

HUNGRY BITES

A PROJECT REPORT

For

Project (KCA451)

Session (2024-25)

Submitted by

Pratham Dhingra

(2300290140126)

Prashant Kumar

(2300290140122)

Sameeksha Varshney

(2300290140155)

Nikhil Nigam

(2300290140108)

**Submitted in partial fulfilment of the
Requirements for the Degree of**

MASTER OF COMPUTER APPLICATION

**Under the Supervision of
Mr. Shashank Bhardwaj
(Associate Professor)**



Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS

KIET Group of Institutions, Ghaziabad

Uttar Pradesh-201206

(February-2025)

CERTIFICATE

Certified that **Pratham Dhingra (2300290140126)**, **Nikhil Nigam (2300290140108)**, **Sameeksha (2300290140155)**, and **Prashant Kumar (2300290140122)** have carried out the project work titled **“Hungry Bites” (Project - KCA451)** for Master of Computer Application from **Dr. A.P.J. Abdul Kalam Technical University (AKTU)** (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the students themselves and the contents of the project report do not form the basis for the award of any other degree to the candidates or to anybody else from this or any other University/Institution.

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Mr. Shashank Bhardwaj

Associate Professor

Department of Computer Applications

KIET Group of Institutions, Ghaziabad

Dr. Akash Rajak

Dean

Department of Computer Applications

KIET Group of Institutions, Ghaziabad

ABSTRACT

Food insecurity and food waste are pressing global challenges, with millions of people going hungry while vast amounts of surplus food are discarded daily. Addressing this dual problem requires innovative solutions to efficiently redistribute surplus food to those in need. Hungry Bites, a MERN stack-based food donation platform, bridges this gap by connecting donors, recipients, and administrators through a user-friendly interface. The platform streamlines food donations with features like user registration, food donation management, recipient request systems, and an admin dashboard for oversight. Donors can list surplus food, recipients such as NGOs can claim or request items, and administrators can monitor and manage the entire process, ensuring transparency and efficiency. By fostering community-driven food sharing, Hungry Bites minimizes food waste, reduces hunger, and promotes sustainability, aligning with global goals like Zero Hunger (SDG 2). Scalable and accessible, Hungry Bites is a powerful tool for creating a compassionate, sustainable ecosystem that ensures no food goes to waste and no one goes hungry.

ACKNOWLEDGEMENT

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **Mr. Shashank Bhardwaj** for his/ her guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to Dr. Akash Rajak, Professor and Dean, Department of Computer Applications, for his insightful comments and administrative help on various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Pratham Dhingra (2300290140126)

Prashant Kumar (2300290140122)

Sameeksha Varshney (2300290140155)

Nikhil Nigam (2300290140108)

TABLE OF CONTENTS

Section	Page No.
Certificate	ii
Abstract	iii
Acknowledgement	iv
Table of Contents	v
List of Abbreviations	vi
List of Tables	vii
List of Figures	viii
Chapter 1: Introduction	1–3
1.1 Problem Summary	1
1.2 Shodh Yatra	1
1.3 Aim and Objective of the Project	2
1.4 Problem Specification	2
1.5 Work Plan	3
Chapter 2: Requirement Analysis	4–9
2.1 Functional Requirements	4–5
2.1.1 For Donors	4
2.1.2 For Volunteers	5
2.1.3 For NGO Management	5
2.1.4 For Admins	5

2.2 Non-Functional Requirements	6
2.3 System Requirements	7
2.3.1 Hardware Requirements	7
2.3.2 Software Requirements	7
2.4 Feasibility Study	8
2.4.1 Technical Feasibility	8
2.4.2 Operational Feasibility	8
2.4.3 Economic Feasibility	8
2.4.4 System Limitations	9
Chapter 3: System Design	10–31
3.1 System Architecture	10–12
3.1.1 Presentation Layer (UI)	10
3.1.1.1 Key Features for Food Donation Website	10
3.1.1.2 Technologies Used	11
3.1.2 Business Access Layer (BAL)	11
3.1.2.1 Key Responsibilities	11
3.1.2.2 Technologies Used	12
3.1.3 Data Access Layer (DAL)	12
3.1.3.1 Key Responsibilities	12
3.1.3.2 Technologies Used	12
3.2 System Diagrams	13–22
3.2.1 Class Diagram	13

3.2.2 Use Case Diagram	14
3.2.3 Activity Diagrams	15–19
3.2.3.1 Activity Diagram for Registration	15
3.2.3.2 Activity Diagram for Login	16
3.2.3.3 Activity Diagram for NGO Management	17
3.2.3.4 Activity Diagram for Donor	18
3.2.3.5 Activity Diagram for Volunteer	19
3.2.4 Sequence Diagrams	20–22
3.2.4.1 Sequence Diagram for NGO Management	20
3.2.4.2 Sequence Diagram for Donor	21
3.2.4.3 Sequence Diagram for Volunteer	22
3.3 Database Design	23–26
3.3.1 E-R Diagram	23
3.3.2 Data Dictionary	24–26
3.4 Business Model Canvas	27–31
3.4.1 Customer Segments	27
3.4.2 Value Proposition	27
3.4.3 Channels	28
3.4.4 Customer Relationships	28
3.4.5 Revenue Streams	28
3.4.6 Key Resources	29
3.4.7 Key Activities	29

3.4.8 Key Partners	29
3.4.9 Cost Structure	30
Chapter 4: System Description	32–33
4.1 Software Description	32
4.1.1 MongoDB	32
4.1.2 Express.js	32
4.1.3 React.js	32
4.1.4 Node.js	32
4.2 Key Benefits of MERN Stack	33
Chapter 5: Implementation	34–36
5.1 Coding Standards	34
5.2 Total Modules	34–36
5.2.1 User Module	34
5.2.2 Admin Module	35
5.2.3 NGO Management Module	35
5.2.4 Volunteer Module	35
Chapter 6: Conclusion	35–36
Chapter 7: Future Scope	36–37
Bibliography	38–39

List of Figure

Figure No.	Title	Page No.
3.1	Three-Tier Architecture	10
3.2	Class Diagram	13
3.3	Use Case Diagram	14
3.4	Activity Diagram for Registration	15
3.5	Activity Diagram for Login	16
3.6	Activity Diagram for NGO Mgmt	17
3.7	Activity Diagram for Donor	18
3.8	Activity Diagram for Volunteer	19
3.9	Sequence Diagram for NGO Mgmt	20
3.10	Sequence Diagram for Donor	21
3.11	Sequence Diagram for Volunteer	22
3.12	E-R Diagram	23
3.13	BMC Canvas	31

3.1	Three-Tier Architecture	10
3.2	Class Diagram	13
3.3	Use Case Diagram	14
3.4	Activity Diagram for Registration	15
3.5	Activity Diagram for Login	16
3.6	Activity Diagram for NGO Mgmt	17
3.7	Activity Diagram for Donor	18
3.8	Activity Diagram for Volunteer	19
3.9	Sequence Diagram for NGO Mgmt	20
3.10	Sequence Diagram for Donor	21

CHAPTER 01

INTRODUCTION

1.1 PROBLEM SUMMARY

In the current scenario, many NGOs face challenges in effectively managing their food donation efforts. Key issues include communication gaps among members, volunteers, and coordinators, difficulty in tracking donation pickups, and challenges in locating volunteers and donation sites.

One of our team members, who is a volunteer for an NGO, particularly the Robin Hood Army, highlighted these issues. The NGO's main objective is to feed underprivileged individuals by collecting surplus food from restaurants, individuals, and other sources. However, the volunteer has faced difficulties such as miscommunication, missed food pickups, and challenges in coordinating with other team members or finding specific locations for food collection and distribution.

To address these challenges, we have developed a Web Application designed specifically for NGOs engaged in food donation. This application will serve as a platform where NGOs, donors, and volunteers can seamlessly coordinate efforts to collect and distribute food, ensuring that it reaches those in need efficiently and effectively.

1.2 SHODH YATRA

Food wastage is a significant issue in densely populated countries like India. This wastage not only exacerbates hunger but also contributes to environmental and economic problems. Instead of letting surplus food go to waste, it can be redirected to those in need, such as individuals in orphanages, old age homes, and other underserved communities.

Our proposed solution, a food donation web application, aims to bridge the gap between donors, volunteers, and NGOs. By leveraging technology, we intend to create an organized and streamlined approach to food donation.

The app, titled "Spread A Smile", was conceptualized and designed after identifying the challenges faced by our team member, a volunteer with the Robin Hood Army. Issues such as

poor communication, missed food pickups, and lack of proper coordination served as the motivation for this project. The application aims to connect all NGOs in a particular city, facilitating the donation and distribution of surplus food while fostering better communication and collaboration among stakeholders.

1.3 AIM AND OBJECTIVE OF THE PROJECT

The aim of our web application is to establish an efficient platform that connects NGOs, donors, and volunteers to facilitate seamless food donation and distribution. The app is designed to address existing challenges in food donation and ensure that surplus food reaches the intended recipients without unnecessary delays or wastage.

Objectives of the Application:

- **Improve Awareness:** Create awareness about food donation and connect donors with NGOs.
- **Enable Seamless Communication:** Provide an effective communication channel between donors, NGOs, and volunteers.
- **Streamline Operations:** Ensure timely pickups and delivery of donated food to those in need.
- **Digitalize the Process:** Leverage technology to make the entire process of food donation faster, transparent, and more efficient.
- **Encourage Volunteer Participation:** Simplify coordination among volunteers and ensure they can easily locate donation sites and recipients.

