Cookery Final Architecture Plan

Dempsey Evans

Capstone Project Final Architecture & Design

Grand Canyon University

Professor Landreth

Revision: 2

May 18th, 2025

**ABSTRACT**

The Cookery Application aims to create a space where users can manage their recipes and ingredients effectively. By allowing users to create their own profile and recipes, the users have more control over their experience while using the application. Cookery also aims to make the meal planning process more efficient by including ingredient price ranges for calculating meal cost and providing a recipe generator feature. Using the recipe generator the user will have the opportunity to generate recipes based on price range, available ingredients, or descriptors. This eliminates user stress in last minute meal planning or when planning on a budget and encourages users to lessen their food waste by using what they have on hand.

The main application functionality is user registration, user login, recipe management, ingredient management, and a recipe generator. User authentication using Spring Security will allow each user to have their own protected account and recipes. Once in the application, users will be able to manage recipes and ingredients by viewing, creating, editing, and deleting them on the recipes page and ingredients page. Simple management functionality will provide consistency that will make the recipes and ingredients easy to read and manage. Using the ingredient price range, cost for recipes will be calculated when the user creates a new recipe. The recipe generator will use the existing recipes to generate a random recipe using user given criteria. Other functionalities include a user feedback form, searching for recipes based on recipe names and recipe descriptors.

By creating a platform for recipe management users will be able to house their recipes in one place, saving time searching for recipes across numerous websites and books. It saves the user time and energy on what is an important yet typically time-consuming and repetitive process. Application growth will be strongly based on user feedback. Users will be able to easily provide feedback on their user profile when logged into the application. By focusing on user needs the application will be able to continue to act as a useful tool for meal management as users and needs change.

|  |
| --- |
| History and Signoff Sheet |

**Change Record**

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Revision Notes** |
|  |  | Initial draft for review/discussion |
|  |  |  |
|  |  |  |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

**Integrated Instructor Feedback into Project Documentation**

Yes  No

**TABLE OF CONTENTS**

Design Overview 4

Detailed High-Level Solution Design 5

Detailed Technical Design 6

Appendix A – Technical Issue and Risk Log 7

Appendix B – References 8

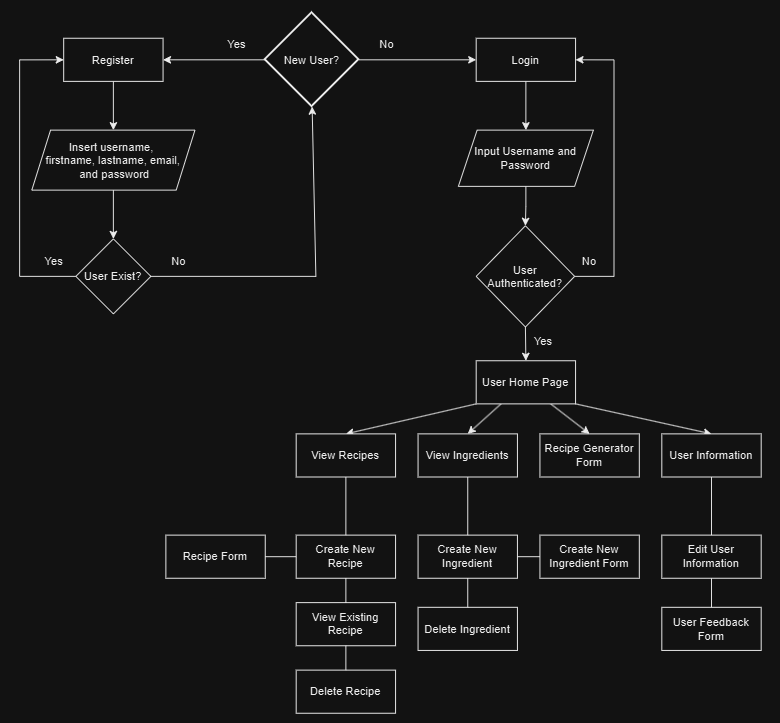
Appendix C – External Resources 9

Design Planning Summary

For many adults the meal planning process feels tedious, overwhelming, or difficult. In the modern world most home cooks find their recipes online. With an endless possibility of platforms for recipes, it becomes difficult to organize and manage the chosen recipes in a singular platform. With Cookery, users can use these recipes to create their personal use online cookbook, giving users quick access to all of their favorite recipes in a singular platform. Another challenge in meal planning is understanding and sticking to a budget. By using user input ingredient cost range, the application calculates the recipe costs. This gives users financial clarity when deciding which meals match their budget. To support effective meal planning the recipe generator uses the users’ recipes and user provided criteria from a provided form to generate a random recipe. This tool will greatly speed up and simplify the meal planning process for many users.

