

# **ASSIGNMENT – 2**

## **APPLICATION DETAILS:**

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]

## **PROPOSED APPLICATION FEATURES:**

- 1) User Management
- 2) Profile Account Management - including preferences on recipe interests
- 3) Recipe Management
- 4) Bookmark/Save (favorites)
- 5) Comments and likes
- 6) Feed, search and filter
- 7) Shopping list for missing items
- 8) In-app Notification
- 9) Meal quantifier and calorie tracker

## **BONUS FEATURES:**

- 1) Follow/Unfollow User
- 2) Meal Planner

## **CHOSEN FEATURE:**

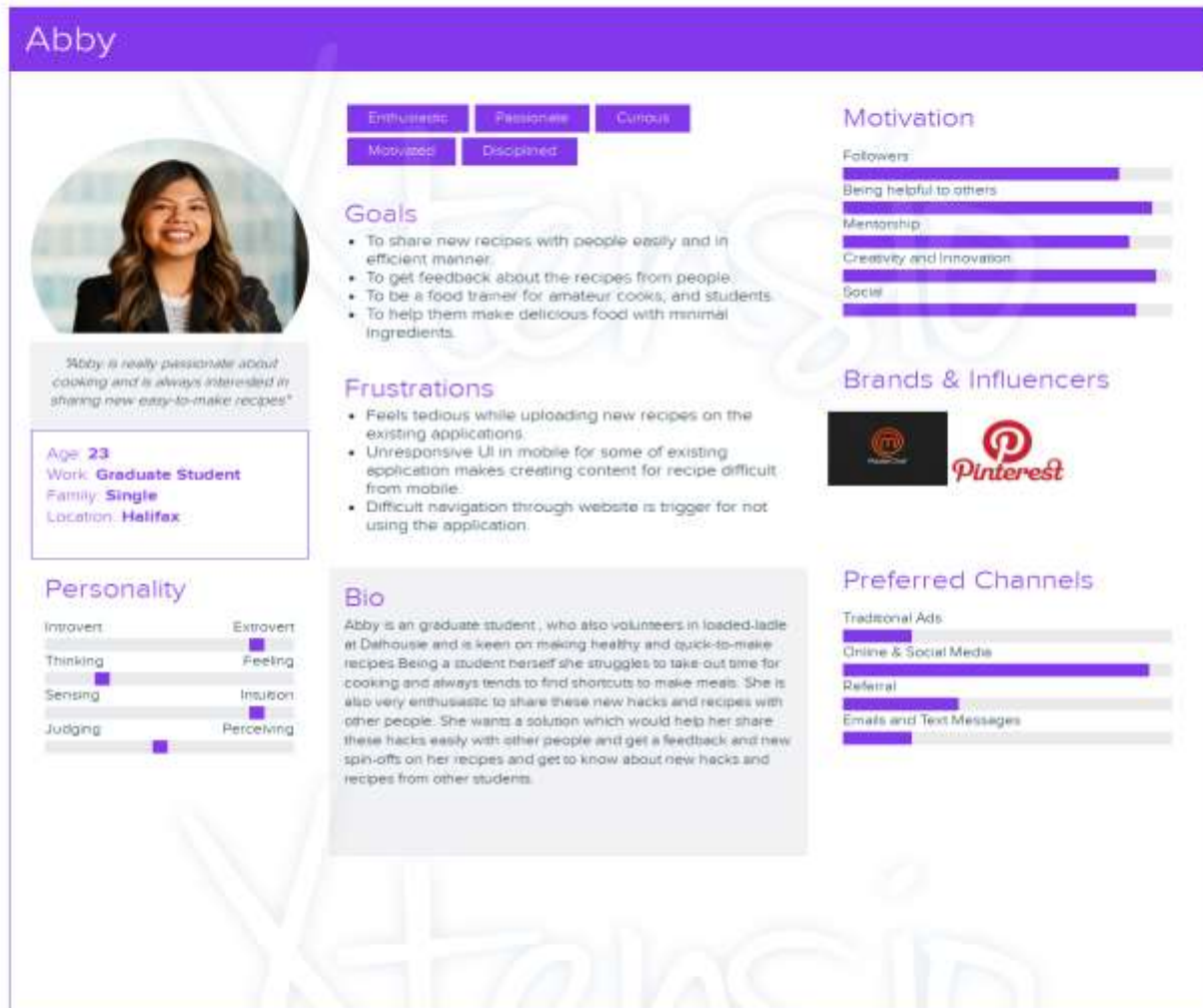
### **Profile Account Management**

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

## **APPLICATION DETAILS:**

### **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. User personas of two different types of users (Abby – who likes to share recipes, and, Lucie – who likes to try out new recipes) is shown in figure 1 and figure 2 [2].



DO NEXT [Customer Journey Map](#)

[See related templates](#)

Figure 1: User Persona for Abby, used xtensio referred from [3]

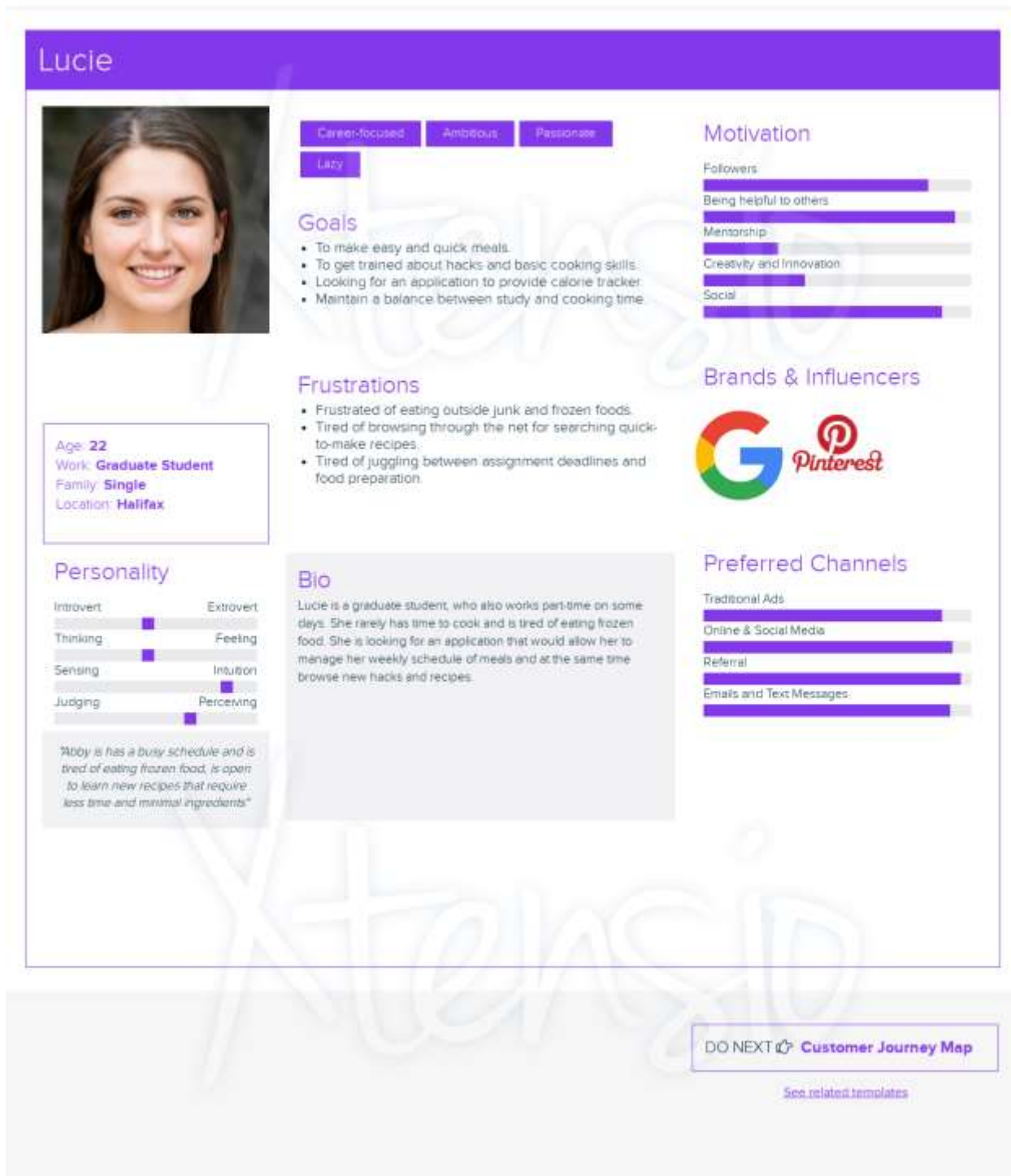


Figure 2: User Persona for Lucie, used Xtensio referred from [3]

## **2. User-centered design approach:**

Based on the Information Architecture which includes proposed sitemap, website design (wireframes), I tried to implement the feature and front-end in the design and layout of the website. Refer figure 3-20 to see the similarity between wireframe and actual website design and layout.

