CHEFS' IDEA

DISTRIBUTED JAVA PROGRAM FOR FINAL PROJECT
ADVANCED PROGRAMMING
KAROLINA STOGA

Chefs' Idea

VIRTUAL COOKBOOK

The Chefs' Idea app is a virtual cookbook.

The app is aimed at young and old alike who want a quick and convenient access to a variety of recipes. System enables to edit the recipes and store their own recipes.

What's more, the app helps to find recipes based on the products the user has in the kitchen.

The application partly uses recipes from https://www.kaggle.com/datasets/paultimothymooney/recipenlg. This dataset is very large so during this presentation we will be using only 2000 record file.

ARCHITECTURE

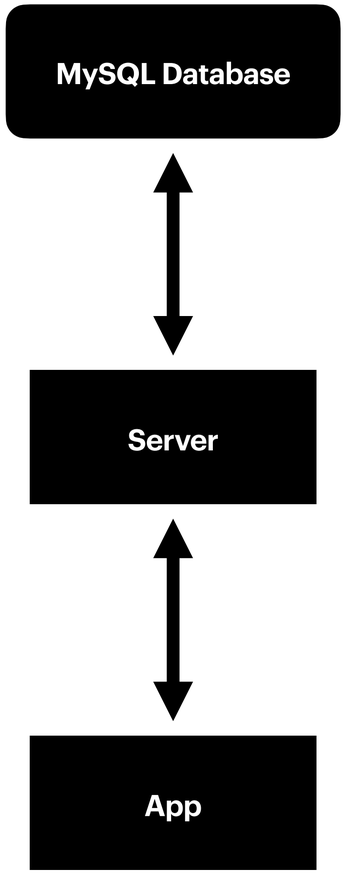
This is distributed Java application and is made up of the following components:

- database
- service
- app

Which communicates with each other via HTTP protocol.

For development I used:

1. Java 19 & Maven 3.8.1 & Spring Boot 3.0.1
2. NetBeans16/Intelij IDE
3. JavaFx 16 & SceneBuilder



INTERESTING TECHNOLOGIES USED

- **Lombok**

It helps to get rid of the boiler-plate code. Using annotations, it is possible to generate both getters and setters, as well as constructors, the hashCode method, equals and much more. It is a very convenient tool that makes code more readable and shorter.

- **JUnit**

A tool for writing tests in Java. It has a wide range of assertions and helps to validate tests.

- **Mockito**

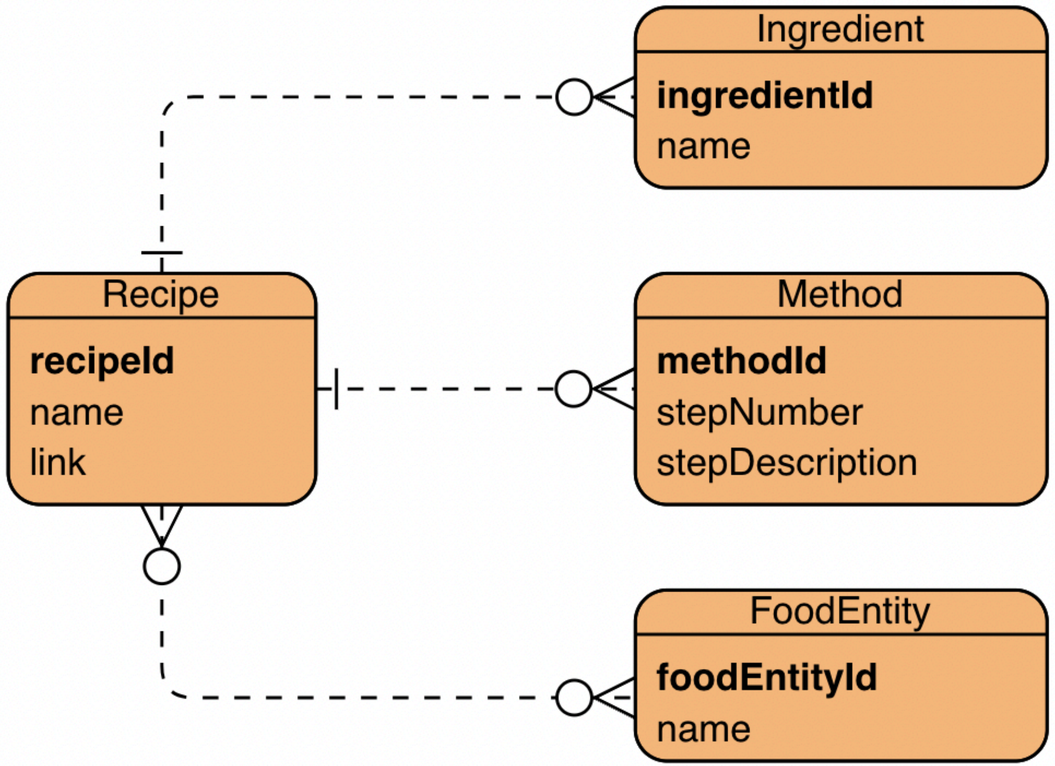
Framework which helps with unit testing. It allows you to mock a given object and its behavior so it is possible to test only a specific service without relying on others.

