



Recipes system: Tasty Tales



Students name	ID
Shatha Ahmed Alanazi - 2	2220006290
Lojain Hafez Alibrahim - 1	2220001991
Zainab Adnan Alaujami - 2	2220000593
Rahaf Adnan Alshammari - 2	2220000977
Kawthar Marzooq - 2	2220000797
Ola Alawad - 3	2220002030
Raghad Ali Alzahrani - 1	2220002910

Instructors:

L. Reem Alshammari - L. Nourah Alqahtani - L. Raja Alyafer

Academic year 1445 – 2023/2024



Introduction:

Food is a way to communicate and share cultures and art .There is unlimited creativity in mixing flavors.

Our program is designed to allow users to review the most famous recipes and share their creativity, culture, or even their way of preparing popular dishes.

Whereas, when the user requests to add a dish, the request will pass through the admin to decide whether to publish it to the public or not.

Goals:

- Cultural Exploration: Promote cultural diversity by including recipes from around the world, encouraging users to explore and appreciate different culinary traditions.
- User-Friendly Interface: Create an intuitive and user-friendly interface that accommodates both novice and experienced cooks, ensuring a seamless and enjoyable experience.
- Detailed Instructions: Provide step-by-step instructions, to empower users with various skill levels to successfully recreate recipes with confidence.
- Nutritional Awareness: Offer nutritional information for each recipe, enabling users to make informed choices aligned with their health and wellness goals.
- Feedback and Improvement: Implement a feedback system to gather user input and continually improve the app's functionality and content.
- Scalability: Design the project with scalability in mind, allowing for future expansion, updates, and the addition of new features to keep users engaged and satisfied.

Tools:

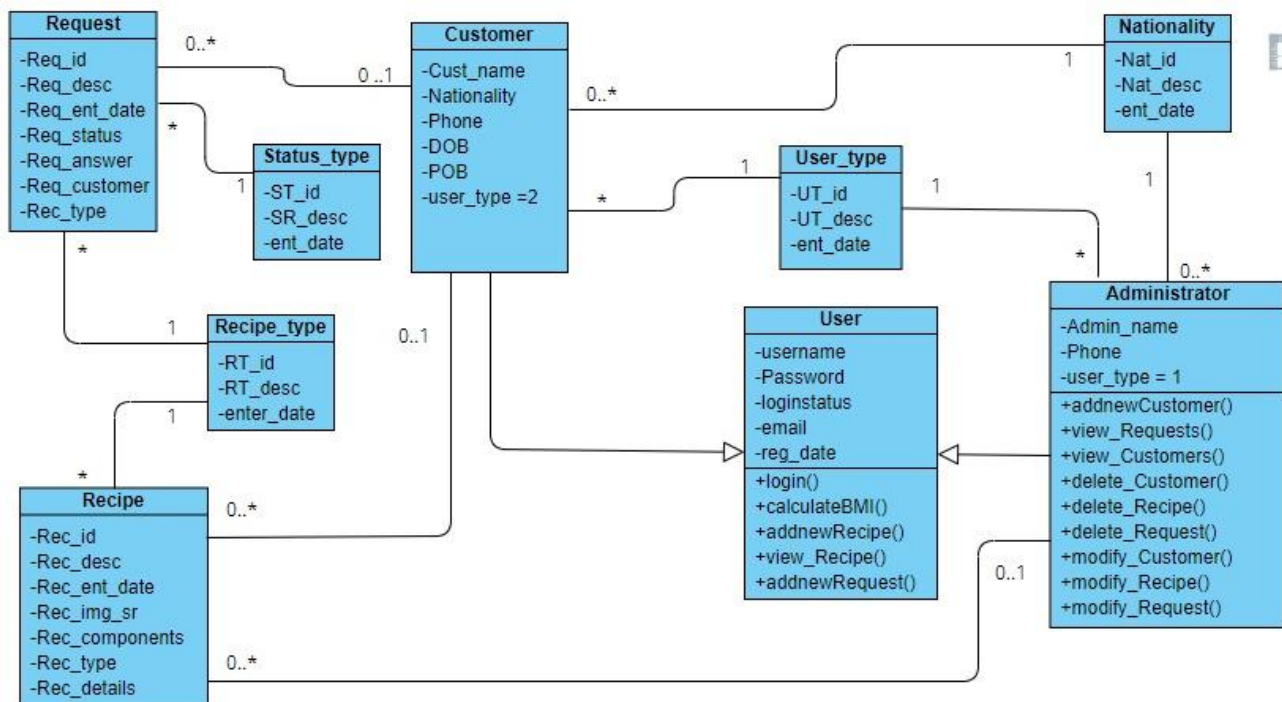
NetBeans & Visual Paradigm



Systems Users (Actors):

