

## **RELIFE**

### **24CSJ303: ADVANCED PROGRAMMING - PROJECT REPORT**

Submitted By

**AJAY N (TKM24CS020)**

**NITHIN M (TKM24CS102)**

**SALMAN A (TKM24CS112)**

**SREERAG V S (TKM24CS130)**

*to*

*TKM College of Engineering (Govt. Aided and Autonomous)*

*in partial fulfillment of the requirements for the award of Bachelor of Technology*

*in Computer Science and Engineering*



**Department of Computer Science and Engineering**

T.K.M College of Engineering, Kollam

October 2025

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING  
T. K. M COLLEGE OF ENGINEERING, KOLLAM**



**CERTIFICATE**

This is to certify that the report entitled '**RELIFE**' submitted by **Ajay N (TKM24CS020), Nithin M (TKM24CS102), Salman A (TKM24CS112) and Sreerag V S (TKM24CS130)** to TKM College of Engineering (Govt. Aided and Autonomous affiliated to the APJ Abdul Kalam Technological University) in partial fulfillment of the B.Tech. degree in Computer Science and Engineering is a bonafide record of the project work carried out by them under our guidance and supervision. This report in any form has not been submitted to any other University or purpose.

**Project Coordinator**

**Head of the Department**

## DECLARATION

We hereby certify that the project report 'RELIFE' presented for partial fulfillment of the requirements in respect of awarding the degree of Bachelor of Technology of APJ Abdul Kalam Technological University, Kerala is an authentic work performed by us under the guidance of Dr. Jini Raju. This submission is our thoughts in my own words and where thoughts or words of others have been incorporated, we have properly and accurately cited and referenced. We further affirm that we have acted in accordance with the ethics of academic honesty and integrity and we have not misrepresented or concocted any data or idea or fact or source in my submission. We acknowledge that any breach of the above will be a reason for disciplinary action by the institute and/or the University and may also attract penal action by the sources which have thus not been duly cited or from whom due permission has not been sought. This report has not hitherto served as the basis for the award of any degree, diploma, or similar title of any other University..

Ajay N

Place: Kollam

Nithin M

Date: 22 October 2025

Salman A

Sreerag V S

## **ACKNOWLEDGEMENT**

We express our deep sense of gratitude to the Almighty and sincere thanks to all who helped us to complete the project successfully.

We express our sincere gratitude to Dr Sadiq A, Principal, for providing us with all the necessary facilities and support for doing the project.

We are extremely grateful to our project coordinator and guide Dr. Jini Raju, Professor of the Computer Science & Engineering Department, for her excellent guidance, positive criticism, and valuable comments.

We are grateful to Dr. Aneesh G Nath, Head of the Department, Department of Computer Science & Engineering, for providing us support and facilities.

We are grateful to our Senior Advisor Dr. Dimple A Shajahan, Department of Computer Science & Engineering, for his support and valuable comments.

We extend our immense gratitude to all Faculties and Technical Staffs in Department of Computer Science & Engineering, for their help and necessary facilities to complete this work.

Our humble gratitude and heartiest thanks also go to our parents and friends, who have supported and helped us during the course of this work.

Ajay N

Nithin M

Salman A

Sreerag V S

## ABSTRACT

RELIFE is an end-to-end disaster relief and recovery system built on Flutter, with the purpose of simplifying coordination and communication in times of need. The platform offers a single solution for performing important disaster-related operations, such as volunteer sign-up, tracking donations and fundraisers, missing person reports, and uploading videos for situation reporting from the ground. With the public, volunteers, and administrators connected to a single platform, RELIFE facilitates quicker and more structured disaster response. The app is built purely on the Flutter platform, providing a cross-platform user interface that is compatible with Android devices without any hitch. The app, as of now, employs local data storage utilizing lists in handling and presenting information like volunteer information, donation history, and reports. This methodology facilitates uninterrupted functionality and offline use during testing and demonstration periods. RELIFE will be combined in later versions with a back-end database for real-time synchronization and multi-user access. While current disaster management software programs are alert system-specific, RELIFE combines several vital functions under a single integrated platform. It enables community involvement through active participation by volunteers and citizens during emergencies, facilitating transparency and coordination in relief services. The main objective of RELIFE is to give communities a digital tool that strengthens readiness, facilitates rapid response, and facilitates recovery in the event of disasters. By mixing easy-to-use design with real-world features, RELIFE is a digital lifeline that assists communities in remaining informed, connected, and resilient during emergencies.

## TABLE OF CONTENT

DECLARATION .....	3
ACKNOWLEDGEMENT .....	4
ABSTRACT .....	5
TABLE OF CONTENT .....	6
CHAPTER 1: INTRODUCTION .....	8
1.1 Project Overview .....	8
1.2 Problem Statement .....	8
1.3 Objectives.....	9
1.4 Project Significance .....	9
1.5 Scope of the Project .....	10
1.7 Challenges and Limitations .....	10
1.8 Organization Of The Report .....	11
CHAPTER 2: BACKGROUND .....	12
2.1 Introduction .....	12
2.2 Existing Systems and Limitations .....	12
2.3 Significance of Relife in Society .....	13
2.4 Conclusion .....	14
CHAPTER 3: DESIGN METHODOLOGY .....	15
3.1 Class Diagram .....	15
3.2. Use Case Diagram .....	21
3.3 Flowchart .....	25
3.4 Database Design .....	28
3.5 Wireframe Model .....	29
CHAPTER 4: TOOLS AND TECHNOLOGIES USED .....	35
4.1 Implementation Tools .....	35
4.2 Design Tools .....	37
4.3 Documentation Tools .....	37
4.4 Version Control .....	38
CHAPTER 5: IMPLEMENTATION AND RESULTS .....	39

5.1 Implementation .....	39
5.2 Results .....	40
5.3 Screenshots of Results .....	42
CHAPTER 6: TIMELINE AND TASK DISTRIBUTION .....	48
CHAPTER 7: COST AND BUDGET ESTIMATION.....	50
CONCLUSION .....	52
REFERENCES .....	53
CHAPTER 7: APPENDIX .....	54
7.1 Appendix I: Vision, Mission and Program Educational Objectives (PEOs) .....	55
7.2 Appendix I: Program Outcomes .....	55
7.3 Appendix II: Course Outcomes (COs) .....	56
7.4 Appendix III: Fulfillment of Programme Outcomes (POs) .....	57

## CHAPTER 1: INTRODUCTION

### 1.1 Project Overview

RELIFE is a disaster response and recovery system aimed at enhancing communication, coordination, and response in emergencies. The platform integrates citizens, volunteers, and authorities into a common platform that includes verified alerts, volunteer registration, tracking of donations, and real-time evacuation route information. In contrast to other applications that aim to notify only with alerts, RELIFE incorporates several modules to facilitate timely support and transparency.

The platform enables people to act responsibly and collectively—volunteers can sign up and get verified assignments, administrators can track ongoing activities, and impacted users can easily find shelters, emergency contacts, and safe evacuation paths. Overall, the platform enables communities to act faster, stay aware, and recover well when disasters hit.

### 1.2 Problem Statement

During disasters, communities face several major challenges that affect timely response and recovery:

- *Lack of Centralized Coordination:* Communication between the public, volunteers, and authorities is often unorganized, leading to delays in rescue and relief efforts.
- *Spread of Misinformation:* Unverified news, images, and videos spread quickly on social media, creating panic and confusion among the public
- *Poor Accessibility of Reliable Information:* People often struggle to find trusted sources for updates, helplines, and shelter locations during emergencies.
- *Inefficient Resource Management:* There is no proper system to track volunteer activities, donations, or relief fund usage transparently.

RELIFE addresses these issues by offering a unified platform that ensures effective communication, reliable information sharing, and organized disaster management for faster and safer recovery.

## 1.3 Objectives

The primary objectives of the RELIFE platform include:

- *Improving Disaster Coordination:* Establish a centralized system to connect citizens, volunteers, and authorities for organized and timely disaster response.
- *Ensuring Reliable Information Flow:* Provide verified alerts, news updates, and visual reports to prevent the spread of misinformation during emergencies.
- *Facilitating Volunteer and Donation Management:* Enable efficient registration, tracking, and allocation of volunteers and donations to areas in need.
- *Supporting Rescue and Recovery Efforts:* Offer real-time evacuation routes, helpline access, and resource availability to guide affected individuals safely.
- *Promoting Transparency and Trust:* Build a secure and user-friendly platform that enhances accountability and encourages active community participation during disasters.

## 1.4 Project Significance

The RELIFE platform also helps to enhance disaster response and recovery systems by using technology-powered coordination.

- *Improves Disaster Response:* Links citizens, volunteers, and authorities through a unified platform for quicker and coordinated action in times of emergency.
- *Avoids Misinformation:* Provides assured delivery of credible alerts and updates, minimizing fear and confusion due to false news.
- *Fosters Transparency:* Monitors volunteer activities, contributions, and distribution of resources to ensure accountability and build trust.
- *Enhances Coordination:* Coordinates different roles to facilitate timely decision-making and effective relief operations.
- *Facilitates Rescue and Recovery:* Offers exposure to real-time evacuation paths, helplines, and vital resources to aid affected persons.

*Empowers Communities:* Engages active involvement, readiness, and resilience through a secure and easy-to-useisaster management system.

## 1.5 Scope of the Project

RELIFE is designed to enhance disaster management through enhanced coordination, communication, and recovery during emergencies. The project scope comprises:

- *Geographic Target*: Disaster-prone areas and scalable for both local and state-level.
- *User Roles*: Three primary roles — Public Users, Volunteers, and Administrators — are supported for streamlined coordination.
- *Core Features*: Volunteer registration, donation and fundraiser tracking, missing person report, and video upload panel for ground reports.
- *Technology Stack*: Built with Flutter for front-end and MongoDB for database functionality.
- *Integration*: Utilizes Google Maps API for evacuation pathways and location-based support.
- *Usability*: Provides a simple-to-use interface which is even accessible under low network conditions to ensure seamless communication during emergencies.

## 1.7 Challenges and Limitations

Some challenges faced during the development RELIFE include:

- *Data Security and Privacy*: Ensuring the safe handling of user information such as volunteer details, donations, and reports is a major concern.
- *Accurate Location Tracking*: Integrating Google Maps for evacuation routes and real-time updates requires stable internet connectivity and accurate GPS data.
- *Video Upload and Storage*: Managing and storing large video files uploaded by users can affect performance and require efficient database management.
- *Network Dependency*: Many features depend on internet access, which may limit usability in severely affected or low-connectivity areas.
- *User Engagement*: Encouraging users and volunteers to actively use the platform during non-disaster periods may require awareness and motivation campaigns.

## 1.8 Organization Of The Report

This report is organized as follows:

- *Chapter 1: Introduction*, Provides an overview of the RELIFE project, including its objectives, scope, significance, and overall structure.
- *Chapter 2: Background*, Discusses existing disaster management systems, their limitations, and the motivation behind developing RELIFE.
- *Chapter 3: Design Methodology*, Explains the system architecture, data flow, and working model of the platform.
- *Chapter 4: Tools and Technologies Used*, Describes the tools, frameworks, and technologies utilized
- *Chapter 5: Implementation and Results*, Screenshot of the results and a brief writeup understand the flow of the process shown in the screenshot.
- *Chapter 6: Timeline and Task Distribution*, workload distribution and setting deadlines for pleasant project completion.
- *Chapter 7: Appendix*, Programme and Course Outcomes.

## CHAPTER 2: BACKGROUND

### 2.1 Introduction

Natural calamities like floods, earthquakes, and landslides tend to inflict large-scale damage, disrupting communication, transport, and access to basic services. In most cases, the absence of a common platform for coordination among citizens, volunteers, and authorities leads to slow response and inefficient relief activities. Citizens are unable to find missing people, request assistance, or get real-time information during such peak moments.

RELIFE seeks to overcome these issues through a centralized disaster management platform that brings together impacted people, volunteers, and administrators. The platform allows for missing persons reporting, donation tracking, on-ground updates through video posting, and access to evacuation routes through Google Maps integration. Through enhanced communication and coordination, RELIFE ensures more rapid and better-organized disaster response, reducing confusion and lives lost.

With its technology-based solution, RELIFE overcomes the shortcomings of current disaster management systems and helps create a secure, more resilient community that can respond effectively to crises.

### 2.2 Existing Systems and Limitations

Disaster response and coordination now depend on individual platforms and government apps that generally operate in isolation. Though the systems deliver valuable services like alerts and volunteer registration, they typically do not have integration, real-time collaboration, and user-generated reporting capabilities. Consequently, communication gaps and misinformation may slow down rescue and relief efforts in the event of emergencies.

- **AmritaKripa App (Amrita University):** Designed to assist disaster relief operations in natural calamities, this application has volunteer registration and camp information

modules. It does not have the modules for tracking donations, missing persons report, and video-based situational updates.

- **Sachet App (NDMA):** A government-designed platform concentrated on disaster alerts, awareness, and preparedness. Though capable of sending alerts, it lacks volunteer management, fundraising, and live reporting features.
- **Other Alert Systems:** Numerous apps and websites provide early alerts but lack tools for public involvement, field reporting, or open coordination among rescue teams.

