User Requirement Specification Document

Individual Project

Recipeasy

Date	12/01/2024
Version	0.7
State	Draft
Author	Z. Imad Aqil M.

1. User Requirements

i. Non-Functional Requirements

No	General Requirements	MosCoW
1	Usability	Must
2	Security	Should
3	Compatibility	Should
4	Maintainability	Could

ii. Functional Requirements

No	App type	General Requirements	MoSCoW
FR-01	Desktop	User Log in (Desktop)	Must
FR-02	Desktop	Create new user (Admin)	Must
FR-03	Website	Register new account	Must
FR-04	Website	User Log in (Website)	Must
FR-05	Desktop	Edit existing users (Admin)	Must
FR-06	Website	Change account details	Must
FR-07	Desktop	Delete/Hide existing Users (Admin)	Must
FR-08	Website	Delete Account	Must
FR-09	Desktop/Website	Add new recipes	Must
FR-10	Desktop/Website	Edit existing recipes	Must
FR-11	Desktop/Website	Delete/Hide existing recipes	Must
FR-12	Website	Leave Comment on recipes	Should
FR-13	Website	Add recipe to favorites	Should
FR-14	Desktop/Website	Search for recipe by name	Should
FR-15	Desktop/Website	Filter by type	Should
FR-16	Website	User changes password	Could
FR-17	Website	System recommends recipes	Could

2. Use Cases

i. Non-Functional Requirements

- **Usability**

The application must be easy to use for the client.

- **Security**

The application should be secure in order to avoid data leaks and other security attacks.

- **Compatibility**

The application Should wok on desktop hardware and mobile hardware.

- **Maintainability**

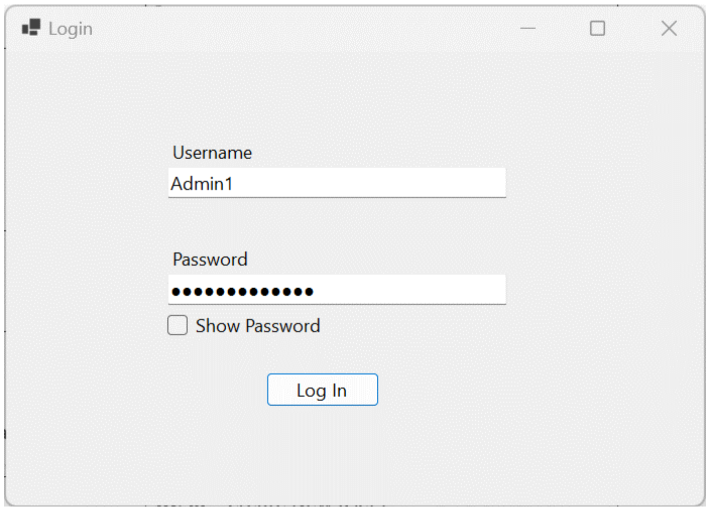
In case the application runs into a critical failure, the application could be fixed easily.

ii. Functional Requirements

UC-01. User Log in (Desktop)

Admin will need to log on in order to access the desktop application

Use case:	**UC-01**: User logs in (Desktop)
Functional Requirements	**FR-01**
Actor:	Admin
Pre-condition:	-
Main Success Scenario:	1. Actor starts desktop application 2. System shows login field 3. Actor fills in required field 4. System checks if the credentials are valid 5. System logs user into application
Extensions:	4a) credentials invalid 1. Error message shown 2. Return to step 3

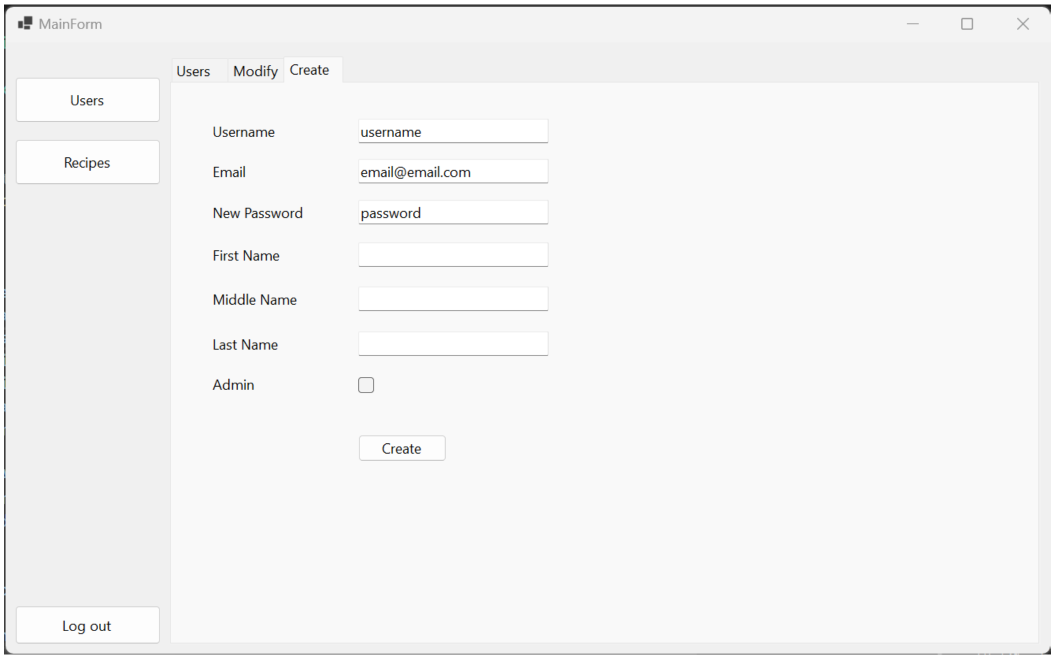


Picture 1.1 Admin creating new user on the desktop application

UC-02. Create new users

A system administrator can add a new user to the database by providing essential details. They could add either admin user or normal user

Use case:	**UC-02**: Create New users
Functional Requirements	**FR-02**
Actor:	Admin
Pre-condition:	Admin needs to log in (desktop)
Main Success Scenario:	1. Actor requests to create new user 2. System shows field that required to be entered 3. Actor filled in required data 4. Actor clicks on add user 5. System checks input data 6. User added 7. End use case
Extensions:	5a) Data not valid 1. System shows error 2. Return to step 3



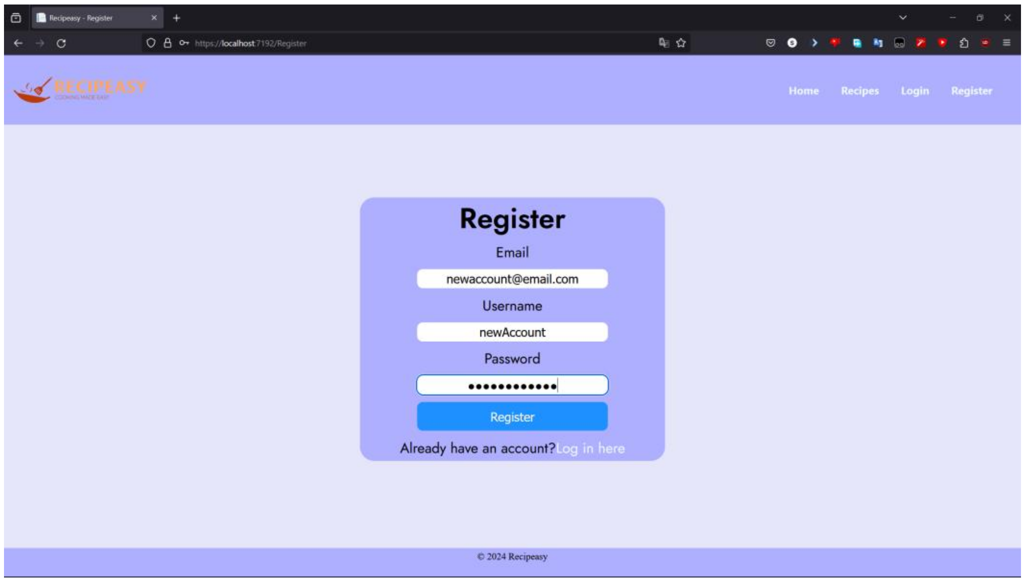
Username	username
Email	email@email.com
New Password	password
First Name	
Middle Name	
Last Name	
Admin	☐

Picture 2.1 Admin creating new user on the desktop application

UC-03. Register new account

User can register a new account through the website

Use case:	**UC-03**: Register new account
Functional Requirements	**FR-03**
Actor:	User
Pre-condition:	Website opened
Main Success Scenario:	1. Actor goes to register user 2. System shows required input field 3. Actor adds necessary details 4. Actor clicks on Register 5. System checks user input 6. System registers data 7. System shows success message 8. End use case
Extensions:	5a) Data not valid 1. System shows error 2. Return to step 3



Register

Email: newaccount@email.com

Username: newAccount

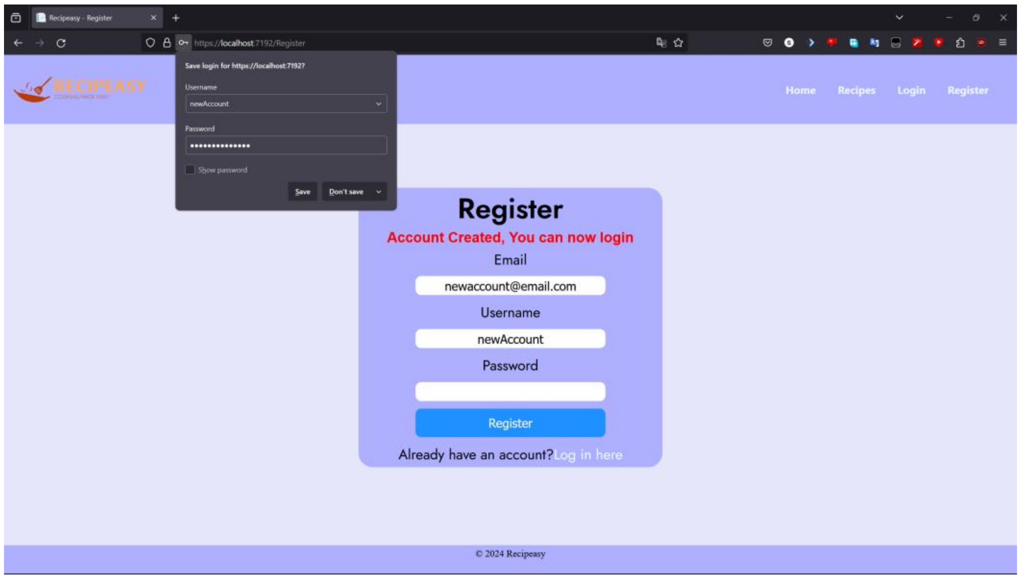
Password: •••••••••••••

Register

Already have an account? Log in here

© 2024 Recipeasy

Picture 3.1 User can register an account on the website through the register page



Save login for https://localhost:7192?