1.4 PROBLEM SPECIFICATION

Effective communication and coordination are crucial for successful food donation. Our application addresses the following problems:

- **Lack of Communication:** Miscommunication among volunteers, donors, and NGO coordinators often leads to inefficiencies.
- **Missed Pickups:** Food pickups are sometimes missed due to poor planning and coordination.
- **Lack of Awareness:** Many potential donors are unaware of the donation process or available NGOs.

- Difficulty in Locating Sites: Volunteers face challenges in identifying donation locations or coordinating with other team members.

Our web application is designed to solve these issues by providing an intuitive and user-friendly platform for all stakeholders.

1.5 WORK PLAN

We have successfully completed the requirement gathering and analysis phase for our web application. The project team has engaged in detailed discussions, including identifying key challenges faced by NGOs and volunteers in food donation. Preparatory documents have also been finalized.

Additionally, we conducted studies of relevant problem statements and developed essential design and development tools such as:

- AEIOU Canvas
- Empathy Canvas
- Ideation Canvas
- Product Development Canvas

The design phase for the web application is already underway. The application will include features such as real-time communication, location tracking, and task management for volunteers. With a clear roadmap in place, our next steps involve implementing the frontend and backend features using the MERN stack to deliver an efficient and reliable solution.

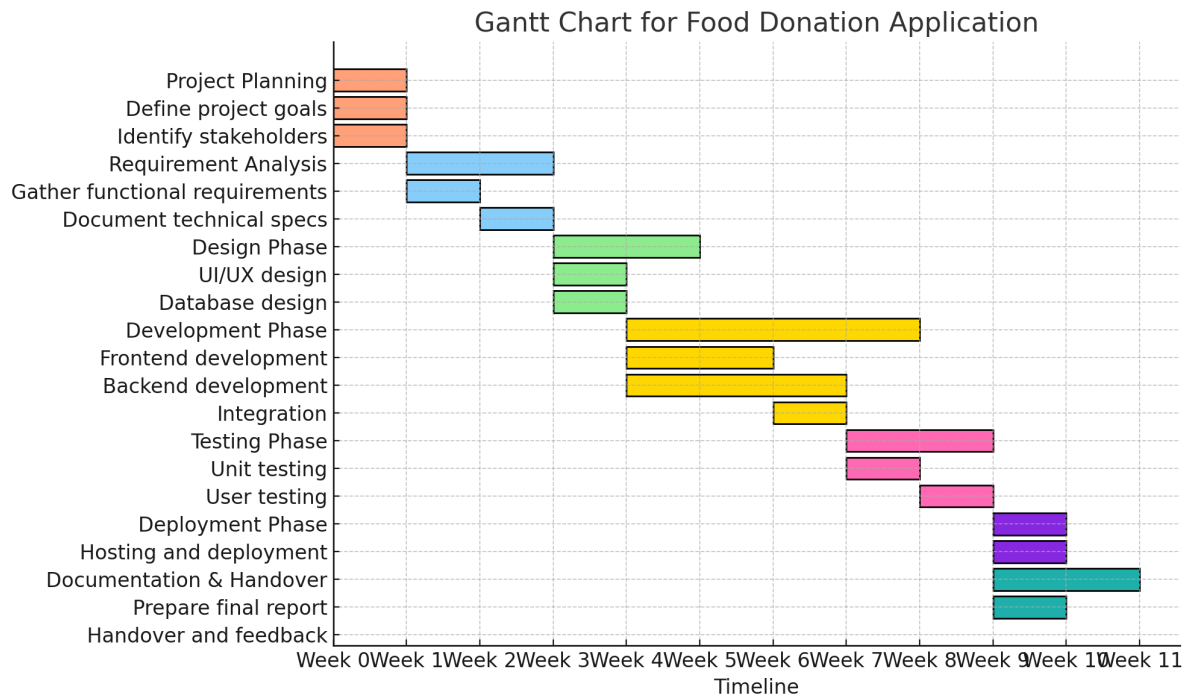


Fig. 1.1 Gantt Chart/Timeline chart

CHAPTER 02

REQUIREMENT ANALYSIS

2.1 FUNCTIONAL REQUIREMENTS

Functional requirements define the core capabilities and behaviors that the system must exhibit to meet user needs. These requirements are grouped based on the roles within the system: donors, volunteers, NGO management, and administrators.

2.1.1 For Donors

- **Donate Items:**
 - Donors can initiate a donation by selecting the type of item (e.g., food), its form (e.g., packaged or fresh), and scheduling a pickup by a volunteer.
 - Option to upload photos or provide details of the item for better clarity.
- **Track Donation Status:**
 - Real-time tracking of the volunteer assigned to their request, including location updates and estimated time of arrival.
- **Search and Select NGOs:**
 - Donors can search for NGOs based on criteria like location, focus area (e.g., feeding the poor, supporting orphanages), and past activities.
- **Donation History:**
 - View a record of past donations, including the recipient NGO and status.

2.1.2 For Volunteers

- **Manage Requests:**
 - Volunteers can accept or decline requests assigned by NGO management.
 - Receive notifications for new tasks, including pickup details and donor location.
- **Navigation and Task Execution:**
 - Access GPS-based directions to reach donors and distribution points.
 - Update the status of tasks, such as "Item Picked," "En Route," or "Delivered."
- **Performance Dashboard:**

- Track their performance, including completed pickups, delivery success rate, and feedback from donors or NGOs.

2.1.3 For NGO Management

- Request Management:
 - Accept, reject, or reassign donation requests from donors based on availability and logistics.
- Volunteer Coordination:
 - Assign volunteers to specific donation requests and monitor their activities in real-time.
- Performance Monitoring:
 - Maintain records of volunteers' activities, efficiency, and reliability.
- Feedback System:
 - Collect feedback from donors and recipients to improve operations.

2.1.4 For Admins

- Database Management:
 - Oversee and maintain donor, volunteer, and NGO information in the database.
- System Monitoring:
 - Manage app settings, ensure smooth operation, and address any system-related issues.
- Analytics and Reporting:
 - Generate statistical reports on donations, NGO activities, and volunteer performance.

2.2 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements define the quality attributes and constraints of the system, ensuring it meets user expectations and performs efficiently.

- Portability:

- As a web-based application, it shall be accessible across multiple platforms, including desktop and mobile browsers, ensuring broad user access.
- Usability:
 - Intuitive user interface with easy navigation and user-friendly features for donors, volunteers, and NGOs.
 - Clearly labeled menus and consistent design to enhance user experience.
- Privacy and Security:
 - Secure login and authentication mechanisms for all users.
 - Data protection measures to prevent unauthorized access to donor and NGO information.
 - Compliance with data privacy regulations.
- Performance:
 - The system should respond to user actions within a few seconds to ensure smooth operation.
 - Efficient handling of simultaneous requests, ensuring no delays even during peak usage.
- Scalability:
 - Designed to accommodate a growing number of users, NGOs, and data as the application expands.
 - Supports dynamic scaling to handle increased traffic and requests.
- Reliability:
 - Ensures consistent performance without crashes or downtime.
 - Robust error-handling mechanisms to prevent system failures.

2.3 SYSTEM REQUIREMENTS

System requirements are the technical and operational specifications the application must meet to fulfill its intended functionality.

2.3.1 Hardware Requirements

- Processor: 2.6 GHz or higher.
- Memory: Minimum 2GB RAM.
- Storage: At least 500MB of free disk space.
- Internet: Stable connection for real-time updates.

2.3.2 Software Requirements

- Operating System: Windows, macOS, Linux, or any modern browser for web access.
- Frameworks and Technologies:
 - Frontend: React.js for building a dynamic user interface.
 - Backend: Node.js and Express.js for server-side operations.
 - Database: MongoDB for efficient data storage and retrieval.
 - APIs: Integration of Google Maps API for location tracking and navigation.
- Tools and Libraries:
 - npm or yarn for managing dependencies.
 - Deployment tools such as Docker or Kubernetes for scaling the app.

2.4 FEASIBILITY STUDY

2.4.1 Technical Feasibility

- The system utilizes the MERN stack, a proven technology for scalable and efficient web applications.
- GPS-based APIs and tracking mechanisms will enable real-time location services.
- The use of MongoDB allows flexible data handling and ensures robust storage solutions.
- Web hosting services (e.g., AWS, Google Cloud) will provide a scalable and secure environment.

2.4.2 Operational Feasibility

- The system is designed with a simple GUI to ensure ease of use, even for users with minimal technical skills.
- Volunteers and NGO staff are familiar with basic navigation and task execution on web platforms.
- Automated notifications and real-time updates simplify operational workflows.

2.4.3 Economic Feasibility

- The application leverages open-source technologies, reducing development costs.
- Cloud-based hosting ensures cost-effective scalability.

- Long-term benefits of efficient food donation outweigh initial development and deployment costs.

2.4.4 System Limitations

- **Platform Dependency:**
 - The app may not have full functionality on older devices or browsers with limited support for modern web technologies.
 - Currently optimized for desktop and mobile browsers, with plans to expand to dedicated mobile apps in the future.
- **Internet Dependency:**
 - Requires an active internet connection for real-time features, which may limit usability in areas with poor connectivity.

Despite these limitations, the system is designed to provide reliable and efficient service to facilitate food donation and distribution.

