



FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY

Programme: RSD (Group: 5)

Assignment

BACS2163 SOFTWARE ENGINEERING

Name(Block Letters)	Registration No.	Signature	Marks
1.Ong Yi Xin	24WMR09097		
2. Chia Ming Yi	24WMR09040		
3. Lim Jun Wei	24WMR09078		
4. Ng Ian Kai	24WMR09090		
5. -	-	-	-

Tutor's Name: YASOTHA MUNIRETNAM

Date of Submission: 15/9/2024



FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY

Plagiarism Statement

Read, complete, and sign this statement to be submitted with the written report. **We confirm that the submitted works are all our own work and are in our own words.**

Name(Block Letters)	Registration No.	Signature	Date
1. Ong Yi Xin	24WMR09097		
2. Chia Ming Yi	24WMR09040		
3. Lim Jun Wei	24WMR09078		
4. Ng Ian Kai	24WMR09090		
5. -	-	-	-

Table of Content

Part 1	6
1.1 Organization Background	6
1.2 Requirement Gathering Techniques	8
Questionnaire	8
Time Effective	8
Privacy Concern	8
Cost-Effectiveness	8
Standardization	9
Information Gathered	9
Observation	10
Realistic Data Collection	10
Determination of Errors and Inefficiencies	11
Profound Requirement Description	11
Detection of Emotion Changes	11
Information Gathered	12
1.3 Problems of Existing System	13
1. Limited Online Ordering Channels	13
2. Inadequate Customer Support	13
3. Incomplete Menu Information	14
4. Poor Menu Layout	14
5. Lack of Availability Status	15
1.4 Software Quality Attribute	16
1. Availability: The probability that a system will work as required during the period of a mission.	16
2. Functional: The system will perform the functions that it is required to.	16
3. Useable: Provides users with a satisfying experience (not a source of dissatisfaction).	16
4. Reliable: Not prone to hardware or software failure, will deliver the functionality when the users want it.	17
5. Flexible: Capable of being adapted to new users, to run in different countries or to be moved to a different platform.	17
1.5 Software Process Model	18
Comprehensive requirements of system	19
Quick Introduction into Market	20
Part 2	21
2.1 Project Planning and Schedule	21
2.2 System Requirement (Functional Requirements)	27
Order (Ong Yi Xin)	27

Delivery and Track Order (Ong Yi Xin)	29
Product (Chia Ming Yi)	32
Payment (Chia Ming Yi)	34
Table Reservation (Lim Jun Wei)	36
Member Registration (Lim Jun Wei)	37
Membership (ian kai)	39
Promotion (ian kai)	40
2.3 Non-Functional Requirements	42
Product	42
Organization	43
External	43
2.4 View-Point Oriented Analysis	44
Customer (Ong Yi Xin)	44
Manager (Lim Jun Wei)	45
Auditor (Chia Ming Yi)	47
Marketing Staff (Ng Ian Kai)	49
Part 3	50
3.1 An Architectural Design	50
Client-Server Model	50
Catalogue Server	50
Picture Server	50
Database Server	51
Front-End Server	51
Back-End Server	52
Food Server	52
Justification	52
3.2 Test Plan	55
Test Objective	55
Testing Process	55
Unit Testing	55
Module Testing	56
Sub-System Testing	57
System Testing	57
User Acceptance Testing (UAT)	58
Tested Items	58
Testing Techniques	59
Testing Environment	61
Testing Constraints	61
3.3 Test Case	63
Add to Cart (Ong Yi Xin)	63
Positive Test Case	63

Negative Test Case	63
Schedule Delivery Time (Ong Yi Xin)	64
Positive Test Case	64
Negative Test Case	65
Process Payment (Chia Ming Yi)	67
Positive Test Case	67
Negative Test Case	67
Create New Product (Chia Ming Yi)	68
Positive Test Case	68
Negative Test Case	69
Reserve Table (Lim Jun Wei)	71
Positive Test Case	71
Negative Test Case	71
Register as Member (Lim Jun Wei)	72
Positive Test Case	72
Negative Test Case	73
Update Membership Points Balance (Ng Ian Kai)	75
Positive Test Case #1	75
Positive Test Case #2	75
Add New Promotion (Ng Ian Kai)	77
Positive Test Case	77
Negative Test Case	77
3.4 UI Design Principles	79
Ong Yi Xin (Modify Shopping Cart)	79
User Familiarity	79
Consistency	80
Recoverability	80
User Guidance	81
Chia Ming Yi (Create New Product)	82
User familiarity	83
Consistency	83
Minimal Surprises	83
User Guidance	83
Lim Jun Wei (Table Reservation)	84
User Guidance	84
Consistency	85
Recoverability	87
Minimal Surprise	88
Ng Ian Kai (View Points History)	89
User Familiarity	89
User Diversity	90

User Guidance	90
Consistency	90
References	91

Part 1

1.1 Organization Background

Sushi Mentai is a Japanese food restaurant which was founded by a group of passionate entrepreneurs at the September of Year 2017, which included 2 directors and 2 shareholders or members. They are *inspired by their love for Japanese cuisine, and decided to introduce an authentic yet accessible sushi dining experience to Malaysians.* (adib, 2023) Poo Wee Ming is the director and also shareholder of Sushi Mentai Enterprise SDN. BHD., Tan Ng Seng position as a secretary and there is one more shareholder who is called Liew Kam Chuan. The business address of Sushi Mentai is at the Muar of Johor where Wee Ming also lives. In 2024, Sushi Mentai already had up to 50 outlets in Malaysia and is also tapping into the Singapore Market, becoming a household name for affordable Japanese restaurants.

Sushi Mentai is a Japanese restaurant chain specializing in affordable sushi and a wide variety of other Japanese dishes. Restaurants typically use a conveyor belt system, also known as rotating sushi, where trays of sushi and other food items circulate through the dining area, allowing customers to pick and choose their favorite dishes at will. In addition to an extensive sushi menu, Sushi Mentai serves a variety of Japanese foods, including curry fried chicken rice, teriyaki chicken rice, and miso soup ramen. To enhance the dining experience, they also offer two types of noodles, udon and soba, to cater to different taste preferences and ensure that every customer is satisfied with their meal.

Sushi Mentai focuses on efficient supply chain management and operational practices to keep prices reasonable while maintaining quality. For example, the price of sushi that Sushi Mentai sells is always maintained at RM1.80 and RM 2.80. The casual dining atmosphere attracts a wide range of customers, including families, students and young professionals. Regular promotions, a vibrant online presence and menu innovations help attract and retain customers. Sushi Mentai combines quality, affordability and convenience to successfully deliver an enjoyable Japanese dining experience.

Sushi Mentai urgently requires an online food ordering system of its own as they currently only have a food ordering function via Whatsapp but does not apply on every branch. Some of the branches can order via WhatsApp and deliver but some can only order for pickup or dine in and some branches even do not allow for the food ordering through WhatsApp. Not only that, Sushi Mentai also does not cooperate with any outside food delivery organization such as GrabFood or FoodPanda. This has greatly reduced the developable market for Sushi Mentai.

An online ordering system allows customers to place an order at any time through a restaurant's website or mobile app. The system typically includes a user-friendly interface where customers can browse menus, select items, customize their order and securely pay. Once the order is placed, the system sends the order details to the restaurant kitchen for preparation and coordinates delivery or pickup options. The system provides convenience to customers and streamlines operations. Customers can order sushi and Japanese cuisine from their homes or offices, increasing sales and satisfaction. The system also efficiently handles large volumes of orders and reduces human error, which not only reduces the burden on staff and increases efficiency, but also provides data on customer preferences to help Sushi Mentai customize products and promotions. In addition, the integration of loyalty programs and discounts encourages repeat business and builds customer loyalty to Sushi Mentai.

1.2 Requirement Gathering Techniques

Questionnaire

In requirement gathering techniques, questionnaire is one of the effective ways to collect the information from Sushi Mentai online food ordering system users. This technique will enable users to answer the question that is related to the system requirement via questionnaires distribution to a broad audience.

Time Effective

The technique of questionnaires can increase the time-effectiveness when collecting the response from the users. This is because the questionnaires can be distributed at once for large numbers of users at different places or platforms such as email, social media and online questionnaire tools. Meanwhile, it also indicates that they are able to complete the questionnaire simultaneously so the answer can be collected quickly.

Privacy Concern

When users are answering the questionnaires, they will feel more comfortable and safe as privacy is concerned and protected. Users can anonymously answer the questionnaire without specifying their name, IC number, address or other personal information unless it is crucial to collect that information for research only. Thus, it can directly prevent the questionnaire originator from revealing the users' sensitive information. Since other people will not be able to know the users who answered the questionnaire, the users can put down their concerns and physiological alert to answer all the questions honestly. As a result, we can also collect the most honest answer from the users for implementing the most appropriate online food ordering system.

Cost-Effectiveness

Since the questionnaire can be distributed in various way, one of the cost-effective way is via online and email questionnaires. This type of questionnaire distribution can reach a large audience at minimal cost as it does not require printing, postage and physical distribution. Besides, it can also reduce the labor costs or even not require labor costs because it does not need

to employ interviewers or helpers to find users and schedule the interviews for different users. Via distributing the questionnaires on online platforms, it can still obtain various responses from the users even without planning to find the right users. On the other hand, the same questionnaires can be reused multiple times for different users. This indicates that no extra questionnaire design cost is needed to design the questionnaire. Moreover, most of the questionnaire tools have provided automated data processing features. It can automatically analyze the response given by the users and even generate useful charts. Thus, it can effectively save the time and cost to execute analysis on the users' answers.

Standardization

When it comes to standardization of questions asked to the users, the questionnaire can easily achieve it by distributing the same content of the questionnaire to all of the users. Once the questionnaire questions are designed well, it can be used until the end of requirement gathering phase. This can promise that standardized format of question is applied and all the users are asked precisely the same questions placed identically. (David, 2020) Meanwhile, sufficient answers for the same question can be collected. Thus, we can analyze the responses uniformly and make a reliable comparison with other modes of questionnaire. (David, 2020) In order to further standardize the questionnaire, we can design some open-ended questions for giving the freedom to respondents to answer in free form. (Bhasin, 2020) Besides, we can also design closed-ended questions which obtain the answer of "Yes or No" only. This can ease the respondents to understand the question and answer required. This type of question can also enable the respondents to clearly express their positions.

Information Gathered

Via enabling users to answer the designed questionnaires, we are able to determine the method of online ordering food preferred by them nowadays. This information helps the system to develop and implement an efficient ordering operation which is familiar to the users. Users can directly apply their common habit of online ordering food in the system instead of wasting time for learning the new way of online ordering food from the beginning. Moreover, we have also collected the ratings given by the users which concerns the satisfaction of the current

implemented online food ordering methodology. Thus, we will be able to revamp the existing operation or plan a new preferred method of ordering food.

Besides, we have also gathered feedback which concerns the integrity of food menu information from the users through questionnaires. For example, users are not able to clearly view the price of each food item in the menu page. It will immediately distract the users from ordering the food as the uncertainty of food item information will easily cause the users not to take the risks to order food. Meanwhile, we have also gathered and analyzed the information about the possible improvement of the system which is desired by most of the users. This can help in building a more user-friendly and productive online food ordering system.

Observation

In order to gather the crucial and detailed requirements from the users, we have also applied observation as our requirement gathering technique. Observation is a methodology where the observer will spend time with a person or group of people to see how they perform their tasks in a real-life job setting. (Indeed Editorial Team, 2023) When the observation is undergoing, we have to be careful not to disturb the users' current task performance. We should always keep observing, take notes, remain unbiased and keep from making judgments. (Indeed Editorial Team, 2023)

Realistic Data Collection

Sometimes what the users say they do is definitely not what they actually do, it might cause some misinterpretation when we are analyzing the information gathered. Thus, observation is applied for avoiding the misinterpretation as we can actually observe the actions and activities of users who are dealing with the current system. We can gather the real data responded by the users via recording the actions, taking pictures or even asking them the questions on the spot. Via observing the real context where they perform their tasks, we will be able to gain the true understanding of what the actual issues they are facing and what improvements they need to increase efficiency and productivity. (Jama software, 2023)

Determination of Errors and Inefficiencies

Observation is not just for observing how the users complete their tasks using the current system, it also includes finding the inefficiencies and errors that users might overlook or consider insignificant. Although it might be a small system error, it is still able to cause a huge drop in the overall performance of the system. For example, users have to obviously take a long time when they want to checkout the food items and purchase for it due to the tedious steps required in the checkout pages and the checkout button is difficult to be found. Meanwhile, they often accidentally cancel their selected food items due to the system errors. This type of issue will affect the efficiency and effectiveness of the current system. Via observation, we can accurately identify the root cause of the problem and immediately implement a series of corrective action. We can even locate the page where the error is happening and know which action or behavior of users will cause the error.

Profound Requirement Description

Since users might not be able to fully describe their requirements and challenges faced in the current system during the interviews and questionnaires, observation is an effective way to determine the profound requirements by users based on how they interact with the system. During observation, users can operate the current system and show the location where the issue happens to the observers. Then, they can also provide their desired improvements to the observers with the help of the user interface shown on the screen. Eventually, we will be able to know what is the profound requirement needed by the users via observing and understanding the actions and behaviors of the users.

Detection of Emotion Changes

In some cases, users' facial expressions and body language can directly indicate their emotion when interacting with the current system. Those imperceptible characteristics may be able to represent users' frustration, satisfaction, confusion or confidence while using the system. For example, a user may show indecisive looks on his or her face when he or she is not sure which button they should click on in a confusing user interface. The observer can immediately determine that the user is not able to understand the instructions shown on the user interface and he or she is not satisfied with this. Although the user does not speak out anything due to the fear

of embarrassment, the observer still can roughly determine the user's inner thoughts via observing their facial expressions. Besides, we can also identify the user's certainty and uncertainty about a matter via observing their dexterity in eye and hand movements. If the users' movements are obviously slowed down in certain sites compared to other sites, it means that the users might not be familiar with the user interface or instructions stated by the system. Another reason might be that the users cannot clearly understand the information displayed on those sites when reading the instructions for the first time. They may need to read it multiple times to clearly understand what the instructions mean. All those detection of tiny changes can be crucial clues and direction for identifying the issues present in the current system and navigating the improvement process.

Information Gathered

Via effective observation, we can get to know the real issues faced by the current users. During the observation, we found that there is a lack of customer support designated for users when they are facing problems in the current system. Thus, the users are not able to immediately obtain the method to resolve the current problem, they can only get help from feedback provided by previous users using external search engines. It indicates that users are desiring to get sufficient customer support when dealing with the system.

Besides, we found out that there is a lack of availability of food items displayed in the menu. Based on the behavior and emotion changes of users which are observed by us, the users always have to access the detail page of each food item repeatedly to check its availability. It would bring a serious inefficiency to users and dissatisfaction of users which can be observed from their facial expression. In order to avoid this, we have concluded that users are requiring a more comprehensive feature in the menu list to display the availability status of each food item. Thus, they can directly check the status of each food item without repeatedly accessing the detail pages.

1.3 Problems of Existing System

1. Limited Online Ordering Channels

Currently, Sushi Mentai's **official website only offers a display function and does not support online ordering**. The only non-physical ordering channel available is via WhatsApp. This limitation poses several problems. Customers who prefer to order online have no choice but to use WhatsApp, which may not be their preferred platform. Orders placed through WhatsApp are prone to miscommunication due to the lack of a structured ordering system. Without predefined order forms or an integrated menu system, there is a higher chance of errors in order details, which can lead to incorrect or incomplete orders being prepared. Furthermore, customers might experience delays in receiving order confirmations or responses to inquiries. This delay is often because communication on WhatsApp relies on manual handling by restaurant staff, which can be slow and inconsistent, especially during peak hours. Such delays can lead to frustration among customers who expect quick and efficient service. In some cases, the lack of timely responses might even result in customers abandoning their orders altogether, leading to lost sales and decreased customer satisfaction.