## A. INFORMATION ARCHITECTURE

### 1) Proposed Sitemap:

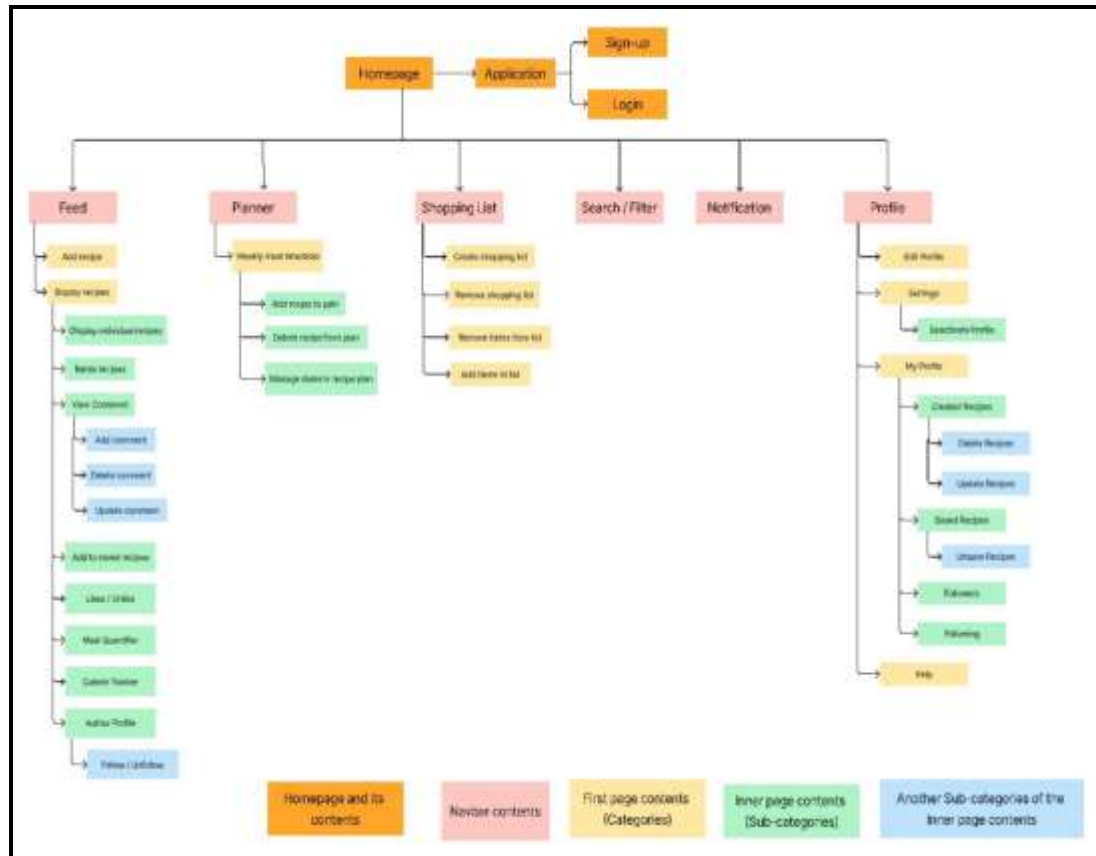


Figure 3: Sitemap [4]

## 2) Website Design:

- **TASK 1 - UPDATE USER PROFILE**

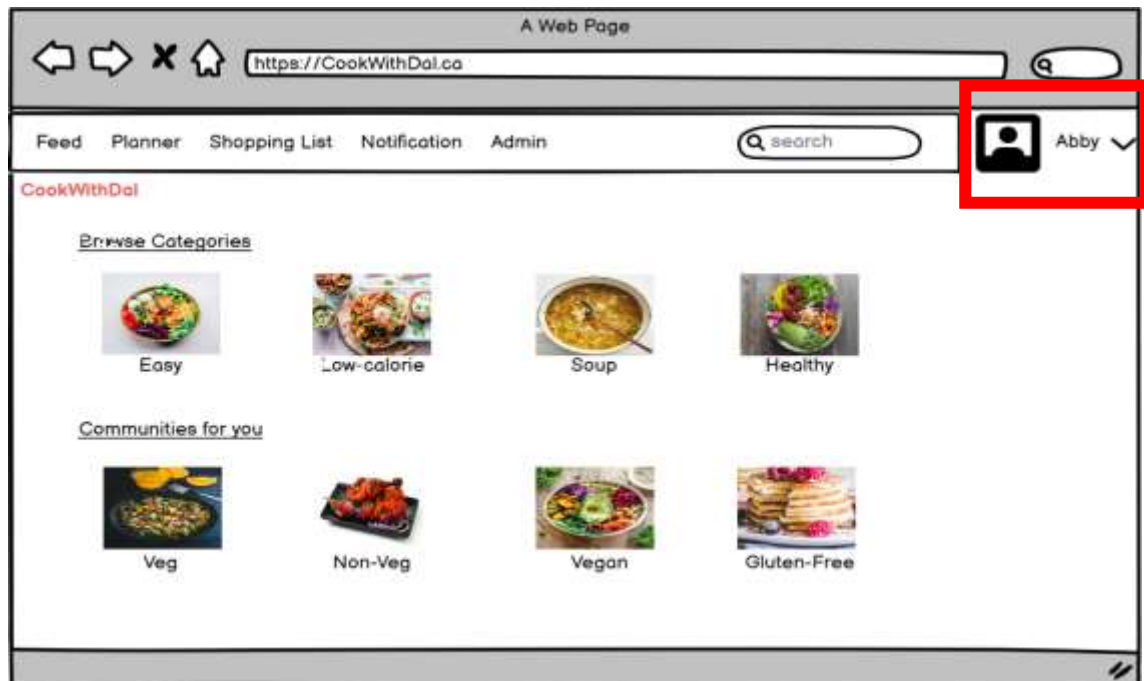


Figure 4: Feed page showing profile icon [5]



Figure 5: Profile icon showing dropdown list [5]

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 6: Update profile form [5]

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: Profile updated successfully!

Save Deactivate Account

Figure 7: Profile updated successfully [5]



- **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 8: Profile form to deactivate [5]

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 9: Alert message for confirming deactivation [5]





Figure 10: Profile deactivated successfully [5]

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



Figure 11: Display profile page [5]

## B. DESIGN AND LAYOUT

- **TASK 1 - UPDATE USER PROFILE**

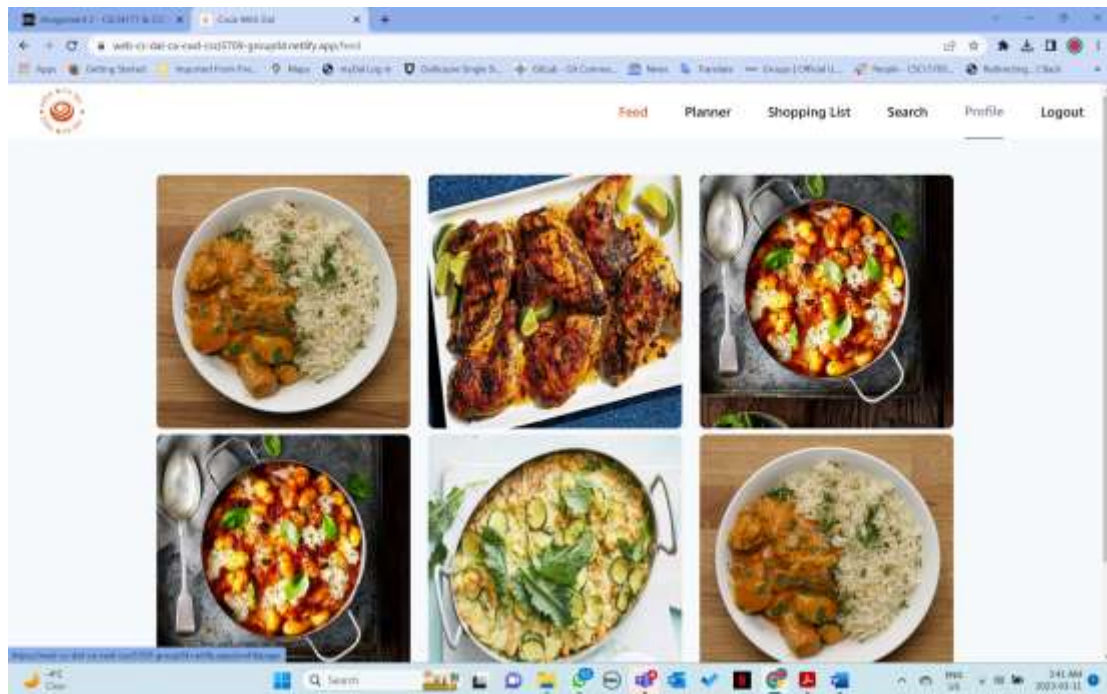


Figure 12: Feed Page of the CookWithDal website [6]