	User	Privileges
1	Admin	he can view and edit both customer and recipes and also accepts or rejects requests from customer
2	User	They can view recipes and add new ones by sending add requests or asking us to provide a specific recipe. Ability to use a weight calculator
3	visitor	He can view the list of recipes

UML:





Interfaces & Outputs:

```
run:
ArrayLists are Declared :
Choose one of the User type :
1: Administrator
2: Customer
3: Claculate BMI
4: Visitor
1
Administrator
you have full access
Choose one of the List :
1: Login
2: Register
3: Customer Methods
4: Recipe Methods
5: Request Methods
6: Nationality Methods
1
User Name :
Admin
Password :
admin
Login failed for user: Admin
Enter 99 For Exit Or Enter 1 to Continue
1
Choose one of the User type :
1: Administrator
2: Customer
3: Claculate BMI
4: Visitor
1
Administrator
you have full access
Choose one of the List :
1: Login
```

```
Choose one of the List :
1: Login
2: Register
3: Add new Recipe
4: Add new Request
5: view Recipes
1
Enter username
Cust
Enter Password
Pass
Login failed for user: Cust
Enter 99 For Exit Or Enter 1 to Continue
1
Choose one of the User type :
1: Administrator
2: Customer
3: Claculate BMI
4: Visitor
2
Customer
Choose one of the List :
1: Login
2: Register
3: Add new Recipe
4: Add new Request
5: view Recipes
1
Enter username
Cust
Enter Password
123
Login successful for user: Cust
Enter 99 For Exit Or Enter 1 to Continue
```

In this output, the program allows users to interact with a database using an ArrayList. It has features like Login, Register, Customer Methods, Recipe Methods, Request Methods, and Nationality Methods. When started, the program displays a menu of options. In this scenario, the user chooses to be an Administrator. They enter the username "Admin" and the password "admin" for login, but the password is incorrect. The program shows an error message: "Login failed for user: Admin." The program returns to the main menu, and the user is asked to choose a user type again. They select Administrator, and the program presents the same list of operations. This cycle repeats until the user enters 99 to exit the program.

To log in successfully, the correct password must be entered.



```

run:
ArrayLists are Declared :
Choose one of the User type :
1: Administrator
2: Customer
3: Claculate BMI
4: Visitor
2
Customer
Choose one of the List :
1: Login
2: Register
3: Add new Recipe
4: Add new Request
5: view Recipes
1
Enter username
Cust
Enter Password
Pass
Login failed for user: Cust
Enter 99 For Exit Or Enter 1 to Continue
1
Choose one of the User type :
1: Administrator
2: Customer
3: Claculate BMI
4: Visitor
2
Customer
Choose one of the List :
1: Login
2: Register
3: Add new Recipe
4: Add new Request

```

```

3: Customer Methods
4: Recipe Methods
5: Request Methods
6: Nationality Methods
1
User Name :
Admin
Password :
admin
Login failed for user: Admin
Enter 99 For Exit Or Enter 1 to Continue
1
Choose one of the User type :
1: Administrator
2: Customer
3: Claculate BMI
4: Visitor
1
Administrator
you have full access
Choose one of the List :
1: Login
2: Register
3: Customer Methods
4: Recipe Methods
5: Request Methods
6: Nationality Methods
1
User Name :
Admin
Password :
123
Login successful for user: Admin
Enter 99 For Exit Or Enter 1 to Continue
1

```

In this output, the program enables users to interact with ArrayLists for simulating login, registration, and recipe management systems.

In the given output, the user selects "Customer" as the user type and chooses the "Login" operation. They enter the username "Cust" and the password "Pass". However, since there is no existing user with those credentials, the program displays the error message: "Login failed for user: Cust".

Afterward, the user is given the option to enter '99' to exit the program or '1' to continue. In this case, the user chooses to continue.



```
ArrayLists are Declared :
Choose one of the User type :
1: Administrator
2: Customer
3: Claculate BMI
4: Visitor
1
Administrator
you have full access
Choose one of the List :
1: Login
2: Register
3: Customer Methods
4: Recipe Methods
5: Request Methods
6: Nationality Methods
2
User Name :
Shatha
Password :
123
Admin full name :
Shatha Ahmed
Phone :
159
Email :
Shatha@company.com
Administrator Add Successful
Enter 99 For Exit Or Enter 1 to Continue
```

In this output, the demonstrates a simulation of a login and registration system. An administrator can add new users with various details such as username, password, full name, phone number, and email. Administrators have access to all functionalities within the system.

