



Software Requirements Specification

Project Flood Aid



“Whoever does good, it is for his own soul.”

Project Title: Flood Aid

Semester: Fall-2025



*Department of Computer Science
Rachna College of Engineering and Technology,
Gujranwala
(A Constituent College of UET Lahore)*

Course Name: Software Engineering (CSC-204)

Section:

B

Submitted By:

Muhammad Mustqeem (2024-CS-762)

Muhammad Umair (2024-CS-480)

Abdullah Umar (2024-CS-496)

Farhan Shakeel (2024-CS-461)

Submitted To:

Mam Natasha Nigar

Software Requirements Specification

Project Flood Aid

Table of Contents

Table of Contents.....	1
1.0 Introduction.....	3
1.1 Purpose.....	3
1.2 Scope of Project.....	3
1.3 Glossary	3
1.4 References	3
1.5 Overview of Document	4
1.6 Feasibility Study.....	4
1.6.1 Technical Feasibility.....	4
1.6.2 Economic Feasibility	4
1.6.3 Operational Feasibility	4
2.0 Overall Description	5
2.1 System Environment (Deployment Diagram).....	5
2.2 Functional Requirements (Use Case Specifications)	6
2.2.1 Victim Use Cases.....	7
2.2.2 Volunteer Use Cases	7
2.2.3 Donor Use Cases.....	8
2.2.4 Administrator Use Cases.....	9
3.0. Requirements Specification.....	9
3.1 External Interface Requirements.....	9
○ User Interfaces:	9
○ Hardware Interfaces:	9
○ Software Interfaces:	9
3.2 Functional Requirements (Detailed logic for each use case above).....	10
○ FR-01 (Victim):.....	10
○ FR-02 (Volunteer):.....	10
○ FR-03 (Inventory):.....	10
○ FR-04 (Admin):.....	10
○ FR-05 (Donor):	10
3.3 Detailed Non-Functional Requirements.....	10

Software Requirements Specification

Project Flood Aid

3.3.1 Logical Structure of the Data (Class Diagram).....	10
3.3.2 Security.....	11
○ Authentication:.....	11
○ Encryption:.....	11
○ Access Control:	11
4.0 WBS	12
4.1 Chart.....	12

List of Figures:

FIGURE 1: DEPLOYMENT DIAGRAM	5
FIGURE 2: COMBINED USE CASE DIAGRAM	6
FIGURE 3: VICTIM USE CASE DIAGRAM	7
FIGURE 4: VOLUNTEER USE CASE DIAGRAM.....	8
FIGURE 5: DONOR USE CASE DIAGRAM	8
FIGURE 6: ADMINISTRATOR USE CASE DIAGRAM.....	9
FIGURE 7: CLASS DIAGRAM (UML NOTATION)	10
FIGURE 8: CLASS DIAGRAM (CROW FOOT NOTATION)	11
FIGURE 9: WBS (USING SMARTART)	12
FIGURE 10: WBS (USING UML CODE).....	12

Software Requirements Specification

Project Flood Aid

1.0 Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) is to define the functional and non-functional requirements for **FloodAid**, a web-based flood relief management system. This document serves as the primary agreement between the development team and stakeholders, detailing how the system will coordinate victims, volunteers, donors, and administrators during disaster relief operations.

1.2 Scope of Project

Flood Aid is a centralized platform designed to bridge the gap between flood victims and relief efforts.

- **In Scope:** The system includes a **Victim Portal** for submitting help requests with geolocation; a **Volunteer Dashboard** for accepting and tracking rescue tasks; a **Donor Module** for contributing funds and goods; and an Admin Console for inventory management and reporting.
- **Out of Scope:** The system will not cover **real-time weather forecasting** (we can implement this if project is completed early using API of some weather station) or **offline-first mobile capabilities** in this initial version (v1.0).

1.3 Glossary

- **Victim:** Any user located in a flood-affected area submitting a request for help.
- **Volunteer:** A verified user assigned to deliver aid or perform rescue operations.
- **Admin:** A super-user responsible for managing inventory, verifying users, and viewing system-wide reports.
- **OCI:** Oracle Cloud Infrastructure (the hosting platform).

1.4 References

- *Flood Aid Project Brief / Semester Project Guidelines.*
- *UML Diagrams Course Slides (Week 12).*
- *SRS Template (Online Shopping System).*

Software Requirements Specification

Project Flood Aid

1.5 Overview of Document

This document is organized into three main parts:

1. **Overall Description:** A high-level view of the system environment and user functions.
2. **Requirements Specification:** Detailed technical requirements, interfaces, and data models.
3. **WBS:** The project schedule and breakdown of tasks.