Design Introduction

The goal of the Cookery application is to provide users with a clean and simple platform to create and manage recipes. New users register for an account using an unregistered e-mail, then log in to access the application. When logging in the users must first be authenticated using Spring Security. If they are successfully authenticated the user is then directed to their homepage which contains a navigation bar to access the other pages, such as the recipe generator and ingredients page. The recipe form to create a new recipe can be found on the View Recipes (or home) page, and the create new ingredient form can be accessed on the Ingredients page.



Flowchart

User functionality includes the registration and login process as well as creating, modifying, and deleting recipes and ingredients. Users will also be able to access the recipe generator functionality to randomly generate existing recipes with an ingredient-based criteria option. Other functionalities include searching for an existing recipe, updating user information, and a user feedback form. Wireframes are further broken down in the [User Interface Diagrams](#UserInterfaceDiagrams) section.

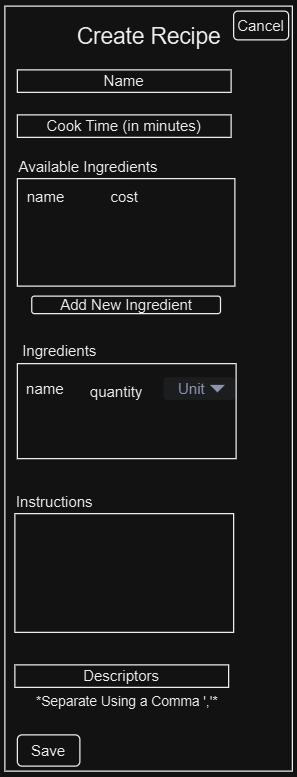
A screenshot of a login screen

Description automatically generated A screenshot of a black screen

Description automatically generated

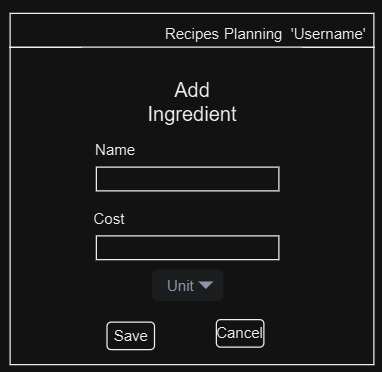
User Login Page User Registration Page

A screenshot of a recipe

Description automatically generated 

Application Recipe/Home Page Create new Recipe Form

A screenshot of a recipe menu

Description automatically generated 

Ingredients Page Add New Ingredient Form

A screenshot of a recipe

Description automatically generated

Recipe View Form

A screenshot of a recipe

Description automatically generated

Recipe Generator

A screenshot of a computer

Description automatically generated

User Information Page

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Deliverable Acceptance Log | | | | | |
| ID | Deliverable Description | Comments | Evaluator (internal or external as applicable) | Status | Date of Decision |
| 1 | API Design | APIs, endpoints, and methods design | Internal | Pending | TBD |
| 2 | Project Proposal | Initial proposal for project with project scope, timeline, and digrams | Internal | Pending | TBD |
| 3 | Project Requirements | Functional and Nonfunctional requirements | Internal | Pending | TBD |
| 4 | User Stories | Describes user-based and system functionality | Internal | Pending | TBD |
| 5 | Development | Create the backend and frontend of the application | Internal | Pending | TBD |
| 6 | Maintenance Plan | Describes steps and procedures for maintenance | Internal | Pending | TBD |

Detailed High-Level Solution Design

A screenshot of a computer

Description automatically generated

The UML Component Diagram above breaks down the application into user interface, business logic, data access, and database components. Listed under each component is the elements for the component.

A diagram of a web server

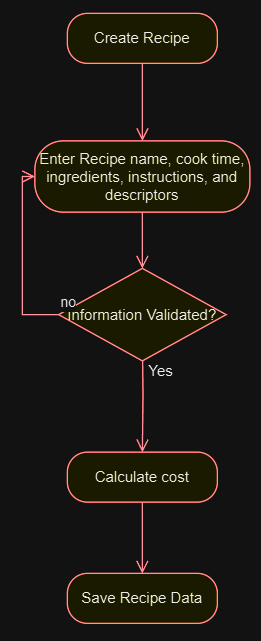
Description automatically generated

The UML Deployment Diagram is shown above. The user uses their device to connect to the internet and web server where the application is held. The authentication server for user authentication and the MySQL database is held on the web server as well.

