**Project Charter**

**Summary:**

FoodieFinds is a restaurant discovery and dining recommendation application designed to assist users in locating nearby eateries, browsing reviews, and making reservations seamlessly.

**Motivation:**

FoodieFinds aims to enhance the client's workflow by providing a centralized platform for restaurant discovery and reservation management. By streamlining the process of finding and booking dining options, FoodieFinds will improve user satisfaction and operational efficiency.

**Required Functionality:**

For a detailed overview of the required functionality, please refer to the Functional Requirements Document.

It sounds like you're looking to develop a comprehensive task management application, and you're interested in integrating similar features into your food delivery platform, FoodieFinds. Let's break down how these features could apply to your platform:

1. **Task Management**: This could be utilized for managing orders, deliveries, and restaurant operations. Each task could be assigned to specific individuals or teams.
2. **Deadline Management**: Set deadlines for order preparation, delivery, or reservation confirmations to ensure smooth operations.
3. **Reliability**: Ensure that the platform is always accessible and functions properly to avoid disruptions in service.
4. **Scalability**: As more restaurants and users join the platform, it should be able to handle increased traffic without slowing down.
5. **Easy to Use**: A simple and intuitive interface for users, restaurants, and delivery personnel to manage their tasks and orders efficiently.
6. **Calendar View**: A calendar view could be useful for restaurants to manage their availability for reservations and for users to plan their orders ahead of time.
7. **Backend Stuff**: Secure user accounts, encrypted data storage, and reliable server infrastructure are essential for protecting user information and ensuring platform stability.
8. **Real-Time Updates**: Instant notifications for new orders, order status changes, and delivery updates provide a seamless experience for users and restaurants.
9. **Security**: Strong login mechanisms, data encryption, and compliance with data protection regulations are crucial for protecting user privacy and data security.
10. **Database**: A database that can handle the platform's requirements efficiently, whether it's a standard SQL database or a more flexible NoSQL solution, based on the application's needs.

**Roles and Responsibilities:**

| **Name and Signature** | **Role** | **Position** | **Contact Information** |
| --- | --- | --- | --- |
| Malaya Sugandhini | Developer | Project Manager | [S566637@nwmissouri.edu](mailto:S566637@nwmissouri.edu) |
| Spandana Pandi | Quality Assurance | Software Engineer | [S565959@nwmissouri.edu](mailto:S565959@nwmissouri.edu) |
| Ramakotireddy Ragipindi | Manager | Software Engineer | [S566606@nwmissouri.edu](mailto:S566606@nwmissouri.edu) |
| Akhila Mylavarapu | Developer | Software Engineer | [S566600@nwmissouri.edu](mailto:S566600@nwmissouri.edu) |
| Jahnavi chava | Quality Assurance | Team Member | [S566948@nwmissouri.edu](mailto:S566948@nwmissouri.edu) |