2. Inadequate Customer Support

Sushi Mentai's current system **lacks an efficient and accessible customer support mechanism** for online orders. This can lead to several issues. Customers may find it difficult to get quick responses to their queries or resolve issues related to their orders. Without a dedicated support system, customers often experience delays in communication, which can result in frustration and a negative customer experience. The absence of a structured customer support system also makes it challenging for the restaurant to manage and track customer feedback effectively. Without a centralized way to log and address customer complaints, issues may go unresolved, leading to repeated problems and ongoing customer dissatisfaction. Furthermore, this lack of organization hinders the restaurant's ability to identify common issues and make necessary improvements to their service. As a result, customers may feel neglected and

undervalued, which can damage the restaurant's reputation and lead to a decline in repeat business.

3. Incomplete Menu Information

The online menu **does not list prices for all items**, which can cause uncertainty and hesitation among consumers. This issue results in a lack of transparency, as customers may feel uncertain about the cost of their order, which could discourage them from making a purchase. Without clear pricing, customers might spend more time deciding what to order or need to contact the restaurant for price inquiries, adding unnecessary steps to the process. This additional effort can be frustrating and may deter customers from completing their orders. Moreover, the absence of price information can make it difficult for customers to plan their meals within their budget, further complicating the ordering experience. The need to ask for prices can also slow down the ordering process and create a less efficient system overall. Consequently, customers might seek out alternative dining options that provide more transparent and straightforward pricing information, leading to a potential loss of sales for Sushi Mentai.

4. Poor Menu Layout

The layout of the online menu is problematic, as it **requires users to navigate between multiple pages to view different categories of food**. This can cause several issues. Constantly switching between pages can be annoying and time-consuming, leading to a poor user experience. Customers might abandon the ordering process due to the cumbersome navigation, resulting in lost sales. Additionally, users find it challenging to compare different food items, which can hinder their decision-making process. The inability to view and compare all options on a single page can lead to customer frustration and a decrease in overall satisfaction with the online ordering experience. A disjointed menu layout can also make it harder for customers to discover new items or promotions, limiting their exposure to the full range of the restaurant's offerings. Ultimately, a poorly designed menu can negatively impact the efficiency and enjoyment of the ordering process, driving customers away from using the online platform.

5. Lack of Availability Status

The menu **does not display the availability status of the items**, which can lead to significant problems. Customers may visit the restaurant expecting to order a specific dish, only to find it unavailable, leading to dissatisfaction. Customers who specifically visit the restaurant for a particular item might feel their time is wasted if it's not available. Repeated experiences of unavailability can lead to a negative perception of the restaurant's reliability and inventory management. This issue can cause customers to lose trust in the restaurant's ability to meet their expectations and may result in them choosing to dine elsewhere in the future. Additionally, the lack of availability information can disrupt the kitchen's workflow, as staff might need to frequently update customers on the status of certain dishes, leading to inefficiencies and potential errors. Providing real-time availability status could significantly enhance the customer experience by setting accurate expectations and reducing frustration.

1.4 Software Quality Attribute

1. Availability: The probability that a system will work as required when required during the period of a mission.

For an online food ordering system, particularly for a popular restaurant like Sushi Mentai, it is crucial that the system is available whenever customers want to place an order. This ensures that the business can operate smoothly without interruptions, which is essential especially during peak hours such as lunch and dinner times. High availability minimizes downtime, thus enhancing customer satisfaction and trust in Sushi Mentai.

2. Functional: The system will perform the functions that it is required to.

The Sushi Mentai online food ordering system must perform the required functions effectively, such as browsing the menu, placing orders, making payments, and tracking delivery status. If the system fails to perform any of these functions, it would directly impact the customer experience and operational efficiency of Sushi Mentai. Hence, ensuring the functionalities work as intended is vital for meeting customer expectations, satisfaction and business requirements.

3. Useable: Provides users with a satisfying experience (not a source of dissatisfaction).

A usable system ensures that customers have a satisfying experience while using the Sushi Mentai online food ordering system. This includes an intuitive user interface, easy navigation within the system, and a smooth ordering process. If the system is not user-friendly, customers might get frustrated and abandon their orders, leading to a loss in sales and potential repeat business.

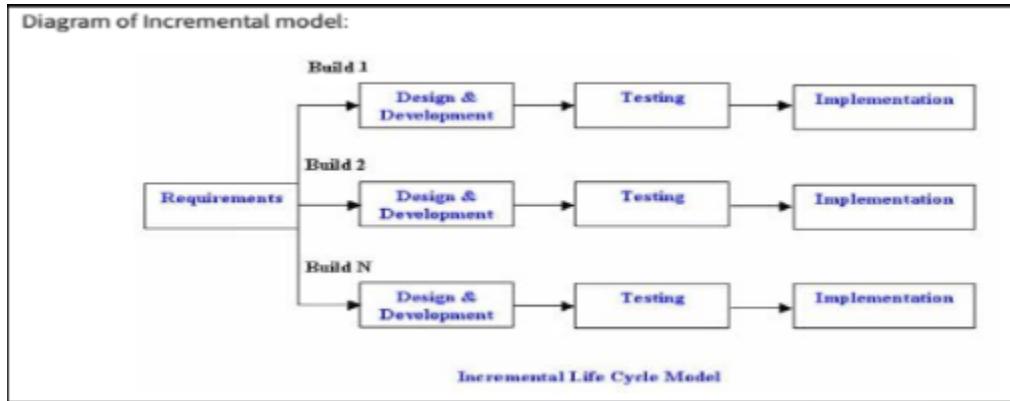
4. Reliable: Not prone to hardware or software failure, will deliver the functionality when the users want it.

Reliability is crucial for maintaining customer trust and satisfaction. The Sushi Mentai online food ordering system must be dependable, delivering the required functionality consistently whenever users interact with the system. Any failure in the system, whether due to hardware or software issues, can lead to missed orders, incorrect billing, or delays in delivery, all of which can negatively impact the restaurant's reputation and customers' satisfaction.

5. Flexible: Capable of being adapted to new users, to run in different countries or to be moved to a different platform.

The system should be adaptable to different new users, capable of running in different countries, and easily moveable to different platforms. Flexibility ensures that the system can accommodate diverse customer bases, including those from different regions with varying languages and cultural preferences.

1.5 Software Process Model



The Software Process Model that we have chosen for this project is the **Incremental Model**. In this model, requirements are broken down into multiple standalone modules of the software development cycle. Thus, this model requires an unambiguous and comprehensive system requirement to achieve the distribution of modules. In the Incremental Model, each module passes through the requirements, design, implementation and testing phases. Via the repeating of design, implementation and testing, the integrity of the system can be improved over time based on the defined requirements.

Since the Movement Control Order was ended in Malaysia, the overall consumer spending has started growing. This phenomenon was not only represented in shopping malls and travels, it also happened in restaurants. In Sushi Mentai, there was a gradual increase in customers coming to the restaurant to enjoy food and services. Thus, the business in Sushi Mentai was also boosting enormously as well. However, a long queue was inevitable most of the time outside the restaurant since all the seats inside were already filled. Customers can place the order only after having the seats, this causes serious time wasting. Although Sushi Mentai has applied online ordering and delivery through Whatsapp, this feature is not applicable for all branches and it is not a systematic

method in the long run. There is no user interface for users to choose food items, add to cart, place order and other crucial interactive functions. Customers can only place orders via entering the name of food items they want to order in the chat box manually. They also have to enter their home address manually to get the delivery service.

In order to avoid this inefficient method, Sushi Mentai wants to develop a systematic online ordering system. For the core functions, the system should **allow customers to view the item menu and register as a member**. It should also allow admin to **manage staff such as adding new staff and updating staff information**. The system should be done **within 10 weeks** for introducing the new system to the customers and enable them to start using the system. Sushi Mentai also plans to develop a **food ordering service, payment and table reservation** feature in the future. So, the customers can easily reserve their table and order the food using their mobile devices early, then make payment for the checkout order. After that, Sushi Mentai also wants to develop a **delivery service and membership functions for customers**. Customers can enjoy their food at their own places without coming to the restaurant. Not only that, they can also earn the membership points to redeem vouchers or gifts by ordering food via online. Additionally, the **report generation and promotion functions** will also be developed for the staff. So, they can generate the various business reports and create promotions using the system for acknowledging the customers about the ongoing events in the restaurant.

Comprehensive requirements of system

Based on the scenario stated above, there are clearly defined requirements for the Sushi Mentai online food ordering system. The core requirements are **allowing customers to use the fundamental food ordering functionality**. For example, viewing item menu, add item to cart, place order, online payment and view receipt. These functionalities are sufficient for customers to complete a whole ordering process from end-to-end even though it is still not perfect without secondary features.

For the secondary requirements, the system should **allow customers to enjoy the delivery service** provided by the system. After customers place their order, the deliverer

in Sushi Mentai will deliver the food directly to the customer's place based on the provided address. For the subsequent orders, the system can refer to the previously used address to execute delivery. Moreover, there is also another requirement which is **customer support live chat**. It is planned to be developed after finishing the development of the delivery feature. It provides customer service for solving the problems faced by customers via one-to-one online consultation. On the other hand, the **event posting function** will also be developed for staff to post any latest announcement or promotion. Meanwhile, the customers can view the announcement and get updated.

Quick Introduction into Market

Since the Sushi Mentai online food ordering system must be introduced into the market **within 10 weeks**, it is ideal for applying an incremental model to this system development. The **first increment** should be focusing on the **core requirement and functionalities of the system**. By excluding the secondary functionalities, the time required to develop the first increment can be shortened effectively. Thus, the customers can use the first incremental version of the system which **provides the view menu, add to cart, place order, payment, view receipts and other crucial features**. Meanwhile, the useful feedback can also be gathered from the user's experience for planning corrective actions and improvements.

For the **second increment**, the designer can start to design the prototyping and user interface for the **delivery feature**. Delivery feature is a sub requirement for the system which allows users to order the food and receive their food at doorstep. After developing the delivery feature, a series of testing, integration, user acceptance testing and feedback gathering will be executed to ensure the second incremental version of the system can be used in production.

For the **customer service live chat** and **event posting features**, the incremental model will still use the same strategies to integrate the features into the system when maintaining the excellent performance of the system. The more the increments, the higher the integrity of the Sushi Mentai online ordering system.

Part 2

2.1 Project Planning and Schedule

ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names	21	Aug '24	28	4
1	?	Planning and Requirement									
2	?	Planning	0.5 wks	Fri 2/8/24	Tue 6/8/24						
3	?	Requirement Gathering	0.5 wks	Tue 6/8/24	Thu 8/8/24	2					
4	?	Requirement Specification Analysis	0.5 wks	Fri 9/8/24	Tue 13/8/24	3					
5	?	Requirement Validation	0.5 wks	Tue 13/8/24	Thu 15/8/24	4					
6	?										
7	?	Build 1:									
8	?	Design and Development									
9	?	Product listing	3 wks	Fri 16/8/24	Thu 5/9/24	5					
10	?	Member registration	3 wks	Fri 16/8/24	Thu 5/9/24	5					
11	?	Staff management	1 wk	Fri 16/8/24	Thu 22/8/24	5					
12	?	Testing									
13	?	Unit Testing	1 wk	Fri 6/9/24	Thu 12/9/24	9,10,11					
14	?	Integration Testing	0.8 wks	Fri 13/9/24	Wed 18/9/24	13					
15	?	User Acceptance Testing (UAT)	0.2 wks	Thu 19/9/24	Thu 19/9/24	14					
16	?	Implementation									
17	?	Deploy into production	0.4 wks	Fri 20/9/24	Mon 23/9/24	15					
18	?	Testing in deployed environment	0.6 wks	Tue 24/9/24	Thu 26/9/24	17					
19	?										
20	?	Build 2:									
21	?	Design and Development									
Project: Project1 Date: Thu 8/8/24		Task		Inactive Summary		External Tasks					
		Split		Manual Task		External Milestone					
		Milestone		Duration-only		Deadline					
		Summary		Manual Summary Rollup		Progress					
		Project Summary		Manual Summary		Manual Progress					
		Inactive Task		Start-only							
		Inactive Milestone		Finish-only							

Page 1

ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names	21	28	Aug '24
22		Ordering	3 wks	Fri 27/9/24	Thu 17/10/24	18				
23		Payment	3 wks	Mon 7/10/24	Fri 25/10/24	22				
24		Table reservation	2 wks	Fri 27/9/24	Thu 10/10/24	18				
25	?	Testing								
26		Unit Testing	1 wk	Fri 8/11/24	Thu 14/11/24	24,22,23				
27		Integration Testing	0.8 wks	Fri 15/11/24	Wed 20/11/24	26				
28		User Acceptance Testing (UAT)	0.2 wks	Thu 21/11/24	Thu 21/11/24	27				
29	?	Implementation								
30		Deploy into production	0.4 wks	Fri 22/11/24	Mon 25/11/24	28				
31		Testing in deployed environment	0.6 wks	Tue 26/11/24	Thu 28/11/24	30				
32	?									
33	?	Build 3:								
34	?	Design and Development								
35		Delivery	1 wk	Fri 29/11/24	Thu 5/12/24	31				
36		Report generation	1 wk	Fri 29/11/24	Thu 5/12/24	31				
37		Membership	3 wks	Fri 29/11/24	Thu 19/12/24	31				
38		Promotion	2 wks	Fri 29/11/24	Thu 12/12/24	31				
39	?	Testing								
40		Unit Testing	1 wk	Fri 20/12/24	Thu 26/12/24	34,35,36,37				
41		Integration Testing	0.8 wks	Fri 27/12/24	Wed 1/1/25	40				
42		User Acceptance Testing (UAT)	0.2 wks	Thu 2/1/25	Thu 2/1/25	41				

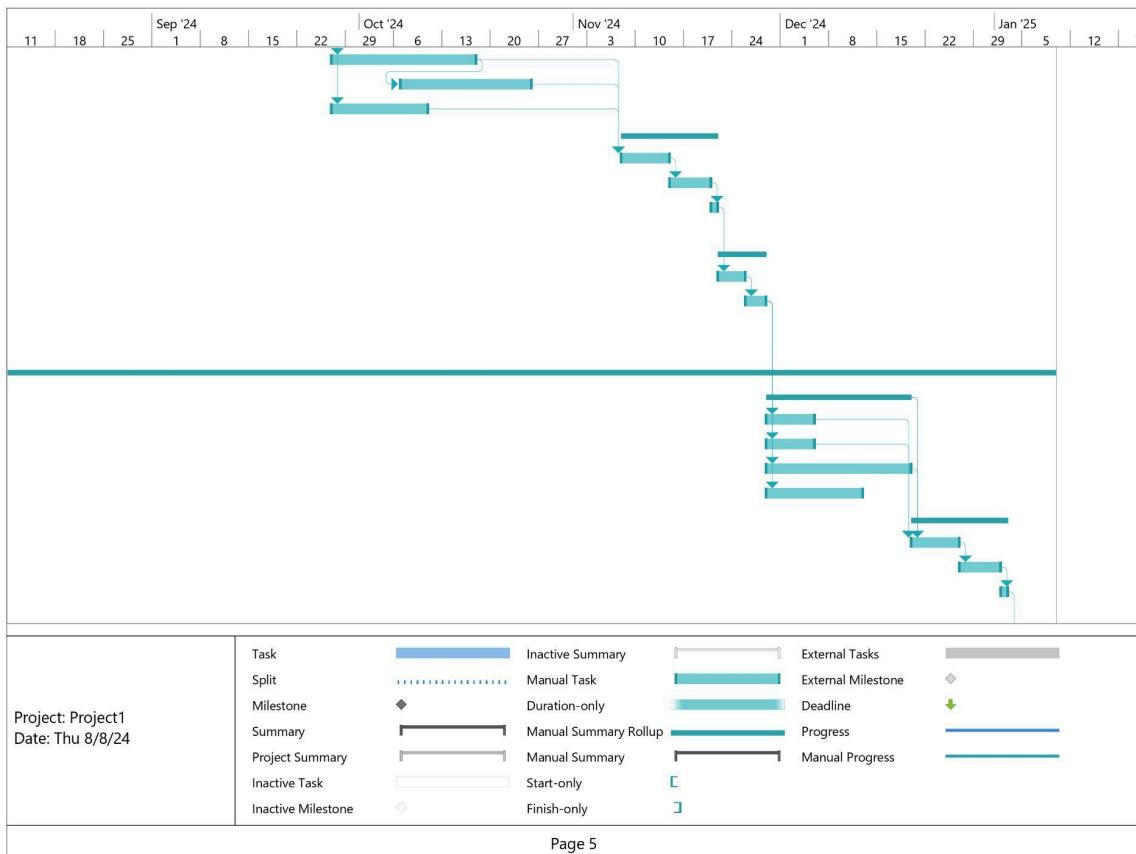
Project: Project1 Date: Thu 8/8/24	Task	Inactive Summary	External Tasks
	Split	Manual Task	External Milestone
	Milestone	Duration-only	Deadline
	Summary	Manual Summary Rollup	Progress
	Project Summary	Manual Summary	Manual Progress
	Inactive Task	Start-only	C
	Inactive Milestone	Finish-only	D