CHAPTER 03

SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

The system will be developed using 3-tier architecture:

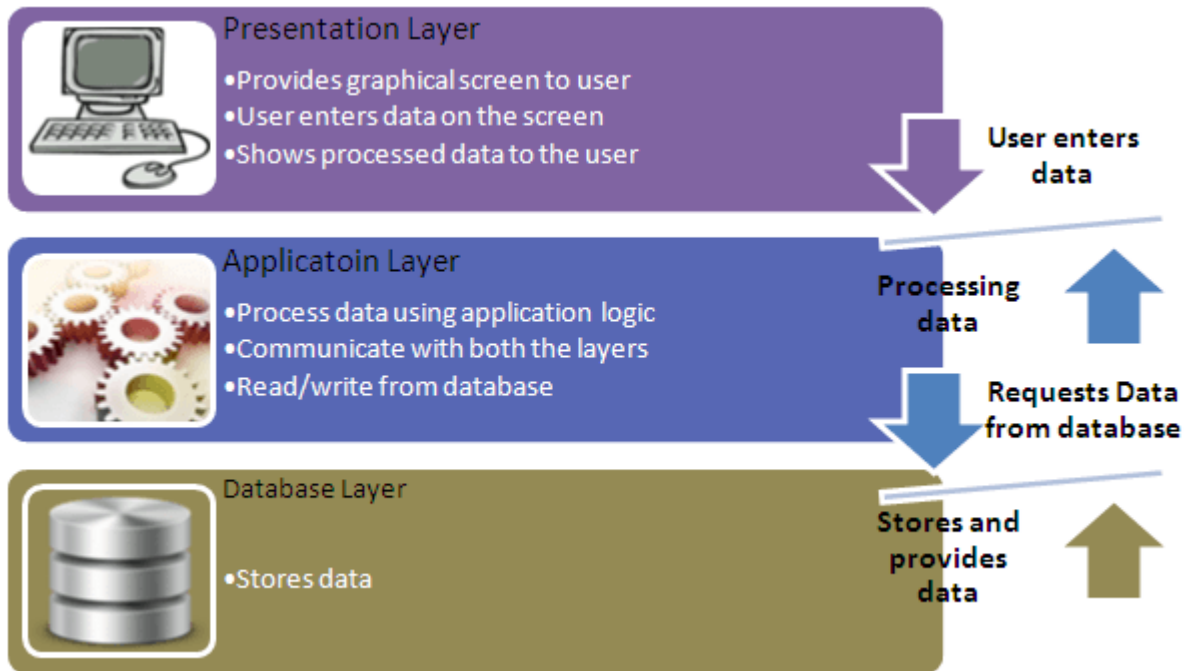


Fig. 3.1 Three Tier Architecture

3.1.1 Presentation Layer (UI):

The **Presentation Layer** is the user-facing part of the food donation website. It is responsible for displaying data to the user and capturing their input. This layer typically includes web pages (.aspx for ASP.NET or HTML/CSS/JS for modern web development) or forms for user interaction.

3.1.1.1 Key Features for Food Donation Website:

- **Home Page:** Displays a welcoming message, introduces the mission, and highlights statistics or success stories about food donations.
- **Registration and Login Forms:** Allow donors, recipients, and volunteers to register and log in to their accounts.

- **Donation Form:** A form where donors can specify the type of food, quantity, and pick-up or drop-off details.
- **Request Form:** Enables organizations or individuals in need to request food donations.
- **Dashboard:** Personalized dashboard for users showing their donation or request history and status.
- **Contact Us and About Us Pages:** Provide information about the organization and ways to get in touch.

3.1.1.2 Technologies Used:

- Frontend Frameworks: HTML, CSS, JavaScript, and libraries like React, Angular, or Vue.js.
- Design Tools: Bootstrap or TailwindCSS for responsive UI design.
- Interaction: AJAX or Fetch API for seamless interaction between UI and backend services.

3.1.2 Business Access Layer (BAL) or Business Logic Layer:

The **BAL** contains the core business logic of the application. It acts as a mediator between the UI and the data access layer to process user requests, validate inputs, and enforce rules.

3.1.2.1 Key Responsibilities for Food Donation Website:

- **Validation Logic:** Ensures all inputs (e.g., food details, recipient requests) are valid and conform to business rules.
- **Donation Matching:** Matches food donations with nearby requests based on location, type of food, and urgency.
- **User Role Management:** Defines and enforces roles such as donor, recipient, and administrator.
- **Notification Services:** Sends updates via email or SMS about the status of donations or requests.
- **Reports Generation:** Creates reports for admins, such as total donations, food types, and impact statistics.

3.1.2.2 Technologies Used:

- Programming Languages: C#, Java, Python, or Node.js.
- Frameworks: ASP.NET Core, Django, or Express.js.
- Libraries: Validation libraries like FluentValidation (C#) or Joi (Node.js).

3.1.3 Data Access Layer (DAL):

The **DAL** directly interacts with the database to perform CRUD (Create, Read, Update, Delete) operations. It serves as the intermediary between the business logic layer and the data storage system.

3.1.3.1 Key Responsibilities for Food Donation Website:

- **Data Retrieval:** Fetches data such as available donations, pending requests, and user details from the database.
- **Data Manipulation:** Handles adding new donations, updating the status of requests, and deleting expired records.
- **Database Connectivity:** Establishes and manages connections to the database securely.
- **Error Handling:** Ensures database errors are logged and handled gracefully without crashing the application.

3.1.3.2 Technologies Used:

- Database Management Systems: SQL Server, MySQL, PostgreSQL, or NoSQL solutions like MongoDB.
- ORMs (Object-Relational Mappers): Entity Framework (C#), Hibernate (Java), or Sequelize (Node.js).
- Query Execution: Executes raw SQL queries or uses stored procedures to interact with the database.

We selected 3-tier architecture because the three important modules like the UI, logic and database are independent of each other and are clearly defined. Also modifying any one tier will not affect the other. Along with it we get the following benefits for using 3-tier architecture:

- **Scalability:** Each tier can scale horizontally. For example, you can load-balance the Presentation tier among three servers to satisfy more Web requests without adding servers to the Application and Data tiers.
- **Performance:** Because the Presentation tier can cache requests, network utilization is minimized, and the load is reduced on the Application and Data tiers. If needed, you can load-balance any tier.
- **Availability:** If the Application tier server is down and caching is sufficient, the Presentation tier can process Web requests using the cache.

3.2 SYSTEM DIAGRAMS

A System Diagram is a visual representation of the components, processes, and interactions within a system. It helps in understanding the system's structure, behavior, and flow of information or resources between different parts. System diagrams are widely used in various fields such as software development, business processes, engineering, and project management.

