

USE CASE DIAGRAM AND USE CASE DESCRIPTIONS

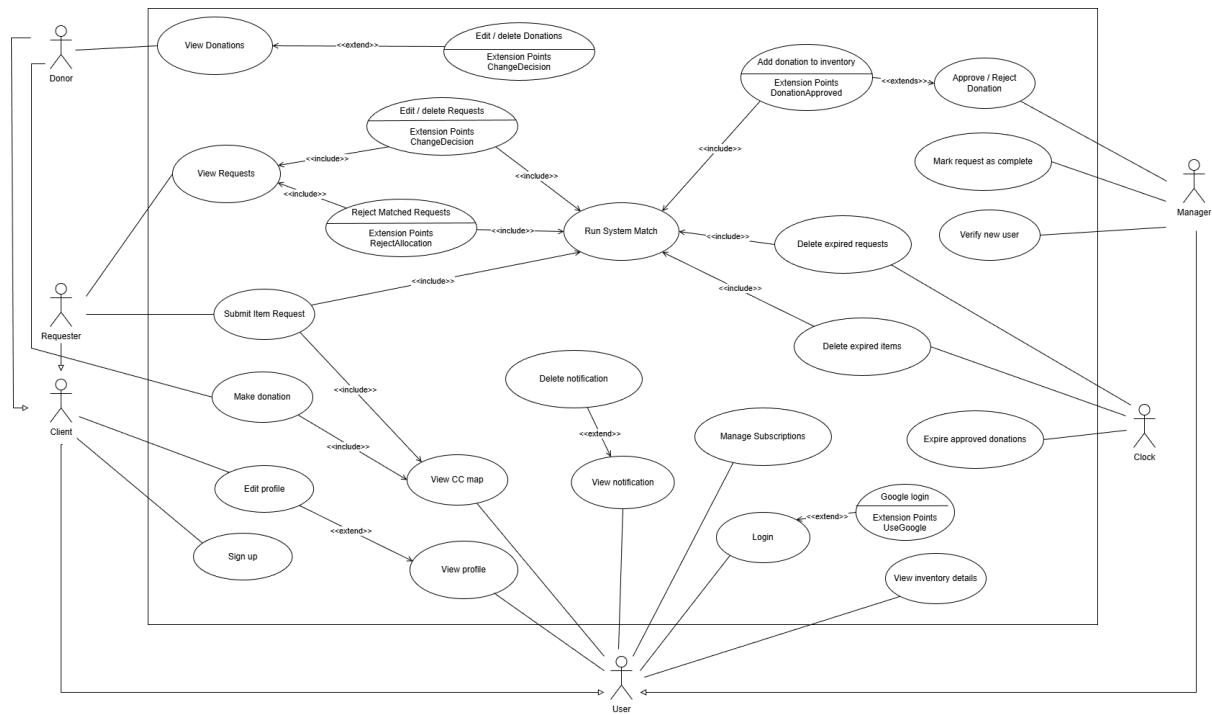
TABLE OF CONTENTS

CONTENTS		PG. NO.
A. USE CASE DIAGRAM		3
B. USE CASE DESCRIPTIONS		
UC 1.X.X	USER	
UC 1.1.1	Login	4
UC 1.1.2	Google Login	5
UC 1.2.1	View CC Map	6
UC 1.3.1	View Notifications	7
UC 1.3.2	Delete Notifications	8
UC 1.4.1	View Profile	9
UC 1.5.1	Manage Subscriptions	10
UC 1.6.1	View Inventory Details	11
UC 2.X.X	CLIENT	
UC 2.1.1	Edit Profile	12
UC 2.2.1	Sign-up	13
UC 3.X.X	REQUESTER	
UC 3.1.1	Submit Item Request	14 - 15
UC 3.1.2	Edit/Delete Request	16
UC 3.1.3	Reject Matched Requests	17
UC 3.1.4	View Requests	18
UC 4.X.X	DONOR	
UC 4.1.1	Make Donation	19
UC 4.1.2	Edit/Delete Donations	21

UC 5.X.X	MANAGER	
UC 5.1.1	Approve/Reject Donation	22
UC 5.1.2	Add Donation to Inventory	23
UC 5.2.1	Mark Request as Completed	24
UC 5.3.1	Verify New User	25
UC 6.X.X	CLOCK	
UC 6.1.1	Delete Expired Requests	26
UC 6.1.2	Delete Expired Items	27
UC 6.1.3	Expire Approved Donations	28
UC 7.X.X	SYSTEM	
UC 7.1.1	Run System Match	29

A. USE CASE DIAGRAM

If the image is unclear, please refer to the raw .png file uploaded with the document.



