ResQ

Phase 1

Team 3

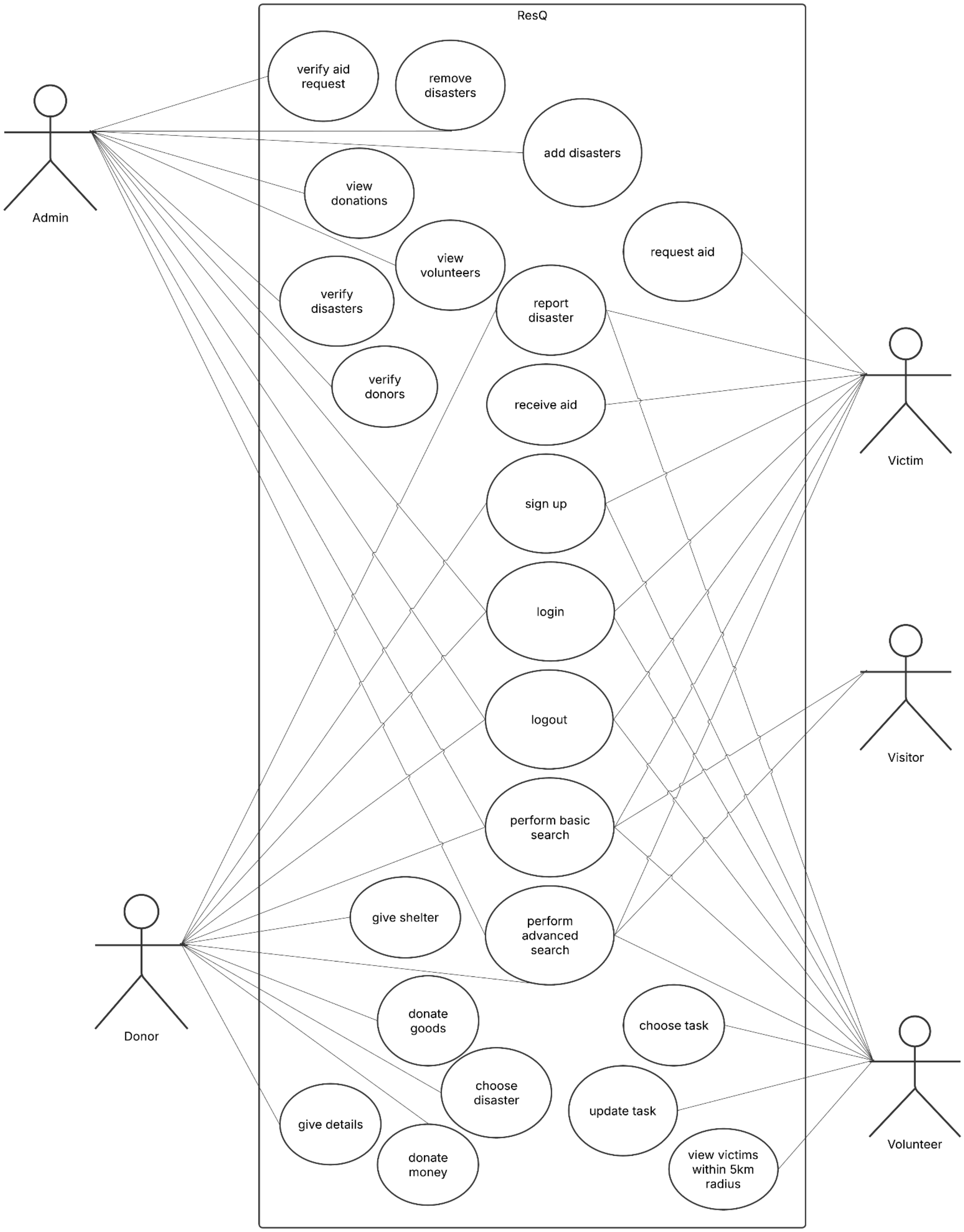
Member Name	Member Roll #	Primary Responsibility
Muhammad Zaigham Asif	23L-3058	UCD, ACD, NFR, UC-1, UC-2, UC-3
Muhammad Abdullah Latif	23L-3080	UCD, ACD, NFR, UC-4, UC-5, UC-6, UC-7
Muhammad Shuja Jamal	23L-3024	UCD, ACD , NFR, UC-8, UC-9, UC-10, UC-11
Abu Bakar Afzal	23L-3062	UCD, ACD , NFR, UC-12, UC-13, UC-14, UC-15
Walija Fatima	23L-3055	UCD, ACD , NFR, UC-16, UC-17, UC-18, UC-19
Fatima Tuz Zahra	23L-3099	UCD, ACD , NFR, UC-20, UC-21, UC-22, UC-23

Table of Contents

https://lucid.app/users/registerOrLogin/free?showLogin=false&invitationId=inv_be2467d5-e345-4345-98f0-c002a2885c9f&productOpt=chart&invitationType=documentAcceptance&returnUrlOverride=%2Flucidchart%2Fb9d6d66d-7cca-41c3-a7ad-ca4ed35bee55%2Fedit%3FinvitationId%3Dinv_be2467d5-e345-4345-98f0-c002a2885c9f%26page%3D0_0

https://lucid.app/lucidchart/b912cedc-0a3f-4b75-9fd2-134a9c3de23c/edit?invitationId=inv_bb106055-cfb2-4d9c-8c32-011e3ce23506&page=0_0#

