

**RECIPE ORGANIZER**

**Software Requirement Specification**

– Ho Chi Minh, May 2023 –

# Record of changes

| **Date** | **A\* M, D** | **In charge** | **Change Description** |
| --- | --- | --- | --- |
| May 15 | A | TramLU | Add Functional Recipe display Detail (II.3) |
| May 15 | A | CongCV | Add Functional Detail |
| May 15 | A | KhoaND | Add Functional Detail |
| May 17 | A | PhiDN | Add Functional Detail |
| May 17 | A | DuyVQK | Add Functional Detail |
| May 21 | A | TramLU | Add Functional Detail (II.4.b,c) |
| May 22 | A | CongCV | Add User Authentication |
| May 29 | M | PhiDN | Modified Functional Detail |
| May 30 | M | KhoaND | Double check All |

\*A - Added M - Modified D - Deleted

# 

# Table of Contents

[I. Overview 4](#_30j0zll)

[1. Introduction 4](#_1fob9te)

[2. System Functions](#_3znysh7) 7

[3. Entity Relationship Diagram](#_2et92p0) 9

[II. Functional Requirements](#_tyjcwt) 10

[1. Authentication](#_3dy6vkm) 10

1. [Register account](#_1t3h5sf) 10
2. Confirm email 12
3. [Login account](#_4d34og8) 14
4. Forgot password 15
5. Resend confirm password 17

[2. Meal planning](#_2s8eyo1) 17

1. Create meal plan 17
2. Update meal plan 20
3. View meal plan 22

3. Search 24

1. Search by keyword 24
2. Filter result search 25

4. Recipe display 26

1. Recipe Details 26
2. Collection 27
3. Feedback 28

5. Category 28

1. Add category. 28
2. Manage category 30

6. Category 31

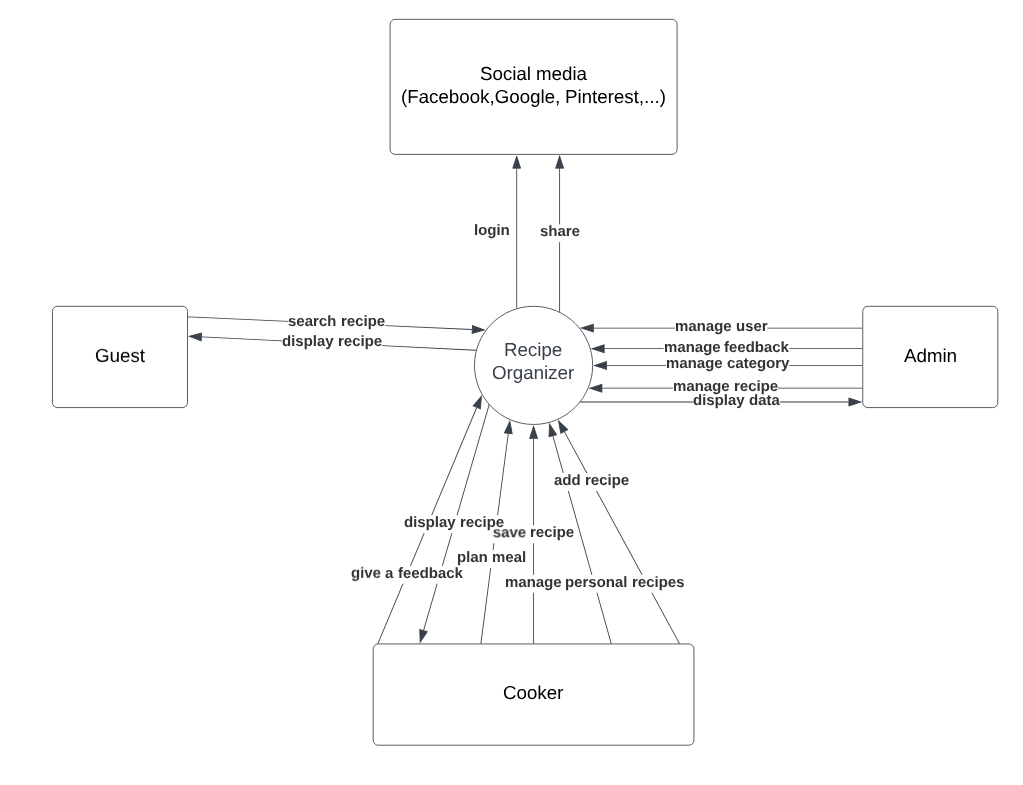
1. View user list . 31
2. Update user status 32
3. View user details 34

# I. Overview

## 1. Introduction

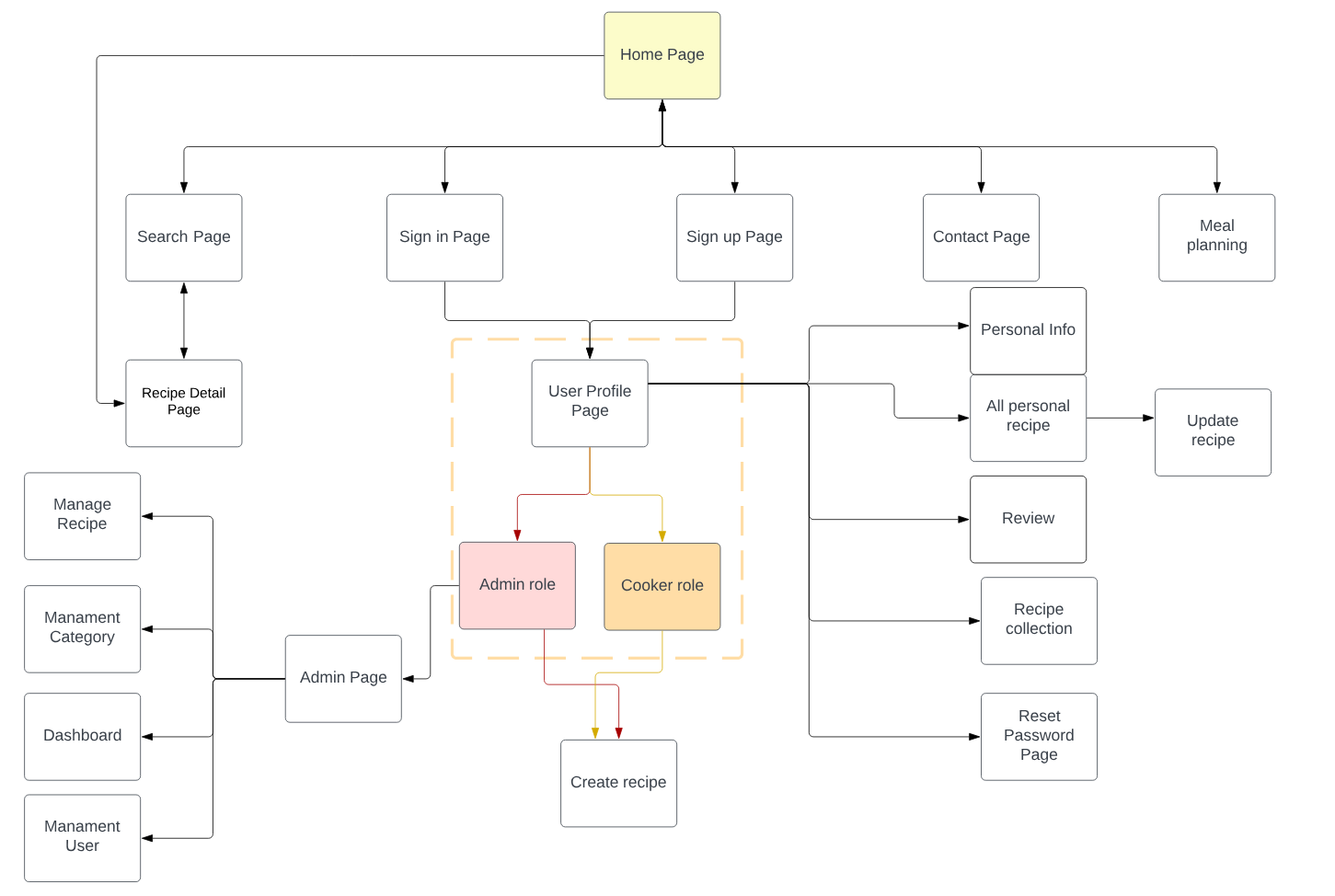
*Many families struggle to locate culinary recipes that are appropriate for their hobbies or eating habits.*

A recipe organizer's purpose is to offer a centralized area for storing and organizing recipes, making meal planning and preparation simpler for users. A recipe organizer may help users save time and decrease the stress of meal planning and preparation by offering features such as search and filtering, meal planning, and recipe sharing. A recipe organizer may also assist users in maintaining a healthy and balanced diet by offering access to nutritional information and the option to filter recipes depending on dietary needs. Overall, the purpose of a recipe organizer is to make meal planning and preparation simpler, as well as to make it easier for consumers to enjoy great and healthful meals.



## 2. System Functions

#### a. Screen Flow



#### b. Screen Details

| **#** | **Feature** | **Screen** | **Description** |
| --- | --- | --- | --- |
| 1 | Authentication | Register account | Register for an account |
| 2 | Authentication | Login Page | Login to the screen |
| 3 | Search | List recipe Page | Search for a recipe by name |
| 4 | Manage User | User List Page | View the list of all accounts in the system |
| 5 | Manage User | Add User Page | Add an account in the system |
| 6 | Manage User | Update User Page | Edit information of an account |
| 7 | Manage User | User Profile Page | View all information of an account |
| 8 | Manage Recipe | Recipe List Page | Show list of all recipe |
| 9 | Manage Recipe | Add Recipe Page | Add a recipe to the system |
| 10 | Manage Recipe | Update Recipe Page | Edit information of a recipe |
| 11 | Meal Planning | Meal Planning Page | Plan meal for a day or more |
| 12 | Recipe Display | Recipe Detail Page | Show details include image, description, ingredients, recipe,... of a dish |

#### c. User Authorization

| **Screen** | **Admin** | **Cooker** | **Guest** |
| --- | --- | --- | --- |
| Home Page | X | X | X |
| User Login | X | X | X |
| User Register | X | X | X |
| Forgot Password | X | X | X |
| Change Password | X | X |  |
| Recipe Recommendations | X | X | X |
| Search and Filter Recipe | X | X | X |
| Save Recipe | X | X |  |
| Recipe Manage (List) | X | X |  |
| Create New Recipe | X | X |  |
| Update Recipe | X | X |  |
| Import Recipe | X | X |  |
| Recipe Sharing | X | X | X |
| Recipe Detail | X | X | X |
| Category Screen | X | X | X |
| Category Manage (List) | X |  |  |
| Add New Category | X |  |  |
| Update Category | X |  |  |
| Meal Planner | X | X |  |
| Settings Screen | X |  |  |
| Profile Screen | X | X |  |
| Tutorial Screen | X | X | X |
| Give a Feedback Screen | X | X | X |
| Recipe Collections | X | X |  |
| Recipe History | X | X |  |
| Recipe Video | X | X | X |
| Cooking Tips | X | X |  |