Page 2

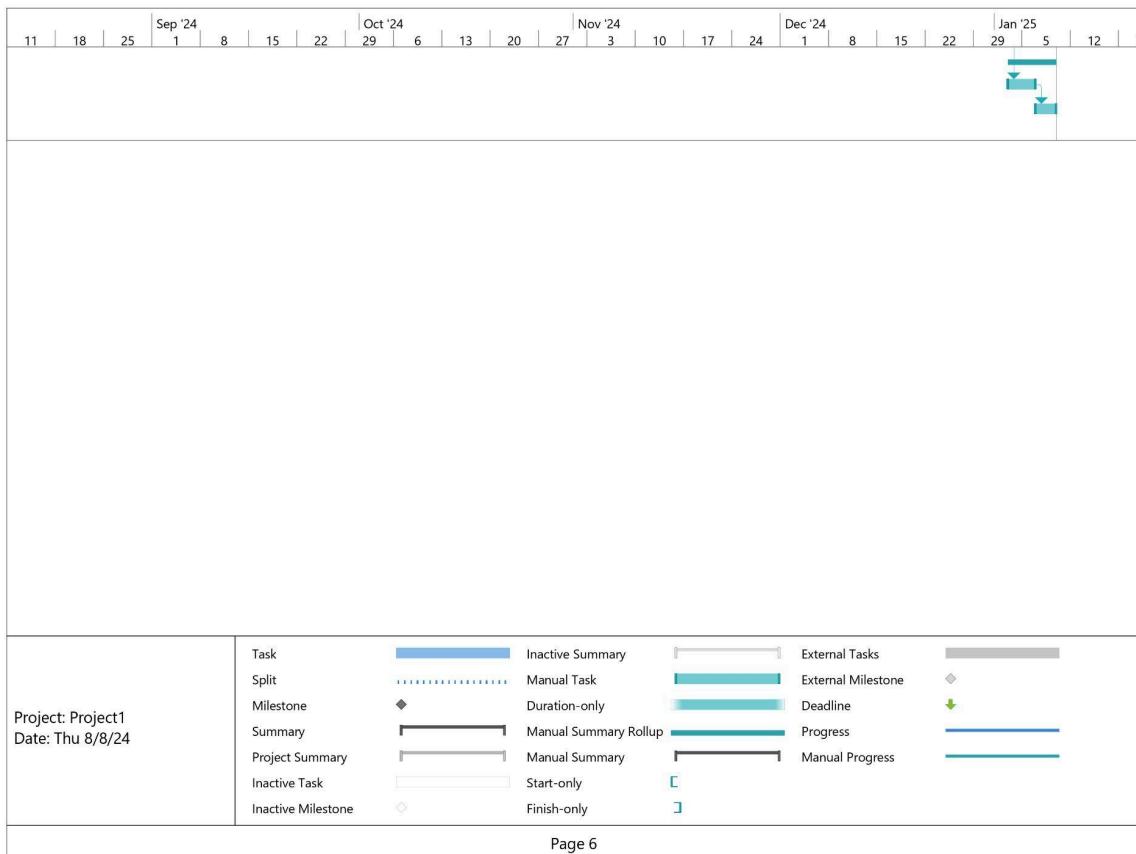
ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names	Aug '24
								21 28 4
43	Implementation							
44		Deploy into production	0.4 wks	Fri 3/1/25	Mon 6/1/25	42		
45		Testing in deployed environment	0.6 wks	Tue 7/1/25	Thu 9/1/25	44		

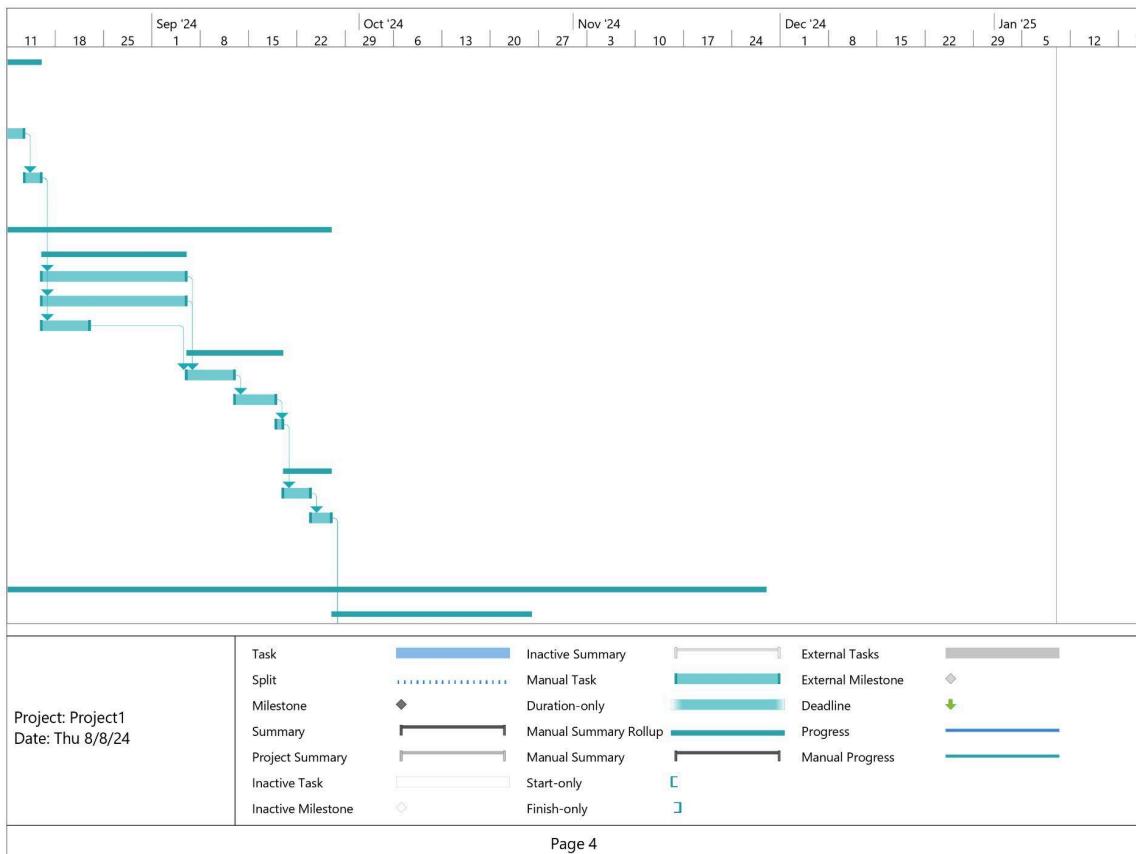
Project: Project1 Date: Thu 8/8/24	<p>Task Inactive Summary External Tasks</p> <p>Split Manual Task External Milestone</p> <p>Milestone Duration-only Deadline</p> <p>Summary Manual Summary Rollup Progress</p> <p>Project Summary Manual Summary</p> <p>Inactive Task Start-only Manual Progress</p> <p>Inactive Milestone Finish-only </p>
---------------------------------------	---

Page 3



Page 5





2.2 System Requirement (Functional Requirements)

Order (Ong Yi Xin)

1.0 Add to Cart Function

- 1.1 The system shall allow users to select items from the menu to add to their cart.
- 1.2 The system shall prompt users to specify the quantity of each selected item.
- 1.3 The system shall allow users to add special instructions for each item .
- 1.4 The system shall add the selected items and quantities to the user's cart.
- 1.5 The system shall display a confirmation message when items are successfully added to the cart.
- 1.6 The system shall update the cart summary to reflect the newly added items.
- 1.7 The system shall display the subtotal cost of items in the cart.
- 1.8 The system shall support updating item quantities or removing items from the cart after addition.

2.0 View Cart Function

- 2.1 The system shall display a detailed list of items currently in the cart, including names, quantities, and prices.
- 2.2 The system shall show the total cost of the items in the cart.
- 2.3 The system shall allow users to modify quantities of items in the cart.
- 2.4 The system shall allow users to remove items from the cart.
- 2.5 The system shall show the updated total cost when items are added or removed.
- 2.6 The system shall provide an option to apply discount codes or promotional offers.
- 2.7 The system shall allow users to view and edit special instructions for each item in the cart.
- 2.8 The system shall prompt users to review their order before proceeding to checkout.

3.0 Place Order Function

- 3.1 The system shall prompt users to review their complete order before submission.
- 3.2 The system shall allow users to confirm and submit their order.
- 3.3 The system shall provide an order confirmation with a unique order ID.

- 3.4 The system shall display an estimated preparation time for the order.
- 3.5 The system shall allow users to select payment methods (e.g., credit card, cash).
- 3.6 The system shall provide a summary of payment details and total cost.
- 3.7 The system shall confirm successful order submission and provide an option to print or save the order receipt.
- 3.8 The system shall update the order status to "Processing" and notify users of the status.

4.0 Cancel Order Function

- 4.1 The system shall allow users to cancel their order before it is being processed.
- 4.2 The system shall prompt users to confirm the cancellation request.
- 4.3 The system shall update the order status to "Canceled" upon successful cancellation.
- 4.4 The system shall provide a confirmation message once the order is canceled.
- 4.5 The system shall handle refunds if applicable, based on the restaurant's policy.
- 4.6 The system shall notify users of any potential issues or restrictions regarding the cancellation.
- 4.7 The system shall allow users to view the cancellation details and reason.

Delivery and Track Order (Ong Yi Xin)

1.0 Schedule Delivery

- 1.1 The system shall allow users to choose a delivery date from available slots.
- 1.2 The system shall allow users to choose a delivery time from available slots.
- 1.3 The system shall validate that the selected delivery time is within the restaurant's operational hours.
- 1.4 The system shall allow users to schedule delivery for a future date or time.
- 1.5 The system shall confirm the scheduled delivery time and date to the user.
- 1.6 The system shall provide an option for users to reschedule the delivery time before the order is processed.
- 1.7 The system shall send a confirmation email with the scheduled delivery details.
- 1.8 The system shall notify users of any delays to the scheduled delivery time.
- 1.9 The system shall notify users of any changes to the scheduled delivery time.
- 1.10 The system shall allow users to select delivery preferences.

2.0 Enter Delivery Address

- 2.1 The system shall allow users to enter a new delivery address.
- 2.2 The system shall allow users to select from saved addresses.
- 2.3 The system shall validate the address format to ensure it is complete and accurate.
- 2.4 The system shall provide address suggestions based on user input.
- 2.5 The system shall provide address auto-completion based on user input.
- 2.6 The system shall allow users to edit the delivery address before finalizing the order.
- 2.7 The system shall save frequently used addresses for easy future use.
- 2.8 The system shall display the entered address for user verification.
- 2.9 The system shall confirm the entered address for user verification before proceeding.
- 2.10 The system shall notify users if the entered address is outside the delivery area.
- 2.11 The system shall provide an option to specify delivery instructions.

3.0 Delivery Charges

- 3.1 The system shall calculate delivery charges based on the delivery address and order amount.

- 3.2 The system shall display the calculated delivery charges before the order is finalized.
- 3.3 The system shall apply any available promotional discounts to the delivery charges.
- 3.4 The system shall update the total cost to include delivery charges.
- 3.5 The system shall reflect the updated total cost in the order summary.
- 3.6 The system shall provide an option to choose a delivery method.
- 3.7 The system shall notify users for additional charges for specific delivery options or addresses.
- 3.8 The system shall include a breakdown of delivery charges in the final order summary.

4.0 View Order Status

- 4.1 The system shall display the current status of the order.
- 4.2 The system shall provide an estimated time of arrival for the order.
- 4.3 The system shall update the order status in real-time.
- 4.4 The system shall provide a visual representation of the order status.
- 4.5 The system shall notify users of significant status changes via email.
- 4.6 The system shall allow users to view historical status changes for their orders.
- 4.7 The system shall allow users to view timestamps for their orders.
- 4.8 The system shall provide detailed information about each status update, including reasons for delays if applicable.
- 4.9 The system shall allow users to contact support if there are discrepancies or issues with the order status.

5.0 Order History

- 5.1 The system shall provide users with access to their past order.
- 5.2 The system shall display details for each past order.
- 5.3 The system shall allow users to search and filter their order history by date.
- 5.4 The system shall allow users to search and filter their order history by status.
- 5.5 The system shall provide a summary of previous orders with the option to view more details.
- 5.6 The system shall display order history in a list.
- 5.7 The system shall allow users to re-order items from their order history.

- 5.8 The system shall provide users with options to export their order history.
- 5.9 The system shall ensure that the order history is secure and accessible only to the user.

6.0 Order Tracking

- 6.1 The system shall provide a unique tracking number for each order.
- 6.2 The system shall display real-time tracking information provided by the delivery service.
- 6.3 The system shall update tracking information periodically.
- 6.4 The system shall provide an estimated delivery route.
- 6.5 The system shall allow users to view detailed tracking history.
- 6.6 The system shall allow users to view delivery updates.
- 6.7 The system shall notify users of any significant changes or delays in the tracking status.
- 6.8 The system shall notify users of delays in the tracking status.
- 6.9 The system shall allow users to report issues with the tracking information.
- 6.10 The system shall ensure that tracking information is accurate and up-to-date.

Product (Chia Ming Yi)

1.0 View Product

- 1.1 The system shall display a list of all products.
- 1.1 The system shall display detailed information for each product, including name, description, price, and images.
- 1.2 The system shall display the products in categories.
- 1.3 The system shall allow users to enter product name to search for products.
- 1.4 The system shall allow users to filter the product items displayed in product list.

2.0 Create Product

- 2.1 The system shall allow staff to create new products.
- 2.2 The system shall allow staff to enter a new product name, cost and price.
- 2.3 The system shall allow staff to upload a product image.
- 2.4 The system shall check the format of the new products' data.
- 2.5 The system shall display the error messages when the users' input is incorrect.
- 2.6 The system shall update the system database when there are new or updated product details.

3.0 Update Product

- 3.1 The system shall show the current details of the selected product.
- 3.2 The system shall allow staff to modify the existing products' name, images, quantity, cost and price.
- 3.3 The system shall check the format and the size of uploaded files.
- 3.4 The system shall validate the format of the updated product details.
- 3.5 The system shall display error messages when there is incorrect users' input.
- 3.6 The system shall display confirmation dialog to users before saving updated product details.
- 3.7 The system shall allow users to confirm the product update action.
- 3.8 The system shall display successful messages to user after successful updating product details.
- 3.9 The system shall allow staff to delete products.