In this scenario, the administrator selects the "Administrator" user type and adds a new administrator named Shatha Ahmed with the username "Shatha", password "123", phone number "159", and email "Shatha@company.com". The system confirms the successful addition.

The administrator can either exit the system by entering "99" or continue by entering "1".

```
Choose one of the User type :
1: Administrator
2: Customer
3: Claculate BMI
4: Visitor
1
Administrator
you have full access
Choose one of the List :
1: Login
2: Register
3: Customer Methods
4: Recipe Methods
5: Request Methods
6: Nationality Methods
3
1: Add Customer
```

```
6: Nationality Methods
3
1: Add Customer
2: View Customer
1
Enter the full Information
Enter User Name
Ahmed
Enter full name
Ahmed
Enter Password
123
Enter Phone
147
Enter DOB
31/12/1990
Enter POB
KSA
Enter Nationality
Nationality ID : Nationality description :
1 KSA
2 UAE
3 JOR
4 Egypt
1
Enter Email
Ahmed@my.com
Customer add successful
Enter 99 For Exit Or Enter 1 to Continue
1
```

In this output, the user selects "Administrator" as the user type and is informed that they have full access. They then choose "Customer Methods" from the available options. Within Customer Methods, they select "Add Customer" and choose "Administrator" as the user type to add. The script confirms their full access.

```
1: Login
2: Register
3: Customer Methods
4: Recipe Methods
5: Request Methods
6: Nationality Methods
4
1: Add Recipe
2: View Recipe
1
Enter Recipe Description
Mansaf
Enter Recipe image source
c:\\img\\mansaf.jpg
Enter Recipe type
ID Type
1 Breakfast
2 Lunch
3 Dinner
4 Dessert
5 Snak
2
Enter Recipe components
meet and rice and ugert
Enter Recipe Details
good food
Done
Add Recipe Successful
Enter 99 For Exit Or Enter 1 to Continue
```

In this output, the represents a command-line program simulating a recipe management system. Users can login or register and choose options to add recipes, view recipes, and manage customer requests. The program provides functionalities such as Customer Methods, Recipe Methods, Request Methods, and Nationality Methods.

To add a recipe, users are prompted to provide details like description, image source, type, components, and details. The program confirms the successful addition of the recipe.

Users can continue using the program by selecting options from the main menu or exit by choosing option 99.



```
Administrator
you have full access
Choose one of the List :
1: Login
2: Register
3: Customer Methods
4: Recipe Methods
5: Request Methods
6: Nationality Methods
5
1: Add Request
2: View Request
1
Enter Request Description
Mathbi
Enter Request Customer
Ahmed
Enter Recipe type
ID      Type
1       Breakfast
2       Lunch
3       Dinner
4       Dessert
5       Snak
2
Enter Request Answer
....
Done
Enter 99 For Exit Or Enter 1 to Continue
```

In this output, the Request Methods section, the user selects option 1 to add a request. They are prompted to provide the request description, customer details, recipe type, and request answer. The system adds the request and confirms its success. The user is then given the choice to exit (entering 99) or continue (entering 1).

```
Choose one of the User type :
1: Administrator
2: Customer
3: Claaculate BMI
4: Visitor
1
Administrator
you have full access
Choose one of the List :
1: Login
2: Register
3: Customer Methods
4: Recipe Methods
5: Request Methods
6: Nationality Methods
3
1: Add Customer
2: View Customer
2
User Name | Password | Email | Full Name | DOB | POB | Phone | Nationality
Cust pass email cust
Ahmed 123 Ahmed@my.com Ahmed 31/12/1990 KSA 147 1
Enter 99 For Exit Or Enter 1 to Continue
```

In this output, the user, acting as an administrator, views customer details and successfully adds a new customer. They decide to continue using the program.

```
Choose one of the User type :
1: Administrator
2: Customer
3: Claaculate BMI
4: Visitor
1
Administrator
you have full access
Choose one of the List :
1: Login
2: Register
3: Customer Methods
4: Recipe Methods
5: Request Methods
6: Nationality Methods
4
1: Add Recipe
2: View Recipe
2
id | description | type | img_sr | components | details
1 | Saleet al tomato | 1 | c:// | tomato | tomato
2 | Mansaf | 2 | c:\\img\\mansaf.jpg | meet and rice and uvert | good food
Enter 99 For Exit Or Enter 1 to Continue
```

