Milestone 4: Project Implementation

By Ryan Davis, Ivan Farfan, Johan Jaramillo, and Cory Vitanza

We pledge our Honor that we have abided by the Stevens Honor System

Table of Contents

Introduction

Our group was tasked with creating a meal sharing application that allows users to host and join meals. Users can assume both roles under one account, and hosts are able to share information on a meal and invite others to join them while guests can view meals nearby and request to join the meal. The current milestone employs the following UML diagrams: use case diagram, activity diagram, class diagram, and sequence diagram. In doing so, the implementation of the design will be much smoother and more efficient, as the group will have an improved understanding of the system design.

In this milestone, our group concentrated on enhancing interactivity and user engagement within the Meal Sharing Application. Key features implemented included guest-side meal browsing with filters for location, date, and meal details, as well as a request system allowing guests to ask to join meals. On the host side, we developed functionality to view, accept, or reject incoming requests. Additionally, we introduced a structured rating system for both guests and hosts after a meal concludes. These updates mark a significant progression in user-driven functionality, emphasizing dynamic interactions between hosts and guests while ensuring smooth database integration and flow continuity.

Team Collaboration

- **Ryan Davis**
    - Implemented request logic allowing guests to request joining a meal and hosts to approve the request.
- **Ivan Farfan**
    - Manages Supabase DB. Implemented guest's ability to browse meals based on location, time, and meal information.
- **Johan Jaramillo**
    - Implemented rating system for users and hosts, as well as hosts rejecting a guest's join request.
- **Cory Vitanza**
    - Focused on documentation. Performed bug testing/fixing.

Note that these contributions / graphs do **not** include merge commits.
(Contributions per week to main, excluding merge commits)

Introduction to Project Repository

https://github.com/IvanFarfan08/MealSharing

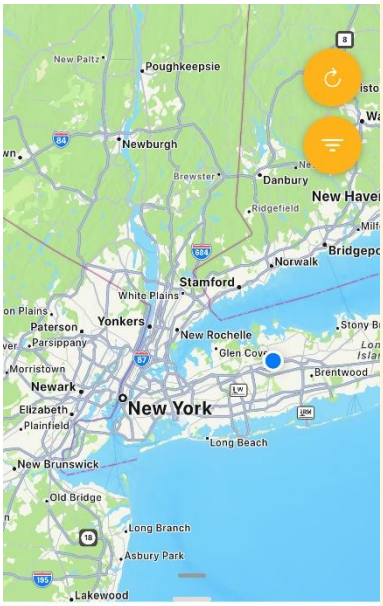
Since this project is deployed on GitHub, the team implemented GitHub tools to maintain proper version control. With different branches, the team was able to develop the application in different components, allowing each developer to write different sheets of the app. Furthermore, when a developer completed a task on their branch, they were able to create a pull request, in which another developer can review the code before merging it into the main branch for deployment. Lastly, the team utilizes a features board, acting as an agile board, which manages the assignment and completion of tasks. With such measures, the team efficiently provided a functional first implementation of the Meal Sharing Application.

In the repository, a structured system of folders allows for the clear development of the application. The assets folder holds icons and images for the user interface. The components folder holds the typescript files, which hold the logic and the programming for the applications implementation. The lib folder holds configuration files for the Supabase database we are implementing. Other configuration files in the repository provide a set of rules and requirements for a machine to run the program. Such tools are essential for developers on different machines to have equivalent run time configurations. This repository's design includes simplicity and configurations allowing for simple and clear development.

Summary of Changes

Since milestones 2 and 3, milestone 4 has had a focus on the action of joining a meal based on a variety of arguments. Previously, joining a meal entailed clicking on a meal from a list of meal cards, where the user would receive a notification stating that they have joined the meal. In milestone 4's implementation meals are presented on a map, relative to the user's location, where users can choose to filter and join meals. There is also a new page which displays meals a user has joined, requested, is hosting, and previous meals to be reviewed for guests and hosts.

