

## Group Project Log

<b>Group Name:</b>	<b>Group 4</b>
--------------------	----------------

<b>Group Members:</b>	<b>6</b>
-----------------------	----------

<b>Deliverable:</b>	<b>Project Proposal and Front-End code (via Gitlab)</b>
---------------------	---

<b>Group Member Name</b>	<b>Work Done (%)</b>
ANUJ DAWAR ( <a href="mailto:an877696@dal.ca">an877696@dal.ca</a> )	<u>16.66%</u>
FAIZA UMATIYA ( <a href="mailto:faizaumatiya@dal.ca">faizaumatiya@dal.ca</a> )	<u>16.66%</u>
PARUL RAICH ( <a href="mailto:pr678267@dal.ca">pr678267@dal.ca</a> )	<u>16.66%</u>
RUCHIKA ( <a href="mailto:rc315087@dal.ca">rc315087@dal.ca</a> )	<u>16.66%</u>
SAGARKUMAR VAGHASIA ( <a href="mailto:sg682034@dal.ca">sg682034@dal.ca</a> )	<u>16.66%</u>
SAIF ALI PRASLA ( <a href="mailto:sf527330@dal.ca">sf527330@dal.ca</a> )	<u>16.66%</u>
<b>Total:</b>	<b>100%</b>

# PROJECT PROPOSAL

## PROJECT GROUP 04

CookWithDal – A recipe-sharing web application for Dalhousie students that enables students to share quick to make recipes.

---

### Members and Contributors

**Anuj Dawar**  
B00927005  
[an877696@dal.ca](mailto:an877696@dal.ca)

**Faiza Umatiya**  
B00899642  
[faizaumatiya@dal.ca](mailto:faizaumatiya@dal.ca)

**Parul Raich**  
B00903203  
[pr678267@dal.ca](mailto:pr678267@dal.ca)

**Ruchika**  
B00885185  
[rc315087@dal.ca](mailto:rc315087@dal.ca)

**Sagarkumar  
Vaghasia**  
B00878629  
[sg682034@dal.ca](mailto:sg682034@dal.ca)

**Saif Ali Prasla**  
B00899528  
[sf527330@dal.ca](mailto:sf527330@dal.ca)

---

Faculty of Computer Science  
Dalhousie University

28 February, 2023

# CONTENTS

---

- 1. PROJECT BACKGROUND ..... 1
  - 1.1 Project Objectives ..... 1
  - 1.2 Live Project URL ..... 1
- 2. APPLICATION DETAILS ..... 2
  - 2.1 Target User Insight..... 2
  - 2.2 Brand Attributes..... 2
  - 2.3 Competitive Landscape ..... 2
  - 2.4 Project Scope..... 3
    - 2.4.1 Project Goals ..... 3
    - 2.4.2 Proposed Features..... 3
    - 2.4.3 Bonus Features..... 5
  - 2.5 Information Architecture ..... 5
    - 2.5.1 Proposed Sitemap ..... 5
    - 2.5.2 Website Design ..... 6
  - 2.6 User Experience ..... 53
- 3. Asset Inventory ..... 108
- 4. Group Roles ..... 108
- 5. REFERENCES..... 109

# 1. PROJECT BACKGROUND

Recipe sharing has been a vital part of communication that helps people form bonds and explore new cultures. In this project, we intend to make a recipe-sharing web app for Dalhousie students. Students usually have a busy schedule and tend to rely on quick ready-to-eat meals to focus more on their studies. CookWithDal enables students to share quick to make recipes with minimal ingredients, or even recipes for regional delicacies with fellow students. This would not only help students learn new recipes, but also promote a sense of community and collaboration with students with a shared interest.

## 1.1 Project Objectives

For students at Dalhousie University, CookWithDal is a recipe and meal planning app that attempts to make it simpler for them to find, save, and organise their favourite recipes. Due to the hectic schedules of students and the time limits imposed by their academic objectives, CookWithDal's primary goal is to make recipe preparation simpler. It's important to have a balanced meal that doesn't interfere with their body's nutritional needs. Meal preparation and grocery shopping also assist in this by offering customers individualised recommendations based on their dietary choices, cooking skills, and ingredient accessibility.

Users of CookWithDal can explore a large collection of recipes from numerous sources, store their favourites, generate shopping lists, and even prepare meals in bulk using the meal quantifier tool. Users may also plan their meals for the upcoming week using CookWithDal's meal planner tool, which also creates a shopping list based on the selected recipes. Also, the app's meal quantifier tool helps users resolve any questions they may have about the numbers to consider when preparing dinner for one person or hundreds. Essentially, CookWithDal's goal is to save consumers time and stress when it comes to grocery shopping and meal preparation.

## 1.2 Live Project URL

Netlify Link:

<https://web-cs-dal-ca-cwd-csci5709-group04.netlify.app/>

Git Lab repository Link:

<https://git.cs.dal.ca/anujd/winter-2023-csci-5709-group-04.git>

Folder name:

**Group04\_CookWithDal**

## **2. APPLICATION DETAILS**

### **2.1 Target User Insight**

The focus of this application is on the students of Dalhousie who are keen on trying new recipes or are cooking enthusiasts or both. Student's primary goals and motivations on using this application is their need for a platform that would help them to browse vast range of recipes and keep a track of their favorite and most frequently go-to recipes easily. Other motivator of the students for using this application would be their need for an easy-to-use integrated shopping list feature with the application to keep track of missing ingredients from the recipe they want to make. Furthermore, the application provides a medium for students to get to know people with similar interests in cooking and provide constructive feedbacks using the like and comments functionality on recipes of our application.

Considering our main target audience to be students for now and considering that they are already acquainted with social media and web applications, they require no pre-requisite knowledge and training required.

### **2.2 Brand Attributes**

The main objective of CookWithDal is to provide students with a social and collaborative platform that would enable them to share their recipes and quick hacks and tricks with fellow students. We intend to make the application authentic and interactive by encouraging them to put the real pictures and videos along with recipes to make this recipe sharing experience more inclusive. In addition, the application would also allow users to like and comment on fellow student's recipes. The main concern of most of the students is on how to scale "serves 2" recipe to maybe "serves 10" or "serves 1" as it is usually cumbersome, our app focuses handling this issue by introducing a meal quantifying tool in app. To help diet-conscious students to be updated on their calorie intake, they could use our calorie tracker feature of our application.

We intended to keep our application creative, relevant and resourceful, and hence we included features such as "Shopping lists for missing items" and "favorite recipes" that would help students to keep track of missing recipes from the recipes they want to make and a list of their saved favorite recipes.

To keep our application user-friendly and easily accessible, we also have provided users with profile management functionalities which would help them manage their profile information, created and saved recipes, and other personal information.

### **2.3 Competitive Landscape**

Many recipe and meal planning applications exist, and they might be viewed as CookWithDal's competitors. One of them is given below:

1. Epicurious [6] is a recipe and culinary app that provides a large collection of recipes from a variety of publications, such as cookbooks and food magazines. Users at

Epicurious can save and organise recipes, make grocery lists, and plan meals—just as on CookWithDal. However, based on the user's tastes, Epicurious does not provide personalised recipe recommendations [6, 7].

CookWithDal stands out among the many recipe and meal planning applications available because it can effortlessly combine meal planning and grocery lists. Additionally, CookWithDal provides users with personalised recipe recommendations based on their dietary choices, item accessibility, and cooking skills. This feature saves users time and enables them to plan meals with greater knowledge [7].

- **User-Friendly Interface:** CookWithDal is accessible to a broad spectrum of users because of its straightforward navigation and clean, user-friendly design. This design includes elements that make the app simple to use scrolling, interactive buttons, and an appealing UI [7].

## **2.4 Project Scope**

### **2.4.1 Project Goals**

The goals of CookWithDal are as follows:

- To build a community where students can come together and try each other's recipes.
- To allow students to explore different recipes so they can try them out.
- To allow students to filter recipes based on their likings and mood.
- To allow students with interest in cooking, to share their recipes.
- To allow students to save a recipe.
- To allow students to like and comment a recipe and give feedback to the author.
- Allow students to create a meal planner so they plan their meals well in advance.
- To allow students manage their shopping lists.

### **2.4.2 Proposed Features**

#### **1) User Management**

- Login
- Registration
- Forgot/Reset Password (with security questions)
- Logout

#### **2) Profile Account Management - including preferences on recipe interests**

- Display Profile (List of recipes (my recipes, saved recipes))
- Update profile
- Deactivate profile

**3) Recipe Management**

- Add new recipe
- Update recipe
- Delete recipe
- Display recipe

**4) Bookmark/Save (favorites)**

- Save recipe
- Remove saved recipe

**5) Comments and likes**

- Add comment
- Update comment
- Delete comment
- View comment
- Like
- Unlike

**6) Feed, search and filter**

- Prep Level (Quick, expert, pro)
- Category: Veg/Non-veg/gluten free/dairy
- Calorie intake

**7) Shopping list for missing items**

- Create personal shopping list
- Add items to shopping list
- Remove items from shopping list
- Delete shopping list

**8) In-app Notification**

- Like notification
- Comment notification

**9) Meal quantifier and calorie tracker**

- Display quantity of ingredients required based on number of serves
- Show calories

### 2.4.3 Bonus Features

#### 1) Follow/Unfollow User

- Like profile
- Unlike Profile

#### 2) Meal Planner

- Add recipe to plan
- Delete recipe from plan

## 2.5 Information Architecture

Provide a complete sitemap of your application or web service and wireframes for each page you intend to develop for your application (i.e., pages within the scope of you project).

### 2.5.1 Proposed Sitemap

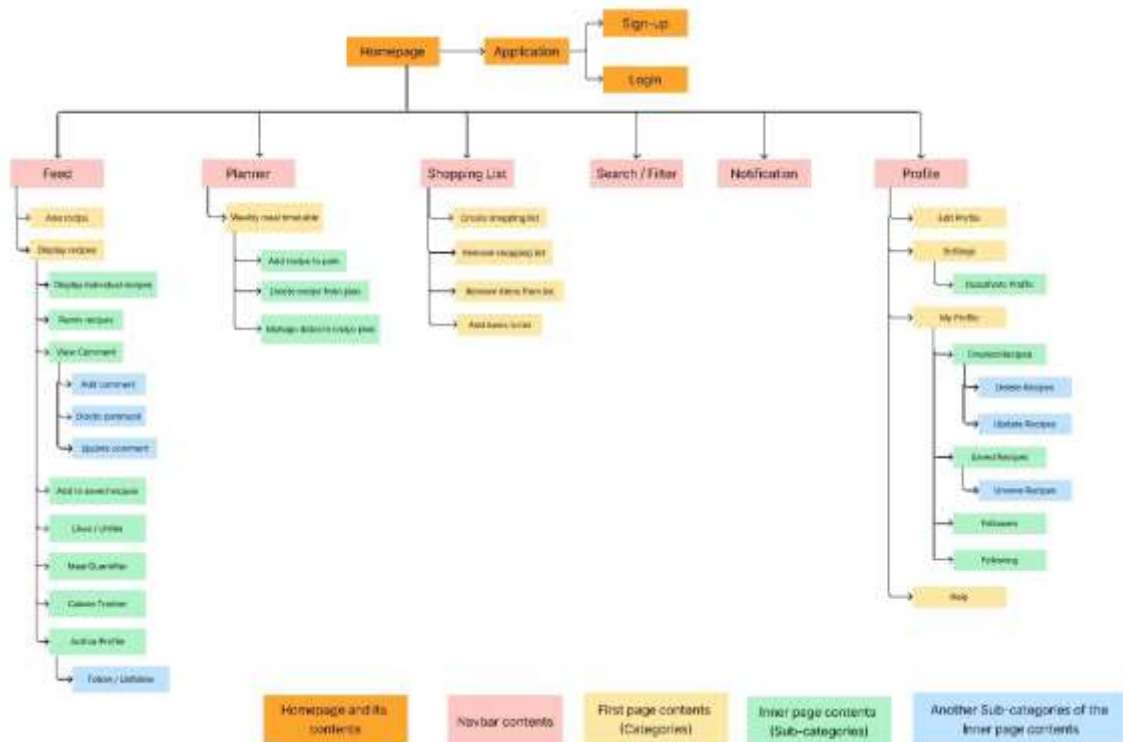


Figure 1: Sitemap [1]



## 2.5.2 Website Design

### FEATURE – User Management

- **TASK 1** – CREATE USER ACCOUNT

The image shows a web browser window with the title 'A Web Page'. The address bar contains 'https://CookWithDal.ca'. The main content area is titled 'Sign - Up Page' and contains the following form fields:

- First name:
- Last name:
- Email:
- Security question:
- Security answer:
- Password:
- Confirm Password:

At the bottom of the form is a 'Sign-up' button.

Figure 2: Sign-up page [4]

- **TASK 2 – LOGIN INTO THE ACCOUNT**

The screenshot shows a web browser window titled "Cook with Dal" with the URL "https://cookwithdal.com/". The page content includes a large "Login" heading, followed by "Email" and "Password" labels next to their respective input fields. A "Login" button is positioned below the fields. To the right of the button is a blue link labeled "Forgot Password". Below the button, the text "Not registered?" is followed by a blue link labeled "Sign Up".

Figure 3: Login page [4]

**Reset Password:**

The screenshot shows a web browser window titled "A Web Page" with the URL "https://CookWithDal.ca". The page content includes an "Email:" label next to a single input field. Below the field is a "Next" button.

Figure 4: Email - Reset Password page [4]

A Web Page

https://CookWithDal.ca

Security Question:

Security answer:

Next

This is a web browser window titled "A Web Page" with the address bar showing "https://CookWithDal.ca". The page contains two text input fields. The first is labeled "Security Question:" and the second is labeled "Security answer:". Below these fields is a button labeled "Next".

Figure 5: Security question page for reset password [4]

A Web Page

https://CookWithDal.ca

New Password:

Confirm Password:

Submit

This is a web browser window titled "A Web Page" with the address bar showing "https://CookWithDal.ca". The page contains two text input fields. The first is labeled "New Password:" and the second is labeled "Confirm Password:". Below these fields is a button labeled "Submit".

Figure 6: Password confirmation page for reset password [4]

## FEATURE – User Profile Management/ User Dashboard

- **TASK 1 - UPDATE USER PROFILE**

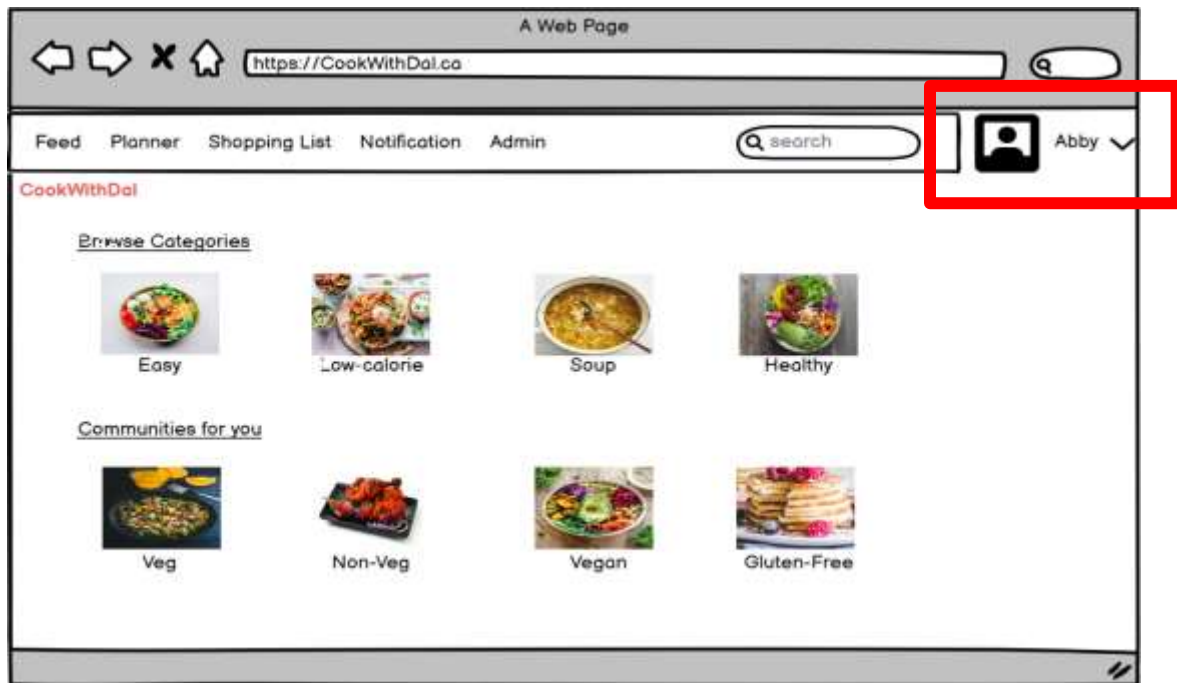


Figure 7: Feed page showing profile icon [4]



Figure 8: Profile icon showing dropdown list [4]



Figure 9: Update profile form [4]



Figure 10: Profile updated successfully [4]

- **TASK 2 - DEACTIVATE USER PROFILE**

A Web Page  
https://CookWithDal.ca

Feed Planner Explore Shopping List Notifi

CookWithDal

Browse Categories

Easy Low-calorie

Communities for you

Veg Non-Veg

First name: Abby

Last name: Johns

Email id: abbyjohns@gmail.com

Preference: Non-veg

Profile Picture: Choose File No chosen File

Save Deactivate Account

Figure 11: Profile form to deactivate [4]

A Web Page  
https://CookWithDal.ca

Feed Planner Shopping List Notifi

CookWithDal

Browse Categories

Easy Low-calori

Communities for you

Veg Non-Veg

First name: Abby

Last name: Johns

Email id: abbyj@gmail.com

Preference:

Profile Picture:

Alert  
Are you sure you want to deactivate your account?

No Yes

Save Deactivate Account

Figure 12: Alert message for confirming deactivation [4]



Figure 13: Profile deactivated successfully [4]

- **TASK 3 - DISPLAY LIST OF RECIPES (CREATED AND SAVED)**



Figure 14: Display profile page [4]



## FEATURE – Recipe Management

- **TASK 1** – ADD RECIPE

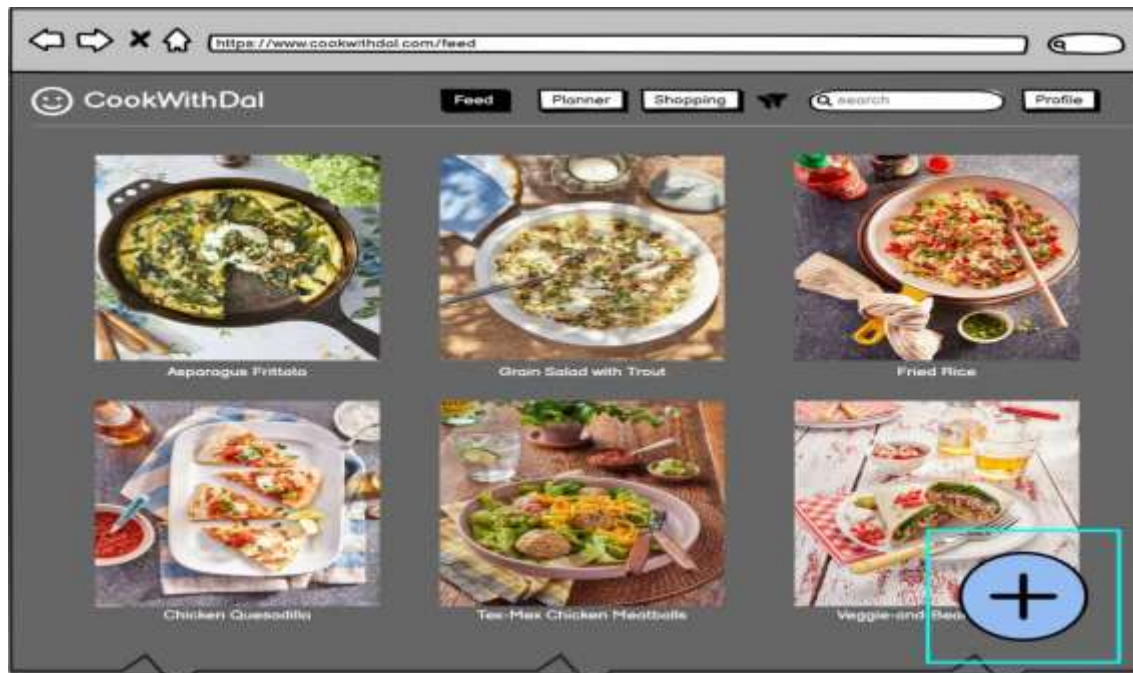


Figure 15: Wireframe for adding recipe [4]

The image shows a web browser window with the URL <https://www.cookwithdal.com/addrecipe>. The page header is identical to Figure 15. The main content area is titled "Add New Recipe" and contains a form with the following fields and controls:

- Title\*: A single-line text input field.
- Add Photos\* or videos: A text input field with a small image icon on the right.
- Description\*: A single-line text input field.
- Ingredients\*: A single-line text input field.
- Instructions\*: A single-line text input field.
- Servings\*: A single-line text input field.
- Prep time\*: Two input fields labeled "Hours" and "Minutes".

At the bottom of the form are two buttons: "CANCEL" and "SAVE".

Figure 16: Wireframe to add recipe form [4]





Figure 17: Wireframe showing recipe added successfully [4]

- **TASK 2 – DELETE RECIPE**

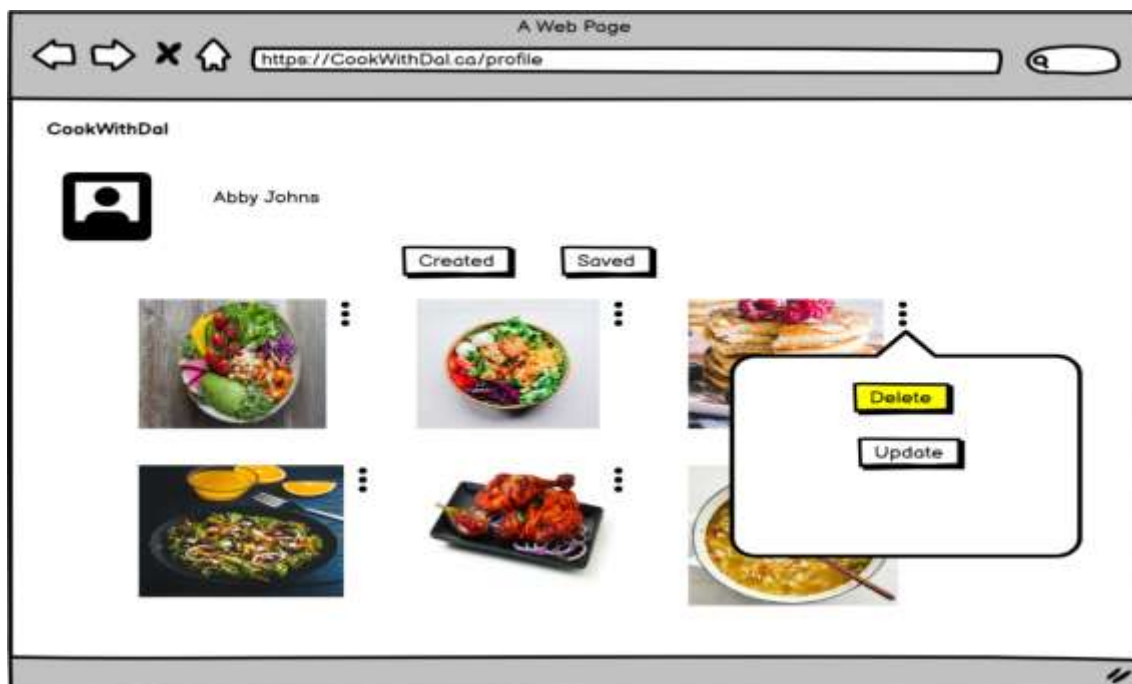


Figure 18: Delete button to delete recipe [4]

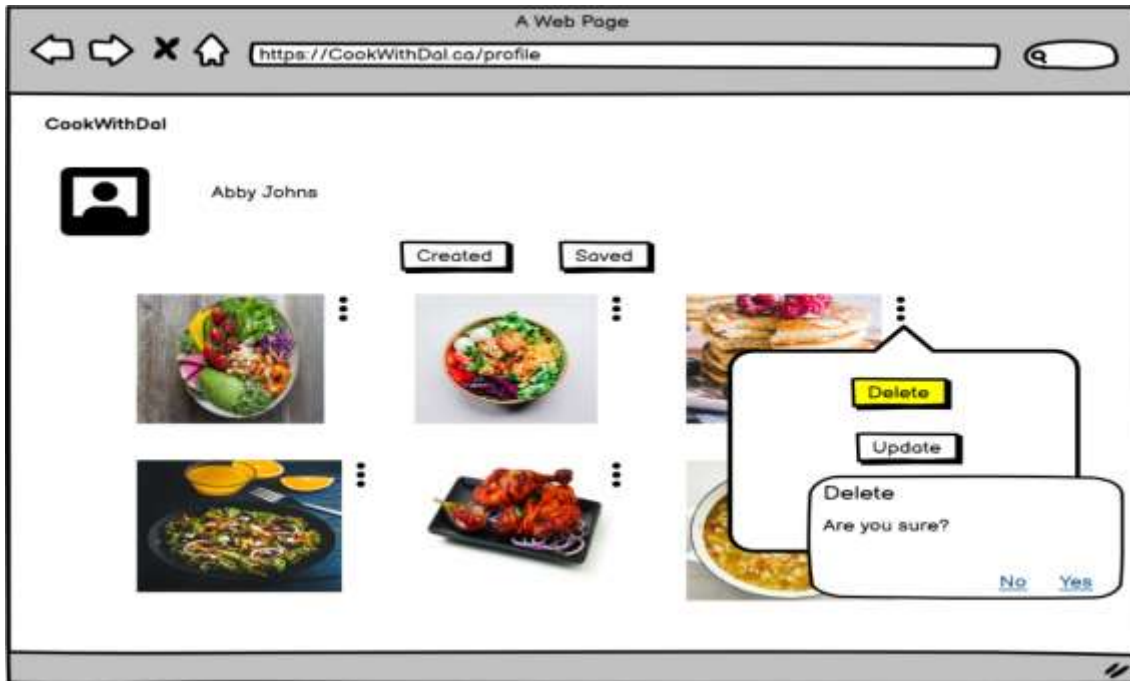


Figure 19: Alert message for confirming deletion [4]

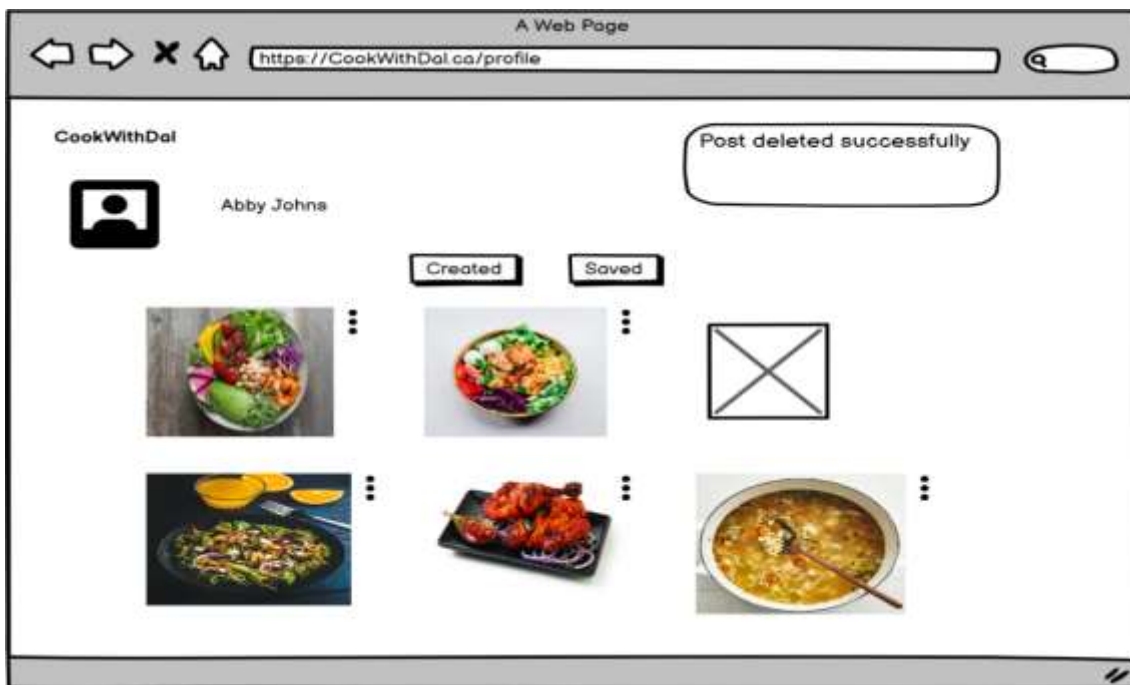


Figure 20: Alert message for post deleted successfully [4]

- **TASK 3 – UPDATE RECIPE**

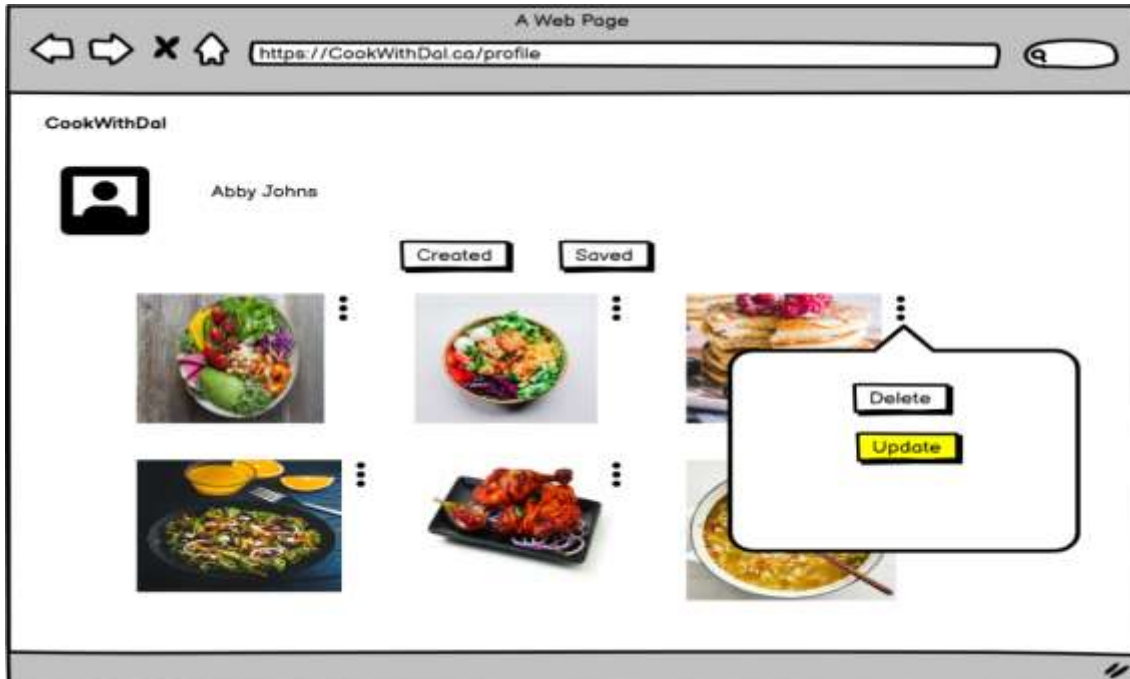


Figure 21: Update recipe button [4]

The screenshot shows a web browser window titled 'A Web Page' with the address bar displaying 'https://CookWithDal.ca/Update'. The page content includes the 'CookWithDal' logo and the title 'Update Recipe'. The form consists of the following fields and buttons:

- Title: [Text input field]
- Add Photo/Video: [Text input field]
- Description: [Text input field]
- Ingredients: [Text input field]
- Instructions: [Text input field]
- Servings: [Text input field]
- Buttons: Cancel, Update

Figure 22: Update recipe form [4]


A Web Page

https://CookWithDal.ca/Update

CookWithDal

Update Recipe

Title: Healthy food recipe

Add Photo/Video: 

Description: Contains boneless chicken and white rice

Ingredients: chicken, rice, tomato, onions, chillies, ginger garlic paste, cilantro

Instructions: Cook boneless chicken with spice on low flame, Boil rice and add with the chicken

Servings: 5

Cancel Update

Update  
Are you sure?  
[yes](#) [no](#)

Figure 23: Alert message for confirming updation [4]

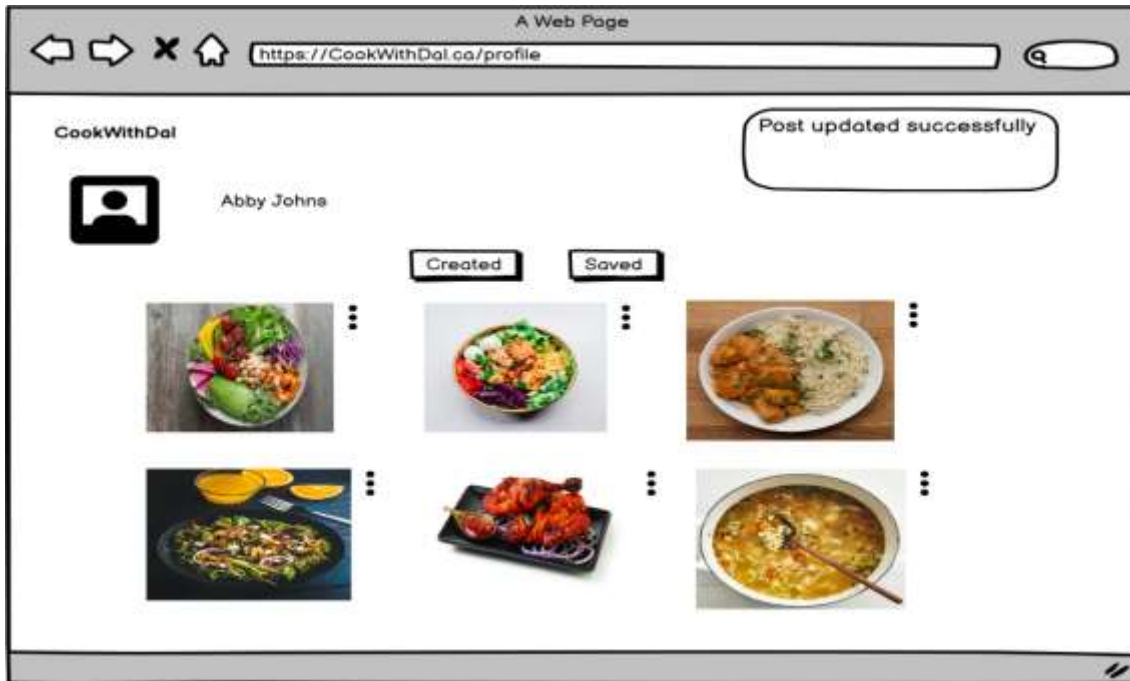


Figure 24: Alert message for post updated successfully [4]