1.6 Feasibility Study

1.6.1 Technical Feasibility

The project utilizes a modern and robust technology stack (.NET Core and React) which is well-documented and scalable. The team possesses the necessary skills to implement these technologies on the Oracle Cloud Infrastructure (OCI).

1.6.2 Economic Feasibility

The project uses open-source development tools (VS Code) and the OCI "Always Free" tier for hosting, ensuring the project requires zero financial investment for infrastructure during the development phase.

1.6.3 Operational Feasibility

The system addresses a critical need for centralized disaster management. The user interfaces are designed for high accessibility, ensuring that non-technical users (victims) can easily interact with the system during emergencies.

Software Requirements Specification

Project Flood Aid

2.0 Overall Description

2.1 System Environment (Deployment Diagram)

The FloodAid system operates on a cloud-native architecture. The frontend is built with **React.js** and deployed on **Vercel** for global content delivery. The backend API is built with **ASP.NET Core**, hosted on **Oracle Cloud Infrastructure (OCI)** Compute instances. Data is stored in an **Oracle Autonomous Database** to ensure high availability and security.

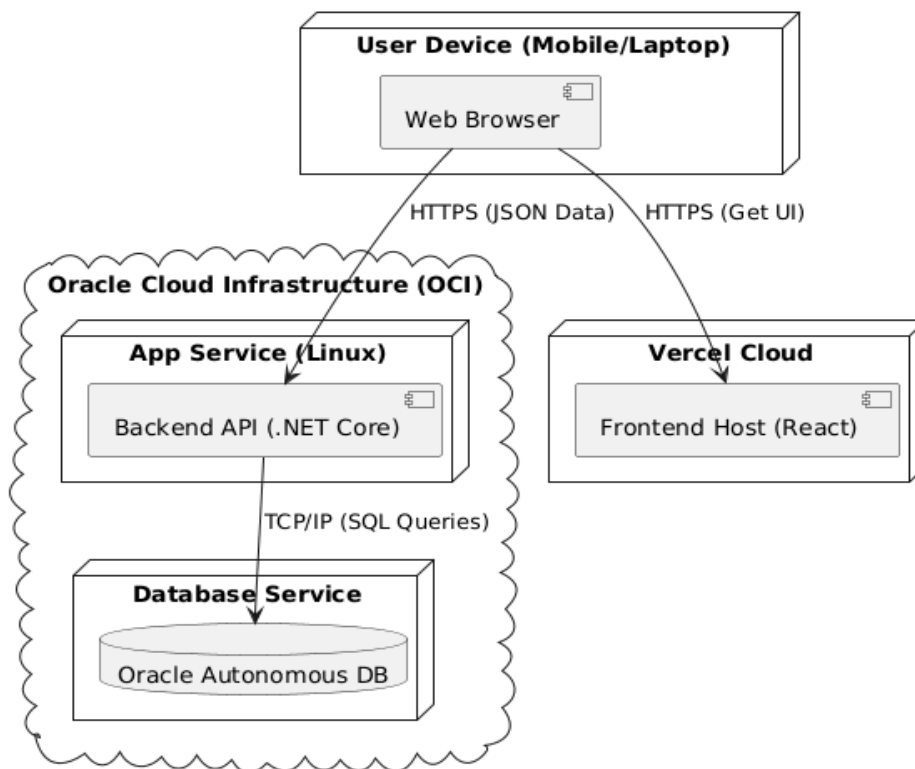


Figure 1: Deployment Diagram

Software Requirements Specification

Project Flood Aid

2.2 Functional Requirements (Use Case Specifications)

The system supports four primary actors: Victims, Volunteers, Donors, and Administrators. The interactions between these actors and the system are defined below.

