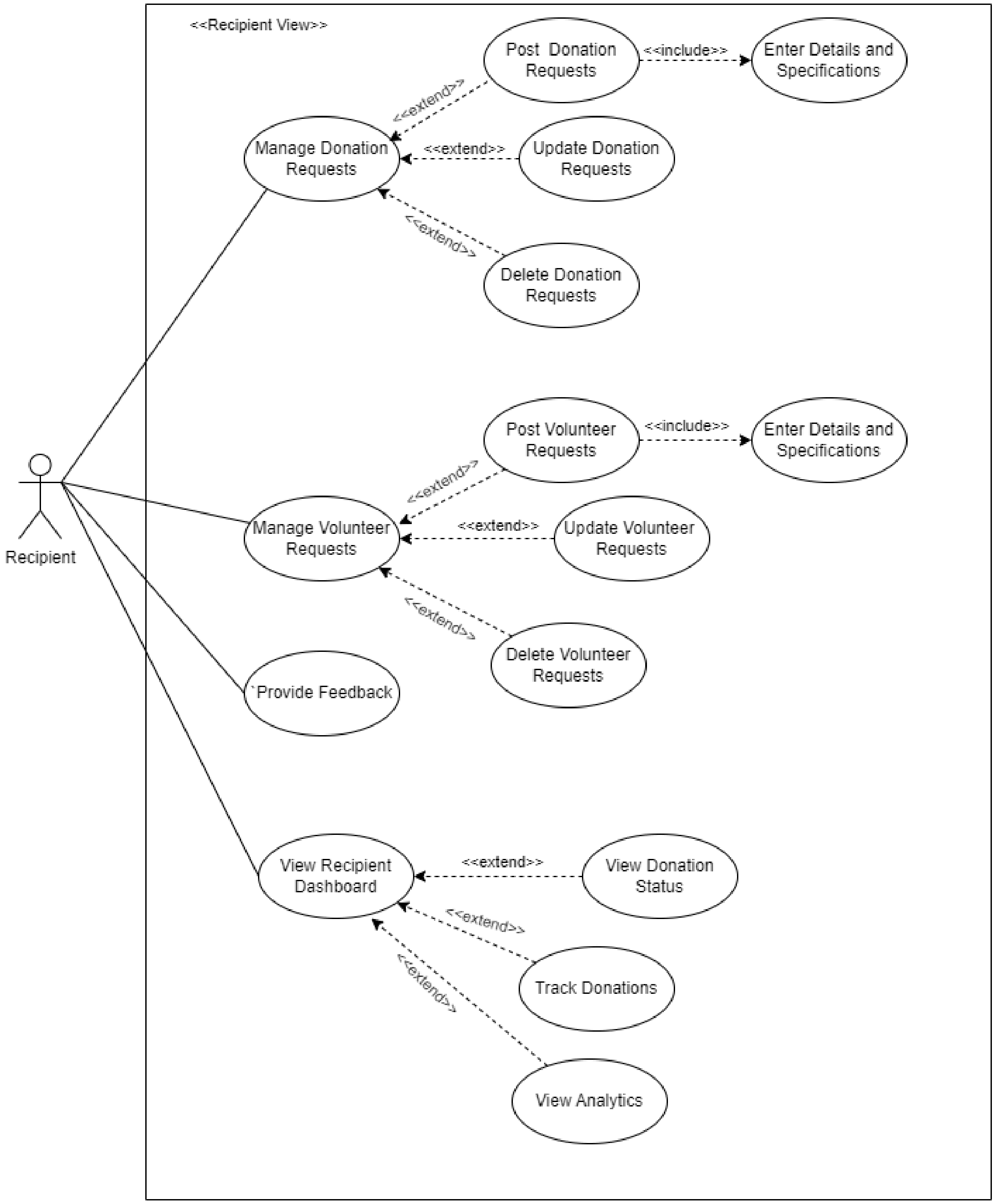
1. Donor View



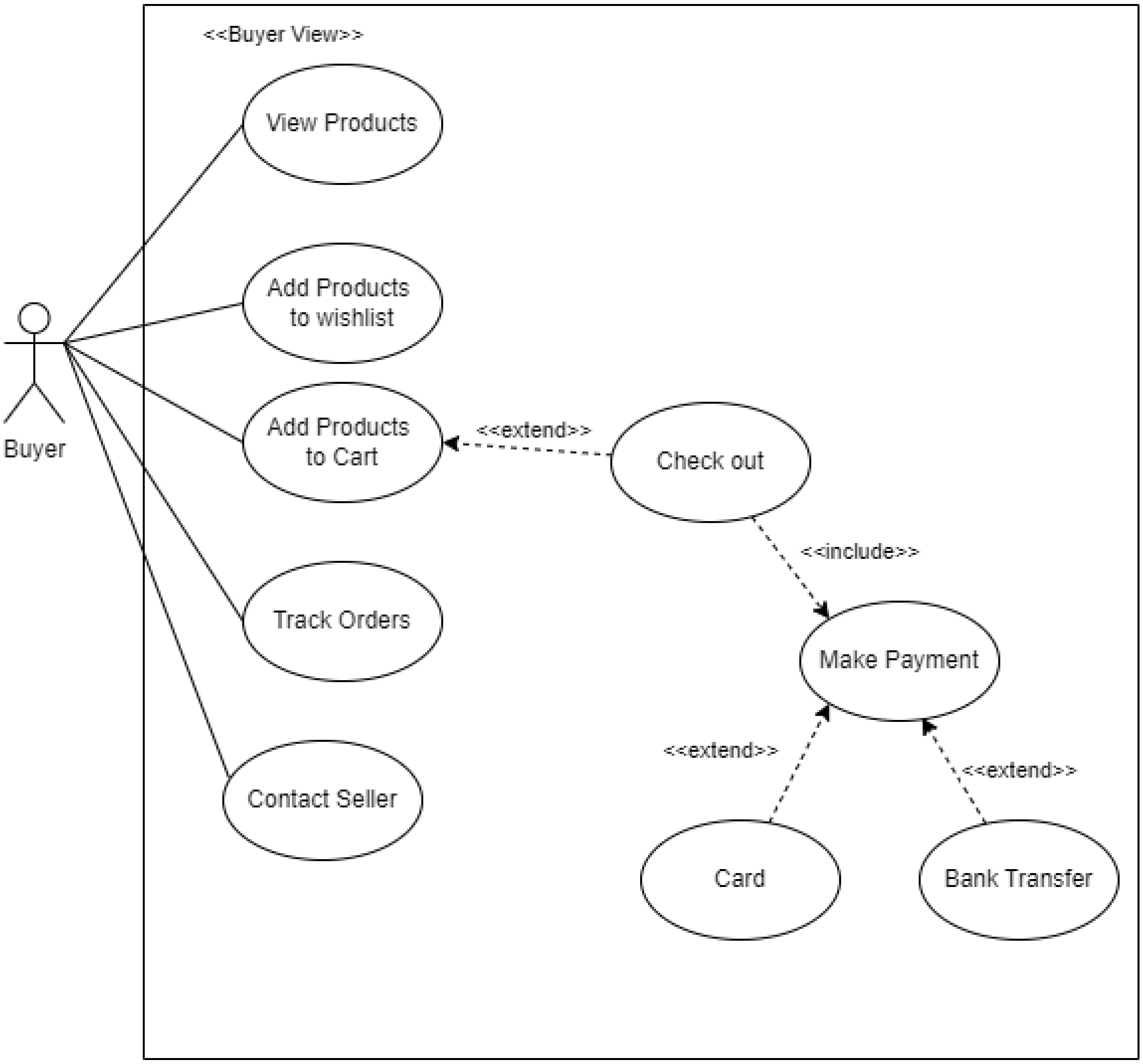
2. Volunteer View



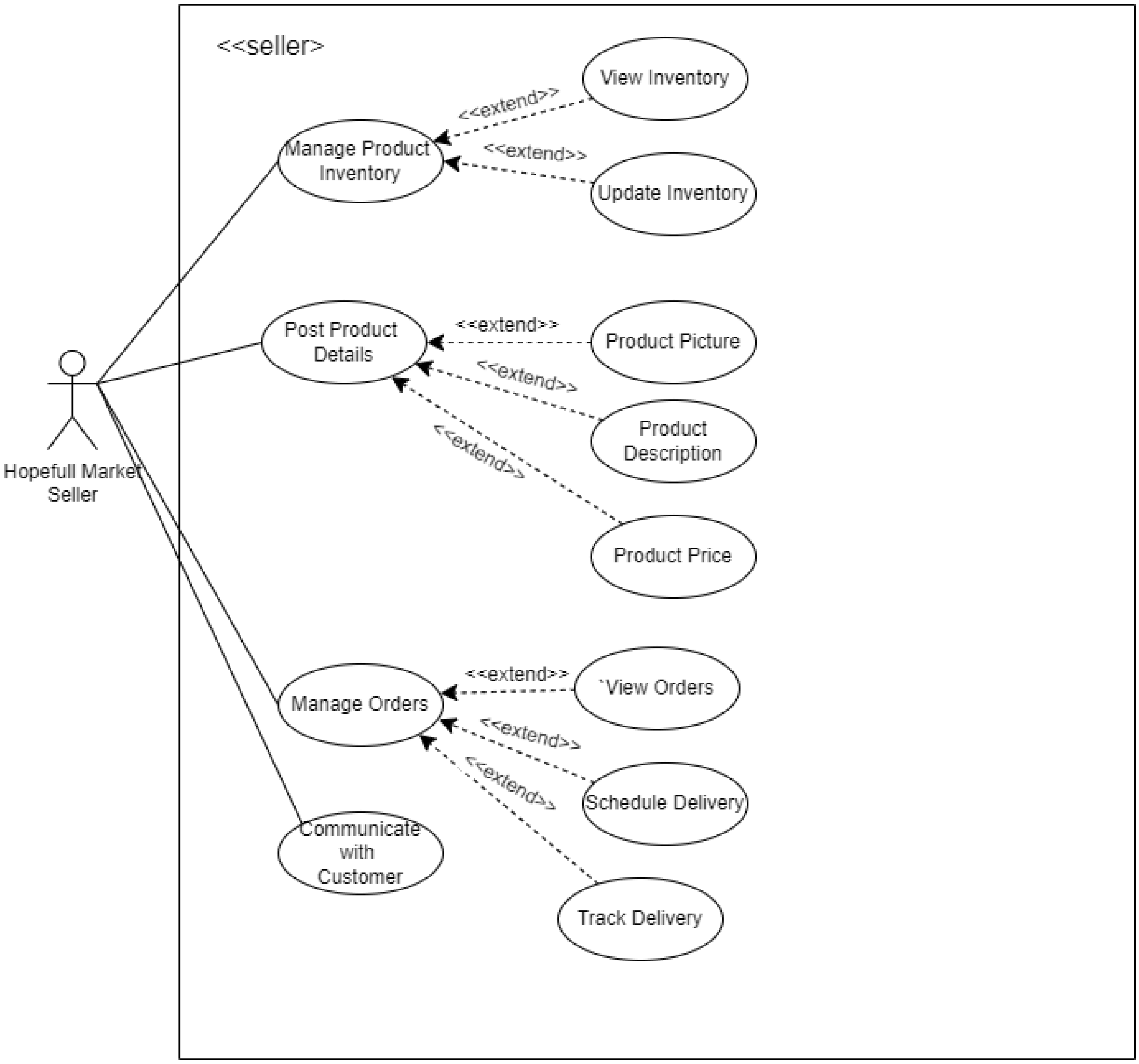
3. Recipient View



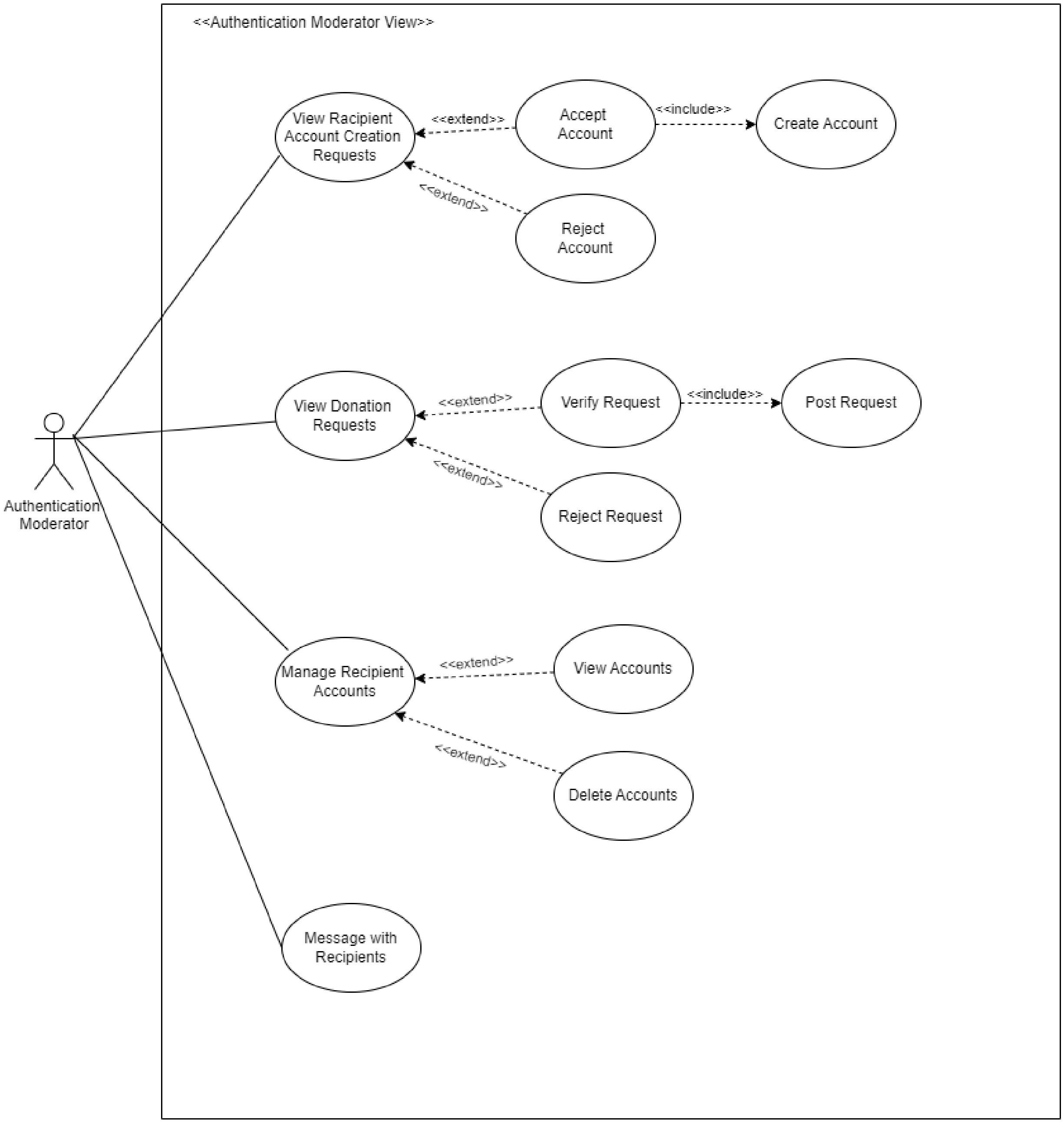
4. MarketPlace Buyer View



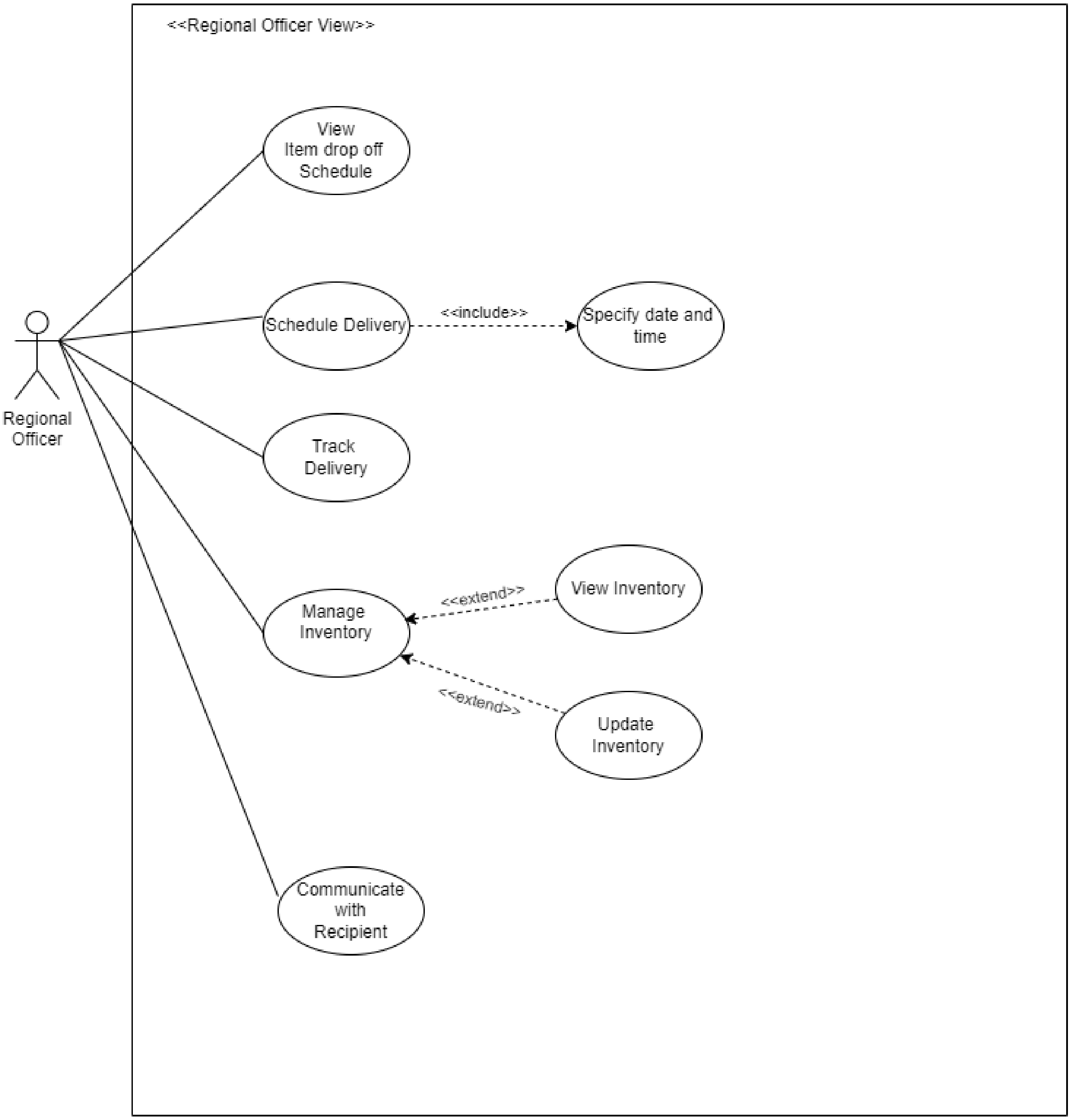
5. MarketPlace Seller



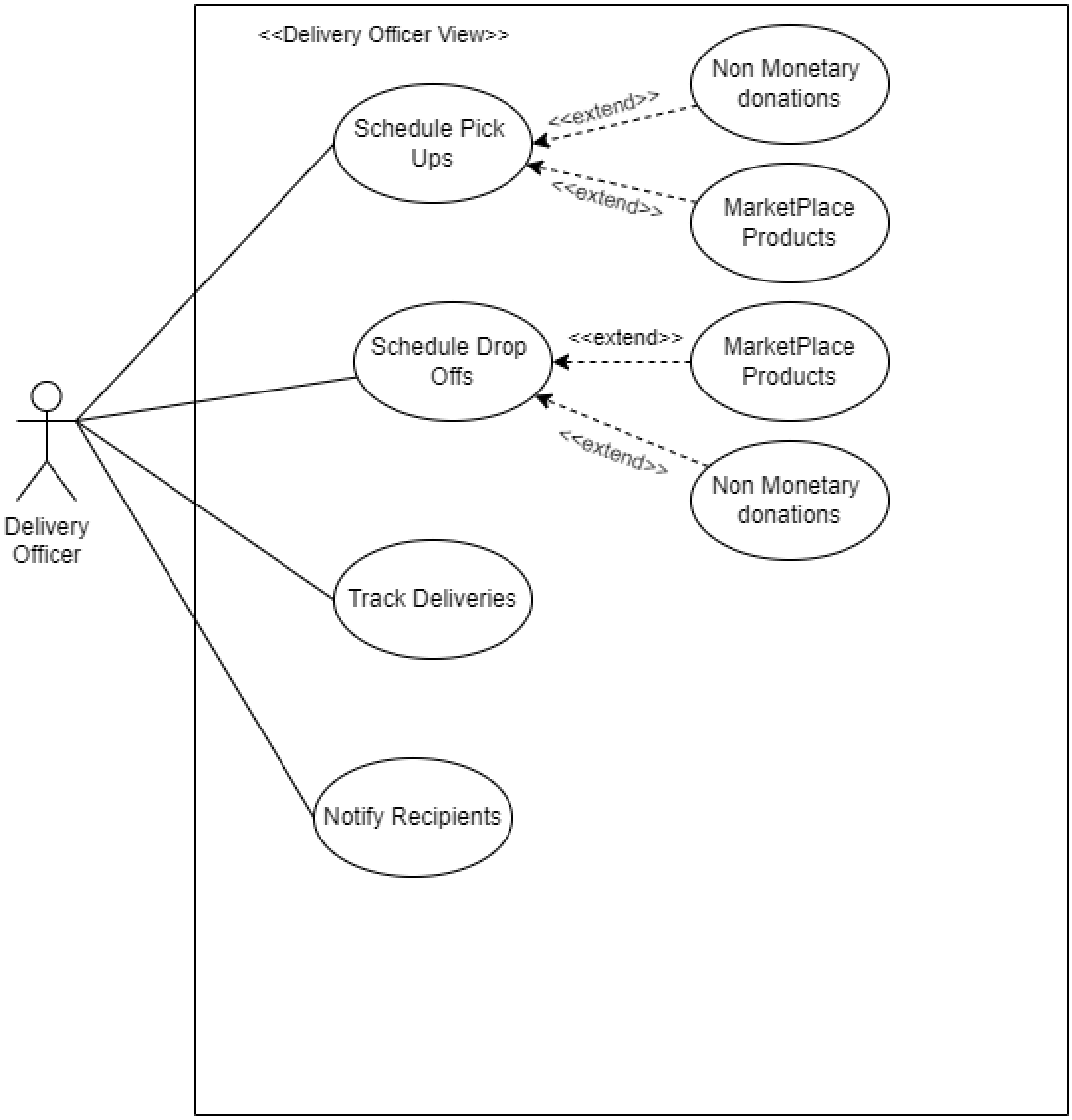
6. Authentication Moderator View



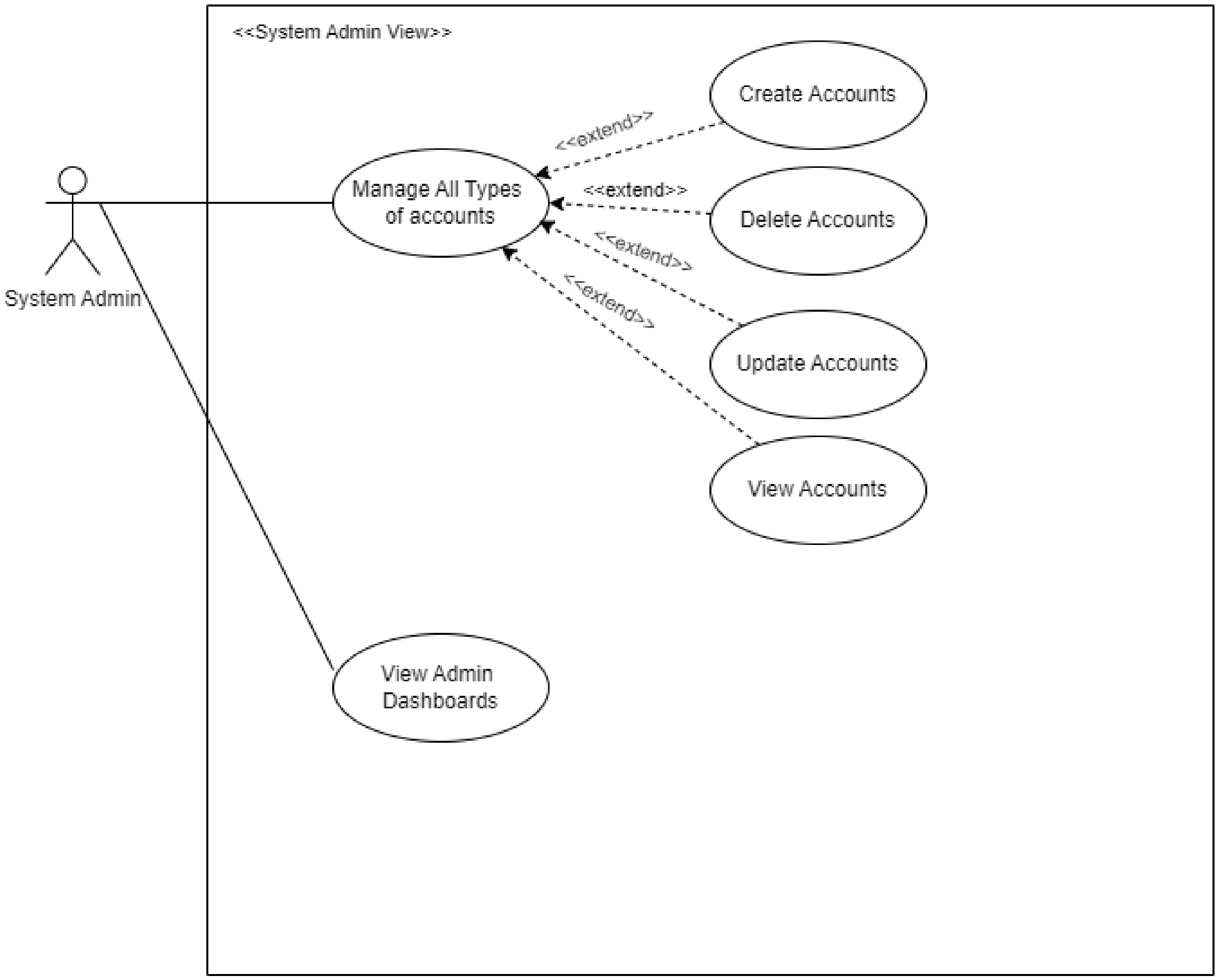
