# PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

## 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:**

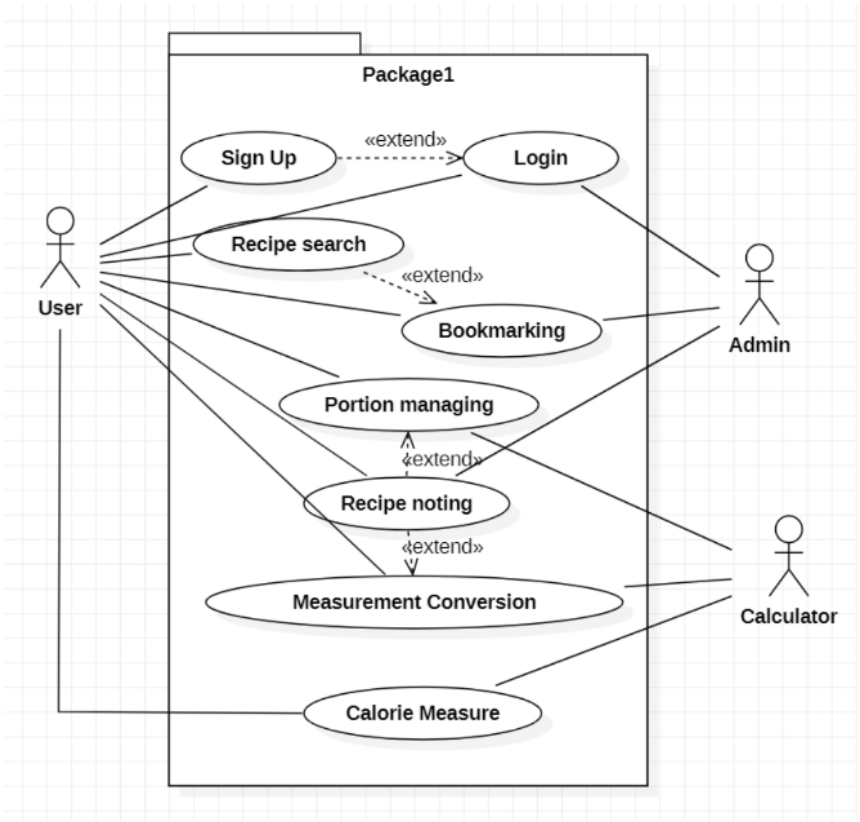
An app that makes cooking a hassle free enjoyable process. Aimed at professional and home chefs with passion for cooking, who would like to store their recipes someplace safely, expand their cooking knowledge. This app will also simplify conversions of measurements for ingredients and make it hassle-free. 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:**