1. UCD


ResQ
Admin
Victim
Visitor
Donor
Volunteer
verify aid request
remove disasters
add disasters
view donations
view volunteers
request aid
verify disasters
report disaster
verify donors
receive aid
sign up
login
logout
perform basic search
give shelter
perform advanced search
donate goods
choose task
choose disaster
update task
give details
donate money
view victims within 5km radius

2. Use Cases

2.1 Admin

<table>
<tr><td><b>Identifier</b></td><td colspan="2">UC-1</td></tr>
<tr><td><b>Name</b></td><td colspan="2">Add Disaster</td></tr>
<tr><td><b>Summary</b></td><td colspan="2">Create a new disaster entry in the system so it can be managed.</td></tr>
<tr><td><b>Priority</b></td><td colspan="2">High</td></tr>
<tr><td><b>Actor(s)</b></td><td colspan="2">Administrator</td></tr>
<tr><td><b>Pre-condition(s)</b></td><td colspan="2">The Administrator is logged into the system.The system is operational and connected to the disaster database.</td></tr>
<tr><td><b>Post-condition(s)</b></td><td colspan="2">A new disaster record is created and stored in the system. The system acknowledges successful addition of the disaster.</td></tr>
<tr><td colspan="3"><b>Typical Course of Action</b></td></tr>
<tr><td><b>S#</b></td><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>1</b></td><td>Navigates to the “Add Disaster” section.</td><td></td></tr>
<tr><td><b>2</b></td><td></td><td>Displays a form requesting disaster information and validates the data.</td></tr>
<tr><td><b>3</b></td><td>Fills the details and click the “Submit” button.</td><td></td></tr>
<tr><td><b>4</b></td><td></td><td>Creates and saves a new disaster entry in the database and prompts message.</td></tr>
<tr><td colspan="3"><b>Alternate Course of Action(Invalid Data)</b></td></tr>
<tr><td><b>S#</b></td><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>3</b></td><td></td><td>System prompts an error and requests corrections.</td></tr>
<tr><td colspan="3"><b>Goes back to Step 1</b></td></tr>
</table>

Identifier	UC-2
Name	Remove Disaster
Summary	Remove a disaster entry from the system.
Priority	Medium
Actor(s)	Administrator
Pre-condition(s)	The Administrator is logged into the system. A disaster record already exists in the system.
Post-condition(s)	The specified disaster record is removed. Users cannot perform new actions on that disaster if it is already removed.

Typical Course of Action

S#	Actor Action	System Response
1	Navigates to the "Remove Disasters" section.	
2		Displays the disaster details.
3	Selects the target disaster from the list and Click the "Remove" button.	
4		Asks for confirmation.
5	Confirms the removal.	
6		Deletes the disaster record and prompt success message.

Alternate Course of Action(Associated Data Found)

S#	Actor Action	System Response
3		System must restrict removal until those associated items are present.

Goes back to Step 3

Identifier	UC-3		
Name	Verify Disaster		
Summary	Review newly reported incidents/disasters and approve them for inclusion in the system.		
Priority	High		
Actor(s)	Administrator		
Pre-condition(s)	Administrator is logged into the system.One or more 'unverified' disaster reports exist.		
Post-condition(s)	The newly reported disaster is either verified and added to the main disaster list or can be rejected.		
Typical Course of Action			
S#	**Actor Action**	**System Response**	
1	Accesses the “Pending Incidents” button.		
2		Displays full details of the reports.	
3	Selects an incident to review.		
5	If verified.		
6		System adds it as an official disaster entry.	
7	If rejected		
8		System marks the report as invalid.	
Alternate Course of Action(Already Exist Disaster)			
S#	**Actor Action**	**System Response**	
6		System prompts error message.	
Goes back to Step 1			

Identifier	UC-4
Name	View Volunteers

Summary	See a list of all registered volunteers, their roles, and assigned tasks.
Priority	Medium
Actor(s)	Administrator
Pre-condition(s)	The Administrator is logged into the system. Volunteers have already registered in the system.
Post-condition(s)	A comprehensive list of volunteers is displayed.

Typical Course of Action

S#	Actor Action	System Response
1	Goes to the ‘View Volunteer' section.	
2		Displays all volunteers.
3	Reviews volunteer details.	

Alternate Course of Action(No volunteer exist)

S#	Actor Action	System Response
2		Displays a 'No Volunteers Found' message.

Goes back to Step 1

Identifier	UC-5
Name	View Donations
Summary	Review incoming donations and track how they are allocated to disasters.
Priority	Medium
Actor(s)	Administrator
Pre-condition(s)	The Administrator is logged into the system. Donations exist in the system.
Post-condition(s)	Administrator can see donation details and track allocation status.

Typical Course of Action

S#	Actor Action	System Response
1	Navigates to the 'Donations' section.	

