Foody Thought

RECIPE GENERATOR APP

Team

Darshil Dholakia

Nayan Gurung

Shirhan Mohamed

Abdiqani Afrah

William Burdett

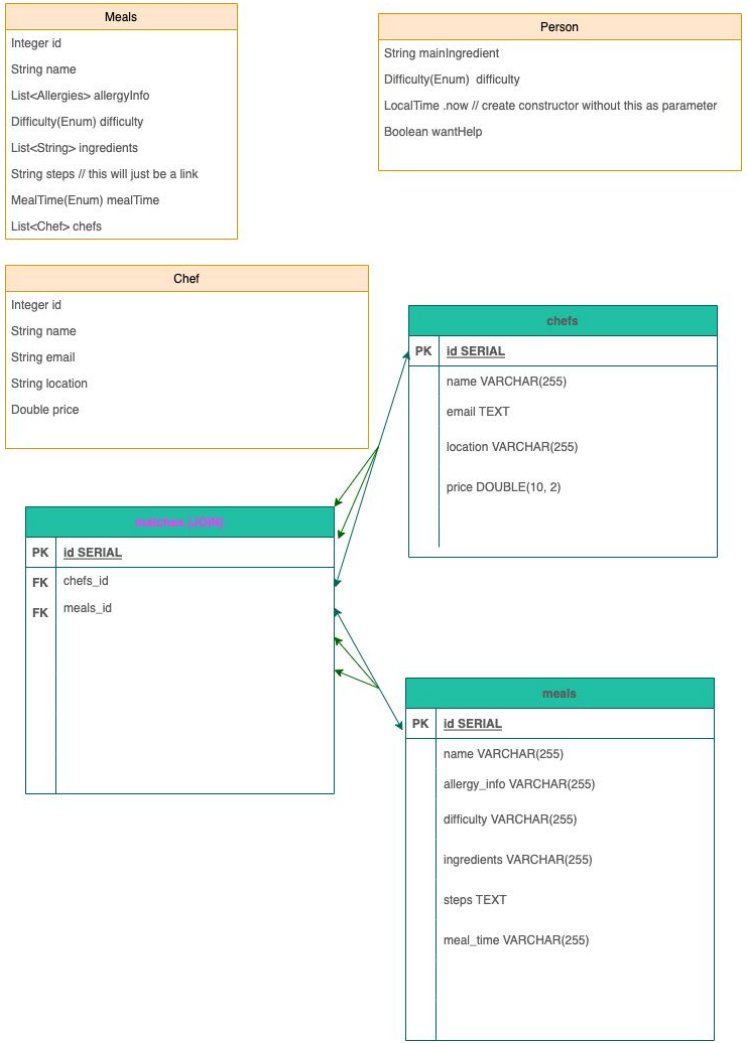
Contents page

Food4Thought
RECIPE GENERATOR APP

What does our app do and why?

- Generates a random recipe using the ingredients a user enters.
- Filters this by the users cooking ability and the time of day.
- Option to find Chefs that offer cooking lessons for particular dishes.
- Helps with cooking inspiration and reduces food wastage.

