Tapau at Inti

Name: Lim Chen Teik

Student ID: P21013267

Supervisor Name: Ms. Vaithegy Doraisamy

Bachelor of Science (Hons) in Computing, 3+0 in Collaboration with Coventry University

School of Engineering and Technology, INTI International College Penang

10th January 2021

Acknowledgment

Firstly, I would like to express my gratitude to my lecturer, Ms. Vaithegy Doraisamy, who has encouraged me with her patience and knowledge in this semester. This work can only be done dure to her continuous guidance, support, and encouragement. Under her guidance, I successfully solve many errors and learned a lot. I have been very glad to have her as my lecturer

Besides, I would like to thank Miss Siti Syakirah Sazali for teaching me the Python language. She provides me with patient guidance when teaching me Python programming language. She will clear my doubts when I have questions about programming in class. The Python programming language is very useful for me when doing this project. Without her guidance, I would have this Python programming language knowledge, I would not be able to accomplish my task.

Lastly, I wish to take this opportunity to sincerely acknowledge my family for the assistance they provided me through my entire like and in particular, I must acknowledge my parents, Mr. Lim Soon Huat and Madam Chin Mei Har, because without those passion, support and motivation, I would not have finished this project.

Abstract

Global health crisis has been triggered due to the outbreak of coronavirus disease 2019 (COVID-19) pandemic. Malaysia. In order to prevent the outbreak of Corona virus, Malaysian government is implementing Standard Operating Procedure (SOP) which apply to all the citizen of Malaysia. Citizen of Malaysia need to follow the SOP in all of the place as long as people is inside the Malaysia region. People need to maintain one meter among each other when waiting for their turn during the queue due to SOP. Recently, the INTI students are required to back to campus to study. It could be wasting a lot of time for students to follow SOP while waiting for their food in the canteen. The people inside the canteen are also being limited due to SOP. People are also encouraged to do the cashless transaction. In order to solve this problem, we will implement a food ordering system which makes student can get their food without waiting which is Tapau at Inti. The main feature of Tapau at Inti is it is a food ordering system which can saved the data of the customer or student inside a database which is sqlite3 and able to be retrieved by the admin or staff. It is a very convenient tool to store the data which makes the customer can order by using the laptop or mobile phone. This is a web-based application which is provided with a lot of attractive images with advance tools which is interactive. For example, student can make comment to the public. The user can share the recipe to their friend. This project aims to develop a food order web-based application for students and staff of INTI college. To enable students or staff to make orders online anywhere and anytime to reduce close contact and reduce waiting time. To provide a free user-friendly and easily navigable food ordering system application to students and staff of Inti college. The methodology chosen is the Waterfall model as it is more user-friendly, and the full scope of the project is known at the beginning. The expected research outcome is that the students at Inti can order online anywhere and anytime without facing hassle of waiting for long queue in the canteen. The system developed consist of one mobile application for student at inti and one web-based administration panel for canteen admin.

Table of Contents

List of Figures

List of Tables

CHAPTER 1

SYSTEM OVERVIEW

1.2 Introduction

Nowadays, there are still a lot of cases of Covid 19 in Malaysia. In order to prevent the outbreak of Corona virus, Malaysian government is implementing Standard Operating Procedure (SOP) which apply to all the citizen of Malaysia. Citizen of Malaysia need to follow the SOP in all of the place as long as people is inside the Malaysia region. People need to maintain one meter among each other when waiting for their turn during the queue due to SOP. Recently, the INTI students are required to back to campus to study. It could be wasting a lot of time for students to follow SOP while waiting for their food in the canteen. The people inside the canteen are also being limited due to SOP. People are also encouraged to do the cashless transaction. In order to solve this problem, we will implement a food ordering system which makes student can get their food without waiting.

In recent year web-based food ordering system is become a trend due to the pandemic of Covid 19. A web-based food ordering system which is implementing can ensure the students can buy the food at everywhere and every time. It is more convenient compared to traditional food ordering system. A web-based ordering system is software that lets any food seller receive and manage orders through online platform. An online food ordering system generally has a website or app that allows customers to view the menu and place an order and an admin site that enables the restaurant to receive and fulfil customer orders. (Square, 2020)

The first aim of this food ordering system is developing a food order web-based application for students and staff of INTI college. Second aim is enabled students or staff to make orders online anywhere and anytime to reduce close contact and reduce waiting time. Student can buy the food early then they can get their food straight away after going to the canteen. Third aim is providing a free user-friendly and easily navigable food ordering system application to students and staff of Inti college. It could be more troublesome for student if the food ordering system is difficult to use. This food ordering system will also concern about the security issue and privacy of the students. Rating system will be created for the improvement of this food ordering system. Student feedback is very important for our developer to make some changes on our application. This food ordering system will be zero advertisement to ensure good experience of the students.

1.3 Problem Statement

Recently, Inti college does not have any online system to make food orders online.

Students and staffs have to wait for so long to wait for their turn to buy food because they need to follow SOP at the same time.

Most of the available online system such as Air Asia Deliveroo and Shopee food is either not user-friendly, highly charged, or lack promotion.

1.4 Detailed Research Question

1. Who is the target user of the food ordering system?
2. How useful of the food ordering system and the secure of the data of food ordering system?
3. How user friendly and navigability of the food ordering system.

1.5 Keywords

Covid 19, Standard Operating Procedure, pandemic

1.6 Project Title

Tapau at Inti

1.6 Objectives:

1. To develop a food order web-based application for students and staff of INTI college.
2. To enable students or staff to make orders online anywhere and anytime to reduce close contact and reduce waiting time.

3. To provide a free user-friendly and easily navigable food ordering system application to students and staff of Inti college.

1.7 Client, Audience and Motivation

The client of this web system consists of:

- Canteen Admin
- Customer

This system allows the canteen admin to manage the order more easily and quickly. This system can reduce the number of staff needed in canteen and reduced labor work involved. The canteen admin no need to receive phone call to receive the order that will lead to confusion or mistake during preparing the food. (Mayurkumar, 2015). Customer such as students or teacher can order the food without waiting for the long queue since the speed of execution of online ordering system is very fast. The system is very easily to use for both the canteen admin and customer. The customer can order the food from anywhere and anytime. So, the customer can get the food immediately after reaching the canteen. My motivation to do this project is because during this pandemic, it could be a hassle and time consuming for INTI students to queue while following the SOP at the same time, so this application can create an online platform for students to ordering food anywhere and anytime and get the food straight away without having much problem at the canteen.

1.8 Primary Research Plan

The primary research plan involves creating and executing one mobile application to Inti students and canteen Admin. Canteen admin can make use of the online food ordering system to view the order from the customer. Student can use mobile application to order food. Development methodology that is chosen to develop the web system is waterfall model/ In this methodology, the project is completed in a continuous flow. This implies that any phase in the development process begins only if the earlier phase is ended. The project is separated into a sequence of tasks, with the highest-level grouping referred to as phases. (Sherman, 2015). The main reason the waterfall model is chosen is because it is more easy to apply and it is more appropriate to novice programmers. Our project, which is Tapau at Inti, will not always change. Software that we want to design which is an Tapau at Inti is not a new system that requires changes. There are already a lot of existing Tapau at Inti. It is also not a long-term and complex project because we just make a simple project. We can also make our own decisions as a newbie programmer in the development process. Progress is more efficiently tracked as the full scope of the project is known from the beginning. Another reason we chose the Waterfall model is because a project's clients may only need to be devoted massively in the beginning scoping phase and at the delivery phase in this project (Agile,2020) The waterfall model's sequential phase is visualized in Figure 1.1.

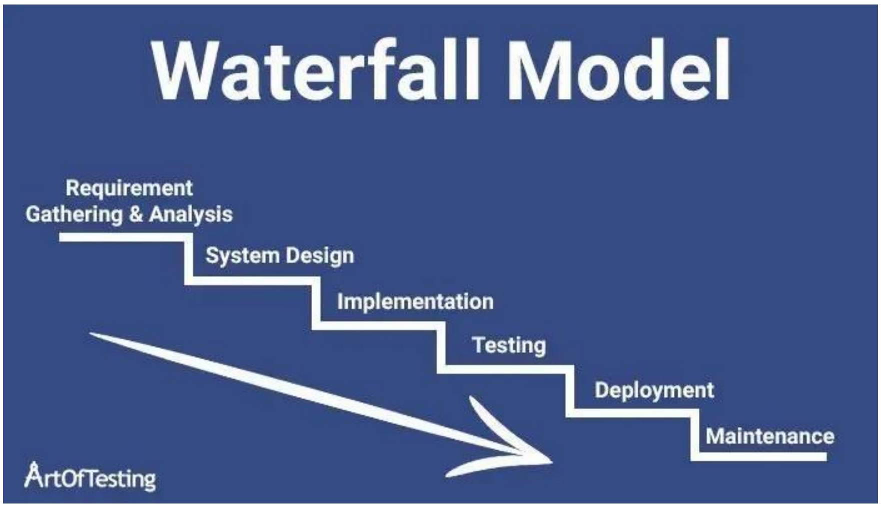


Figure 1.1 Waterfall Model in Software Engineering (Rana,2020)

1.9 Summary

Topic of Chapter 1 discussed about the introduction of Tapau at Inti project. The reason for implementing this project and how is the project is implemented is discussed in the introduction. The aim of this project is also proposed in this chapter. The first aim of this food ordering system is developing a food order web-based application for students and staff of INTI college. Second aim is enabled students or staff to make orders online anywhere and anytime to reduce close contact and reduce waiting time. Student can buy the food early then they can get their food straight away after going to the canteen. Third aim is providing a free user-friendly and easily navigable food ordering system application to students and staff of Inti college. It could be more troublesome for student if the food ordering system is difficult to use. This food ordering system will also concern about the security issue and privacy of the students. The waterfall model is used in this project because of its user-friendliness.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Introduction A literature review is a study of educated references on a certain topic. It outlines current facts, allowing you to know related ideas, approaches, and gaps in the existing research. (McCombes, 2021). In this chapter we will discuss review on existing food ordering system, review on web application Python Graphic User Interface, review on Captcha, and review on system development life cycle, (SDLC). For the review on existing system. Comparison between the existing food ordering system's feature and limitation is made to determine the feature that can be improve in this food ordering system. For the review on web application Python Graphic User Interface, we compare the web application graphic use interface to determine which framework is most suitable to be used in this project. For the review on Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA), the type of CAPTCHA is compared to determine which type of CAPTCHA is most suitable to use in this project. For the review on SDLC, SDLC model is compared to determine which model is which suitable to be used in this project.

2.2 Review on Existing System

One of the existing food ordering systems is foodpanda. One of the features of foodpanda is the user has a relatively smooth ordering and payment process. Another feature of foodpanda is users can easily use the intuitive user interface. (Joycelyn, T.) The limitation of foodpanda is late delivery. Some delivery drivers mark the order as delivered after a very long time. Second drawback of foodpanda is some people face the problem of not logging in back after logging out. Third disadvantage of foodpanda is users cannot cancel the order immediately after ordering the food. Users have to make conversation with the rider and sometimes the rider will reply late. (Trustpilot, n.d.)

Second existing food ordering system is Grab food. One of the features of Grab food is can arrange the order for late. For, example, user can make a Grab food order to deliver in the future up to two days later. The limitation of Grab food is food refund policy. One user paid for extra toppings but did not get them in his food. He ordered in the evening and reported to Grab the next day. Grab declined his request as they have an 8-hour policy, any report later than 8 hours of receiving food, user will not get a chance of refund. (Truspilot, n.d.)

Third existing food ordering system is Airasia food. First After choosing what you want to order you can pay with a credit card and get discounts with vouchers or big points. The good news is you've accumulated a lot of big points and don't have anywhere to spend it. 500 big points can be exchanged for RM10. The limitation of Airasia food is bad user experience AirAsia food probably had bad user interfaces for its ordering page. The user wanted to order a caramel iced latte, but instead of having the usual drop-down menu to choose for the user’s flavour, user had to type out this preference as a note, which is a step that can be easily missed by users. Another limitation of AirAsia is AirAsia food didn’t allow for immediate deliveries and prompted users to choose a scheduled delivery time. If a user wanted to order his food at 1.55 PM, he had to choose the most immediate time slot which is 2.30 PM. The payment process was very slow as there were long buffer times in between the switching of pages on the app. (Joycelyn, T. 2021)

Fourth existing food ordering system is Shopee food, The feature of Shopee food is intuitive interface and reliable payment system. The disadvantage of Shopee food is long delivery time. Another limitation is there is no promotions and the prices of food are higher than other competitors. (Joycelyn, T. 2021)

Fifth existing food ordering system is Deliveroo. The feature of Deliveroo is very enticing and easily navigable customer and user-friendly interface. Similar to other platforms they list all available restaurants in your area including local shops or grocery options. The limitation of Deliveroo is limited coverage. User still have to search a few platforms in order to find a restaurant that works because Deliveroo only work with one third party app. Another limitation is less offer and promotion. (G2, n.d.)

Existing Projects	Features	Problem/Lacking	Citation

Foodpanda	Smooth ordering process and payment process. User interface that is user-friendly	Late delivery. Cannot cancel order instantly.	(Trustpilot, n.d.) (Joycelyn, T.)
Grab food	Can schedule an order for late.	Food refund policy.	(Alanna, 2020) (Truspilot, n.d.)
Airasia food	Can pay with a credit card and get discounts with vouchers or big points.	Bad user experience AirAsia food probably had one Immediate deliveries and prompted users to choose a scheduled delivery time is not allowed. Slow payment process.	(Joycelyn, T. 2021) (Mr. X, 2021)
ShopeeFood	Intuitive interface and reliable payment system.	Long delivery time. With no promotions, prices are higher than competitors	(Joycelyn, T. 2021)
Deliveroo	Very enticing and easily navigable customer and user-friendly interface.	Limited coverage. Less offer and promotion.	(G2, n.d.)

Table 2.1: Comparison between Existing Food Ordering System

2.3 Review on Web application Python Graphic User Interface

2.3.1 What is Web Framework

A web framework is a code library that produces a more rapid and easy web development and equipping typical ways for creating trustworthy, scalable and maintainable web applications. After the early 2000s, professional web development projects always use an existing web framework except in very unusual situations. Web frameworks encapsulate what developers have learned over the past twenty years while programming sites and applications for the web. Frameworks make it easier to reuse code for common HTTP operations and to structure projects so other developers with knowledge of the framework can quickly build and maintain the application. Web frameworks are a concept implemented by Django, Flask, Bottle, Pyramid, Morepath, TurboGears and several other libraries. (Full Stack Python, n.d.)

2.3.2 Functionality of Web Framework

First function of web framework is Uniform resource Locater (URL) routing. Second function of web framework is it can be used in input form handling and validationThird function of web framework is can be used in HTML, XML, JSON, and other output formats with a templating engine. Database connection configuration and persistent data manipulation through an object-relational mapper (ORM). Web application contain web security against Cross-site request forgery (CSRF), SQL Injection, Cross-site Scripting (XSS) and other common malicious attack.