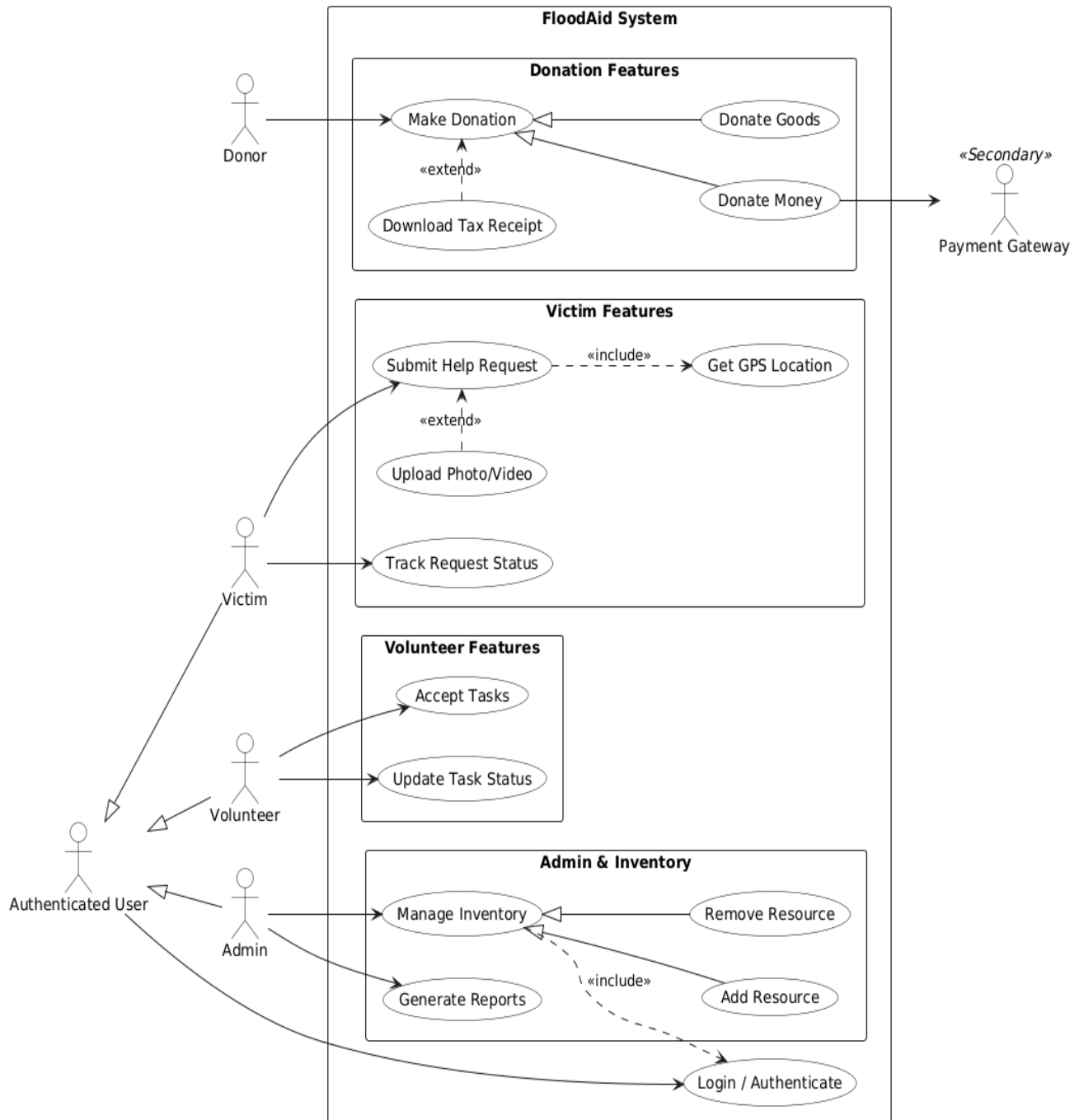


Figure 2: Combined Use Case Diagram

Software Requirements Specification

Project Flood Aid

2.2.1 Victim Use Cases

Use case 01: Submit Help Request

- **Goal:** Allow a victim to ask for aid.
- **Trigger:** Victim clicks "I need Help".
- **Precondition:** Victim must provide a valid phone number.
- **Flow:** System captures GPS location, User selects aid type (Food, Medical, Rescue), System saves request.

Use case 02: Track Request Status

- **Goal:** Allow victim to see if help is coming.
- **Postcondition:** System displays "Pending", "Assigned", or "Resolved".