2		Displays all the donations.
Alternate Course of Action(No donation Exist)		
S#	**Actor Action**	**System Response**
2		Displays a “No Donations” message.
Goes back to Step 1		

Identifier	UC-6		
Name	Verify Donor		
Summary	The Admin verifies a donor's details before approving their ability to donate.		
Priority	High		
Actor(s)	Administrator		
Pre-condition(s)	Admin logs in the system and finds some pending donor verification requests.		
Post-condition(s)	The donor is verified by the admin and approved to make donations.		
Typical Course of Action			
S#	**Actor Action**	**System Response**	
1	Navigates to the "Verify Donors" section.		
2		Displays a list of pending donor verification requests.	
3	Selects a donor from the list.		
4		Displays the donor's personal information.	
5	Reviews the provided information.		
6		Allows the Admin to approve or reject the donor.	
7	Clicks "Approve" if the donor's details are valid.		
8		Marks the donor as verified and grants access to donation features.	
9	Clicks "Reject" if details are invalid.		

<table>
<tr><td colspan="2"><b>Identifier</b></td><td colspan="2">UC-6</td></tr>
<tr><td colspan="2"><b>Name</b></td><td colspan="2">Verify Donor</td></tr>
<tr><td colspan="2"><b>Summary</b></td><td colspan="2">The Admin verifies a donor's details before approving their ability to donate.</td></tr>
<tr><td colspan="2"><b>Priority</b></td><td colspan="2">High</td></tr>
<tr><td colspan="2"><b>Actor(s)</b></td><td colspan="2">Administrator</td></tr>
<tr><td colspan="2"><b>Pre-condition(s)</b></td><td colspan="2">Admin logs in the system and finds some pending donor verification requests.</td></tr>
<tr><td colspan="2"><b>Post-condition(s)</b></td><td colspan="2">The donor is verified by the admin and approved to make donations.</td></tr>
<tr><td colspan="4"><b>Typical Course of Action</b></td></tr>
<tr><td><b>S#</b></td><td colspan="2"><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>1</b></td><td colspan="2">Navigates to the "Verify Donors" section.</td><td></td></tr>
<tr><td><b>2</b></td><td colspan="2"></td><td>Displays a list of pending donor verification requests.</td></tr>
<tr><td><b>3</b></td><td colspan="2">Selects a donor from the list.</td><td></td></tr>
<tr><td><b>4</b></td><td colspan="2"></td><td>Displays the donor's personal information.</td></tr>
<tr><td><b>10</b></td><td colspan="2"></td><td>Marks the donor as unverified.</td></tr>
<tr><td colspan="4"><b>Alternate Course of Action(No verification request)</b></td></tr>
<tr><td><b>S#</b></td><td colspan="2"><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>4</b></td><td colspan="2"></td><td>Prompts “No record found”.</td></tr>
<tr><td colspan="4"><b>Goes back to Step 1</b></td></tr>
</table>

Identifier	UC-07
Name	Verify Aid Request (with System Limit Check)
Summary	This use case allows the **Victim** to request aid, with the system verifying if the requested amount exceeds the predefined limit.
Priority	High
Actor(s)	victim
Pre-condition(s)	The victim is logged into the system.

Post-condition(s)	If the request is within the limit, it is marked as **Verified**.If the request exceeds the limit, it is marked as **Pending** for admin approval.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Navigates to the Request Aid section.	
2		The system displays available aid categories.
3	The victim selects the type and amount of aid needed.	
4	Then the victim enters his/her family information.	
5		The system checks if the requested amount is within the predefined limits.
6		System stores the entered information.
7	Victim submits the request	
8		If the request is within the limit, the system marks it as Verified.
9		The victim is notified by prompt that their request has been successfully submitted.
Alternate Course of Action 1(amount exceeds the limit)		
S#	**Actor Action**	**System Response**
8		The system marks it as **Pending,** and prompts the victim that the request is sent to the admin for verification.
Alternate Course of Action 2(an invalid or empty request)		
7		The system displays an error message: **"Invalid request. Please enter valid aid details."**
Goes back to Step 2.		

2.2 General

Identifier	UC-8	
Name	Perform Basic Search.	
Summary	This use case allows the actor to search for disasters using general keywords without any filtering options.	
Priority	High	
Actor(s)	Visitor, Volunteer, Admin, Donor, Victim	
Pre-condition(s)	The system is accessible. The actor is logged in (if required).	
Post-condition(s)	The system displays a list of disasters matching the search keywords.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Enters a keyword related to a disaster	
2	Submits the search request.	
3		Processes the search query.
4		Retrieves and displays relevant disasters based on the keyword.
Alternate Course of Action 1 (invalid keyword entered)		
S#	**Actor Action**	**System Response**
4		Displays a "No results found" message.
Goes back to Step 1		

