

Group Project Log

Note: all information must be filled out. You must hand in the project log along with each group project deliverable for this course (e.g., milestones, proposals, reports). The percentage of work allocated to each group member must add up to 100%.

Group Name:	Implementers
--------------------	--------------

Group Members:	5
-----------------------	---

Deliverable:	Foodably Project Report
---------------------	-------------------------

Group Member Name	Work Done (%)
Mykhailo Herasimov	20
Muppidi Shashank Reddy	20
Sneha Jayavardhini Doguparthi	20
Raviteja Kase	20
Ujwal Vikas Chanda	20
Total:	100%

PROJECT REPORT

PROJECT GROUP 2 - IMPLEMENTERS

FOODABLY- ONLINE PLATFORM TO SHARE, DISCOVER, AND POST RECIPES

Members and Contributors

Mykhailo Herasimov
B00799960
my590678@dal.ca

Muppidi Shashank Reddy
B00826400
sh375383@dal.ca

Sneha Jayavardhini Doguparthi
B00846995
sn536318@dal.ca

Raviteja Kase
B00823644
rv270985@dal.ca

Ujwal Vikas Chanda
B00848764
uj225642@dal.ca

**Faculty of Computer Science
Dalhousie University**

10, April, 2020

ABSTRACT

This project report elucidates the steps involved to build a web application called Foodably. The project's purpose is to provide a single portal for the ultimate cooking experience. It will accumulate most of the processes related to food preparation. This application allows us to post and share recipes of their choice on a social platform, thus creating a perfect niche for this application in the food industry. We have achieved major functionalities of Profile management, creating and finding recipes with advanced search, creating a shopping list, providing comments and subscription. Every functionality is implemented from the designing phase with a process workflow and wireframe, which will be discussed in the report.

KEYWORDS

Architecture, Cooking, Food, Home cooking, Ingredients, Interaction design, Recipe, Target user, Wireframe, Workflow

CONTENTS

1	INTRODUCTION	1
1.1	Project Objectives	1
1.2	Live Project URL	1
2	BACKGROUND	2
2.1	Competitive Landscape	2
2.2	Problem and Approach	2
3	APPLICATION DETAILS	3
3.1	Target User Insights	3
3.1.1	User age	3
3.1.2	Location	4
3.1.3	Gender	4
3.1.4	Interests	4
3.2	User-Centered Design Approach	4
3.2.1	Information Architecture	4
3.2.2	Design and Layout	5
3.2.2.1	Low fidelity prototypes	5
3.2.2.2	High fidelity prototypes	9
4	APPLICATION WORKFLOW	23
4.1	Interaction Design	23
4.1.1	Scenario I	23
4.1.1.1	Use case—search for a recipe with specific requirements	23
4.1.1.2	Use case—view a recipe and its preparation instructions	24
4.1.2	Scenario II	24
4.1.2.1	Use case—register on the website	25
4.1.2.2	Use case—login to the website	25
4.1.2.3	Use case—create a blog post	26
4.1.3	Scenario III	27
4.1.3.1	Use case—subscribe to the author	27
4.1.3.2	Use case—create a shopping list	28
4.1.4	Scenario IV	29
4.1.4.1	Use case—find grocery stores nearby	29
4.1.4.2	Use case—contact developers/support	30
4.1.5	Scenario V	31
4.1.5.1	Use case—leave a comment	31
4.2	Process and Service Workflow	32
5	REFERENCES	47

INTRODUCTION

The project's purpose is to provide a single portal for the ultimate cooking experience. It accumulates most of the processes related to food preparation. Each customer of this web application can find something for themselves. The intention is not only to create a functional technical solution for searching and preparing recipes but make it entertain so that anyone can visit our web application and have a fun time. The project provides a social platform for cooking enthusiasts where they could share their experiences. The platform is inclusive as possible, meaning that despite customer profile or background they should find a recipe appropriate to their skill level, timing requirements or level of engagements. The web application comprises a set of useful utilities that would eliminate the necessity to distract the customer from the main content or primary purpose of visiting the website. The cooking application will provide a sufficient level of accessibility. Customer can help and assistance at any point of interaction with the website.

Project Objectives

Food is the most primitive form of comfort. The main objective of Foodably is to create a common platform for food lovers. The market has several applications that concentrate on building a social relationship but there are only a few applications that bring together food lovers as a single community. This application allows us to post and share recipes of their choice on a social platform thus creating a perfect niche for this application in the food industry.

Foodably lets users post recipes they love and discover recipes based on the ingredients of their choice. Their main goal is to create an environment for food lovers to share their most favourite recipes and how they have improvised existing recipes. This also allows them to learn and explore new recipes. The application also creates a path for beginners to develop their cooking skills.

As a result of our work, we managed to create a large portion of tasks that we initially planned. In the current version of the website, the user can fully manage profiles and recipes. The users can search using the name of the recipe or filter by many various parameters. The only functionality that we did not manage to complete in full is a blog post. We create a front end to presents a list of the blogs as well as an individual blog post; however, we were unable to create the backend and connect it to the front end of the website. But as for the rest of the application, we developed and hosted the API. We planned a set of collections for our NoSQL database and connected it to the backend. The frontend implementation is fully interactive and compatible with backend endpoints.

Live Project URL

Project is hosted on AWS and since modifications aren't done to the code after Assignment 4 we are using the providing the same URL, which we have used for Assignment 4 and the modifications aren't done due to time constraints and most of the features were completed by Assignment 4 itself.

Foodably website URL:

<http://csci5709-a4.us-east-1.elasticbeanstalk.com>

The latest code for frontend is in the develop branch of the git repository. Gitlab repository link:

<https://git.cs.dal.ca/herasimov/foodably.git>

The latest code for backend is in the develop branch of the git repository. Gitlab repository link:

https://git.cs.dal.ca/herasimov/group2_implementers_foodably_backend.git

BACKGROUND

Competitive Landscape

As Foodably web application is for food enthusiasts, we have referred following web applications to facilitate ideas for our project development:

- a) kitchen stories
- b) allrecipes
- c) tasty
- d) food network
- e) epicurious

Foodably has some similarities and differences with the above list. Our application has mainly concentrated for finding and posting new and delicious food recipes. The overall design and the flow of application are made simple and easy to understand by the user.

User need not sign up for viewing a recipe or watching the video of a dish, so this will help them to get the required results as soon as they type the dish name.

The layout is consistently maintained along different pages.

User can select the required recipe with various filters like cuisines, preparation time, ingredient specific etc. The special feature in our application is the ingredients selection i.e. the user can do an advance search for a recipe based on the ingredients they have.

It differs from our competitors as it is not confined only to the chefs to post the dishes; any common person who wants their recipe to be known to everyone can use Foodably.

Problem and Approach

While contemplating the project idea, we talked a lot about our current life, and the only topic that repeatedly was coming up is the food selection and how our food habits had to change due to migration to the other country. Most of us do not have extensive experience Implementers

with home cooking. There is a set of recipes that we all know and love from birth; however, we only saw how others would cook them for us. As you start to live on your own, especially with the limited budget, home cooking becomes a vital part of our life.

To help other people that have a similar lifestyle, we decided to create a website that would be able to cope with cooking. After analysis of the competitive land field, we did not find that application that would be able to combine food recipes, effective instructions, utilities and cultural experience exchange in one technological solution. We identified the most important aspects of our application during numerous discussions. The utilities had to be efficient and useful; the instructions concise and on point; recipes simple and tasty. To fulfill identified tasks, we created wireframes and high fidelity prototype for evaluation of the intermediate results.

In order to create the final product, we specified the technological stack for the website. We decided upon developing our application using Express.js, Angular and MongoDB. The task flows and interaction inside the website will be discussed in detail in the further sections of this document.

APPLICATION DETAILS

Target User Insights

The description of the target audience will be based on the four main parameters: age, location, gender, and interests.

User age

We expect the Foodably users will fall into one of the three age ranges.

First, teenagers and college students that start their independent life. Since in most cases, when young people have a tight budget, they tend to cook and prepare food on their own. We think that Foodably can be an excellent starting platform for this audience due to the wide variation of filters and search tools.

Second, the significant portion of the user base is young and mid-age working professionals. Due to their hectic lifestyle and need for a balanced diet, they are most likely to need a creative approach to the optimization of food preparation. We expect them to create and share their personal experience with the way they tend to approach food culture in the office environment.

And the last user group is people that are close to retirement or already retired. Their motivation to use Foodably would be primarily enthusiastic. The older people tend to have more free time and usually start to travel and spend their time experiencing entertainment with low intensity where food preparation could play a major part. We expect from that user group to be involved in the preparation of sophisticated recipes that take a lot of time. Also, they can be the backbone of article writers that would like to share their secrets after experimenting with various ingredients and recipes.

The only one user group with we will not consider—small children because food preparation often requires cutting ingredients with sharp knives or the use of hot cooking utensils. We believe that adult supervision must be present during the familiarization of kids with cooking.

Location

The website will be primarily used at places where people have access to desktop, laptop or tablet. Foodably content is better consumed on computer screens because users can publish blogs and media content. However, such a feature of the website as a shopping list can be accessed using mobile phones. It was created specifically for the user usability purpose, where users can quickly review the shopping list with ingredients in any location.

Gender

The website will be a gender-neutral system. The colour palette or any visual components will not differentiate any specific gender group. Foodably is inclusive for everyone, regardless of gender identification.

Interests

We expect people with various backgrounds to share their knowledge and experience on the Foodably website. However, we anticipate that there will be some groups of recipes made by people that are interested in travelling, traditions of particular cultures, street food, and high-level culinary. The connoisseurs of good restaurant food and general food lovers can create a great community based on interests in targeted types of food.

User-Centered Design Approach

We have taken reviews and frequently asked questions (FAQ's) from similar kind of websites in the market into consideration to understand the expectations of users. During the designing phase of the application, based upon the analysis obtained we have designed the components like styles, features, and support.

Information Architecture

During the planning of the site map, we identified 12 web pages that cover all the functionality of the website. Four of them require user authentication. Most of them primarily related to the creation of content or personalization of the user experience. The site map components were grouped into individual modules that represent one web page. Site map items present the general distribution of features and navigation between them. The full diagram available in [Fig. 1]. For the creation of the sitemap and the rest of the graphical resources and diagrams, we used Sketch extended with libraries or design kits [1].

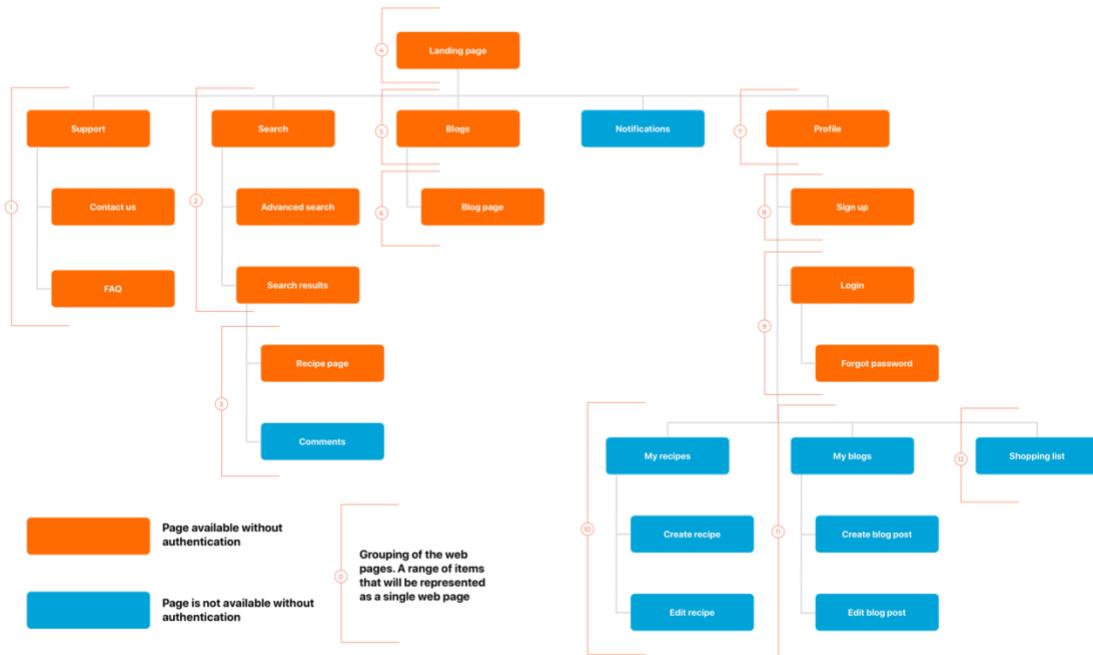


Figure 1 Foodably Site map [1]

Design and Layout

As a reference for that layout, we used concepts from a similar website called Kitchen Stories [2]. The illustrations used are a modified version of free vector files available on the unDraw website [3]. All images were taken from the Unsplash image bank that provided free stock high-resolution items [4]. As for the icons I used icons kit provided by Apple Inc. The SF Symbols have a set of thousands of free icons available free for developers [5].

1.1.1.1 Low fidelity prototypes

Fig. 2 is the wireframe for a recipe page. This page contains the video of the recipe, steps followed to create the recipe and ingredients required for recipe. The comment section is provided, where users can be able to comment and reply to the comment for each recipe. There is a section where it recommends, more similar dishes like the recipe which is in the video.

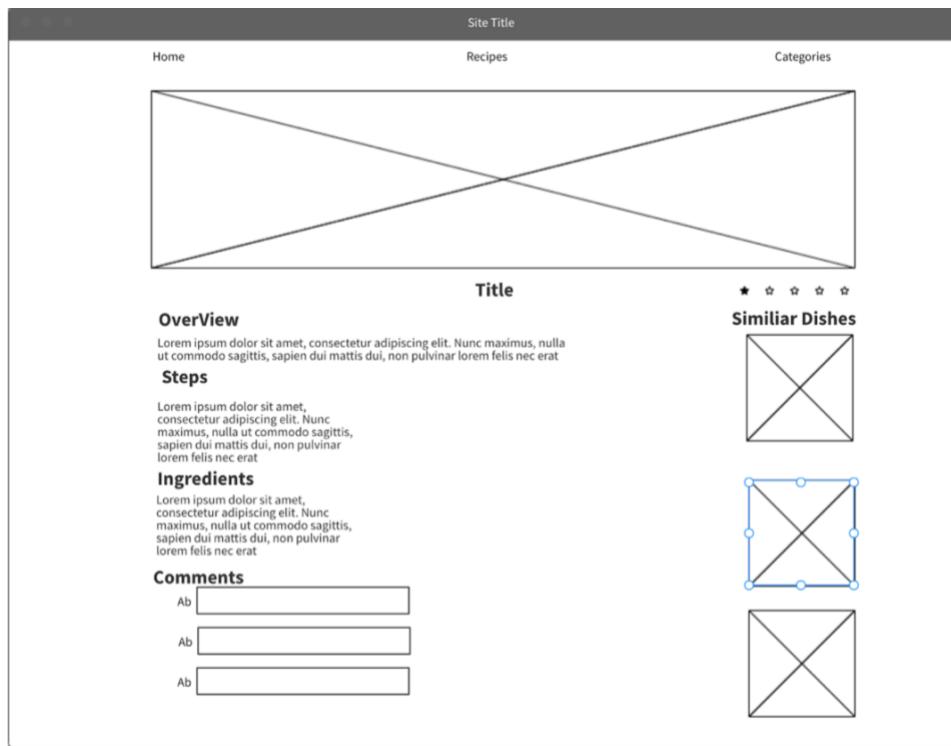


Figure 2. Recipe main page [7]

Fig. 3 shows a create recipe page. The header and footer will have a different appearance. The high-fidelity prototypes section demonstrates these items in greater detail. As for the creation form, it will have separate sections for describing the main content of the recipe, ingredient list, and preparation instructions.

This high-fidelity prototype of a 'Create Recipe' page is designed to look like a web browser window. The title bar says 'A Web Page' and shows a URL 'https://foodably.com'. The main content area has a header with 'FoodAble' and other navigation links like 'Home', 'Recipes', 'Categories', 'Maps', and 'Subscribe'. To the right of these links is a 'MyAccount' button. Below the header, the page displays the breadcrumb 'Home > MyAccount > Create Recipe'. The main form area contains several input fields: 'RecipeName' with a placeholder box, 'Enter Details' with a large text area, and 'Steps' with another text area. At the bottom of the form is a 'Save' button. The footer of the page includes links for 'About', 'FAQ', 'Contact', 'Facebook', 'Instagram', and an ellipsis (...).

Figure 3. Create recipe page [7]

Fig. 4 is the wireframe for the profile page. This page will be available or visible to the user only after the user logs in. The profile page enables the user to post a recipe, create a blog, and create a shopping list of ingredients. Based on each button, the user can perform different activities.



Figure 4. Profile page [7]

Fig. 5 is the wireframe of the shopping list page. This page will be available in users my profile section only after the user logs in. The profile page enables the user to post a recipe, create a blog, and create a shopping list of ingredients and users can view the nearest stores to buy groceries. The shopping list is helpful when the user wants to make an ingredient list to buy in the nearest stores.

