



**UNSW**  
SYDNEY

Wenqian (James) Zhang – z5286111  
LinFang (Eleanor) Dou - z5323991  
Zeal Liang – z5325156  
Qifeng (Rex) Zheng – z5345131  
YiKun Wang – z5365090

## Table of Contents

<b>Project Overview .....</b>	<b>3-8</b>
System Introduction.....	3
Primary Update from Proposed System .....	3
System Expectations .....	3
System Objectives .....	4
Main Designs and Themes .....	5
User Stories and Functionalities .....	6
Layers .....	6-8
Presentation Layer (Frontend).....	6
Business Layer (Backend).....	7
Database .....	7-8
Final Architecture / Structural Diagram.....	8
<b>Functionalities / Challenge Overview .....</b>	<b>4</b>
Feature 1 – Account Management .....	9-12
Feature 2 – Search Bar .....	12-14
Feature 3 – Voucher Releasing .....	14-16
Feature 4 – Voucher Redeeming .....	16-19
Feature 5 – Voucher Sharing.....	19-20
Feature 6 – Voucher Overview .....	20-21
Feature 7 – Menu Configuration (NF).....	21-23
Feature 8 – Review / Rating (NF) .....	23-24
Novelty Features Summary.....	25
Technical Challenges.....	26-27
Data Type Mismatch .....	26
Multithread .....	27
Testing.....	27
<b>Diagrams.....</b>	<b>28</b>
ER Diagram.....	28
Main Logic Structure .....	28
<b>Setup and User Manual.....</b>	<b>29</b>
Frontend Setup .....	29
Backend Setup.....	30-31
<b>Reference.....</b>	<b>32</b>

# Project Overview

## System Introduction

The proposed system that the group has designed and implemented is an eatery management system. The overall functionality of this system is to provide vouchers to users that allows them to use for discount at restaurant. Within this given context, we have identified two main users of this system:

- Customers:  
The customers are users who log onto the platform, and able to browse all restaurant and find possible and potential discounts that suit their needs. A primary goal of the team is to ensure customers have the best user experiences when browsing over the list.
- Restaurant Owners:  
The second biggest group of users for this platform refers restaurant owners. They will be able to release vouchers for their business and use this platform as opportunities for advertisement to attract more customers.

After stakeholder analysis, the system that the team has proposed will need to be inclusive, easy to access, and efficient. While the main objectives are set, the team took other similar system into considerations to identify their weaknesses and make sure they are strengthened in the systems.

## Primary Update from Proposed System

During the proposing stage for the system, the team has decided to develop an iOS mobile phone software using SWIFT and Xcode as the platform. However, later the team has confirmed with the tutor that this may not be feasible. Therefore, the team has chosen a web-based platform developed by React instead. This change happened late in the proposing stage, causing in differences in the actual implementation from the proposed features.

## System Expectations

As mentioned in the system introduction, the main target users can be categorised into two parties. The team has had frequent discussion throughout the development cycle to discuss potential goals that needs to be addressed. A main sequence of performed tasks can be identified:

1. User 1 registers account and classifies as a restaurant owner.
2. User 2 registers account and classifies as a customer.
3. User 1 releases vouchers.
4. User 2 collects and uses the voucher.

## System Objectives

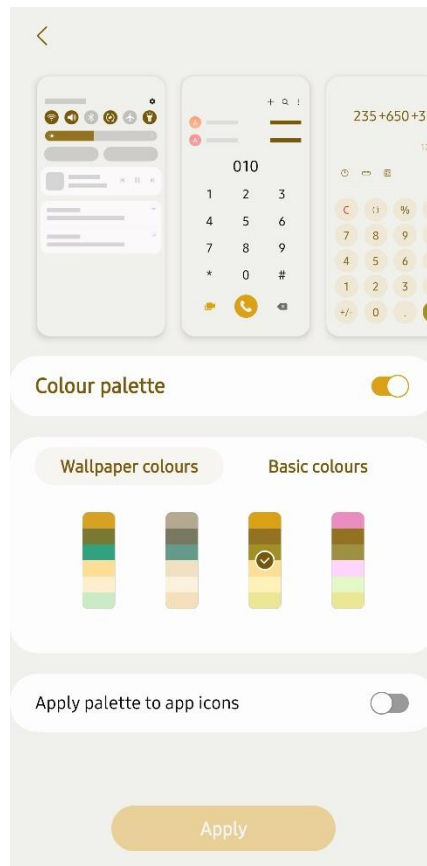
Based on the main sequence, the team carefully identified series of relevant potential objectives that need to be fulfilled alongside. With reference to the team's proposal, a final and updated table of sub-objectives are identified.

Restaurants	Customers
Register and maintain their accounts.	Register and maintain their accounts.
Change restaurant related information.	Change personal information.
Retrieve forgotten password.	Retrieve forgotten password.
Restaurants are visible and accessible to all customers.	Have access to all the existing restaurants.
Able to configure the menu.	Browse restaurants through the list or a searching function.
Able to upload picture for each dish.	Able to see the voucher information such as expiring time.
Able to browse and select different types of vouchers.	Can only collect the same voucher once.
Able to set a timer for automatic voucher releasing.	Can use the voucher after it is converted into a short code.
Able to set different properties of the vouchers such as description and shareable properties.	Able to regenerate the code if the code is expired.
Able to reply to other customers' comments.	Able to add text reviews.
Able to report comments.	Able to rate alongside the composed review.
Able to identify the corresponding discount according to the voucher code provided by the customers.	Able to view other customers' comments.
	Able to reply to other customers' comments.
	Able to post comments anonymously.
	Able to report comments.
	Able to overview the collected vouchers.
	Able to share vouchers to others if allowed.

## Main Designs and Themes

To ensure that the main design of website has professional looking, it is important for the team to decide what colours to be used over different sections of the website. To do so, the team has taken Google's approach as a reference.

Using Android 13 as an example, the system allows users to set colour palette. This is done by the system extracting main colours from the wallpaper, then apply those colours and integrate them throughout system options and pages such as the notification centre, the dial page, and other genuine application such as calculator.



Instead of using wallpaper to generate the main colours to be used as theme, the team held meeting and have decided a main logo should be designed first. Once the logo is designed, then colours used for the logo can be extracted and reapply those colours back to the website as main themes.

The platform being a voucher related food platform has inspired the team on the logo design as well as naming the platform. As a result, the name that the team has come up with is "Donut Voucher", which is direct and easy to understand. Correspondingly, the main logo design focuses on the "Donut" section of the naming. The main theme will be the colour of the donut, and colours of the lollies as the toppings will be use on different types of vouchers.



## User Stories and Functionalities

1. Basic Functionalities	2. Novelty Functionalities
Account management	Menu Configuration
Search Bar	Review / Rating
Voucher Releasing	
Voucher Redeeming	
Voucher Sharing	
Voucher Overview	

## Layers

As proposed earlier, the team went as planned. Overall, the team used React, PostgreSQL, and Python to develop the project.

### - **Presentation Layer (Frontend)**

In the front-end development, the team has decided to use React instead of React Native as our primary development language. This change is caused by the change in the group's main objective, primarily due to the concerns relating to the compatibility for the final demo. The key benefit of React is its component-based architecture, which promotes code reusability and modularity. This not only reduced the team's workload on similar frontend page design, but also contributed heavily on design consistencies. With a vast ecosystem of pre-built UI components, the team could leverage existing libraries and community-driven solutions to accelerate development and enhance user experiences while optimising code performance. React offers excellent performance due to its rendering capabilities, where it utilises native components and APIs, providing high level performance and responsiveness ([Business, I.T.& and Name \(2022\)](#)).

Another important capability offered by React is live rendering and hot reloading, this allows the team to directly access the design that have been implemented during the development without needing to recompile the entire application. Furthermore, React also enjoys strong community support and large number of resources. Overall, React Native offers the following benefits or advantages that increases work efficiency while contributing to user experiences of final project:

- Robust Framework
- Excellence Performance
- Vibrant Community

## - **Business Layer (Backend)**

Regarding back-end development, our team decided to use Python as the main development language. Python is a widely used programming language and most of the team members are familiar and had experienced with it. Its versatility, portability and robustness are the reasons why the team chose this language. As mentioned above in the Features/Functionalities section, the team has divided different users into different roles, although they share some of the same properties. Python has object-oriented features, which also helps the team in this case.

In addition to Python, we chose to develop the backend using the Flask framework, a popular Python backend framework that provides a comprehensive set of tools and libraries for building robust and scalable applications. The team chose to use Flask in the main project for several reasons ([Coursera \(2023\)](#)).

### *Simplicity:*

Flask is designed to keep things simple and streamlined, without imposing too many restrictions on developers. It doesn't have too many levels of abstraction, allowing developers to better understand and control how the application works.

### *Flexibility:*

Flask allows developers to customise and extend it according to their project needs. The team can choose to use the plugins and libraries needed without being limited by the constraints of the framework itself.

### *Community Support:*

Flask has an active community with plenty of documentation, tutorials, and resources to refer to. The team can get help from the community, solve problems, and learn best practices.

To summarise, Python and Flask are a powerful combination for building robust, scalable back-end applications. The team believes that choosing them as development tools was a wise decision.

## - **Database**

For data processing and database management, we have chosen PostgreSQL. It is a powerful and versatile open-source relation database management system ([Amazon AWS \(2023\)](#)).

### *High Performance:*

Due to its robust features and excellent performance, PostgreSQL has the capability to handle large volume of data while ensuring data integrity and reliability.

### High Compatibilities:

PostgreSQL has supports over different data types, including JSON, XML, and even geospatial data. Those supported file format will benefit us when pulling data from the internet for building our back-end database.

To pull data from the Internet, we will write a crawler using Python, which has many powerful third-party libraries such as Beautiful Soup, Requests, etc. These libraries provide a variety of tools and features for crawling and parsing web pages, making it easier to develop crawlers. For example, allowing developers to extract specific data elements such as text, images, links, and structured data from a website through customization options. As an overview, Python Crawler has the following advantages that benefit us in this project.

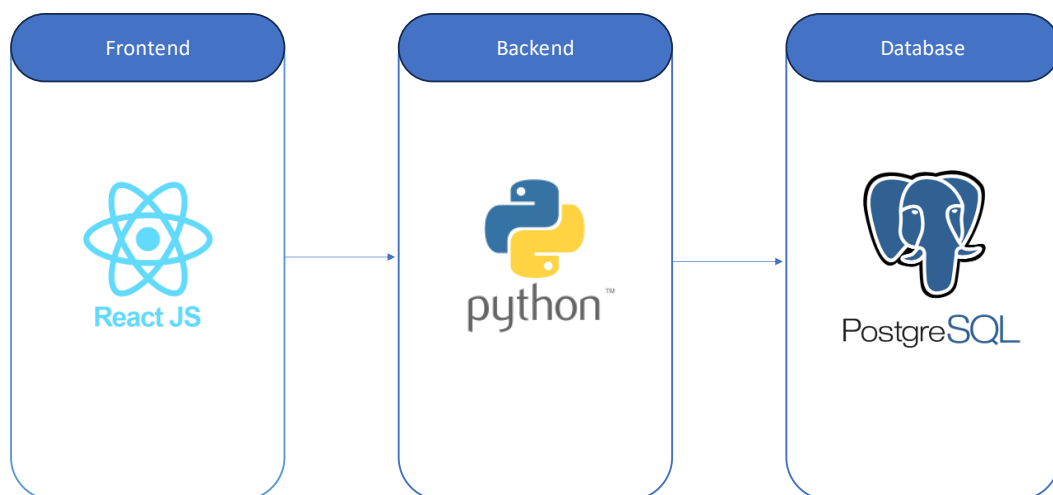
### Supportive Resources:

Python crawlers also benefit from a wide range of community support and extensive documentation. The availability of pre-existing code snippets, tutorials, and online resources makes it easier for us to get started with web scraping projects and troubleshoot any issues they may encounter along the way.