3.2.1 Class Diagram

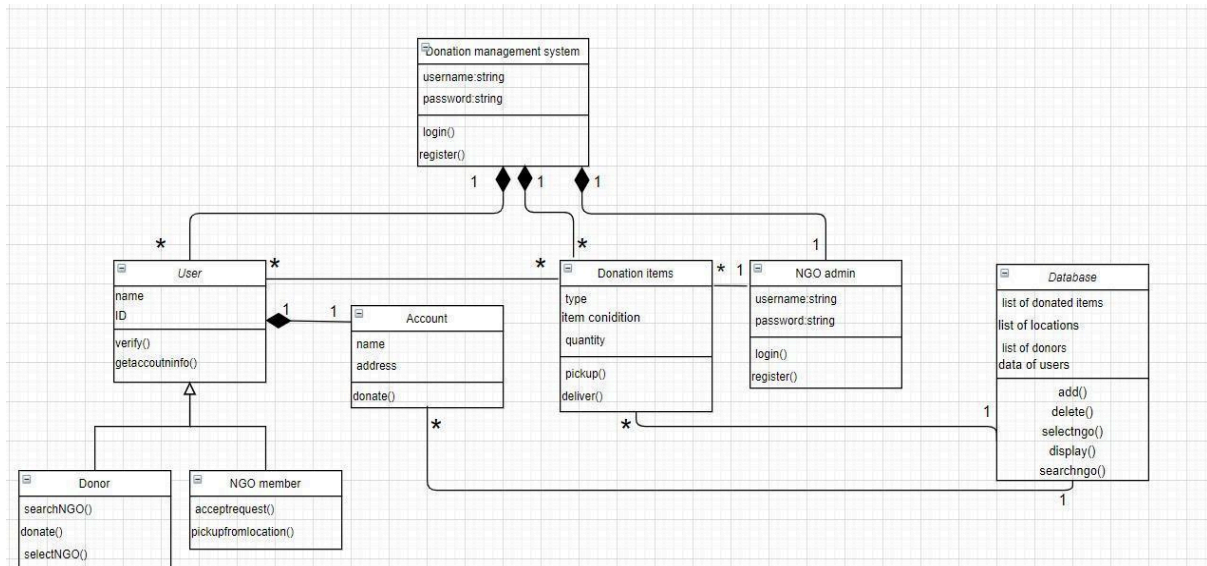


Fig 3.2 Class Diagram

3.2.2 Use Case

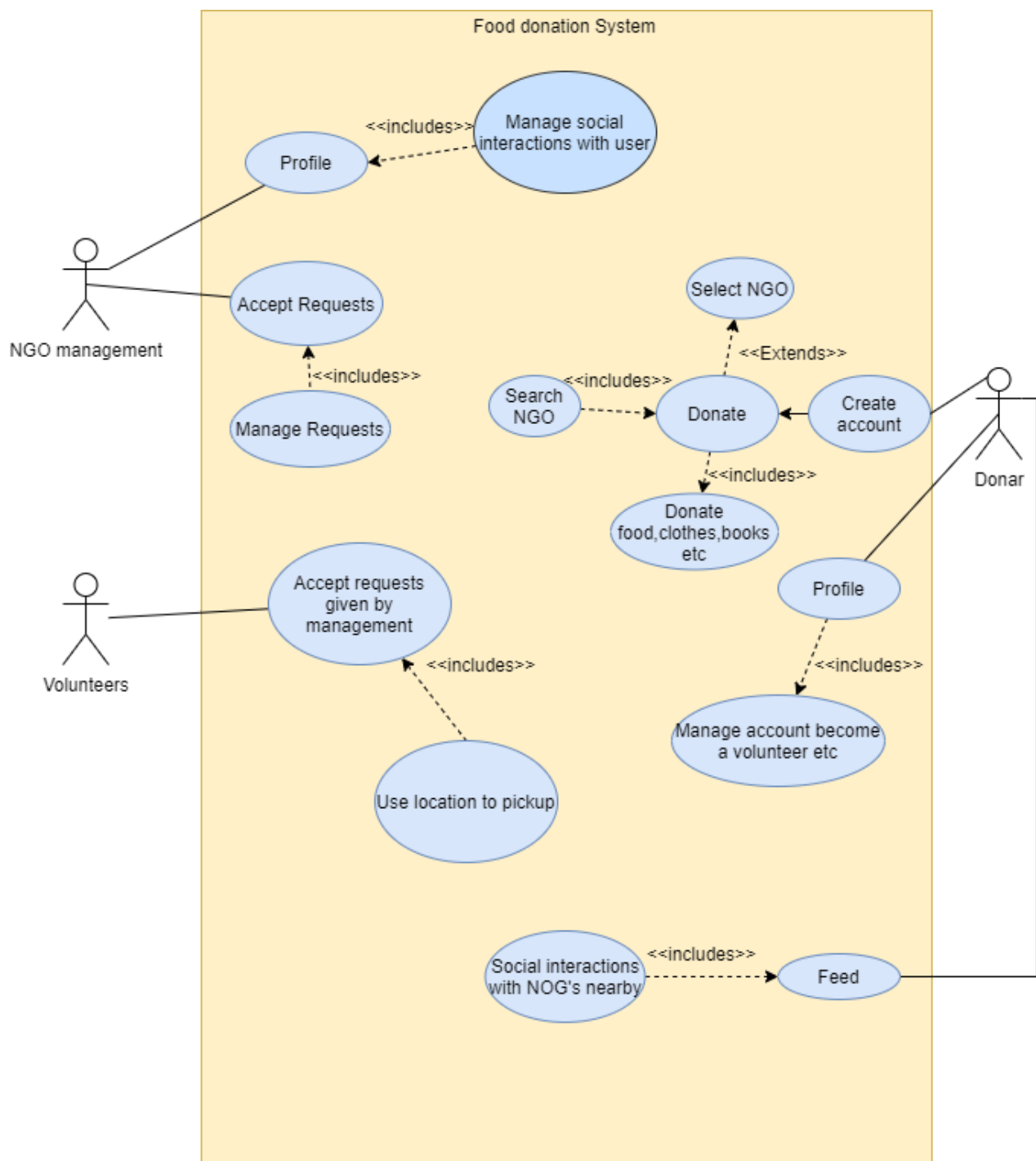


Fig 3.3 Use Case Diagram

3.2.3 Activity Diagram

3.2.3.1 Activity Diagram for Registration

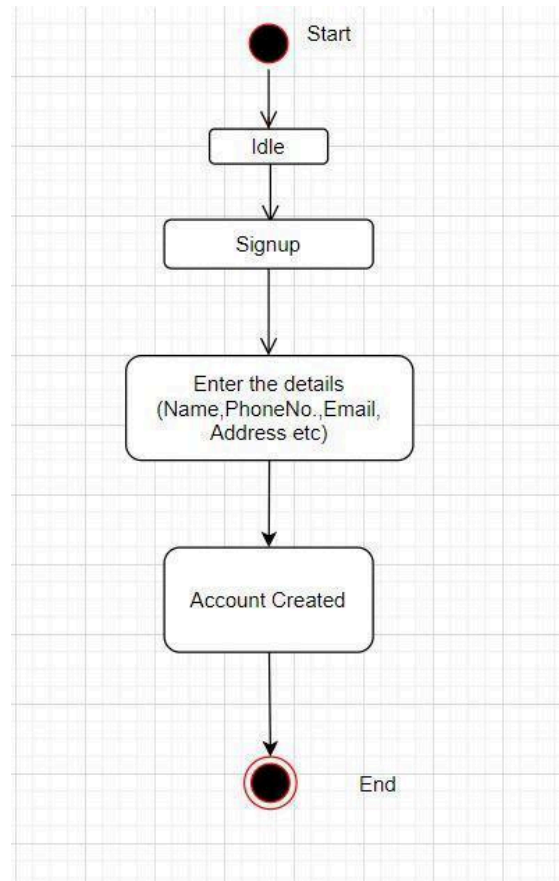


Fig 3.4 Activity Diagram for Registration

3.2.3.2 Activity Diagram for Login

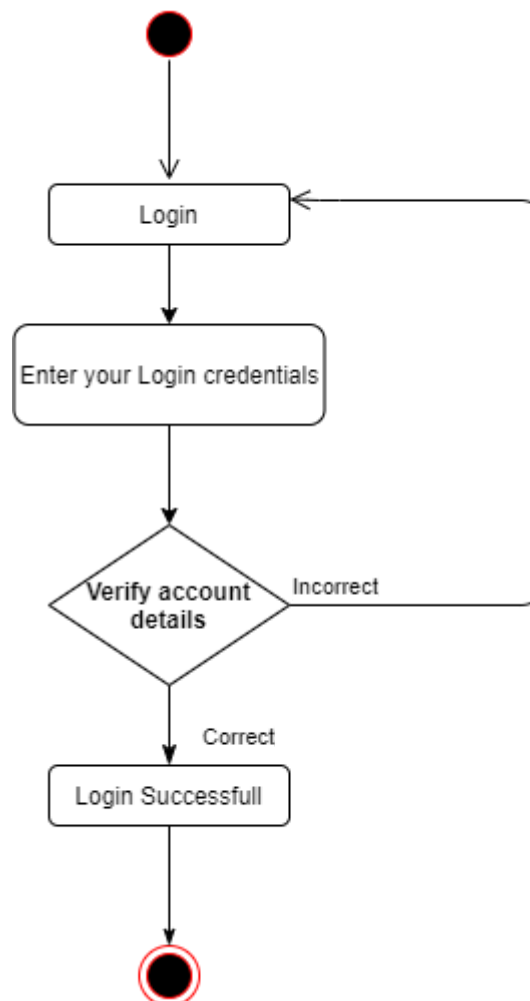


Fig 3.5 Activity Diagram for Login

3.2.3.3 Activity Diagram for NGO Management

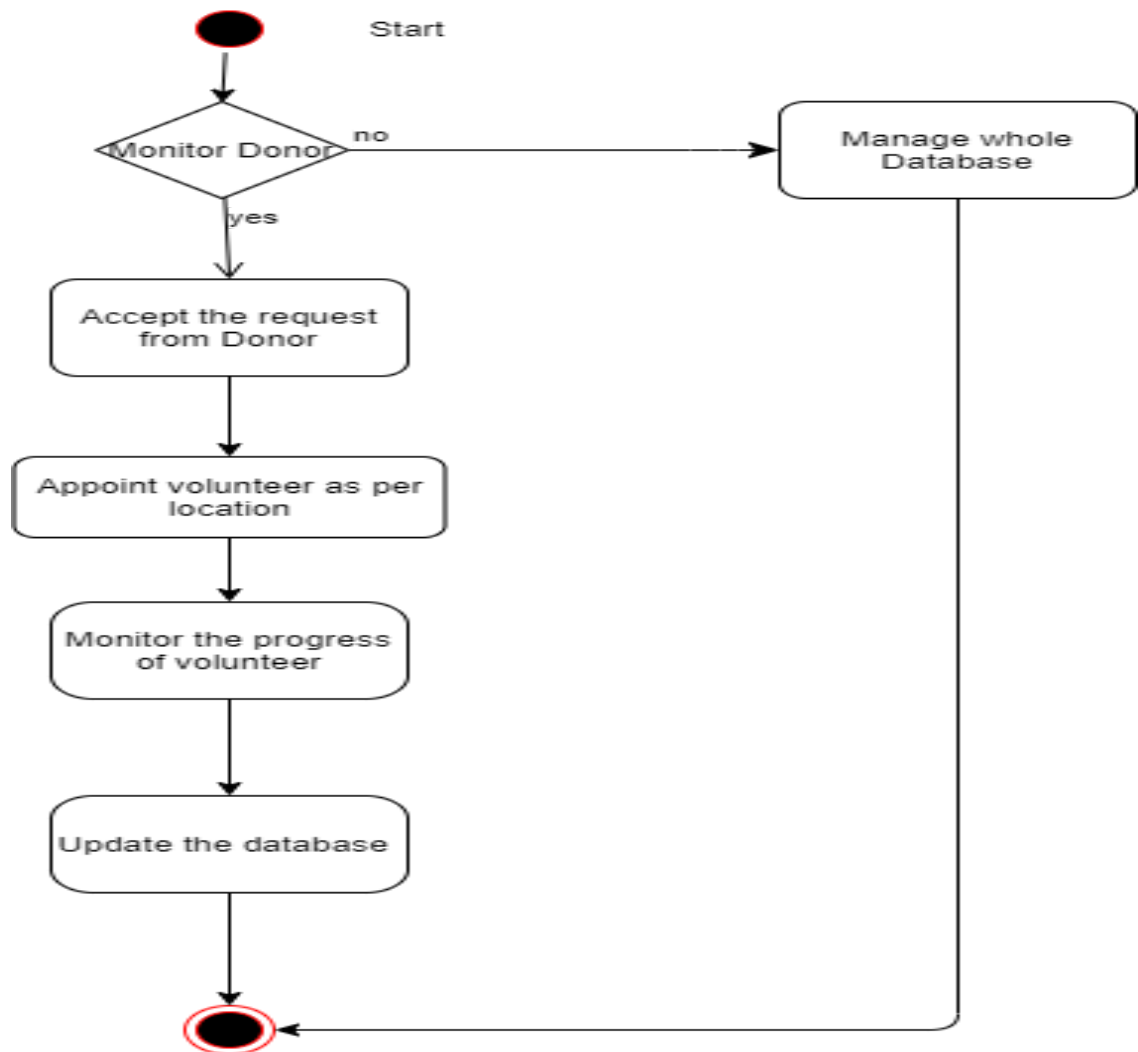


Fig 3.6 Activity Diagram for NGO Management

3.2.3.4 Activity Diagram for Donor

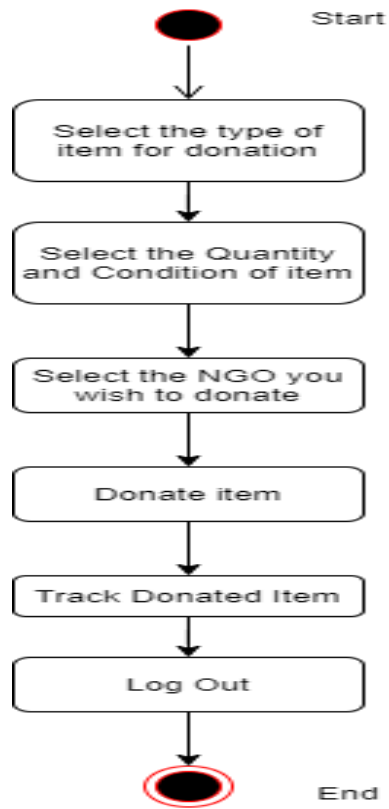


Fig .7 Activity Diagram for Donor

3.2.3.5 Activity Diagram for Volunteer

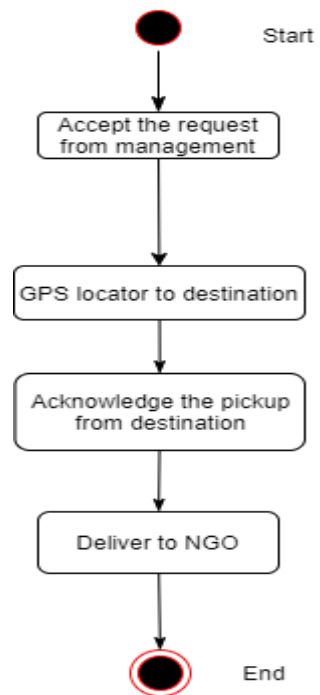


Fig 3.8 Activity Diagram for Volunteer

3.2.4 Sequence Diagram

3.2.4.1 Sequence Diagram for NGO Management

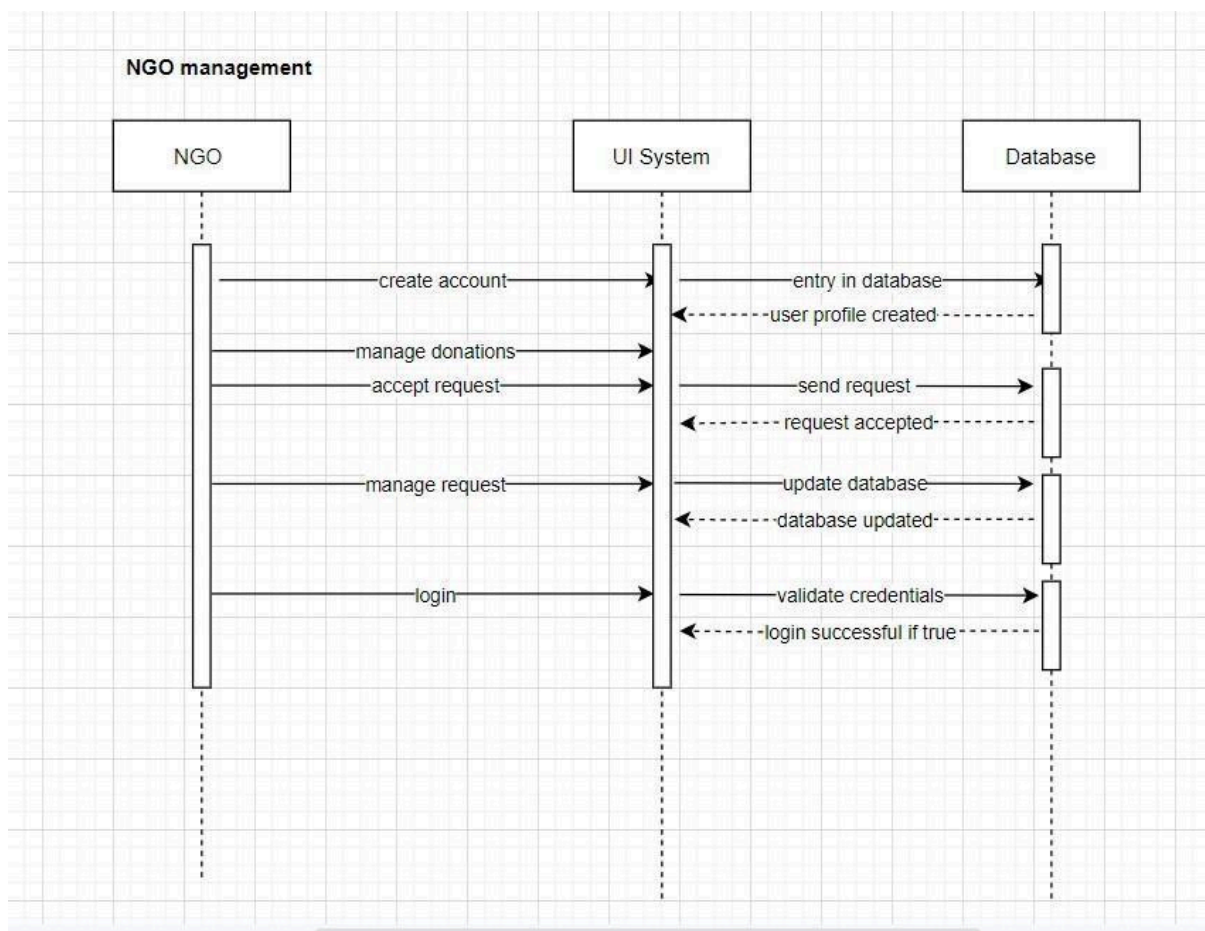


Fig 3.9 Sequence Diagram for NGO Management

3.2.4.2 Sequence Diagram for Donor

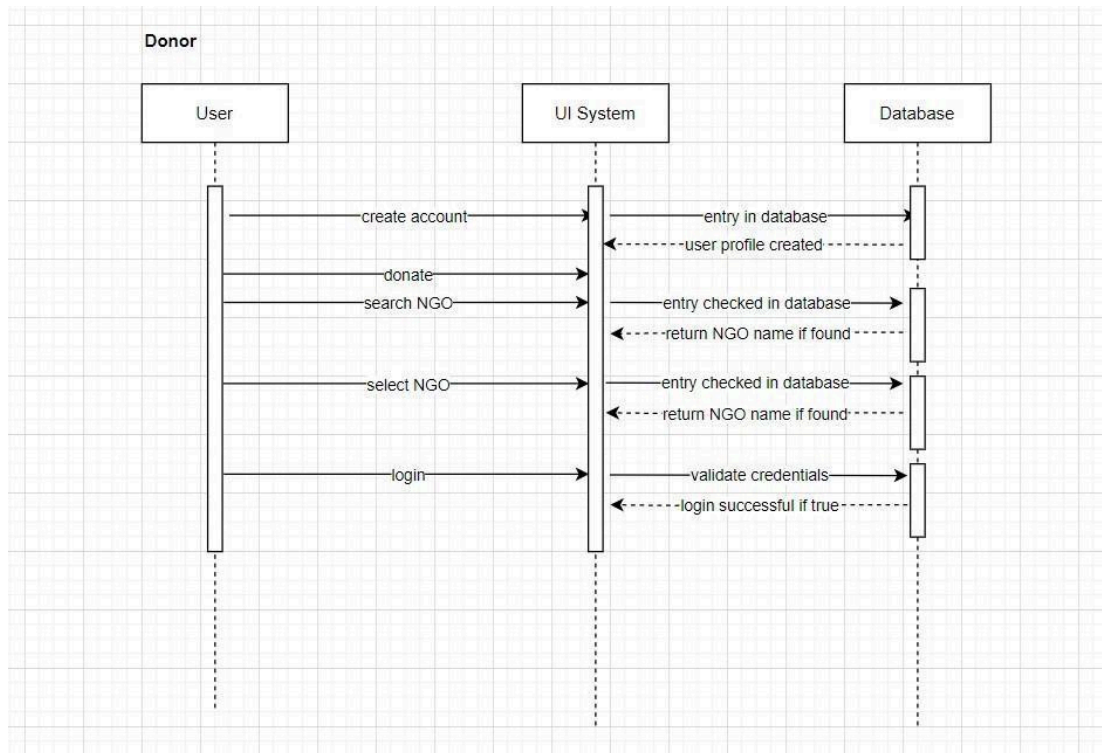


Fig 3.10 Sequence Diagram for Donor

3.2.4.3 Sequence Diagram for Volunteer

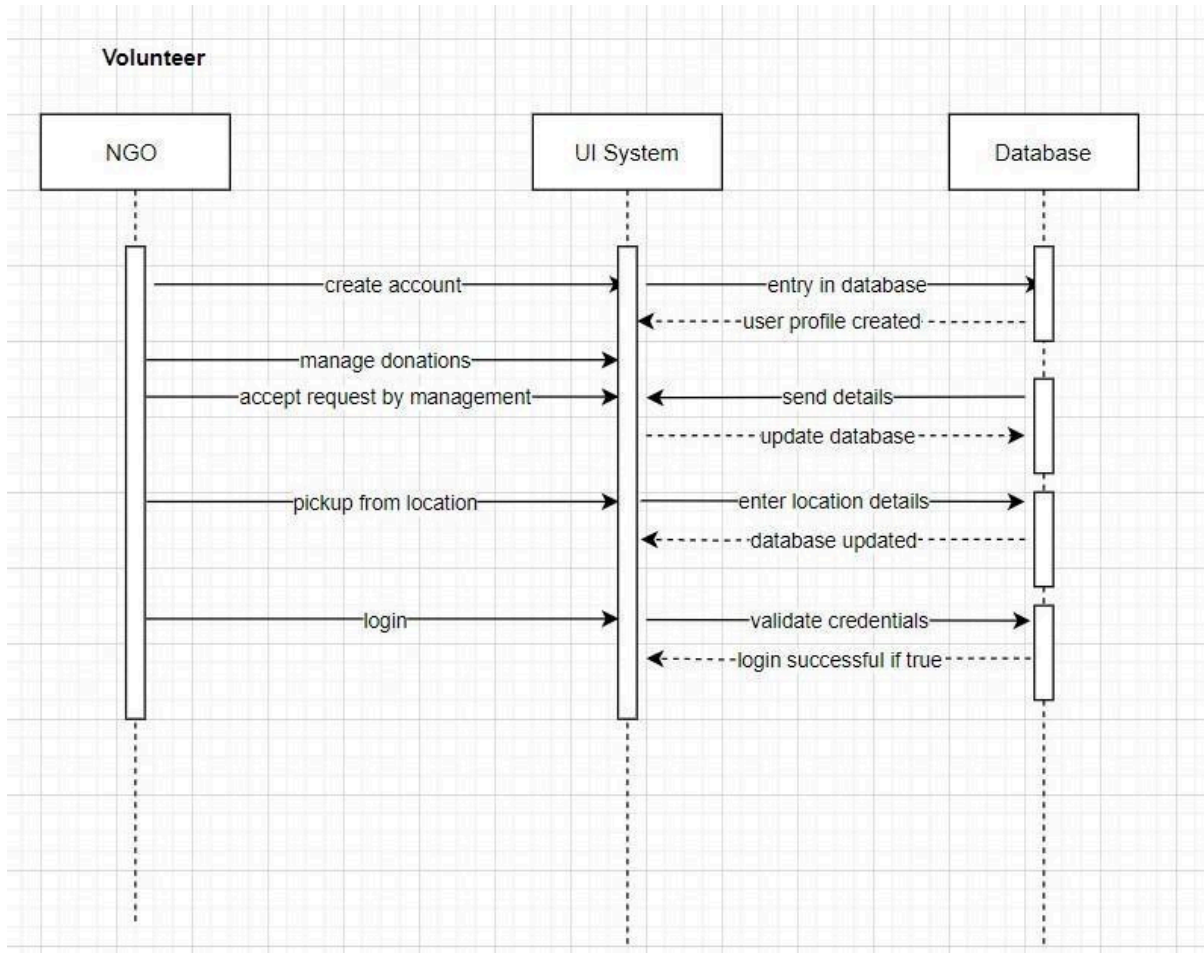


Fig 3.11 Sequence Diagram for Volunteer

3.3 DATABASE DESIGN