7. Regional Officer View



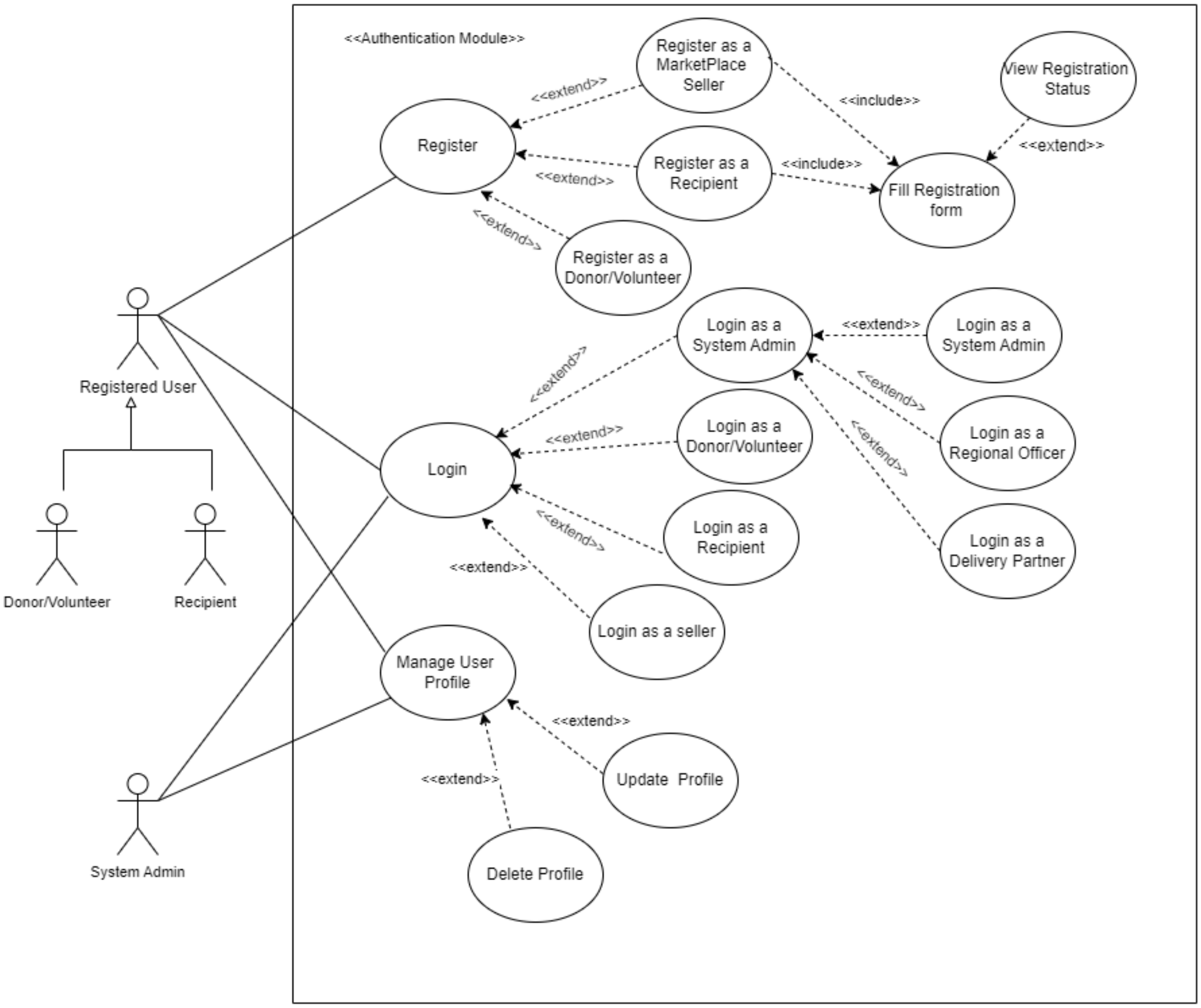
8. Delivery Officer View



9. System Admin



10. Authentication Module for users



User Narratives

1. Donor View Narrative

Use Case ID	01
Use Case Name	Select Donation Type
Primary Actor	Donor
Description	The donor selects a category of organizations (education, healthcare) to narrow down the recipient organizations.
Pre-Conditions	The donor must be logged in
Main Scenario	● Donor logs in to the platform ● Navigate the Hopefull donation page ● Choose a category to choose a customized selection ● The system displays organizations in the selected category.
Expectations	Donations are filtered based on the filtration and there should be no errors
Post - Conditions	The system updates the donor's preferences and displays filtered organizations

Use Case ID	02
Use Case Name	Select Donation Type
Primary Actor	Donor
Description	Donor Selects between monetary and non-monetary donations
Pre-Conditions	-
Main Scenario	● Donor views donation options ● Choose between Monetary and non-monetary donations ● The system redirects to the respective workflow

Expectations	The system ensures only valid options are displayed. There should be no interruptions in the selection flow
Post - Conditions	Donation workflow is determined based on the type selected.