2.3.3 Comparison Between Types of Web-Application Python Framework

2.3.3(a) Pros and Cons of using Django

First advantage of developer use the Django web application is wide range of features and functionality. Instead of writing his own code, he just needs to import the packages for adding functionalities. This not only saves a lot of time for the developers but also allows them to focus on adding advanced functionalities. Moreover, since it is open-source, one can use the packages made by world-class community members. By simply importing the packages, developers can start building applications in no time. (Benchmark, n,d,)

Second advantage of developer using Django is having capability of develop intelligence application. Developer can use Django framework to develop the Intelligence application such as Internet of Things (IoT), Machine Learning and Artificial intelligence. It is also compatible with some powerful machine learning libraries like PyTorch, NumPy and others.
(Benchmark, n,d,)

Third advantage of developer using Django is Django having the capabilities of python. Since Django is based on Python language, it has some of the key benefits of Python. It inherits all of Python's benefits like simplicity, productivity boost, and external library support. All such things speed up the development process. Django make it easier for programmers to develop applications with clean, readable, and maintainable code by benefitting from the syntax rules of Python. It also helps developers easily curtail the development time by building custom web applications without the need to write additional code. (Benchmark, n,d,)

Fourth advantage of using Django is Django has faster development time in creating project. Developers can thus utilize it to rapidly create Minimum Viable Products (MVP) and get an early-mover advantage over the rivals. (Benchmark, n,d,)

First disadvantage of using Django is lack of convenience. A big reason why many programmers dislike Django is that it lacks conventions. As opposed to other frameworks like Ruby on Rails, everything needs to be explicitly defined in Django. This leads to configuration boilerplate and, therefore, a slowed development process. It makes it difficult for developers who are used to working with frameworks with convention over configuration. (Benchmark, n,d,)

Second disadvantage of Django is not appropriate for smaller scale projects. The Django web framework comes lots of code that takes the server's processing and time. This can take its toll on low-end websites that run on very little bandwidth. That is why it is not suitable for projects and products with only a few features and requirements. (Benchmark, n,d,)

Third disadvantage of using Django is difficult to learn. It comes with a lot of features that cannot be easily understood by developers. This can be really tough for developers who are making a switch from other languages. Having a different syntax also makes it hard for developers to quickly get a hang of it. Python can thus be hard to master. (Benchmark, n,d,)

No.	Advantage	Disadvantage
1	wide range of features and functionality.	lack of convenience
2	having capability of develop intelligence application.	Not appropriate for smaller scale project
3	having the capabilities of python.	Hard to master
4	faster development time in creating project.	

Table 2.2: Advantage and disadvantage of using Django

2.3.3(b) Pros and Cons of using Web2py

First advantage of using Web2py is Web2py is a flexible, powerful, and extensible web-based application. Web2Py is great for large-scale projects as well as for simple ones and can be handled even by beginners. Despite the sufficient number of built-in features, Web2Py can be extended with external libraries. (DDI Development, 2020)

Second advantage of using Web2py is Web2py is supported and easily adaptable to the needs of any programming developer team. With Web2Py, there will be no situation when it’s necessary to understand and adapt the code written by the previous team. Even if the code is outdated, it will be compatible with updates and the latest version of the framework. (DDI Development, 2020)

Third advantage of using web2py is documentation is written in form of a book which is good for beginners. Web2py documentation does not follow the common pattern of using Sphinx, MkDocs or ReadTheDocs which is goods for inexperienced developers. Although documentation in form of a book is very easy and good for beginners. Turning web2py the most easy and comprehensive framework to learn and also to teach. (Slant, n.d.)

First disadvantage of Web2py is Inflexible plug-in system. Second disadvantage of Web2py is difficulty in managing the system for loading data models. Third disadvantage is no Python 3 support. (DDI Development, 2020)

No	Advantage	Disadvantage
1	Powerful, flexible and extensible web-application	Inflexible plug-in system
2	Supported and easily adaptable to the needs of developer team	Difficult in managing the system for loading data models
3	Documentation is written in form of a book which is good for beginners.	Python 3 do not support

Table 2.3 Advantage and disadvantage of using Web2py

2.3.3(c) Pros and Cons of using Pyramid

First Advantage of using Pyramid is it is easy to configure and flexible. Any framework component can be replaced by any other one. Besides, two different databases or two different URL display methods can be used within the same application or to implement the same functionality. (DDI Development, 2020)

Second advantage of Pyramid is it is easy to use Ajax requests. By using pyramid web-application, sending Ajax requests is easy with XHR views and decorators. Third advantage is Pyramid is having SQLAlchemy support. This is good news for SQL-based projects. SQLAlchemy is a powerful database that makes it easy to create complex queries. (DDI Development, 2020)

First disadvantage of using Pyramid is The Pyramid needs advanced tool preparation, which is time-consuming. The second disadvantage of using pyramid is only highly experienced technicians can properly configure the Pyramid elements. Third disadvantage is SQLAlchemy is not necessary or even useless in case of simple projects. (DDI Development, 2020)

No	**Advantage**	**Disadvantage**
1	Easy to configure and flexible	Needs advanced tool preparation
2	Easy to use ajax requests	Only highly experienced technicians can properly configure the pyramid elements
3	Having SqlAlchemy support.	SQL Alchemy is not necessary in case of simple project

Table 2.4 Advantage and Disadvantage of using Pyramid

2.3.3(d) Pros and Cons of using Flask

First advantage of using Flask is scalable. Size is everything, and Flask’s status as a microframework means that you can use it to grow a tech project such as a web app incredibly quickly. If you want to make an app that starts small, but has the potential to grow quickly and in directions you haven't completely worked out yet, then it's an ideal choice. Its simplicity of use and few dependencies enable it to run smoothly even as it scales up and up. (Deery, 2021)

Second advantage of using Flask is Flexible. Not only is this helpful in terms of allowing your project to move in another direction easily, it also makes sure that the structure won't collapse when a part is altered. The minimal nature of Flask and its aptitude for developing smaller web apps means that it's even more flexible than Django itself. (Deery, 2021)

Third advantage of using Flask is easy to understand for web developer. Like Django, being able to find your way around easily is key for allowing web developers to concentrate on just coding quickly, without getting bogged down. At its core, the microframework is easy to understand for web developers, not just saving them time and effort but also giving them more control over their code and what is possible. (Deery, 2021)

The fourth advantage of using Flask is lightweight. When we use this term in relation to a tool or framework, we're talking about the design of it—there are few constituent parts that need to be assembled and reassembled, and it doesn't rely on a large number of extensions to function. This design gives web developers a certain level of control. Flask also supports modular programming, which is where its functionality can be split into several interchangeable modules. Each module acts as an independent building block, which can execute one part of the functionality. Together this means that the whole constituent parts of the structure are flexible, moveable, and testable on their own. (Deery, 2021)

The first disadvantage of using Flask is do not has a lot of tools. Inevitably there are some downsides to this microframework's lightweight nature. Chief among them is that unlike Django, Flask lacks a large toolbox. This means that developers will have to manually add extensions such as libraries. And, if you add a huge number of extensions, it may start to slow down the app itself due to a multitude of requests.

Second disadvantage of using flask is not easy to learn with a larger Flask app. Because of the fact that development of a web app using Flask can take a variety of twists and turns, a web developer arriving to the project mid-way through can struggle to come to terms with how it's been designed. The modular nature of the microframework that we mentioned earlier can come back to haunt coders, who will have to familiarize themselves with each constituent part.

Third disadvantage of using Flask is maintenance costs. Because it is so versatile in terms of which technologies it can interface with, quite often a company using Flask will incur extra costs of supporting those technologies. For example, if a technology interfacing with your Flask app becomes obsolete or is discontinued, then the company will have to scramble to find a new compatible one. The more complicated the app becomes, the higher the potential maintenance and implementation costs.

No.	Advantage	Disadvantage
1	Scalable and able to grow a tech project such as a web app incredibly quickly.	Do not provide a lot of tools. Need extension.
2	Flexible in terms of allowing programmer project to move in another direction easily and makes sure that the structure won’t collapse when a part is altered.	Not easy to learn
3	The microframework is easy to understand	Higher maintenance cost.
4	Lightweight because does not rely on a large number of extensions to function.	

Table 2.5 Advantage and Disadvantage of using flask.

2.3.3(e) Conclusion for Comparison between Web-Application Python Framework

After consider all the pros and cons of the web-application python framework, Django is being selected as the development tool in this project. This is because this project required to complete in short period, the Django has faster development time in creating project. It also has capabilities of python. This ability can make the development time faster. Django also has wide range of feature and function. Instead of writing Django’s code, developer can just import from the packages and saves a lot of time. So, Django is chosen for its efficiency to develop project.

2.4 Review on System Development Life Cycle (SDLC)

System Development Life Cycle (SDLC) SDLC or the Software Development Life Cycle is a strategy that produces software with the high quality and low cost in the fastest time possible. SDLC provides a well-organized flow of stages that help a business to develop high-quality software which is well-tested and designed for production use faster. The SDLC involves six phases which have been discussed in the introduction. Famous SDLC models include the waterfall model, spiral model, and Agile model. (Altvater, 2020)

SDLC works by decreasing the cost of software development while simultaneously improving quality and decreasing production time. SDLC achieves these apparently divergent goals by following a plan that removes the regular traps of software development projects. That plan begins by evaluating existing systems for flaws. (Altvater, 2020)

Next, SDLC describes the requirements of the new system. After that, SDLC makes the software through the stages of analysis, planning, design, development, testing, and deployment. By expecting costly errors like forgetting to ask the end-user or client for feedback, SLDC can eradicate duplicate work and after-the-fact fixes.
(Altvater, 2020)

It’s also essential to understand that there is a strong emphasis on the testing phase. As the SDLC is a repetitious methodology, code quality needs to be ensured at every cycle. Many organizations tend to pay little effort on testing while a focus strongly on testing can save them a lot of redundant work, wasted time and money. (Altvater, 2020)

2.4.1 Comparison between System Development Life Cycle

Comparison is done to determine the best SDLC which will be implemented in this project. Agile model and waterfall model are selected to do the comparison. These two System development life cycles are compared to decide the most suitable technique to complete our application more efficiently and effectively.

2.4.1(a) Advantages of Agile Model

There are a few advantages of the Agile model. The first advantage of using the Agile model is can fulfil the customer satisfaction due to quick, constant delivery of practical software. Another advantage is people and interactions are highlighted rather than technique and application in agile model. The third advantage is clients, developers and testers frequently interact with each other which lead to higher effectiveness in completing project. Fourth advantage is operating software is shown often. Fifth advantage is the agile method of Face-to-face communication. There is always daily collaboration between industry people and developers, continuous engagement to excellence and useful design. Another advantage is there is always adaptation to varying circumstances and even late changes in requirements can be accepted. (360 Logical, n.d.)

2.4.1(b) Disadvantages of Agile Model

There are a few disadvantages of Agile. It is very hard to assess the effort required at the starting of the software development life cycle. There is a lack of focus on required design and documentation. The project can be easily not in track if the customer representative is not precise of the future plan that they want. Only experienced programmers are able to make important decisions required throughout the development process. Hence, only capable programmer can engage in project and make decision. (360 Logical, n.d.)

2.4.1(c) Advantage of Waterfall model

Some of the major benefits of the Waterfall Model are simple and user-friendly, manageable due to the flexibility of the model. Each phase has explicit deliverables and a review process. Another advantage is stages are processed and finished wholly at one time. Another advantage is that it performs well for smaller projects where needs are very well determined. Another advantage is clearly defined phases. Another advantage is well understood future result. Another advantage is tasks are easily organized. Another advantage is the process and results are well established. (Tutorials point, n.d.)

2.4.1(d) Disadvantage of waterfall model

Disadvantage of the waterfall model is that no effective software is created until the very late during the life cycle. There is high risk and uncertainty in this model. The Waterfall model is not a good model for complex and object-oriented projects. It is not a suitable model for long and continuous projects. Not appropriate for the projects where requirements are at a moderate to high chance of changing. Hence, risk and uncertainty are high with this process model. Progress is very hard to determine within stages. Waterfall model cannot accommodate changing requirements. Developer cannot adjust scope during the life cycle in this model. (Tutorials point, n.d.)

Agile Methodology	Waterfall methodology
Separates the project development lifecycle into sprints.	Splits the software development into different phases
Follows an incremental method	Follows a linear sequential process
It is a flexible procedure	It is a fixed structural methodology
Agile can be considered as a grouping of various projects	It will be done as one single project.

It is easy to make modifications because of its flexibility	It is difficult to make modifications
The test plan is reviewed after the completion of each sprint	The test plan is discussed during the testing phase
Project requirements are subject to change	The waterfall is suitable for projects with definite requirements. Changes are not easy. Agile vs Waterfall
Agile is a client-focused methodology. Changes on the product will be according to client's requirements	Waterfall methodology relies on definite project requirements
A fixed project budget is not suitable for the implementation of agile methodology. There may be some change in budget	Since all the requirements and the project scope is defined well, the project budget does not change too much.
High communication and cooperation among the team members	Low communication and cooperation among the team members

The product owner and the project team collect the requirements every day.	All the requirements are defined before starting the project.
Follows an iterative development approach so that planning, development, and testing phases may appear more than once.	Phases are linear and sequential. Once a phase is completed, the next phase starts
Testing is performed in conjunction with software development.	The testing phase comes once the build phase is completed.

Table 2.4 Comparison between Agile methodology and Waterfall methodology.

2.4.1(e) Conclusion for comparison between System Development Life cycle

After analyzing the advantages and disadvantages of Agile model and Waterfall Model, Agile model is mainly used for new systems that will change. However, a newbie programmer cannot make the important decision because only senior programmers can make the decision in the development process. The Waterfall model is not a good model for the project that requires high risk of changes. It is also not suitable for complex or long-term projects. However, it is easy to understand and use. The Waterfall model is chosen because it is more suitable to newbie programmers. Tapau at inti will not always be changed. Software that is designed in a food ordering system is not a new system that requires changes. There are already a lot of existing food ordering system. It is also not a long-term and complex project. The decision can be made as a newbie programmer in the development process. So, the waterfall model is more suitable to be used in this project.