Username: newAccount

Password: ••••••••••••••

Register

Account Created, You can now login

Email: newaccount@email.com

Username: newAccount

Password:

Already have an account? Log in here

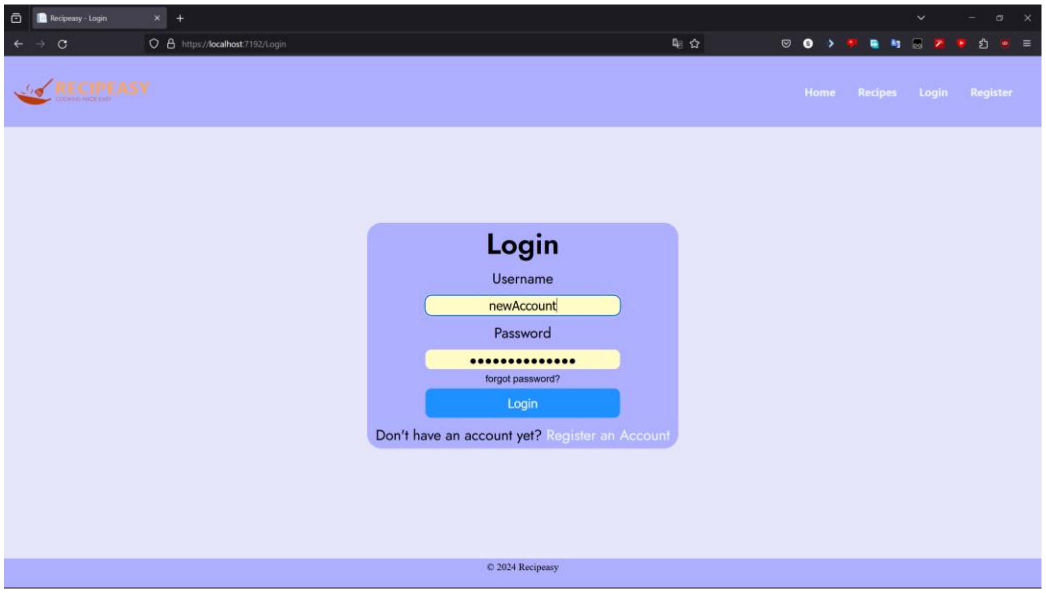
© 2024 Recipeasy

Picture 3.2 Message shown after creating an account

UC-04. User Log in (Website)

User will need to be able to log onto the website in order to use the application

Use case:	**UC-04**: User Log in (Website)
Functional Requirements	**FR-04**
Actor:	User
Pre-condition:	-
Main Success Scenario:	1. Actor navigates to Login page 2. System shows login field 3. Actor fills in required field 4. System checks if the credentials are valid 5. System logs user into application
Extensions:	4a) credentials invalid 1. Error message shown 2. End use case



Login

Username

newAccount

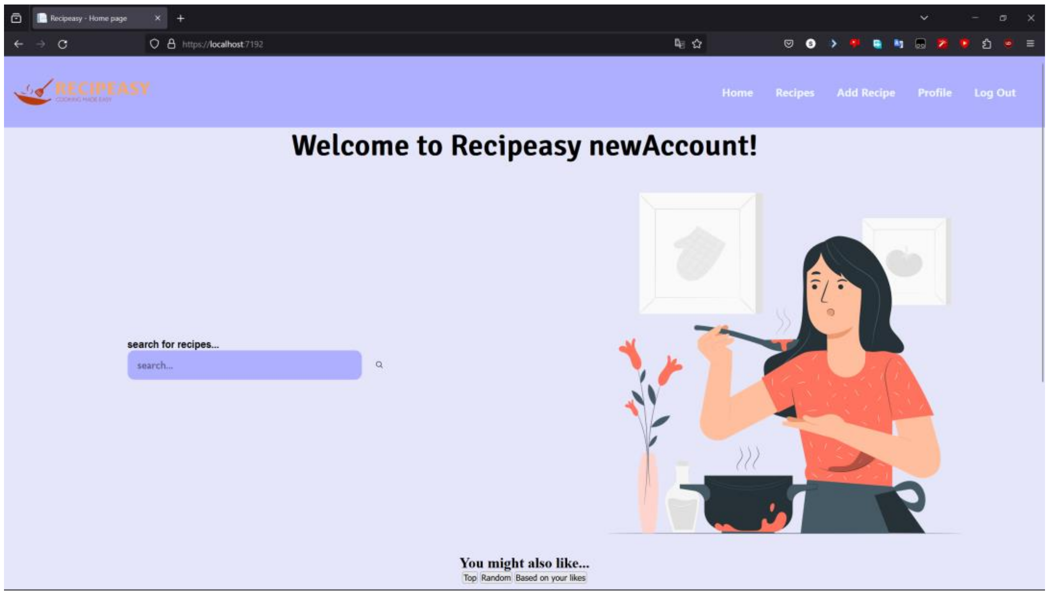
Password

••••••••••••••

Don't have an account yet? Register an Account

© 2024 Recipeasy

Picture 4.1 Login Page



Welcome to Recipeasy newAccount!

You might also like...

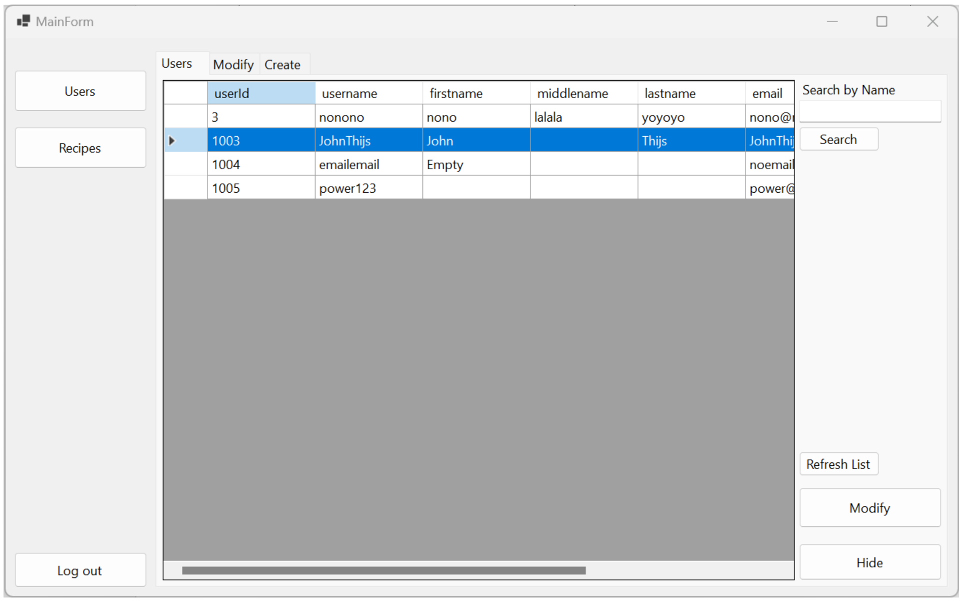
Picture 4.2 The index page after user has logged in

UC-05. Edit existing user

Admin can edit user details

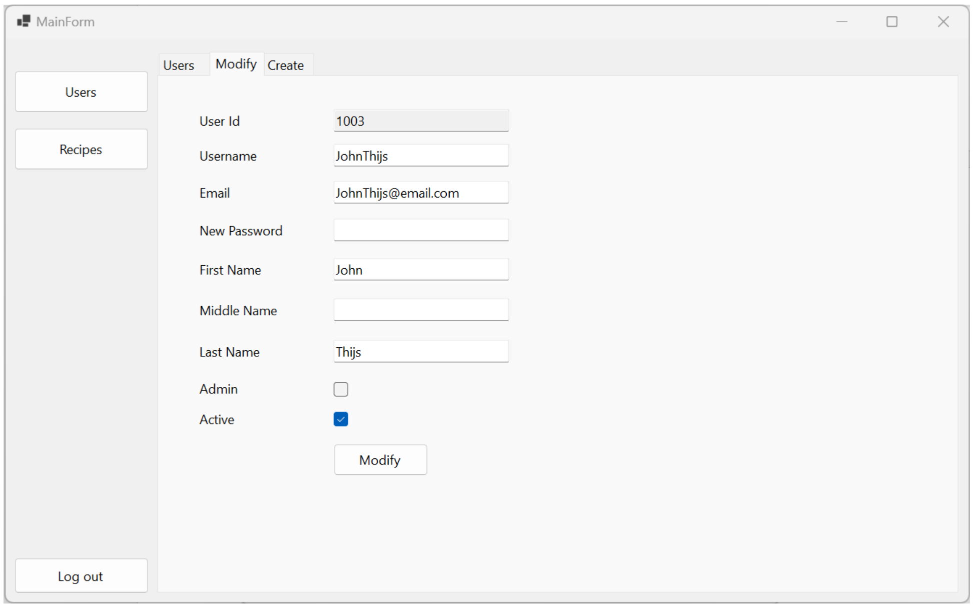
Use case:	**UC-05**: Edit existing user
Functional Requirements	**FR-05**
Actor:	Admin

Pre-condition:	Be logged in (desktop)
Main Success Scenario:	1. Actor selects a specific user to edit 2. System shows fields that are currently filled with the user’s details 3. Actor changes a detail from the wanted fields. 4. System shows confirmation message 5. Actor confirms changes 6. System checks user input 7. System displays new user details 8. End use case
Extensions:	5a) Actor cancels operation 1. Operation cancelled 2. End use case 6a) Data not valid 1. System shows error 2. Return to step 3



userId	username	firstname	middlename	lastname	email
3	nonono	nono	lalala	yoyoyo	nono@
1003	JohnThijs	John		Thijs	JohnThi
1004	emailemail	Empty			noemail
1005	power123				power@

Picture 5.1 Admin selects desired user to modify



User Id	1003
Username	JohnThijs
Email	JohnThijs@email.com
New Password	
First Name	John
Middle Name	
Last Name	Thijs
Admin	☐
Active	✓