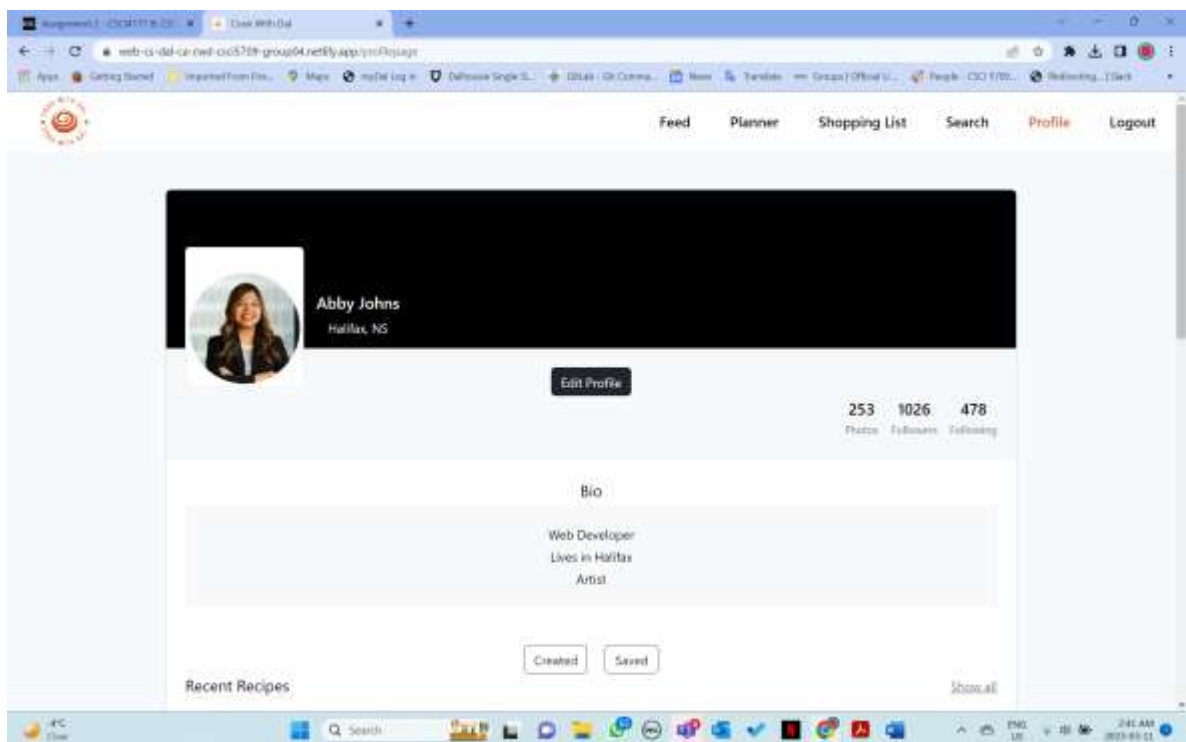


Figure 13: Profile Page [6]

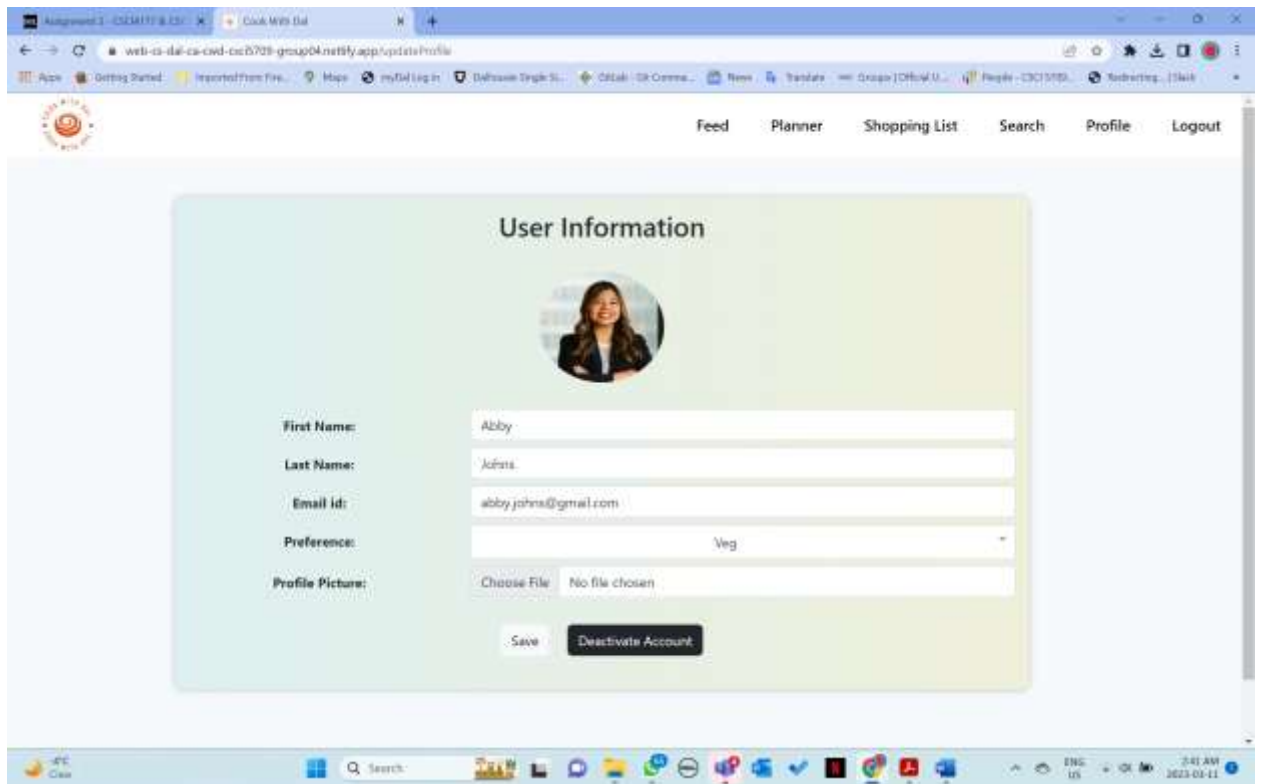


Figure 14: Edit user information page [6]

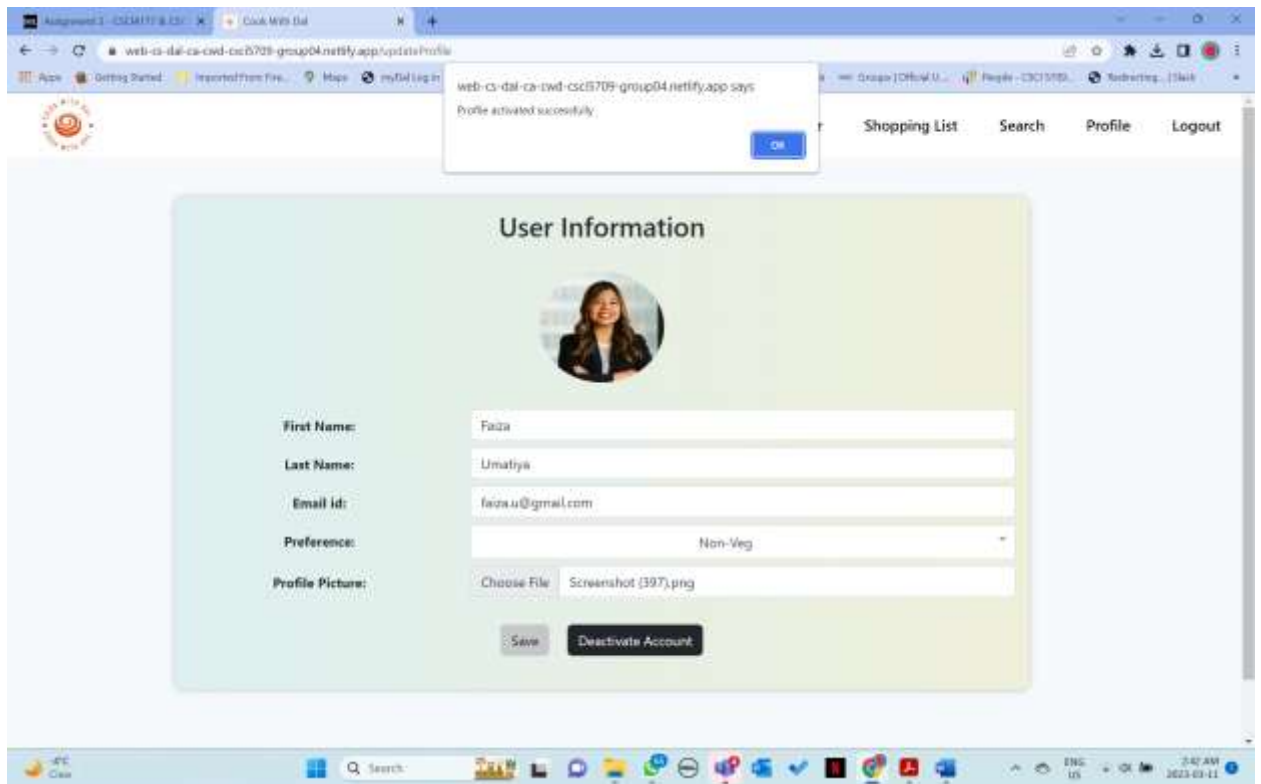


Figure 15: User information updated successfully [6]