- 3.10 The system shall allow staff to list or delist the products.
- 3.11 The system shall update the system database when there are new or updated product details.

4.0 Navigate the Menu of Products

- 4.1 The system shall display real-time product availability status.
- 4.2 The system shall allow users to exclude products that contain their allergies.
- 4.3 The system shall allow users to browse products by categories.
- 4.4 The system shall allow users to search for products by any keywords.
- 4.5 The system shall allow users to filter products by price range and category.
- 4.6 The system shall display search results based on user queries.
- 4.7 The system shall display product recommendations based on user preferences and purchase history.
- 4.8 The system shall allow users to view the product details page for the selected product.
- 4.9 The system shall display other related products on the product detail page.

5.0 Post Reviews and Ratings

- 5.1 The system shall allow users to submit reviews for products.
- 5.2 The system shall allow users to rate products.
- 5.3 The system shall display product reviews and ratings to users.
- 5.4 The system shall check the validity of product reviews.
- 5.5 The system shall allow users to upload multiple images or video files.
- 5.6 The system shall validate the format and size of the uploaded files.
- 5.7 The system shall allow staff to delete customers' reviews.

Payment (Chia Ming Yi)

1.0 Authenticate user

- 1.1 The system shall authenticate users before allowing access to the payment module.
- 1.2 The system shall support multiple authentication methods.

2.0 Process payment

- 2.1 The system shall support payments in multiple currencies.
- 2.2 The system shall display payment amounts in the user's preferred currency.
- 2.3 The system shall allow users to select their preferred currency.
- 2.4 The system shall calculate the total payment amount based on the selected order item.
- 2.5 The system shall allow users to enter their bank account ID and password.
- 2.6 The system shall check the validity of bank account ID and password.
- 2.7 The system shall display the error messages when users' provided information is incorrect.
- 2.8 The system shall allow users to select "Yes" or "No" option before confirming payment.
- 2.9 The system shall display successful messages to users after completing payment.

3.0 Generate receipts and notifications

- 3.1 The system shall generate receipts for successful payments.
- 3.2 The system shall allow users to download the receipts.
- 3.3 The system shall send email receipts to users.
- 3.4 The system shall push notifications for completed transactions.
- 3.5 The system shall send SMS confirmations for completed transactions.

4.0 View transaction history

- 4.1 The system shall allow users to view their transaction history.
- 4.2 The system shall allow users to filter the transaction history.
- 4.3 The system shall allow users to search for transactions.

5.0 Process refunds

- 5.1 The system shall allow users to request refunds within 12 hours.
- 5.2 The system shall allow users to enter reason when requesting refunds.
- 5.3 The system shall check the validity of the reason field.
- 5.4 The system shall allow users to upload image and video files.
- 5.5 The system shall validate the format of the uploaded files.
- 5.6 The system shall display error messages when users have entered incorrect data.
- 5.7 The system shall allow staff to view user refund requests.
- 5.8 The system shall allow staff to search the refund records.
- 5.9 The system shall allow staff to approve or deny user refund requests.
- 5.10 The system shall allow staff to enter remarks for refund records.
- 5.11 The system shall display the request status to the user.
- 5.12 The system shall update the refund status after approving or denying user refund requests.
- 5.13 The system shall automatically transfer the money to users' account once the refund request has been approved.

Table Reservation (Lim Jun Wei)

1.0 Reserve table in restaurant

- 1.1 The system shall display the available tables, time slots for the selected date and time and available location of restaurants.
- 1.2 The system shall allow users to enter the number of customers for each reservation.
- 1.3 The system shall update the table availability from time to time.
- 1.4 The system shall allow users to enter their mobile number.
- 1.5 The system shall verify the mobile number.
- 1.6 The system shall allow users to select the location of restaurants.
- 1.7 The system shall verify the availability of the restaurants.
- 1.8 The system shall allow users to select the available time slot for each reservation.
- 1.9 The system shall display the error messages when users' input is wrong.
- 1.10 The system shall send the reservation confirmation message to the users.

2.0 Track the table reservation

- 2.1 The system shall allow users to track the current status of their reservation.
- 2.2 The system shall allow users to view the reservation details.
- 2.3 The system shall update the reservation status to the users real-time.
- 2.4 The system shall provide the phone number and email address as contact details to users.
- 2.5 The system shall send notification to the users before 1 hour from the table reservation time.
- 2.6 The system shall show the location of the restaurant which has been reserved by the users.
- 2.7 The system shall display the table number reserved by the users.

3.0 Update table reservation

- 3.1 The system shall prevent users from updating the table reservation which has been created over 3 hours.
- 3.2 The system shall display the reservation update form to the users.

3.3 The system shall allow users to enter the mobile number, date and time of reservation and location of the restaurant.

3.4 The system shall check the validity of the mobile number, date and time of reservation and location of the restaurant.

3.5 The system shall display the confirmation dialog to the users.

3.6 The system shall update the latest reservation information to the system database.

3.7 The system shall send the successful update reservation notification to users via email or SMS.

3.8 The system shall allow users to cancel the reservation.

Member Registration (Lim Jun Wei)

1.0 Register as member

1.1 The system shall allow users to enter their name, mobile number, email address and password.

1.2 The system shall verify the users' mobile number.

1.3 The system shall check the valid format of users' email address.

1.4 The system shall check the validity of users' passwords.

1.5 The system shall prompt error messages and remind the users to re-enter the valid input.

1.6 The system shall send verification email or One-Time-Password to users' email or SMS.

1.7 The system shall require users to accept the terms and conditions and privacy policy before completing registration.

1.8 The system shall store member's registration details into the system database.

1.9 The system shall notify the users when they have registered as members successfully.

2.0 Integrating with social media platforms

2.1 The system shall allow users to choose the social media platform account to be integrated with.

2.2 The system shall redirect users to the social media platform login page.

2.3 The system shall allow users to enter the email address, username and password for the social media account.

2.4 The system shall check the validity of the media social account credentials input.

2.5 The system shall display error messages and prompt users to re-enter the valid credentials if they have provided incorrect credentials.

2.6 The system shall store the integrated social media account credentials into the system database.

2.7 The system shall display the successful integration message to users via email.

3.0 Recover password

3.1 The system shall allow users to enter their mobile phone number or email address.

3.2 The system shall check whether the mobile phone number or email address exists in the system database.

3.3 The system shall display error messages if the mobile phone number or email address is not valid.

3.4 The system shall send verification email or One-Time-Password via SMS to users.

3.5 The system shall allow users to click the verification link embedded in email.

3.6 The system shall allow users to enter One-Time-Password in the provided form.

3.7 The system shall verify the One-Time-Password entered by users.

3.8 The system shall refuse the One-Time-Password if it is incorrect or expired.

3.9 The system shall allow users to enter the new password for their accounts.

3.10 The system shall display a successful message to users via email or SMS if their password has been recovered successfully.

Membership (ian kai)

1.0 Update Membership Points Balance

1.1 The system shall update the user's membership points balance upon receiving successful order information from the payment module.

1.1.1 The system shall verify the accuracy of the points calculation based on a predefined rule of how many points per currency unit.

2.0 View Membership Points Balance

2.1 The system shall allow users to view their current membership points balance.

2.2 The system shall show the expiration date of how many points when user hovered over the points balance

2.3 The system shall show alerts when the user has points that are about to expire.

3.0 View Points History

3.1 The system shall allow users to view a detailed points history of points earned and points redeemed.

3.1.1 The detailed points history shall be displayed as a table with the columns as "Date", "Type", "Action", "Points", and "Status".

3.2 The system shall allow users to filter the detailed points history by date.

3.3 The system shall provide advanced filtering options including filter by type (earned/redeemed), status (approved, pending), and points amount.

3.4 The system shall allow users to sort the detailed points history by date, type, points amount, and status.

4.0 Display Voucher Catalog

4.1 The system shall display a catalog of vouchers available for redemption using membership points.

4.2 The system shall allow users to browse through the voucher catalog.

4.3 The system shall allow users to view detailed information about each voucher, including points required, validity period, and terms of use.

4.4 The system shall allow users to search vouchers by name.

4.5 The system shall allow users to filter vouchers by points required and expiration date.

4.6 The system shall update the voucher catalog in real-time to reflect availability and expiration of vouchers.

5.0 Redeem Vouchers

5.1 The system shall allow users to redeem vouchers using their membership points.

5.2 The system shall verify the availability and validity of the selected voucher before confirming redemption.

- 5.3 The system shall display an appropriate error message when the users have insufficient points for the selected redemption.
- 5.4 The system shall deduct the appropriate number of points from the user's membership points balance upon successful redemption.
- 5.5 The system shall display a confirmation message when the voucher is successful redeemed.

Promotion (ian kai)

1.0 Display Promotion List

- 1.1 The system shall display a list of promotions for users to view.
- 1.2 The system shall display additional buttons only for staff to add, modify, and remove promotions.

2.0 Add New Promotion

- 2.1 The system shall allow staff to add a new promotion to the system.
- 2.2 The system shall allow staff to specify key promotion details such as start date, end date, eligibility criterias, discount value, associated products, and redemption limits.
- 2.3 The system shall display an error message if any mandatory promotion details are missing before adding the promotion.
 - 2.3.1 The mandatory promotion details include start date, end date, discount values, and associated products.
 - 2.3.2 The non-mandatory promotion details including eligibility criterias and redemption limits, shall be defaulted to anyone is eligible and no redemption limits, respectively.
- 2.4 The system shall display a confirmation message when a promotion is successfully added.

3.0 Modify Existing Promotion

- 3.1 The system shall allow staff to search for existing promotions by name.
- 3.2 The system shall allow staff to filter existing promotions by validity, discount values, and associated products.
 - 3.2.1 The promotion's validity shall be determined by the start date, end date, and redemption limit, and associated products quantity.
- 3.3 The system shall allow staff to select an existing promotion for modification.
- 3.4 The system shall allow staff to modify the existing promotion's details including start date, end date, eligibility criteria, discount value, associated products, and redemption limits.

3.5 The system shall display an error message if mandatory promotion details are missing during modification.

3.5.1 The mandatory promotion details include start date, end date, discount values, and associated products.

3.5.2 The non-mandatory promotion details including eligibility criterias and redemption limits, shall be defaulted to anyone is eligible and no redemption limits, respectively.

3.6 The system shall display a confirmation message when a promotion is successfully modified.

4.0 Remove Existing Promotion

4.1 The system shall allow staff to search for existing promotions by name.

4.2 The system shall allow staff to filter existing promotions by validity, discount values, and associated products.

4.2.1 The promotion's validity shall be determined by the start date, end date, and redemption limit, and associated products quantity.

4.3 The system shall allow staff to select existing promotions for removal.

4.4 The system shall display a confirmation prompt before removing a promotion.

4.3 The system shall display a confirmation message when a promotion is successfully removed.

5.0 Promotion Notifications

5.1 The system shall notify users of upcoming promotions that are about to start.

5.2 The system shall notify users of ongoing promotions that are about to expire.

5.3 The system shall send notifications of promotions only to eligible users.

2.3 Non-Functional Requirements

Product

1.0 Availability

- 1.1 The new system shall be available 99.95% of the time, ensuring minimal downtime.
- 1.2 The new system's scheduled maintenance shall be limited to non-peak hours and should not exceed 2 hours per month.
- 1.3 In the event of unexpected server downtime, the system should recover and be fully operational within 5 minutes.

2.0 Functional

- 2.1 The new system shall ensure that all functionalities are accessible and functioning as expected across all platforms and devices, including different operating systems and different device types.

3.0 Useable

- 3.1 The new system shall have a user interface that is intuitive and easy to use, allowing even novice users to make orders without requiring external assistance.
- 3.2 The new system shall maintain consistency in user interface design across all pages and platforms, including mobile devices and desktop devices.

4.0 Reliable

- 4.1 The new system shall be able to handle up to 20,000 concurrent users without noticeable degradation of performance.
- 4.2 The system shall ensure data integrity during payment transactions, with no loss of information even in case of system failure.

5.0 Flexible.

- 5.1 The system shall support localization, enabling it to be run seamlessly in varying countries, by allowing easy translation of text, adaption to date format, and currency conversions.

Organization

Organization: delivery, implementation, standards

1. The system should be developed using PHP programming language with Laravel framework.
2. The system should use the MySQL database management system to store system data.

External

1. The system must be able to process payments through major payment gateways like PayPal and Stripe.
2. The system shall ensure compliance with the Payment Card Industry Data Security Standard (PCI DSS) for handling payment information.

2.4 View-Point Oriented Analysis

Customer (Ong Yi Xin)

The customer is the primary user of the online food ordering system. They are the end users who interact directly with the system, performing basic tasks such as navigating menus, placing orders, making payments, and tracking deliveries. They are also the **decision makers**; customers decide what to order, when to order it, and how to pay for it, and are the driving force behind the core operations of the system. Customers are also the **feedback providers for the system**, and their experience and feedback are critical to the continuous improvement of the system, influencing updates and enhancements.

Customers will use the system to **place sushi orders**, making them the focal point of the entire ordering process. After placing the order they select and execute a payment method to ensure the transaction is complete. Customers **use the system's membership features**, including earning rewards, viewing promotions, and managing personal profiles. Customers' perceptions of the system's ease of use, functionality and responsiveness have a direct impact on their satisfaction and likelihood of repeat patronage. Therefore, when designing and following the system, you need to ensure the customer experience.

The first point of importance of the customer perspective is that it has a **great impact on system usability and design**. Based on the fact that the customer is the primary system user, the system design must take into account the customer's needs and preferences, ensuring that navigation is intuitive, information is easy to access, and actions, such as placing an order or making a payment, must be simple and straightforward. Customers also **make systems more personal**, and designers or merchants may personalize functions to accommodate customer preferences, such as remembering past orders or offering tailored promotions, which can enhance the user experience and encourage repeat business.

Customers also **generate revenue for the business**. They interact with the system by ordering, paying, and participating in promotions, becoming the main source of revenue for the organization. Customer loyalty and repeat business are also very important. The system's

membership features, such as loyalty points and exclusive promotions, are designed to retain customers and cultivate brand loyalty. The customer perspective also helps Sushi Mentai's system differentiate from other systems, and a system that effectively meets customer needs can make Sushi Mentai stand out from competitors. This includes offering unique features such as real-time order tracking, multiple payment methods, and personalized promotions.

Manager (Lim Jun Wei)

Manager is one of the crucial indirect viewpoints for the online food ordering system. He will **oversee the weekly, monthly or annual analysis report** prepared by the subordinates. For example, analysis reports about sales trends, customer satisfaction levels and operational efficiency. Instead of directly interacting with the system, the manager will review the summary represented by the subordinates to understand the current situation faced by the system in a quick glance. Thus, he can easily **monitor and evaluate the performance of the system and organization**. Meanwhile, he is also able to draw up an appropriate plan or precautionary measures based on the current status acknowledged from the report to improve the comprehensive performance of the system and organization.

Besides, a manager's decision can also affect most of the trends of user interface design in the online food ordering system. At the beginning, the manager will meet with the UI designers to **discuss and decide the direction of the system user interface design**. After the system has been introduced into the market, he will regularly carry out conferences with UI designers and other related subordinates to probe into the effectiveness of the user interface design via overseeing the user experience survey report. If the result did not achieve the expectation of the manager, he will continue to discuss with UI designers to **plan an improvement action on creating more user-friendly interface design** for better user experience. If the result has achieved the expectation, they will discuss creating more innovation strategies to provide more constructive design for attracting more users to use the system.

In order to allocate resources fairly within the organization and system, the manager is taking an essential responsibility to plan a reasonable scheme via collaborating with his employees. Before making any decision, he will overview all the useful reports such as the human resource monthly

report and finance report to determine the current status of the organization and system overall. Then, the manager will **decide the resources allocation for each section of the organization** such as system maintenance cost, human resource allocation and time scheduling for different projects. Via a reasonable allocation of resources, the productivity and effectiveness of organization can be optimized as sufficient resources can be utilized by each section in organizations. Meanwhile, the system usability can also be improved or maintained as well.