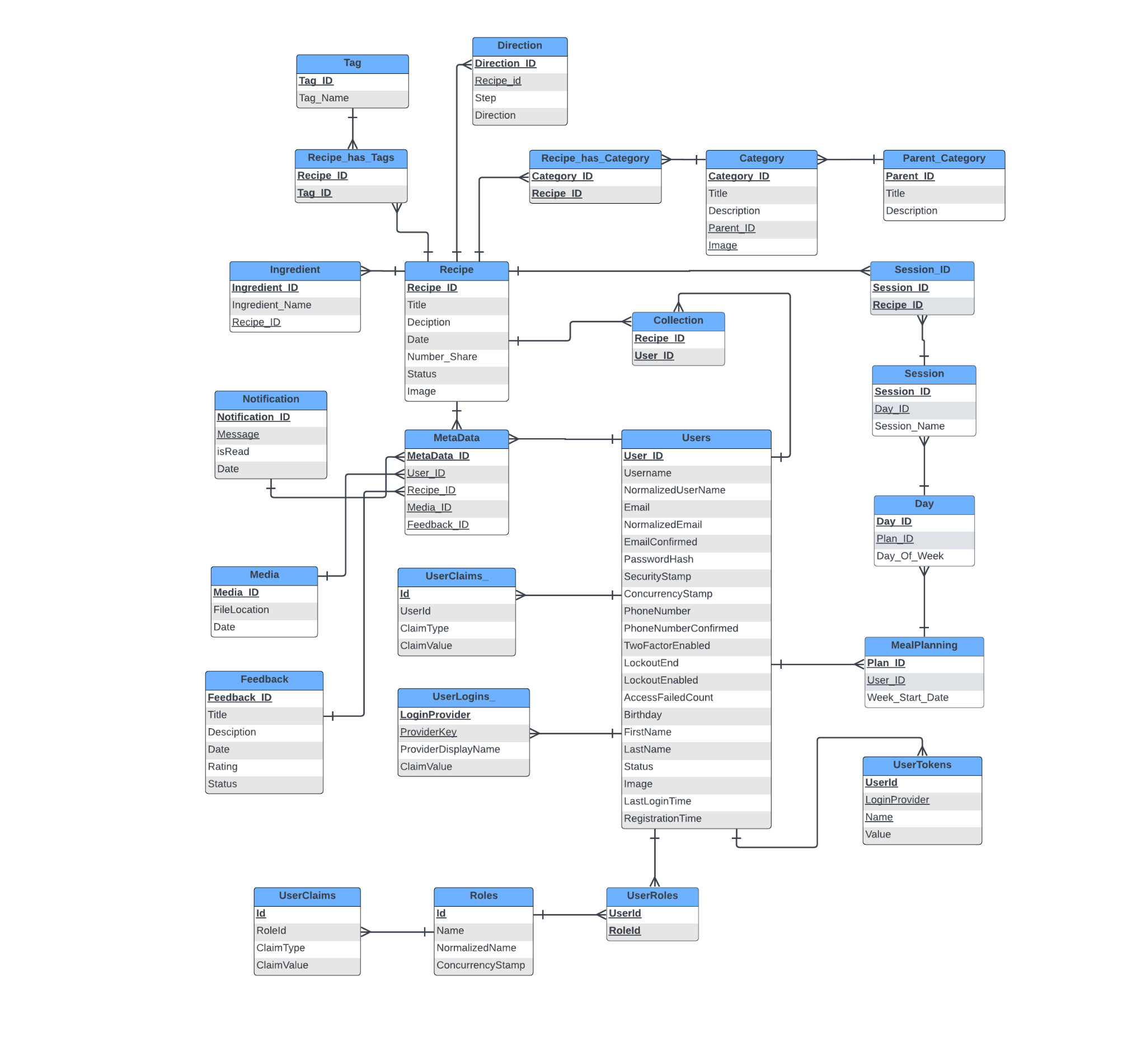
In which:

* **Admin**: The admin is responsible for managing the overall functionality and user access of the Recipe Organizer app.
* **Cooker**: The cooker is a registered user who can create and manage their own recipes and can also create their meal planning within the app.
* **Guest**: The guest is an unregistered user who can browse and search for recipes within the app

#### d. Non-Screen Functions

| **#** | **Feature** | **System Function** | **Description** |
| --- | --- | --- | --- |
| 1 | Remember me | Login | Allows users to stay logged in to the app after they have closed it or shut down their device. The app stores their login credentials |
| 2 | Common | Sign out | Allows users to log out or disconnect from their current user account. Ending their current session and disconnecting from the account. |

## 3. Entity Relationship Diagram

**

**Entities Description**

### 

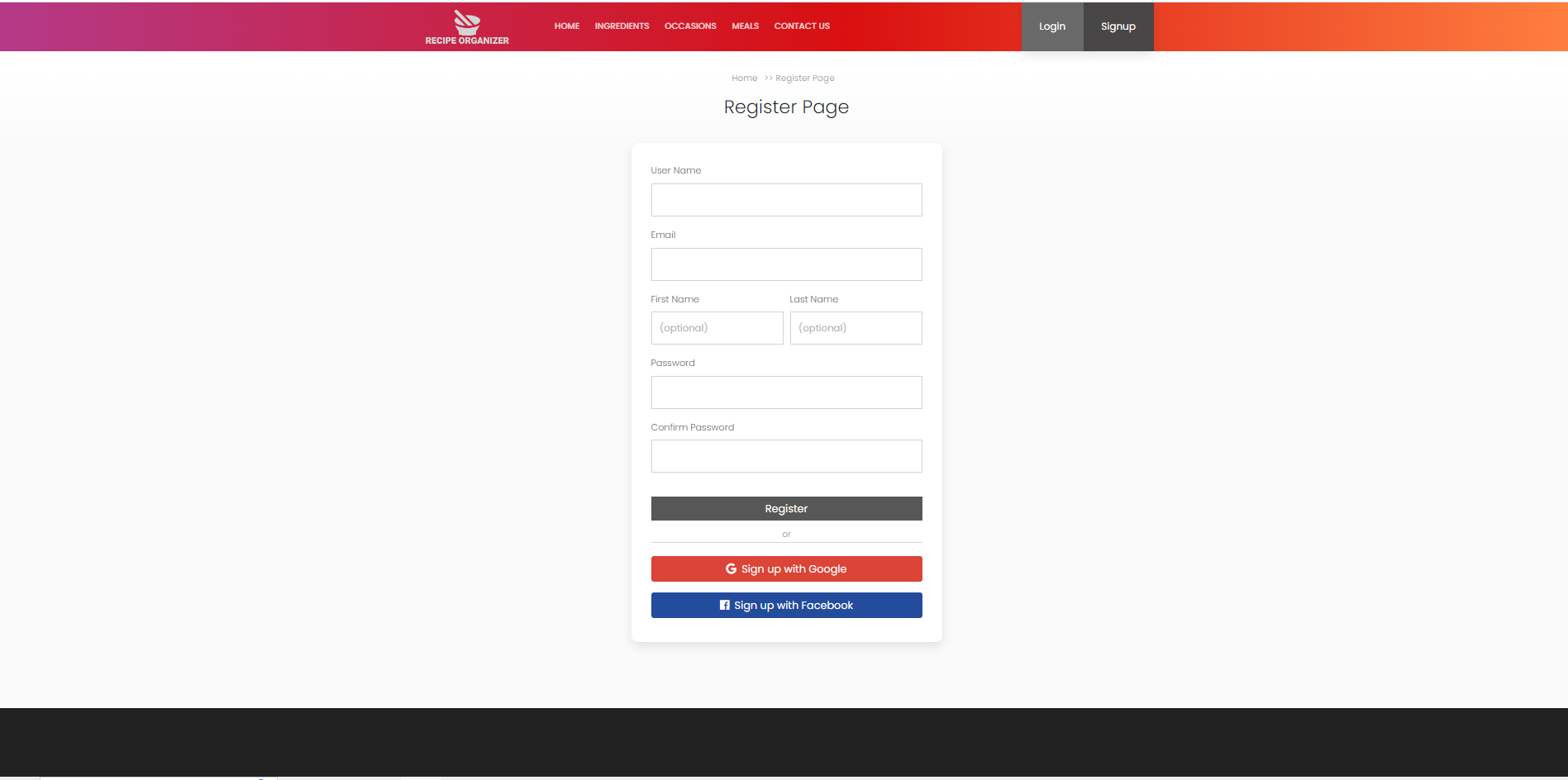
| **No** | **Table** | **Description** |
| --- | --- | --- |
| 01 | AspNetUsers | User-specific information such as user ID, username, email, password hash, security stamp, and other user-related properties. |
| 02 | Role | Role-related information such as role ID and role name. Roles are used to group users with similar permissions or access levels. |
| 03 | UserRole | Establishes a relationship between users and roles. It stores the mapping of user IDs and role IDs, indicating which users belong to which roles. |
| 04 | UserClaims | User claims or assertions about the user's identity. Claims represent specific user attributes, such as name, email, or custom properties. |
| 05 | UserLogins | External logins associated with a user account. It maintains the information about the external provider, such as Google or Facebook, and the user's unique identifier provided by that provider. |
| 06 | UserTokens | Security tokens for password reset and email confirmation processes. It includes fields such as user ID, token type, and token value. |
| 07 | UserRoleClaims | The claims associated with each role. Claims represent additional information or attributes about a role, such as its permissions or any custom claims defined in the application. |
| 08 | Notification | Store notification when user create new recipe and when admin approve or reject recipe |
| 09 | Recipe | Store recipe detail includes Title, description, date, status, number share, avg rating |
| 10 | Category | category include recipe one a type recipe, make user see easy by type |
| 11 | Recipe\_has\_Category | Map 1-n;n-1 from recipe,category, its net n-n |
| 12 | Collection | Store recipes that the user has saved |
| 13 | Feedback | Its is QA, comment form user it is possible vision all viewer |
| 14 | MealPlanning | The specific information related to a meal plan, such as the selected recipes, their quantities, cooking instructions, and any additional notes or preferences provided by the user |
| 15 | Media | Store link (location) of images |
| 16 | Parent\_Category | The parent item of the category of recipe |
| 17 | Day | Store day of plan |
| 18 | Session | Store session of day |
| 19 | SessionHasRecipe | Store recipes of session |
| 20 | MetaData | Map 1-n from recipe-metadata, user-metadata, feedback-metadata, media-metadata, notification-metadata |
| 21 | Tag | It’s store hashtag for recipe |
| 22 | Ingredient | Store ingredients or each recipe |
| 23 | Direction | Store step-by-step direction of recipe |
| 24 | Recipe\_has\_Tags | Map m-m from recipe-tag |

# II. Functional Requirements

## 1. Authentication

### a. Register account

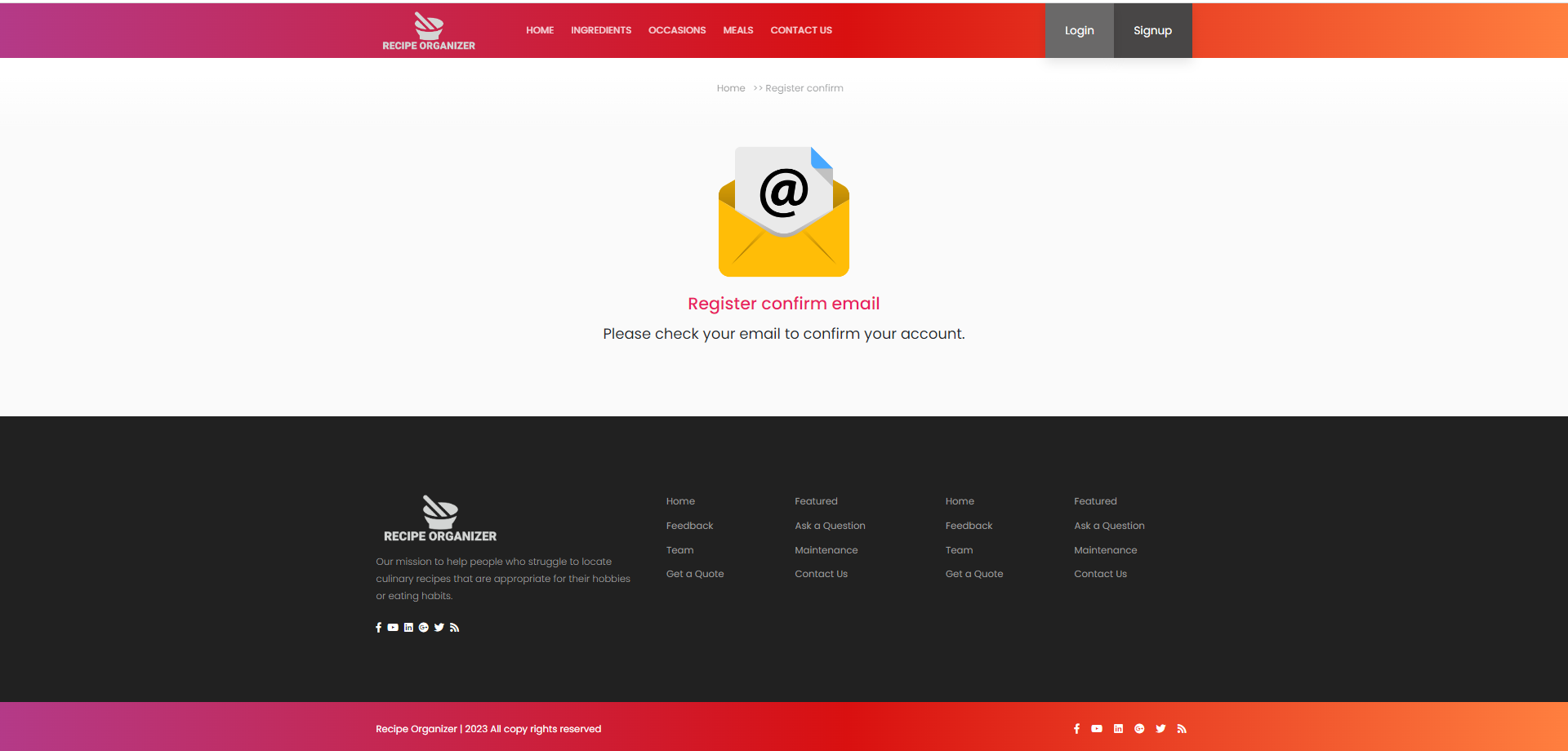
* Function Trigger: This function is triggered when a user initiates the registration process by accessing the registration page or clicking on the "Sign Up" button.
* Function Description:
  + Actors/Roles: New User, Recipe Organizer Application, Email Service
  + Purpose: The purpose of this function is to allow new users to create an account in the Recipe Organizer app and verify their email address through a confirmation process.
  + Interface: The user will be presented with a registration form where they can enter their desired username, email, and password.
  + Data Processing:
    - The function will validate the user's input data, including username, email, and password, to ensure correctness and adherence to any specified requirements.
    - If the data passes validation, a new user record will be created and stored in the database with the provided information.
    - An email with a confirmation link will be sent to the user's provided email address for email verification.
* Screen Layout:

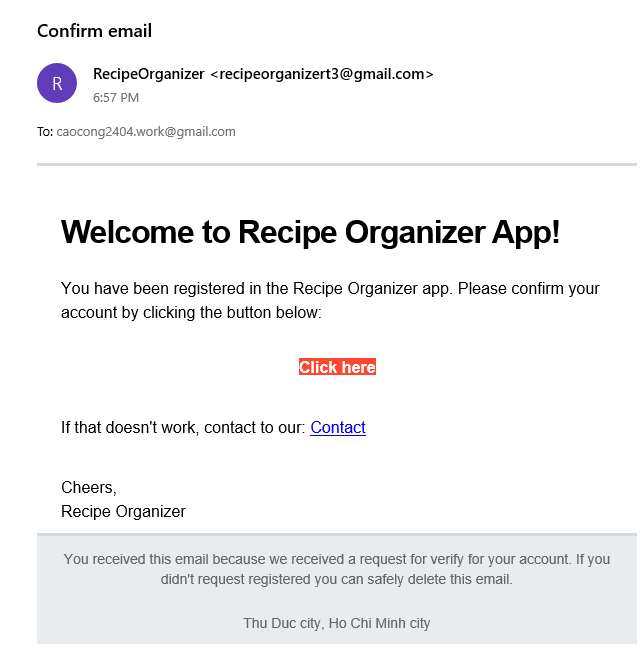


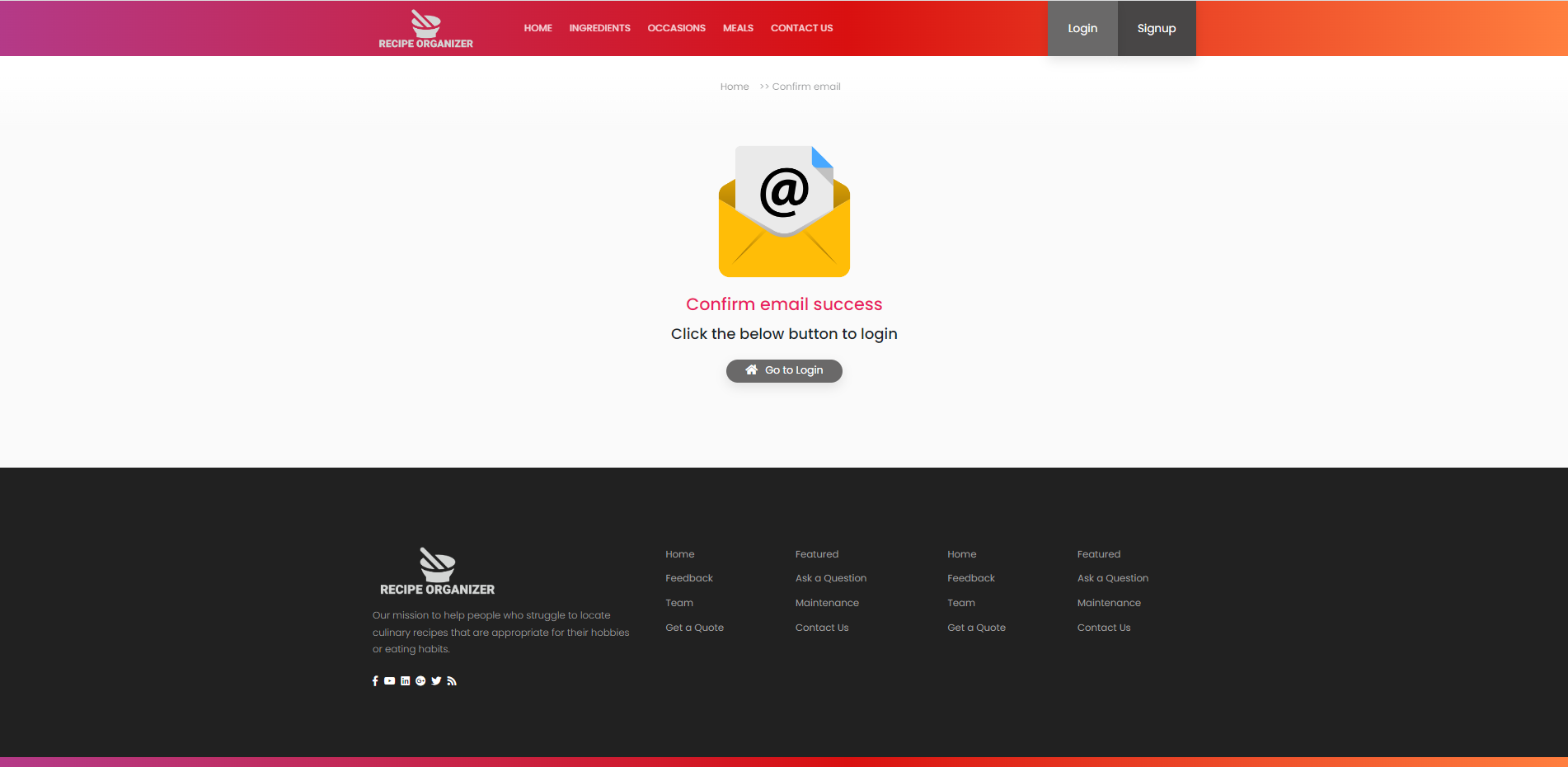
* Function Details:
  + Data:
    - Username: The desired username entered by the user.
    - Email: The email address entered by the user.
    - Password: The password entered by the user.
  + Validation:
    - Username and email uniqueness validation: The function will check if the entered username and email are unique and not already registered by another user.
    - Password complexity validation: The function may enforce password complexity requirements, such as minimum length, special characters, or alphanumeric combinations.
  + Business Logic:
    - If the entered username and email are unique, the function will create a new user record in the database with the provided data.
    - The function will generate a unique confirmation token and associate it with the user account.
    - An email containing the confirmation link with the generated token will be sent to the user's provided email address using an email service.
* Functionality:
  + Normal Case: The user successfully completes the registration process and receives an email with a confirmation link.
  + Abnormal Cases:
    - Duplicate Username or Email: If the entered username or email is already registered by another user, the function will display an error message prompting the user to choose a different username or email.
    - Invalid Password: If the entered password does not meet the specified complexity requirements, the function will display an error message and ask the user to provide a valid password.

### b. Confirm email

* Function Trigger: This function is triggered when a user clicks on the confirmation link received in their email.
* Function Description:
  + Actors/Roles: Registered User, Recipe Organizer Application
  + Purpose: The purpose of this function is to confirm the user's email address by validating the provided confirmation token.
  + Interface: The user will click on the confirmation link in the email, which will redirect them to a confirmation page in the Recipe Organizer app.
  + Data Processing:
    - The function will validate the confirmation token provided in the URL against the user's stored confirmation token in the database.
    - If the token is valid and matches the user's record, the user's email address will be marked as confirmed in the database.
* Screen Layout:



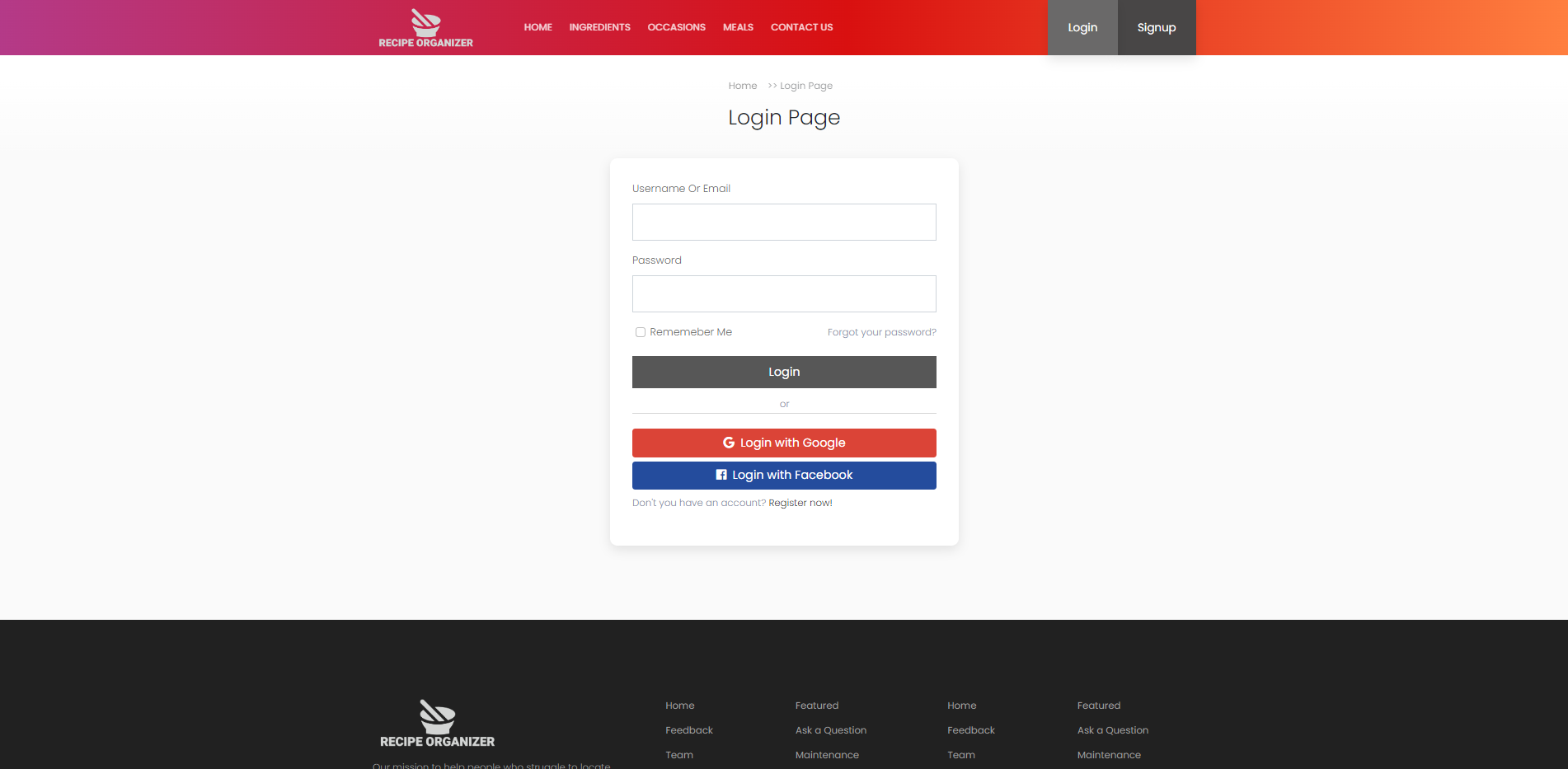




* Function Details:
  + Data:
    - Confirmation Token: The unique token generated during the user registration process and included in the confirmation link.
  + Validation:
    - Confirmation Token Validation: The function will compare the provided token against the user's stored confirmation token in the database to ensure their match.
  + Business Logic:
    - If the provided token is valid, the user's email address will be marked as confirmed, allowing them to access their account and use the Recipe Organizer app.
* Functionality:
  + Normal Case: The user clicks on the confirmation link, and their email address is successfully confirmed.
  + Abnormal Cases:
    - Invalid or Expired Token: If the provided confirmation token is invalid or has expired, the function will display an error message notifying the user and provide instructions for resending the confirmation email.