### Customisation:

Allowing us to specify what data we want to collect from the internet.

## - Final Architecture / Structural Diagram





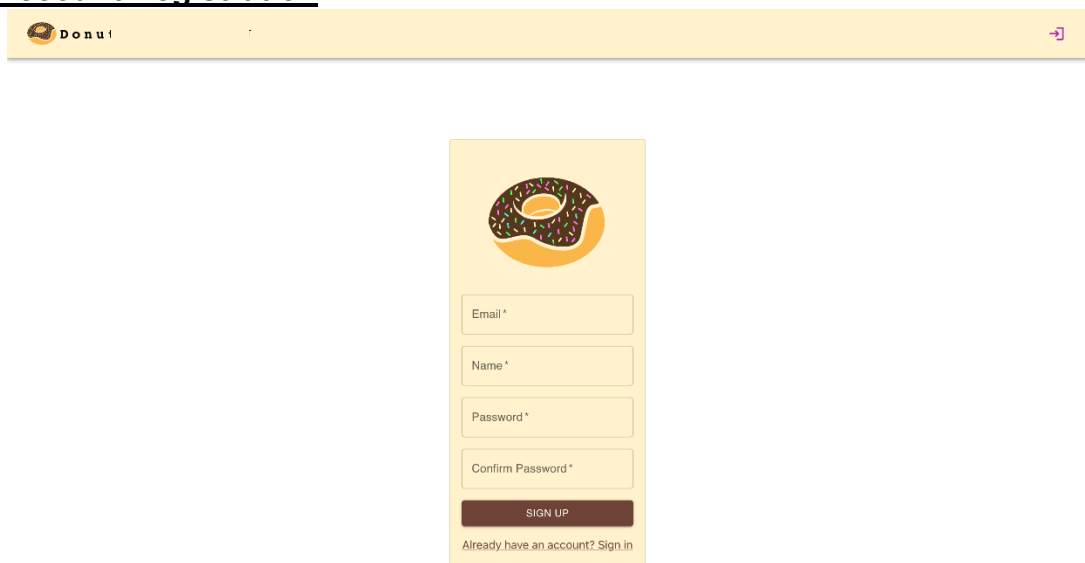
# Functionalities / Challenges Overview

This section of the report will mainly be focusing on the specific functionalities that the team has successfully implemented. This includes the name of the features as well as the specific screenshots taken from the actual implementation that address those issues. Further, this section of the report will also include explicit references to the proposed user stories and Jira board. For each proposed functionality, the team held meetings to identify the basic features and advance features that rely on the basic ones.

## Feature 1 - Account Management

This functionality mainly focuses on users managing their account starting from becoming a user to the platform. Within this function, the basic functionalities identified are the following:

- **Account Registration**

The image shows a web browser window with a yellow header bar. On the left of the header is the 'Donut' logo, and on the right is a small purple icon. Below the header is a registration form. At the top of the form is a large donut icon with colorful sprinkles. Below the icon are four input fields, each with an asterisk indicating it is required: 'Email \*', 'Name \*', 'Password \*', and 'Confirm Password \*'. Below these fields is a dark brown 'SIGN UP' button. At the bottom of the form, there is a link that says 'Already have an account? Sign in'.

(Figure 1.1)

As a newly arrived user, they should be able to register their personal account by providing the system with their personal information. Taking the criteria in the proposal and the outcomes of team meetings, the following information is selected to be critical for account registration.

- Email Address
- Username
- Password

With the provided email address, the system will automatically check the format of the provided email address. Password is also checked with criteria to ensure basic safety. The system will also require users to input the password twice to confirm the final password. ZEAL 说点啥

## - Account Log in/out

The screenshot shows the Donut application interface. At the top is a yellow header bar with the Donut logo on the left and a small icon on the right. Below the header, there are two main components. On the left is a login form with a donut icon at the top. It contains two input fields: 'Email \*' with the value 'James' and 'Password \*' with masked characters. Below these is a 'SIGN IN' button. At the bottom of the form are two links: 'Do not have an account? Sign up here!' and 'Forgot password?'. On the right is a user profile card. It has a yellow header with a user icon. Below this is a white box containing three options: 'Profile', 'liked Restaurant', and 'Logout'.

(Figure 1.2)

As a registered user to the platform, the team expected them to be able to sign in using the email address and the corresponding password. If a user is logged in, they should also be able to log out.

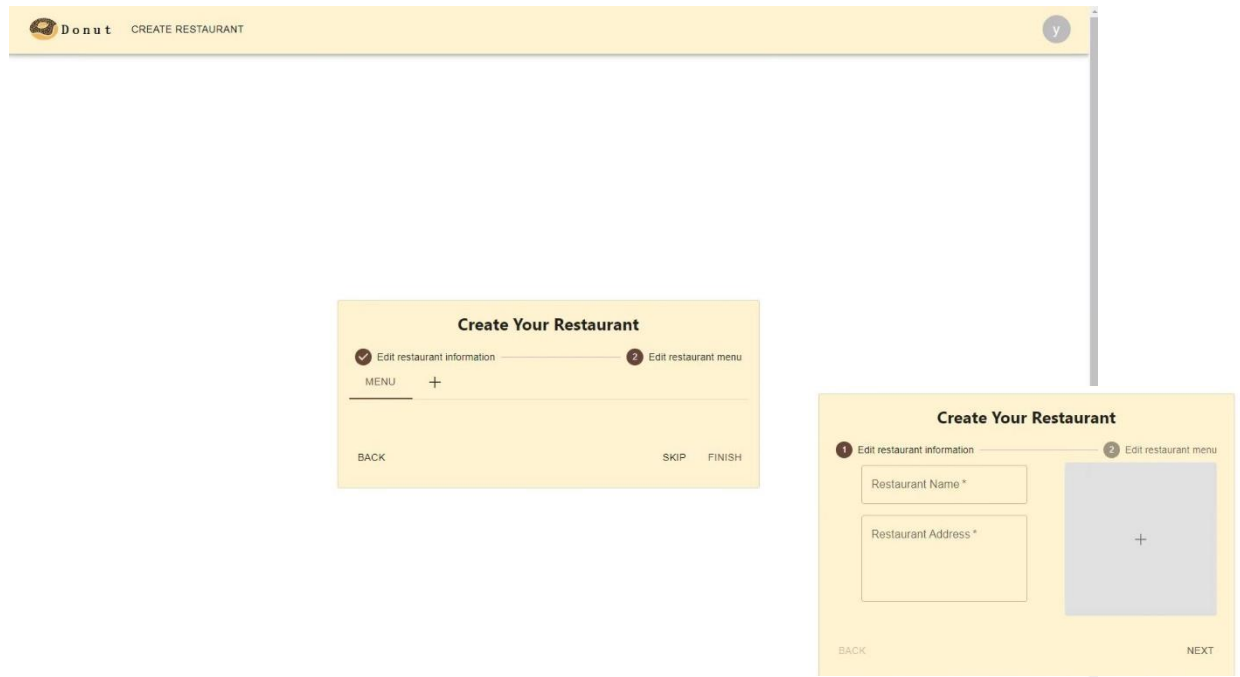
## - Retrieve Password

The screenshot shows the Donut application interface for password retrieval. It features a yellow header bar with the Donut logo on the left and a small icon on the right. Below the header is a form with a donut icon at the top. The form contains an 'Email \*' input field with the value 'zhangwenqian6915@gmail.co'. Below this is a 'RESEND CODE (43)' button. Further down are two more input fields: 'Code \*' and 'New Password \*'. At the bottom of the form is a 'SUBMIT' button. Below the button are two links: 'Do not have an account? Sign up here!' and 'Remember password? Sign in here!'.

(Figure 1.3)

The team has taken the case where users may not be able to log in due to forgotten password into consideration. Just like other platform, Donut Voucher will be able to send a code to registered users' email address, then they will be able to reset their password using this code. This can be done by clicking the forgot password button at the bottom of the screen as shown in figure 1.2. Then, the system will take the user to the following page.

## - Create Restaurant

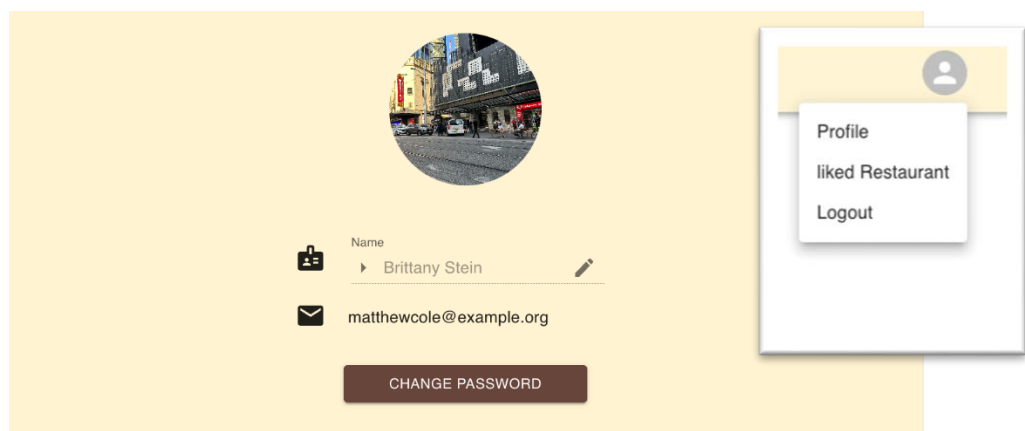


(Figure 1.4)

When users first registered on this platform, they are registered as a customer. For them to become a restaurant owner or account manager, they will need to click on the create restaurant on the status bar near the top of the page. When users are redirected, they will need to the following descriptions:

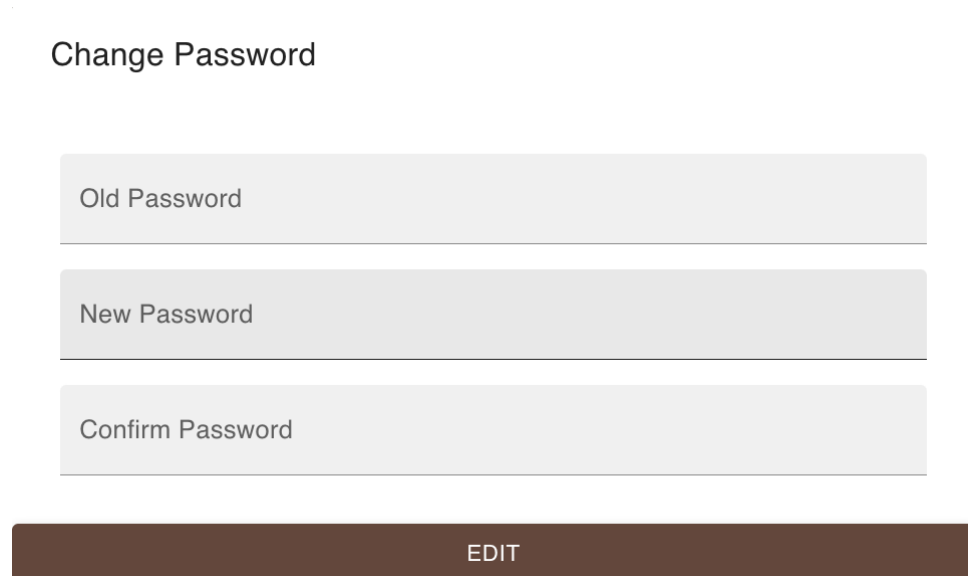
- Restaurant Name
- Restaurant Address
- Upload an image as the thumbnail.
- Creating Menu

## - Personal Information Configuration



(Figure 1.5)

The team also took the fact that users may change their username into consideration. From the home page, by clicking on the profile option from the pull-down menu, the users will be able to change their username and password. As email address in the primary information that users provide for signup, the system will not allow this information to be changed. When the users click on the “CHANGE PASSWORD” button, the following page is showed.