Identifier	UC-09

<table>
<tr><td colspan="2"><b>Name</b></td><td colspan="2">Perform Advanced Search.</td></tr>
<tr><td colspan="2"><b>Summary</b></td><td colspan="2">This use case allows the actor to search for disasters using filters like area code and city name.</td></tr>
<tr><td colspan="2"><b>Priority</b></td><td colspan="2">High</td></tr>
<tr><td colspan="2"><b>Actor(s)</b></td><td colspan="2">Visitor, Volunteer, Admin, Donor, Victim</td></tr>
<tr><td colspan="2"><b>Pre-condition(s)</b></td><td colspan="2">The system is accessible. The actor is logged in (if required).</td></tr>
<tr><td colspan="2"><b>Post-condition(s)</b></td><td colspan="2">The system displays a refined list of disasters based on the selected filters.</td></tr>
<tr><td colspan="4"><b>Typical Course of Action</b></td></tr>
<tr><td><b>S#</b></td><td colspan="2"><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>1</b></td><td colspan="2">Selects the Advanced Search option.</td><td></td></tr>
<tr><td><b>2</b></td><td colspan="2"></td><td>Presents filtering options as dropdowns or checkboxes (e.g. area code, city name).</td></tr>
<tr><td><b>3</b></td><td colspan="2">Selects one or multiple filters.</td><td></td></tr>
<tr><td><b>4</b></td><td colspan="2"></td><td>Applies the filters in real-time.</td></tr>
<tr><td><b>5</b></td><td colspan="2">Submits the search request.</td><td></td></tr>
<tr><td><b>6</b></td><td colspan="2"></td><td>Retrieves and displays disasters matching the selected criteria.</td></tr>
<tr><td colspan="4"><b>Alternate Course of Action(invalid area code or city selected)</b></td></tr>
<tr><td><b>S#</b></td><td colspan="2"><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>6</b></td><td colspan="2"></td><td>Displays a "No results found" message.</td></tr>
<tr><td colspan="4"><b>Goes back to Step 1</b></td></tr>
<tr><td colspan="4"><b>Alternate Course of Action 2(proceeds without selecting any filter)</b></td></tr>
<tr><td><b>5</b></td><td colspan="2"></td><td>Treats the search as a Basic Search and retrieves results accordingly.</td></tr>
<tr><td colspan="4"><b>Goes back to Step 1 of Basic Search</b></td></tr>
</table>

Identifier	UC-10

<table>
<tr><td><b>Name</b></td><td colspan="2">Login</td></tr>
<tr><td><b>Summary</b></td><td colspan="2">The actors log into the system.</td></tr>
<tr><td><b>Priority</b></td><td colspan="2">High</td></tr>
<tr><td><b>Actor(s)</b></td><td colspan="2">Administrator, Volunteer,Victim,Donor</td></tr>
<tr><td><b>Pre-condition(s)</b></td><td colspan="2">The actors have valid credentials.</td></tr>
<tr><td><b>Post-condition(s)</b></td><td colspan="2">The actors are logged in and gain access to the system.</td></tr>
<tr><th colspan="3">Typical Course of Action</th></tr>
<tr><th>S#</th><th>Actor Action</th><th>System Response</th></tr>
<tr><td><b>1</b></td><td>Navigates to the login page.</td><td></td></tr>
<tr><td><b>2</b></td><td></td><td>Displays login page.</td></tr>
<tr><td><b>3</b></td><td>Enters username and password.</td><td></td></tr>
<tr><td><b>4</b></td><td></td><td>Validates credentials.</td></tr>
<tr><td><b>5</b></td><td>Clicks "Login" button.</td><td></td></tr>
<tr><td><b>6</b></td><td></td><td>Grants access and redirects to relevant actors dashboard.</td></tr>
<tr><th colspan="3">Alternate Course of Action(Invalid Data)</th></tr>
<tr><th>S#</th><th>Actor Action</th><th>System Response</th></tr>
<tr><td><b>4</b></td><td></td><td>Prompts ‘Invalid Credentials’ error.</td></tr>
</table>

Identifier	UC-11
Name	Report a Disaster
Summary	Allows to report a new disaster/incident to the system.
Priority	High
Actor(s)	Victim,Donor, volunteer
Pre-condition(s)	The Actor(s) is logged into the system.
Post-condition(s)	The disaster/incident is reported and pending verification by the admin.

Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Navigates to the 'Report Disaster' section.	
2		Displays a form to report a disaster.
3	Enter disaster details (e.g name, address, severity, required resources).	
4	Clicks 'Submit'.	
5		Validates input fields.
6		Saves the disaster report in the database.
7		Displays a confirmation message: "Disaster reported successfully."
Alternate Course of Action 1(Incomplete Disaster details)		
S#	**Actor Action**	**System Response**
5		Prompts 'Incomplete Details' error.
Goes back to Step 2.		

Identifier	UC-12
Name	Logout
Summary	Actors log out of the system.
Priority	Low
Actor(s)	Administrator, Volunteer, Victim, Donor
Pre-condition(s)	The Actors are logged into the system.
Post-condition(s)	The Actors are logged out and redirected to the home page.

Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Clicks the "Logout" button.	