Before the execution of any planning to develop the system, the manager has to **specify the direction of the strategies** such as standardization of programming language used and pace of the system development for each increment. Thus, system consistency can be promised when ensuring every incremental version of the system can be delivered on time as well. Not only that, the manager also takes the role to regularly be updated with the progress of every project. When there is a bottleneck faced during implementation of a project plan, the manager can immediately take actions to **make adjustments on the scope or requirements of the projects**. Thus, all the project plans can be accomplished successfully in the right direction. Via a good guidance of development and evolution of the system, the reputation of the online food ordering system will be increased and more market share can be gained.

Auditor (Chia Ming Yi)

The auditors play a crucial role in ensuring that Sushi Mentai's food ordering system operates smoothly and in full compliance with all relevant regulations. Their primary responsibility is to **ensure that the system complies with local and international laws**, particularly in areas such as data protection and food safety. By ensuring that customer data is handled securely and that the restaurant meets hygiene standards, the Auditor helps Sushi Mentai avoid legal penalties and maintains a trustworthy reputation, which is essential for the restaurant's continued success.

Another key aspect of the auditor's role is **financial oversight**. They carefully **review the financial transactions processed** by the ordering system to ensure that sales, payments, discounts and refunds are accurately recorded and accounted for. This oversight is critical to maintaining accurate financial statements, which are essential not only for tax reporting and financial planning, but also to prevent fraud or financial discrepancies that could harm the business.

Security is also an important priority for auditors. They regularly **assess the security measures in place**, focusing on how customer information is stored and transmitted, particularly during payment processing. By identifying and addressing potential vulnerabilities, the Auditor helps protect Sushi Mentai from data breaches that could result in significant financial loss and damage to the brand's reputation.

Operational efficiency is another area where the auditor's expertise is invaluable. They **evaluate the entire order process**, from the moment an order is placed through to fulfillment and delivery, identifying any inefficiencies or bottlenecks that could slow down operations. By improving efficiency, Sushi Mentai can ensure timely delivery of fresh sushi, increase customer satisfaction and reduce waste, all of which are critical to the restaurant's success in a competitive market.

Finally, the auditor is responsible for **risk management**. They **identify potential risks** within the food ordering system, such as technical failures, supply chain disruptions or food safety issues. By proactively addressing these risks, the auditor helps Sushi Mentai maintain smooth

operations, ensuring that the restaurant can continue to provide quality service and food to its customers without interruption.

Marketing Staff (Ng Ian Kai)

The importance of the marketing staff's view point is in the **planning and designing of marketing strategies** for the online food ordering system, with the goal of **attracting more customers and retaining existing returning customers**. Furthermore, the marketing staff's involvement in shaping the **strategic planning aspects of marketing** ensures that the restaurant and the system **remain competitive in a rapidly evolving market**. Their ability to **anticipate customer needs and respond with targeted marketing** and rewarding helps **maintain the system's relevance and appeal**, contributing significantly to overall success of the online food ordering system. Their insights are instrumental in designing benefits or events that **enhance customer loyalty**.

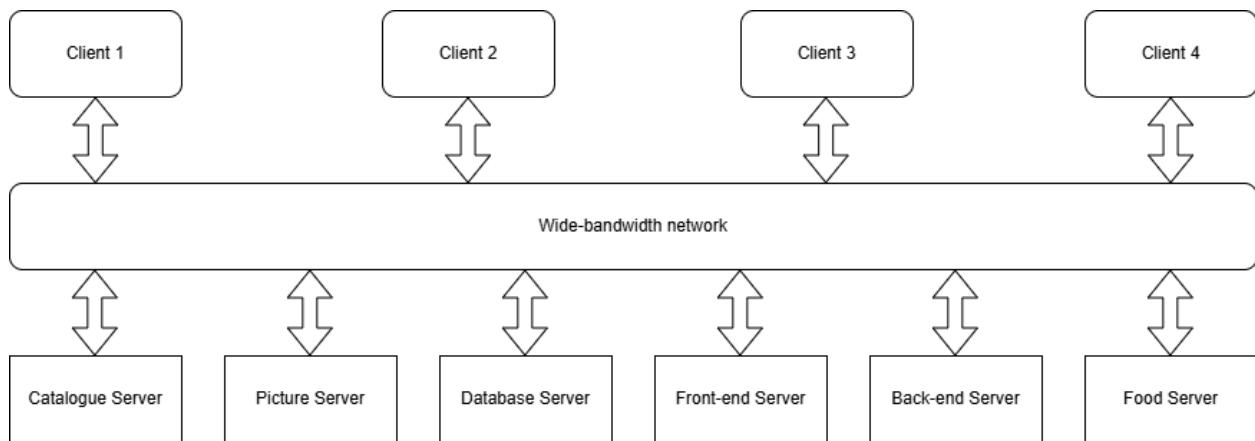
For example, the marketing staff proposed a membership points reward system that **incentivizes customers to order and make payments through the online food ordering system**. Customers can accumulate points with each successful order, which can then be redeemed for vouchers offering discounts on future orders or even for specific food products. This strategy not only **encourages returning customers** but also **fosters a sense of loyalty** among customers, making them more likely to continue using the system to make orders.

In addition, the marketing staff plays a key role in **planning promotions and events that drive customer traffic**. While they may not directly interact with the system to implement these promotions, they collaborate closely with the system administrator who is responsible for managing the system so that the planned promotions can be added into the system. This collaboration ensures that **marketing strategies are effectively integrated** into the platform, allowing the system to respond to market trends.

Part 3

3.1 An Architectural Design

Client-Server Model



Catalogue Server

The catalogue server is aiming for providing services to customers to **view the available food items in restaurants**. Meanwhile, it also provides usable services to staff to **create, view, modify and delete food items data**. From the view of customers, they can access the catalogue server when they want to order food items in the food menu interface via mobile devices over the network. For staff, they can access the catalogue server when there are new food items introduced by the restaurant (adding new food items). They can also access it when they need to make any update on the current existing menu item details. When there are any food items being taken off the shelves, staff will have to delete the food items from the catalogue server.

Picture Server

Picture server is also an important part in client-server model as it **hosts and serves all images related to the online food ordering system**. Before the customers select and buy the food items, they will need to view the reference picture of the food items. So, they can make sure that they are looking for the right food items before making the order. In order to handle this situation, a

picture server can be used to handle the food menu item images offered by the restaurant. It can **serve the images and present it on the customers' food menu page**. Meanwhile, the picture server can also be used for **storing the latest restaurant logo images and showing the images in the system user interface**. Besides, it is also useful for providing the relevant images when **dealing with the advertisements, deals and special offers**. Moreover, it can also be used to **handle the images uploaded by the customers**, especially when the customers are submitting the reviews with uploaded pictures.

Database Server

Database server is essential for the whole online food ordering system. It is because it **stores sensitive information of all parties related to the system or even restaurants**. For customers and staff, it stores the information about their **personal information such as name, gender, ages, passwords** and so on. It can be used for identifying the login credentials of the users and help the system to know which user is trying to use the system. Besides, the **restaurant data such as names, locations, operating hours and reviews** will also be stored in the database server as well. The data stored can be used as an informative purpose for acknowledging the users about the basic information of the restaurant. Apart from that, when customers and staff want to **review the order history or records** in the system, all the related information can be directly retrieved from the database server without requiring the users to recall back the order details. After the ordering session, the database server will also have to **store the delivery details of the customers' orders**. Thus, the deliverer can view the information and know which foods are prepared for the specific customers and where to deliver the food to.

Front-End Server

In the client-server model, the front-end server is crucial for **providing the user interface and handling the interaction between users and the system**. It **stores all the front-end pages design** components such as font, assets and icons, and codes such as HTML, CSS and JavaScript. This can help for rendering the system web pages or applications. At the same time, it is also able to **handle the basic operation initiated by users**. For example, dynamic form field data validation, page navigation and content displaying. It also **acts as an intermediary to send requests to the back-end server and receive responses from the back-end server**, then

displays it to the user's device screen. Not only that, it takes an important role for **offering a good user experience to users** as it designs all the display pages which are seeable by users. The higher the readability and unambiguity of the system interface, the better the user experience.

Back-End Server

In order to complete all operations initiated by the users and successfully interact with the database server, the back-end server is important for **receiving the input from the front-end server**. Then, it will **process the data when dealing with the database** such as retrieve data from the database, insert, update and delete data in the database. After that, it will **send the response back to the front-end server** for providing feedback to the users. In most cases, the back-end server will deal with the registration and login process to identify the user's credentials. Besides, it is also used for **processing order creation** when customers are submitting the food orders. In the payment session, the back-end server will **process the payment** done by the users by integrating with the payment gateway and record transaction details.

Food Server

Food server is useful for **tracking and managing the orders**, it is especially used by the restaurant staff such as chefs, kitchen staff and waiters. The food server can **real-time update the latest order and show the order on the system interface** used by the chefs. So, the chefs can immediately get notified about the incoming orders and start preparing the dishes. After preparing all the dishes needed from the orders, they can **update the order status via interacting with the food server interface**. So, the waiters can get notified about which table's ordered dishes are ready to be served. Then, they can deliver the dishes to the specified table and mark the order status as completed.

Justification

Client-Server Model is a distributed system model which **separates the workloads and tasks into several parts and each part is handled by stand-alone providers of specific resources** or services, called servers. Meanwhile, there will be various devices which request for the resources or services, called clients. In order to enable clients to engage or connect with the servers, the **network will be used as an intermediary**.

In the online food ordering system, the client-server model would be the suitable model to implement. This is because it has a **straightforward data distribution** since all of the data are stored and managed centrally on the server. When there is any operation required to deal with the data such as update and retrieve data, it could be easy to distribute the data to all clients from a single location. For example, when the staff want to update the catalogue data, the system can access the catalogue server and picture server to retrieve all of the relevant data. So, the staff can view the catalogue with attached pictures and select his or her wanted food item to execute the update function. While there is any update in the database data, the **data can be immediately updated in respective servers to ensure the consistency across the system**. For example, after the staff has submitted the updated details of the food item, the system will access the catalogue server and picture server again to update the catalogue details and food item pictures respectively. It can make sure that all of the information is up-to-date on different servers.

From the perspective of **cost-effectiveness**, the client-server model can achieve this via **effective use of networked systems**. Since this model separates the tasks into different subtasks and utilises different servers to handle different subtasks, the client can **perform heavy computations efficiently** by separating the workloads to various servers. Meanwhile, it only **costs a low cost to buy the less powerful servers** as each task has been broken down into smaller subtasks, and each subtasks with smaller workload can be easily handled by an average performance server. For example, the customer wants to add food items in the system cart, then create the order and make payment for the submitted order. Before breaking these tasks down into several subtasks, this could be difficult and time consuming to be handled by one server. After breaking it down into multiple parts and enabling each server to handle each respective task, the catalogue server and picture server just have to retrieve and show the food item details only. For the database server, it just has to store the user's selected food item only (cart). For the back-end server, it just needs to create the order and process the payment made by the customer only. For the front-end server, it provides the system user interface, request and accept resources or services to and from the back-end server. In overall, **each server will only have low workloads and the task can be completed efficiently** as each of the servers just have to handle 1 to 3 subtasks only. Even though there are heavy transactions from online users of the system

that have to be processed, the system still is able to handle this situation via distributing the proper amount of subtasks to different servers.

Another reason to implement a client-server model is it has **high scalability**. It can **easily adapt to different requirements of users amount** as the data volume and user amount are increasing. Since the system is using stand-alone servers to operate, it can **easily upgrade the servers** via replacing the specific servers independently or upgrade the storage of the specific servers. This approach can effectively prevent clients from not being able to access the system. Meanwhile, the system can also **simply add new servers** to handle more tasks when there are new functionalities or the tasks are getting heavier.

3.2 Test Plan

Test Objective

The test objective is to ensure that the system is able to perform all functionalities without causing any errors. Meanwhile, the test can also help with determining the existing errors in the system. So, the developers will be able to fix those errors and maintain the system reliability and performance.

Testing Process

Unit Testing

Unit testing is one of the important testing stages. It **breaks the whole system functionalities down into various single individual components**. Then, it will **test them one by one**. Before carrying out the unit testing, an appropriate test case with testing condition and input and expected output should be prepared. All the possible conditions, input and output will be needed to be written down in the test case for a comprehensive understanding of component accuracy. Thus, the tester can check whether the single component is running well or doing the expected task based on the expected output written down in the test case during the unit testing. After unit testing, there might be a few unexpected results tested out. Based on the existing issues, the system developers can start locating the root cause of the problem, then start fixing them. Via repetitive unit testing and bug solving, the accuracy, correctness and reliability of every individual component would be enhanced gradually.

For example, in the order module, we will separate it into several single components. One of the components is to add food items to the cart. Before start testing the component, we will create a test case with various possible condition, inputs and expected results. For instance, adding one item, 2 items, 5 items to cart. By comparing the actual results with the expected results, the unlogical or unexpected result can be determined such as incorrect cart item addition, cart item cannot be added when cart is empty or incorrect cart item price calculation.

Via the same way, we will also break the delivery and track order module down into several individual components for carrying out unit testing. For example, we will carry out unit testing for the component “enter delivery address”. We will prepare several test cases such as enter delivery address within the limited length, enter delivery address which exceeds the maximum length, mix the delivery address with special characters and submit an empty delivery address. Via this testing, the tester will be able to know which situation might cause an error and the developers can also fix the relevant problem based on the unexpected result from the test case.

Module Testing

Module testing is mainly used for testing on individual modules or collection of components. Usually, it is done by developers as they familiar with the code and they can easily determine the potential problems (Module Testing, 2022). This can help for promising all the functionalities in the module are working properly and successfully fulfil the requirements. When there are any bugs or errors found when module testing, the developers can immediately locate the bugs and fix the relevant code before it is being used in production.

For example, we will start the module testing once the unit testing is completed. We will integrate the components “manage product information”, “navigate menu of products” and “post reviews and ratings” into one module called product module. Then, we will test all the functionalities in the module but also testing the consistency and propagation of updated data between various functions. If there is any unexpected situation tested out, the developer will determine whether the errors happened before integration or after integration. Then, they will be able to locate the issues and solve them within a short period of time.

On the other hand, we will also integrate the functions of “Authenticate user”, “Process Payment”, “Generate Receipts and Notifications”, “View Transaction history” and “Process Refunds” into one module called “Payment” module. After integration, we will start the module testing by repeating the similar way with the unit testing but testing all the integrated functions in the modules at once. Thus, the tester can find out which functions have caused errors or bugs after integration. Not only that, this testing can also ensure the reliability and completeness of payment functions from one end to another end.

Sub-System Testing

After the successful integration and testing of different units, all of the modules will be integrated into a sub-system. Then, the sub-system testing will be started to prepare. This testing is aiming for testing the collection of modules, which have been integrated into the sub-system. Meanwhile, it will also determine how different modules are interacting with each other in terms of data flow and error handling.

In our system, we will combine all of the modules into 2 subsystems which are customers' sub-system and staff sub-system. The purpose of customers' sub-system is to enable customers to use the online food ordering system to view food menus, make orders, track orders, make payment and other customer related functionalities. For the staff sub-system, it allows staff to manipulate the internal database of the system and obtain the latest information about the business operation. For example, staff will be able to add new food items into the system database, modify existing menus, view and update the ongoing order status and other useful staff related features. Via the result obtained from the sub-system testing, the tester and developers will know how well the system can operate and how reliable the real-time data flows between customer's sub-system and staff sub-system.

System Testing

The system testing is mainly used for checking the compatibility of individual modules. Meanwhile, it also tests the integration of each sub-system in the entire system. During the system testing, there might be some discrepancies between the system and its stated objective, current specifications and systems documentation before. Thus, the system developers will have to execute a series of adjustments to bring the system back to the correct track if the discrepancies between each other are lower than the expectation or ideal level.