The image shows a web form titled "Change Password". It contains three input fields: "Old Password", "New Password", and "Confirm Password". Each field is a light gray rectangle with its label inside. Below these fields is a dark brown button with the word "EDIT" in white capital letters. The entire form is enclosed in a thin black border on the right and bottom sides.

Change Password

Old Password

New Password

Confirm Password

EDIT

**(Figure 1.6)**

In this page, users may also choose to modify their password. Unlike resetting or retrieving password, password modification is faster given that the users are able to provide their current password. Just like the sign-up process, users are also to require inputting their new password twice for confirmation.

## **Feature 2 – Search Bar**

A search bar is a function that fetches users with the results that match to their searching criteria. In the context of our system, this can be the name of the restaurant. The main advantage or object of this functionality is to boost users' efficiency when browsing. In particular, when users are not getting the information that they desire, a search bar will always be there as an alternative that increases user experiences.

Another side feature that the team chose to implement in assistance to the search bar is fuzzy search. Users may not always put the exact name of the desired restaurant into the search bar. With this feature implemented, users will be able to find the restaurant without having to know exactly what they are looking for.

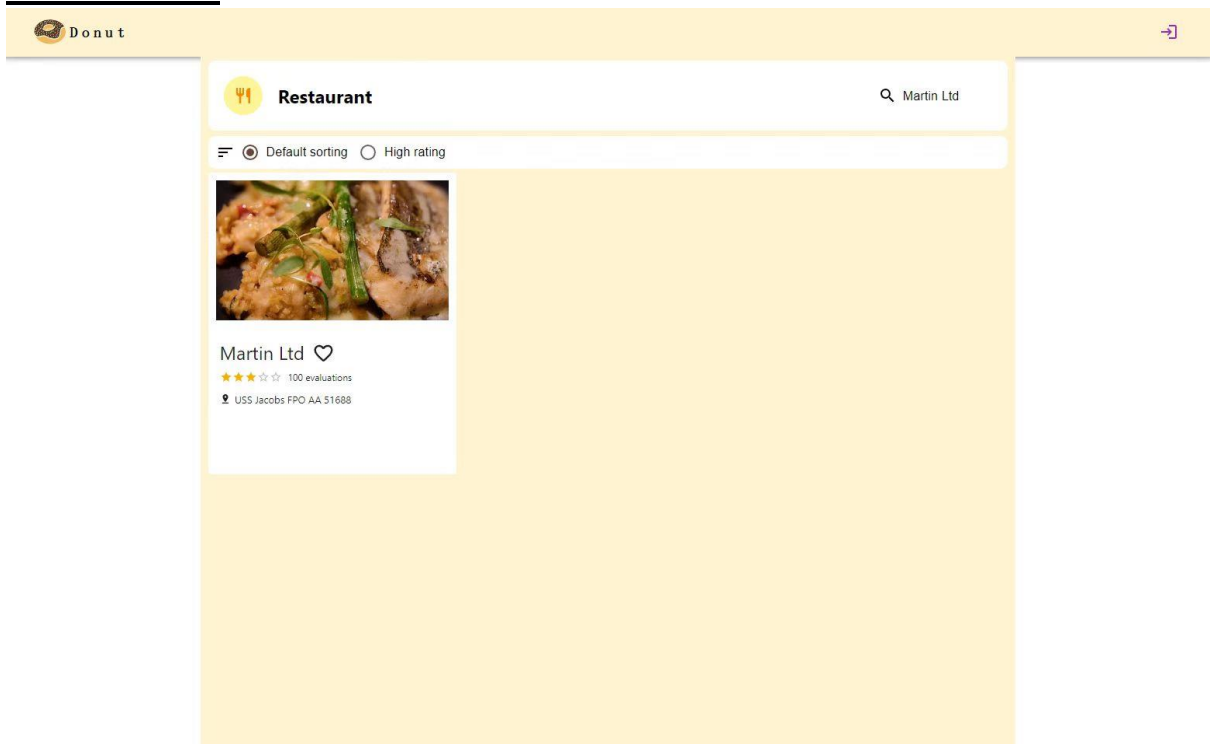
## - Access to the search functionality



(Figure 2.1)

One of the main objectives that the team proposed is to have the logo be obviously visible on the main page. Just like mainstream websites, the team has decided to use logo from Material and place the search option on the top right of the page. That way, the website provides users with the easiest and most common way of accessing the search function.

## - Search Result

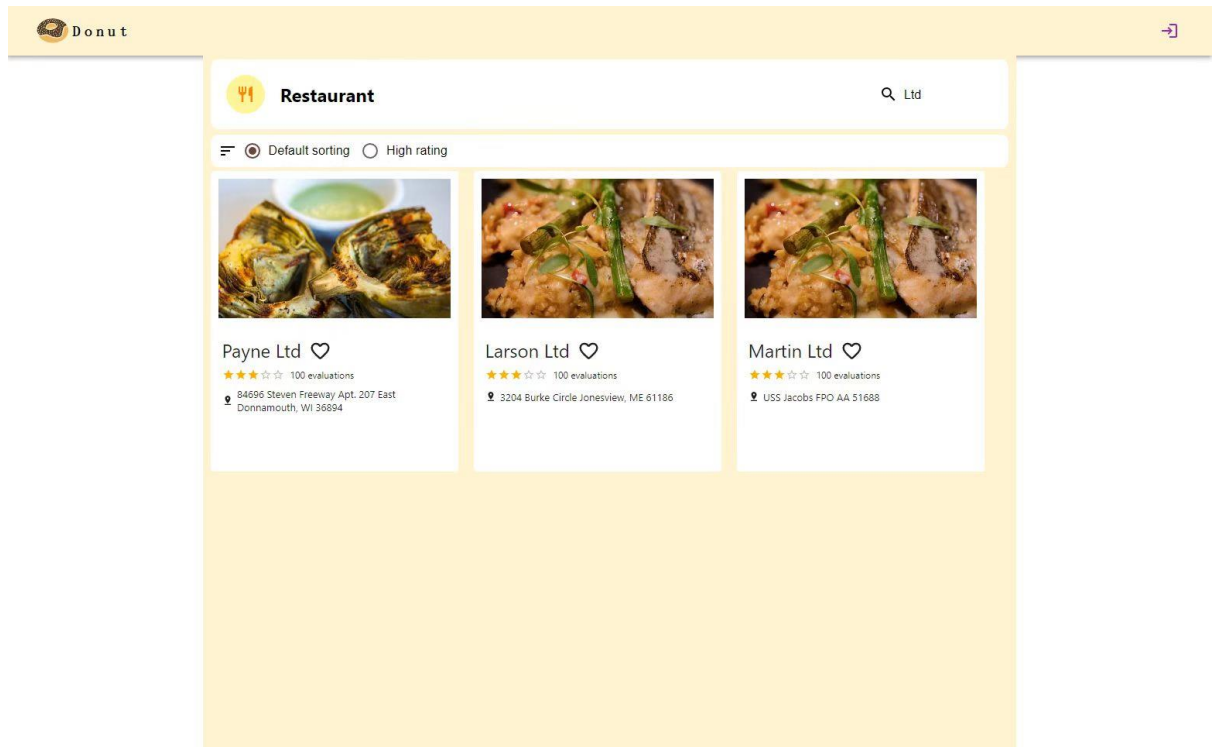


(Figure 2.2)

When the search button is clicked, the users will be provided with access to the textbox. The result page is also carefully designed by the team which doesn't just show the name, but also other details such as the address, average ratings, and the overall number of reviews received. Users can also subscribe restaurant from result list.

### - **Fuzzy Search**

As mentioned above, users may not know the exact names of restaurants that they are looking for, and this is where fuzzy search that is not cases-sensitive comes in place. As demonstrated in figure 1.7, the search input is “Martin Ltd”, however, if the search is only “Ltd”, the following three options with Ltd related to them.



(Figure 2.3)

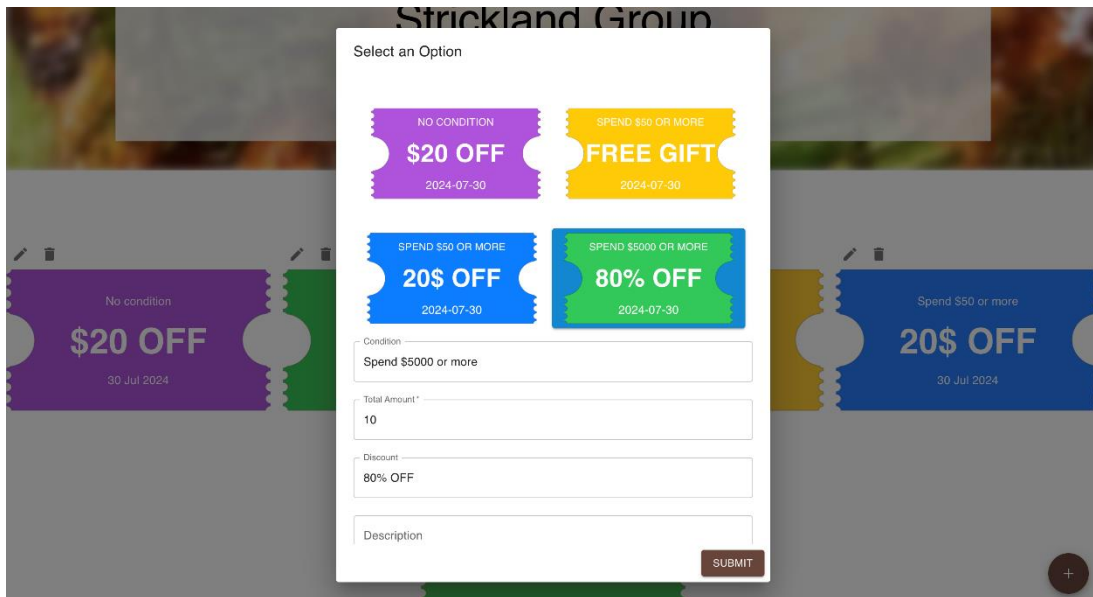
## **Feature 3 – Voucher Releasing**

The main objective of the project being a voucher related site is to allow restaurant owners to have smooth experiences when bridging the gap between their businesses and their potential target customers. This feature includes something such as “Percentage discount (By default)”, “Voucher for free food”, and “Vouchers that allow dollars discount rather than percentage discount”. The main advantage that this feature targets is to boost the efficiency and reduce the time spent of restaurant managers when releasing vouchers.

Another proposed feature that goes with this section is restaurant subscription notification. If a user subscribes to a restaurant, the system will automatically notify the users if this restaurant has released new vouchers via email. With reference to the automatic releasing voucher feature, users that are subscribed to the corresponding restaurant will also get notified at the beginning of every automatic releasing time frame.

- **Voucher Releasing Basic Configuration**

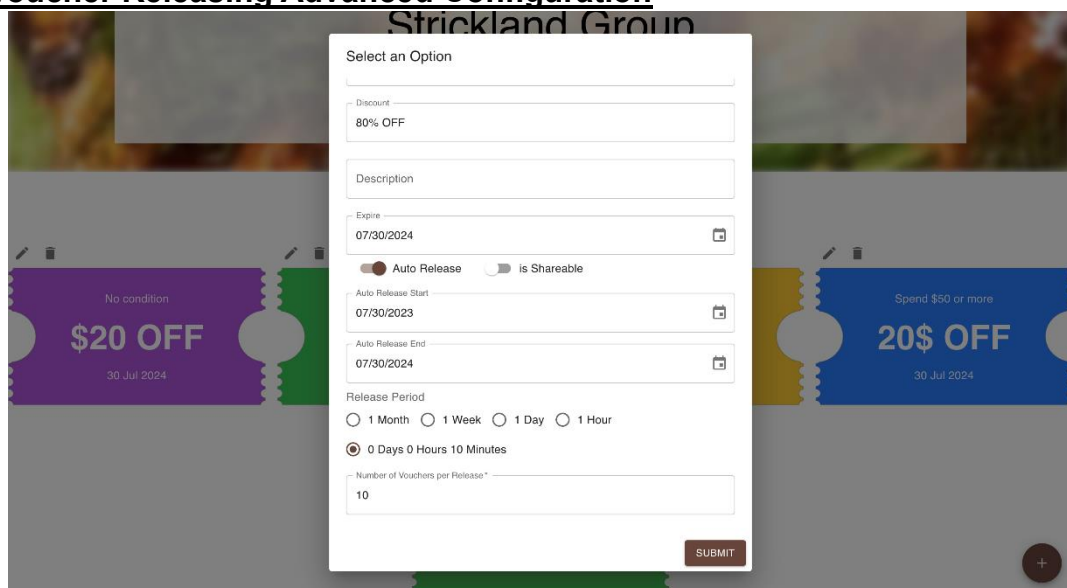
Once users have created their restaurant, there will be a create button at the bottom right of the restaurant main page for users to create vouchers.