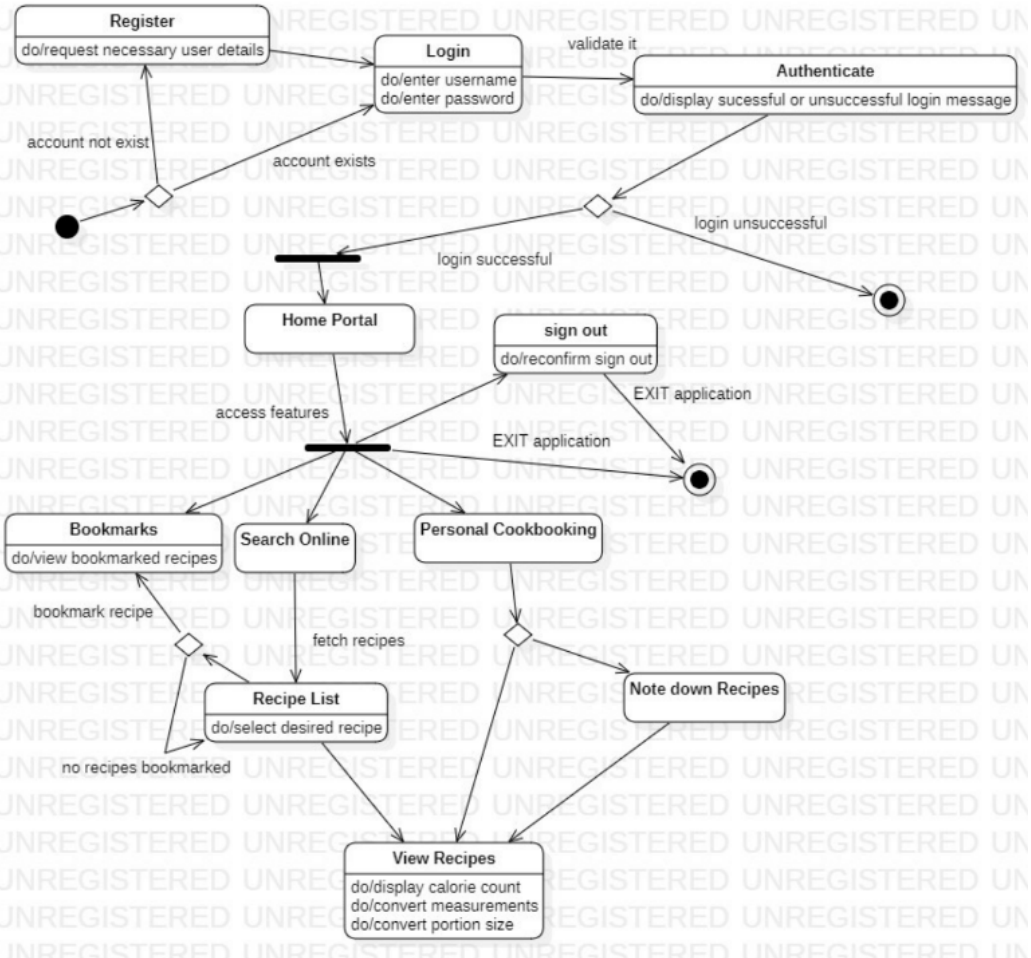
- Conversion of measurements (from cups to gm, gm to pound, etc)
- Changing the quantity of ingredients needed according to the portion size.
- Finding popular recipes, visiting their sites through the app
- Space for users to note down their recipes so they won't forget.
- Bookmarking recipes feature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