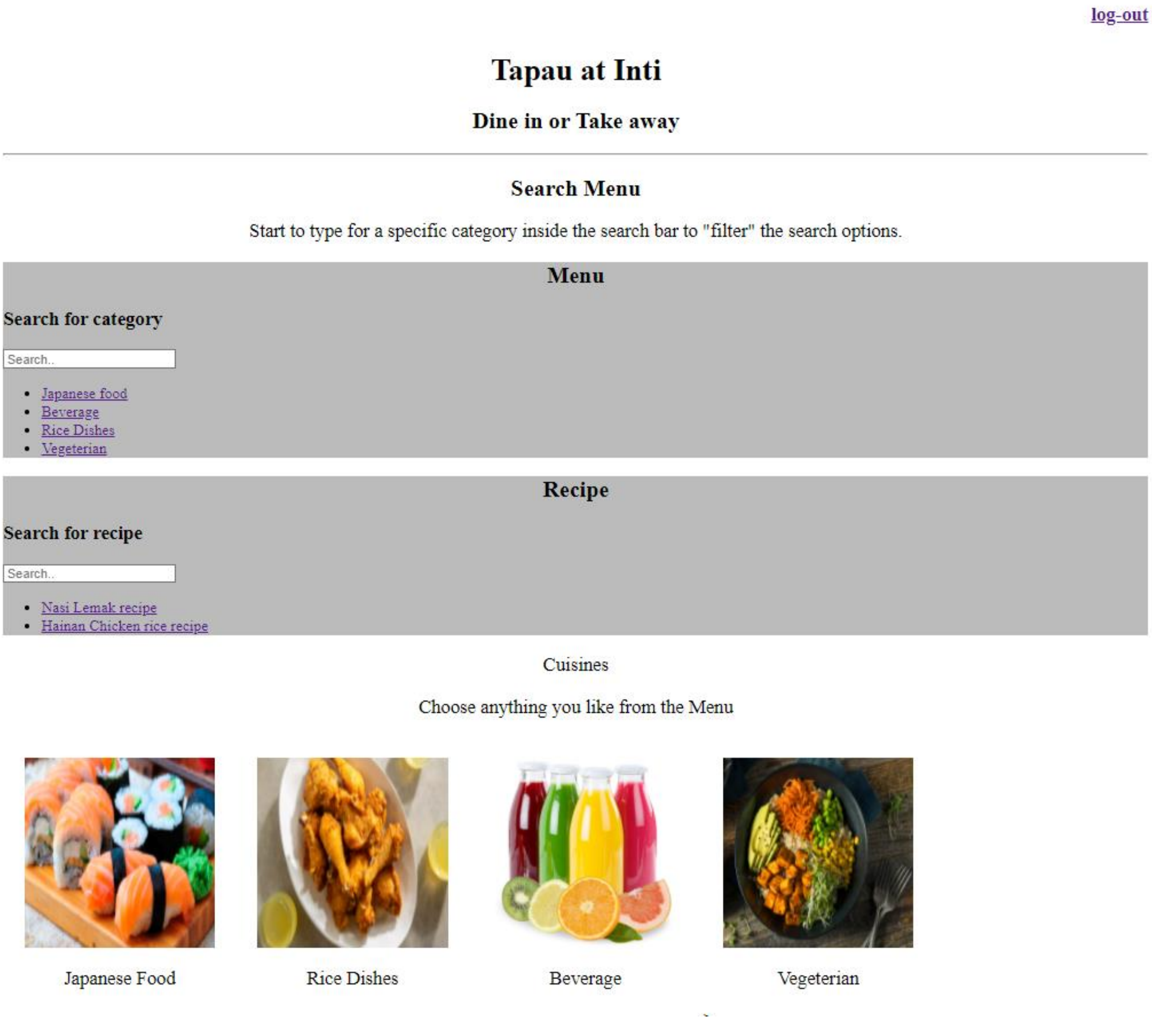
2.5 Summary

Reviews on existing food ordering system, review on Web application Python Graphic User Interface and review on System Development Life Cycle (SDLC) has been discussed. For the review on existing food ordering system, the existing food ordering app’s features and limitations are compared in order to know what features that can be improved in our food ordering application. For the review on Web application Python Graphic User Interface, comparison between the web application GUI framework is made to determine which GUI framework is most suitable to be used in our project. For the comparison of system development life cycle, the pros and cons of Agile model and Waterfall model are compared. Waterfall model is chosen because of its user-friendliness. Chapter three will discuss the waterfall which will be used to design of the system.

CHAPTER 3
RESEARCH MEHODOLOGY

3.1 Food Ordering System

Food ordering system is a system to enable users to make orders online. The users of the system are students, staff and canteen operators. The main reason the system is developed to enable students and staff to make food orders online to avoid crowd during the pandemic season and at the same time allowing the students to fully adhere SOPs. In order to solve this problem food ordering system which is Tapau at Inti is designed and implemented. The main feature of Tapau at Inti is it is a food ordering system which can saved the data of the customer or student inside a database which is sqlite3 and able to be retrieved by the admin or staff. It is a very convenient tool to store the data which makes the customer can order by using the laptop or mobile phone. This is a web-based application which is provided with a lot of attractive images with advance tools which is interactive. For example, student can make comment to the public. The user can share the recipe to their friend. The Food Ordering System is very user-friendly because Tapau at Inti use language which very simple. The user can navigate easily by using the navigation provided. The system can be search for the category of food. User can also navigate by clicking the image of food. Before entering the main page, user need to navigate to register first. Then, user can login into the system by using login page. There are four category of food which are Japanese food, rice dishes, Beverage and vegetarian. User can search by category of the food by clicking the list below the search bar. User can also straight away click the image of the food to navigate to page related with that category. Inside the pages of each category, the price of the menu in that category will be showed. User can think about which food that they want. After that, user can navigate to the order form pages to by clicking the order button. User need to enter their name, contact number, Japanese food, Japanese food quantity, Beverage, Beverage quantity, rice dishes, rice dishes quantity, vegetarian, vegetarian quantity, order date, order time, payment method and taking method. The pages will show message which is ‘Please Wait, your order will be arrived soon! The user can also click the comment button to navigate to the comment page. After the user click the comment to public. The message which is nickname of user and comment is posted will be showed. The staff can view the order of the students by registered as an admin and login as an admin. The main page of Food_Ordering_System is illustrated in Figure 3.1.



log-out

Tapau at Inti

Dine in or Take away

Search Menu

Start to type for a specific category inside the search bar to "filter" the search options.

Menu

Search for category

Search..

- Japanese food
- Beverage
- Rice Dishes
- Vegeterian

Recipe

Search for recipe

Search..

- Nasi Lemak recipe
- Hainan Chicken rice recipe

Cuisines

Choose anything you like from the Menu

Japanese Food

Rice Dishes

Beverage

Vegeterian

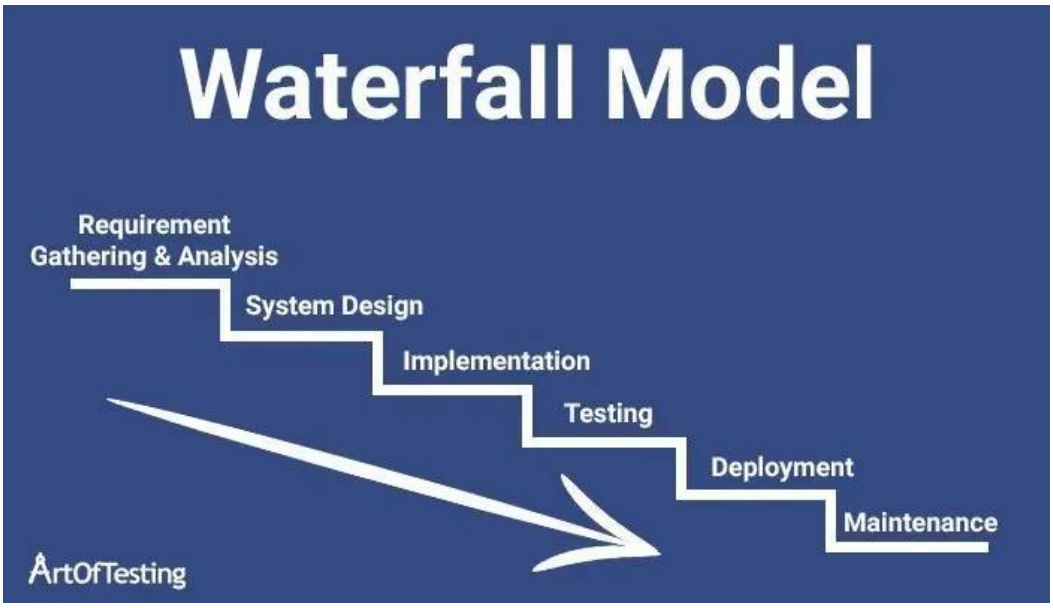
Figure 3.1 Main page of Tapau at Inti

3.2 Waterfall Model

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model.It is very user-friendly. In a waterfall model, each phase must be completed before the next step can begin and there is no overlapping in the stages. The Waterfall model is the earliest SDLC approach that was us for software development. The waterfall Model illustrates the software development process in a linear sequential flow.

This means that any phase in the development process starts only if the previous phase is complete. In this waterfall model, the phases do not overlap. (Tutorials point, n.d.).

The waterfall methodology consists of six phases, which are requirement gathering and analysis, system design, implementation, testing, deployment and maintenance. The processes in each phase will be explained in this section. The waterfall methodology and iteration process flow are visualized in Figure 3.1. The waterfall methodology consists of six stages: requirement gathering and analysis, system design, implementation, testing, deployment, and maintenance. The procedures in each phase will be explained in this section. The waterfall methodology and iteration process flow are illustrated in Figure 3.2



Waterfall model is developed as guidance during web development. Waterfall methodology is chosen because of its user-friendliness. Then in the waterfall model we need to follow its linear flow of sequence. There are six phases of sequential flow of Waterfall Model. First phase is requirement gathering and analysis. Before starting to write the code to design the application, some preparation needs to be made. Requirement in this project will be analyzed and gathered by make some review on existing system, Web application Python Graphic User Interface and System Development Life Cycle. Budget and time will also be discussed in this phase. All possible requirements of the system to be created are obtained in this stage and documented in a requirement specification document. (Tutorials point, n.d.).

Second phase is system design. The requirement specifications from the first phase are investigated in this phase and the system design is prepared. This system design helps in defining hardware and system requirements and helps in defining the overall system structure. Then**,** design phase will made by understanding how to choose the best solution before writing the code by designing the Use case diagram, ER-diagram, flowchart and wireframe. (Tutorials point, n.d.)

After that implementation will be made. In this phase coding take place and the code is integrated to form a whole application. With inputs from the system design, the system is first produced in small applications called units, which are combined in the next phase. Each unit is produced and tested for its functionality, which is referred to as Unit Testing. (Tutorials point, n.d.). Fourth phase is integration and testing. All the parts produced in the implementation phase are merged into a system after testing of each unit. Post integration, the entire system is examined for any faults and failures. (Tutorials point, n.d.). Testing phase require required the test case to examine whether the button or feature can be function.

Fifth phase is deployment of the system. Once the functional and non-functional testing is made; the product is deployed in the client environment or published into the market. In this phase, the Tapau at Inti application will be distributed to real world if it is usable. (Tutorials point, n.d.) Final phase is Maintenance. There are some problems that come up in the client environment. Some better versions and applications are published to fix those problems and enhance the product. Maintenance is done to deliver these changes in the client environment. (Tutorials point, n.d.)

3.2.1 Requirement

This is the first phase of sequential flow of the Waterfall model. In this phase, all requirements of the project are analysed and documented in a specification document and a feasibility analysis is done to check if these requirements are valid. It is essential to consider any limitations and constraints (example, time, budget constraints) which can affect the development process. (Zulqadar, A.) Literature review that we have made before is web application Graphic User Interface review in python. The comparison will be made based on feasibility, efficiency and user-friendliness of the tools. Django is chosen as a development tool because of its fastest development time. Review on the existing system of food ordering system also is made to determine the limitations and features of existing applications. The development tools that were required are visual studio, python, sqlite3, command prompt, Django and crispy forms. The further explanation of the development tool will be explained in 3.7 Development tools. The time to complete this project is allocated properly to be able to complete the time on task. The tool that will be used in this project is Django that is open source and free to use. The information about the business opportunity of Tapau at Inti is gathered. The target user of Tapau at Inti also will be determined in this phase. Suitable System development life cycle also is determined in this phase.

3.2.2 Design

The Design Phase is where programmers look at the many potential solutions and narrow down the choices to determine the most effective and efficient way to construct the solution. The Design Phase answers the questions about "how" programmers will build the best solution. At the end of the Design Phase, programmers will have a logical solution defined. The solution is "logical" because it exists on paper or in a design tool. The design tool that has been used in this project is which will be discussed in 3.5.1 Wireframe of Tapau at Inti. This logical solution is then passed to the Construct Phase, where the logical solution is turned into a physical solution. Construction is similar with implementation which will be discussed in 3.8 implementation. (tenstep, n.d.)

3.2.2 (a) Use Case Diagram

Star UML is used to design the use case diagram. Use case diagrams identify the interactions between the system and its actors. Use-case diagrams illustrate and define the context and requirements of either an entire system or the important parts of the system. While gathering requirements, we create use-case diagrams to capture the system requirements and to present to others what the system should do. (IBM, n.d.). For example, we can see Japanese Food class, Western class and Vegetarian class which will further inherit into different types of Japanese Food, different types of Western food and so forth. We can clearly see that Each class has one game. Has option to choose dine in or take away then further inherit into credit card or cash and finally can pick the time and comment.

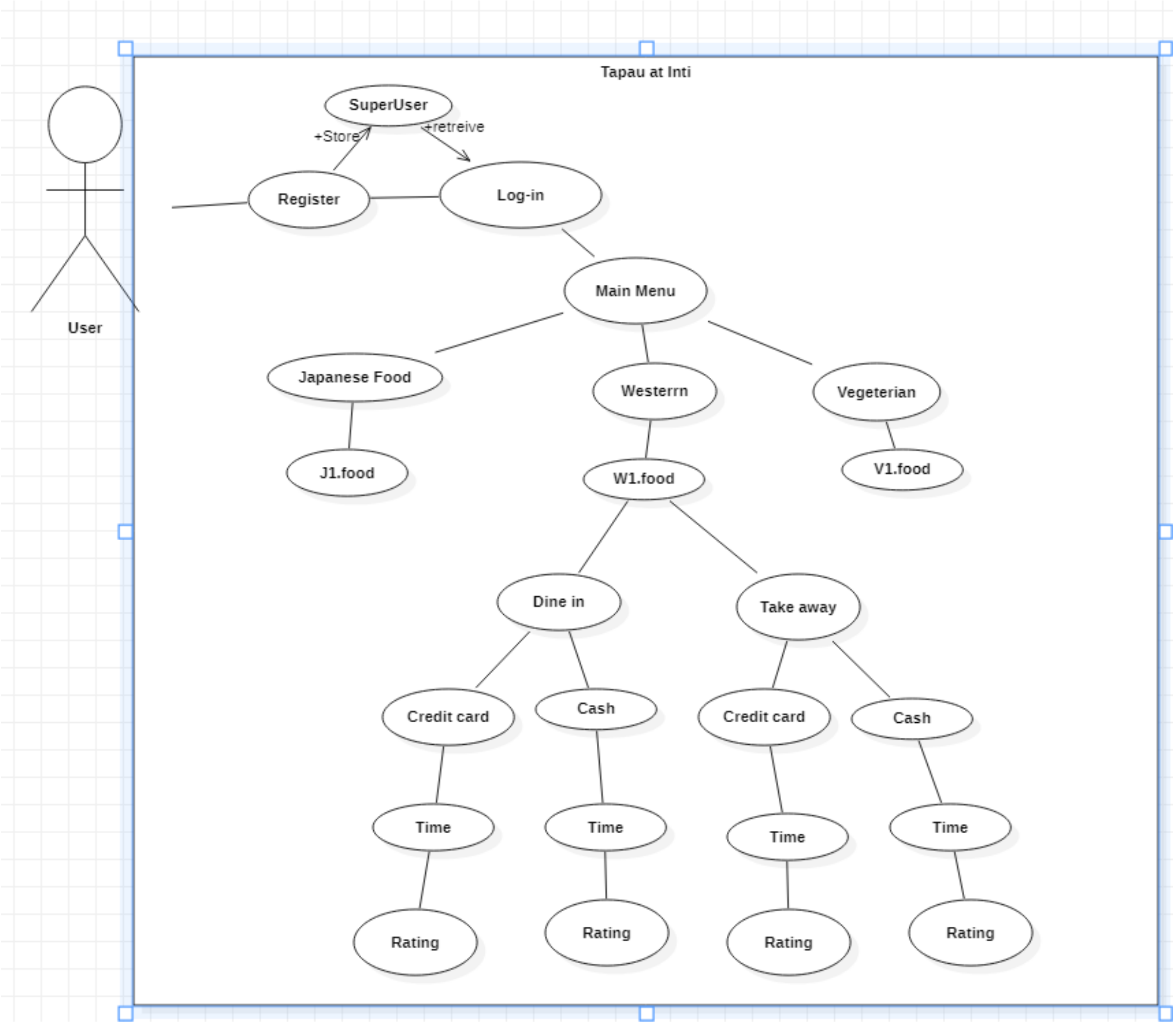


Figure 3.2 Use case diagram of Tapau at Inti

3.2.2 (a1) Explanation of Use Case Diagram

User that is registered will be stored inside superuser and can be retrieved from superuser when log in. User can navigate to main menu after log in. After log in, user can choose between Japanese Food, Western, Vegetarian and other category. User can further choose the food type from the food category. Then after select food category, user can choose between dine in or take away. After choose between dine in and take away, user can choose between credit card and cash. After that, user can choose the time. Then, user can do the rating.