### c. Login account

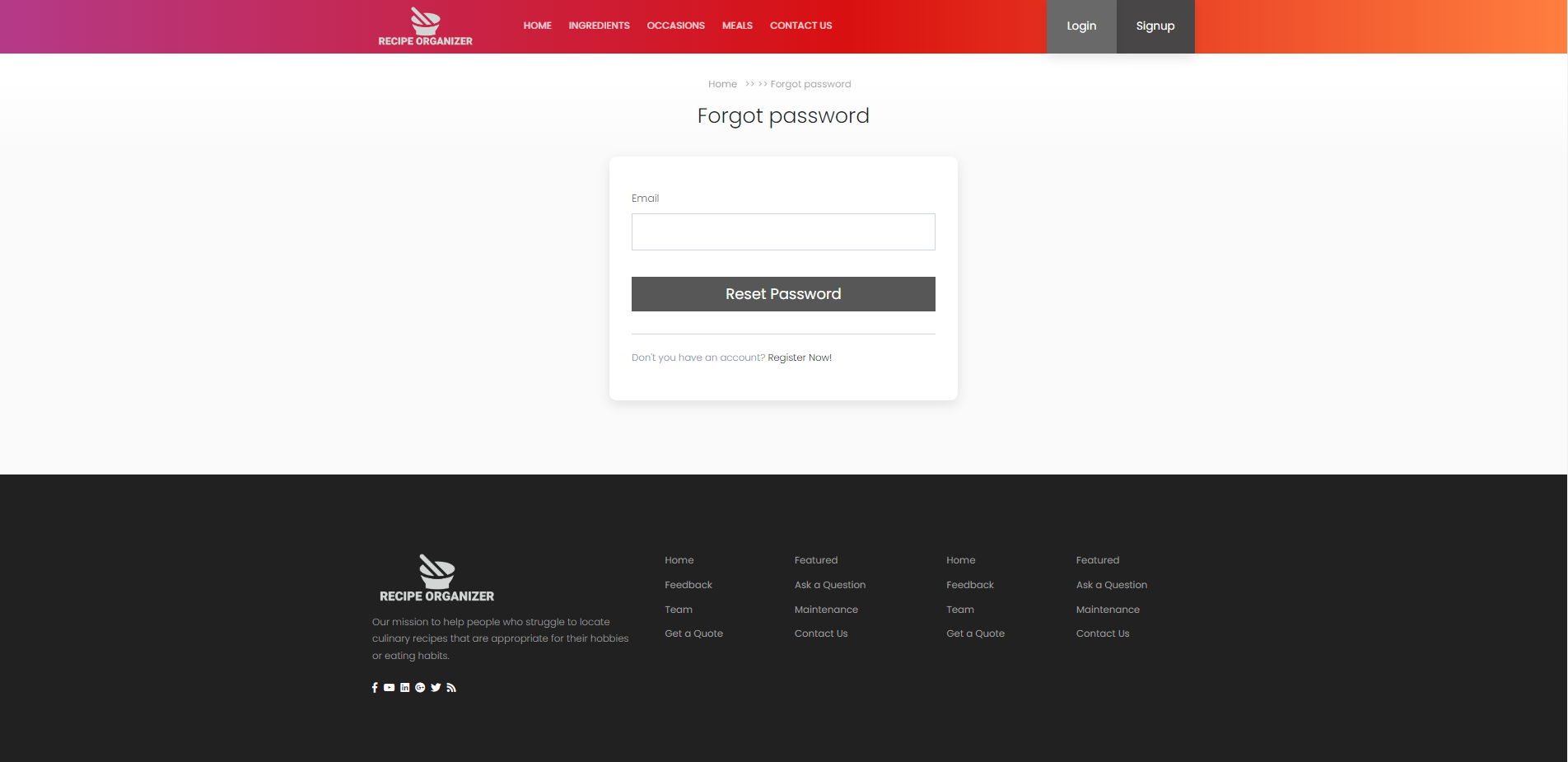
* Function Trigger: This function is triggered when a user attempts to log in to their Recipe Organizer account.
* Function Description:
  + Actors/Roles: User, Recipe Organizer Application
  + Purpose: The purpose of this function is to authenticate the user's login credentials (username or email and password) and grant them access to their account if the credentials are correct.
  + Interface: The user will be prompted to enter their username or email and password on the login page of the Recipe Organizer application. If the user's credentials are correct, they will be granted access to their account.
  + Data Processing:
    - The function will validate the user's login credentials by checking them against the stored user data in the ASP.NET Identity database.
    - The function will log the user in and grant them access to their account if their credentials are correct.
* Screen Layout:



* Function Details:
  + Data:
    - Username/Email: The string entered by the user to identify their account.
    - Password: The string entered by the user to authenticate their account.
  + Validation:
    - Credentials Validation: The user's entered username or email and password will be validated to ensure that they match the stored user data in the ASP.NET Identity database.
  + Business Logic:
    - If the user's entered credentials match the stored user data, the function will log the user in and grant them access to their account.
    - If the user's entered credentials do not match the stored user data, the function will return an error message and prompt the user to try again.
* Functionality:
  + Normal Case: The user will be able to log in with their correct username and password, and will be granted access to their account.
  + Abnormal Cases:
    - Incorrect Username or Password: If the user's entered username or password is incorrect, the function will return an error message and prompt the user to try again.

### d. Forgot password

* Function Trigger: This function is triggered when a user requests to reset their password because they have forgotten it.
* Function Description:
  + Actors/Roles: User, Recipe Organizer Application
  + Purpose: The purpose of this function is to allow users to reset their password in case they have forgotten it.
  + Interface: The user will navigate to the "Forgot Password" page or click on a "Forgot Password" link on the login page of the Recipe Organizer application.
  + Data Processing:
    - The function will prompt the user to enter their email address associated with their account.
    - The function will validate the provided email address to ensure it matches an existing user account.
    - If the validation is successful, the function will generate a password reset token and send a password reset link to the user's email address.
* Screen Layout:



* Function Details:
  + Data:
    - Email: The user's email address associated with their account.
  + Validation:
    - Email Validation: The function will validate the provided email address to ensure it matches an existing user account.
  + Business Logic:
    - If the provided email address passes the validation, the function will generate a password reset token and send a password reset link to the user's email address.
    - If the provided email address does not match any existing user account or fails the validation, the function will return an error message and prompt the user to provide a valid email address.
* Functionality:
  + Normal Case: The user will successfully request a password reset by providing a valid email address associated with their account. The function will generate a password reset token and send a password reset link to the user's email address.
  + Abnormal Cases:
    - Invalid Email Address: If the provided email address does not match any existing user account or fails the validation, the function will return an error message and prompt the user to provide a valid email address.

### e. Resend confirm account

* Function Trigger: This function is triggered when a user requests to resend the email confirmation after registering a new account.
* Function Description:
  + Actors/Roles: User, Recipe Organizer Application
  + Purpose: The purpose of this function is to allow users to request a resend of the email confirmation to verify their registered email address.
  + Interface: The user will navigate to the "Resend Email Confirmation" page or click on a "Resend Confirmation Email" link in their account settings within the Recipe Organizer application.
  + Data Processing:
    - The function will retrieve the user's email address associated with their account.
    - The function will generate a new email confirmation token and send a confirmation email containing the token to the user's email address.
* Screen Layout: N/A (non-screen function)
* Function Details:
  + Data:
    - Email: The user's email address associated with their account.
  + Business Logic:
    - The function will retrieve the user's email address from their account.
    - The function will generate a new email confirmation token and send a confirmation email containing the token to the user's email address.
* Functionality:
  + Normal Case: The user will successfully request a resend of the email confirmation. The function will generate a new email confirmation token and send a confirmation email to the user's email address.
  + Abnormal Cases: None

## 2. Meal planning

### a. Create meal plan

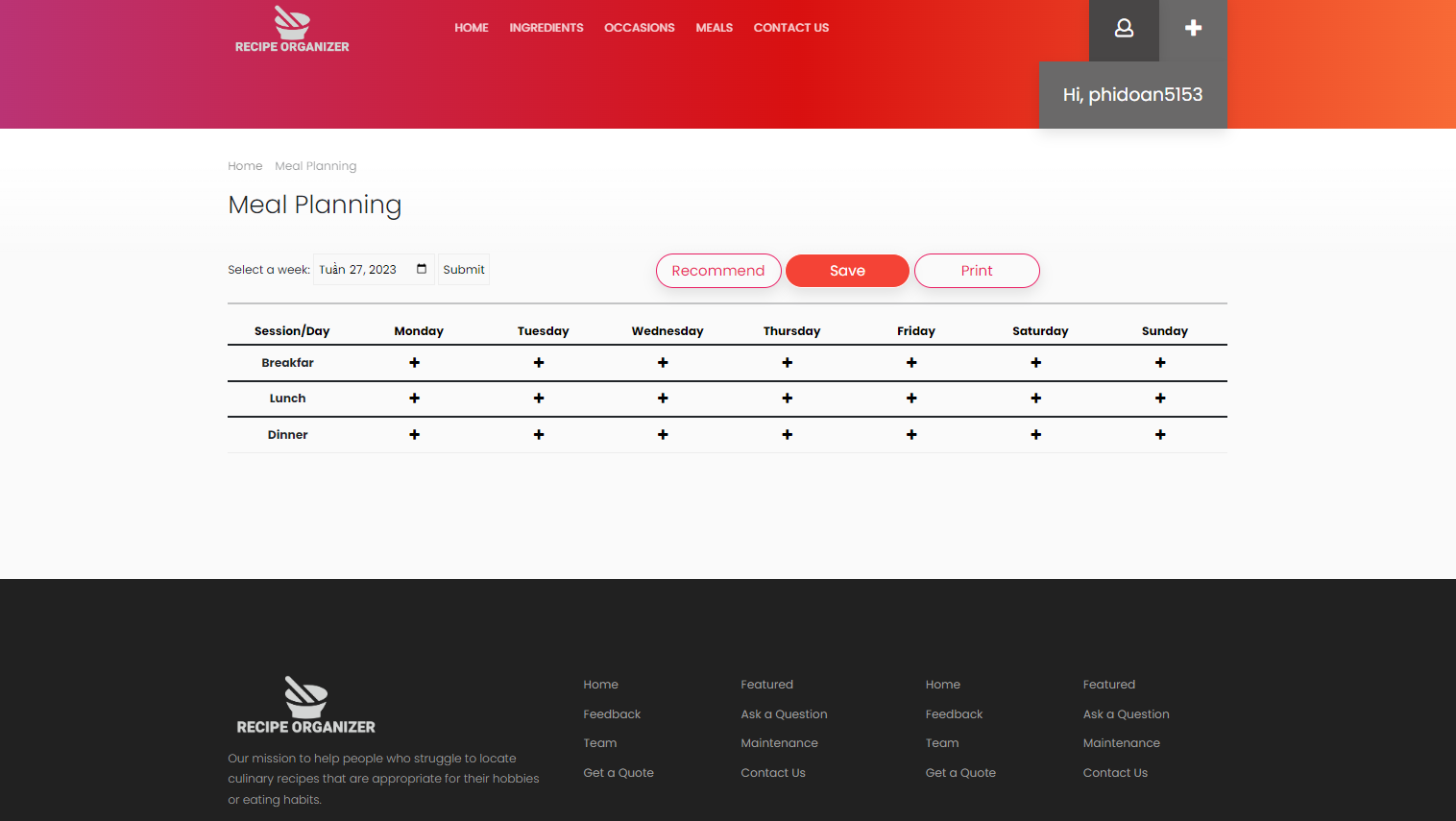
* Function Trigger: This function can be triggered by the user navigating to the "Meal Planning" section within the Recipe Organizer app or by selecting a "Create Meal Plan" button within the app.
* Function Description:
  + Actors/Roles:
    - Cooker: The person who is using the Recipe Organizer app to plan their meals.
    - System handle: The Recipe Organizer application that allows the user to plan their meals and create meal plans.
  + Purpose: The purpose of this function is to enable the user to create a customized meal plan for a specific period, such as a week or a month.
  + Interface: The user will be presented with a screen where they can specify the meals for each day. The interface may include options to search for recipes or add custom recipes.
  + Data Processing:
    - The function will retrieve the user's saved recipes from the Recipe Organizer app's database.
    - The function will store the created meal plan, including the selected dates and associated recipes, in the app's database.
    - The function will display the created meal plan to the user.
* Screen Layout:
  + The "Meal Planning" screen will include fields and controls for selecting the duration of the meal plan and specifying the meals for each day. The screen layout may look similar to the following mockup:

### 

* Function Details:
  + Data:
    - Meal Plan: The created meal plan, including the selected dates and associated recipes for each meal.
    - Saved Recipes: The user's saved recipes retrieved from the app's database.
  + Validation:
    - Date Range Validation: The function will validate that the selected date range is valid and within an acceptable range, such as a week or a month.
    - Recipe Selection Validation: The function will validate that the selected recipes exist in the user's saved recipes.
  + Business Logic:
    - Selecting Recipes: The function will enable the user to choose recipes for each meal within the specified dates. This can be done by searching for recipes or adding custom recipes.
    - Saving Meal Plan: The function will save the created meal plan, including the selected dates and associated recipes, to the app's database.
  + Functionality:
    - Normal Case: The user will be able to create a customized meal plan by choosing recipes for each meal. The created meal plan will be saved and displayed to the user.
    - Abnormal Cases:
    - Invalid Recipe Selection: If the user selects recipes that do not exist in their saved recipes, the function will return an error message and prompt the user to select valid recipes.