The implementation of milestone 4 includes browsing meals on a map, filtering such meals, requesting to join a meal, viewing and accepting or denying requests, and a rating system for users. The implementation of such functionalities has called for a change in design, hence the development of the My Meals page. Now, users have a page to display all the meals they are involved in.



Available Meals 1 meals found

Find My Meals Host Notificatio... Account

11:21

Account

Username:
ryry91021@gmail.com

Email:
ryry91021@gmail.com

User ID:
e1160d03-f873-4218-b849-17ac117dff1d

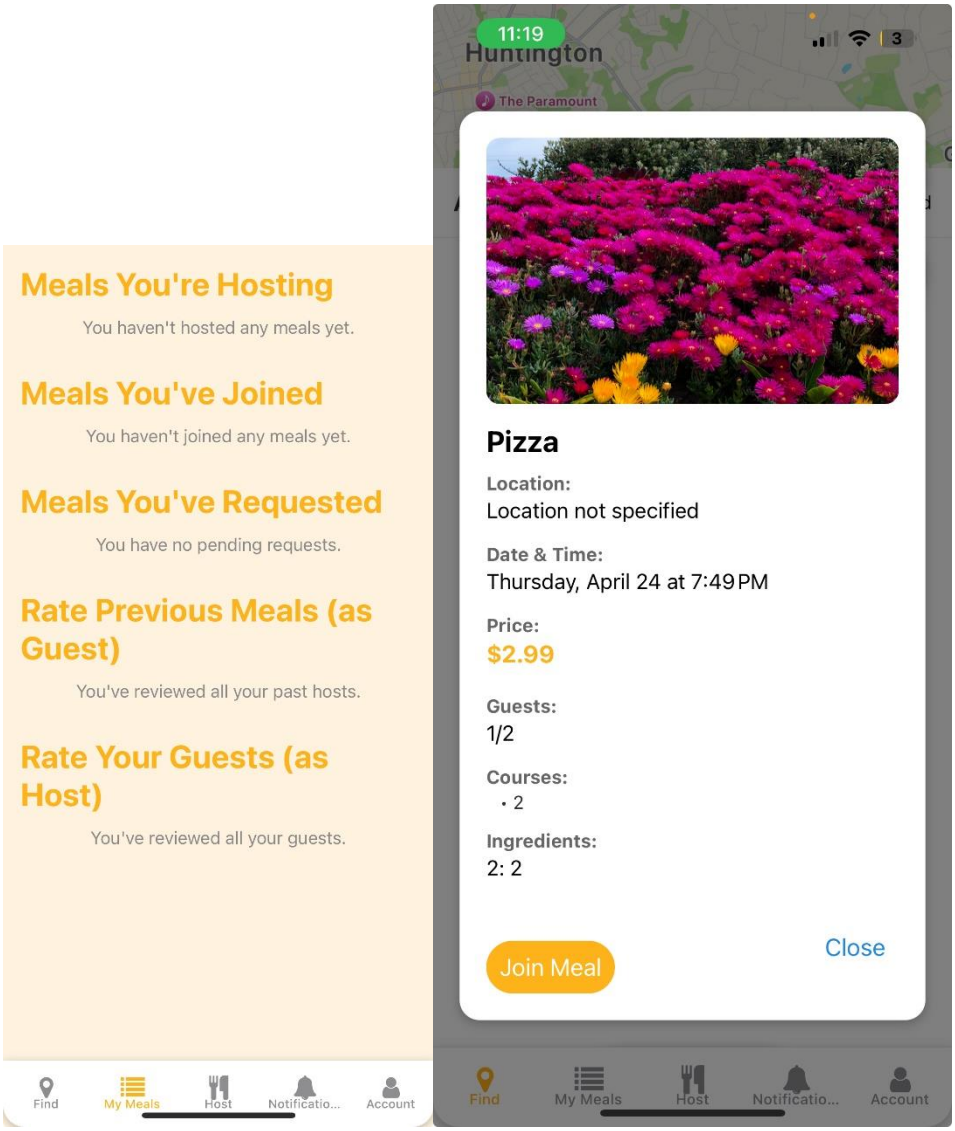
Average Rating:
4.0 / 5.0

Reviews

★★★★☆
No comment
4/16/2025

Log Out

Find My Meals Host Notificatio... Account


