

Object Oriented Analysis and Design using Java (UE20CS352)

Mini Project

Title: Chef's Portal

Teammates:

Name	SRN
Charvi Bannur	PES1UG20CS638
Chaitra Bhat	PES1UG20CS635
Ankitha C	PES1UG20CS626
Jeffrey S Varghese	PES1UG20CS651

Project Description:

Abstract:

An app that makes cooking a hassle free enjoyable process. Aimed at professional and home chefs with passion for cooking, who would like to store their recipes someplace safely, expand their cooking knowledge. This app will also simplify conversions of measurements for ingredients and make it hassle-free. Users can change the quantity of their recipe's ingredients based on the portion size, flexibly.

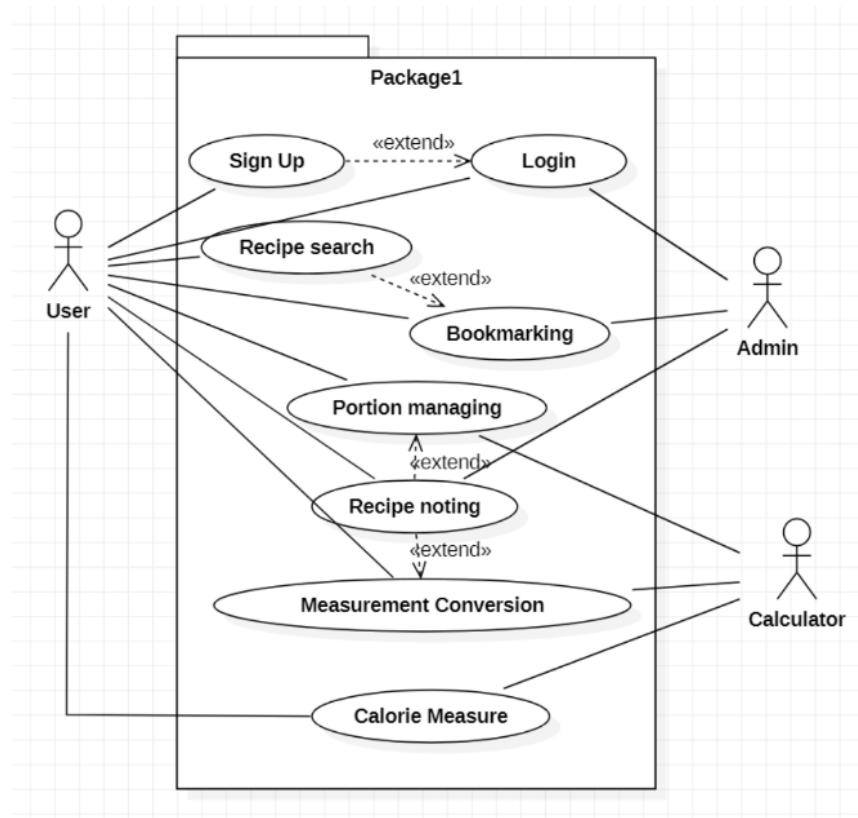
Recipes can be uploaded easily, with pictures and other flairs. A secure personal cookbook feature will be provided.

Functionalities:

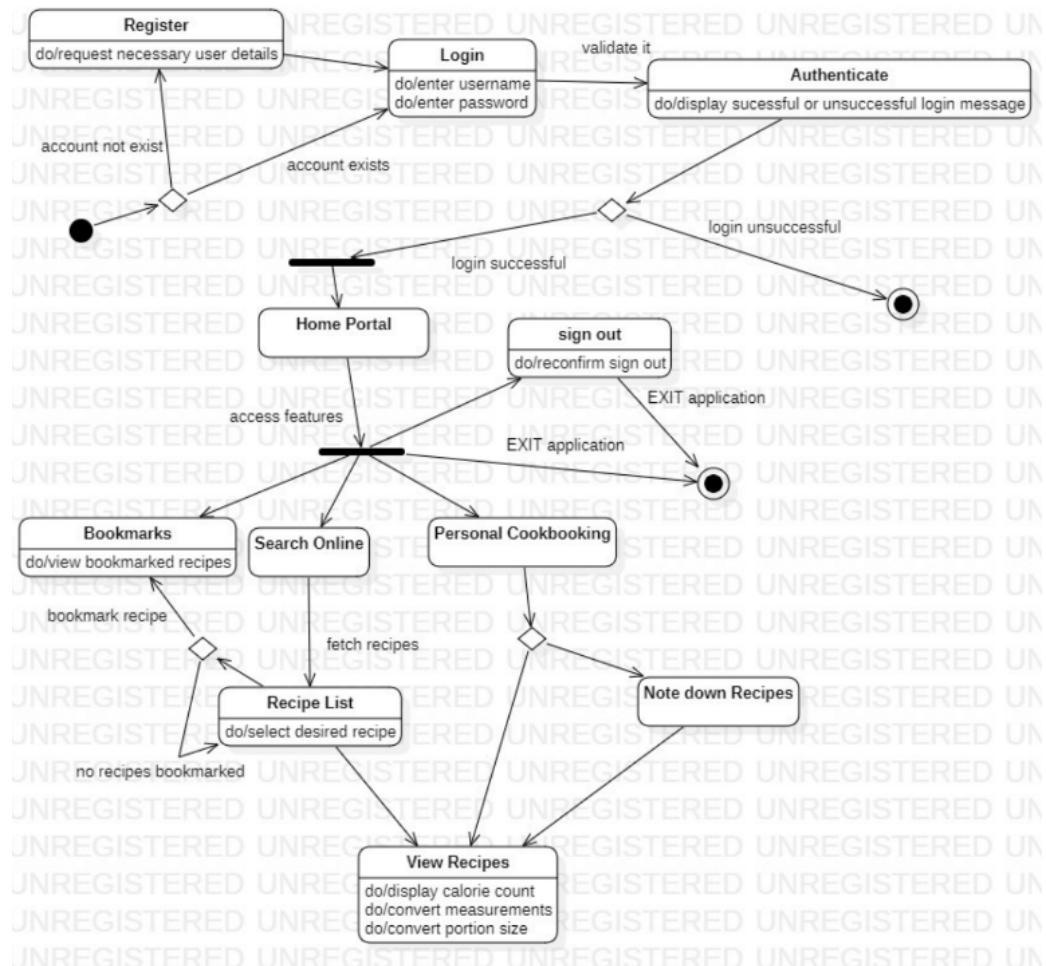
- Conversion of measurements (from cups to gm, gm to pound, etc)
- Changing the quantity of ingredients needed according to the portion size.
- Finding popular recipes, visiting their sites through the app
- Space for users to note down their recipes so they won't forget.
- Bookmarking recipes feature
- A calorie count measurement

UML Diagrams:

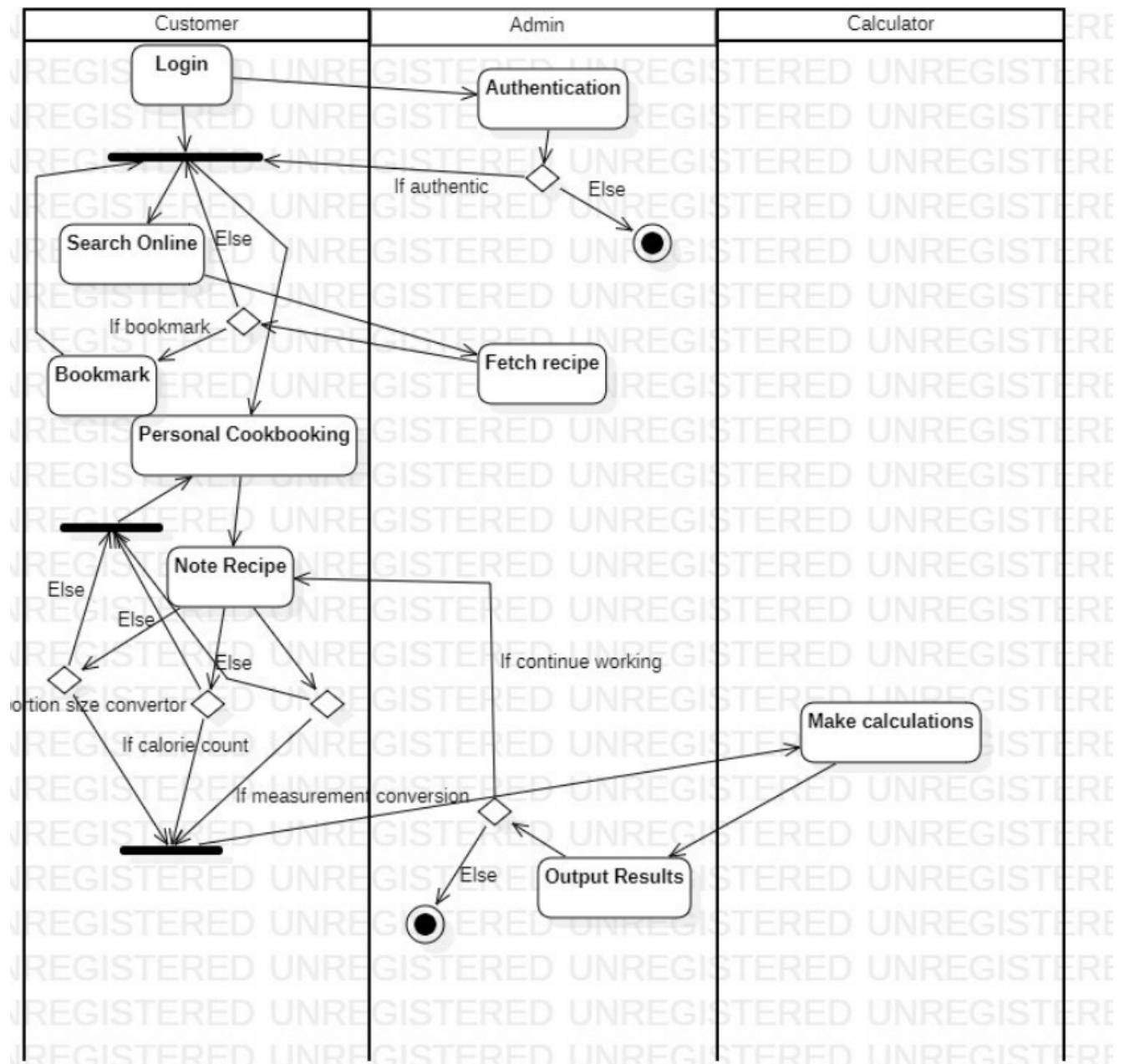
USE Case:



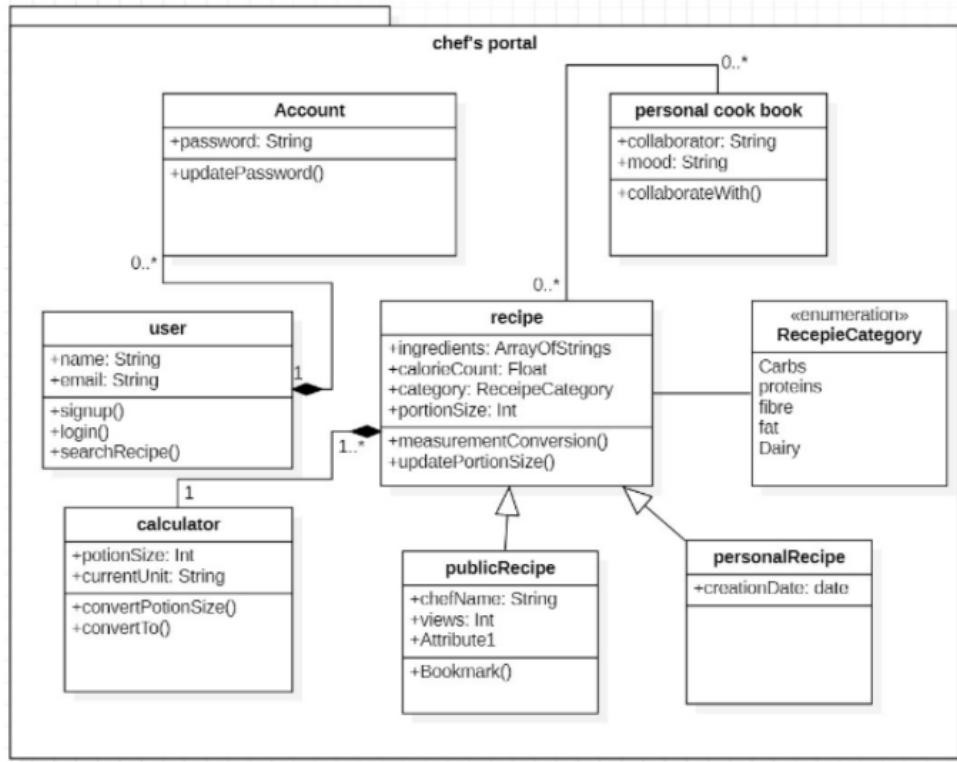
State diagram:



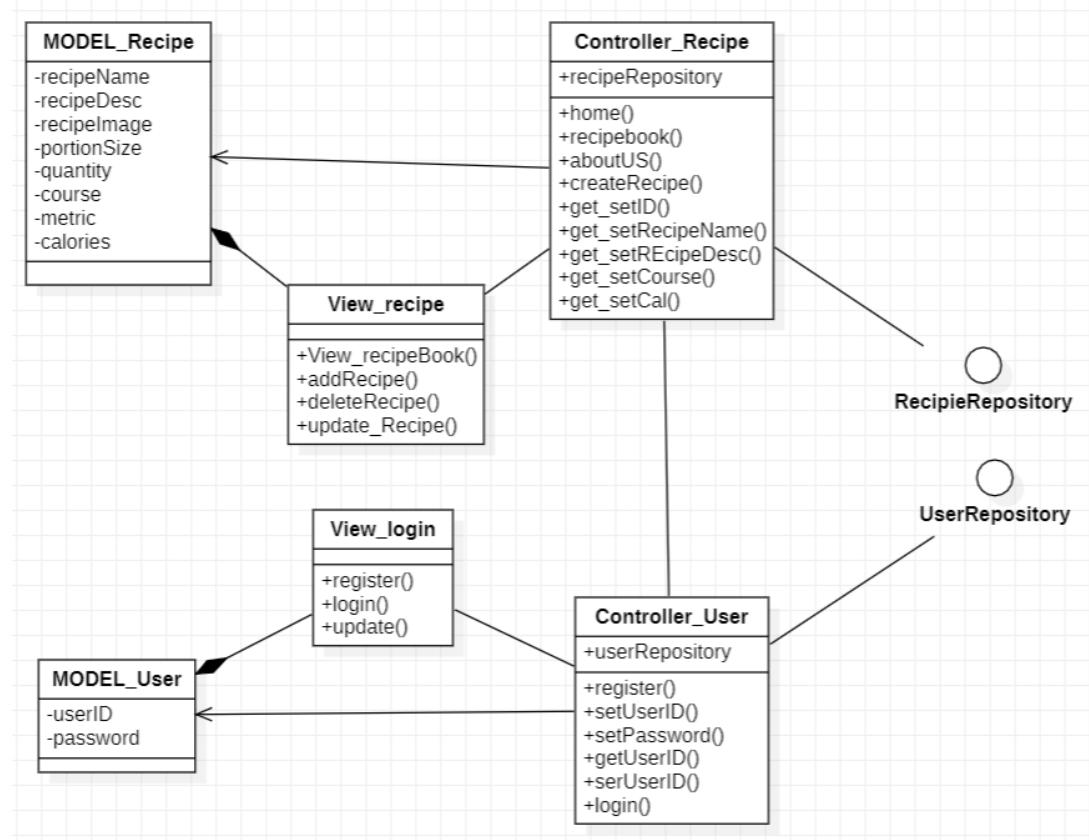
Activity Diagram:



Class Diagram:



MVC Class diagram:



Architecture and Design Patterns:

1. **Model-View-Controller (MVC) Pattern:** This pattern is used to separate the user interface (view) from the application logic (controller) and the data storage (model). In the project, the MVC pattern is applied to divide the system into three interconnected components - the user interface, the business logic, and the data access layer.
2. **Single Responsibility Principle (SRP):** The SRP is a design principle that states that a class or module should have only one reason to change. The project uses this principle to ensure that each class is responsible for only one aspect of the system's functionality.
3. **Open-Closed Principle (OCP):** The OCP is a design principle that states that a class should be open for extension but closed for modification. The project uses this principle to ensure that the system is easily extensible without having to modify existing code.
4. **Observer Pattern:** This pattern is used to define a one-to-many dependency between objects, so that when one object changes state, all its dependents are notified and updated automatically. The project uses this pattern to notify the system's components, such as the recipe book, when a recipe is added or removed from the database.

Github link to the Code base:

<https://github.com/charvibannur/OOAD-Mini-Project>

Individual contributions of the team members:

Name	Contributions
Charvi Bannur	Use Case Diagram, StateDiagram Model: Recipe.java Controller: RecipeController.java View: RecipeRepository
Chaitra Bhat	Use Case Diagram, Activity diagram Model: Userr.java Controller: LoginController.java & RegistrationController.java View: UserRepository
Ankitha C	Use Case Diagram, Class Diagram Model: Recipe.java Controller: RecipeController.java View: RecipeRepository
Jeffrey S Varghese	Use Case Diagram, MVC Class diagram Model: Userr.java Controller: LoginController.java & RegistrationController.java View: UserRepository

Screenshots with input values populated and output shown

Database- before registering user:

The screenshot shows the H2 Database Console interface. The left sidebar lists databases: RECIPE, USERR, INFORMATION_SCHEMA, Sequences, and Users. The main area shows the results of the SQL query `SELECT * FROM USERR;`. The results are displayed in a table with columns `USER_ID` and `PASSWORD`.