### b. Save m**eal** p**lan**

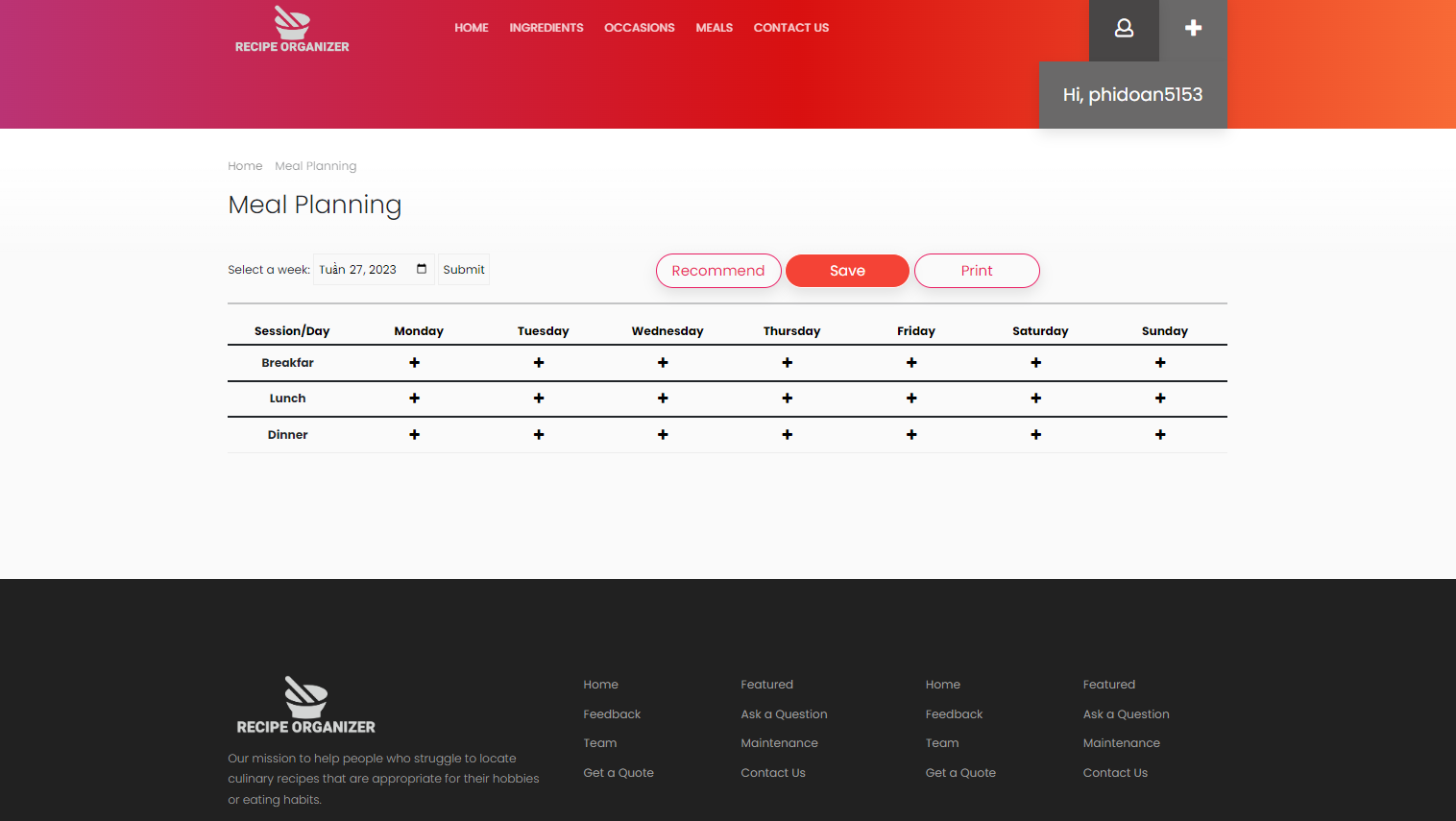
* Function Trigger: This function can be triggered when the user selects the option to edit a specific meal plan from the Meal Planning section.
* Function Description:
  + Actors/Roles:
    - Cooker: The person who is using the Recipe Organizer app and wants to edit their meal plan.
    - System handle: The Recipe Organizer application that allows the user to plan meals and store meal plans.
  + Purpose: The purpose of this function is to enable the user to make changes to an existing meal plan, such as modifying the selected recipes, adjusting meal times, or adding/removing meals.
  + Interface: The user will be presented with a screen that displays the existing meal plan, including the associated recipes and meal details. The interface will allow the user to make modifications, such as selecting different recipes, changing meal times, or adding/removing meals.
  + Data Processing:
    - The function will retrieve the selected meal plan from the app's database.
    - The function will fetch the necessary recipe and meal information from the database for display and modification.
    - The function will update the meal plan and associated recipes in the app's database based on the user's modifications.
* Screen Layout:
  + The "Update Meal Plan" screen will display the existing meal plan and provide options for modifying the recipes, meal times, or adding/removing meals. The screen layout may look similar to the following mockup:



* Function Details:
  + Data:
    - Meal Plan: The existing meal plan to be edited.
    - Associated Recipes: The recipes associated with each meal in the meal plan.
  + Validation:
    - Recipe Availability: The function will validate that the selected recipes are available and exist in the user's saved recipes.
    - Meal Time Validation: The function will validate that the modified meal times are within acceptable ranges and do not conflict with each other.
  + Business Logic:
    - Fetching Meal Plan: The function will retrieve the selected meal plan from the app's database.
    - Retrieving Recipe Information: The function will fetch the corresponding recipe information for each meal in the meal plan to display and allow modification.
    - Updating Meal Plan: The function will update the meal plan and associated recipes in the app's database based on the user's modifications.
  + Functionality:
    - Normal Case: The user will be able to make modifications to the existing meal plan, such as changing recipes, adjusting meal times, or adding/removing meals. The updated meal plan will be saved in the app's database.
  + Abnormal Cases:
    - Conflicting Meal Times: If the user tries to set conflicting meal times or times that are outside acceptable ranges, the function will return an error message and prompt the user to select valid meal times.

### c. View meal plan

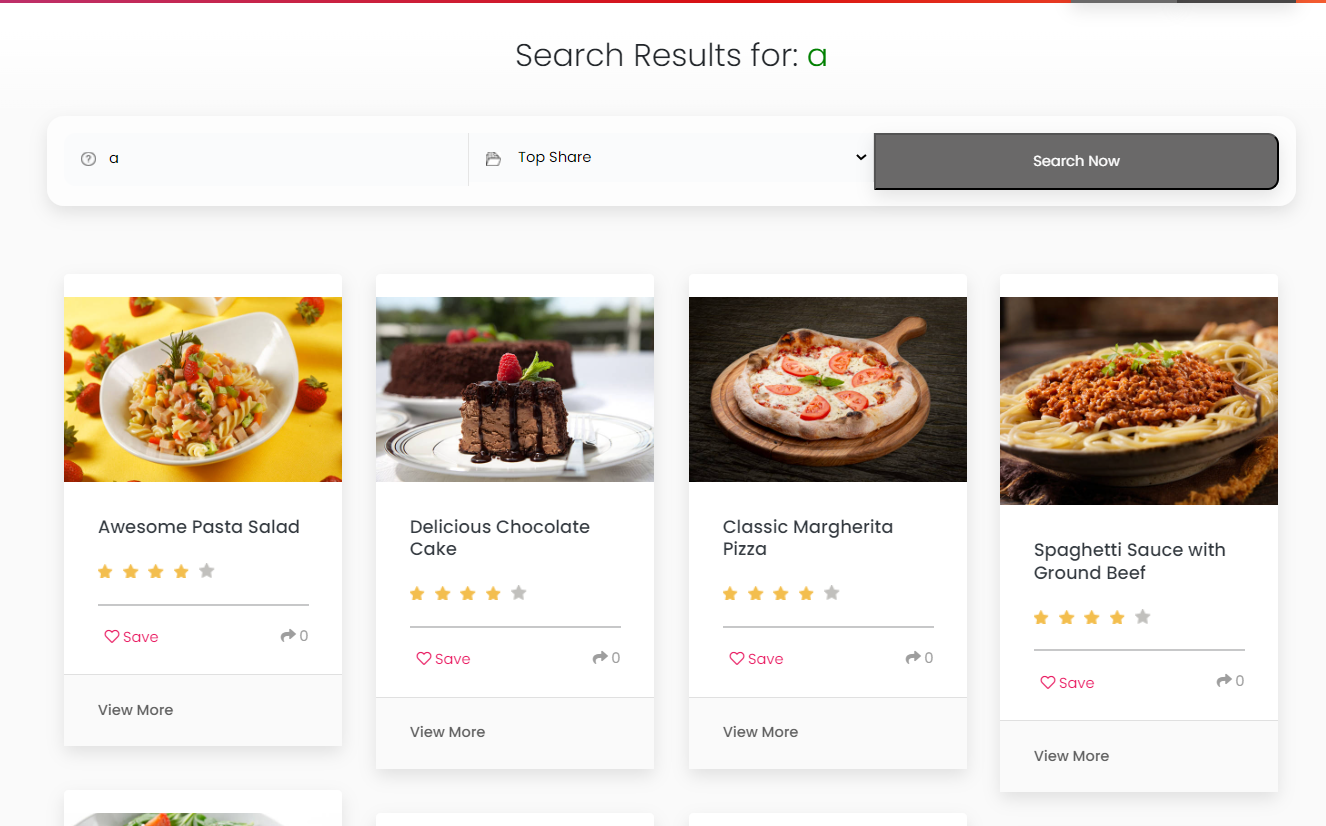
* Function Trigger: This function can be triggered when the user selects a specific meal plan from the Meal Planning section or when the app automatically displays the current meal plan based on the date.
* Function Description:
  + Actors/Roles:
    - Cooker: The person who is using the Recipe Organizer app and wants to view their meal plan.
    - App: The Recipe Organizer application that allows the user to plan meals and store meal plans.
  + Purpose: The purpose of this function is to display the user's selected meal plan for a specific date.
  + Interface: The user will be presented with a screen that shows the selected meal plan for the chosen date or range of dates. The interface may include details such as the meal names and any additional notes or instructions.
  + Data Processing:
    - The function will retrieve the selected meal plan from the app's database based on the specified date.
  + The function will fetch the corresponding recipe information from the database to display along with the meal plan.
* Screen Layout:
  + The "View Meal Plan" screen will display the selected meal plan, including the associated recipes and any additional details. The screen layout may look similar to the following mockup:



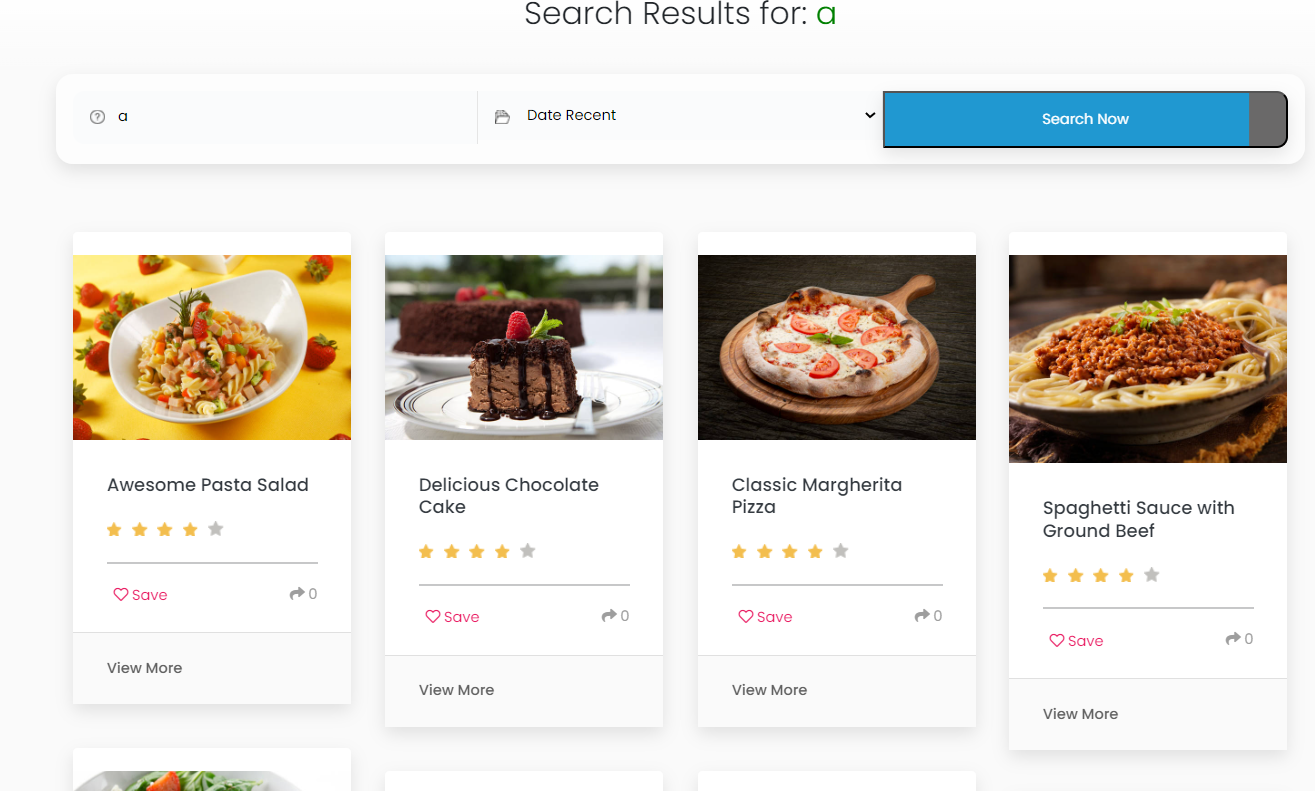
* Function Details:
  + Data:
    - Meal Plan: The selected meal plan to be displayed.
    - Associated Recipes: The recipes associated with each meal in the meal plan.
  + Validation:
    - Date Range Validation: The function will validate that the selected date is valid and within the available meal plans.
* Business Logic:
  + Fetching Meal Plan: The function will retrieve the selected meal plan from the app's database based on the specified date.
  + Retrieving Recipe Information: The function will fetch the corresponding recipe information for each meal in the meal plan to display along with the meal plan.
* Functionality
  + Normal Case: The user will be able to view the selected meal plan, including the associated recipes, for the chosen date.
  + Abnormal Cases:
  + Invalid Date Selection: If the user selects a date or range of dates that does not have a corresponding meal plan, the function will return an error message and prompt the user to select a valid date.
  + Missing Recipe Information: If there are missing recipe details or the associated recipes cannot be found in the database, the function will display a placeholder or indicate that the recipe information is unavailable.