B. USE CASE DESCRIPTIONS

UC 1.1.1 - Login

Use Case ID:	UC 1.1.1		
Use Case Name:	Login		
Created By:	Aaron	Last Updated By:	
Date Created:	03.11.25	Date Last Updated:	

Actor:	User
Description:	Standard authentication flow (email/password or third-party); grants access tokens and loads user profile and role.
Preconditions:	1. User has an account.
Postconditions:	1. Session established; role determines UI features.
Priority:	High
Frequency of Use:	Several times per week.
Flow of Events:	<ol style="list-style-type: none">1. User enters credentials2. System authenticates:3. If success:<ol style="list-style-type: none">a. Redirect to appropriate landing.
Alternative Flows:	1.2.1.AC.1: If authentication, unsuccessful, try again
Exceptions:	-
Includes:	-
Special Requirements:	<ol style="list-style-type: none">1. Brute-force protections2. Session timeout.
Assumptions:	-
Notes and Issues:	-

UC 1.1.2 - Google Login

Use Case ID:	UC 1.1.2		
Use Case Name:	Google Login		
Created By:	Aaron	Last Updated By:	
Date Created:	03.11.25	Date Last Updated:	

Actor:	User
Description:	Optional OAuth sign-in via Google; on first sign-in a new user account may be created and queued for verification (if necessary).
Preconditions:	1. User selects Google login option.
Postconditions:	1. Authenticated session established; account created/linked if first time.
Priority:	High
Frequency of Use:	Several times per week.
Flow of Events:	<ol style="list-style-type: none"> 1. User selects Google 2. OAuth flow 3. System receives assertion 4. Create or link account 5. Login.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Extends:	UC 1.1.1 - Login
Special Requirements:	1. OAuth token validation; map external scopes to roles.
Assumptions:	-
Notes and Issues:	-

UC 1.2.1 – View CC Map

Use Case ID:	1.2.1		
Use Case Name:	View CC Map		
Created By:	Aaron	Last Updated By:	Aaron
Date Created:	30.08.25	Date Last Updated:	03.11.25

Actor:	User
Description:	User views interactive map with CC anchor points, sees CC details (name, address, fulfillment rate) and severe-shortage markers; search and subscribe features available.
Preconditions:	1. User is logged in.
Postconditions:	Map rendered; user can select CC to view details or subscribe.
Priority:	High
Frequency of Use:	Very frequent (daily)
Flow of Events:	<ol style="list-style-type: none"> 1. User opens Map 2. System fetches CC dataset 3. Displays markers with status overlays (red for severe shortage) 4. User selects CC.
Alternative Flows:	<u>1.2.AC.1:</u> If no relevant listings available, System displays appropriate message
Exceptions:	1.2.2.EX.1: System error prevents map from being temporarily displayed.
Includes:	None
Special Requirements:	Real-time updates, search bar, subscribe UI. Privacy for details per role
Assumptions:	Data API is available
Notes and Issues:	-

UC 1.3.1 – View Notifications

Use Case ID:	1.3.1		
Use Case Name:	View Notifications		
Created By:	Aaron	Last Updated By:	Aaron
Date Created:	30.08.25	Date Last Updated:	03.11.25

Actor:	User
Description:	User views the list of notifications relevant to them (provisional matches, approvals, expiry notices); may act on them (Reject collection, navigate to request/donation).
Preconditions:	<ol style="list-style-type: none"> 1. User is logged in. 2. Notifications are enabled
Postconditions:	<ol style="list-style-type: none"> 1. Notifications displayed
Priority:	High
Frequency of Use:	Many times a week
Flow of Events:	<ol style="list-style-type: none"> 1. User opens notifications 2. System shows sorted list 3. User selects one to view details or act.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	Timely delivery; ensures actions are idempotent.
Assumptions:	-
Notes and Issues:	-

UC 1.3.2 – Delete Notifications

Use Case ID:	1.3.2		
Use Case Name:	Delete Notifications		
Created By:	Aaron	Last Updated By:	Aaron
Date Created:	30.08.25	Date Last Updated:	03.11.25