- **TASK 4 - DISPLAY RECIPE**

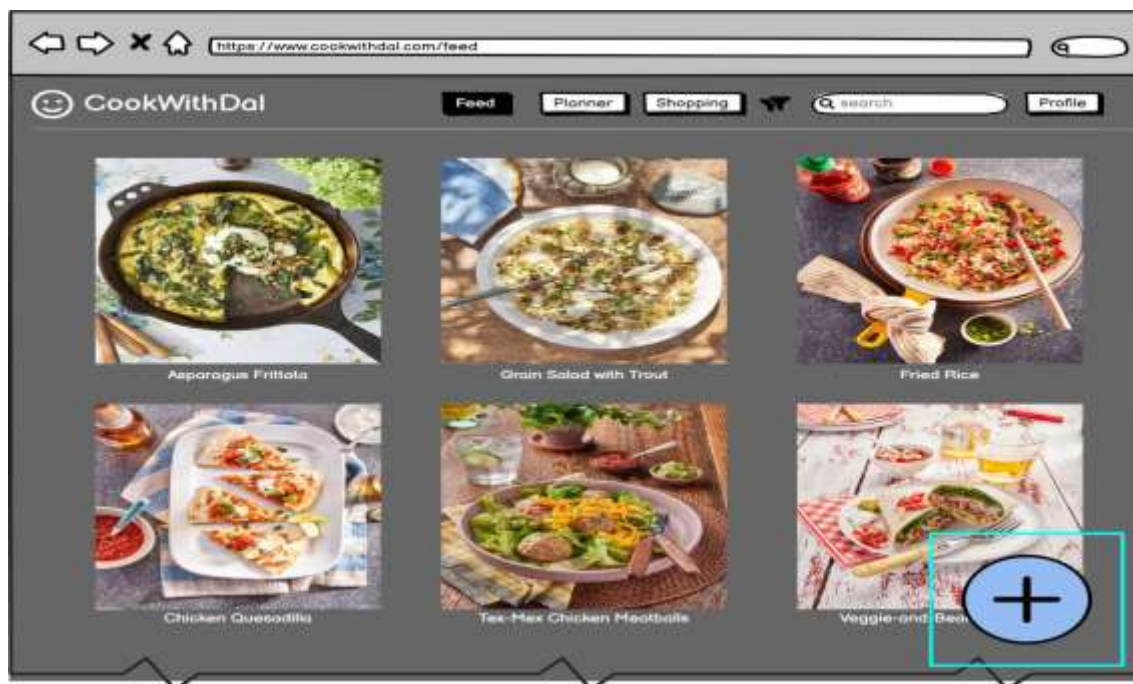


Figure 25: Display recipe page [4]



## FEATURE – Follow/Unfollow User

- **TASK 1 - FOLLOW USER**



Figure 26: Recipe post page [4]

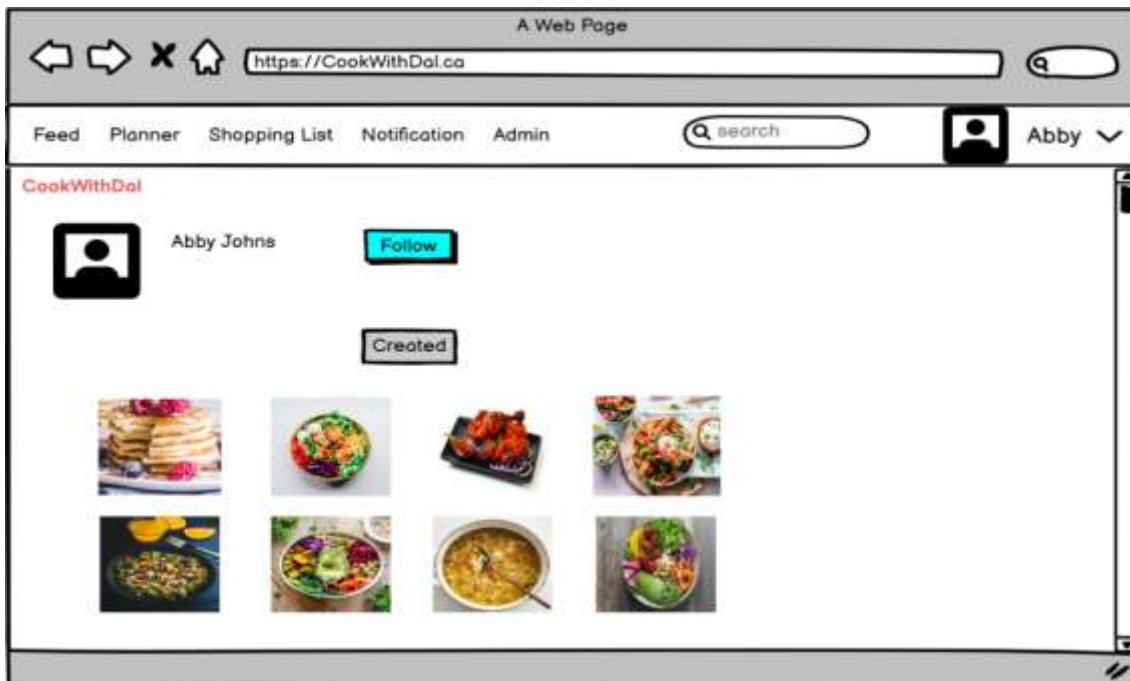


Figure 27: Follow button in profile page [4]



Figure 28: User followed the profile [4]

- **TASK 2 - UNFOLLOW USER**



Figure 29: Recipe post page [4]



Figure 30: Unfollow button in profile page [4]

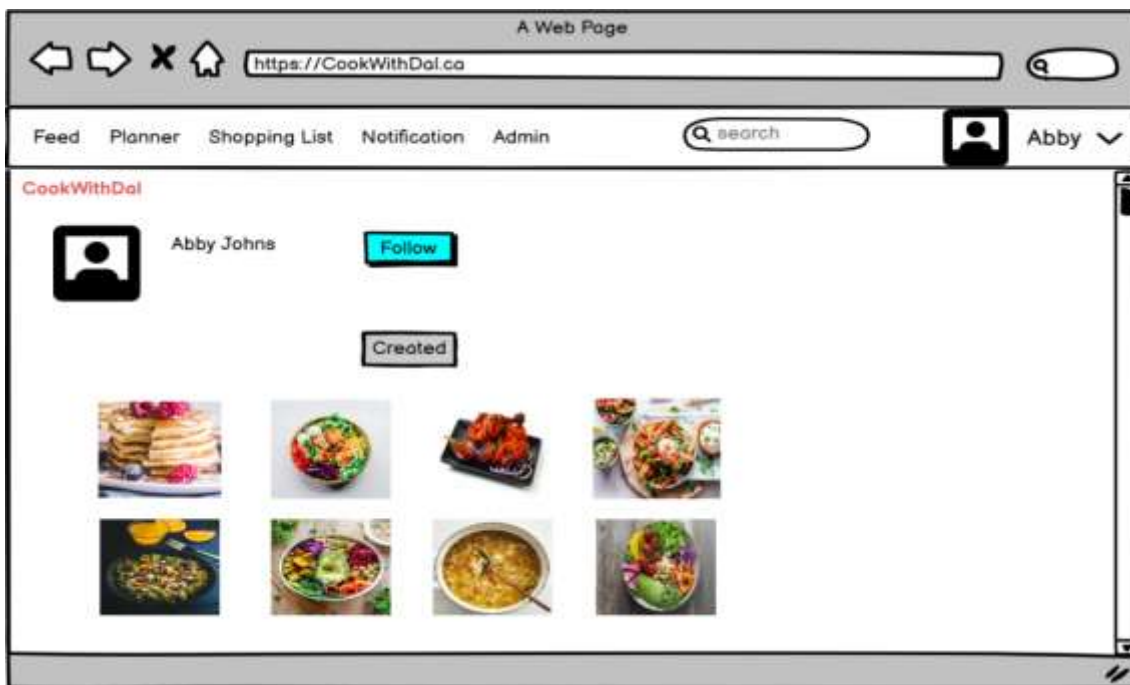


Figure 31: User unfollowed the profile [4]



## FEATURE – Likes and Comments

- **TASK 1 - LIKE RECIPE**

If the user likes the recipe, they can hit the like button to increase the like count.

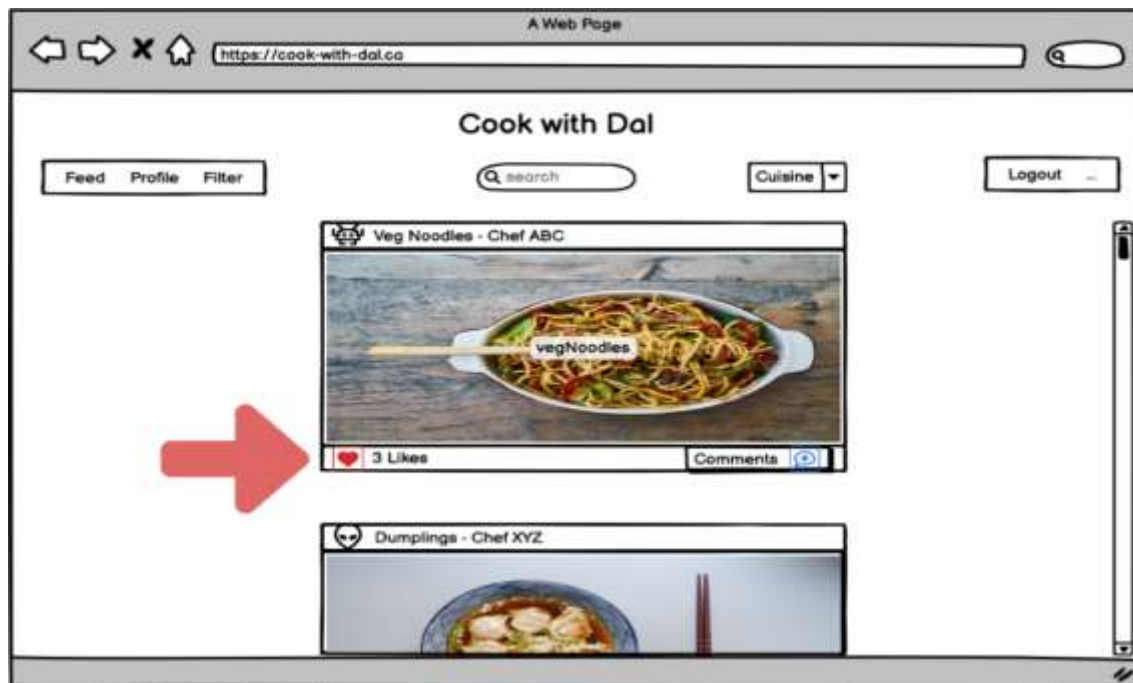


Figure 32: Wireframe – Liked recipe [4]

- **TASK 2 - UNLIKE RECIPE**

If the user has clicked on the like button by mistake, or has changed their mind, they can undo the like action by hitting the like button again.

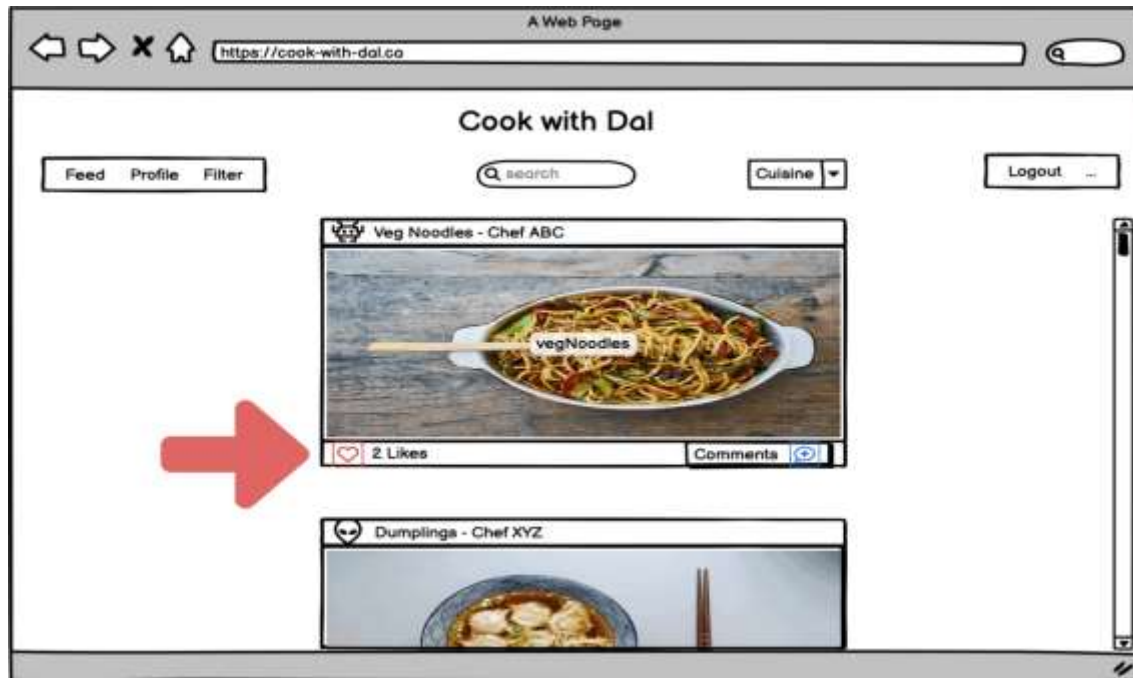


Figure 33: Wireframe – Unliked Recipe [4]

- **TASK 3 – VIEW COMMENTS**

The red arrow is pointing to the view comments button.



Figure 34: Wireframe – View comments button [4]

On clicking the view comments button, a dialog will be displayed with the list of comments. If there are no comments, a message will be displayed stating that there are no comments yet.

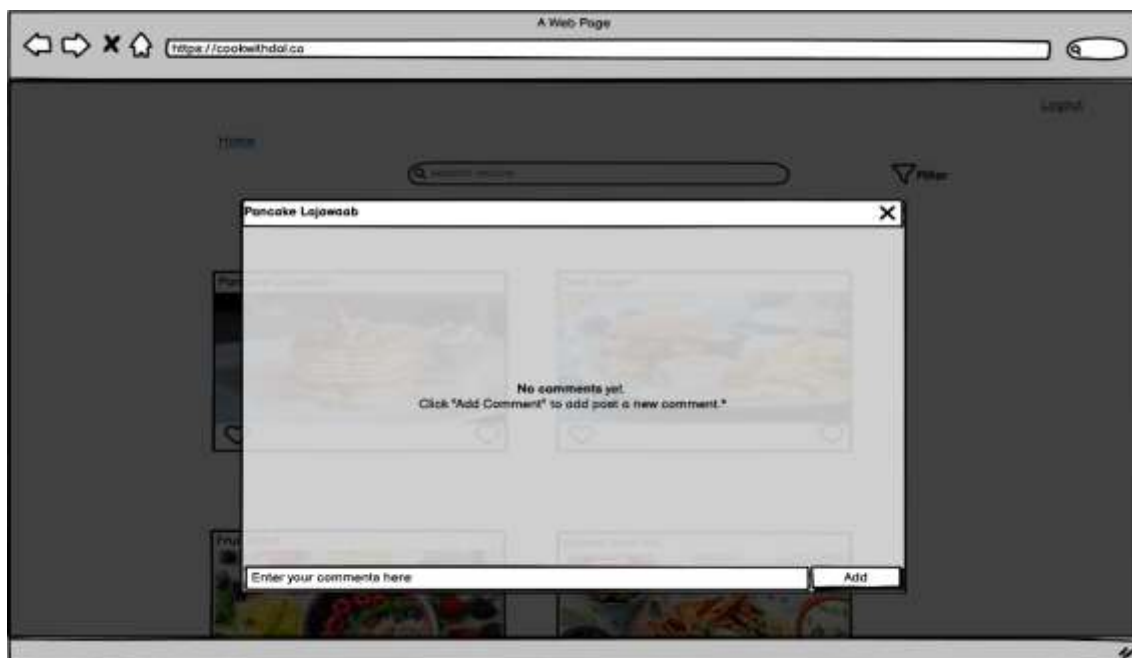


Figure 35: Wireframe – View comments [4]

- **TASK 4 – ADD COMMENT**

User will be prompted with an input box at the bottom of the dialog where they can enter their comments. The red arrow is pointing to the add comment button.

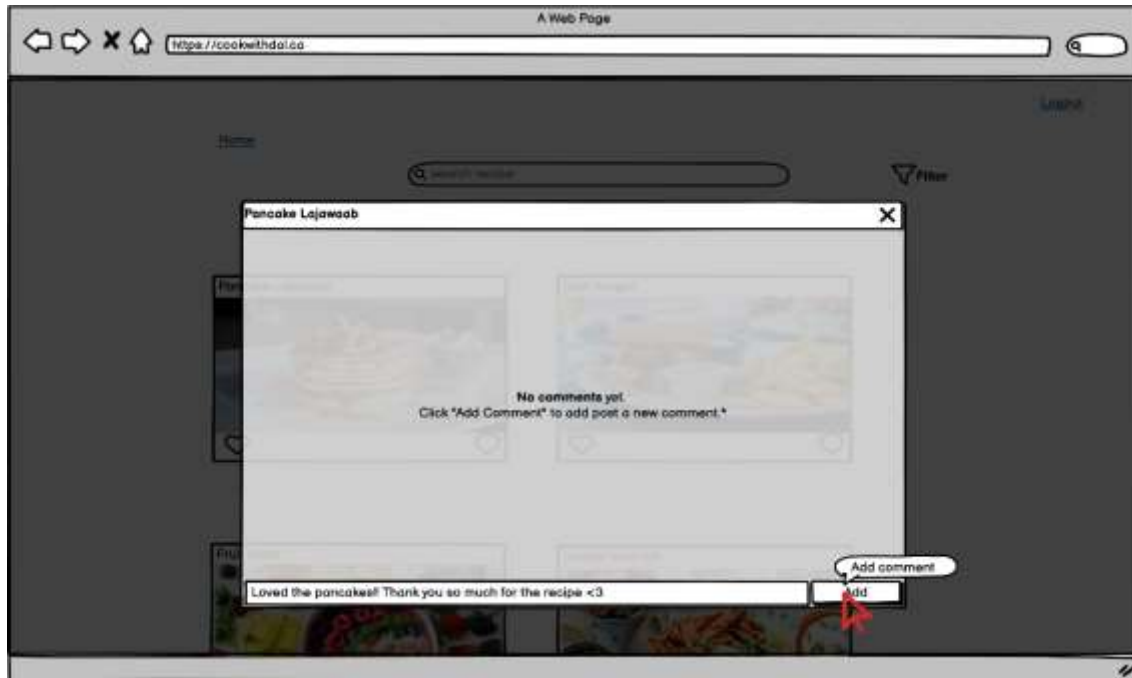


Figure 36: Wireframe – Add comments [4]

A new comment is added to the comments list for the recipe.

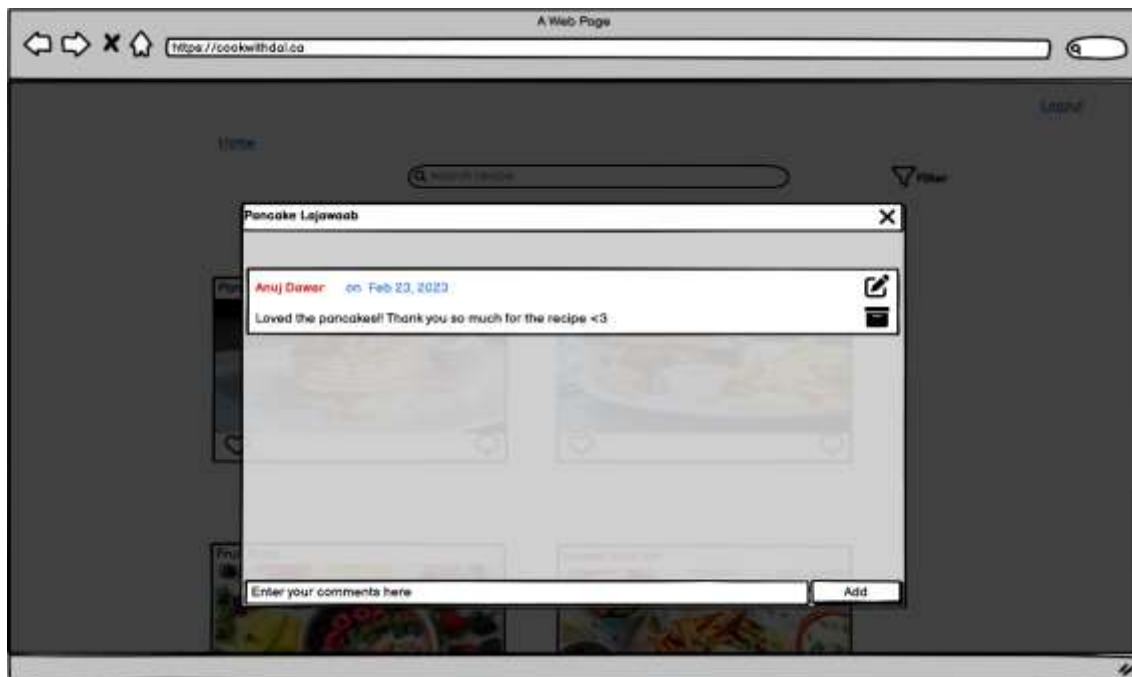


Figure 37: Wireframe –New comment added [4]

- **TASK 5 – EDIT COMMENT**

User can edit an existing comment by clicking on the Edit button associated with the comment (red arrow pointing in the figure).

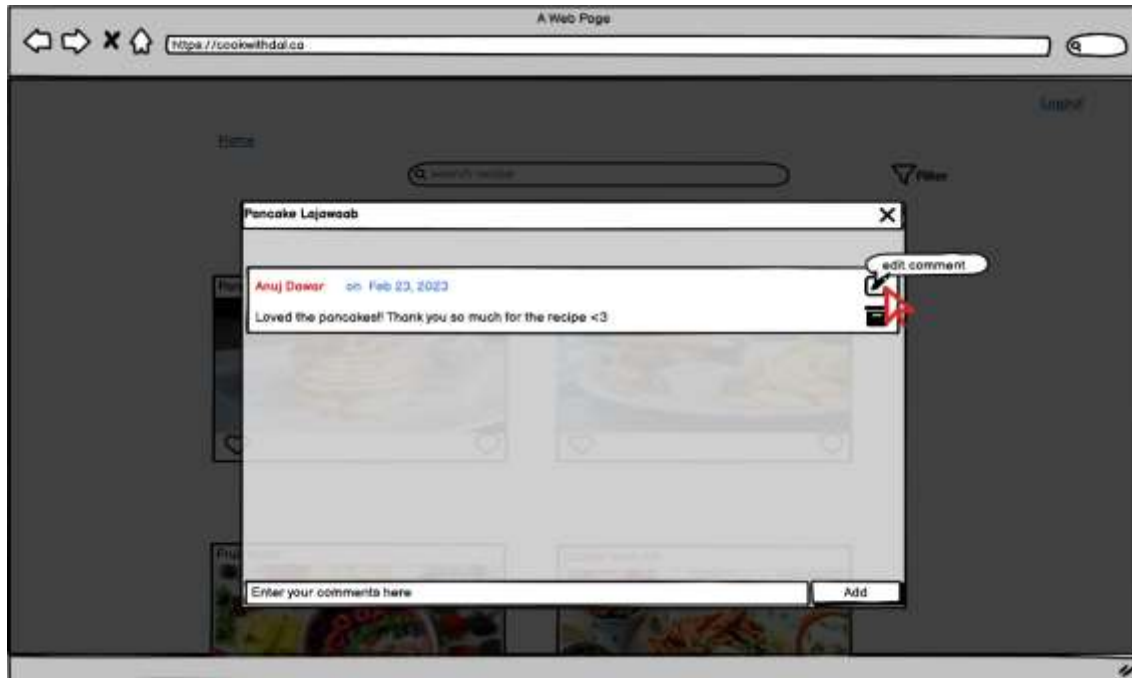


Figure 38: Wireframe –Edit comment [4]

User can edit the existing comment.

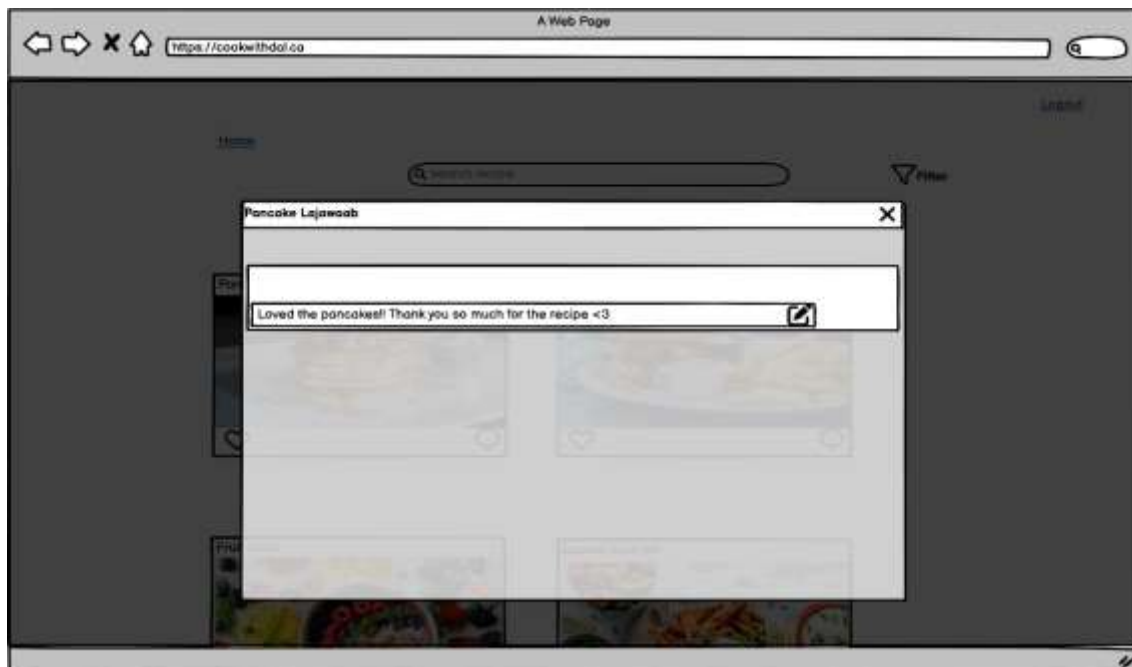


Figure 39: Wireframe –Editing comment [4]

User can update the edited comment by clicking on the update button.

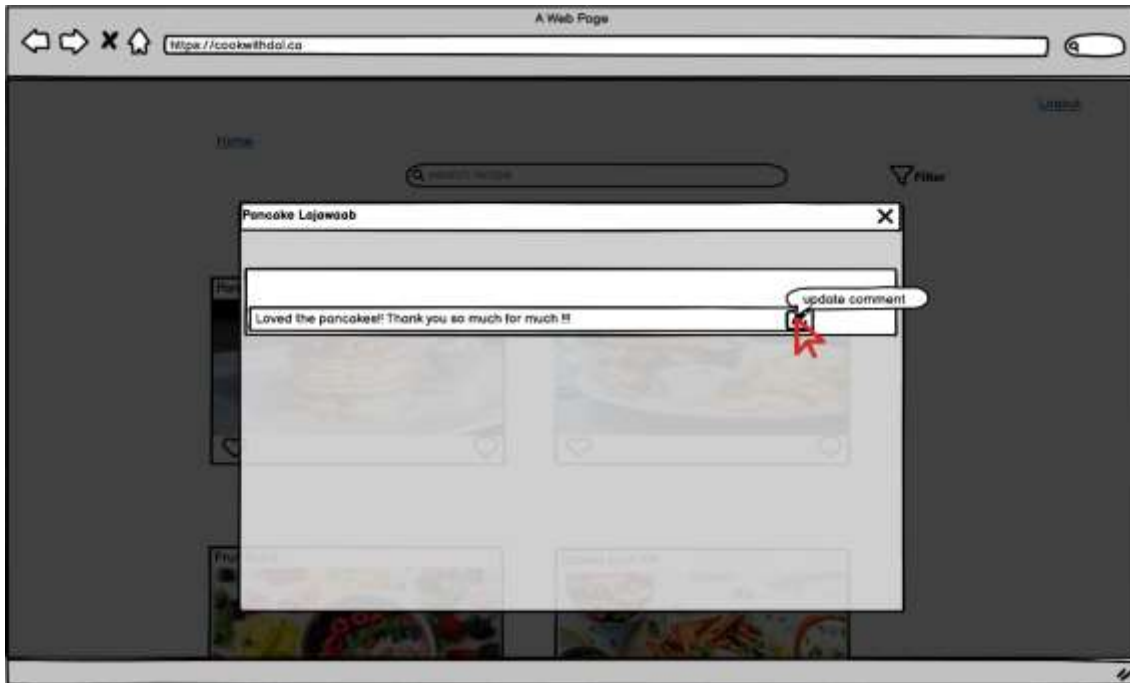


Figure 40: Wireframe –Update comment [4]

On successful update, the user will be prompted by a confirmation message.



Figure 41: Wireframe –Comment updated [4]

- **TASK 6 – DELETE COMMENT**

To delete an existing comment, user can click on the delete comment associated with the comment.

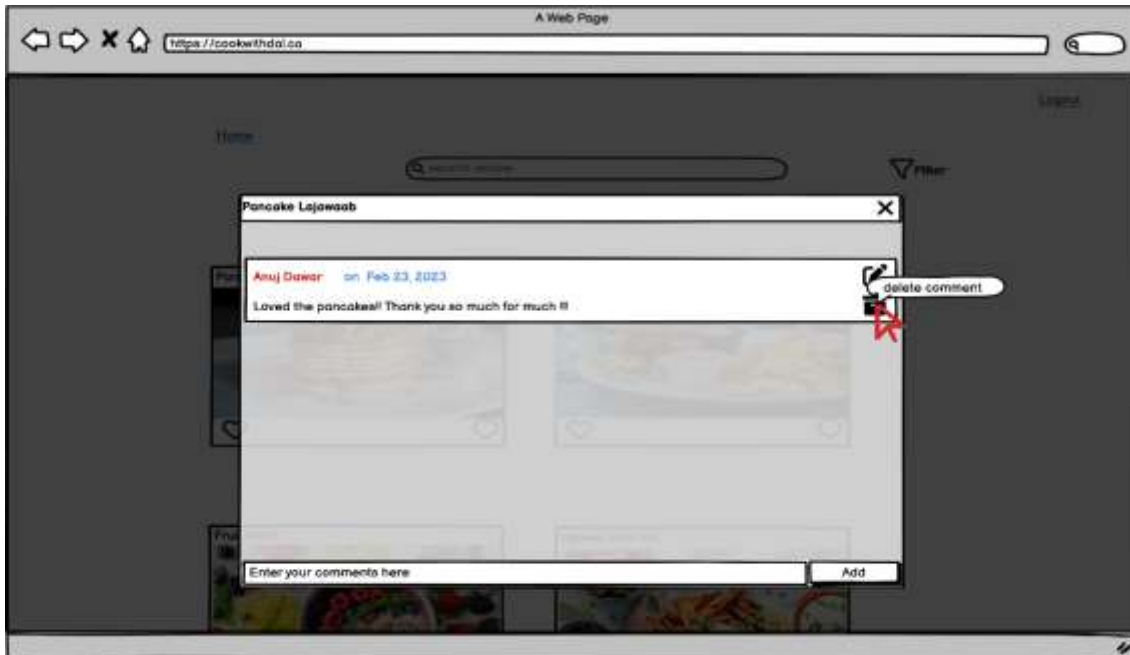


Figure 42: Wireframe –Delete comment button [4]

User will be prompted with a message box to confirm whether they want to delete the comment.



Figure 43: Wireframe –Delete comment confirmation [4]

A confirmation that the comment has been deleted will be displayed to the user.

