

Project Report – Recipe App

1. Project description:

- The Recipe Showcase App is a web application designed to provide users with access to a wide variety of recipes.
- The main objective of the project is to create a user-friendly platform where users can discover, search, and explore different recipes based on their preferences.
- The app aims to cater to cooking enthusiasts, home cooks, and anyone looking for culinary inspiration.
- The Recipe Showcase App allows users to browse through a collection of recipes, search for specific recipes by keywords or categories, filter recipes based on dietary preferences or meal types, view detailed information about each recipe including ingredients and instructions, rate recipes based on their experience, save favorite recipes for future reference, and share recipes with others.
- The app targets a broad audience interested in cooking, nutrition, and exploring new dishes.

2. Technologies used, where used in your project and purpose or justification:

Backend Technologies:

- Java: A versatile programming language used for backend development.
- Java Servlets: Java classes that handle HTTP requests and responses in web applications.
- JDBC (Java Database Connectivity): API for connecting Java applications to relational databases.

- Apache Tomcat: An open-source web server and servlet container for deploying Java web applications.

Frontend Technologies:

- HTML (Hypertext Markup Language): Standard markup language for creating web pages.
- CSS (Cascading Style Sheets): Stylesheet language used for styling HTML elements.
- JavaScript: Programming language for adding interactivity and dynamic behavior to web pages.
- JSP (JavaServer Pages): Technology for creating dynamic web pages using Java syntax.

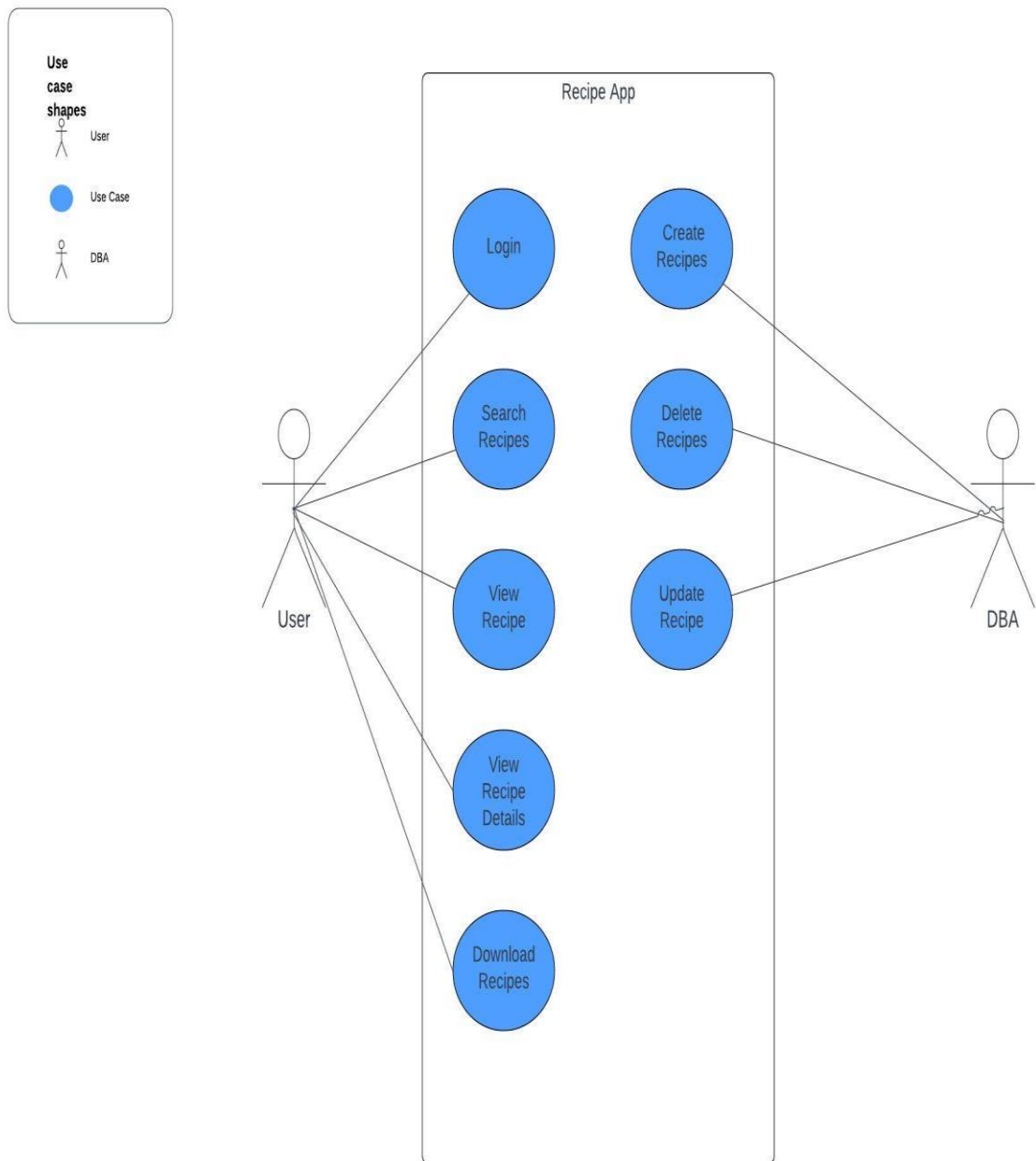
Database Technology:

- MySQL: An open-source relational database management system (RDBMS) used for storing user and recipe data.