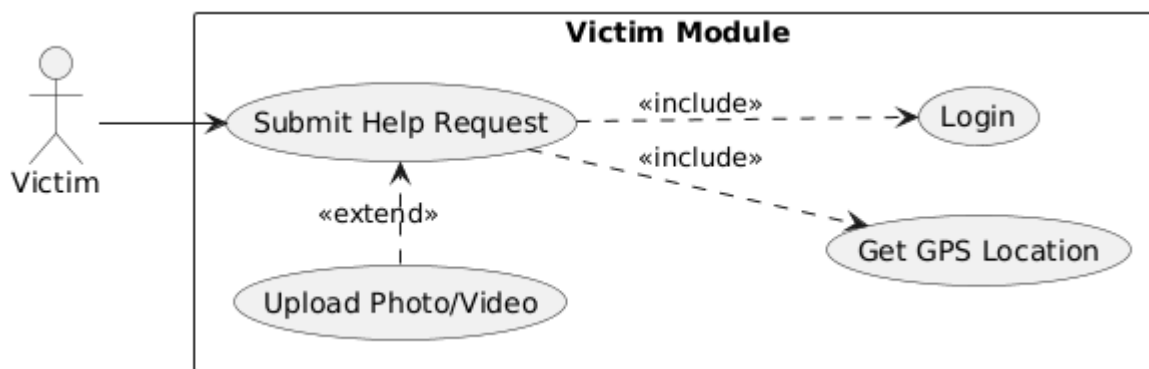


Figure 3: Victim Use Case Diagram

2.2.2 Volunteer Use Cases

Use case 01: Accept Tasks

- **Goal:** Assign a volunteer to a specific victim.
- **Logic:** Volunteer views map of nearby requests and clicks "Accept". The system locks the request so no other volunteer can take it.

Use case 02: Update Task Status

- **Goal:** Report progress.
- **Logic:** Volunteer marks request as "In Progress" upon arrival and "Resolved" upon delivery of aid.

Software Requirements Specification

Project Flood Aid

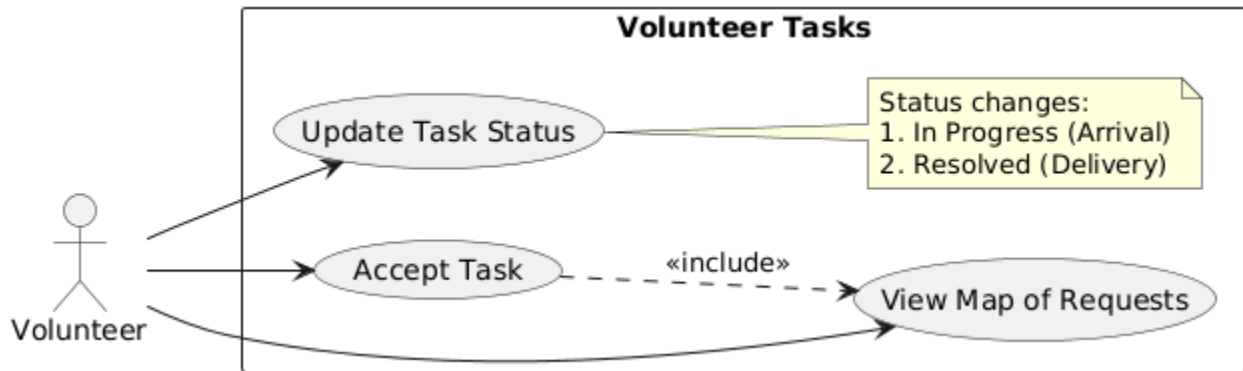


Figure 4: Volunteer Use Case Diagram

2.2.3 Donor Use Cases

Use case 01: Donate Funds (Credit Payment)

- **Goal:** Process monetary donations securely.
- **Flow:** Donor enters amount -> System redirects to Payment Gateway -> Transaction confirmed.

Use case 02: Donate Goods

- **Goal:** Pledge physical items (blankets, food).
- **Flow:** Donor lists items -> Admin approves pledge -> Donor receives drop-off location.

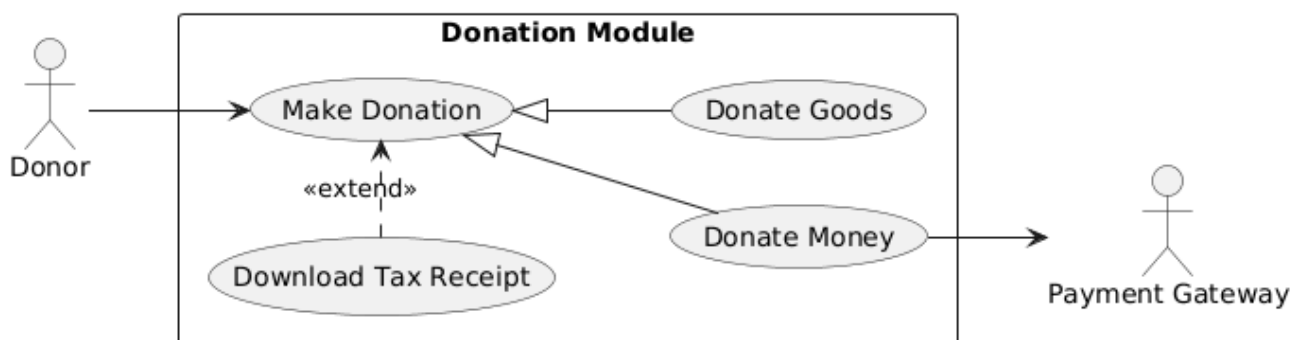


Figure 5: Donor Use Case Diagram

Software Requirements Specification

Project Flood Aid

2.2.4 Administrator Use Cases

Use case 01: Service Authentication (Identity Provider)

- All admins must authenticate via multi-factor authentication before accessing the dashboard.

