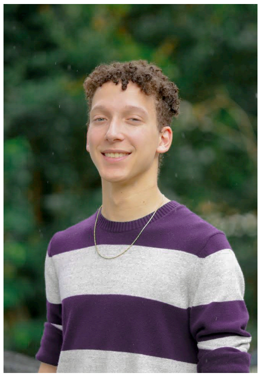
Indy 4 - MyKitchen

CS 4850 - Section 02 - Spring 2024

Jan 30, 2024

Merrick 	Nick 	Minchul

Name	**Roles**	**Major responsibilities**	**Cell Phone / Alt Email**
Merrick McPherson	Team Leader	Handles meetings with instructor and other members. Will serve as manager and documenter.	404-956-0341 merrickscottmcpherson@gmail.com
Nick Nettleton	Developer	Handles backend and software architect.	770-595-2486 nick@zoty.us
Minchul Ha	Developer	Handles frontend and testing.	404-917-7484 billyha0703@gmail.com
Sharon Perry	Advisor / Instructor	Facilitate project progress; advise on project planning and management.	770-329-3895

Project Overview

The MyKitchen project proposes a platform that transforms home kitchens into ghost kitchens, enabling individuals to sell their home-cooked meals to local customers. This platform will be a web app that has two types of actors, customers and kitchen owners, where by the kitchen owners will be able to make their own page with filled with pictures of the dish, price, and description and the customers will have access to different kitchens in their location to purchase from. Customers would make payments through online or choose to pay in cash on pickup.

The MyKitchen project will be utilizing the javascript framework React to create the frontend, and will use the backend language PHP for calls to APIs. Since the project will require a place to store data on prices, kitchens, and pictures the project will house our database in MySQL. Lastly for hosting, the project will go with AWS.

Project website

https://studentweb.kennesaw.edu/~mmcphe13

Deliverables:

1. Project Plan
2. Requirements
3. Design of Backend
4. Design of Frontend
5. Development of Backend
6. Development of Frontend
7. Prototype
8. Testing
9. Final Report
10. Final Project

Milestone Events

1. Project Plan

2. Requirements- An agreed upon set of requirements is drawn up and has been read by all members.

3. Software Design- Design documents have been created to to turn the abstract requirements into concrete design. Establishing the architectural design, technical specifications, data structures, interface representations, and any security that comes with it.

4. Development of Software- Set up of technologies, frameworks, databases, and cloud servers (if any) will be the first start of development leading into coding the platform.

5. Prototype and Testing- At the end of the development period a finished prototype will be tested allowing for any adjustments to be made.

6. Final Report- A document of the SDLC of the product is completed.

Meeting Schedule Date/Time

Meetings will be held through Teams or in-person. They will take place every Friday at 3:00pm to 4:00pm.

Collaboration and Communication Plan

Communication Method — Teams, Cellphones (call/text)

Collaboration — Teams Call

Version Control — Github

For status updates, it is expected that when a developer has made progress that it will be reported through text or teams. This will be accompanied along with the proper procedures that of version control that the developer would follow. Having a text to announce the developers progress will allow the documenter to keep track of development.

Version Control Plan

Overview: For the Kitchenator project, effective version control is crucial to manage code changes, track progress, and facilitate collaboration among team members. We plan to utilize GitHub, a widely recognized platform for software development version control, along with integration into our chosen cloud provider.

1. GitHub Organizational Account:

- **Account Creation:** We will establish an organizational account on GitHub specifically for this project. This ensures that all project-related code and documentation are stored under a collective entity rather than individual accounts.
- **Repository Setup:** Within this account, we will create separate repositories for different components of the project (e.g., backend, frontend, documentation).
- **Access Control:** Team members will be granted appropriate access levels to contribute to the repositories. We will manage permissions to ensure security and proper collaboration flow.

2. Branching Strategy:

- **Main Branch:** The 'main' branch will be the primary branch where the final source code resides.
- **Development Branches:** Feature-specific branches will be created from the 'main' branch for new features and bug fixes. After completion and testing, these branches will be merged back into the 'main' branch.

- **Release Branches:** For every major release, a release branch will be created from the 'main' branch.