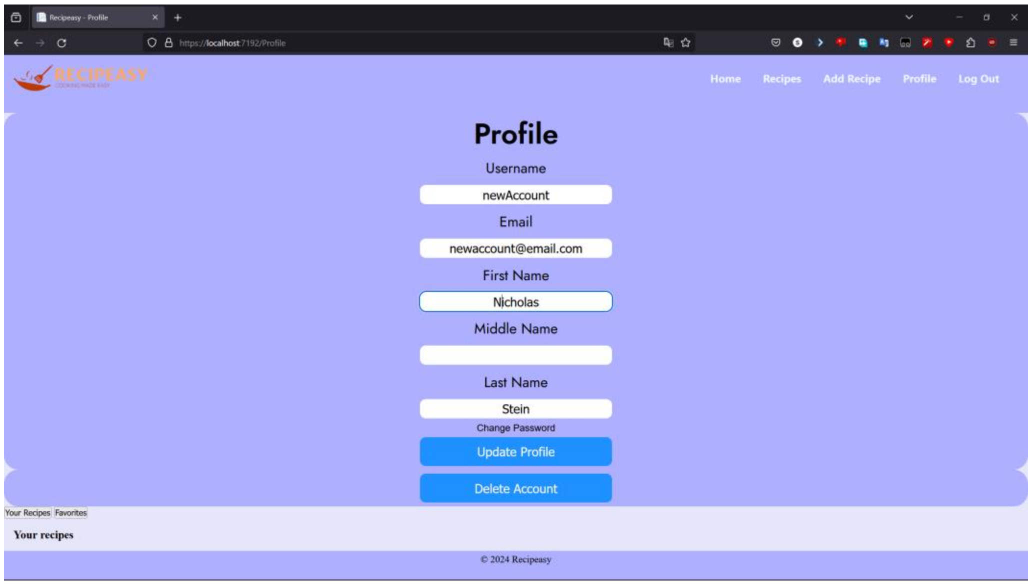
Picture 5.2 After selecting the user, admin is then redirected to the Modify User tab

UC-06. Change account details

User can change their account details once they are logged on in the website

Use case:	**UC-06**: Change account details
Functional Requirements	**FR-06**
Actor:	User
Pre-condition:	Be logged in (Website)
Main Success Scenario:	1. Actor navigates to profile page 2. System shows current account details 3. Actor changes a detail from the wanted fields. 4. System shows confirmation message 5. Actor confirms changes 6. System checks user input 7. System displays new account details 8. End use case
Extensions:	5a) Actor cancels operation 1. Operation cancelled 2. End use case 6a) Data not valid 1. System shows error

	2. Return to step 3


https://localhost:7192/Profile

Profile

Username: newAccount

Email: newaccount@email.com

First Name: Nicholas

Middle Name:

Last Name: Stein

Your recipes

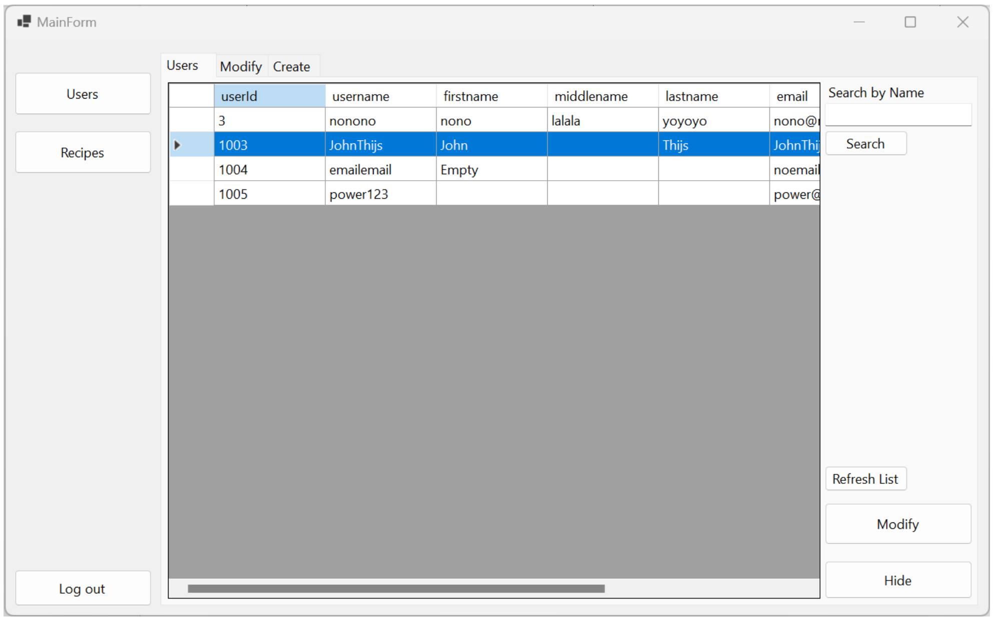
© 2024 Recipeasy

Picture 6.1 User changing their details in the profile page

UC-07. Delete existing users

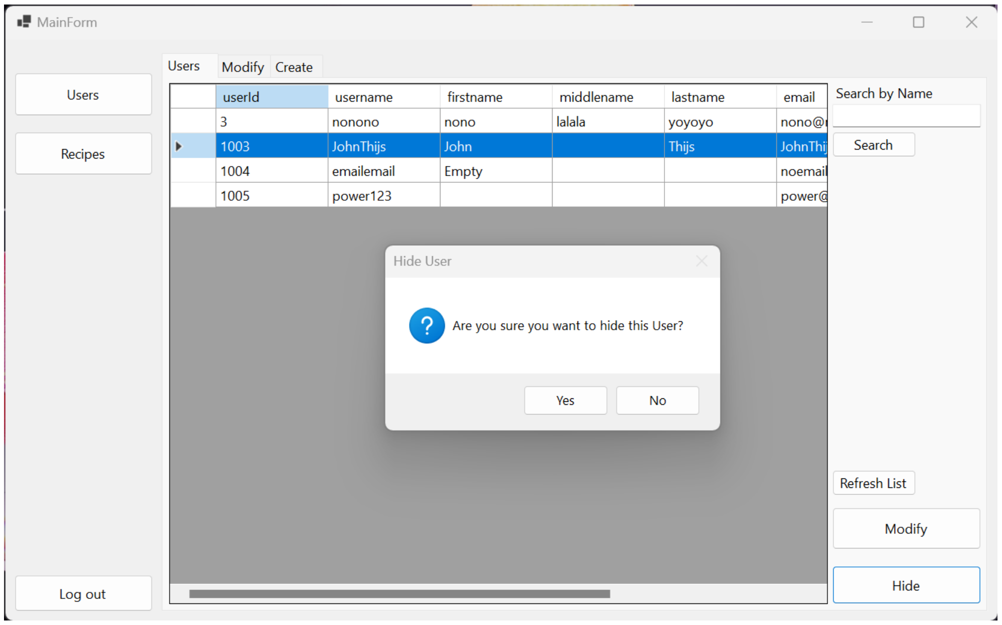
Admin is able to delete users from the desktop application

Use case:	**UC-07**: Delete existing user
Functional Requirements	**FR-07**
Actor:	Admin
Pre-condition:	Be logged in
Main Success Scenario:	1. Actor requests to delete user 2. System asks actor for confirmation 3. Actor confirms 4. User deleted 5. End use case
Extensions:	3a) Actor cancels the action 1. User is not deleted End use case



userId	username	firstname	middlename	lastname	email
3	nonono	nono	lalala	yoyoyo	nono@
1003	JohnThijs	John		Thijs	JohnThi
1004	emailemail	Empty			noemail
1005	power123				power@

Picture 7.1 Admin selects desired user to Delete/hide



userId	username	firstname	middlename	lastname	email
3	nonono	nono	lalala	yoyoyo	nono@
1003	JohnThijs	John		Thijs	JohnThi
1004	emailemail	Empty			noemail
1005	power123				power@

Are you sure you want to hide this User?

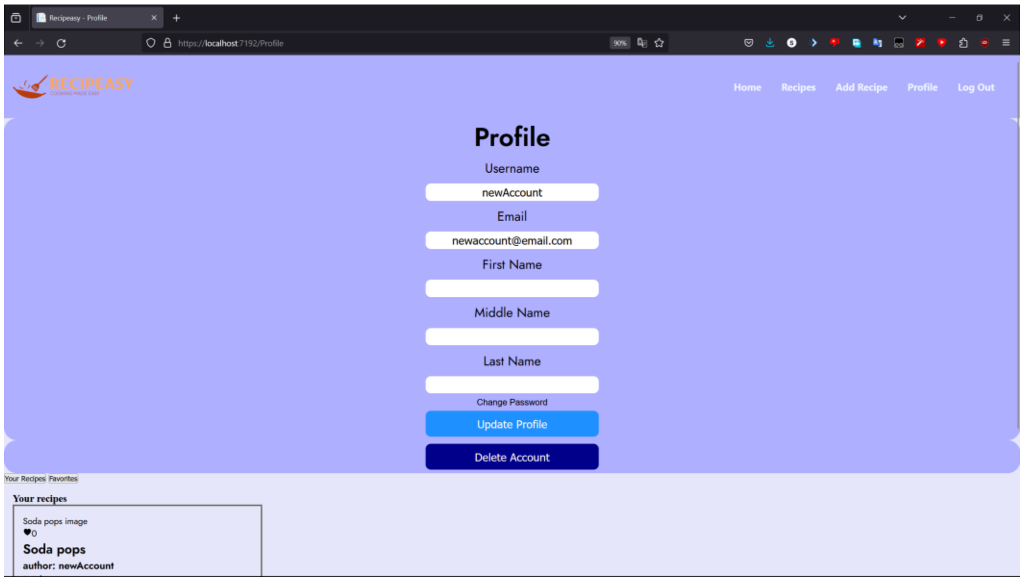
Picture 7.2 Confirmation message after clicking the Hide button (Bottom Right)

UC-08. User Deletes Account

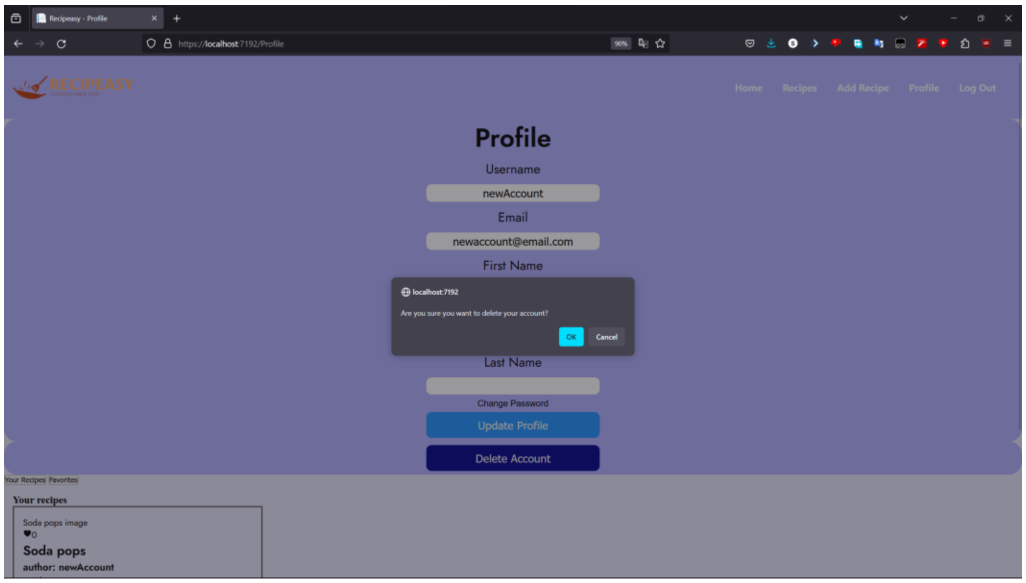
User can choose to the delete their account.

