**CS3354 Software Engineering Final Project Deliverable 1**

Recipe Generator

Group #4

Rhea Aemireddy, Shannon Carter, Jay Chae, Roslyn Collings, Adair Gonzalez, Hafa Kazi, Veom Nemade, Ayush Velhal

**1.  Project Task Delegation for 1st and 2nd deliverables ( Updated version of Proposal)**

**Total Tasks**

**Deliverable # 1** [All Group tasks]

* **UX/UI Designing and Prototyping:** Develop wireframes on Figma and create a working prototype of the recipe generator.
  + Pages to develop:
    - Landing Page (with navbar: Saved Recipes, Start a Recipe)
      * Assigned: [Shannon Carter](mailto:shannon.m.carter.18@gmail.com) [rhea.a.reddy@gmail.com](mailto:rhea.a.reddy@gmail.com)
    - Recipe Generator Flow:
      * Enter Ingredients (with submission button)
      * Resulting recipe
      * Assigned: [ayush.velhal@gmail.com](mailto:ayush.velhal@gmail.com) [Adair Gonzalez](mailto:axg230107@utdallas.edu)
    - Saved Recipes:
      * Categories:
        + Appetizers
        + Breakfast
        + Lunch
        + Dinner
        + Dessert
        + Drinks
        + Snacks
      * Assigned: [cotlqlc21@gmail.com](mailto:cotlqlc21@gmail.com) [Roslyn Collings](mailto:rccollings27@gmail.com) [hafakazi@gmail.com](mailto:hafakazi@gmail.com)
* **Environment Setup:** Getting the appropriate technologies installed on everyone’s machine. Connecting this project to a repository in GitHub.
  + Technologies Needed:
    - VSCode
    - Git/Github
    - Node.js
    - React.js
    - Flask
  + Packages we may use for Front-End:
    - TailwindCSS
    - Framer Motion
  + Packages we may use for Back-End:
    - FastAPI or Flask
    - Axios
  + Assigned: [rhea.a.reddy@gmail.com](mailto:rhea.a.reddy@gmail.com) [Shannon Carter](mailto:shannon.m.carter.18@gmail.com) [hafakazi@gmail.com](mailto:hafakazi@gmail.com) [cotlqlc21@gmail.com](mailto:cotlqlc21@gmail.com) [ayush.velhal@gmail.com](mailto:ayush.velhal@gmail.com) [veom2004@gmail.com](mailto:veom2004@gmail.com) [axg230107@utdallas.edu](mailto:axg230107@utdallas.edu) [Roslyn Collings](mailto:rccollings27@gmail.com)

**Deliverable # 2** [All Group tasks]

* **Implementing Front-End Pages**
  + Landing Page
    - Assigned: [Shannon Carter](mailto:shannon.m.carter.18@gmail.com) [rhea.a.reddy@gmail.com](mailto:rhea.a.reddy@gmail.com)
  + Recipe Generator Flow:
    - Assigned: [Adair Gonzalez](mailto:axg230107@utdallas.edu) [ayush.velhal@gmail.com](mailto:ayush.velhal@gmail.com)
  + Saved Recipes:
    - Assigned:  [cotlqlc21@gmail.com](mailto:cotlqlc21@gmail.com) [Roslyn Collings](mailto:rccollings27@gmail.com) [hafakazi@gmail.com](mailto:hafakazi@gmail.com)
* **Implementing Backend Schema**
  + Assigned: [veom2004@gmail.com](mailto:veom2004@gmail.com) [ayush.velhal@gmail.com](mailto:ayush.velhal@gmail.com)
* **Connecting Front-End pages to Backend with API Calls**
  + Assigned:[Adair Gonzalez](mailto:axg230107@utdallas.edu)
* **Feature Testing**
  + Navbar functionality
    - Assigned: [Shannon Carter](mailto:shannon.m.carter.18@gmail.com)
  + Recipe Generation:
    - Assigned: [Roslyn Collings](mailto:rccollings27@gmail.com)
  + Saved Recipes:
    - Assigned: [cotlqlc21@gmail.com](mailto:cotlqlc21@gmail.com) [hafakazi@gmail.com](mailto:hafakazi@gmail.com)

**2.  ​OUR REPOSITORY URL LINK:​** [***https://github.com/shannonmcarter18/CS3354-Recipe-Generator***](https://github.com/shannonmcarter18/CS3354-Recipe-Generator)

**2.1 Figma page design Link**

[CS 3345: Recipe Generator](https://www.figma.com/proto/07YaUxtu4BKF3OnapjvXzC/CS-3345%3A-Recipe-Generator?node-id=6-2&t=PQqDxRwvKj9Xod9q-1&scaling=contain&content-scaling=fixed)

**3. Delegation of tasks:**

|  |  |
| --- | --- |
|  |  |
| Hafa Kazi: | Outlining Software Process Model, Design “Saved Recipes” page, Set up environment |
| Shannon Carter: | Project Leader, Fill out delegation of tasks, Create and Share Figma Design File, Design and Prototype “Landing” Page, create Github Repository, Set up environment, Applying Architecture |
| Rhea Aemireddy: | Final Project Draft Description, Design “Landing” Page, Set up environment |
| Rosyln Collings : | Document Submission, Design “Saved Recipes” Page, Set up environment |
| Jay Chae: | Design “Saved Recipes” Page, Set up environment |
| Veom Nemade: | Creating Sequence Diagrams for Use Cases, Configure MongoDB/PostgreSQL, ensure all endpoints are functional |
| Adair Gonzalez: | Handling Software Requirements both functional and non-functional, design “Recipe Generator” Flow, planning for frontend and backend connections |
| Ayush Velhal: | Design “Recipe Generator” Flow, Creating Use Case Diagrams, Designing Class Diagrams, Document Revision |

**4. Software Process Model​**

We will be using an Agile process model due to its adaptability and effectiveness on a small scale. We will be using the Scrum methodology.