Meals You're Hosting
You haven't hosted any meals yet.
Meals You've Joined
You haven't joined any meals yet.
Meals You've Requested
You have no pending requests.
Rate Previous Meals (as Guest)
You've reviewed all your past hosts.
Rate Your Guests (as Host)
You've reviewed all your guests.
Find
My Meals
Host
Notificatio...
Account
11:19
Huntington
The Paramount
Pizza
Location:
Location not specified
Date & Time:
Thursday, April 24 at 7:49 PM
Price:
$2.99
Guests:
1/2
Courses:
• 2
Ingredients:
2: 2
Join Meal
Close
Find
My Meals
Host
Notificatio...
Account

Available Meals 1 meals found

Pizza

Thu, Apr 24 at 7:49 PM

$2.99

Guests: 1/2

Join Meal

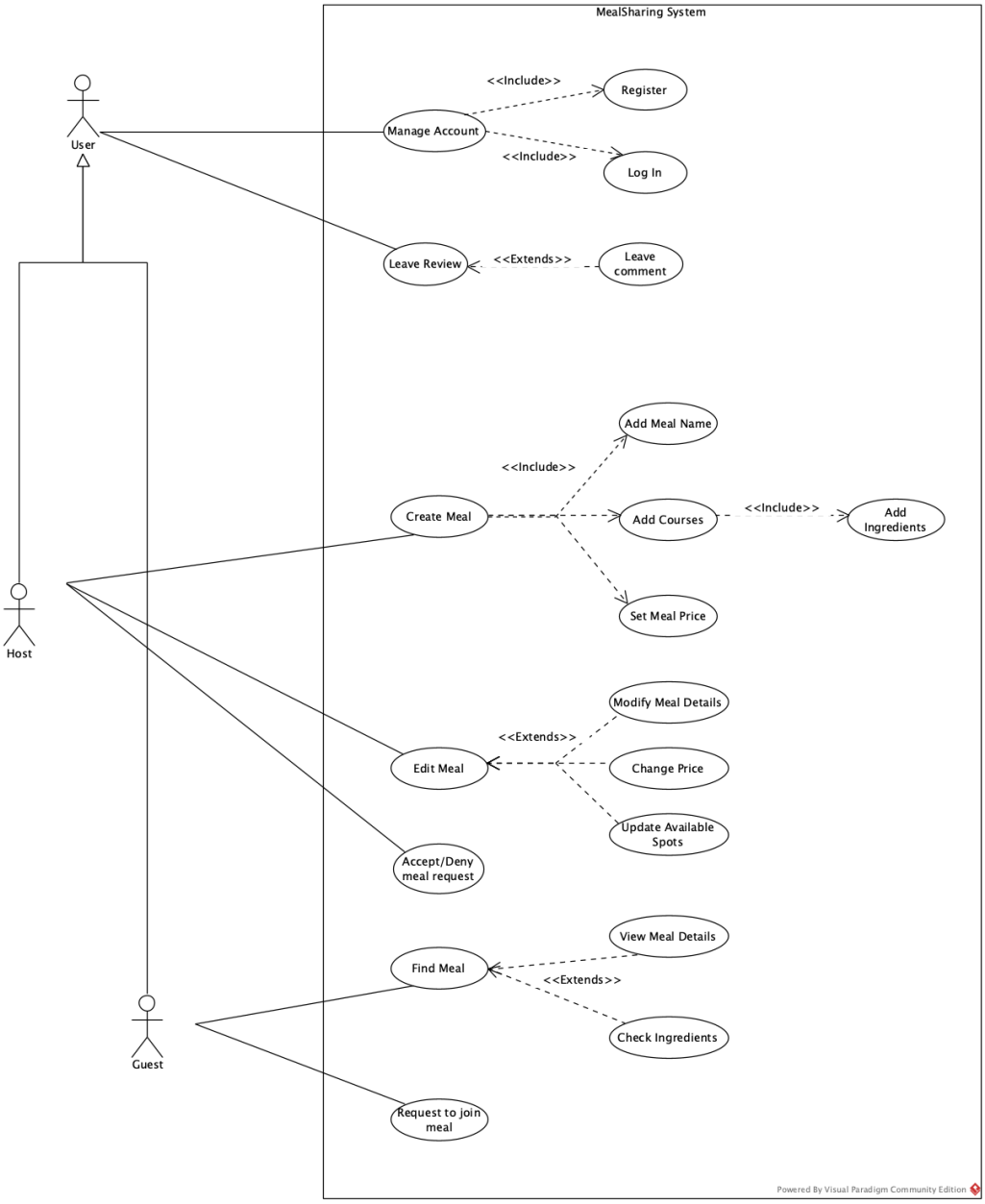
Find My Meals Host Notificatio... Account