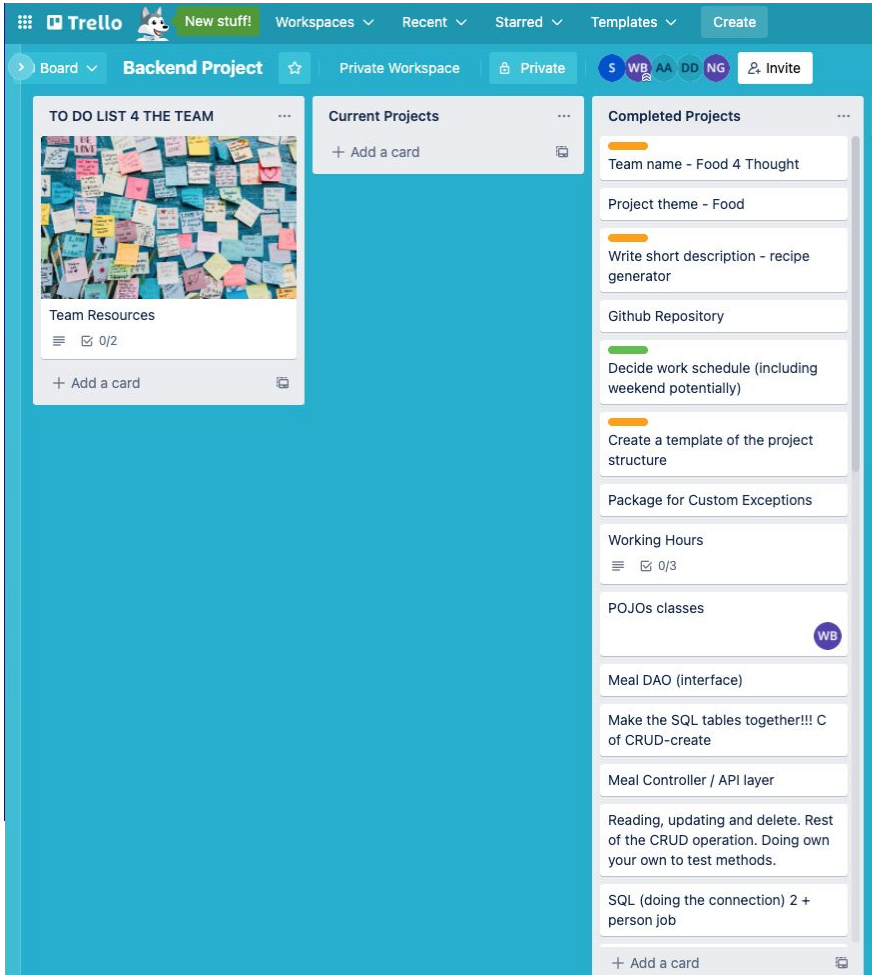
Food4Thought
RECIPE GENERATOR APP



Planning

- Created an Entity Relationship Diagram on app.diagrams.net.
- Helped visualise our POJO's and database relationships.
- Many-to-Many relationship between chefs and meals.

Food4Thought
RECIPE GENERATOR APP



Planning

- Used Trello to list and organise tasks.
- Labelled as MVP (minimum viable product) or Extension.

mockito

spring boot

Java™

H2

PostgreSQL

POSTMAN

Food4Thought

RECIPE GENERATOR APP

Collaboration

- Sprint-based Model (Agile Framework)

- Met in the morning to distribute tasks

- Met in the afternoon to reflect, troubleshoot and plan for the next day

Food4Thought
RECIPE GENERATOR APP

DEMO

Food4Thought
RECIPE GENERATOR APP

Key bits of code - User Requests

Check-list:

- ➔ Ingredient(s)
- ➔ Difficulty
- ➔ Want help
- ➔ Time of meal

```
public Meals selectMealByPerson(Person person) {}
```

```
// creating new person to return LocalTime.now()
        Person request = new Person(person.getMainIngredient(), person.getDifficulty(), person.getWantHelp());
```

Food4Thought

RECIPE GENERATOR APP

Key bits of code - User Functionality

Ingredient(s) ✅

```
String[] personIngredientArr = personIngredients.split(",");
        String ingredientBuilder = "";
        for (int i = 0; i < personIngredientArr.length; i++) {
            if (i + 1 == personIngredientArr.length) {
                ingredientBuilder += "LOWER(ingredients) LIKE '%" + personIngredientArr[i] + "%')";
            } else {
                ingredientBuilder += "LOWER(ingredients) LIKE '%" + personIngredientArr[i] + "%' OR ";
            }
        }
}
```

Food4Thought
RECIPE GENERATOR APP

Key bits of code - User Functionality

Difficulty ✅
Wants Help ✅

```
String personDifficulty = String.valueOf(request.getDifficulty());
Boolean personWantsHelp = request.getWantHelp();
```