Use case 02: Manage Inventory

- The admin dashboard allows full CRUD (Create, Read, Update, Delete) operations on the relief goods database.

Use case 03: Generate Reports

- Admin can export PDF summaries of daily rescue operations.

Use case 04: Add Relief Resource

- Admin adds new shipments of food/medicine to the virtual inventory.

Use case 05: Remove Relief Resource

- Admin removes expired/damaged goods from the inventory.

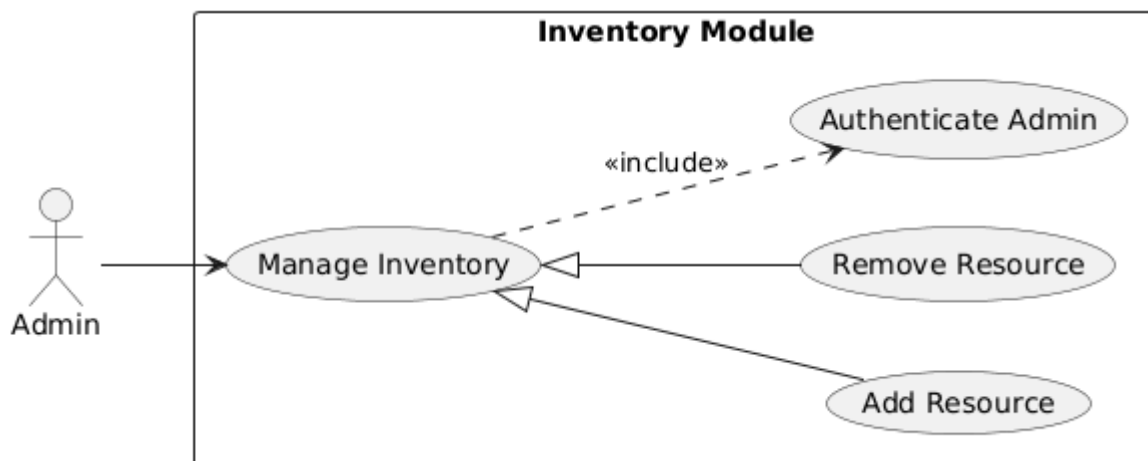


Figure 6: Administrator Use Case Diagram

3.0. Requirements Specification

3.1 External Interface Requirements

- **User Interfaces:** The system shall provide a Responsive Web Interface compatible with Chrome, Safari, and Edge.
- **Hardware Interfaces:** The system shall interface with the user's device GPS module to retrieve geolocation coordinates (Latitude/Longitude).
- **Software Interfaces:** The system shall integrate with the **Google Maps API** for location rendering and **Brevo** for email notifications.

Software Requirements Specification

Project Flood Aid

3.2 Functional Requirements (Detailed logic for each use case above)

- **FR-01 (Victim):** The system shall allow victims to submit a help request with a maximum of 3 photos.
- **FR-02 (Volunteer):** The system shall prevent a volunteer from accepting more than 3 active tasks simultaneously.
- **FR-03 (Inventory):** The system shall automatically deduct stock from the "Inventory" table when a Volunteer marks a task as "Aid Delivered".
- **FR-04 (Admin):** The system shall generate a "Low Stock Alert" when any item quantity falls below 50 units.
- **FR-05 (Donor):** The system shall send an automated tax receipt via email immediately after a successful credit payment transaction.

3.3 Detailed Non-Functional Requirements

3.3.1 Logical Structure of the Data (Class Diagram)

The data model utilizes **Composition** for critical dependencies (User-Request) and **Aggregation** for loose groupings (Team-Volunteer), adhering to strict object-oriented principles .