Notifications

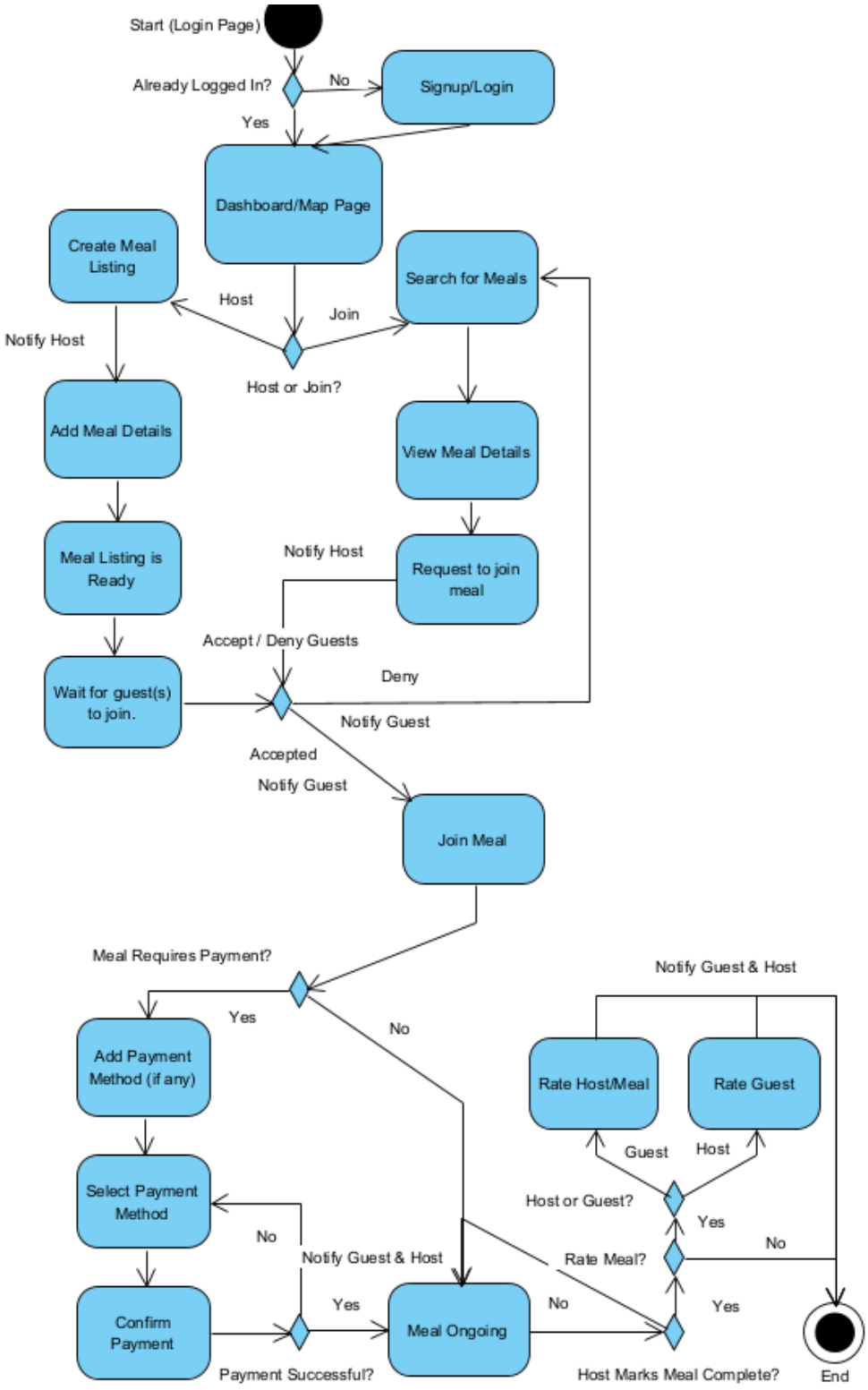
You have no notifications

Find My Meals Host Notificatio... Account

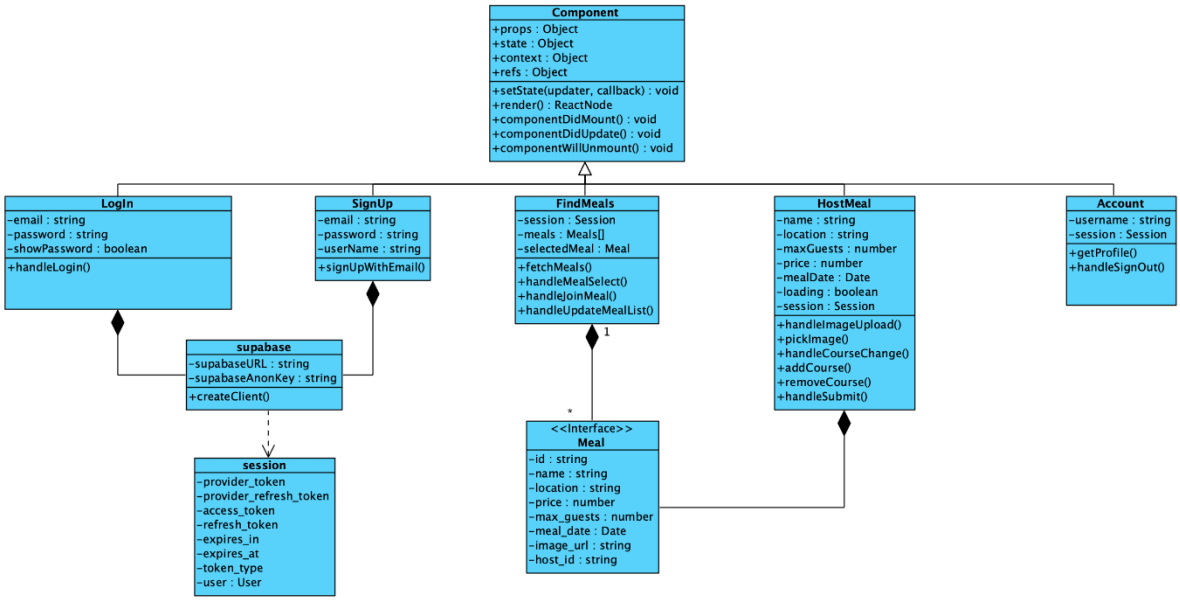
Updated Use-Case Diagram



Updated Activity Diagram



Updated Class Diagram



UPDATED WORKFLOW:

1. Start & Login Flow

The process begins at the Login Page where the system checks if the user is already logged in:

- Yes: The user is directed to the Dashboard/Map Page.
- No: The user is redirected to the Signup/Login screen to authenticate.

2. Choosing an Action: Hosting or Joining a Meal

Upon reaching the Dashboard/Map Page, the user decides whether to Host or Join a meal.