A screenshot of a mobile application interface showing a modal titled "Select an Option" for creating a voucher. The modal displays four voucher options: "NO CONDITION \$20 OFF", "SPEND \$50 OR MORE FREE GIFT", "SPEND \$50 OR MORE 20\$ OFF", and "SPEND \$5000 OR MORE 80% OFF". Below these options, there are input fields for "Condition" (set to "Spend \$5000 or more"), "Total Amount" (set to "10"), "Discount" (set to "80% OFF"), and "Description". A "SUBMIT" button is at the bottom right. The background shows a blurred view of the restaurant main page with a "Strickland Group" header and various voucher icons.

(Figure 3.1)

To begin with the creating process, the restaurant owners can choose the type of vouchers that they want to release, that is, picking one out of the four types of vouchers. In addition to that, the users will be able set up description for the specific voucher that they are releasing, which will render onto the voucher shaped icon as visual representations. Other than that, more basic settings such as releasing amount and discount amount can also be configured on this page.

- **Voucher Releasing Advanced Configuration**

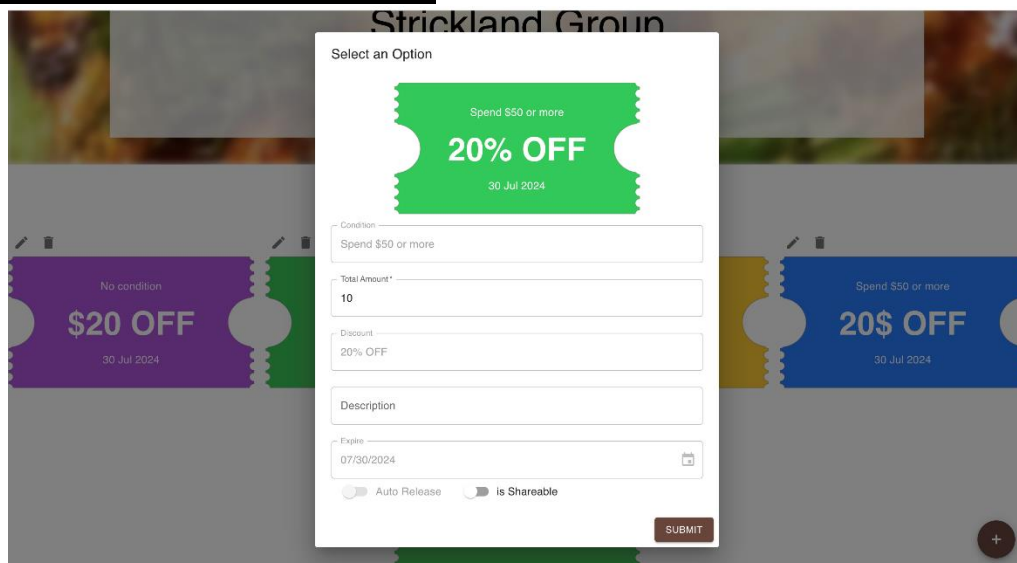
A screenshot of a mobile application interface showing a modal titled "Select an Option" for advanced voucher configuration. The modal includes input fields for "Discount" (set to "80% OFF"), "Description", "Expire" (set to "07/30/2024"), "Auto Release" (checked), and "Is Shareable" (unchecked). It also has fields for "Auto Release Start" (set to "07/30/2023") and "Auto Release End" (set to "07/30/2024"). Below these, there are radio buttons for "Release Period" with options "1 Month", "1 Week", "1 Day", "1 Hour", and "0 Days 0 Hours 10 Minutes" (selected). At the bottom, there is a field for "Number of Vouchers per Release" (set to "10") and a "SUBMIT" button. The background shows a blurred view of the restaurant main page with a "Strickland Group" header and various voucher icons.

(Figure 3.2)

When users scroll further down this small window, they will be able to access the automatic releasing feature. The automatic release of voucher can either start directly and finish at desired time, or users can decide both of the start and end timestamp for automatic releasing. Within this timeframe, the user will be able to set a releasing gap and releasing amount per gap. Restaurant manager can also decide whether this voucher can be shared or not through the swipe.

To explain this, an example can be used. Suppose the starting date and ending date timestamp are both 31/07/2023, and a releasing gap of 1 hour with 10 voucher per release, then from the starting time, the system will automatically release 10 vouchers per hour, until 23.59pm on the 31<sup>st</sup> of July.

#### - **Released Voucher Modification**



(Figure 3.3)

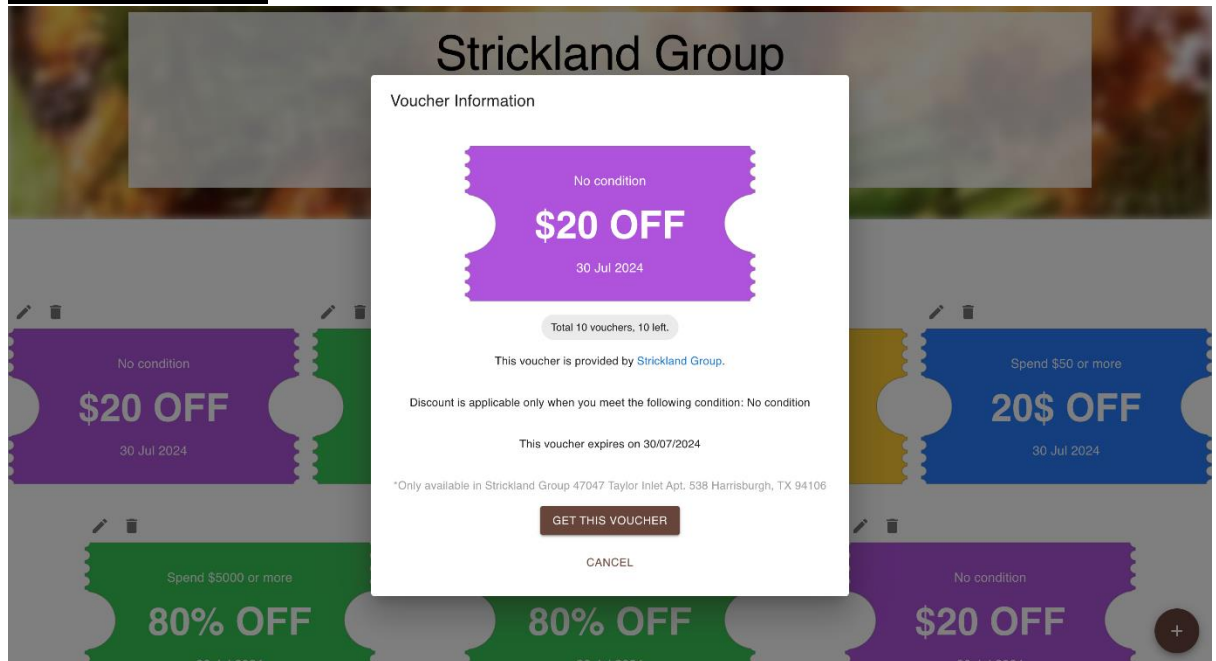
The team also provided owners access to modify their existing released vouchers. This function allows user to supply customers with more of the same type of the vouchers. However, fundamental conditions cannot be changed.

## **Feature 4 – Voucher Redeeming**

This feature allows customers to collect and use their vouchers. Once a voucher is collected, it will show in customers' individual "Wallet" or "Inventory". Initially during the proposing stage, the way for customers to use their vouchers is to provide their personal information to the restaurant to redeem the discount. However, the team has evolved that idea and redesigned it by converting the voucher into a 4-digit time limited code, which can then be used to redeem at the restaurant. Another proposed feature for users is the ability for them share vouchers. All those can be done by clicking into the restaurant page to view and collect feasible vouchers.



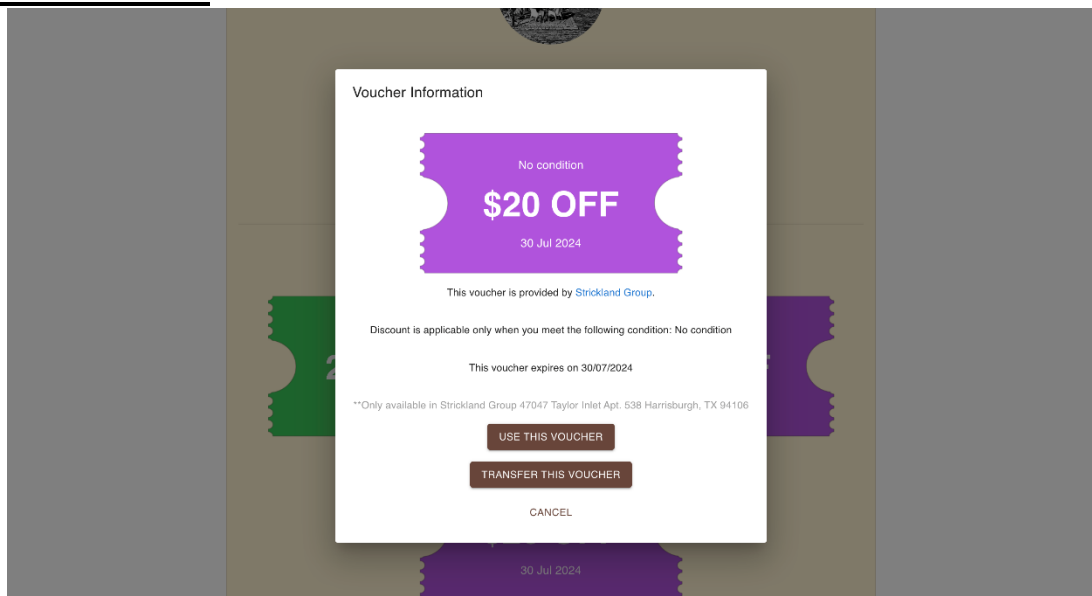
- **Collect Vouchers**



**(Figure 4.1)**

When vouchers are released by restaurant, all users will have the ability to collect them. This is done by clicking the preview of the vouchers which will then be able to be collected. Information such as voucher details, and remaining voucher count will be provided over this screen.

