

Caring Plates

Sai Manoj Dintakurti
MEng CSA
Virginia Tech
NCR Campus
saimanoj@vt.edu

Sai Krishna Gandhi Boyidi
MEng CSA
Virginia Tech
NCR Campus
krishnagandhi7@vt.edu

Divakara Rao Annepu
MEng CSA
Virginia Tech
NCR Campus
adivakararao@vt.edu

Aparna Alamanda
MEng CSA
Virginia Tech
NCR Campus
aparnaa@vt.edu

Radhika Bhumireddi
MEng CSA
Virginia Tech
NCR Campus
radhikab@vt.edu

Abstract—In a world where hunger and food waste coexist, numerous people are willing to help and contribute. Though these contributions may vary from acts of service to charities or donations, the shared objective is to feed people in need and distress while avoiding food wastage. We intend to improve the collaboration of people and organizations working with a common intent of reducing food wastage and providing food to the needy by means of providing a medium for them to connect. Taking a step forward to make this system coordinated, we propose a website that brings everyone involved in the cause to work together by connecting the restaurant owners with surplus food to people who are in need or NGO (Non-governmental organization) workers. Our project, “Caring Plates” enables restaurant owners to list the extra food availability on the website, so that anyone willing to outsource the food can request food pickup. A significant enhancement in the redistribution of excess food supply can be estimated with this platform when it comes into the picture.

BACKGROUND

Keywords: Non-Governmental Organization(NGO), Food Insecurity, Food Rescue, Food Waste Management, Sustainable Development Goals(SDG), Volunteer Engagement, React, Spring Boot.

I. INTRODUCTION

More than enough food is produced in the world to feed everyone on the planet. Yet, as many as 783 million people still go hungry globally[1]. About 30 million children are suffering from malnutrition[2]. Considering this as a serious problem, The United Nations framed “ZERO HUNGER” as one of the sustainable development goals and aims to achieve by 2030[3]. In our personal lives, we experienced food wastage during a wedding reception we hosted at our place. A huge amount of food that was catered for the event has remained prompting us to reach out to various NGOs by contacting them on their hotline numbers. After a lot of effort, though the food was outsourced, some of the supplies were spoiled by the time they reached those in need. Reflecting on this situation, a systematically managed medium for distributing surplus food could have helped in rescuing the

food in time. Our motivation for this project stemmed from this situation.

Technology has been at the forefront of solving many impactful humanitarian problems. A well-built web application connecting Restaurant Owners, NGOs, Volunteers, and people suffering from hunger can be a helping hand to some extent to address the problem of shortage of food supply for people in need through food rescue. By developing a reliable website, we plan to make our first attempt aligning with personal motivations and SDG (Sustainable Development Goals) for a better community within our vicinity and scope. This web application acts as a medium between Restaurants and NGOs for rescuing leftover food and providing meals to needy people. React and Spring Boot are used in the development of the “Caring Plates” website within the scope of this course.

II. MOTIVATING EXAMPLE

Consider a restaurant owner with connections to a limited set of Non-Governmental Organizations (NGOs) for outsourcing excess food. However, logistical challenges arise as the NGOs may not always synchronize with the restaurant’s timings for efficient collection and distribution. Enter “Caring Plates”, our website. In such a scenario, the restaurant owner can turn to the platform, listing the surplus food available. Any individual or NGO eager to collect the excess can easily claim it, ensuring timely rescue and distribution to those in need. Here, “Caring Plates” becomes the crucial link, seamlessly connecting the restaurant owner with potential food rescuers.

Conversely, consider an NGO with connections to a limited number of restaurants. There can be instances when these restaurants may not have surplus food to distribute. In this situation, the NGO faces challenges in fulfilling its mission to feed those in need. “Caring Plates” transforms

this situation by offering a diverse array of restaurants listing surplus food on the website. The NGO gains access to a broader pool of potential food sources, ensuring a consistent supply to meet the demands of their humanitarian efforts. In this way, “Caring Plates” plays a vital role in connecting NGOs with a wider network of restaurants, contributing to a more effective and coordinated approach in the battle against both hunger and food waste.

III. RELATED WORK

The work[4] has highlighted the problem of daily food wastage. Some methods have been suggested and are in use to cut down on food wastage and put food waste to good use. One such approach is based on the AHP-TRIZ design model, which explains how an intelligent household food waste recycling product works[5]. This involves creating a smart, semi-automated recycling product and a user-friendly mobile app. On the other hand, to tackle food waste, there are NGOs offering to redistribute food, and they’ve set up hotline numbers and email services for this purpose[6]. Plus, there are online platforms connecting people with extra food to those who need it[7]. We are taking this idea a step further by rewarding volunteers by rating them based on their number of deliveries when they help distribute excess food. NGOs(Non- governmental organizations) pay the incentives to the volunteers basis their rating.

