COMP3900 Project Proposal

MealMatch

Submitted: 21 June 2020

Team Name: **A Group Name**

Members:

Name	zID	Email	Role
Di Shun Huang	5108453	z5108453@unsw.edu.au	Frontend / Admin
Yiting Liu	5211008	z5211008@unsw.edu.au	Backend
Kevin Tran	5164322	z5164322@unsw.edu.au	Backend
Henry Pan	5160601	z5160601@unsw.edu.au	Full stack support

Table of Contents

Background

Problem Statement

The accessibility of online resources is changing the way people learn how to cook. With the new fast-paced culture that the new generation is growing into, people do not have the time to go to cooking lessons but want a quick and healthy meal. Rather than reading from a cookbook or taking a cooking lesson, people want the recipes from a simple Google Search. More and more so, chefs, bloggers or the general public already post their recipes on the internet in which everyone can access. What people want is quick, and accessible resources.

Currently, what's widely available are recipes that people can Google Search. The main problem is that generally people will find that they are missing several ingredients. A one-stop-go-to, meal-matching platform will provide people with flexibility and accessibility of cooking recipes in which they already have the ingredients for. By allowing the users to select ingredients that they already have, the platform will be able to suggest recipes (along with the filters such meal-type, and available cooking equipment).

Existing User Experience Issues

Currently, there already exists similar platforms that provide recipes for users, but there are certain drawbacks that can be improved upon. From researching the websites www.taste.com.au, www.delish.com, https://www.supercook.com/, these websites have similar platforms in that they all recommend recipes. We can see from the existing systems, the drawbacks:

- www.supercook.com/ The website allows entering ingredients by checkbox which is preferred, but when typing ingredients - if it is misspelt or the ingredient is not registered in their database, the user will just be confused as the ingredient did not go through. It should have an autocomplete, or list out possible ingredients (Dropbox) for users to select when typing.
- https://www.delish.com/ and www.taste.com.au These websites do not have a search by ingredient, and results in the original problem, where users will not have the necessary ingredients when browsing recipes.
- www.taste.com.au and www.delish.com These websites do not support searching by category.
- www.supercook.com/ The website lists out the ingredients when you are on the 'type' section, but if users choose to use the checkbox to select ingredients, they may forget what ingredients they chose. In our system, we can have a list of selected ingredients which is always visible to the user, so they can easily deselect or view their selected ingredients.

- www.supercook.com/ The webpage doesn’t show the list of ingredients selected if you click categories. It should always show a list of selected ingredients so the user knows what ingredients he has selected, and is able to deselect it if he made an error.
- www.supercook.com/ does not list ingredient amounts or cooking utensils required and can be improved by having these filters in our system.

User Stories and Sprints

We divide our target users into three groups, users, recipe explorers and recipe contributors. Both recipe explorers and contributors share the same user stories and thus they are the extension of the common users. However, any recipe explorer can also be a contributor at the same time.

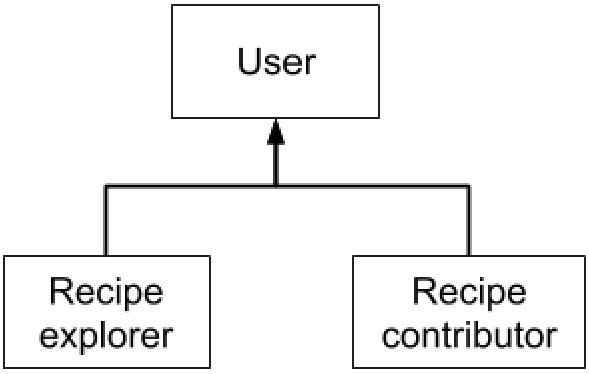


Figure 1 User Type Hierarchy

User Stories

As a user, I want

1. to be able to register an account so that I can have my data persisted for future usage.
2. to be able to log in so that I can recover any stored data on the app.
3. to be able to login with a secret key so that the only person who can access my data is myself.
4. myself to be kept in login state unless I explicitly log out so that minimal time is consumed on the security-related issue
5. to be able to change my password so that I can minimise the chance of my account being stolen

As a recipe explorer, I want

1. to be able to select ingredients from a series of ingredient categories so that I can find the recipes which are made with my favourite categories.
2. to be able to search a recipe by ingredients with detailed quantities so that I can make sure I have enough materials to cook.
3. to be able to directly input my ingredient list so that it will take less time for me to find the recipes I need.
4. to be able to get some suggestions about the next ingredients while I am inputting ingredients so that the process of searching will be much faster.
5. to be able to review all the ingredients I've picked so that I can make sure I didn't pick any ingredients by mistake.
6. to be able to search for all the available recipes with my ingredient list so that I will know which dishes are suitable for the ingredients I have.
7. to be able to see the full details of the recipe I've selected so that I will know what materials I need to buy and how to cook the dishes.
8. to be able to sort the recipes by meal-type so that some of the types that I don't want can be excluded.
9. to be able to filter recipes by utensils required so that the recipes shown to me must be followable with the utensils I have.
10. to be able to filter recipes by cooking time so that I can make sure that I have enough time to do the cooking.