These currently available systems point to the need for a comprehensive, interactive, and community-based disaster management solution. RELIFE fills this gap by integrating volunteer organization, donation tracing, missing individual reporting, and situational information into one accessible system that can be used both by users and the authorities in the event of a crisis.

### 2.3 Significance of RELIFE in Society

- **Enhancing Disaster Response and Coordination:** The greatest strength of RELIFE is its capability to unify citizens, volunteers, and officials on a single, centralized interface. In the event of disasters, coordination tends to get fragmented across different channels, resulting in delays and confusion. RELIFE simplifies communication through the process of allowing volunteers to sign up, administrators to control operations, and the public to post real-time updates, enhancing overall disaster response and recovery efficiency.
- **Minimizing Misinformation and Building Trust:** False news, doctored videos, and unsubstantiated updates are common occurrences during crises, inducing panic and misinformation. RELIFE reduces the possibility of such instances by enabling users to upload ground-level data that can be monitored and authenticated by official staff prior to being made available to the public. This helps to ensure individuals are given accurate and credible information at times when it is most valuable, hence building community confidence and security.
- **Enabling Communities in Emergencies:** RELIFE empowers citizens to play an active part in disaster management as opposed to being mere passive recipients of assistance. With functionalities such as missing person reports, tracking of donations, and volunteer coordination, the platform empowers individuals to make a direct contribution towards relief and recovery operations. This enhances community responsibility and resilience during emergencies.

- **Facilitating Transparency and Resource Utilization:** By monitoring donations and volunteer engagement, RELIFE supports equity and openness in the allocation of resources. It ensures relief assistance goes to the most vulnerable individuals, eliminating duplication and abuse of relief services.

## 2.4 Conclusion

The shortcomings of available disaster management platforms call for an integrated and community-driven solution such as RELIFE. In contrast to conventional systems that merely send out alerts or elementary information, RELIFE provides a converged approach that unites citizens, volunteers, and authorities in one platform. Through facilitating volunteer registration, donation tracking, missing person reporting, and video-based situational reports, it facilitates coordination and proper disaster response in a timely manner.

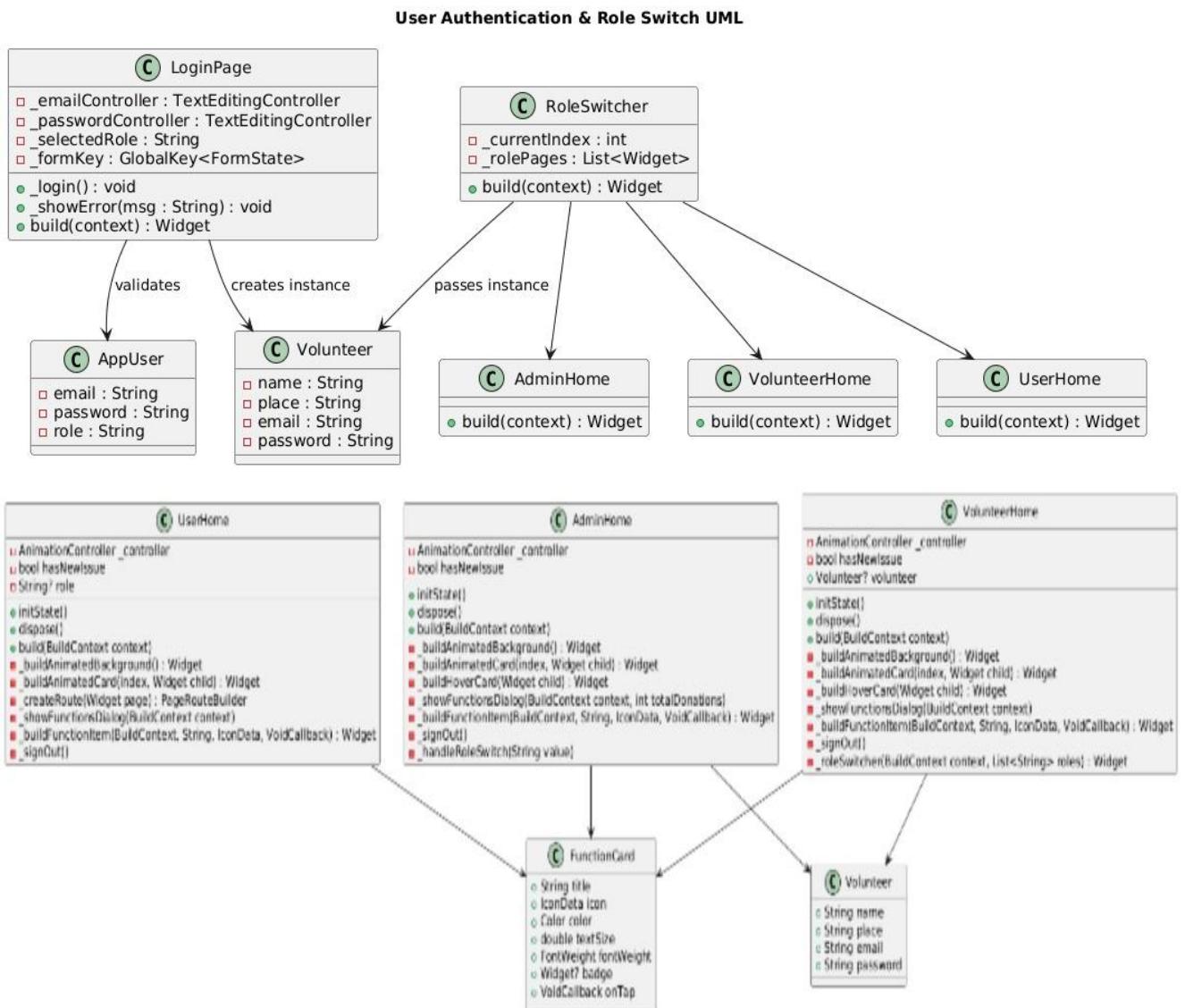
By emphasizing real-time collaboration and openness, RELIFE helps play an important role in minimizing misinformation, enhancing communication, and enhancing community resilience. Not only does this project aid quicker recovery in emergency situations, but it also enables people to actively contribute towards relief efforts. In general, RELIFE helps create a safer, more structured, and technologically empowered society ready to tackle disasters efficiently

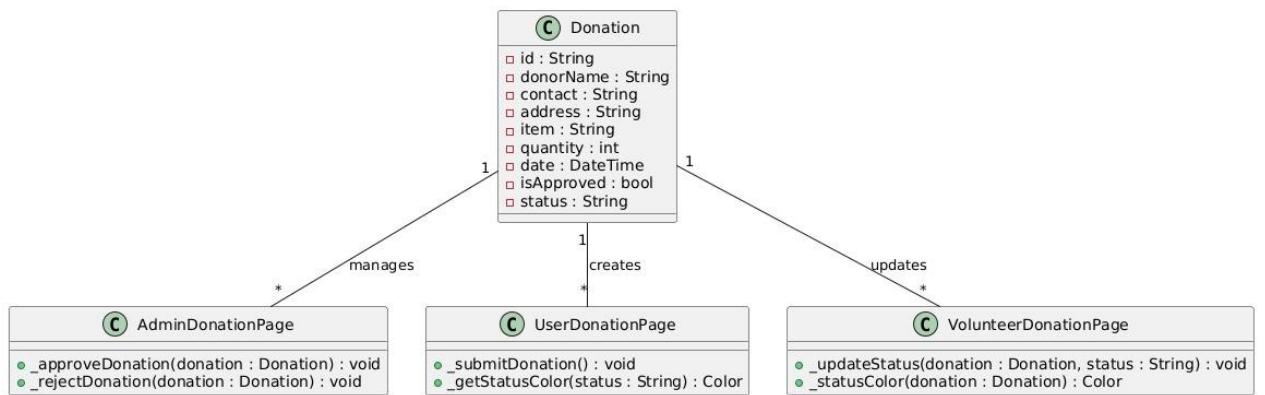
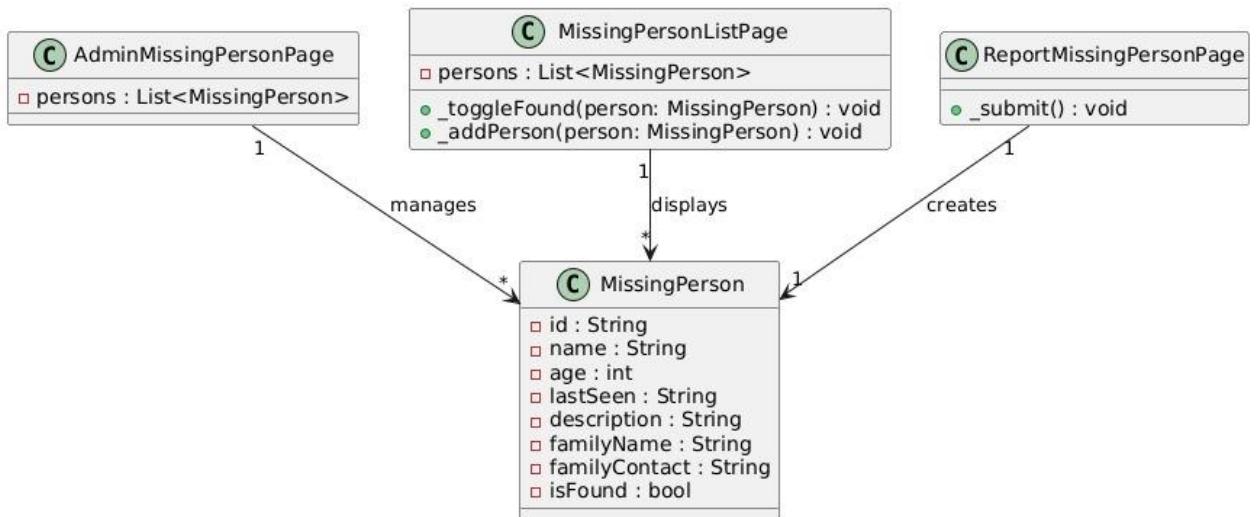
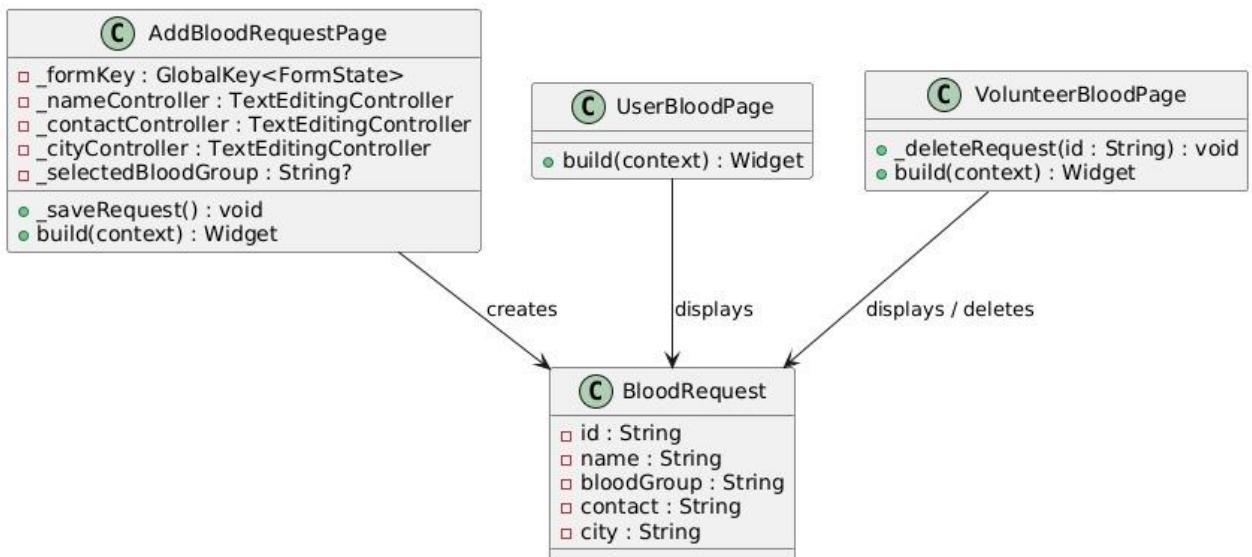
## CHAPTER 3:

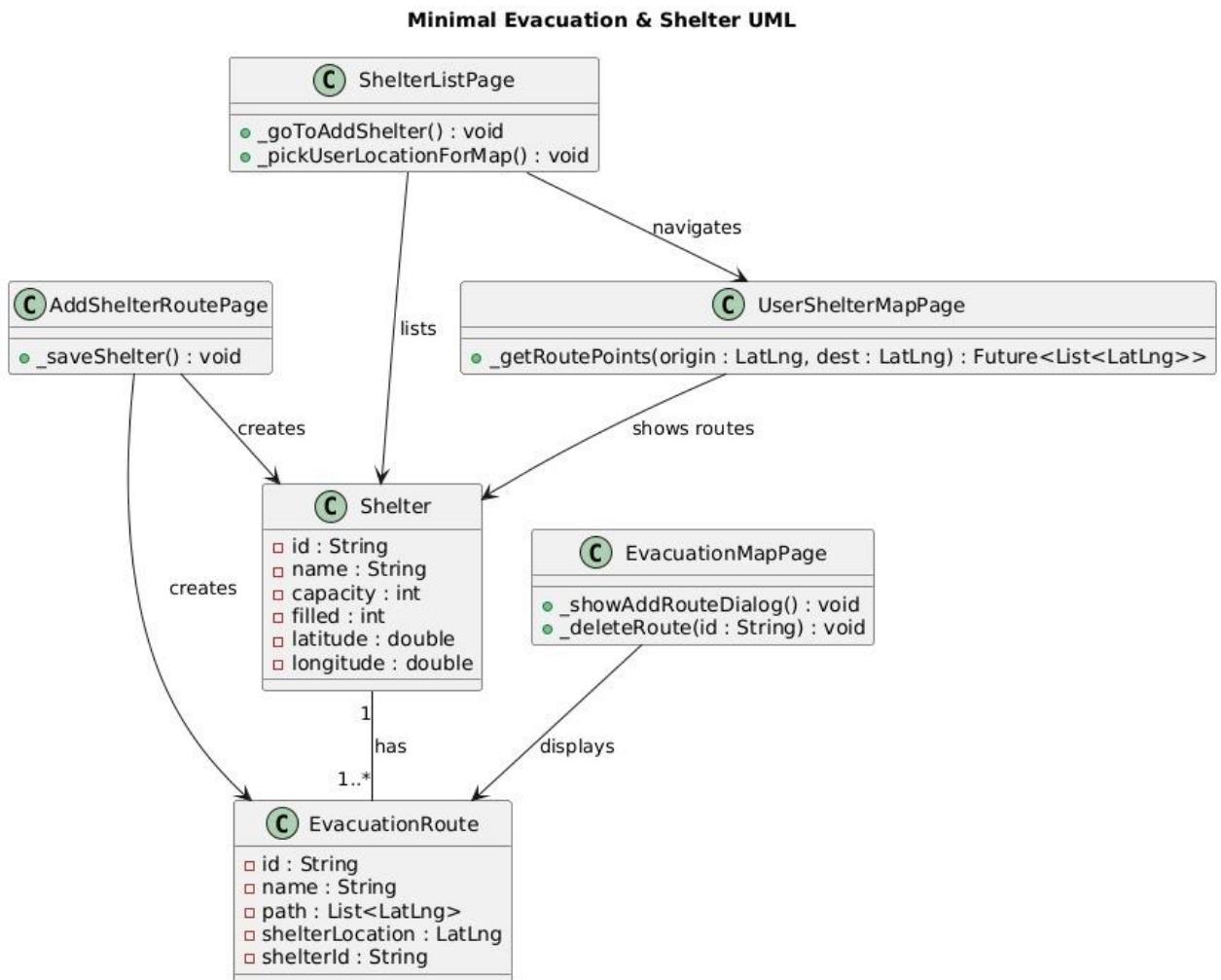
### DESIGN METHODOLOGY

#### 3.1 Class Diagram

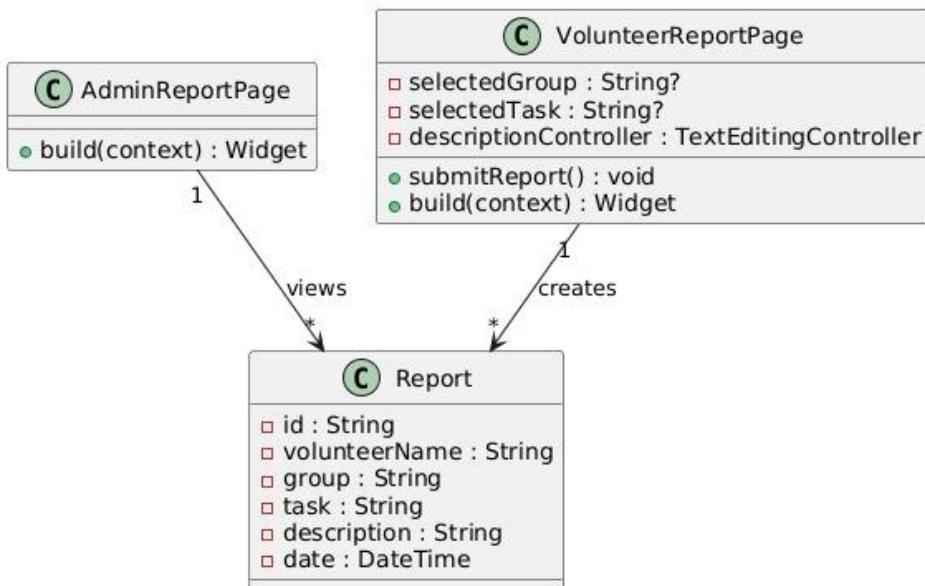
The RELIFE class diagram demonstrates the fundamental organization of the disaster management system, emphasizing relationships and collaborations between its principal classes. This design depicts the system's fundamental elements and their roles, enveloping each module's primary attributes and procedures. It is depicted in Figure 3.1 below.



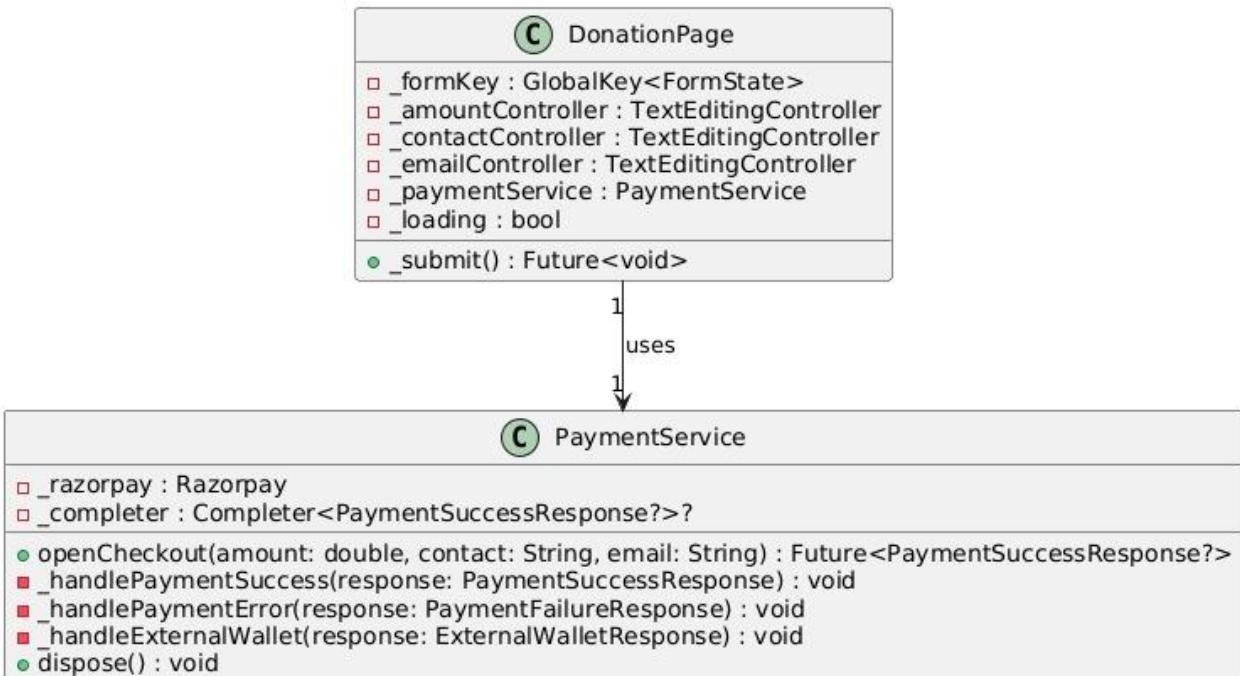
**Donation System UML****Missing Person Management UML****Blood Donation Request UML**



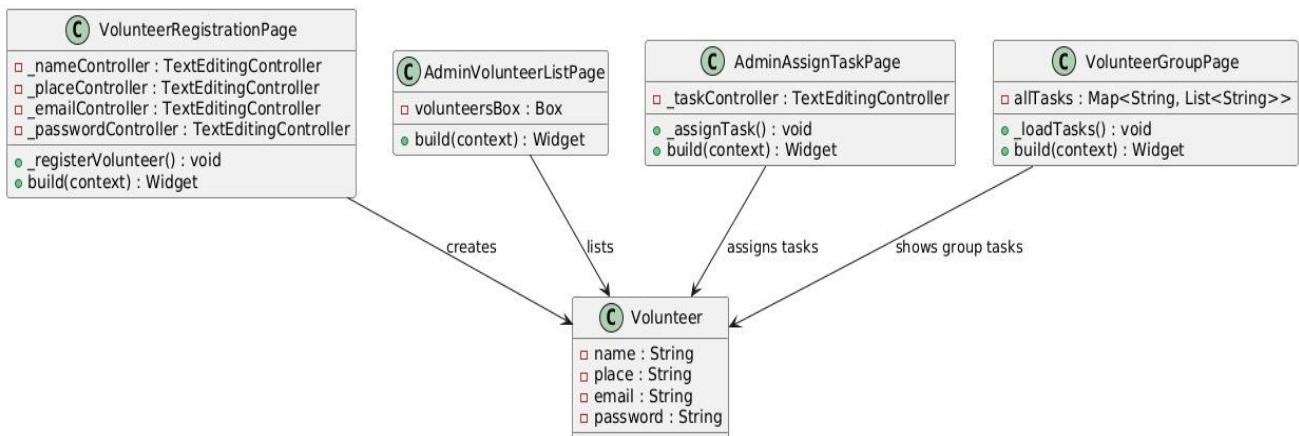
### Volunteer Report Management UML



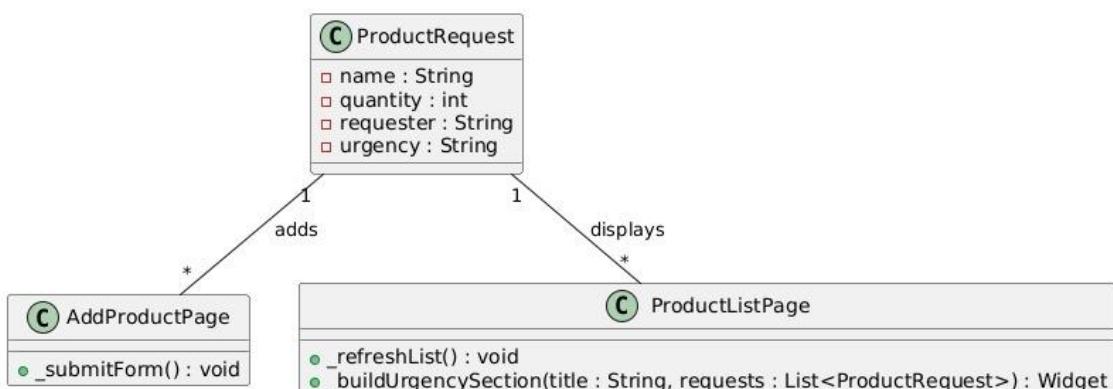
### Donation & Payment System UML



### Volunteer Management UML



### Product Request System UML



*Diagram:* Include a class diagram displaying the main classes in RE-LIFE, such as User, Volunteer, Donation, Shelter, MissingPerson, BloodRequest, and PaymentService.

