**San Francisco State University**

**CSC648/848 - Software Engineering**

**Milestone 1**

**Section #1 Team 07**

**September 27, 2023**

**Team Members: Lusi Acuna Mendez, Jooho Chang, Gwangwoo Lee, Leslie Mora Ponce, Dorrie Shen, Vedang Sakxena**

Table of Contents

[**1. Executive Summary**](#_jj1deskgl451) **2**

[**2. Personal and User Stories**](#_w5n4y42riw1l) **3**

[**3. Data Definitions**](#_ifj4z8dk5ysj) **4**

[**4. Initial List of Functional Requirements 5**](#_jmnqiwhknwmy)

[**5. List of Non-Functional Requirements**](#_d7u80pbprx8w) **8**

[**6. Competitive Analysis 8**](#_fojnmdf7kayi)

[**7. High-Level System Requirements 9**](#_85pc2xp204vb)

[**8. Team 9**](#_yj7d8doxa8ih)

[**9. Checklist 10**](#_xexv5q7zg6qr)

# Executive Summary

The United States holds the unenviable distinction of being the world's largest producer of food waste, as reported by the Natural Resources Defense Council (NRDC). This issue poses a considerable challenge for the restaurant industry, with an alarming estimated annual food waste of 11.4 million tons in the United States alone. The repercussions of this wasteful practice extend to both financial and environmental domains, exacting substantial costs on businesses and negatively impacting the planet.

According to the National Restaurant Association, the restaurant sector contributes to a staggering 22 to 33 billion pounds of food waste each year in the United States. Modern Restaurant Management's findings align with this, reporting an annual production of 915,400 tons of food waste within the restaurant sector. Notably, NPR highlights that food waste from restaurants alone constitutes 15% of the total food waste that ultimately finds its way into landfills. *(Source: moveforhunger.org article based on food waste management in restaurants)*

Several factors contribute to this issue. Full-service restaurants, with their diverse and complex menus, tend to generate more waste due to a wider range of ingredients and dishes. Similarly, establishments offering buffet-style or self-service options face the challenge of preparing for unpredictable customer demand, often resulting in a surplus of food that ultimately goes to waste.

From an environmental perspective, food waste contributes significantly to greenhouse gas emissions, primarily in the form of landfill waste. The financial implications of food waste are equally substantial, exerting a considerable financial burden on businesses in the restaurant industry.

Our Web-Application ‘Last Call’, provides a simpler solution to food waste management. It is a platform that enables Restaurants to recycle their leftover food and sell it at a cheaper price. This application connects customers to restaurants and stores that have surplus unsold food. The food on the app is priced at one-third its original price. This reduces the waste of food that would otherwise be discarded.

‘Last Call’ provides a simpler solution for Restaurants to recycle their left-over food supplies, and also earn extra money in the process. It is seemingly better than bearing the loss of food wastage, and the cost implied on disposing the waste into landfills.

As for the users, they’re provided with good quality food at a cheaper price. This platform is beneficial for the category of users who are not equipped enough to cook for themselves and are regularly searching for restaurants that provide cheap food.

# Personal and User Stories

* **Student: Fernanda Ortiz**
  + Twenty-one-year-old Fernanda is a first-generation student. She lives by herself in a one-room shared home in San Francisco because the rest of her family is undocumented.
  + She works a full-time job in order to be able to pay her $1,200 monthly rent, meals, and tuition.
  + She works as a hostess from 5:00 pm-1:30 am
  + She goes full-time for her first class starting at 11:30 am
  + Fernanda commutes to and from school for 1 hour every day, most days she has no time to cook at home so she ends up getting food at the mall that's right across from her school, but she ends up spending around $10-15 for a good breakfast
  + Her last class ends at 4:30 pm so she has to get lunch at school again for the total food in school she ends up spending $35-40 dollars every day
  + The only free time she has is spent studying since she wants to go to a good medical school
* **Minimum Wage Worker: Johoo Lee**
  + Twenty-nine-year-old Johoo works in a sushi restaurant
  + He lives in a San Francisco apartment where rent can be too high $2,700 dollar every payment
  + Johoo works extra shifts throughout the week
  + He loves to cook but many times of the week he is really tired and ends up getting fast food which he ends up spending around $60-70 for a meal for him and his mother.
* **Restaurant Owner: Mariano** 
  + Mariano was a person who grew up in one of the poorest communities, and he has always known how hard it is to get food for a good meal
  + Since he was a child he dreamed of becoming a chef and owning his own restaurant and it took him hours of dedication
  + Mariano has decided that he doesn't want food to be wasted and has decided to sell with an 80% discount
  + He thinks is a good waste to reduce waste and he believes throwing food away is bad luck for his business