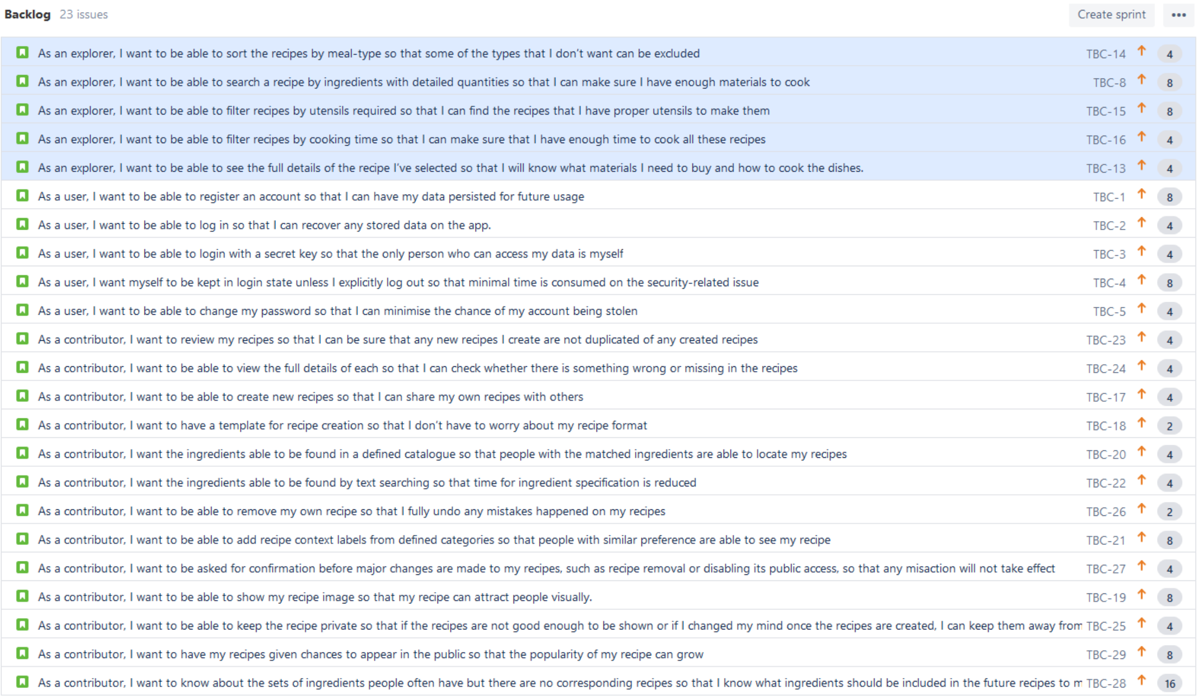
As a recipe contributor, I want

1. to be able to create new recipes so that I can share my own recipes with others.

2. to have a template for recipe creation so that I do not have to worry about my recipe format.
3. to be able to show my recipe image so that my recipe can attract people visually.
4. the ingredients able to be found in a defined catalogue so that people with the matched ingredients are able to locate my recipes.
5. the ingredients able to be found by text searching so that time for ingredient specification is reduced.
6. to be able to add recipe context labels from defined categories so that people with similar preference are able to see my recipe.
7. to review my recipes so that I can be sure that any new recipes I create are not duplicated of any created recipes.
8. to be able to view the full details of each so that I can check whether there is something wrong or missing in the recipes.
9. to be able to keep the recipe private so that if the recipes are not good enough to be shown or if I changed my mind once the recipes are created, I can keep them away from the public.
10. to be able to remove my own recipe so that I fully undo any mistakes happened on my recipes.
11. to be asked for confirmation before major changes are made to my recipes, such as recipe removal or disabling its public access, so that any misaction will not take effect.
12. to know about the sets of ingredients people often have but there are no corresponding recipes so that I know what ingredients should be included in the future recipes to make them popular.
13. to have my recipes given chances to appear in the public so that the popularity of my recipe can grow.