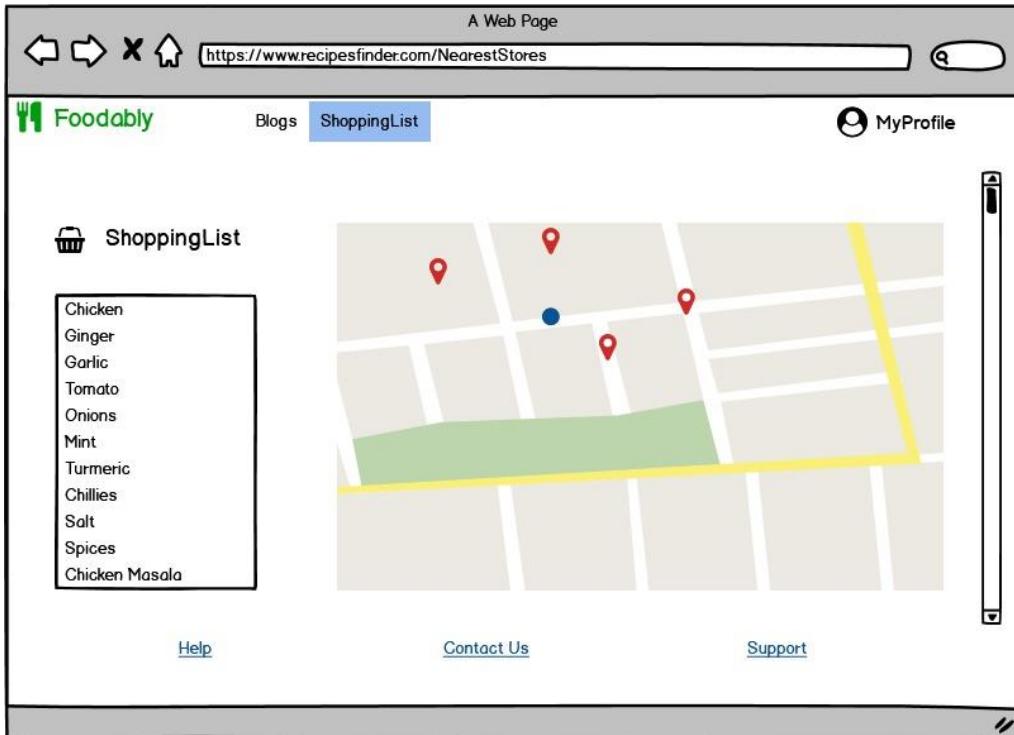


Figure 5: Shopping List [7]

Fig. 6 is a wireframe for the FAQ page. It contains the frequently asked questions and answers.

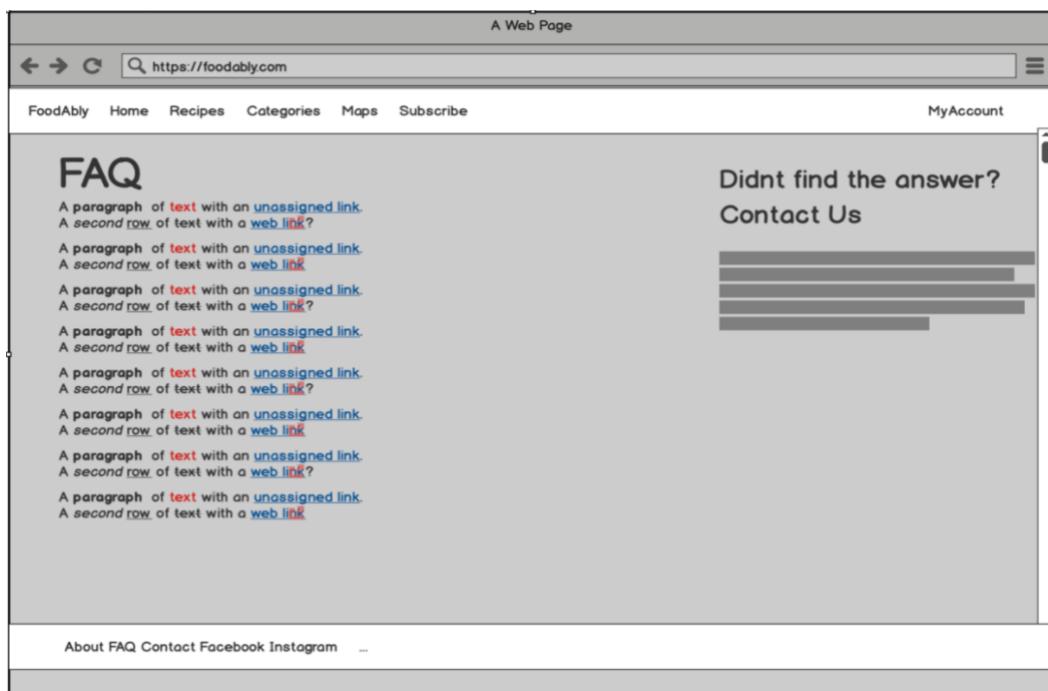


Figure 6:FAQ [7]

1.1.1.2 High fidelity prototypes

The wireframe in Fig. 7 represents a sign-up form. It will be accessible on any page of the website. Users can create a new one using an email address and the passwords are encrypted which provides security. Server-side validations are done to check if the email already exists or not while registering.

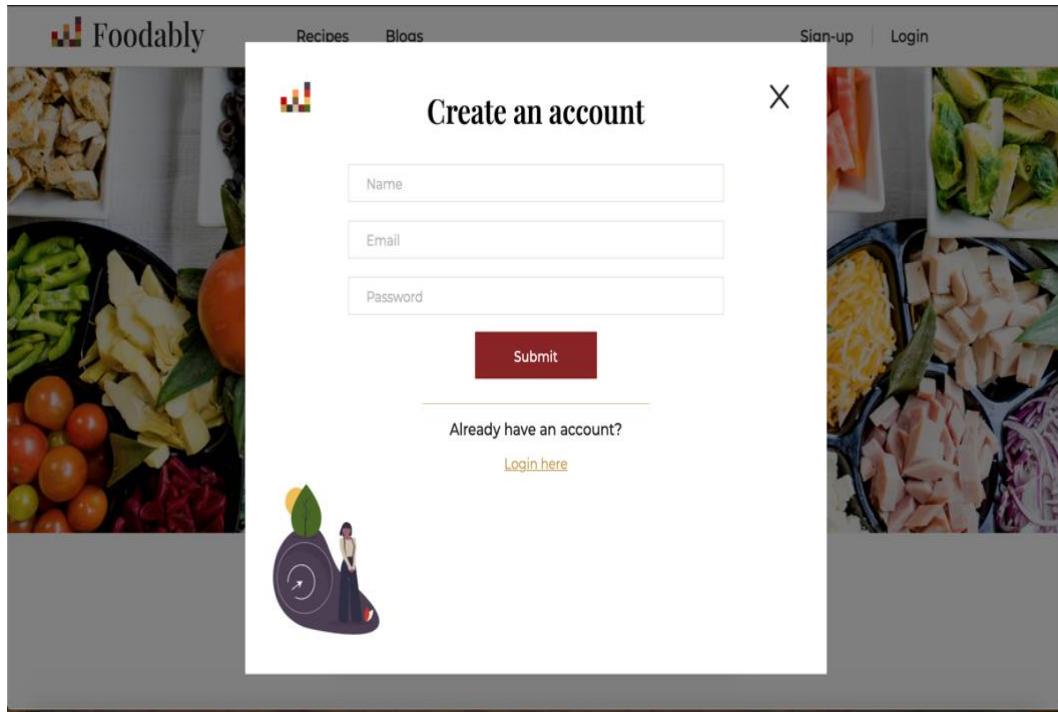


Figure 7. Registration page[1]

Fig. 8 shows a login form that lets users log in using email. It also has a reference to the password recovery flow. Server-side validations are done to check if the email exists or not while signing in.

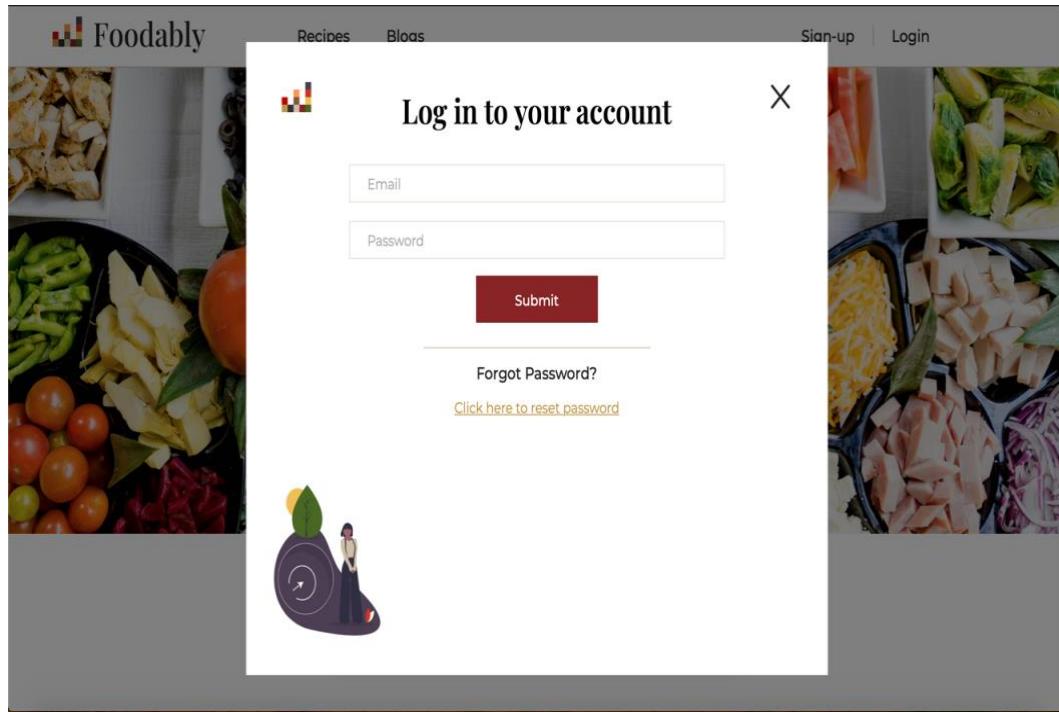


Figure 8. Login page[1]

The wireframe in Fig. 9 shows a Forgot Password page. The user needs to enter the registered email to reset the password, server-side validations are done to check if the user is registered with Foodably or not.

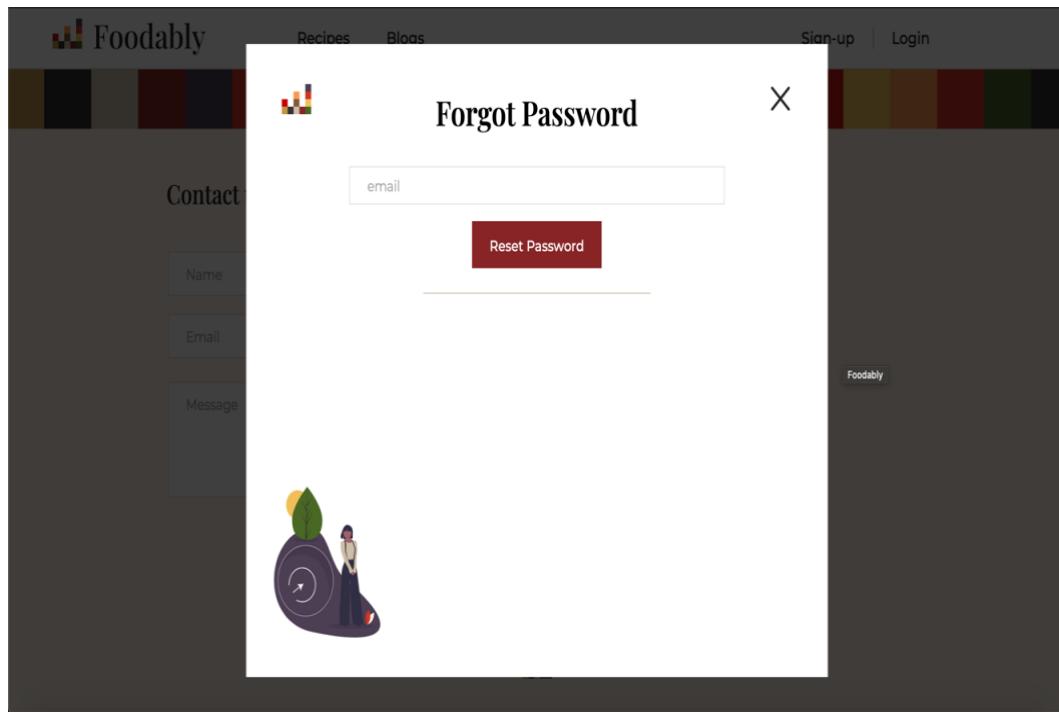


Figure 9. Forgot Password Page[1]

Fig. 10 demonstrates the Reset Password page, which will be accessible once the user provides a valid email in the Forgot password page and an email will be sent to reset the password. On clicking the link, this page gets enabled. User can reset the password.

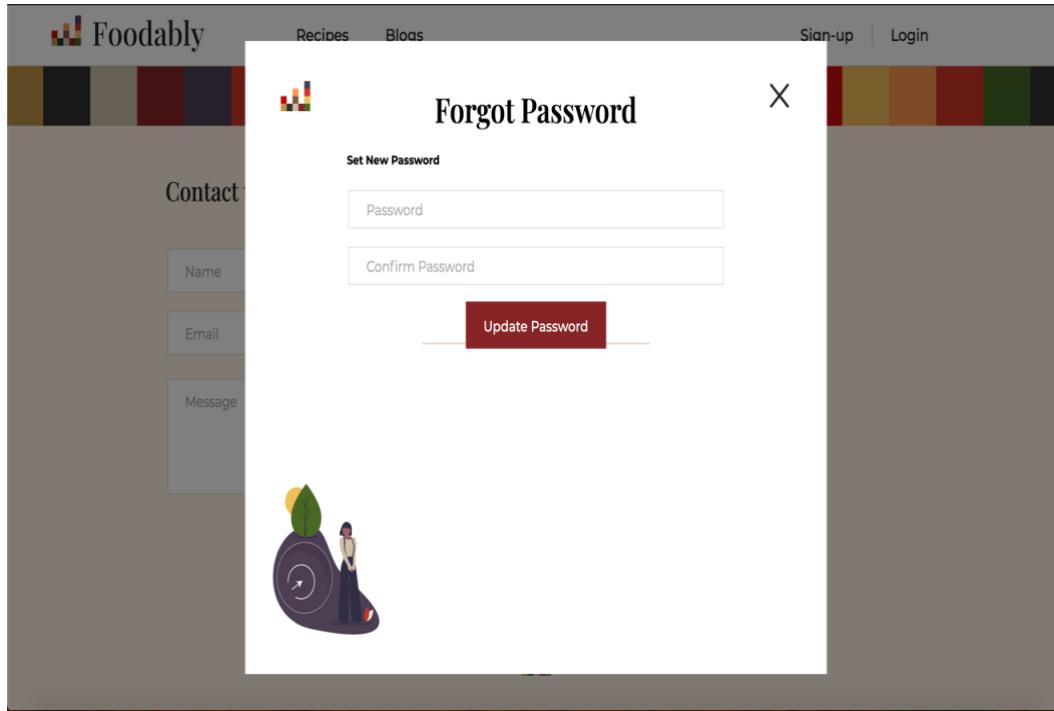


Figure 10. Reset Password[1]

Fig. 11 demonstrates the Landing page, where user can be able to view recipes and search for the recipe. Users can sign up and log in, once the user logs in profile and logout options are enabled.

Fig. 12 demonstrates the blog articles page where at the top, there will be three most-read articles from a specific date. The main section will contain all the articles created by users; some of them will be marked as featured. Featured articles will be differentiated by card looking appearance that contains a title image of the article.

Fig. 13 depicts the concept of the blog article page. It will contain information about the author, content of the article and related recipes that users can attach while creating an article. The page will be dynamically generated based on types of sections. There will be two types of sections; first is the text that represents one paragraph or an image.

Fig. 14 illustrates the wireframe for the search page. After the user searches with a generic keyword in the search bar from the home page, the user is directed to this page. This page displays all the recipes related to the keyword searched by the user. A list of recipes from the result can be sorted based on rating, recently published, preparation time, and Relevance.