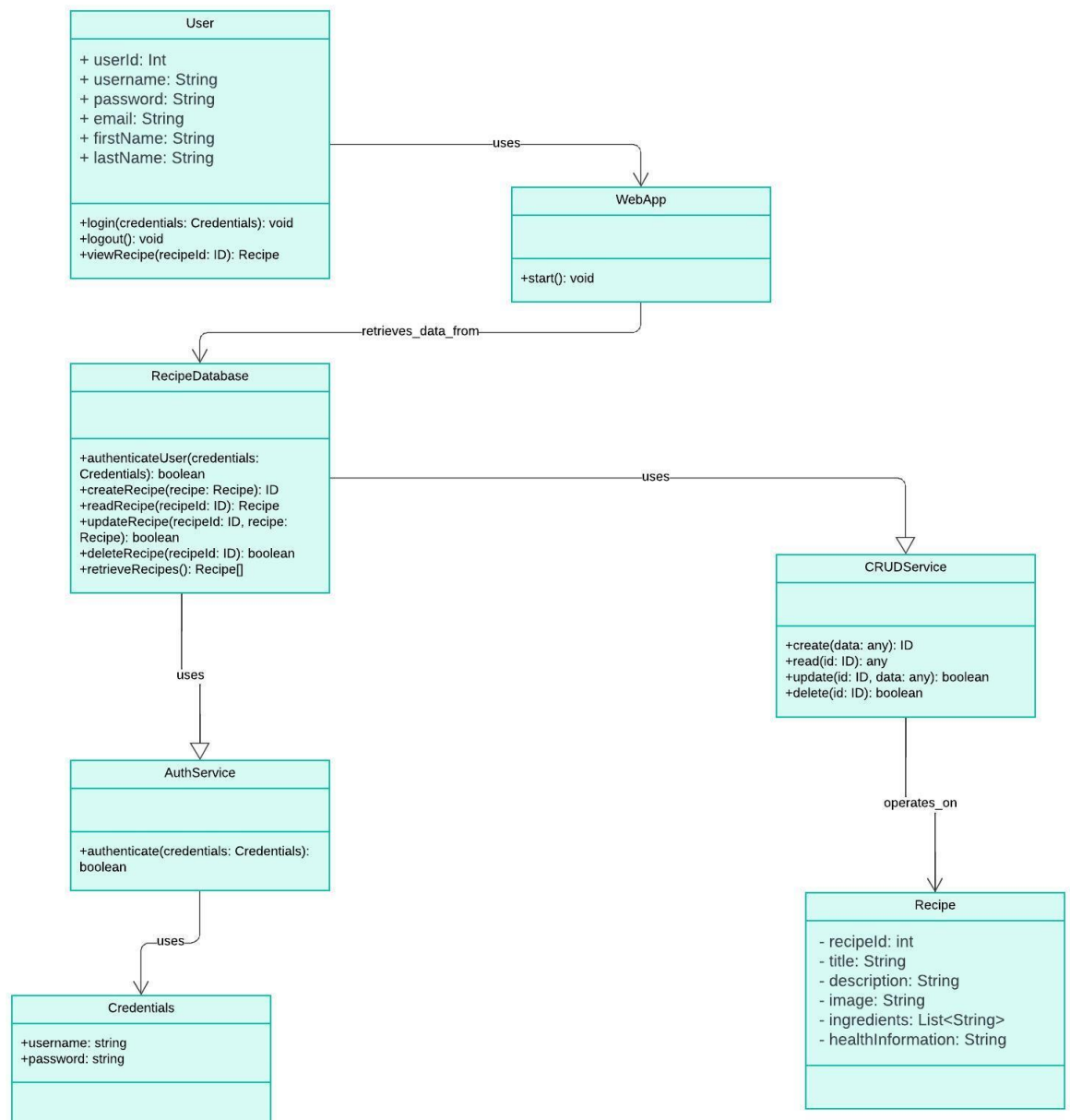
Development Tools:

- Integrated Development Environment (IDE): Software tools such as Eclipse, IntelliJ IDEA, or NetBeans for writing, debugging, and testing Java code.