In our system, we will conduct the system testing from top to the bottom. It can include many test scenarios as long as it is applicable and approachable in reality. From the perspective of users, the system testing will check all the functionalities from users logging into the system until users tracking the order status. From the perspective of staff, the system testing would involve all of the possible operations done by staff in real life such as modifying food items menu, update

order status and food delivery management. The system testing will even simulate the real life usage environment which closely mimics the production environment. Via this comprehensive testing, the system effectiveness and efficiency will be able to maintain or reach a higher level.

User Acceptance Testing (UAT)

User Acceptance Testing (UAT) is the final stage in the testing process before the system is accepted for operational use. This testing is conducted by the actual end-users or stakeholders to ensure that the system can fulfill their expectations or requirements. In this testing, the system will be used by the real people in the real-world scenario to reveal the actual condition and performance of the system. In this testing, the acceptance criteria has to be clearly defined so that the system will be tested based on those criterias. Once all the criterias can be approached, it indicates that the system can be approved for production use. After that, the users will offer a formal approval or “sign-off” to prove the achievement of user satisfaction from using this system and the agreement of the system to be employed in production.

In our system, we will invite the real staff in restaurants and real customers of the restaurant to experience our developed online food ordering system. They are allowed to do any operation in the system as long as it is logical. For example, the staff will use the food delivery features to test whether the functions can actually work well and properly. Not only for the functionalities, they will also test the performance and user interface of the system to check whether the response times are short enough or the user interface is human-readable and user-friendly.

Tested Items

- Add item to cart
- Schedule delivery time of an order
- Process Payment
- Create new product
- Reserve table in restaurant
- Register as member

- Update membership points balance
- Add new promotion

Testing Techniques

Unit Testing Stage

Technique: Component/ Unit testing (Individual Function/Method in an Object)

Example:

For the “View Points History” function within the “Membership” module, we can test the function to filter points history by date by using various test data, such as different users, different dates and observing if the output is our desired output. This technique is the simplest, and can be used to design subsequent test cases.

Module Testing Stage

Technique: White Box testing

Example:

For the “Membership” module, we can test the integration of functions within this module such as “Update Membership Points Balance”, “View Membership Points Balance”, and “View Points History” by analyzing the code. We can verify that the points are correctly calculated based on purchase values, promotions, and discount vouchers, then correctly updates the user’s points earned history in the database. This technique ensures that the internal code structure is correct before integrating different modules into a sub-system.

Sub-system Testing Stage

Technique: Bottom-Up testing

Example:

For the “Member” sub-system, we begin by testing the lowest-level functions (unit testing), such as the function in “Redeem Vouchers” within “Membership” module that appropriately deducts points from the user’s membership points balance after redeeming a voucher. Once the lowest-level functions are tested and verified, we move upward to test the integration of these low-level functions into a module. After that, we test the

integration of different modules into the “Member” sub-system. For example, we verify whether the redeemed vouchers from “Membership” module are still available when placing an order in the “Order” module, ensuring data consistency across different sub-systems.

System Testing stage

Technique: Regression testing (For incremental model)

Example:

For the entire sushi online ordering system, after integrating a new increment with new functions into the system, we rerun the tests for previous increments to ensure that functions in previous increments are still working as expected. If the previous increment’s functions are having problems after the new increment, we should check whether the current new increment exposed the problems, or the current new increment created the problems.

User Acceptance Testing Stage

Technique: Black-Box Testing (Release testing)

Example:

For the “Membership” module, real users will test the system by interacting with key features like “View Membership Points Balance” and “Redeem Vouchers” without knowing the internal code structure. Users will log in to the system and perform various tasks such as checking their current membership points balance after earning/ redeemed points through a recent purchase, or redeeming a voucher through the voucher catalog. They will also attempt to redeem a voucher but does not have sufficient points, and the system is expected to show an appropriate error message indicating that the user has insufficient points for redeeming this voucher. This technique is used pre-release and post-release, to ensure that by providing different inputs in different environments, based on the system requirement specification, the related outputs should behave correctly and according to the user’s expectations, without seeing the internal structure of the system.

Testing Environment

Operating System

- Window
- Linux
- Mac
- Android
- IOS

Hardware

- Desktop
- Laptop
- Smart Phone
- Tablet
- POS machine

Browser

- Google Chrome
- Microsoft Edge
- FireFox

Testing Constraints

1. Testing during off-peak hours:

To minimize the risk of disruptions to real customers and the daily operations of the system, all system testing activities should be scheduled between **10:00 PM and 6:00 AM**. These off-peak hours are chosen based on usage patterns, ensuring that system performance tests, feature tests, or bug fixes do not interfere with peak usage times when the system is heavily relied upon by customers and staff. By conducting tests during these less active hours, testers can assess system functionality and performance without impacting normal business activities.

2. Time zone considerations:

Given the nature of the system serving many Sushi Mentai restaurants in multiple geographical locations, it is important to factor in time zone differences during testing. For example, a test scheduled for a restaurant operating in **New York (EST)** must be conducted according to **Eastern Standard Time**, while a restaurant in **Kuala Lumpur, Malaysia (MYT)** must follow **Malaysia Time (GMT+8)**. Testing teams must account for the local operating hours and time zones of each restaurant to ensure that tests reflect the actual working environment of the restaurant, providing more accurate and relevant test results.

3. Database maintenance window:

For tests involving critical updates, such as modifying membership points balances or managing table reservations, testing should be performed during a predefined database maintenance window that occurs **outside of the restaurant's operating hours**. This prevents conflicts with live transactions and data processing, ensuring the integrity of customer data and smooth operation of the restaurant management system. Careful coordination is required between database administrators and testers to ensure that the database is available for updates while avoiding any downtime or disruption during business hours.

4. Limited network bandwidth consideration:

In order to simulate real-world conditions where network bandwidth may be limited, it is essential to conduct tests during scheduled periods when bandwidth-limiting tools are accessible. These tests help assess how the system performs under constrained network conditions, which may occur in locations with poor internet connectivity or during high-traffic times. By scheduling these tests, testers can better understand how the system behaves when network resources are limited, ensuring that the system is resilient and responsive even in less-than-ideal circumstances.

3.3 Test Case

Add to Cart (Ong Yi Xin)

Positive Test Case

Tester's Name		Date Tested	Test Case (Pass/Fail/Not Executed)	
S #	Prerequisites:	S #	Test Data	
1	Login to the system as client role	1	product_id = P0001	
2		2	quantity = 2	
3		3		
4		4		
Test Scenario	Add the product which is exist and the quantity is available			
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Click on the "Add cart" button of the product	Pop up a window to let user select quantity		
2	Click one times on the plus(+) button select quantity	The quantity showing updated		
3	Click on "Confirm" button	Product and quantity selected success added to shopping cart		
4				

Negative Test Case

Tester's Name		Date Tested	Test Case (Pass/Fail/Not Executed)	
S #	Prerequisites:	S #	Test Data	
1	Login to the system as client role	1	product_id = P0001	
2		2	quantity = -2	
3		3		
4		4		
Test Scenario	Add the product which is exist and the quantity is invalid			

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Click on the "Add cart" button of the product	Pop up a window to let user select quantity		
2	Fill in the quantity field.	The quantity showing change to -2		
3	Click on "Confirm" button	Display an error message, "Quantity invalid" beside the input field and the product did not add in to cart.		
4				

Schedule Delivery Time (Ong Yi Xin)

Positive Test Case

Tester's Name	Date Tested	Test Case (Pass/Fail/Not Executed)		
S #	Prerequisites:	S #	Test Data	
1	Login to the system with admin role	1	orderId = O0001	
2	An order is in process status that waiting to schedule delivery	2	deliveryTime = 2024/09/04 14:30:00	
3		3	riderId = R0002	
4		4	confirmOption = Yes	
<u>Test Scenario</u>	Schedule delivery time for order which status is at processing to 2024/09/04 14:30:00:00 that is within the restaurant operation hour and the delivery time selected not early than current time and available rider.			
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Select order.	Pop up window to select option for futher action to take of the order		
2	Select "Schedule Delivery" option	Redirect user to schedule delivery page		
3	Select time at the	Show list of available		

	delivery time fields	rider for the time to delivery		
4	Select rider.	Pop up window for confirmation to schedule the delivery		
5	Select "Yes"	Delivery schedule successful and pop up a success message		

Negative Test Case

Tester's Name		Date Tested	Test Case (Pass/Fail/Not Executed)	
S #	Prerequisites:		S #	Test Data
1	Login to the system with admin role		1	orderId = O0001
2	An order is in process status that waiting to schedule delivery		2	deliveryTime = 2024/09/04 14:45:00
3	Current time is later than delivery time want to select		3	
4			4	
<u>Test Scenario</u>	Schedule delivery time for order O0001 which status is at processing to 2024/09/04 14:45:00:00 that is early than current time.			
Step #	Step Details	Expected Results	Actual Results	
Step #	Step Details	Expected Results	Actual Results	
1	Select order.	Pop up window to select option for futher action to take of the order		
2	Select "Schedule Delivery" option	Redirect user to schedule delivery page		
3	Select time at the delivery time fields	Display error message with "Delivery time cannot early than current time" and ask user re-enter again the delivery time.		
4				
5				

Tester's Name		Date Tested	Test Case (Pass/Fail/Not Executed)	
S #	Prerequisites:		S #	Test Data
1	Login to the system with admin		1	orderId = O0001

	role								
2	An order is in process status that waiting to schedule delivery		2	deliveryTime = 2024/09/05 07:00:00					
3			3						
4			4						
<hr/>									
Test Scenario	Schedule delivery time for order O0001 which status is at processing to 2024/09/04 14:45:00:00 that is early than restaurant operation time.								
Step #	Step Details	Expected Results	Actual Results		Pass / Fail / Not executed / Suspended				
1	Select order.	Pop up window to select option for futher action to take of the order							
2	Select "Schedule Delivery" option	Redirect user to schedule delivery page							
3	Select time at the delivery time fields	Display error message with "Delivery time are not in restaurant operation time. Try to schedule another time." and ask user re-enter again the delivery time.							
4									
5									

Process Payment (Chia Ming Yi)

Positive Test Case

Tester's Name		Date Tested	Test Case (Pass/Fail/Not Executed)	
S #	Prerequisites:	S #	Test Data	
1	Access to Chrome Browser	1	Preferred Currency = MYR	
2	User have login and the role of account is member	2	payment_method = online_banking	
3	Order that want to do payment must be exists	3	bankAccountId = 1234567890	
4		4	bankPassword = 123456@#%	
5		5		
6		6		
Test Scenario		Process payment with valid currency bank account and password		
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Click "Process to Payment" on the check out page	redirect to the payment process page		
2	Select preferred currency from the currency selection list.	MYR is selected and displayed as the preferred currency.		
3	Select payment method.	Display the bank account and password input field.		
4	Enter the bank account ID and password in the designated fields.	Bank account ID and password are accepted and no error message is displayed.		
5	Select "Yes" to confirm the payment.	The selection of "Yes" is recorded, and the confirmation process proceeds.		
6	Confirm the payment.	Payment is processed successfully, and a confirmation message is displayed indicating successful payment.		

Negative Test Case

Tester's Name		Date Tested		Test Case (Pass/Fail/Not Executed)	
1	Access to Chrome Browser		1	Preferred Currency = XYZ	
2	User have login and the role of account is member		2	payment_method = online_banking	
3	Order that want to do payment must be exists		3	bankAccountID = 123	
4			4	bankPassword = abc	
5			5		
6			6		
Test Scenario		Process payment with invalid currency bank account and password			
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended	
1	Click "Process to Payment" on the check out page	redirect to the payment process page			
2	Enter currency to the select currency field.	Error message is displayed indicating that the selected currency is invalid.			
3	Select payment method.	Display the bank account and password input field.			
4	Enter the bank account ID and password in the designated fields.	Payment fails and an error message is displayed indicating missing or incorrect information.			
5	Select "Pay" to confirm the payment.	Payment fails, and an appropriate error message is displayed.			
6	Confirm the payment with invalid data and verify that no success message is displayed.	Payment does not go through, and no success message is shown.			
6					

Create New Product (Chia Ming Yi)

Positive Test Case

Tester's Name		Date Tested		Test Case (Pass/Fail/Not Executed)	
1	Login the system with the account that		1	productName = Tamago Sushi	

	role is staff							
2				2	cost = 0.80			
3				3	price = 2.20			
4				4	photo = tamago_sushi.jpg			
5				5	description = Sweet omelet atop vinegared rice.			
Test Scenario	Create a new product which name as tamago sushi with valid cost and price.							
Step #	Step Details	Expected Results	Actual Results			Pass / Fail / Not executed / Suspended		
1	Click on "Create Product" button	Redirect to the create new product form page.						
2	Input product name text field	Display green thick icon beside the input field.						
3	Input product cost field	Display green thick icon beside the input field.						
4	Input product price field	Display green thick icon beside the input field.						
4	Select photo and upload	Display green thick icon beside the input field.						
5	Input description to the description field.	Display green thick icon beside the input field and able the create button to click.						
6	Click on "Create" button	Pop up the success add new product message.						

Negative Test Case

Tester's Name		Date Tested	Test Case (Pass/Fail/Not Executed)					
S #	Prerequisites:		S #	Test Data				
1	Login the system with the account that role is staff		1	productName = ""				
2			2	cost = -0.80				
3			3	price = -2.20				
4			4	photo = ""				
5			5	description = ""				
Test Scenario	Enter invalid information when filling the create product form							
Step #	Step Details	Expected Results	Actual Results			Pass / Fail / Not executed / Suspended		

1	Click on "Create Product" button	Redirect to the create new product form page.		
2	Click on the product name input field and does not enter anything	Display a red cross and error message as "Product name cannot be empty" beside the input name field.		
3	Input product cost field	Display a red cross and error message as "Cost must be greater than 0" beside the input cost field and ask user to re-enter the cost.		
4	Input product price field	Display a red cross and error message as "Price must be greater than 0" beside the input price field and ask user to re-enter the price.		
4	Click on the input file field but not select image	Display a red cross and error message as "Product image cannot be empty" beside the input file field and ask user to upload a product image.		
5	Click on the description input field and does not enter anything	Display a red cross and error message as "Product description cannot be empty" beside the input description field.		
6				

Reserve Table (Lim Jun Wei)

Positive Test Case

Tester's Name		Date Tested	Test Case (Pass/Fail/Not Executed)	
S #	Prerequisites:	S #	Test Data	
1	Login the system as member	1	Date = 6/9/2024	
2		2	Time slot = 3.00 p.m.	
3		3	Number of customers = 4	
4		4	Table number = 1	
		5	Mobile number = 012-5236854	
Test Scenario		Reserve the restaurant tables with reservation date of 6/9/2024 which is restaurant operation date and the time slot selected is at 3.00 p.m. which is within operation hour. The number of customers chosen is 4 people which is within the acceptable range of person number per table. The restaurant table number selected is 1 which is within the available table range. The mobile number entered is 012-5236854 which is a valid mobile number.		
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Select the table reservation date in the table reservation form field.	Display a range of available time slot drop down list based on the selected date.		
2	Select the time slot as table reservation time.	Display the available number of person per table in select list		
3	Enter the number of person per table	Display the available table number based on the selected date, time slot and pax per table		
4	Select the table number.	Display the form field for entering mobile number.		
5	Enter mobile number and click on submit button.	Display successful table reservation message and navigate to the table reservation history page.		

Negative Test Case

Tester's Name		Date Tested	Test Case (Pass/Fail/Not Executed)	
S #	Prerequisites:	S #	Test Data	
1	Login the system as member	1	Date = 15/9/2024	
2		2	Time slot = 5.00 p.m.	