**A diagram of a computer

Description automatically generated**

Login and Registration Activity Diagram

** A diagram of a process

Description automatically generated**

Create Recipe Activity Diagram Create Ingredient Activity Diagram

A diagram of a recipe

Description automatically generated A diagram of a flowchart

Description automatically generated

Delete Recipe/Ingredient Activity Diagram Modify Recipe/Ingredient Activity Diagram Diagram

A diagram of a recipe

Description automatically generated A screenshot of a cell phone

Description automatically generated

Recipe Generator Activity Diagram User Feedback Activity Diagram

**Configuration Changes**

|  |  |
| --- | --- |
| Configuration | Changes |
| Database Configuration | -Create database schema |
| Server Configuration | -Environment variables |
|  | -Application testing |
| User Authentication | -Spring Security user authentication setup |

**Meeting Non-Functional Requirements**

Non-functional requirements will be met in this solution using official documentation such as the Spring Security “Getting Started” guide and other credible resources provided in the references below. Using spring security, endpoints will be secured for authorized user access only. To ensure that error messages are available and effective, error messages will be implemented in the controllers for methods associated with user inputs and outputs. In approaching the recipe generator, efficient database queries will be essential to pull recipes correctly based on any given ingredient criteria. Resources to support this include the SQL Tutorial by W3Schools for correct SQL query usage.

**Frameworks and Third-Party Libraries**

* Spring Boot
* Spring Security
* Spring Data JPA
* MySQL

|  |  |  |
| --- | --- | --- |
| Proof of Concepts | |  |
| **Description** | **Rationale** | **Results** |
| 1. Spring Security User Authentication | Validate the implementation of user authentication and authorization using Spring Security. | Complete |
| 1. Password Reset Functionality | Ensure that the user can successfully reset their password. | Pending |
| 1. Recipe Generator Performance | Verify that the database queries support correct recipe generation. | Complete |
| 1. Recipe Management Performance | Validate the creation, modification, and deletion of recipes by the user. | Complete |
| 1. Ingredient Management Performance | Validate the creation, modification, and deletion of ingredients by the user. | Complete |
| 1. User Feedback Form | Ensure that the user feedback form is user-friendly, and user feedback is correctly stored. | Complete |
| 1. Thymleaf & CSS | Thymleaf template to process and render UI data in a user friendly way. | Complete |
| 1. Maven | Dependency management and automated build process | Complete |

|  |
| --- |
| Hardware and Software Technologies |
| 1 – Integrated Development Environment. VS Code will be used for this project. |
| 2 – Email account |
| 3 – MySQL Database Server. The local database using MAMP will be used in this project. |
| 4 – Github Account |
| 6 - Git |
| 7 – Maven |
| 8 – Testing Framework |
| 9 – Postman |
| 10 – GitHub CI/CD Configuration |