3.2.2 (b) Database Design

We need to do the database connection to maintain the privacy of people and be able to pay for the premium feature. We need to gather information about what is the built-in package that we need to import to create the database connection. We also need to find a software to store the information of the user. It is important to ensure data integrity and safety. In order to do this, we need to design a register page and login page. Our register page requires the user to fill in student id, Password, Re-enter password and contact number. After the data is saved it will be stored inside the superuser. The admin can check the information related to user by open the superuser.

3.2.2 (b1) Entity of Tapau at Inti Database:

User: User_ID. Name, Contact_Number, Gender, Country, Password

Admin: Admin_ID, Name, Contact_Number

SuperUser: User_ID. Name, Contact_Number, Gender, Country, Password

3.2.2 (b2) Entity Relationship Diagram (ERD)

The login System of Tapau at Inti will be illustrated using ER-diagram. The ER-diagram will be illustrated in Figure 3.3.

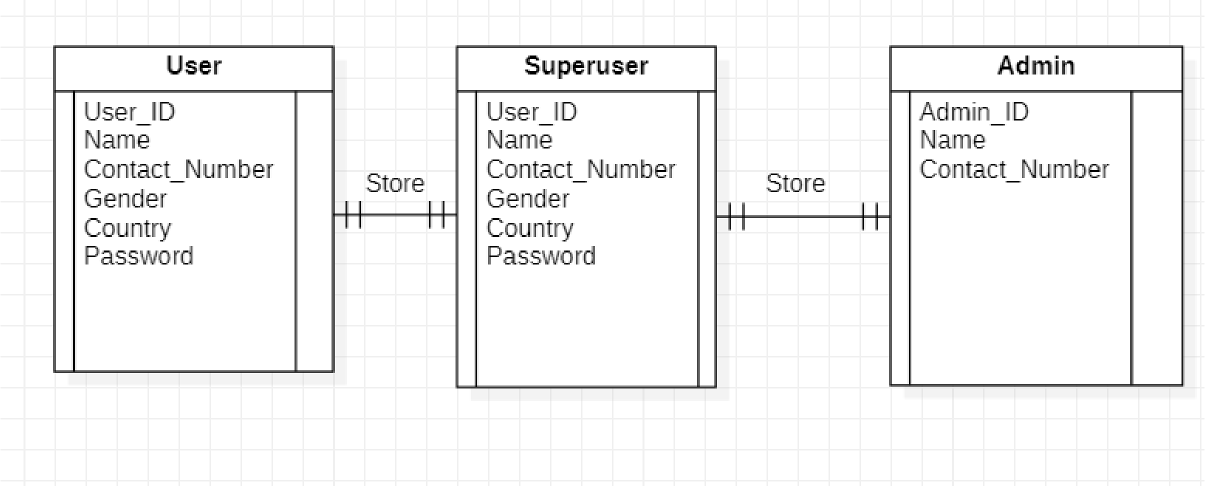


Figure 3.3 ER-diagram of Tapau at Inti

3.2.2 (b3) Explanation of ER-diagram:

User will store the User ID, Name, Email, Contact Number, Gender, Country and password inside Db browser. Email and password of user will be retrieved when user want to login to the system. Admin will store his own Admin ID name and contact number inside Superuser.

3.2.2 (c) Module Design

A modular design is a method for designing a product which is applied to build a final product by merging or combining smaller components that are independent of each other. With the modular design approach, a complex product (for example, a car) can be splited or divided into smaller and simpler components that are independently designed and manufactured. Each of these individual components is then integrated (or assembled) together to form the final product. (Upadhyay, 2020)

Modular design enables one to customize, reuse, upgrade, or maintain product designs. Additionally, the modular product's independent parts follow a regular interface to suit into each other as the final product easily. Moreover, works with non-modular designs (for example, an electric switch or electronic products) are not easy to customize or maintain. (Upadhyay, 2020)

Modular design is usually used in many consumer products like cars, smartphones, computers, and televisions. Another modular design examples include software engineering, software product design, and even large websites (with each webpage as an individual component). A famous example of modularity is LEGO. These plastic toys contain elements that can be simply assembled and reused to develop different final products. (Upadhyay, 2020)

3.2.2 (c1) Overall System Flowchart

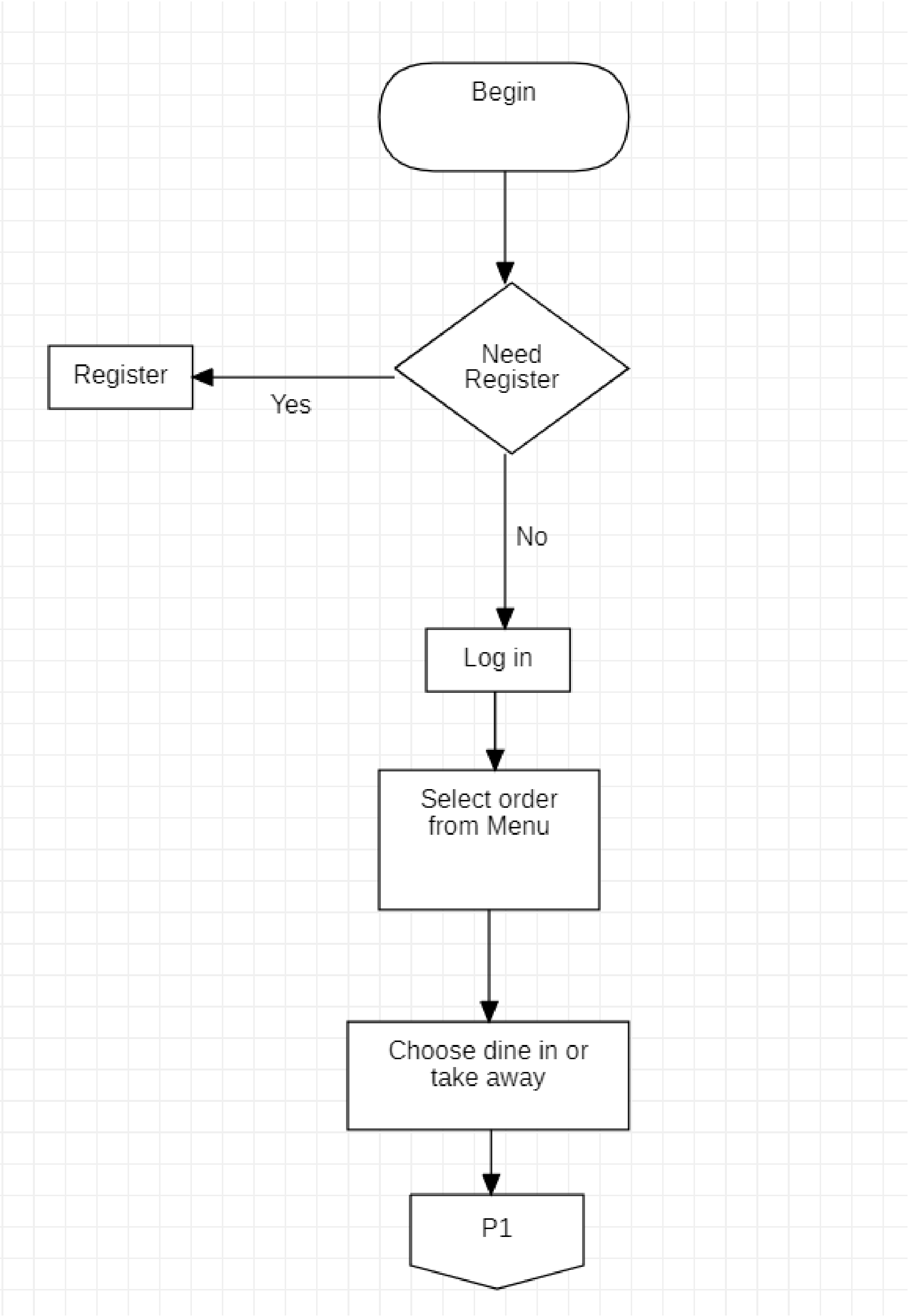


Figure 3.4 Flow Chart of Tapau at Inti

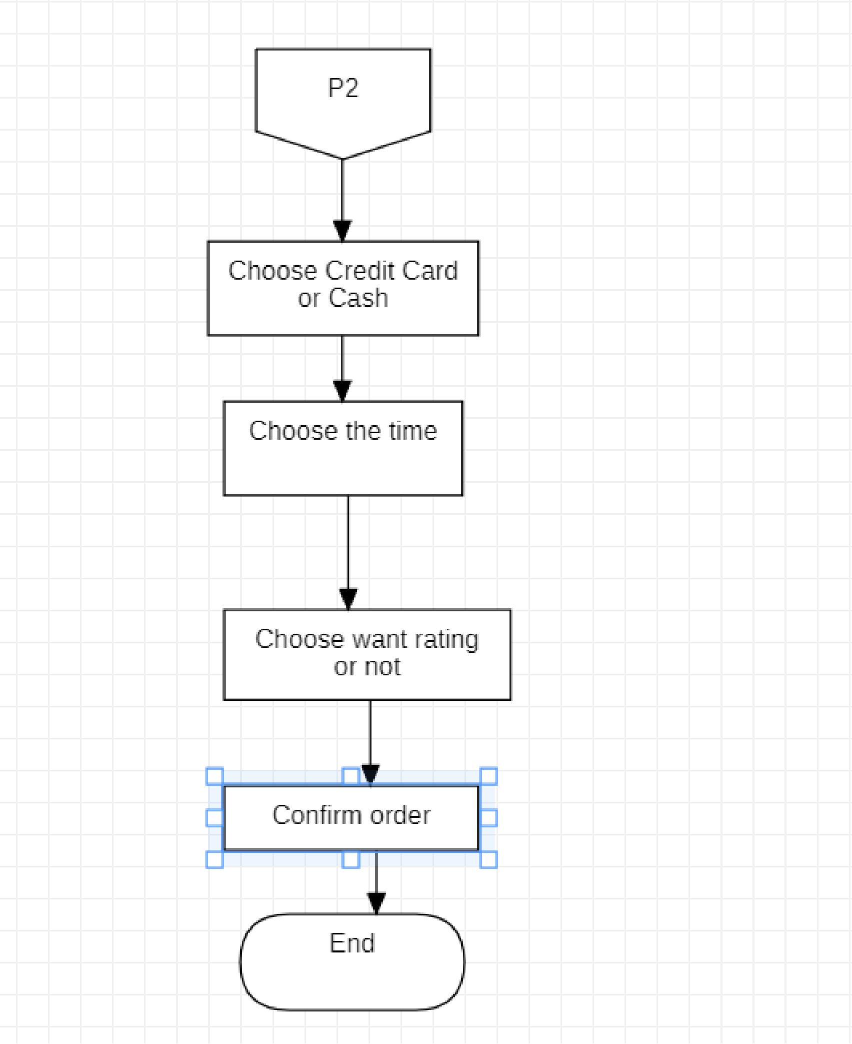


Figure 3.4 Flow Chart of Tapau at Inti

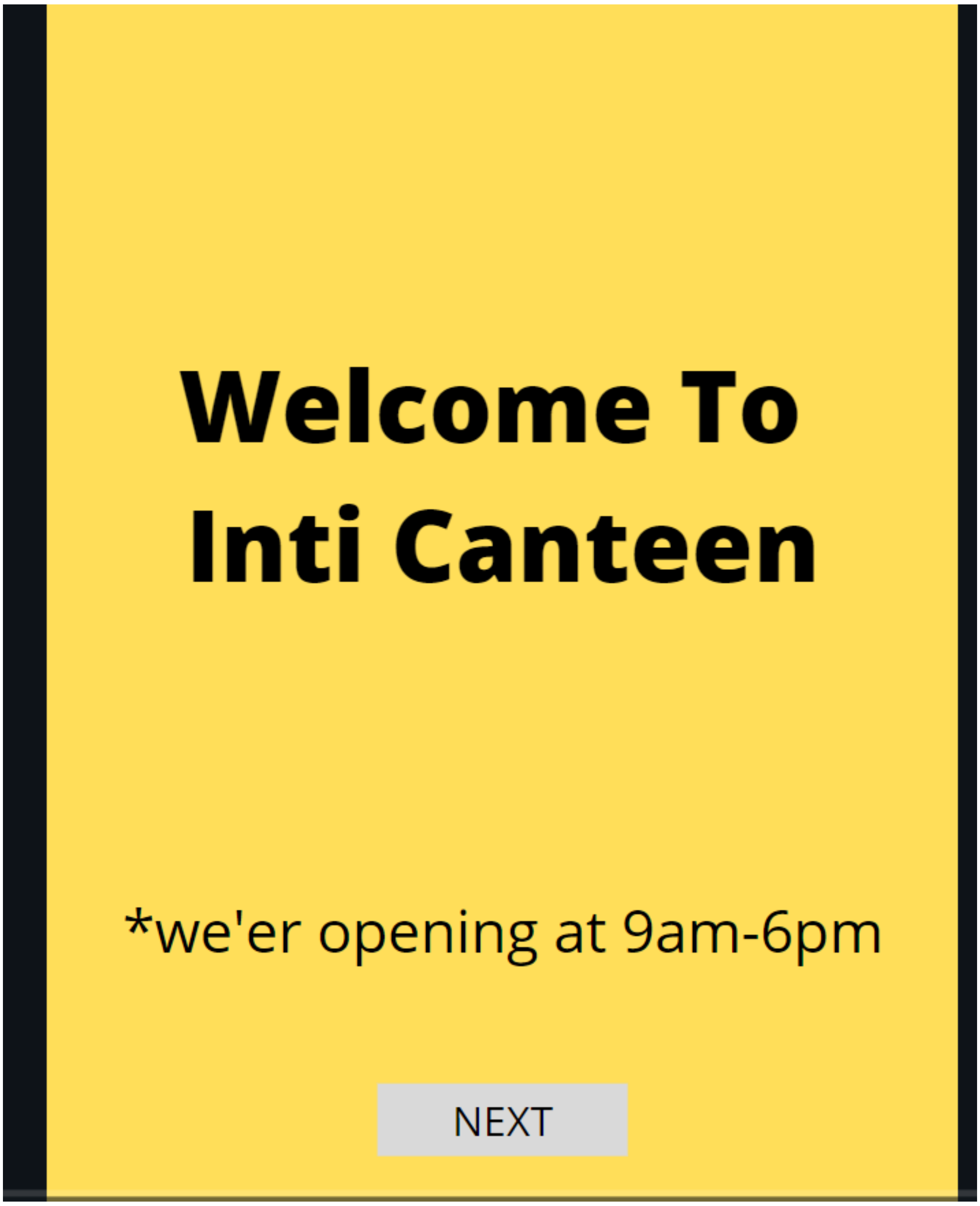
3.2.2 (c2) Explanation of flowchart:

User that is registered will be stored inside superuser and can be retrieved from superuser when log in. User can navigate to main menu after log in. After log in, user can choose between Japanese Food, Western, Vegetarian and other category. User can further choose the food type from the food category. Then after select food category, user can choose between dine in or take away. After choose between dine in and take away, user can choose between credit card and cash. After that, user can choose the time. Then, user can do the rating. Then user can click the confirm order.