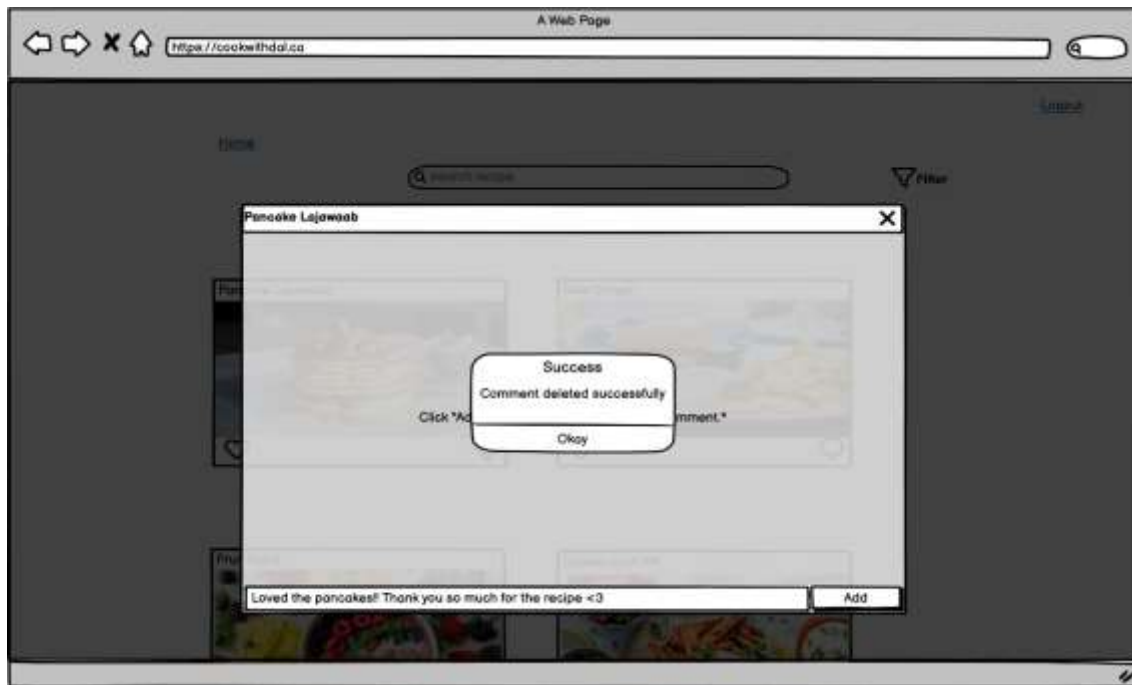


Figure 44: Wireframe –Deleted comment [4]



## FEATURE – Shopping List

- **TASK 1 - CREATE A SHOPPING LIST**

**Figure 45** shows the wireframe of the Shopping List page for "CookWithDal" website. The page is displayed with a message “Oops, you don't have any shopping list” and “GET STARTED” button if this is the first-time user is going to create list.

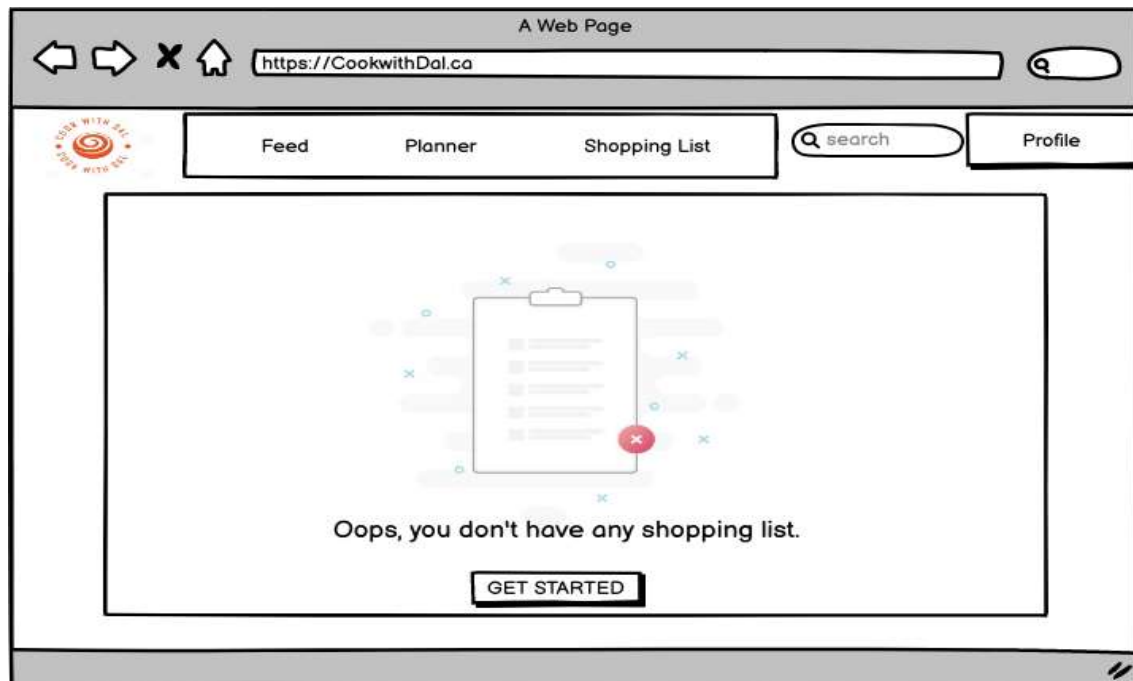


Figure 45: Shopping List Page Wireframe [2]

**Figure 46** shows the wireframe of the dialog box which gets displayed when the user clicks on the “GET STARTED” button asking for Shopping List Name. User can give the new Shopping list name and click on Save to get it created. Also, user can click on cancel if he doesn’t want to create any new list.



Figure 46: Shopping List Page with Shopping List Name Dialog Box Wireframe [4]

**Figure 47** shows the wireframe of the error message which gets displayed when the user clicks on Save button without entering the Shopping List Name in field.



Figure 47: Shopping List Page with error message Wireframe [4]

**Figure 48** shows the wireframe when user entered the Shopping List Name.



Figure 48: Shopping List Page with valid Shopping List Name Wireframe [4]

**Figure 49** shows the wireframe of the “Shopping List Page” with the newly created Shopping list once the user gives the Shopping List name and click on Save button. When the user clicks on newly created Shopping List, the blank shopping list will be displayed in right panel with a dropdown to add ingredients/items.



Figure 49: Shopping List Page with newly created Shopping List Wireframe [4]

- **TASK 2 - ADD ITEMS TO THE SHOPPING LIST.**

**Figure 50** shows the wireframe of the “Adding Items to Shopping List” with the newly created Shopping list. The left panel will display the shopping list name whereas the right panel will display the ingredients added to that list. In the right panel, there will be a dropdown to select items that the user wants to add to his list.

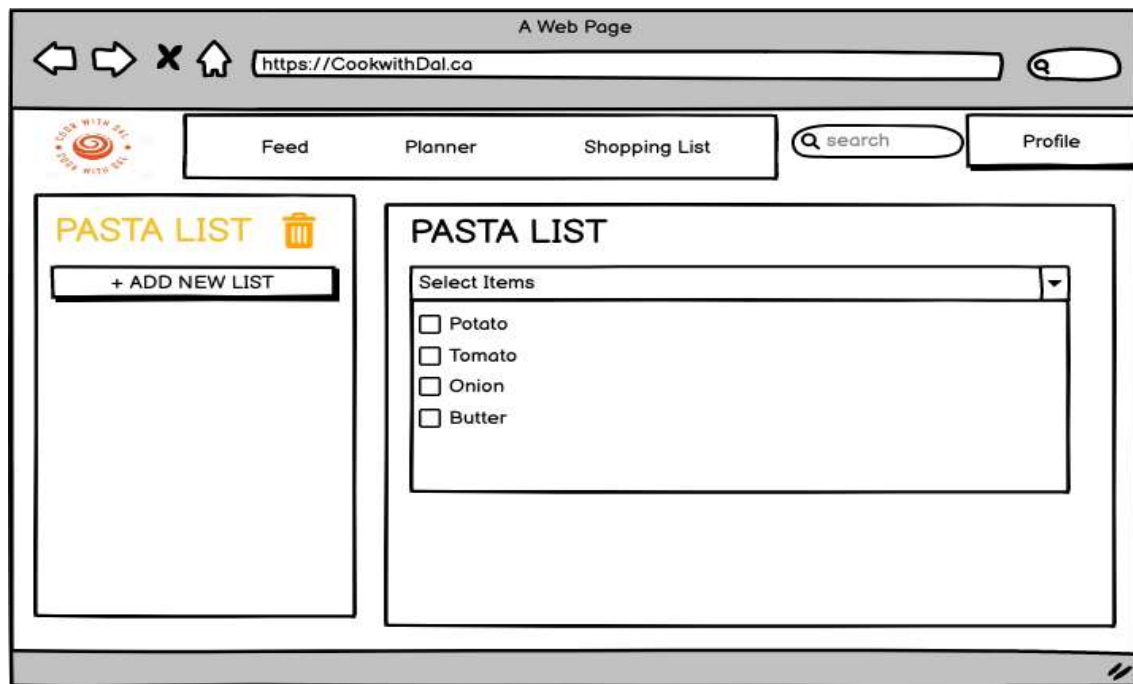


Figure 50: Shopping List Page with Adding Items to Shopping List from Dropdown [4]

**Figure 51** shows the wireframe of the “Adding Items to Shopping List” with the newly created Shopping list and added items to that list under “Selected Item(s)”. Each added item will have a checkbox associated with it so that the user can check it once he has purchased it or no longer need it.

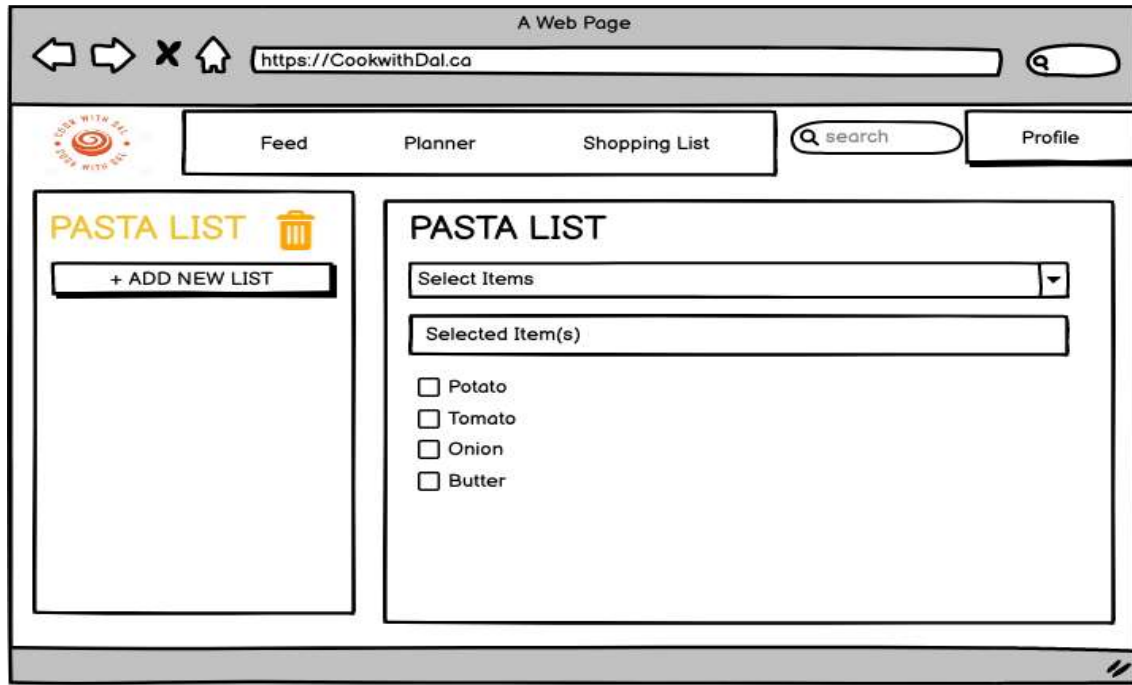


Figure 51: Shopping List Page with Added Items to Shopping List [4]

- **TASK 3 - REMOVE ITEMS FROM THE SHOPPING LIST**

**Figure 52** shows the wireframe of the “Removing Items from Shopping List” with the newly created Shopping list and added items in it. Once the user checks the checkbox associated with the item, the item will be moved under “Checked Item(s)” with strike off.

The wireframe depicts a web browser window titled "A Web Page" with the URL "https://CookwithDal.ca". The browser's address bar and navigation icons are visible. The website's header includes a logo on the left and navigation tabs for "Feed", "Planner", "Shopping List", and "Profile". A search bar is located to the right of the "Shopping List" tab. The main content area is divided into two panels. The left panel, titled "PASTA LIST" with a trash icon, contains a button labeled "+ ADD NEW LIST". The right panel, also titled "PASTA LIST", features a "Select Items" dropdown menu, a "Selected Item(s)" text box, and a list of items: "Potato" and "Tomato" (each with an unchecked checkbox), and "Onion" and "Butter" (each with a checked checkbox). Below the list is a "Checked Item(s)" text box. The items "Onion" and "Butter" are shown with a strike-through, indicating they have been moved to the checked list.

Figure 52: Shopping List Page with Removing Items from Shopping List [4]

- **TASK 4 - DELETE THE SHOPPING LIST**

**Figure 53** shows the wireframe of the “Removing Shopping List” with the newly created Shopping list and added items in it. Each shopping list created has a trash icon beside it.

Once the user completed the list or no longer needed, he will click on trash icon beside the list to delete that shopping list completely.

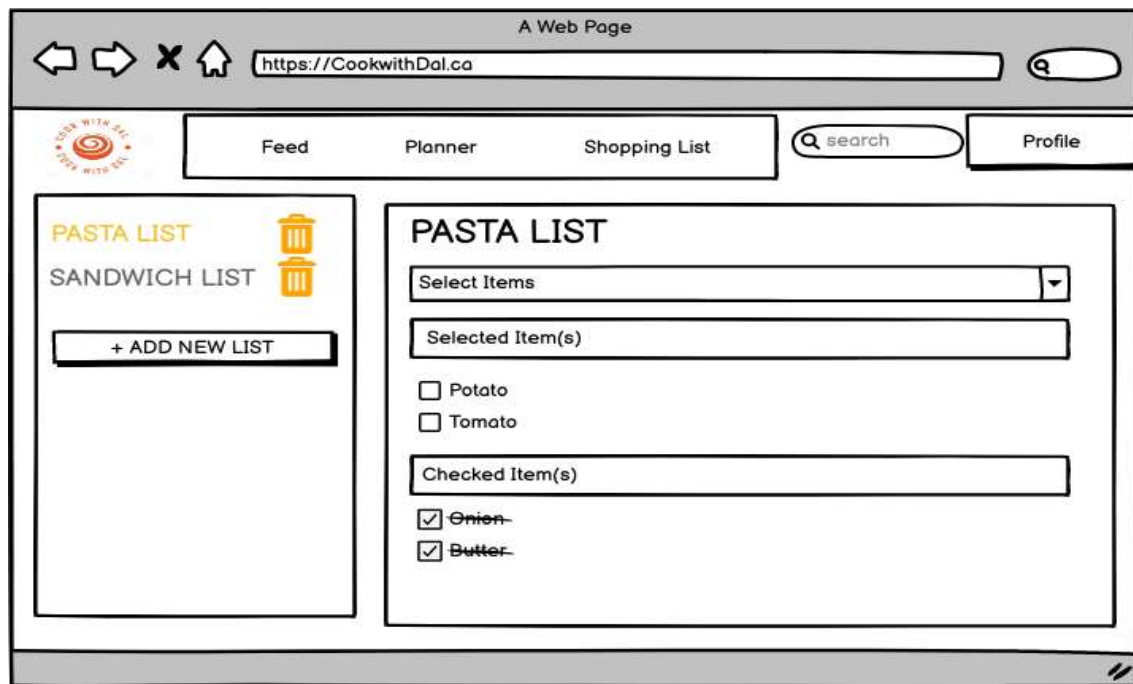


Figure 53: Removing Shopping List Wireframe [4]



**Figure 54** shows the wireframe of the “Removing Shopping List” after the shopping list is deleted (if any other shopping list is present in his account)

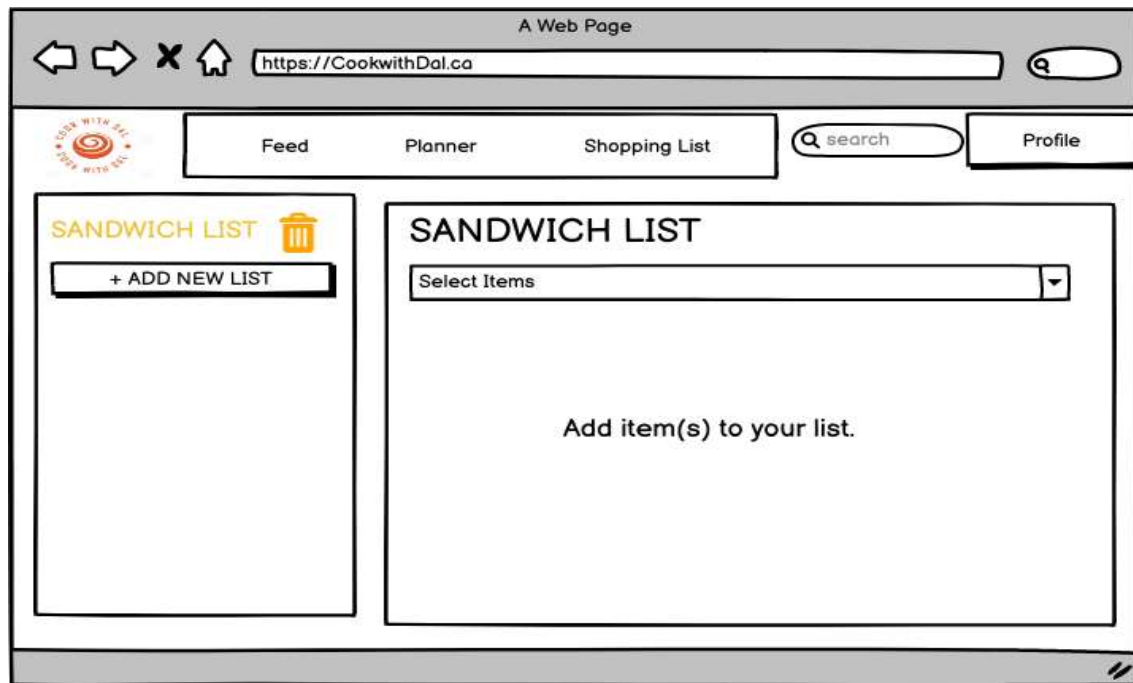


Figure 54: Shopping List Page with removed Shopping List wireframe (if any other shopping list is present in user account) [4]

**Figure 55** shows the wireframe of the “Removing Shopping List” after the shopping list is deleted (if no shopping list is present in his account)

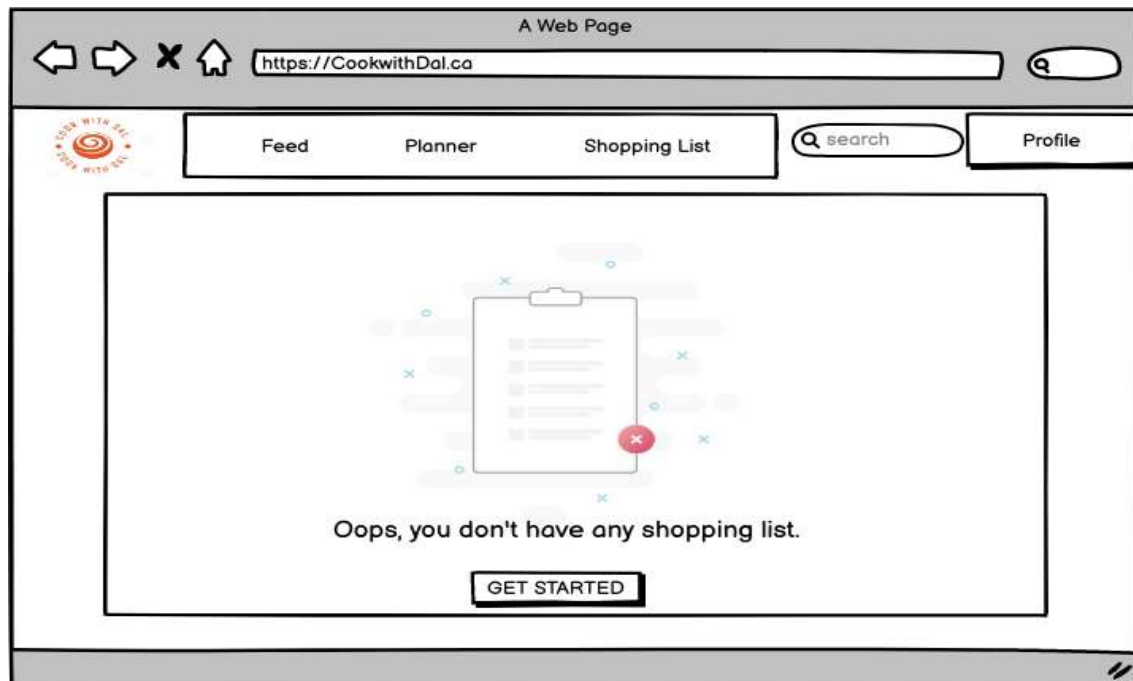


Figure 55: Shopping List Page with removed Shopping List wireframe (No shopping list is present in user account) [4]

## FEATURE – Meal Quantifier & Calorie Tracker

- **TASK 1 - MEAL QUANTIFIER.**

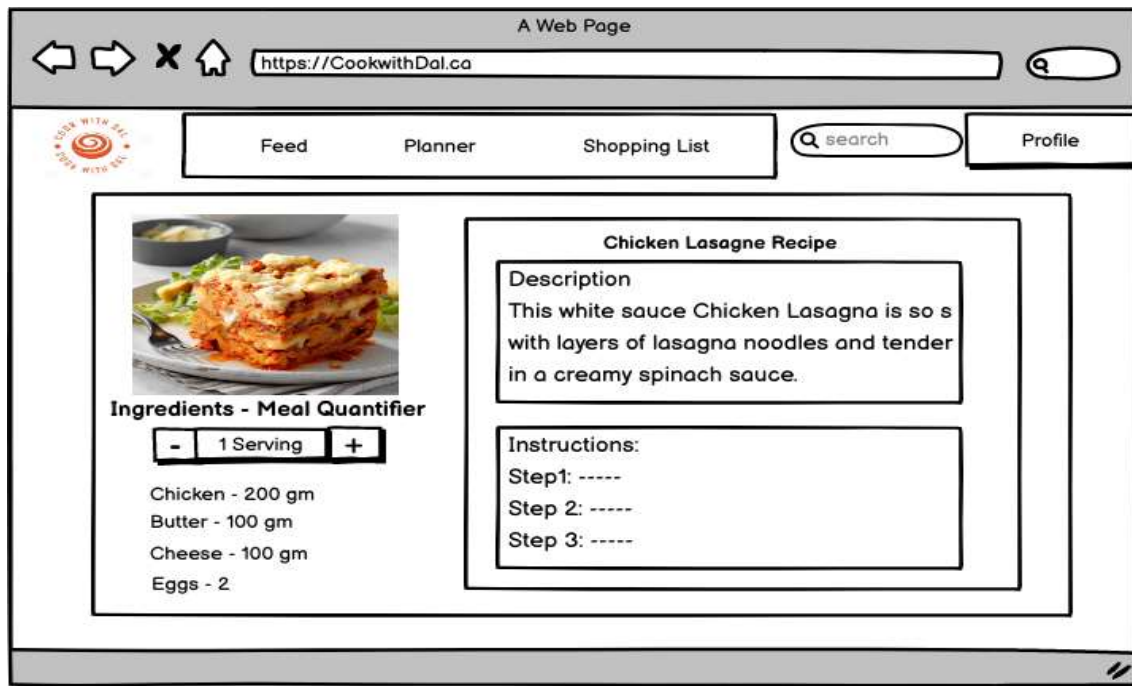


Figure 56: Meal quantifier page Wireframe [4]

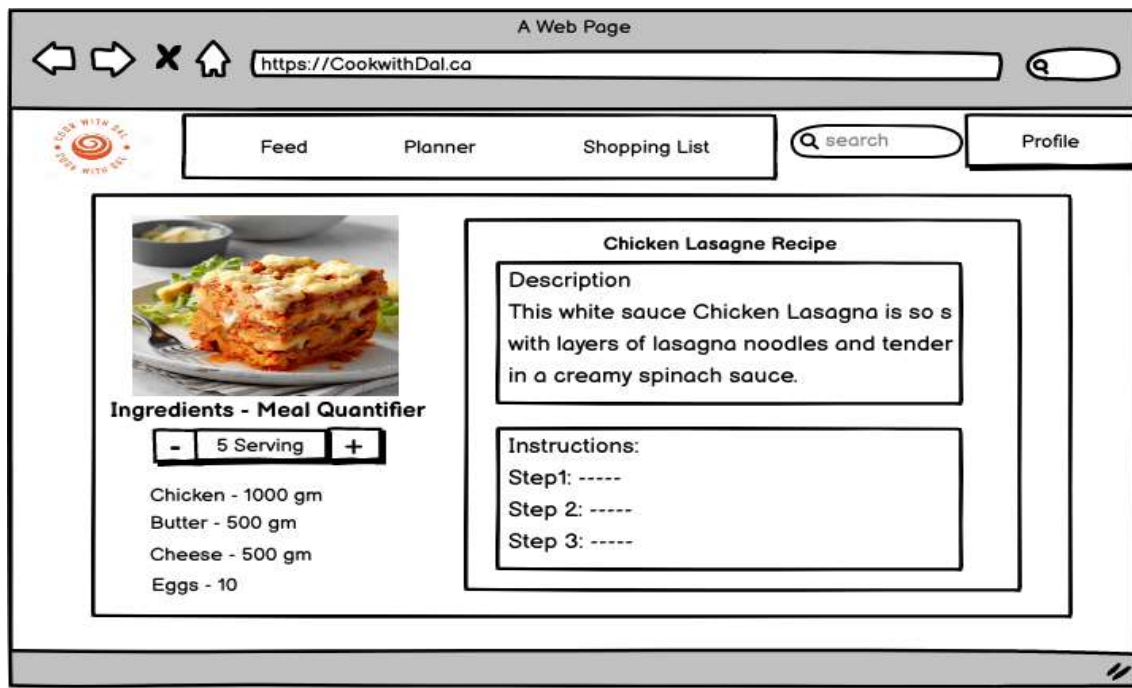


Figure 57: Increasing number of servings Wireframe [4]

- **TASK 2 - CALORIE TRACKER**



Figure 58: Calorie tracking Wireframe [4]

## FEATURE – Search and Filter Recipe

- **TASK 1 – SEARCH RECIPE**

The landing/feed page of the application has no recipes searched.

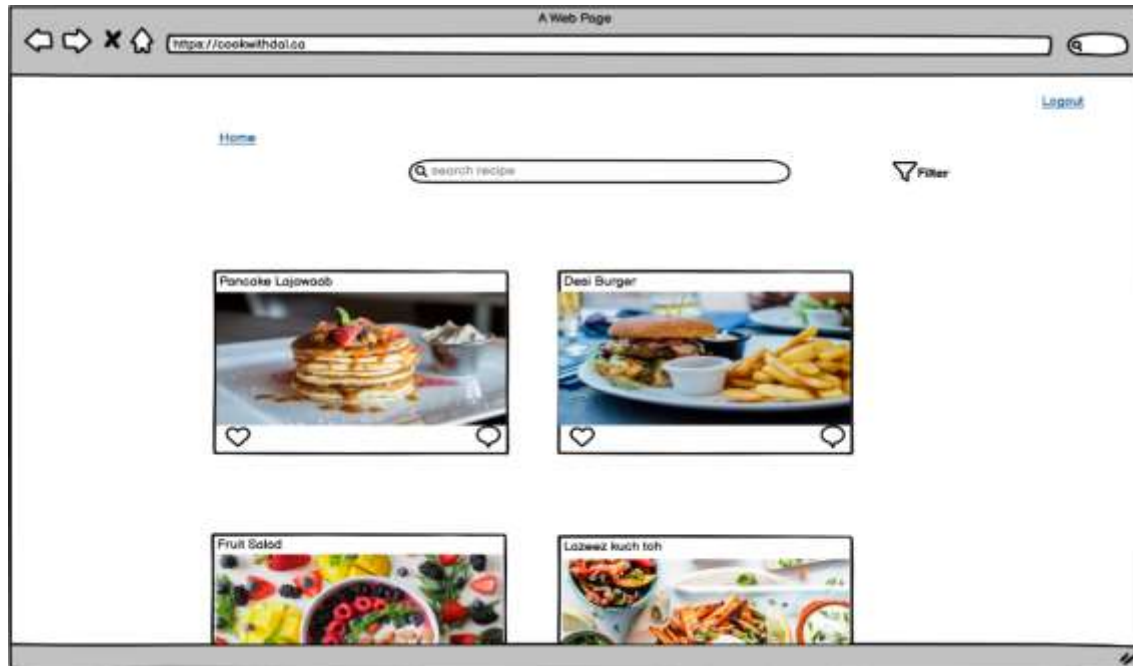


Figure 59: Wireframe – Landing Page without search applied [4]

User can search for a dish (example burger in this case).



Figure 60: Wireframe – Searched Recipe [4]

- **TASK 2 - FILTER RECIPE**

User can filter the results by clicking on the Filter button.



Figure 61: Wireframe – Filter options [4]

User can add or remove filters.



Figure 62: Wireframe – Filters edited [4]

Apply filter button will apply the filter.



Figure 63: Wireframe – Filter Applied [4]

## FEATURE – In-App Notification

- **TASK 1 – LIKE AND COMMENT NOTIFICATION**

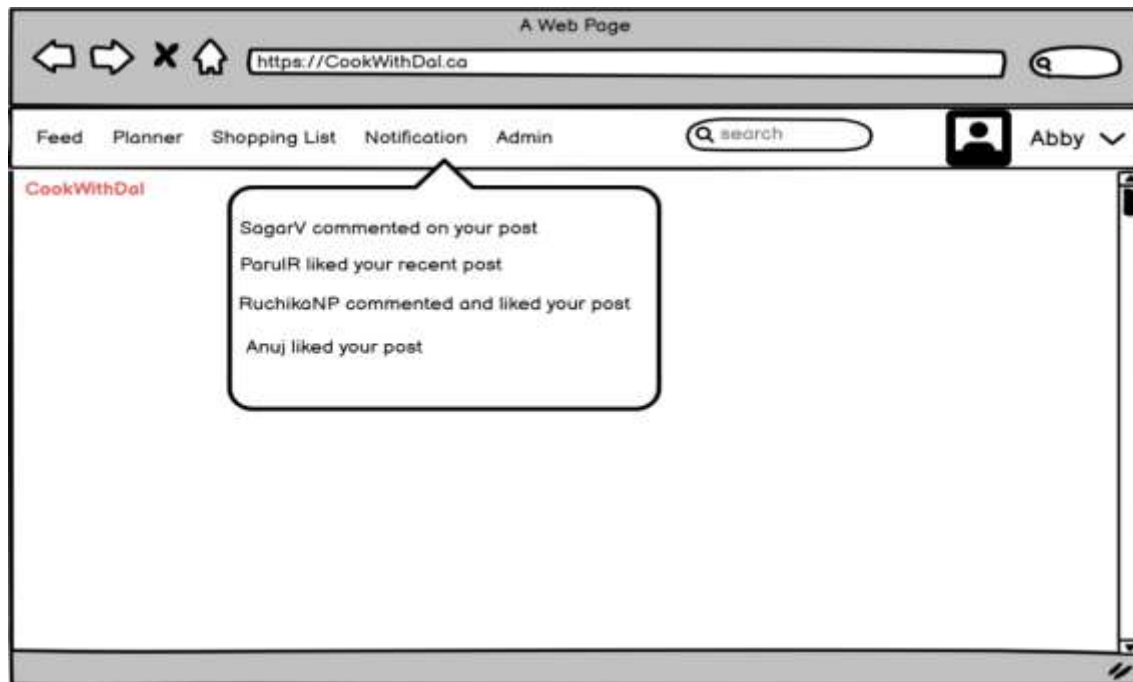


Figure 64: Notification dropdown panel [4]



## FEATURE – Meal Planner

- **TASK 1** – ADD THE RECIPE TO PLAN

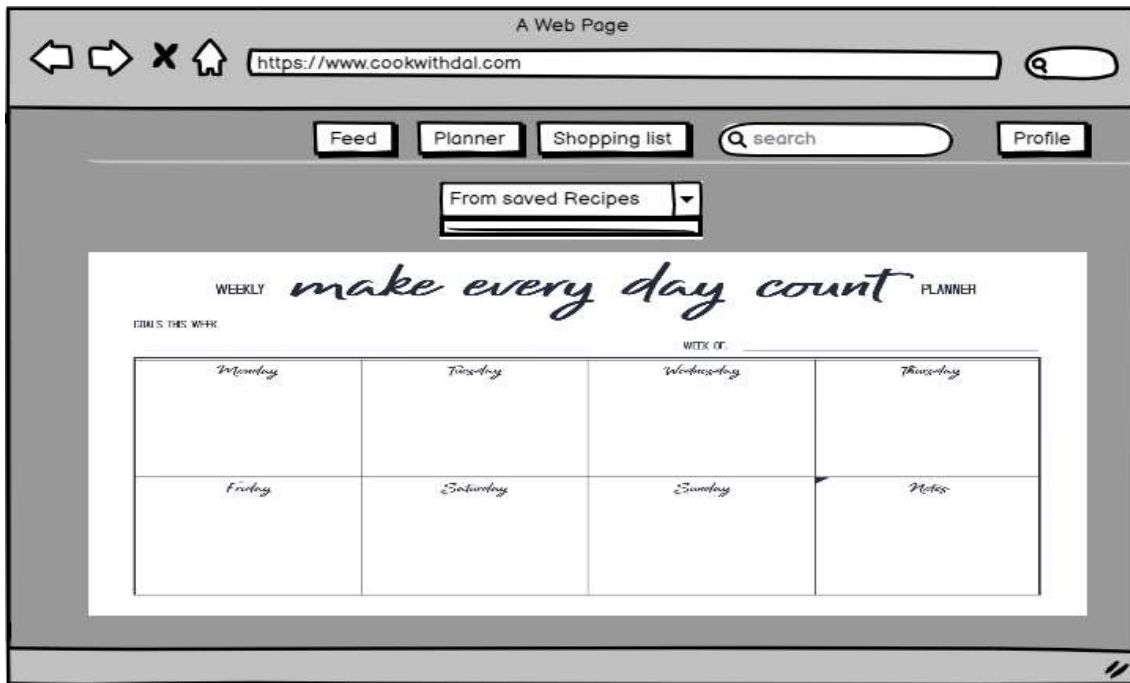


Figure 65: Recipe plan page [4]