### Detailed Descriptions:

- *User Class:* Holds information about public users like name, email, contact, and location. Users can donate, report missing individuals, or see shelters through the app.
- *Volunteer Class:* Holds information regarding volunteers such as name, email, duty allocated, and location. Volunteers can process donation requests, report field updates, and aid during emergencies.
- *Donation Class:* Holds donation information like donor name, item or value, contact details, status, and approval.
- *Shelter Class:* Handles shelter information such as shelter name, capacity, coordinates, and availability status.
- *MissingPerson Class:* Holds details such as name, age, last location seen, description, and family contact details, with methods to report and update found status.
- *BloodRequest Class:* Models volunteer-registered blood requests such as requester information, blood group, city, and contact number, enabling users to respond as donors.
- *PaymentService Class:* Facilitates secure donations and fundraiser payment transactions through payment gateways like Razorpay, handling payment success and failure responses

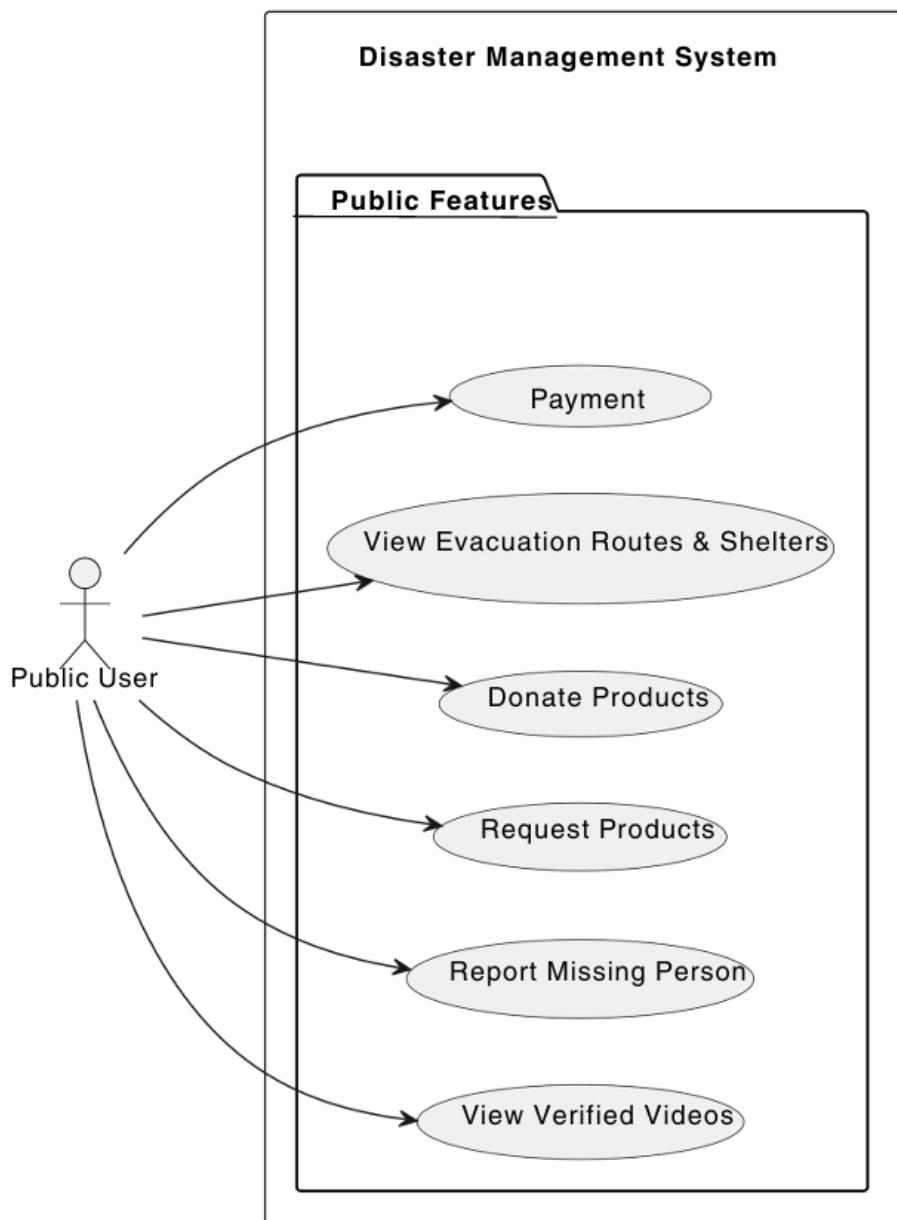
### Relationship

- **User and Donation:**
  - *Type of Relationship:* One-to-Many (1.\*)
  - *Explanation:* A user may have multiple donations done by him/her, in the form of product donation or financial aid. One user is associated with every donation record.
- **Volunteer and Donation:**
  - *Type of Relationship:* One-to-Many (1.\*)
  - *Explanation:* Multiple donation requests and updates can be handled by a volunteer. One volunteer is associated with every donation record for verification or approval.
- **Volunteer and MissingPerson:**
  - *Type of Relationship:* One-to-Many (1.\*)
  - *Explanation:* A volunteer can update or report multiple missing person cases, whereas a single missing person entry is managed by a particular volunteer.

- **User and BloodRequest:**
  - *Relationship Type:* One-to-Many (1.\*)
  - *Explanation:* Multiple blood requests are responded to by a user, whereas every blood request is created and updated by a volunteer.
- **Volunteer and Shelter:**
  - Relationship Type: One-to-Many (1.\*)
  - Explanation: Multiple shelters are added or managed by a volunteer, whereas one responsible volunteer maintains each shelter record.
- **Shelter and EvacuationRoute:**
  - Relationship Type: One-to-Many (1.\*)
  - Explanation: Multiple evacuation routes can point to a single shelter, but one route points to a single shelter only.
- **PaymentService and Donation:**
  - Relationship Type: One-to-One (1.1)
  - Explanation: Each payment service instance processes a single donation transaction.  
This provides secure and traceable handling of every donation.

### 3.2. Use Case Diagram

The RE-LIFE use case diagram provides a overview of the main functionalities and interactions among the public users, volunteers, and administrators , as shown in Figure 3.2, and its detailed description is given below.



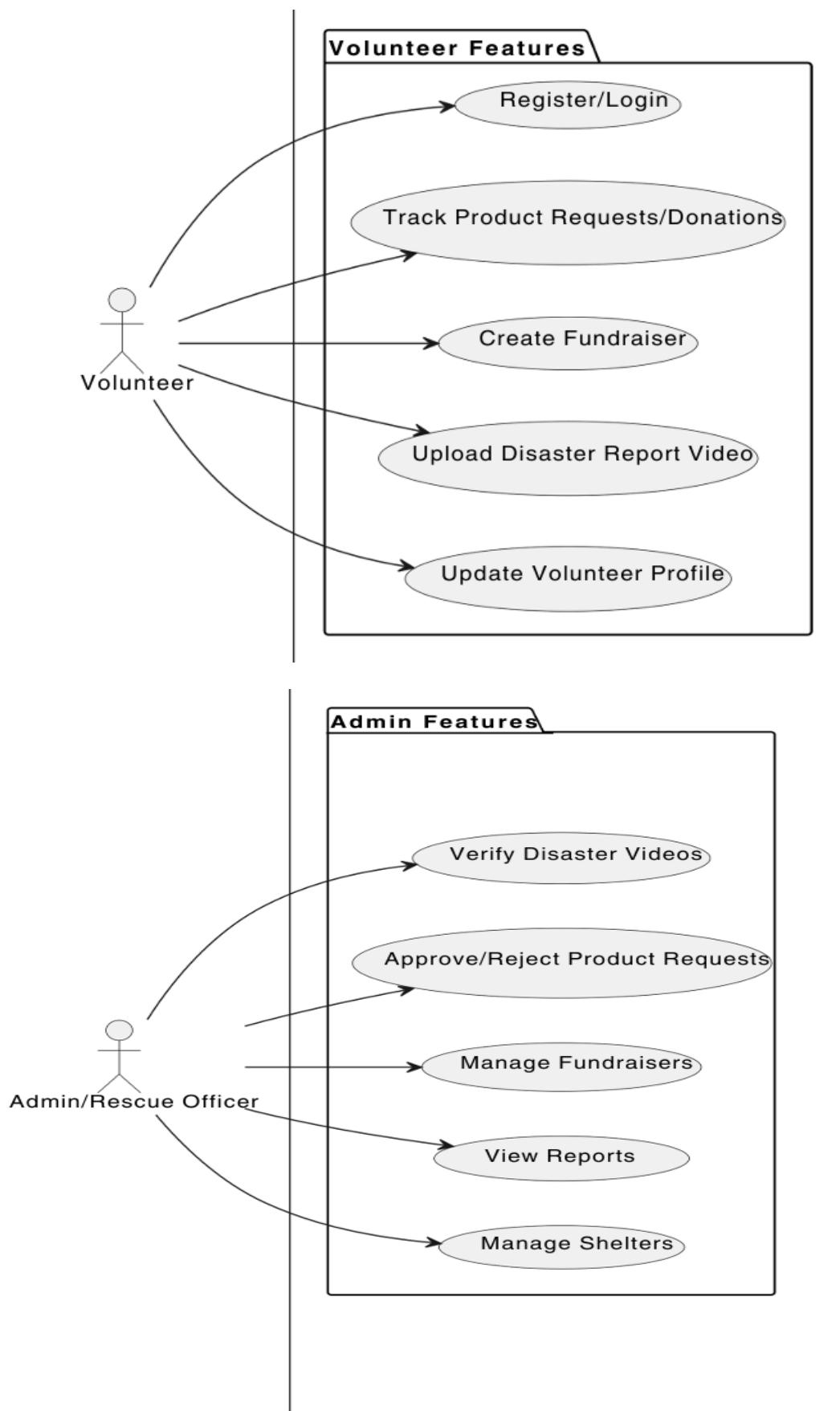


Figure 3.2

**Actors:**

- *Public User*: Represents individuals affected by disasters who can report missing persons, view updates, and access evacuation routes.
- *Volunteer*: Represents registered users who assist in relief activities and manage their availability through the platform.
- *Administrator*: Represents authorized personnel responsible for verifying reports, managing donations, and overseeing overall system operations.

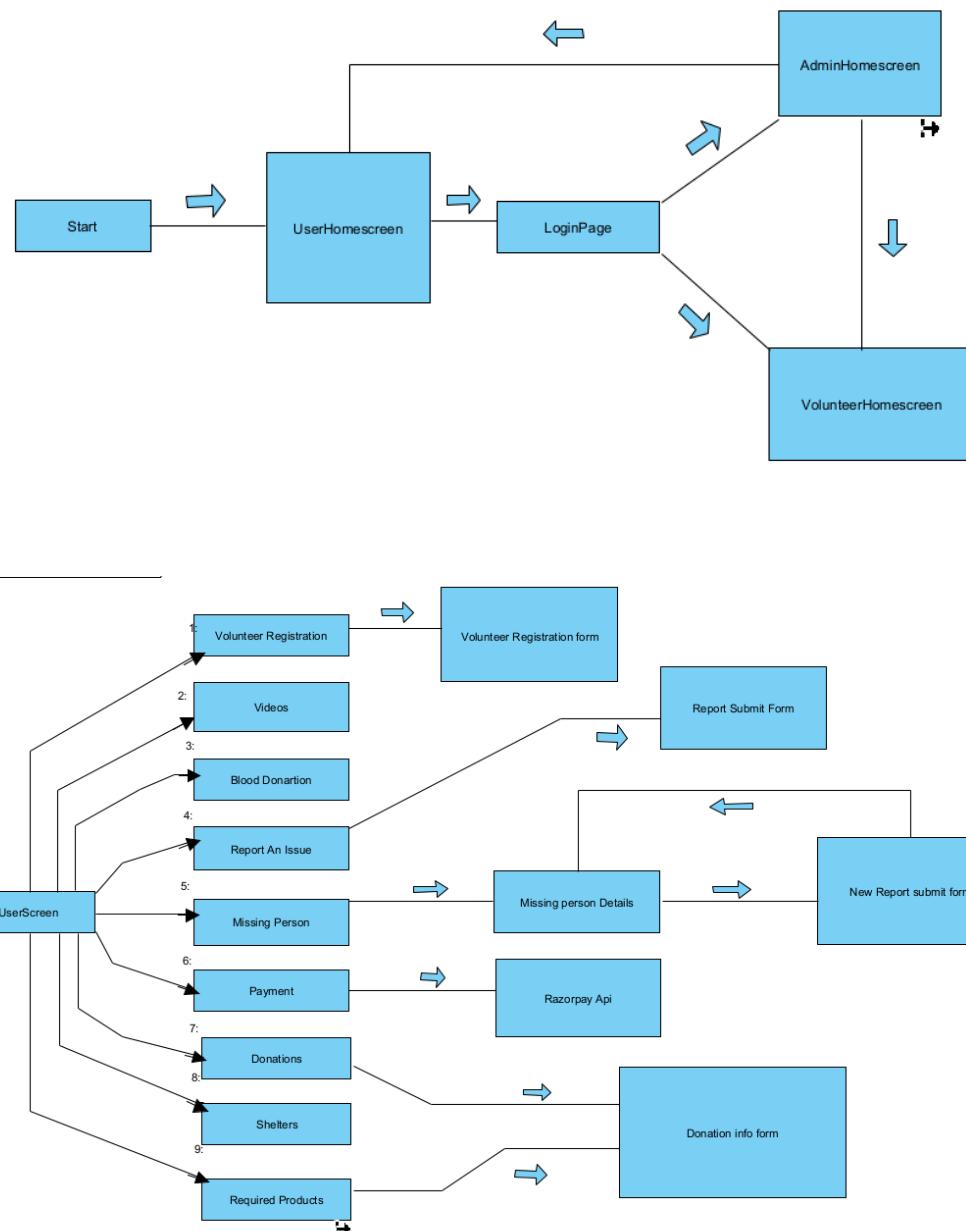
**Use Cases and Relationships:**

- Login / Register (Public User, Volunteer, Admin): Enables all users to register or login into the system in order to use their respective functionalities.
- Relationships: Includes relationship with Manage Profile, as users may change personal information following registration.
- Missing Person Report (Public User): Enables users to report missing persons by giving the critical information like name, age, and place last seen.
- Relationships: Maintains an extend relationship with View Reports so that users can monitor updates concerning their submitted reports.
- Upload Disaster Report Video (Public User, Volunteer): Enables volunteers or users to upload videos of actual disaster situations to create awareness and provide relief.
- Relationships: Maintains an include relationship with View Verified Videos, as uploaded videos can eventually be viewed once verified by the admin.
- Donate Products (Public User): Allows users to donate relief items and community essentials.
- Relationships: Extends a relation with Track Product Requests/Donations so that users can track the donation status.
- Request Products (Public User): Facilitates users to request necessary relief or supply items in case of disasters.
- Relationships: Includes a relation with Approve/Reject Product Requests where the admin approves and processes requests.