Sprints

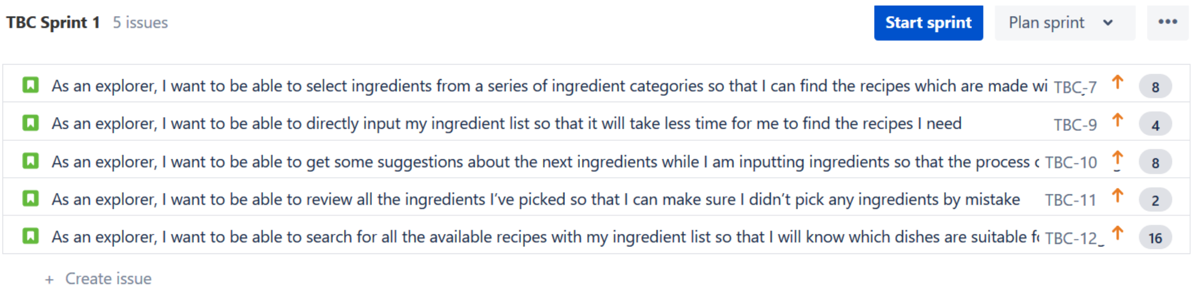
Sprints are separated into weekly manner so that the productivity can be more accurately estimated, and we can take advantage of the Tuesdays tutorial session for any discussion with various timespan. The implementation is expected to start from Monday Week 4 and ends on Friday Week 9. There are in total 6 weekly sprints, except that the last sprint will last for 5 days. Tasks are concentrated in the earlier sprints than later ones so that the measurement of our sprint velocity is accurate and tasks are more likely to be completed on time due to varying workload of coursework in other courses throughout the project timeline than having the tasks aggregated until the end of the timeline. The user stories have been set up on our Jira team as shown in Figure 2and Figure 3.



Backlog 23 issues

As an explorer, I want to be able to sort the recipes by meal-type so that some of the types that I don't want can be excluded	TBC-14	↑	4
As an explorer, I want to be able to search a recipe by ingredients with detailed quantities so that I can make sure I have enough materials to cook	TBC-8	↑	8
As an explorer, I want to be able to filter recipes by utensils required so that I can find the recipes that I have proper utensils to make them	TBC-15	↑	8
As an explorer, I want to be able to filter recipes by cooking time so that I can make sure that I have enough time to cook all these recipes	TBC-16	↑	4
As an explorer, I want to be able to see the full details of the recipe I've selected so that I will know what materials I need to buy and how to cook the dishes.	TBC-13	↑	4
As a user, I want to be able to register an account so that I can have my data persisted for future usage	TBC-1	↑	8
As a user, I want to be able to log in so that I can recover any stored data on the app.	TBC-2	↑	4
As a user, I want to be able to login with a secret key so that the only person who can access my data is myself	TBC-3	↑	4
As a user, I want myself to be kept in login state unless I explicitly log out so that minimal time is consumed on the security-related issue	TBC-4	↑	8
As a user, I want to be able to change my password so that I can minimise the chance of my account being stolen	TBC-5	↑	4
As a contributor, I want to review my recipes so that I can be sure that any new recipes I create are not duplicated of any created recipes	TBC-23	↑	4
As a contributor, I want to be able to view the full details of each so that I can check whether there is something wrong or missing in the recipes	TBC-24	↑	4
As a contributor, I want to be able to create new recipes so that I can share my own recipes with others	TBC-17	↑	4
As a contributor, I want to have a template for recipe creation so that I don't have to worry about my recipe format	TBC-18	↑	2
As a contributor, I want the ingredients able to be found in a defined catalogue so that people with the matched ingredients are able to locate my recipes	TBC-20	↑	4
As a contributor, I want the ingredients able to be found by text searching so that time for ingredient specification is reduced	TBC-22	↑	4
As a contributor, I want to be able to remove my own recipe so that I fully undo any mistakes happened on my recipes	TBC-26	↑	2
As a contributor, I want to be able to add recipe context labels from defined categories so that people with similar preference are able to see my recipe	TBC-21	↑	8
As a contributor, I want to be asked for confirmation before major changes are made to my recipes, such as recipe removal or disabling its public access, so that any misaction will not take effect	TBC-27	↑	4
As a contributor, I want to be able to show my recipe image so that my recipe can attract people visually.	TBC-19	↑	8
As a contributor, I want to be able to keep the recipe private so that if the recipes are not good enough to be shown or if I changed my mind once the recipes are created, I can keep them away from	TBC-25	↑	4
As a contributor, I want to have my recipes given chances to appear in the public so that the popularity of my recipe can grow	TBC-29	↑	8
As a contributor, I want to know about the sets of ingredients people often have but there are no corresponding recipes so that I know what ingredients should be included in the future recipes to m	TBC-28	↑	16

Figure 2 Jira Backlog (Without tasks allocated to Sprint 1)