3.2.2 (d) User Interface (UI) Sketches

Wireframe of Tapau at Inti is made by using Canva. The wireframe of Tapau at Inti is illustarted in Figure 3.5

3.2.2 (d1) Wireframe of Tapau at Inti



Welcome To Inti Canteen

*we'er opening at 9am-6pm

Figure 3.5 User interface of Tapau at Inti

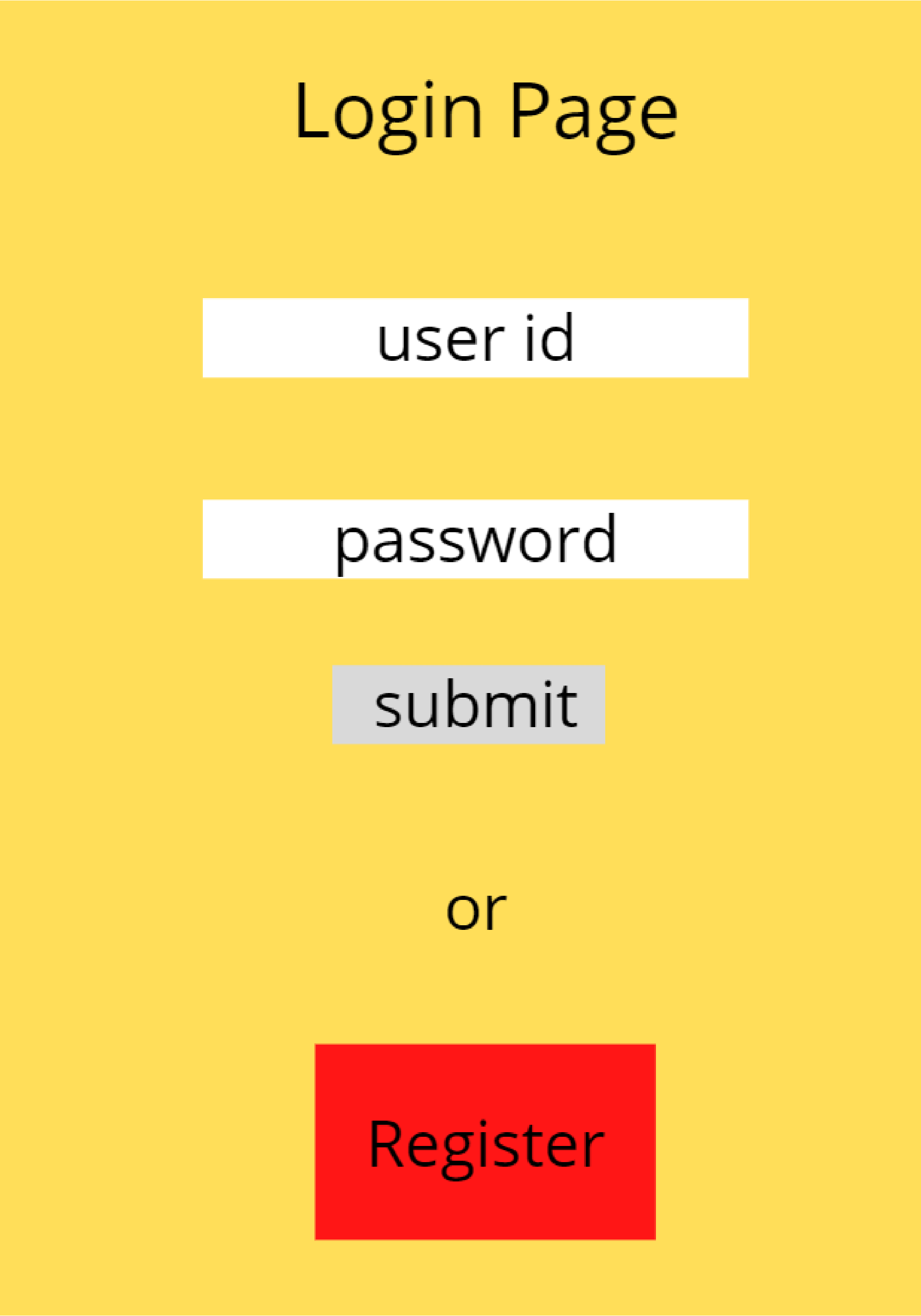


Figure 3.5 User interface of Tapau at Inti

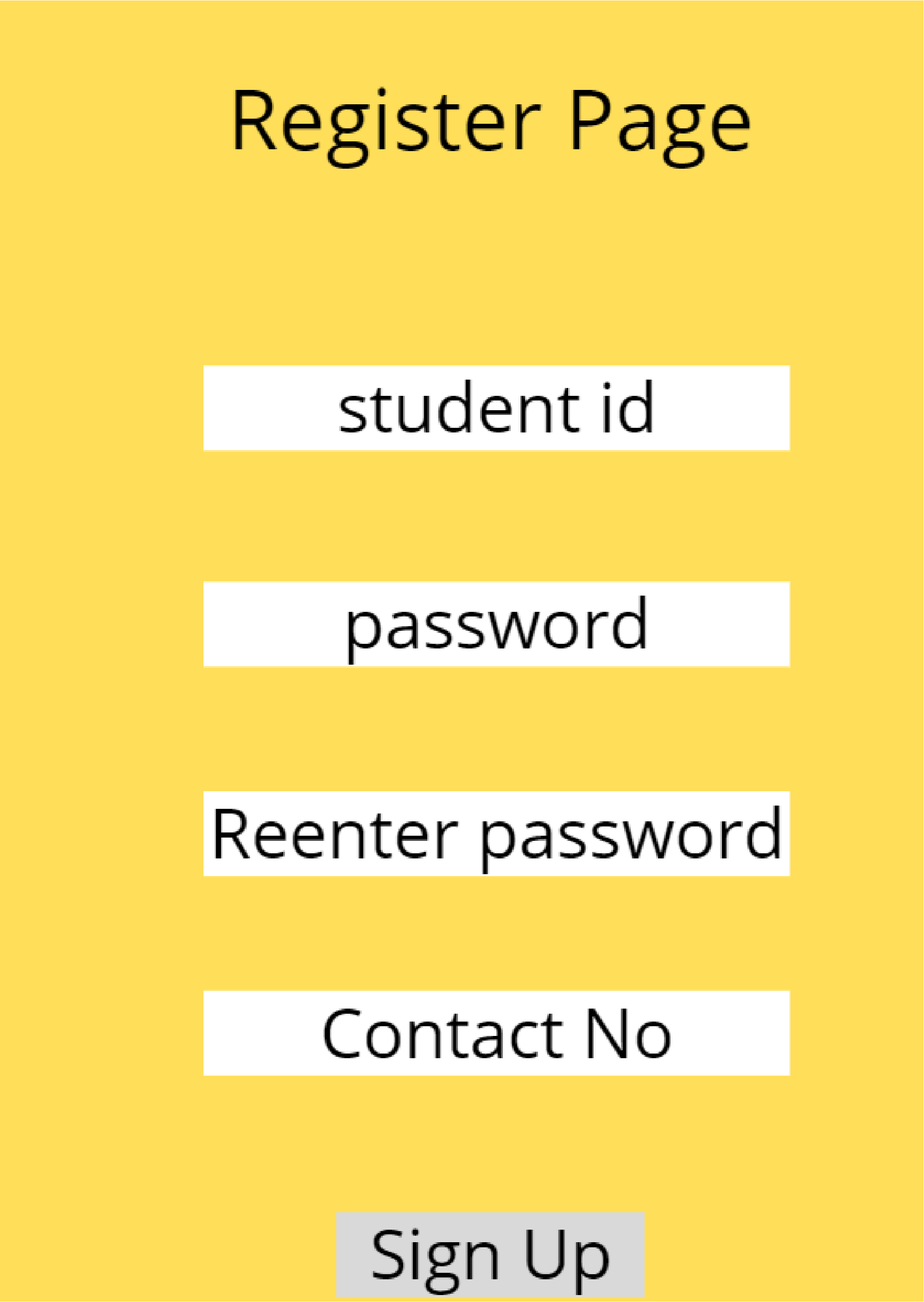