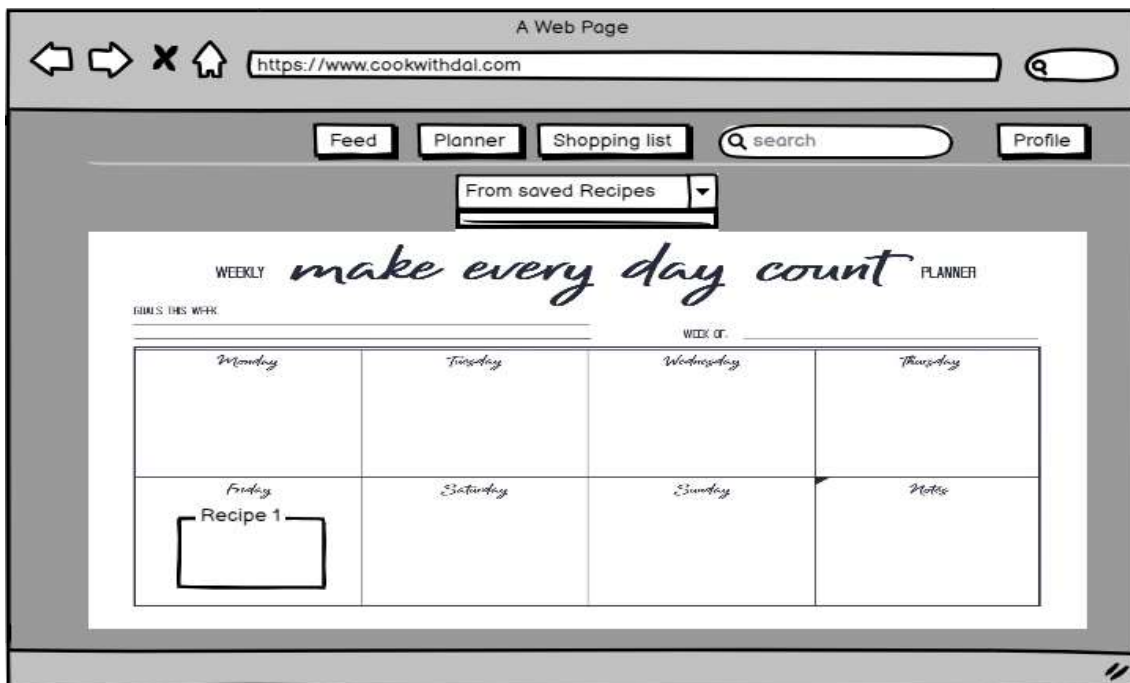


Figure 66: Recipe added in plan [4]

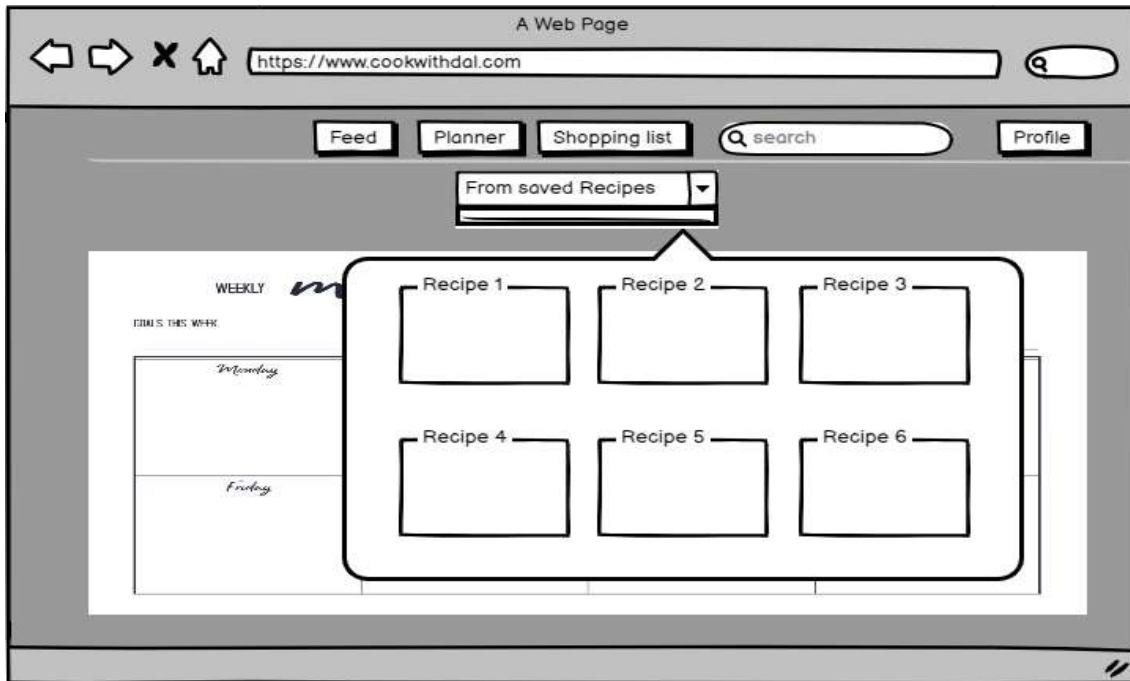


Figure 67: Saved recipes panel [4]

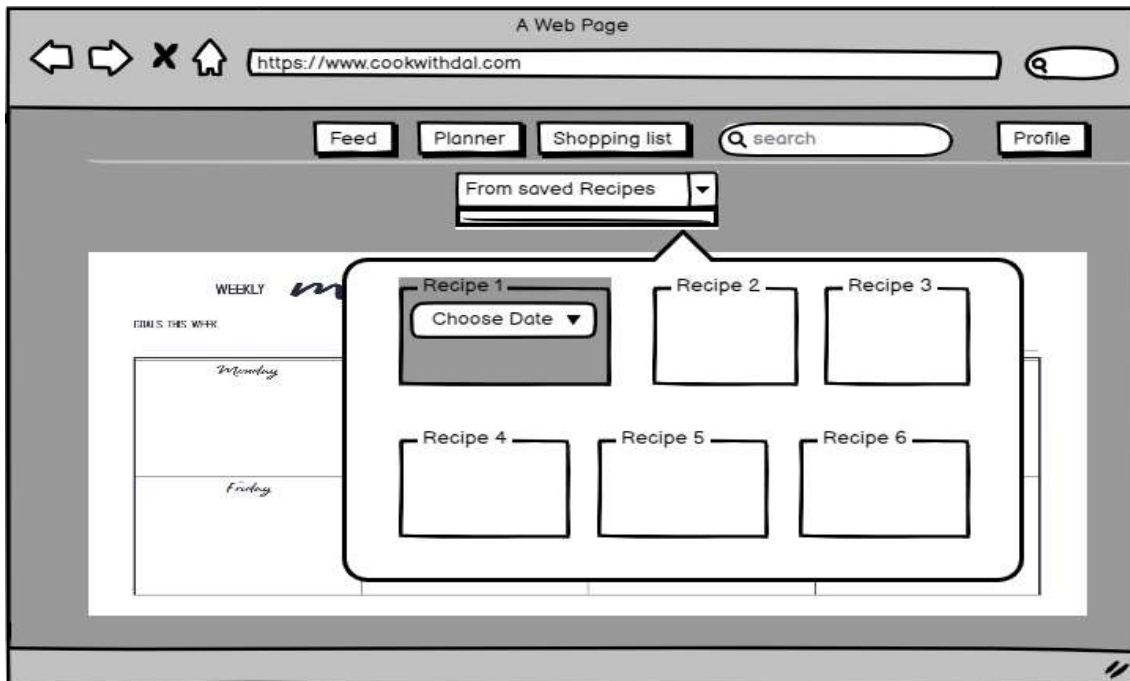


Figure 68: Choose data from saved recipe's panel [4]

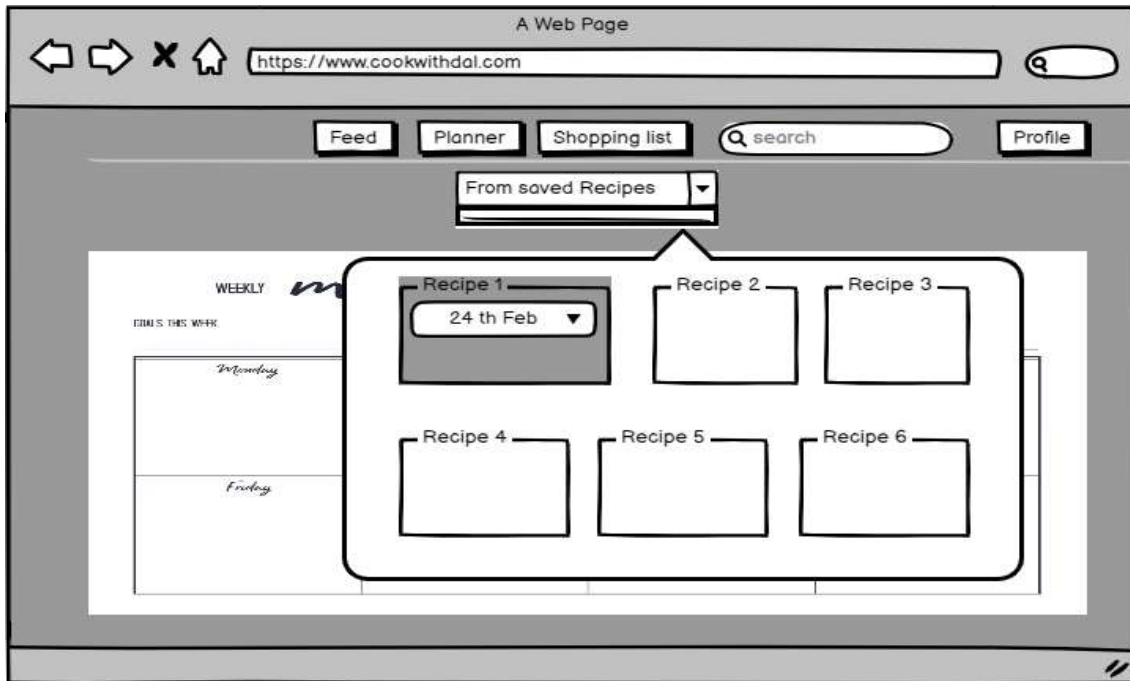


Figure 69: Date added [4]

- **TASK 2 - DELETE THE RECIPE FROM THE PLAN**

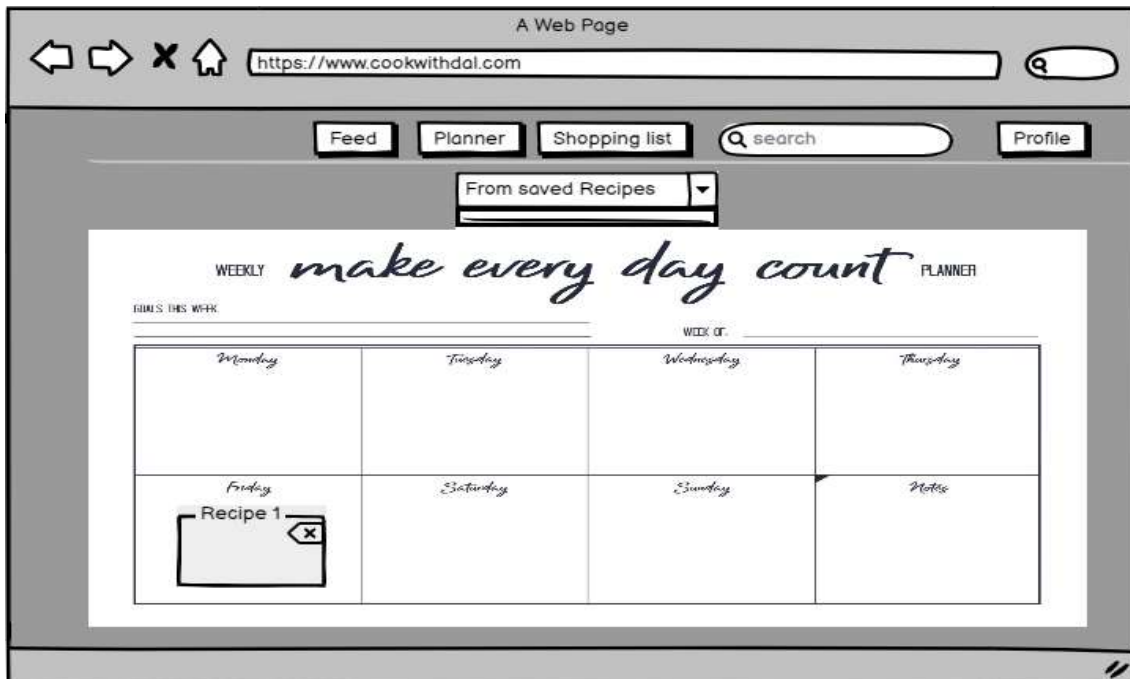


Figure 70: Recipe having delete option[4]

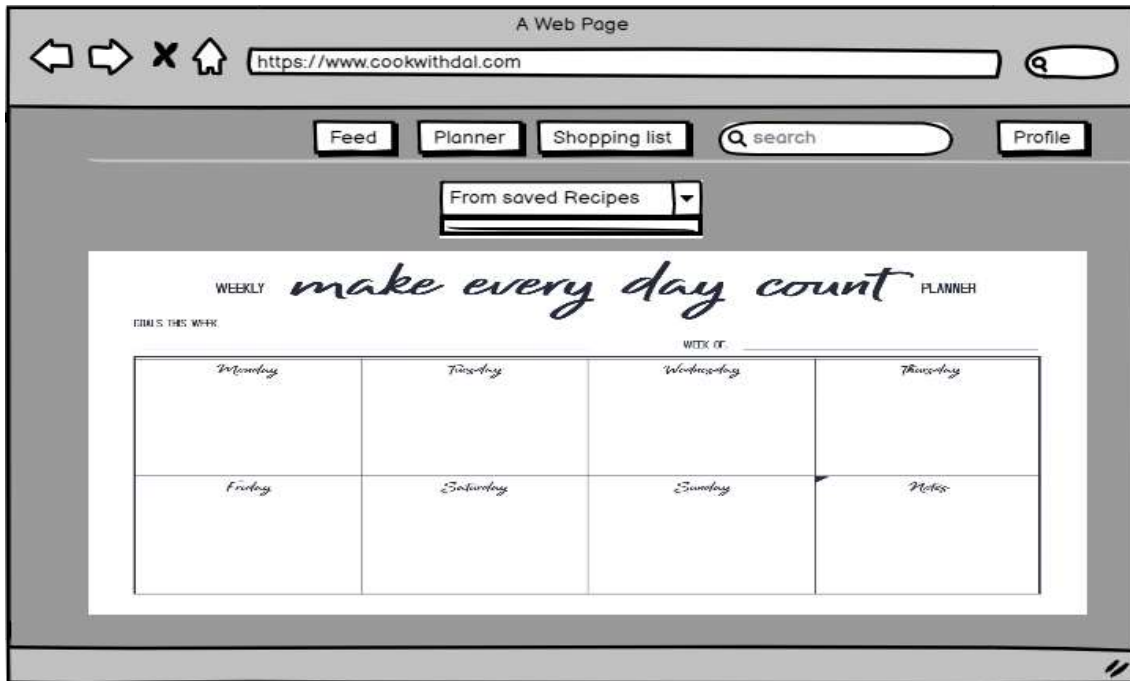


Figure 71: Recipe deleted [4]

## FEATURE - Save favorite recipes

- **TASK 1 - SAVE RECIPE**



Figure 72: Save recipe bookmark [4]

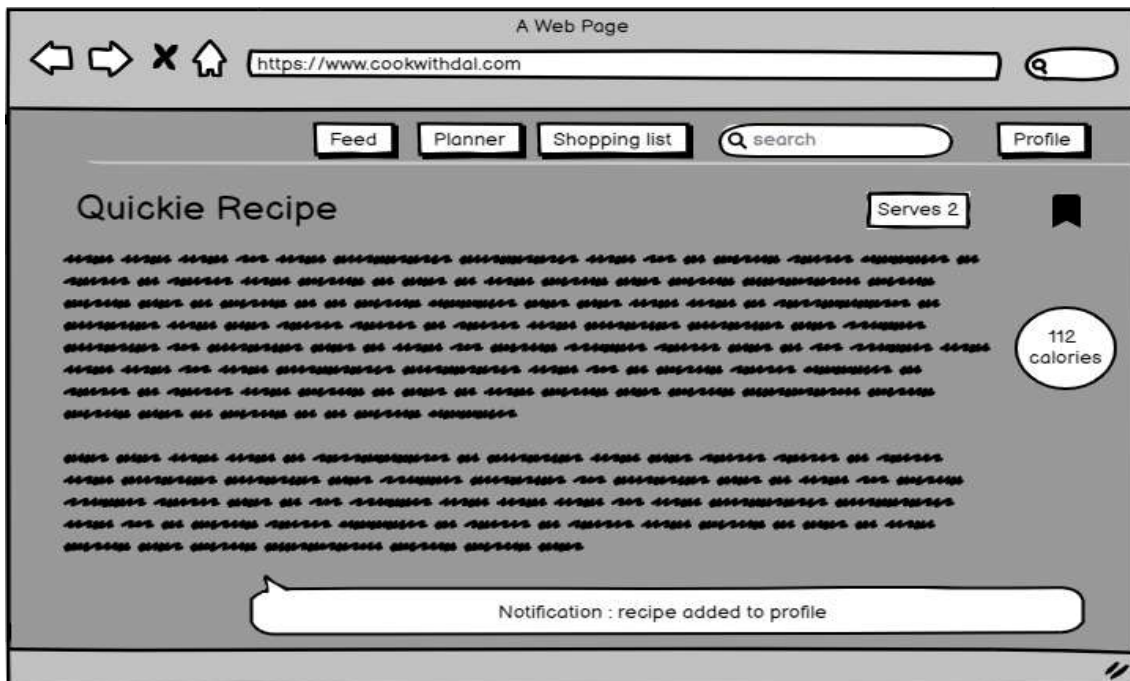


Figure 73: Recipe saved [4]

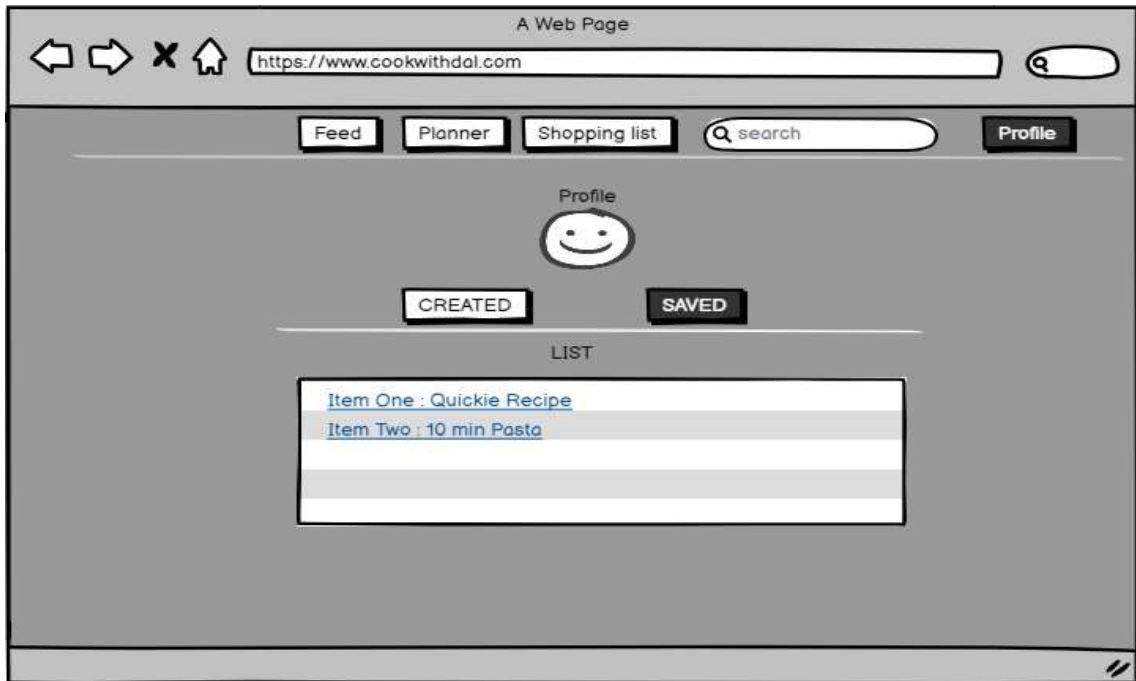


Figure 74: Saved recipe list [4]

- **TASK 2 - REMOVE RECIPE FROM SAVED**

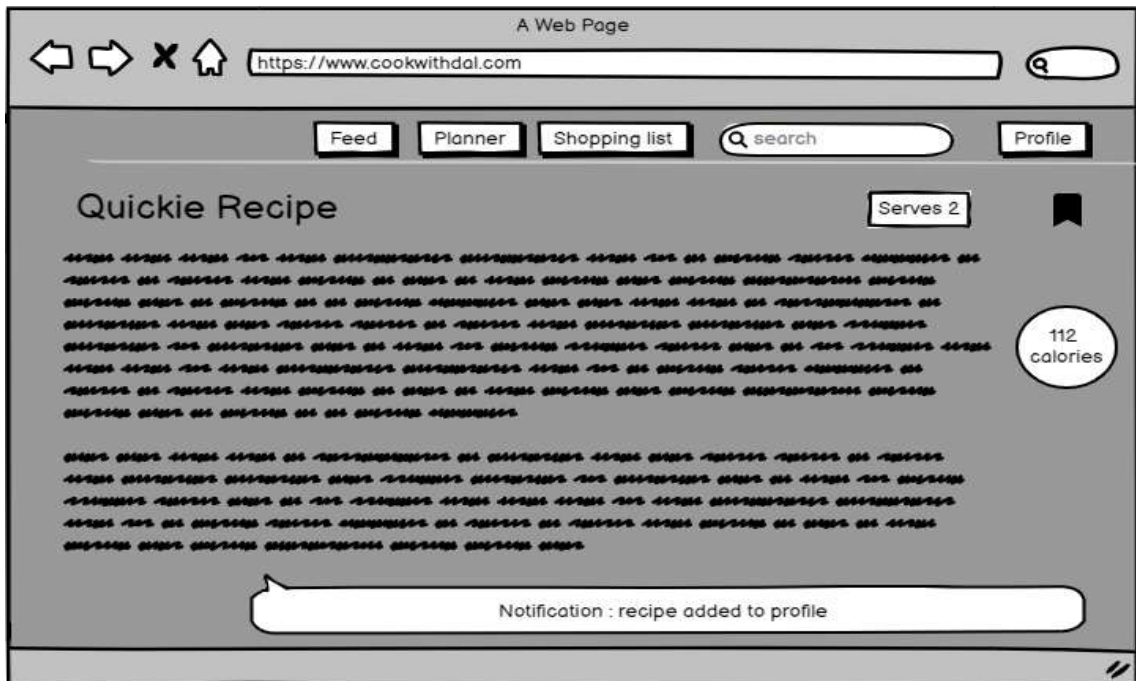


Figure 75: Removing recipe from saved [4]

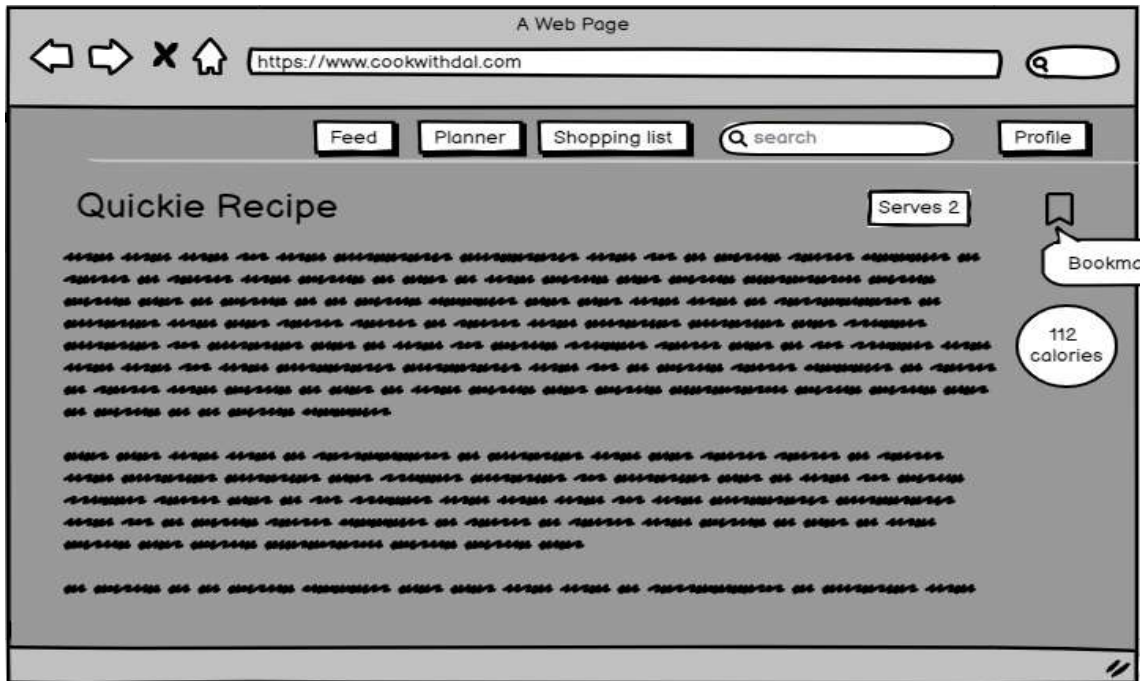


Figure 76: Removed recipe from saved [4]

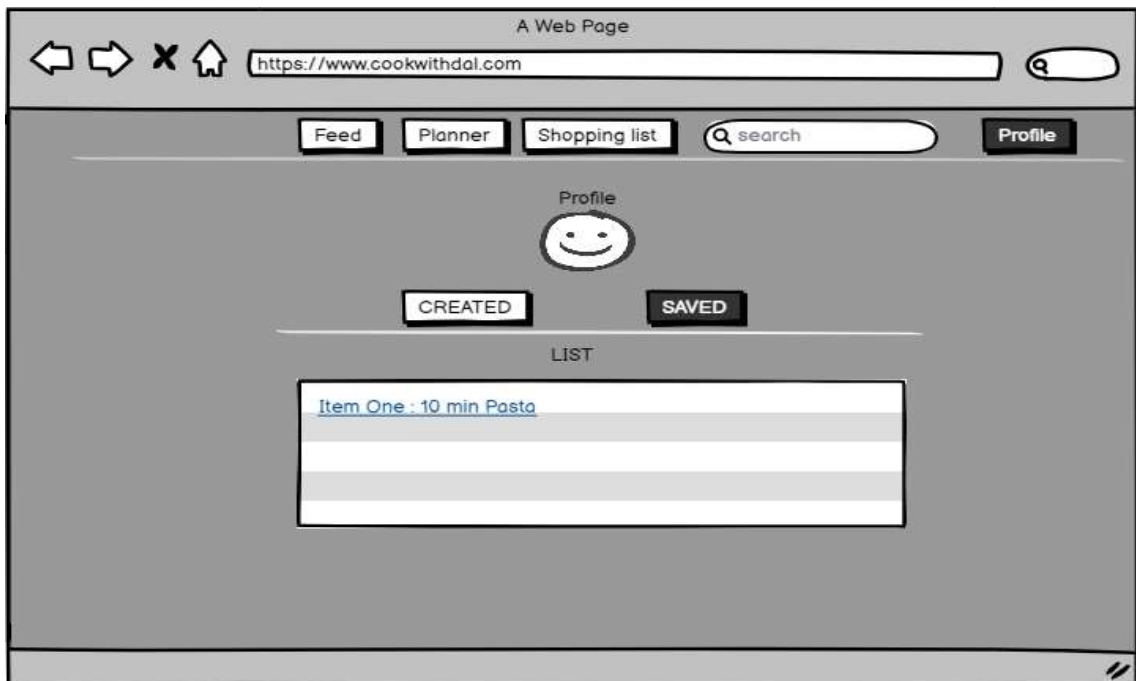


Figure 77: List showing recipe removed [4]

## 2.6 User Experience

Provide a short description of the scenarios in which you see your application being used or useful, and the use cases or tasks (e.g., request a taxi, send an email) users will be able to complete when using your application. Provide user task flow diagrams for each of your use cases, you may use the examples covered in class to help you. Only provide use cases, and task flow diagrams for the features you will be developing (as mentioned in Section 2.4). Justify your decisions.

### FEATURE - User Management

- **TASK 1 - CREATE USER ACCOUNT**

**SCENARIO:** Abby, a graduate student from abroad at Dalhousie, is keen on learning quick and simple recipes so she may prepare delicious meals at home. She recently got to know about our application from one of her friends and she wants to register with the website so she could save her time but at the same time relish the delicious food.

**PERSONA:** Abby (A graduate student)

**NEED:** Cooking delicious meals at home

**CONTEXT:** Learning quick and simple recipe

**USE CASE:** Create a user account.

1. User opens browser and enters URL on our website.
2. System displays the login page of our website.
3. User clicks on the sign-up link which is next to the message “Already have an account”.
4. System displays sign-up page which contains fields like email id, first name, last name, security question list, security question answer, password, and confirm password.
5. User enters all the information.
6. User clicks on the sign-up button.
  - 6.1. System displays "Email id already registered " message.
    - 6.1.1. System prompts to enter a new email id.
    - 6.1.2. User enters new email id and clicks on signup button.
  - 6.2. System displays an error message “Enter a valid email address” below the email id field.
    - 6.2.1. User enters valid email id.
    - 6.2.2. User clicks on signup button.
  - 6.3. System displays an error message “First name cannot be empty” below the first name field.
    - 6.3.1. User enters a valid first name.
    - 6.3.2. User clicks on signup button.
  - 6.4. System displays an error message “Security question must be selected” below security question drop down list.
    - 6.4.1. User selects a question from the dropdown list.



- 6.4.2. User clicks on signup button.
- 6.4.3. System displays an error message “Security answer cannot be empty” below the security answer field.
- 6.4.4. User provides an answer to the security question.
- 6.4.5. User clicks on signup button.
- 6.5. System displays an error message “Security answer cannot be empty” below the security answer field.
- 6.5.1. User provides an answer to the security question.
- 6.5.2. User clicks on signup button.
- 6.6. System displays an error message “Password must have minimum 8 characters and it must contain an uppercase, lowercase letter, special character, and number” below password field.
- 6.6.1. User enters a valid password.
- 6.6.2. User clicks on signup button.
- 6.7. System displays an error message “Password must match” below confirm password field.
- 6.7.1. User re-enters password in confirm password field.
- 6.7.2. User clicks on signup button.
- 6.8. System creates an account for the user.
- 6.9. System redirects the user to the login page.

## TASK FLOW:

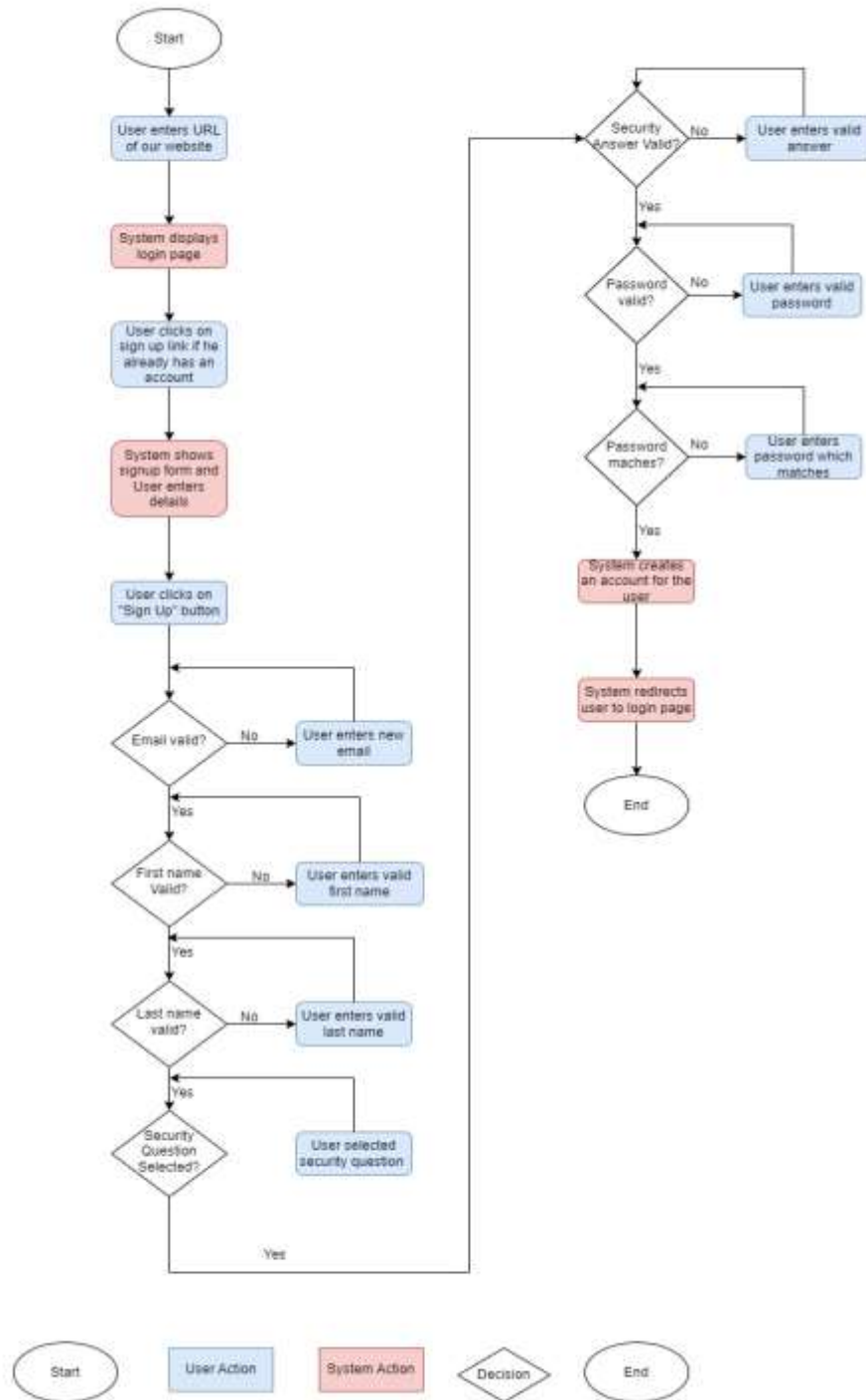


Figure 78: Taskflow Diagram to create user account [3]

- **TASK 2 - LOGIN INTO THE ACCOUNT**

**SCENARIO:** Abby, a student at Dalhousie, is a passionate cook and wants to share her favorite recipes so that other students can cook the same delicious food at home. She logs onto our website and shares her recipes with the users of our website.