IV. PROPOSED WORK

Creating a restaurant and NGO connecting application involves several key features to ensure efficient communication and coordination between restaurants and NGOs. Here are some essential features of our application:

A. User Management and Interaction:

- *User Authentication:* User registration and login for restaurants, individuals and NGO representatives. User login credentials are verified against their registration details every time they login.
- *Restaurant Profile Management:* Restaurants can create and manage their profiles. Restaurants can update information such as restaurant name, Address, contact number, and landmark.
- *NGO Registration and Profile:* When users register on the website, they have the option to select their role as a restaurant owner, individual, or NGO volunteer. If they choose the role of a restaurant owner, they gain access to the platform with specific functionalities tailored to their needs. As a restaurant owner willing to contribute by donating food, they are provided with the option to list the available food items and manage their profile page.
- *Individual/Volunteer Registration and Profile:* If users opt for the role of an individual or NGO volunteer during the registration process, they unlock the ability to claim

food from the available listings on the platform. Upon successfully claiming and delivering food, these users are rewarded with points.

- *Email Notifications and Alerts:* When food is claimed on the platform, email notifications are automatically dispatched to the respective users. For restaurant owners, the email contains pertinent details about the individual who claimed the food. This includes the claimant’s name and phone number, providing restaurant owners with essential information for coordination.
- *Customer Support:* We have provided a support system for users to report issues or seek assistance.

Conversely, individuals and NGO volunteers receive detailed emails specifying the particulars of the claimed food. This includes comprehensive information about the associated restaurant, such as the restaurant’s name, contact number, and location. Additionally, the email includes specifics about the food item to be picked up, such as its name, available quantity, and the designated pickup time. These comprehensive details enable individuals and NGO volunteers to efficiently coordinate and collect the donated food.

B. Donation and Food Management:

- *Food Donation Listings:* Restaurants can create listings for surplus food items available for donation. Include details like food type, quantity, expiration date, and pickup time.
- *Food Pickup Requests:* NGOs or Volunteers can browse available food listings and request pickups.
- *Donation History:* All history of all donations made by restaurants and received by NGOs or Volunteers can be viewed on the website.
- *Rewards for the volunteers:* Upon successfully claiming and delivering food, users are rewarded with points. These reward points serve as a recognition and incentive system, acknowledging their contribution to the food donation initiative. This feature encourages active participation from individuals and NGO volunteers, fostering a collaborative and engaging environment on the website.

C. Information and Accessibility:

- *Geolocation and Maps:* The email notification dispatched upon claiming an available food listing incorporates a valuable feature— a Google Maps link providing the geo location of the respective restaurant. Upon clicking this link, users are seamlessly directed to Google Maps for streamlined and user-friendly navigation. This enhancement aims to facilitate and simplify the process for individuals and NGO volunteers, ensuring they can effortlessly locate and reach the restaurant for food pickup.

V. METHODOLOGY

“Caring Plates” is a tech-driven platform that aims to connect stakeholders in the food donation ecosystem. ReactJS[8] powers the front end, providing a dynamic and

engaging user interface, while Spring Boot serves as the robust backend framework, ensuring optimal performance and efficiency across various functionalities.

A. IMPLEMENTATION

- *Client-Server Architecture:* In crafting the implementation design for our system, we strategically opted for a client-server architecture to foster concurrent development on both the frontend and backend components. Client-server architecture is a computing model in which client devices (such as computers, smartphones, or tablets) request services or resources from servers. In this architecture, the server provides resources and services to multiple clients simultaneously.
- *Monolithic Architecture:* Given the manageable scale of user and food donation entities, a judicious choice was made to embrace a monolithic architecture. This decision not only streamlined the development process but also ensured an efficient and cohesive system. Monolithic architecture refers to a traditional software design approach where an application is built as a single, indivisible unit. In this architecture, all components of the application are interconnected and run on a single platform.
- *React JS:* On the frontend, React emerged as our technology of choice, where we adeptly harnessed the power of reusable components to seamlessly collect user information across diverse forms. React is a free and open-source front-end JavaScript library for building user interfaces based on components. It provides good maintenance services for keeping the application running quickly and dynamically.
- *Java Mail Service:* The application employs the Java Mail Sender to facilitate the seamless sending of notification emails to individuals who have claimed available food donations. This email notification is crucial for keeping users informed about their successful claims and plays a pivotal role in fostering engagement and enhancing the overall user experience. JavaMail is a Java API that allows developers to send, receive, and manipulate email messages, including support for various protocols such as SMTP, IMAP, and POP3.
- *Google Maps JavaScript API:* The notification emails received by individuals contain essential details, including the location of the corresponding restaurant. To provide the appropriate location of the navigation for the food claim, the location of the restaurant is sent to the individual using Google Maps JavaScript API. The Google Maps JavaScript API is a set of programming interfaces and tools provided by Google to enable the integration of Google Maps functionality into web applications.