TBC Sprint 1 5 issues

As an explorer, I want to be able to select ingredients from a series of ingredient categories so that I can find the recipes which are made wi	TBC-7	↑	8
As an explorer, I want to be able to directly input my ingredient list so that it will take less time for me to find the recipes I need	TBC-9	↑	4
As an explorer, I want to be able to get some suggestions about the next ingredients while I am inputting ingredients so that the process c	TBC-10	↑	8
As an explorer, I want to be able to review all the ingredients I've picked so that I can make sure I didn't pick any ingredients by mistake	TBC-11	↑	2
As an explorer, I want to be able to search for all the available recipes with my ingredient list so that I will know which dishes are suitable fo	TBC-12	↑	16

Figure 3 Jira First Sprint

Link to Project Objectives

As the project objective describes the main features of explorers and contributors, we create relevant user stories in recipe explorer and recipe contributor sections above. In project objectives, it mentions that recipe explorers can “ *directly input a running list of their available ingredients*”, “*locate the ingredient(s) to input through a list of ingredient categories*” and “*look through the full details of matching recipes*”, which are corresponding to our the first four user stories in recipe explorer section. Additionally, it also states that our system must include a “*next ingredient*” suggestion system and a recipe sorting system, which are defined in our 5th and the last three user stories in the explorer section.

For the recipe contributors, the project objective states that “*contributors must be able to contribute complete recipe details*”. To achieve this objective, we defined a series of user stories (e.g. the first seven user stories in the contributor section) to describe all the process of creating a new recipe on the contributor’s view. Besides, implementing an ingredient categories suggestion system and a lack-of-recipe notification system are also included in the project objective. To satisfy them, we define the fourth and the ninth user stories in the contributor’s part, in which we talk about how these systems work and how they benefit the contributors.

Beyond Existing Systems

Compared with existing systems, our system has some different features. Unlike other existing systems, we allow explorers to search for recipes through ingredients with their quantities, which is defined in the second user story in the recipe explorer section above. Through this feature, explorers can get more accurate recipe recommendations than existing systems. Except this, we also allow explorers to filter the recipes by utensils, which is not allowed by other recipe websites. In this way, our explorers will find the recipes they have enough materials to cook. We also defined this feature in the last user story of the explorer section above.

Interface and Flow Diagrams

The user interface is divided into three major components, which are user profile, recipe exploration and recipe creation. The initial page displayed to any user when they access the app initially is the explorer main page as most users are expected to be explorers. From there, the user can access any other functions of the app via the links in the app bar. The app bar is attached to every non-popup page so that the user can switch to another functionality instantly with ease.

Currently, only the access for the profile or the recipe creation pages requires user login. If the users are not in login state, they are redirected to the login page (or the register page if they choose to register new accounts) Once they have login-ed (or registered), they are led to the page of the respective functionality and subsequent access to these pages are guaranteed to be exempted from manual login (or register) as long as they have not logout or close the browser.