## 3. Search

### a. Search keyword

* Function Trigger: This function is triggered when a user wants to search for items on the platform. The user can access it through the search bar or by navigating to the items section of the website or app.
* 
* Function Description: The Item Search and Filtering function allows users to search for items based on various parameters such as ingredients, prep time, difficulty level, and so on. The function provides a basic search capability as well as sophisticated filtering tools, which enable users to narrow down search results based on specified criteria.
  + Actors/Roles:
    - The primary actor for this function is the user who wants to search for an item. The system also plays a role in providing the search and filtering capabilities.
    - Interface:
    - The interface for this function includes a search bar and various search filters such as ingredient, prep time, difficulty level, and so on. The search results are displayed in a list format, and each item is accompanied by a thumbnail image, title, and a brief description.
  + Data Processing:
    - The function processes a large amount of item data to provide search results. The items are stored in a database and indexed by various parameters such as ingredients, prep time, and difficulty level. The function retrieves the relevant items based on the user's search query and applies the specified filters to narrow down the results.
  + Screen Layout:
    - The mockup prototype for the Item Search and Filtering screen might include a search bar at the top of the page and a set of filter options on the left-hand side. The search results are displayed in the main area of the screen in a list format, with each item accompanied by a thumbnail image, title, and brief description.
* Function Details:
  + Basic Search: The basic search capability allows users to search for items using keywords. The function retrieves items that match the search query and displays them in the search results list.
  + Sophisticated Filtering: The sophisticated filtering tools allow users to narrow down search results based on various parameters such as ingredients, prep time, and difficulty level. The function retrieves items that meet the specified criteria and displays them in the search results list.
  + Data Validation: The function validates user input to ensure that it is in the correct format and within acceptable ranges. For example, if the user enters a prep time of less than zero or more than 24 hours, the function displays an error message.
  + Business Logic: The function applies business logic to ensure that only relevant items are displayed in the search results. For example, if the user searches for vegetarian items, the function will not display items that contain meat.
  + Functionalities:
    - Normal Cases: The function retrieves relevant items based on the user's search query and displays them in the search results list. The user can then view the item details, including the ingredients, instructions, and nutritional information.
    - Abnormal Cases: If the user enters an invalid search query or selects an invalid filter option, the function displays an error message. If the function is unable to retrieve any items that match the specified criteria, it displays a message indicating that no results were found.

### b. Filter search result

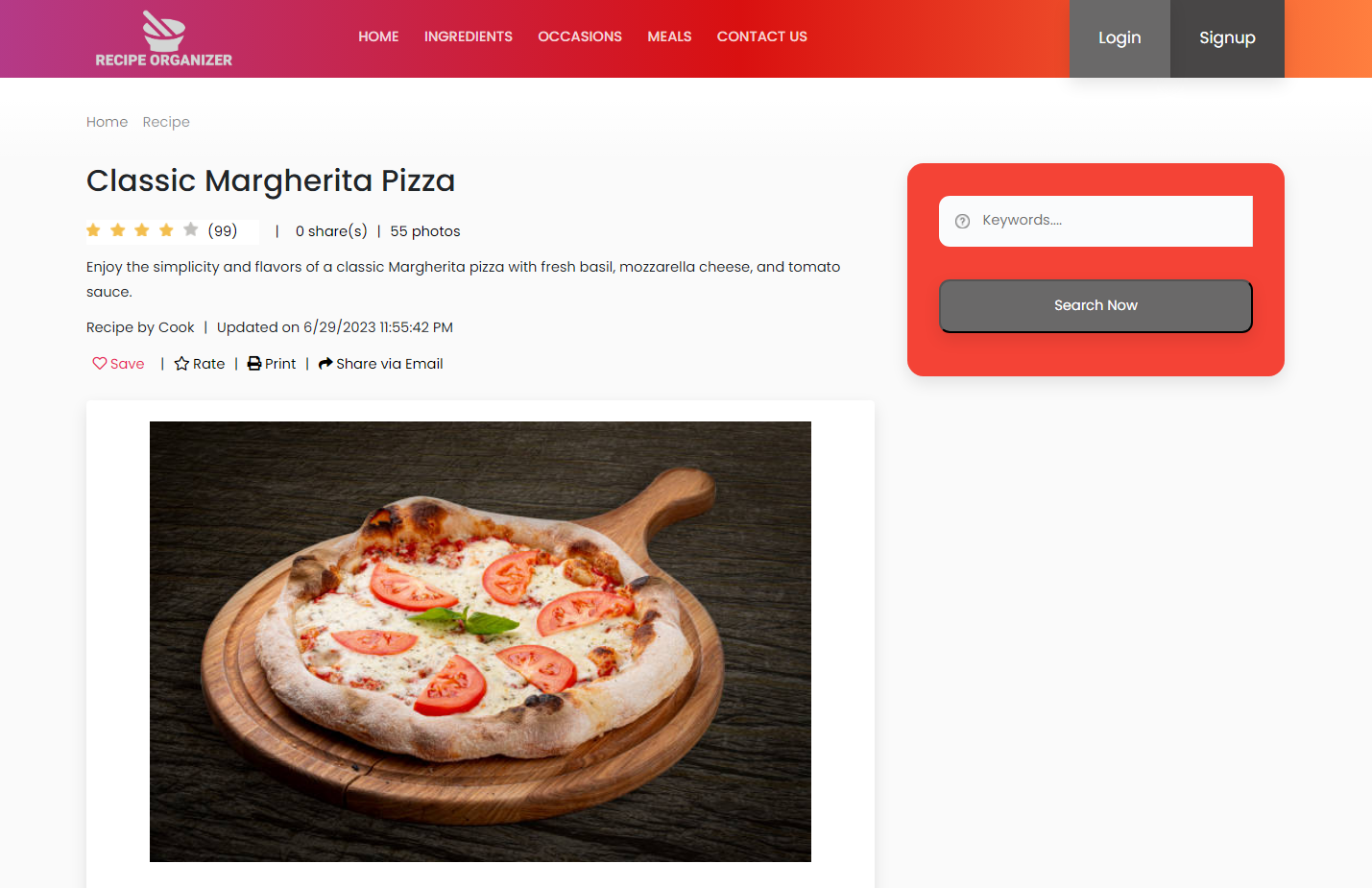
* Function Trigger: The function is triggered when the user clicks on the "Filter" button on the recipe search page.
* Function Description: The purpose of this function is to allow users to filter their recipe search results based on various parameters such as ingredients, prep time, difficulty level, and other criteria. Users can select one or more filters to narrow down their search results. The interface for this function includes a menu of filter options with checkboxes or dropdown menus for selecting specific parameters. The data processing involves querying the recipe database and displaying the search results based on the selected filters.
* 
* Screen Layout: The screen layout for this function would include a search bar, a list of search results, and a menu of filter options on the left-hand side of the screen. The mockup prototype for this screen might resemble a typical e-commerce product search page, with a menu of filter options on the left and a list of search results in the centre of the page.
* Function Details: When the user selects one or more filters, the function validates the selected data to ensure that it meets the required format and criteria. For example, if the user selects "gluten-free" as a filter for ingredients, the function will only display recipes that do not contain gluten. The business logic for this function involves querying the recipe database for recipes that meet the selected criteria and then displaying the search results on the screen. In abnormal cases, the function may display an error message if the user selects an invalid filter option or if there are no search results that meet the selected criteria.

## 4. Recipe display

### a. Recipe details

The functional requirement for the Recipe Display function is to provide users with a detailed view of a recipe's information and instructions. This feature allows users to access comprehensive details about a specific recipe, enhancing their cooking experience and enabling them to follow the recipe accurately.

* Function trigger: The recipe display feature is triggered when a user selects a specific recipe from their collection or performs a search for a particular recipe within the Recipe Organizer application.
* Function description:
  + Actors/Roles: User (included Cooker and Guest)
  + Purpose: To provide users with detailed information and instructions for a selected recipe.
  + Interface: The recipe displays screen/interface.
  + Data Processing: Retrieving and presenting recipe data from the application's database.
* Screen layout: Mockup prototype of the recipe display screen



* Function Details:
  + Data: The function retrieves recipe details such as the recipe title, description, cooking time, serving size, difficulty level, and dietary restrictions.
  + Validation: The system ensures that the selected recipe exists in the user's collection or search results before displaying the details.
  + Business Logic: The function displays a list of ingredients required for the recipe, including their quantities and units of measurement.
  + Functionalities:
    - Display Instructions: The function presents step-by-step instructions on how to prepare the recipe, ensuring clarity and ease of comprehension.
    - Print or Save: Users have the option to print the recipe details or save them for later reference, enabling convenient offline access.
    - Images: If provided, the function includes images or visual representations of the recipe to help users visualise the final dish.
    - Nutritional Information: If available, the function displays nutritional information related to the recipe, such as calorie count, macronutrient composition, and allergen warnings.
    - Ratings and Reviews: Users can view ratings and reviews provided by other users, offering social validation and feedback on the recipe's quality and taste.
    - Related Recipes: The system may suggest related or similar recipes to enhance the user's cooking experience and provide additional ideas.

### b. Collection

The functional requirement for the Collection function is to allow users to create and manage their personal collection of recipes within the Recipe Organizer application. This feature enables users to organize their favourite recipes, bookmark recipes for later use, and easily access their saved recipes.

* Function trigger: The Collection feature is triggered when a user selects the "Collection" tab or option within the Recipe Organizer application.
* Function description:
  + Actors/Roles: User (Cooker)
  + Purpose: To provide users with a convenient way to organize and access their favourite recipes.
  + Interface: The Collection screen/interface.
  + Data Processing: Retrieving and displaying recipe data from the application's database, allowing users to add or remove recipes from their collection.
* Function Details:
  + Data: The function retrieves the user's saved recipes from the application's database, including the recipe title, image, and other relevant information.
  + Validation: The system verifies that the user has appropriate permissions to access and modify their collection.
  + Business Logic: The function provides options to add or remove recipes from the user's collection, allowing them to curate their personal recipe library.
  + Functionalities:
    - View Collection: The function displays the user's saved recipes in a visually appealing and organized manner, allowing for easy browsing and selection.
    - Add to Collection: Users can add new recipes to their collection by selecting the "Add to Collection" option within the recipe details or search results.
    - Remove from Collection: Users can remove recipes from their collection, either individually or in bulk, to keep their collection up to date and organized.

### c. Feedbacks

The functional requirement for the Feedback function is to provide users with the ability to provide feedback and reviews on recipes within the Recipe Organizer application. This feature allows users to share their experiences, rate recipes, and provide comments or suggestions for improvement.