**UML Diagrams:**

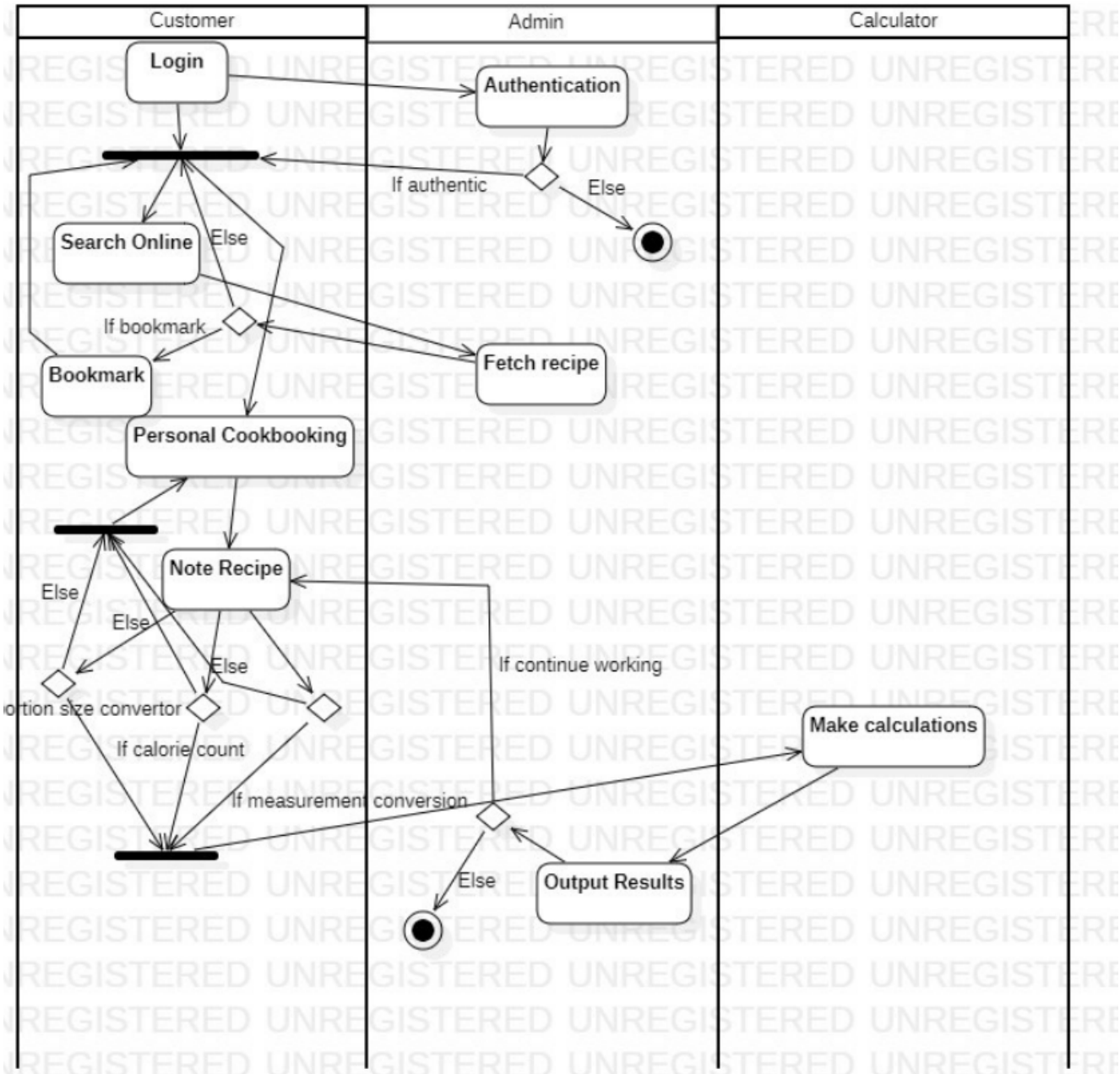
**USE Case:**



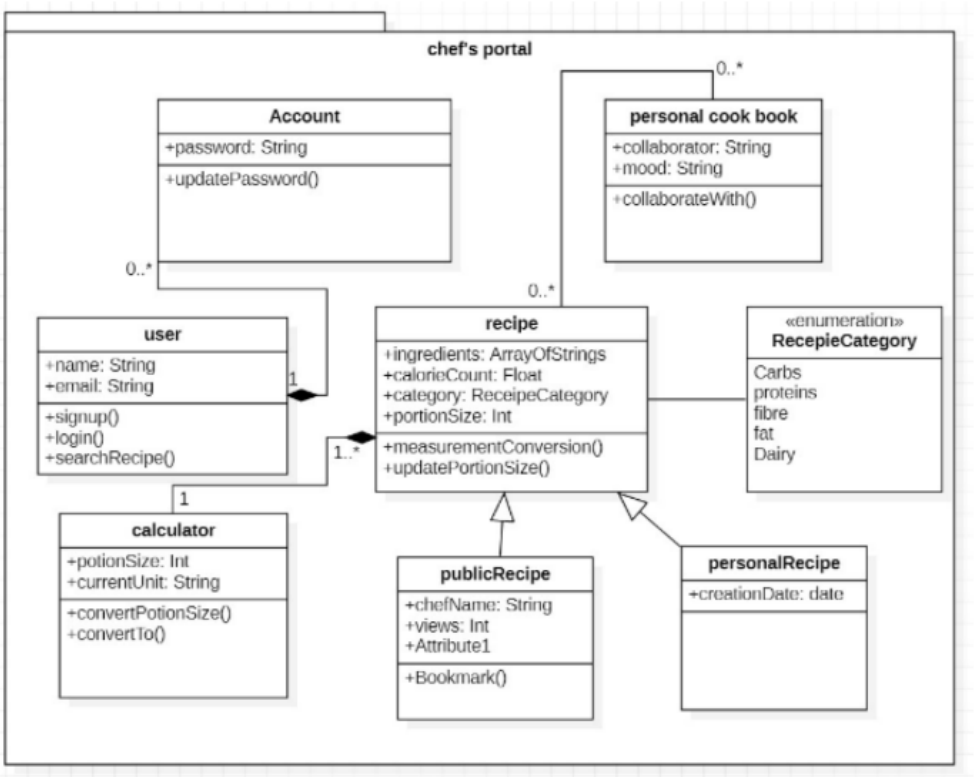

**State diagram:**



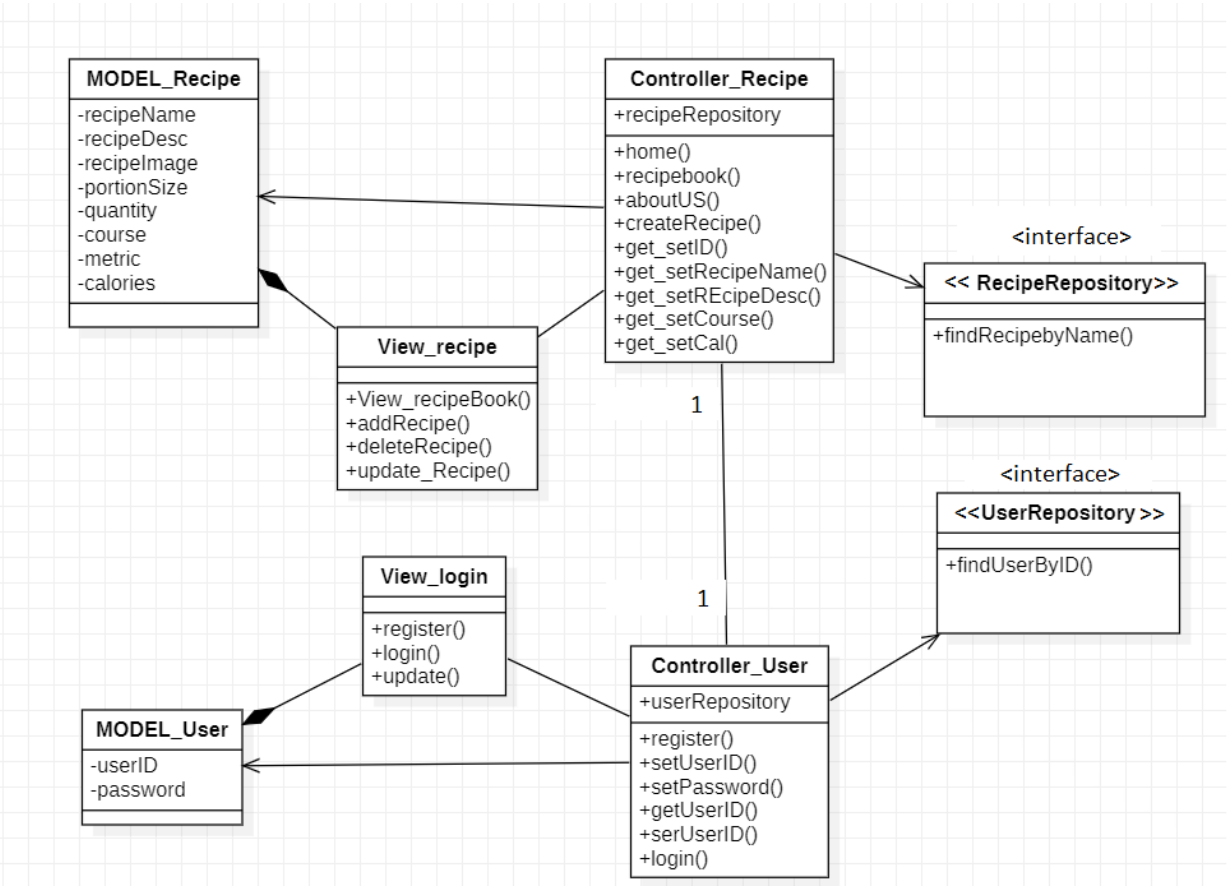
**Activity Diagram:**



**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. **Factory Method(Creational Pattern):** This pattern provides an interface for creating objects, but allows subclasses to decide which class to instantiate. It encapsulates the object creation logic and provides flexibility in object creation. The project uses JpaRepository interface which is an abstract interface that defines methods for performing common database operations such as CRUD (Create, Read, Update, Delete) operations on entities.
5. **Observer Pattern(Behavioural Patterns)**: 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

