Sprint 1 Materials

Team 11
Sam Bush
Dong Ha Cho
Blake Dejohn
Nicholas Petersilge
Shantanu Raghavan

Retrospective Summary

Overall, this sprint was mostly successful. The two big issues that we ran into for this sprint were not adding enough items to the sprint backlog and not protecting the main branch of our project. For the issue of not adding enough items, because of the novelty of working with a new language and framework, we thought that we would need much more time to get familiar with the environment we would be using to create this project. However, all in all, it did not take the team too long to adapt to the new environment we were introduced to. To remedy this for the next sprint, we will make sure to add more items than we did for this sprint (or at the very least add more difficult tasks). Moving on, the oversight of not protecting the main branch of the project was something that the team completely overlooked. Continuing, we will make sure that any changes to the project's main branch will be done through pull requests/merges from other branches. Finally, to talk about the procedure we used for this sprint, we are very happy with the process we went through to complete this sprint. We think that we will use the same procedure for future sprints and consequently are only updating our product log by taking out the items that we completed during this sprint.

GitHub Release Link

GitHub Release link:
https://github.com/csce-315-331-2024a/project-3-full-stack-agile-web-project-3-902-11/releases/tag/sprint

MVP link: https://rev-pos-331.vercel.app/

(Note: the only feature not yet tested is the updating of the stock. We are looking to push testing of this task to the next sprint)

Product Backlog

Task Description	Priority	Dependencies	Estimated Time	Status	User Story Points
Design orders table	High	None	1-2 Days	Complete	2
Design food table	High	None	1-2 Days	Complete	2
Design ingredients table	High	None	1 Day	Complete	1
Join tables where necessary	High	Design of all 3 tables	1 Day	Complete	1
Populate the database	Medium	Database schema completed	2-5 Days	Complete	5
Design an API to interact with the database from frontend for cashier/customer	High	Database schema	2-3 Days	In progress	3
GET endpoint to fetch entire available menu	High	API design, database populated	1-2 Days	Complete	2
GET endpoint to fetch menu items by category	High	API design, database populated	1 Day	Complete	1
POST endpoint to submit a new order	High	API design, database schema	2-3 Days	Complete	3
PATCH endpoint to update inventory stock values when orders placed	High	API design, database schema, order endpoint	2 Days	In progress	2
Design an API for manager use	High	Database schema	2-3 Days	In progress	3
Allow user to manually update inventory stock values	Medium	API design for manager, database schema	1-2 Days	Not started	2
Fetch excess ingredient usage data given certain dates	Medium	Database schema, API design for manager	2-3 Days	Not started	3
Fetch ordering trend data from database	Medium	Database schema, API design for manager	2-3 Days	Not started	3
Fetch commonly ordered pairs data given certain dates	Medium	Database schema, API design for manager	2-3 Days	Not started	3
Update menu items	Medium	API design for manager, database schema, frontend	1-2 Days	Not started	2
Create new menu item with inputted data	Medium	API design for manager, database schema, frontend	1-2 Days	Complete	2

Delete menu item	Medium	API design for manager, database schema, frontend	1-2 Days	Not started	2
Fetch past orders data within certain dates	Medium	API design for manager, database schema	1-2 Days	Complete	2
Option to log into cashier and manager views	High	Authentication system	2-3 Days	Not started	3
Create button to log into cashier view	Medium	Authentication system, UI design	1 Day	Not started	1
Create button to log in to manager view	Medium	Authentication system, UI design	1 Day	Not started	1
Create button to display menu items in a separate tab	Medium	Frontend structure, menu data fetching	1-2 Days	In progress	2
Fetch menu items with sufficient inventory	Medium	API endpoint , database design	1-2 Days	Not started	2
Display menu items on the newly generated tab and do not create any interactivity	Medium	Tab creation, data fetching	1-2 Days	Not started	2
Display menu items in a static manner based on menu data fetched from the database	Medium	Data fetching	1 Day	Complete	1
Integrate translation API for accessibility	Medium	API selection, frontend structure	2-4 Days	Not started	4
Accommodate for WCAG 2.1 guidelines	High	Throughout frontend development	1-3 days per view	Not started	3
Create frontend design to allow users to choose which category to view	Medium	Menu data structure	1-2 Days	Complete	2
Fetch menu items grouped per category	Medium	API endpoint	1-2 Days	Complete	2
Update GUI based on category selection	Medium	Frontend logic, data fetching	1 Day	Complete	1
Create frontend design to allow user to add items to their cart	High	Frontend structure, cart logic	2-3 Days	Complete	3
Create frontend design to allow users to view their cart and make modifications	High	Frontend structure, cart logic	2-3 Days	In progress	3
Create frontend design to allow user to complete their order and check out their cart	High	Frontend structure, cart logic, API calls	2-3 Days	In progress	3