Figure 3.5 User interface of Tapau at Inti

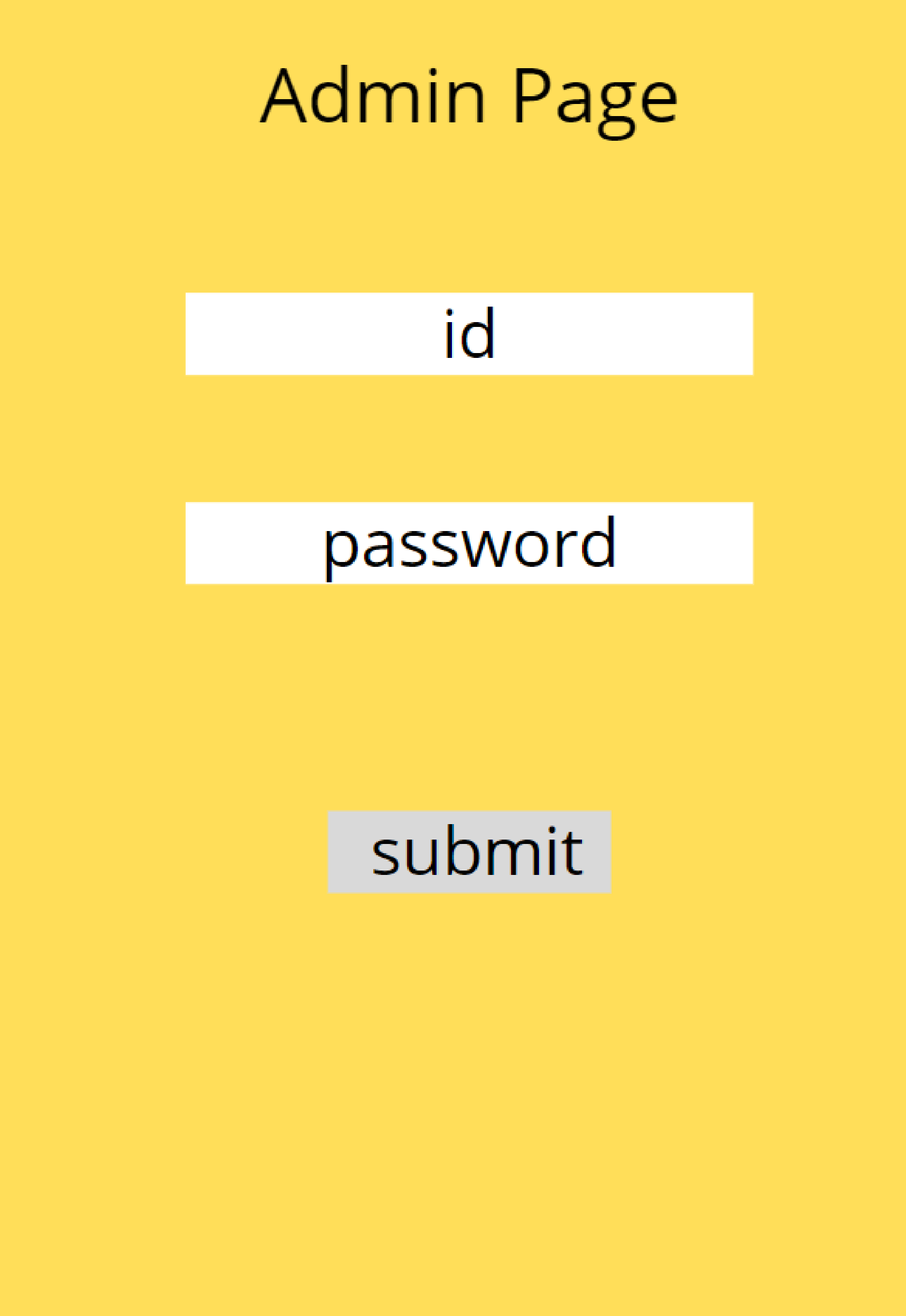


Figure 3.5 User interface of Tapau at Inti

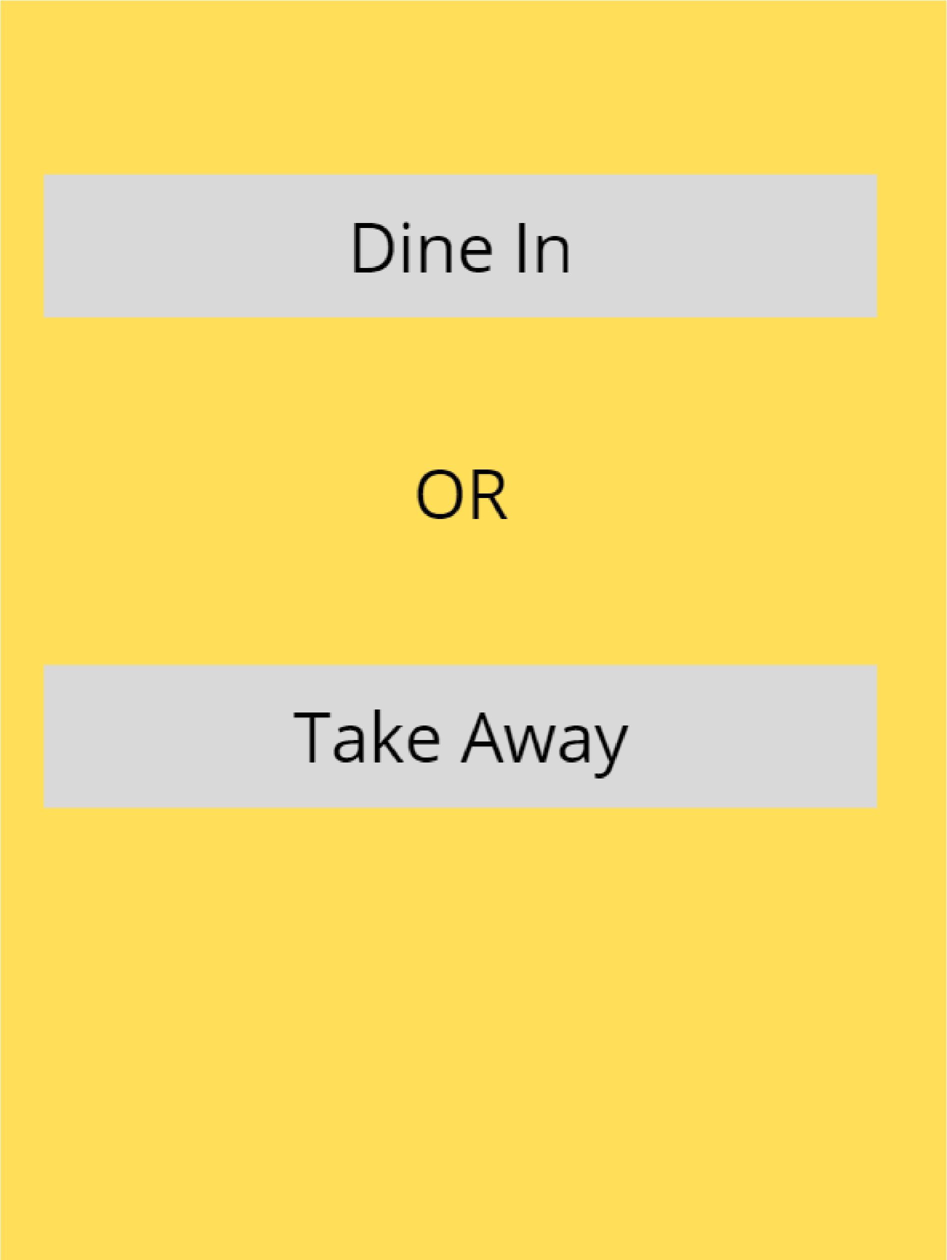


Figure 3.5 User interface of Tapau at Inti

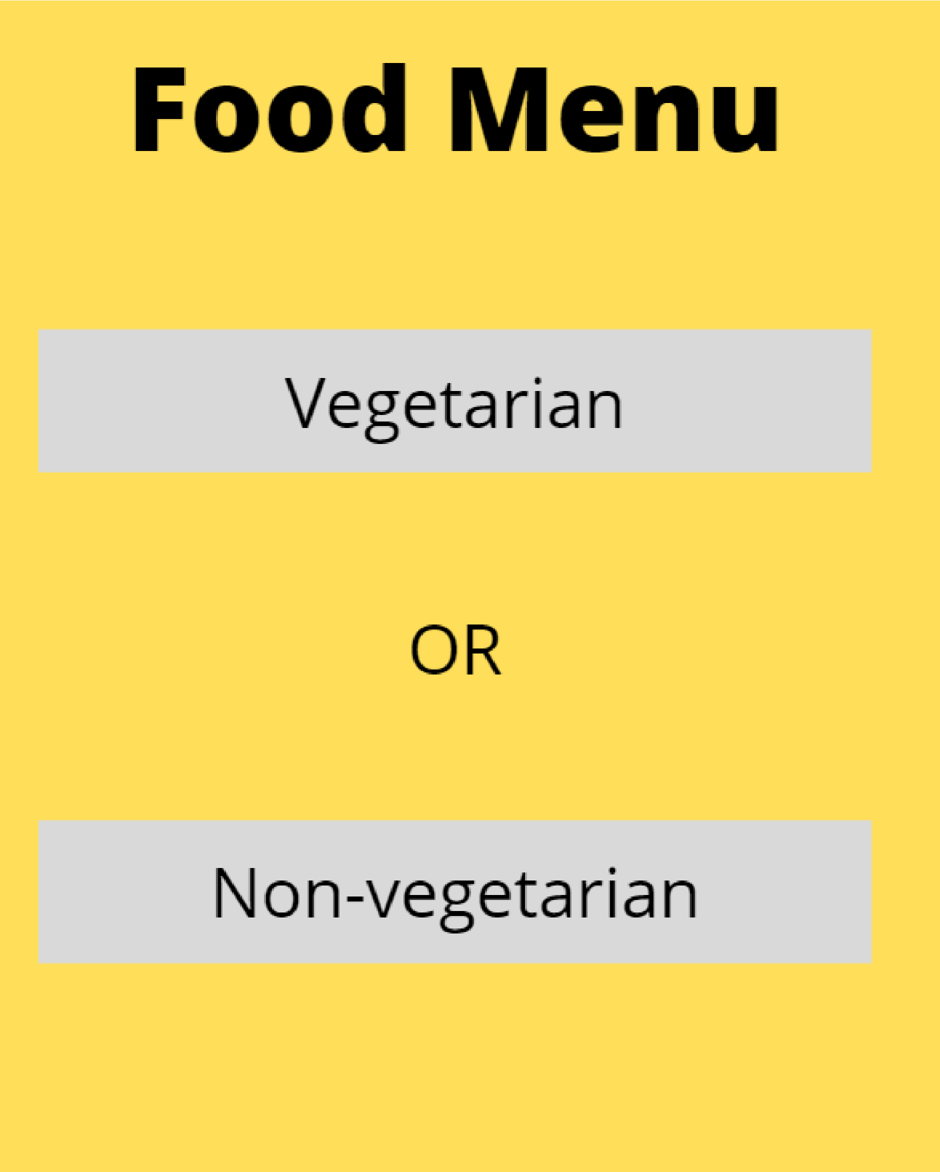


Figure 3.5 User interface of Tapau at Inti

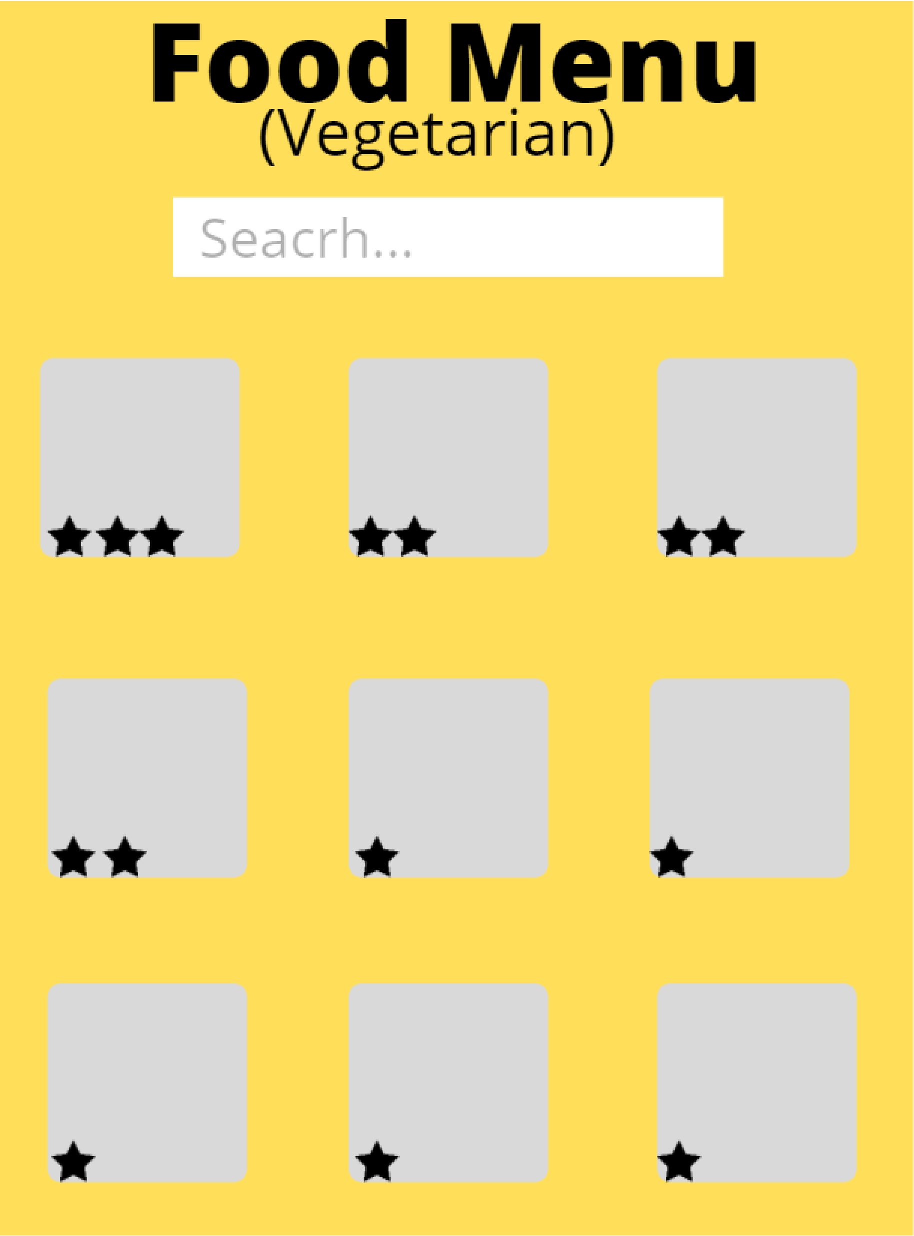


Figure 3.5 User interface of Tapau at Inti

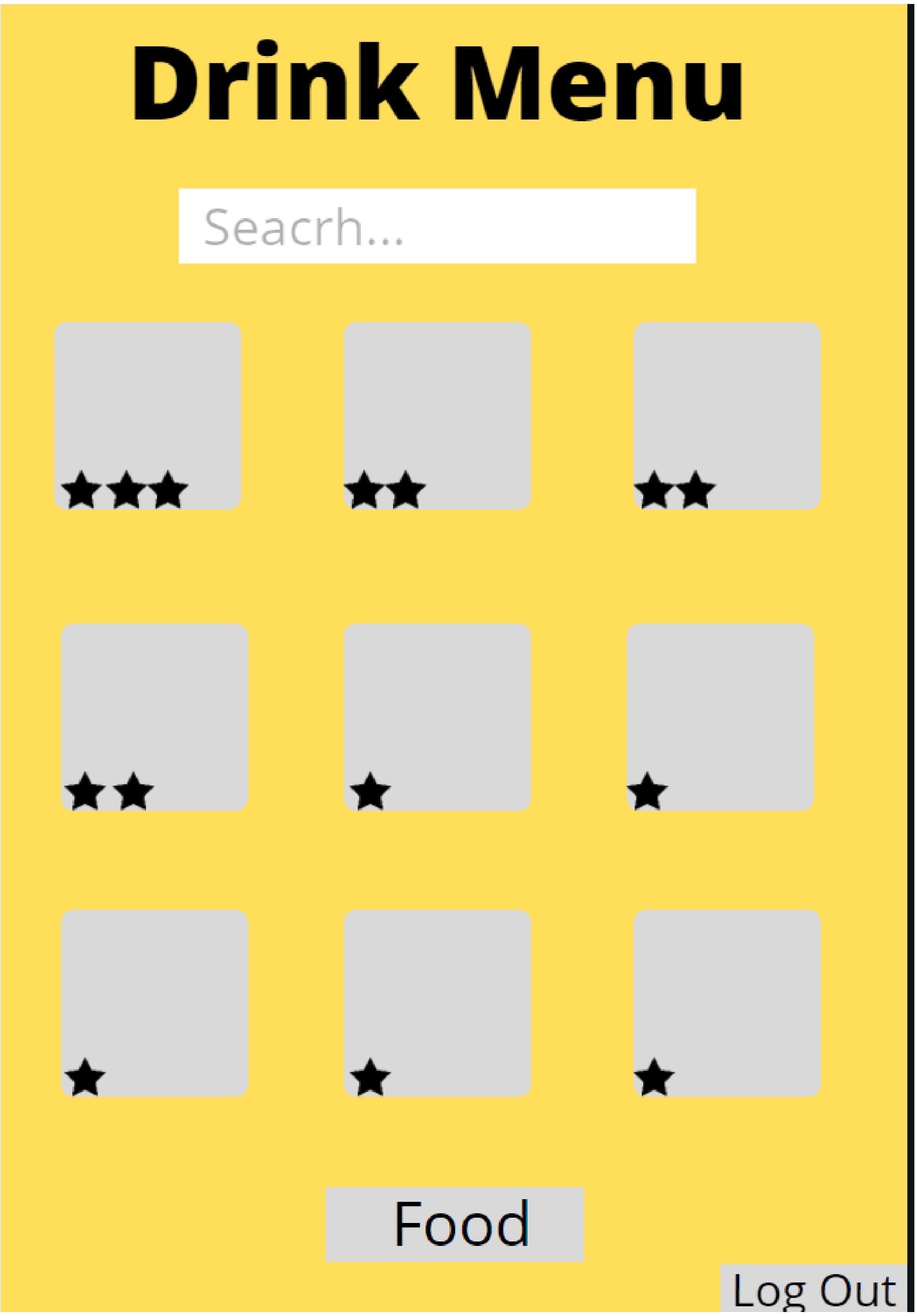