**PERSONA:** Abby (A graduate student)

**NEED:** To share the recipe with everyone

**CONTEXT:** Other users can view the recipe and cook delicious food at their place.

**USE CASE:** Login into the account

1. User opens browser and enters URL on our website.
2. The system displays the login page of our website.
3. User enters their email id and password.
4. User clicks on the login button.
  - 4.1. System displays incorrect email id/password error message.
  - 4.2. System prompts user to enter email id and password.
  - 4.3. System displays incorrect email id/password error message.
  - 4.4. User clicks on forget password link.
  - 4.5. System prompts user to select a security question.
  - 4.6. User selects a question and provide an answer to the question.
    - 4.6.1. User clicks on the login button.
    - 4.6.2. System displays “answer does not match” error message.
    - 4.6.3. System prompts users to enter an answer again.
    - 4.6.4. Users provides an answer to the question and clicks on the login button.
  - 4.7. System prompts user to enter a new password and type in the password in confirm password field.
  - 4.8. User clicks on change password button.
    - 4.8.1. System displays an error message “Password must have minimum 8 characters and it must contain an uppercase, lowercase letter, special character, and number” alert message.
    - 4.8.2. System prompts user to type the new password and to confirm the password.
    - 4.8.3. User clicks on the change password button.
    - 4.8.4. System displays an error message “Password must match” below confirm password field.
    - 4.8.5. User re-enters password in confirm password field.
    - 4.8.6. User clicks on login button.
5. System updates password and authenticates user.
6. System redirects user to the landing page.

## TASK FLOW:

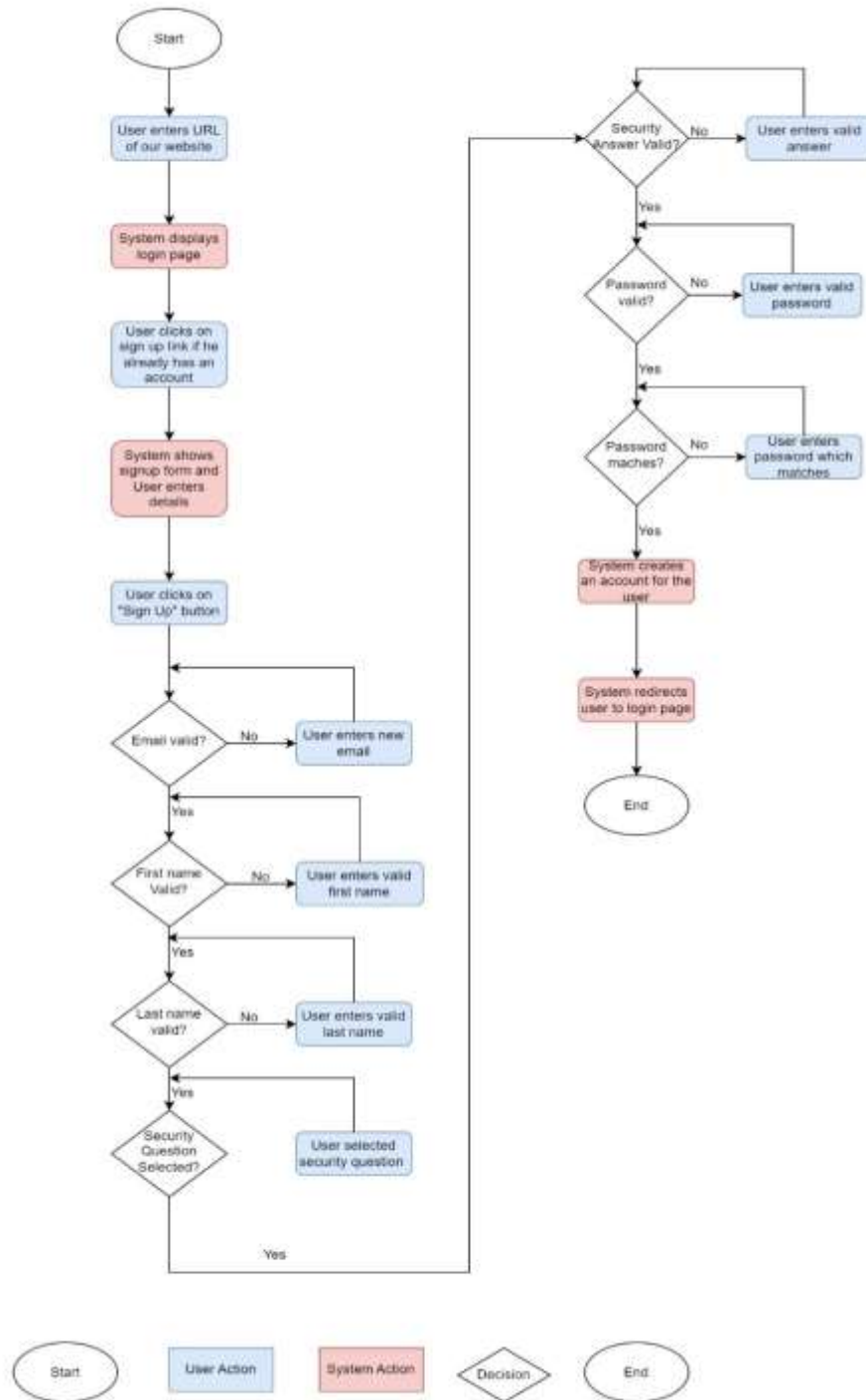


Figure 79: Taskflow Diagram to login into the account [3]

- **TASK 3 - LOGOUT FROM THE ACCOUNT**

**SCENARIO:** Abby, a graduate student, has logged into our website from a Desktop PC at Halifax Public Library. Since the PC is used by many people, she wants to make sure that her credentials are not compromised in any way. So, she logs out of our website.

**PERSONA:** Abby (A graduate student)

**NEED:** Securing credentials.

**CONTEXT:** Logging into our website from a public computer.

**USE CASE:** Logout of the account

1. System displays the landing page of our website.
2. User clicks on the drop-down list on the Navigation bar.
3. System displays list of options in the drop-down list.
4. User clicks on the logout option from that list.
5. System redirects user to the login page.

## TASK FLOW:

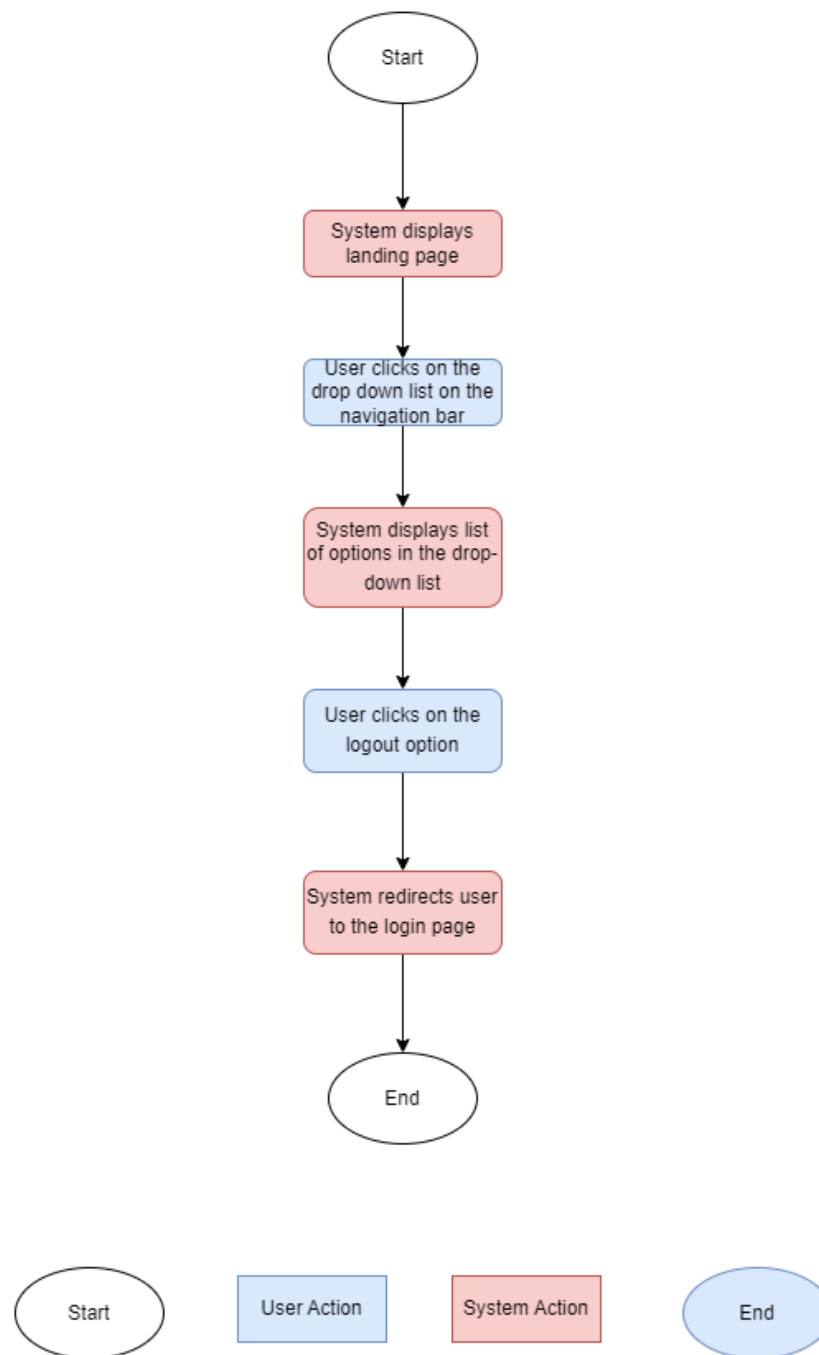


Figure 80: Taskflow Diagram to log out from the account [3]

## **FEATURE - User Profile Management/ User Dashboard**

- **TASK 1 - UPDATE USER PROFILE**

**SCENARIO:** Abby mentioned her first name and last name while account creation. However, she got married and her last name changed. So now she wants to update her last name in the profile.

**PERSONA:** Abby (A graduate student)

**NEED:** Some information entered needs to be updated if it entered incorrectly.

**CONTEXT:** Entered wrong last name while account creation and now user want to update it.

**USE CASES:** Update user profile

**Assumption:** User is already logged in the system.

1. User clicks on the user profile icon in the navigation bar.
2. On clicking the user profile icon, it displays menu.
3. User selects “Edit Profile” option in the menu.
4. System displays “Edit Profile” page.
5. User enters a value in the field that needs to be updated. Such as first name, last name, email, preferences, and display picture.
6. After adding the values, user clicks on the update button.
  - 6.1. User will be prompted to add first name by sending alert message as “First name cannot be empty”.
    - 6.1.1. User enters valid first name.
    - 6.1.2. User clicks on the update button.
  - 6.2. User will be prompted to add last name by sending alert message as “Last name cannot be empty”.
    - 6.2.1. User enters valid last name.
    - 6.2.2. User clicks on the update button.
  - 6.3. User will be prompted to add valid email address by sending alert message as “email must contain an ‘@’”.
    - 6.3.1. User enters valid email address.
    - 6.3.2. User clicks on the update button.
  - 6.4. User can add preferences (Vegetarian, Non-Vegetarian, Gluten-free recipes, etc) and clicks on update button.
  - 6.5. User can add profile picture and clicks on update button.
7. System updates User Profile.
8. System displays alert message as “User profile updated successfully”.

## TASK FLOW:

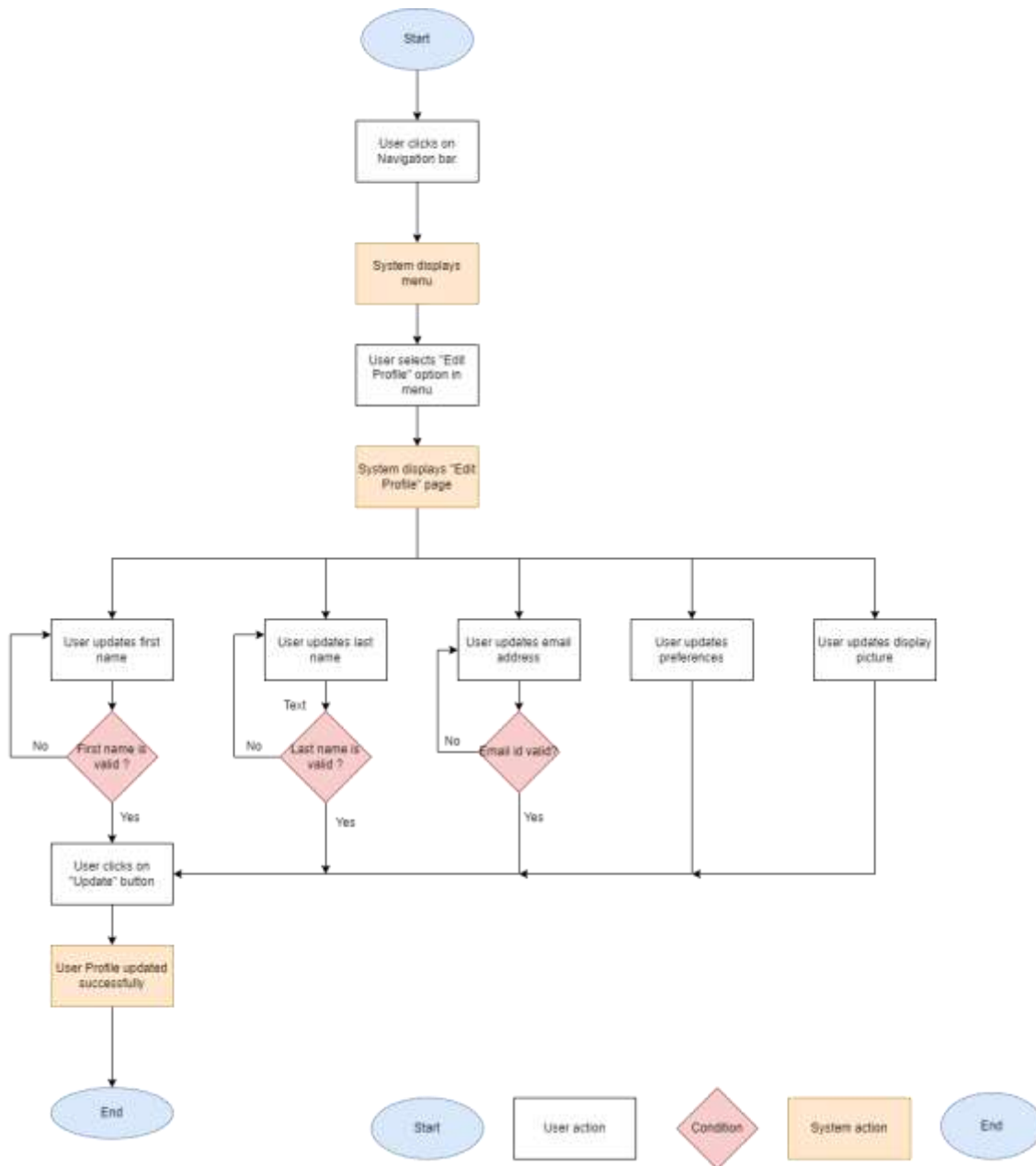


Figure 81: Task flow for updating User Profile [3]



- **TASK 2 - DEACTIVATE USER PROFILE**

**SCENARIO:** Abby has created two accounts using different email ids entered the wrong details like wrong last name while creating an account. Now they want to update their wrong details (Last name).

**PERSONA:** Abby (A graduate student)

**NEED:** Deactivate user account.

**CONTEXT:** User created two accounts with different email ids and want to deactivate one.

**USE CASES:** Deactivate user profile.

**Assumption:** User is already logged in the system.

1. User clicks on the user profile icon in the navigation bar.
2. On clicking the user profile icon, it displays menu.
3. User selects “Edit Profile” option in the menu.
4. System displays “Edit Profile” page.
5. User sees the “Deactivate” after scrolling down to the page.
6. User clicks on the “Deactivate” button.
7. System displays a confirmation box to confirm account deactivation.
  - 7.1. User clicks on the cancel button.
    - 7.1.1. System removes confirmation box from the screen.
    - 7.1.2. System displays profile page.
  - 7.2. User clicks on confirm button.
8. System deactivates User Profile.
9. System displays alert message as “User profile deactivated successfully”.

## TASK FLOW:

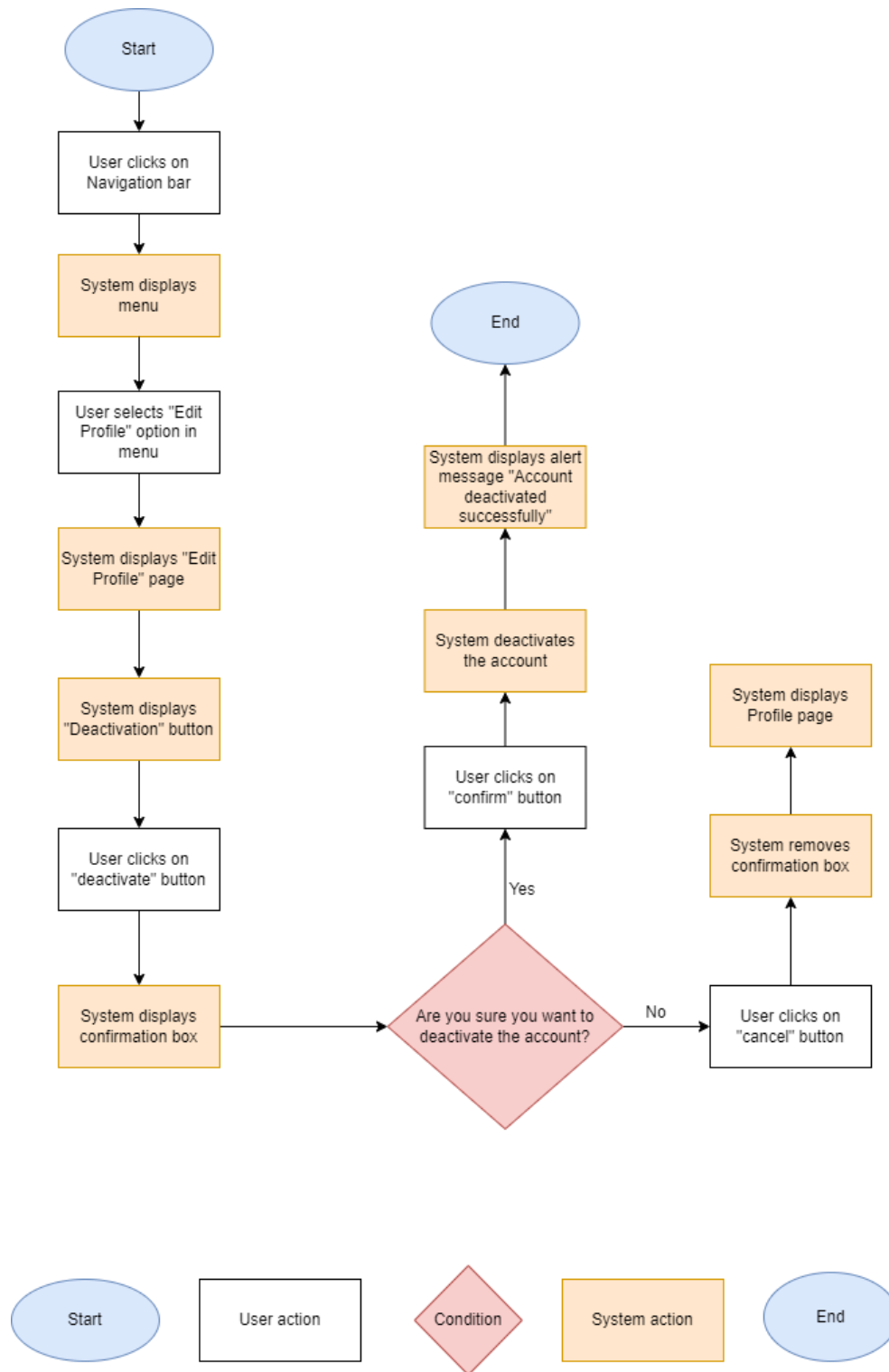


Figure 82: Task flow for Deactivating User Profile [3]

- **TASK 3 - DISPLAY LIST OF RECIPES (CREATED AND SAVED)**

**SCENARIO:** Abby likes to save the recipes that are easy to make and delicious, so that he can make it without searching at other places. This saves Abby's time. Also, she can see her own created recipes to keep a track of her own recipes and can make variations in his recipes in for future posts.

**PERSONA:** Abby (A graduate student)

**NEED:** To easily see the saved recipes and keep a track of user's own created post.

**CONTEXT:** Student is in hurry and want to look for the easy and quick recipe from the saved section or created from his/her own created posts.

**USE CASES:** Display List of created and saved recipes.

1. User clicks on the user profile icon in the navigation bar.
2. On clicking the user profile icon, it displays menu.
3. User selects "Profile" option in the menu.
4. System displays "Profile" page.
5. User sees two tabs "Created" and "Saved" recipes.
  - 5.1. User clicks on "Created" to see the post created by the user.
  - 5.2. User clicks on "Saved" to see the saved posts which is saved by the user.

## TASK FLOW:

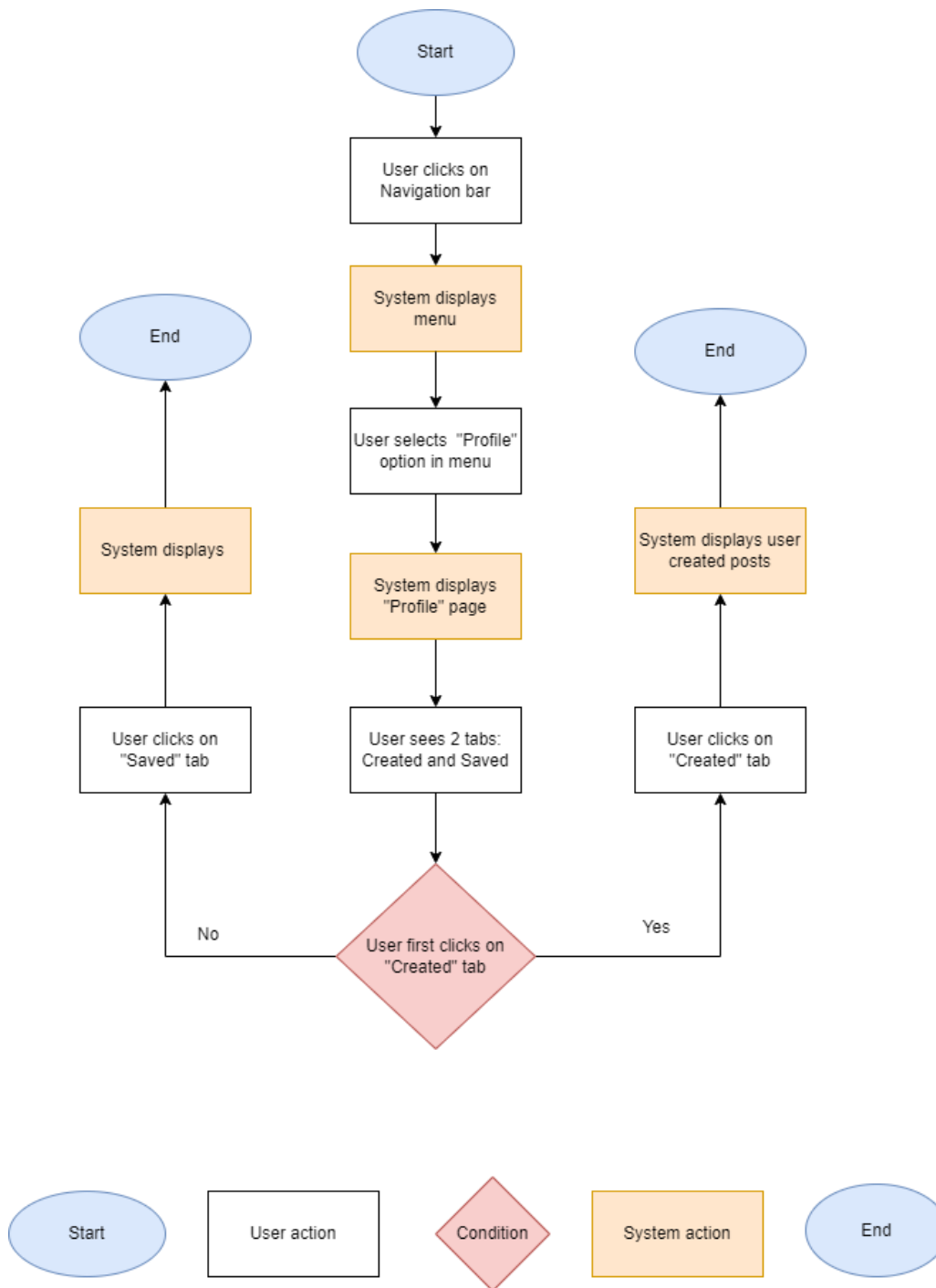


Figure 83: Task flow for displaying list of recipes [3]

## FEATURE - Recipe Management

- TASK 1- Add new recipe**

After logging into the website, the user will click the “+” icon from the feed page to add a post of new recipe to the feed. Then user will enter all the required information and the validation will take place. If the validation is not passed, then the system will show error message. At the end, when user will click submit the recipe will be added.

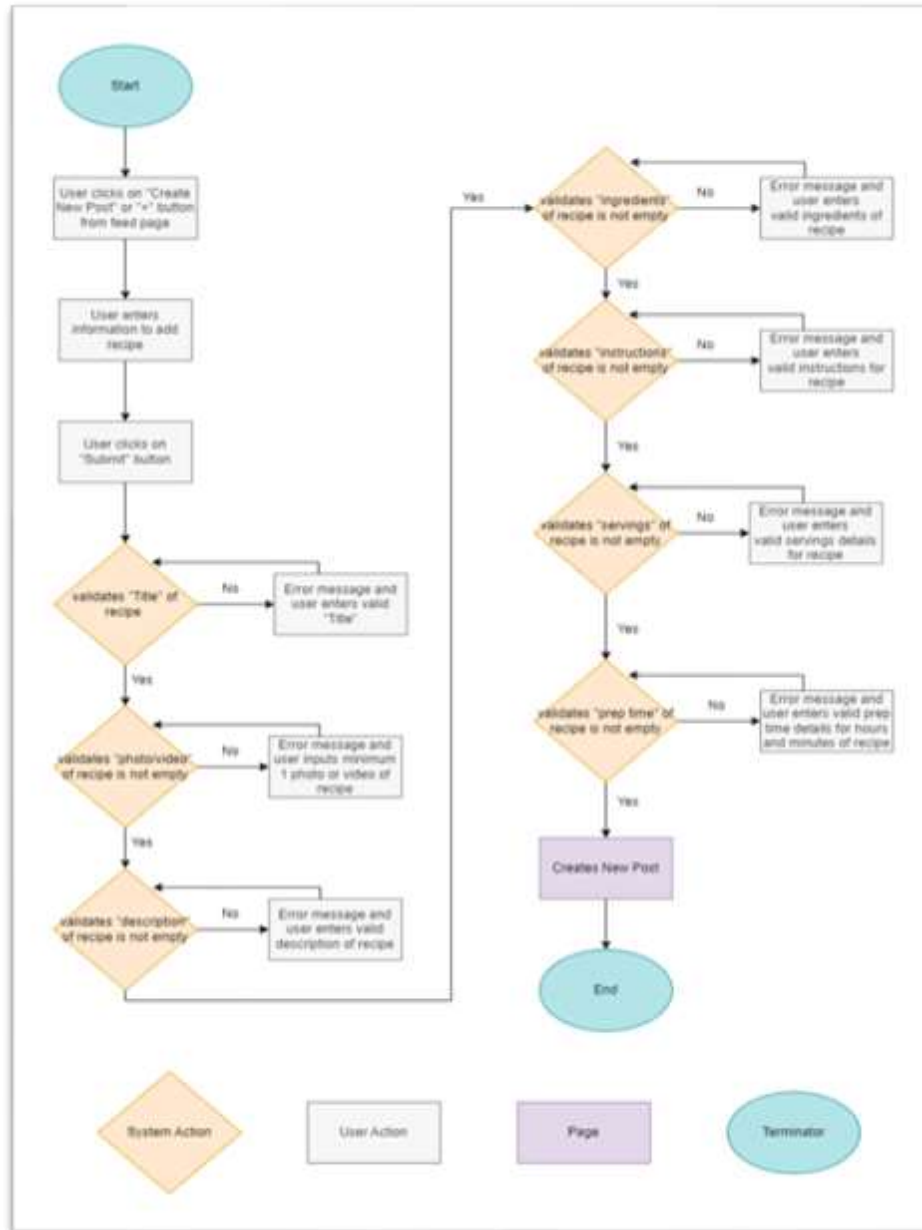


Figure 84: Taskflow for adding recipe [3]

- **TASK 2 - Update Recipe**

If the user wants to update the information of the created recipes, then the user will go to profile section. From that under “Created” tab the user will be able to see all the recipes created. After that the user will click on the ellipses and clicks on “Update Recipe” button and enters the information that needs to be updated. At the end, by pressing submit button the system will show that the recipe is updated.

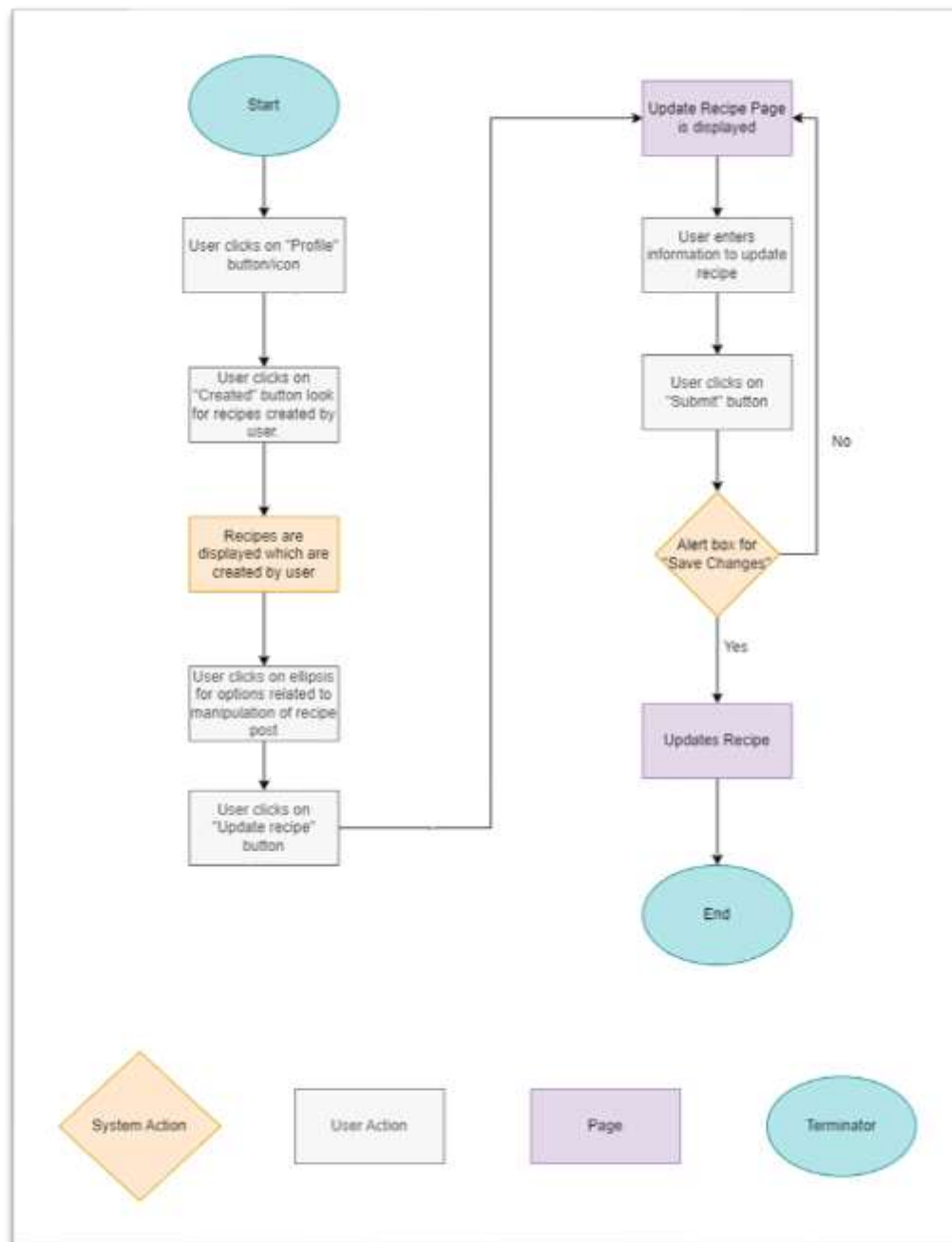


Figure 85: Taskflow to update recipe [3]

- **TASK 3 - Delete Recipe**

If the user wants to delete one of the created recipes, then the user will go to profile section. From that under “Created” tab the user will be able to see all the recipes created. After that the user will click on the ellipses and click on “Delete Recipe” button. At the end, by pressing the submit button the system will ask the user for deletion confirmation. As soon as the user clicks “yes” button the recipe will be deleted.