3			3	Number of customers = 30
4			4	Table number = 10
			5	Mobile number = 017-3568214

Test Scenario	Reserve the restaurant tables with reservation date of 15/9/2024 which is restaurant operation date and the time slot selected is at 5.00 p.m. which is within operation hour. The number of customers chosen is 30 people which is out of the acceptable range of person number per table.
----------------------	---

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Select the table reservation date in the table reservation form field.	Display a range of available time slot drop down list based on the selected date.		
2	Select the time slot as table reservation time.	Display the available number of person per table in select list		
3	Enter the number of person per table	Display error message "Number of person per table cannot exceed 10 person", disable table number field and submit button, and prompt user to re-enter the valid number of person per table.		
4	Click on disabled table number select field.	Display error message "Please re-enter a correct number of person per table within range of 1 - 10." and prompt user to re-enter the valid number of person per table.		
5	Enter the mobile number and click on disabled submit button.	Display error message "Please complete the form with valid data before submission".		

Register as Member (Lim Jun Wei)

Positive Test Case

Tester's Name		Date Tested	9/5/2024	Test Case (Pass/Fail/Not Executed)
S #	Prerequisites:		S #	Test Data
1	Access Chrome or Microsoft Edge browser		1	Username = Aaron Leong
2			2	Mobile number = 011-5632452

3			3	Email address = aaron123@gmail.com
4			4	Password = Aaron2543!\$
			5	One-Time-Password = 532971

Test Scenario	Register as member with the name "Aaron Leong" which is a valid name and mobile number 011-5632452 which is an valid format of mobile number. The email address entered is aaron123@gmail.com which is also a valid email address. The password used is "Aaron2543!\$" which is an valid and strong password format. After the verification email is sent to the email mailbox, the One-Time-Password is 532971. Then, enter the same One-Time-Password into the registration form field.
----------------------	---

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Enter user name in the username data field.	Display a green tick icon beside the username data field.		
2	Enter mobile number in the mobile number data field.	Display a green tick icon beside the mobile number data field.		
3	Enter email address in email address data field.	Display a green tick icon beside the email address data field.		
4	Click on the "Verify Email" button.	Display a successful message "The verification email has been sent".		
5	Check the verification email sent by the system in the mailbox of the email address entered and enter the One-Time-Password in the One-Time-Password data field in the registration form.	Display a green tick icon beside the One-Time-Password data field.		
6	Enter the password	Display a green tick icon beside the password data field.		
7	Click on the "Submit" button.	Display a successful message "The registration is successful.", then navigate to the member dashboard page.		

Negative Test Case

Tester's Name		Mark	Date Tested		9/5/2024	Test Case (Pass/Fail/Not Executed)	Fail
S #	Prerequisites:		S #	Test Data			
1	Access Chrome or Microsoft Edge browser		1	Username = 123			

2			2	Mobile number = 017-2363215
3			3	Email address = liting23@gmail.com
4			4	Password = liting
<hr/>				
<u>Test Scenario</u>	Register as member with the name "123" which is an invalid name and mobile number 017-2363215 which is a valid format of mobile number. The email address entered is liting23@gmail.com which is an invalid email address. The password used is "liting" which is an invalid and weak password format.			
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Enter user name in the username data field.	Display a red cross icon beside the username data field and prompt user to re-enter the username.		
2	Enter mobile number in the mobile number data field.	Display a green tick icon beside the mobile number data field.		
3	Enter email address in email address data field.	Display a red cross icon beside the email address data field and disable the "Verify Email" and form's "Submit" button.		
4	Enter password in password data field.	Display a red cross icon beside the password data field.		
5	Click on disabled "Submit" button.	Display the error message "Please complete the form with correct data before submission" and prompt user to re-enter the valid email address.		

Update Membership Points Balance (Ng Ian Kai)

Positive Test Case #1

Tester's Name		Date Tested	Test Case (Pass/Fail/Not Executed)	
S #	Prerequisites:		S #	Test Data
1	Logged in to the system as customer		1	user_id = 5
2			2	points_balance = 0
3			3	order_amount = 50
4			4	
Test Scenario	Check if the user's membership points balance updates accordingly, based on points earned through orders			
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Place a new order with a purchase amount of RM50	The system should calculate and credit the appropriate amount of membership points, in this case, 50 points to the user's points balance.		
2	Navigate to the "View Membership Points Balance" page	The user's current membership points balance should reflect the additional points earned from the RM50 order, which is now 50.		

Positive Test Case #2

Tester's Name		Date Tested	Test Case (Pass/Fail/Not Executed)	
S #	Prerequisites:		S #	Test Data
1	Logged in to the system as customer		1	user_id = 5
2			2	points_balance = 50
3			3	order_amount = 100
4			4	
Test Scenario	Check if the user's membership points balance updates accordingly, based on points earned through orders			
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Place a new order with a purchase amount of RM100	The system should calculate and credit the appropriate amount of membership points, in this case 100 points, to the user's points balance.		

2	Navigate to the "View Membership Points Balance" page	The user's current membership points balance should reflect the additional points earned from the order with a purchase amount of RM100, which is now 150.		
---	---	--	--	--

Add New Promotion (Ng Ian Kai)

Positive Test Case

Tester's Name		Date Tested	Test Case (Pass/Fail/Not Executed)	
S #	Prerequisites:	S # Test Data		
1	Logged in to the system as staff	1	promotion_name = "Christmas Sale"	
2		2	start_date = 20-12-2024	
3		3	end_date = 27-12-2024	
4		4	discount_value = 25%	
		5	associated_products = [product_id = 1, 2, 3, 4]	
		6	redemption_limits = 100	
<u>Test Scenario</u>		Adds a new promotion to the system when all promotion details are correctly entered.		
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Click on "Add new promotion" button.	The system displays a form to enter new promotion details, where the eligibility criteria field is defaulted to none and redemption limits field is defaulted to 99999.		
2	Fill in the form's fields with the provided test data (start date, end date, discount value, associated products, redemption limits).	The system should accept the input values without any error.		
3	Submits the form by clicking the "Save" button.	A confirmation message is displayed, indicating that the Christmas Sale promotion was successfully added.		

Negative Test Case

Tester's Name		Date Tested	Test Case (Pass/Fail/Not Executed)	
S #	Prerequisites:	S # Test Data		
1	Logged in to the system as staff	1	promotion_name = "New Year Sale"	
		2	discount_value = abc	
<u>Test Scenario</u>		Tryin to add a new promotion to the system without specifying the details.		

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Click on "Add new promotion" button.	The system displays a form to enter new promotion details, where the eligibility criteria field is defaulted to none and redemption limits field is defaulted to 99999.		
2	Fill in the form's promotion name and discount value fields with the provided test data, without filling other fields.	The system should display an error message below the discount value field, indicating that the input value is in an incorrect format.		
3	Correct the discount value input field to 25%.	The system should accept the input values without any error.		
4	Submits the form by clicking the "Save" button.	The system should not allow the promotion to be added and should display an error message indicating the following mandatory fields are missing: Start Date, End Date, Associated Products.		

3.4 UI Design Principles

Ong Yi Xin (Modify Shopping Cart)

The screenshot shows a shopping cart interface for "Sushi Mentai". On the left, there are three product cards: "Unagi Nigiri Sushi - 2pcs" (RM 3.00), "Sushi Collection #03" (RM 18.00), and "Salmon Sushi #07 - 3pcs" (RM 6.00). Each card includes a small image, the product name, a brief description, and quantity selection buttons (-, +). On the right, the "Order Summary" section displays the subtotal (RM 39.00), a discount of -10% (-3.90), a delivery fee of +6.00, and a total of RM 41.10. It also features a "Add promo code" input field and an "Apply" button, followed by a large "Go to Checkout" button.

Order Summary	
Subtotal	RM 39.00
Discount (-10%)	-3.90
Delivery Fee	+6.00
Total	RM 41.10

Shopping cart interface applied 4 UI design principle

User Familiarity

User familiarity refers to designing interfaces that utilise common design patterns and interactions that users are already familiar with. In the shopping cart interface, user familiarity is applied in a way that utilises a layout and structure similar to other popular e-commerce platforms such as Shopee, Lazada, or Taobao. For example, the product listings on the left side of the screen, including product images, names, prices, and quantity buttons, are design elements that many users will recognize from previous online shopping experiences. This familiarity is important because it reduces the cognitive load on users, allowing them to navigate and interact with the shopping cart more efficiently. Since users don't have to learn a new system from scratch, they are more likely to actively engage with the interface.

Consistency

Consistency in user interface design refers to ensuring that similar operations or elements are presented in a uniform manner throughout the interface. In the shopping cart interface of the online food ordering system we designed, consistency is achieved across all products by aligning the buttons used to modify shopping cart items (e.g., adding or removing items or deleting products) in a uniform manner.

For example, each product in the shopping cart is displayed in a frame that includes the product image, name, and price, as well as buttons to modify the quantity or remove the product. The “Remove” button is always located on the right side of each product and is represented by a red trash can icon, allowing the user to see at a glance what the button does. The red color is usually associated with actions such as delete or remove, which deepens the user's understanding without the need for text. Similarly, the buttons for increasing or decreasing the number of items use the “+” and “-” symbols, and all items are arranged horizontally. This uniform arrangement allows users to anticipate how to interact with any item in the cart, which improves usability and reduces the likelihood of error.

Consistency also extends to visual elements such as fonts, colours, and spacing, all of which are applied consistently throughout the interface, giving it an organised, professional look. This consistency is critical because it reduces confusion and ensures that users can easily find and use the features they need no matter where they are in the cart.

Recoverability

Recoverability is the ability of a user to undo or reverse their actions without causing irreversible harm. In this shopping cart interface, recoverability is achieved by allowing the user to cancel the checkout process if they realise they made a mistake or forgot to add an item to their order. If the user clicks “Go to Checkout” but then decides to return to the cart, the items in the cart will remain unchanged and nothing will be lost. This principle is critical to creating a sense of control and security for the user. Knowing that their actions can be undone reduces anxiety and

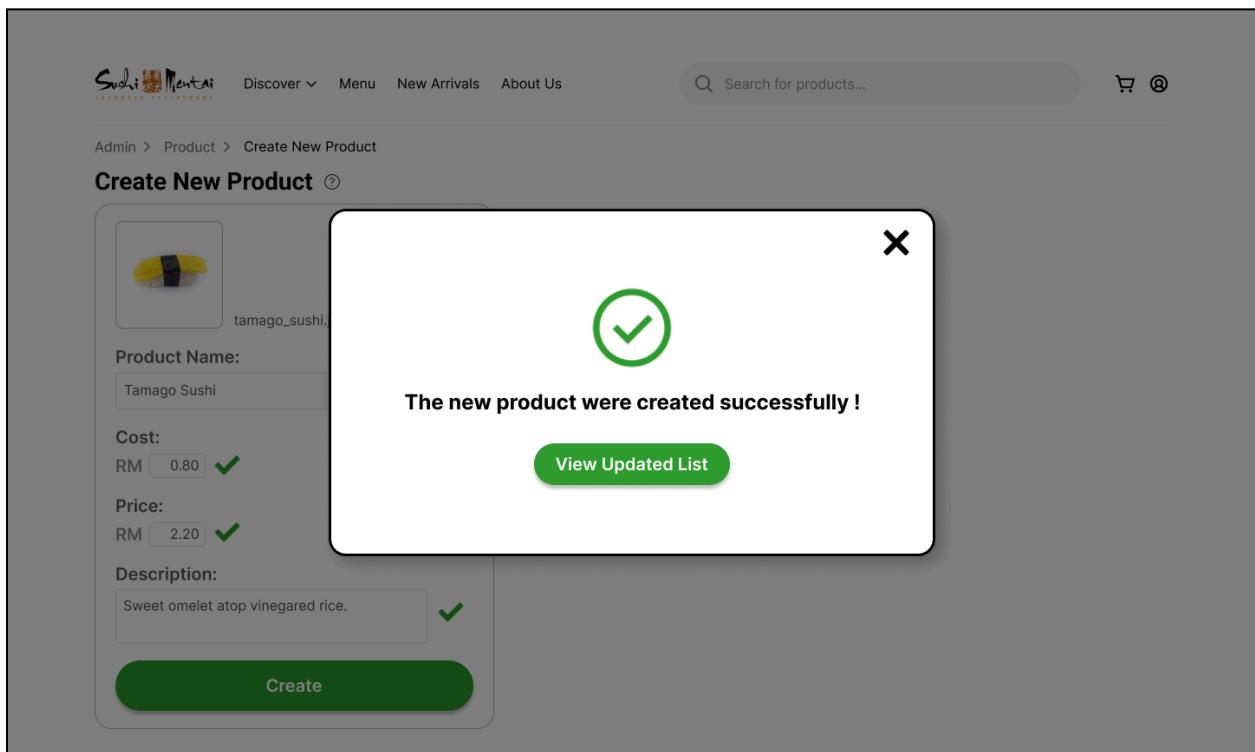
encourages users to explore and use the interface without worrying that making a mistake will negatively impact the shopping experience.

User Guidance

The principle of user guidance is to provide users with clear instructions or help to help them navigate the interface. In this shopping cart interface, user guidance is provided through a help button located next to the “Your Cart” heading. When the user clicks on it, the button provides the user with information about actions that can be taken in the shopping cart, such as modifying the number of items or deleting items, as well as selecting items for checkout. This kind of guidance is critical to ensure that users quickly understand how to interact with the system without feeling lost or confused. In-place guidance improves usability and makes it easier for users to complete tasks efficiently.

Chia Ming Yi (Create New Product)

The screenshot shows the 'Create New Product' page on the Sushi Tentai website. At the top, there is a navigation bar with links for Discover, Menu, New Arrivals, and About Us. A search bar and a user icon are also present. Below the navigation, the breadcrumb path Admin > Product > Create New Product is shown. The main form has a title 'Create New Product'. It includes a placeholder image of a tamago sushi roll, a file input field containing 'tamago_sushi.jpg' with a green checkmark, and a 'Product Name' field containing 'Tamago Sushi' with a green checkmark. There are also fields for 'Cost' (RM 0.80) and 'Price' (RM 2.20), both with green checkmarks. A 'Description' field contains the text 'Sweet omelet atop vinegared rice.' with a green checkmark. A large green 'Create' button is at the bottom of the form.



User familiarity

This principle ensures that users recognize and understand terms and design patterns based on their prior experiences. Familiar terms like "Product Name," "Cost," and "Price" label are used, along with intuitive icons like the green checkmark for successful validation. It minimizes the learning curve and makes the interface more intuitive, helping users navigate easily without confusion.

Consistency

Consistency means performing similar actions in similar ways across the interface. The form fields are uniformly styled, with consistent use of green checkmarks and the "Create" button, maintaining a uniform design language. It enhances user experience by making interactions predictable, reducing the need for users to relearn how different components work.