Use Case ID	03
Use Case Name	View Types and Quantities of Items
Primary Actor	Donor
Description	Donor views requested non-monetary items and their required quantities.
Pre-Conditions	The donor has selected Non-Monetary as the donation type
Main Scenario	● The donor selects a donation request ● The system displays a list of items and their quantities. ● Donor reviews the list
Expectations	Accurate item and quantity details are displayed
Post - Conditions	The donor can proceed to schedule the drop-off

Use Case ID	04
Use Case Name	Schedule Drop off
Primary Actor	Donor
Description	Donor schedules the time and location for item drop-off
Pre-Conditions	The donor selected items for donations
Main Scenario	● Donor navigates the scheduling page ● Select drop off point ● Select the date and time ● Confirm the drop-off Schedule

Expectations	The drop off scheduled without conflict
Post - Conditions	The drop-off details are saved and recipients are notified

Use Case ID	05
Use Case Name	Make Payment
Primary Actor	Donor
Description	The donor makes monetary donations via card or bank transfer
Pre-Conditions	The donor has selected Monetary as the donation type and viewed the donation amount
Main Scenario	● The donor selects the payment method ● Enters payment details ● Confirms the payment ● The system processes the payment
Expectations	The transaction is securely processed. There should be no payment errors.
Post - Conditions	The donation is logged and the recipient is notified

Use Case ID	06
Use Case Name	View the donation history dashboard
Primary Actor	Donor
Description	Donor views a dashboard with their past donations
Pre-Conditions	The donor should have an account and log in
Main Scenario	● Donor accesses the donation history dashboard ● The system displays a list of past donations
Expectations	Complete and accurate records are displayed

Post - Conditions	The donor gains insights into their contribution history

2. Volunteer View

Use Case ID	07
Use Case Name	Select Event and View Details
Primary Actors	Volunteer
Description	Allows the volunteers to browse and select the events they wish to participate in. And display detailed information about a selected event
Pre-Conditions	Volunteers should have an account and should be logged in and need to have access to the platform.
Main Scenario	● Volunteer logs into the systems ● Navigates the system and select an event for liking ● Display the details like date,time , location and description
Expectations	.Volunteers are presented with an intuitive list of events to select from
Post - Conditions	The selected event is stored for further actions.

Use Case ID	09
Use Case Name	Schedule Participation
Primary Actors	Volunteer
Description	Allows the volunteer to plan their participation schedule based on the approval
Pre-Conditions	The volunteer has approved requests for one or more events
Main Scenario	● After the approval, the volunteer schedules the

	participation for the event ● Set their availability
Expectations	Schedule conflicts are minimized and availability is successfully saved
Post - Conditions	Scheduled participation is stored in the system

Use Case ID	10
Use Case Name	Manage Participation
Primary Actor	Volunteer
Description	Enables the volunteer to update or cancel their participation in events they are scheduled for.
Pre-Conditions	The volunteer should be scheduled for one or more events
Main Scenario	● Volunteer logs in ● Navigates manage Participation ● Updates their participation details or cancels participation.
Expectations	Participation is updated accurately and cancellations are handled smoothly
Post - Conditions	Updated participation details are stored in the system.

3. Recipient View

Use Case ID	11
Use Case Name	Post Donation Requests
Primary Actors	Recipient
Description	Enables a recipient to create donation requests

Pre-Conditions	The recipient should have a verified account
Main Scenario	● The recipient navigates to the post donations request page ● Enter the donation details ● Confirms and submits the request
Expectations	Provide an interface for the recipients to enter their requests
Post - Conditions	A new donation request is created and listed

Use Case ID	12
Use Case Name	Manage Donation Requests
Primary Actors	Recipient
Description	Allows the recipient to update or delete requests
Pre-Conditions	Volunteers should have an account and should be logged and need to have access to the platform.
Main Scenario	● Volunteer logs into the systems ● Navigates the system and select an event for liking ● Display the details like date,time , location and description
Expectations	.Volunteers are presented with an intuitive list of events to select from
Post - Conditions	The selected event is stored for further actions.

Use Case ID	13
Use Case Name	Provide Feedback
Primary Actors	Recipient
Description	Allows the recipient to give feedback on the services.

Pre-Conditions	The feedback module is accessible
Main Scenario	- The recipient navigates to the feedback section. - Fill in the feedback form. - Submits feedback
Expectations	Gives an opportunity for the recipients to share their gratitude.
Post - Conditions	Feedback is submitted successfully.

Use Case ID	14
Use Case Name	View Recipient Dashboard
Primary Actors	Recipient
Description	Allows the recipients to monitor the donation status and analytics
Pre-Conditions	The dashboard is configured for the recipient
Main Scenario	- The recipient logs into the system - Navigate to the dashboard - Views donations statuses, tracking information, and analytics.
Expectations	The information should be organized and displayed with clarity
Post - Conditions	Insights are available for decision-making

4. Buyer View

Use Case ID	15
Use Case Name	Add products to the cart
Primary Actors	Buyer

Description	Enables the buyer to add items to their cart
Pre-Conditions	Buyers should view and select items
Main Scenario	● Buyer selects a product ● Adds the product to the cart and proceeds to the cart page
Expectations	Should give the buyers a perfect shopping experience
Post - Conditions	Selected Products are in the cart

Use Case ID	16
Use Case Name	Check Out
Primary Actors	Buyer
Description	Allows the buyer to complete the purchase process.
Pre-Conditions	The cart contains selected products
Main Scenario	● The buyer reviews the cart ● Proceeds to check out ● Enters payment details and confirms.
Expectations	Easy checkout without any interruptions.
Post - Conditions	Payment is processed, and the order is placed

Use Case ID	17
Use Case Name	Track Orders
Primary Actors	Buyer
Description	Enables the buyer to track the status of their orders
Pre-Conditions	Order details are available in the system
Main Scenario	● Buyer navigates to the order tracking page ● Selects an order to track ● Views real-time order updates

Expectations	Buyers are able to smoothly track their orders in real-time
Post - Conditions	Buyers are informed of the order status

Use Case ID	18
Use Case Name	Contact seller
Primary Actors	Buyer
Description	Allows the buyer to reach out to the seller for inquiries
Pre-Conditions	Seller contact information is available
Main Scenario	● The buyer selects the seller ● Sends a message or query ● Waits for a response
Expectations	Streamline smooth interrupted communication with the marketplace admin
Post - Conditions	Communication is initiated with the seller

5. Authentication Moderator View

Use Case ID	19
Use Case Name	View Recipient Account Creation requests
Primary Actors	Authentication Moderator
Description	Enables moderators to review and process new account requests.
Pre-Conditions	New recipient account requests are available
Main Scenario	● The moderator logs into the system ● Views pending account requests ● Selects specific request

Expectations	Prevent fraud accounts from being created
Post - Conditions	Account requests are available for further actions

Use Case ID	20
Use Case Name	Accept Account
Primary Actors	Authentication Moderator
Description	Allows the moderator to approve a recipient’s request.
Pre-Conditions	The recipient’s account request is verified and ready for approval.
Main Scenario	● The moderator reviews the account request ● Confirms the details ● Accepts the request and creates the account
Expectations	Only verified accounts are created
Post - Conditions	The recipient account is created and Activated.