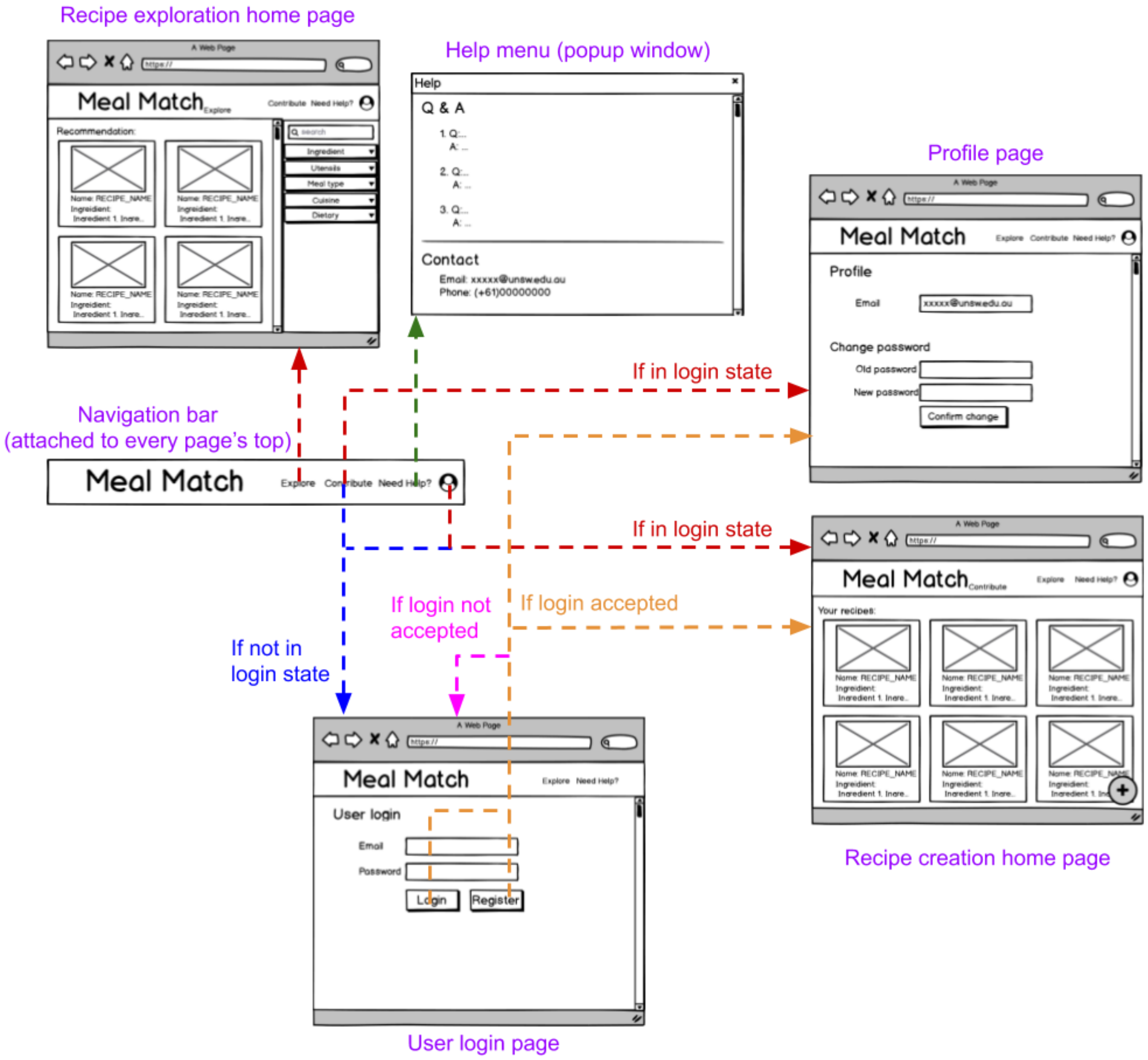


Figure 4 Interface Flow Chart: Navigation between Different Submodules

In the recipe exploration home page, since the users have not specified any parameters regarding their anticipated recipes, the page is filled with recommended recipes which can be possibly based on their popularity, novelty (period since its creation) and suitability to the user, but leaves a side panel for the user to provide their preference and constraint details to the system by selecting the specific labels.

There are two distinctive approaches, one is to look into the hierarchical list of labels grouped by the categories they belong to. The user can search for the anticipated label by repetitively selecting the category block, to which the label is expected to belong, in the list of one deeper level at a time until the respective label block is reached. Visually, whenever the category block is clicked, a list (one deeper level) containing its blocks of its subcategories and labels is expanded below the category block.

Another approach, which is more user friendly, is to search by entering keywords in the search box. Whenever the keyword is updated, the search box will provide suggestions of keyword-matching labels. The selection of labels are then identical to the first approach, except the list contains only the labels satisfying the keywords.

After a label has been selected, the respective label block will be attached to the exploration page. If additional setting is required for the selected label, such as the quantity of the ingredient the users possess, the label must go through configuration in a popup window before its attachment. All labels can be removed simply by clicking on the respective label blocks.