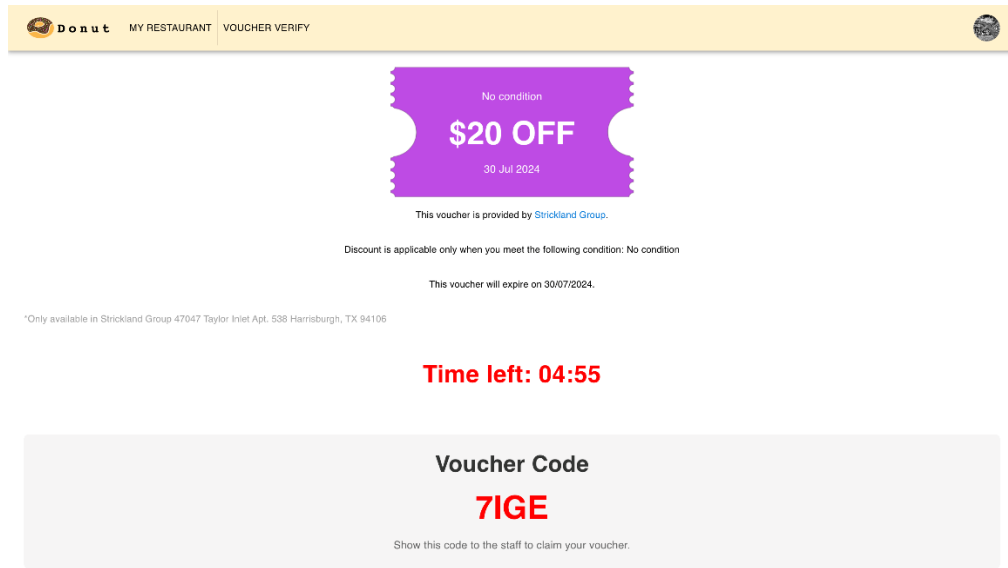
- **Use Vouchers**



**(Figure 4.2)**

When users have collected their desired voucher, the platform allows vouchers to be used by converting them into a 4-digit code. By providing this code to the restaurant, the discount can be applied. This is done by users

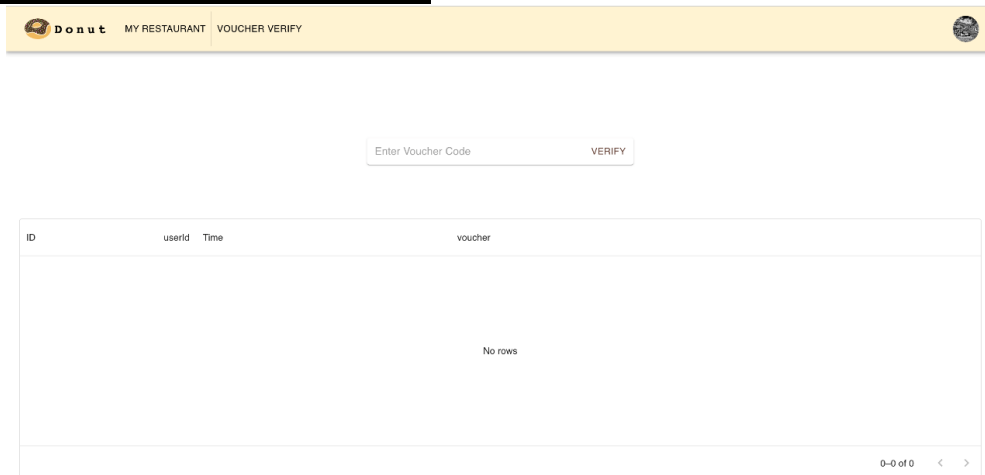
going into their inventory, then click on one of the collect vouchers, a page (Figure 4.2) will pop up.



(Figure 4.3)

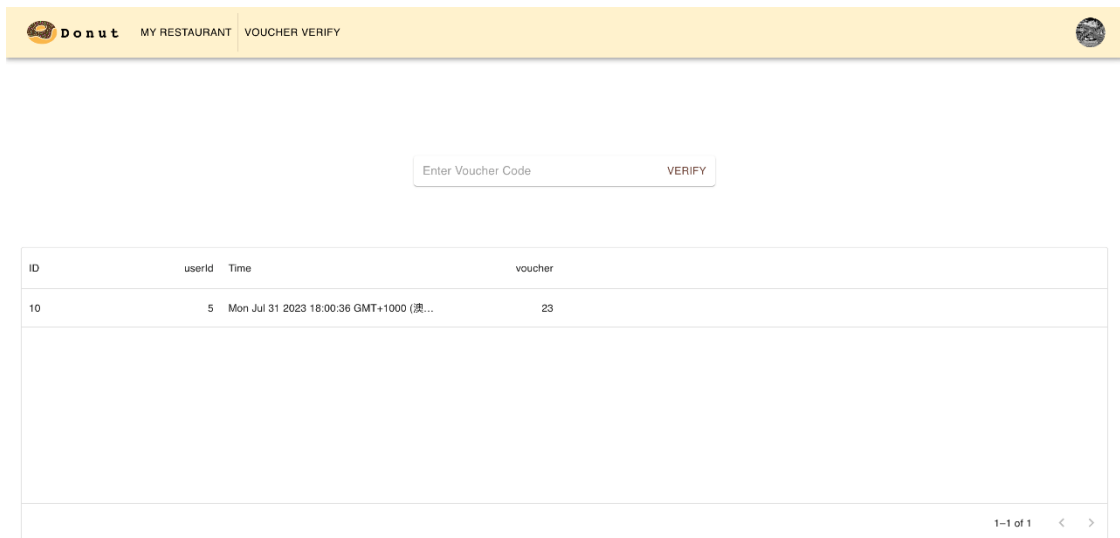
If users want to use this specific voucher, they can click on the “Use voucher” button which directs them to the digit page (Figure 4.3) with countdown. Users will be able to reacquire this voucher if the time runs out.

#### - Restaurant Voucher Verification



(Figure 4.4)

This feature primarily focuses on verifying the code provided by customers before restaurant can apply any discount. To do so, restaurant owner can click on the “Voucher Verify” button on the top status bar, which will lead them to the verification page (Figure 4.4).



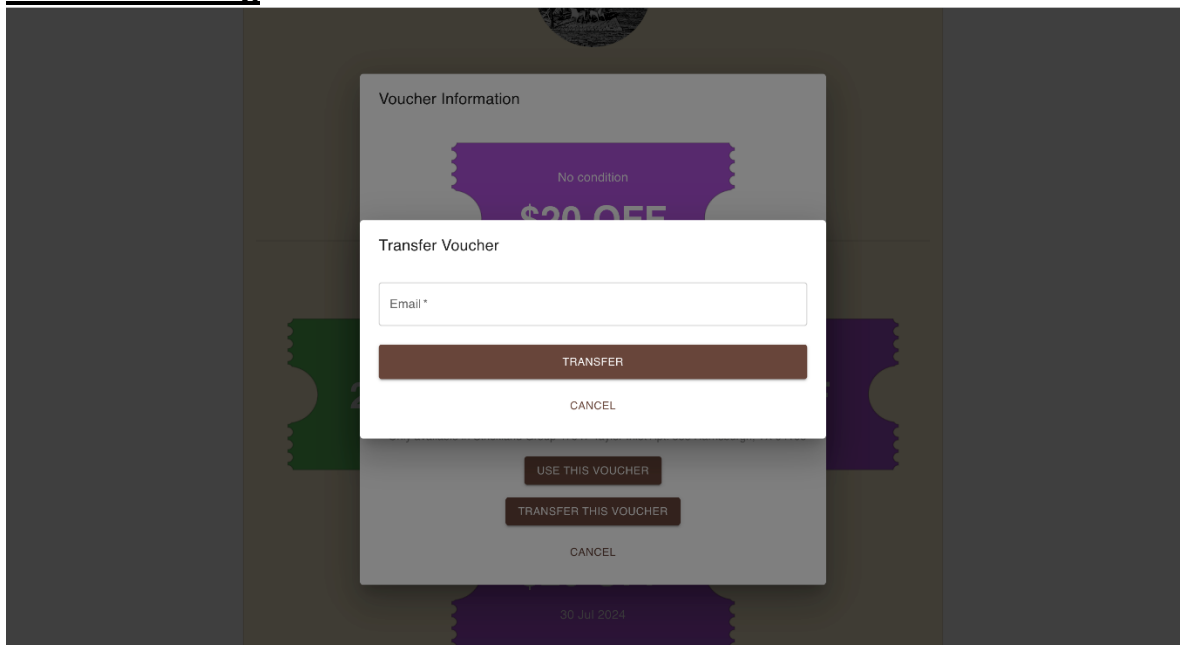
(Figure 4.5)

For successfully validated vouchers, they will show as records down in the main screen.

## Feature 5 – Voucher Sharing

This is one of the inspiring features proposed by the team, and it is also not a feasible feature on another similar platform. In this feature, the team has removed the setting of not able to collect the same voucher after sharing it, making the system more logical.

### - Voucher Sharing



(Figure 5.1)

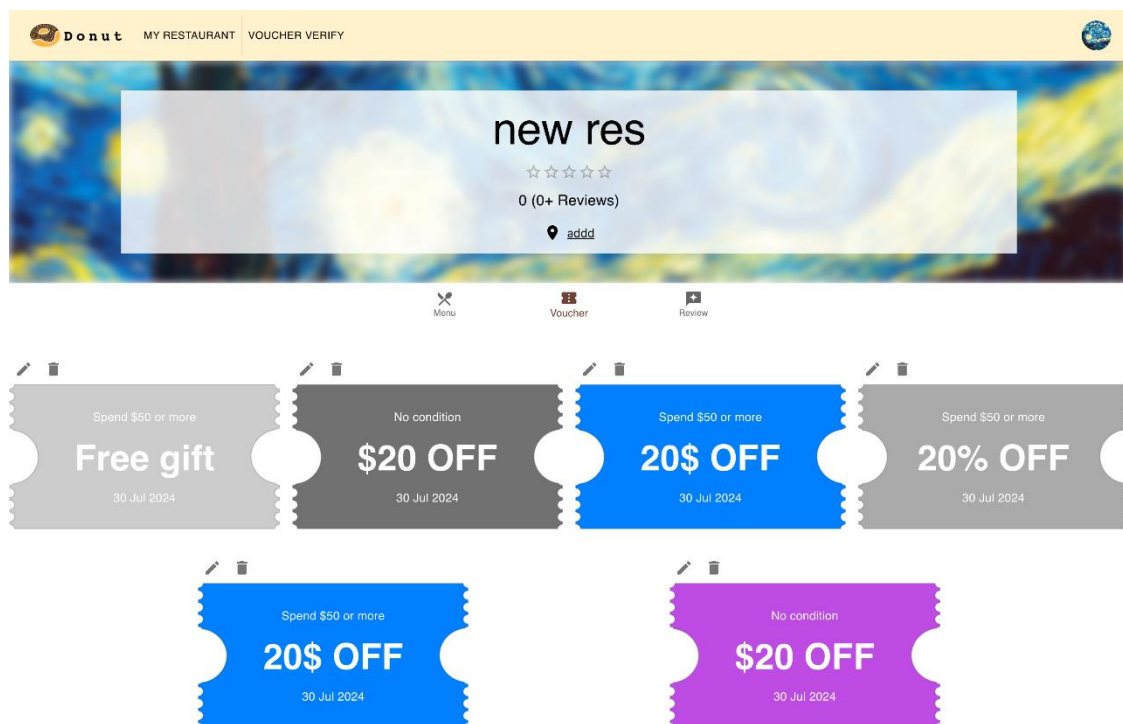
On the voucher detail page as shown in figure 4.2, the user will be able to share or transfer this voucher to a target receiver via providing the correct

email for the receiver, the system then removes the corresponding voucher from this user's inventory and create the same corresponding voucher in the target receiver's inventory. It is important to note that this button will not show up on vouchers' detail page if they are explicitly set by the owner that they cannot be shared among users.

## Feature 6 – Voucher Overview

The main objective of this feature is to provide both customers and restaurant owners with an overview of their previous activities. With respect to the overall conformity of the theme, the team has decided to use coloured vouchers to represent the unused voucher (available vouchers) and greyed out colours to represent the used voucher (unavailable vouchers).