Our approach was further fortified by the astute utilization of Axios calls and strategic re-rendering of React com-

ponents, resulting in a marked enhancement of system performance. This meticulous selection of architectural paradigms and technologies enabled us to create a robust and scalable solution tailored to the unique needs of our application.

B. TESTING

In our testing strategy, we utilized Mockito specifically for testing service classes, leveraging its mock object capabilities to isolate and verify the behavior of these classes. Additionally, for web API testing, we employed a robust framework, such as RestAssured, to validate the functionality and endpoints of our APIs. RestAssured is a popular Java-based framework designed for simplifying API testing, providing a fluent interface to expressively define and execute API requests. This framework allowed us to perform comprehensive testing, including verifying response status codes, payload content, headers, and other aspects of our web APIs. Combining Mockito for service classes and RestAssured for web API testing has fortified our testing suite, ensuring the reliability and effectiveness of both backend service components and external API interactions. Meanwhile, on the frontend, manual testing played a crucial role in validating user interface and experience.

VI. DEPLOYMENT PLAN AND MAINTENANCE

If we were to release the Caring Plates website for users, our deployment plan and maintenance would be as follows

A. INFRASTRUCTURE SETUP

The deployment of the Food Rescue website will be hosted on Amazon Web Services (AWS) to leverage its scalable and reliable infrastructure.

- *EC2 Instances for Java Backend:* We will utilize AWS EC2 instances to host the Java backend of the website. These instances will handle the server-side logic and processing. Amazon Elastic Compute Cloud (EC2) instances are virtual servers provided by Amazon Web Services (AWS) that allow you to run applications in the cloud.
- *RDS for Database:* Amazon RDS would be implemented to host the database for the website. RDS offers a managed database service, ensuring high availability, scalability, and ease of maintenance. Amazon RDS (Relational Database Service) is a managed relational database service provided by Amazon Web Services (AWS). It is designed to simplify the process of setting up, operating, and scaling a relational database in the cloud. It supports several popular database engines.
- *S3/CloudFront for React Frontend:* React frontend would be hosted on Amazon S3 and utilize CloudFront as a content delivery network (CDN) to ensure low latency and high performance for end-users. Amazon S3 (Simple Storage Service) and Amazon CloudFront are AWS services commonly used in conjunction to serve static assets for a React frontend application.

B. BUILD AND PACKAGE

To streamline the build and packaging process, AWS CodeBuild will be employed.

- *AWS CodeBuild for JAR and Static Builds:* Implement AWS CodeBuild to automate the creation of production-ready JAR files for the Java backend and static builds for the React frontend. This ensures consistency and efficiency in the deployment process. AWS CodeBuild is a fully managed continuous integration service provided by Amazon Web Services (AWS). It is designed to compile source code, run tests, and produce software artifacts that are ready to deploy. CodeBuild supports a variety of programming languages, build tools, and workflows, making it versatile for different types of projects.

C. DATABASE MIGRATION

For seamless database migration, AWS Database Migration Service (DMS) or custom scripts will be employed.

- *AWS DMS or Migration Scripts:* We would choose between AWS Database Migration Service or custom migration scripts to migrate the database. This step ensures a smooth transition of data to the production environment. AWS DMS, or AWS Database Migration Service, is a fully managed service provided by Amazon Web Services (AWS) that allows you to migrate databases seamlessly and securely. AWS DMS supports various database engines, including Amazon Aurora, MySQL, MariaDB, PostgreSQL, Oracle, Microsoft SQL Server, and others. It facilitates both homogeneous migrations (e.g., Oracle to Oracle) and heterogeneous migrations (e.g., Oracle to Amazon Aurora).

D. SECURITY SETUP

Security is paramount for the Food Rescue website, and AWS provides robust tools to achieve this.

