Object-Oriented Programming III

RECIPE AND MEAL PLANNING PROJECT PLAN

Group 1

ShareABite - Project Plan Document:

Introduction:

The Recipe and Meal Planning Application - **ShareABite** - is designed with a user-centric approach. It aims to create a comprehensive tool for users to discover recipes, plan meals, and track nutritional intake.

The application will empower users to explore new dishes while filtering through various dietary preferences and restrictions, provide a seamless meal-planning experience (with additional details regarding grocery items for their desired meal), and build a community where users can share their meals and recipes for others also to enjoy.

Project Objectives:

- Develop a user-friendly desktop browser application for meal planning and recipe discovery.
- Implement sorting and filtering features to manage dietary preferences
- Implement a form of informatics that allows the user(s) to view nutritional information about selected meals.
- Foster community interaction through recipe sharing and social features.
- Integrate itemized grocery lists to help prepare users when selecting a meal to prepare

Scope:

- Design and develop a desktop browser application.
- Include secure user authentication and profile management.
- Create a comprehensive recipe database with advanced search and filter options.
- Implement an intuitive meal planner with automatic grocery list generation.
- Provide nutritional tracking and community interaction features.

Assumptions:

- Users will access the application via desktop browsers (chrome, safari, etc.)
- Nutritional data for recipes, grocery lists, and other client data will be sourced from a MongoDB database created through verified user data entry or from pre-seeded data.

Technology Stack

The following is an overview of the technology our team plans to use to build this application. The technology used is being considered from the client's UI as the front-end extending to the back-end for database technologies (MongoDB) and finally accounting for the entire project's progression using version control from GitHub.

Front-end:

These are the selected/discussed technologies that will be used to build the solution for this application.

- HTML/CSS/JavaScript
- React.js for dynamic user interfaces

Back-end:

These are the selected/discussed technologies that will be used to build the solution for this application.

- Node.js with Express for server-side logic
- MongoDB for a flexible NoSQL database

Other Tools:

These are additional tools that can/will be used during this application's creation to help monitor tracking, keep milestones contained within reasonable timelines, and further support the success of this application.

- Git for version control
- Trello for project management
- Jest for unit testing

Timeline

A brief overview of the schedule highlighting dates as stated in the project requirements documentation.

Milestones and Deadlines:

These are the pre-defined milestones and deadlines, as highlighted in the primary project documentation, that also align with the course schedule for an ideal timeline for the project deliverables.

- Week 2: Project selection and team formation
- Week 4: Project plan submission
- Week 6: Requirements document submission
- Week 9: Design and data specification report
- Week 12: Status report and mid-term review
- Week 14: Final presentation and project submission

Team Roles

Here is a detailed breakdown of the team's decided roles and various duty breakdowns for how the team plans to contribute to the success of this application through its development and deployment.

- **Project Lead**: Kuldeep Mohanta
  - Organize plans and actions
  - Coordinate team efforts

- **Frontend Developer**: Oluwakibati Omotosho-Oboro
  - Design and implement user interface
  - Ensure cross-browser compatibility
  - Develop responsive layouts optimized for desktop browsers
  - Implement interactive features using JS frameworks (will adjust usage based on needs)

- **Backend Developer**: Bharat Gahlot
  - Develop server-side logic and database interactions
  - Design and implement RESTful APIs
  - Ensure security and data protection measures
  - Optimize server performance and scalability
  - Collaborate with frontend developers to ensure seamless data flow

- **QA/Test Engineer**: James Swaine, Princess Emmanuel
    - Create and execute comprehensive test plans and test cases
    - Perform functional, integration, and user acceptance testing
    - Identify, document, and track bugs and issues
    - Conduct regression testing to ensure stability after updates

- **Documentation Specialist**: Kuldeep Mohanta, Princess Emmanuel
    - Create and maintain formal documentation
    - Compile reports, requirements documents, and user manuals
    - Ensure documentation accuracy and consistency

Development Methodology

- **Agile Methodology**:
    - Iterative development with bi-weekly sprints
    - Regular reviews and peer-to-peer check-ins with the team to address concerns, highlight milestones, etc.
    - Continuous feedback and improvements based on the application's client requirements and consistent bug fixes, application errors, etc.

Requirements Document

Functional Requirements:

- Detailed list of features and functionalities including:
    - User registration and authentication
    - Profile management
    - Recipe upload and management
    - Advanced recipe search and filtering
    - Meal planning and grocery list generation
    - Nutritional tracking
    - Community features for recipe sharing and social interaction

Non-Functional Requirements:

- **Performance**: Ensure the application loads within 5 seconds.
- **Security**: Implement HTTPS, data encryption, and secure user authentication.
- **Usability**: Design an intuitive and easy-to-use interface.
- **Scalability**: Ensure the system can handle increasing numbers of users and data.

Risk Management

Potential Risks:

- **Technical Challenges**: Difficulty integrating various technologies and ensuring smooth functionality.
- **Security Vulnerabilities**: Potential breaches in user data security.
- **Timeline Delays**: Falling behind schedule due to unforeseen issues.
- **Requirement Changes**: Modifications in project requirements or scope.

Mitigation Strategies:

- **Technical Challenges**:
    - Conduct thorough research and prototyping before full-scale implementation.
    - Utilize experienced developers and conduct regular code reviews with other members on the team.

Security Vulnerabilities:

    - Implement robust security measures such as HTTPS, data encryption, and regular security audits.
    - Educate team members on best security practices and coding practices.

- **Timeline Delays**:
    - Establish a realistic project timeline with buffer periods for unexpected delays.

    - Hold regular progress reviews and adjust the timeline as necessary.

- **Requirement Changes**:
    - Maintain open communication with stakeholders/team members to manage expectations and changes.
    - Use Agile methodology to accommodate and adapt to changes efficiently.

- **User Adoption**:
    - Conduct user research and testing to ensure the application meets user needs.
    - Implement marketing strategies to attract users and gather feedback for continuous improvement.