# Data Definitions

1. Customer
   1. A Customer is a registered user who chooses items displayed on our website
   2. A Customer can be able to create at most one account by using a unique email address
   3. A Customer shall be located in only one Location
2. Restaurant
   1. A Restaurant is a registered business user who can sell items they want to Customer
   2. A Restaurant shall be able to create only one account
   3. A Restaurant can create the food list they have under their account
   4. A Restaurant can add food or remove food from their account
   5. A Restaurant shall be located in only one Location
3. Owner
   1. An Owner is the only account that can add a Customer and a Restaurant account
   2. An Owner is the only account that can remove Customer and Restaurant account
   3. An Owner is the only account that can control the Penalty System to the Customer account
   4. An Owner is the only account that can add Base Point Payment
4. Food
   1. Food is the item that can only displayed by a Restaurant
   2. Food is an item that can only purchased by the Customer
   3. Each food item can only belong to one Restaurant
   4. Each food item can only purchased by at most one Customer
5. Base Point Payment
   1. Each Base Point can used by the Customer
   2. Base Point Payment can only added by the Owner
6. Penalty System
   1. The Penalty System can only be controlled by the Owner
   2. Penalty System shell has many Customers
7. Reservation
   1. Each Reservation has a unique ID
   2. Each Reservation belongs to one Customer
   3. Each Reservation belongs to one Restaurant
   4. Each Reservation can have many foods
8. Location
   1. A Location shall have many Restaurants
   2. A Location shall have many Customers
9. Review
   1. A review shall be created by zero or many customers
   2. A review shall be owned by one restaurant
10. Open Status
    1. An open status shall represent one restaurant’s status.

# Initial List of Functional Requirements

1. Search Bar
   * The Search Bar function allows customers to enter search queries to find specific restaurants. For this function, the users can type text and there should be a button to search.
2. Open Now
   * This feature displays restaurants that are currently open. This function needs to be displayed next to each restaurant.
3. Log In
   * Log In function allows users to access their accounts. For this function, the user can type username and password and there should be a login button.
4. Sign Up
   * Sign-up function allows users to create their accounts. For this function, the customers enter their name, email, and password, and confirm the password. Also, restaurants enter restaurant names, addresses, email, passwords, and confirm passwords to sign up for our application. After that, all user information is stored in the database.
5. Base Point System
   * This function allows customers to earn points that they can redeem for discounts. This needs to be displayed on the account page and when customers check out.
6. Favorite Restaurant List
   * This function allows customers to add restaurants they like to a list of favorites for easy access and reference. For this function, we need to make a button next to the restaurant's name.
7. Log Out
   * Users can log out of their accounts to secure their information and end their session. This function could be done on every page.
8. Search History
   * This feature keeps a record of the customers’ past search queries for reference. This function can looked up on the account page that showed up search history in 2 weeks.
9. Platform logins.
   * Allows users to log in to their accounts using different authentication methods such as Gmail and Facebook. This function needs to be placed on the login page with the platform icon.
10. Log-In Authentication
    * This refers to web cookies that are used for authentication within the platform to enhance user experience.
11. Checkout / Reservation
    * Users can use this function to make a reservation at the restaurant. This function needs to be placed on the restaurant's page after selecting food.
12. Penalty System
    * This function may require fees for canceling a reservation without notice. Users can check their penalty on their account page.
13. Code for Reservation order
    * Customers use this code when they pick up their food to identify their specific order. This code will be displayed on the top of every page.
14. List up Restaurant
    * This function displays a list of restaurants to customers on the restaurant’s page.
15. Order History
    * Users can view a record of their past orders. Users can check their order history on their account page.
16. Find Restaurants around the User
    * This function displays a list of restaurants based on customers’ current location on the restaurants page.
17. Suggest Restaurant
    * This function displays a list of popular restaurants to customers on the restaurants page.
18. Restaurant Food Stocks.
    * This feature displays the availability of specific menus at restaurants on the restaurants page.
19. Show How much money we saved.
    * This displays the amount of money a user has saved by using our platform. This will be showed on users’ account page.
20. Review
    * This function allows customers to review their order. This feature will show up after users get their order
21. Rate Restaurants
    * This function displays the rate of each restaurant on the restaurants page.