Use case:	**UC-08**: User deletes account
Functional Requirements	**FR-08**

Actor:	User
Pre-condition:	Be logged in
Main Success Scenario:	1. Actor requests to delete account 2. System gives confirmation message 3. Actor confirms 4. System deletes user account 5. Actor redirected to Index page 6. End use case
Extensions:	3a) Actor cancels the action 1. User is not deleted 2. End use case



Picture 8.1 Highlighted in dark blue is the delete account button



Profile

Username: newAccount

Email: newaccount@email.com

First Name

Last Name

localhost:7192

Are you sure you want to delete your account?

Your recipes

Soda pops image

♥0

Soda pops

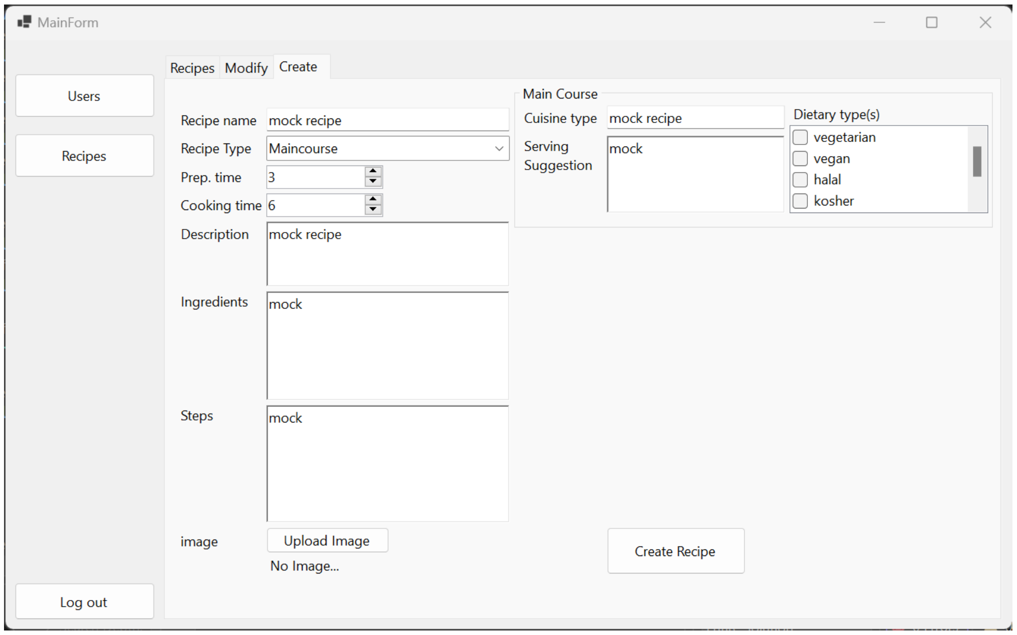
author: newAccount

Picture 8.2 Confirmation message to delete the account

UC-09. Add new recipes

Both user and admin can add recipes into the application

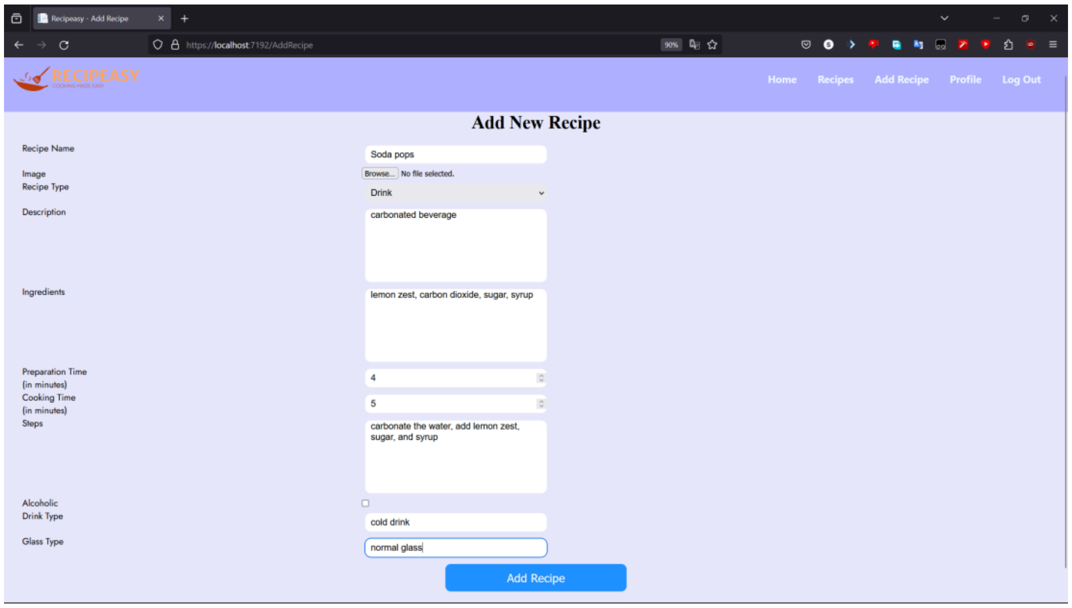
Use case:	**UC-09:** Add new recipes
Functional Requirements	**FR-09**
Actor:	Admin, User
Pre-condition:	Be logged in
Main Success Scenario:	1. Actor navigates to Add Recipe page 2. System shows required fields 3. Actor filled in required data 4. Actor clicks on add recipe 5. System checks input 6. System adds recipe into database 7. End use case
Extensions:	5a) data is invalid 1. System shows error message 2. Return to step 3



Recipe name: mock recipe
Recipe Type: Maincourse
Prep. time: 3
Cooking time: 6
Description: mock recipe
Ingredients: mock
Steps: mock
image: No Image...

Main Course
Cuisine type: mock recipe
Serving Suggestion: mock
Dietary type(s): vegetarian, vegan, halal, kosher

Picture 9.1 Admin creating new recipe on the desktop application



Add New Recipe

Recipe Name: Soda pops
Image: No file selected.
Recipe Type: Drink
Description: carbonated beverage
Ingredients: lemon zest, carbon dioxide, sugar, syrup
Preparation Time (in minutes): 4
Cooking Time (in minutes): 5
Steps: carbonate the water, add lemon zest, sugar, and syrup
Alcoholic: ☐
Drink Type: cold drink
Glass Type: normal glass

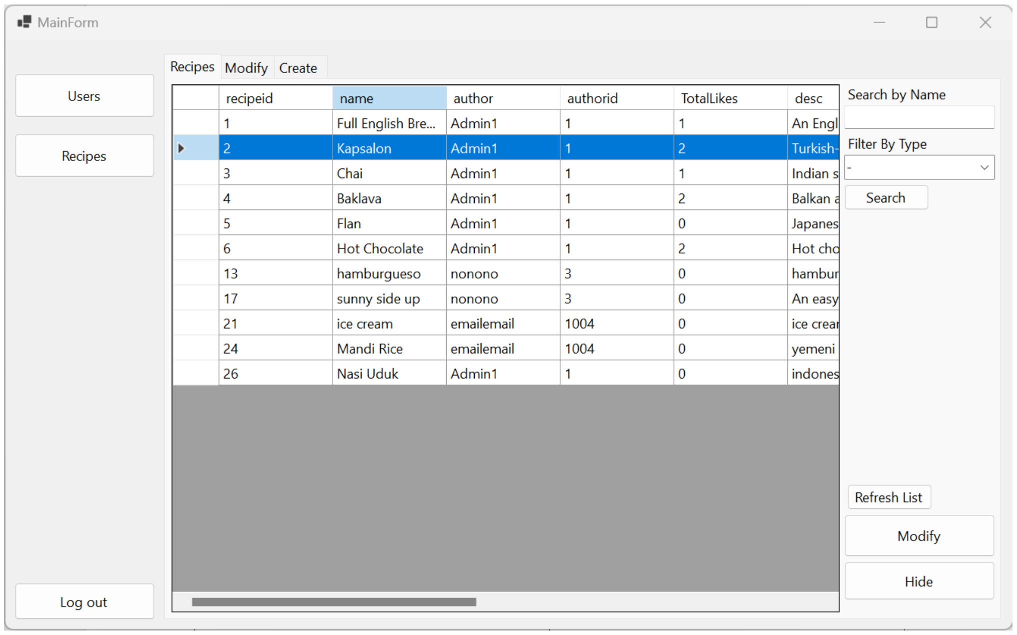
Picture 9.2 Creating recipe on the Website

UC-10. Edit existing recipes

Both user and admin can edit recipes that they made. Admin can edit recipes that normal users made, but normal users will only able to edit recipes that they made

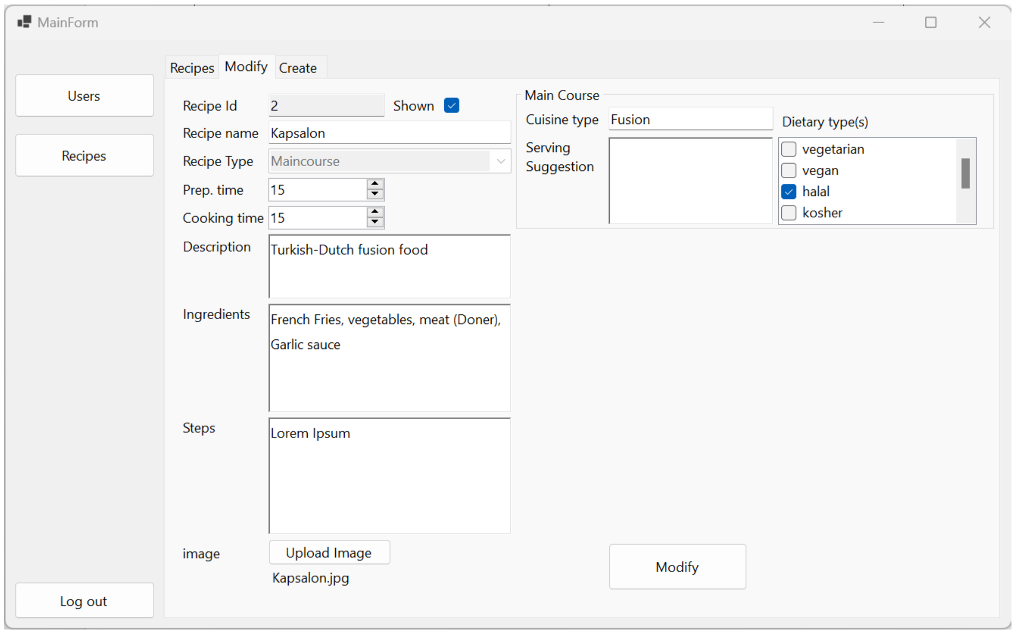
Use case:	**UC-10: Edit existing recipes**
Functional Requirements	**FR-10**