3.3.1 E-R Diagram

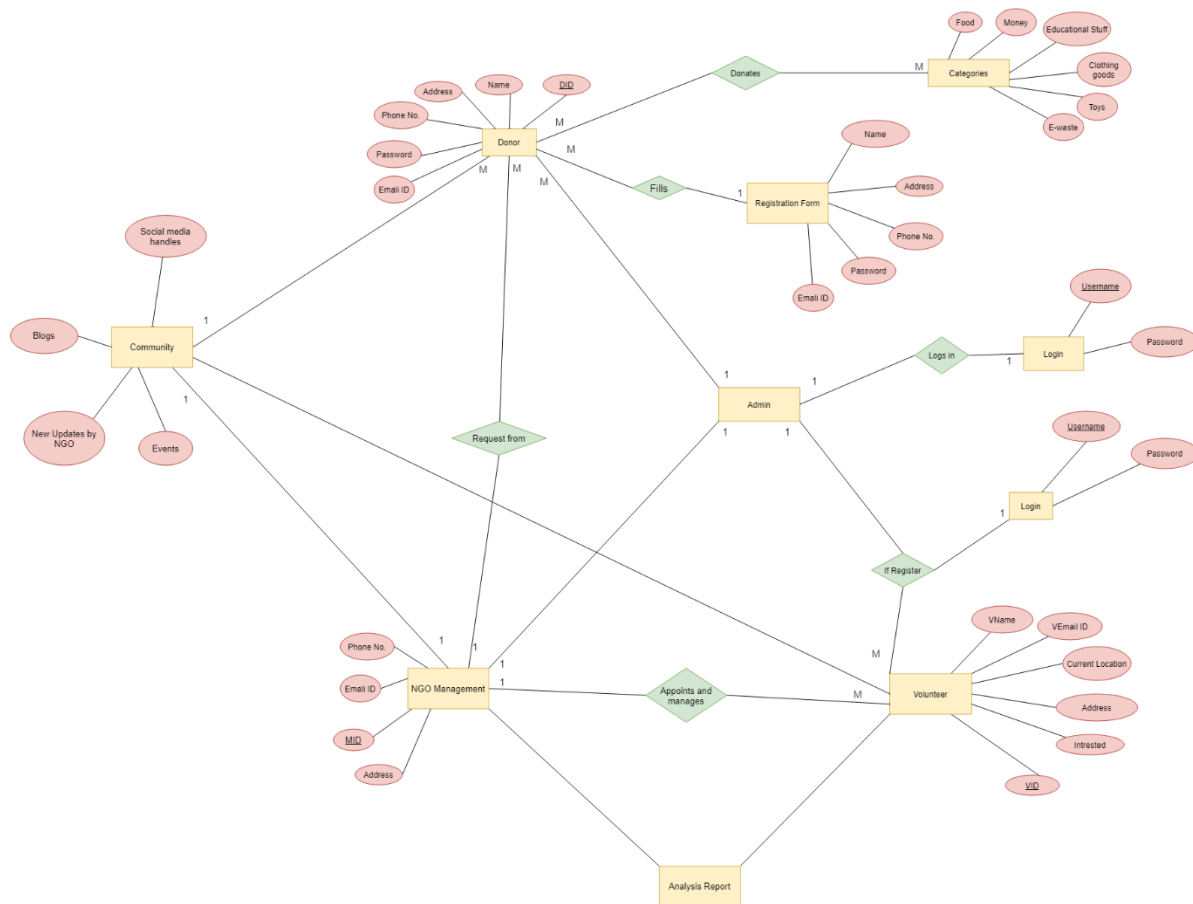


Fig. 3.12 E-R Diagram

3.3.2 Data Dictionary

3.3.2.1 Donor table

Field Names	Data Type	Constraint
DID	Varchar2(50)	Primary Key
Name	Varchar2(25)	Not Null
Address	Varchar2(60)	Not Null
Phone No.	Numeric (20,0)	Not Null
Password	Varchar (50)	Not Null
Email ID	Varchar (50)	Not Null

Table 3.1 Donor Table

3.3.2.2 Volunteer Table

Field Names	Data Type	Constraint
VID	Varchar2(50)	Primary Key
VName	Varchar2(50)	Not Null
Address	Varchar2(60)	Not Null
Phone No.	Numeric (20,0)	Not Null
Password	Varchar (50)	Not Null
Email ID	Varchar (50)	Not Null
Current Location	Varchar (25)	Not Null
Drives No.	Varchar (30)	Not Null

Table 3.2 Volunteer table

3.3.2.3 NGO management table

Field Names	Data Type	Constraint
MID	Varchar2(50)	Primary Key
Name	Varchar2(25)	Not Null
Address	Varchar2(60)	Not Null
Phone No.	Numeric (20,0)	Not Null
Password	Varchar (50)	Not Null
Email ID	Varchar (50)	Not Null

Number of requests	Numeric (30)	Not Null
--------------------	--------------	----------

Table 3.3 NGO management table

3.3.2.4 Registration details table

Field Names	Data Type	Constraint
Name	Varchar2(25)	Primary Key
Address	Varchar2(60)	Not Null
Phone No.	Numeric (20,0)	Not Null
Password	Varchar (50)	Not Null
Email ID	Varchar (50)	Not Null

Table 3.4 Registration Details Table

3.3.2.5 Registered user table

Field Names	Data Type	Constraint
User ID	Varchar2(25)	Primary key
Username	Varchar2(25)	Not Null
Password	Varchar2(60)	Not Null

Table 3.5 Registered user table

3.3.2.6 Donated items table

Field Names	Data Type	Constraint
Item number	Numeric	Primary key
Type	Varchar2(25)	Not Null
Quantity	Numeric	Not Null
Delivery status	Boolean	Not null
Pickup location	Varchar2(25)	Not null
Delivery location	Varchar2(25)	Not null

Table 3.6 Donated items table

3.4 BUSINESS MODEL CANVAS

The **Business Model Canvas** is a strategic management tool that helps in defining the key aspects of a business model. It provides a structured overview of the business through nine essential building blocks that define how the business creates, delivers, and captures value. For the **Food Donation Web Application**, the business model is designed to effectively connect donors with NGOs, ensuring food donations reach the needy in an efficient and transparent manner. Below is an expanded explanation of each building block of the Business Model Canvas as applied to the Food Donation Web Application.