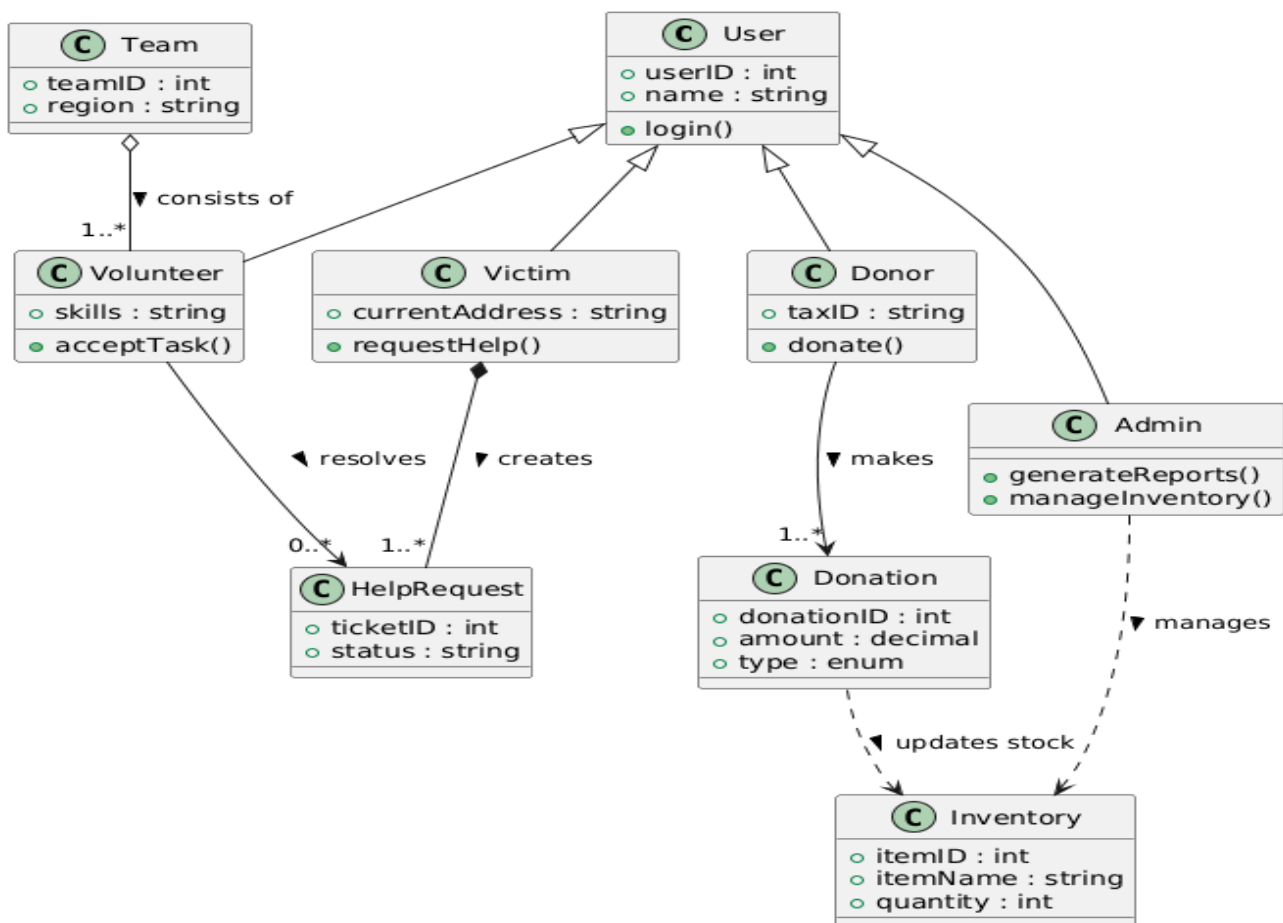


Figure 7: Class Diagram (UML Notation)