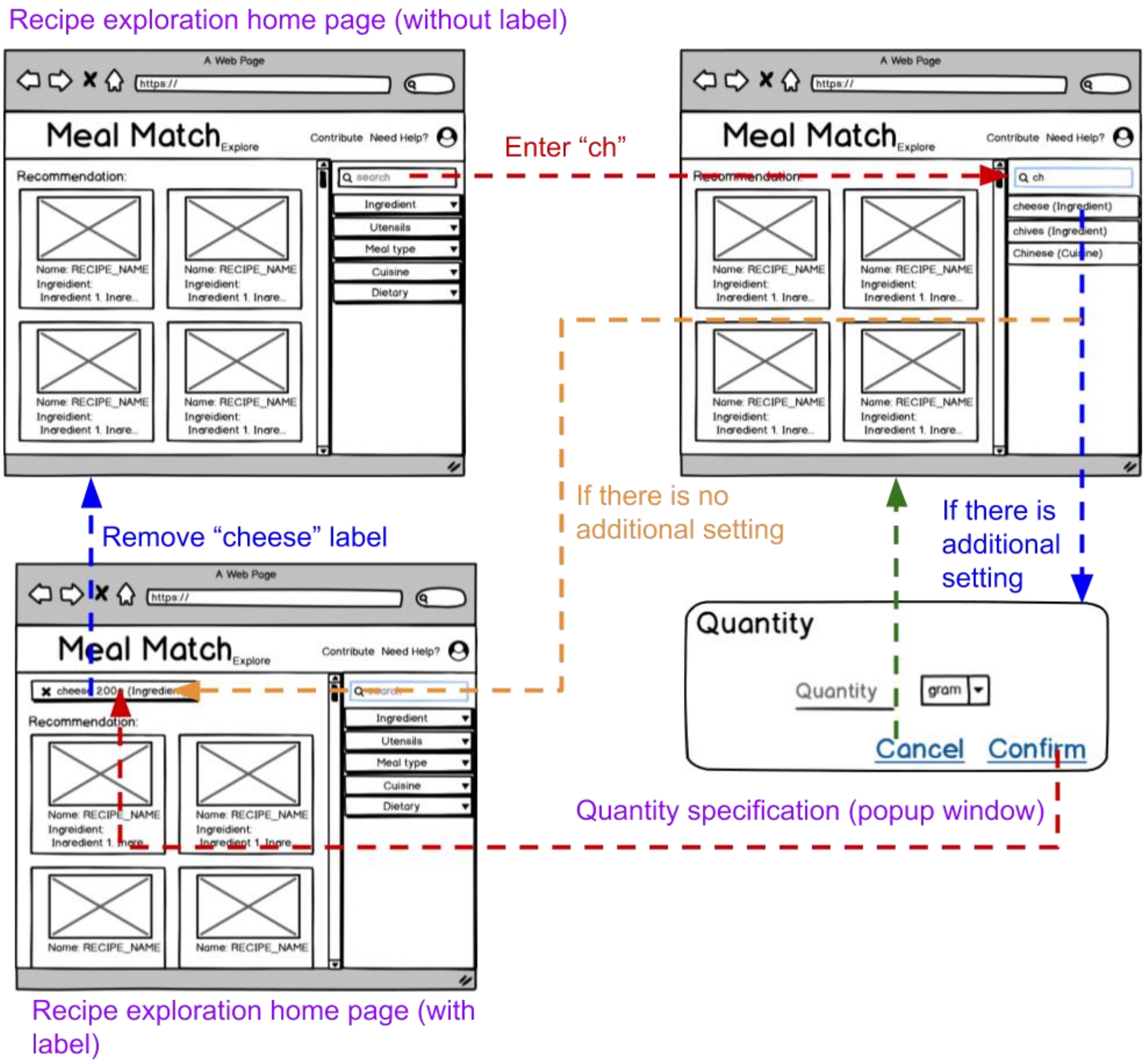


Figure 5 Interface Flow Chart: Recipe Exploration Homepage Label Appending Example

Every time the user selects a new label, the recipe list panel is automatically replaced with the recipes fitting all the labels the user chooses. The user can scroll along the recipe list panel to see the recipes hidden due to the display size limit and view the details of the target recipe by single clicking on it. The user can return to the recipe list page by simply clicking the "return" symbol.

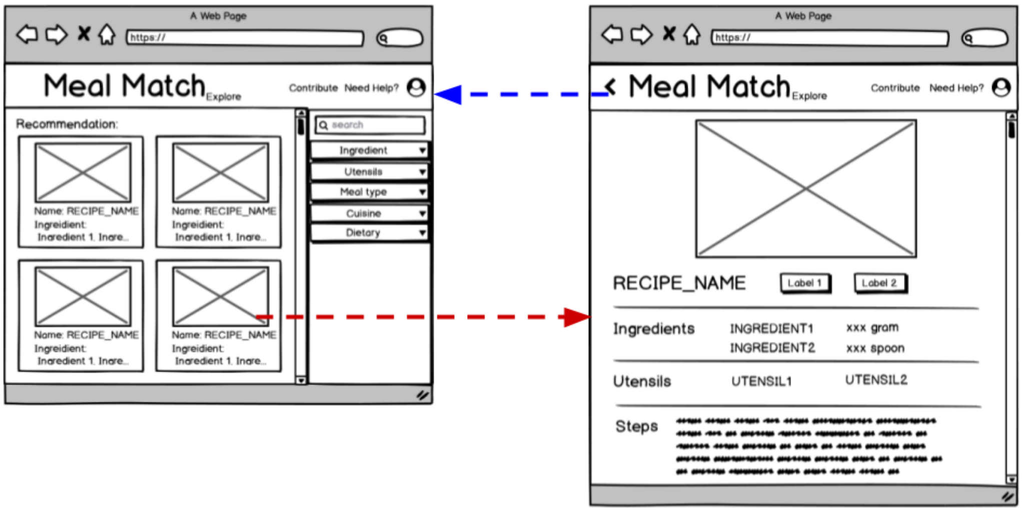


Figure 6 Interface Flow Diagram: Recipe Exploration Navigation

On the recipe creation home page, the interface is almost equivalent to that of the recipe exploration homepage to minimise the user learning curve if the recipe contributor is also a recipe explorer. The only differences are, for the recipe list panel, there are “reading” symbol indicating they are publicly accessible and an “add” floating button as the channel for new recipe creation, and for the recipe details page, there are “edit” and “dispose” floating buttons for editing and disposal of the corresponding recipes respectively, as well as a text display indicating whether the recipe is publicly accessible.