Call API to update the database with the new order	Medium	Frontend logic, API endpoint	1 Day	Complete	1
Call API to update ingredient stock	Medium	Frontend logic, API endpoint	1 Day	Not started	1
Display weather information from third-party API	Low	API selection	1-2 Days	Not started	2
Allow cashier to quickly add items to the current order	High	Menu item data, cart logic (frontend)	2-3 Days	Not started	3
Fetch menu items grouped per category	Medium	API endpoint for category-based fetching	1-2 Days	Not started	2
Display menu items in any given category	Medium	Frontend structure, data fetching	1-2 Days	Complete	2
Allow items to be added to an order with the option of choosing a quantity of a given item	High	Frontend structure, cart logic	2-3 Days	Complete	3
Create frontend design to allow cashier to check out the current order	High	Frontend structure, cart logic, API calls	2-3 Days	Not started	3
Call API to update database with the new order	Medium	Frontend logic, API endpoint	1 Day	In progress	1
Call API to update database's appropriate ingredient stock data	Medium	Frontend logic, API endpoint	1 Day	Not started	1
Create frontend interface for manager to modify menu items	Medium	Frontend form design, API for updating	2-4 Days	In progress	4
Create API calls from frontend to update menu item	Medium	API design for updating items	1-2 Days	Not started	2
Create frontend interface for manager to create a new menu item	Medium	Frontend form design, API for creation	2-4 Days	Complete	4
Create API calls from frontend to update database with inputted data	Medium	API design for item creation	1-2 Days	In progress	2
Create API calls from frontend to fetch inventory stock data	Medium	API design for inventory	1-2 Days	Not started	2

Use fetched data from API to update GUI to stock data	Medium	Frontend UI for stock display	1-2 Days	Not started	2
Create API calls from frontend to update inventory stock data	Medium	API design for inventory update	1-2 Days	Not started	2
Create frontend design for manager to view a visualized representation of excess ingredient usage data	Medium	Data visualization library, decision on chart types	2-3 Days	Complete	3
Create API calls from the frontend to fetch excess ingredient usage data	High	API design for excess ingredient data, backend logic to calculate	2-3 Days	Not started	3
Use fetched data from API to create a visualization of excess ingredient usage data	Medium	Data visualization library, frontend logic	2-3 Days	Not started	3
Create frontend design for manager to view a visualization of commonly ordered item pairs	Medium	Data visualization library, decision on chart types	2-3 Days	Not started	3
Create API calls from frontend to fetch commonly ordered pairs data	High	API design, backend logic to calculate pairs	2-3 Days	Not started	3
Use fetched data from API to create a visualization of commonly ordered item pairs data	Medium	Data visualization library, frontend logic	2-3 Days	Not started	3
Create frontend design for manager to view a visualization of past order data	Medium	Data visualization library, backend API for past orders	2-3 Days	Not started	3
Create API calls from frontend to fetch past order data	Medium	API design for past orders	1-2 Days	Complete	2
Use fetched data from API to create a visualization of past order data	Medium	Data visualization library, frontend logic	2-3 Days	Not started	3
Create frontend interface for manager to view all menu items	Medium	Frontend UI structure, backend API for menu items	2-3 Days	Complete	3
Create API calls from frontend to fetch menu items	Medium	API design for menu items	1-2 Days	Complete	2
Use fetched data from API to update GUI to show menu items	Medium	Frontend logic	1-2 Days	Not started	2
Create frontend interface for manager to modify menu items	Medium	Frontend form design, backend API for updating items	2-4 Days	Complete	4
Create API calls from frontend to update menu item	Medium	API design for item updates	1-2 Days	Not started	2