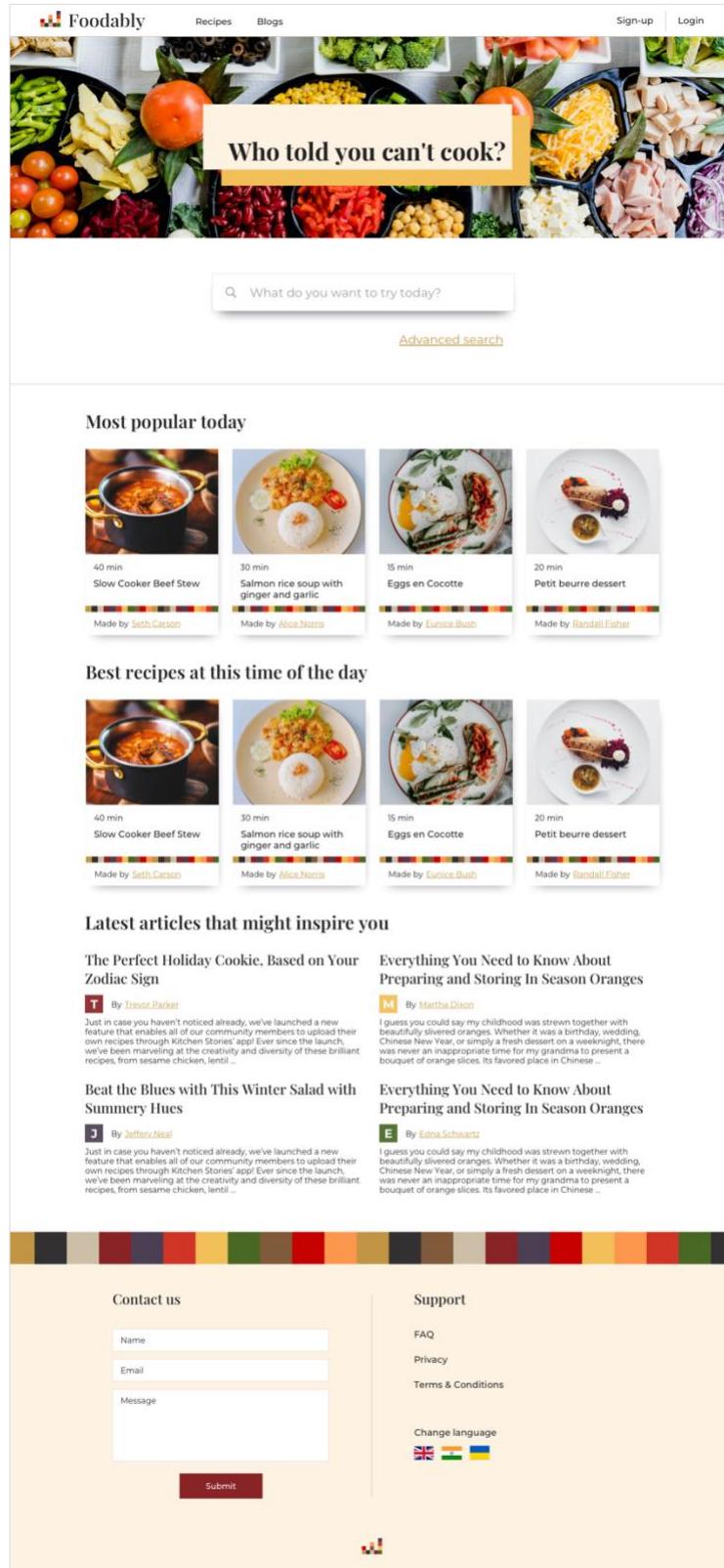


Figure 11. Landing page[1]

Figure 12. Blog articles page[1]

 Foodably

Recipes Blogs

Sign-up Login

Maui By Air The Best Way Around The Island

 Written by Trevor Parker
Sun 16, 2014



Choosing A Static Caravan

Placeholder text used in web design, typography, layout, and printing in place of English to emphasize design elements over content. It's also called 'decorative' (or 'filler') text. It's a common tool for mockups or 'heavily stylized' documents.

Placeholder text is mostly a part of a Latin text by the classical author and philosopher Cicero. Its words and letters were originally added by additional or removed, to deliberately render its content nonsensical. It is not genuine, correct, or complete Latin anymore. While lorem ipsum still resembles classical Latin, it actually has no meaning whatsoever. As Cicero's text doesn't contain the letters K, W, or Z, alien to latin, these are often added to the original. In addition, it is used to mimic the typographic appearance of European languages, as are digraphs not to be found in the original.

In a professional context it often happens that private or corporate clients demand a publication to be made and presented with the actual content still not being ready. Think of a news blog that's filled with content hourly on the day of going live. However, reviewers tend to be distracted by comprehensible content, say, a random text block, instead of the actual message. Placeholder text is often used to fill up space and time in such cases. Besides, random text risks to be unintentionally humorous or offensive, an unacceptable risk in corporate environments. Lorem ipsum and its many variants have been employed since the early 1960ies, and quite likely since the sixteenth century.



Placeholder text is mostly a part of a Latin text by the classical author and philosopher Cicero. Its words and letters were originally added by additional or removed, to deliberately render its content nonsensical. It is not genuine, correct, or complete Latin anymore. While lorem ipsum still resembles classical Latin, it actually has no meaning whatsoever. As Cicero's text doesn't contain the letters K, W, or Z, alien to latin, these are often added to the original. In addition, it is used to mimic the typographic appearance of European languages, as are digraphs not to be found in the original.

Most of its text is made up from sections 110.32-3 of Cicero's *De finibus bonorum et malorum* (On the Boundaries of Good and Evil). This may also be translated as *On the Nature of Good and Evil*. Neque committitur quod quisq; dolorem ipsum quia dolor est, committit vero quia dolor est. Nihil est in mundo tristius nisi est dolor. (Neither is there anyone who loves grief itself since it is grief and thus wants to obtain it"). It was found by Richard McClinton, a philologist, director of publications at Hampden-Sydney College in Virginia; he searched for citations of conjecturæ in classical Latin literature, a term of remarkably low frequency in that literary corpus.

Related recipes

	
40 min Slow Cooker Beef Stew <small>Made by Seth Carson</small>	20 min Petit beurre dessert <small>Made by Randall Fisher</small>

[See comments \(114\)](#)



Contact us

Support

[FAQ](#)

[Privacy](#)

[Terms & Conditions](#)

[Change language](#)





Figure 13. Blog article page [1]

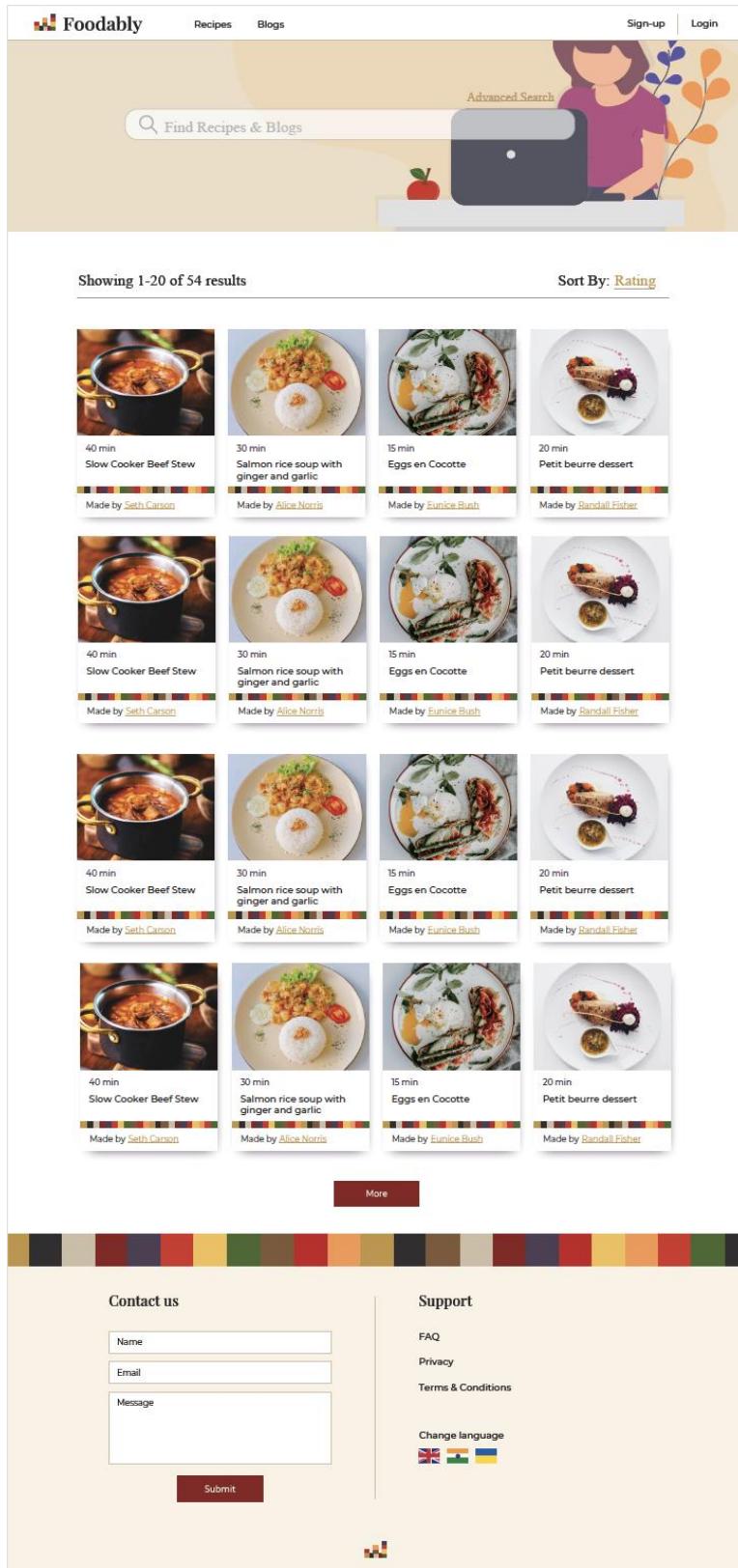


Figure 14. Recipe search page [1]

Fig. 15 is the wireframe for the advanced search page. The user is given an advanced search option below the search bar, where Users can filter searching tasks at a more specific level. User is given options to choose like a type of dish, diet, time taken to prepare a dish, and ingredients.

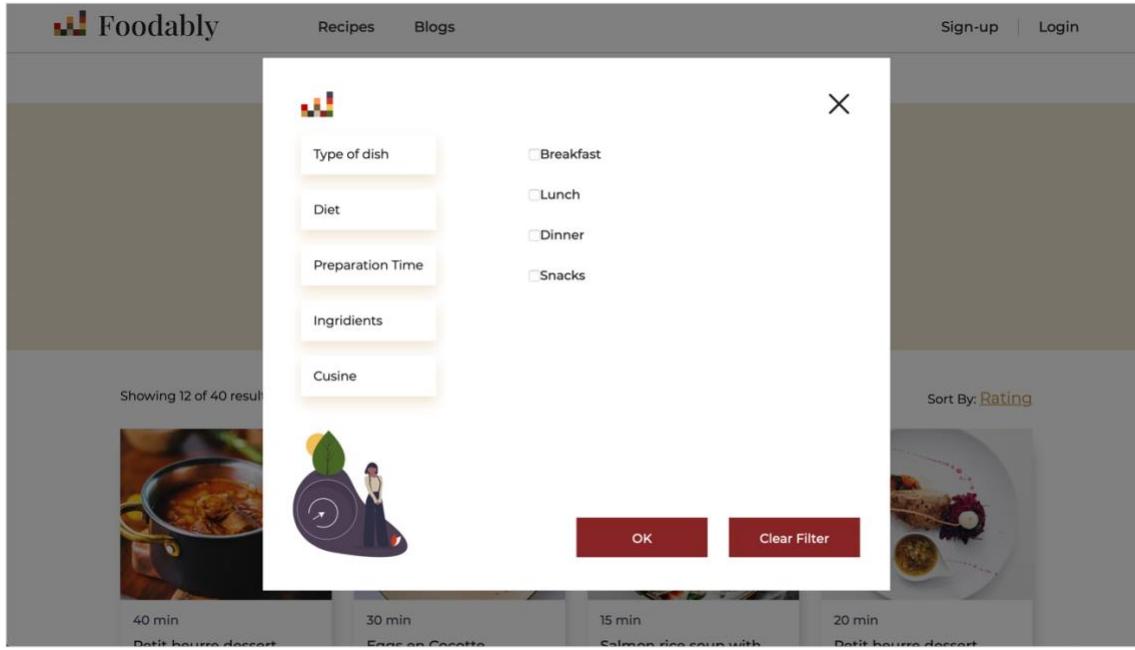


Figure 15. Advanced search pop up[1]

Fig. 16 demonstrates the Profile page, which enables user to Add/Edit Bio and Delete Profile. Users can be able to view the Shopping list and recipes created by the user. Users can also be able to view the nearest locations of grocery stores on the map.

Ingredients	Quantity	Units
$\frac{1}{2}$ cup warm water	1	g
4 cups all-purpose flour (500 g), plus more for dusting	1	kg
1 teaspoon baking powder	1	tbsp
1 teaspoon baking soda	1	g

Figure 16. Profile Page [1]

Fig. 17 is the wireframe for creating a blog. Users can create and post a blog with their experiences on foods like a scenario where a user has visited a new place and wanted to share their experience with the local food or a good restaurant they have been. Users can add a display picture, which is used as a post of the respective blog in other pages of the website along with the title. While creating a blog, users must provide title, description, and content of the blog. Users have additional options to add location, images, and videos.

Fig. 18 below is a wireframe for editing the blog. Users are given an option to edit their blogs as part of profile management. Users need to login to edit their blog and can access the security features of their blog. Users can also add the list of additional ingredients at any point of instance.

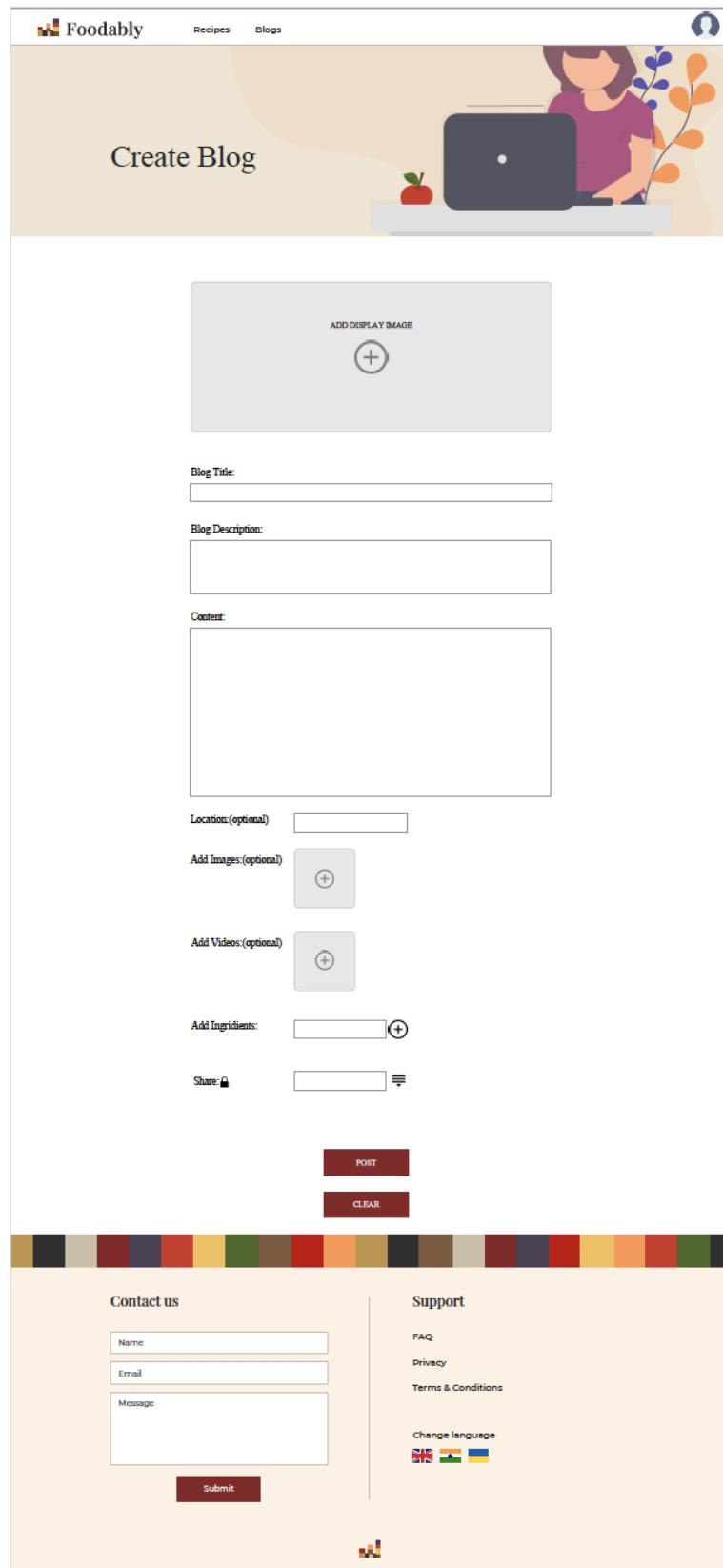


Figure 17.Create blog[1]

The screenshot shows the Foodably 'Edit Blog' interface. At the top, there's a header with the Foodably logo, a navigation bar with 'Recipes' and 'Blogs' links, and a user profile icon. Below the header is a decorative illustration of a person working at a laptop with a bowl of fruit nearby. The main title 'Edit Blog' is centered above a large image of a bowl of chili with sour cream and garnish. A red 'EDIT' button is visible at the bottom right of the image.

Blog Title: Lemony Salmon Creamy Spinach

Blog Description:

Season salmon filets generously with salt and pepper. Quarter mushrooms. Roughly chop parsley and chives. Finely dice garlic. Halve lemon, juice one half in a small bowl, and quarter the other half into wedges.

Content:

Heat vegetable oil in a large frying pan over medium heat. When the oil is hot, add salmon to the pan, skin-side down, and let cook for approx. 3 min., or until crispy. Add butter to the pan and let fry for another 2 min. Turn off heat, flip salmon, and let cook for another 3 min., or until salmon is at your preferred doneness. Remove from pan and let rest.

Location (optional): Halifax

Add Images (optional):

Add Videos (optional):

Add Ingredients: salmon filets, baby spinach, mushrooms

Share:

POST **CLEAR**

Contact us

Name:
 Email:
 Message:

Submit

Support

FAQ
 Privacy
 Terms & Conditions
 Change language

Figure 18 Edit blog[1]

The design that is presented on the Fig. 19 shows a form that will be used for the creation of the recipe. It is divided into three sections. The first is responsible for the primary information about the recipe. The second one helps to create a list of ingredients. This section is expendable. When a user adds a new ingredient to the list, the set of controls will show up to fill all the necessary information. The last section helps to create preparation steps for the individual recipe. The user can add the description for the step along with the associated list of ingredients. The ingredients can be selected from the dropdown list. The available options will be predefined based on the information filled in the ingredient list section of the recipe.