2		Logs the user out.
3	Redirects to the login page.	
4		Confirms logout with a message.

<table>
<tr><td><b>Identifier</b></td><td colspan="2">UC-13</td></tr>
<tr><td><b>Name</b></td><td colspan="2">Sign Up</td></tr>
<tr><td><b>Summary</b></td><td colspan="2">Allows a user to create an actor(s) account in the system.</td></tr>
<tr><td><b>Priority</b></td><td colspan="2">High</td></tr>
<tr><td><b>Actor(s)</b></td><td colspan="2">Volunteer, Victims, Donor</td></tr>
<tr><td><b>Pre-condition(s)</b></td><td colspan="2">The system is operational and accessible. The user does not already have a relevant account.</td></tr>
<tr><td><b>Post-condition(s)</b></td><td colspan="2">A new actor's account is created. The user can now log in and access their functionalities.</td></tr>
<tr><th colspan="3">Typical Course of Action</th></tr>
<tr><th>S#</th><th>Actor Action</th><th>System Response</th></tr>
<tr><td><b>1</b></td><td>Navigates to the "Sign Up" page.</td><td></td></tr>
<tr><td><b>2</b></td><td></td><td>Displays the registration form.</td></tr>
<tr><td><b>3</b></td><td>Selects desired role.</td><td></td></tr>
<tr><td><b>4</b></td><td></td><td>Updates form options accordingly.</td></tr>
<tr><td><b>5</b></td><td></td><td>Prompts option to select NGO if desired role is ‘Donor’.</td></tr>
<tr><td><b>6</b></td><td>Fills in required details and clicks "Submit."</td><td></td></tr>
<tr><td><b>7</b></td><td></td><td>Validates input fields.</td></tr>
<tr><td><b>8</b></td><td></td><td>Saves user information and creates account.</td></tr>
<tr><th colspan="3">Alternate Course of Action 1 (If account already exists)</th></tr>
<tr><th>S#</th><th>Actor Action</th><th>System Response</th></tr>
</table>

7		Prompts to use a different email account.
Goes back to Step 6		
Alternate Course of Action 2 (Invalid data)		
S#	**Actor Action**	**System Response**
7		Prompts an error and requests corrections.
Goes back to Step 6		

2.3 Volunteer

Identifier	UC-14	
Name	Choose Volunteer Role	
Summary	Allows volunteers to specify whether they will work on-field or off-field.	
Priority	Medium	
Actor(s)	Volunteer	
Pre-condition(s)	The user is logged in as a Volunteer. The user has not yet set a role.	
Post-condition(s)	The volunteer's role is recorded in the system. Role Relevant assignments are displayed.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Navigates to the "Role Selection" section.	
2		Displays available roles.
3	Selects a role (on-field/off-field).	
4		Updates role information.
5	Provides additional details if required (e.g. skills, working hours, etc).	

6		Saves input.
7	Confirms role selection	
8		Updates volunteer profile.
9	Confirmation message displayed	
10		Role selection is now saved.
Alternate Course of Action(Missing Information)		
S#	**Actor Action**	**System Response**
6		Prompts the user to fill in missing details.
Goes back to Step 5		

Identifier	UC-15	
Name	Choose a Task.	
Summary	A volunteer selects a task from available tasks.	
Priority	High	
Actor(s)	Volunteer	
Pre-condition(s)	The user is logged in as a Volunteer.	
Post-condition(s)	The task is assigned to the volunteer.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Navigates to "Choose Task" section.	
2		Displays available tasks.
3	Selects a task from the list.	
4		Displays task details.
5	Confirms task selection.	
6		Assigns the task and updates the system.

7	Receives confirmation message.	
8		Displays a success notification.
Alternate Course of Action(No task available)		
S#	**Actor Action**	**System Response**
4		Prompts "No task available " message.

Identifier	UC-16	
Name	View Google Map	
Summary	A volunteer views victims within a 5km radius using Google Maps.	
Priority	High	
Actor(s)	Volunteer	
Pre-condition(s)	The Volunteer is logged into the system. Location services and Google Maps access are enabled.	
Post-condition(s)	The volunteer will be able to view victims in the 5km radius.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Navigates to "View Victims" section.	
2		Displays a Google map with victim locations within a 5km radius marked on it.
Alternate Course of Action(No disaster detected)		
S#	**Actor Action**	**System Response**
2		Display map only and Prompts "No Victim locations nearby" message.
Goes back to Step 1		

Identifier	UC-17
Name	Update Tasks.
Summary	The Volunteer updates progress on an assigned task.
Priority	Medium
Actor(s)	Volunteer
Pre-condition(s)	The Volunteer is logged into the system and has an assigned task.
Post-condition(s)	The task details are updated with the latest status.
Typical Course of Action	

S#	Actor Action	System Response
1	Navigates to the "Update Tasks" section.	
2		Displays the list of assigned tasks.
3	Select the task to update.	
4		Displays Task Details.
5	Updates task status.	
6		Validates and saves the update.
7	Confirms submission.	
8		Displays a success message.
Alternate Course of Action(Incomplete Details)		
S#	**Actor Action**	**System Response**
4		Prompts the user to retry.
Goes back to Step 5		