USER_ID	PASSWORD
chaitra	123
chaitraa	1234
chaitraaaaaaaaa	1234

(3 rows, 10 ms)

[Edit](#)

SQL statement: `SELECT * FROM USERR;`

Registering user:

The screenshot shows a web-based registration form titled "Register". The form has two input fields: "User ID:" containing "jeffrey" and "Password:" containing "123". A green "Register" button is visible. The background of the form is a photograph of bread slices with toppings like cheese and tomatoes.

Updated Database:

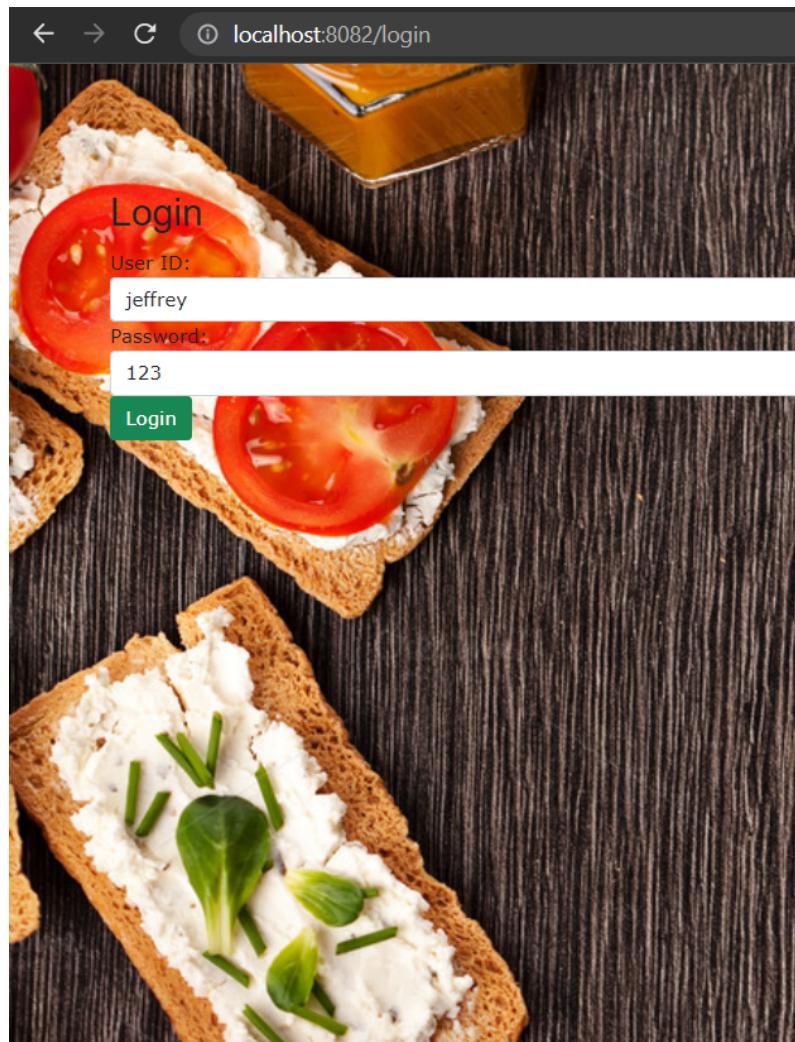
The screenshot shows the H2 Database Console interface. On the left, there's a sidebar with database objects: RECIPE, USERR, INFORMATION_SCHEMA, Sequences, and Users. The main area has a toolbar with 'Auto commit' checked, 'Max rows: 1000', 'Run', 'Run Selected', 'Auto complete', 'Clear', and 'SQL statement:'. Below that is a SQL query window with the command 'SELECT * FROM USERR;'. The results show a table with four rows:

USER_ID	PASSWORD
chaitra	123
chaitraa	1234
chaitraaaaaaaaa	1234
jeffrey	123

(4 rows, 0 ms)

A red arrow points to the last row, 'jeffrey'.