Create recipe

Primary Recipe Info:

- Title:** Recipe title (input field) and Category (dropdown menu).
- Video presentation:** Youtube link to the recipe video (input field) and Preview image (input field).
- Description:** Text area for describing the recipe.
- Cuisine:** Cuisine (dropdown menu).
- Dish type:** Dish type (dropdown menu).

Time to prepare: 10 min. **Is this recipe vegetarian?** (checkbox)

Ingredients for one serving:

Ingredient	Quantity	Units
Ingredient name	1	Select
Ingredient name	1	Select

Preparation steps:

Step	Description	Ingredients
1/2	Describe preparation step in details.	Ingredients: Select, 1, Select
2/2	Describe preparation step in details.	Ingredients: Select, 1, Select

Buttons: New ingredient, New step, Add ingredient, Cancel, Submit.

Figure 19. Create recipe form[1]

Fig. 20 demonstrates the redesigned recipe page. We had to redesign it because the previous version did not include information related to the preparation steps as well as metadata about the recipe. New design now included ranking information along with likes and preparation time. The update also better fits the design theme of the website. All the colours and general elements such as buttons, dividers and controls were reused from other web page designs.

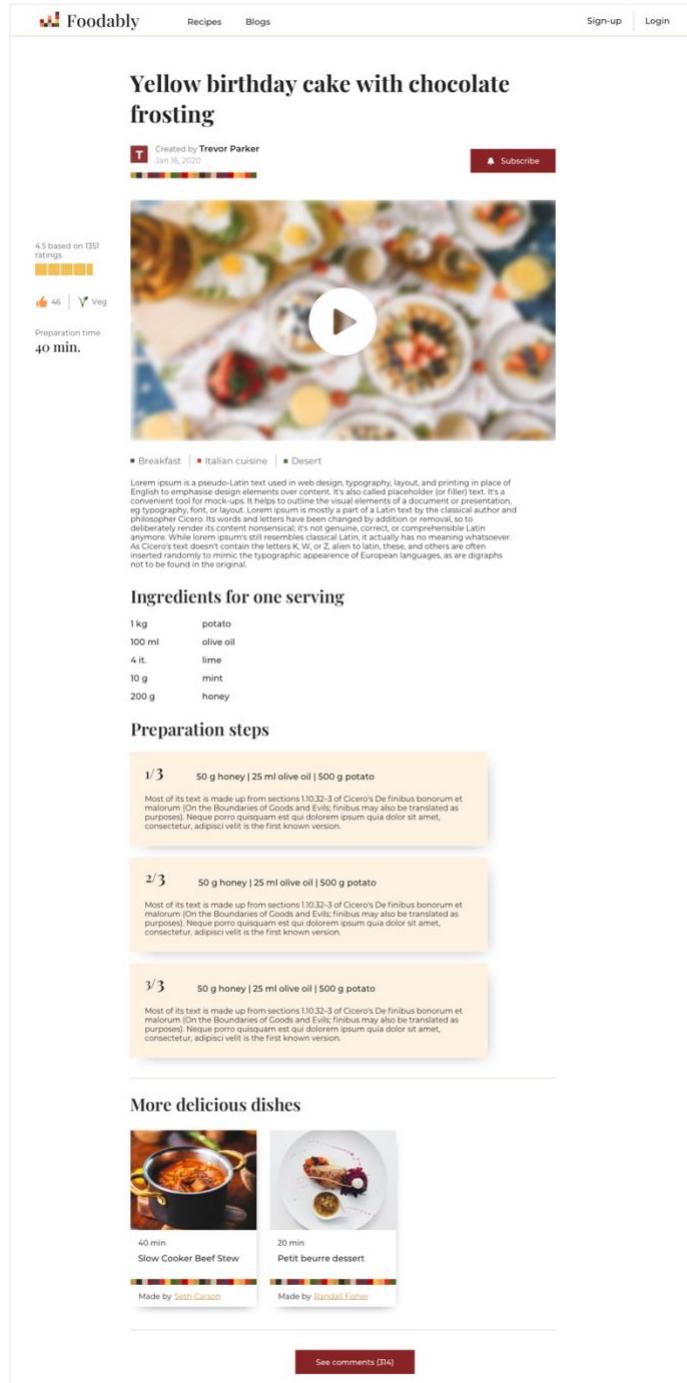


Figure 20. Recipe page[1]

The prototype in Fig. 21 represents a list of user recipes. The design has controls for creating new recipes as well as edit existing ones. Unlike recipe cards from the search results, the user recipe card contains less information. It is expected that the users will be acquainted with their own work; hence they will be able to navigate freely across the full list

The screenshot shows a web-based application for managing user recipes. At the top, there is a navigation bar with the logo 'Foodably' (a stylized 'F' icon), 'Recipes', 'Blogs', 'Sign-up', and 'Login'. Below the navigation bar, a profile section for 'Misha Gerasimov' is displayed, featuring a large green square placeholder for a profile picture with a white 'M' on it. The profile bio reads: 'Biker, risk-taker, drummer, reclaimed wood collector and recent OCAD grad. Making at the sweet spot between art and computer science to create not just a logo, but a feeling. I prefer clear logic to decoration'.

Below the profile section are three navigation buttons: 'My recipes', 'My blog posts', and 'Shopping list'. The 'My recipes' button is highlighted with a dark grey background and white text.

The main content area displays a list of recipes. The first item is a dark grey card with a '+' icon and the text 'New recipe'. To its right is a thumbnail of a yellow birthday cake with chocolate frosting, followed by the title 'Yellow birthday cake with chocolate frosting' and a truncated description: 'Lorem ipsum is a pseudo-Latin text used in web design, typography, layout, and printin...'. Below this is another card for the same recipe, showing a different angle of the cake.

Further down the list are two more cards for the same recipe, each with a different image of the cake. The last card in the list is highlighted with a yellow background and features icons for edit, delete, and trash.

At the bottom of the main content area is a 'More' button in a dark red box. Below the main content area is a decorative footer bar consisting of a horizontal row of colored squares (gold, black, grey, red, blue, yellow, green, etc.).

The footer is divided into two sections: 'Contact us' on the left and 'Support' on the right. The 'Contact us' section includes input fields for 'Name', 'Email', and 'Message', and a 'Submit' button. The 'Support' section includes links for 'FAQ', 'Privacy', 'Terms & Conditions', and a 'Change language' dropdown with icons for English, Indian, and Ukrainian.

Figure 21. User recipes

APPLICATION WORKFLOW

Interaction Design

To describe possible events on the web site, we created ten use cases that cover nine features. For the creation of the task flow diagrams, we used the “User Flow Diagram - Template” library created by Jaroslaw Ceborski [6]. To avoid repetitions in task flow diagrams, we used the concept of references where a portion of the task flow can be inferred from the other one. On the diagrams, reference points illustrated as red circles. The grey squares represent a web site page. The diamonds serve as decision points, where the outcome can be only positive or negative. The last piece of the diagram is orange and blue oval-shaped items. Orange represents user actions and blue system actions.

Scenario I

You are a student that returned home after a busy day of classes, and you want to prepare a quick dinner but do not know what to cook and how to cook it.

1.1.1.3 Use case—search for a recipe with specific requirements

Search recipe using the advanced search.

1. User visits cooking website homepage
2. User clicks advance search button
3. The system shows a popup with corresponding fields
4. User selects an item from “preparation time” dropdown
5. User enters a list of available ingredients
6. User clicks “search” button
 - 6.1. System does not find any results and displays a communicate with text guiding to change search parameters
7. System shows a list of available recipes (Next, review and prepare recipe use case)

The first diagram [Fig. 16] shows a task flow required for searching for recipes. Users can go through a regular search path or advanced one (it provides options for filtering and searching with advanced parameters).

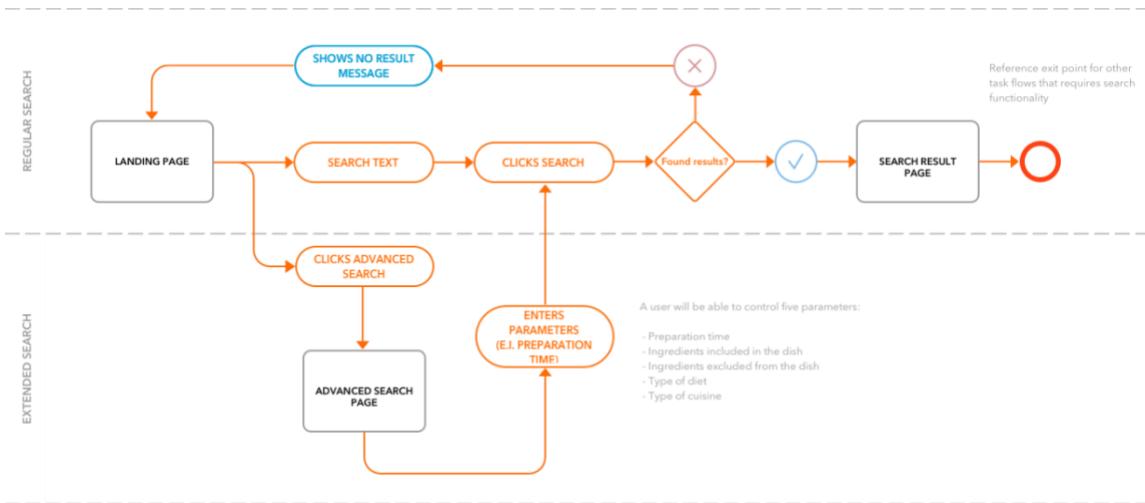


Figure 16. Use case—search recipe using regular or advanced search

1.1.1.4 Use case—view a recipe and its preparation instructions

Review the recipe and go through instructions.

1. User clicks on a recipe
2. System shows a recipe page
3. User click on the “play video” button
4. System shows a video clip with the recipe preparation instructions
5. User scrolls down the page
6. System shows a list of ingredients and step-by-step instructions
7. User follows provided instructions

The next diagram [Fig 17.] shows a number of ways a user can access and interact with the recipe page. The flow assumes that the user has finished the search flow first.

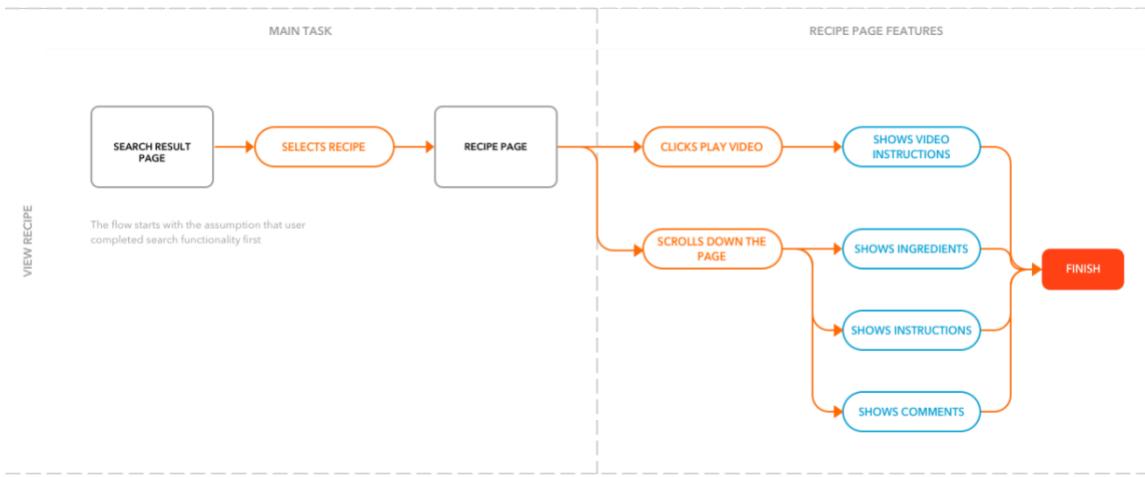


Figure 17. Use case—view recipe and explore available sections

Scenario II

You are a traveller that visited a new country and tried local cuisine. You just found an interesting fact about the local food preparation technique and want to share this experience. You found out about an Foodably cooking website and decided to write a blog post.

1.1.1.5 Use case—register on the website

Registration on Foodably web site.

1. User visits cooking website home page
2. User clicks “sign up” button
3. System shows a popup with options to sign-up with Google or email
 - 3.1. User selects to sign up with Google
 - 3.1.1. System shows a Google login page
 - 3.1.2. User logs-in with Google
 - 3.2. User enters a username, email, password, and confirmation of the password
 - 3.3. System makes a real-time validation and shows a message if some field is already registered in the database or does not comply with the rules
 - 3.3.1. If the email exists in the database, the system offers to log in
 - 3.3.1.1. User selects “log in” button (Next, use case 4.1.2.2)
 - 3.3.2. If the password does not comply with security rules, the system suggests changing it accordingly
 - 3.4. User makes changes if needed
 - 3.5. User click “submit” button
4. System shows information that the user is successfully registered
5. System redirects the user to the homepage and shows username at the profile panel

The registration task flow [Fig. 18] is a part of profile management features. Due to the complexity of registration tasks, it has a reference to the login task flow [Fig. 19]. The login and sign-up processes reference each other, creating a closed cycle where users can easily switch between tasks and successfully accomplish the process of authentication.

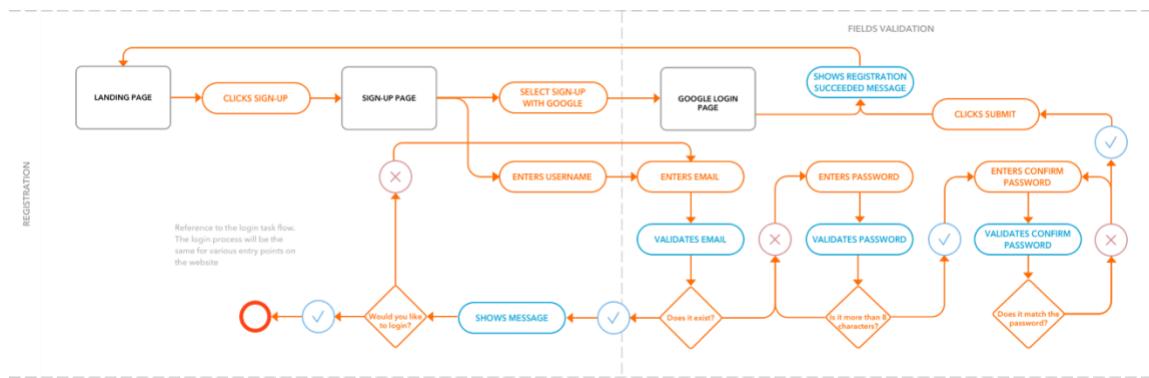


Figure 18. Use case—register a new user

1.1.1.6 Use case—login to the website

Prerequisites: User has an account

1. User visits cooking website home page
2. User clicks “login” button
3. System shows a popup with options to log-in with Google or email
 - 3.1. User selects to login with Google
 - 3.1.1. System shows a Google login page
 - 3.1.2. User logs-in with Google
 - 3.2. User enters email and password
 - 3.2.1. If password or password does not comply with the validation rules, the system makes online validation and highlights errors
4. User clicks “login” button
5. System validates user
 - 5.1. If authentication fails, the system shows a message and asks the user to correct the user’s credentials
6. System shows landing pages with visible “profile” button

The login task flow [Fig. 19] has an entry reference point since there are many use cases that require authentication. The login form will be available on any page of the web site as part of the header. Users will have two options of authentication, either using personal email or Google account.

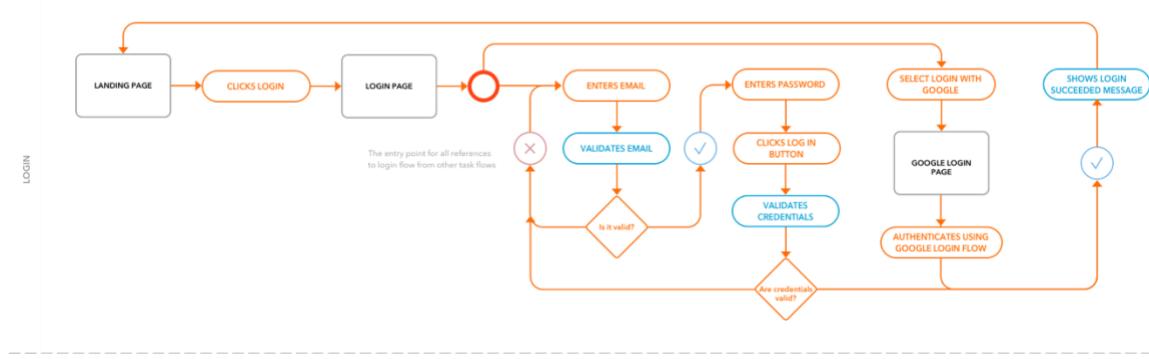


Figure 19. Use case—login to an existing account

1.1.1.7 Use case—create a blog post

Prerequisites: User is logged in.

Create a blog post and submit it to the blog articles feed.

1. User clicks on the profile icon
2. System shows user profile information
3. User clicks on “my blog” tab
4. System shows a list of blog posts created by the user
 - 4.1. If the list is empty, the system shows an empty page with “create” button
5. User clicks on “create button”
6. System shows create blog form
7. User enters in the text of the blog, its title, and optionally link to the recipe on the website

- 7.1. If the main text of the blog less than 50 characters, system blocks “submit” button
- 7.2. If the main text of the blog reaches 2500 characters, the system blocks entering of more text
8. User clicks “submit” button
9. System saves a blog post
10. System shows a new blog post in the profile of the user

The remaining task flows have a prerequisite requirement—the user must be authenticated before completion of the task flow.