Meal time ✅

```
// determining the meal_time listed based on the time
        String personMealtime;
        if (request.getLocalTime().getHour() < 11){
            personMealtime = "'BREAKFAST'";
        } else {
            personMealtime = "'SNACK') OR LOWER(meal_time) = LOWER('MAIN'";
        }
```

Food4Thought
RECIPE GENERATOR APP

Key bits of code - User Functionality

Check-list:

- ➔ Ingredient(s) ✅
- ➔ Difficulty ✅
- ➔ Want help ✅
- ➔ Time of meal ✅

```
String sql = "SELECT id, name, allergy_info, difficulty, ingredients, steps, meal_time FROM meals WHERE (" +
ingredientBuilder + " AND (LOWER(meal_time) = LOWER(" + personMealtime + ")) AND LOWER(difficulty) = LOWER('" +
personDifficulty + "')";
        Meals meal = mealDAO.selectMealByPerson(sql, personWantsHelp);
        return meal;
```

Food4Thought
RECIPE GENERATOR APP

Key bits of code –Email Validator in Chef Utilities Class

```
public class ChefUtilities {

        public static boolean validateEmail(String email) {
            String regex = "[\\w-]{1,20}@\\w{2,20}\\.\\w{2,3}$";
            Pattern pattern = Pattern.compile(regex);
            Matcher matcher = pattern.matcher(email);
            if (!matcher.matches()) {
                throw new EmailInvalidException("Please re-enter your email again.");
            }
            return true;
        }
    }
```

Food4Thought
RECIPE GENERATOR APP

Key bits of code –Email Validator in Chef Service Class

```
 public void insertChef (Chef chefs){
        int rowsChanged = 0;
        if (checkIfEmailIsUnique(chefs.getEmail()) && ChefUtilities.validateEmail(chefs.getEmail()) &&
ChefUtilities.validatePrice(chefs.getPrice())) {
            rowsChanged = chefDAO.insertChef(chefs);
        }
        if (rowsChanged != 1) {
            throw new RowNotChangedException("Chef " + chefs.getName() + " not added.");
        }
    }
```

Food4Thought
RECIPE GENERATOR APP

Key bits of code – Chef Service Class

➔ Price validator feature

Chef utilities class

```
public static boolean validatePrice(Double price) {
            if (price <= 0) {
                throw new PriceInvalidException("Please enter a valid price.");
            }
            return true;
        }
```

Food4Thought
RECIPE GENERATOR APP

Key bits of code – Chef Service Class

- ➔ Issues encountered
- ➔ How the issues were resolved

```
    public void updateChefsById (Integer chefId, Chef update){

        int rowsChanged = 0;
        if (chefDAO.selectChefById(chefId) == null ) {
            throw new ChefNotFoundException("Chef with id " + chefId + " could not be found.");
        } else if (chefDAO.selectChefById(chefId) != null &&
ChefUtilities.validatePrice(update.getPrice()) && ChefUtilities.validateEmail(update.getEmail())) {
            rowsChanged = chefDAO.updateChefsById(chefId, update);
        }
        if (rowsChanged != 1) {

            throw new RowNotChangedException("Chef with id " + chefId + " was not updated.");
        }
```

Food4Thought
RECIPE GENERATOR APP

Key bits of code- Meal Service

Jackson Serialization Annotations

```
@JsonCreator
   public static Allergies fromString(String type) {
       for (Allergies value : Allergies.values()) {
           if (value.value2.equals(type.toUpperCase())) {
               return value;
           } else if(value.value1.equals(type.toUpperCase()){
               return value;}
       }return null;}
@JsonValue
  public String getAllergies() {
      return this.value1.toUpperCase();}
```

```
public void updateById(Integer mealId, Meals update) {
       try{
           if(!isStepsValid(update.getSteps())) {
               throw new LinkInvalidException("Invalid link.Check again!");
           } else if (mealDAO.selectMealById(mealId)!=null){
               mealDAO.updateMeals(mealId,update);
           } else if(mealDAO.updateMeals(mealId,update)!=1){
               throw new RowNotChangedException("Meal with id " + mealId + " was not updated");
           }
       } catch(EmptyResultDataAccessException e) {
           throw new MealNotFoundException("Meal with id number "+ mealId + " does not exist");
       }
   }
```