Use Case ID	21
Use Case Name	Reject Account
Primary Actors	Authentication Moderator
Description	Allows the moderator to reject a recipient’s account request
Pre-Conditions	A recipient’s account request is flagged or deemed invalid.
Main Scenario	● Moderator Reviews the account request ● Identifies issues. ● Rejects the account request.
Expectations	Make sure that those who are truly needed are the ones making the accounts
Post - Conditions	The account request is marked as rejected

Use Case ID	22
Use Case Name	View Donation Requests
Primary Actors	Authentication Moderator
Description	Enables moderators to review and verify donation requests
Pre-Conditions	Donation requests are submitted by the recipients
Main Scenario	● The moderator navigates to the donation request section ● Views pending donation requests ● Selects a request for review
Expectations	To validate the requests
Post - Conditions	Donation requests are visible for action

Use Case ID	23
Use Case Name	Verify and reject requests
Primary Actors	Authentication Moderator
Description	Allows to approve valid donation requests and reject the other ones
Pre-Conditions	There should be requests that are pending to be verified
Main Scenario	● The moderator reviews the donation request ● Verify the details ● Approve the request for posting ● Or reject it
Expectations	To prevent faulty requests from being posted
Post - Conditions	The donation request is approved and posted or rejected and informed

Use Case ID	24
Use Case Name	Manage Recipient Accounts
Primary Actors	Authentication Moderator
Description	Allows the moderators to ban or unban recipient accounts
Pre-Conditions	Recipient accounts exist in the system
Main Scenario	● The moderator navigates to the account management page ● Views recipient accounts ● Selects an account to view or delete
Expectations	To clean the platform of possible scams
Post - Conditions	Recipient accounts are managed as needed

6. Regional Officer View

Use Case ID	25
Use Case Name	View Item drop-off Schedule
Primary Actors	Regional Officer
Description	The regional officer views the schedule for item drop-offs
Pre-Conditions	The systems must have existing schedules stored
Main Scenario	● The officer logs into the system ● The officer navigates to the scheduled selection ● The officer views the list of item drop-off schedules
Expectations	The system displays accurate schedules
Post - Conditions	The officer gains information about upcoming drop-offs

Use Case ID	26
Use Case Name	Schedule Delivery
Primary Actors	Regional Officer
Description	The regional officer schedules item deliveries to recipients
Pre-Conditions	Items must be available for delivery
Main Scenario	● Officers specify the date and the time ● Confirm the delivery schedule
Expectations	Delivery is added to the system's schedule
Post - Conditions	A new delivery schedule is created.

Use Case ID	27
Use Case Name	Track Delivery
Primary Actors	Regional Officer
Description	Regional Officer tracks the progress of the scheduled deliveries
Pre-Conditions	Deliveries must have been previously scheduled
Main Scenario	● Officer selects track delivery ● The system displays current delivery status ● The officer checks the details of a specific delivery
Expectations	Delivery Status are updated in real-time
Post - Conditions	The officer stays informed about the delivery progress

Use Case ID	28
Use Case Name	Manage Inventory
Primary Actors	Regional Officer
Description	Regional officer views and updates inventory details
Pre-Conditions	Inventory items must exist in the system.
Main Scenario	● Officer goes to the inventory page ● Officer views inventory list ● Officer updates inventory details
Expectations	Changes to inventory are saved successfully
Post - Conditions	Inventory reflects updated information

7. Delivery Officer View

Use Case ID	29
Use Case Name	Schedule Pick Ups and drop-offs
Primary Actors	Delivery Officer
Description	Delivery officer schedules item pick-ups from various locations
Pre-Conditions	Pick-up requests must be logged in the system
Main Scenario	● Go to the scheduled pick up page ● The officer assigns as date and time ● The officer confirms the schedule
Expectations	The system stores the pick-up-schedule
Post - Conditions	A new pick-up/drop-off schedule is created.

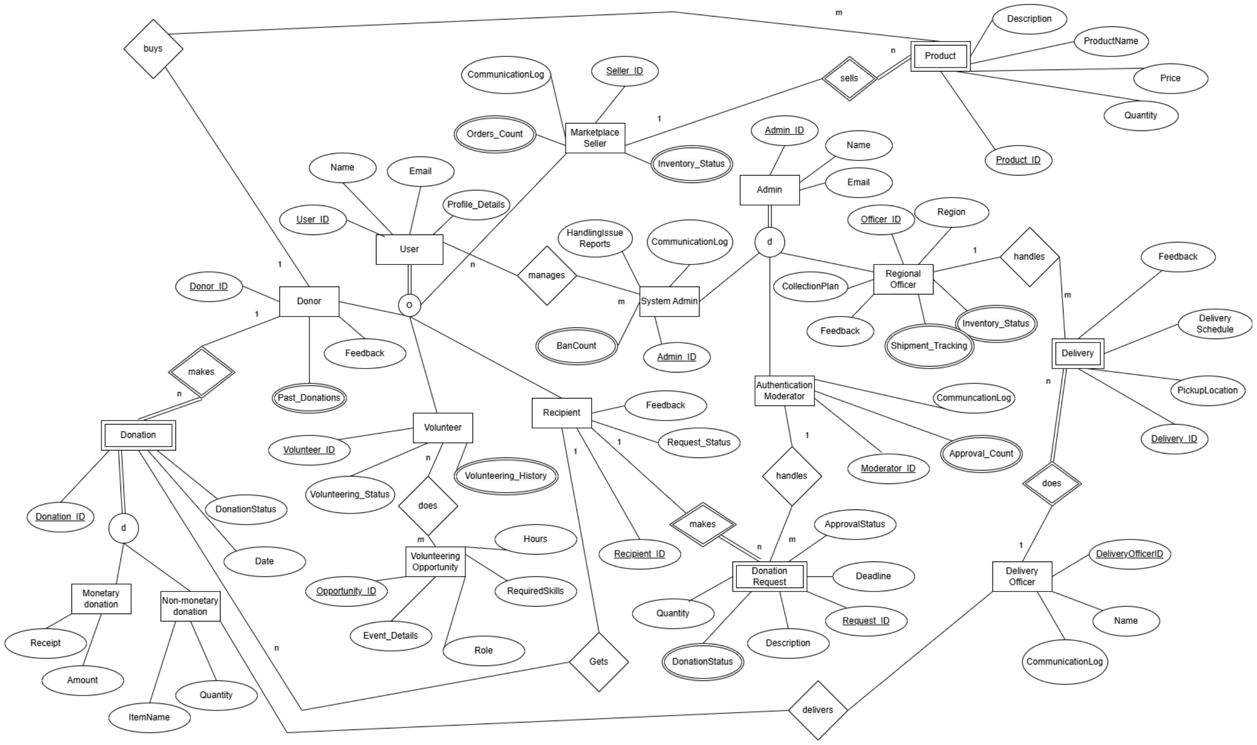
Use Case ID	30

Use Case Name	Notify Recipients
Primary Actors	Delivery Officers
Description	Delivery officer notifies recipients of scheduled deliveries
Pre-Conditions	Recipients contact details must exist in the system
Main Scenario	● When the delivery is brought to the location ● The officer must notify the recipient
Expectations	Notifications are delivered successfully
Post - Conditions	Recipients are informed of upcoming deliveries.

8. System Admin

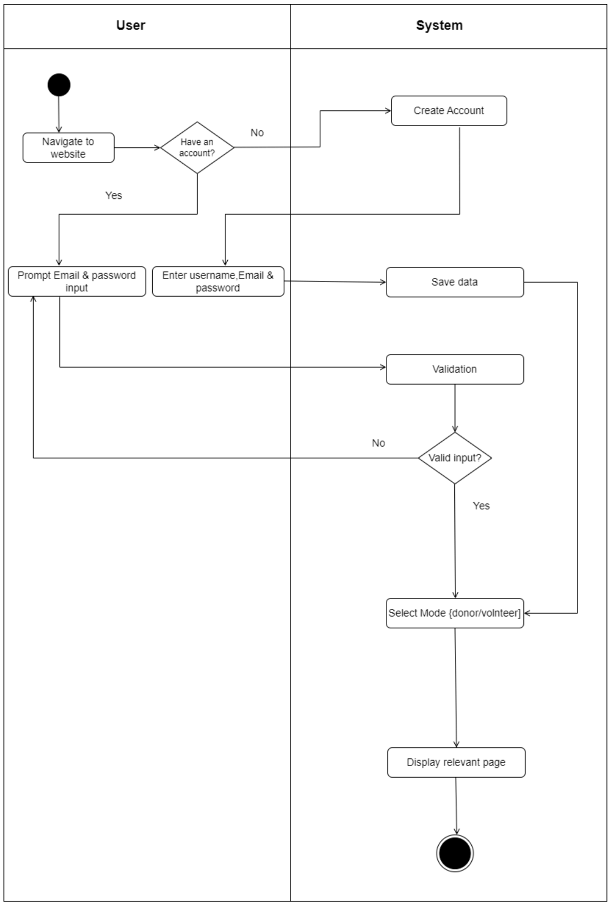
Use Case ID	31
Use Case Name	Manage System
Primary Actors	System admin
Description	Should manage overall user accounts
Pre-Conditions	Should have access as the system admin
Main Scenario	● View all the user accounts ● Do modifications them ● Delete them
Expectations	To run the system smoothly
Post - Conditions	