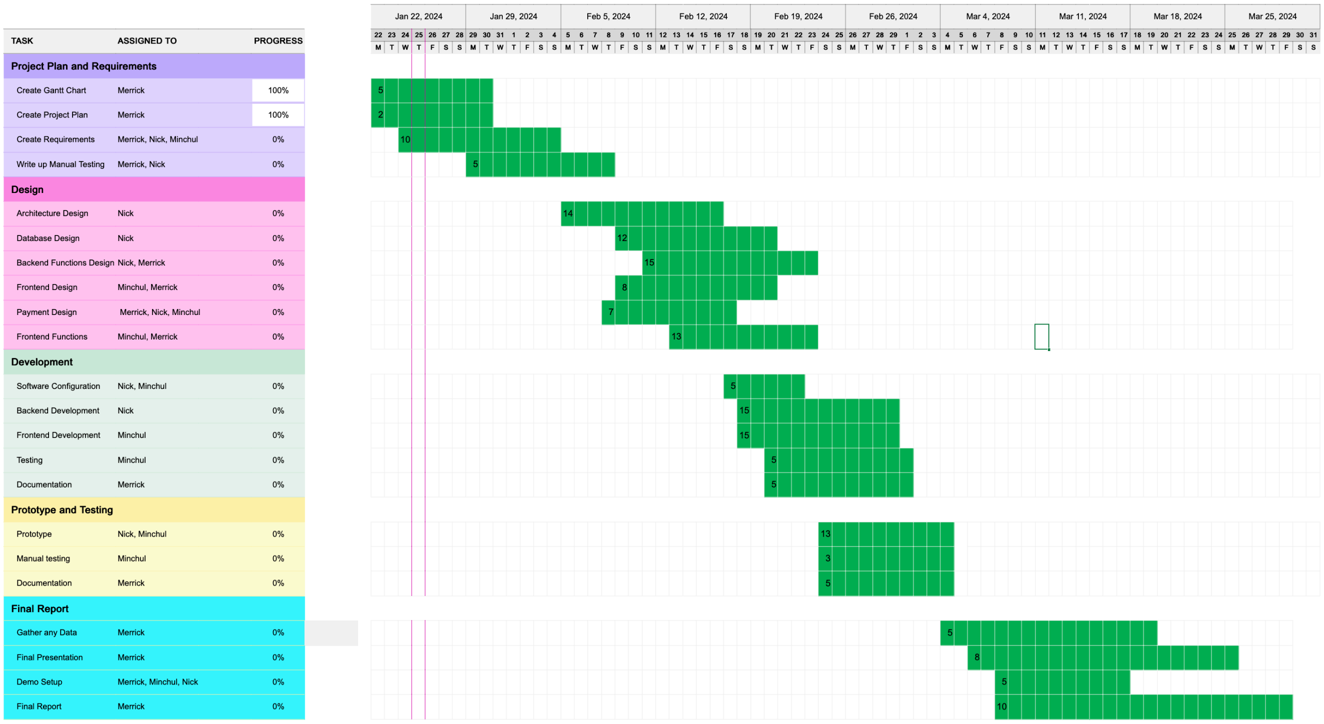
3. **Commit Guidelines:**

- **Descriptive Messages:** Every commit should have a clear and descriptive message.
- **Atomic Commits:** Commits should be as small as possible while maintaining the functionality to make tracking and reverting changes easier.

4. **Code Reviews and Pull Requests:**

- **Pull Requests:** All changes must be submitted as pull requests (PRs) and reviewed by at least one other team member before merging.
- **Review Process:** The review process will ensure code quality, adherence to project standards, and detection of potential issues.

Project Schedule and Task Planning



TASK	ASSIGNED TO	PROGRESS
Project Plan and Requirements		
Create Gantt Chart	Merrick	100%
Create Project Plan	Merrick	100%
Create Requirements	Merrick, Nick, Minchul	0%
Write up Manual Testing	Merrick, Nick	0%
Design		
Architecture Design	Nick	0%
Database Design	Nick	0%
Backend Functions Design	Nick, Merrick	0%
Frontend Design	Minchul, Merrick	0%
Payment Design	Merrick, Nick, Minchul	0%
Frontend Functions	Minchul, Merrick	0%
Development		
Software Configuration	Nick, Minchul	0%
Backend Development	Nick	0%
Frontend Development	Minchul	0%
Testing	Minchul	0%
Documentation	Merrick	0%
Prototype and Testing		
Prototype	Nick, Minchul	0%
Manual testing	Minchul	0%
Documentation	Merrick	0%
Final Report		
Gather any Data	Merrick	0%
Final Presentation	Merrick	0%
Demo Setup	Merrick, Minchul, Nick	0%
Final Report	Merrick	0%