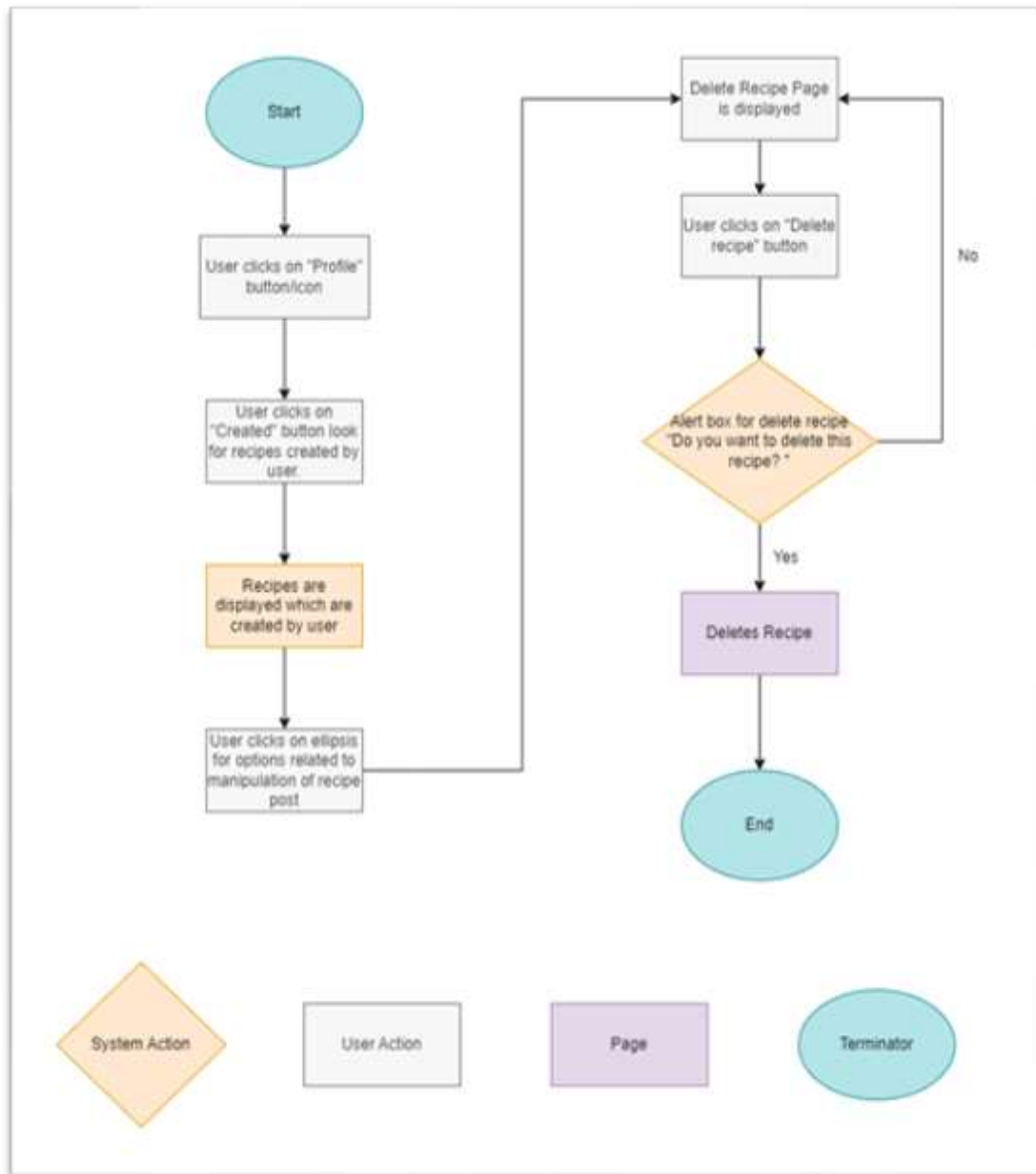


Figure 86: Taskflow for deleting recipe [3]

- **TASK 4** - Display Recipe

When the user will enter the credentials and logs in to the website, it will be redirected to the feed page. Here, all the recipes are displayed. If there are no recipes to display, then the system will show greeting message.

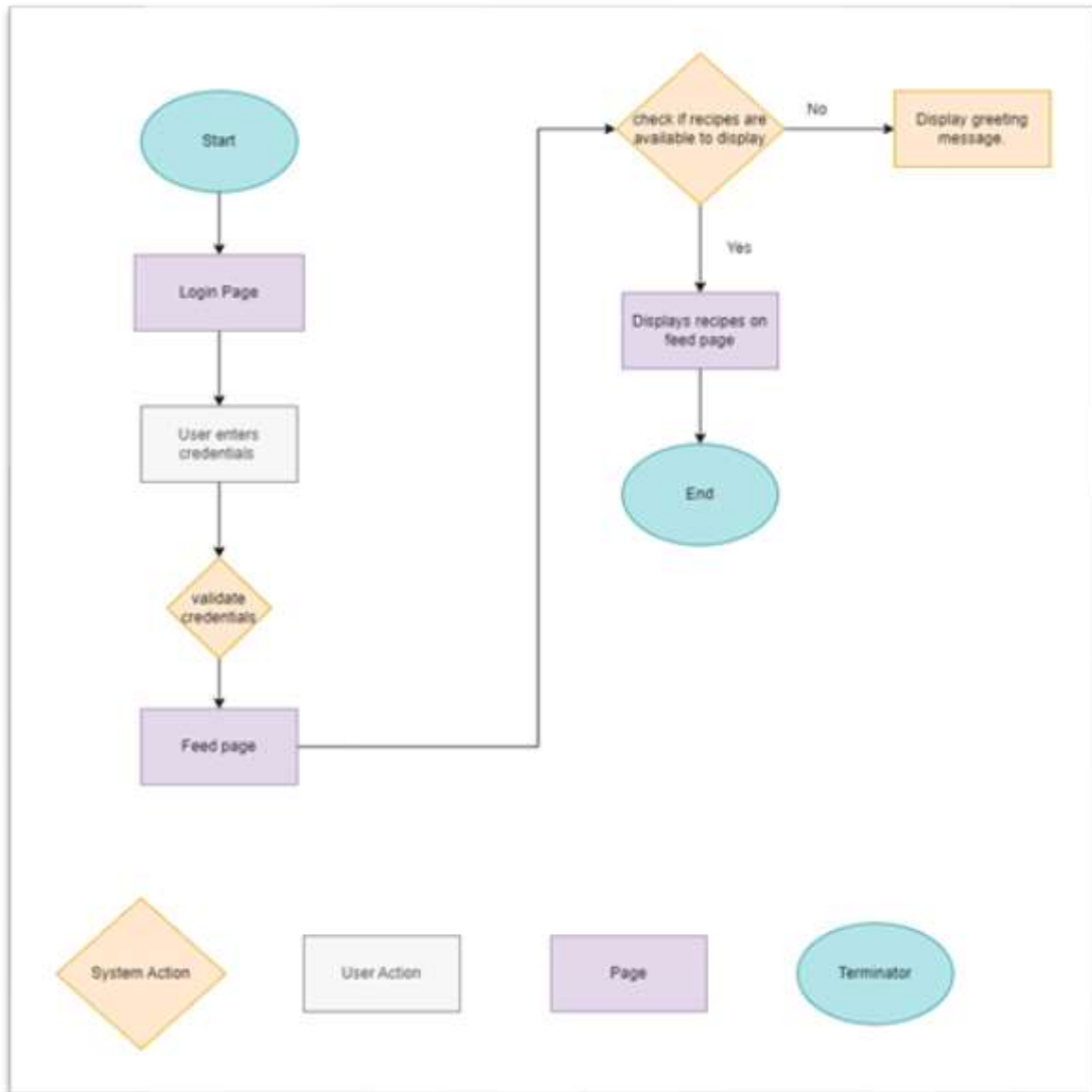


Figure 87: Taskflow for displaying recipe [3]



## FEATURE - Follow/Unfollow User

### • **TASK-1: FOLLOW USER**

**SCENARIO:** Lucie wants to check other users' profile and found Abby's profile interesting and wants to follow her. So lucie visits Abby's profile and follows her.

**PERSONA:** Lucie (A graduate student)

**NEED:** User needs to click on "follow" button to follow that page/person.

**CONTEXT:** User found the other user's profile interesting so he/she wants to follow it to get new updates or see recipe posts from the followed user.

### USE CASES: Follow user

**Assumption:** User is already logged in the system.

1. User clicks on the author's name whose recipe posts they found interesting as shown in fig.88.
2. On clicking the author's link, it displays user profile to whom user wants to follow.
3. User sees Abby's Profile having "Follow button" as shown in fig.88.
4. User clicks on "Follow" button
5. User sees two buttons "Followed" and "Unfollow".

### TASK FLOW:

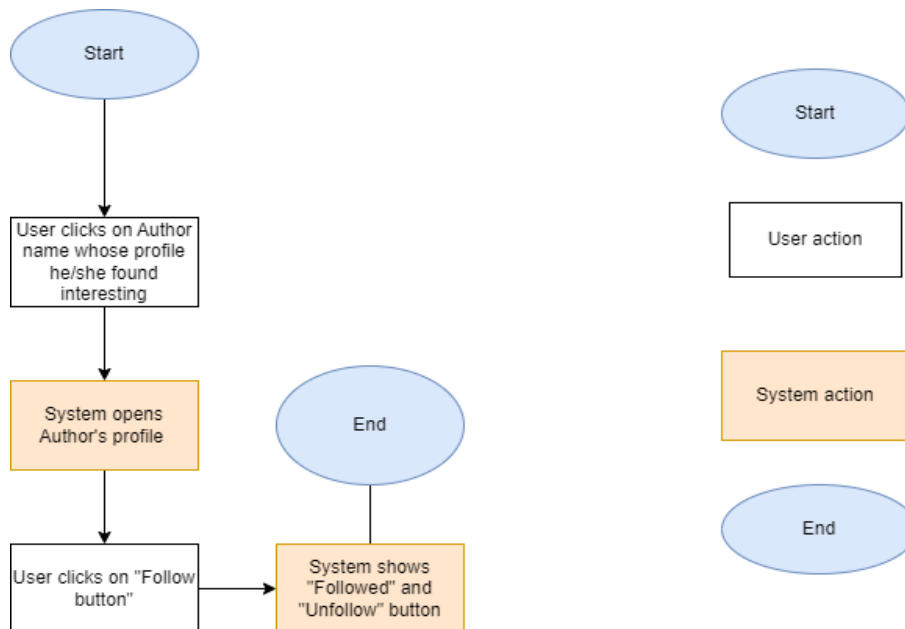


Figure 88: Task flow to follow user [3]

- **TASK-2: UNFOLLOW USER**

**SCENARIO:** Lucie found irrelevant posts or not so interesting of the Abby and now she wants to unfollow her. So lucie visits Abby's profile and unfollows her.

**PERSONA:** Lucie (A graduate student)

**NEED:** User needs to click on "unfollow" button to unfollow that page/person.

**CONTEXT:** User found the other user's profile irrelevant or not based on his/her interests, so he/she wants to unfollow it so that they don't get any new updates or see recipe posts from them.

**USE CASES:** Unfollow user

**Assumption:** User is already logged in the system.

1. User clicks on the author's name whose recipe posts they found not interesting as shown in fig. 89.
2. On clicking the author's link, it displays user profile to whom user wants to unfollow.
3. User sees Abby's Profile having "Unfollow button" as shown in fig. 89.
4. User clicks on "Unfollow" button.
5. User sees two buttons "Unfollowed" and "follow".

**TASK FLOW:**

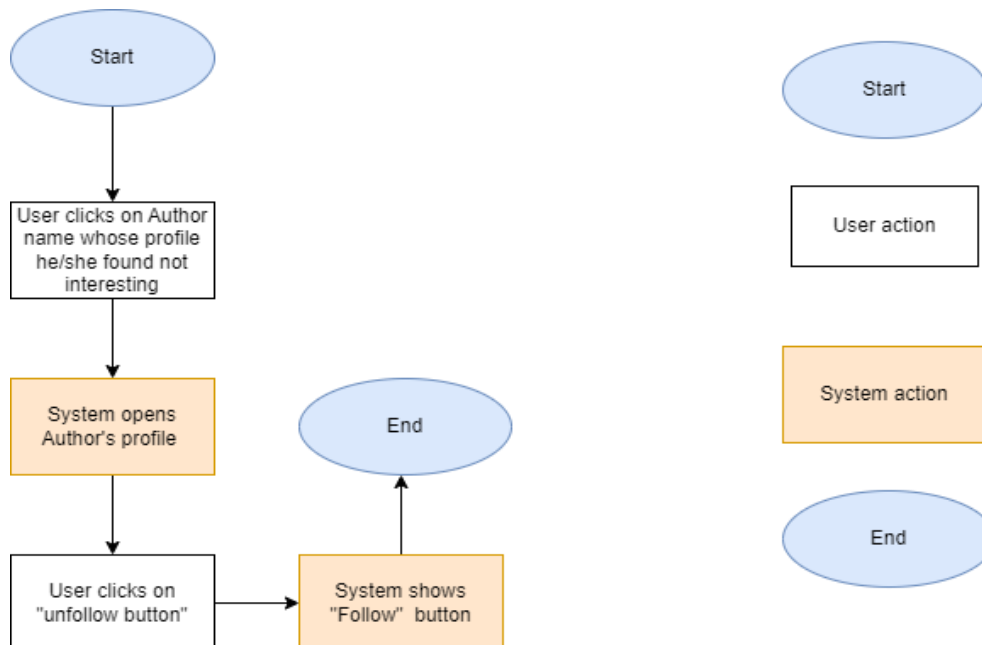


Figure 89: Taskflow to unfollow user [3]

## **FEATURE - Comments & Likes**

- **TASK 1 - VIEW COMMENT**

**SCENARIO:** Lucie is looking for ideas for her next meal. She is browsing through the feed looking for easy to prepare recipes. She found a recipe which interests her. It makes sense to read the comments and see what other users think about the recipe.

**USE CASE:**

**Assumption:** User is already logged in to the system.

1. User clicks on “Comments” button associated with a recipe.
2. Comments section will appear
  - 2.1 There are existing comments associated with the recipe
    - 2.1.1 User scrolls through the comments
  - 2.2 There are no comments associated with the recipe
    - 2.2.1 A message will appear indicating the user that there are no comments
3. User clicks on close button
4. Comments section will hide

## TASK FLOW:

After logging into the application, the user can see recipes on their feed. When a user clicks on the comments button of a recipe, the user can view all the comments associated with that recipe.

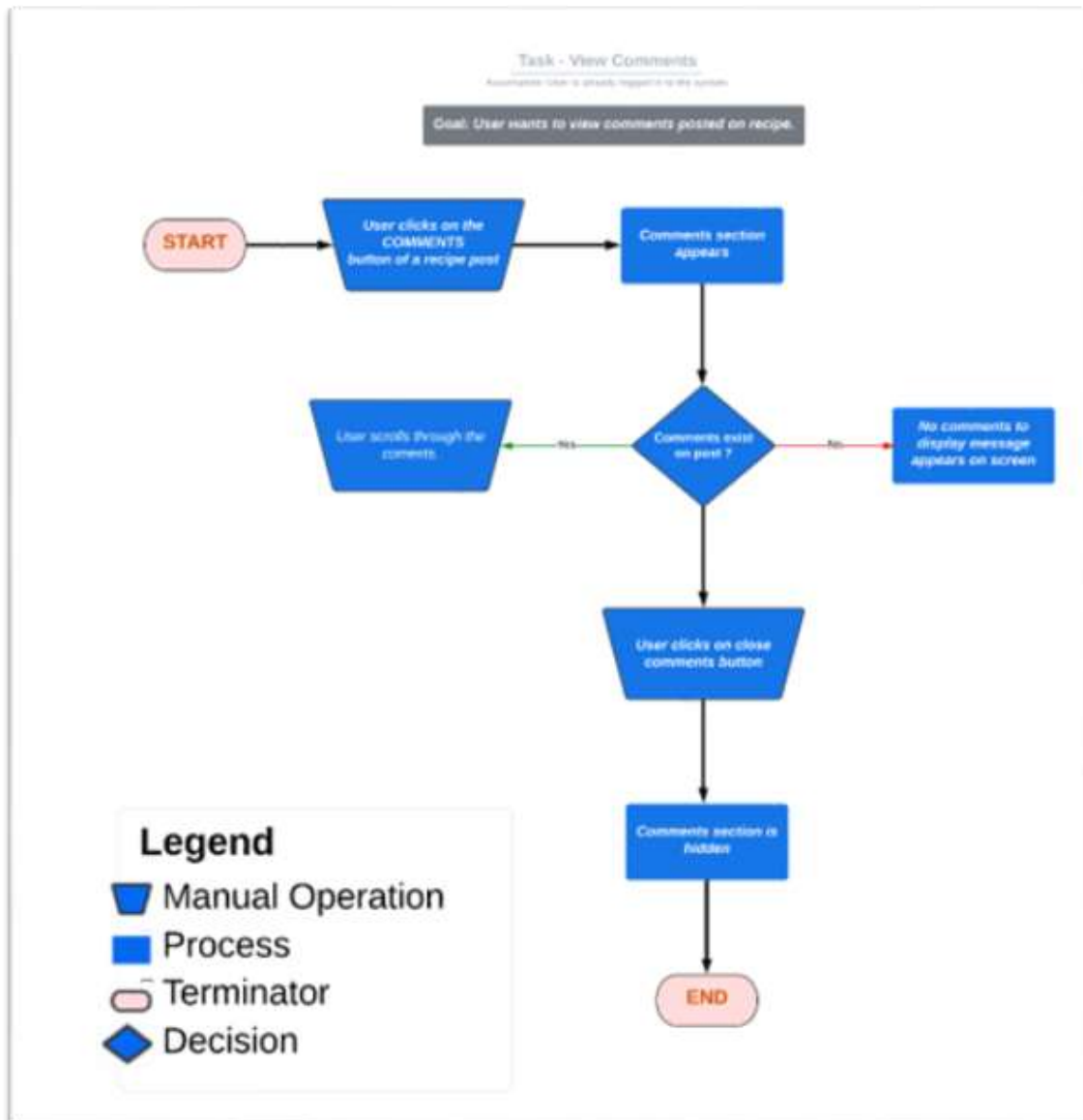


Figure 90: Task flow – View comments [5]

- **TASK 2 - ADD COMMENT**

**SCENARIO:** Lucie tried a recipe she was interested in. The recipe worked out well for her. She is happy with the recipe and would like to share her thoughts with other users and possibly give a compliment to the author of the recipe. She opens the post, navigates to the comments section so she can add a new comment.

**USE CASE:**

**Assumption 1:** User is already logged in to the application.

**Assumption 2:** User has opened the comments section of a recipe.

**Assumption 3:** Input box available to the user for entering their comments.

1. User has option to enter comments.
  - 1.1 User tries to post comment without entering any comment.
    - 1.1.1 The user is notified with a warning message stating that comment text is required.
  - 1.2 User enters the comment.
2. User clicks on Add Comment button
3. Comment will be added to the recipe
4. User clicks on close button
5. Comments section hides

## TASK FLOW:

If the user is interested in a recipe and wishes to give feedback, or share their thoughts, they can do so by adding comments to a recipe.

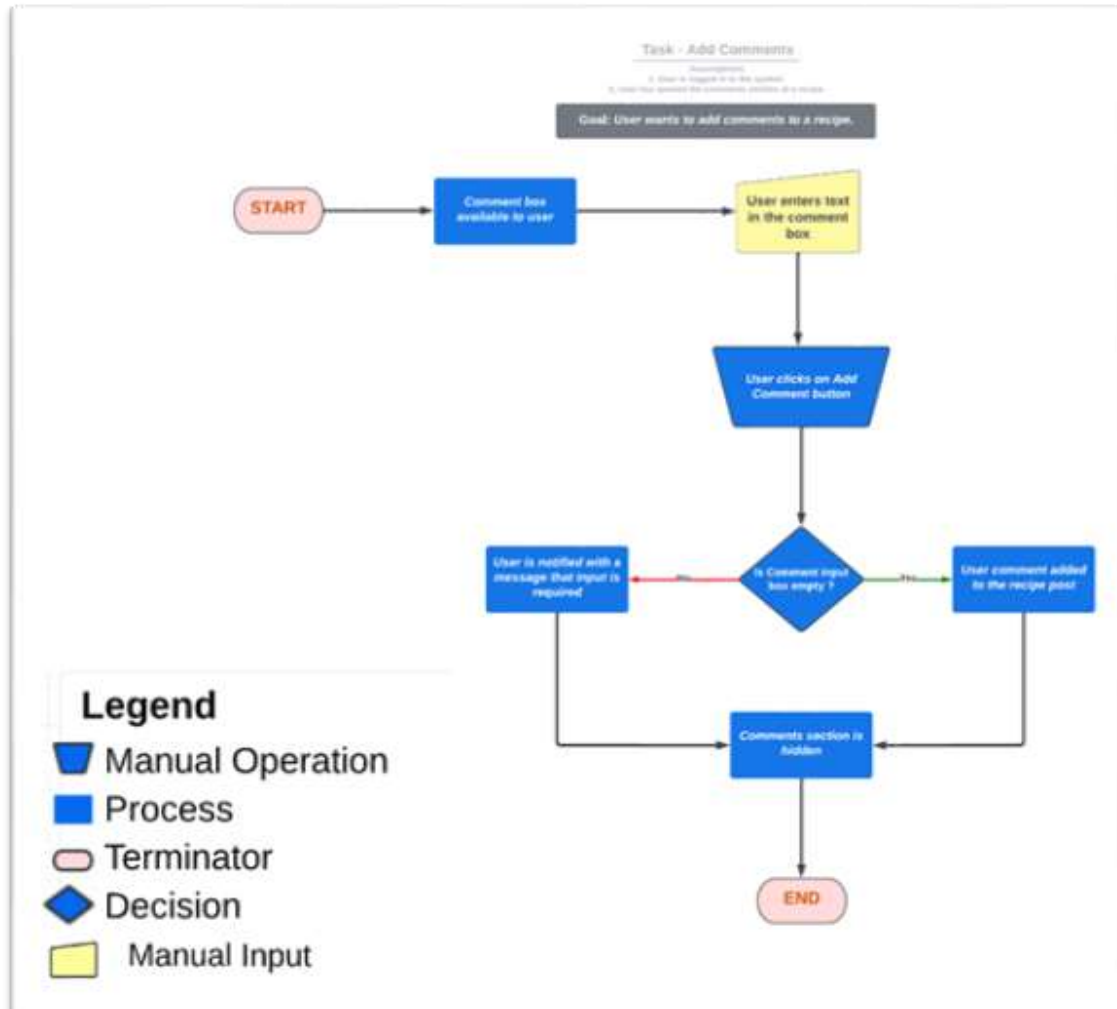


Figure 91: Task flow – Add comment [5]

- **TASK 3 - UPDATE COMMENT**

**SCENARIO:** Lucie has posted a new comment on a recipe but soon realizes a typing error in the comment. She would like to edit the comment in order to fix the typo.

**USE CASE:**

**Assumption 1:** User is logged in.

**Assumption 2:** User has opened the comments section of a recipe.

**Assumption 3:** User has posted at least one comment in the recipe.

1. User clicks on options button beside their comment
2. User selects Edit
3. User modifies the comment
  - 3.1 User decides to update the comment.
    - 3.1.1 User clicks on Save button.
    - 3.1.2 A confirmation message will appear to notify user of the changes.
  - 3.2 User decides to discard the changes.
    - 3.2.1 User clicks on Cancel button.
4. User can see the updated comment
5. User closes the comments section

## TASK FLOW:

The user can update their existing comment if they wish to modify it.

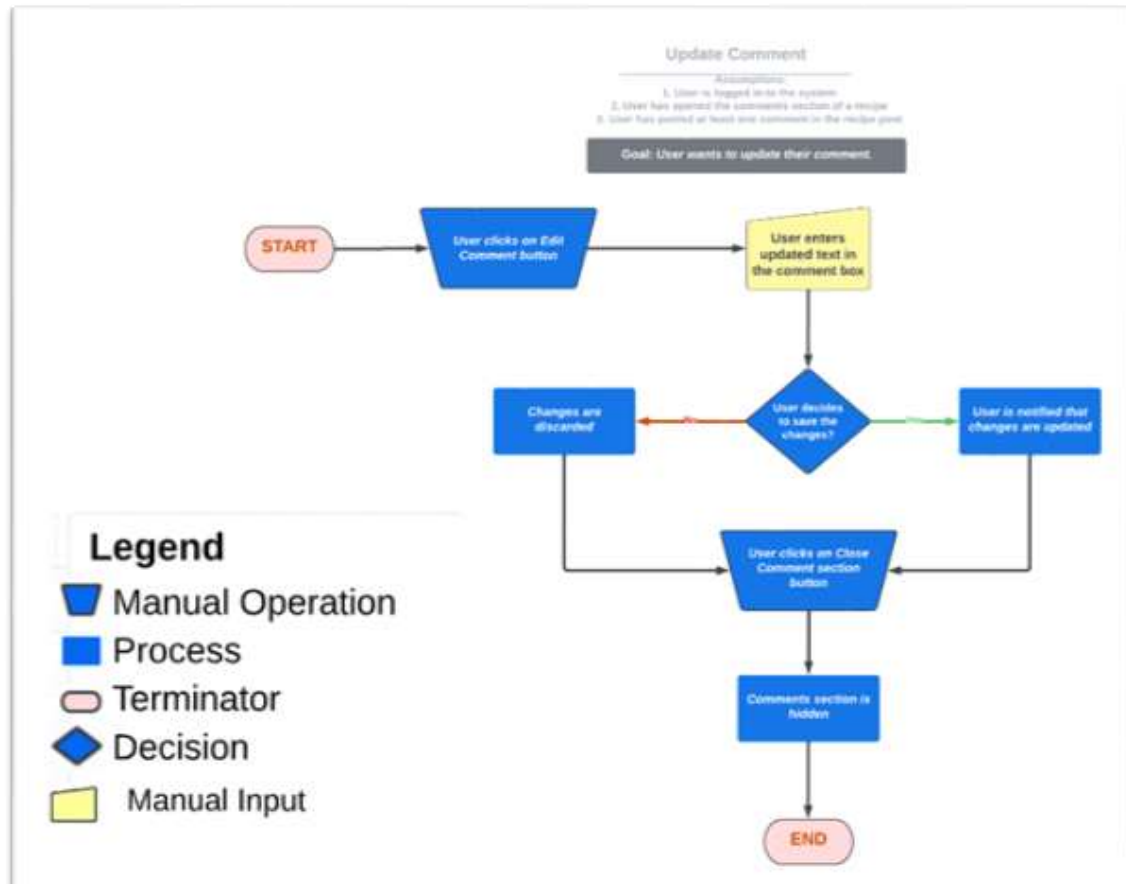


Figure 92: Task flow – Update comment [5]



- **TASK 4 - DELETE COMMENT**

**SCENARIO:** Lucie made a dish following a new recipe from the application. Unfortunately, though, the taste was not great. She decides to share her experience with others by adding a new comment. In her new comment, she expressed her feelings about the dish. Soon she realizes that she missed a key ingredient, and it was her fault and not the recipe. She goes back to the post and decides to delete the comment.

**USE CASE:**

**Assumption 1:** User is logged in.

**Assumption 2:** User has opened the comments section of a recipe.

**Assumption 3:** User has posted at least one comment in the recipe

1. User clicks on options button beside their comment
2. User selects Delete
3. User is prompted with a confirmation dialog
  - 3.1 User clicks on Delete.
    - 3.1.1 Comment is deleted.
  - 3.2 User clicks on Cancel.
    - 3.2.1 Comment is not deleted.
4. User can see the comments section
5. User clicks on close comments section

## TASK FLOW:

The comments on a post can be deleted by using the delete button on a comment.

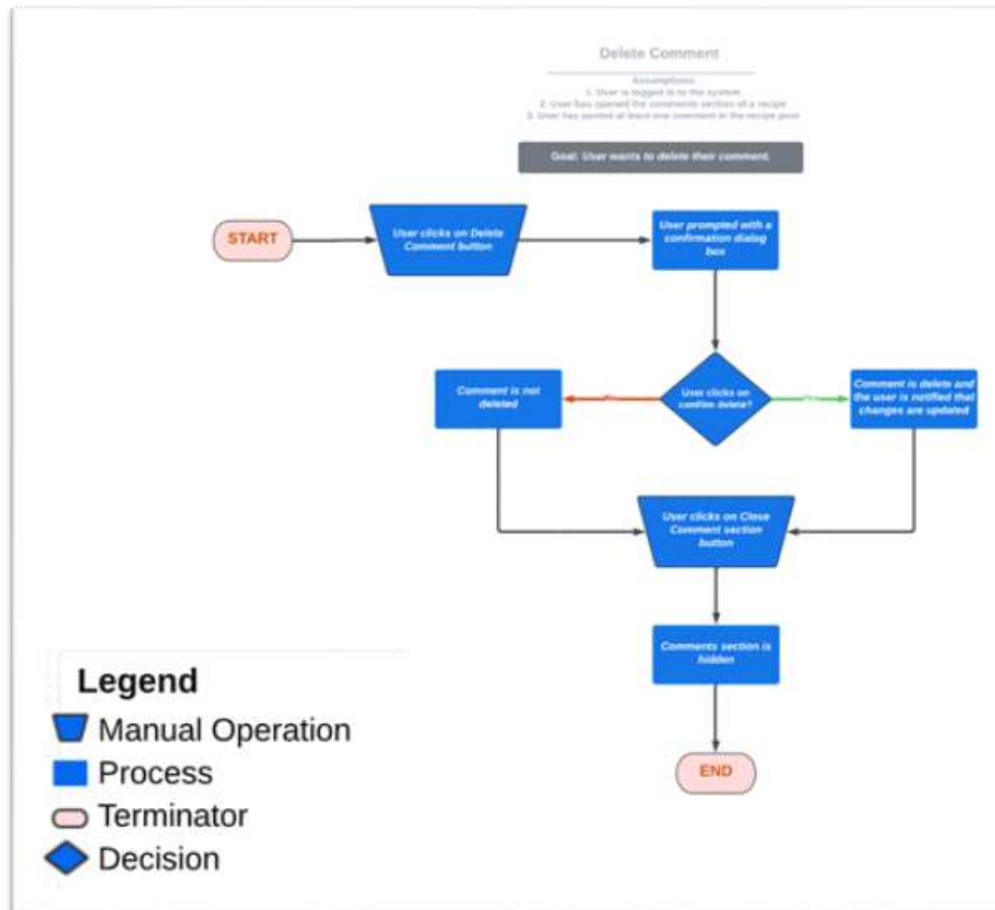


Figure 93: Task flow – Delete comment [5]

- **TASK 5 – Like recipe**

**SCENARIO:** Lucie tried a few recipes, some of which she really liked. To express her feelings, to promote the author, she decides to give the recipe a quick LIKE as it takes a very small time and effort to LIKE a post.

**USE CASE:**

**Assumption 1:** User is logged in.

**Assumption 2:** User is on the FEED page.

**Assumption 3:** There is at least 1 recipe in the FEED page.

1. User scrolls through the recipe(s).
2. User clicks on LIKE button/icon.
3. Like button/icon color is modified indicating that the post is liked.
4. Like counter is incremented.

**TASK FLOW:**

The user can hit the like button on a recipe post to encourage the author.

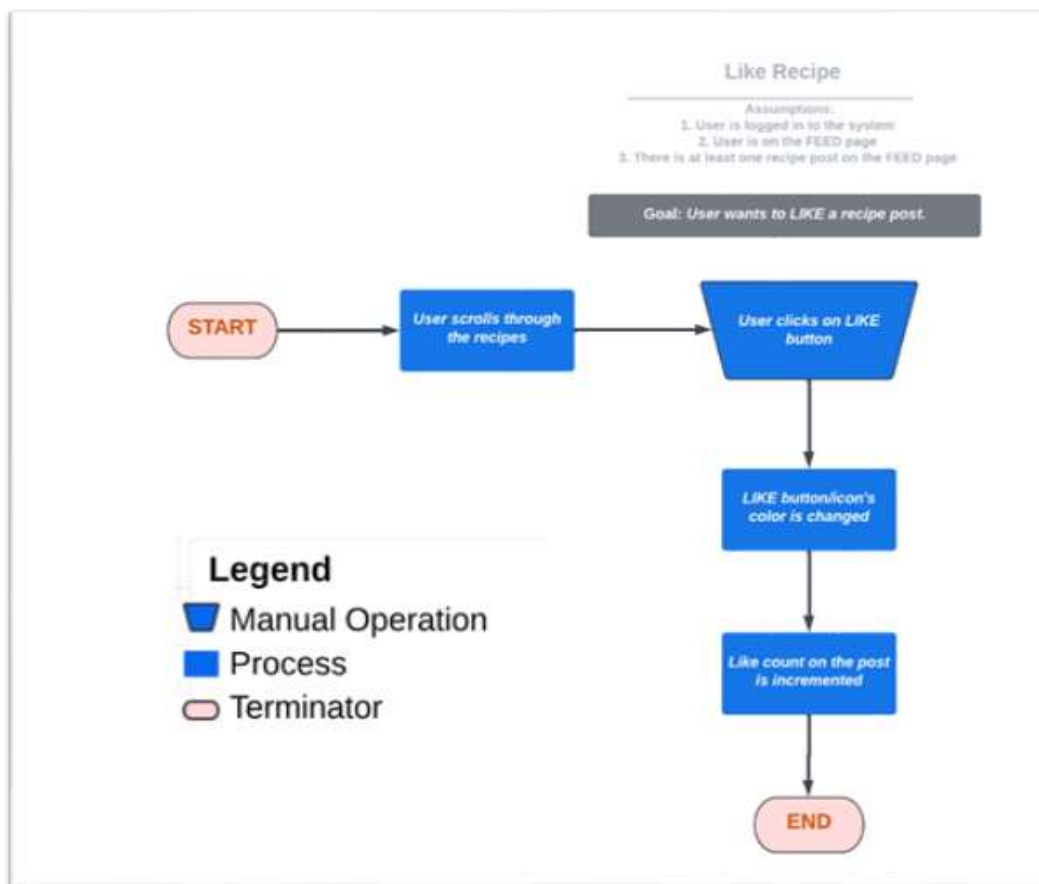


Figure 94: Task flow – Like recipe [5]

- **TASK 6 – UNLIKE RECIPE**

**SCENARIO:** While browsing through the FEED of recipes, Lucie accidentally LIKED a post. To make things right, she would like to undo the LIKE action. She decides to UNLIKE the post to make keep the usage of the application fair for all.