Example of custom exceptions:
→ LinkInvalidException,
→ RowNotChangedException,
→ MealNotFoundException

Food4Thought
RECIPE GENERATOR APP

Testing 1

Application.properties

```
spring.datasource.url=jdbc:h2:mem:testdb;DB_CLOSE_DELAY=-1
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=password
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
spring.h2.console.enabled=true
spring.jpa.hibernate.ddl-auto=create-drop
spring.jpa.defer-datasource-initialization=true
spring.profiles.active=test
```

Schema.sql

```
INSERT INTO chefs(name,email,location,price)
VALUES('Sue Lopez', 'suelopez@gotmail.com','Grimsby', 75.00);
```

Testing

```
@Test
    void canSelectById(){
        Chef expected = new Chef(1,"Sue Lopez", "suelopez@gotmail.com","Grimsby", 75.00);
        Chef actual = underTest.selectChefById(1);
        assertThat(actual).isEqualTo(expected);
    }
```

Food4Thought
RECIPE GENERATOR APP

Key bits of code - Mocking

```
@Test
  void selectMealByPerson() {
      Person testPerson = new Person("kielbasa", Difficulty.BEGINNER, false);
      Meals expected = new Meals(2, "Pasta Salad", Difficulty.BEGINNER, List.of(Allergies.DAIRY),
      List.of("Kielbasa", "Noodles"), "https://www.inspiredtaste.net/38019/easy-pasta-salad-
      recipe/#itr-recipe-38019", MealTime.SNACK, null);
      given(fakeMealDao.selectMealByPerson(anyString(), anyBoolean())).willReturn(expected);

      Meals actual = underTest.selectMealByPerson(testPerson);

      assertThat(actual).isEqualTo(expected);
  }
```

Food4Thought
RECIPE GENERATOR APP

Key bits of code - Testing with H2

```
DataSource dataSource = new EmbeddedDatabaseBuilder().setType(EmbeddedDatabaseType.H2)
                .addScript("classpath:jdbc/meals-schema.sql")
                .addScript("classpath:jdbc/meals-test-data.sql")
                .build();
```

```
DROP TABLE IF EXISTS meals;
CREATE TABLE meals (
    id SERIAL PRIMARY KEY,
    name character varying(255) NOT NULL,
    allergy_info character varying(255),
    difficulty character varying(255) NOT NULL,
    ingredients character varying(255) NOT NULL,
    steps text NOT NULL,
    meal_time character varying(255) NOT NULL
);
```

meals-schema.sql

```
INSERT INTO meals VALUES (1, 'Test Meal 1', null, 'Intermediate', 'Chicken',
'https://www.testlink1.com', 'Main');

INSERT INTO meals VALUES (2, 'Test Meal 2', 'Dairy', 'Beginner', 'Kielbasa,Noodles',
'https://www.testlink2.com', 'Snack');
```

meals-test-data.sql

Food4Thought

RECIPE GENERATOR APP

What we are proud of!

- Our planning and entity relationship (ERD) diagram

- Comprehensive testing and mocking for different layers

- IntelliJ code formatting and use of package structure

Food4Thought
RECIPE GENERATOR APP

Reflections

Key learnings:

- Collaborating on GitHub
- Spring Starter Security
- Working with H2 database

Future improvements:

- Add a feature to allow users to save the meals they received
- Configure security to a database

Food4Thought

RECIPE GENERATOR APP

Thank you!

Link to GitHub - https://github.com/WillBurdett/Food4Thought

Food4Thought
RECIPE GENERATOR APP