2.4 Donor

<table>
<tr><td><b>Identifier</b></td><td colspan="2">UC-18</td></tr>
<tr><td><b>Name</b></td><td colspan="2">Donate Currency as a Donor</td></tr>
<tr><td><b>Summary</b></td><td colspan="2">Allows the user to donate currency in general.</td></tr>
<tr><td><b>Priority</b></td><td colspan="2">Medium</td></tr>
<tr><td><b>Actor(s)</b></td><td colspan="2">Donor</td></tr>
<tr><td><b>Pre-condition(s)</b></td><td colspan="2">The user is logged in and verified as a Donor.</td></tr>
<tr><td><b>Post-condition(s)</b></td><td colspan="2">The system updates account balance accordingly.</td></tr>
<tr><td colspan="3"><b>Typical Course of Action</b></td></tr>
<tr><td><b>S#</b></td><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>1</b></td><td>Selects ‘Donate Money’.</td><td></td></tr>
<tr><td><b>2</b></td><td>Enter the amount to donate.</td><td></td></tr>
<tr><td><b>3</b></td><td>Selects ‘General’.</td><td></td></tr>
<tr><td><b>4</b></td><td>Clicks ‘Confirm’.</td><td></td></tr>
<tr><td><b>5</b></td><td></td><td>Validates input fields.</td></tr>
<tr><td><b>6</b></td><td></td><td>Prompts a thanks to the user.</td></tr>
<tr><td><b>7</b></td><td></td><td>Updates account balance and stores user information and amount.</td></tr>
<tr><td colspan="3"><b>Alternate Course of Action 1 (Enters a negative amount)</b></td></tr>
<tr><td><b>S#</b></td><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>5</b></td><td></td><td>Prompts ‘Invalid Amount’ error and requests a correction.</td></tr>
<tr><td colspan="3"><b>Goes back to Step 2</b></td></tr>
</table>

Identifier	UC-19
Name	Donate Items as a Donor.
Summary	Allows the user to donate items in general.
Priority	Medium
Actor(s)	Donor
Pre-condition(s)	The user is logged in and verified as a Donor.
Post-condition(s)	The system updates inventory accordingly.

Typical Course of Action

S#	Actor Action	System Response
1	Selects ‘Donate Items’.	
2		Display list of items.
3	Selects what item to donate from the list, and enters quantity of said item.	
4	Selects ‘General’.	
5	Clicks ‘Confirm’.	
6		Validates input fields.
7		Displays address information to send items to.
8		Prompts a thanks to the user.
9		Updates inventory.
10		Stores user information and items donated.

Alternate Course of Action 1 (Enters a negative amount)

S#	Actor Action	System Response
6		Prompts a ‘Invalid Amount’ error and requests a correction.

Goes back to Step 3

Identifier	UC-20	
Name	Donate to Specific Disaster.	
Summary	Allows the user to donate to a specific disaster	
Priority	Medium	
Actor(s)	Donor	
Pre-condition(s)	The user is logged in and verified as a Donor. The user has already selected whether to donate currency or items and has entered the amount.	
Post-condition(s)	The system updates specific disaster inventory and balances accordingly.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Selects 'Specific Disaster'.	
2		Displays List of Disasters.
3	Selects disaster to donate to.	
4	Clicks 'Confirm'.	
5		Validates inputs.
6		Displays a thank you message.
7		Updates specific disaster inventory/balance.
8		Stores user information.
Alternate Course of Action (Does not choose a disaster)		
S#	**Actor Action**	**System Response**
5		Prompts 'No Disaster Chosen' error and requests a disaster to be chosen.
Goes back to Step 3		

<table>
<tr><td><b>Identifier</b></td><td colspan="2">UC-21</td></tr>
<tr><td><b>Name</b></td><td colspan="2">Offer shelter to victims.</td></tr>
<tr><td><b>Summary</b></td><td colspan="2">Allows the user to offer shelter on basis of beds available.</td></tr>
<tr><td><b>Priority</b></td><td colspan="2">Low</td></tr>
<tr><td><b>Actor(s)</b></td><td colspan="2">Donor</td></tr>
<tr><td><b>Pre-condition(s)</b></td><td colspan="2">The user is logged in and verified as a Donor.</td></tr>
<tr><td><b>Post-condition(s)</b></td><td colspan="2">The system updates available beds accordingly.</td></tr>
<tr><td colspan="3"><b>Typical Course of Action</b></td></tr>
<tr><td><b>S#</b></td><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>1</b></td><td>Selects ‘Offer Shelter’.</td><td></td></tr>
<tr><td><b>2</b></td><td>Enter the amount of beds available.</td><td></td></tr>
<tr><td><b>3</b></td><td>Enters name and address.</td><td></td></tr>
<tr><td><b>4</b></td><td>Selects disaster.</td><td></td></tr>
<tr><td><b>5</b></td><td>Clicks ‘Confirm’.</td><td></td></tr>
<tr><td><b>6</b></td><td></td><td>Prompts a disclosure agreement of sharing shelter.</td></tr>
<tr><td><b>7</b></td><td>Clicks ‘Confirm’.</td><td></td></tr>
<tr><td><b>8</b></td><td></td><td>Validates input fields.</td></tr>
<tr><td><b>9</b></td><td></td><td>Updates available beds.</td></tr>
<tr><td><b>10</b></td><td></td><td>Prompts a thank you message.</td></tr>
<tr><td colspan="3"><b>Alternate Course of Action 1 (Enters a negative amount of beds)</b></td></tr>
<tr><td><b>S#</b></td><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>8</b></td><td></td><td>Prompts ‘Invalid Amount’ error and requests a correction.</td></tr>
<tr><td colspan="3"><b>Goes back to Step 2</b></td></tr>
<tr><td colspan="3"><b>Alternate Course of Action 2 (Does not select a disaster)</b></td></tr>
</table>