- Create Fundraiser (Public User, Volunteer): Allows users or volunteers to initiate a fundraiser campaign for certain disaster-related purposes.
- Relationships: Includes a relationship with Manage Fundraisers, as admins maintain and track all live fundraisers.
- View Evacuation Routes & Shelters (Public User, Volunteer): Offers map-based views of safe evacuation routes and surrounding shelters through Google Maps integration.
- Update Volunteer Profile (Volunteer): Allows volunteers to update personal information, availability, and relief zone assignments.
- Manage Shelters (Admin/Rescue Officer): Allows administrators to add, edit, or delete shelter addresses and their respective information.
- Relationships: Has an include relationship with View Evacuation Routes & Shelters since updates to it impact the data made available to users.
- Verify Disaster Videos (Admin/Rescue Officer): Enables admins to inspect and validate submitted videos to ensure there is no cyber misinformation.
- Relationships: Has an extend relationship with View Verified Videos to only show approved videos to the public.
- Approve/Reject Product Requests (Admin/Rescue Officer): Allows the admin to control product requests made by the users.
- Relationships: Includes relationship with Track Product Requests/Donations, as tracking is reliant on approval status.
- •\tManage Fundraisers (Admin/Rescue Officer): Allows admins to view, approve, or close fundraisers to provide authenticity and transparency.
- •\tTrack Product Requests/Donations (Volunteer, Admin): Allows volunteers and admins to track donation and product request statuses in real time.
- • Payment (Public User): Manages donations given to fundraisers via integrated payment processing

### 3.3 Flowchart

The RELIFE flowchart illustrates the step-by-step process flow for the primary interactions in the application as shown in Figure 3.3 given below.



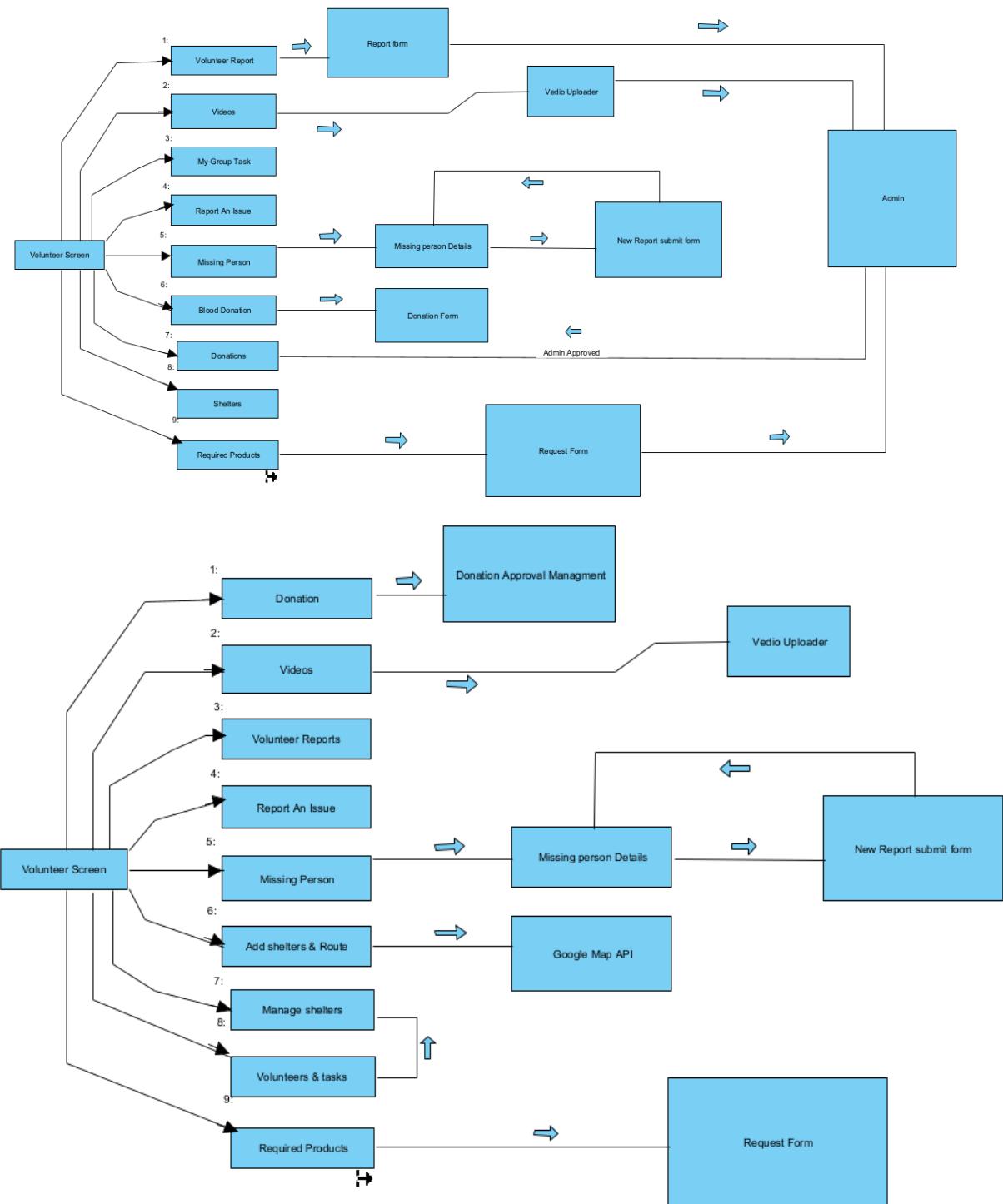


Figure 3.3

**Diagram:** Present flowcharts for key operations, such as the volunteer registration process, missing person reporting, donation and fundraiser workflows, and the disaster video upload process.

### Flow of Execution:

1. *Start:* The process begins when the user opens the RELIFE application.
2. *Login:* The user is prompted to log in or register based on their role Public User, Volunteer, or Admin/Rescue Officer.
  - o If login is successful, the user is redirected to their respective Home Page.
  - o If login fails, the user is prompted to re-enter credentials or reset the password.
3. *Home Page:* After logging in, users can access the main modules available for their role.
4. *Role-Based Navigation:* From the Home Page, the system directs users to different workflows depending on their selected role:
  - o Public User Path
  - o Volunteer Path
  - o Admin/Rescue Officer Path

#### Path 1: Public User

5. Report Missing Person: Users can submit missing person details such as name, age, and last seen location using a form.
6. Create Fundraiser / Donate Products: Users can start a fundraiser or donate essential items for relief efforts.
7. Upload Disaster Video: Allows users to share on-ground visuals from affected areas to assist rescue teams.
8. View Evacuation Routes and Shelters: Users can view safe routes and nearby shelters through Google Maps integration.

#### Path 2: Volunteer

9. Volunteer Registration: Volunteers can register by providing personal details, contact number, and area of availability.
10. Assist in Requests: Volunteers can view and manage donation or product requests assigned to them.
11. Upload Reports: Volunteers can upload videos or situation reports directly from affected locations to help authorities.

#### Path 3: Admin/Rescue Officer

12. Manage Requests: Admins can view and approve or reject product and fundraiser requests.
13. Verify Disaster Videos: Admins verify uploaded videos to ensure authenticity before making them visible to the public.
14. Manage Shelters: Admins can add, edit, or update shelter and evacuation information.
15. Monitor System Activity: Admins oversee overall operations, ensuring data accuracy and proper coordination
16. End: Once tasks are completed, users can log out or return to the home screen to perform additional actions.

### 3.4 Database Design

#### Schema Details:

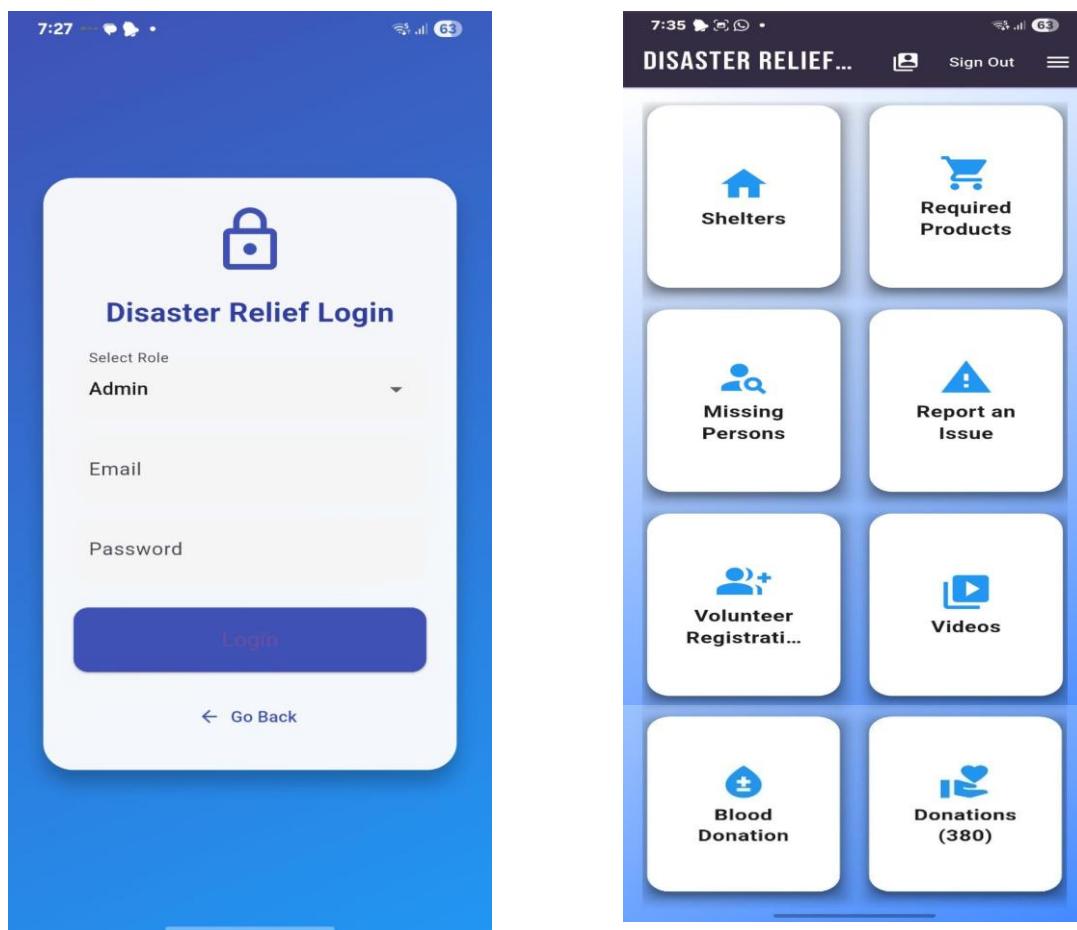
Since the current version of RE-LIFE is developed using Flutter with local list-based storage, all data is temporarily stored within in-app data structures for testing and demonstration.

- *Users*: Stores basic user information such as User ID, Name, Email, Password, and Role (Public User, Volunteer, or Admin).
- *Volunteers*: Contains volunteer-specific details including Volunteer ID, Name, Contact, Availability Area, and Assigned Tasks.
- *Missing Persons*: Records details of missing individuals, including Report ID, Name, Age, Last Seen Location, Contact Person, and Status.
- *Donations*: Tracks all donation-related activities with fields such as Donation ID, Donor Name, Donation Type, Amount/Product Details, and Date.

- **Fundraisers:** Stores information on active fundraisers including Fundraiser ID, Title, Description, Goal Amount, Raised Amount, and Created By.
- **Videos:** Maintains details of uploaded disaster videos, including Video ID, Uploaded By, Description, Date, and Verification Status.
- **Shelters:** Keeps data about available shelters and evacuation routes such as Shelter ID, Location, Capacity, and Contact Details.
- **Requests:** Stores records of Product or Help Requests made by users, along with their Request ID, Request Type, Description, and Approval Status.

### 3.5 Wireframe Model

The RELIFE wireframe model shows the layout and design of the main user interfaces, indicating how public users, volunteers, and administrators engage with the core application features. The wireframes emphasize ensuring that the experience is intuitive and accessible, with simple navigation and clearly structured sections for the registration of volunteers, tracking donations, reporting missing persons, creating fundraisers, and viewing shelters or evacuation routes.



**Shelters**

- Kollam Town Hall Shelter  
Kollam Town | 25/100
- Chinnakada Community Shelter  
Chinnakada, Kollam | 40/80
- Asramam Park Shelter  
Asramam, Kollam | 70/120
- Kilikollur Shelter  
Kilikollur, Kollam | 50/90
- Tangasseri Beach Shelter  
Tangasseri, Kollam | 30/70

**Report Missing Person**

Please provide details carefully.  
This will help in locating the missing person.