**Logical Solution Design:**

A diagram of a user flow

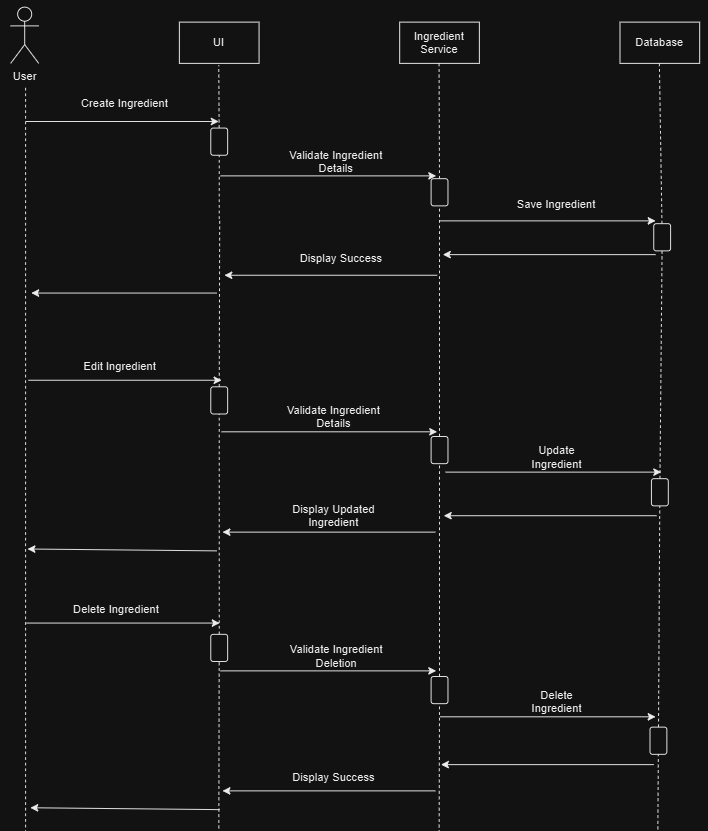
Description automatically generated

Registration and Login Sequence Diagram

A black screen with white arrows

Description automatically generated

Recipe Management Sequence Diagram

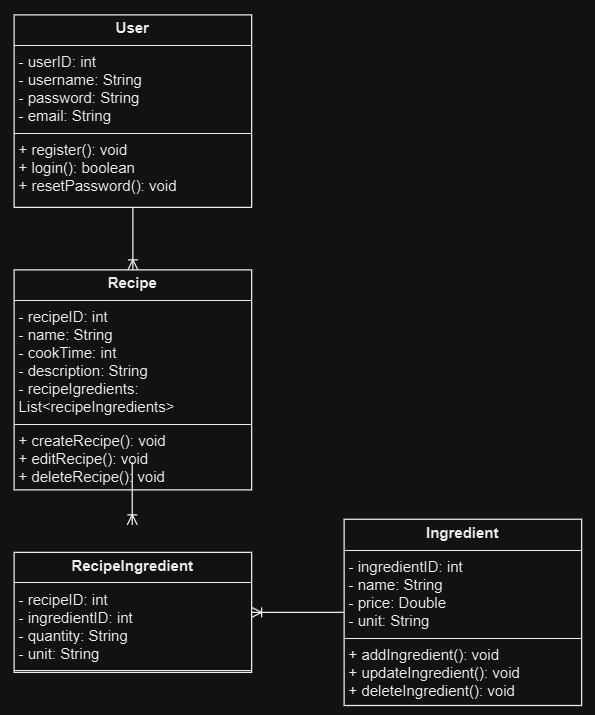


Ingredient Management Sequence Diagram

A diagram of a recipe

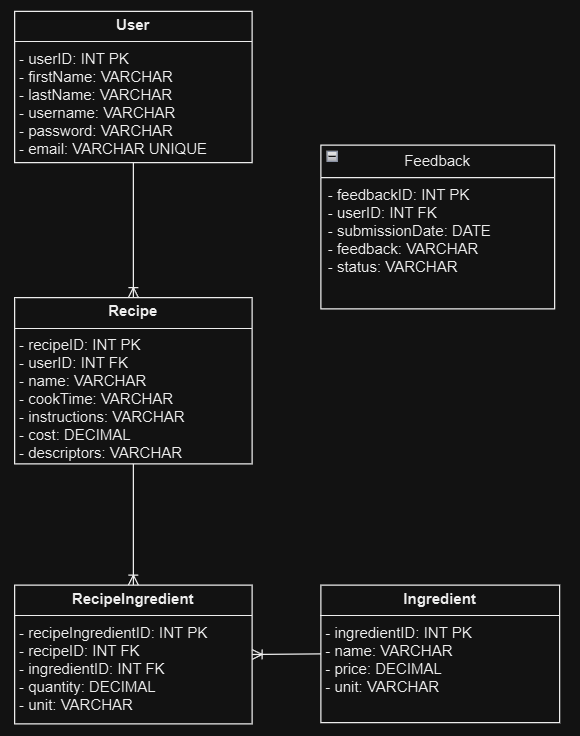
Description automatically generated

Recipe Generator Sequence Diagram



[Class Diagram](#UMLDiagrams)

**Physical Solution Design:**



Database Schema

Detailed Technical Design

**General Technical Approach:**

The Cookery application is designed to focus on manageability, scalability, and positive user experience. Using the Model-View-Controller architecture with a focus on Separation of Concerns, the application with be easier to develop, test, and manage.

In the frontend, Thymleaf and CSS will ensure responsive and user-friendly pages. New users will be required to register to ensure a separation between users and more personalized experience. Spring Security will manage user authentication and secure the REST APIs to prevent unwanted access to the application and its features.

In the backend, the Sprint Boot application will support user management, recipe management, and ingredient management. User management consists of user registration, authentication, and user information. Recipe and ingredient management will use user service classes and repositories to query the database and allow users to create, modify, and delete recipes and ingredients. When a user creates a new recipe, the user will choose ingredients then input the quantity and unit for each ingredient. This will be used with the ingredients price range, which is input using the create ingredient form, to calculate the cost of new recipes. The recipe generator functionality will have a tight dependency on the recipe management, sorting through existing recipes then display a random recipe that is either completely random or aligns with the given criteria. Additional functionalities include a user feedback form, search bar for recipes, and password recovery.

The application will use a MySQL database, which can be set up using the provided DDL script. The database has a User, Recipe, Ingredient, RecipeIngredient, and Feeback table. Foreign keys are used to create relationships, such as the Recipes to the User table using the UserID foreign key and RecipeIngredients to the Ingredient and Recipe table using the IngredientID and RecipeID foreign key.