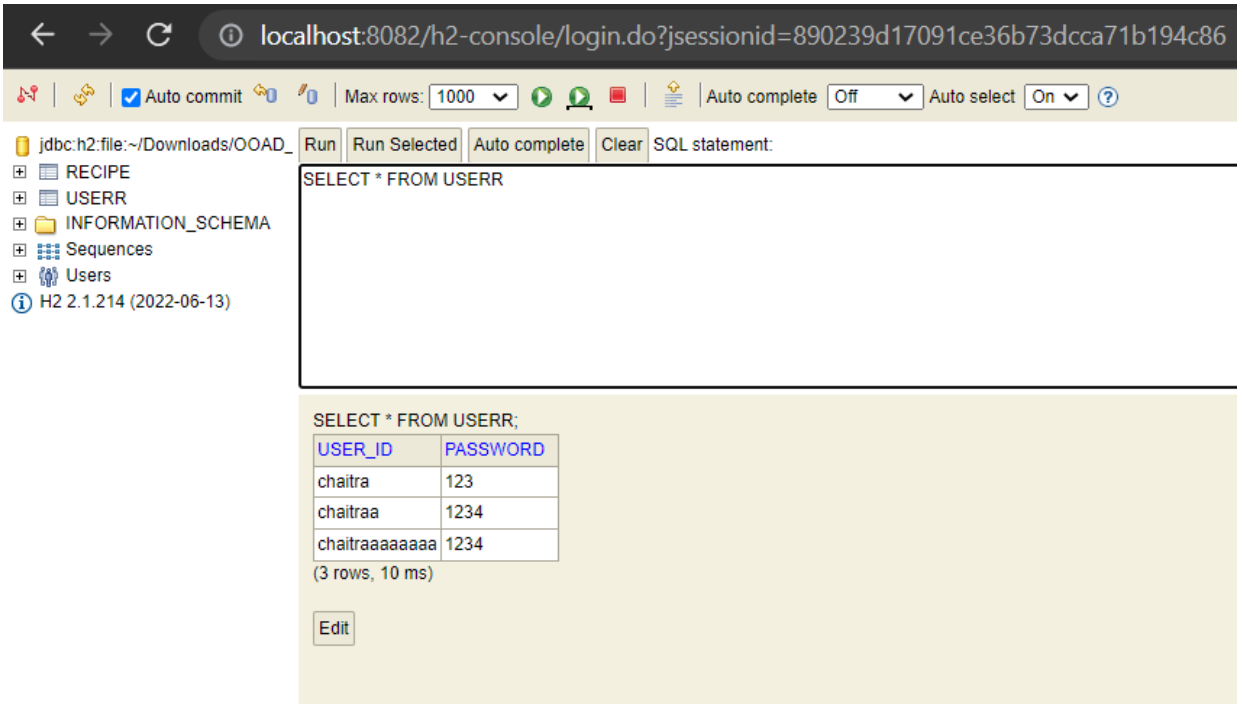
### Individual contributions of the team members:

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

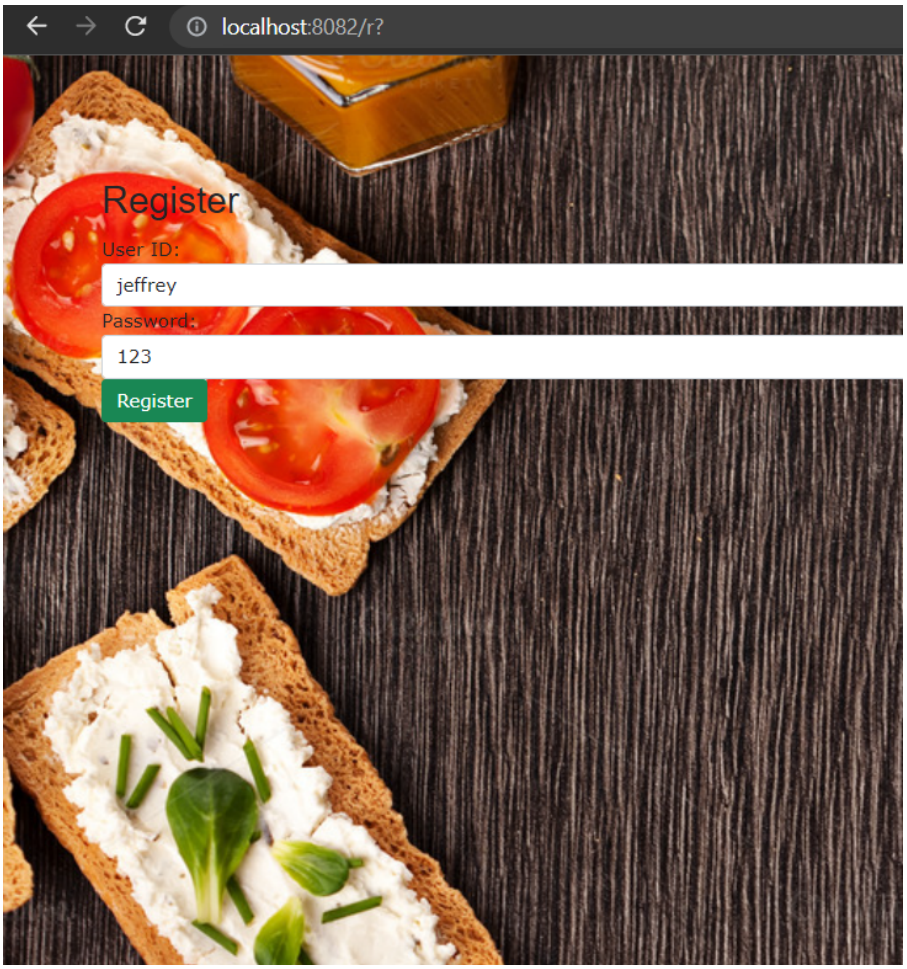
| | |
|---|---|
| | Controller: LoginController.java & RegistrationController.java<br>View: UserRepository |