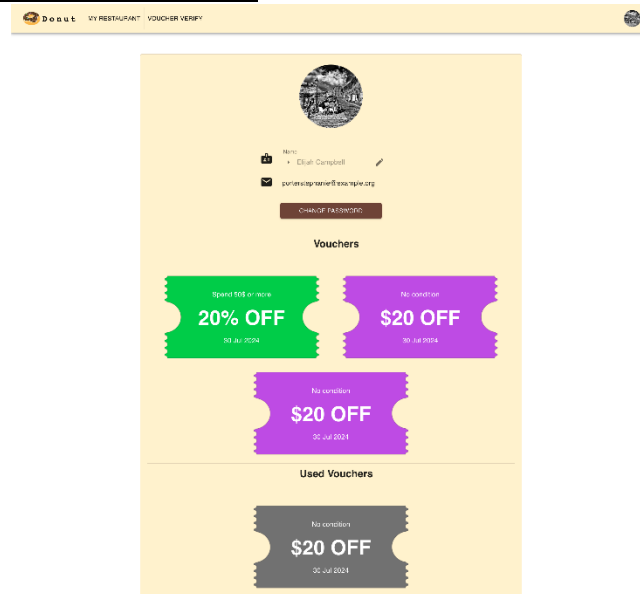
### - Voucher Overview – Restaurants



(Figure 6.1)

On the restaurants' pages, the owners will be able to view the existing vouchers and expired or unavailable vouchers. The team has also provided a remove voucher option which can remove the history of the vouchers.

## - Voucher Overview – Customers



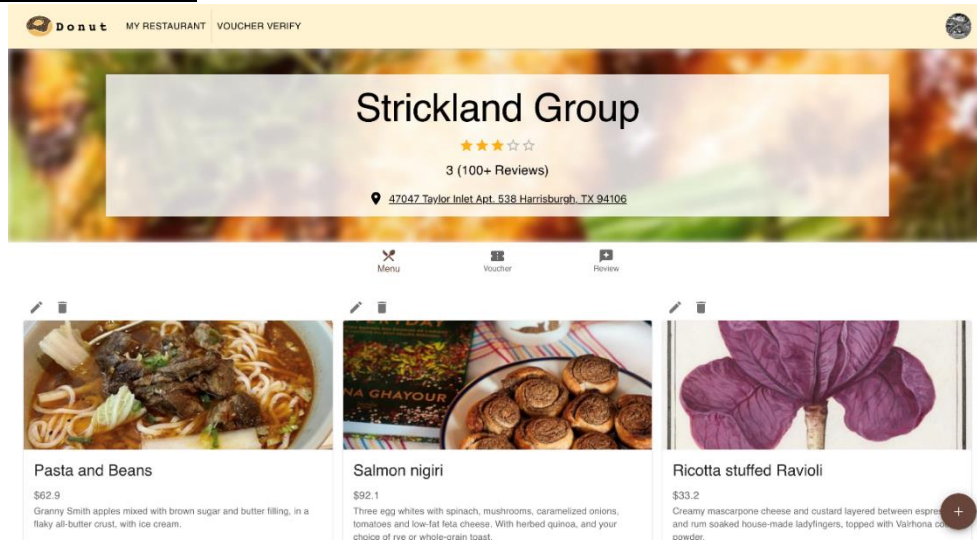
(Figure 6.2)

Similar to the overview on the restaurant's side, customers will also have access to the inventory with all collected vouchers. This is the page where customer get an overview of what they have already and what they have used in the past.

## Feature 7 – Menu Configuration (NF)

This option allows restaurant account managers to upload their unique menu. They can choose to have the menu typed in. That is, the name of the dishes, one picture associated with each dish and a brief to detail description of the dish. Those can also be edited so that new dishes can be added to the menu, and existing dishes can be removed from the menu. The advantage of this feature is that restaurants are able to maximise their customer satisfaction through live interactions by updating new offers as well as removing options that are no longer available which further increases reputation.

## - Place for Editing

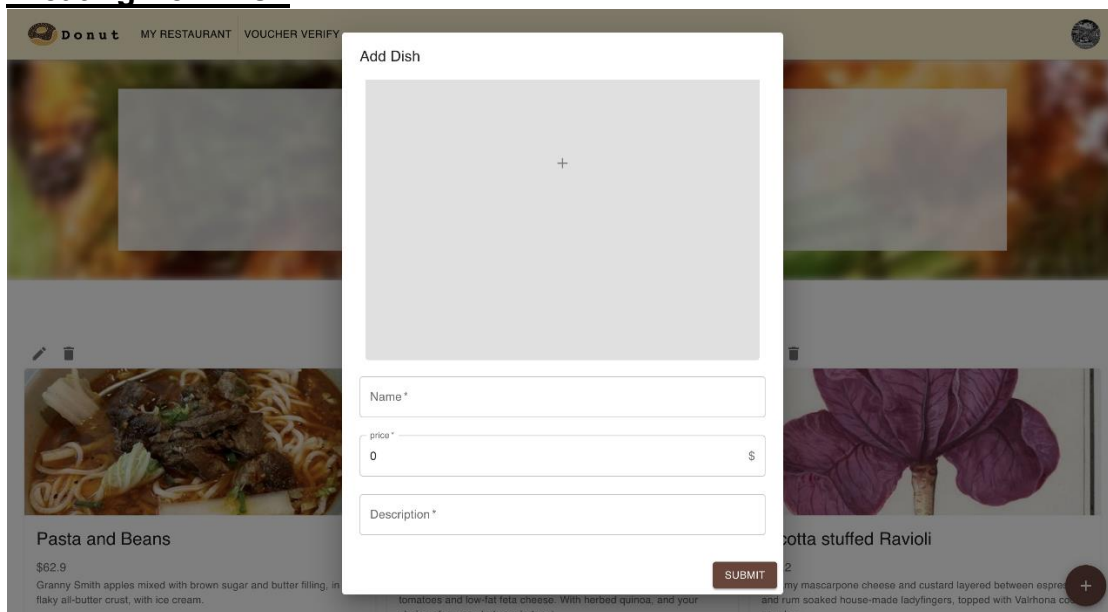


(Figure 7.1)

There are two approaches to fulfil the main objective of editing/creating dishes onto the menu. The first approach is to create new dishes. This is done by restaurant owners clicking the add button on the bottom right of the screen. This will take them to figure 7.2 with a specific dish creating page.

The second approach will be editing the existing dishes. For configuration and changing the information of the existing dish, restaurant owners can click the pencil icon on the top left section of each dish, which will take them to figure 7.3. It is a similar interface as 7.2, however, this page will keep the existing information for editing.

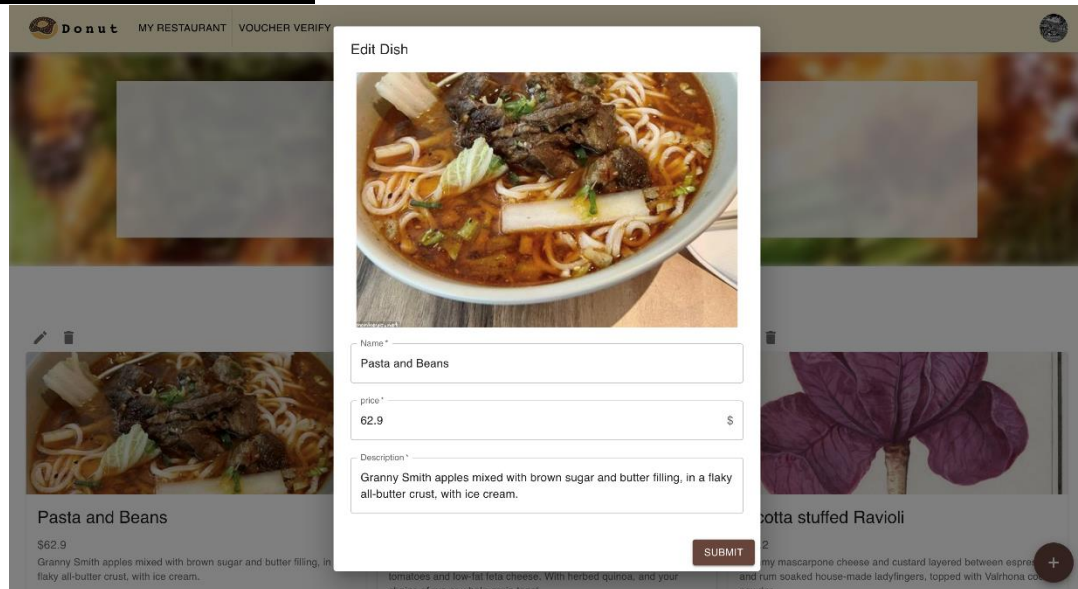
## - Creating New Dish



(Figure 7.2)

This will be the rough template for restaurant owners when creating sections for their new dishes. The information boxes are blank for fill in.

#### - Editing Existing Dish



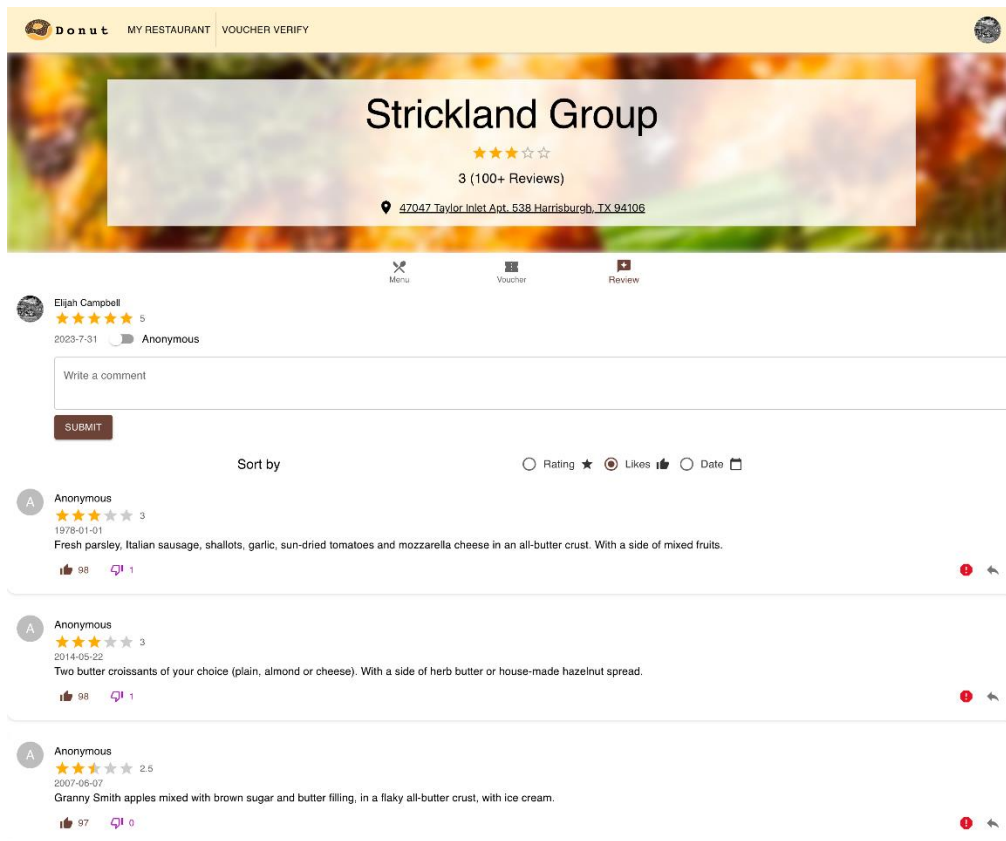
(Figure 7.3)

The editing page will keep the existing information of the dish, which allows restaurant owners to add or remove existing information related to certain dishes.

## Feature 8 – Review / Rating (NF)

This feature allows customers to rate and review certain restaurants they have visited. This can include the quality of the food that they are served with and the quality of the services. Furthermore, they can also comment on the pricing, locations, and so on. Alongside a text review, the customers can also rate the restaurant, so that an average rating of the restaurant can be calculated and displayed in the main page. All users can reply to review, and report comments. This feature benefits the user community as the restaurant owners can make improvements or adjustments according to customers' feedback, and customers will be able to get a rough idea of what the quality of certain restaurant is like by reading over other customers' experience of their previous visit. Comments can also be sorted.

Alongside the reviewing system, the team has also introduced a report feature. This feature automatically removes reviews that are flagged multiple times by other users.