Figure 3.5 User interface of Tapau at Inti

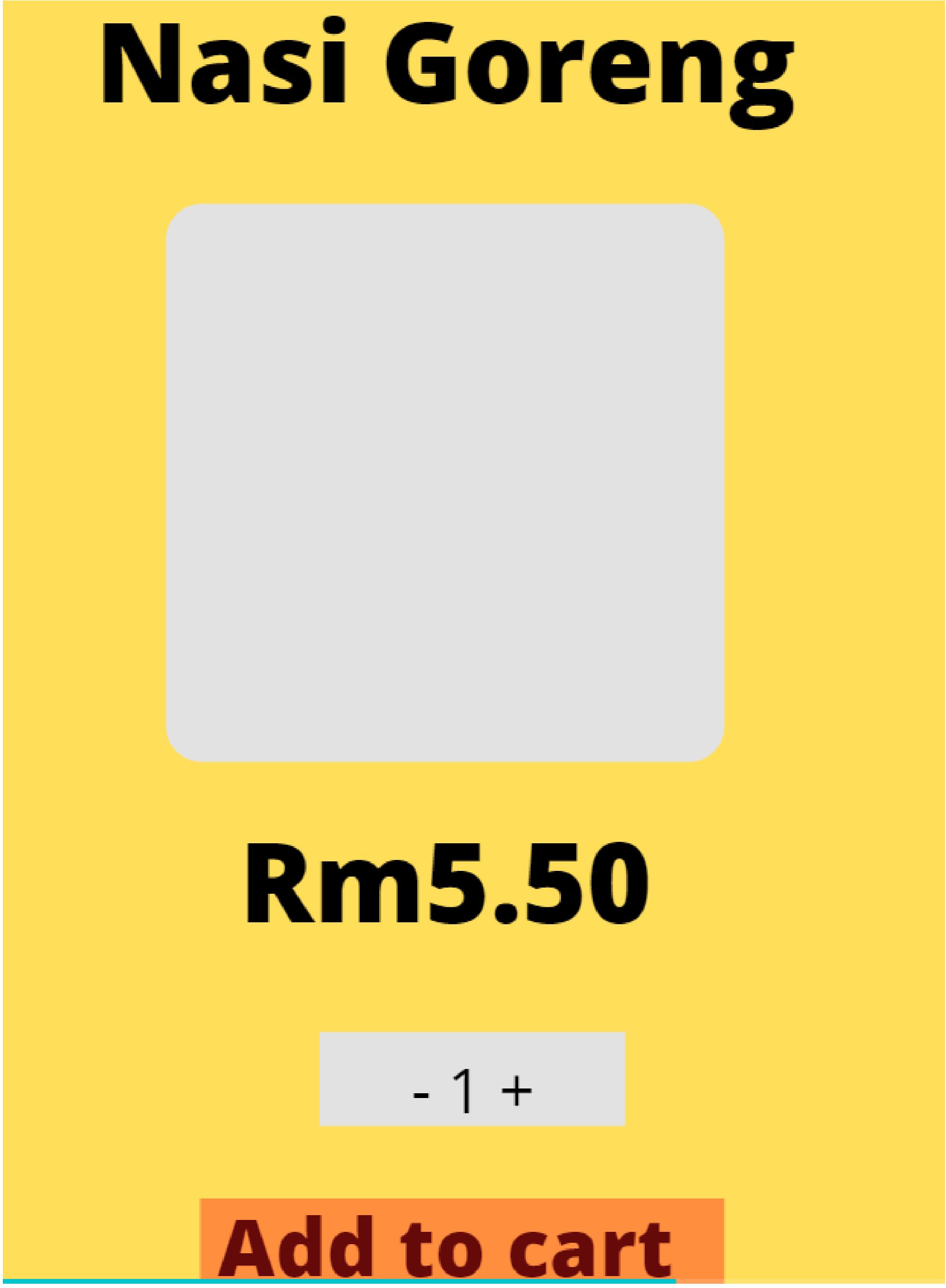


Figure 3.5 User interface of Tapau at Inti

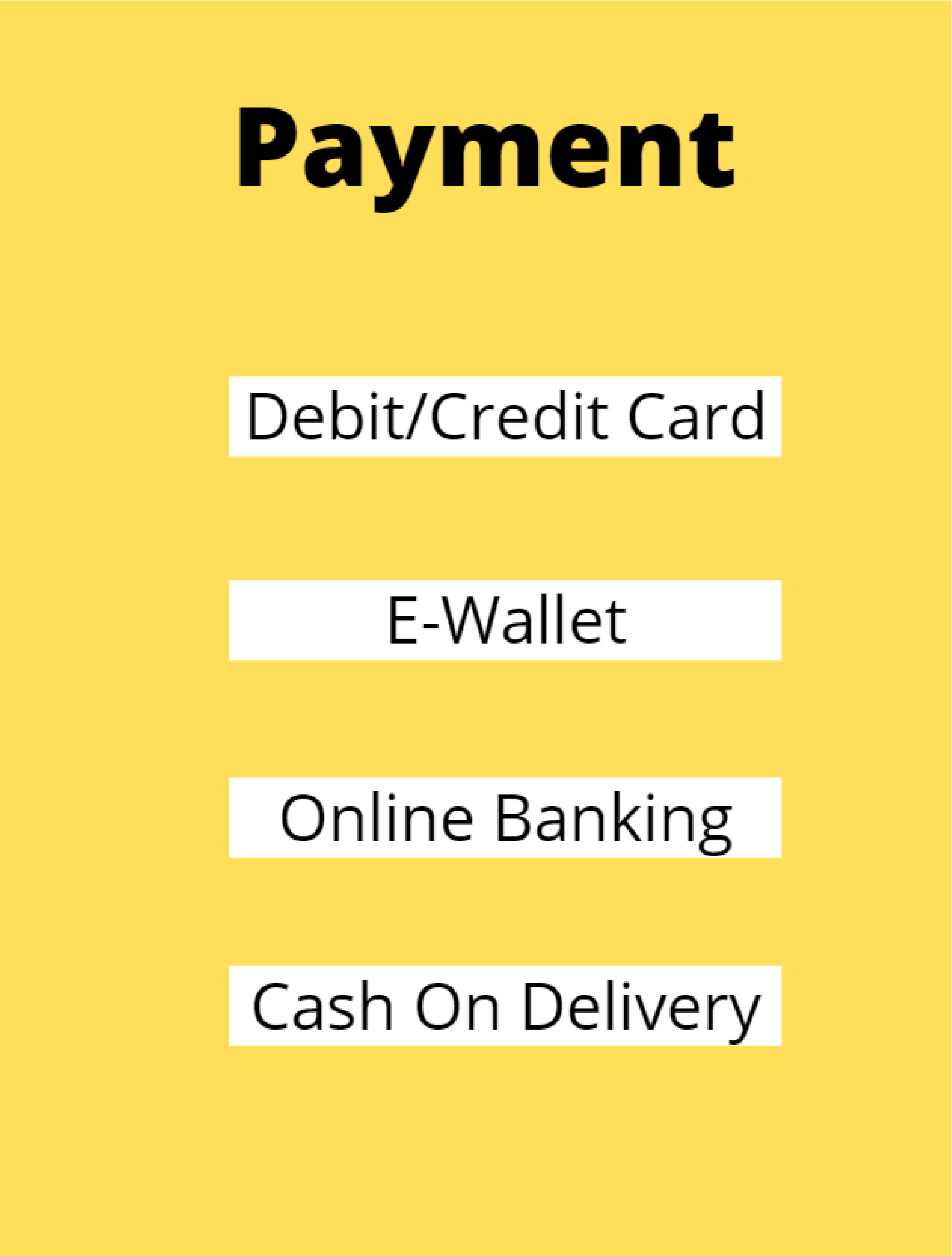
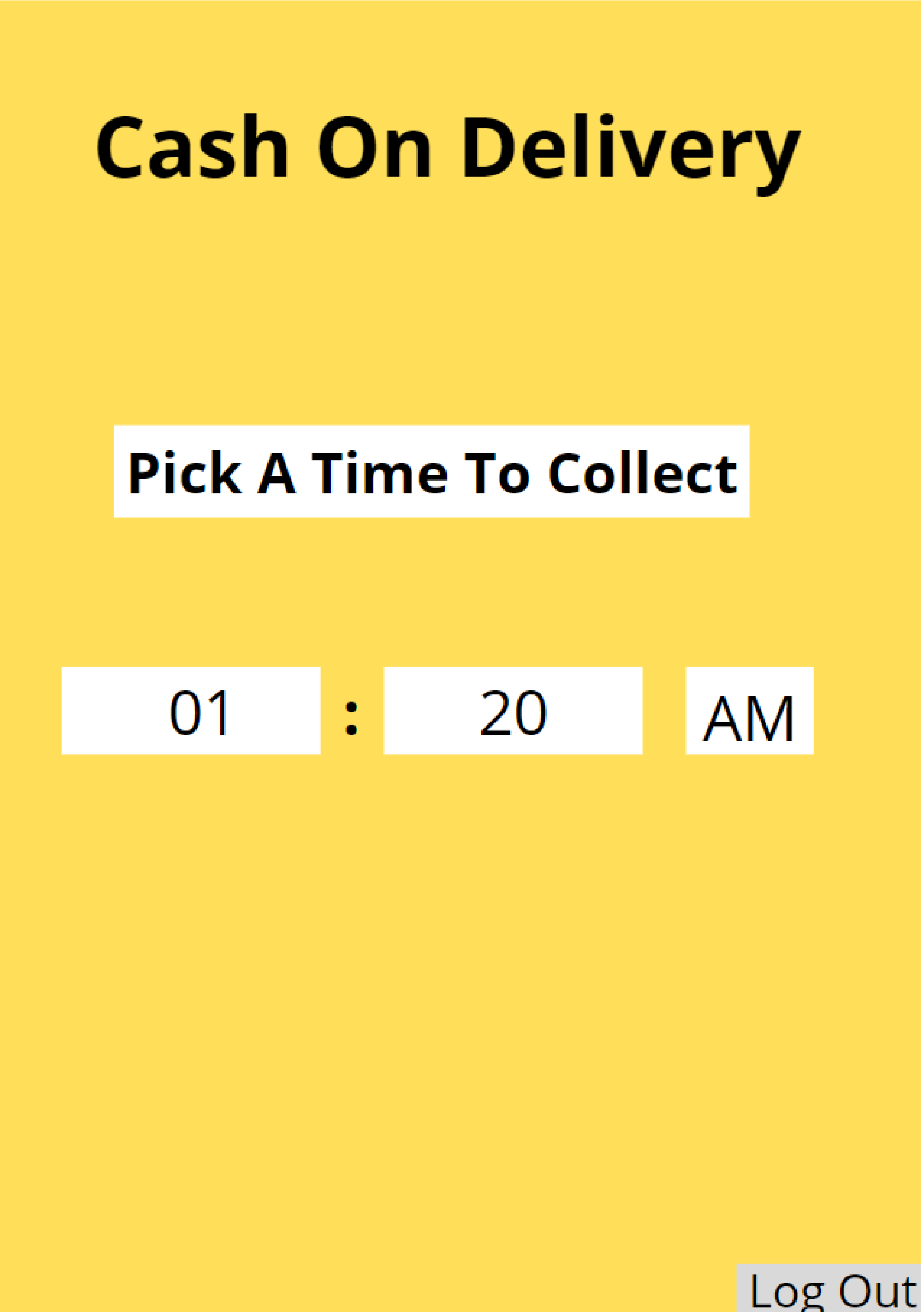


Figure 3.5 User interface of Tapau at Inti

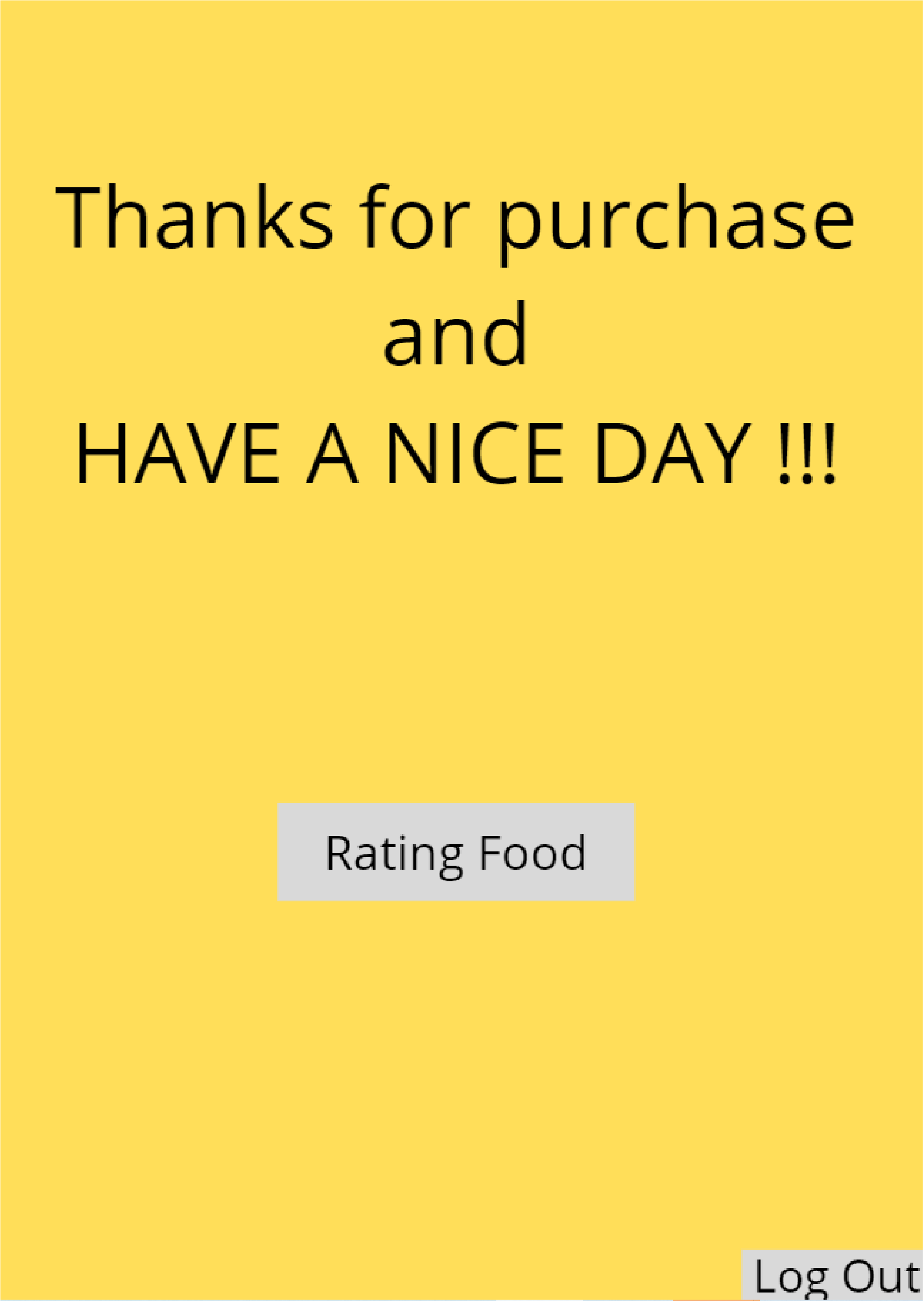


Cash On Delivery

Pick A Time To Collect

01 : 20 AM

Figure 3.5 User interface of Tapau at Inti



Thanks for purchase
and
HAVE A NICE DAY !!!