**Key Technical Design Decisions:**

|  |  |  |
| --- | --- | --- |
| Technology/Framework | Purpose | Reason |
| CSS | Ensure a user-friendly interface. | - Ease of use  - Many resources for support. |
| Thymleaf | Process and render UI data | - Easy to use templates  - Many resources for support |
| Spring Boot | Handle the backend logic for the application | - Many available features for simple and quicker development  - Easy integration with Spring Security. |
| Spring Security | Manage user authentication and secure REST APIs | - Flexible framework with password management, secure REST API, and user authentication abilities, - Large community with many resources for support |
| MySQL Database | To store and manage application data | - Reliable and efficient database |
| GitHub | Manage the git repository and track issues | - Easy to use and track issues  - Easy integration with Azure |
| Maven | Dependency management and build automation | - Easier development |
| Log4j2 | Configuration logging to print messages to the user | - effective logging framework |

**Database ER Diagram:**

A diagram of a company

Description automatically generated with medium confidence

The entity relationship diagram above shows the relationship between the entities and their attributes. One user can have many userFeedback submissions as well as many recipes. One recipe can have many ingredients. The recipeIngredients connects the recipe and ingredients by representing the ingredients use in the recipe. This allows the recipeIngredient to be modified without modifying the ingredient stored in the database.

**Database DDL Scripts:**

--Create User Table—

CREATE TABLE ‘User’ (

‘userID’ INT PRIMARY KEY AUTO\_INCREMENT,

‘firstName’ VARCHAR(50) NOT NULL,

‘lastName’ VARCHAR(50) NOT NULL,

‘username’ VARCHAR(50) UNIQUE NOT NULL,

‘password’ VARCHAR(50) NOT NULL,

‘email’ VARCHAR(100) UNIQUE NOT NULL );

-- Create Recipe Table --

CREATE TABLE ‘Recipe’ (

‘recipeID’ INT PRIMARY KEY AUTO\_INCREMENT,

‘userID’ INT,

‘name’ VARCHAR(100) NOT NULL,

‘cookTime’ VARCHAR(100) NOT NULL,

‘instructions’ TEXT NOT NULL,

‘cost’ DECIMAL(10, 2),

‘descriptors’ VARCHAR(200),

FOREIGN KEY (‘userID’) REFERENCES User(‘userID’));

-- Create Ingredient Table --

CREATE TABLE Ingredient (

‘ingredientID’ INT PRIMARY KEY AUTO\_INCREMENT,

‘name’ VARCHAR(100) NOT NULL,

‘price’ DECIMAL(10, 2) NOT NULL

‘unit’ VARCHAR(50) NOT NULL );

-- Create RecipeIngredient Table –

CREATE TABLE ‘RecipeIngredient’ (

‘recipeIngredientID’ INT PRIMARY KEY AUTO\_INCREMENT,

‘recipeID’ INT,

‘ingredientID’ INT,

‘quantity’ DECIMAL(10, 2) NOT NULL,

‘unit’ VARCHAR(50) NOT NULL,

FOREIGN KEY (‘recipeID’) REFERENCES Recipe(‘recipeID’),

FOREIGN KEY (‘ingredientID’) REFERENCES Ingredient(‘ingredientID’);

-- Create Feedback Table –

CREATE TABLE ‘Feedback’ (

‘feedbackID’ INT PRIMARY KEY AUTO\_INCREMENT,

‘userID’ INT,

‘submissionDate’ DATE NOT NULL,

‘feedback’ TEXT NOT NULL,

‘status’ VARCHAR(50) NOT NULL,

FOREIGN KEY (‘userID’) REFERENCES User(‘userID’);

**Flow Charts/Process Flows:**

**A diagram of a user home page

Description automatically generated**

Flow chart

A diagram of a computer

Description automatically generated

Registration and Login Activity Diagram

Registration and Login Activity Diagram. The user accesses the login page then either registers or logs in. After logging in the user is directed to the Home/Recipes page.

A diagram of a flowchart

Description automatically generated A diagram of a process

Description automatically generated

Create Recipe Activity Diagram Create Ingredient Activity Diagram

When a user creates a new recipe or ingredient they enter the necessary information, which is then validated to ensure that each field is properly filled out. When creating a recipe, the cost range is then calculated after the information is validated. After validation (and cost calculation for recipes) the information is saved.

A diagram of a recipe

Description automatically generated A diagram of a flowchart

Description automatically generated

Recipe & Ingredient Deletion Activity Diagram Recipe & Ingredient Modify Activity Diagram

When deleting a recipe or ingredient the user selects the item to be deleted, confirms the deletion, then the item is deleted from the database. When modifying a recipe or ingredient the user selects the item to be modified, modifies the desired fields, then saves the updated item to the database.

A diagram of a recipe

Description automatically generated

Recipe Generator Activity Diagram

When generating a random recipe the user optionally inputs their desired criteria, the recipe generates the random recipe, then the recipe is displayed for the user.

A screenshot of a cell phone

Description automatically generated

User Feedback Form Activity Diagram

To submit feedback the user enters their feedback into the user feedback form, saves the feedback, then displays a successful submission confirmation for the user.