ER Diagram

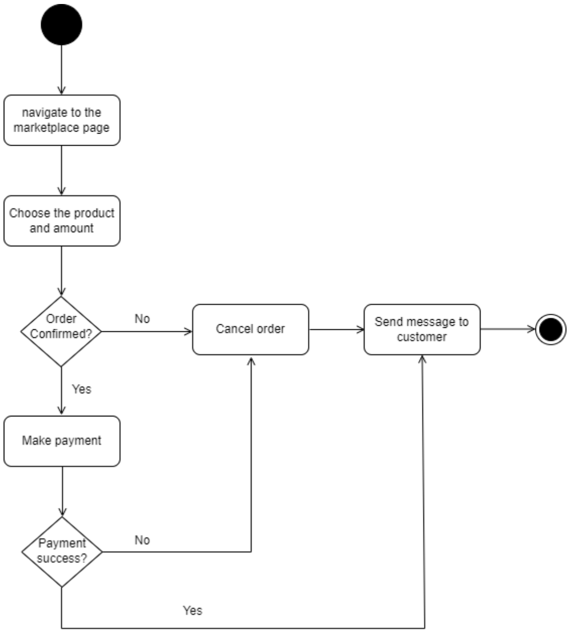

m
buys
Description
n
Product
ProductName
Seller_ID
CommunicationLog
sells
Price
Quantity
1
Orders_Count
Marketplace Seller
Admin_ID
Name
Inventory_Status
Product_ID
Name
Email
Email
Admin
Profile_Details
User_ID
HandlingIssue Reports
CommunicationLog
Officer_ID
Region
d
User
1
handles
Feedback
n
manages
Regional Officer
1
Donor_ID
CollectionPlan
m
Donor
o
m
System Admin
1
Inventory_Status
Delivery Schedule
Feedback
BanCount
Feedback
Shipment_Tracking
Delivery
Admin_ID
makes
Authentication Moderator
n
PickupLocation
Past_Donations
Feedback
n
Donation
Volunteer
Recipient
CommuncationLog
Request_Status
1
Volunteer_ID
1
1
Delivery_ID
n
Volunteering_History
Moderator_ID
Approval_Count
handles
does
Volunteering_Status
DonationStatus
does
Donation_ID
makes
ApprovalStatus
d
m
Hours
m
n
1
Volunteering Opportunity
Recipient_ID
DeliveryOfficerID
Date
Deadline
Donation Request
Delivery Officer
Monetary donation
Non-monetary donation
Opportunity_ID
RequiredSkills
Quantity
Request_ID
Name
Event_Details
Receipt
Description
n
Role
Gets
DonationStatus
CommunicationLog
Amount
Quantity
ItemName
delivers

Activity Diagrams

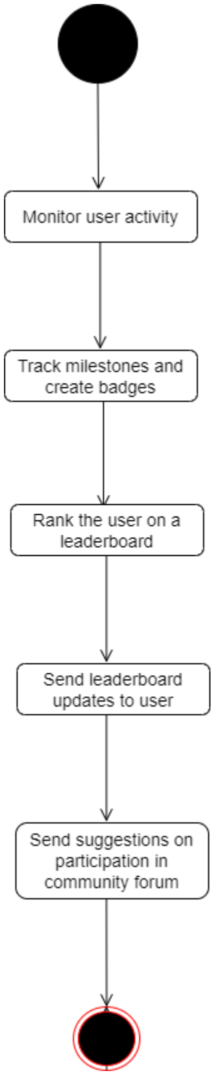
1) Sign in and Login



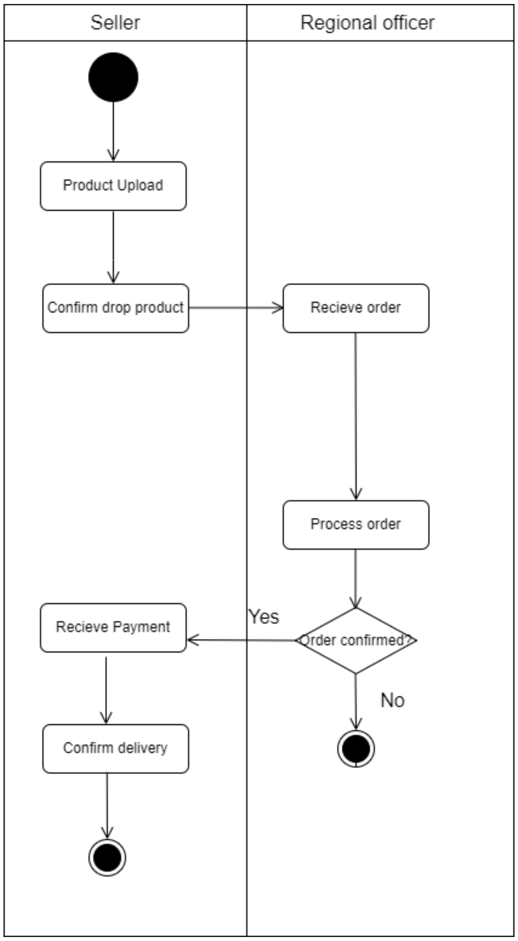
2) Buyer Process



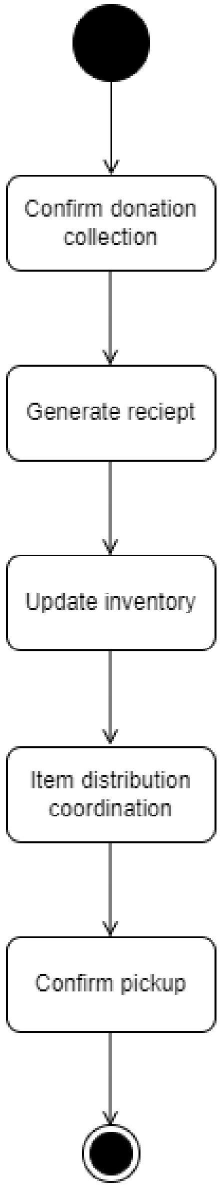
3) Gamification Process


Monitor user activity
Track milestones and create badges
Rank the user on a leaderboard
Send leaderboard updates to user
Send suggestions on participation in community forum