* Function trigger: The Feedback feature is triggered when a user selects the "Feedback" option within the Recipe Organizer application or completes a recipe they have prepared.
* Function description:
  + Actors/Roles: User (Cooker)
  + Purpose: To enable users to share their feedback, ratings, and comments on recipes to contribute to the recipe community and improve recipe quality.
  + Interface: The Feedback screen/interface.
  + Data Processing: Collecting and storing user feedback and ratings in the application's database, displaying user-generated content.
* Function Details:
  + Data: The function collects user feedback, ratings, and comments related to a specific recipe.
  + Validation: The system ensures that the user has prepared and completed the recipe before allowing them to provide feedback.
  + Business Logic: The function associates user feedback with the corresponding recipe, allowing for aggregation and presentation.
  + Functionalities:
    - Rating System: Users can rate recipes based on their satisfaction level, using a predefined rating scale (e.g., stars, thumbs up/down).
    - Comment Section: Users can provide comments, suggestions, or personal notes related to the recipe, sharing their insights or modifications.

## 5. Category

### a. Display category

* Function trigger: This function is triggered when a user wants to view a specific category in the Recipe Organizer application.
  + Actor/Roles:

Cooker/User: The person who logs in and wants to view a specific category.

System handle: The Recipe Organizer application that allows the user to organize their recipes into different categories and view them.

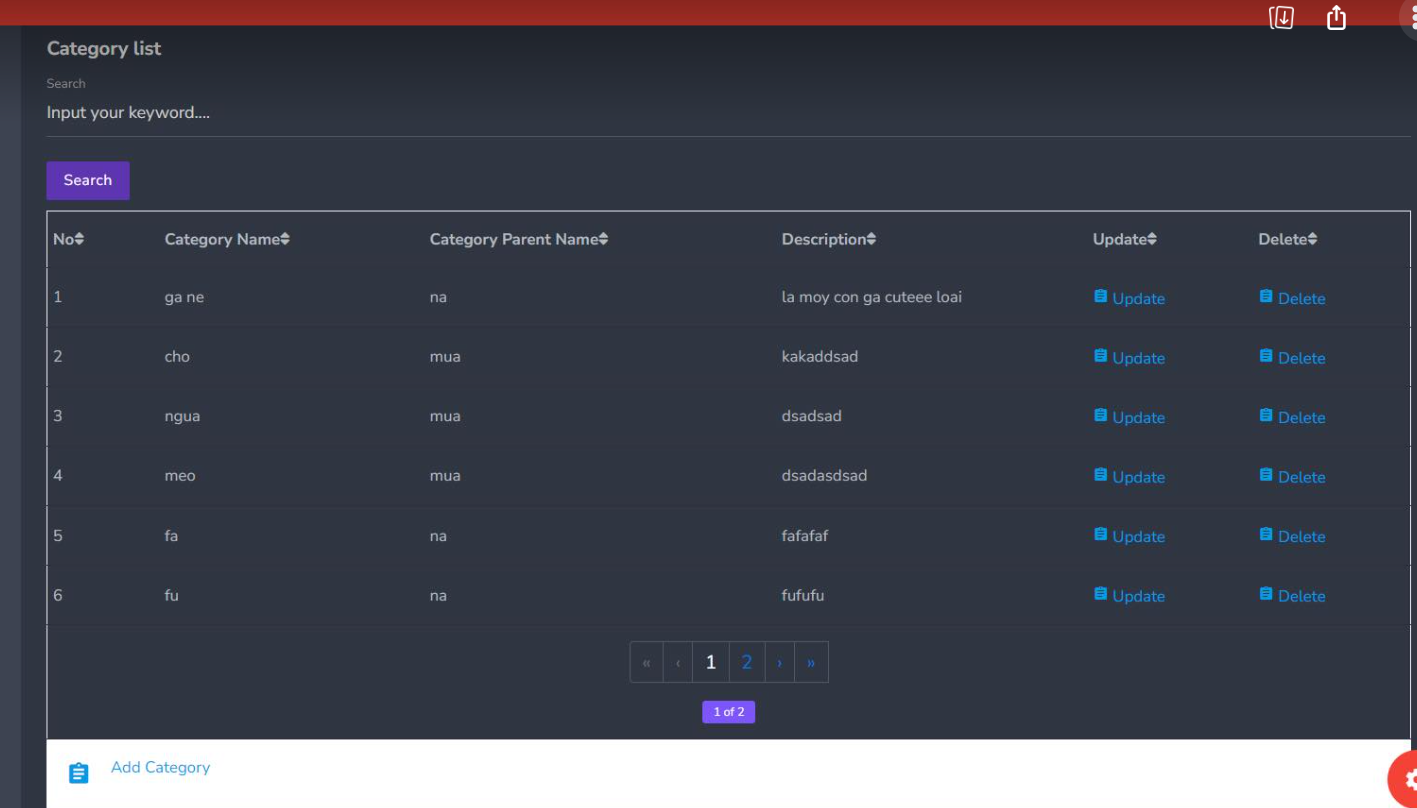
* + Purpose: The purpose of this function is to enable the user to view their recipes that are organized in a specific category.
  + Interface: The user selects a category from the category list, and the application displays the recipes associated with that category in a structured format.
  + Data Processing:

The function retrieves the selected category's information from the application's database.

* + The function retrieves the recipe information associated with the selected category from the application's database.
  + The function presents the recipe information associated with the selected category in a structured format.
  + Screen Layout: The recipe details page will display the selected category's recipes in a structured format, allowing the user to view and manage them.
* Function Details:
  + Data:
    - Selected Category: The category selected by the user.
    - Associated Recipes: The recipes associated with the selected category.
  + Functionality:
    - Normal Case: Normal Case: The user can view the recipes associated with the selected category, enabling them to organize and manage their recipes effectively.

### d. Manage Category

* Function Trigger: This function is triggered when a user selects a specific category from the category list in the "Manage Categories" section of the Task Management application.
* Function Description:
  + Actors/Roles: User, Task Management Application
  + Purpose: The purpose of this function is to provide users with the ability to manage task categories by performing various operations such as creating, reading, updating, and deleting categories.
  + Interface: The user selects a category from the category list, and the application displays a category details page or modal with relevant information and options for managing the category.
  + Data Processing:
  + The function retrieves the selected category's information from the application's database.
  + The function presents the category details, including name, description, and any additional attributes.
* Screen Layout: The category details page or modal will display the selected category's information in a structured format, allowing the user to manage the category.

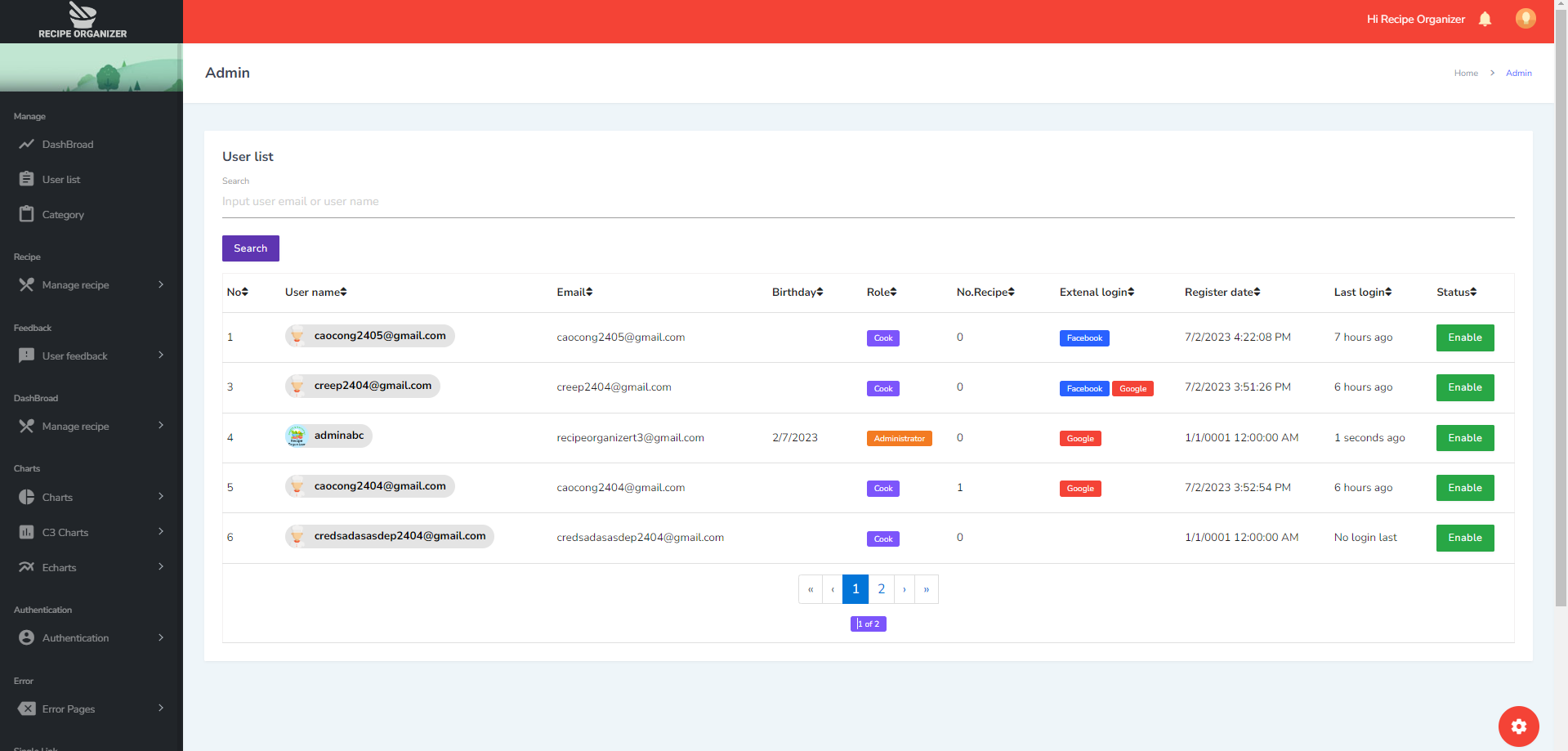


* Function Details:
  + Data:
  + Selected Category: The task category selected by the user.
  + Business Logic:
  + The function retrieves the selected category's information from the database.
  + The function allows the user to perform various operations on the category, such as creating, reading, updating, and deleting.
* Functionality:
  + Normal Case: The user can manage the selected category by creating, reading, updating, or deleting it, enabling them to organize their tasks efficiently.
  + Exceptional Cases: None.

## 6. Manage user

### a. View user list

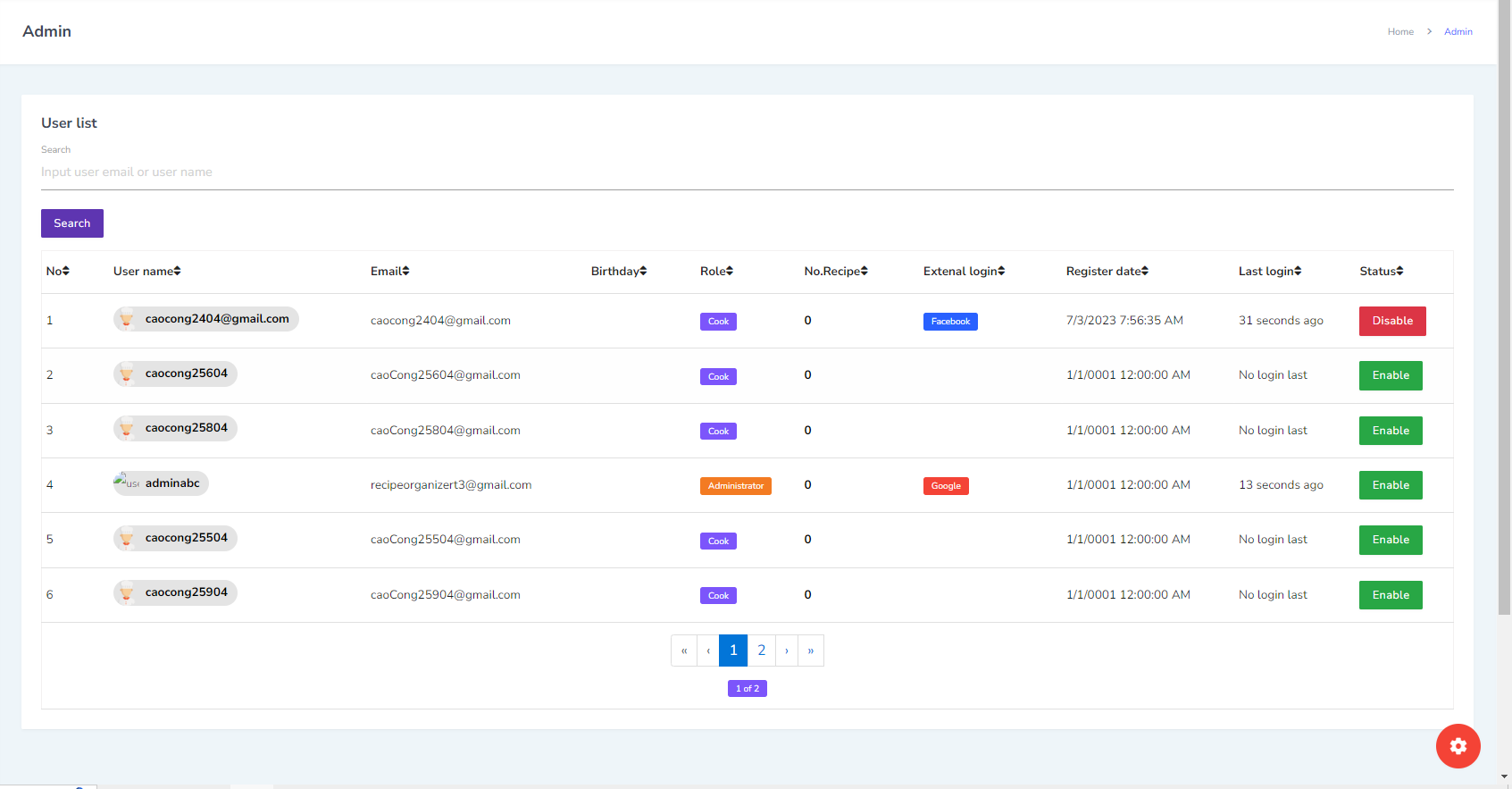
* Function Trigger: This function is triggered when an administrator or privileged user accesses the "Manage User" functionality within the Recipe Organizer application.
* Function Description:
  + Actors/Roles: Administrator, Recipe Organizer Application
  + Purpose: The purpose of this function is to provide administrators with the ability to manage user accounts within the Recipe Organizer application.
  + Interface: The administrator will access the "Manage User" section or navigate to the user management dashboard within the Recipe Organizer application.
  + Data Processing:
    - The function retrieves a list of user accounts from the application's database.
    - The function allows the administrator to perform various user management actions, such as viewing user details, editing user status, activating or deactivating user accounts.
* Screen Layout: The "Manage User" screen or dashboard will display a list of user accounts with relevant details, such as user name, email, birthday, roles, number of their recipe, external login, register account date, last login and account status. It will also provide options for editing user status, activating/deactivating accounts.