**Sitemap Diagram:**

A diagram of a recipe

Description automatically generated

**User Interface Diagrams:**

To access the application new users must first register. On the login page the user has a link to access the registration page, where they must submit an email, first name, last name. username, and password. If the email is unused and the user successfully registers, they will then be directed to the login page. On the login page the user submits their username and password to be authenticated using Spring security. If the login is successful, the user will be directed to their home page.

A screenshot of a black screen

Description automatically generated A screenshot of a login screen

Description automatically generated

Registration Page User Login Page

The Recipes page acts as the home page, presenting the users’ recipes as a list. Using the search bar users can quickly find existing recipes and modify or delete existing recipes. The functionality to create new recipes is accessed on the Recipes page using a button on the bottom of the screen to bring up the Create Recipe form. On this form the user inputs the information for the new recipes including descriptors that will be used with the search functionality. When including ingredients the user can pick from existing ingredients then input the price range for each ingredient. The user can also add new ingredients on the create recipe form for efficiency. After submitting the recipe, the user will see the recipe added on their Recipes page. The user can then click on an existing recipe to view the recipe form with all of the information for the chosen recipe.

A screenshot of a recipe

Description automatically generated A screenshot of a computer

Description automatically generated

Recipe Home Page Create Recipe Form

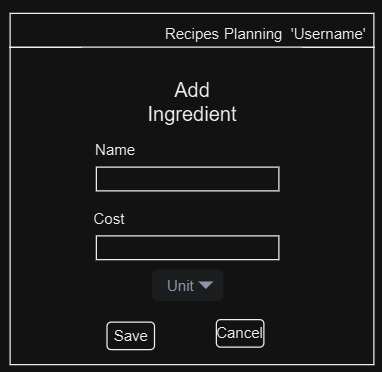
A screenshot of a recipe

Description automatically generated

Recipe View Form

On the ingredients page the user can add, edit, and delete ingredients, as well as search for ingredients by name. The Add New Ingredients button brings the user to the Add Ingredient form where the user inputs the name and price range of the ingredient. The price range is then used in the recipe form to calculate the cost of the recipe.

A screenshot of a recipe menu

Description automatically generated 

Ingredients Page Add Ingredient Form

The meal generator allows the user to insert a minimum and maximum price range, ingredients, or descriptors. The user then clicks the generate recipe, the existing recipes are sorted into a list using the user given criteria, then a random recipe is selected and displayed on the bottom of the page.

A screenshot of a recipe

Description automatically generated

Recipe Generator

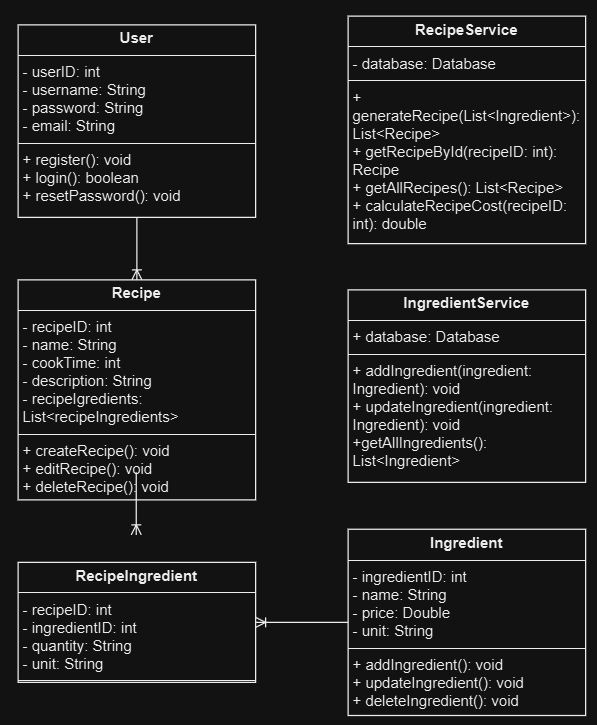
The user information page allows users to view their personal information that was submitted upon registration. On this page the user can edit their information or submit user feedback using the feedback form on the bottom of the page

A screenshot of a computer

Description automatically generated

User Information Page

**UML Diagrams:**

****

Class Diagram

The key classes in this project are the User, Recipe, RecipeIngredient, and Ingredient classes. The user class manages user information and methods for registering, logging in, and resetting the password. The recipe class contains the same attributes as the recipe database table with a list for ingredients and methods for creating, editing, and deleting recipes. The recipeIngredient class is used to represent the ingredients used in the recipe. The ingredient class attributes align with the ingredient table with methods for adding, updating, and deleting an ingredient. The application also uses a recipe and ingredient service class and repositories to interact with the database and manipulate data. The recipe service class has methods to generate, search for, and get recipes as well as a method to calculate recipe cost. The ingredient service has methods for getting ingredients. The repository will allow data to be pulled from and saved to the database.