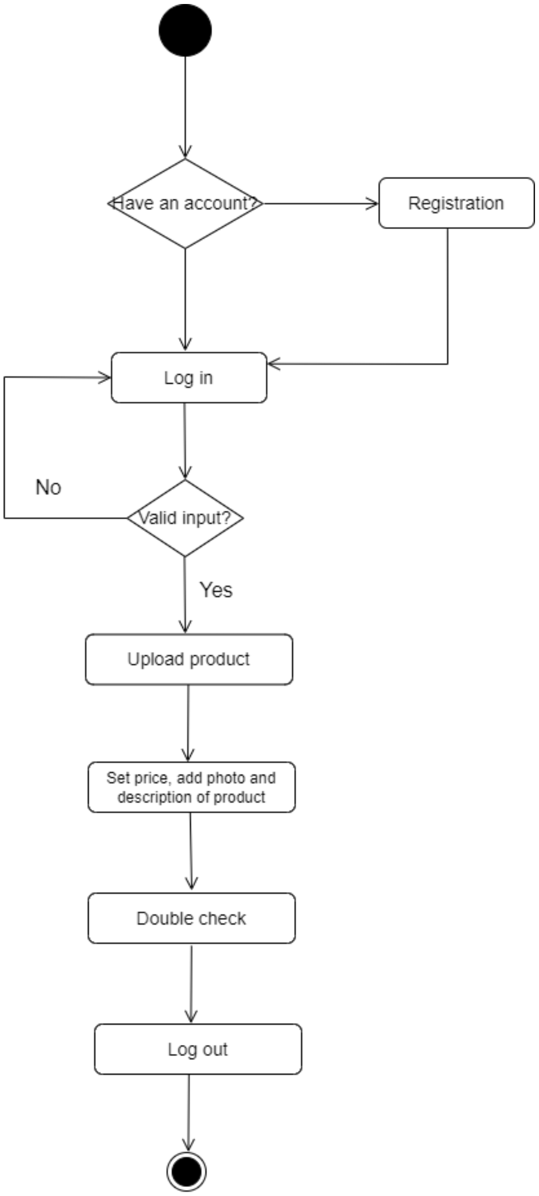
4) Delivery Process



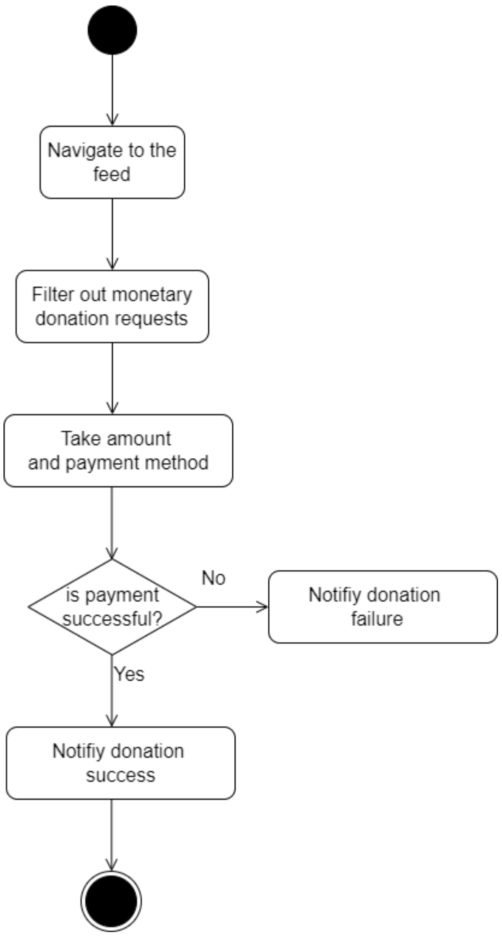
5) Inventory Management Process



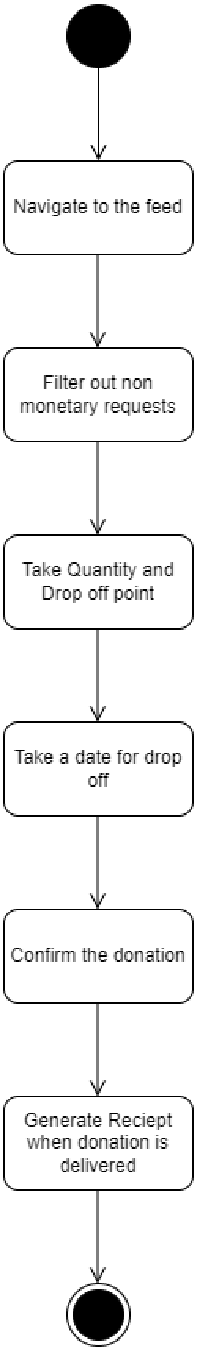
6) Marketplace Seller process



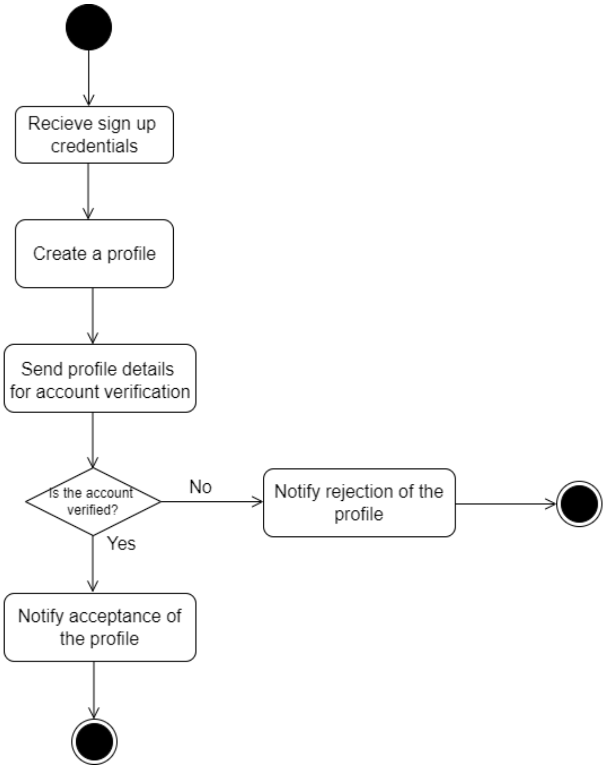
7) Monetary donation process



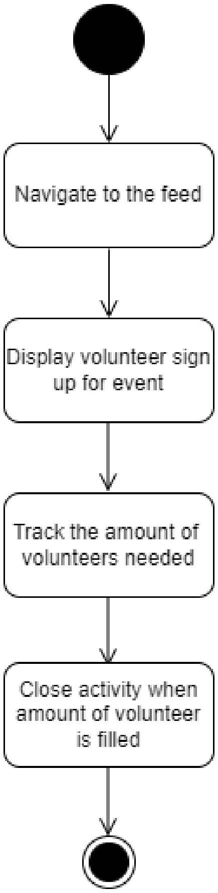
8) Non-monetary donation process


Navigate to the feed
Filter out non monetary requests
Take Quantity and Drop off point
Take a date for drop off
Confirm the donation
Generate Reciept when donation is delivered

9) Profile management and registration


Recieve sign up credentials
Create a profile
Send profile details for account verification
Is the account verified?
No
Notify rejection of the profile
Yes
Notify acceptance of the profile

10) volunteer management process


Navigate to the feed
Display volunteer sign up for event
Track the amount of volunteers needed
Close activity when amount of volunteer is filled

Current Progress for Hopefull Application

System Development Progress Concerning System Requirements

- Development of the Hopefull platform is ongoing with key features being implemented progressively across different sections.
- Functional areas such as User Registration, Donation Management (monetary and non-monetary), Profile Management, Request Creation and Approval, and Market Place have been partially developed and are undergoing testing.
- Backend functionalities such as request approval workflows, inventory management by regional officers, and delivery tracking are under active development.
- The gamification features (leaderboards, badges, milestones) and community engagement functionalities (forums) are in the initial design phase.

Estimated Completion Percentage

Total Progress Estimate

Approximately 70% of the system has been completed, with core functionalities like user management, donation management, and request handling in place.

- User Interfaces - 100% complete.
- Backend and Admin Functionality: About 70% complete.
- Market Place and Delivery Coordination: 50% complete.
- Gamification & Community Engagement: 20% complete.

Remaining Tasks / Work

Backend Development

- Completing the backend services for donation workflows, and managing non-monetary donations.
- Finalizing the integration of the inventory management system for regional officers, including real-time updates and coordination with marketplace sellers.
- Implementing the recipient feedback and impact report systems to enable donors and volunteers to view the results of their contributions.

Testing

Continuous testing for system functionality, especially for edge cases like donation cancellations, and testing of the delivery coordination system.

Gamification & Community Engagement

Implementing the leaderboard and community forums.

Toggle Button Functionality

Finalizing the backend logic for the donor/volunteer toggle button, ensuring that users can switch between "Donor Mode" and "Volunteer Mode" seamlessly, with the relevant information displayed accordingly for each mode.

Each Member’s contribution

Sudila (Donor Features): Focused on donor-related features, including donor registration, profile management, and toggling between donor/volunteer modes. Sudila is working on implementing donation workflows, such as browsing donation requests, making monetary and non-monetary donations, and managing donor dashboards.

Dulmini (Market Place): Responsible for the development and management of the Hopefull Marketplace. This includes seller registration, product management, and inventory updates. Dulmini is also handling order management and coordinating with regional officers to ensure that products are available and delivered to buyers.

Ayeshini (Recipient Features): Focused on the architecture and the recipient side of the platform, including registration and profile management. Ayeshini is developing the process for recipients to create and manage donation requests, post volunteer opportunities, and provide feedback. Ayeshini is also working on integrating the payment gateway into our application.

Sasanka (Admin Side Interfaces and Functions): Responsible for designing and implementing the admin-side features, including user profile management, handling issue reports, and providing technical support. Sasanka is also overseeing the development of the system’s backend to manage authentication, account approvals, and communication with users.