In this output, the user views and adds recipes by entering their details. They can continue adding recipes or exit the program from the main menu.



```
1 Choose one of the User type :
2 1: Administrator
3 2: Customer
4 3: Calculate BMI
5 4: Visitor
6
7 Administrator
8 you have full access
9 Choose one of the List :
10 1: Login
11 2: Register
12 3: Customer Methods
13 4: Recipe Methods
14 5: Request Methods
15 6: Nationality Methods
16
17 1: Add Request
18 2: View Request
19
20 id | Description | Date | Status | Answer | who make it | recipe type
21 |-----|-----|-----|-----|-----|-----|-----|
22 1 | tomato | 2023/12/03 20:03:49 | 1 | not yet | hashim | 1
23 2 | Mathbi | 2023/12/03 20:03:49 | 1 | ..... | Ahmed | 2
24
25 Enter 99 For Exit Or Enter 1 to Continue
```

In this output, the user, as an administrator, performs tasks within the system by selecting actions from a list. The system provides confirmation and options to continue or exit.

```
1
2 Choose one of the User type :
3 1: Administrator
4 2: Customer
5 3: Calculate BMI
6 4: Visitor
7
8 Weight :
9 100
10 Height :
11 177
12 BMI = 31.91930798940279
13 Enter 99 For Exit Or Enter 1 to Continue
14
15 Choose one of the User type :
16 1: Administrator
17 2: Customer
18 3: Calculate BMI
19 4: Visitor
20
21 Visitor
22 you can just view the list of recipes
23 id | Description | type | img_src | components | details
24 1 | Saleet al tomato | 1 | c:/// | tomato | tomato
25 2 | Haneef | 2 | c:\\img\\haneef.jpg | meat and rice and ugert | good food
26
27 Enter 99 For Exit Or Enter 1 to Continue
```

In this output, the program simulates a restaurant menu system where users can calculate their BMI or view recipes. Users can choose a user type and perform actions accordingly. The program repeatedly prompts the user to select a user type and continues until the user decides to exit.

```
1
2 Choose one of the User type :
3 1: Administrator
4 2: Customer
5 3: Calculate BMI
6 4: Visitor
7
8 Administrator
9 you have full access
10 Choose one of the List :
11 1: Login
12 2: Register
13 3: Customer Methods
14 4: Recipe Methods
15 5: Request Methods
16 6: Nationality Methods
17
18 1: Add Nationality
19 2: View Nationality
20
21 Nationality ID : Nationality description :
22 1 KSA
23 2 UAE
24 3 JOR
25 4 Egypt
26 5 Qatar
27
28 Enter 99 For Exit Or Enter 1 to Continue
29 99
30 BUILD SUCCESSFUL (total time: 1 minute 12 seconds)
```

In this output, the user selects the Administrator user type and is informed about their full access. They choose to add a new nationality. The system displays a list of nationality IDs, descriptions, and related data. The user is given the option to exit (entering 99) or continue (entering 1). The user selects 99 to exit the system.



Difficulty:

We were having some errors that took time to figure out.

- When we run the program in a place where we must enter something the compiler skips it and goes to the next output without letting the user enter anything.
This problem is caused by NetBeans itself and to solve that problem we created more input objects with a scanner.
- The user class is a superclass for both the customer and Administrator classes.
So, we wanted to inherit some attributes from the super to the subclass.
one of them is Use_type.
We tried to make user_type in the customer class always with the value 2 and in the administrator Class with the value 1, Assigning value didn't work out.
So, to solve the error, we decided to delete the user type from the user class and create in each class variable with a value.
- On the part of the work team, there were problems finding a suitable time for everyone to discuss what we would do next.

Therefore, we decided that when any member finishes writing apart, she must record a video clip explaining what she did and suggesting a next step.

Future work:

1. Creating interactive interfaces in OOP2
2. Improving user capabilities
3. Research in the field of software engineering: This can include working on a research project, presenting a research paper at a conference, or publishing an article in a scientific journal.



Conclusion:

Our recipe program stands as a testament to the boundless creativity and cultural richness that food embodies. By fostering a platform for users to explore and share traditions from around the globe,

The program user-friendly interface, detailed instructions, and nutritional awareness empower users of all skill levels to confidently recreate delicious dishes while making informed choices aligned with their health goals. We are committed to continuous improvement,

As we move forward, we envision an interactive interface that enhances the user experience, expanded user capabilities that foster deeper engagement, and innovative research contributions in the field of software engineering.

The role of each member:

Shatha	Test class, Administrator class, User class, The report
Lojain	Test class, Administrator class, User class, User type class
Zainab	Test class, Customer class, User class, Usertype class, The report
Rahaf	Customer class, Requests class, Status type class
Kawthar	Recipe type class, Recipes class, Test class
Ola	Recipe type class, Recipes class, Test class
Raghad	Customer class, Requests class, Status type class

References:

Liang (2013) *Introduction to java programming: Comprehensive version*. Boston: Pearson.

Some YouTube videos.