- **TASK 2 - DEACTIVATE USER PROFILE**

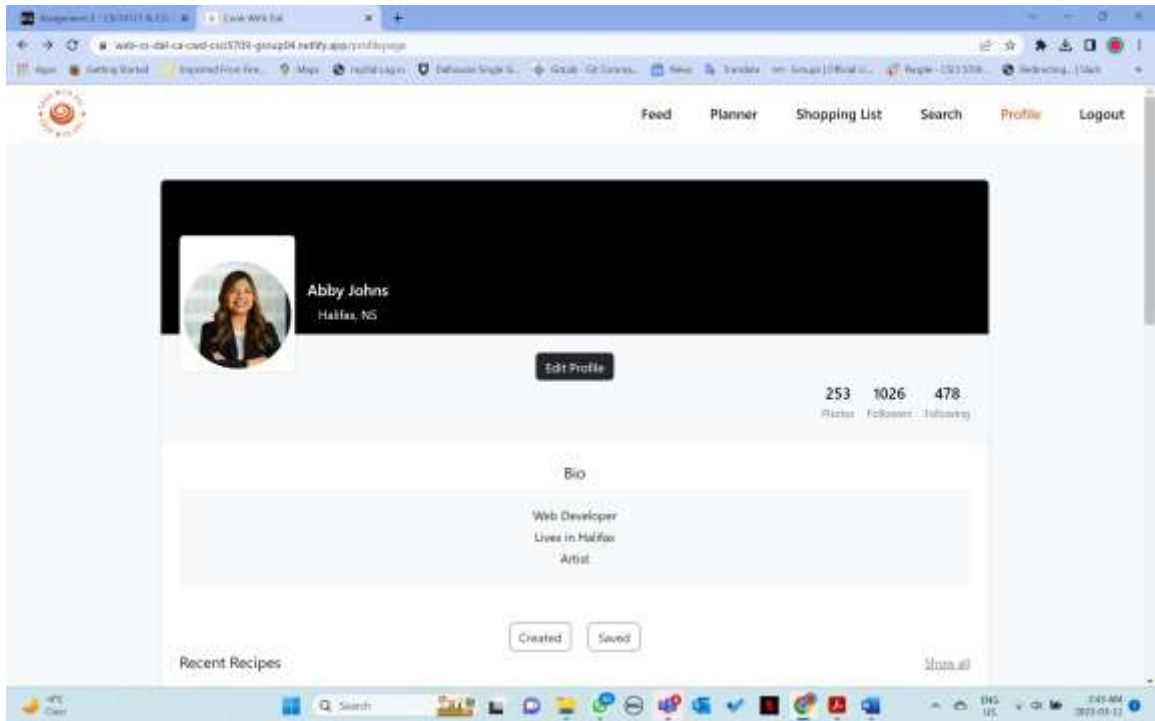


Figure 16: Profile page [6]

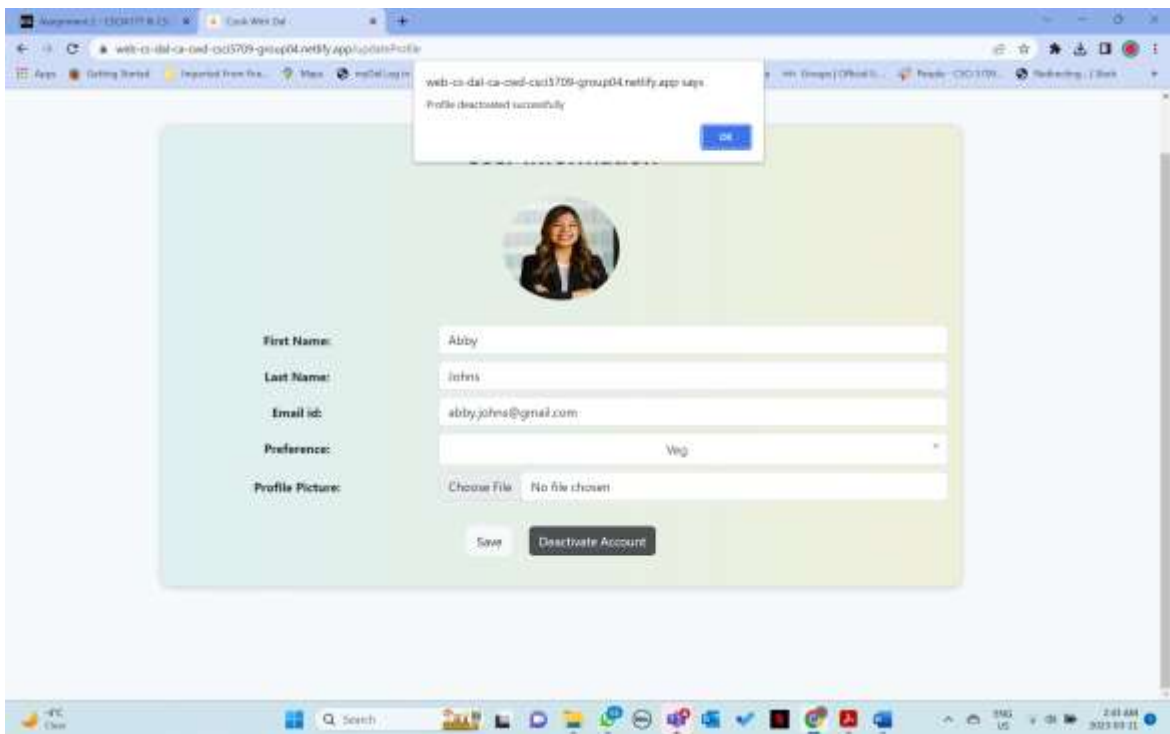


Figure 17: Profile deactivated successfully [6]



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

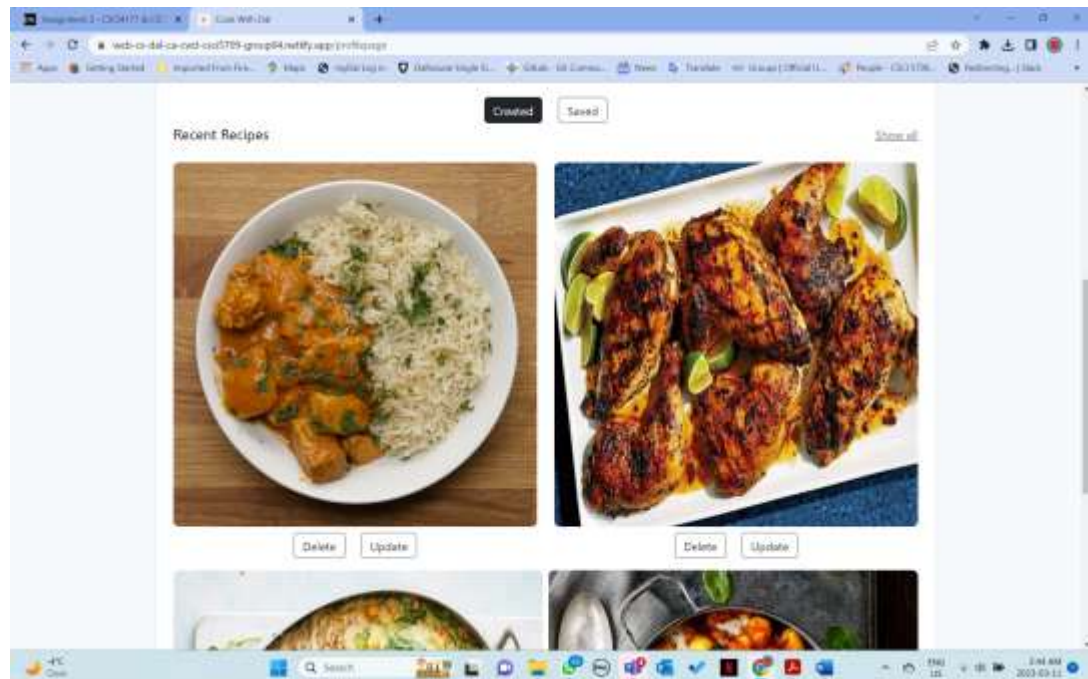


Figure 18: Display profile page having list of “Created” recipes [6]

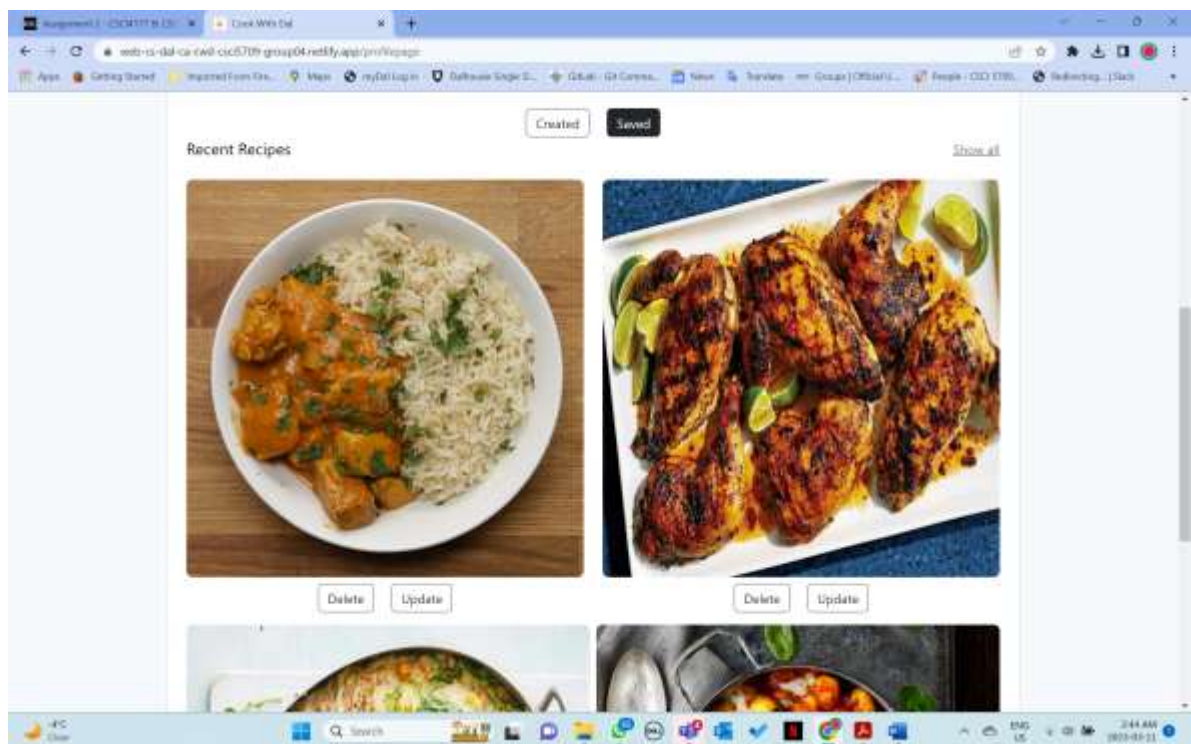


Figure 19: Display profile page having list of “Saved” recipes [6]