Actor:	Admin, User
Pre-condition:	Be logged in, On Website->Profile Page
Main Success Scenario:	1. Actor selects a specific recipe to be edited 2. System shows fields with current recipe details 3. Actor changes details from the wanted fields. 4. System shows confirmation message 5. Actor confirms changes 6. System checks input 7. System displays that the user details has been changed
Extensions:	5a) Actor cancels operation 1. Operation cancelled 2. End use case 6a) Data not valid 1. System shows error 2. Return to step 3



recipeid	name	author	authorid	TotalLikes	desc
1	Full English Bre...	Admin1	1	1	An Engl
2	Kapsalon	Admin1	1	2	Turkish-
3	Chai	Admin1	1	1	Indian s
4	Baklava	Admin1	1	2	Balkan a
5	Flan	Admin1	1	0	Japanes
6	Hot Chocolate	Admin1	1	2	Hot cho
13	hamburgueso	nonono	3	0	hambur
17	sunny side up	nonono	3	0	An easy
21	ice cream	emailemail	1004	0	ice crea
24	Mandi Rice	emailemail	1004	0	yemeni
26	Nasi Uduk	Admin1	1	0	indones

Picture 10.1 Admin selects desired recipe to modify

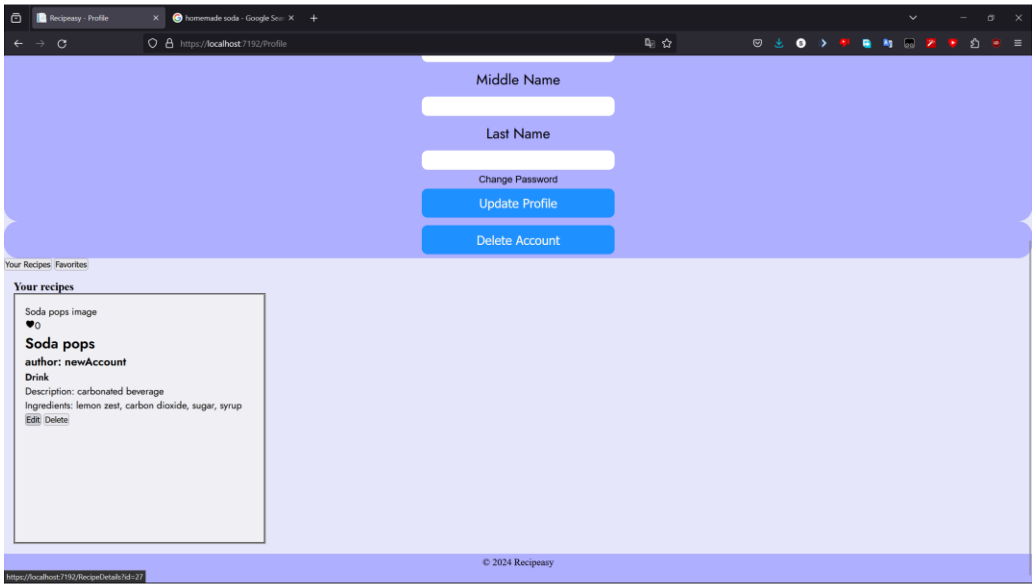

MainForm

Recipe Id	2
Shown	✓
Recipe name	Kapsalon
Recipe Type	Maincourse
Prep. time	15
Cooking time	15
Description	Turkish-Dutch fusion food
Ingredients	French Fries, vegetables, meat (Doner), Garlic sauce
Steps	Lorem Ipsum
image	Kapsalon.jpg

Main Course

Cuisine type	Fusion
Serving Suggestion	
Dietary type(s)	vegetarian [ ], vegan [ ], halal [✓], kosher [ ]

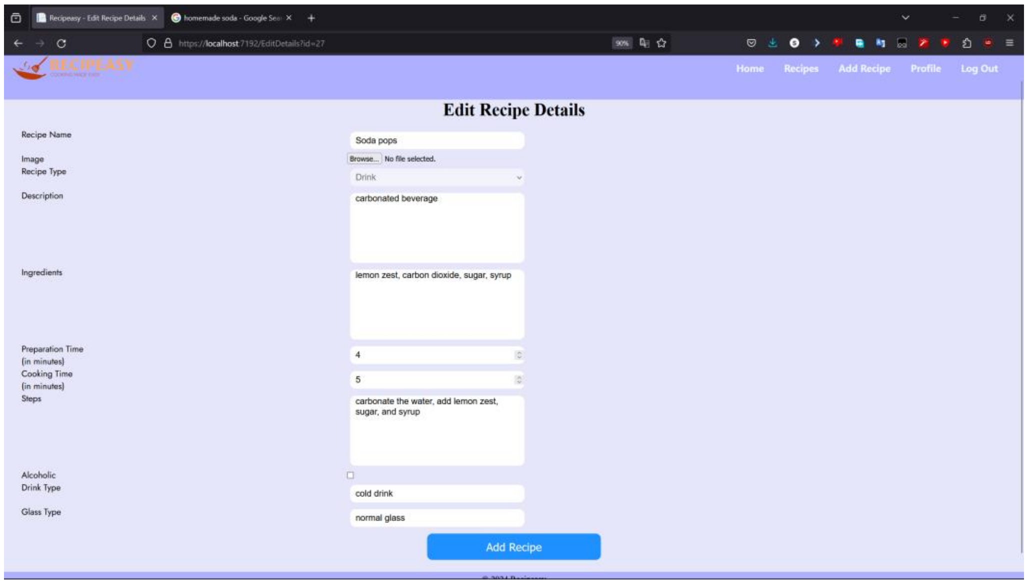
Picture 10.2 After selecting the recipe, admin is then redirected to the Modify User tab



Your recipes

Soda pops image
♥0
Soda pops
author: newAccount
Drink
Description: carbonated beverage
Ingredients: lemon zest, carbon dioxide, sugar, syrup

Picture 10.3 Highlighted in dark grey the edit button of a recipe


Recipeasy - Edit Recipe Details
homemade soda - Google Sea
https://localhost:7192/EditDetails?id=27
RECIPEASY
Home
Recipes
Add Recipe
Profile
Log Out

Edit Recipe Details

Recipe Name	Soda pops
Image	Browse... No file selected.
Recipe Type	Drink
Description	carbonated beverage
Ingredients	lemon zest, carbon dioxide, sugar, syrup
Preparation Time (in minutes)	4
Cooking Time (in minutes)	5
Steps	carbonate the water, add lemon zest, sugar, and syrup
Alcoholic	☐
Drink Type	cold drink
Glass Type	normal glass

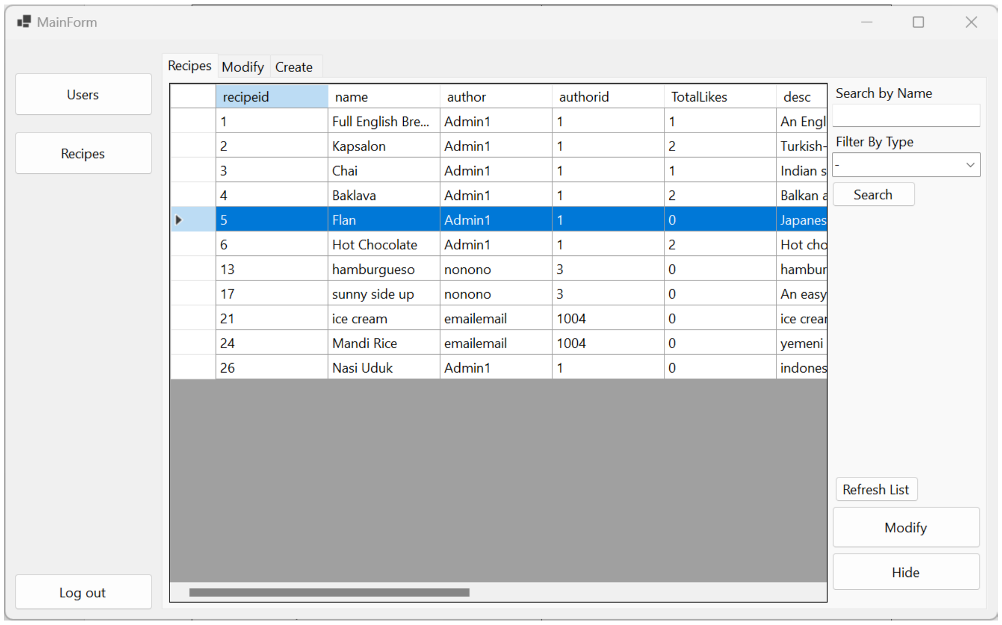
Add Recipe

Picture 10.4 Recipe details are filled in after clicking the edit button

UC-11. Delete existing recipe

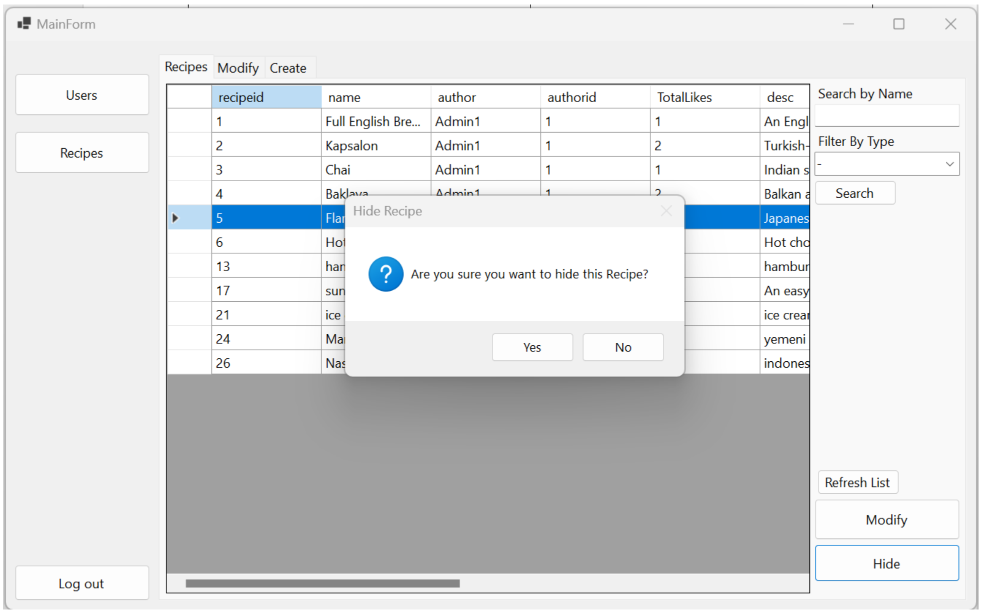
Both user and admin can delete recipes that they made. Admin can delete recipes that normal users made, but normal users will only able to delete recipes that they made