Actor:	User
Description:	User can dismiss or delete notifications (in-app); deletion affects only the user's notification view, not underlying events.
Preconditions:	<ul style="list-style-type: none"> 3. User is logged in. 4. Notifications are enabled
Postconditions:	<ul style="list-style-type: none"> 2. Notification removed from user inbox 3. Event logs retained for audit.
Priority:	High
Frequency of Use:	Many times a week
Flow of Events:	<ul style="list-style-type: none"> 4. User opens notification center 5. Deletes chosen notification 6. System records deletion for that user.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	Soft delete per user; server retains canonical event.
Assumptions:	-
Notes and Issues:	-

UC 1.4.1 – View Profile

Use Case ID:	1.4.1		
Use Case Name:	View Profile		
Created By:	Aaron	Last Updated By:	Aaron
Date Created:	30.08.25	Date Last Updated:	03.11.25

Actor:	User
Description:	User views their profile and related account metadata; from here they may navigate to edit profile.
Preconditions:	1. User is logged in.
Postconditions:	Profile displayed.
Priority:	High
Frequency of Use:	Not very frequent
Flow of Events:	<ol style="list-style-type: none">1. User opens profile view2. System renders stored info3. User may choose Edit Profile.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

UC 1.5.1 – Manage Subscriptions

Use Case ID:	1.5.1		
Use Case Name:	Manage Subscriptions		
Created By:	Aaron	Last Updated By:	Aaron
Date Created:	30.08.25	Date Last Updated:	03.11.25

Actor:	User
Description:	User subscribe to a CC for shortage broadcast.
Preconditions:	1. User is logged in.
Postconditions:	2. User is subscribed to the CCs of his choice.
Priority:	High
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none">1. User views subscriptions2. User can choose to unsubscribe to a CC3. From a drop-down, user can choose a CC to subscribe to
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

UC 1.6.1 – View Inventory Details

Use Case ID:	1.6.1		
Use Case Name:	View Inventory Details		
Created By:	Aaron	Last Updated By:	Aaron
Date Created:	30.08.25	Date Last Updated:	03.11.25

Actor:	User
Description:	All users will see overview of request fulfilment rates for all CCs. Managers can choose one CC to see detailed information.
Preconditions:	1. Navigated to inventory page
Postconditions:	Item details displayed.
Priority:	High
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> 1. All users will see donation and request quantities, fulfilled request quantities and fulfilment rate. 2. Only managers can click on a CC to view the breakdown of each item
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	Hide sensitive info for donors (requester identities).
Assumptions:	-
Notes and Issues:	-

UC 2.1.1 – Edit Profile

Use Case ID:	2.1.1		
Use Case Name:	Edit Profile		
Created By:	Aaron	Last Updated By:	Aaron
Date Created:	30.08.25	Date Last Updated:	03.11.25

Actor:	Client
Description:	Client updates his personal profile fields (contact, address, income data subject to verification). Changes affect eligibility and communication settings.
Preconditions:	2. Client is logged in.
Postconditions:	Profile updated and revalidated if income/eligibility changed; audit recorded.
Priority:	High
Frequency of Use:	Not very frequent
Flow of Events:	<ol style="list-style-type: none"> 1. Client edits allowed fields 2. System validates 3. Persists 4. Triggers re-evaluation of eligibility.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	Sensitive fields require confirmation/audit.
Assumptions:	-
Notes and Issues:	-

UC 2.2.1 - Sign-up

Use Case ID:	UC 2.2.1		
Use Case Name:	Sign-up		
Created By:	Aaron	Last Updated By:	
Date Created:	05.11.25	Date Last Updated:	

Actor:	Client
Description:	A new user creates an account by providing required details (email, password, name, income). The system validates input, prevents duplicates, persists the account as Pending..
Preconditions:	1. Client does not have an account.
Postconditions:	1. Account created after manager verifies (refer UC 5.3.1)
Priority:	High
Frequency of Use:	Several times per week.
Flow of Events:	<ol style="list-style-type: none"> 1. User opens sign-up page 2. User enters required credentials 3. System validates inputs (format, password strength) and checks for duplicates 4. If valid details: <ol style="list-style-type: none"> a. System notifies CC Manager and sets Sign-up status to Pending 5. System displays confirmation UI instructing user to verify email or await approval.
Alternative Flows:	1.1.3.AC.1: If authentication, unsuccessful, try again
Exceptions:	1.1.3 EX.1: If account already exists with provided email, account not displayed, 'account exists' message provided
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

UC 3.1.1 – Submit Item Request

Use Case ID:	3.1.1		
Use Case Name:	Submit Item Request		
Created By:	Aaron	Last Updated By:	Aaron
Date Created:	30.08.25	Date Last Updated:	03.11.25