**USE CASE:**

**Assumption 1:** User is logged in.

**Assumption 2:** User is on the FEED page.

**Assumption 3:** There is at least 1 recipe in the FEED page with LIKED post.

1. User scrolls through the liked recipe(s).
2. User clicks on LIKE button/icon.
3. Like button/icon color is modified indicating that the post is unliked.
4. Like counter is decremented.

**TASK FLOW:**

If a user wants to unlike a recipe post, they can do so by hitting the like button again.

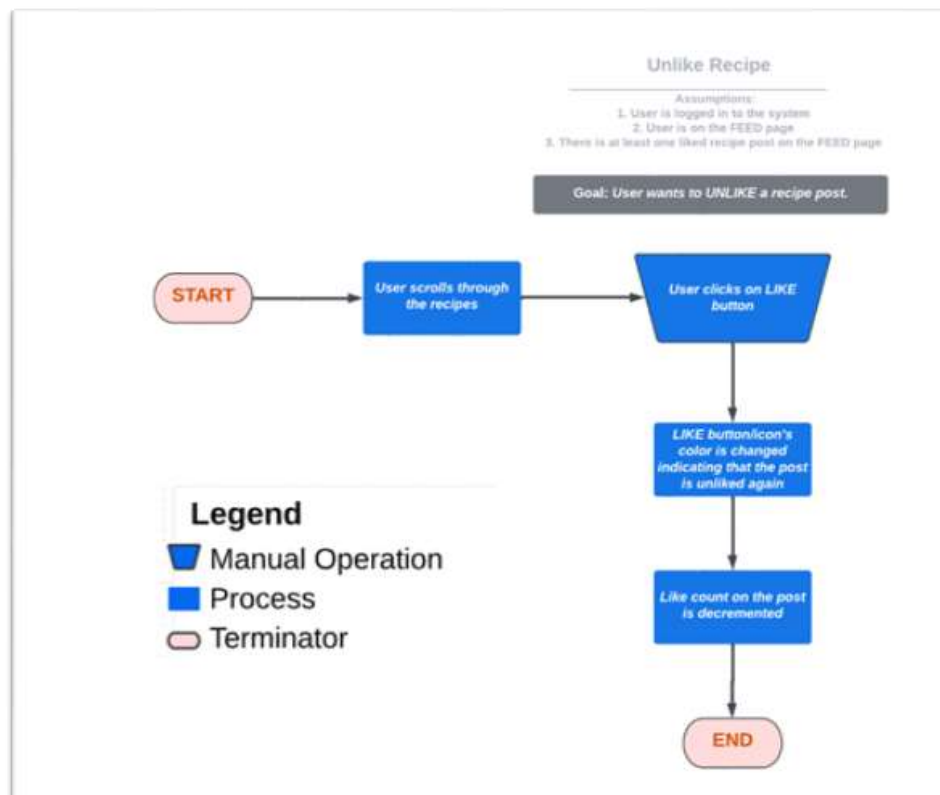


Figure 95: Task flow – Unlike recipe [5]

## FEATURE – Shopping List

- **TASK 1 – CREATING SHOPPING LIST**

**SCENARIO:** Lucie, a graduate student is using our Recipe Sharing application where she liked a recipe that she wants to try but she saw that there are some ingredients in the recipe are not available in her kitchen. So, she wants to create a shopping list to make sure she doesn't forget the ingredients while shopping. She wants a facility in the application that can show her all shopping lists with one click.

**PERSONA:** Graduate Student.

**NEED:** Creating the shopping list for the missing items of recipes that the user needs to purchase.

**CONTEXT:** All the shopping lists get accumulated in one place.

**USE CASES:** Create a shopping list.

**Assumptions:** The user is logged into the application.

1. User is on the Application Homepage.
2. User clicks on the “Shopping List” from the Navigation Bar.
  - 2.1. Shopping List Screen will be displayed with the Message “Oops, you don't have any shopping list” and “GET STARTED” button.
3. User clicks on the “GET STARTED” button.
4. A dialogue box will appear asking for the Shopping list name.
5. User enters the shopping list name.
6. User clicks on the Save button.
  - 6.1. List name entered is not empty.
    - 6.1.1. A new shopping list will get created and displayed on the left panel with a blank list on the right panel.
    - 6.1.2. User is provided with “+ ADD NEW LIST” on the left panel to add more shopping lists.
  - 6.2. List name entered is empty.
    - 6.2.1. User will get Error Message stating “Please Enter the Shopping List Name.”
    - 6.2.2. User enters the Shopping List name.
    - 6.2.3. User clicks on the save button.
    - 6.2.4. A new shopping list will get created.
    - 6.2.5. User is provided with “+ ADD NEW LIST” on the left panel to add more shopping lists.
7. User clicks on the Cancel button to cancel the list.

### **TASK FLOW:**

**Figure 96** represents the Task Flow diagram of **“Create Shopping List”** Use Case where the user is displayed with the message “Oops, you don't have any shopping list” and “GET STARTED” button if he hasn't created any list till now. The user clicks on the **“GET STARTED”** button and is asked to enter the Shopping List name. Upon entering the shopping list name and clicking Save, it will create a new Shopping list with the provided name. Now he will be displayed with **“+ ADD NEW LIST”** to create more lists.

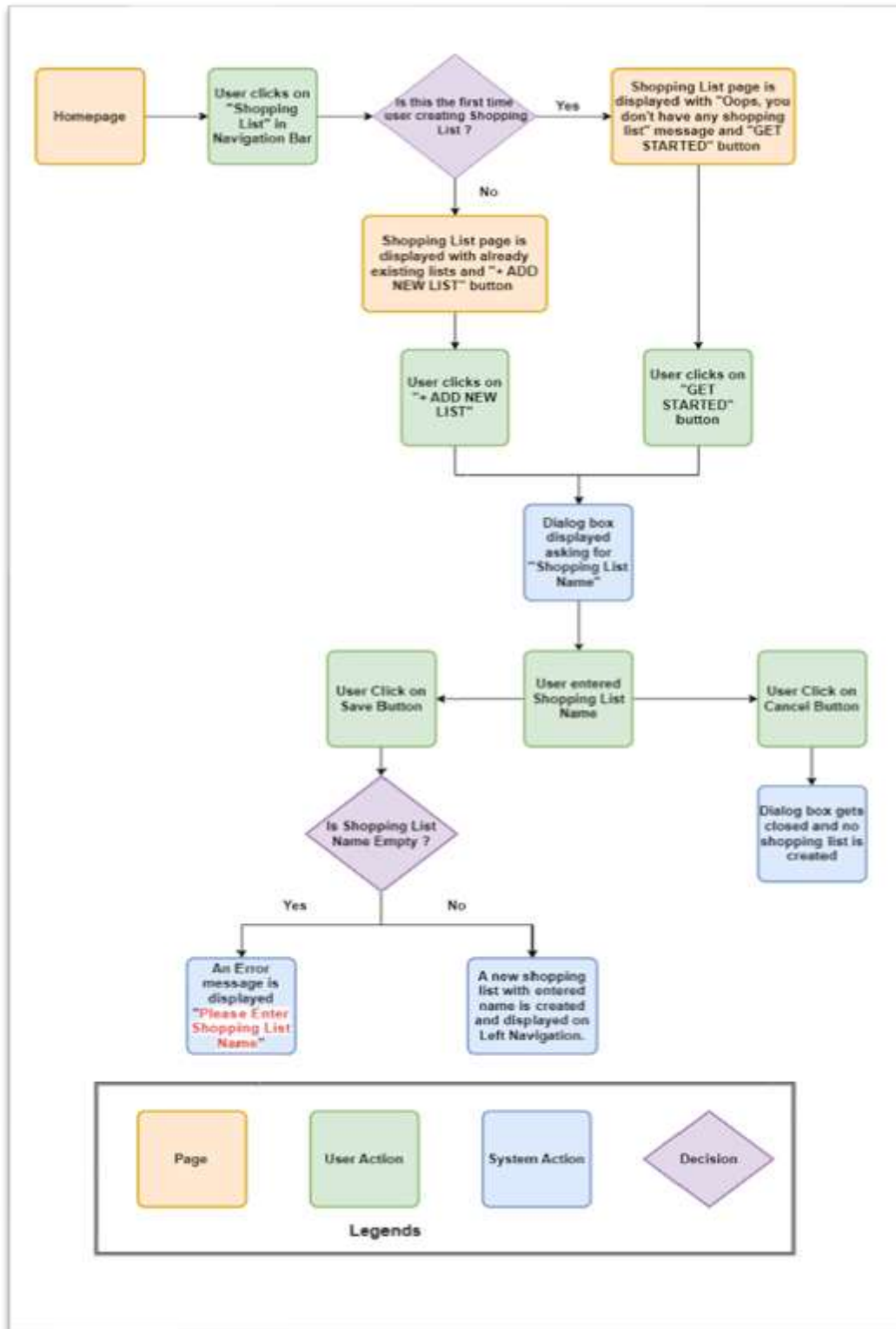


Figure 96: Create Shopping List Task Flow Diagram [3]

- **TASK 2 - ADD ITEMS TO THE SHOPPING LIST**

**SCENARIO:** Lucie, a graduate student is using our Recipe Sharing application where she liked a recipe that she wants to try but she saw that there are some ingredients in the recipe are not available in her kitchen. She has already created a shopping list and she wants to add items to the list to make sure she doesn't forget the ingredients while shopping. She wants a facility in the application where she can add different items/ingredients required for the recipe to the shopping list.

**PERSONA:** Graduate Student.

**NEED:** Add the missing items in the shopping list that the user needs to purchase.

**CONTEXT:** All the missing items will get accumulated in a list in one place.

**USE CASE:** Add items to the shopping list.

**Assumptions:** The user has already logged in to the application and created a shopping list for the recipe.

1. User is on the Shopping List page by clicking on "Shopping List" from Navigation Bar.
2. User clicks on the shopping list that he has created from the left Navigation panel.
3. User is displayed with an empty shopping list with a dropdown to add items to the shopping list.
4. User clicked on the dropdown, and he will be displayed with some popular items with a checkbox to select.
5. User clicks on the checkbox to add the item to the shopping list.
  - 5.1. Item gets added to the Shopping list under "Selected Item(s)".
  - 5.2. User can add more items to his Shopping list.
6. All the items added by the user are displayed in created Shopping List.

**TASK FLOW:**

**Figure 97** represents Task Flow Diagram of "Adding Items to Shopping List" Use Case where user is displayed with popular ingredients used in recipe and he/she can user can add them in their shopping list so that they can retrieved them in one go.



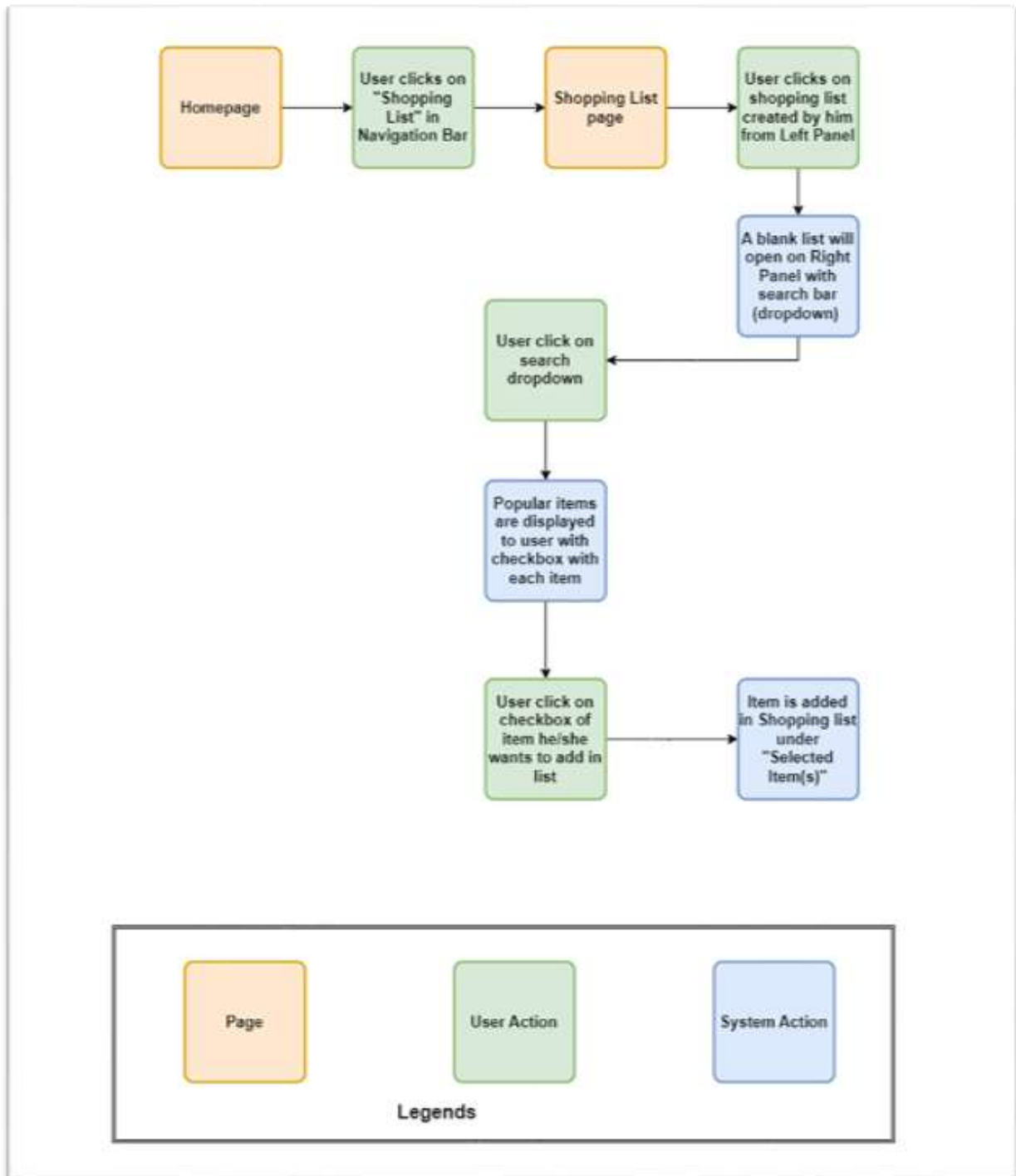


Figure 97: Add Item in Shopping List Task Flow Diagram [3]

- **TASK 3 - REMOVE ITEMS FROM THE SHOPPING LIST**

**SCENARIO:** Lucie, a graduate student is using our Recipe Sharing application where she liked a recipe that she wants to try but she saw that there are some ingredients in the recipe are not available in her kitchen. She has already created a shopping list and added items to the list that she wants to purchase. She wants a facility in the application where she can remove different items/ingredients from her shopping list once she has purchased it or doesn't want it anymore.

**PERSONA:** Graduate Student.

**NEED:** Remove the items from the shopping list once purchased or don't want anymore.

**CONTEXT:** Items no longer required or purchased get removed from the shopping list.

**USE CASE:** Remove items from the shopping list.

**Assumptions:** The user has already logged in to the application, created a shopping list for the recipe and added items that need to be purchased.

1. User is on the Shopping List page by clicking on "Shopping list" on Navigation Bar.
2. User clicks on the shopping list that he has created from the left panel.
3. User is displayed with a shopping list along with the items added to that shopping list.
4. For each of added item in the shopping list, there will be a checkbox associated with that item.
5. User has purchased an item or is no longer required for that item.
6. User selects the checkbox of that item.
  - 6.1. Item gets crossed and added under "Checked Item(s)" in the shopping list.
  - 6.2. User has selected the checkbox of that item by mistake.
    - 6.2.1 User deselects the checkbox of that item.
    - 6.2.2 Item gets added back to "Selected Item(s)" in the shopping list.

**TASK FLOW:**

**Figure 98** represents Task Flow Diagram of "**Removing Items from Shopping List**" Use Case where user can remove the ingredients from Shopping list once they purchased them or no longer needed by checking the checkbox displayed with them. If they have checked the checkbox by mistake, they can revert it back by simply unchecking the check box.

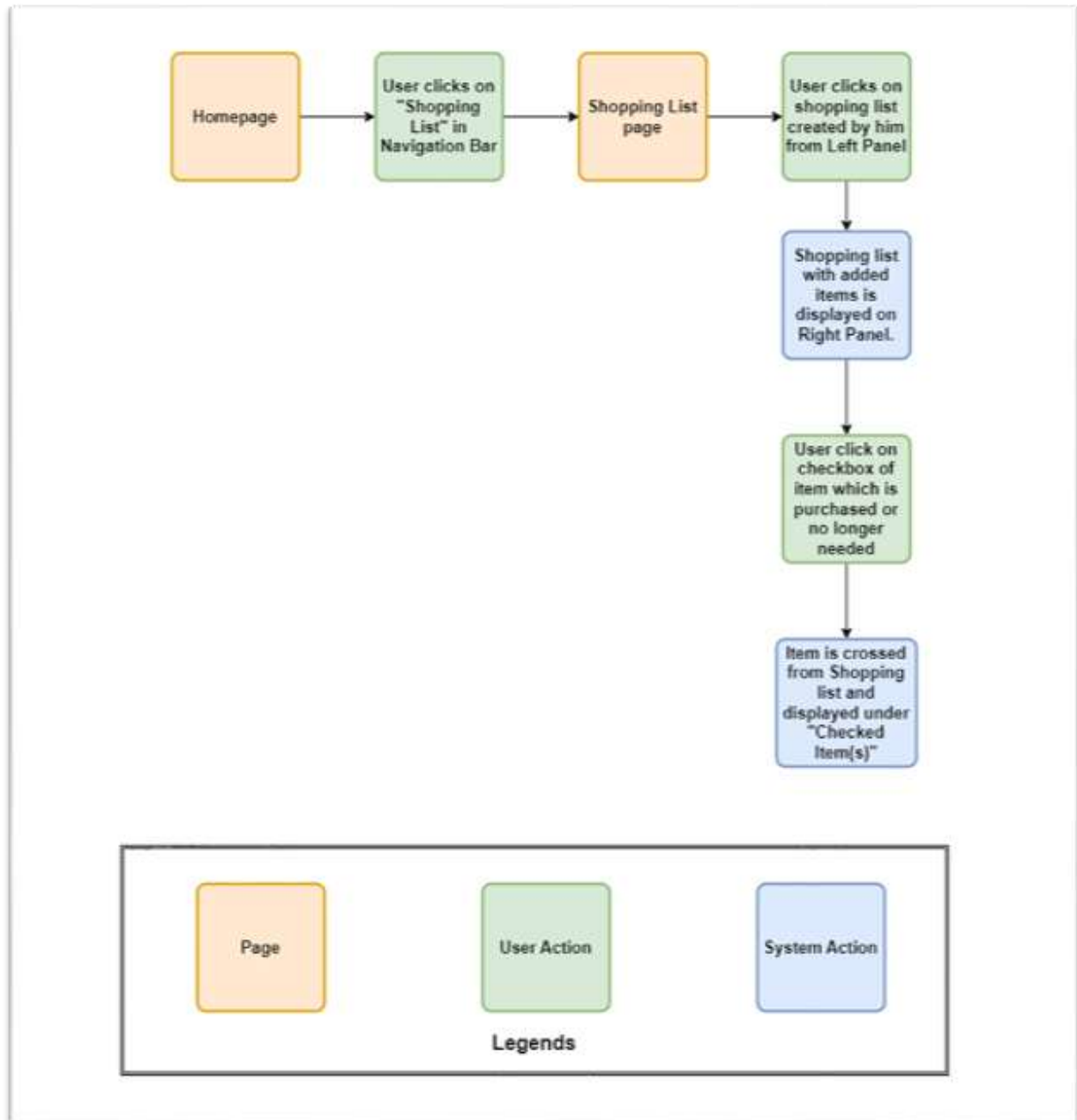


Figure 98 : Remove Item from Shopping List Task Flow Diagram [3]

- **TASK 4 - DELETE THE SHOPPING LIST**

**SCENARIO:** Lucie, a graduate student is using our Recipe Sharing application where she liked a recipe that she wants to try but she saw that there are some ingredients in the recipe are not available in her kitchen. She has already created a shopping list and added items to the list that she wants to purchase. She has purchased all the items and doesn't want to remove all the items one by one. She wants a facility in the application where she can remove the entire shopping list in one click.

**PERSONA:** Graduate Student.

**NEED:** Remove the complete shopping list once purchased all items or no longer required.

**CONTEXT:** Once the recipe is done, delete the entire shopping list instead of removing items one by one.

**USE CASE:** Delete the Shopping List.

**Assumptions:** The user has already logged in to the application, created a shopping list for the recipe and added items that need to be purchased.

1. User is on the Shopping List page by clicking on "Shopping List" from Navigation Bar.
2. User clicks on the shopping list that he has created from the left navigation.
3. User is displayed with a shopping list along with the items added to that shopping list.
4. User has purchased all the items or is no longer required for that item. So, the user wants to remove the entire Shopping list instead of removing items one by one.
5. User clicks on the trash icon present beside the name of the shopping list.
6. The list gets removed from the left panel of the shopping page.

**TASK FLOW:**

**Figure 99** represents Task Flow Diagram of "**Removing Shopping List**" Use Case where user can delete the entire Shopping List once they have purchased all ingredients from Shopping List or no longer needed. Instead of removing the ingredients one by one, they have the option of removing entire shopping list.

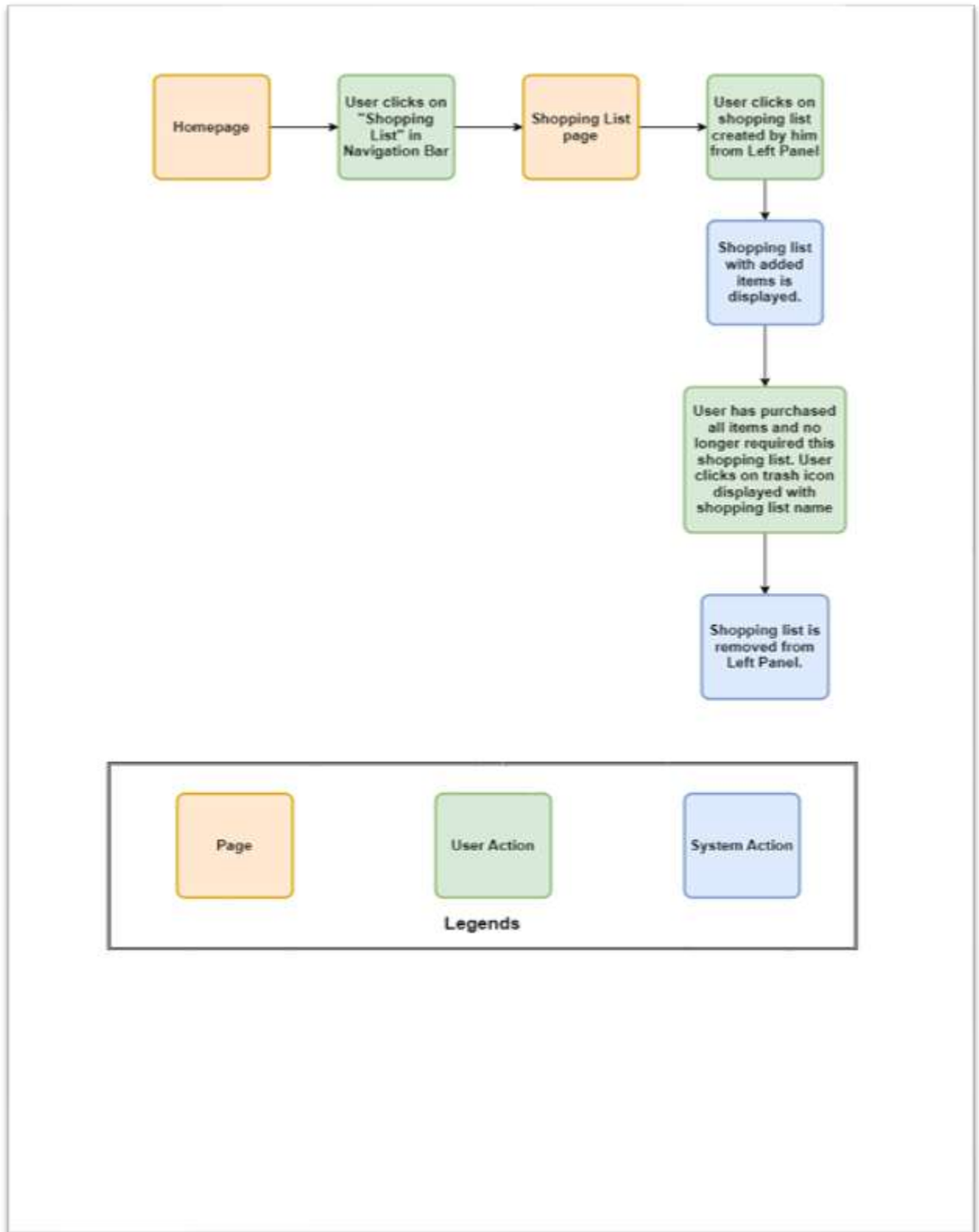


Figure 99: Taskflow for removing shopping list [3]

## **FEATURE – Meal Quantifier & Calorie Tracker**

- **TASK 1 - MEAL QUANTIFIER - DISPLAY THE QUANTITY OF INGREDIENTS REQUIRED BASED ON THE NUMBER OF SERVES**

**SCENARIO:** Lucie, a graduate student is using our Recipe Sharing application where she liked a recipe that she wants to try but she saw that the recipe is according to 1 serving but she wants to prepare it for her 5 friends. She wants a facility in the application where she can get the exact quantity of ingredients needed to make the recipe for 5 serves.

**PERSONA:** Graduate Student.

**NEED:** Getting the exact quantity of ingredients as per the servings of the meal.

**CONTEXT:** The quantity of ingredients is calculated as per the servings of the meal.

**USE CASES:** Meal Quantifier - Display the Quantity of Ingredients required based on the number of serves.

**Assumptions:** The user is logged into the application.

1. User is on the Feed page and liked a recipe.
2. User opened the recipe to get the complete details.
3. User is displayed with ingredients required for 1 serving of the meal.
4. User is provided with a Meal Quantifier button to increase/decrease the serving by clicking the +/- button.
5. User changes the serving.
  - 5.1. User clicks on + icon.
    - 5.1.1. Quantity of ingredients is increased as per servings selected.
  - 5.2. User clicks on the – icon.
    - 5.2.1 Quantity of ingredients is decreased as per servings selected.

### **TASK FLOW:**

**Figure 100** represents the Task Flow Diagram of the “Meal Quantifier” Use Case where the user can get the quantity of ingredients required for the meal as per servings by clicking on the +/- button.

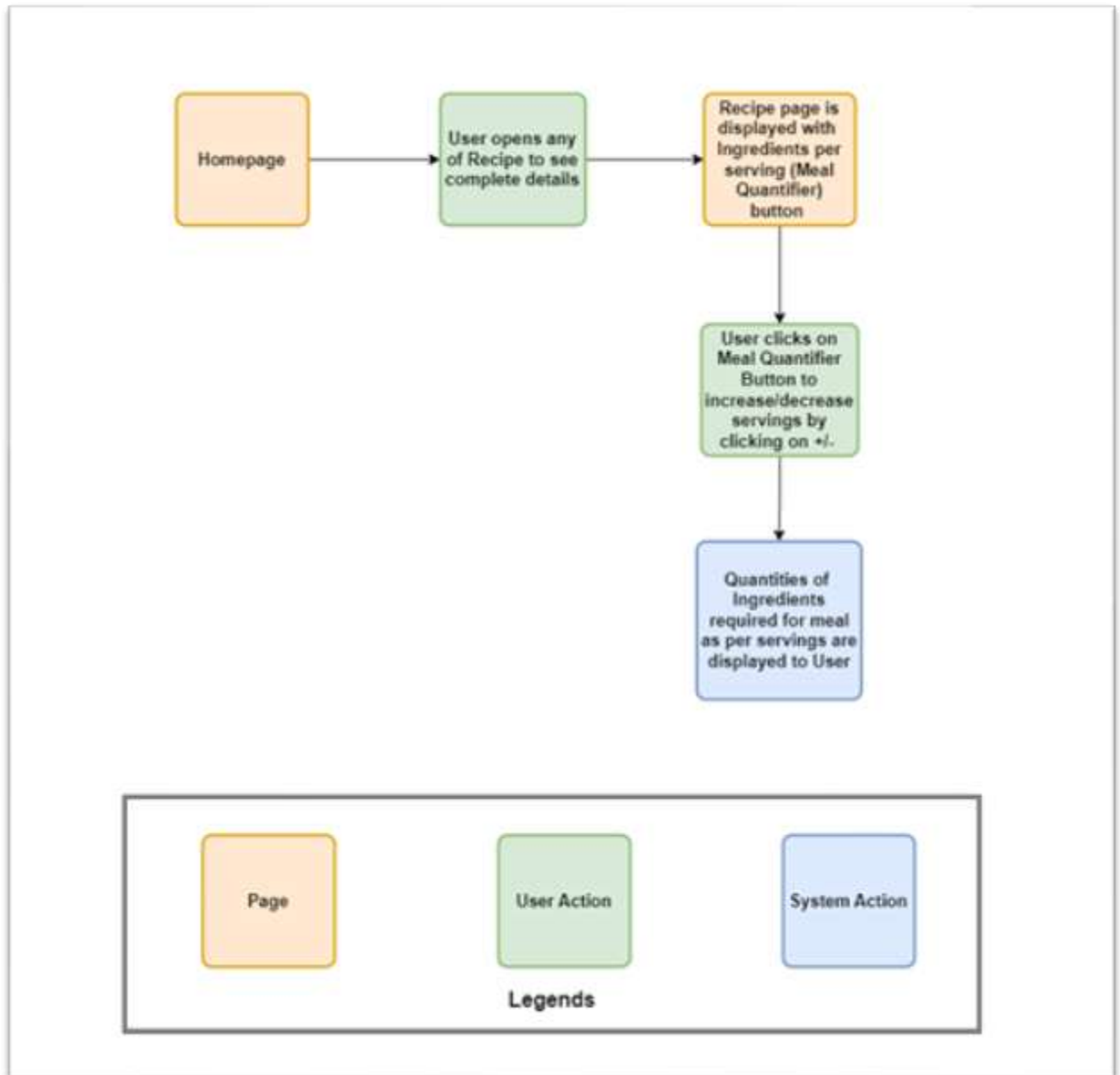


Figure 100: Taskflow for meal quantifier

- **TASK 2 - Calorie Tracker**

**SCENARIO:** Lucie, a graduate student is using our Recipe Sharing application where she liked a recipe that she wants to try but she is concerned about his health. She wants a facility in the application where she can get the calorie count in meals and check if it's good for her health or not.

**PERSONA:** Graduate Student.

**NEED:** Getting the calorie count in the meal and checking if it's good for health or not.

**CONTEXT:** The calorie in the meal is displayed along with the health score.

**USE CASE:** Calorie Tracker - Calorie indicator with respect to 1 serve.

**Assumptions:** The user has already logged in to the application.

1. User is on the Feed page and liked a recipe.
2. User opened the recipe to get the complete details.
3. User is displayed with calories in the recipe as per 1 serving.
4. User is also displayed with a Health score stating whether the recipe is good for health or not.

**TASK FLOW:**

**Figure 101** represents the Task Flow Diagram of the “**Calorie Tracker**” Use Case where the user can get the calorie count in the meal as per 1 serving along with the Health score.





Figure 101: Taskflow for calorie tracker [3]

## **FEATURE – Feed search and filter**

- **TASK 1 – Search for a recipe**

**SCENARIO** – Lucie has a dish in mind. She already has some information about the dish, like the name of the dish, how the dish looks like. She is ready to prepare the dish now but needs the application to find the recipe. She is on the landing page of the application and sees the Search bar for a quick lookup of the recipe.