Use case:	**UC-11**: Delete existing recipe
Functional Requirements	**FR-11**
Actor:	Admin, User
Pre-condition:	Be logged in
Main Success Scenario:	1. Actor requests to delete specific recipe 2. System asks actor for confirmation 3. Actor confirms 4. System deletes recipe 5. End use case
Extensions:	3a) Actor cancels the action 1. recipe is not deleted 2. End use case



	recipeid	name	author	authorid	TotalLikes	desc
	1	Full English Bre...	Admin1	1	1	An Engl
	2	Kapsalon	Admin1	1	2	Turkish-
	3	Chai	Admin1	1	1	Indian s
	4	Baklava	Admin1	1	2	Balkan a
▸	5	Flan	Admin1	1	0	Japanes
	6	Hot Chocolate	Admin1	1	2	Hot cho
	13	hamburgueso	nonono	3	0	hambur
	17	sunny side up	nonono	3	0	An easy
	21	ice cream	emailemail	1004	0	ice crea
	24	Mandi Rice	emailemail	1004	0	yemeni
	26	Nasi Uduk	Admin1	1	0	indones

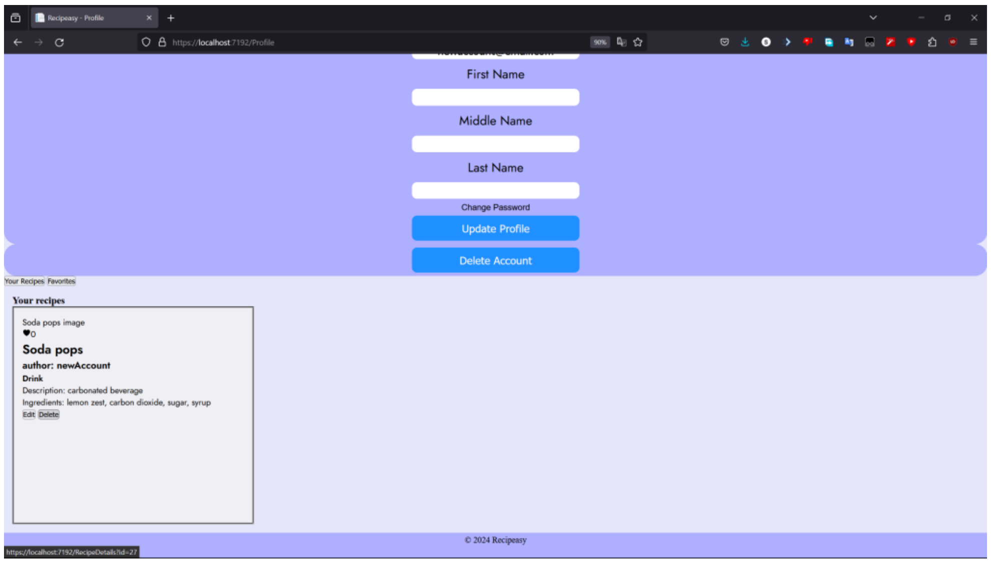
Picture 11.1 Admin selects desired recipe to Delete/hide



Are you sure you want to hide this Recipe?

	recipeid	name	author	authorid	TotalLikes	desc
	1	Full English Bre...	Admin1	1	1	An Engl
	2	Kapsalon	Admin1	1	2	Turkish-
	3	Chai	Admin1	1	1	Indian s
	4	Baklava	Admin1	1	2	Balkan a
▸	5	Flar				Japanes
	6	Hot				Hot cho
	13	har				hambur
	17	sun				An easy
	21	ice				ice crea
	24	Ma				yemeni
	26	Nas				indones

Picture 11.2 Confirmation message after clicking the Hide button (Bottom Right)

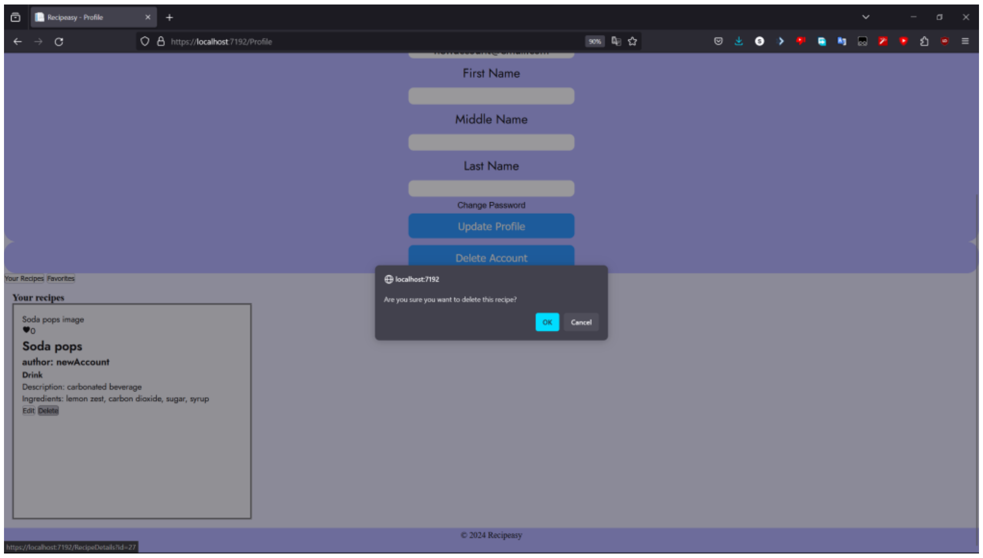


Your recipes

Soda pops image
♥0
Soda pops
author: newAccount
Drink
Description: carbonated beverage
Ingredients: lemon zest, carbon dioxide, sugar, syrup

© 2024 Recipeasy

Picture 11.3 User can only delete their own recipe



Are you sure you want to delete this recipe?

Your recipes

Soda pops image
♥0
Soda pops
author: newAccount
Drink
Description: carbonated beverage
Ingredients: lemon zest, carbon dioxide, sugar, syrup

© 2024 Recipeasy

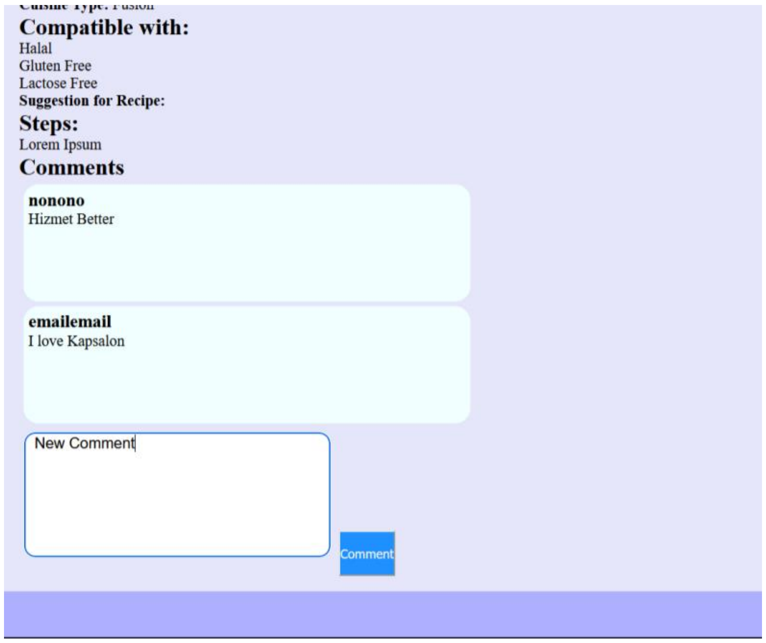
Picture 11.4 confirmation message

UC-12. Leave comment on recipes

Both user and admin can leave comment(s) on any recipes

Use case:	**UC-12: Leave comment on recipes**
Functional Requirements	**FR-12**
Actor:	Admin, User
Pre-condition:	Be logged in

Main Success Scenario:	1. Actor selects a specific recipe to comment on 2. System shows Recipe Details along with all previous comments 3. Actor fills the comment box 4. Actor clicks on Send Comment button 5. System checks user input 6. System stores comment in the database 7. System shows comment on Details Page 8. End Use Case
Extensions:	5a) invalid input 1. System shows error message 2. Return to step 3


Cuisine Type: Fusion

Compatible with:

Halal
Gluten Free
Lactose Free

Suggestion for Recipe:

Steps:

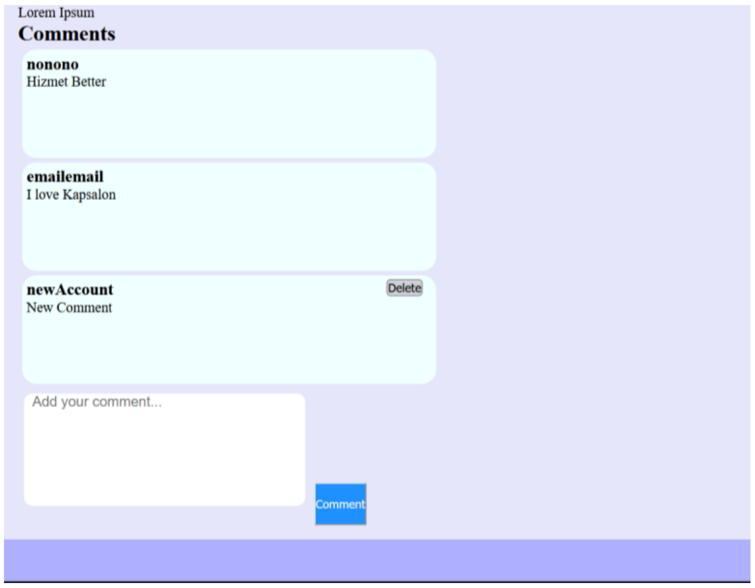
Lorem Ipsum

Comments

nonono
Hizmet Better

emailemail
I love Kapsalon

Picture 12.1 Comment box (white background)