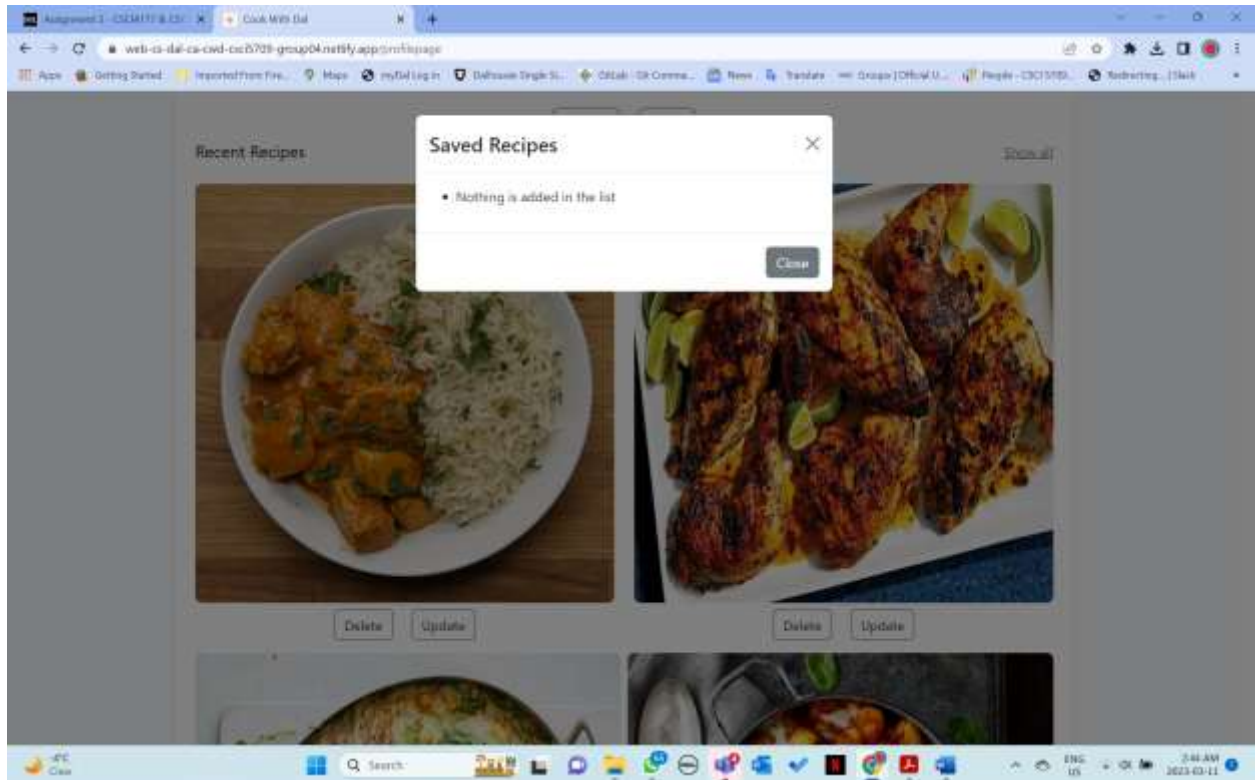


Figure 20: Page showing no "Saved" recipes in the list [6]

## **APPLICATION WORKFLOW:**

### **1. Interaction Design:**

#### **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:

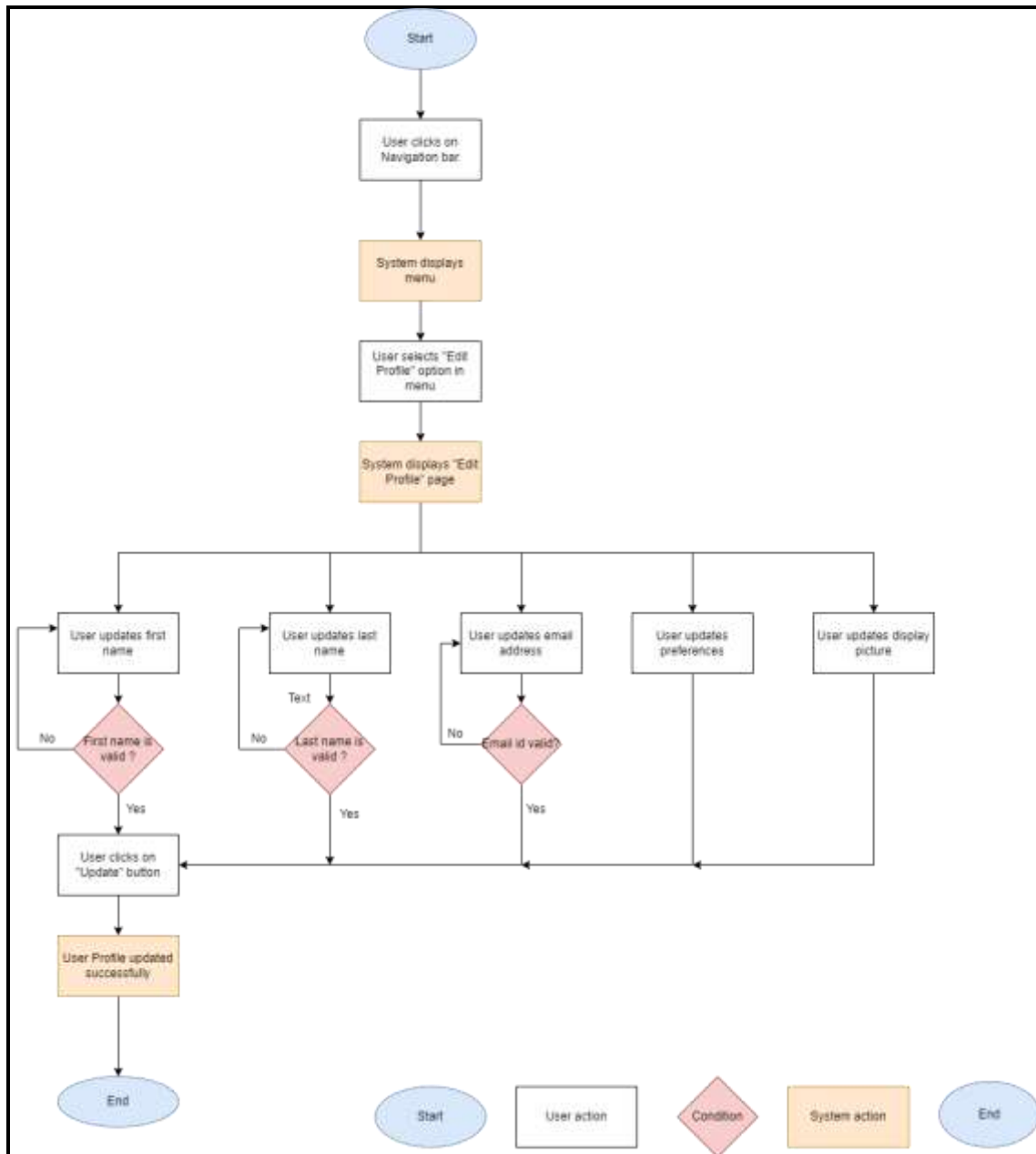


Figure 21: Task flow for updating User Profile [7]

## CLICK STREAM:

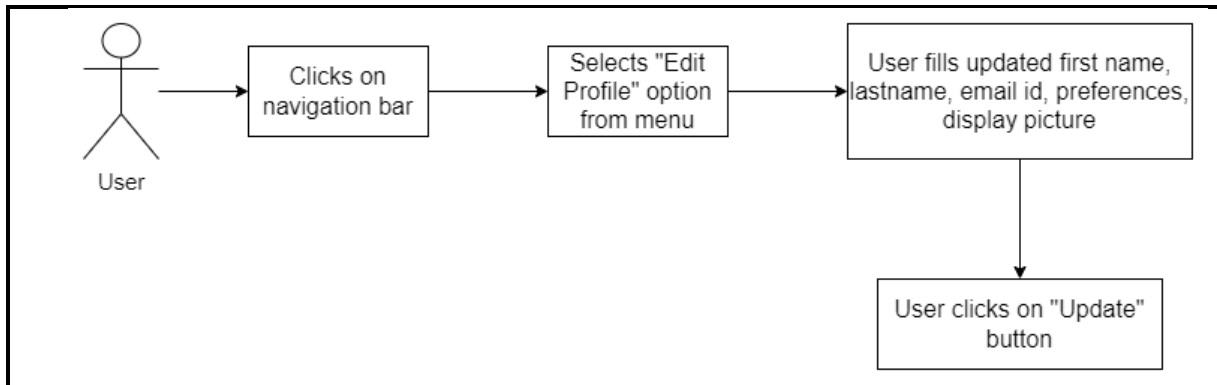


Figure 22: Click stream to update user profile details [7]