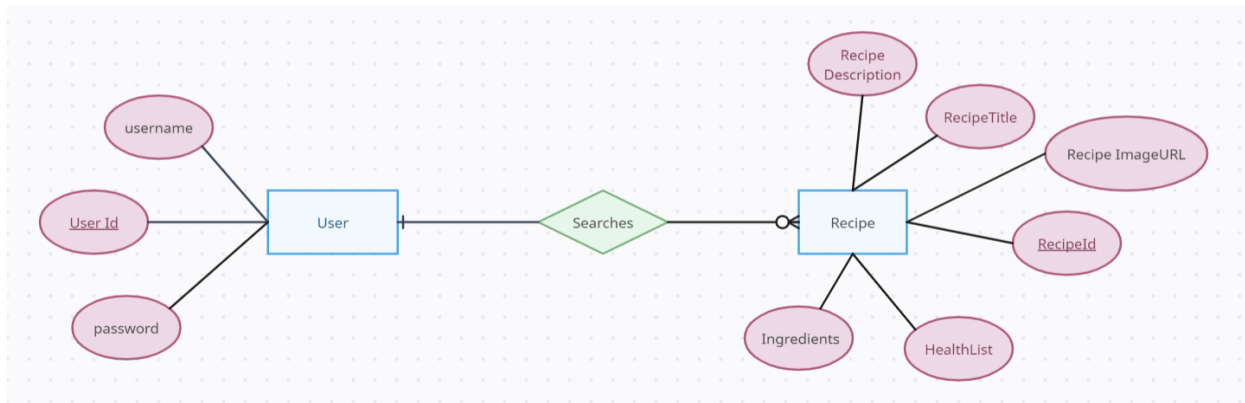
3. Use-case diagram



4. Class diagram



5. ER Diagram:



6. Database design, in case your project includes database:

1. User

userId (Primary Key)

username

password

2. Recipe

recipeId (Primary Key)

title

description

image

ingredients

healthInformation

7. Test use-cases:

Use Case Name: View Recipe Details

Actor: User

Description: This use case describes the scenario where a user views the details of a specific recipe.

Preconditions:

- The user is logged in to the Recipe Showcase App.
- The user has navigated to the recipe details page.

Basic Flow:

- The user selects a recipe from the list of recipes displayed on the app.
- The app retrieves the details of the selected recipe from the database.
- The app displays the title, description, image, ingredients, instructions, rating, category, and health information of the recipe on the recipe details page.
- The user reads the details of the recipe to gain an understanding of its ingredients, preparation method, and nutritional information.
- The user may choose to rate the recipe based on their experience.

Alternative Flow:

- If the selected recipe is not found in the database:
- The app displays an error message indicating that the recipe details could not be found.
- The user is prompted to select another recipe or return to the previous page.

Postconditions:

- The user has successfully viewed the details of the selected recipe.
- The user may choose to rate the recipe or perform other actions based on the information displayed.

Test Cases:

- **Test Case 1:**

Input: User selects a recipe with valid recipe Name.

Expected Output: The app displays the details of the selected recipe, including its title, description, image, ingredients, instructions, rating, category, and health information.

- **Test Case 2:**

Input: User selects a recipe with an invalid recipe Name.

Expected Output: The app displays an error message indicating that the recipe details could not be found.

8. Implementation details:

Backend Development (Java Servlets)

- Create servlet classes to handle different functionalities such as user management, recipe management, authentication, etc.
- Implement methods to handle HTTP requests (GET, POST) and generate appropriate responses.
- Define servlet mappings in the web.xml deployment descriptor file to map URL patterns to servlet classes.
- Specify servlet mappings for each servlet created to ensure proper routing of requests.
- Use JDBC (Java Database Connectivity) to establish a connection to the MySQL database.
- Execute SQL queries to retrieve and manipulate data from the database within servlets.
- Implement session management to maintain user authentication state across multiple HTTP requests.
- Store session attributes such as user credentials, user ID, etc., to identify and authenticate users.

Frontend Development (JSP)

- Create JSP files to generate dynamic web pages that interact with the servlets.
- Embed Java code and expressions within JSP files to dynamically generate HTML content.
- Design user interface components using HTML and CSS to create visually appealing and responsive web pages.
- Include HTML forms, buttons, navigation menus, etc., in JSP files to capture user input and display dynamic content.
- Use JSP tags and expressions to bind dynamic data fetched from servlets to HTML elements.
- Display user information, recipe details, search results, etc., dynamically on web pages.

Database Integration (MySQL):

- Set up a MySQL database to store user and recipe data.
- Define database tables and columns based on the database schema designed for the application.
- Implement connection pooling to efficiently manage database connections and improve application performance.
- Configure database connection properties such as URL, username, password, etc., in a properties file.
- Use JDBC to execute SQL queries and interact with the database from servlets.
- Implement methods to perform CRUD operations on database tables (e.g., insert, update, delete, select).