Create frontend interface for manager to create a new menu item	Medium	Frontend form design, backend API for item creation	2-4 Days	Complete	4
Create API calls from frontend to update database with inputted data	Medium	API design for item creation	1-2 Days	Complete	2
Create API calls from frontend to fetch inventory stock data	Medium	API design for inventory	1-2 Days	Not started	2
Use fetched data from API to update GUI to stock data	Medium	Frontend UI for stock display	1-2 Days	Not started	2
Create API calls from frontend to update inventory stock data	Medium	API design for inventory update	1-2 Days	Not started	2
Allow manager to switch to the menu page	Low	Frontend navigation	1 Day	Not started	1
Obtain OAuth2 credentials from the Google Developer Console	High	None	1-2 Days	Not started	2
Implement the backend authorization endpoint	High	OAuth2 library	2-3 Days	Not started	3
Implement token exchange and secure storage logic	High	OAuth2 library, authorization endpoint	2-3 Days	Not started	3
Implement user profile retrieval using Google's API	Medium	OAuth2 library, token exchange	1-2 Days	Not started	2
Design and implement the "Login with Google" UI element	Medium	UI design guidelines	1-2 Days	Not started	2
Implement frontend logic to initiate and handle the OAuth2 flow	Medium	OAuth2 library, "Login with Google" UI	2-3 Days	Not started	3
Implement token transmission to the backend	Medium	Frontend OAuth2 logic	1 Day	Not started	1

Sprint Backlog

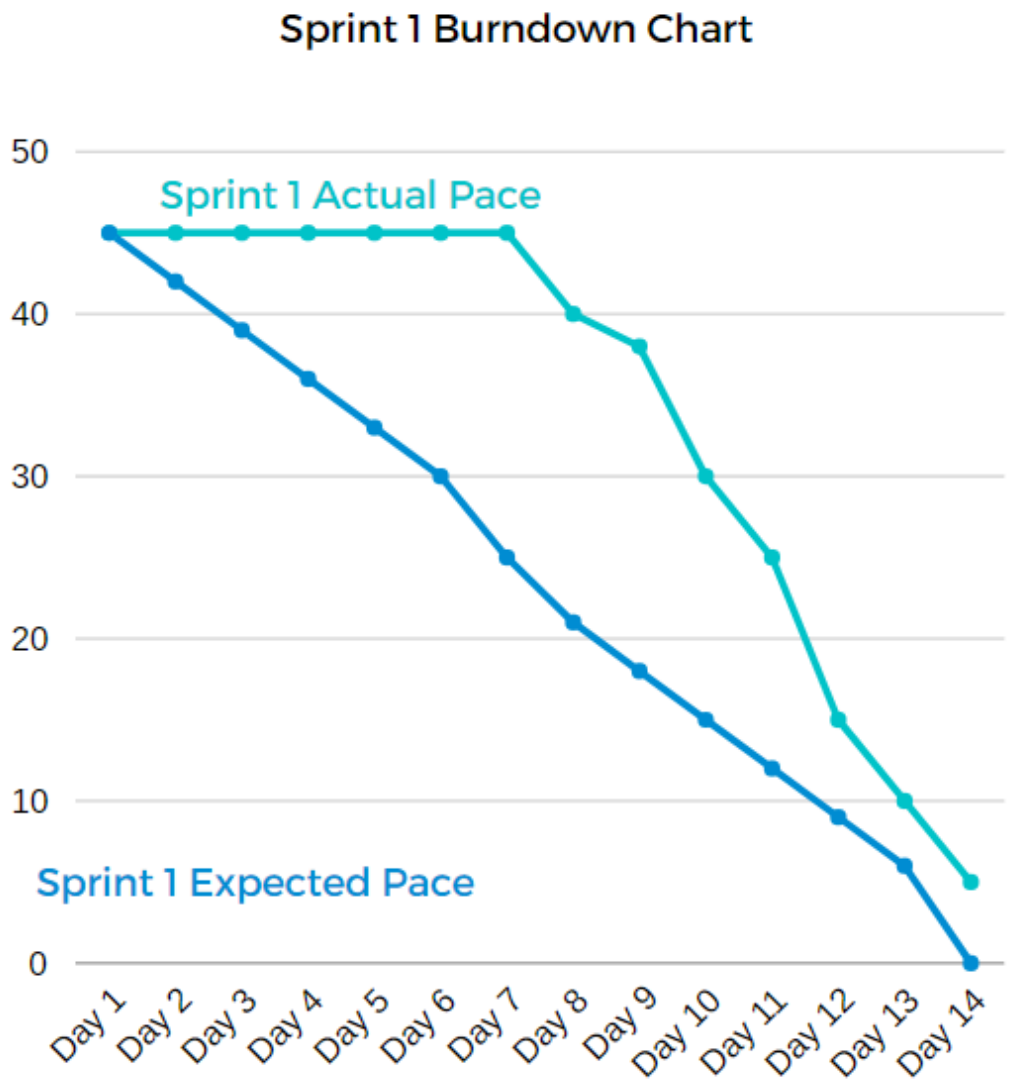
Task	User Point Value	Assigned To	Actual Time Spent (days)	Status
Design frontend interfaces for category selection and dynamic menu display.	3	Dong Ha Cho	2	Completed
Design frontend interfaces for cart viewing, adding/removing items, and modification.	4	Dong Ha Cho	3	Completed
Implement and connect frontend designs to fetch and display menu items from the backend.	3	Nicholas Petersilge	2	Completed
Design and create tables for: Orders, Food Items, and Ingredients	5	Sam Bush	4	Completed
Design relationships between tables, ensuring data integrity and efficient querying.	1	Sam Bush	1	Completed
Populate the database with initial data (as needed).	5	Blake Dejohn	4	Completed
Develop the GET endpoint to retrieve the entire menu.	2	Shantanu Raghavan	2	Completed
Develop the GET endpoint to fetch menu items filtered by category	1	Nicholas Petersilge	1	Completed