Actor:	Requester
Description:	Requesters from approved low-income accounts can submit requests for items. When available stock exists in the selected CC (i.e., after InventoryAdded), the system will attempt to create a provisional match. If stock is not available, the request is marked Pending and queued for future matching.
Preconditions:	<ol style="list-style-type: none"> 1. Requester is a registered, logged-in user flagged as low-income. 2. Requester has selected a CC.
Postconditions:	<ol style="list-style-type: none"> 1. If matching stock exists at the time matching runs (after InventoryAdded): request → Matched until Rejected/Expired. 2. If no stock → Pending and queued.
Priority:	High
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> 1. Requester selects CC from the map and opens Request form. 2. Requester enters Name, Category, Quantity. 3. System validates inputs. 4. System stores the request with Status = Pending and timestamp. 5. Matching engine will evaluate Pending requests and create reservation records when inventory permits and notifies requester once fully fulfilled

Alternative Flows:	3.1.1.AC.1: Requester cancels before submission → request not created.
Exceptions:	3.1.1.EX.1: Invalid data (negative quantity, unsupported category) → reject submission with error.
Includes:	UC 1.2.1 - View CC Map UC 7.1.1 - Run System Match
Special Requirements:	<ul style="list-style-type: none"> 3. Input validation. 4. Requests are queued with timestamps for matching priority. 5. Matching/reservation operations must be atomic and idempotent; concurrency control required to prevent double allocation.
Assumptions:	-
Notes and Issues:	Matching is only triggered by InventoryAdded events or periodic request expiry and items expiry runs, not by donation approval.

UC 3.1.2 – Edit/Delete Requests

Use Case ID:	3.1.2		
Use Case Name:	Edit/Delete Requests		
Created By:	Aaron	Last Updated By:	Aaron
Date Created:	30.08.25	Date Last Updated:	03.11.25

Actor:	Requester
Description:	Requester edits or deletes their own requests while requests are still editable (Status = Pending). Editing re-queues the request for matching. Deletion releases any partial reservations.
Preconditions:	<ol style="list-style-type: none"> 1. User is logged in 2. User has a request with Status $\in \{\text{Pending}\}$.
Postconditions:	<ol style="list-style-type: none"> 1. Request updated with new timestamp or removed; matching engine re-runs for edited requests.
Priority:	High
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> 1. Requester selects request 2. Edits allowed fields (name, category, qty) or deletes 3. System validates 4. Saves and requeues.
Alternative Flows:	-
Exceptions:	3.1.2.EX.1: Cannot edit immutable fields (requester ID, CC); cannot edit once Matched/Completed/Expired.
Includes:	UC 7.1.1 - Run System Match
Extends	UC 3.1.4 - <i>View Requests</i> .
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

UC 3.1.3 – Reject Matched Requests

Use Case ID:	3.1.3		
Use Case Name:	Reject Matched Requests		
Created By:	Aaron	Last Updated By:	Aaron
Date Created:	30.08.25	Date Last Updated:	03.11.25

Actor:	Requester
Description:	Requester can choose to reject Matched requests.
Preconditions:	<ol style="list-style-type: none"> 1. User is logged in 2. User has a request with Status $\in \{\text{Matched}\}$.
Postconditions:	<ol style="list-style-type: none"> 1. Request deleted
Priority:	High
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> 1. Requester selects request 2. Reject
Alternative Flows:	-
Exceptions:	-
Includes:	UC 7.1.1 - Run System Match
Extends	UC 3.1.4 - View Requests
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

UC 3.1.4 – View Requests

Use Case ID:	3.3.1		
Use Case Name:	View Requests		
Created By:	Aaron	Last Updated By:	Aaron
Date Created:	30.08.25	Date Last Updated:	03.11.25

Actor:	Requester
Description:	Requester views their own requests and CC pending requests (depending on role/permission). List is filterable by status and category and shows allocation/fulfillment info.
Preconditions:	1. Requester is logged in
Postconditions:	1. Request list displayed 2. Requester can act (edit/delete) on eligible requests.
Priority:	High
Frequency of Use:	Not very frequent
Flow of Events:	1. User opens Requests 2. System fetches user's requests 3. Displays requests.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Extends:	-
Special Requirements:	Real-time status updates; privacy of requester identities when shown to donors.
Assumptions:	-
Notes and Issues:	-

UC 4.1.1 – Make Donation

Use Case ID:	4.1.1		
Use Case Name:	Make Donation		
Created By:	Aaron	Last Updated By:	Aaron
Date Created:	30.08.25	Date Last Updated:	03.11.25