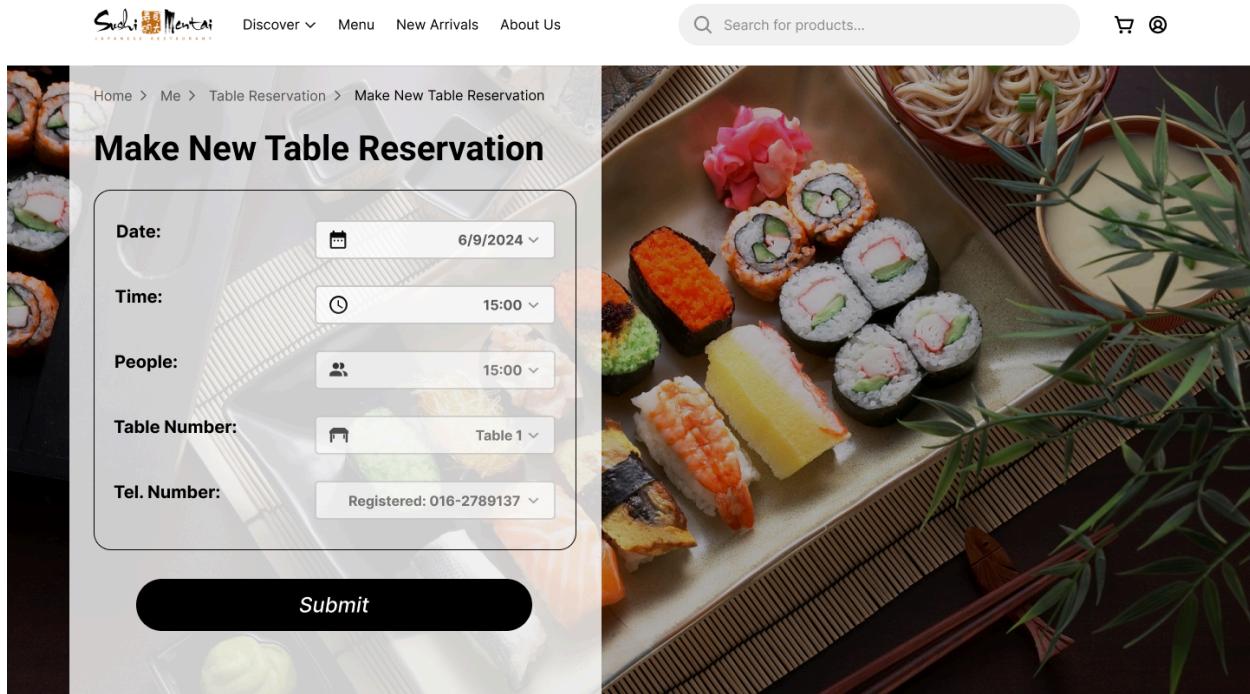
Minimal Surprises

This principle ensures that the system behaves as users expect. Actions like filling out forms and clicking "Create" result in predictable outcomes, such as the success dialog that indicates the new product created successfully, which aligns with typical user expectations. It builds user trust and confidence by ensuring the system is reliable and predictable, leading to a smoother user experience.

User Guidance

User guidance involves providing feedback and help throughout the user's interaction. The UI uses green checkmarks for validation feedback and a success dialog with a next step button, "View Updated List." It helps users complete tasks efficiently and accurately, reducing errors and ensuring users know what to do at every step, enhancing overall satisfaction.

Lim Jun Wei (Table Reservation)



User Guidance

User guidance is one of the crucial UI design principles which enables the system interface to show meaningful messages to users. Not only that, the system interface will also display the output messages or feedback as response for users' action. Thus, it can help users to understand the current state of the system and realise whether the incorrect steps are taken.

In table reservation pages, the user guidance principle is applied into the design of the reservation form. The system will show the clear and large title at the top of the table reservation form which helps users to know what the current page shows. The system will also display a clear and short label on the left of each form field with understandable placeholders as references. So, the users will be able to know the information they should fill into the form fields. Meanwhile, the limited list options will be applied when selecting the range of available time slot and table number. This can effectively guide users to input the correct information via selecting the right option. The submit button will be placed at the bottom of the form, it can help users to conceive information that it is the last interaction within the table reservation form.

Consistency

When designing a good user interface within a system, the designer must obey the principle of consistency to build a clear and unambiguous content for experts and novice users. This principle means that all of the similar operations should be activated in the same way. Thus, the interface design should also be similar with other system interfaces when they have the similar actions to be taken.

In the table reservation page, the system will show a similar form design for the label and form field. This is because the operation and format of all the form fields are almost the same. For example, both the number of person fields and mobile number fields are providing a space for users to input the corresponding data. The position of labels and data field for different sections will be arranged in such a way that the label is located at the left side of the respective data field.



Home > Me > Table Reservation > Reservation Details

Table Reservation Details



Date:

Time:

People:

Table Number:

Tel. Number:



Home > Me > Table Reservation > Reservation Details

Edit Table Reservation Details

Date:

Time:

People:

Table Number:

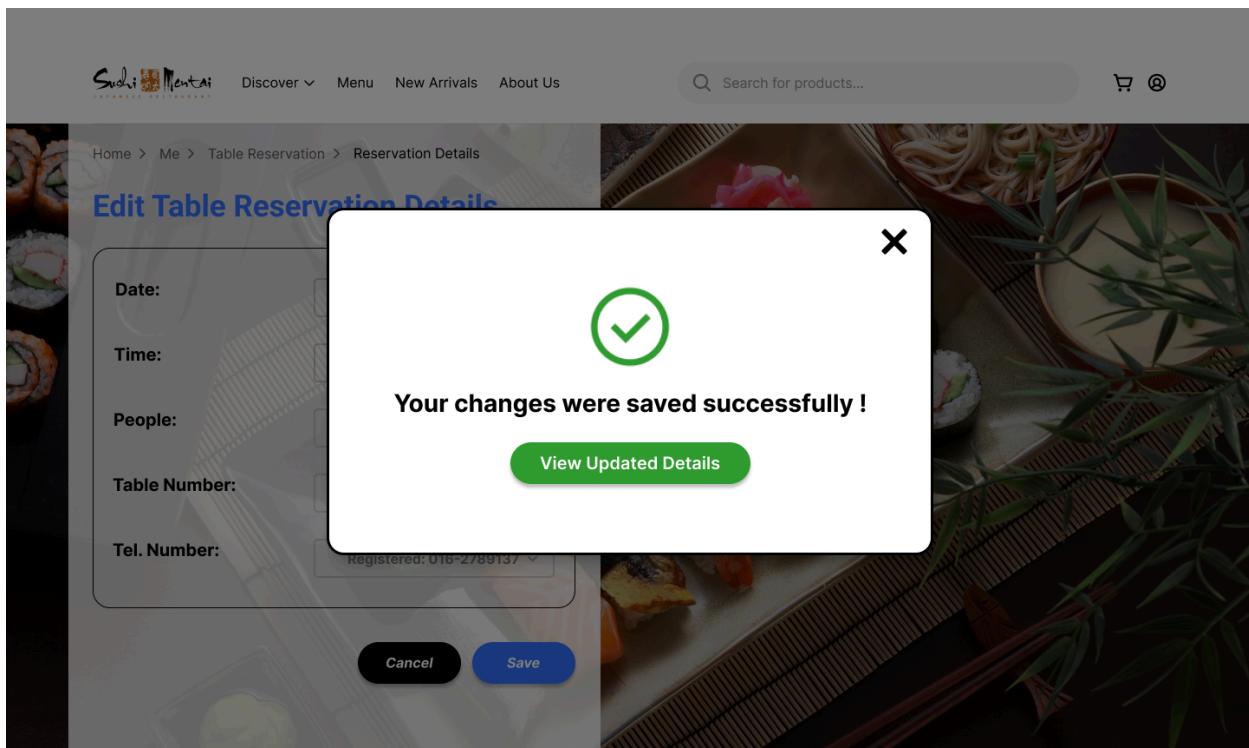
Tel. Number:



Recoverability

Sometimes customers might have accidentally made a mistake when dealing with the system. Thus, the recoverability should be applied into the system for providing a high fault-tolerance rate user interface design via allowing customers to undo their operations. So, the user experience can be improved effectively. The customers also would not worry about human-errors when executing some actions.

In the table reservation module, the system has provided a mechanism for updating the table reservation details for customers. After customers have reserved a table, they are allowed to correct the reservation details immediately within 3 hours once they find out that they have filled in the form with incorrect data. They can decide to reselect the table reservation date and time and modify the number of people, table number and mobile number. They can even cancel the table reservation. This mechanism has maximised the fault-tolerance as customers still have chances to change the current information even though they have created the reservation.

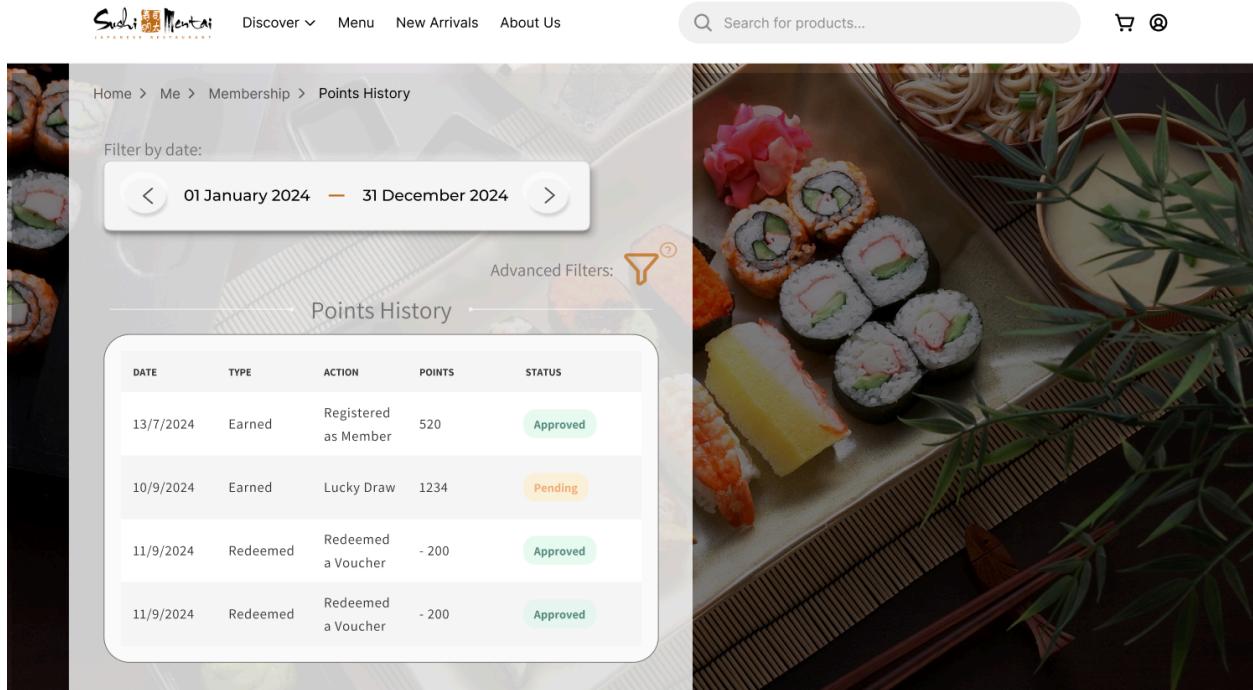


Minimal Surprise

In order to develop a system with a smooth flow of operation, the principle of minimal surprise should be implemented into the system. It is aiming for preventing customers from being surprised by the behaviour of the system. It indicates that the system should provide sufficient and useful messages or content after each of the system and user operation most of the time. Thus, the customers can keep updated when being acknowledged about the current stages of the system. The confusing and anxiety feelings of customers can also be minimised.

When applying for a table reservation in an online food ordering system, the minimal surprise will be implemented. After the customers have clicked on the submit button, the system will show a successful creating reservation dialog to the customers if all the customers' input are correct. Then, the system will just navigate customers to the table reservation history page. So, the customers will be able to know that the table reservation is completed successfully. A sense of relief can also be achieved for the customers. If there is no message dialog after successful reservation, the customers will not be able to ensure whether the reservation is completed. They might think that the process is not completed or there are some input errors inside the table reservation form.

Ng Ian Kai (View Points History)



(Figure: prototype of ‘View Points History’ page)

User Familiarity

This principle refers to designing interfaces in ways that users find familiar, leveraging their prior knowledge to help them navigate the system more easily. In this UI, **familiar elements** such as the **filter and help icons, color codes, and breadcrumbs** are key examples. Icons like the filter and help icons are widely recognized across platforms, ensuring users intuitively understand their function and purpose without needing extensive guidance. The use of familiar color codes such as green for “Approved” and yellow for “Pending” enables users to instantly recognize statuses at a glance, reducing the need to read and interpret text. Additionally, breadcrumbs provide a clear and familiar navigation path, helping users easily understand their location within the application and navigate back when needed. This boosts user confidence, reduces frustration, and minimizes errors.

User Diversity

This principle emphasizes designing for a wide range of users, taking into account different experience levels and preferences. In this case, the UI offers both **a simple date filter and more advanced filters**, catering to the varying needs of different users. Novice users, who may prefer quick and straightforward filtering, can benefit from the simple date filter, while advanced users, seeking more control, can utilize the more granular advanced filters to fine-tune their results. This flexibility ensures that the system meets the needs of both novice and advanced users, thus adhering to the principle of "User Diversity."

User Guidance

This principle involves providing users with tools or features that help guide them through complex tasks or interactions. In the UI, the **help icon associated with the advanced filter** serves as a proactive guide for users who might be unfamiliar with advanced filtering options. This minimizes the risk of confusion and ensures that all users can efficiently navigate the more complex aspects of filtering without excessive trial and error. By offering such guidance, the design improves overall user satisfaction and efficiency.

Consistency

Consistency is a key design principle focused on maintaining uniformity across different parts of the UI to reduce cognitive load. In this interface, **consistent layouts and styles** such as those seen in the **breadcrumbs, navigation bar, and tables** are applied across different pages. This consistency ensures that users do not have to relearn the interface as they move from one section to another. Consistent structures and patterns help make the experience more intuitive, allowing users to focus more on their tasks rather than on how to use the interface itself.

References

1. Adib. (2023, July 28). Jom kenali asal usul Sushi Mentai. Purple Durian. <https://purpledurian.my/sejarah/jom-kenali-asal-usul-sushi-mentai/>
2. Mentai, S. (n.d.). Sushi Mentai | 寿司明太. Sushi-Mentai.com. © Copyright 2024. Sushi Mentai. <https://sushi-mentai.com/EN/our-restaurants>
3. Suruhanjaya Syarikat Malaysia. (2024). Company profile. In SSM Search (pp. 1–6). https://storage.googleapis.com/prod-ssm-search-bucket/pdf/SUSHI_MENTAI_ENTERPRISE_SDN_BHD_FKR12B_EN_Company_Profile_17206727177402WINKX.pdf
4. David, E. (2020). 12 advantages & disadvantages of questionnaires. ProProfs Survey Blog. <https://www.proprofssurvey.com/blog/advantages-disadvantages-of-questionnaires/>
5. Bhasin, H. (2020, September 21). Questionnaire: Definition, characteristics & advantages. Marketing91. <https://www.marketing91.com/questionnaire/>
6. Indeed Editorial Team. (2023, March 11). 12 techniques for requirement gathering. Indeed Career Guide. <https://www.indeed.com/career-advice/career-development/requirement-gathering-techniques>
7. Jama Software. (2023). Requirements gathering techniques for agile product teams. Jama Software. <https://www.jamasoftware.com/requirements-management-guide/requirements-gathering-and-management-processes/11-requirements-gathering-techniques-for-agile-product-teams>
8. Unicommerce. (n.d.). Top challenges of order management system in e-commerce. <https://unicommerce.com/blog/top-challenges-of-order-management-system-in-e-commerce/>
9. Storfox. (n.d.). 16 online shopping problems and how to fix. <https://storfox.com/blog/16-online-shopping-problems-and-how-to-fix/>

10. Stratii. (n.d.). IT issues with solutions.
<https://www.stratiis.com/blog/it-issues-with-solutions/>
11. TechTarget. (n.d.). Replacing vs maintaining legacy systems.
<https://www.techtarget.com/searchcio/feature/Replacing-vs-maintaining-legacy-systems>
12. Tutorialspoint. (n.d.). System analysis and design: Planning.
https://www.tutorialspoint.com/system_analysis_and_design/system_analysis_and_design_n_planning.htm
13. ResearchGate. (n.d.). Chapter three: Analysis of existing systems.
https://www.researchgate.net/publication/332180629_CHAPTER THREE 31 Analysis of existing systems
14. Module Testing. (2022, November 24). GeeksforGeeks.
<https://www.geeksforgeeks.org/module-testing/>