When the “add” button (or the “edit button”) is clicked, it leads the user to the recipe edit page connected to a new empty recipe (or the corresponding recipe). When the user adds any label, a search column identical to the one in the recipe exploration page is shown to the user in a popup window for the user to search and select the labels in the same manner. Every label can be edited, if the “edit” buttons next to them exist, or removed, if the user clicks on the corresponding buttons next to the label. The image can be changed by clicking its “edit” button at its corner, which will pop up an image selection window for the user to upload the photo related to the recipes. The user can click the “reset” button to undo all the changes to the last saved version, and the “tick” button to save the recipe, or the “return” button to exit to the previous page without saving. Every time the user changes the public access permission, exits edit with modification, reset the edited content or delete a recipe, a confirmation popup window is shown before the action is taken place to avoid unanticipated changes.

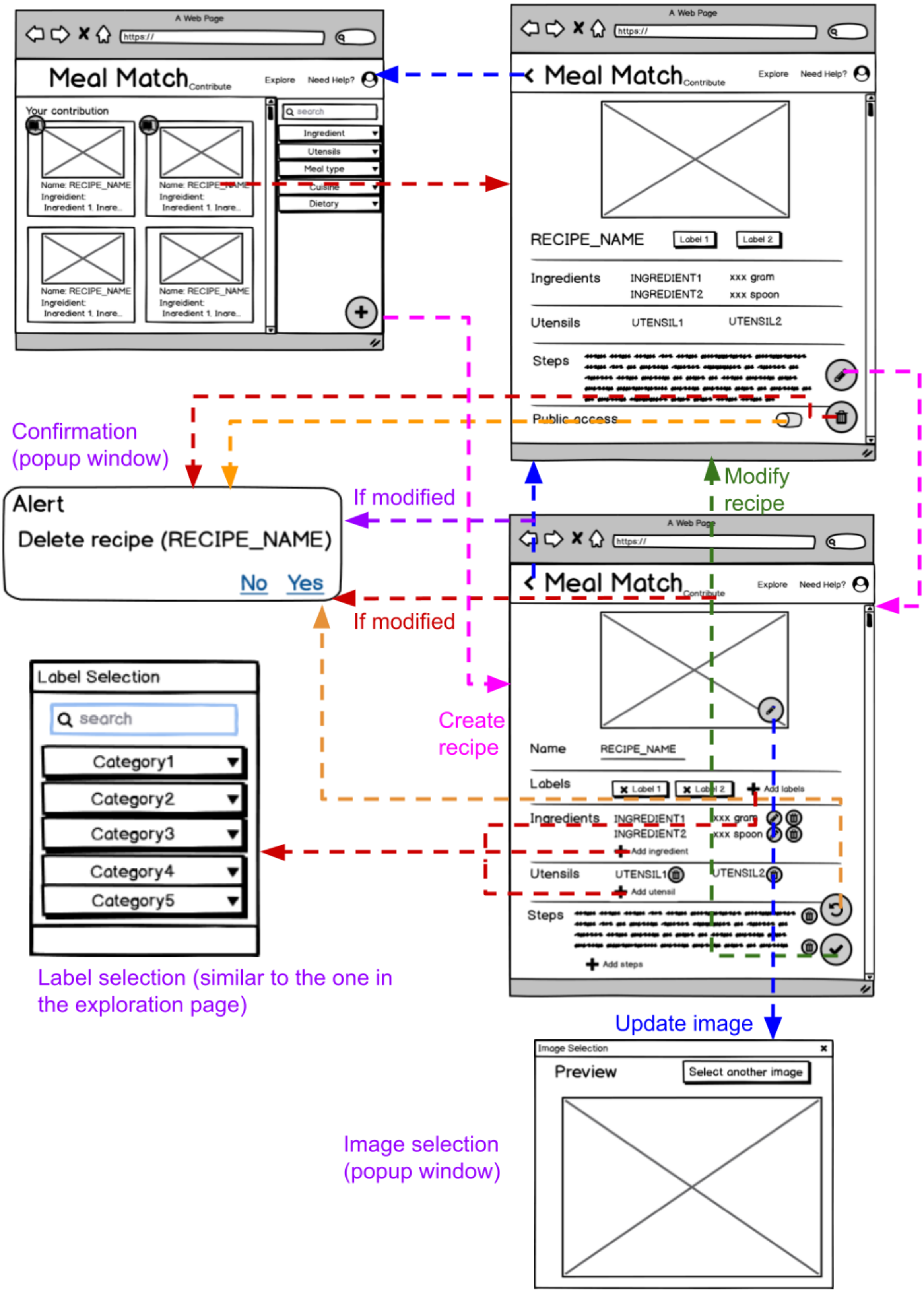


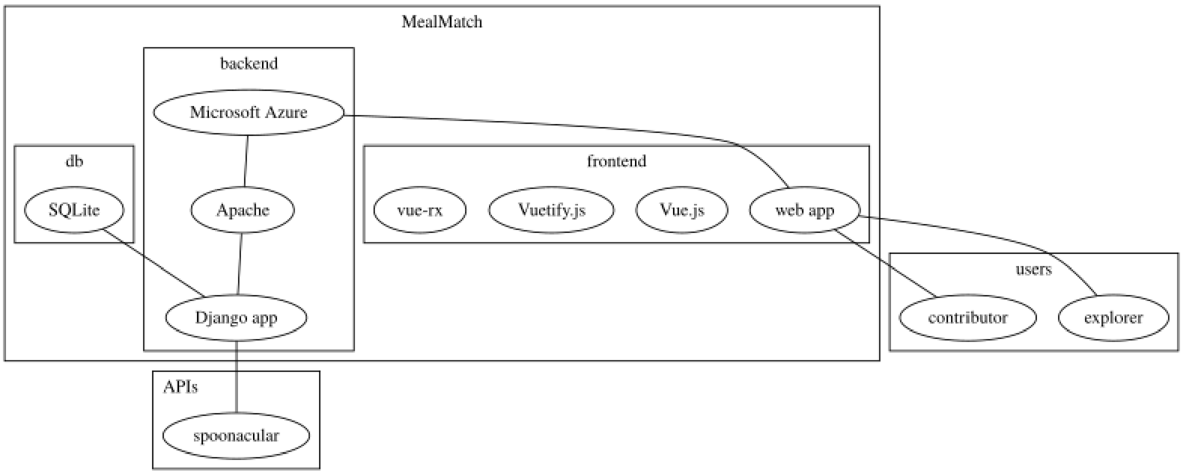
Figure 7 Interface Flow Diagram: Recipe Creation Navigation and Editing Procedure

System Architecture

For the backend, we use Django [1] to route requests from the web browser to an Apache [2] server, and use REST to make http requests and handle Create, Read, Update and Delete (CRUD) operations. We will be using SQLite [3] as the database to store all the data. The web application will be hosted on Microsoft Azure [4]. Python was chosen due to general familiarity with the language within the whole group.