**5.  Software Requirements**

**5.a.) Functional Requirements:**

**Recipe Generation from Input**

* The system shall allow users to input a list of ingredients and/or a dish name.
* The system shall generate a recipe based on the provided ingredients and/or dish name.
* The generated recipe shall include a list of ingredients, preparation steps, and cooking instructions.

**Recipe Display**

* The system shall display the generated recipe to the user in a clear and readable format via the frontend (React.js).

**Recipe Saving**

* The system shall provide an option for the user to save the generated recipe.
* If the user chooses to save, the system shall store the recipe in the database (MongoDB/PostgreSQL).
* The system shall allow users to retrieve previously saved recipes from the database.

**User Interaction**

* The system shall provide a user-friendly interface (via React.js) for entering inputs and viewing recipes.
* The system shall confirm successful saving of a recipe with a notification or message to the user.

**5.b.) Non-functional requirements**

**Scalability**

* The system shall support scaling individual services (frontend, backend, database) independently to handle increased user demand, as justified by the Microservices Architecture.

**Performance**

* The system shall generate and display a recipe within 5 seconds of receiving user input under normal load conditions.
* The system shall retrieve saved recipes from the database within 2 seconds.

**Maintainability**

* The codebase shall be well-documented to facilitate future development by team members.

**Reliability**

* The system shall ensure that saved recipes are accurately stored and retrievable with a 99% success rate.
* The system shall handle invalid inputs (e.g., empty ingredient list) gracefully by providing appropriate error messages.

**Usability**

* The frontend shall be intuitive and responsive, ensuring users can easily input data and view recipes on both desktop and mobile devices.
* The system shall support a consistent user experience across different browsers (e.g., Chrome, Firefox).

**Security**

* The system shall protect user data (e.g., saved recipes) by implementing secure database access controls.
* The system shall validate user inputs to prevent injection attacks or malicious data entry.

**Availability**

* The system shall be available 95% of the time, excluding planned maintenance, to ensure users can access the recipe generator when needed.

**6.  Diagrams (Use-Case/Sequence/Class/Activity)**

Sequence Diagram:

A diagram of a recipe

AI-generated content may be incorrect.

Use Case diagram:

A diagram of a recipe

AI-generated content may be incorrect.

Class diagram:

A diagram of a recipe

AI-generated content may be incorrect.

**7.  Architectural design**

**7.1.  Describe why the pattern is selected**

The **Microservices Architecture** pattern is chosen for scalability and maintainability. The system is divided into:

* **Frontend (React.js)**: Handles UI interactions.
* **Backend (Flask/FastAPI)**: Manages recipe generation logic and database interactions.
* **Database (MongoDB/PostgreSQL)**: Stores user-generated recipes.
* **External API (AI Model, if applicable)**: Fetches or generates recipe data.

**Justification for Microservices Architecture:**

* **Scalability**: Individual services can scale independently.
* **Flexibility**: Backend can support multiple frontends.
* **Maintainability**: Clear separation of concerns.

**8.  Final Project Draft Description**

**Project Overview**

The **Recipe Generator** is a web-based application designed to help users generate recipes based on available ingredients or a dish name. The system allows users to **input ingredients, view generated recipes, and save them for future use**. Built with **React.js (frontend), Flask/FastAPI (backend), and MongoDB/PostgreSQL (database)**, it follows a **Microservices Architecture** to ensure scalability, flexibility, and maintainability.

**Motivation & Goals**

The project aims to assist users in **meal planning, reducing food waste, and saving time** by providing tailored recipe recommendations. As busy college students, our team recognizes the need for **a quick and efficient way to find recipes** based on available ingredients.

**Key Features**

* Users can **input ingredients or a dish name** to generate recipes.
* Recipes include a **list of ingredients, preparation steps, and cooking instructions**.
* Users can **save and retrieve recipes** from a database.
* The UI is **user-friendly, responsive, and accessible on multiple devices**.
* The system ensures **fast performance, security, and high availability**.

**Technology Stack**

* **Frontend:** React.js (with TailwindCSS, Framer Motion)
* **Backend:** Flask or FastAPI (with Axios for API calls)
* **Database:** MongoDB/PostgreSQL
* **Version Control:** Git/GitHub

**Development Approach**

The project follows an **Agile development model using Scrum**, ensuring iterative improvements. Tasks were divided across two deliverables:

1. **Deliverable 1:** Prototyping UI (Figma), system design, and environment setup.
2. **Deliverable 2:** Frontend-backend implementation, database integration, and feature testing.

**Expected Outcomes**

By implementing a **scalable, efficient, and secure recipe generation system**, we aim to provide users with a **convenient tool for meal planning**, ultimately helping them make **better use of available ingredients and reduce food waste**.