### • **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:

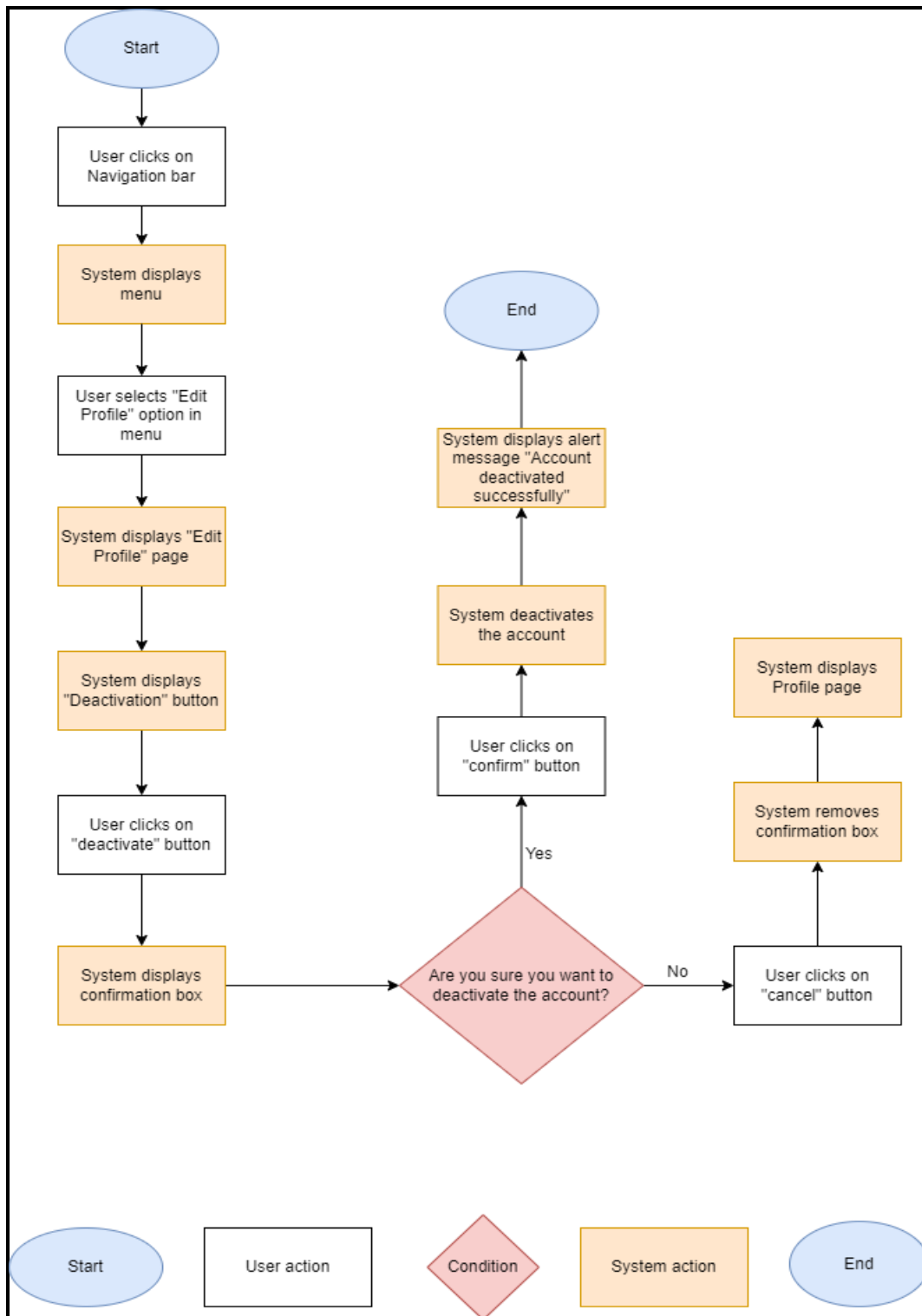


Figure 23: Task flow for Deactivating User Profile [7]

### CLICK STREAM:

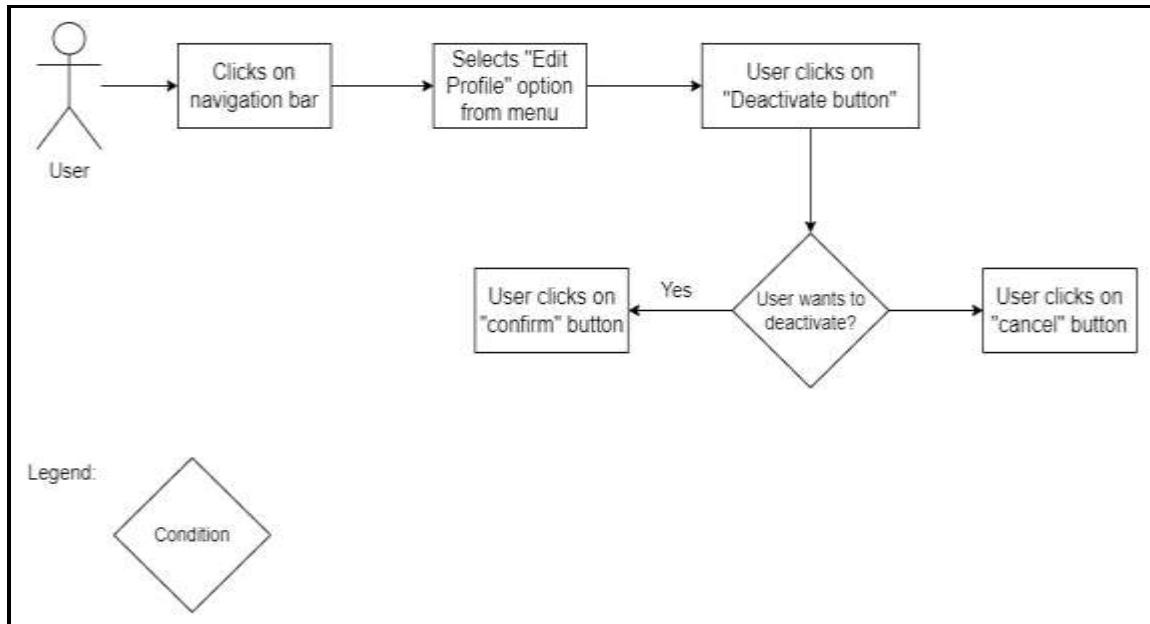


Figure 24: Click stream to deactivate user account [7]

- **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:

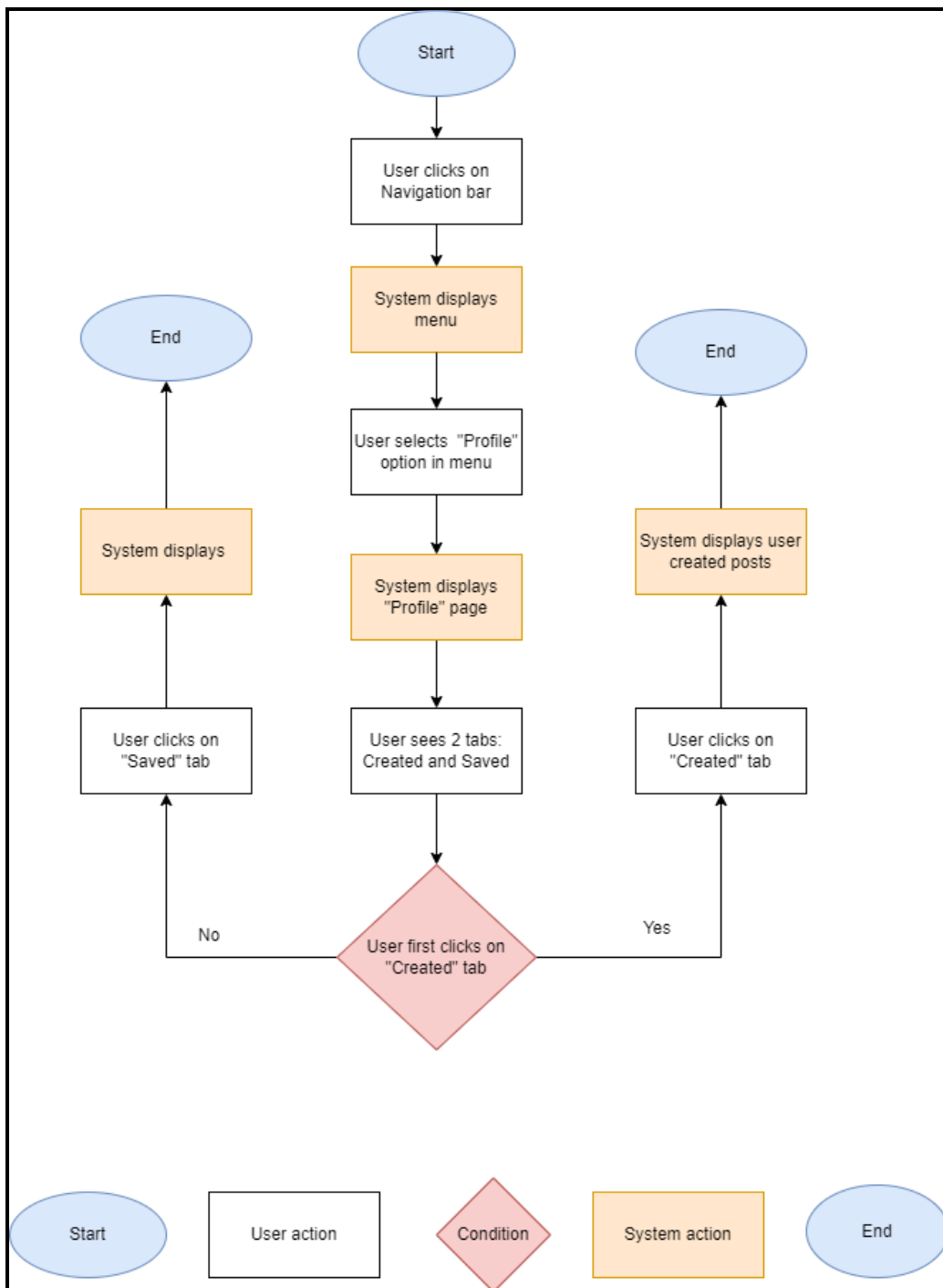


Figure 25: Task flow for displaying list of recipes [7]