Hosting a Meal

- The host selects "Create Meal Listing."
- Adds meal details such as name, time, and location, as well as an image for the listing.
- The system confirms that the Meal Listing is Ready.
- The system waits for guests to join before proceeding.

Joining a Meal

- The guest searches for meals on the map.
- Clicks/taps on a meal to view the meal details.
- A “Wait to Join Meal” step is introduced where the user requests to join a meal.

    - The user must wait for the host to accept them as a guest before they can join.
- The system then allows the user to join the meal.

3. Handling Payments (If required)

- A decision node checks if the meal requires payment.
- If payment is required:
    - The user is prompted to Add a Payment Method (if none exists).
    - The user selects the payment method and Confirms Payment.
    - A decision checks if the Payment is Successful:
        - Yes: The user proceeds to the Meal Ongoing step.
            - Notification to Guest: "Payment confirmed for '[Meal Title]'."
        - No: The user must retry payment.
            - Notification to Guest: "Payment failed. Please try again to join '[Meal Title]'."
- If no payment is required, the system moves directly to Meal Ongoing.

4. Meal Ongoing and Completion

- Once the meal is ongoing, the system waits for the Host to Mark the Meal as Complete.
- After the host marks the meal complete:
    - A decision checks if the user is a Host or Guest.
    - The system prompts for ratings:
        - Guest: Rates the host/meal.
        - Host: Rates the guest.
            - Notification to Both: "Please rate your experience with [Username] for '[Meal Title]'."
    - If the user skips the rating, the flow proceeds to the End.

5. End of Flow
After the meal is complete and any necessary ratings are submitted (or skipped), the workflow concludes, and the user's journey ends.

Key Updates/Changes:

- Implemented meal browsing functionality, allowing guests to search meals by:
    - Location proximity
    - Date and time
    - Meal/course information (e.g., ingredients, course names)
- Built request flow for guests to request to join a meal:
    - Guests can send a join request to meal hosts
    - Hosts can view incoming requests and choose to accept or reject them
    - Guests receive updates on the status of their request (accepted or rejected)
- Integrated request logic to update the guest list once a host approves a request
- Developed a rating system allowing:
    - Hosts to rate guests after the meal
    - Guests to rate the host and meal experience
- Performed extensive bug testing and fixed across features including request flow and UI
- Updated documentation to reflect new request and rating features

Challenges and Strengths

Through this milestone, the team encountered several technical and collaborative challenges. Working with Supabase required careful attention to how data was formatted and retrieved, and early development involved a steep learning curve with React Native. Additionally, resolving merge conflicts became necessary as developers worked on overlapping components and pushed code from separate branches. Our group also ran into some issues when using our software on Android devices, which led to us being unable to test certain areas of our code.

Despite these challenges, the team demonstrated strong problem-solving and collaboration throughout the milestone:

- Resolved merge conflicts stemming from parallel development on shared files
- Gained proficiency in React Native while actively building out core features
- Successfully implemented new functionality including guest request flows and host approval handling
- Used scenario testing to validate user journeys and ensure stability
- Maintained consistent communication and pull request discipline to manage progress effectively

The most critical contributor to our success was the team's ability to communicate consistently and adapt quickly to new technologies and tools. This enabled smooth development and feature integration across the board.

Looking Forward

As we move into the final phase of development, our focus will be on ensuring all core system requirements are fully met and functioning smoothly. We will solidify the ability for users to create, post, and view meal invitations, ensuring hosts can easily manage their events. On the guest side, we aim to enhance browsing experience by implementing search capabilities that allow users to filter meals by key attributes such as location, date, and meal details.

Additionally, we will complete the integration of the payment system using the Stripe API, allowing guests to seamlessly submit payments when joining paid meals. This will include handling payment method selection, confirmation, and potential errors (e.g., failed transactions). We also plan to implement the AI in our system. We have yet to decide how to do so, however

we will decide throughout this coming final milestone. By tying all these components together, we aim to deliver a polished and fully functional meal-sharing experience that supports the complete lifecycle of a hosted meal event.