**USE CASE:**

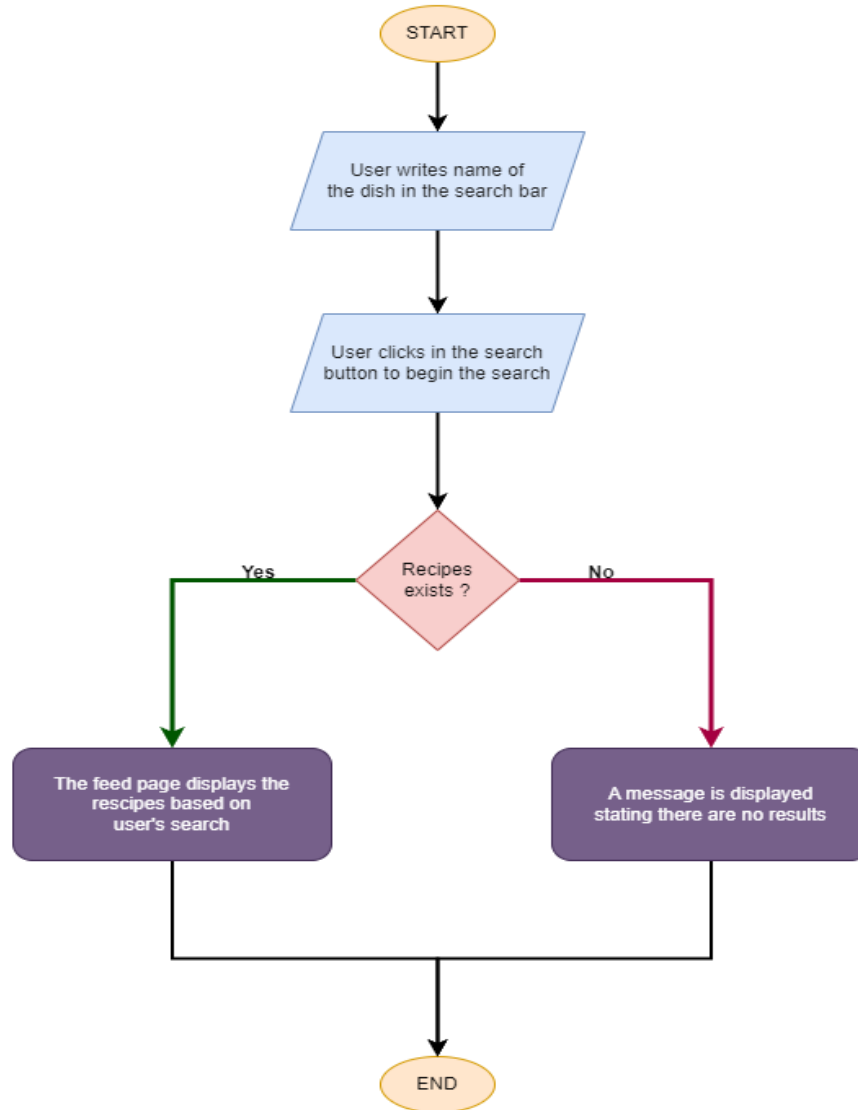
**Assumptions:** Lucie is already logged in to the application and is on the landing page.

1. User writes the name of the dish in the search bar and hits the search button.
  - 1.1. Recipe of the dish exists. A list of recipes matching the searched dish appears as a list.
  - 1.2. Recipe does not exist. A message stating no results found will be displayed the user.

**TASK FLOW:**

The user can use the search bar to search for any recipe if they already know what dish they are looking for. The search will return the matching recipes. If no match is found, a message will be displayed that no recipe match found.

### Task - Search Recipe



Legend:



Figure 102: Task flow – Search recipe [3]

- **TASK 2 – Filter recipes**

**SCENARIO** – Lucie is picky about the category/ingredients of a recipe and would like to filter the results/recipes on the landing page for an optimized and custom view.

**USE CASE:**

**Assumptions:** Lucie is already logged in to the application and is on the landing page with some recipes already available on the page.

1. User clicks on the filter option at the top.
2. Filter dialog/panel opens with multiple filter options to choose from.
3. User selects the desired options from the available filters.
  - 3.1 User clicks on Apply filter button.
    - 3.1.1 Search results on the feed page are modified and are filtered.
  - 3.2 User clicks on Cancel button.
    - 3.2.1 Filter is not applied and the recipes on the page remain as is.
4. Filter dialog/panel hides.

**TASK FLOW:**

Users can filter the search results by various categories, including the type of dish, cuisine or if they have any dietary restrictions or preferences. They can play with the filters and then click on the Apply Filter button to apply the filter. The filtered result can be removed by clicking on the clear filter button.

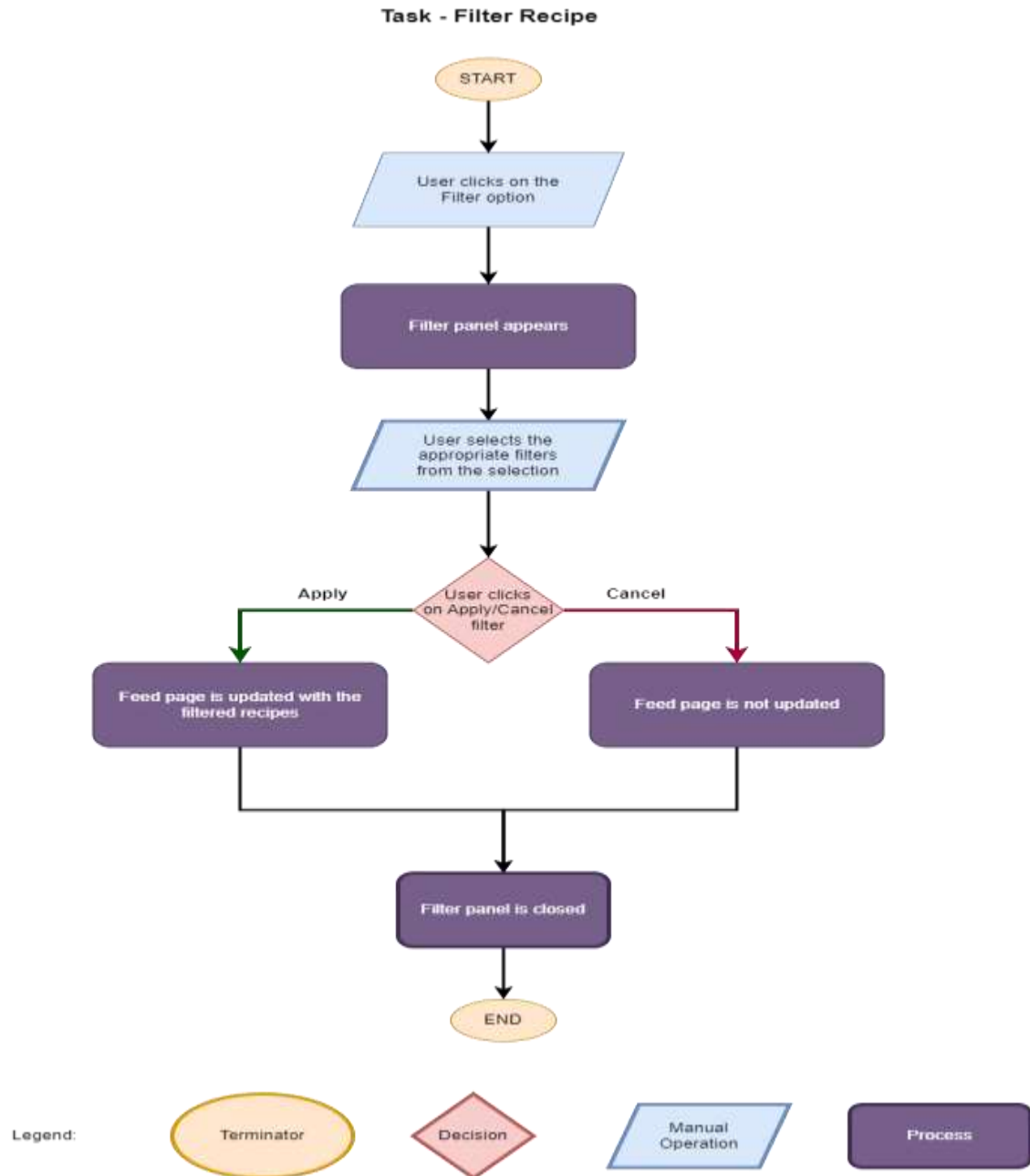


Figure 103: Task flow – Filter recipe [3]

## FEATURE - In-App Notification

- **TASK 1 - LIKE AND COMMENT NOTIFICATION**

**SCENARIO:** When lucie posts her recipe on the CookWithDal app, she receives likes and comments on her posts. She can engage with those notifications through “Notification” feature present in the navbar as shown in fig. .

**PERSONA:** Lucie (A graduate student)

**NEED:** User needs to click on “Notification” button to see the “Likes” and “Comments” on the particular posts.

**CONTEXT:** User wants to find out how many comments or likes he/she received on the post which gives the insights about the engagement of the post. And user can feel more encouraged to be consistent in posting recipe contents.

**USE CASES:** Receive “Likes” and “Comments” notifications.

**Assumption:** User is already logged in the system.

1. User clicks on the “Notification” option present in the navbar.
2. On Clicking, the user can see the list of people who have commented or liked the post.

### TASKFLOW:

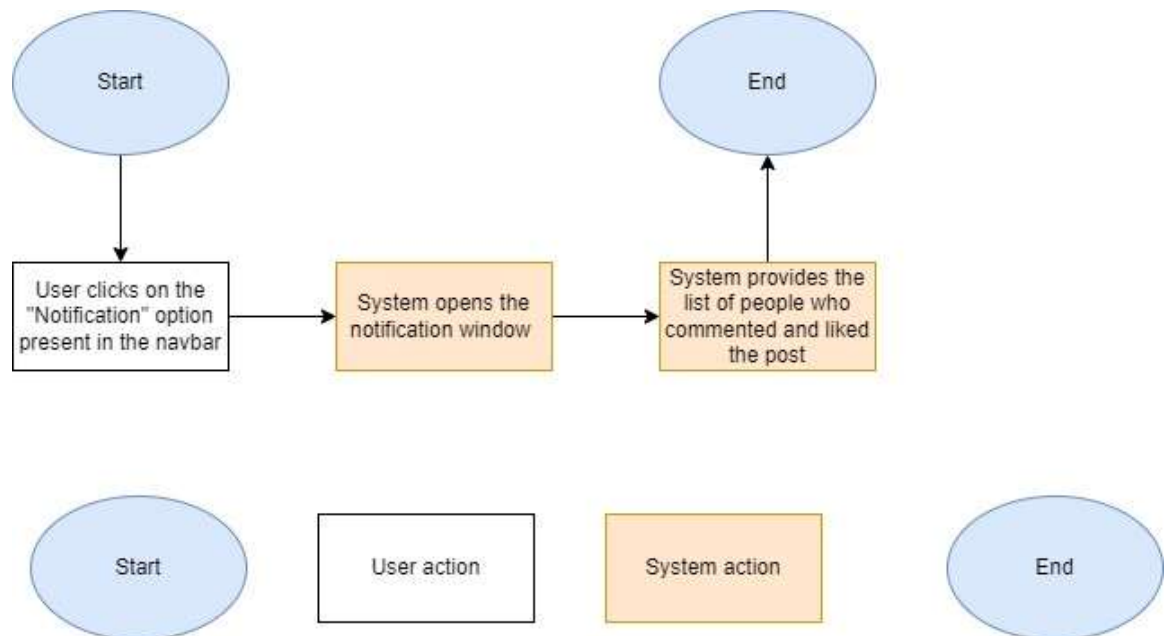


Figure 104: Taskflow for like and comment notification [3]

## FEATURE - Meal Planner

- **TASK 1 - Add the recipe to plan**

**SCENARIO 1:** Lucie, a graduate student, and the app user saw an amazing quick-to-eat recipe in the application, and she wants to store it in the app and add it to next week's schedule for Saturday meal. After saving the recipe under favorites, she goes to the meal planner and sees an option "add recipes from favorites" and then gets prompted to schedule the recipe for a specific date from the dropdown. The application prompts the "recipe added to the schedule" and she can view the schedule by choosing the weekly calendar grid.

**PERSONA:** Lucie (a graduate student)

**NEED:** Add a recipe to a day in a weekly schedule

**CONTEXT:** For adding a recipe directly to a day in a week so that she can plan for a recipe she wants to try beforehand.

**ASSUMPTION:** Lucie is already a registered user.

**USE CASE:** Save the recipe to the planner

1. The user opens the app and goes to the feed section of the application.
2. Feed displays various recipes in it, the user sees a particular recipe.
3. To view the recipe completely, the user clicks on the recipe tile, which loads the recipe as a new page.
4. The user likes a recipe and saves it using the bookmark button.
5. The user wants to make the recipe on a specific date, so the user goes to the "planner" option from the navigation bar.
6. The user sees a button "from saved recipes", which pops up a window with saved favorite recipes.
7. Users select a specific recipe that they wish to schedule for a specific day and see a "choose date" button from the selected recipe icon.
8. User selects the specific date from the "choose date" drop-down button.
  - 8.1 The recipe gets added to the specific date and the user gets the prompt "recipe got added to the planner". The user could also see the recipe added under the specific calendar grid for that date.
  - 8.2 The recipe couldn't get added for that specific date and the user receives a prompt that "recipe couldn't be added, please try again later".
9. User returns to the feed or adds another recipe to the planner.

## TASK FLOW:

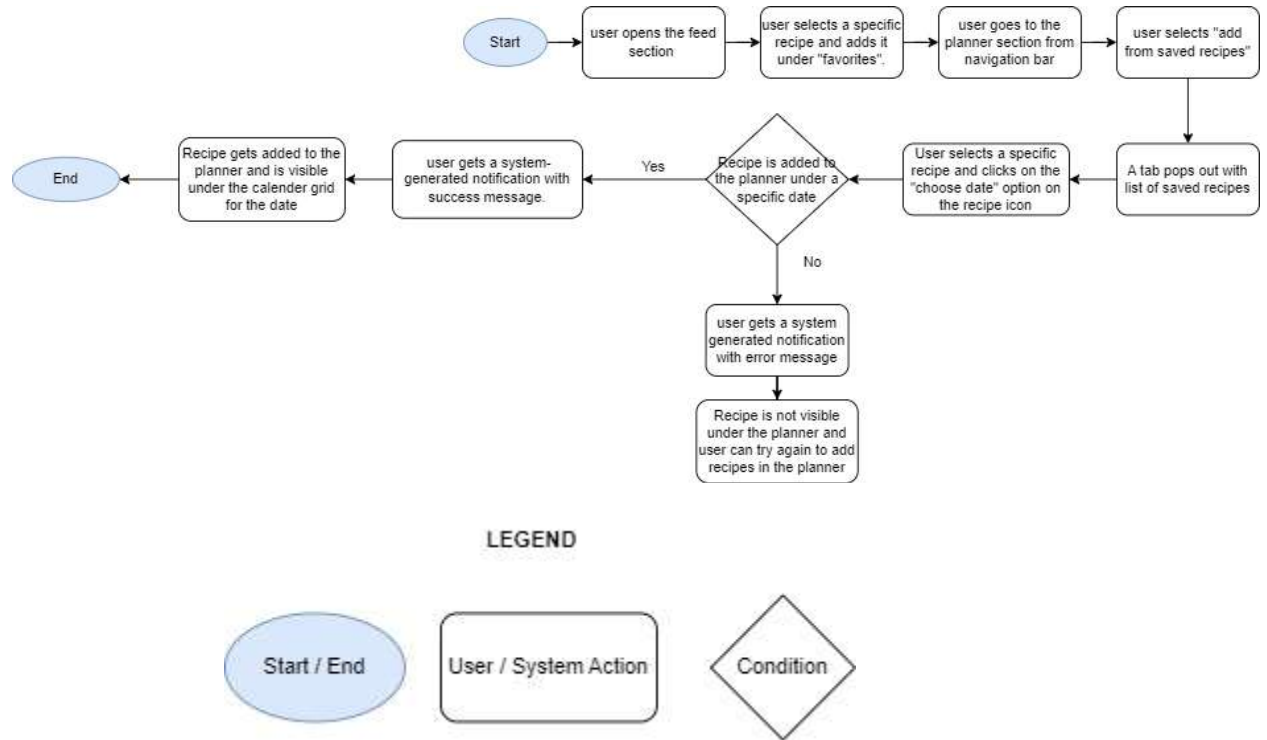


Figure 105: Taskflow for adding recipe to planner [3]

**SCENARIO 2:** Lucie, a graduate student, and the app user has multiple recipes that she has saved under favorites in the application. She wishes to add it to next week's schedule for the Saturday meal. she goes to the meal planner and sees an option "add recipes from favorites" and then gets prompted to schedule the recipe for a specific date from the dropdown. The application prompts the "recipe added to the schedule" and she can view the schedule by choosing the weekly calendar grid.

**PERSONA:** Lucie (a graduate student)

**NEED:** Add a recipe to a day in a weekly schedule

**CONTEXT:** For adding a recipe directly to a day in a week so that she can plan for a recipe she wants to try beforehand.

**ASSUMPTION:** Lucie is already a registered user.

**USE CASE:** Save the recipe to the planner

1. The user wants to make the recipe on a specific date, so the user goes to the "planner" option from the navigation bar.
2. The user sees a button "from saved recipes", which pops up a window with saved favorite recipes.



3. Users select a specific recipe that they wish to schedule for a specific day and see a “choose date” button from the selected recipe icon.
4. User selects the specific date from the “choose date” drop-down button.
  - 4.1 The recipe gets added to the specific date and the user gets the prompt “recipe got added to the planner”. The user could also see the recipe added under the specific calendar grid for that date.
  - 4.2 The recipe couldn’t get added for that specific date and the user receives a prompt that “recipe couldn’t be added, please try again later”.
5. User returns to the feed or adds another recipe to the planner.

#### TASK FLOW:

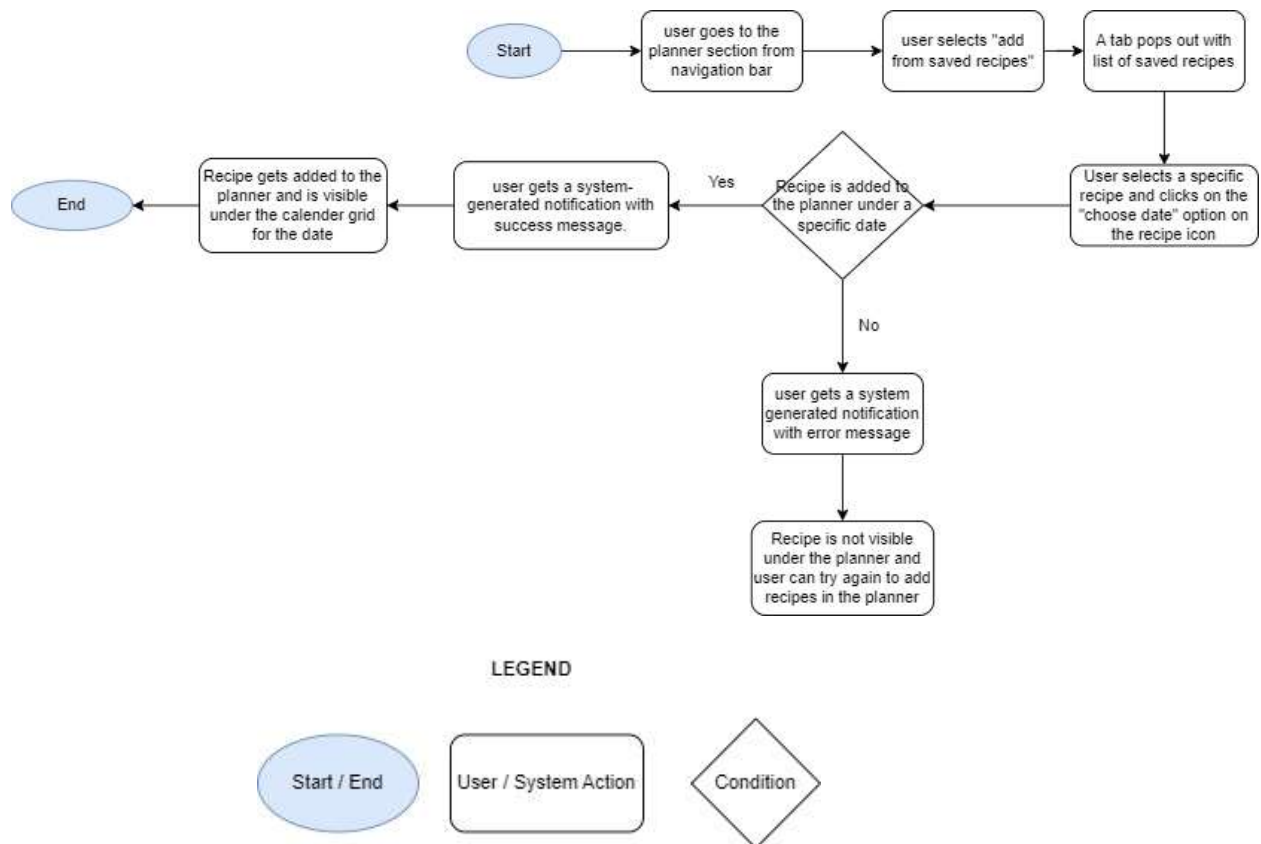


Figure 106: Taskflow for saving recipe to planner [3]

- **TASK 2 - Delete the recipe from the plan**

**SCENARIO:** Lucie, a graduate student, and the app user saw an amazing quick-to-eat recipe in the application, and she added a saved “favorite” recipe to date in the meal planner. She feels won’t be able to make that recipe on that specific day and wants to remove it from the meal planner. She notices a delete “cross” button

on the recipe image displayed under the calendar grid. She clicks on that button to remove the recipe from the planner.

**PERSONA:** Lucie (a graduate student)

**NEED:** Remove a recipe from the planner

**CONTEXT:** Removing the recipe that is not needed anymore from the planner.

**ASSUMPTION:** Lucie is already a registered user.

**USE CASE:** Remove a recipe for a specific date from the planner.

1. The user visits the planner section of the application, and wishes to remove a recipe from it for a specific date.
2. User opens the calendar grid for the week that the specific date falls under, and clicks on the “cross” button on the recipe image icon displayed in the planner schedule grid.
  - 2.1. The recipe gets deleted from the planner and is no longer visible under the calendar grid.
  - 2.2. The recipe couldn't get deleted from the planner and user gets prompt that “recipe couldn't get deleted, please try again deleting later”.

#### TASK FLOW:

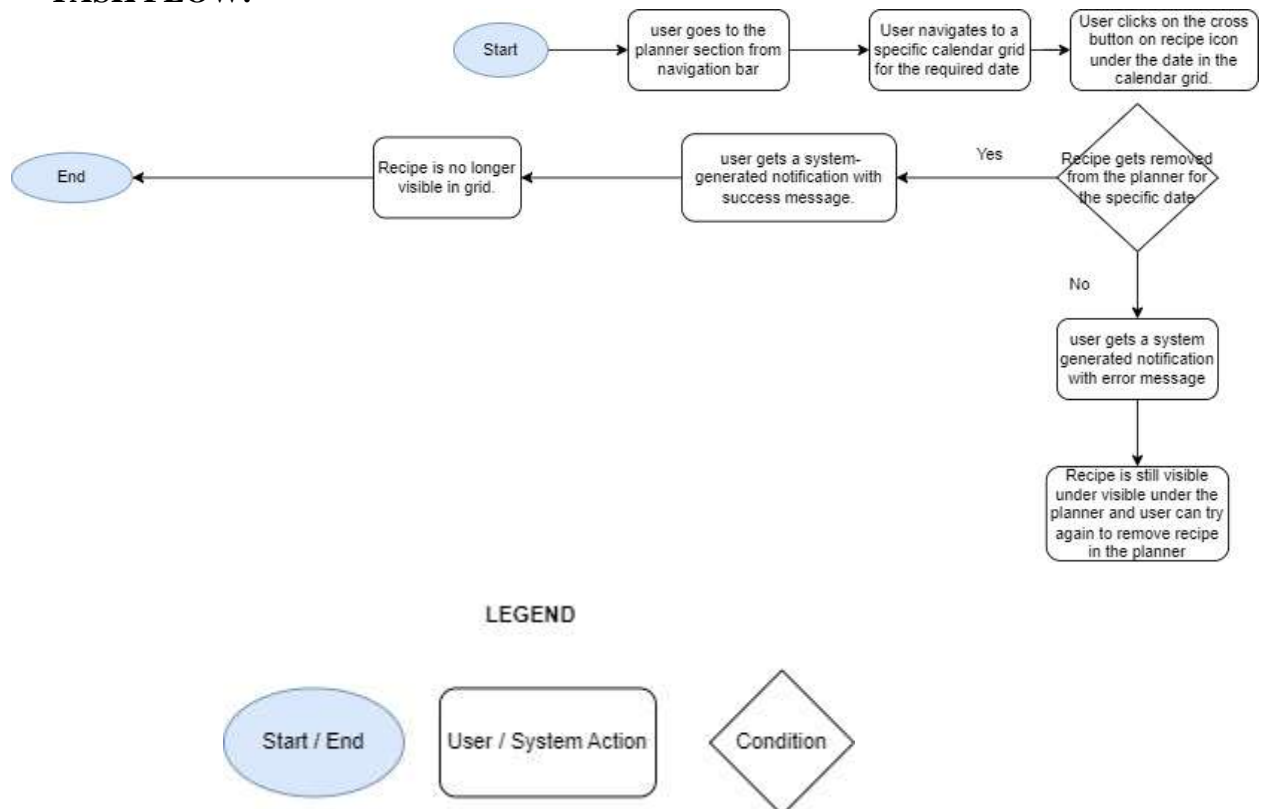


Figure 107: Delete recipe from planner [4]

## **FEATURE - Save favourite recipes**

- **TASK 1 - SAVE RECIPE**

**SCENARIO:** Lucie, a graduate student, and the app user saw an amazing quick-to-eat recipe in the application, and she wants to store it in the app so that she can refer to it when she needs it again. She sees a save button with a bookmark icon on it and clicks on it. The application prompts her that the recipe is saved under her profile section.

**PERSONA:** Lucie (a graduate student)

**NEED:** Save a new recipe so that it can be referred to when needed

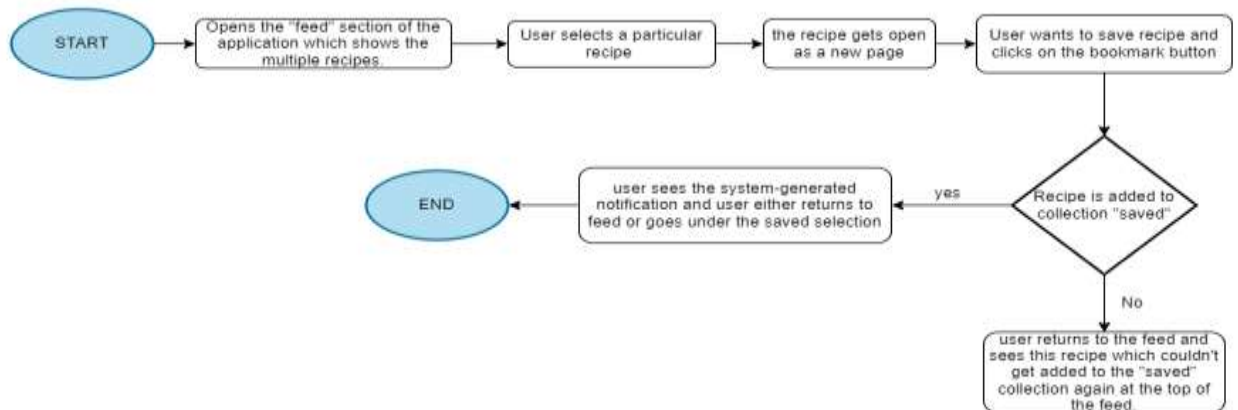
**CONTEXT:** For saving favorite recipes so that they can be accessed easily when needed again.

**Assumption:** Lucie is already a registered user.

**USE CASE:** Save the recipe to profile.

1. User opens the app and goes to the feed section of the application.
2. Feed displays various recipes in it, the user sees a particular recipe.
3. To view the recipe completely, the user clicks on the recipe tile, which loads the recipe as a new page.
4. The user likes the recipe and wants to save it and sees a bookmark icon on the page.
5. When the user hovers over the button, the button gives a hint that it's a bookmark button, and the user clicks the button.
  - 5.1 The recipe gets added to the saved list under the user profile section and the user gets a notification that "the recipe got added to the saved collection under the profile section".
  - 5.2 The recipe doesn't get added to the saved list under the user profile and the user gets a notification that "recipe couldn't get added to the saved collection, please try again later" and the application added it under the recipe list suggestion to be shown in the feed. When the user returns to the feed, the user sees this recipe that couldn't get added last time as at the top of the feed now.
6. The user visits the profile section and sees a subsection "saved" which has all the saved recipes under it or returns to the feed.

## TASK FLOW:



### LEGEND

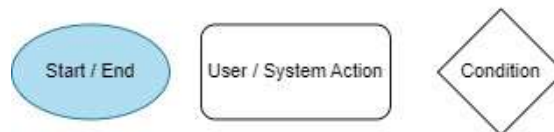


Figure 108: Saved recipe to profile [4]

- **TASK 2 - REMOVE SAVED RECIPE**

**SCENARIO:** Abby, a graduate student, and the app user saw an amazing quick-to-eat recipe in the application, and she wants to store it in the app so that she can refer to it when she needs it again. She sees a save button with a bookmark icon on it and clicks on it. While browsing her saved recipes she notices that she doesn't want a specific recipe in her saved section as she doesn't need it, she notices that if she unselects the bookmark button, the recipe gets removed from the saved recipe section. She clicks on the save button to unselect it and the recipe gets removed.

**PERSONA:** Lucie (a graduate student)

**NEED:** Remove a saved recipe from the collection as it is not needed anymore

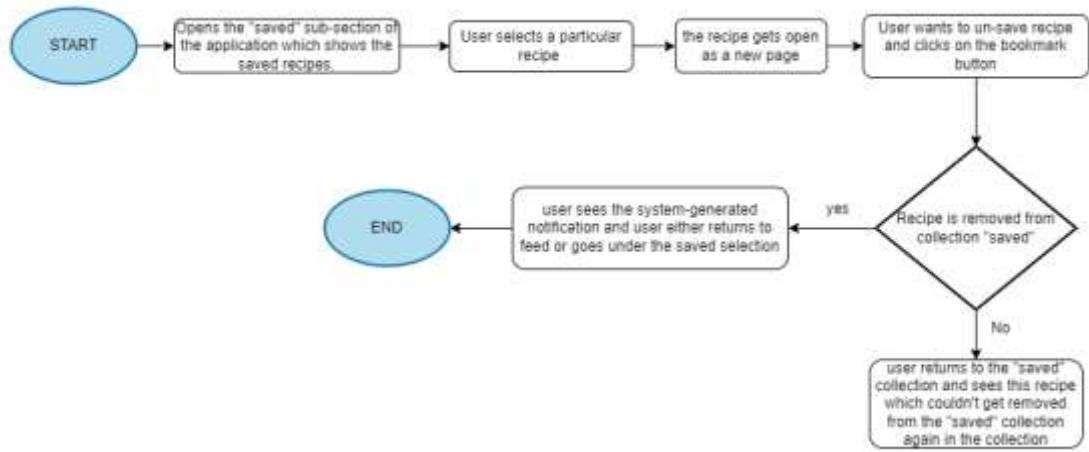
**CONTEXT:** Removing the recipe that is not needed anymore from the saved collection.

**Assumption:** Lucie is already a registered user.

**USE CASE:** Remove a saved recipe from subsection "saved" of the user profile

1. User opens the profile section and goes to the saved sub-section of the profile.
2. The user browses the recipes under the "saved" subsection and sees a recipe that is not required anymore in the saved sub-section.
3. The user clicks the recipe and hovers again to the bookmark icon button, this time when the user hovers over it, it shows "choose this button to un-save the recipe".
4. The user clicks the bookmark icon button.
  - 4.1 The recipe gets removed from the saved collection and the user gets a notification that the "recipe was removed from the application".
  - 4.2 The recipe couldn't get removed from the saved collection and the user gets a notification that the "recipe was not deleted, please try again to delete later".

## TASK FLOW:



### LEGEND

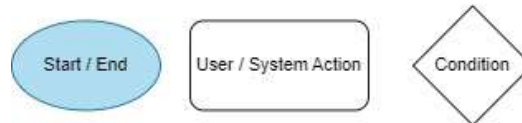


Figure 109: Remove recipe from saved [3]

### 3. Asset Inventory

#### Client–Side Technologies:

1. HTML5 – It is used to create the structure of the website pages.
2. CSS – It is used to provide styling to the elements on the web pages.
3. React – Front and framework for performing state management on the website components.
4. Bootstrap 5 – CSS framework that provides standard styling for various web components.
5. React UI Material - React components that implement Google's Material Design.
6. React 17 UI kit – Standard web components provided with the React Library.

#### Server–Side Technologies:

1. NodeJS – Cross-platform open-source JavaScript runtime environment
2. ExpressJS – Backend framework for creating REST APIs in NodeJS
3. MongoDB - NoSQL document database that scales horizontally.

#### Images:

Below herewith is the list of image types that we used in our website:

1. Jpg
2. Jpeg
3. png

### 4. Group Roles

Table 1: Group roles

Team Members	Primary Role	Secondary Role
ANUJ DAWAR	FULL – STACK DEVELOPER	DATABASE MANAGER AND TESTING
FAIZA UMATIYA	FULL – STACK DEVELOPER	DOCUMENT MANAGER AND TESTING
PARUL RAICH	FULL – STACK DEVELOPER	SCRUM MASTER AND TESTING
RUCHIKA .	FULL – STACK DEVELOPER	DESIGNER AND TESTING
SAGARKUMAR VAGHASIA	FULL – STACK DEVELOPER	DOCUMENT MANAGER AND TESTING
SAIF ALI PRASLA	FULL – STACK DEVELOPER	INTEGRATOR AND TESTING

## 5. REFERENCES

- [1] "Figma," Figma. [Online]. Available: <https://www.figma.com/files/recent?fuid=1203216882011676942>. [Accessed: 07-Feb-2023].
- [2] "Create powerful business content together with Xtensio," Xtensio.com, 14-Jun-2022. [Online]. Available: <https://xtensio.com/>. [Accessed: 07-Feb-2023].
- [3] "Flowchart maker & online diagram software," Diagrams.net. [Online]. Available: <https://app.diagrams.net/>. [Accessed: 06-Feb-2023].
- [4] "Balsamiq cloud," Balsamiq.cloud. [Online]. Available: <https://balsamiq.cloud/sgvkuqp/pgqrj2b/rE39C>. [Accessed: 07-Feb-2023].
- [5] "Intelligent diagramming," Lucidchart. [Online]. Available: <https://www.lucidchart.com/>. [Accessed: 06-Feb-2023].
- [6] *Epicurious.com*. [Online]. Available: <https://www.epicurious.com/>. [Accessed: 26-Feb-2023].
- [7] *Whisk.com*. [Online]. Available: [https://my.whisk.com/?utm\\_source=website&utm\\_medium=homepage&utm\\_campaign=redirect](https://my.whisk.com/?utm_source=website&utm_medium=homepage&utm_campaign=redirect). [Accessed: 26-Feb-2023].