## CLICK STREAM:

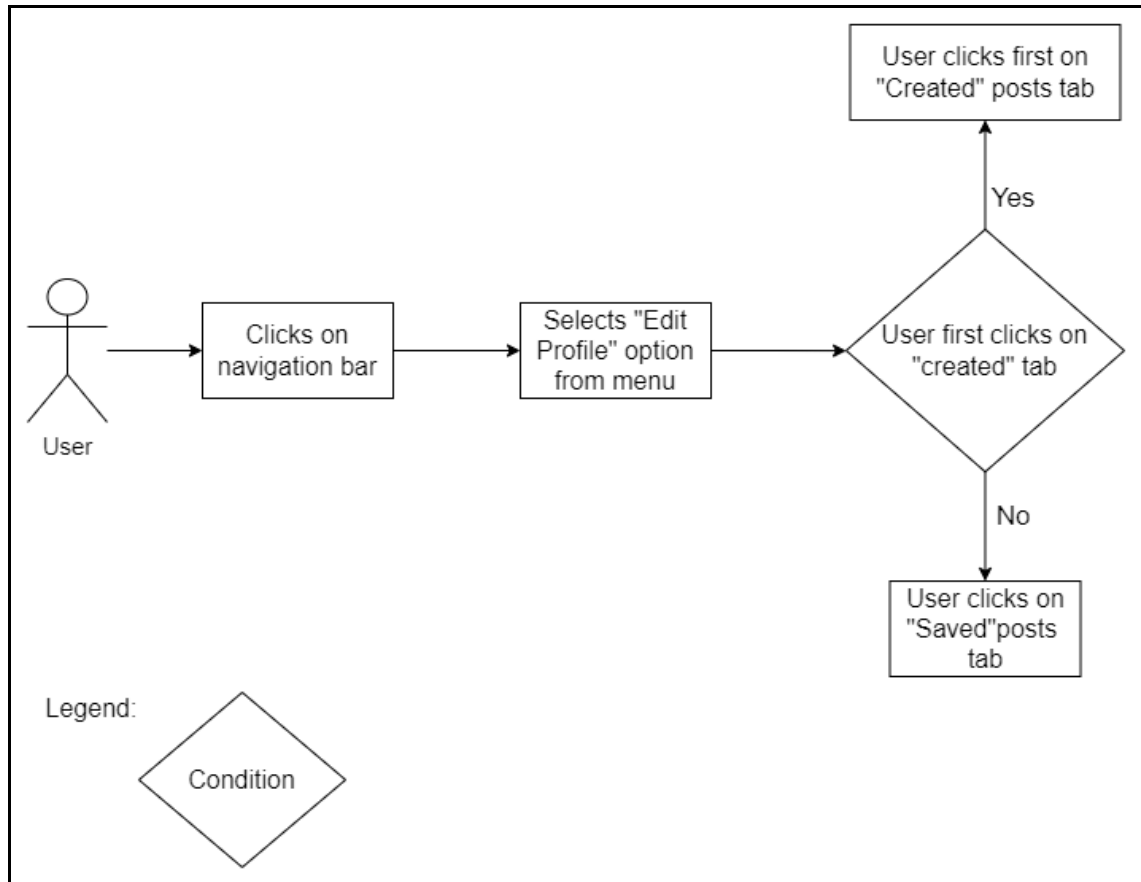


Figure 26: Click stream for displaying user details [7]

## 2. Process and Service Workflow:

The figure 27. shows the application workflow of CookWithDal. The users would be able to access the react application through web browsers. When the user clicks on any feature on the react frontend application, the react components prepare a request and send it to the back-end development node server. The node server checks for the request to be handled and fulfilled by connecting with MongoDB in the database management phase. After the request is processed and data response is received, the react components will then display the results to the users.

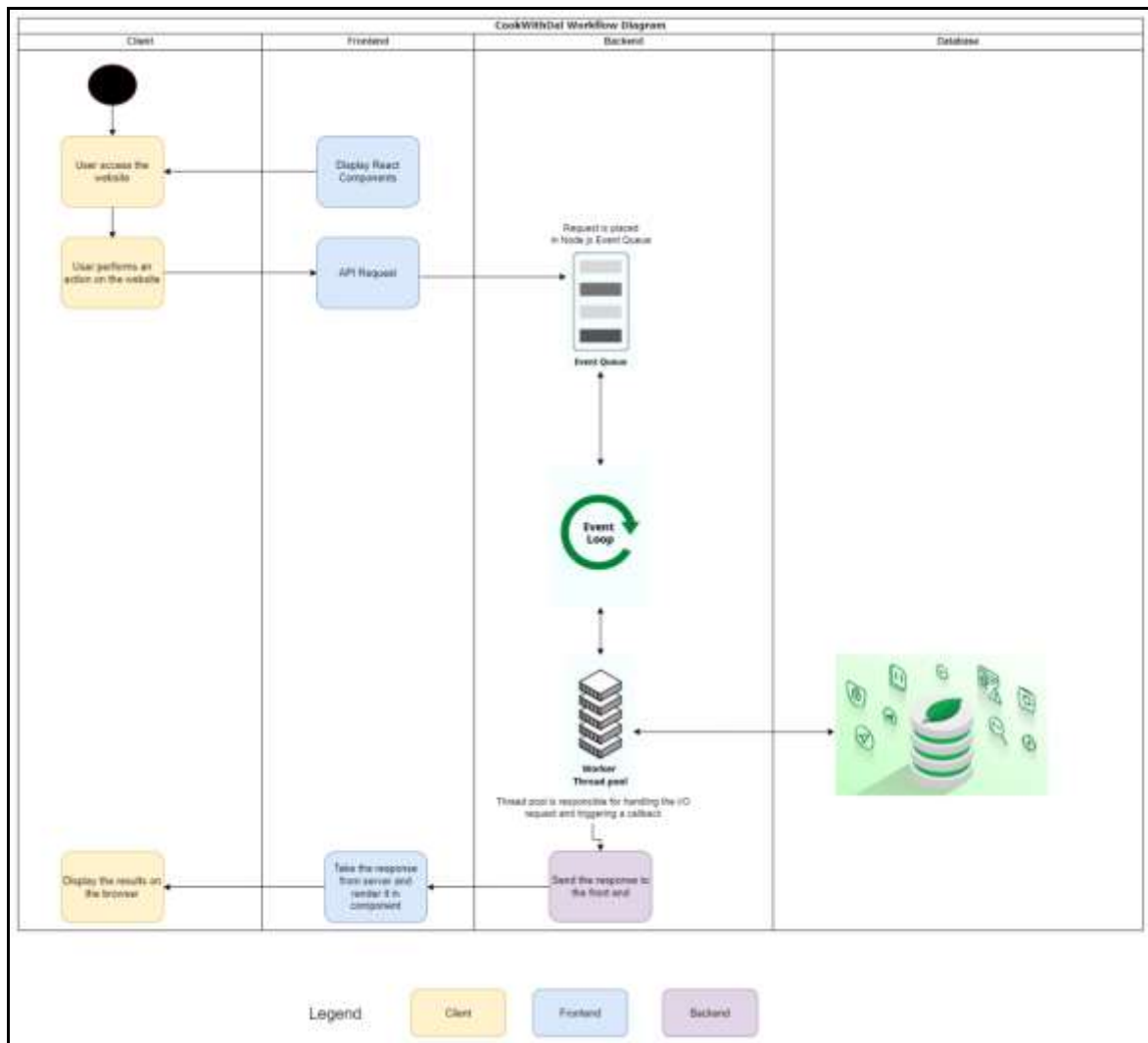


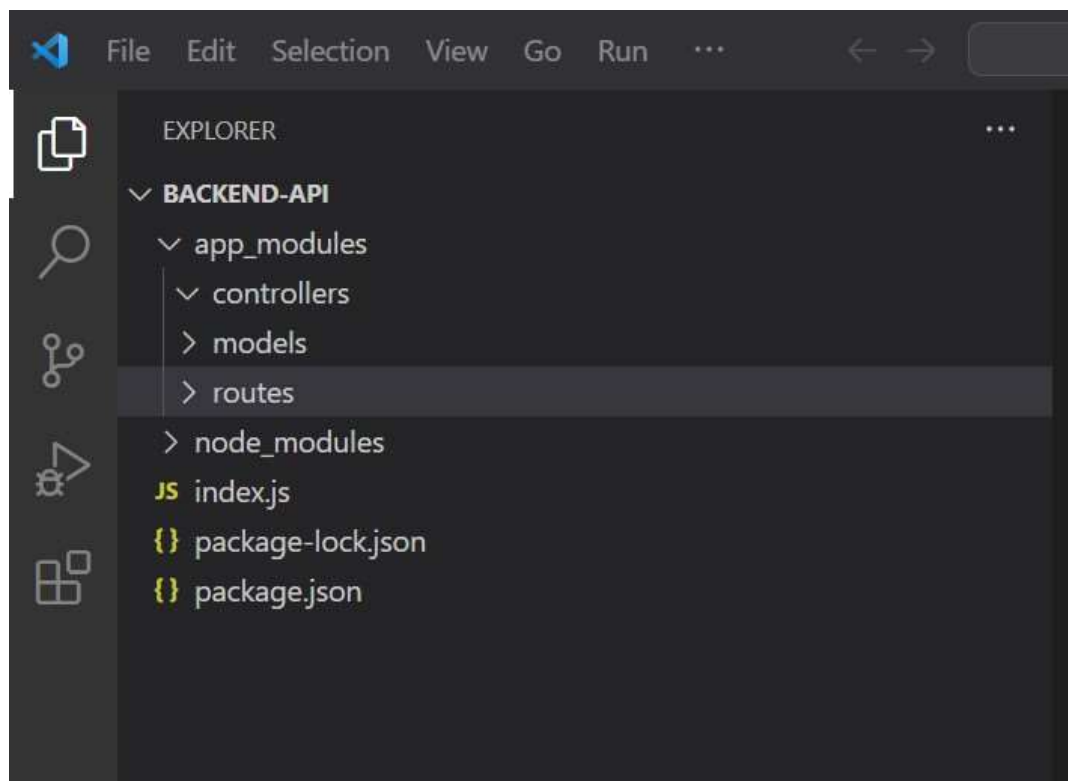
Figure 27: Process workflow of application [7][8][9]