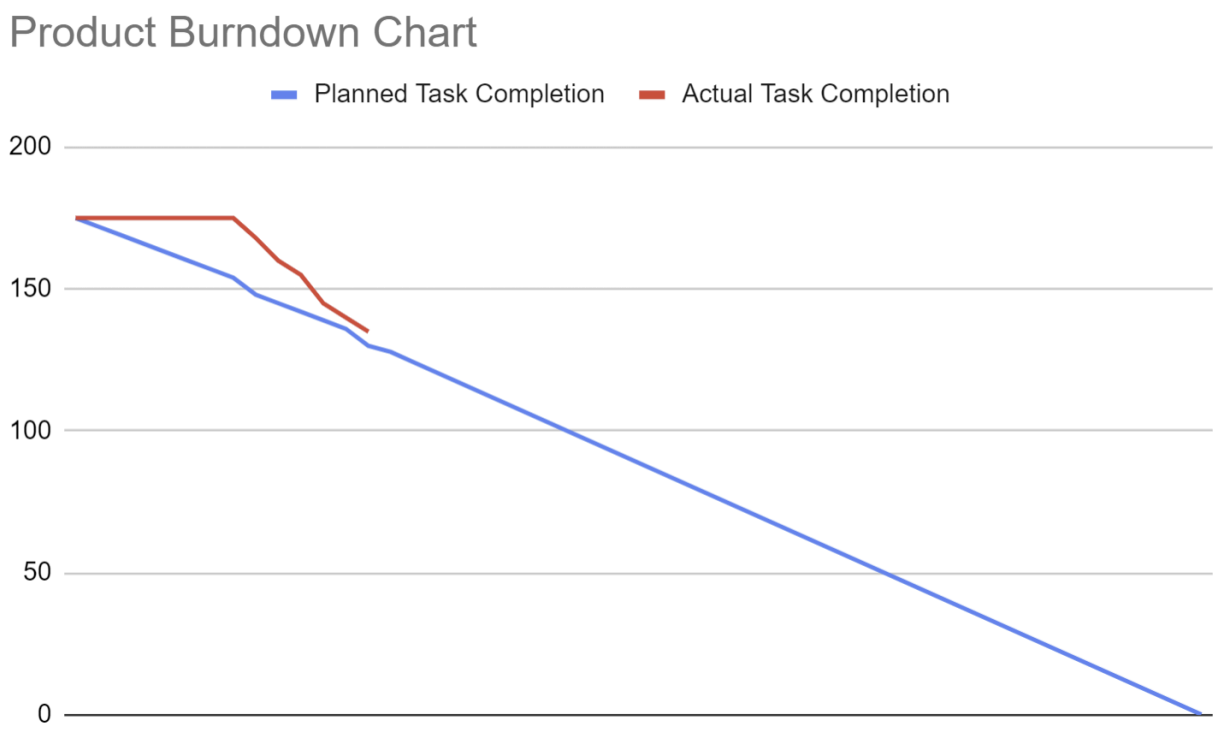
Develop the POST endpoint for submitting new orders.	3	Nicholas Petersilge	3	Completed
Develop the PATCH endpoint to update inventory stock after order placement.	2	Nicholas Petersilge	2	In Progress
Establish backend connectivity to the database for data retrieval and updates.	2	Shantanu Raghavan	2	Completed
View past orders API endpoint	2	Shantanu Raghavan	2	Completed
View Static Menu	3	Blake Dejohn	2	In progress
Add new menu item API endpoint	2	Shantanu Raghavan	2	Completed
Adjust Inventory stock	3	Sam Bush	2	In progress