Actor:	Donor
Description:	Donor submits an application describing goods. This creates a Submitted donation application that a CC manager will review. This flow ends at the Approval step; no inventory is created here.
Preconditions:	<ol style="list-style-type: none"> 2. Donor is logged in. 3. A CC hub has been selected.
Postconditions:	Donation application persisted with Status = Submitted.
Priority:	High
Frequency of Use:	Very frequent (daily)
Flow of Events:	<ol style="list-style-type: none"> 1. Donor selects CC and clicks “Donate.” 2. Donor completes form (donation category, donation item, quantity, upload image of donation and fill in expiry date if donating food or drinks). 3. Manager validates images (check if donation is in good condition). 4. If donation is valid, donors are notified to go to the cc for physical handover.

Alternative Flows:	1. Manager chooses not to accept the donation, donation is deleted and donor is notified
Exceptions:	-
Includes:	UC 7.1.1 - Run System Match UC 1.2.1 - View CC Map
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

UC 4.1.2 – Edit/Delete Donation

Use Case ID:	4.1.2		
Use Case Name:	Edit/Delete Donation		
Created By:	Aaron	Last Updated By:	Aaron
Date Created:	30.08.25	Date Last Updated:	03.11.25

Actor:	Donor
Description:	Donor edits or deletes their pending donation applications prior to final intake/approval; edits on approved applications reset status per business rule.
Preconditions:	1. Donor is logged in; donation application exists and is in editable state (Pending).
Postconditions:	Application updated or deleted; status transitions as required (e.g., edit → Submitted).
Priority:	High
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> 1. Donor navigates to their donation 2. Chooses Edit/Delete 3. Changes saved or application removed 4. System logs change and notifies manager if needed.
Alternative Flows:	-
Exceptions:	If the application status is already Approved//Added, editing is restricted.
Includes:	-
Extends	UC 4.1.2 - <i>View Donations</i>
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

UC 5.1.1 – Approve/ Reject Donation

Use Case ID:	5.1.1		
Use Case Name:	Approve/Reject Donation		
Created By:	Aaron	Last Updated By:	Aaron
Date Created:	30.08.25	Date Last Updated:	03.11.25

Actor:	Manager
Description:	Manager reviews incoming donation applications (images, expiry, donor info) and approves or rejects them for physical drop-off.
Preconditions:	Donation application exists with required image(s).
Postconditions:	Application status set to Approved or Rejected; donor notified. Approved flows proceed to check-in and Intake steps.
Priority:	High
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> 1. Manager views application 2. Inspects images/expiry 3. Chooses Approve/Reject 4. System records decision and notifies donor.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	Decisions logged with timestamp and reviewer ID.
Assumptions:	-
Notes and Issues:	-

UC 5.1.2 – Add Donation to Inventory

Use Case ID:	5.1.2		
Use Case Name:	Add Donation to Inventory		
Created By:	Aaron	Last Updated By:	Aaron
Date Created:	30.08.25	Date Last Updated:	03.11.25

Actor:	Manager
Description:	CC manager records intake outcomes when donor delivers goods and, upon successful intake, adds accepted goods to CC inventory (creating InventoryAdded events).
Preconditions:	Donation application status = Approved and donor has arrived/check in.
Postconditions:	Inventory record created/updated; InventoryAdded event emitted; matching triggered.
Priority:	High
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> 1. Manager verifies goods and images 2. Marks Added/Rejected 3. If Added, create item entry with batch/expiry metadata 4. Commit and emit event.
Alternative Flows:	-
Exceptions:	-
Includes:	UC 7.1.1 Run System Match
Extends	UC 5.1.1 Approve/Reject Donation
Special Requirements:	Audit trail, idempotency, timestamping.
Assumptions:	-
Notes and Issues:	-

UC 5.2.1 – Mark Request as Completed

Use Case ID:	5.2.1		
Use Case Name:	Mark Request as Completed		
Created By:	Aaron	Last Updated By:	Aaron
Date Created:	30.08.25	Date Last Updated:	03.11.25

Actor:	Manager
Description:	Manager confirms physical collection by a beneficiary and marks the associated request as Completed; inventory is decremented accordingly.
Preconditions:	Request status = Matched/Accepted; beneficiary present for collection.
Postconditions:	Request → Completed; inventory final decrement occurs; audit recorded.
Priority:	High
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> 1. Manager verifies beneficiary identity and match 2. Confirms collection in system 3. System decrements inventory and updates request state
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	Logging and ability to undo within short window if mistaken.
Assumptions:	-
Notes and Issues:	-