- *Security Groups and Network ACLs:* Configuring AWS Security Groups and Network ACLs would control inbound and outbound traffic, ensuring a secure environment for the application. In Amazon Web Services (AWS), Security Groups and Network Access Control Lists (Network ACLs) are two important components that help control access to resources within a Virtual Private Cloud (VPC). They serve as mechanisms for implementing network security policies but operate at different layers of the networking stack.
- *HTTPS with AWS Certificate Manager:* Enabling HTTPS by obtaining SSL/TLS certificates from AWS Certificate Manager, enhances the security of data in transit. AWS Certificate Manager (ACM) is a service provided by Amazon Web Services that simplifies the process of obtaining, managing, and deploying SSL/TLS (Secure Sockets Layer/Transport Layer Security) certificates for your applications and websites.

E. LOAD BALANCING

Efficient traffic distribution will be achieved using an Application Load Balancer (ALB).

- *Application Load Balancer (ALB):* Implementing an ALB to distribute incoming traffic across multiple EC2 instances, improves scalability and availability. An ALB is a service provided by AWS that distributes incoming web traffic across multiple targets, such as Amazon EC2 instances, containers, and IP addresses, within one or more availability zones.

F. MONITORING AND LOGGING

Continuous monitoring and centralized logging are essential for maintaining a healthy application.

- *Amazon CloudWatch for Monitoring:* Setting up monitoring using Amazon CloudWatch tracks key metrics, detects anomalies, and ensures optimal performance. Amazon CloudWatch is a monitoring and observability service provided by Amazon Web Services (AWS). It allows users to collect and track various metrics, collect and monitor log files, and set alarms. CloudWatch provides a centralized platform for monitoring the performance and health of AWS resources and applications.
- *CloudWatch Logs for Centralized Logging:* Utilizing CloudWatch Logs centralizes logs, facilitates easier troubleshooting and analysis of application behavior. CloudWatch Logs simplifies log management, providing a scalable and secure solution for analyzing, troubleshooting, and gaining insights from log files.

G. CI/CD

Establishing a robust CI/CD pipeline will automate the deployment process and enhance development workflows.

- *AWS CodePipeline for CI/CD:* Implementing AWS CodePipeline would facilitate continuous integration and continuous deployment. This pipeline will automate the testing, building, and deployment processes, ensuring a swift and reliable release cycle. AWS CodePipeline is a fully managed continuous integration and continuous delivery (CI/CD) service provided by AWS. It helps automate and streamline the software release process, allowing development teams to deliver code changes more rapidly, reliably, and consistently. CodePipeline facilitates the end-to-end orchestration of your release pipeline, from source code changes to deployment to production.

VII. FUTURE WORK AND LIMITATIONS

As the project is in its nascent stages, there is considerable potential for future enhancements and the addition of new features to further elevate the usability and functionality of the website. Potential areas that are the limitations in the current setup that need expansion include the following with details

- *User Authentication:* The authenticity of the restaurant owners and NGO volunteers can be verified.

- *Rating and Feedback:* NGOs and restaurant owners can be allowed to rate and provide feedback on the restaurant's donations. This encourages transparency and accountability.
- *Payment and Expense Tracking:* NGOs can be allowed to track expenses related to food pickup and distribution. This supports donations to NGOs for operational costs.
- *Accessibility and Multilingual Support:* The application can be made accessible to all users, including those with disabilities. Multiple languages can be supported to cater to a diverse user base.
- *Scheduling and Coordination:* A scheduling system can be introduced to coordinate food pickups. This allows restaurants, NGO volunteers and individuals to confirm, reschedule, or cancel pickups.

VIII. CONCLUSION

In summary, "Caring Plates" addresses the stark dissonance between global hunger and the excessive food produced by introducing an innovative web platform. The project hinges on the fusion of technology and compassion, serves as a collaborative hub connecting restaurant owners, NGO volunteers, and individuals with a shared mission of curbing food waste and providing nourishment to those in need.

The implementation journey encompasses a client-server architecture, a monolithic design for efficiency, and the strategic application of technologies like ReactJS, Java Mail Sender, and Google Maps JavaScript API. Rigorous testing using Mockito and RestAssured ensures the reliability of the system, while the deployment plan envisions leveraging Amazon Web Services (AWS) for scalability, reliability, and security. "Caring Plates" transcends being a mere technological solution; it symbolizes a compassionate initiative, fostering transparent processes, engaging environments, and a reward system for volunteers. As we conclude, "Caring Plates" is not just a digital innovation but a heartfelt endeavor contributing tangibly to the global battle against hunger and food waste. It represents a vision where technology seamlessly aligns with empathy, creating positive transformations in communities.

REFERENCES