MYSQL DATABASE

The service uses the Hibernate framework to take advantage of the Object Relational Mapping (ORM).

I created my tables as entity classes in Java where I specified all the columns, their types, constraints and relationships between them.

I also used JPA Repository to easily query data from database.



SERVER

The server was implemented in the Spring Boot framework.

I used a layered architecture there, which divided my code into the following layers:

- presentation layer - this layer is responsible for communication with the frontend, all endpoints were contained in it.
- business logic layer - this layer contains all the application logic, including exception handling
- data layer - this layer is used to communicate with my database

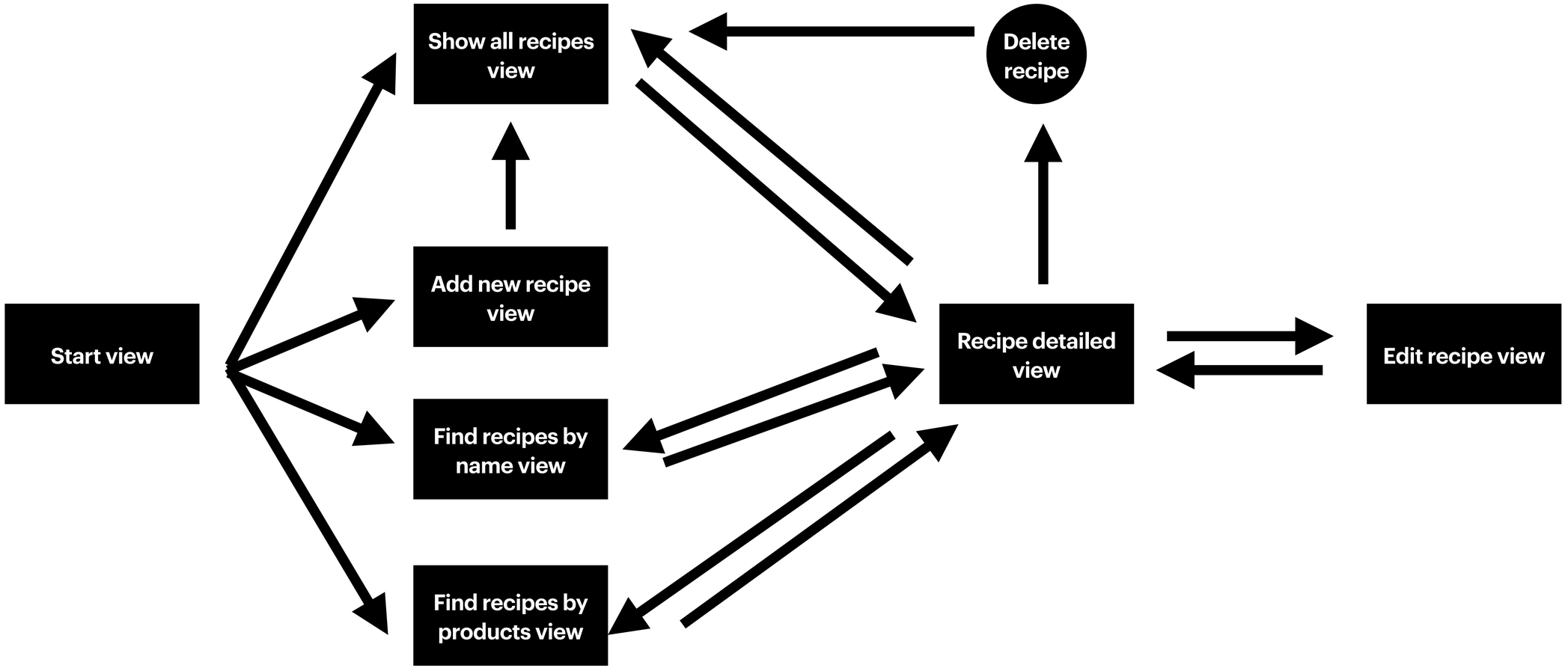
APPLICATION

The application is an user interface that is both responsible for communicating with the server and displaying the data provided by the server to the user.

It communicates with the server using Apache HTTPClient.

The graphic interface is built using JavaFX.

SCENES


Start view
Show all recipes view
Add new recipe view
Find recipes by name view
Find recipes by products view
Delete recipe
Recipe detailed view
Edit recipe view

USE CASES

PRESENTATION OF SYSTEM FUNCTIONS WITH SHORT EXPLANATION

- Initialize database
- Show all recipes
- Add new recipe
- Display detailed view recipe
- Find recipes by products
- Find recipes by name
- Edit recipe
- Delete recipe

USE CASE DIAGRAM