8		Prompts 'Select Disaster' error.
Alternate Course of Action 3 (Does not confirm agreement to share shelter.)		
8		Prompts a 'Agreement Not Signed' error.
Goes back to Step 7		

2.5 Victim

Identifier	UC-22
Name	Request Aid (with System Limit Check)
Summary	This use case allows the **Victim** to request aid, with the system verifying if the requested amount exceeds the predefined limit.
Priority	High
Actor(s)	victim
Pre-condition(s)	The victim is logged into the system.
Post-condition(s)	If the request is within the limit, it is marked as **Verified**.If the request exceeds the limit, it is marked as **Pending** for admin approval.

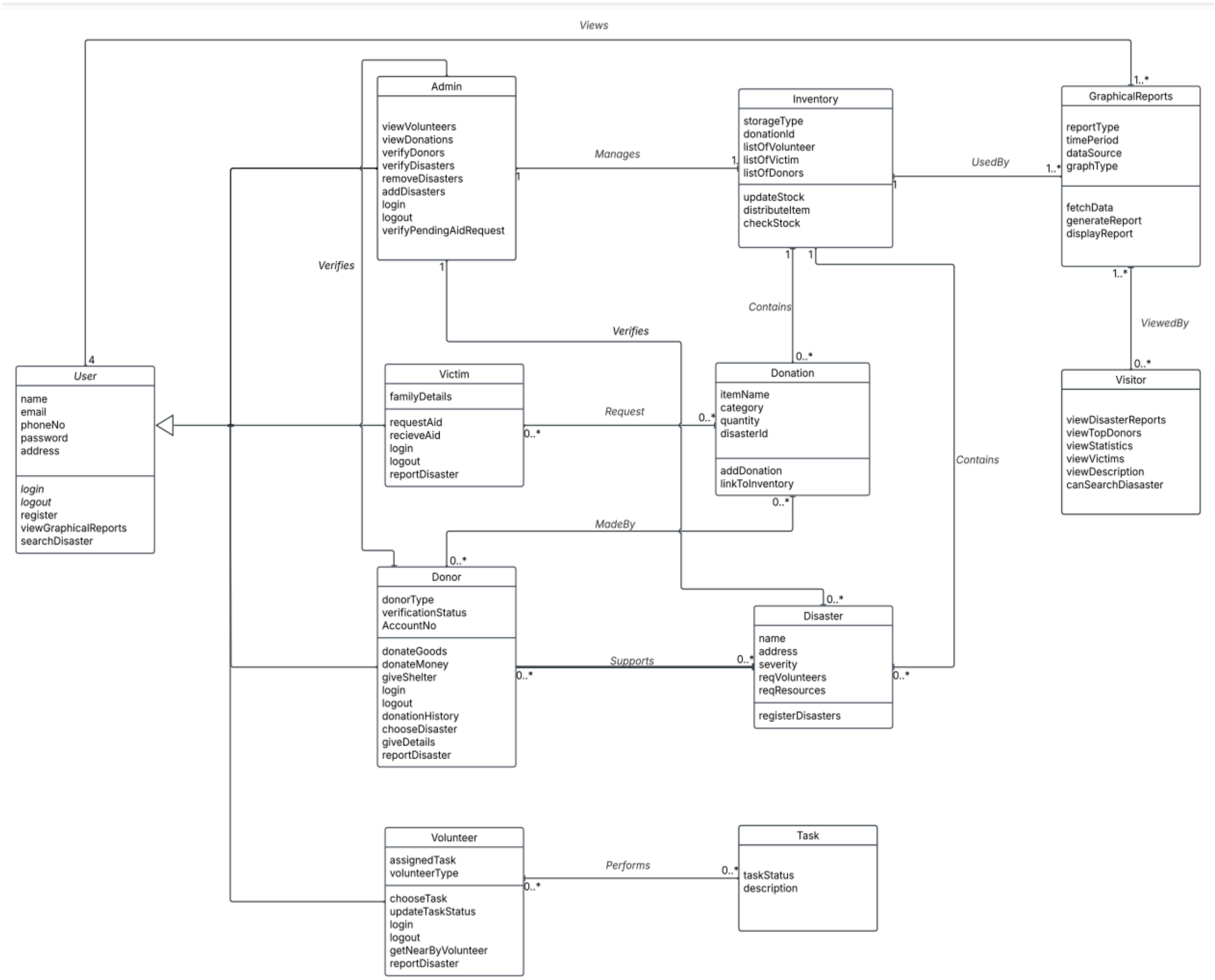
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	The victim navigates to the **Request Aid** section.	
2		The system displays available aid categories.
3	The victim selects the type and amount of aid needed.	
4	Then the victim enters his/her family information.	

