




MEAL SHARING - MILESTONE 5:

FINAL DELIVERY
AND DESIGN REFLECTION

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



Team Introduction and Member Roles

Ryan Davis

- *Focused on documentation. Implemented merge requests and bug fixes.*

Ivan Farfan

- *Manages SupaBase DB. Implemented AI categorizations functionality.*

Johan Jaramillo

- *Implemented host meal cancellations and the notification center for meal status updates.*

Cory Vitanza

- *Focused on documentation, as well as performed bug testing/fixing. Reviewed pull requests*

Functional Requirements for Milestone 5

Guest:

- Cancel a join request (if not yet accepted)
- View request history and upcoming meals
- Receive in-app updates for request outcomes

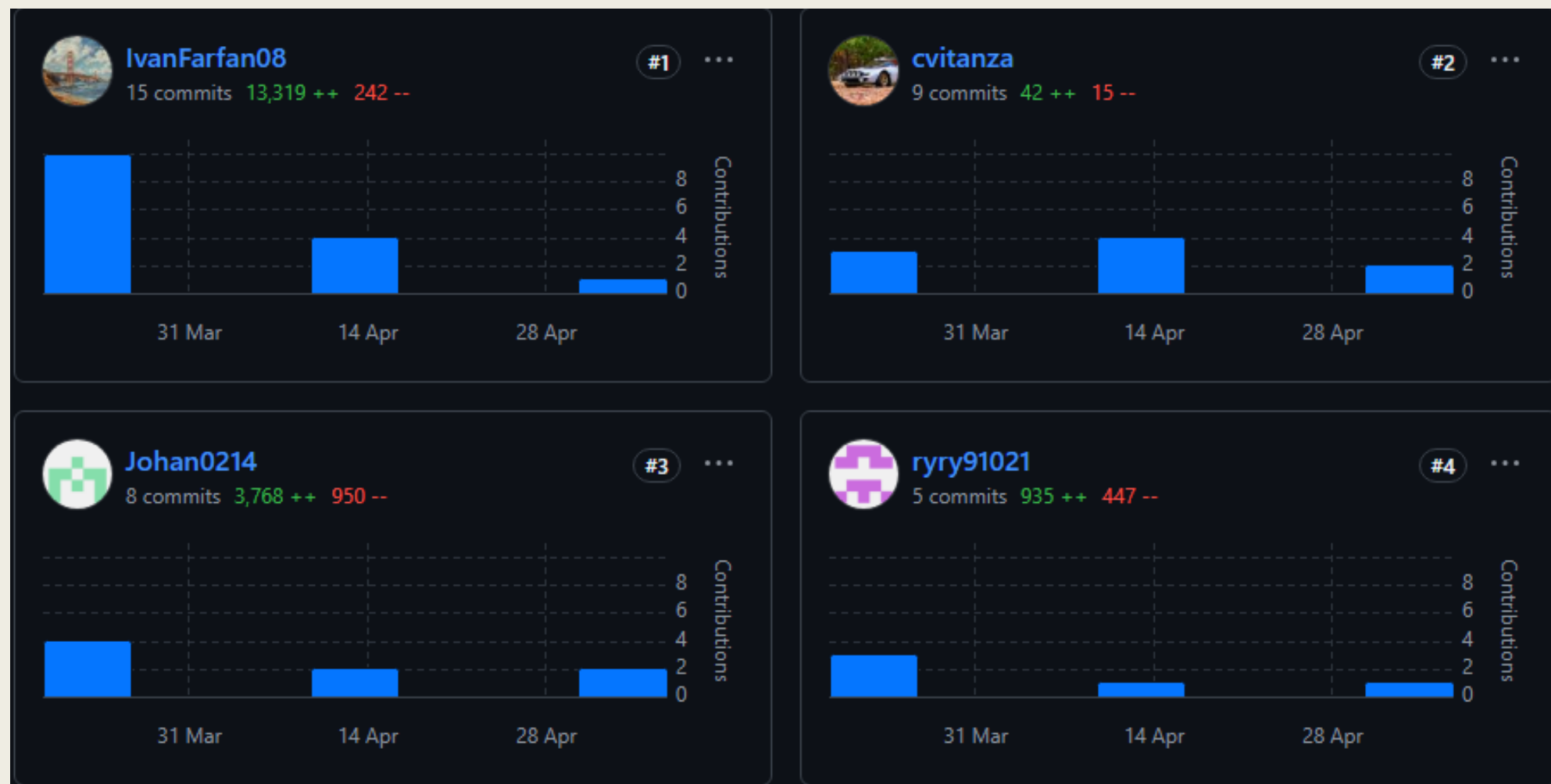
Host:

- Cancel or reschedule meals
- View and manage guests for each meal

System:

- Provide a centralized notification center for meal status updates
- Handle edge cases like full meals or re-joining canceled ones

Development Effort Overview



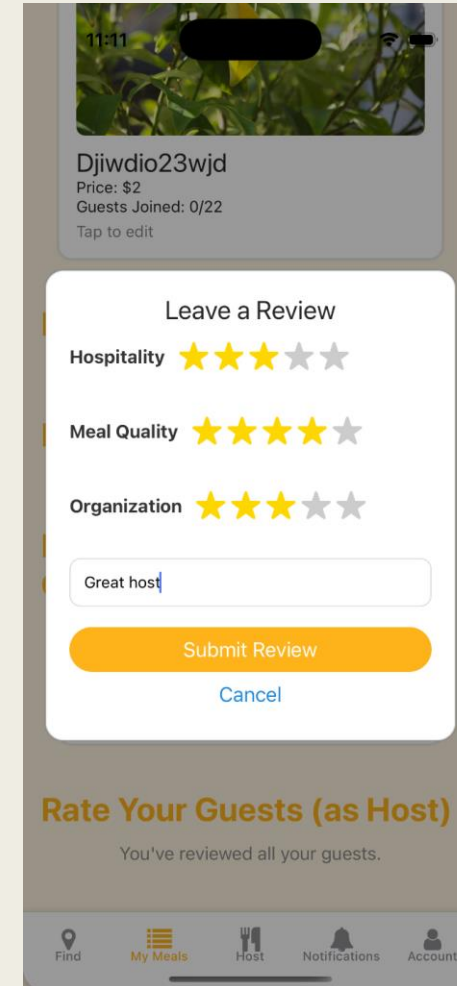
Note that these contributions / graphs do **not** include merge commits.
(Contributions per week to main, excluding merge commits)



OUR DEMO

IMPLEMENTATION HIGHLIGHTS

-Expanded rating system by integrating categories for both guest ratings and host ratings.



Challenges and Strengths

Challenges

- Resolving merge conflicts when making changes in the same file.
- Making updates in the database schema to support new request features.
- Ran into major package issues, leading to a long list of problems

Strengths

- Recent familiarity with React has improved programming skills.
- Pull requests are implemented to ensure minimal merge conflicts.
- The diversification of tasks allows developers to work in different components of the system.

Lessons Learned



Merge conflict management is critical



Node package and environment stability impacts development flow



Diversifying tasks improves production



THANK YOU!

Questions?