The [Fig. 20] is an extended version of the use case described above. The sequence of tasks illustrates steps for the creation, edition, and deletion of the blog post. All of these options will be available on the "my blog articles" tab of the profile page. In there, users can review all previously created blog posts as well as create a new one.

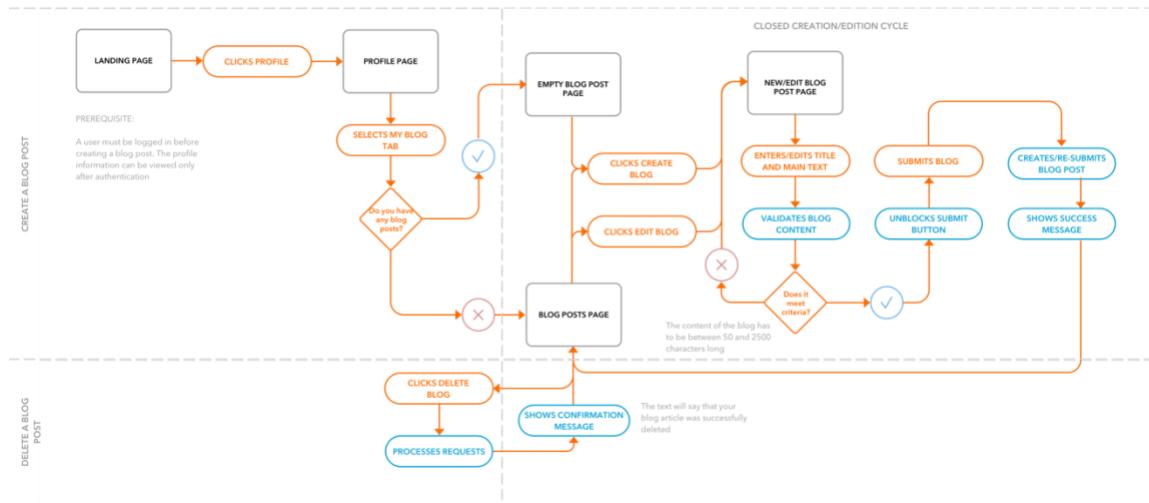


Figure 20. Use case—create/edit/delete and view user’s blog posts

Scenario III

You are a working professional that found a great author who posts recipes for a balanced diet. You decided to prepare a dish by following instructions from one of the recipes; however, you lack some of the ingredients. You decided to create a shopping list of groceries.

1.1.1.8 Use case—subscribe to the author

Prerequisites: User is logged in.

Find a recipe and subscribe to the new ones posted by the author.

1. Repeat steps mentioned in use case 4.1.1.1 ([Search for a recipe with specific requirements](#))
2. User clicks on a recipe
3. System shows a recipe page
4. User scrolls down the page to find information about the author

5. System shows the author's information such as name, profile photo and a short bio.
6. User clicks on the "subscribe" button
7. System confirms subscription by changing the style of the "subscribe" button

The task flow illustrated in [Fig. 21] describes how a user can subscribe or unsubscribe from updates made by an author. The intention for the current feature—create social groups that have relatable tastes or preferences. All recipes will have an author section; it will allow users to quickly get information about a person and subscribe to the content posted by that author.

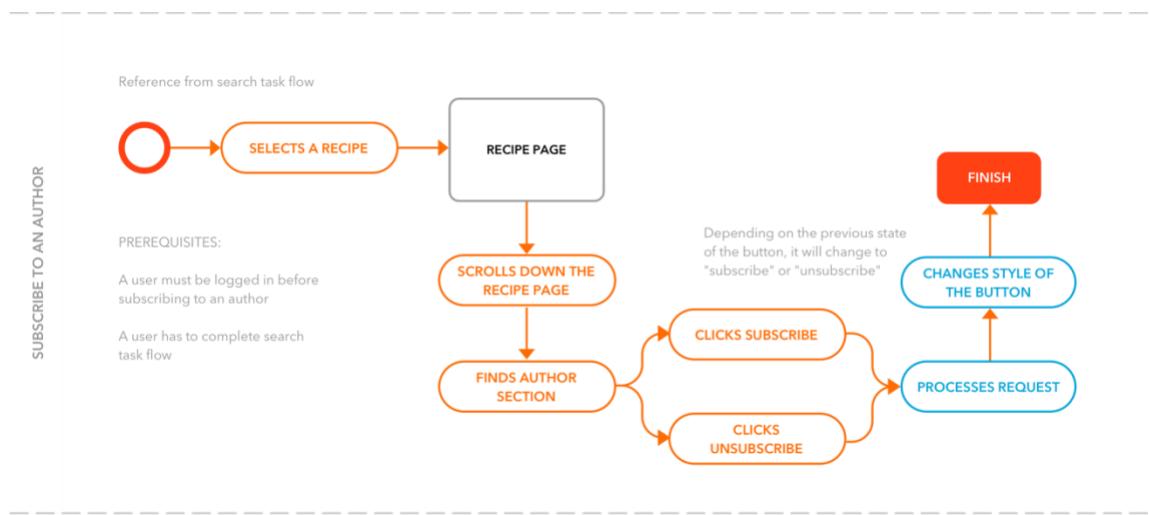


Figure 21. Use case—subscribe to a recipe author

1.1.1.9 Use case—create a shopping list

Prerequisites: User is logged in.

Find a recipe and export its ingredients to the shopping list. View the list of exported items.

1. Repeat steps mentioned in use case 4.1.1.1 ([Search for a recipe with specific requirements](#))
2. User clicks on a recipe
3. System shows a recipe page
4. User scrolls down the page to list of ingredients
5. System shows ordered list of ingredients required for the preparation of the recipe
6. User clicks on the "add to shopping list" button
7. System confirms action by showing a message
8. User click on the profile icon
9. System shows profile information
10. User clicks on the shopping list tab
11. System shows exported items in the shopping list)

The task flow presented in [Fig. 22] covers a feature called the shopping list. The intention of this feature to let a user easily create and access list of shopping items exported form a

recipe. The export button will be available in the ingredients section of the recipe page. The exported items can be viewed using a separate tab on the profile page.

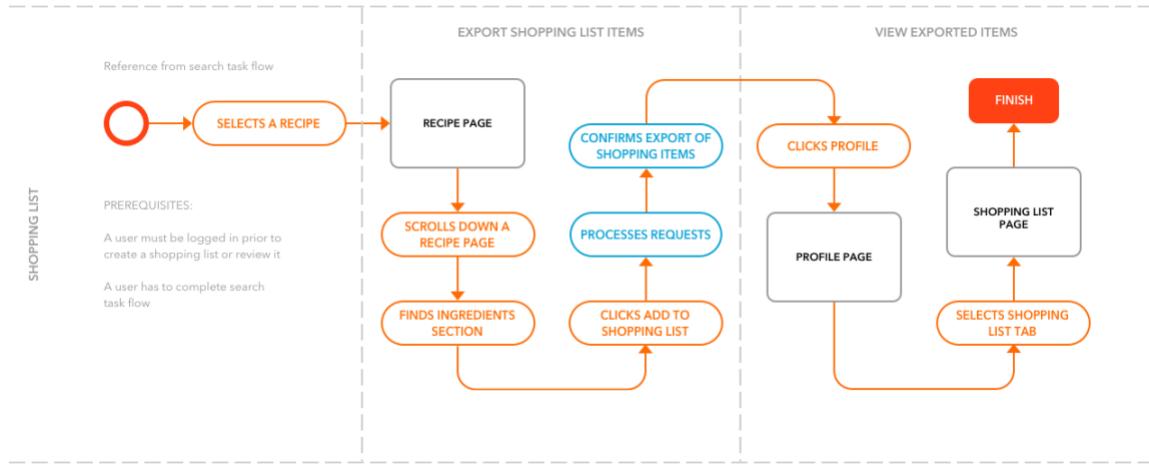


Figure 22. Use case—create and view a user’s shopping list

Scenario IV

A tech-savvy user that likes to try all features offered by the system. A user who does not afraid to write an email or notify developers about faults.

1.1.1.10 Use case—find grocery stores nearby

Prerequisites: User is logged in, and the user has at least one item on the shopping list.
Show grocery shops near the user.

1. User clicks on the profile icon
2. System shows user’s information along with related content
3. User click on the shopping list tab
4. System shows a shopping list items and the map with the search field
 - 4.1. User enters location
5. User click on the “my location” button
6. System prompts a user to give access to a user’s current location
 - 6.1. User denied access
 - 6.1.1. The system shows a message that access has been denied and suggests user to enter address manually
 - 6.2. User grants access
 - 6.2.1. System reads current location
7. System shows a list of annotations that represent grocery stores

The interaction with the map illustrated in [Fig. 23]. To use the functionality of the map, the user has to have some shopping items added to the list. The map will offer store search by the location where stores will be shown as annotations. The user can also use the geolocation to show stores nearby.

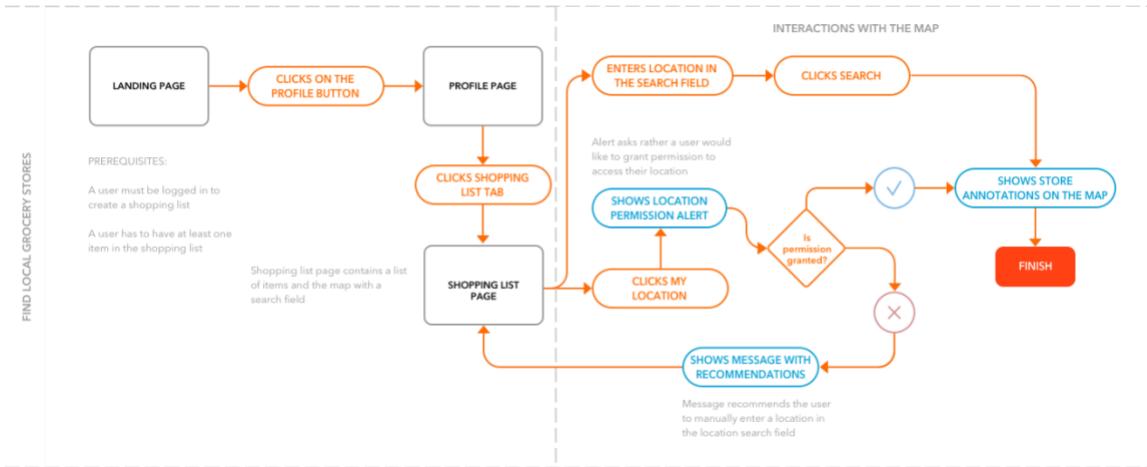


Figure 23. Use case—find local grocery stores

1.1.1.11 Use case—contact developers/support

Write a suggestion, complaint or recommendation to the support team

1. Open any page of the website
2. System shows relevant information
3. User scrolls down to the very bottom of the website
4. System shows the footer of the website
5. System shows “contact us” form
6. User enters the content of the message
 - 6.1. If the user is not registered, the system asks about an email address of the customer
 - 6.1.1. If the user enters invalid email, the system does not allow to submit the form
7. User clicks on “submit” button
8. System shows that the message has been sent successfully
 - 8.1. If the submission failed, the system asks a user to resubmit the form
 - 8.2. User prompted to step 8
9. System sends an email to the customer with the confirmation of the enquiry

The [Fig. 24] shows the task flow that describes the process of contacting support. This feature will be available throughout the web site as part of the footer.

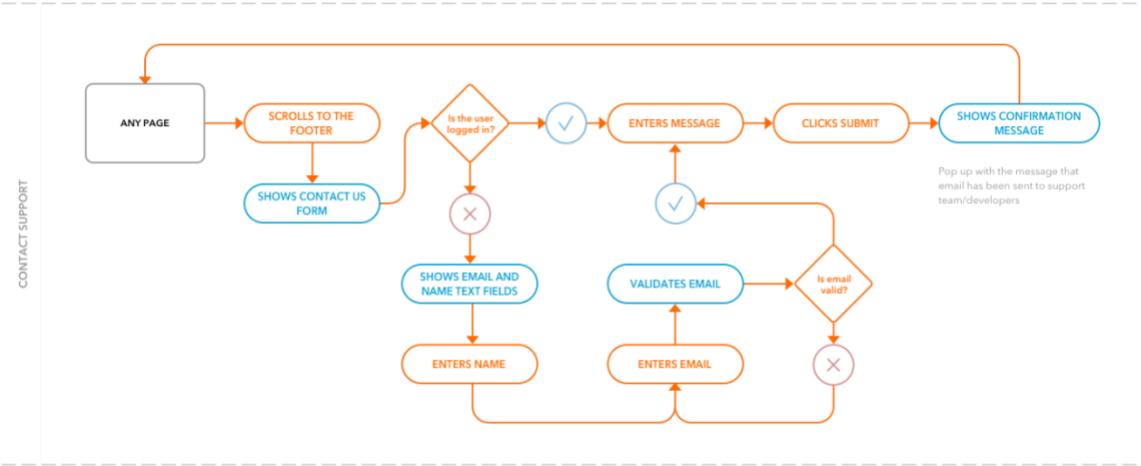


Figure 24. Use case—contact support/developers

Scenario V

You are a customer of the cooking website that decided to ask an author or community a question regarding the preparation process.

1.1.1.12 Use case—leave a comment

Leave a comment under a recipe

1. User visits cooking website homepage
2. System shows landing page with a search bar
3. User enters the name of the dish
4. User clicks “search” button
5. System shows a list of available recipes
6. User clicks on a recipe
7. System shows a recipe page
8. User scrolls down the page
9. System shows a comment section
 - 9.1. If a user is not logged in, the system shows login button
 - 9.1.1. User follow steps described in use case 4.1.2.1 ([Registration on the website](#))
10. User enters a comment text
11. User clicks “submit” button
12. System refreshes comment section and adds a new comment to the section

The commenting lets users to share information about a recipe or a blog. User has to be authenticated before leaving a comment. The task flow diagram for this task [Fig. 25] assumes that a user has completed a recipe search first. It also contains a reference to the login task flow in case a user is not authenticated.

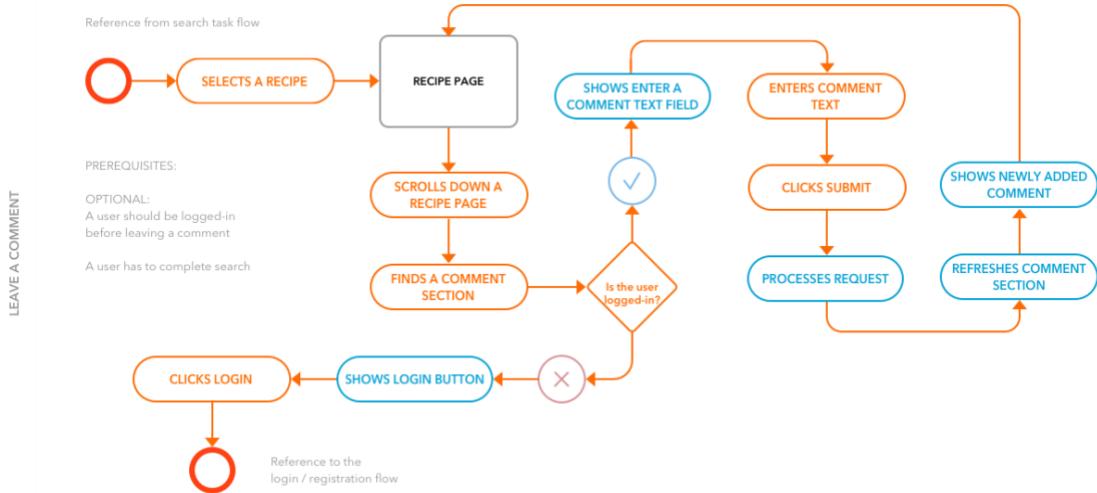


Figure 25. Use case—leave a comment under the recipe

Process and Service Workflow

To describe backend processes, we created task flows and sequence diagrams that show interactions between the client, frontend and backend of the website. For the creation of the diagrams, we used various online tools such as Lucidchart, Draw.io and others [7, 8, 9].