Name

# Age

Last Seen Location

Description

Family Name

Family Contact

Submit Report

**Missing Persons**

- Amit Sharma  
Age: 30  
Last seen: Delhi Railway Station  
Missing Mark Found
- Priya Singh  
Age: 25  
Last seen: Mumbai CST  
Missing Mark Found
- Rahul Verma  
Age: 28  
Last seen: Bangalore Bus Stand  
Missing Mark Found
- Neha Gupta + Report Missing Mark Found

**Add Shelter & Route**

Shelter Name

Capacity

Currently Filled

Contact

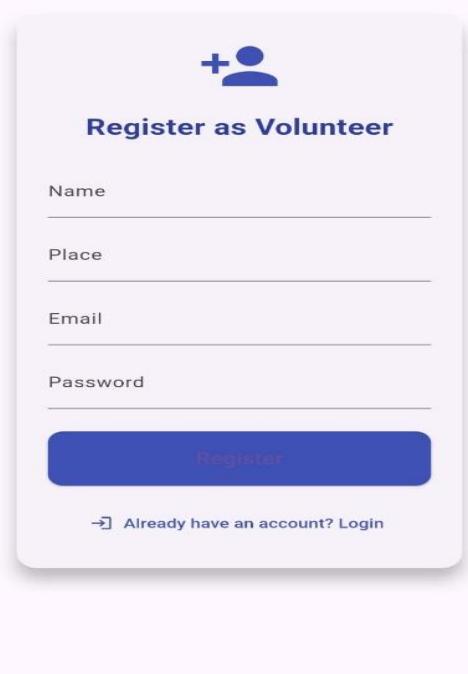
Address

Pick Shelter Location on Map

Save Shelter

8:00 8:00 62

← Volunteer Registration



**Register as Volunteer**

Name

Place

Email

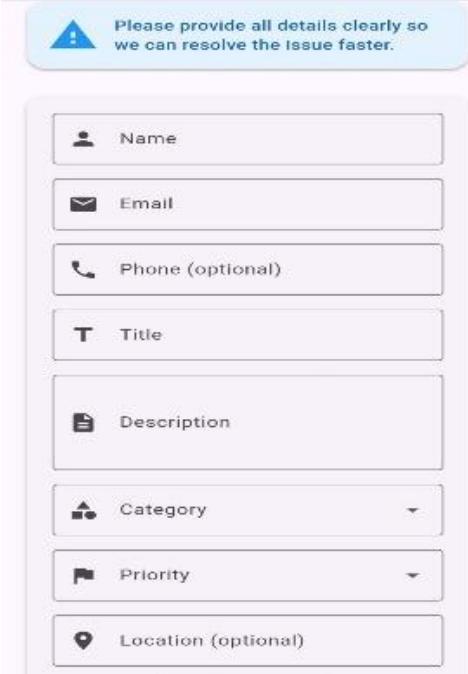
Password

**Register**

→ Already have an account? Login

7:59 7:59 62

← Report an Issue



**Please provide all details clearly so we can resolve the Issue faster.**

**Name**

**Email**

**Phone (optional)**

**Title**

**Description**

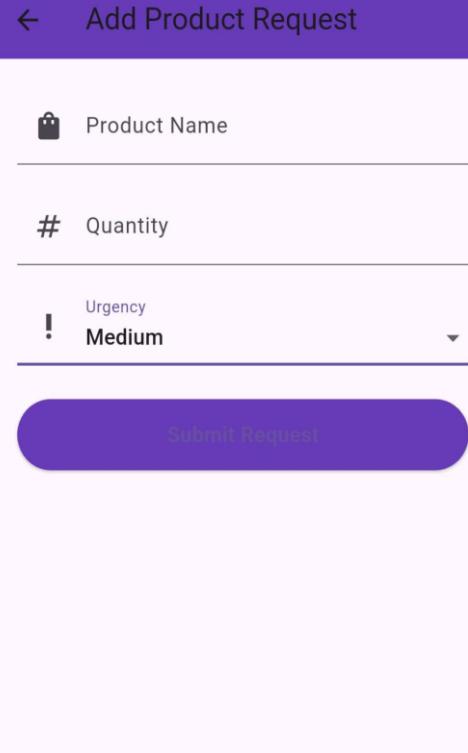
**Category**

**Priority**

**Location (optional)**

8:21 8:21 60

← Add Product Request



**Product Name**

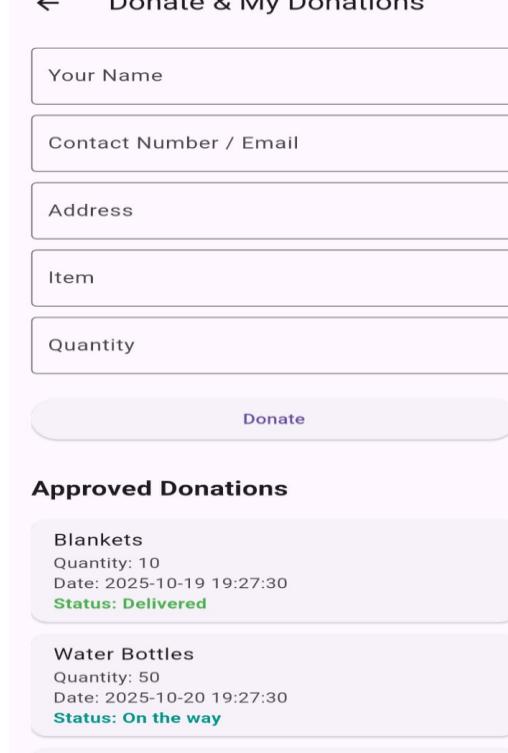
**Quantity**

**Urgency**  
 Medium

**Submit Request**

8:05 8:05 61

← Donate & My Donations



**Your Name**

**Contact Number / Email**

**Address**

**Item**

**Quantity**

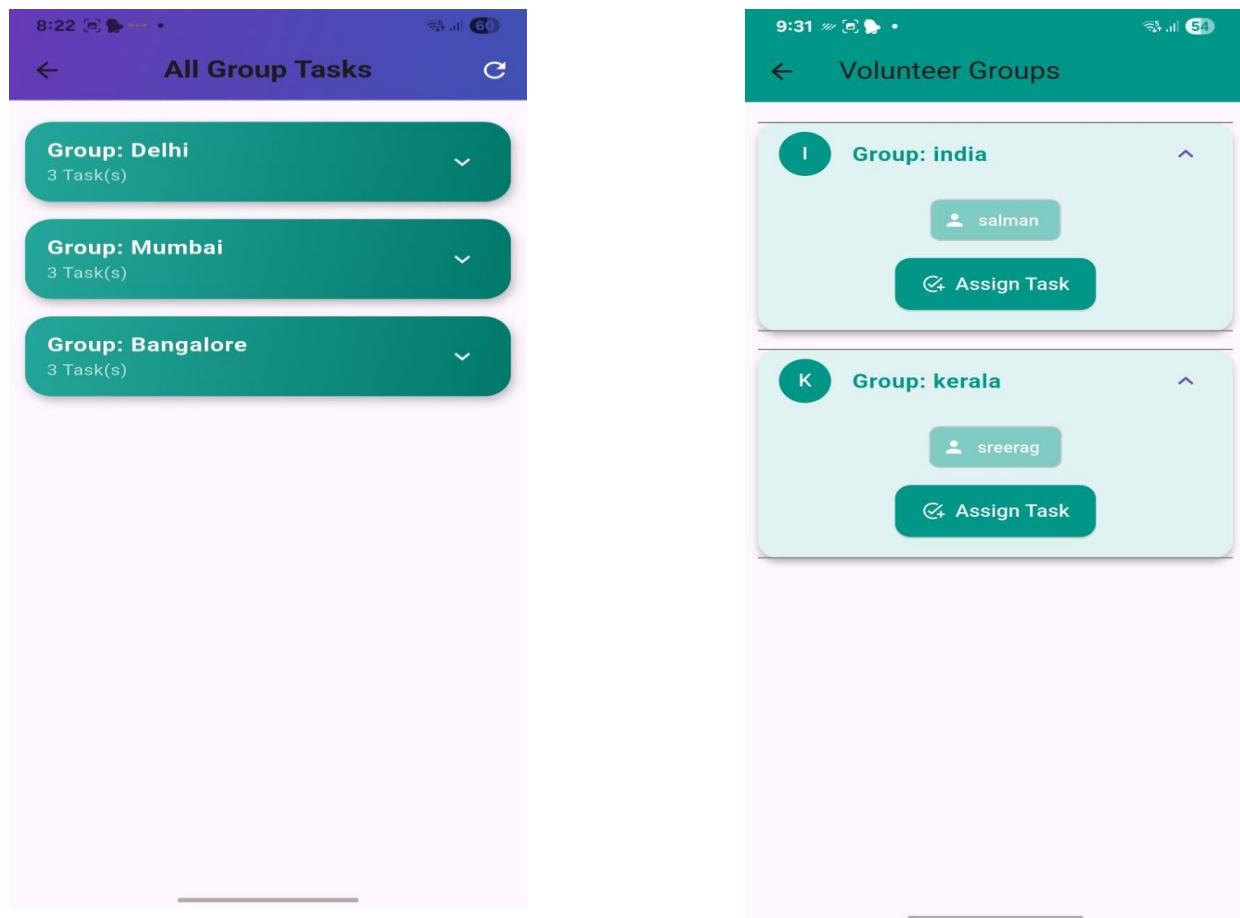
**Donate**

**Approved Donations**

**Blankets**  
 Quantity: 10  
 Date: 2025-10-19 19:27:30  
**Status: Delivered**

**Water Bottles**  
 Quantity: 50  
 Date: 2025-10-20 19:27:30  
**Status: On the way**

**Masks**



The image shows the "Add Blood Request" form. It consists of four input fields: "Name", "Blood Group", "Contact Number", and "City". Below these fields is a large red "Submit" button.

Name
Blood Group
Contact Number
City

Submit

8:27 60

← Submit Volunteer Report

Select Group

Select Task

Report Description

Submit Report

Your Reports:

- 9:32 50
- ← Volunteer Reports
- Delhi (1 reports)
- Amit Sharma - Distribute food packets  
Distributed 50 food packets in sector 5.  
20/10/2025
- Mumbai (1 reports)
- Priya Singh - Help at evacuation center  
Assisted 20 families at the evacuation center.  
21/10/2025
- Bangalore (1 reports)

8:26 60

← Volunteer: Update Status

**Blankets**  
Quantity: 10  
Donor: Alice  
Contact: alice@example.com  
Address: 123 Main St

Delivered

Delivered ▾

**Water Bottles**  
Quantity: 50  
Donor: Bob  
Contact: bob@example.com  
Address: 456 Park Ave

On the way

On the way ▾

**Masks**  
Quantity: 100  
Donor: Diana  
Contact: diana@example.com  
Address: 101 Maple Rd

Approved

Approved ▾

9:32 50

← Admin: Approve Donations

**Canned Food**  
Quantity: 20  
Donor: Charlie  
Contact: charlie@example.com  
Address: 789 Elm St

Approve

Reject

**Gloves**  
Quantity: 200  
Donor: Ethan  
Contact: ethan@example.com  
Address: 202 Oak Ln

Approve

Reject

## CHAPTER 4:

# TOOLS AND TECHNOLOGIES USED

### 4.1 Implementation Tools

- **Flutter**

- *Need:* The primary development environment used to code, design, test, and debug the RELIFE application.
- *Functionality:* Offers a comprehensive platform to develop, design, test, and debug Android apps. The features include a testing emulator, layout editor, as well as a full range of libraries to simplify development.
- **Dart Programming Language**

- *Need :* The main development environment employed to code, design, test, and debug the RE-LIFE application.

*Functionality:* Offers facilities like an emulator for testing, layout editor for UI design and Flutter/Dart plugins for native app integration. It makes it easy to compile and deploy the application on Android devices.

- **Android Studio**

- *Need:* The primary development environment used to code, design, test, and debug the RE-LIFE application.
- *Functionality:* Provides tools such as an emulator for testing, layout editor for UI design, and Flutter/Dart plugins for integrated app development. It simplifies compiling and deploying the app on Android devices.

- **Google Maps API**
  - *Need:* For the display of evacuation routes and location-based support during disasters.
- *Functionality:* To enable users to see nearby shelters or safe paths through the integration of interactive map services within the application interface.
- **Payment Gateway Integration (Razorpay )**
  - *Need:* Introduced to provide secure and open financial transactions for donations and fundraisers in the RE-LIFE app.
  - *Functionality:* Allows users to make instant payments through multiple methods (UPI, card, or wallet) with real-time confirmation. It ensures transaction safety through encryption and verification protocols.
- **Visual Studio Code (Optional)**
  - *Need:* Alternative code editor used for lightweight development and quick UI testing.
- *Functionality:* To facilitate instant payment via various channels (UPI, card, or wallet) with real-time verification. It provides transaction security through encryption and verification systems.
- **Local Data Storage (List-Based)**
  - *Need:* Utilized for storing user data like volunteer information, donation records, and missing person reports temporarily within the application.
  - *Functionality:* As the backend is not yet implemented, data is stored using in-app lists for demonstration and testing. Future versions will include a database such as Firebase or MongoDB for real-time storage and syncing.