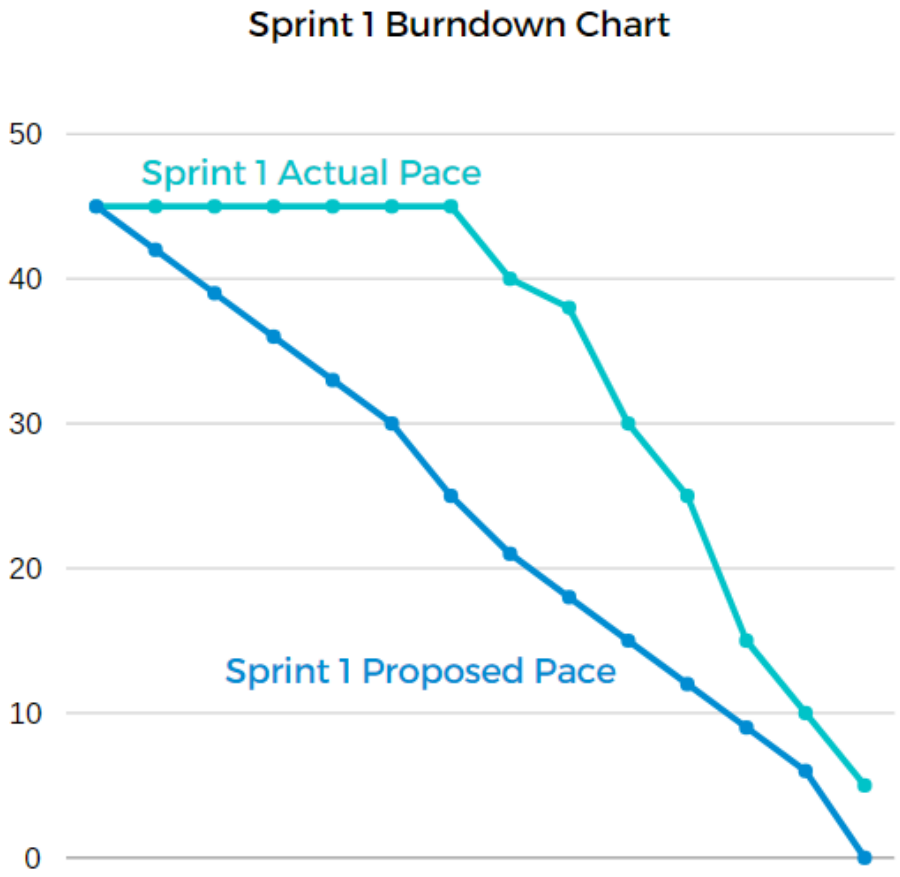
Summary Table

Team Member	**Completed User Story Points**	**Sprint #**
Shantanu Raghavan	8	Sprint 1
Sam Bush	6	Sprint 1
Nicholas Petersilge	7	Sprint 1
Blake Dejohn	7	Sprint 1
Dong Ha Cho	7	Sprint 1

Burn-down Charts




Product Burndown Chart
Planned Task Completion
Actual Task Completion
200
150
100
50
0



Next Sprint's SCRUM Meeting Schedule

For the next sprint, we are looking to meet at least every two days (starting on Monday). The general theme of the next spring will be adding missing features. Another part of this sprint will be cleaning up the application so that it is more intuitive to use and looks better (meaning we will be working on the first objective first and then if we have time, we will beautify the application).