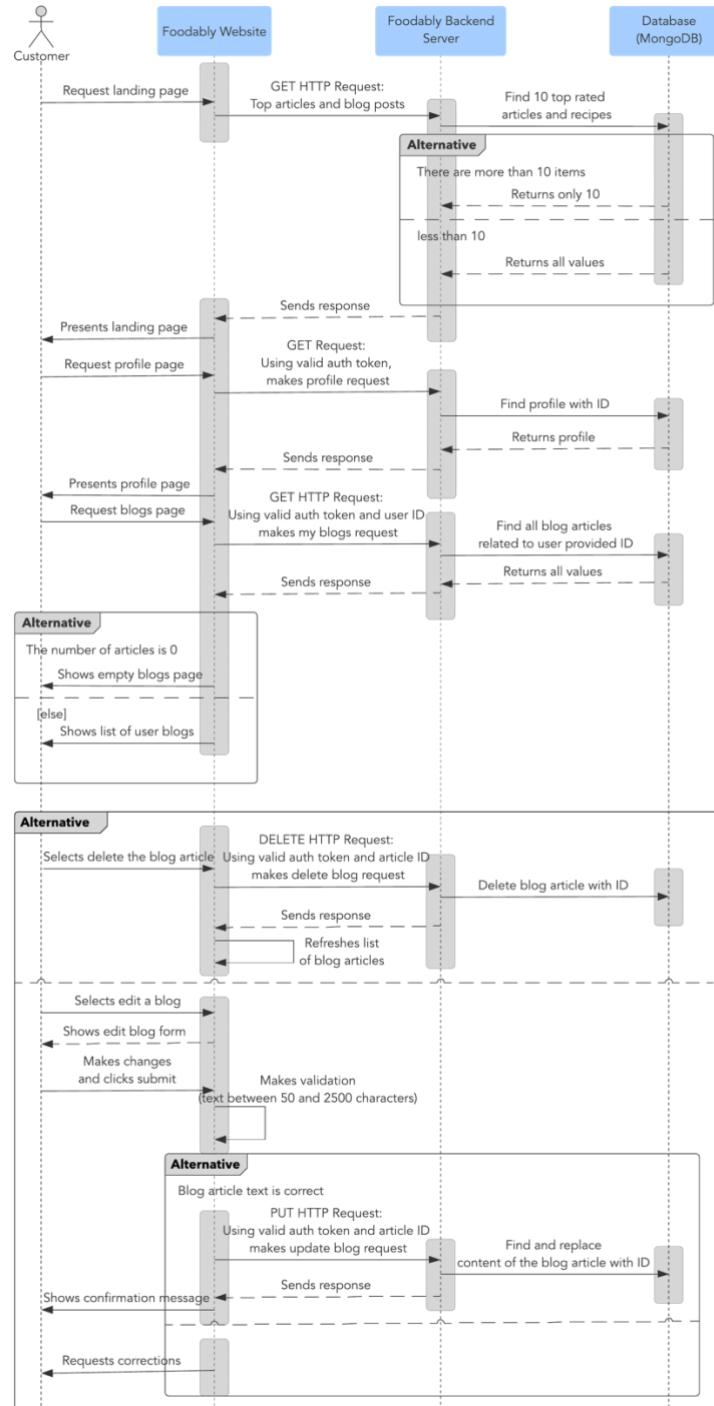
The first sequence diagram [Fig. 26] describes processes that take place during blog article management. To work on an article user has to be logged in. Since another group member will cover the profile management feature, the blog management sequence diagram starts at the point when a user has an account and already logged in to it.

To describe the next feature, which is a subscription, we created two sequence diagrams. The first one [Fig. 27] shows the process of subscription action steps. The second one [Fig. 28] describes processes that take place ones an author posts new updated and the subscribed users need to receive a notification. To notify subscribers, the backend will use the Nodemailer library with a connection to a free SMTP (Simple Mail Transfer Protocol) server [10].

Similarly, to the previous feature, the user has to be logged in before being able to subscribe to the author.

ASSUMPTION:

A user has an account and used it to log in before, so authentication information is available in cookies



The create a blog article task sequence is very similar to the edit blog article (DIFFERENCE: use POST request and insert to the database operation), so it is omitted in the diagram.

Figure 26. Sequence diagram for managing blog articles

ASSUMPTION:

A user has an account and a user used it to log in before, so authentication information is available in cookies. **A user also completed the search for the recipes.**

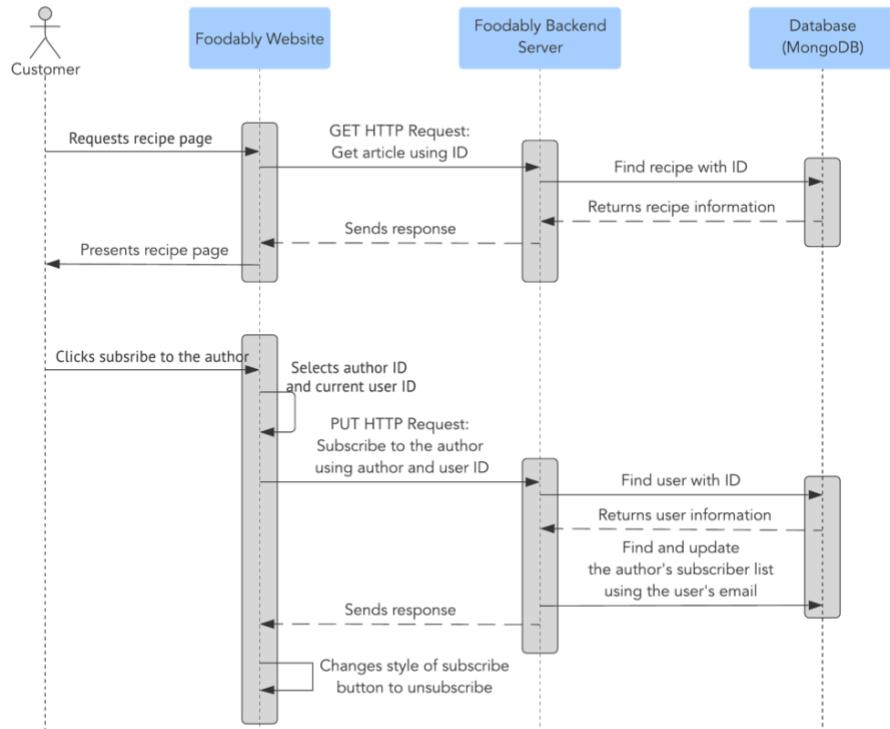


Figure 27. Sequence diagram for handling subscriptions

ASSUMPTION:

A user has an account and a user used it to log in before, so authentication information is available in cookies. **A user entered the profile page and a user is about to submit a new recipe.**

For sending updates via email the backend will use the Nodemailer library.

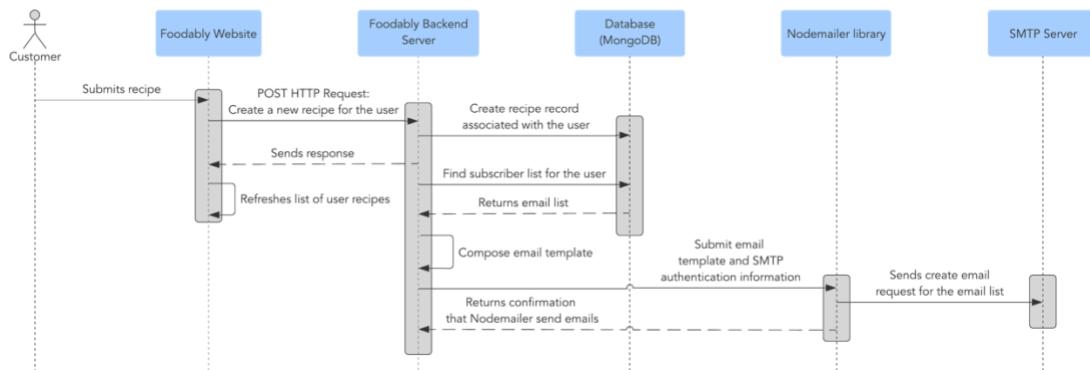


Figure 28. Sequence diagram for sending updates from the author via email

The diagram demonstrated in Fig. 29 shows a process that takes place during the management of the recipes. The diagrams describe such actions as creation, edition, deletion of the recipe [12].

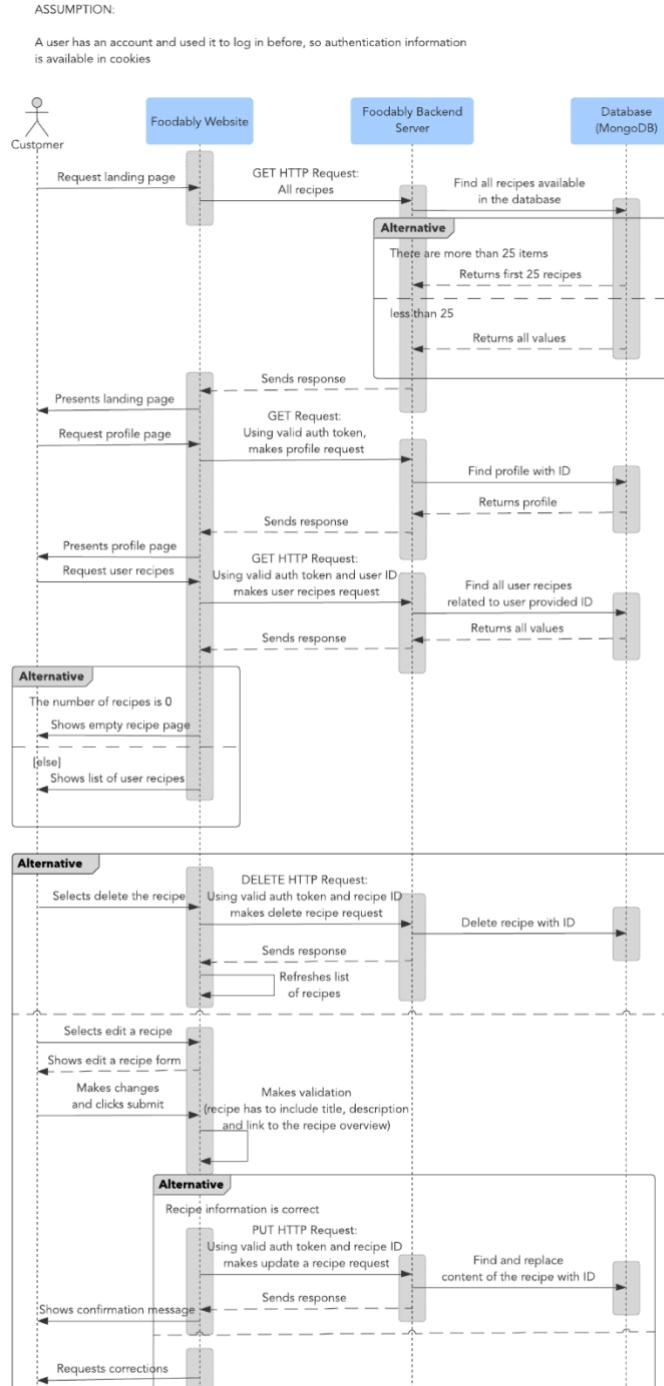


Figure 29. Sequence diagram for managing recipes

Fig. 30 describes the task flow of how a user can provide a comment. The search task flow has to be completed for the user to give a comment. The comment can be posted only if the user is logged in. If the user is not logged in, the login button appears. After the login task flow is completed, the user can continue to provide a comment to the desired recipe/blog.

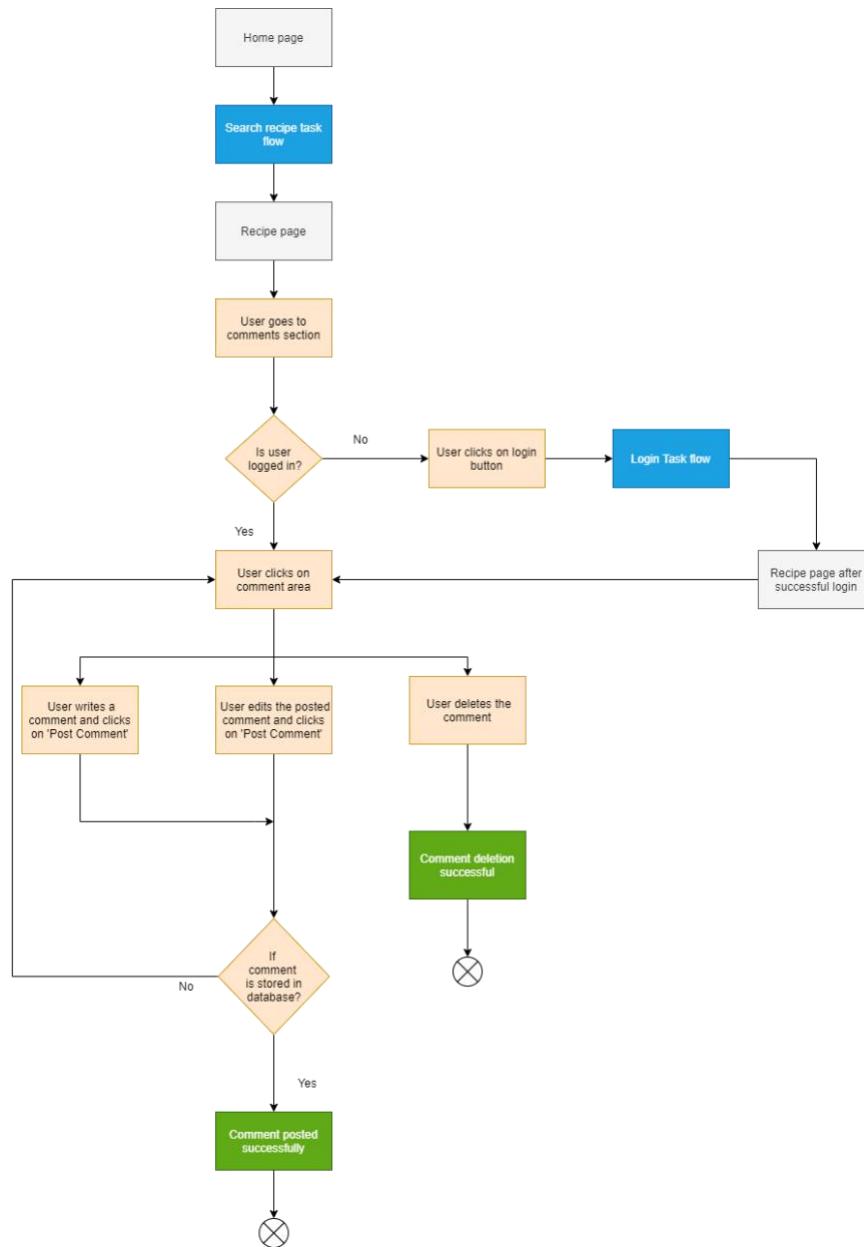


Figure 30. Task flow for posting a comment [9]

Fig. 31 describes the task flow of how a user can provide a rating. The search task flow has to be completed for the user to give a rating. The rating can be given only if the user is logged in. If the user is not logged in, the login button appears. After the login task flow is completed, the user can continue to provide ratings to the desired recipe/blog.

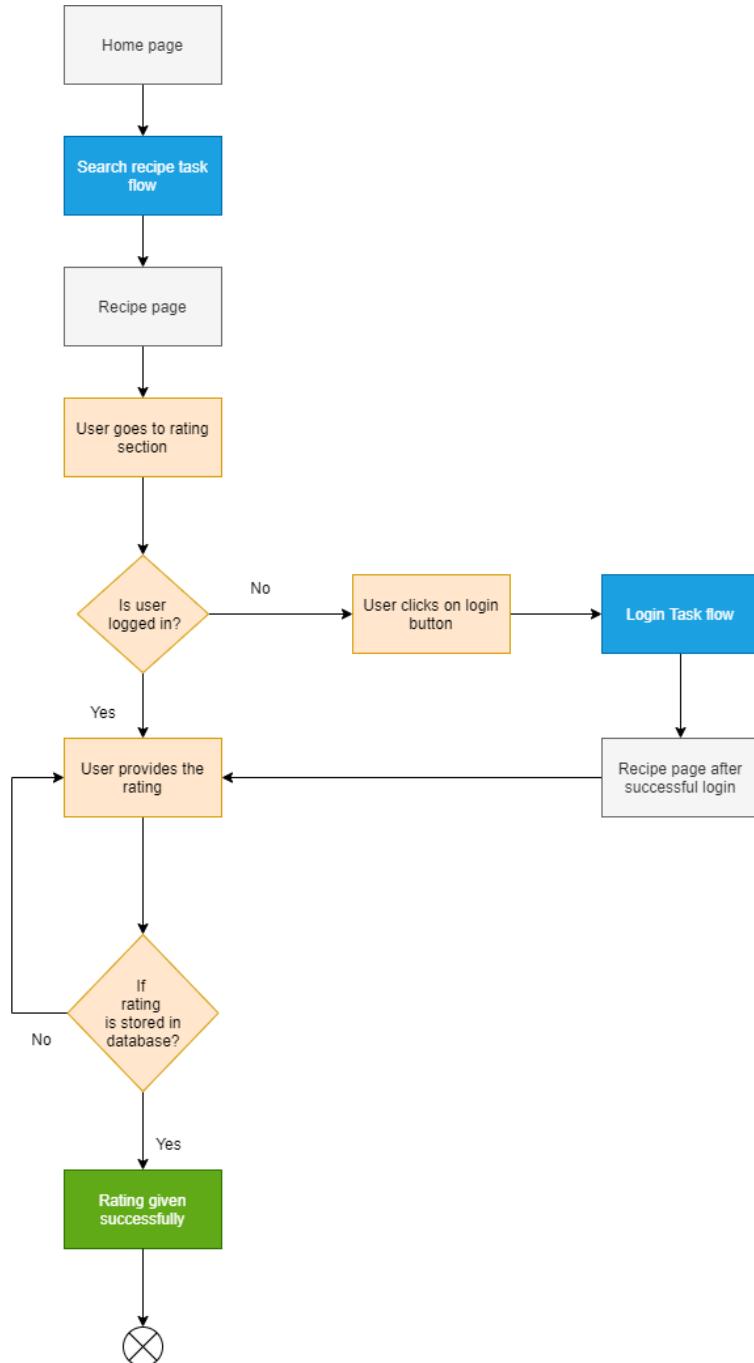


Figure 31. Task flow for providing rating [9]

Fig. 32 depicts the task flow for the search feature of Foodably. The process flow starts from the home page and with a search feature allowing to apply a filter based on user choice, thus displaying the related recipes.