## 4.2 Design Tools

- **Figma**
  - *Need:* Prototyping and wireframe design tool for user interface (UI) layout.

- *Functionality:* Allows creation of interactive prototypes, wireframes, and mockups for screens, ensuring that RELIFE has a user-friendly, intuitive interface. Design features like drag-and-drop tools and customizable components are included.

- **Visual Paradigm**

- *Need:* Diagramming tool for design diagrams (e.g., class diagrams, ER diagrams).
- *Functionality:* Used for creating and visualizing various UML diagrams, including class, ER, and flowcharts, which help in planning the system's architecture and understanding relationships among components.

### 4.3 Documentation Tools

- **Microsoft Word**

- *Need:* Documenting project details, requirements, and design methodologies.
- *Functionality:* Offers a flexible, accessible environment for writing, editing, and

formatting documentation. These features enable collaborative writing, which is helpful in tracking edits and contributions.

- **Canva**

- *Need:* Presenting project reports and demonstrations.
- *Functionality:* Allows creating nice-looking slideshows for presenting important features of RELIFE such as system architecture, workflow, and user interface.

### 4.4 Version Control

- **Git & GitHub**

- *Need:* Version control to track changes and collaborative development.

- *Functionality:* Git manages project history by tracking changes to code and GitHub offers a centralized location for sharing and collaborating. Makes it possible for team members to see code, keep versions, and fix conflicts.

## CHAPTER 5:

# IMPLEMENTATION AND RESULTS

### 5.1 Implementation

The design of RELIFE included a number of major stages, namely user interface design, functionality development, and management of local data. The subsequent sections illustrate the major parts of the application:

#### Frontend Development:

- The user interface (UI) of RE-LIFE was created with the Flutter framework, which provided effective performance and a uniform design across devices.
- Various screens were designed to depict various modules like VolunteerRegistration, Donation Tracking, Missing Person Report, and Video Upload Panel.
- The home screen serves as the main hub, enabling users to access various sections with ease.
- All modules feature user input fields and buttons for submitting data, which are temporarily saved in lists for local processing.
- The design favors simplicity and clarity, hence rendering the app friendly and easy to use even for people with little technical expertise..

#### Local Data Handling:

- As backend integration has not been done yet, all data for the application — including volunteer information, donation history, and missing persons' reports — is being stored using lists in the app.
- Users' data input is reflected on the screen immediately, mimicking real-time updates for demonstration purposes.
- This method allows testing and evaluation to be done without the need for a server or internet.

- Future developments will add database integration (e.g., Firebase or MongoDB) for realtime syncing and multi-user support.

### Core Functionalities:

*Volunteer Registration:* Enables users to fill in personal information like name, contact number, and area of availability. Volunteers can register and then later see their information in the app.

- *Donation Tracking:* Allows users to log donation information, such as donor name, amount, and reason. All locally stored entries for reference.
- *Missing Person Report:* Users can report missing persons by inputting information such as name, age, where last seen, and contact details.

• *Video Upload Panel:* Offers a page where users can mimic video uploads for sharing ground reports or incidents during disasters.

• *Fundraiser Module with Razorpay Integration:* Allows users or volunteers to create relief campaigns with information such as title, description, and target amount. Built-in Razorpay Payment Gateway enables secure, real-time donation through UPI, cards, or wallets, making it transparent and safe.

• *Evacuation Route Mapping:* Integrated with Google Maps API, enabling users to see and navigate through safe routes in emergencies.

### Testing and Debugging:

- Testing was done after every implementation phase to validate correct functionality and fluent navigation from screen to screen.
- The application was tested on Android emulators and physical devices with Android Studio to ensure UI responsiveness and local data storage.
- Navigation, input validation, and list data update-related bugs were resolved during testing to enhance app stability.

- On the "big picture" side, RE-LIFE was "prototypically" realized as a proof of concept for an integrated disaster management platform.

## 5.2 Results

The below outcomes were obtained during the implementation phase:

### **Functional Prototype Development:**

The RE-LIFE application was successfully implemented using Flutter, proving certain functionalities like volunteer registration, donation tracking, missing person reporting, and video upload simulation. Every module serves its respective purpose, enabling users to input and display data effectively in the app.

### **Smooth User Interface and Navigation:**

The UI of the app is clean and simple, providing smooth navigation on every page. Users are easily switch between pages like Home, Volunteer Registration, Donations, and Missing Person Reports. The layout and the design create an uncluttered and easy-to-use interface, even for beginners

### **Local Data Management:**

All user input data is saved utilizing lists within the application. This allows details like volunteer information and donation histories to be shown immediately after submission. The local storage approach functioned well in testing and demonstration verifying the internal data-handling logic of the app in the absence of a backend connection.

**Map and Location Features:** Google Maps API integration allows users to see evacuation paths and safety areas in cases of emergencies. The functionality worked well in the testing process, offering precise navigation assistance.

**Performance and Stability:** Testing on Android smartphones and emulators established that the app performs well without crashes or lag. Page transitions, input processing, and data rendering were smooth and error-free throughout the testing period.

### Future Enhancements:

Future updates will include:

- Integration of a backend database (Firebase or MongoDB) for real-time data storage and multi-user access.
- Implementation of user authentication for secure access control.
- Addition of media upload functionality for verified reports and alerts.
- Improvement of the overall UI with animations and notification support.

### 5.3 Screenshots of Results

The following screenshots showcase the user interface and the functionality of the RELIFE application:

- *Login Screen:* Screenshot showing the login interface where users can enter their credentials. The login options differ for Public Users, Volunteers, and Administrators (Figure 5.1). The Volunteer login page can be accessed directly from the Public User login page.

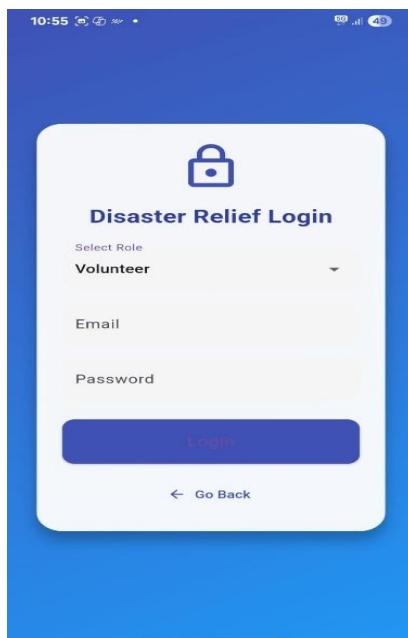


Figure 5.1

- *Volunteer Dashboard:* The volunteer homepage displays available relief tasks, donation updates, and assigned activities. A volunteer can access this dashboard after logging in, as shown below in Figure 5.2.

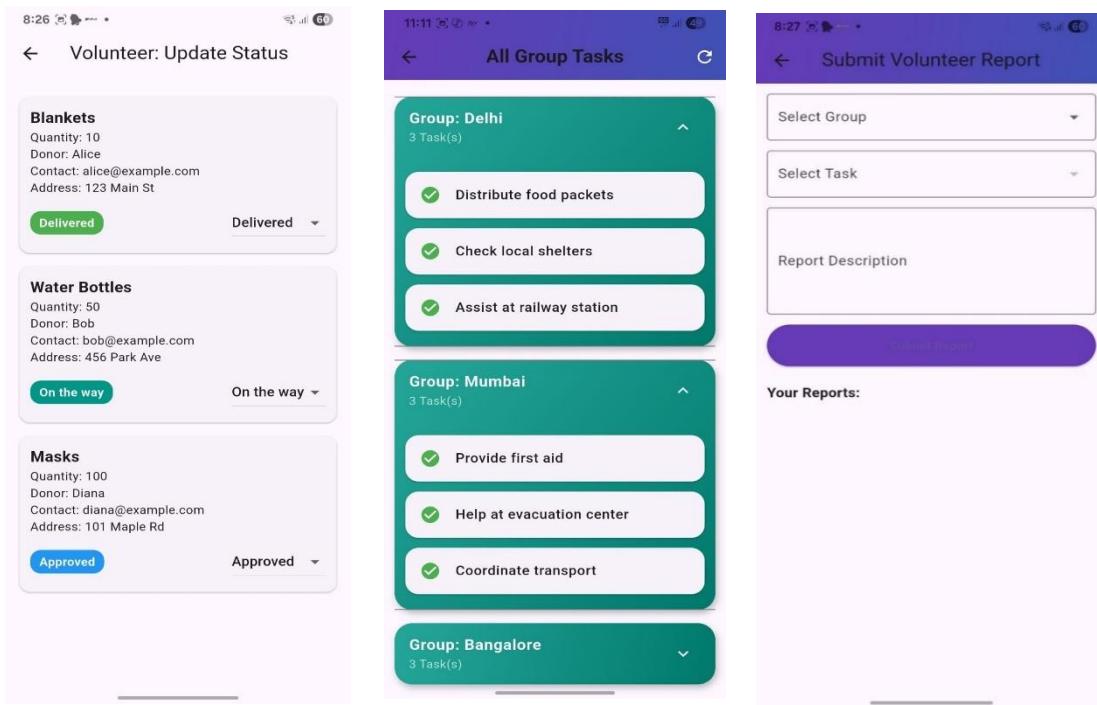


Figure 5.2

- *Home page :* The page where the user reaches upon logging in as shown in Figure 5.3.



Figure 5.3

- *Missing Person:* User can find the list of missing person(Figure 5.4) and a page for add or report the missing person details(Figure 5.5).

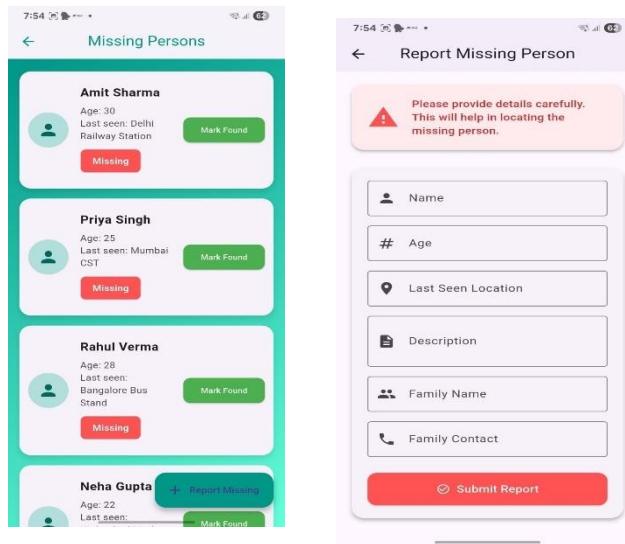


Figure 5.4

Figure 5.5

- *Upload Disaster Report:* A page that allows users or volunteers to upload videos or reports from disaster-affected areas. This feature helps in sharing real-time ground updates with authorities and can be accessed from the Home Page, as shown in Figure 5.6

Figure 5.6

- **Donation Page:** A page where users can contribute by donating essential products or funds for disaster relief(Figure 5.7).Volunteer can request product from user in (Figure 5.8)

**Approvals**

- Blankets  
Quantity: 10  
Date: 2025-10-19 19:27:30  
**Status: Delivered**
- Water Bottles  
Quantity: 50  
Date: 2025-10-20 19:27:30  
**Status: On the way**
- Masks

Figure 5.7

**Product Requests**

- Product Name
- # Quantity
- Urgency
  - Medium

Figure 5.8

- **Blood Donation Page:** A page that allows volunteers to register blood requirements during emergencies (Figure 5.9). Public users can view these requests and donate accordingly (Figure5.10).

Figure 5.9

**Blood Donation Requests**

- John Doe (A+)  
Contact: +91 9876543210  
City: Delhi
- Jane Smith (B-)  
Contact: +91 9123456780  
City: Mumbai

Figure 5.10

- *Shelter Location Page:* The page uses Google Maps integration to help users identify the closest relief centers based on their location. Volunteers and administrators can update shelter information (Figure 5.11).

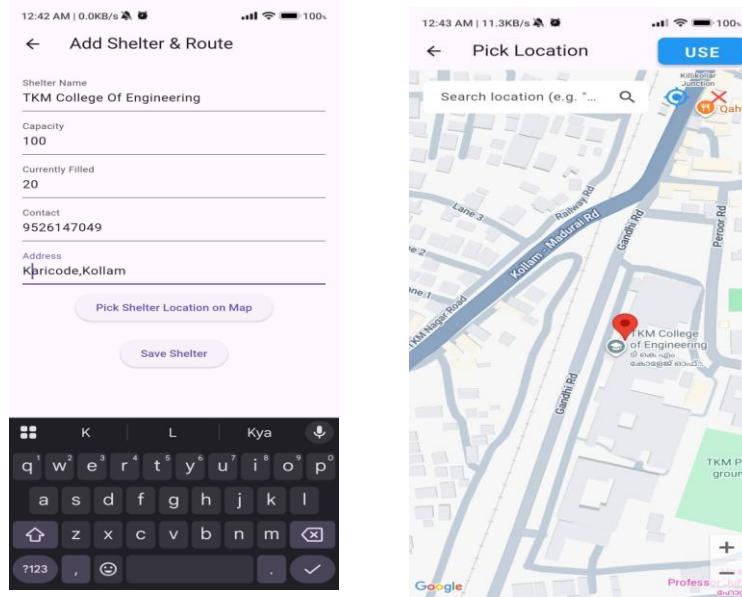


Figure 5.11

- *Search for Available Shelter and Findings Page:* A page that allows users to search for nearby shelters. Users can also view the shelter's location on an interactive Google Map for easy navigation shown in (Figure 5.12).