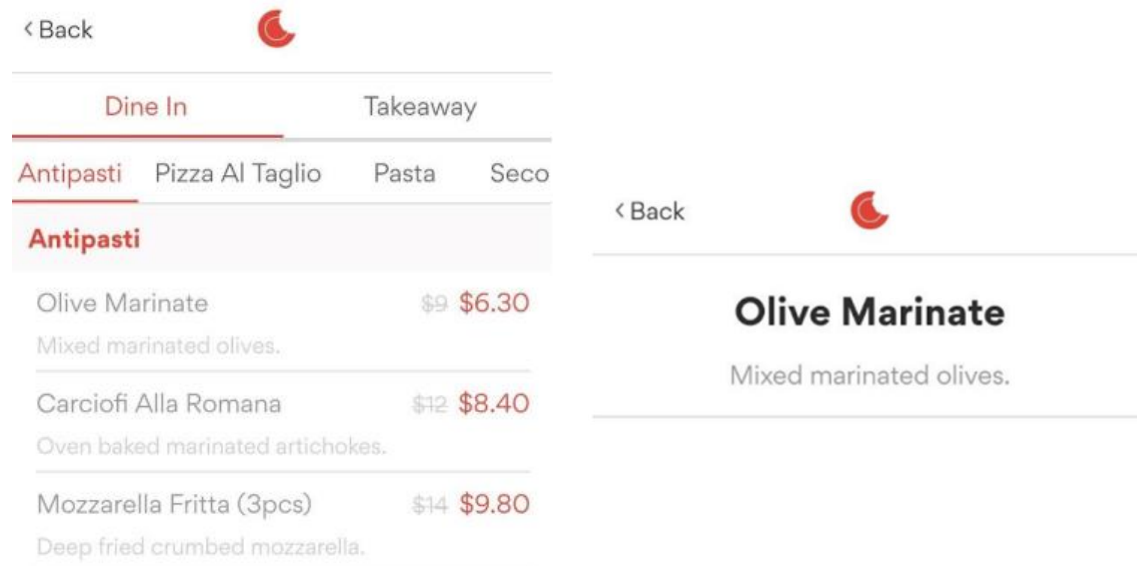
(Figure 8)

When user choose to visit the preview section, page in figure 8 is shown. Near the top of the review section, user will have access to a composing tab with star ratings above. To address potential privacy concerns, an anonymous option is also provided which hides composers' name and replace their profile picture.



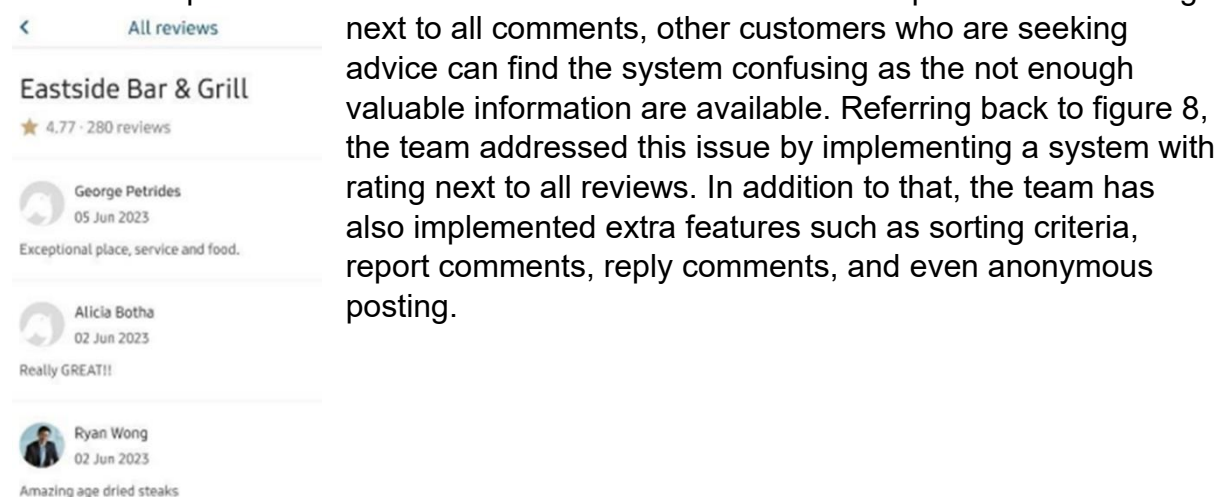
## Novelty Features Summary

To address the issue that the team has identified on similar platform “Eat Club”, the team has implemented a comprehensive menu setting which allows users to insert sufficient information for their potential customers ([Graves, A. \(2022\)](#)).



Both screenshots above showed the lack of information, in particular image descriptions. The team consider this to be very important as they provide an overview on what the customers are paying for before they make the decisions. Referring back to figures 7.1 -7.3, the developed system provides both users with convenient, effective, and user-friendly interfaces when providing dishes related information to customers.

The second novelty feature addresses the lack of ability of user-user interactions on the review system equipped on First Table. This review system provides insufficient interaction options as well as information. With reviews are composed without ratings

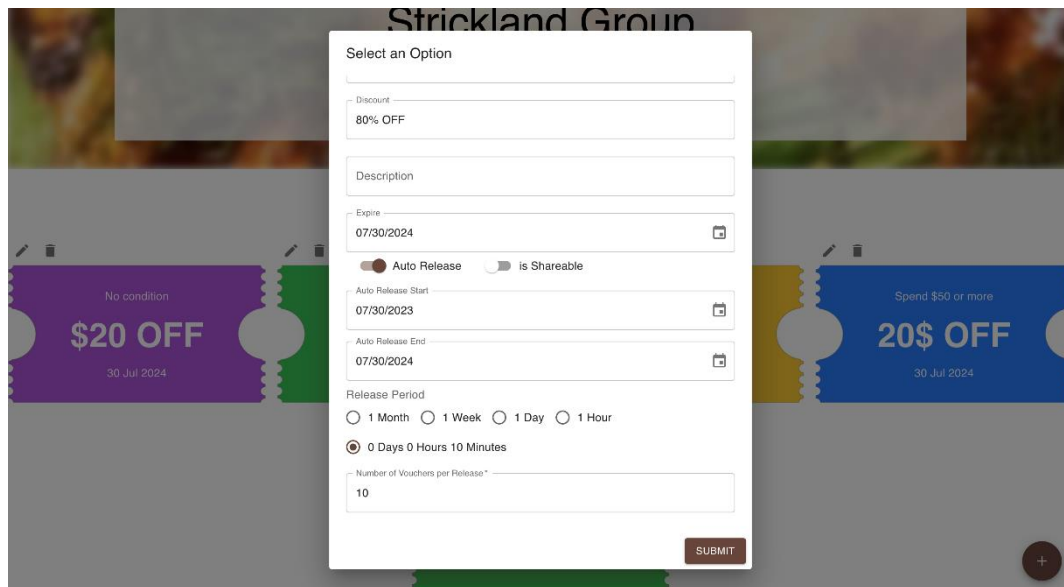


next to all comments, other customers who are seeking advice can find the system confusing as the not enough valuable information are available. Referring back to figure 8, the team addressed this issue by implementing a system with rating next to all reviews. In addition to that, the team has also implemented extra features such as sorting criteria, report comments, reply comments, and even anonymous posting.

# Technical Challenges

## 1) Data Type Mismatch

One of the most difficult bugs that the team has ran into was related to the manipulation of the data types. As proposed as part of the main features of this project, the restaurant owners are able to release vouchers automatically.



To be specific, the system requires user to input days, hours, and limits for automatic voucher releasing. The frontend then converts the time into minutes, then give this data to the backend sections to start the count-down. The bug appears to be the incorrection in the conversion process. For example,  $100 + 1$  results in 1001 in total.

Initially, the group dived deep into lines and liens of code over the frontend page but was unsuccessful to identify the potential cause of the bug. The team later realise that the bug appears to be relating to data type as that 1 seems appended at the end of 100 to form 1001, which matches the characteristic of a char type.

This then directs the team to revisit the section that does the calculation and identified the issues at line 409. To fix this, the team has used the function “parseInt” to make sure the right data types are manipulated to produce the right outcome which can be send to the backend.

```
405     props.setAutoReleaseTimeRange(ReleasePeriodHour * 60 + ReleasePeriodMinute);
406   } else if (e.target.value > 365) {
407     props.setAutoReleaseTimeRange(365 * 24 * 60 + ReleasePeriodHour * 60 + ReleasePeriodMinute);
408   } else {
409     props.setAutoReleaseTimeRange(parseInt(e.target.value) * 1440 + parseInt(ReleasePeriodHour) * 60 + parseInt(ReleasePeriodMinute));
```

The is changed is then tested to ensure the functionality, and the team has concluded that this change is valid that resolves the problem.

## 2) Multithread

React.useEffective runs in a multithread condition, changing values in React.useState does not automatically update and change the values in React.useState. Instead, React.useEffect is used. This posed difficulties when changing multiple values in React.useState.

This is more like the difficulties that the team ran into, and there is no so-called “Solution” to fully resolve this.

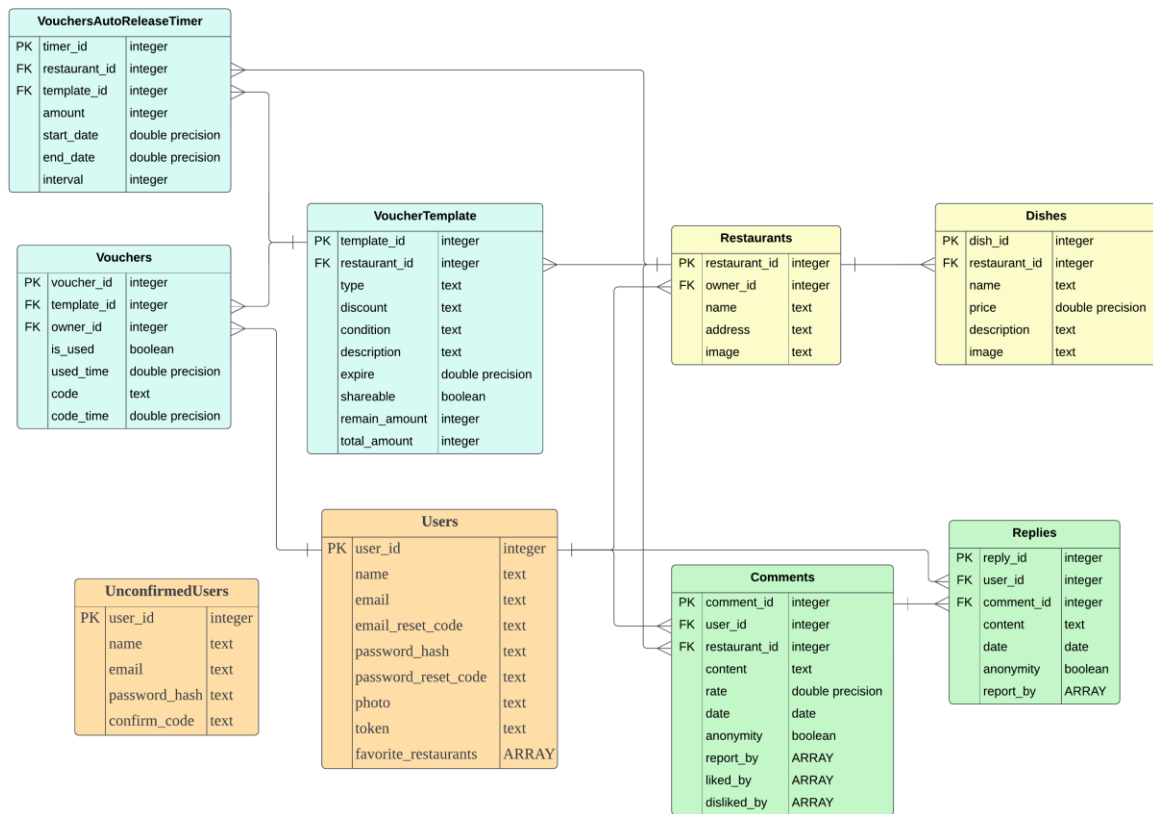
## Testing

For the overall project, the main testing objective for testing is to cover the backend section of the development. The team, Zeal in particular ensured the overall coverage to be above 80. Majority of the cases are tested to ensure functionalities remain obtained as proposed in the initial development stage of this process. Overall, test coverages for all backend files are shown on the following coverage report.

