System Design

CRC Cards:

Class Name: Recipe	
Responsibilities: - Store recipe details such as name, ingredients, instructions, tags (e.g., vegan, gluten-free). - Provide a method for filtering recipes based on ingredients or dietary restrictions. - Provide recipe details when requested (e.g., ingredients, preparation instructions).	Collaborators: - RecipeController: Interacts to send recipe-related data to the user. - RecipeService: Handles business logic and data processing. - Database: Retrieves and stores recipe data.

Class Name: User	
Responsibilities: - Store user preferences such as dietary restrictions (e.g., vegetarian, vegan, gluten-free). - Track user search history and provide personalized recipe recommendations. - Update and manage preferences (e.g., adding new dietary restrictions).	Collaborators: - UserController: Handles user interactions (e.g., updating preferences). - RecipeService: Uses user preferences to filter recipes. - Recipe: Receives and filters recipes based on the user's dietary restrictions.

Class Name: UserController	
Responsibilities: - Handle HTTP requests related to user actions, such as login, registration, or preference updates. - Validate and process user input, passing it to the UserService or Database.	Collaborators: - UserService: Manages the logic for handling user data and preferences. - Database: Stores or retrieves user-related data (e.g., preferences, search history).

Class Name: UserService	
Responsibilities: - Handle user authentication (e.g., login, registration). - Manage user preferences (e.g., dietary restrictions, personal	Collaborators: - UserController: Interacts with UserController to process user requests like updating preferences or logging in.

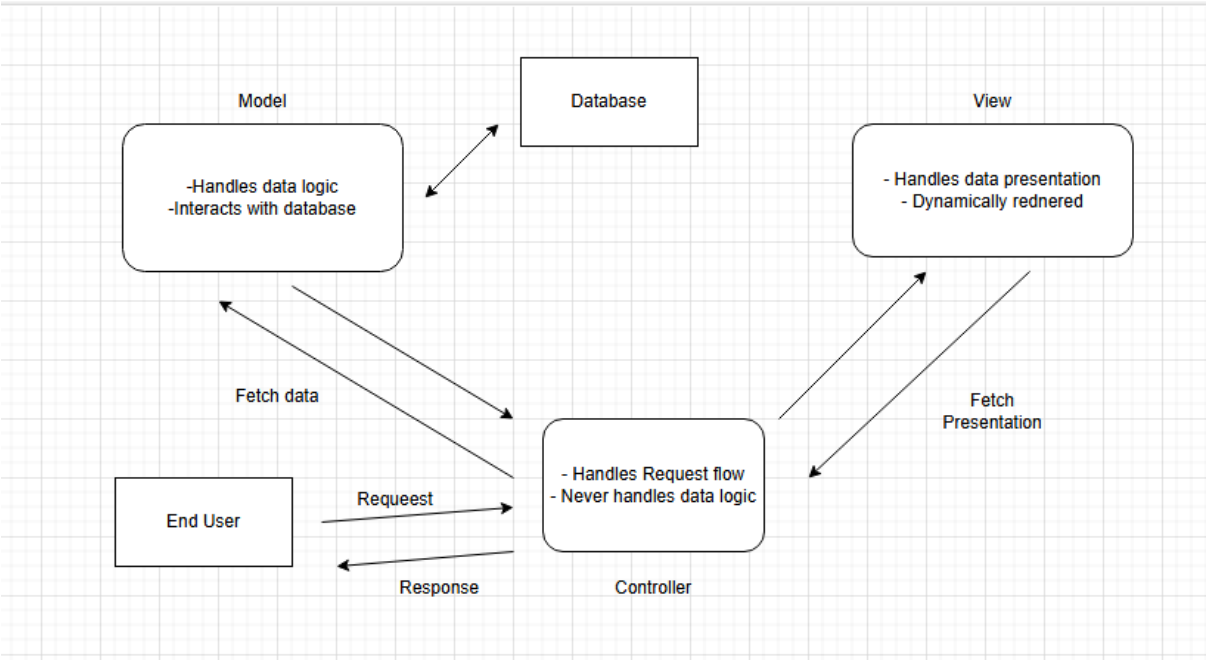
settings). - Track and retrieve user search history for personalized recommendations. - Interface with the Database to store and retrieve user data.	- Database: Stores and retrieves user-related data (preferences, search history). - User: Represents the user data model but is managed by the UserService for business logic.

Class Name: RecipeController	
Responsibilities: - Handle HTTP requests for recipe data (e.g., searching, fetching details). - Call RecipeService to process data and return the result to the user. - Ensure that search results are filtered based on user preferences and query parameters. -	Collaborators: - RecipeService: Handles the logic and querying of recipes. - User: Fetches user preferences for filtering recipes. - Recipe: Serves the data about recipes. -

Class Name: RecipeService	
Responsibilities: - Process recipe queries (e.g., by ingredients, dietary restrictions). - Communicate with the database to fetch and filter recipe data. - Return recipe data to the RecipeController after processing.	Collaborators: - Database: Retrieves recipe data from the database. - RecipeController: Communicates with the controller to return filtered data. - User: Uses user preferences for filtering recipes.

Class Name: Database	
Responsibilities: - Store and manage recipe data, user preferences, and other necessary information. - Provide methods for querying recipes based on different parameters (e.g., ingredients, tags).	Collaborators: - RecipeService: Queries the database for recipe data. - User: Stores and retrieves user preferences from the database.

System Architecture Diagram:



1. Model (M)
   - The Model represents the data layer and the business logic of the application. It's responsible for managing the data, retrieving it from the database, and performing any necessary operations on the data (like filtering, searching, or sorting).
   - For your recipe search engine, the Model would include:
     - Recipe: Represents the recipe data (e.g., name, ingredients, instructions, dietary tags).
     - User: Represents the user data (e.g., preferences, search history).
     - Database: Stores recipes, user data, and other persistent information.
   - Responsibilities of the Model:
     - Fetch, store, and update data in the database.
     - Represent business logic for user preferences, recipe searches, and dietary filters.
2. View (V)
   - The View is the user interface (UI) layer, where users interact with the application. It displays the data received from the Controller and updates it based on user actions.
   - For your application, the View would include:
     - React Components: These are the UI components built using React to display recipes, allow users to input dietary preferences, search for recipes, etc.
     - HTML/CSS: The structure and styling of the pages displayed to the user.
     - JavaScript (React/Redux): Handles dynamic behavior on the front end (e.g., updating the UI based on search results, user preferences).
   - Responsibilities of the View:
     - Present data (recipes, search results) to the user in a readable format.

    - Allow users to interact with the system (e.g., entering search terms, selecting dietary restrictions).
    - Update the UI based on the controller's response.

3. Controller (C)
    - The Controller is the intermediary between the Model and the View. It receives input from the View, processes it (using the Model), and updates the View accordingly. The Controller is responsible for handling user input, making decisions based on business rules, and managing the flow of data.
    - For your recipe search engine, the Controller would include:
        - RecipeController: Handles requests related to recipe searches, such as filtering by ingredients, diet, or preferences.
        - UserController: Manages user-related actions, like logging in, updating preferences, and viewing personal recipe recommendations.
        - UserService: Contains the business logic for handling user preferences and user-specific search history.
    - Responsibilities of the Controller:
        - Accept and process user inputs (e.g., search queries, dietary restrictions).
        - Interact with the Model to retrieve or modify data.
        - Send the processed data back to the View to be displayed to the user.
        - Handle the routing and orchestration of requests (in case of a web application).