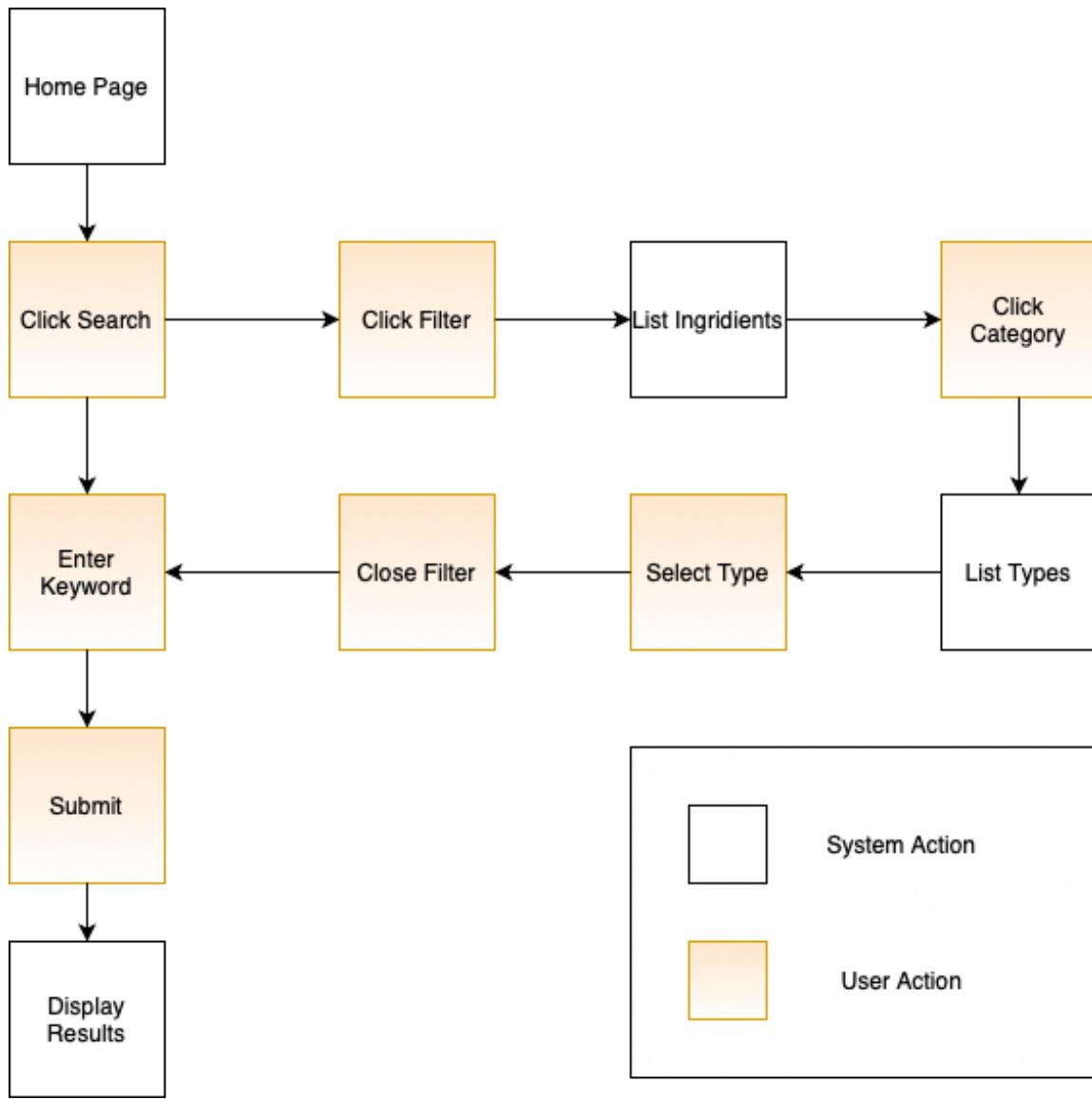


Figure 32. Task flow for searching task [9]

The following Fig. 33 depicts the task flow for the support feature of Foodably. The task flow coordinates the process for contacting the support team to address any queries.

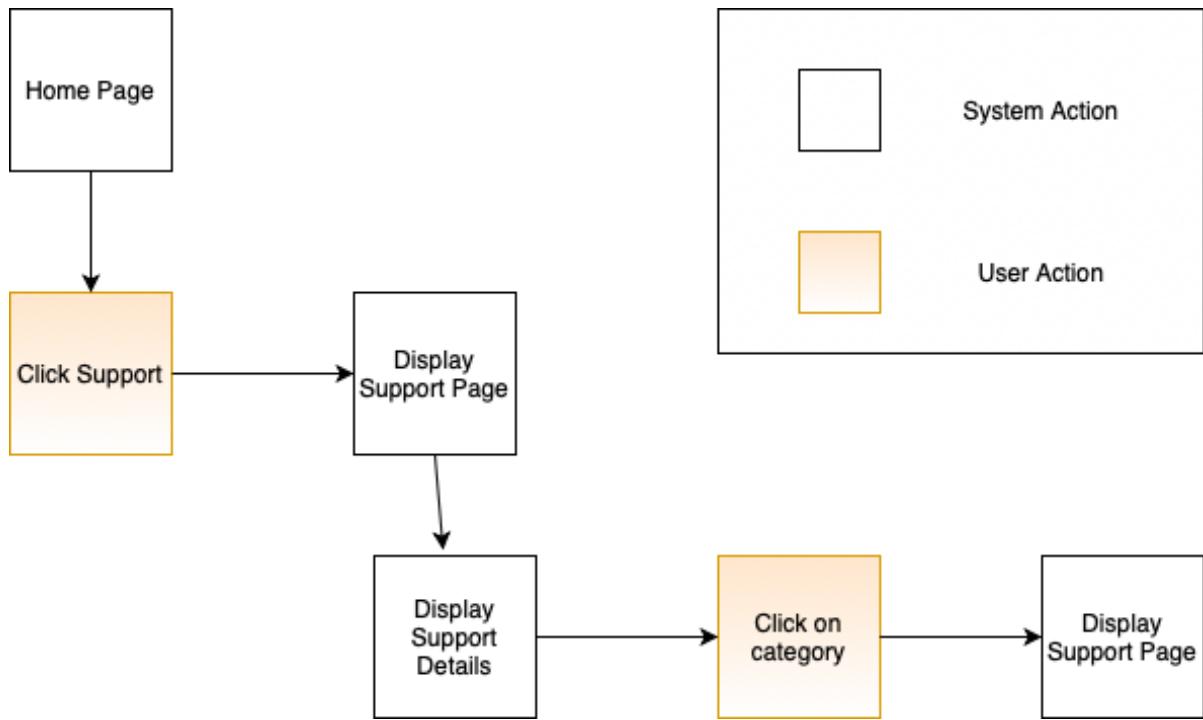


Figure 33. Task flow for support feature [9]

The Fig. 34 depicts the task flow for the shopping list feature of Foodably. The task flow diagram shows the steps to add ingredients to shopping list.

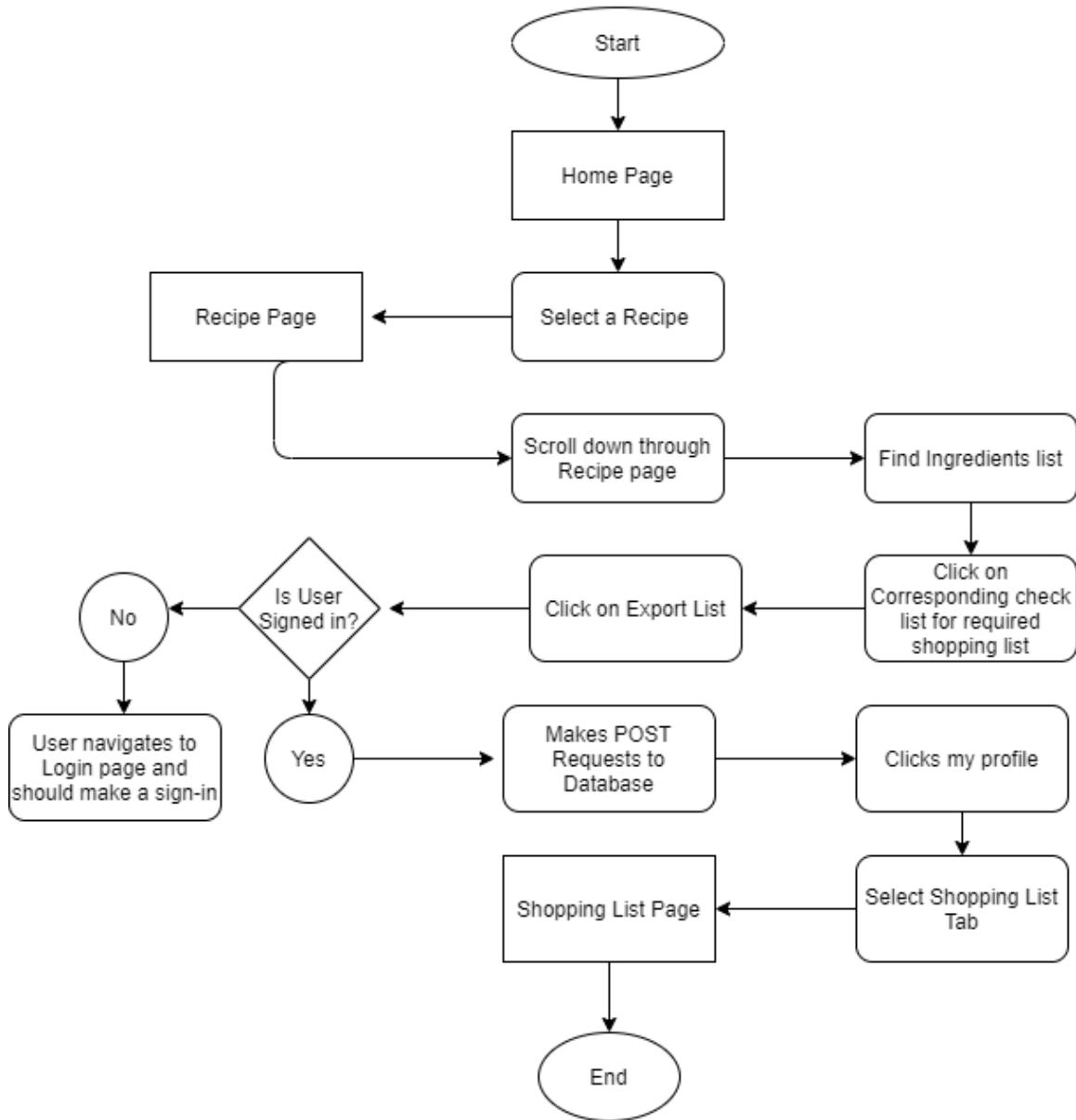


Figure 34. Task flow for shopping list feature

Fig. 35 depicts the task flow for the nearby grocery shopping list feature of Foodably. The task flow diagram shows the steps to display nearby grocery stores.

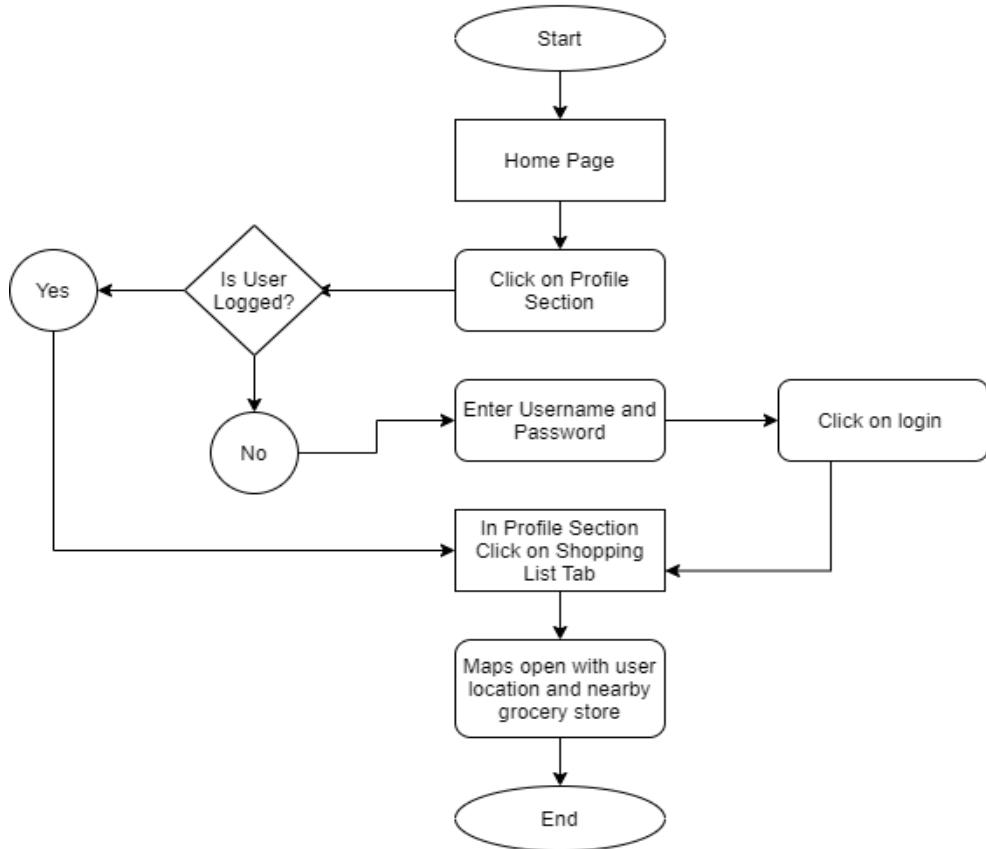


Figure 35. Task flow for nearby grocery store list

Fig. 36 briefly describes the sequence of steps carried from frontend to backend when a user posts a comment to a recipe or a blog post.

Fig. 37 briefly describes the sequence of steps carried from frontend to database when a user edits a comment which is already posted under a recipe or a blog post.

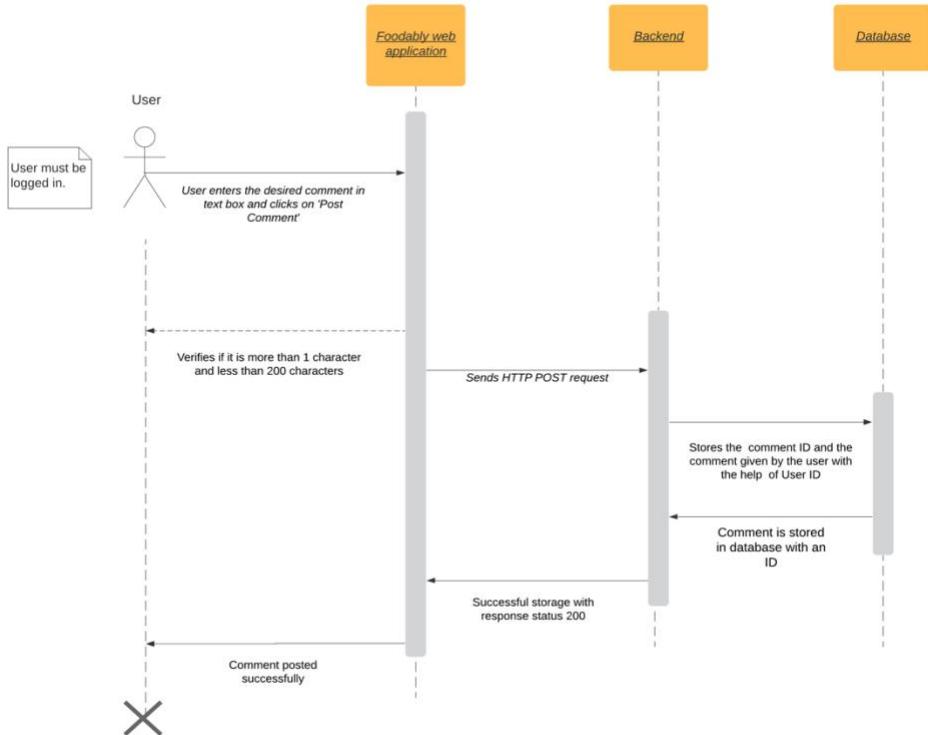


Figure 36. Sequence diagram for posting a comment [10]

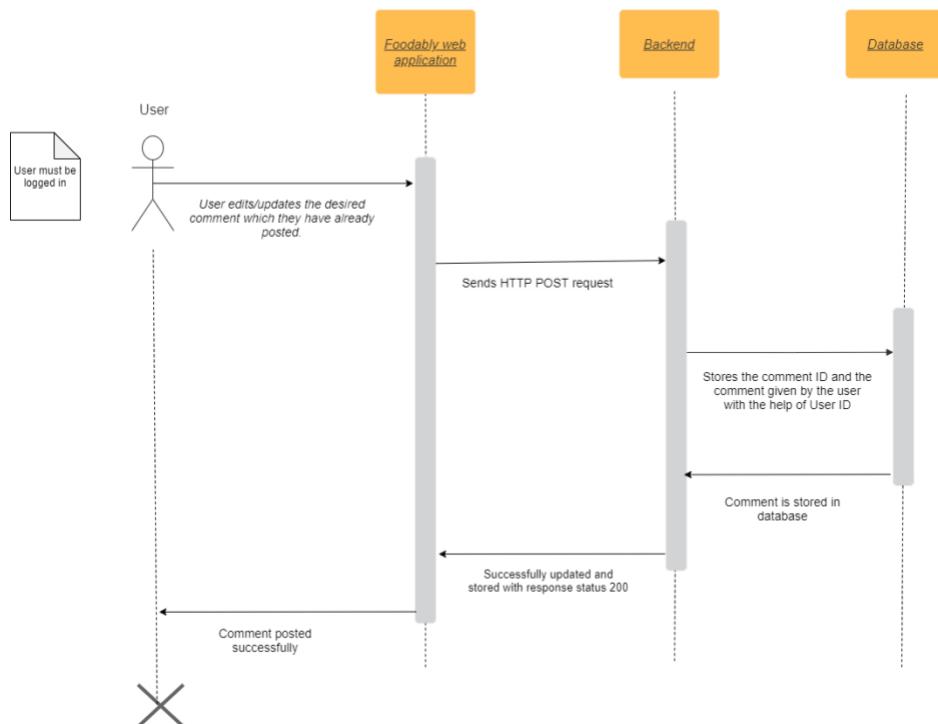


Figure 37. Sequence diagram to edit/update a comment [10]