3.4.1 Customer Segments

The **Customer Segments** for the Food Donation Web Application are divided into three main groups:

1. **Donors:**

- **Individuals or organizations** who are willing to donate surplus food. This includes households, restaurants, event organizers, and businesses that may have excess food or ingredients that can be donated.

2. **NGOs and Charitable Organizations:**

- **Non-governmental organizations** (NGOs) and charitable entities responsible for distributing the donated food to those in need. These organizations include food banks, shelters, community kitchens, and emergency relief programs.

3. **Volunteers:**

- **Individuals** who contribute their time and effort to pick up and deliver food donations from donors to the NGOs. Volunteers are crucial for the operational success of the platform and play a key role in the logistical aspects of food distribution.

3.4.2. Value Proposition

The **Value Proposition** of the Food Donation Web Application is centered around providing a simple, transparent, and efficient platform for food donation. It delivers the following key value to different stakeholders:

- **For Donors:**

- **Hassle-free donation process** through a user-friendly web application.
- **Transparency** by allowing donors to track the status of their donations, ensuring they know where their food is going and who it is helping.
- **Social impact** by contributing to the reduction of food waste and helping feed the underprivileged.
- **For NGOs:**
 - A **streamlined donation process** that enables easy tracking and management of food donations.
 - **Access to a broader network of donors**, facilitating the sourcing of food based on real-time needs.
 - **Data insights** on donation trends, allowing for better planning and resource allocation.
- **For Volunteers:**
 - **Organized platform** for managing pick-up and delivery schedules.
 - **Opportunity to contribute to a meaningful cause**, with an organized system to track performance and make an impact.

3.4.3. Channels

The **Channels** through which the Food Donation Web Application reaches its customers include:

- **Web Platform:** The primary channel where users (donors, NGOs, and volunteers) can interact with the system. The web application provides features such as registration, donation requests, tracking, and reporting.
- **Mobile Access:** In addition to the web application, the service will also be available through a **mobile application** (via platforms like Google Play Store), allowing users to access the platform on-the-go.
- **Social Media:** Social media platforms such as Facebook, Instagram, and Twitter will be used to raise awareness about the application, promote the benefits of food donation, and engage potential users.
- **Partnerships with NGOs:** NGOs can also serve as a channel to promote the application through their existing networks, newsletters, and events.

3.4.4. Customer Relationships

The **Customer Relationships** for the Food Donation Web Application will be established and maintained through:

- **Transparency and Trust:** Donors can track the status of their donations, which provides a sense of fulfillment and confidence in the platform.
- **Ease of Use:** The platform's simple, intuitive interface ensures that donors, NGOs, and volunteers can easily navigate the application, making their experience smooth and hassle-free.
- **Community Engagement:** Regular communication through email updates, newsletters, and notifications will keep users engaged and informed.
- **Feedback Mechanism:** A built-in feedback system will allow users to rate their experiences, helping to improve the platform's efficiency and effectiveness.

3.4.5. Revenue Streams

The **Revenue Streams** for the Food Donation Web Application are currently **non-revenue generating**, as the platform's main goal is to act as an intermediary between donors and NGOs. The application is **free to download** and does not include any direct payment modules.

However, potential future revenue streams could include:

- **Sponsorships or Partnerships:** Collaborations with corporate sponsors who may wish to support the platform in exchange for brand visibility and social responsibility credits.
- **Grants or Donations:** Funds from non-profit organizations or government grants to support the operational costs of the platform.

3.4.6. Key Resources

The **Key Resources** necessary for the functioning of the Food Donation Web Application include:

- **Technology Infrastructure:** The web application and mobile platform require hosting, databases, security measures, and user support systems to ensure the platform runs smoothly and safely.
- **Human Resources:** A team of developers and technical staff to maintain and update the platform. Additionally, customer service representatives will assist users with any issues or inquiries.
- **Volunteers:** Volunteers are crucial for picking up and delivering food donations from donors to NGOs. A well-organized volunteer system ensures that the logistics of the food donation process are efficiently handled.
- **Marketing and Outreach:** Resources for promoting the platform, including digital marketing tools, partnerships with NGOs, and social media outreach.

3.4.7. Key Activities

The **Key Activities** for the Food Donation Web Application include:

- **Platform Development and Maintenance:** Continuous development and enhancement of the web and mobile applications to ensure they meet user needs and remain secure and up-to-date.
- **Donor and NGO Engagement:** Encouraging donations through targeted outreach and building relationships with NGOs to ensure they have the tools to accept and distribute food efficiently.
- **Logistics Coordination:** Assigning volunteers to pick up and deliver food, ensuring that the food reaches the appropriate NGOs in a timely manner.
- **Monitoring and Reporting:** Tracking the flow of donations and providing real-time updates to donors. Generating reports that provide insights into donation patterns and needs.

3.4.8. Key Partners

The **Key Partners** for the Food Donation Web Application include:

- **NGOs and Charitable Organizations:** These organizations are the primary recipients of food donations. They are integral to the distribution process and collaborate closely with the application to ensure donations reach those in need.

- **Volunteers:** Volunteers are essential to the success of the platform, as they help with the physical transportation of food from donors to NGOs.
- **Corporate Partners:** Businesses that can provide support, such as sponsorships or logistical resources like vehicles for transportation.
- **Technology Providers:** Companies that offer hosting services, database management, and application development support.

3.4.9. Cost Structure

The **Cost Structure** for the Food Donation Web Application includes the following:

- **Development and Maintenance Costs:** Initial development costs to build the platform, as well as ongoing maintenance to keep the system functional and secure.
- **Marketing and Outreach Costs:** Expenses related to promoting the application, engaging with donors and NGOs, and creating awareness.
- **Operational Costs:** Costs associated with managing the platform, including volunteer coordination, server hosting, and user support.
- **Logistics Costs:** In the future, if necessary, covering the costs of transportation for volunteers (e.g., fuel, vehicle maintenance).

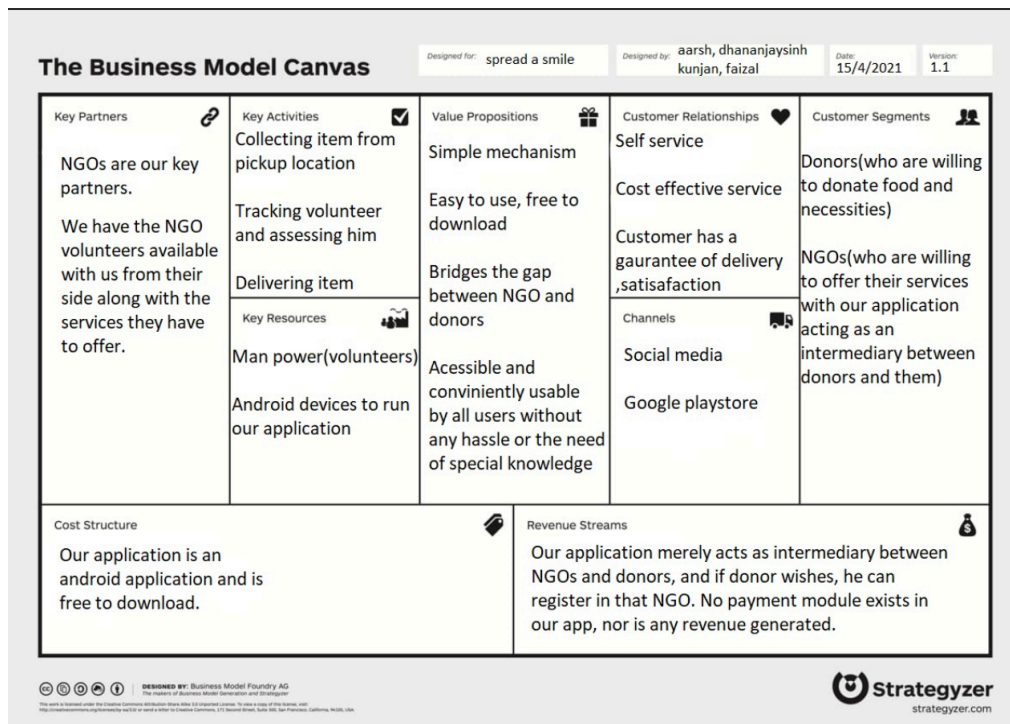


Fig. 3.13 BMC Canvas

CHAPTER 04

SYSTEM DESCRIPTION

4.1 SOFTWARE DESCRIPTION

Our project leverages the MERN stack (MongoDB, Express.js, React.js, Node.js), a robust and modern full-stack JavaScript framework, to ensure efficient development and seamless integration between the front-end and back-end systems. This stack is widely used in the development of dynamic web applications due to its versatility, scalability, and ability to deliver high-performance solutions.