Redirected to login page after Registering user:



Home Page:

The screenshot shows a web browser window for 'localhost:8082/login?userId=jeffrey&password=123'. The title bar has a red arrow pointing to it. The page features a header 'CHEF'S PORTAL' over a dark wood background. Below the header is a section titled 'Trending this week' with five food images. To the left is a large image of a sandwich. On the right, under the heading 'About', is a text block: 'An app that makes cooking a hassle free enjoyable process. Aimed at professional and home chefs with passion for cooking, who would like to store their recipes somewhere safely, expand their cooking knowledge. This app will also simplify conversions of measurements for ingredients and make it hassle-free. Users can change the quantity of their recipe's ingredients based on the portion size, flexibly. Recipes can be uploaded easily, with pictures and other flairs. A secure personal cookbook feature will be provided. Other users' public recipes can be bookmarked for further reference, and be accessed from each user's personal bookmarks section.' Below this is a 'Made with Love by:' section listing names and IDs: Charvi Bannur PESIUG20CS638, Chaitra Bhat PESIUG20CS635, Ankiitha C PESIUG20CS626, and Jeffrey S Varghese PESIUG20CS651.

Empty Recipe Table:

The screenshot shows the H2 console interface at 'localhost:8082/h2-console/login.do?jsessionid=890239d17091ce36b73dcca71b194c86'. The left sidebar shows database objects: RECIPE, USER, INFORMATION_SCHEMA, Sequences, and Users. The main area displays the SQL statement 'SELECT * FROM RECIPE' and its results: '(no rows, 1 ms)'. The results table has columns: ID, CAL, INGLIST, INGPRINT, METRIC, PORTIONSIZE, QUANTLIST, QUANTPRINT, RECIPE_COURSE, RECIPE_DESC, RECIPE_IMAGE, and RECIPE_NAME. There is an 'Edit' button at the bottom of the results table.

Adding Recipe:

The screenshot shows a web application for adding a recipe. The URL is `localhost:8082/create`. The page has a header with links to HOME, ABOUT, and RECIPE BOOK. A blue button labeled "Back to Recipes" is visible. The main form is titled "Add a Recipe". It includes fields for Name (filled with "masala dosa"), Recipe (a list of three steps: 1. Blending the lentil and rice batter, 2. Fermenting Batter, 3. Preparing the savory potato masala filing), Ingredients (Rice, Urad dal, chana dal), Quantity (4), Course (500), and a "Choose a metric" dropdown set to Grams. A portion size of 2 is selected. Below the form is a URL input field containing `https://img5.search.brave.com/HmSgxTw5cNSh_MKNgYDQdK_gn7UcvBUXGrKZoQYzdw/rs:fit:1200:1200:1/g:ce/aHR0cHM6Ly9hcG9s/bG9zdWdhci5jb2`. A green "Save Entry" button is at the bottom.

View Recipe:

The screenshot shows the details of the "MASALA DOSA" recipe. The title is "MASALA DOSA". Below it is a thumbnail image of the dish. There are buttons for "Change Portion" and "Change Metric" (set to Grams). The "Ingredients" section lists "Rice, Urad dal, chana dal". The "Quantity" is 4, and the "Portion" is 2. The "Metric" is gram. The "Recipe" section contains the three steps: 1. Blending the lentil and rice batter, 2. Fermenting Batter, 3. Preparing the savory potato masala filing. The "Course" is 500, and the "Calories" are 0.0. At the bottom are "DELETE" and "EDIT" buttons.

Updated Recipe Table

The screenshot shows a JDBC H2 console window with the following details:

- URL: localhost:8082/h2-console/login.do?sessionid=890239d17091ce36b73dcca71b194c86
- Toolbar buttons: Back, Forward, Home, Refresh, Stop, Auto commit (checked), Max rows: 1000, Run Selected, Auto complete (checked), Off, Auto select (On), Help.
- SQL Statement: SELECT * FROM RECIPE
- Database Structure Tree:
 - RECIPE
 - USER
 - INFORMATION_SCHEMA
 - Sequences
 - UsersH2.2.1.214 (2022-06-13)
- Table Data:

METRIC	PORTIONSIZE	QUANTLIST	QUANTITYINT	RECIPE_COURSE	RECIPE_DESC	RECIPE_IMAGE
gram	2	aced000575200025048009c818922e004202000078700000000140800000	4	500	1. Blending the lentil and rice batter 2. Fermenting Batter 3. Preparing the savory potato masala filling	https://img5.search.brave.com/HmSpxTvs5tNSh_MKNgYDQdk_gn7UcvBUxGrKZoQYzcmwfts1200x1200