Lorem Ipsum

Comments

nonono
Hizmet Better

emailemail
I love Kapsalon

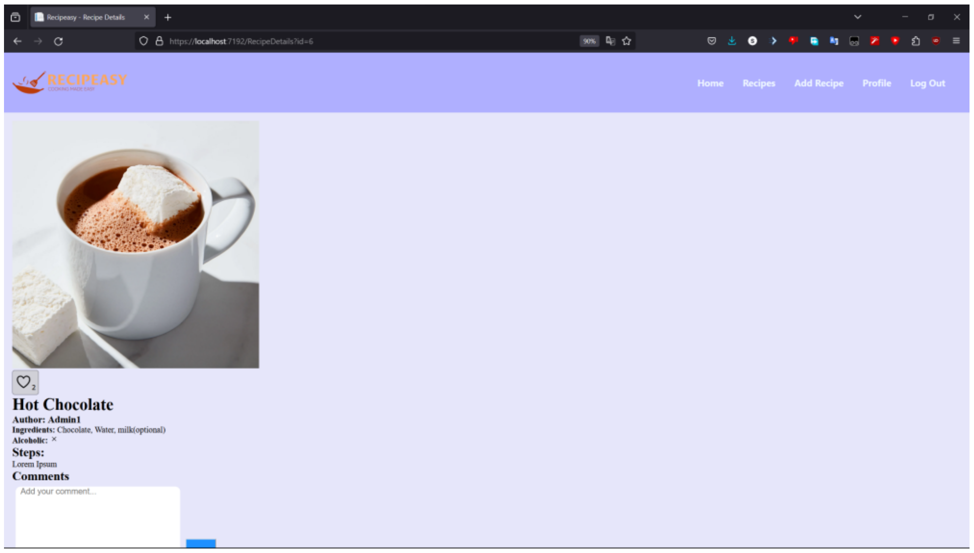
newAccount
New Comment

Picture 12.2 New comment added on the recipe

UC-13. Add recipe to favorites

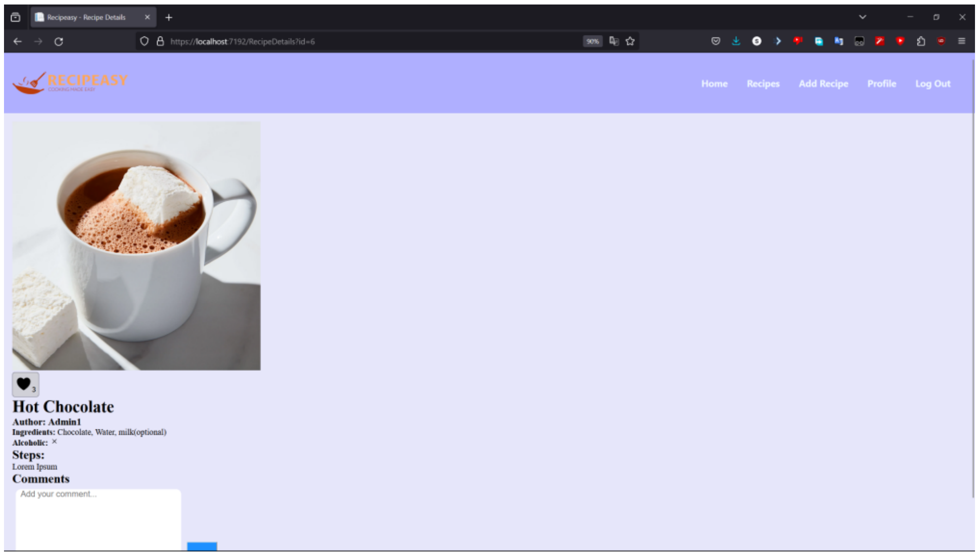
Both user and admin can add any recipes to their favorites

Use case:	**UC-13: Add recipe to Favorite**
Functional Requirements	**FR-13**
Actor:	Admin, User
Pre-condition:	Be logged in
Main Success Scenario:	1. Actor selects a specific recipe to comment on 2. System shows Recipe Details and the number of likes for the given recipe 3. Actor clicks on the heart button 4. System stores users like in the database 5. System shows new number of likes for the given recipe 6. End use case
Extensions:	3a) Actor clicks on the heart button again (dislike the recipe) 1. System removes like of user 2. System shows new number of likes for the given recipe 3. End use case



Hot Chocolate
Author: Admin1
Ingredients: Chocolate, Water, milk(optional)
Alcoholic: ×
Steps:
Lorem Ipsum
Comments

Picture 13.1 User has not liked the recipe. The heart is only outlined



Hot Chocolate
Author: Admin1
Ingredients: Chocolate, Water, milk(optional)
Alcoholic: ×
Steps:
Lorem Ipsum
Comments

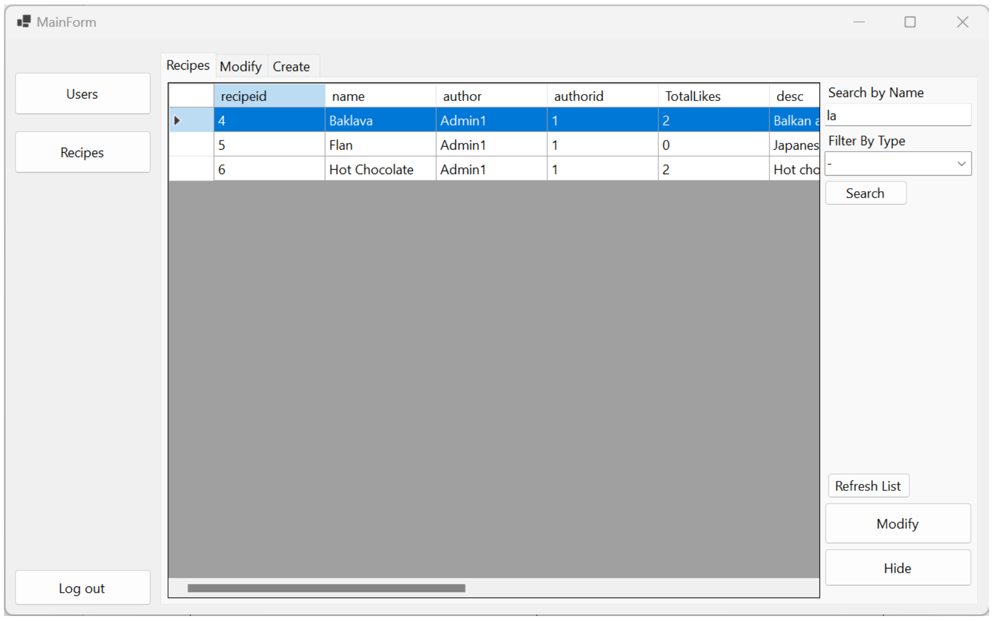
Picture 13.2 User has not liked the recipe. The heart is only outlined

UC-14. Search recipe by name

Both user and admin can search for specific recipes by name

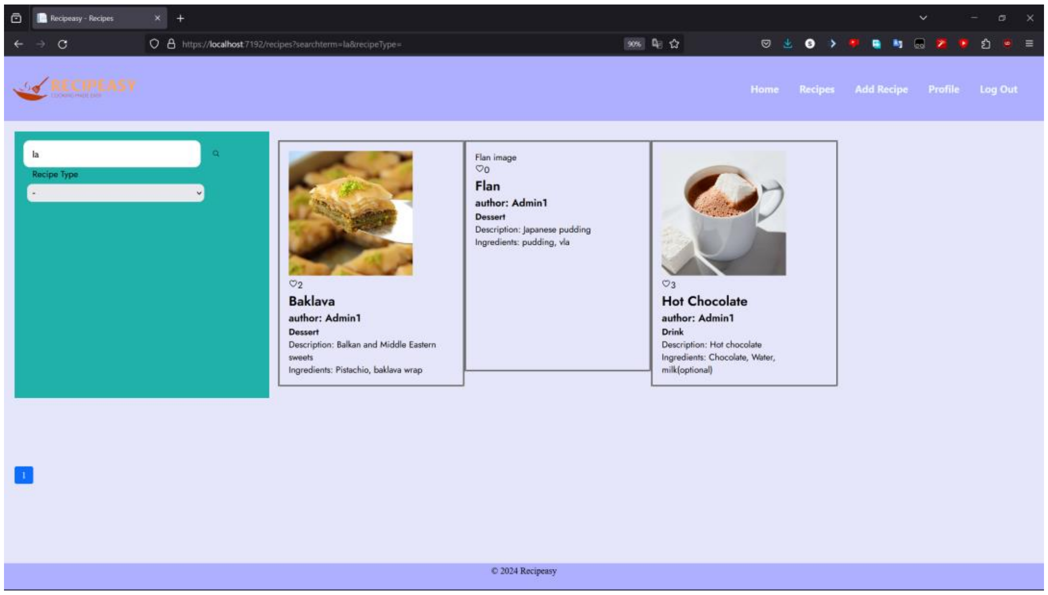
Use case:	**UC-14: Search recipe by name**
Functional Requirements	**FR-14**
Actor:	Admin, User
Pre-condition:	Be logged in

Main Success Scenario:	1. Actor fills search box with specific keyword 2. Actor clicks on the search button 3. System filters recipe by name to get relevant recipes 4. System shows relevant recipes to user 5. End use case
Extensions:	1a) Search box is empty 1. System removes all filters and show all recipes 2. End use case



recipeid	name	author	authorid	TotalLikes	desc
4	Baklava	Admin1	1	2	Balkan a
5	Flan	Admin1	1	0	Japanes
6	Hot Chocolate	Admin1	1	2	Hot cho

Picture 14.1 DataGrid only show recipes that contains “la” as written in the search box (Top Right)



Flan image
♡0
Flan
author: Admin1
Dessert
Description: Japanese pudding
Ingredients: pudding, vla

♡2
Baklava
author: Admin1
Dessert
Description: Balkan and Middle Eastern sweets
Ingredients: Pistachio, baklava wrap