5		The system checks if the requested amount is within the predefined limits.
6		System stores the entered information.
7	Victim submits the request	
8		If the request is **within** the limit, the system marks it as **Verified**.
9		The victim is notified by prompt that their request has been successfully submitted.
Alternate Course of Action 1(amount exceeds the limit)		
S#	**Actor Action**	**System Response**
8		The system marks it as **Pending,** and prompts the victim that the request is sent to the admin for verification.
Alternate Course of Action 2(an invalid or empty request)		
7		The system displays an error message: **"Invalid request. Please enter valid aid details."**
Goes back to Step 2.		

Identifier	UC-23
Name	Receive Aid
Summary	Allows the Victim to confirm receipt of aid (food, clothes, shelter, medical help) from volunteers.
Priority	High
Actor(s)	Victim
Pre-condition(s)	The Victim has requested aid, and the aid has been allocated by volunteers.
Post-condition(s)	The Victim receives the requested aid, and the system updates the aid status.
Typical Course of Action	

<table>
<tr><th>S#</th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1</td><td>Victim navigates to the ‘Aid Status’ section.</td><td></td></tr>
<tr><td>2</td><td></td><td>Displays the status of the requested aid.</td></tr>
<tr><td>3</td><td>Victim confirms receipt of aid.</td><td></td></tr>
<tr><td>4</td><td></td><td>Updates the aid status to "Received" in the database.</td></tr>
<tr><td>5</td><td></td><td>Displays a confirmation message: "Aid received successfully."</td></tr>
<tr><th colspan="3">Alternate Course of Action 1(If not confirmed)</th></tr>
<tr><th>S#</th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1</td><td></td><td>Keeps the aid status as “Verified".</td></tr>
</table>

3. ACD


Views
1..*
4
Admin
viewVolunteers
viewDonations
verifyDonors
verifyDisasters
removeDisasters
addDisasters
login
logout
verifyPendingAidRequest
Manages
1
1
Inventory
storageType
donationId
listOfVolunteer
listOfVictim
listOfDonors
updateStock
distributeItem
checkStock
UsedBy
1
1..*
GraphicalReports
reportType
timePeriod
dataSource
graphType
fetchData
generateReport
displayReport
1..*
ViewedBy
0..*
Visitor
viewDisasterReports
viewTopDonors
viewStatistics
viewVictims
viewDescription
canSearchDiasaster
Verifies
1
Verifies
1
1
Contains
0..*
User
name
email
phoneNo
password
address
login
logout
register
viewGraphicalReports
searchDisaster
Victim
familyDetails
requestAid
recieveAid
login
logout
reportDisaster
Request
0..*
0..*
Donation
itemName
category
quantity
disasterId
addDonation
linkToInventory
0..*
MadeBy
0..*
Contains
Donor
donorType
verificationStatus
AccountNo
donateGoods
donateMoney
giveShelter
login
logout
donationHistory
chooseDisaster
giveDetails
reportDisaster
0..*
Disaster
name
address
severity
reqVolunteers
reqResources
registerDisasters
Supports
0..*
0..*
0..*
Volunteer
assignedTask
volunteerType
chooseTask
updateTaskStatus
login
logout
getNearByVolunteer
reportDisaster
Performs
0..*
0..*
Task
taskStatus
description

4. NFR

1. Performance Requirements

1.1 The system shall process user actions (e.g., logging in, reporting an incident, or making a donation) within **10 seconds** under typical conditions.
1.2 The system shall load all web pages and fetch data within **10 seconds** on an average broadband connection.

2. Safety Requirements

2.1 The system shall prevent unauthorized access to disaster and donation records by implementing **role-based access control (RBAC)** to ensure data integrity and confidentiality.
2.2 The system shall validate all user inputs to prevent incorrect or harmful data submissions using **format checks, length limits, and regular expressions**.
2.2.1 The system shall reject inputs containing **special characters(/,\,^,% etc)** for critical data fields (e.g., names and addresses).

3. Security Requirements

3.1 The system shall require **user authentication** for all roles (victims, volunteers, and administrators) before accessing their respective functionalities.
3.2 The system shall restrict access to administrative actions (e.g., adding or removing disasters) to **authorized users only**.

4. Software Quality Attributes

4.1 Usability

4.1.1 The system shall ensure that at least **90% of test users successfully complete core tasks** (e.g. reporting an incident, donating) on the first attempt.
4.1.2 Navigation menus and UI elements shall remain consistent across different web pages, tested via **user interface testing**.

4.2 Maintainability

4.2.1 The system shall use **modular architecture**, ensuring that updates and bug fixes do not affect more than **10% of system functionality**, verified through regression testing.

4.2.2 The system codebase shall maintain at least **90% unit test coverage** to ensure maintainability and reliability.

4.3 Reliability

4.3.1 The system shall maintain an uptime of 90% over a one-month period, monitored through server logs and automated availability checks.

4.3.2 The system shall **log all errors and system events** for troubleshooting.

4.4 Portability

4.4.1 The system shall run on a local server and be accessible through a web browser in the development environment.

4.4.2 The system shall be tested for functionality on **at least one modern browser** (e.g. Google Chrome).