Software Requirements Specification

Project Flood Aid

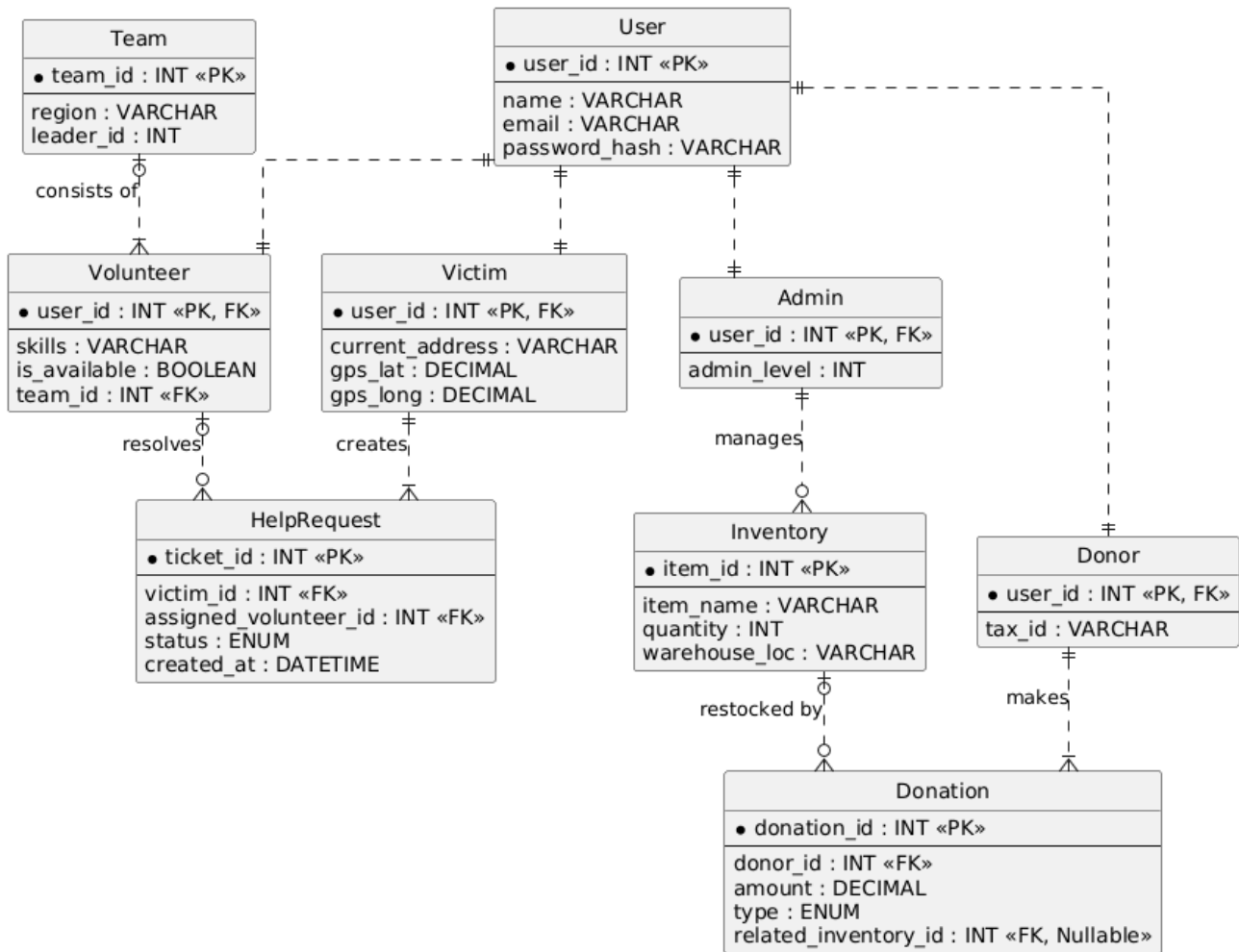


Figure 8: Class Diagram (Crow Foot Notation)

3.3.2 Security

- **Authentication:** All passwords shall be hashed using **Bcrypt** before storage.
 - "While Argon2id is the current state-of-the-art, we have selected Bcrypt (Work Factor 11) for its proven reliability in .NET environments and ease of maintenance."
- **Encryption:** All data in transit must be encrypted via **TLS 1.2+** (HTTPS).
- **Access Control:** The system shall enforce **Role-Based Access Control (RBAC)**.