To show this, this is a very general outline of the tasks we are looking to complete for the sprint 2:

- Updating the stock based on the orders being made (includes error handling such as non-negative stock checks and making sure that the item being ordered can be made with the current stock)
- Adding support for making orders (orders will be updated in the database along with the stock it took to create the items within the order)
- Adding the cashier view (very similar to the self-checkout menu, however, it will be a more streamlined and faster version)

- Adding better navigation (global nav bar?)
- Overall cleaning up of the application (using Tailwind, bootstrap, etc.)

Appendix 1: SCRUM Meeting Agendas and Minutes

SCRUM Meeting 1 for Rev's POS:
Prepared by: Shantanu Raghavan
Meeting Date: 03/25/2024

Meeting Attendees

1. Sam Bush
2. Dong Ha Cho
3. Blake Dejohn
4. Nicholas Petersilge
5. Shantanu Raghavan

Meeting Agenda Items

- Roadblocks/Progress
- Feasibility of Tasks

Status Update Since Last Meeting

Accomplishments:

- Set up frameworks
- Added basic frontend functionality

Tasks Completed:

Task Description	**Assigned to**	**Completed? (yes/no)**
Design frontend interfaces for category selection and dynamic menu display.	Dong Ha Cho	no
Design frontend interfaces for cart viewing, adding/removing items, and modification.	Dong Ha Cho	no
Implement and connect frontend designs to fetch and display menu items from the backend.	Nicholas Petersilge	no
Design and create tables for: Orders, Food Items, and Ingredients	Sam Bush	yes

Design relationships between tables, ensuring data integrity and efficient querying.	Sam Bush	yes
Populate the database with initial data (as needed).	Blake Dejohn	yes
Develop the GET endpoint to retrieve the entire menu.	Shantanu Raghavan	yes
Develop the GET endpoint to fetch menu items filtered by category	Nicholas Petersilge	no
Develop the POST endpoint for submitting new orders.	Nicholas Petersilge	no
Develop the PATCH endpoint to update inventory stock after order placement.	Nicholas Petersilge	no
Establish backend connectivity to the database for data retrieval and updates.	Shantanu Raghavan	yes

Before The Next Meeting

Plans:

- Flesh out backend endpoints
- Create manager page, will complete sprint 1 requirements early

Task Assignments:

Task Description	**Assigned to**
View past orders API endpoint	Shantanu Raghavan
View Static Menu	Blake Dejohn
Add new menu item API endpoint	Shantanu Raghavan
Adjust Inventory Stock	Sam Bush

Minutes from Previous Meeting

Since this was our first meeting since the initial meeting, we were touching base on how things were going for each of us. We were speaking on how Next.js requires a bit of learning for each of us, but it seems like it will be a good choice in the long run. Each of us is having unique difficulties

and challenges in our functions. It seems like we didn't include enough work to do in this sprint, so we may increase the workload for the next sprint.

SCRUM Meeting 2 for Rev's POS
Prepared by: Shantanu Raghavan
Meeting Date: 03/27/2024

Meeting Attendees

6. Sam Bush
7. Dong Ha Cho
8. Blake Dejohn
9. Nicholas Petersilge
10. Shantanu Raghavan

Meeting Agenda Items

- Task Feasibility
- Difficulties and Roadblocks

Status Update Since Last Meeting

Accomplishments:

- Frontend works with backend on customer page
- Can view menu items in manager view

Tasks Completed:

Task Description	**Assigned to**	**Completed? (yes/no)**
Design frontend interfaces for category selection and dynamic menu display.	Dong Ha Cho	yes
Design frontend interfaces for cart viewing, adding/removing items, and modification.	Dong Ha Cho	yes
Implement and connect frontend designs to fetch and display menu items from the backend.	Nicholas Petersilge	yes
Design and create tables for: Orders, Food Items, and Ingredients	Sam Bush	yes
Design relationships between tables, ensuring data integrity and efficient querying.	Sam Bush	yes

Populate the database with initial data (as needed).	Blake Dejohn	yes
Develop the GET endpoint to retrieve the entire menu.	Shantanu Raghavan	yes
Develop the GET endpoint to fetch menu items filtered by category	Nicholas Petersilge	yes
Develop the POST endpoint for submitting new orders.	Nicholas Petersilge	yes
Develop the PATCH endpoint to update inventory stock after order placement.	Nicholas Petersilge	yes
Establish backend connectivity to the database for data retrieval and updates.	Shantanu Raghavan	yes
View past orders API endpoint	Shantanu Raghavan	yes
View Static Menu	Blake Dejohn	no
Add new menu item API endpoint	Shantanu Raghavan	yes
Adjust Inventory stock	Sam Bush	no

Before The Next Meeting

Plans:

- Start thinking about next sprint plans
- Try and use similar code practices across branches/features

Task Assignments:

Task Description	**Assigned to**
View Static Menu	Blake Dejohn
Adjust Inventory Stock	Sam Bush

Minutes from Previous Meeting

We were discussing certain issues we were running into with type declarations in typescript, and how sometimes we may not use typescript files when unnecessary for the ease of use. Additionally, we discussed how we plan on splitting up the api endpoints to ensure we do not have redundant API endpoints and the endpoints make sense given their purpose. We discussed how each of us completed our tasks and some insights we gained from completing our tasks to hopefully help each other in the future.

SCRUM Meeting 3 for Rev's POS:
Prepared by: Shantanu Raghavan
Meeting Date: 03/31/2024

Meeting Attendees

11. Sam Bush
12. Dong Ha Cho
13. Blake Dejohn
14. Nicholas Petersilge
15. Shantanu Raghavan

Meeting Agenda Items

- Status Quo for responsibility
- MVP requirements

Status Update Since Last Meeting

Accomplishments:

- Merged branches together
- Completed MVP

Tasks Completed:

Task Description	**Assigned to**	**Completed? (yes/no)**
Design frontend interfaces for category selection and dynamic menu display.	Dong Ha Cho	yes
Design frontend interfaces for cart viewing, adding/removing items, and modification.	Dong Ha Cho	yes
Implement and connect frontend designs to fetch and display menu items from the backend.	Nicholas Petersilge	yes
Design and create tables for: Orders, Food Items, and Ingredients	Sam Bush	yes
Design relationships between tables, ensuring data integrity and efficient querying.	Sam Bush	yes

Populate the database with initial data (as needed).	Blake Dejohn	yes
Develop the GET endpoint to retrieve the entire menu.	Shantanu Raghavan	yes
Develop the GET endpoint to fetch menu items filtered by category	Nicholas Petersilge	yes
Develop the POST endpoint for submitting new orders.	Nicholas Petersilge	yes
Develop the PATCH endpoint to update inventory stock after order placement.	Nicholas Petersilge	yes
Establish backend connectivity to the database for data retrieval and updates.	Shantanu Raghavan	yes
View past orders API endpoint	Shantanu Raghavan	yes
View Static Menu	Blake Dejohn	yes
Add new menu item API endpoint	Shantanu Raghavan	yes
Adjust Inventory stock	Sam Bush	no

Before The Next Meeting

Plans:

- Sprint 2 planning finalization
- Inventory stock

Task Assignments:

Task Description	**Assigned to**
Adjust Inventory stock	Sam Bush

Minutes from Previous Meeting

Discussed how everything was coming together, and how to resolve merge conflicts. We actually did a pull request in the meeting to ensure everyone was on the same page as to policies for doing

merge requests and what were the standards we were setting as to functionality and design. We discussed how Sam was running into a slight complication for the inventory stock functionality, but we will push that task to the next sprint. We will still be on pace to complete the project, and Sam has already worked out some functionality, so the task is not completely untouched.