# List of Non-Functional Requirements

1. Compatibility
   1. Application can be used on several mobile browsers including Chrome, Mozilla, and different web searches available in the market.
2. Development requirement
   1. Data shall be stored in the team’s chosen database technology on the team’s deployment server Amazon AWS.
   2. The code in the master branch of the team's GitHub repo should be well maintained and tested, and guarantee working at any time.
3. Storage
   1. The Application will be able to have business preparation, food listing, real-time availability, order placement, and pick-up times
4. Usability
   1. The Application shall be easy to use with an interactive search bar, a method to be able to pay face-to-face to the restaurants while supporting a point-based system that allows free items in the application.

# Competitive Analysis

**features that make the app better than ‘too good to go’**

1. Food Reservation System
2. Restaurant Ratings
3. Tracking how much money is saved

| Too good to go ( Competitor ) | Last Call (Our App) |
| --- | --- |
| The app features a restaurant rating system. This allows users to review the restaurants they've visited and share their opinions with others. | We discovered that our competitor's review system is rather ineffective. Users who have never bought food at a restaurant are unable to view reviews until they make a purchase. Essentially, they can't see others' opinions about the restaurant before buying food. In contrast, our app will allow new users to access the review system freely, even before they make a payment. |
| The app displays how much money users save when they make a payment. | Our app will introduce a feature that displays the amount of money a user saves when purchasing food. Additionally, the app will provide users with their total savings. |
| The app features a food reservation system. When a user makes a reservation for food, they must pay within the app. | Our app also includes a food reservation system. However, users aren't required to pay when they make a reservation. To deter no-shows, we've implemented a penalty system. Users can make payments in person.. |

# High-Level System Requirements

1. Server Host: AWS EC2 1vCPU 2GB RAM
2. Operating System: Ubuntu 22.04 Server
3. Database: Mysql 8.0
4. Server Side Language: JavaScript
5. Web server: Nginx 1.25
6. Web Application Framework: Express
7. Additional Tech: React
8. IDE: VS Code

# Team

Team Members

* Dorrie - GitHub Master, Frontend Engineer
* Jooho - Frontend Leader
* Leslie - Frontend Engineer
* Gwangwoo - Backend Master
* Vedang - Scrum Master, Backend Engineer
* Luis - Group Leader, Backend Engineer

Study Plan

Front End

1. ~ by end of September, our team will be studying React, specifically JSX Grammar, component, Hooks (useState, useEffect and etc.), Reducer, Immer
2. ~ by October 14th, our team will be studying React including React Router and React Query
3. ~ by end of October, we will study advanced React grammar and custom hooks
4. In November, we will focus more on the interaction between backend and frontend, as well as the functionality of components.

Back End

1. ~ by end of September, our team will be studying basic Express usage.
2. ~ by October 14th, our team will be studying MySQL, such as how to create tables and use foreign key to make relationships between two tables.
3. ~ by end of October, our team will study for advanced Express usage.

# Checklist

* ~~The team found a time slot to meet outside of the class.~~
* ~~Scrum Master shares meeting minutes with everyone after each meeting.~~
* ~~GitHub master chosen.~~
* ~~Everyone sets up their local development environment from the team’s git repo.~~
* ~~The team decided and agreed together on using the listed SW tools and deployment server.~~
* Team-ready and able to use the chosen back/front-end frameworks.
* For each technology (front/back-end/DB/cloud), the team decides who will lead the study of each technology and what will be the specific goal of the study within one month from the M1 announcement.

Ex: implement DB scheme for main data items

* If you list a detailed study plan for this, earn extra points!
* ~~The team lead ensured that all team members read the final M1 and agreed/understood it before submission.~~