UC 5.3.1 – Verify New User

Use Case ID:	5.3.1		
Use Case Name:	Verify New User		
Created By:	Aaron	Last Updated By:	Aaron
Date Created:	30.08.25	Date Last Updated:	03.11.25

Actor:	Manager
Description:	Manager reviews new user sign-ups and approves or rejects accounts (beneficiary, donor, volunteer). Approval grants role-specific access.
Preconditions:	New user registration exists with required info.
Postconditions:	Account activated (or rejected) and user notified; approvals logged.
Priority:	High
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> 1. Manager views application 2. Checks documents/info 3. Approve/Reject 4. System updates account status 5. Notify user.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	Access control changes applied immediately; audit trail.
Assumptions:	-
Notes and Issues:	-

UC 6.1.1 – Delete Expired Requests

Use Case ID:	6.1.1		
Use Case Name:	Delete Expired Requests		
Created By:	Aaron	Last Updated By:	Aaron
Date Created:	30.08.25	Date Last Updated:	03.11.25

Actor:	Clock (scheduled task), System
Description:	Scheduled process that expires requests not collected within 48 hours after Matched status; reservations released and requests marked Expired.
Preconditions:	Requests with expired Matched requests.
Postconditions:	Expired requests updated, reservations released, notifications sent.
Priority:	High
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> 1. Cron job runs 2. Identifies expired Matched Requests 3. Mark requests Expired 4. Release reserved inventory 5. Notify requesters and managers.
Alternative Flows:	-
Exceptions:	-
Includes:	UC 7.1.1 - Run System Match
Special Requirements:	Ensure idempotency and proper ordering with matching engine; log actions.
Assumptions:	-
Notes and Issues:	-

UC 6.1.2 – Delete Expired Items

Use Case ID:	6.1.2		
Use Case Name:	Delete Expired Items		
Created By:	Aaron	Last Updated By:	Aaron
Date Created:	30.08.25	Date Last Updated:	03.11.25

Actor:	Clock (scheduled task), System
Description:	Scheduled cleanup that removes or marks inventory that has passed expiry (by batch expiry date), adjusting available/reserved quantities and notifying relevant parties; may trigger reallocation/matching.
Preconditions:	Inventory batches with expiry date \leq today.
Postconditions:	Expired items marked/unavailable and removed from matching; reservations involving expired items are released; notifications sent.
Priority:	High
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> 1. Scheduler scans batches 2. Mark expired 3. Decrement/unavailable quantities 4. Release reservations 5. Invoke matching to reallocate if needed.
Alternative Flows:	-
Exceptions:	-
Includes:	UC 7.1.1 - Run System Match
Special Requirements:	Regulatory handling for disposed goods; audit trail.
Assumptions:	-
Notes and Issues:	-

UC 6.1.3 – Expire Approved Donations

Use Case ID:	6.1.3		
Use Case Name:	Expire Approved Donations		
Created By:	Aaron	Last Updated By:	Aaron
Date Created:	30.08.25	Date Last Updated:	03.11.25

Actor:	Clock (scheduled task), System
Description:	Scheduled job to expire donation applications or check-in reservations if donors do not complete physical handover donation 48 hours after Donation is approved .
Preconditions:	Donation application status = Approved but not Added/Rejected within allowed window.
Postconditions:	Application deleted; donor notified.
Priority:	High
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> 1. Scheduler identifies stale Approved applications 2. Transition status 3. Notify donor and manager.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	Business window configurable; logging.
Assumptions:	-
Notes and Issues:	-

UC 7.1.1 – Run System Match

Use Case ID:	7.1.1		
Use Case Name:	Run System Match		
Created By:	Aaron	Last Updated By:	Aaron
Date Created:	30.08.25	Date Last Updated:	03.11.25

Actor:	System (Matching Engine)
Description:	Automatic process that allocates available inventory to pending requests, creates reservations, and triggers notifications.
Preconditions:	InventoryAdded events or pending requests present; configured matching window and policies available.
Postconditions:	Reservations created/updated; inventory quantities adjusted; notifications created.
Priority:	High
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> 1. On trigger (InventoryAdded, request submit, request expiry, item expiry, match request rejected and request deletion) 2. Fetch eligible requests based on type of items 3. Match per policy (FIFO) 4. Create reservations atomically 5. Update inventory 6. Emit notifications.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	Atomicity, idempotency, concurrency control, audit logs.
Assumptions:	-
Notes and Issues:	-