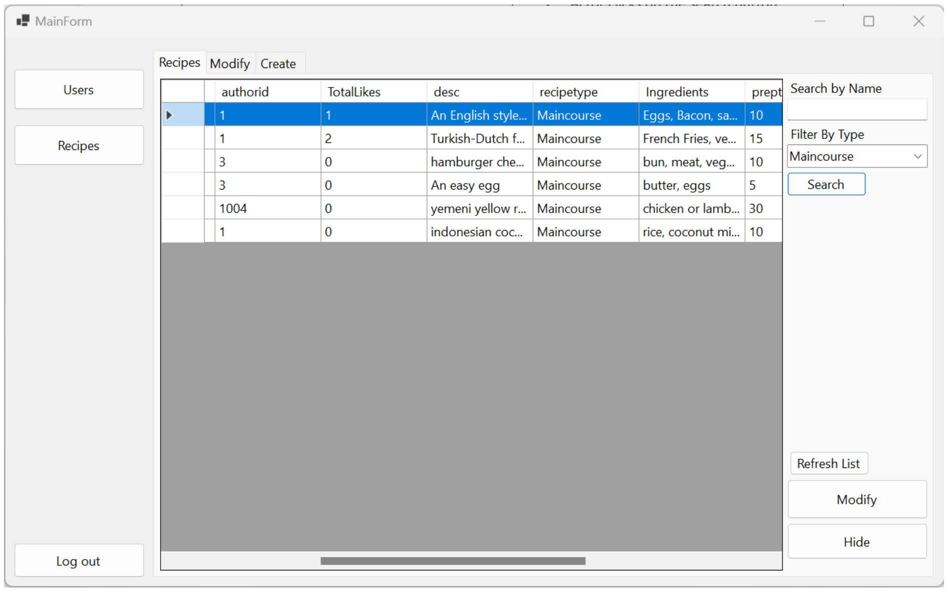
♡3
Hot Chocolate
author: Admin1
Drink
Description: Hot chocolate
Ingredients: Chocolate, Water, milk(optional)

Picture 14.1 Recipes page only shows recipes that contains "la" as written in the search box (Top Left)

UC-15. Filter by type

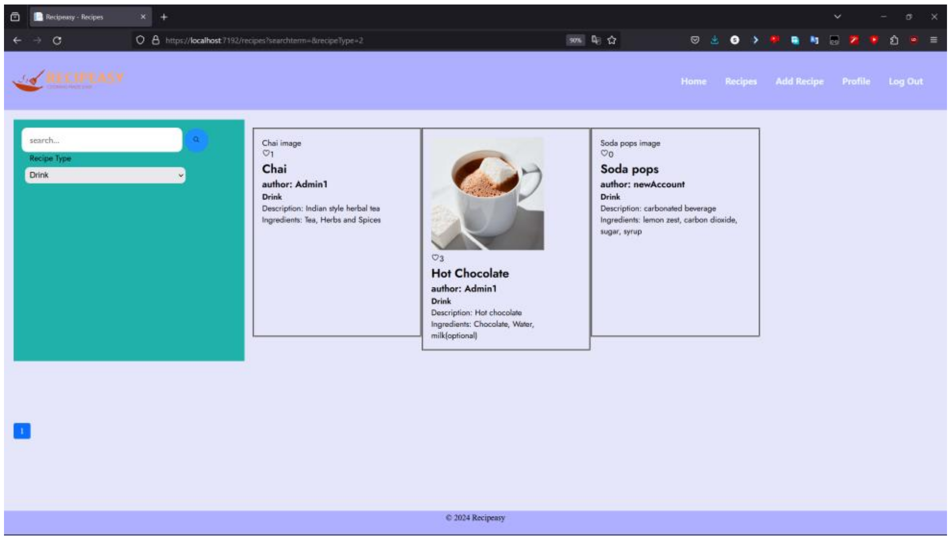
Both user and admin can filter recipes by the type of recipe they want to see

Use case:	**UC-15: Filter by type**
Functional Requirements	**FR-15**
Actor:	Admin, User
Pre-condition:	Be logged in
Main Success Scenario:	1. Actor selects a specific recipe type that desired 2. Actor clicks on the search button 3. System filters recipe by type to get relevant recipes 4. System shows relevant recipes to user 5. End use case
Extensions:	1a)Filter Type is empty 1. System removes all filters and show all recipes 2. End use case



	authorid	TotalLikes	desc	recipetype	Ingredients	prept
▶	1	1	An English style...	Maincourse	Eggs, Bacon, sa...	10
	1	2	Turkish-Dutch f...	Maincourse	French Fries, ve...	15
	3	0	hamburger che...	Maincourse	bun, meat, veg...	10
	3	0	An easy egg	Maincourse	butter, eggs	5
	1004	0	yemeni yellow r...	Maincourse	chicken or lamb...	30
	1	0	indonesian coc...	Maincourse	rice, coconut mi...	10

Picture 15.1 DataGrid only show recipes that is the same recipe type as the filter (Top Right under search bar)



Chai image
♡1
Chai
author: Admin1
Drink
Description: Indian style herbal tea
Ingredients: Tea, Herbs and Spices

♡3
Hot Chocolate
author: Admin1
Drink
Description: Hot chocolate
Ingredients: Chocolate, Water, milk(optional)

Soda pops image
♡0
Soda pops
author: newAccount
Drink
Description: carbonated beverage
Ingredients: lemon zest, carbon dioxide, sugar, syrup

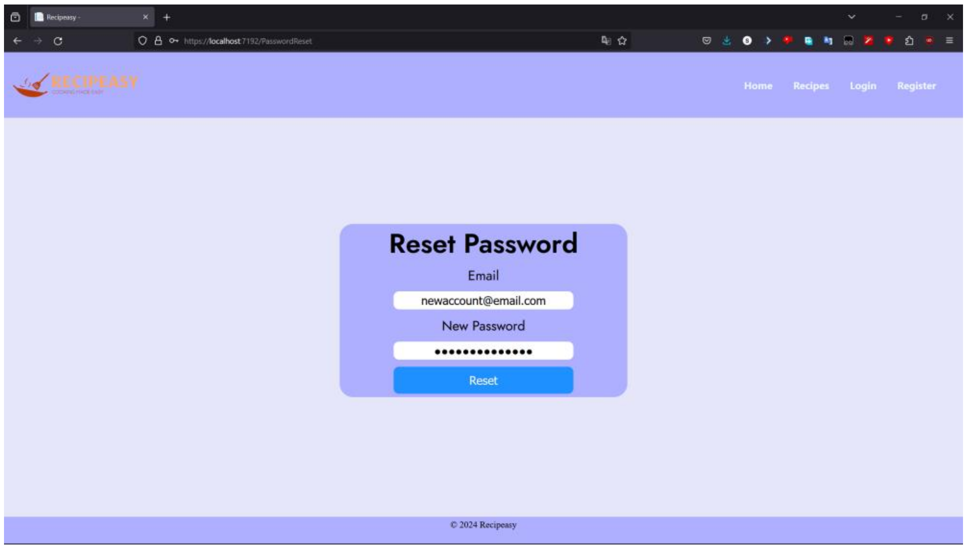
Picture 15.2 Website only show recipes that is the same recipe type as the filter (Left Middle)

UC-16. User changes password

Users can change their password by filling their email address and new Password in the reset password page

Use case:	**UC-16: User changes password**
Functional Requirements	**FR-16**
Actor:	User

Pre-condition:	-
Main Success Scenario:	1. Actor navigates to reset password page 2. System shows necessary fields to fill in 3. Actor fills required fields 4. Actor clicks on submit button 5. System checks user input 6. System resets password 7. Actor is redirected to the Login Page 8. End use case
Extensions:	6a) Data not valid 1. System shows error message 2. Return to step 3



Reset Password

Email: newaccount@email.com

New Password: ••••••••••••••

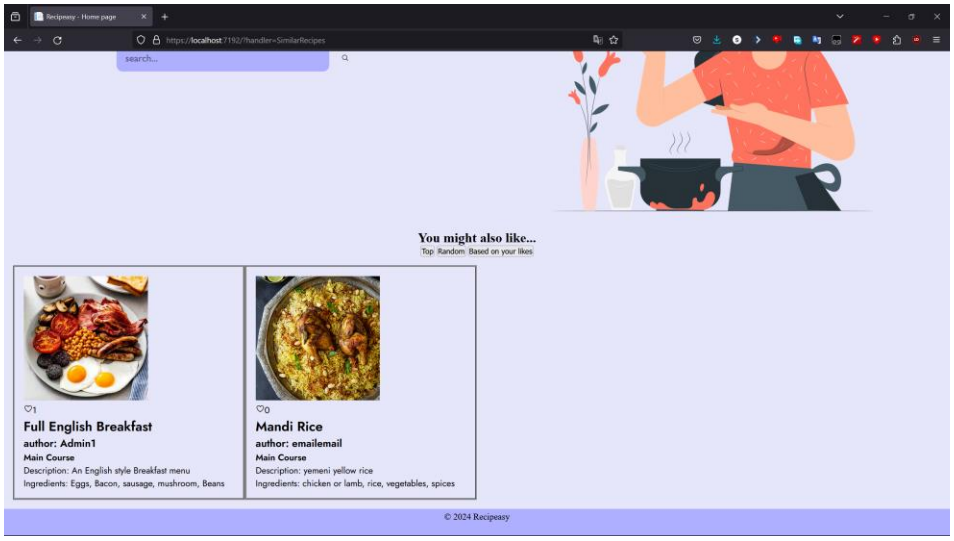
Picture 16.1 Password reset using email

UC-17. System recommends recipes

System will recommend recipes based on the selected preference by the user

Use case:	**UC-17: System recommends recipes**
Functional Requirements	**FR-17**
Actor:	Admin, User
Pre-condition:	Be logged in (optional)

Main Success Scenario:	1. System checks if user is logged in 2. System shows recommendation preferences 3. Actor clicks on Recommend “Based on your likes” button 4. System checks user’s favorite recipes 5. System shows recommended recipes based on user’s preference 6. End use case
Extensions:	1a) User is not logged in 1. System will not show “Based on your likes” button 2. End use case 3a) User selects “Top Recipes” button 1. System will recommend recipes based on the numbers of likes 2. End use case 3b) User selects “Random” button 1. System will recommend random recipes 2. End use case 4a) User don’t have any favorited recipes 1. System will show a message saying “There are no recipes that can be recommended” 2. End use



♡1
Full English Breakfast
author: Admin1
Main Course
Description: An English style Breakfast menu
Ingredients: Eggs, Bacon, sausage, mushroom, Beans

♡0
Mandi Rice
author: emailemail
Main Course
Description: yemeni yellow rice
Ingredients: chicken or lamb, rice, vegetables, spices

© 2024 Recipeasy

Picture 17.1 System shows recipes that can be recommended based on the user’s liking