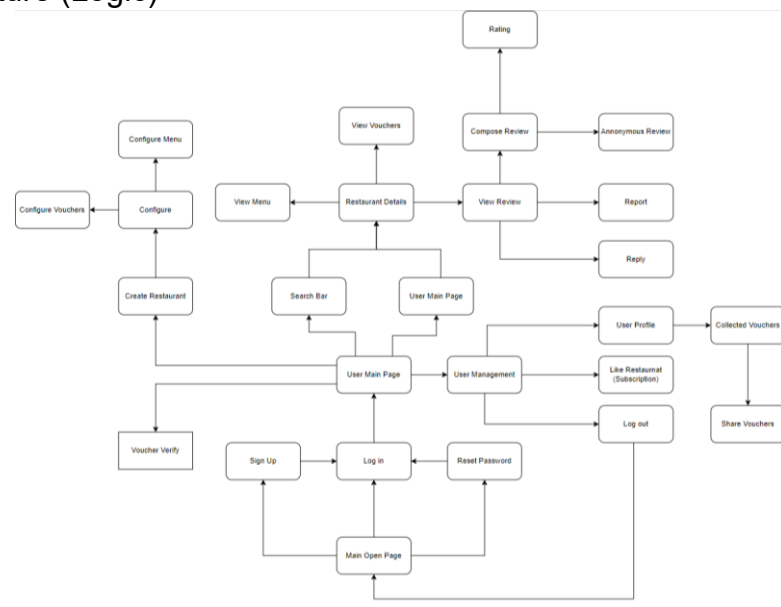
Coverage report: 83%				
coverage.py v7.2.7, created at 2023-07-29 17:17 +1000				
Module	statements	missing	excluded	coverage
backend\app\__init__.py	13	0	0	100%
backend\app\config.py	24	0	0	100%
backend\app\models\__init__.py	26	5	0	81%
backend\app\models\comments.py	80	0	0	100%
backend\app\models\dishes.py	42	0	0	100%
backend\app\models\replies.py	43	0	0	100%
backend\app\models\restaurants.py	38	0	0	100%
backend\app\models\unconfirmedUsers.py	23	8	0	65%
backend\app\models\users.py	66	21	0	68%
backend\app\models\voucherTemplate.py	58	24	0	59%
backend\app\models\vouchers.py	50	20	0	60%
backend\app\models\vouchersAutoReleaseTimer.py	47	17	0	64%
backend\app\routes\__init__.py	33	2	0	94%
backend\app\routes\comments.py	197	2	0	99%
backend\app\routes\dishes.py	171	7	0	96%
backend\app\routes\replies.py	106	1	0	99%
backend\app\routes\restaurants.py	157	1	0	99%
backend\app\routes\users.py	329	76	0	77%
backend\app\routes\vouchers.py	270	147	0	46%
backend\app\services\__init__.py	9	0	0	100%
backend\app\services\comments.py	18	3	0	83%
backend\app\services\dishes.py	21	1	0	95%
backend\app\services\replies.py	10	3	0	70%
backend\app\services\restaurants.py	26	2	0	92%
backend\app\services\users.py	143	93	0	35%
backend\app\services\util.py	58	37	0	36%
backend\app\services\voucherTemplate.py	5	1	0	80%
backend\app\services\vouchers.py	50	35	0	30%
backend\app\services\vouchersAutoReleaseTimer.py	5	1	0	80%
backend\tests\__init__.py	0	0	0	100%
backend\tests\comments_test.py	236	0	0	100%
backend\tests\confTest.py	18	0	0	100%
backend\tests\dishes_test.py	144	0	0	100%
backend\tests\replies_test.py	111	0	0	100%
backend\tests\restaurants_test.py	117	0	0	100%
backend\tests\users_test.py	179	0	0	100%
backend\tests\util.py	28	0	0	100%
Total	2951	507	0	83%
coverage.py v7.2.7, created at 2023-07-29 17:17 +1000				

# Diagrams

- ER Diagram for backend database.



- Main structure (Logic)



# Setup and User Manual

## Frontend Setup on Vlab

1. Clone the Project:

```
git clone git@github.com:unsw-cse-comp3900-9900-23T2/capstone-project-3900w18aZEAL.git
```

2. Move to the frontend directory:

```
cd frontend
```

3. Install dependency package:

```
npm install
```

4. Execute:

```
npm start
```

## Backend Setup

1. Use VirtualBox:

<https://www.virtualbox.org/wiki/Downloads>

Lubuntu 20.04.1 LTS:

[https://sourceforge.net/projects/linuxvmimages/files/VMware/L/lubuntu\\_20.04.1\\_VM.zip/download](https://sourceforge.net/projects/linuxvmimages/files/VMware/L/lubuntu_20.04.1_VM.zip/download)

2. Username: lubuntu  
Password: lubuntu

3. Prerequisites:

psql (PostgreSQL) >=13.4

python >=3.11

4. Clone the Project:

```
git clone git@github.com:unsw-cse-comp3900-9900-23T2/capstone-project-3900w18aZEAL.git
```

5. cd backend:

Change user and password to your psql's username and password in

```
app/config.py
```

6. Install pipenv:

```
pip3 install pipenv
```

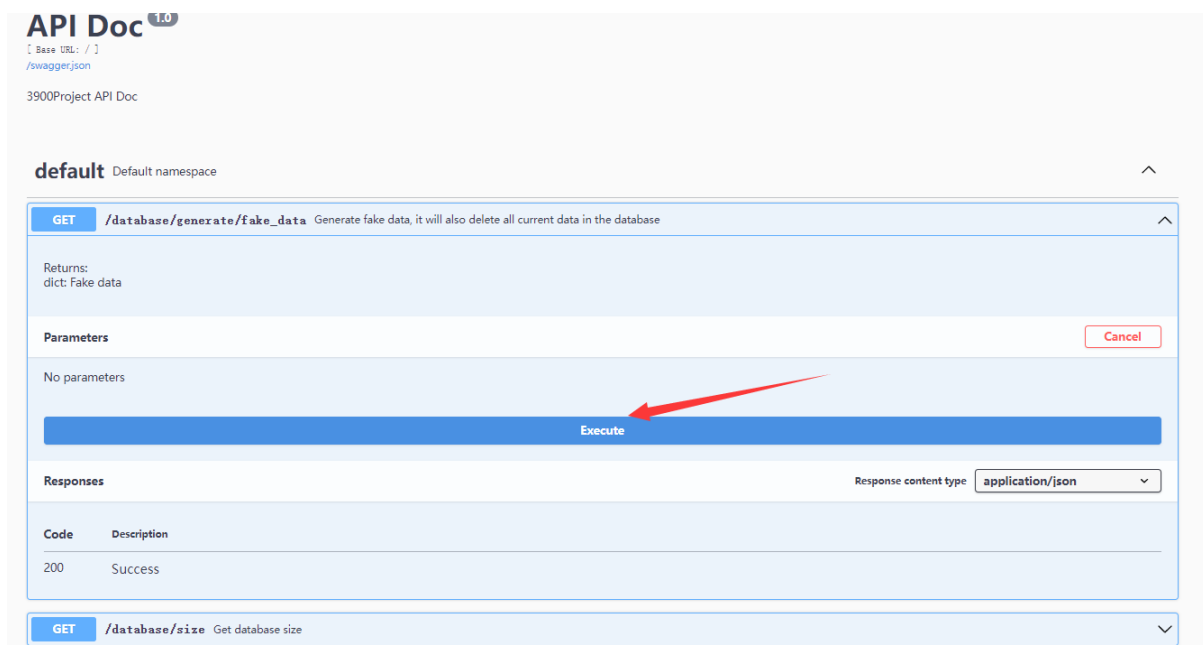
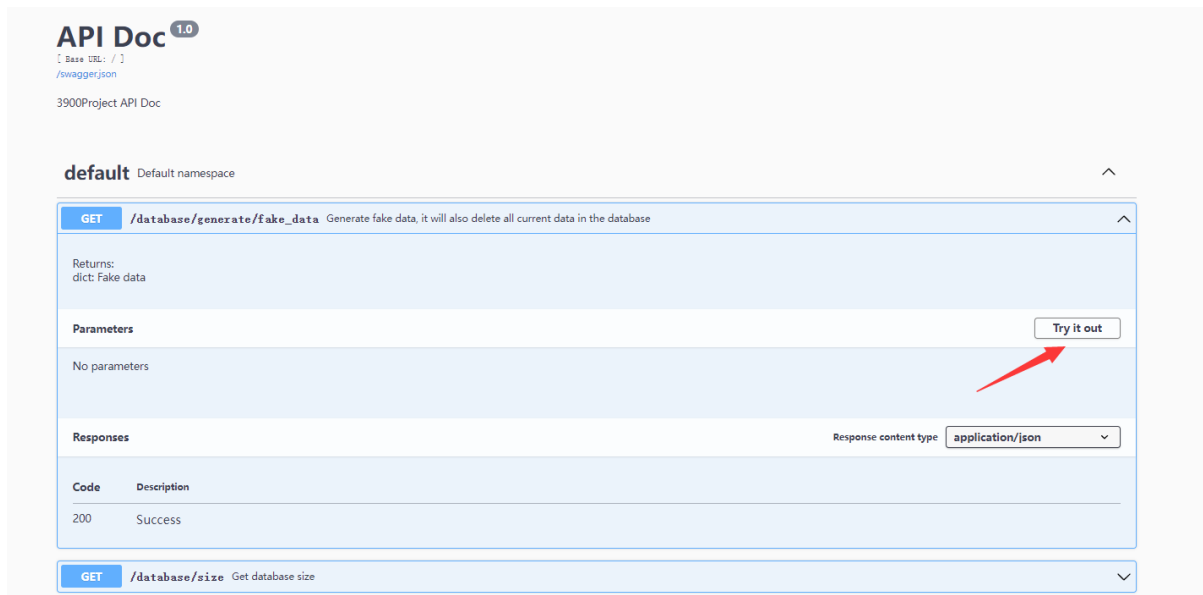
7. Install all dependency package:

```
pipenv run pipenv install --dev
```

8. Launch backend server:

```
pipenv run python run.py
```

If generations of fake data are required, the users will need to copy the link provided by the terminal after the backend is launched, which takes users to the backend API page to execute.



The progress of the generation process is shown on the backend terminal, the estimation time required is around 2 minutes.

# Reference

Coursera (2023) What is python used for? A beginner's guide, Coursera. Available at: <https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python> (Accessed: 28 July 2023).

Amazon AWS (2023) What is PostgreSQL? Amazon. Available at: <https://aws.amazon.com/rds/postgresql/what-is-postgresql/> (Accessed: 28 July 2023).

Graves, A. (2022) Restaurant ratings and reviews: Benefits and tips, Bloom Intelligence. Available at: <https://bloomintelligence.com/blog/ratings-reviews-benefitsbenchmarks/#:~:text=The%20more%20positive%20ratings%20you,on%20your%20overall%20bottom%20line>. (Accessed: 27 July 2023).

## **Supporting Source for Setup:**

(2023) Download lubuntu\_20.04.1\_vm.zip (linux VM images) - sourceforge. Available at: [https://sourceforge.net/projects/linuxvmimages/files/VMware/L/lubuntu\\_20.04.1\\_VM.zip/download](https://sourceforge.net/projects/linuxvmimages/files/VMware/L/lubuntu_20.04.1_VM.zip/download) (Accessed: 03 August 2023).

*Download virtualbox* (2023) Oracle VM VirtualBox. Available at: <https://www.virtualbox.org/wiki/Downloads> (Accessed: 03 August 2023).