## Screenshots with input values populated and output shown

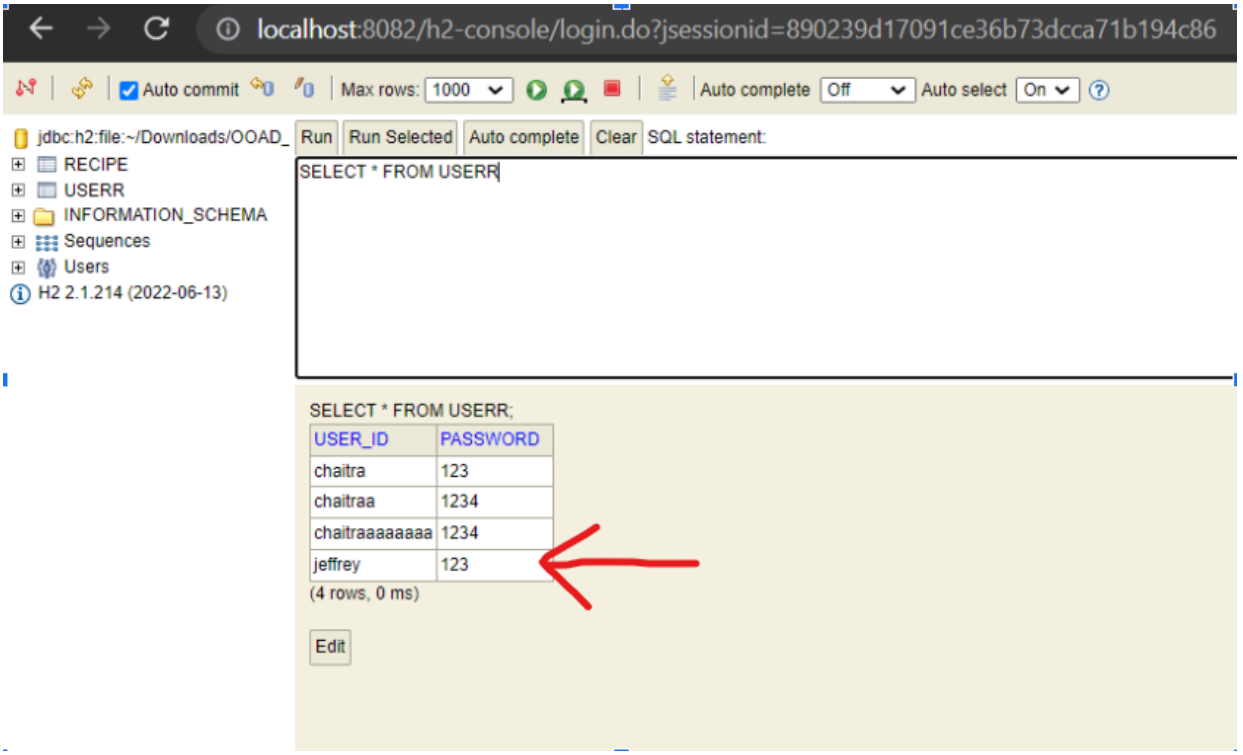
### Database- before registering user:



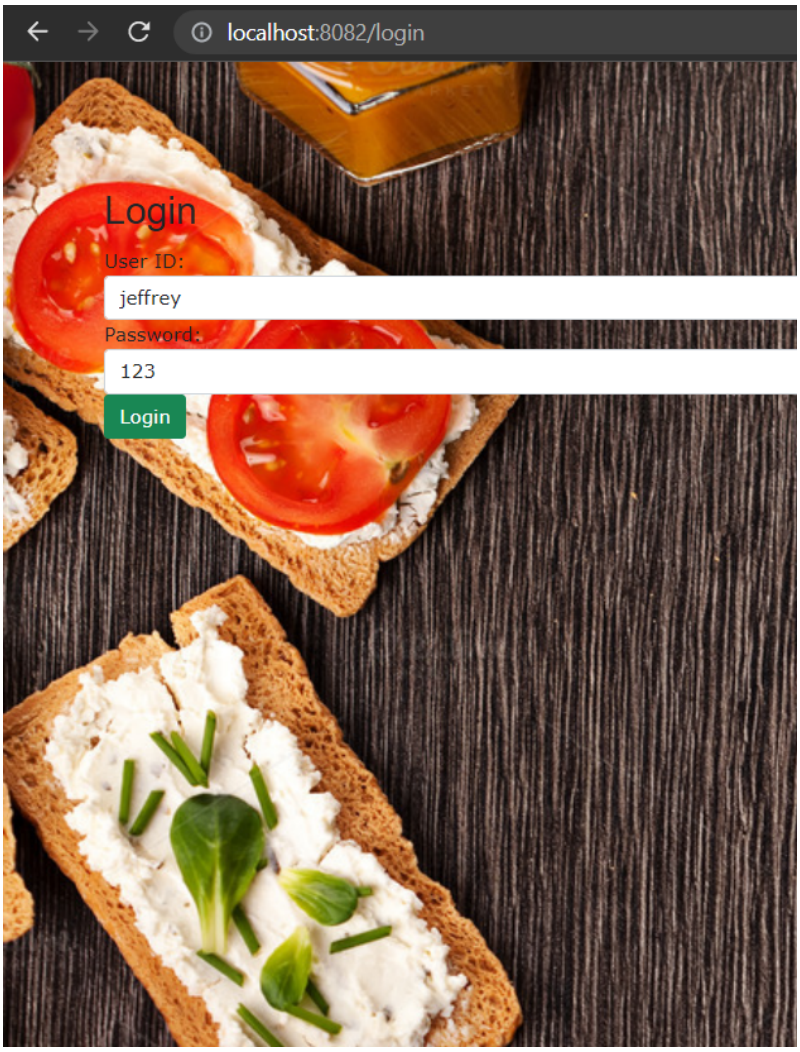
**Registering user:**




**Updated Database:**

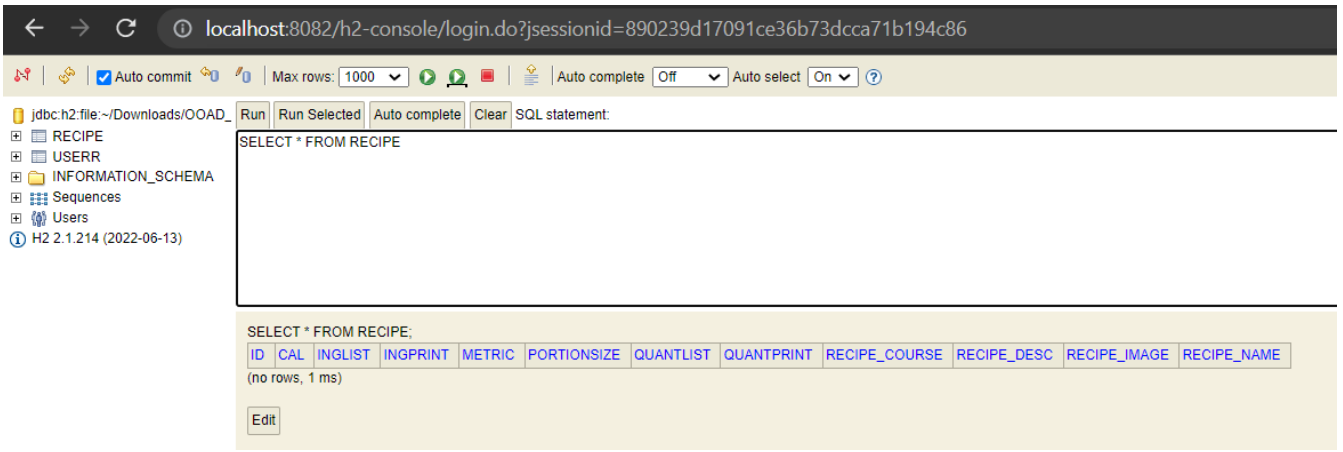


**Redirected to login page after Registering user:**




**Home Page:**

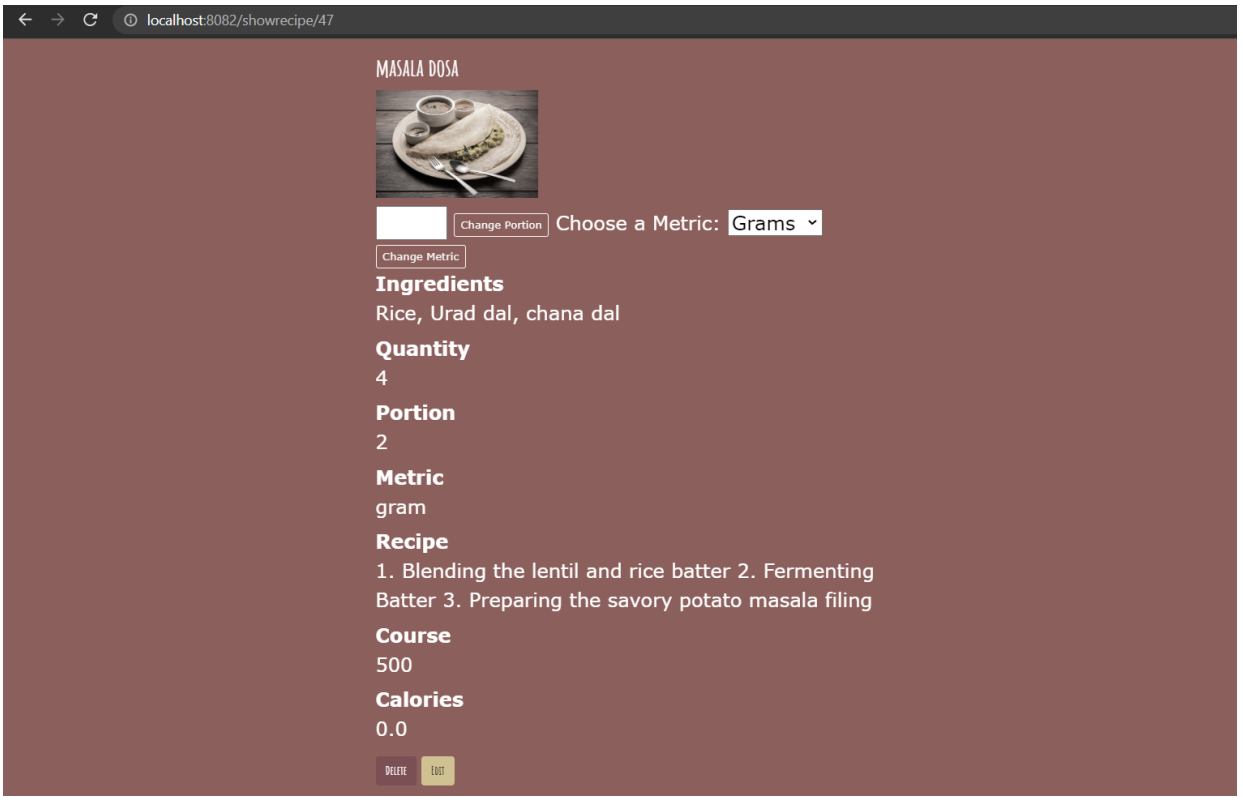
## Empty Recipe Table:




## Adding Recipe:


localhost:8082/create
HOME
ABOUT
RECIPE BOOK
Back to Recipes
Add a Recipe
Name:
masala dosa
Recipe:
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
Choose a metric:
Grams
Portion Size:
2
Provide the food image URL here :
https://imgs.search.brave.com/HmSgxTw5tcNSh_MKNgYDQdK_gn7UcvBUxGrKZoQYzcw/rs:fit:1200:1200:1/g:ce/aHR0cHM6Ly9hcG9s/bG9zdWdhci5jb2
Save Entry

## View Recipe:


localhost:8082/showrecipe/47
MASALA DOSA
Change Portion
Choose a Metric:
Grams
Change Metric
Ingredients
Rice, Urad dal, chana dal
Quantity
4
Portion
2
Metric
gram
Recipe
1. Blending the lentil and rice batter 2. Fermenting Batter 3. Preparing the savory potato masala filing
Course
500
Calories
0.0
DELETE
EDIT

# Updated Recipe Table