Software Requirements Specification

Project Flood Aid

4.0 WBS

4.1 Chart

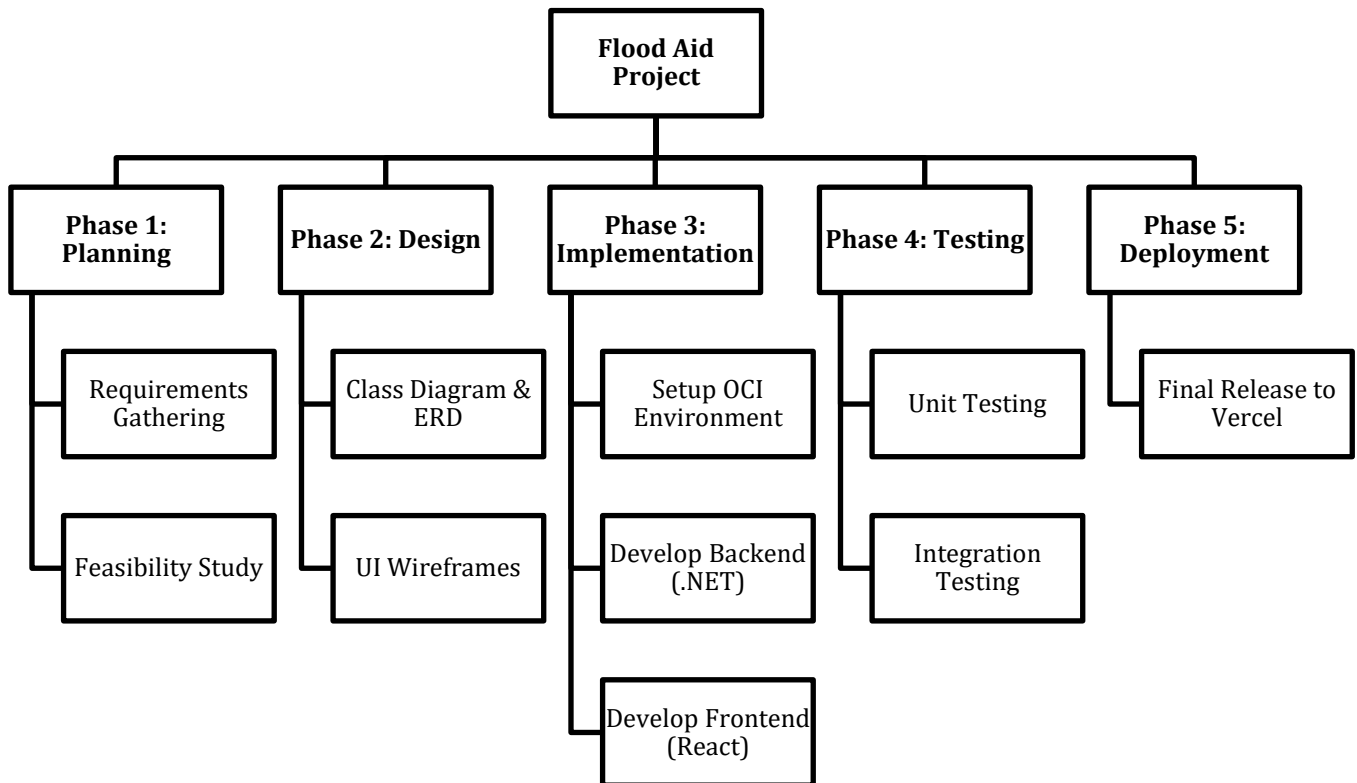


Figure 9: WBS (using SmartArt)

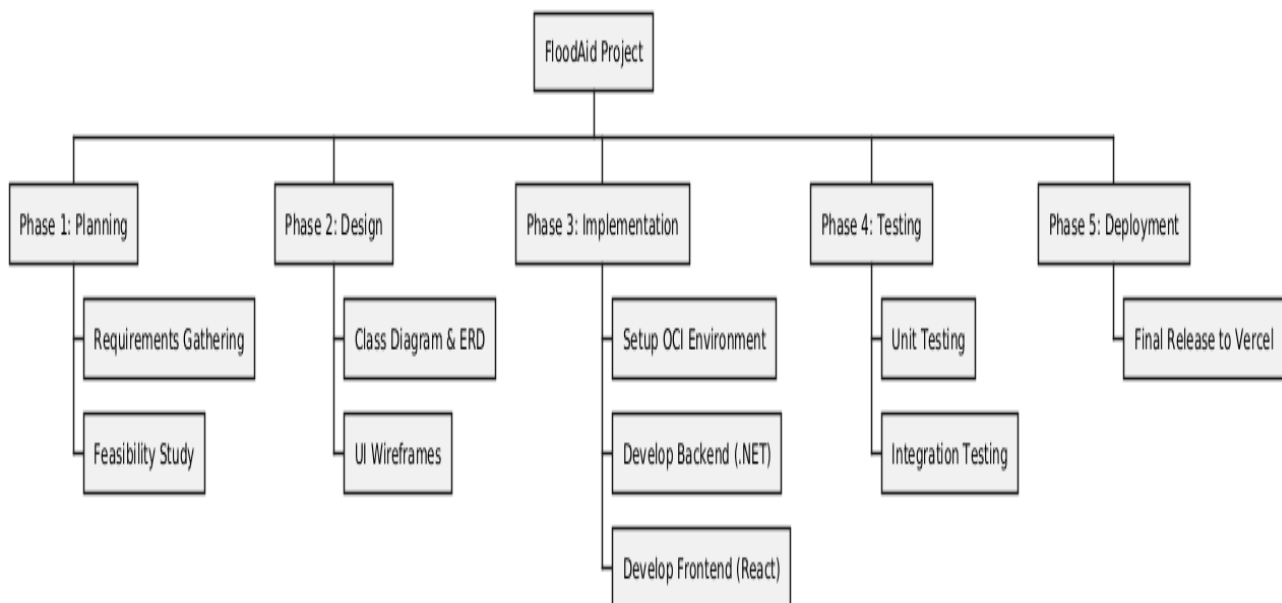


Figure 10: WBS (using UML Code)