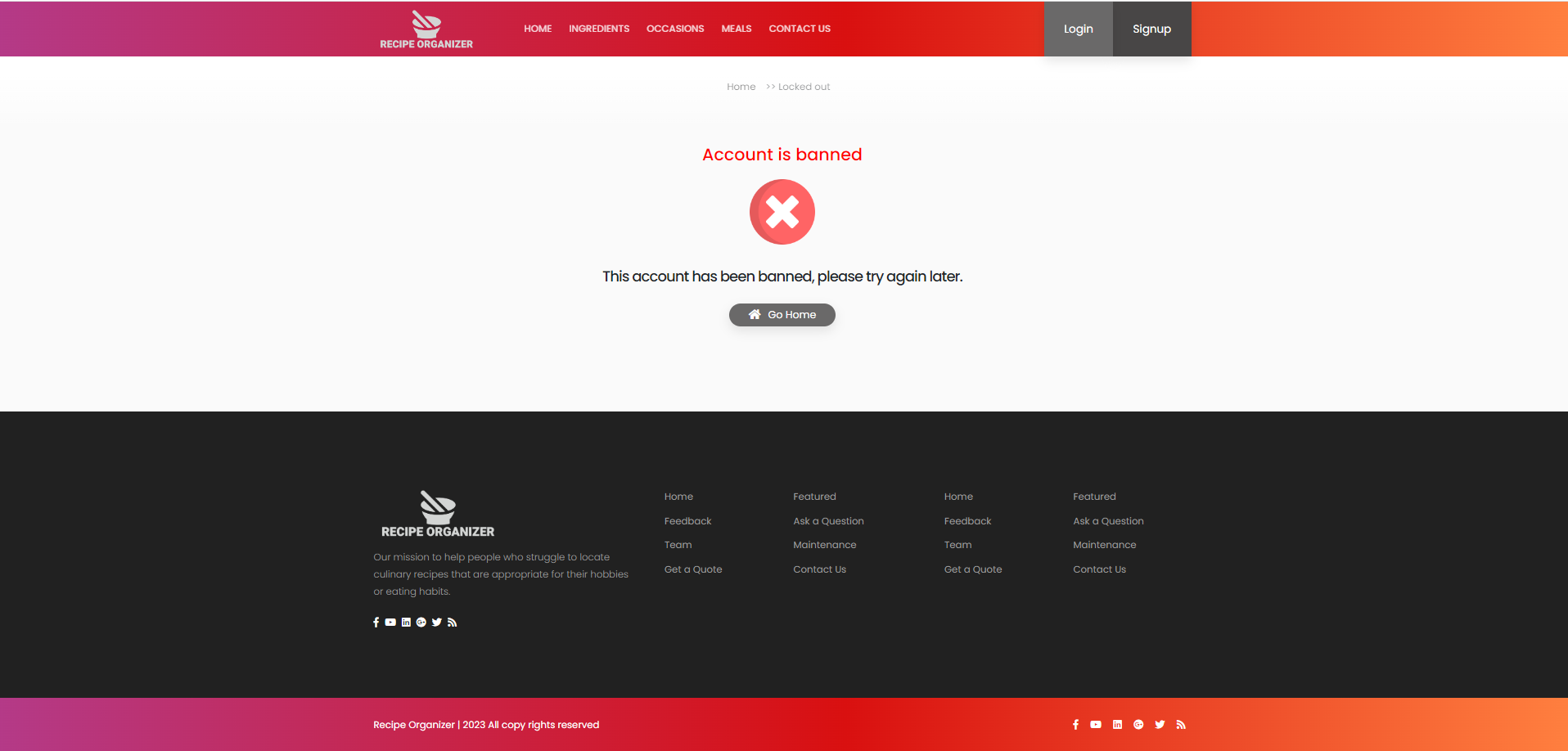


* Function Details:
  + Data:
    - User Accounts: The list of user accounts retrieved from the application's database.
  + Business Logic:
    - The function retrieves the list of user accounts from the database.
    - The function allows the administrator to perform actions such as viewing user details, editing user status
    - The function updates the user account status in the database based on the actions performed.
* Functionality:
  + Normal Case: The administrator can view and manage user accounts effectively. They can perform actions such as updating user status, activating or deactivating accounts.
  + Abnormal Cases: None

### b. Update user status

* Function Trigger: This function is triggered when an administrator clicks the "Status button" option for a specific user in the "Manage User" section of the Recipe Organizer application.
* Function Description:
  + Actors/Roles: Administrator, Recipe Organizer Application
  + Purpose: The purpose of this function is to update the status of a user account, such as activating, deactivating the account.
  + Interface: The administrator selects the "Update Status" option for a user, which opens a modal or a form to update the user's status.
  + Data Processing:
    - The function retrieves the selected user's current status from the application's database.
    - The function allows the administrator to select a new status from available options.
    - The function validates the new status and updates it in the database.
* Screen Layout: The update status form or modal displays the current status of the user and status button

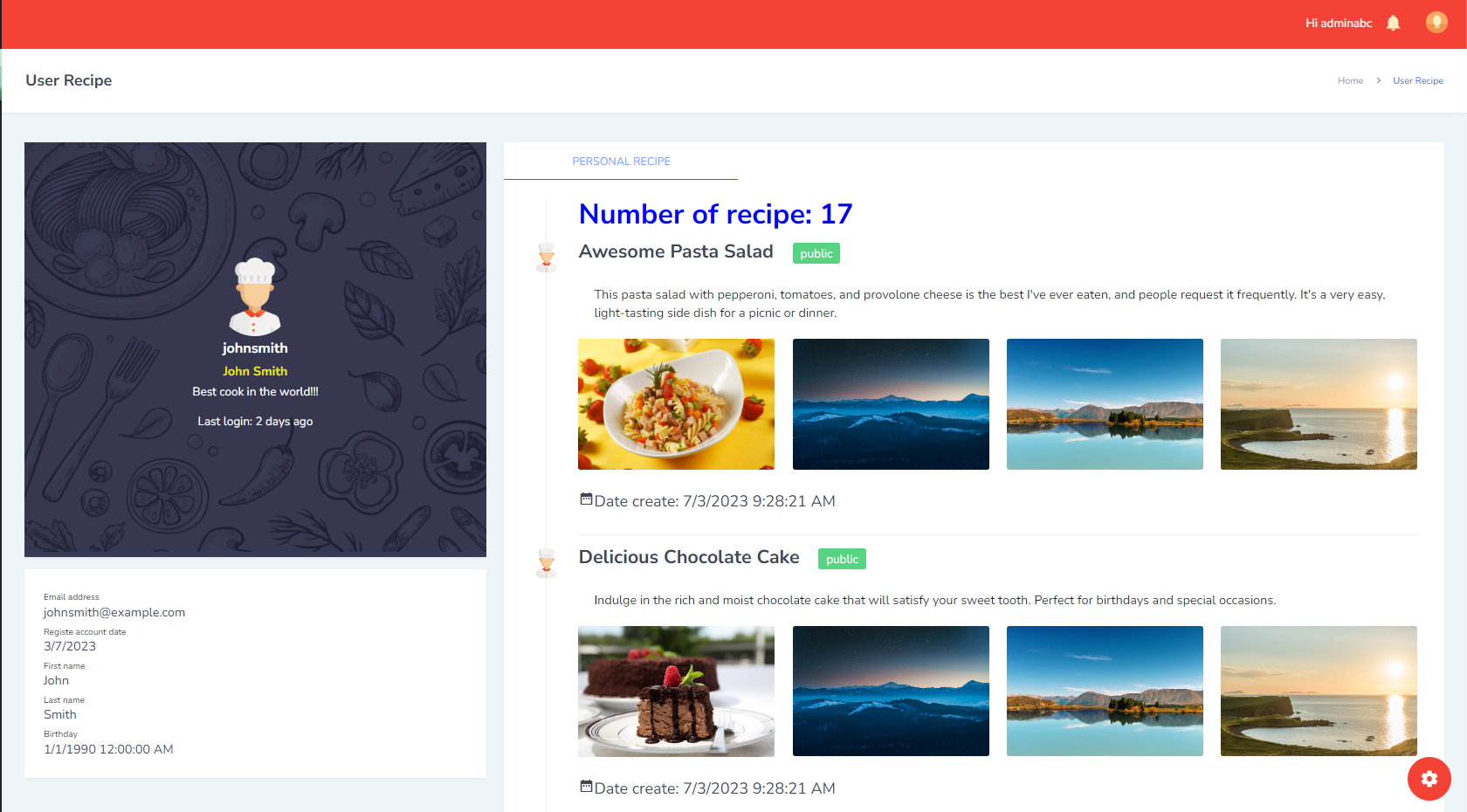




* Function Details:
  + Data:
    - Selected User: The user account selected by the administrator to update the status.
    - Current Status: The current status of the selected user account.
    - New Status: The updated status by the administrator.
  + Business Logic:
    - The function retrieves the selected user's current status from the database.
    - The function presents the current status and available options for updating the status.
    - The administrator clicks the status button to update a new status for the user account.
    - The function updates the user's status in the database.
* Functionality:
  + Normal Case: The administrator can update the status of a user account by clicking the status button.
  + Abnormal Cases:
  + Invalid Status Selection: If the administrator selects an invalid or unsupported status option, the function displays an error message and prevents the status update.
    - Unauthorized Access: If a user or unauthorized administrator attempts to update the status of a user account without proper authorization, the function denies access and displays an access deny page.

### c. Views user details

* Function Trigger: This function is triggered when an administrator selects a specific user from the user list in the "Manage User" of the Recipe Organizer application.
* Function Description:
  + Actors/Roles: Administrator, Recipe Organizer Application
  + Purpose: The purpose of this function is to display detailed information about a selected user and all their recipes created.
  + Interface: The administrator or privileged user selects a user from the user list, and the application displays a user details page or modal with relevant information.
  + Data Processing:
    - The function retrieves the selected user's information from the application's database.
    - The function presents the user details, including username, email, role(s), account status, registration date, and all their recipes created and any additional user-specific information.
* Screen Layout: The user details page or modal will display the selected user's information in a structured format, allowing the administrator to review and analyse it.



* Function Details:
  + Data:
    - Selected User: The user account selected by the administrator.
  + Business Logic:
    - The function retrieves the selected user's information from the database.
    - The function presents the user details, including username, email, role(s), account status, registration date, and any additional user-specific information.
* Functionality:
  + Normal Case: The administrator or privileged user can view the detailed information of the selected user, enabling them to understand the user's account status, roles, and other relevant details.
  + Abnormal Cases: None