- **Folder structure:**

- a. **Backend Project Structure**

The figure 28 shows NodeJS and ExpressJS backend application. The backend project structure comprises of controllers, models and routes folder. The routes folder specifies the URL routes for the controllers, while the controllers contain the necessary code to convert data between the frontend and backend using the ODM library to ensure compatibility with MongoDB. Additionally, the model files facilitate the serialization and deserialization of data between the frontend and database. The project's dependencies and build tasks are listed in the "package.json" file.



*Figure 28: Backend project structure*

## b. Frontend Project Structure

The figure 29 shows the frontend project structure that comprises of src folder, assets folder, package.json, public folder etc. The src folder includes “Components” folder which consists of feature-specific files/pages. “Assets” folder consists of files that consists of contents present in the features. Finally, the public folder contains the images needed to display on the website. ReactJS chose the images directly from the “public” folder.

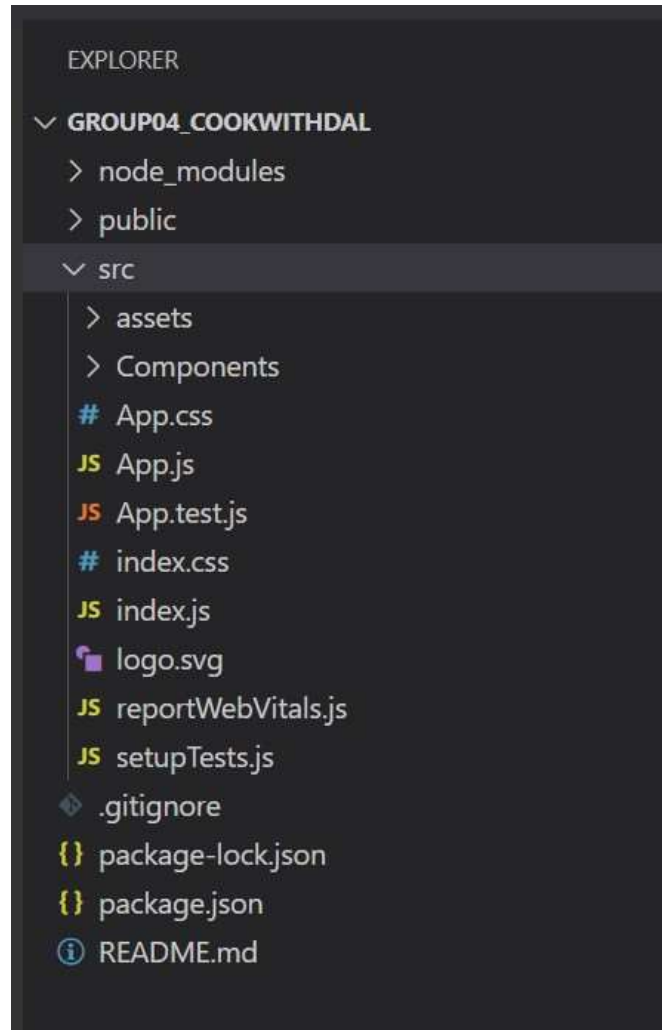


Figure 29: Frontend project structure

### 3. Application Architecture:

We would be following the model-view-controller architecture for both react frontend and NodeJS - ExpressJS backend. For the backend application, we intend on binding model data with ODM so that we could easily convert the data from the frontend and make it compatible with the format that we intend to store in the database and vice-versa. The router will be the point of entry for both the frontend and backend applications which would direct the requests based on the request parameter to specific components or controllers. The service module in the frontend client application would be responsible to communicate with the backend application using HTTP clients such as Axios or Fetch API. In order to implement the separation of concerns principle, the client requests would be handled by the backend application, which would further interact with the underlying database and return back the response from the database.

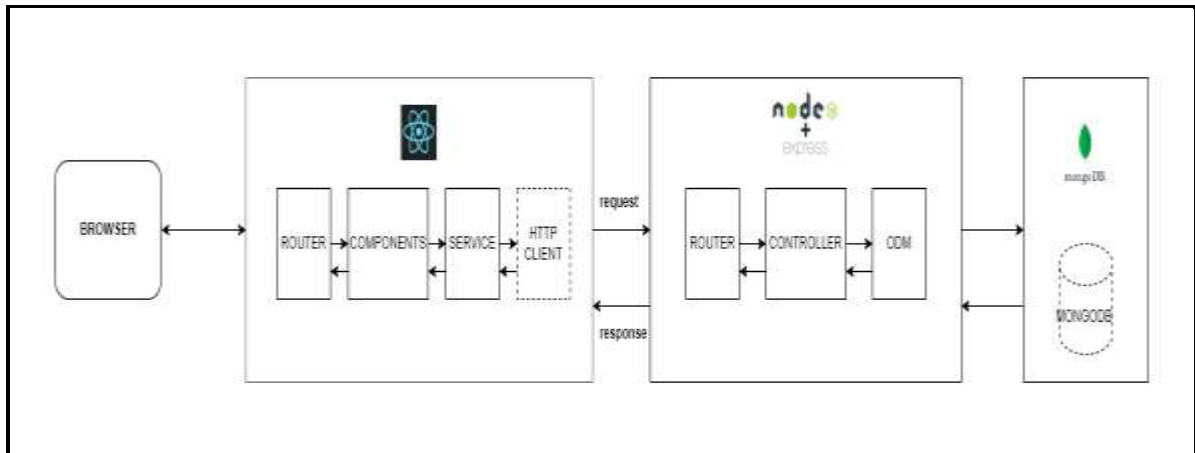


Figure 30: Application Architecture [7] [10] [11]

## **References:**

- [1] ANUJ DAWAR, FAIZA UMATIYA, PARUL RAICH, RUCHIKA, SAGARKUMAR VAGHASIA, SAIF ALI PRASLA “CSCI 5709 ASSIGNMENT 1 (GROUP SUBMISSION) (A1\_GROUP04).” Dalhousie University, [online document], 2023. [Accessed 11-March-2023].
- [2] ANUJ DAWAR, FAIZA UMATIYA, PARUL RAICH, RUCHIKA, SAGARKUMAR VAGHASIA, SAIF ALI PRASLA “PROJECT PROPOSAL (D1\_GROUP04).” Dalhousie University, [online document], 2023. [Accessed 11-March-2023].
- [3] “Create powerful business content together with Xtensio,” Xtensio.com, 14-Jun-2022. [Online]. Available: <https://xtensio.com/>. [Accessed: 07-Feb-2023].
- [4] “Figma,” Figma. [Online]. Available: <https://www.figma.com/files/recent?fuid=1203216882011676942>. [Accessed: 07-Mar-2023].
- [5] “Balsamiq cloud,” Balsamiq.cloud. [Online]. Available: <https://balsamiq.cloud/sgvkuqp/pgqrj2b/rE39C>. [Accessed: 08-Mar-2023].
- [6] “Cook with Dal,” *Netlify.app*. [Online]. Available: <https://web-cs-dal-ca-cwd-csci5709-group04.netlify.app/>. [Accessed: 11-Mar-2023].
- [7] “Flowchart maker & online diagram software,” Diagrams.net. [Online]. Available: <https://app.diagrams.net/>. [Accessed: 09-Mar-2023].
- [8] W. Crowell, “What is MongoDB?,” *OpenLogic by Perforce*, 03-Jun-2021. [Online]. Available: <https://www.openlogic.com/blog/what-is-mongodb>. [Accessed: 11-Mar-2023].
- [9] M. Khare, “What is Node.Js and why you should use it,” *Kinsta®*, 29-Apr-2021. [Online]. Available: <https://kinsta.com/knowledgebase/what-is-node-js/>. [Accessed: 11-Mar-2023].
- [10] “React.js + Node.js + Express + MongoDB example: MERN stack CRUD App,” *BezKoder*, 12-Oct-2019. [Online]. Available: <https://www.bezkoder.com/react-node-express-mongodb-mern-stack/>. [Accessed: 11-Mar-2023].
- [11] *Cloudinary.com*. [Online]. Available: [https://res.cloudinary.com/hevo/image/upload/f\\_auto,q\\_auto/v1626694700/hevo-blog/MongoDB-sm-logo-500x400-1-1.gif](https://res.cloudinary.com/hevo/image/upload/f_auto,q_auto/v1626694700/hevo-blog/MongoDB-sm-logo-500x400-1-1.gif). [Accessed: 11-Mar-2023].