4.1.1 MongoDB

MongoDB serves as the database for our application. As a NoSQL database, it is optimized for handling large volumes of unstructured data. Its document-oriented structure allows storing data in JSON-like format, enabling flexibility and easy adaptation to changing requirements. MongoDB also supports advanced features like horizontal scaling, indexing, and high availability, making it an ideal choice for modern web applications.

4.1.2 Express.js

Express.js is used as the backend framework in our project. Built on Node.js, it simplifies the creation and management of APIs and web servers. Express.js offers powerful routing capabilities, middleware integration, and error-handling mechanisms, allowing us to develop a secure and efficient backend to serve our application's needs.

4.1.3 React.js

For the frontend, we utilized React.js, a JavaScript library known for building interactive and dynamic user interfaces. React's component-based architecture ensures reusability and maintainability of code, while its virtual DOM enhances performance by optimizing updates and rendering. React allows us to deliver a responsive and user-friendly interface tailored to our project requirements.

4.1.4 Node.js

Node.js acts as the runtime environment for the backend development. Its non-blocking, event-driven architecture enables the handling of multiple simultaneous

requests efficiently. Node.js also facilitates smooth communication between the server, database, and client-side applications, serving as the backbone of our project.

4.2 KEY BENEFITS OF THE MERN STACK IN OUR PROJECT

1. **Unified Language:** JavaScript is used across the stack, streamlining development and enhancing collaboration among team members.
2. **Rapid Development:** With React.js on the frontend and Node.js on the backend, the MERN stack provides tools and frameworks to accelerate the development process.
3. **Scalability:** The stack supports scalable solutions, ensuring the application can grow to handle increasing user demands.
4. **Performance:** MongoDB's indexing capabilities and Node.js's event-driven model contribute to high-performance application design.
5. **Flexibility:** The modular architecture of the stack allows for seamless integration of additional features and third-party services.
6. **Community Support:** Being an open-source stack, MERN benefits from extensive community support, ensuring access to a wide range of resources and solutions.

By adopting the MERN stack, our project achieves a modern architecture that aligns with industry best practices for web application development, ensuring a robust, scalable, and maintainable system.

CHAPTER 05

SYSTEM IMPLEMENTATION

5.1 CODING STANDARDS

Coding standards are essential guidelines that ensure consistency, readability, and maintainability in software development. These standards provide a uniform appearance to the code, making it easier for multiple developers to collaborate effectively. The primary benefits of adhering to coding standards include:

- **Uniform Code Appearance:** Ensures that all code follows a consistent style, improving its visual structure and readability for different engineers.
- **Improved Maintainability:** Simplifies future updates and bug fixes by providing well-organized and self-explanatory code.
- **Error Detection and Debugging:** Facilitates easier detection and resolution of errors, as the code is systematically written and structured.
- **Code Reusability:** Promotes efficient reuse of code across different parts of the application, reducing redundancy and saving development time.
- **Sound Programming Practices:** Encourages developers to follow best practices, such as proper naming conventions, modularity, and documentation.
- **Enhanced Productivity:** Improves the overall efficiency of programmers by reducing ambiguities and complexities in the codebase.

By following these standards, the system ensures high-quality code that is scalable, efficient, and easy to maintain, even as the project grows in complexity.

5.2 TOTAL MODULES

The system comprises four main interconnected modules, each responsible for distinct yet interdependent functionalities. These modules work collaboratively to ensure the smooth operation of the application:

5.2.1 User Module

- Primary Role: The users of the application are primarily donors. They play a crucial role in contributing items or funds to support various NGOs.
- Key Features:
 - View a list of registered NGOs.
 - Select an NGO based on preference or cause.
 - Make donations to the chosen NGO.
- Purpose: Simplify the process of connecting donors with NGOs and promote seamless donation experiences.

5.2.2 Admin Module

- Primary Role: The admin acts as the central authority managing the system's overall operation and data integrity.
- Key Features:
 - Maintain the database of users, NGOs, and donations.
 - Add new NGOs to the system or remove inactive NGOs.
 - Ensure the system's smooth operation by monitoring activities and resolving conflicts or issues.
- Purpose: Provide centralized control to ensure data accuracy, operational efficiency, and system security

5.2.3 NGO Management Module

- Primary Role: Manage donation requests and coordinate between donors and volunteers.
- Key Features:
 - Receive and review requests from donors.
 - Assign volunteers to pick up items from donor locations.
 - Update the system with the status of requests and donations.
- Purpose: Bridge the gap between donors and volunteers, ensuring timely pickups and effective management of resources.

5.2.4 Volunteer Module

- Primary Role: Execute the logistics of picking up and delivering donated items.
- Key Features:
 - Accept tasks assigned by the NGO management team.
 - Collect items from donor-specified locations.
 - Deliver items to the respective NGOs and update the system.
- Purpose: Enable the physical collection and transportation of donations, ensuring the fulfillment of donor commitments.

CHAPTER 06

CONCLUSION

The development of the Anna-Mitra application has successfully addressed the challenges associated with facilitating food donations and fostering community engagement. This system was designed to connect donors with NGOs effectively, manage logistical operations efficiently, and streamline communication between all stakeholders, including volunteers and NGO management teams.

Throughout the project lifecycle, extensive efforts were made to ensure that the application met its functional and non-functional requirements. Rigorous testing validated the system's reliability, performance, and user experience, demonstrating its readiness for deployment.

Key accomplishments of the project include:

- **User-Friendly Interface:** A simple yet intuitive interface allows donors to navigate and contribute with ease.
- **Efficient Management:** The admin and NGO management modules ensure smooth operations and accurate record-keeping.
- **Seamless Logistics:** The volunteer module facilitates the efficient collection and delivery of donated items.
- **Real-Time Communication:** Notifications and real-time updates ensure transparency and engagement among users.

In conclusion, Anna-Mitra provides a robust, reliable, and scalable platform for fostering social good by promoting a culture of donation and collaboration. The project has achieved its objectives, making it a valuable tool for communities and organizations dedicated to reducing food wastage and supporting those in need.

CHAPTER 07

FUTURE SCOPE

While the Anna-Mitra application has successfully addressed its primary goals, there are several opportunities for further enhancement and expansion. These improvements aim to increase the system's impact, usability, and scalability to cater to a broader audience.

1. Integration of Advanced Technologies:

- **AI for Donation Prediction:** Implement AI algorithms to predict donation trends and recommend optimal NGO-donor matches based on donation history and geographical location.
- **Chatbots:** Introduce chatbots to assist users in real-time with queries and navigation.

2. Mobile Application Development:

- Develop dedicated mobile applications for Android and iOS to improve accessibility and user experience for donors, NGOs, and volunteers.

3. Support for Multi-Language Interfaces:

- Incorporate multi-language support to cater to a diverse user base across different regions.

4. Gamification:

- Introduce a gamification feature, such as donor badges and rewards, to encourage more participation and regular contributions.

5. Enhanced Security Features:

- Strengthen data security by implementing advanced encryption techniques and multi-factor authentication for users.

6. Scalability for Larger Audiences:

- Optimize the application to handle a larger user base and an increased volume of requests as the system gains popularity.

7. Partnerships with Corporations:

- Expand collaborations with corporate organizations to facilitate large-scale donations and Corporate Social Responsibility (CSR) initiatives.

8. Analytics Dashboard for NGOs and Admins:

- Provide an analytics dashboard to NGOs and administrators for monitoring donation trends, volunteer performance, and overall system impact.

9. Global Expansion:

- Adapt the system to support international NGOs and donors, enabling cross-border contributions and fostering global community support.

By incorporating these future advancements, Anna-Mitra can evolve into a comprehensive, scalable, and globally recognized platform for promoting social welfare and addressing the challenges of food wastage and hunger.

BIBLIOGRAPHY

Web Resources

1. Mozilla Developer Network (MDN). "JavaScript Documentation." <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.
2. React Documentation. "Getting Started with React." <https://reactjs.org/docs/getting-started.html>.
3. Express.js Documentation. "Express Guide." <https://expressjs.com/>.
4. MongoDB Documentation. "MongoDB Manual." <https://www.mongodb.com/docs/>.
5. Socket.IO Documentation. "Socket.IO Guide." <https://socket.io/docs/>.
6. "Build a Real-Time Collaborative Code Editor with React and Socket.IO." Medium, <https://medium.com/>.
7. "Full Stack Web Development with MERN." Tutorial on building applications using MongoDB, Express, React, and Node.js.
8. GitHub. "GitHub Documentation." <https://github.com/>.
9. Visual Studio Code. "VS Code Documentation." <https://code.visualstudio.com/>.

Books

1. Haverbeke Marijn. *Eloquent JavaScript: A Modern Introduction to Programming*. 3rd ed., No Starch Press, 2018.
2. Banks, Alex, and Porcello, Eve. *Learning React: Functional Web Development with React and Redux*. O'Reilly Media, 2017