A diagram of a user flow

Description automatically generated

Registration and Login Sequence Diagram

The sequence diagram above represents the experience of a new user. The user enters their user details on the registration page, the users are validated by the Authorization Service, then the user account is created and saved to the database. After the user has a successful registration they enter their user credentials into the login page, which is then sent from the UI to the Authentication Service for validation. The user is then presented with a successful or failed login.

A black screen with white arrows

Description automatically generated

Recipe Sequence Diagram

The sequence diagram above represents creating, editing, and deleting a recipe. When creating a recipe the user inputs the data, which is then sent to the Recipe Service for validation, then upon successful validation is saved to the database. When editing or deleting a recipe the user first inputs the information or chooses a recipe to delete. The information is then validated through the Recipe Service and, if successful, is saved in the database.

A black screen with white text

Description automatically generated

Ingredient Sequence Diagram

The ingredient sequence diagram above represents creating, editing, and deleting ingredients. This process aligns with the process for recipe management in the previous diagram.

A diagram of a recipe

Description automatically generated

Recipe Generator Sequence Diagram

In the diagram above, which represents the recipe generator, the user inputs criteria, which is then sent to the recipe service. The recipe service then pulls and sorts through recipes in the database and displays a random recipe for the user.

**Service API Design:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **API Endpoint** | **Purpose** | **How to Access** | **Parameters** | **JSON Data Format** |
| …/register | Allow new users to register | Through a link on the login page | - username  - firstName  - lastName  - password  - email | {  "firstName": "string", "lastName": "string", "username": "string", "password": "string", "email": "string" } |
| …/login | Allow users to login | Entering the correct domain name | - username  - password | {  "username": "string",  "password": "string"  } |
| …/recipes | Create a new recipe | On the recipe homepage Create Recipe form | - recipeID  - userID  - name  - cookTime  - ingredients  - instructions  - cost  - descriptors | {  "name": "string", "ingredients": [ { "ingredientId": integer, "quantity": decimal, "unit": "string" } ], "instructions": "string", "userId": integer } |
| …/recipes/{recipeID} | Search for a recipe | On the recipe home page | -name  -descriptors |  |
| …/recipes/{recipeID} | Update an existing recipe | On the recipe home page | - recipeID  - userID  - name  - cookTime  - ingredients  - instructions  - cost  - descriptors | {  "name": "string", "ingredients": [ { "ingredientId": integer, "quantity": decimal, "unit": "string" } ], "instructions": "string", "userId": integer } |
| …/recipes/{recipeID} | Delete an existing recipe | On the home page | -recipeID |  |
| …/ingredients | Create and update ingredients | On the ingredients page | - ingredientID  - name  -price  - unit | {  "name": "string", "price": decimal, "unit": "string"  } |
| …/ingredients/{ingredientID} | Delete an existing recipe | On the ingredients page | - ingredientID |  |
| …/recipe-generator | Generate a recipe based on user criteria | Through the navigation bar within the application | - userID  -ingredientID | {  "userId": integer, "criteria": { "ingredientId": integer, "maxCost": decimal }  } |
| …/users/{userID} | View user information | On the user info page | - userID  - firstName  - lastName  - password  - email | {  "userId": integer, "firstName": "string", "lastName": "string", "username": "string", "email": "string"  } |
| …/users/{userID}/feedback | Submit user feedback | On the user info page | - feedbackID  - userID  - submissionDate  - feedback  - status | {  "userId": integer, "feedback": "string"  } |

**Operational Support Design:**

The specific monitoring tool for long term monitoring is yet to be decided

Logging will be implemented using the Log4j2 logging framework. Messages will be created for users to notify them of failed and successful data submissions, such as when registering, logging in, creating a new recipe, or deleting a recipe. This will also allow the use of warning levels allowing errors to be categorized. A file will be created to store the logs by adding a property in the properties file for maintenance purposes.