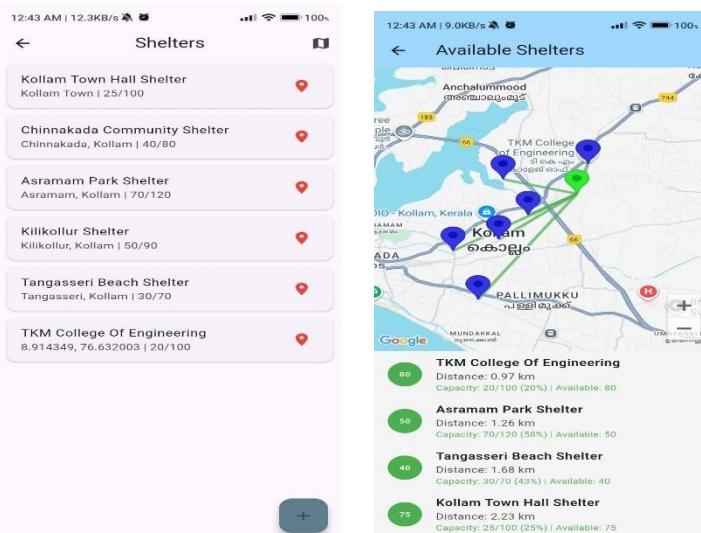


Figure 5.12

- *Payment Gateway Page:* A page that enables users to make secure online payments for donations or fundraiser contributions within the RELIFE application as shown in (Figure 5.13)

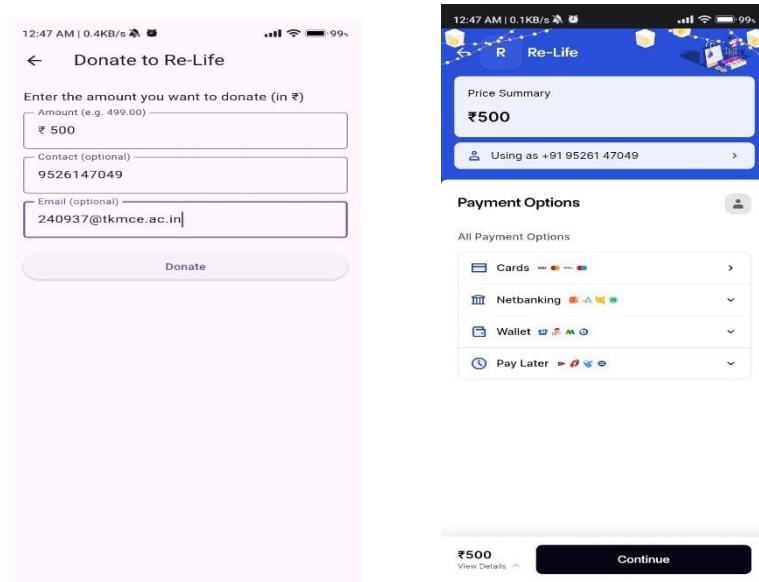


Figure 5.13

## CHAPTER 6:

### TIMELINE AND TASK DISTRIBUTION

#### Timeline:



#### Task Distribution

- *Salman A:*
  - Develop base GUI of the project.
  - Create a project plan or roadmap.
  - Conduct feasibility analysis (technical, financial, time).
  - Major Code Base contributor
- *Sreerag V S:*
  - Implement payment gateway.
  - Ensuring App compatibility and error debugging.
  - Google Maps API implementation

- *Nithin M.*
  - Develop classes and GUI required for Admin
  - Building the presentation
  - Major Code Base contributor
- *Ajay N.*
  - Develop classes and GUI required for User and Volunteer
  - Building project report
  - Review and validate the system design

## CHAPTER 7:

### COST AND BUDGET ESTIMATION

#### 7.1 Introduction

The RELIFE Disaster Relief and Recovery Application was developed with the goal of providing a centralized, user-friendly platform for managing disaster-related activities such as volunteer coordination, donation management, missing person reporting, and shelter tracking. Since the project was created as an academic prototype, the overall development cost was minimal, as most tools used were open-source. However, a future budget estimation has been prepared to evaluate the cost of developing a fully deployable and scalable version of the system.

#### 7.2 Cost Explanation

Category	Description	Estimated Cost (INR)
Software Tools	Flutter SDK, Android Studio, VS Code (Free and Open Source)	₹0
Hardware Resources	Personal laptops and smartphones used for testing and development	₹0
Payment Integration	Razorpay payment gateway setup and testing	₹200
API Services	Google Maps API (Free developer tier)	₹0
Internet & Utilities	Internet connectivity and electricity used during development	₹0
Documentation	Digital documentation and report preparation	₹0
Miscellaneous	Minor operational expenses	₹0

### 7.3 Future Budget Projection

If the RELIFE system is expanded into a fully functional, deployable application with backend services, cloud hosting, and long-term maintenance, the following cost estimation applies:

Category	Description	Estimated Cost (INR)	Future Cost (INR)
Backend Development	Implementation using Node.js / Spring Boot and database integration (MongoDB or Firebase)	₹20,000 – ₹25,000	
Cloud Hosting & Database	Server hosting, Firebase/MongoDB Atlas, and Google Cloud APIs	₹5,000 – ₹10,000 per year	
App Deployment	Google Play Store developer account (one-time)	₹2,500	
UI/UX Enhancement	Improved app interface and cross-device testing	₹5,000	
Security & Payment Setup	Razorpay / UPI verification, secure API handling	₹2,000	
Maintenance & Updates	Bug fixes, feature additions, and server upkeep	₹5,000 – ₹10,000 per year	
Miscellaneous & Contingency	Unexpected costs (resources, device testing, marketing)	₹3,000	

### 7.5 Summary

The current total cost for developing the RELIFE prototype is ₹200, showcasing the efficiency of using open-source and freely available tools. Yet, for a complete, operational deployment with backend integration, hosting, and ongoing maintenance, the estimated budget is between ₹40,000 to ₹55,000.

## CONCLUSION

The RELIFE project effectively fills an essential requirement in disaster management by closing the communication and coordination gap among citizens, volunteers, and authorities. By enabling an easy-to-use and dependable application, the project facilitates well-organized relief operations, transparent donation tracking, and rapid response in emergencies. Through the inclusion of vital features like volunteer registration, fundraiser management, missing person reporting, and video upload for situational reports, RELIFE offers an all-round digital solution for disaster relief and recovery. Technologies such as Flutter were effectively used to develop a scalable, cross-platform system with seamless data handling and interactive performance. Challenges like accurate location mapping, handling video uploads, and ensuring data security were addressed through iterative design, testing, and refinement, resulting in a reliable and user-friendly platform. Feedback and analysis indicate that RELIFE has strong potential to enhance disaster preparedness and response by enabling faster coordination and verified information sharing, helping victims locate assistance while empowering volunteers and organizations to contribute more effectively.

In sum, RELIFE shows how technology can be the key to saving lives, enhancing transparency, and building community resilience in times of disaster. With further improvement and mass adoption, the platform can be a de facto tool in contemporary disaster management, making sure that assistance is always within reach with just a click.

## REFERENCES

1. **AmritaKripa App** – Disaster Relief Application by Amrita University:  
[https://play.google.com/store/apps/details?id=edu.amrita.amrita\\_kripa](https://play.google.com/store/apps/details?id=edu.amrita.amrita_kripa)
2. **Sachet App** – National Disaster Management Authority (NDMA):  
<https://sachet.ndma.gov.in/>
3. **Google Cloud Service:** <https://cloud.google.com/>
4. **RazorPay:** <https://razorpay.com/>
5. **Flutter Documentation:** <https://docs.flutter.dev/>
6. **GeeksforGeeks :** <https://www.geeksforgeeks.org/use-case-diagram/>
7. **YouTube Tutorials** – Flutter App Development & API Integration

## CHAPTER 7: APPENDIX

### 7.1 Appendix I: Vision, Mission and Program Educational Objectives (PEOs)

#### VISION

To be a center of excellence imparting quality education in Computer Science and Engineering and changing students to critical thinkers and lifelong learners with the ability to create environmentally friendly and economically viable solutions for real life problems

#### MISSION

- To provide a strong foundation in Computer Science and Engineering, prepare students for professional career and higher education, and inculcate research interest.
- To be abreast of the technological advances in a rapidly changing world.
- To impart skills to come up with socially acceptable solutions to real world problems, upholding ethical values.

**PROGRAMME EDUCATIONAL OBJECTIVES(PEOs)**

**PEO 1:** Excel in professional career by acquiring knowledge in mathematics, science and engineering and apply the knowledge in the design of hardware and software solution for challenging problems of the society

**PEO 2:** Pursue higher studies and research thereby engages in lifelong learning by adapting to the current trends in the area of Computer Science and Engineering

**PEO 3:** Ability to Provide socially acceptable and economically feasible computer oriented solutions to real world problems with teamwork, while maintaining environmental balance, quality and cognizance of the underlying principles of ethics.

## 7.2 Appendix I: Program Outcomes

1. **PO1: Engineering Knowledge:** Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems.
2. **PO2: Problem Analysis:** Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)
3. **PO3: Design / Development of Solutions:** Design creative solutions for complex engineering problems and design / develop systems / components / processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)
4. **PO4: Conduct Investigations of Complex Problems:** Conduct investigations of complex engineering problems using research-based knowledge including design of

experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8)

5. **PO5: Engineering Tool Usage:** Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)

6. **PO6: The Engineer and The World:** Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7).

7. **PO7: Ethics:** Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)

8. **PO8: Individual and Collaborative Team work:** Function effectively as an individual, and as a member or leader in diverse / multi-disciplinary teams.

9. **PO9: Communication:** Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences

10. **PO10: Project Management and Finance:** Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.

11. **PO11: Life-Long Learning:** Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change..

### 7.3 Appendix II: Course Outcomes (COs)

1. **CO1**- Design a UML structural and behavioural model for solving real world problems (Analyze Level).
2. **CO2**- Develop programs to solve problems using object-oriented design techniques through Java (Apply Level)
3. **CO3**- Illustrate the creation and usages of packages and interfaces in Java. (Understand Level)
4. **CO4**-Demonstrate the exception handling mechanism to handle run time errors (Apply Level)
5. **CO5**-Make use of collection interfaces and classes to store and manipulate data effectively. (Apply Level)
6. **CO6**-Develop Java application to solve a real-world problem by using graphical user interfaces and Event handling techniques (Create Level).

### 7.4 Appendix III: Fulfillment of Programme Outcomes (POs)

#### 1. PO2 - Problem Analysis

- **Outcome:** The project demonstrates the ability to identify and analyze real-world issues related to disaster management, including inefficient communication, uncoordinated volunteer efforts, and lack of centralized reporting.
- **Justification:** Through an analysis of the pitfalls of current disaster relief systems and emergency apps, the team learned major coordination and information flow challenges. RELIFE specifically solves these issues through a single platform covering reporting, resource allocation, and community support during disasters.

#### PO3 - Design/Development of Solutions

- **Outcome:** The project demonstrates capability in designing and creating efficient digital solutions that address emergency response and community needs.
- **Justification:** The group came up with a user-friendly Flutter-based interface with organized modules for lost person reporting, donations, blood donation requests,

and shelter data. The modules make these features easily accessible and offer a sound solution for disaster response and relief management..

## 2. PO5 - Modern Tool Usage

- **Outcome:** Proved ability to work with contemporary development tools and frameworks.
- **Justification:** The project made use of Flutter for frontend development and incorporated

Google Maps API for location-based services. Utilization of up-to-date technologies emphasizes the efficiency of the team in utilizing contemporary tools towards real-time problem-solving in emergency situations.

## 3. PO9 – Individual and Teamwork

- **Outcome:** Smooth cooperation and equal contribution by all team members.
- **Justification:** The team cooperated in several roles, such as UI/UX design, module integration, and testing. They held frequent meetings and communication to ensure smooth flow of work and timely delivery of project milestones.

## 4. PO10 - Communication

- Outcome: Exemplary coordination and communication skills exhibited throughout the project.
- Justification: Clear communication was upheld by the team for planning, problem solving, and documentation to ensure smooth progress and good presentation of

## 5. project results.PO10 – Life-long Learning

- Outcome: Shown flexibility and dedication to constant learning of new technologies and techniques.
- Justification: The team tested new technology such as Flutter and Google Maps API, learning how to incorporate and optimize them for actual application in disaster management.

This is an expression of dedication to ongoing skill development and technological adaptability.