Fig. 38 briefly describes the sequence of steps carried from frontend to database when a user deletes a comment which is already posted under a recipe or a blog post.[13]

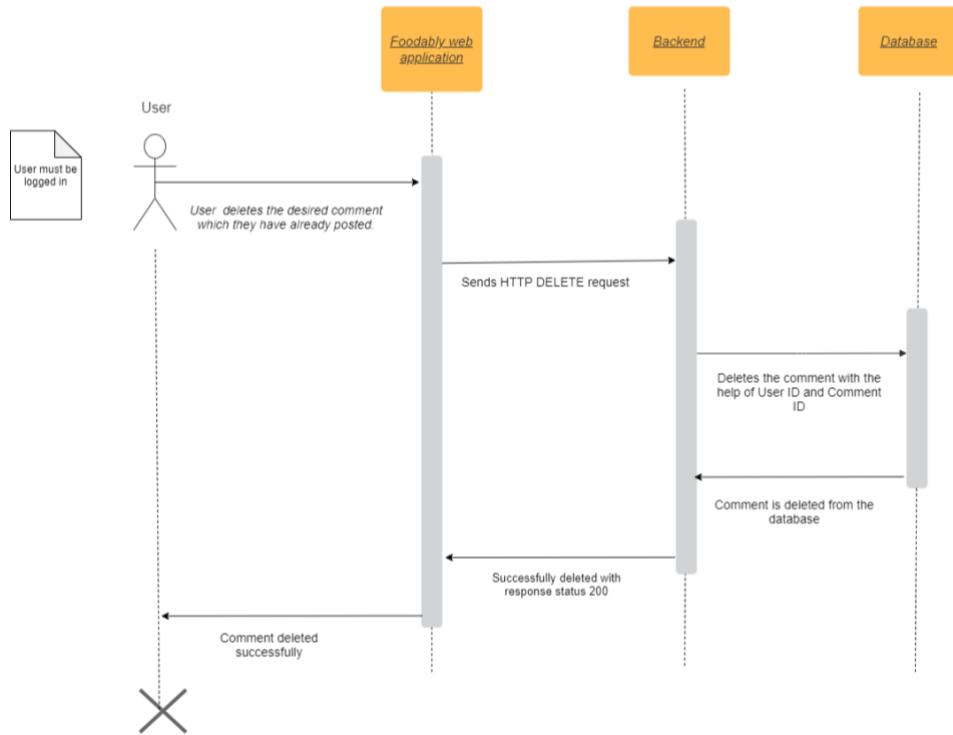


Figure 38. Sequence diagram to delete a comment [10]

Fig. 39 briefly describes the sequence of steps carried from frontend to database when a user provides a rating to a recipe.

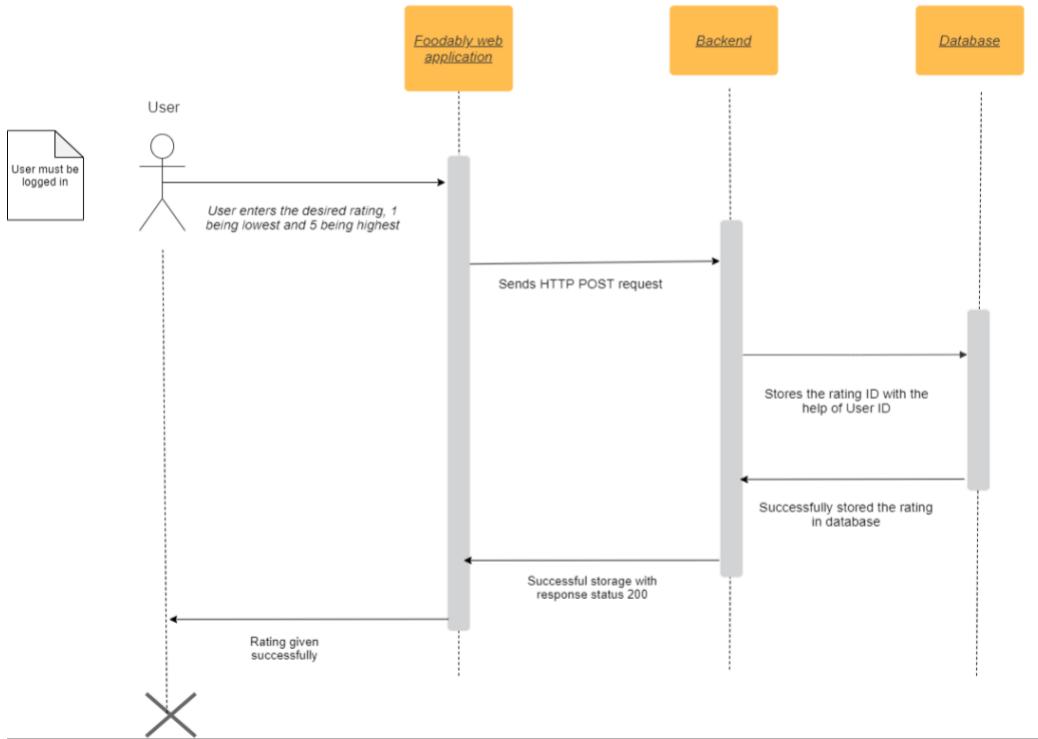


Figure 39. Sequence diagram to provide rating to a recipe [10]

Fig. 40 depicts the sequence flow of users sending a search request for recipes. Users can also apply advance filters on the search and sort the order of recipes displayed.

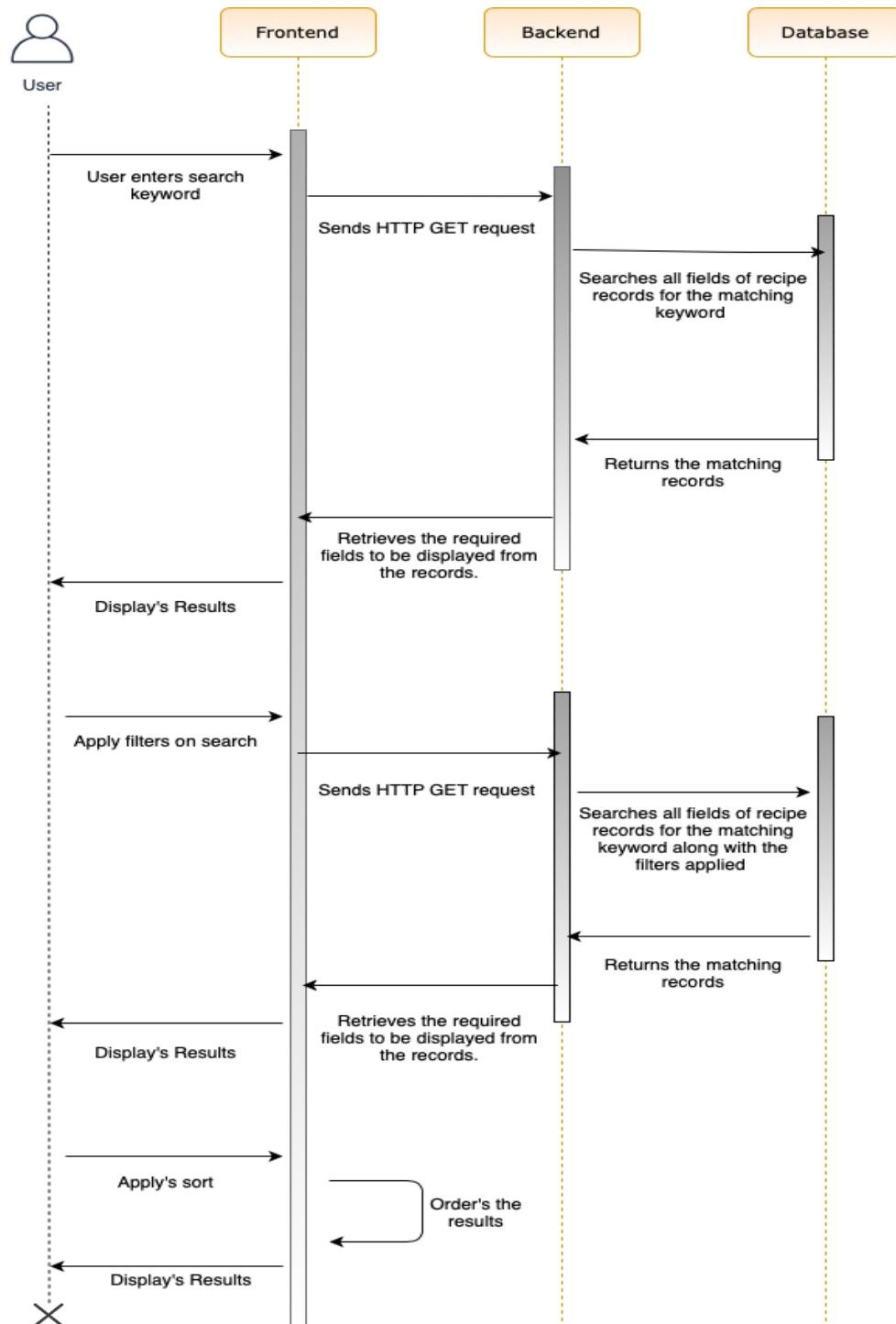


Figure 40. Sequence diagram to search [9]

Fig. 41 depicts the sequence flow of user sending details for support [15].

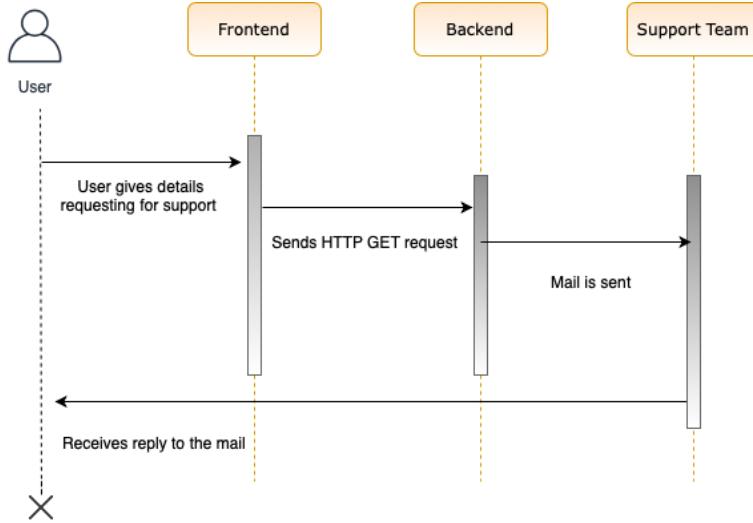


Figure 41. Sequence diagram for support feature [9]

Fig. 42 shows the sequence flow of get shopping list details from frontend to database.

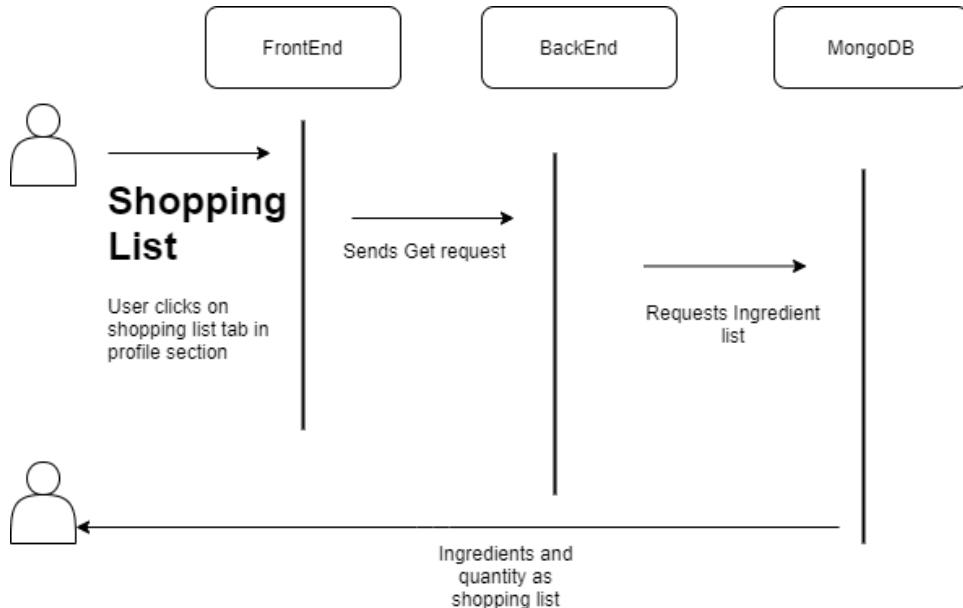


Figure 42. Sequence diagram for shopping list feature [9]

Fig. 43 shows the sequence flow of getting nearby stores in maps from frontend to calling Maps API [16].

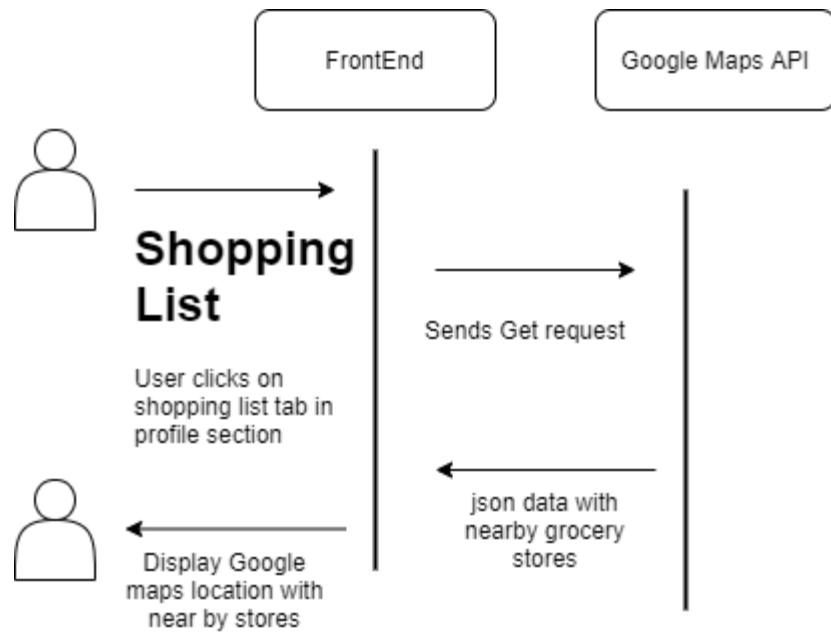


Figure 43. Sequence diagram for nearby stores feature [9]

REFERENCES

- [1] “The digital design toolkit,” Sketch. [Online]. Available: <https://www.sketch.com/>. [Accessed: 10-Apr-2020].
- [2] Kitchen Stories, “A Complete Guide to Vinegars and How to Use Them: Stories,” Kitchen Stories. [Online]. Available: <https://www.kitchenstories.com/en/stories/a-completeguide-to-vinegars-and-how-to-use-them/>. [Accessed: 10-Apr-2020].
- [3] “unDraw,” Search. [Online]. Available: <https://undraw.co/search>. [Accessed: 10-Apr-2020].
- [4] Unsplash, “Beautiful Free Images & Pictures,” Unsplash. [Online]. Available: <https://unsplash.com/>. [Accessed: 10-Apr-2020].
- [5] Apple Inc, SF Symbols - SF Symbols - Human Interface Guidelines - Apple Developer. [Online]. Available: <https://developer.apple.com/design/human-interface-guidelines/sfsymbols/overview/>. [Accessed: 10-Apr-2020].
- [6] “User Flow Diagram - Template,” Dribbble. [Online]. Available: <https://dribbble.com/shots/3019900-User-Flow-Diagram-Template>. [Accessed: 10-Apr-2020].
- [7] “Online Diagram Software & Visual Solution,” Lucidchart. [Online]. Available: <https://www.lucidchart.com/>. [Accessed: 29-Feb-2020].

- [8] "Balsamiq. Rapid, effective and fun wireframing software. | Balsamiq", *Balsamiq.com*, 2020. [Online]. Available: <https://balsamiq.com>. [Accessed: 10-Apr-2020].
- [9] "Flowchart Maker & Online Diagram Software", Draw.io, 2020. [Online]. Available: <https://www.draw.io/> [Accessed: 10-Apr-2020].
- [10] A. Reinman, Nodemailer. [Online]. Available: <https://nodemailer.com/>. [Accessed: 29Feb-2020].
- [11] "Project Proposal of Foodably", Dalhousie University. [Online document], [Accessed: 10-Apr-2020].
- [12] Mykhailo Herasimov, "CSCI:5709-Assignment- 3."Dalhousie University.[Online document]. [Accessed:10-Apr-2020]
- [13] Sneha Jayavardhini, "CSCI:5709-Assignment- 3."Dalhousie University.[Online document]. [Accessed:10-Apr-2020]
- [14] Raviteja Kase, "CSCI:5709-Assignment- 3."Dalhousie University.[Online document]. [Accessed:10-Apr-2020]
- [15] Ujwal Vikas Chanda, "CSCI:5709-Assignment- 3."Dalhousie University.[Online document]. [Accessed:10-Apr-2020]
- [16] Muppidi Shashank Reddy, "CSCI:5709-Assignment- 3."Dalhousie University.[Online document]. [Accessed:10-Apr-2020]