**Other Documentation:**

Task List

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Task | Description | Effort Hours | Start Date | Planned Completion | Sprint |
| 1 | Setup Environment | Configure the Spring Boot environment | 3 | 12/30/2024 | 12/30/2024 | 1 |
| 2 | Create MySQL Database Tables | Create tables for use with entities | 1 | 12/30/2024 | 01/02/2025 | 1 |
| 3 | Implement Spring Security | Create login and registration for Spring Security use | 5 | 01/01/2025 | 01/07/2025 | 2 |
| 4 | Core Recipe Functionality | Create, read, update, and delete methods for recipes | 6 | 01/08/2025 | 01/10/2025 | 2 |
| 5 | Create Recipe Form | Frontend and backend functionality to create a new recipe | 4 | 01/11/2025 | 01/13/2025 | 2 |
| 6 | Core Functionality for Ingredients | Create, update, and delete ingredients | 5 | 01/23/2025 | 01/28/2025 | 3 |
| 7 | Create Ingredient Form | Form to create new ingredients | 3 | 01/29/2025 | 01/30/2025 | 3 |
| 8 | Recipe Cost Calculation | Add logic for cost calculations | 5 | 01/31/2025 | 02/05/2025 | 4 |
| 9 | Recipe Generator Functionality | Logic for random recipe generation | 8 | 02/06/2025 | 02/09/2025 | 4 |
| 10 | Recipe Generator interface | Create page for recipe generator | 3 | 02/06/2025 | 02/13/2025 | 4 |
| 11 | User Acceptance Testing | Test that application meets user requirements | 3 | 02/14/2025 | 02/16/2025 | 5 |
| 12 | Post Deployment Monitoring | Set up monitoring and logging for the application | 4 | 02/27/2025 | 03/01/2025 | 6 |

Appendix A – Technical Issue and Risk Log

1. Use the template to identify and monitor project issues and risks.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Issues and Risk Log | | | | | | | | |
| **Issue or Risk** | **Description** | **Project Impact** | **Action Plan/Resolution** | **Owner** | **Importance** | **Date Entered** | **Date to Review** | **Date Resolved** |
| 1 | Incomplete understanding of recipe generator development implementation | Delay in recipe generator functionality or unsuccessful implementation of functionality | Research methods to handle data in a manner to randomize recipes using criteria | Development Team | *High* | *10/26/2024* | *10/30/2024* | *11/02/2024* |
| 2 | Questionably accurate timeline | Improper development plan, potentially impacting deadlines | Sprint estimation review process to ensure tasks estimations are accurate | Resource Manager | Medium | 10/26/2024 | 10/27/2024 | 10/27/2024 |
| 3 | Scope lacks detail and does not show what is out of scope | Miscommunication throughout the project/development | Review the scope and include what is out of scope | Project Manager | High | 10/30/2024 | 10/30/2024 | 11/10/2024 |
| 4 | Project success measures levels of success is missing | Unable to determine and test for quantitative success | Include scores, rating, and test pass rates | Project Manager | High | 10/30/2024 | 10/31/2024 | 11/11/2024 |
| 5 | Missing stakeholders | Inaccurate account of contributors/impactors of the project | Add missing stakeholders in the stakeholders table | Project Manager | Medium | 10/30/2024 | 10/31/2024 | 10/30/2024 |
| 6 | Missing quality assurance and post deployment testing tasks | Insufficient and inaccurate functionality testing | Add the missing task in the work breakdown structure | Project Manager | High | 10/30/2024 | 11/01/2024 | 10/30/2024 |
| 7 | Changes to task require the timeline to be reviewed | Inaccurate work plan for the development phase | Sprint estimation review process to ensure tasks estimations are accurate | Resource Manager | Medium | 10/30/2024 | 11/01/2024 | 11/04/2024 |
| 8 | Ingredients table missing | Incomplete and unusable database schema | Update the database schema to include an ingredient table | Development Team | High | 10/30/2024 | 11/02/2024 | 10/30/2024 |
| 9 | Missing functionality for password recovery | Inability for registered users to access the application | Add ability for users to recover a forgotten password | Development Team | High | 11/01/2024 | 11/03/2025 | 11/07/2024 |
| 10 | Ingredient quantities missing | Missing or unsuccessful recipe cost calculation functionality | Add ingredient quantity attribute to the ingredient | Development Team | High | 11/21/2024 | 11/23/2024 | 11/28/2024 |
| 11 | Risk of the recipe costs calculation not being complex enough | Complicated or inaccurate user experience for recipe costs calculations | Research methods for conversions and calculating cost | Development Team | High | 11/21/2024 | 11/21/2024 | 02/28/2025 |
| 12 | Missing unit attribute when creating ingredients | Inaccurate or insufficient information to calculate ingredient costs within recipe | Add a unit attribute to the ingredient form so the ingredient price can correlate to the unit | Development Team | High | 12/06/2024 | 12/06/2024 | 12/06/2024 |

Appendix B – References

Eugen, P. (2020). Spring Security – Reset Your Password. <https://www.baeldung.com/spring-security-registration-i-forgot-my-password>

Spring (n.d.). Hello Spring Security. <https://docs.spring.io/spring-security/reference/servlet/getting-started.html>

W3Schools (n.d.). SQL Tutorial. <https://www.w3schools.com/sql/sql_any_all.asp>

Appendix C – External Resources

|  |  |
| --- | --- |
| **GIT URL:** | *https://github.com/dempseyevans/Cookery.git* |