For the frontend, on top of the three major components (HTML, CSS and JavaScript), Vue.js [5] will be the backbone framework because it is lightweight enough for newcomers to grasp the development paradigm quickly and it works in tandem with Django with minimal extra configuration, but complex business logic can still be implemented with minimal effort. For presentation, Vuetify [6] is chosen as it provides highly configured material design components which can produce professional appearance on any devices suitable for browsing with minimal effort. For business logic control, vue-rx [7] is used as it enables necessary logic control for this system, specifically for searching, which is not provided by the official state management component, vuex [8].

We will use Github to host all our code, pushing our updates to our own branches before merging with the master branch in order to always maintain working code in master, and solve conflicts.



Layers

Presentation

- Vue.js
- Vuetify.js

Business

- Vue.js
- vue-rx
- Django
- Apache
- Microsoft Azure

Data

- Vue.js
- vue-rx
- SQLite
- Microsoft Azure

References

[1] D. S. Foundation, “DjangoThe web framework for perfectionists with deadlines,” [Online]. Available: www.djangoproject.com.

[2] T. A. S. Foundation, “Apache HTTP Server Project,” [Online]. Available: httpd.apache.org/.

[3] S. Consortium, “What Is SQLite?,” [Online]. Available: www.sqlite.org/index.html.

[4] M. A. Team, “Microsoft Azure,” Microsoft , [Online]. Available: azure.microsoft.com/en-au/.

[5] Vue.js, “Vue.js The Progressive JavaScript Framework,” [Online]. Available: vuejs.org/.

[6] Vuetify, “Vuetify Material DesignComponent Framework,” [Online]. Available: vuetifyjs.com/en/.

[7] vuejs, “vue-rx,” [Online]. Available: github.com/vuejs/vue-rx.

[8] vuejs, “What is Vuex?,” [Online]. Available: vuex.vuejs.org.