Figure 3.5 User interface of Tapau at Inti

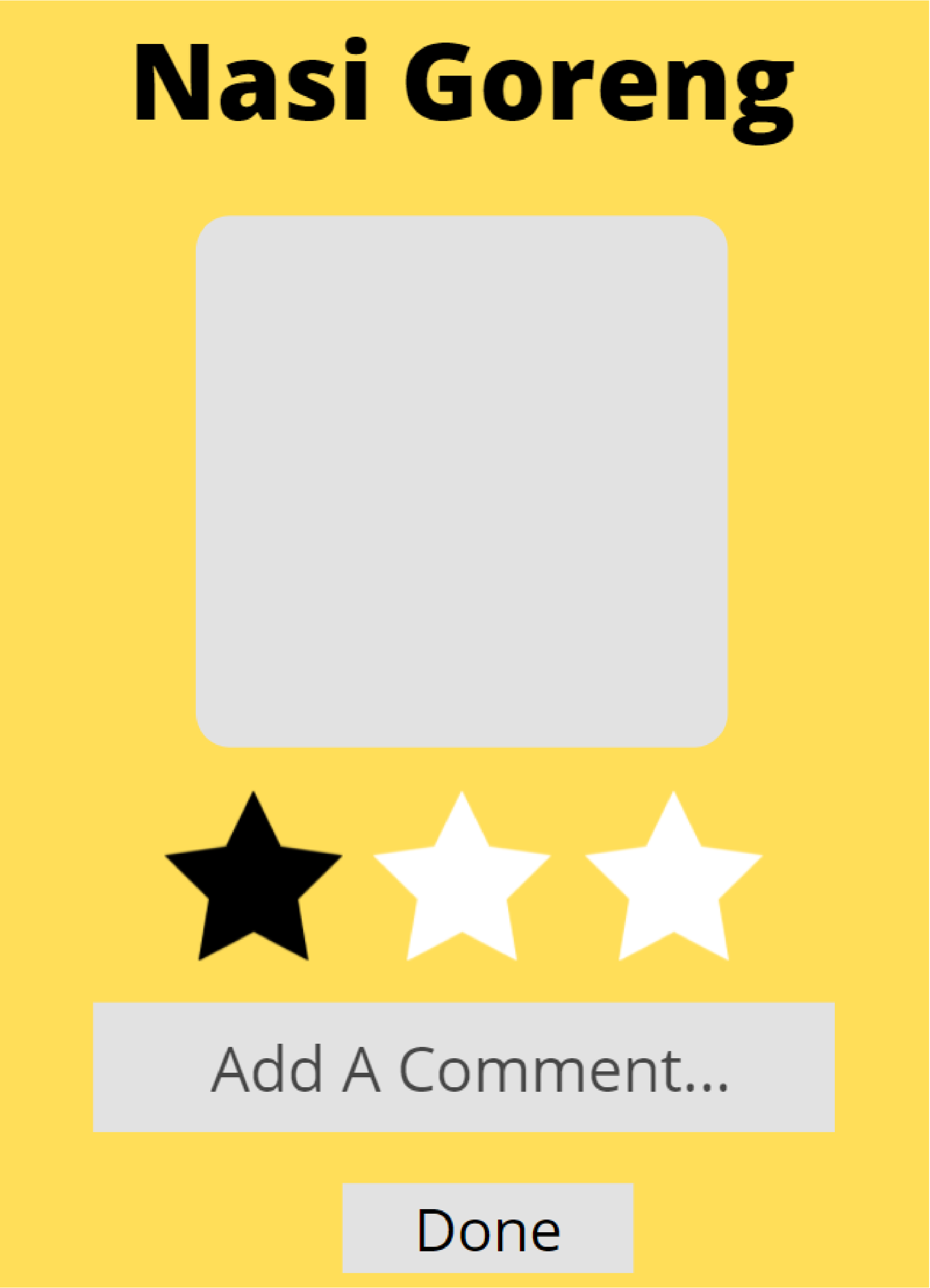


Figure 3.5 User interface of Tapau at Inti

3.2.2 (d2) Explanation of User Interface

The register page contains student id, password, reenter password and contact number. Admin page contain id and password. User can choose between dine in or take away. Food Menu contain vegetarian and non-vegetarian. User can choose the cuisines from food menu. After select the order from food menu, user can select the quantity of food for the food chosen then add to cart. User can choose the payment method from either Debit card, E-wallet, online banking or cash on delivery. Then User need to pick the time of the food to be arrived User can click the rating food to rate the food. Then, user can select the food from the order which he want to rate or add comment then click done.

3.2.3 Implementation

Coding takes place in this phase. Programmers take information from the previous stage and create a functional product. They typically implement code in small pieces, which are integrated at the end of this phase or the beginning of the next. (Lucid Content Team, n.d.). During the implementation phase, every time one page is created, the code is tested to check whether the code can run as expected. Firstly, Pip will be installed. After Pip is being installed. Db browser is downloaded to store the data. Visual studio code is downloaded to edit code and debugging. After that, the registration page and login page are created. For the registration page and login page, crispy form is used to make the form become neater. The Customer table is created by using models.py to store the data related to customer. Then the log-out function is created. Then the main menu which is the main menu.html is created. If user is admin, admin can sign up as admin then sign in as admin to view the customer order. Inside main menu, user can view the menu which are Japanese Food, Rice dishes, Beverage and Vegetarian. User can click the images to navigate to the page related (for example Japanese_food.html). Then inside the Japanese_food.html, user can further view the menu from the Japanese food that they like. After viewing the menu, user can click order now to navigate to order.html to fill in their personal detail and select the order and state the quantity they want. After that, user need to confirm order. After that, the user can go back to the index.html or make some comment inside comment.html. The sequential flow of the user interface will be illustrated in Figure 3.6.

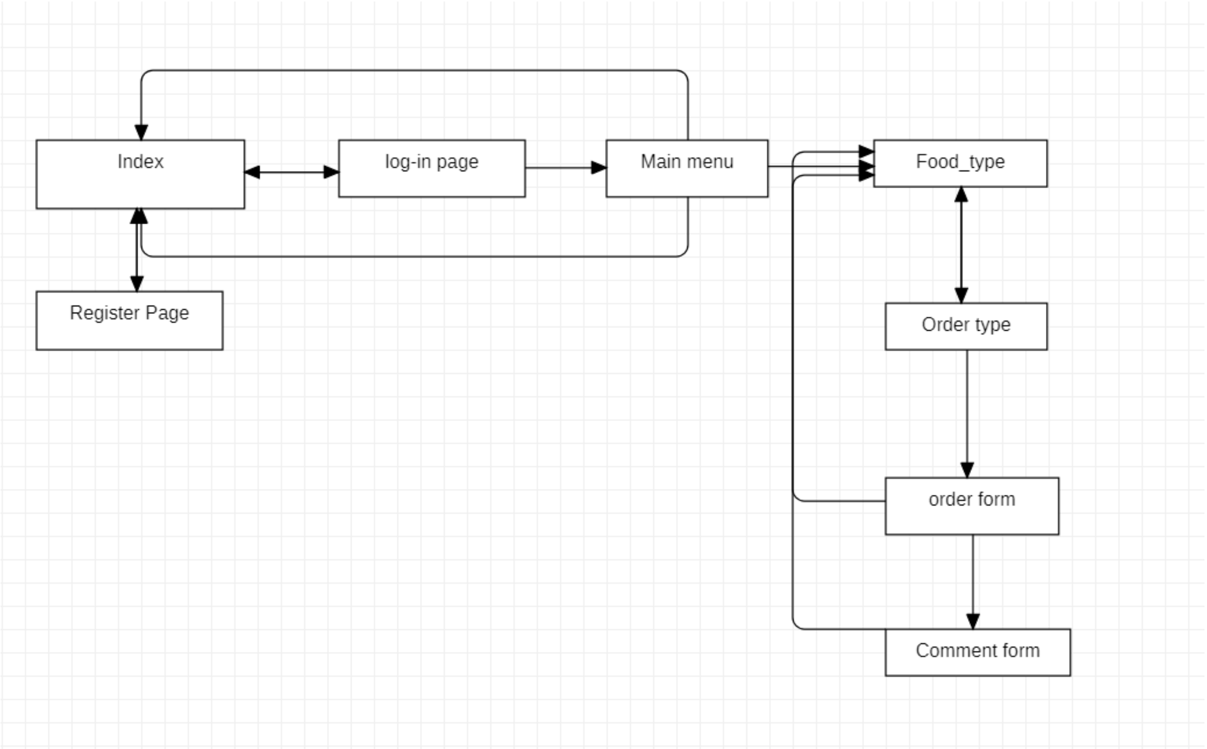


Figure 3.6 Sequential Flow of User Interface

3.2.4 Testing and Evaluation

Testing documentation includes the documentation of artifacts that should be produced before or during the testing of Software. Documentation for software testing helps to evaluate the testing effort required, test coverage, requirement tracking/tracing and others. The chapter 4 will explain one of the usually used documented artifacts that will be done in this Chapter which is the test case. (Tutorialspoint, n.d.)

A test case is a set of procedures performed on a system to decide if it fulfils software requirements and functions correctly. The purpose of a test case is to determine if different features within a system are performing as expected and to confirm that the system satisfies all related standards, guidelines and customer requirements. The process of writing a test case can also help reveal errors or defects within the system. (Brush, n.d.)

3.2.5 Deployment

The deployment phase involves making the software live in the production or real environment after it has been tested thoroughly in the previous phase. The Tapau at Inti will be distributed to the public if it is usable. (Rana,2020)

3.2.6 Maintenance

Over a period of time, a software product may require some updates in order to remain functional in the real-world environment. The maintenance phase takes care of this activity by timely tuning of the software as per the requirement. After the Tapau at Inti application is distributed to the public, improvement of the application will always be made to ensure good quality of the application. (Rana, 2020)

3.3 System Development Tools

In Database Design, we mention that we need to create a database connection. We choose sqlite3 as a tool to develop the database connection. Register page and login page will use this tool to design. We will use the superuser to store and retrieve the user information. The next thing we need is to connect the server to the internet. We will use Django and command prompt as a tool to connect our html file to the internet. To create the html file, visual studio code is needed as a template to create the html file can write the other python file which is created by using command prompt. Git is installed to share the code among team members through the Github.

3.4 Summary

In summary, the waterfall model is chosen because it is easy to understand and use and more suitable for a newbie programmer. Tapau at Inti will not always change. Tapau at Inti is not a new system that requires changes because there are already a lot of existing food ordering system. It is also not a long-term and complex project because a simple project is made. There are six phases of sequential flow of waterfall models which are requirement gathering and analysis, system design, implementation, testing, deployment and maintenance. In the gathering and analysis phase, Initial requirements will be composed, and it will contain the end result of the project, available features and features not initially supported. Besides, literature review will be done in this phase. Once the requirement is confirmed, it will be used to serve as a guideline for the next phase. In this chapter, system design has been discussed. The Design Phase is where you look at the many potential solutions and narrow down the choices to determine the most effective and efficient way to construct the solution. At the end of the design phase, the best solution will be found and illustrated using Er-diagram, Use case diagram, user interface design and flow chart. Development tools which are pip, visual studio code, StarUml, command prompt, Db Browser will be used to design the Tapau at Inti. After this design phase, implementation phase will be carried out by writing the code inside the visual studio code. Then test case will be carried out in testing phase. After that, deployment will be done to publish to the public if it is usable. Finally, maintenance will be made to ensure the good quality of this application.

CHAPTER 4

TESTING AND EVALUATION

4.1 Introduction

This chapter will discuss testing and evaluation in detail and separately. In the previous chapter, we discussed the design and implementation phases of writing the code. The testing phase will be carried out in this chapter to determine the error. Testing is the process of evaluating a system or its components to determine whether it fulfills the specific needs. Testing is executing a system to recognize any flaws, mistakes, or lacking requirements. When the testing phase begins, the testing team will follow the test cases or "scripts" they prepared to carry out the tests and check the Software. The order or group of test cases is called a test suite. (Barlett, 2016)

Testing documentation includes the artifacts that should be created before or during software testing. Documentation for software testing helps assess the testing effort required, testing scope, requirement tracking or tracing, and others. This section explains one of the usually used documented artifacts that will be done in this chapter which is the test case. (Tutorials point, n.d.)

As mentioned before in chapter 3.11, testing, a test case is a set of procedures performed on a system to decide if it fulfills software requirements and functions accurately.

Test cases define what must be done to test a system, including the steps executed in the system, the input data values entered into the system, and the expected results throughout test case execution. Using test cases allows developers and testers to locate errors that may have happened during development or missed defects during ad hoc tests. A practical test case's benefits include ensuring good test coverage, reduced maintenance and Software support costs, reusable test cases, and confirmation that the Software satisfies end-user requirements. (Brush, n.d.)

In this chapter, testing and evaluation will be discussed in detail and separately. In the previous chapter, we discussed the design and implementation phases of writing the code. The testing phase will be carried out in this chapter to determine the error. Testing is the process of evaluating a system or its components to find whether it satisfies the specified requirements or not. In simple words, testing is executing a system to recognize flaws, mistakes, or missing requirements. When the testing phase starts, the testing team follows the test cases or "scripts" they wrote in order to perform the tests and verify the Software. The sequence or group of test cases is called a test suite. (Barlett, 2016)

Once program activities in the logic model should be evaluated and determined, the types of evaluation that can be conducted would start to be identified. There are different types of evaluations that can be carried out. Some of them include formative evaluation, process evaluation, outcome evaluation and impact evaluation. Formative evaluation ensures that a program or program activity is achievable, relevant, and acceptable before it is entirely implemented. Evaluation is usually done when a new software is being created or when an existing one is being changed or modified. Process or implementation evaluation decides whether program activities have been performed as intended. Result or effectiveness evaluation measures program results in the target population by evaluating the progress in the outcomes or outcome objectives that the program is to achieve. Impact evaluation assesses program effectiveness in achieving its ultimate goals. (Types of evaluation, n.d.)

4.2 System Testing

This subchapter will include all the functionality testing in the application. The purpose is to System testing to check the performance of the entire system all together. System testing includes both functional and non-functional testing. The development of testing a combined system to check that it meets or matches requirements that are specified is known as **s**ystem testing. Basic Purpose of System Testing is to assure that a system or application satisfies its specification and any non-functional requirements (such as stability and throughput) also meets the requirements with its users. Second purpose is to include a different test series to fully exercise the computer-based system. Third purpose is to confirm an application or system accuracy and completeness in achieving the designed functions. In conclusion, System testing means testing of the entire software. The main purpose of system testing is to detect whether the software satisfies its requirements or not. (Null, n.d.)

4.3 Test Cases

The following tables shows the test cases and their output of the application

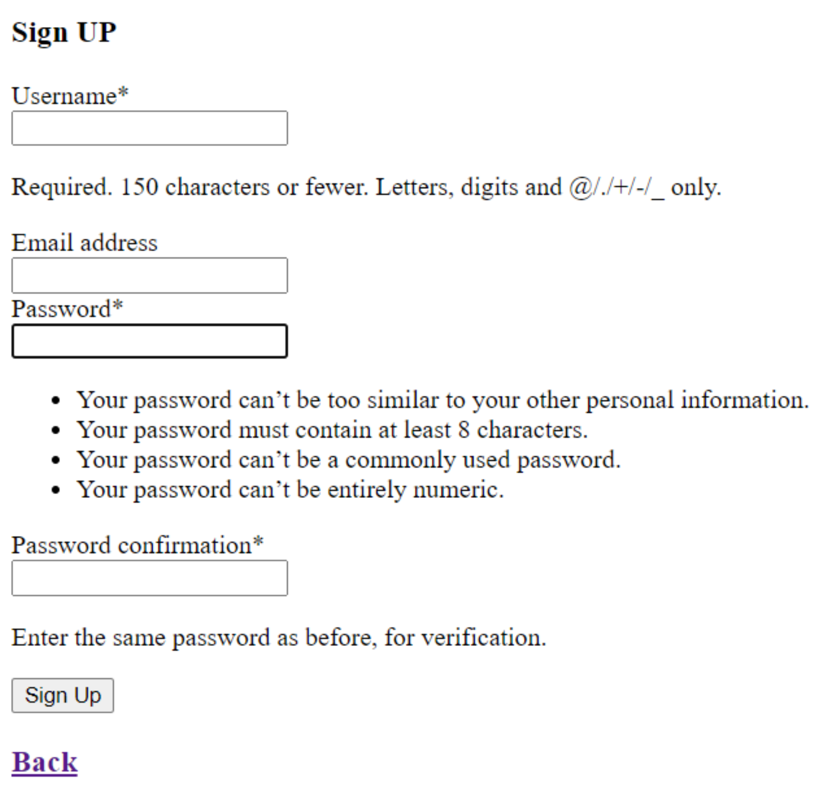
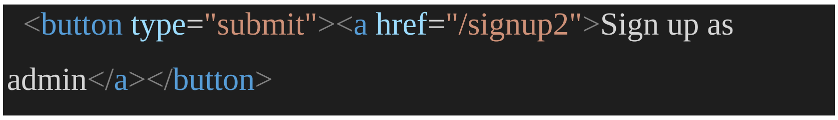
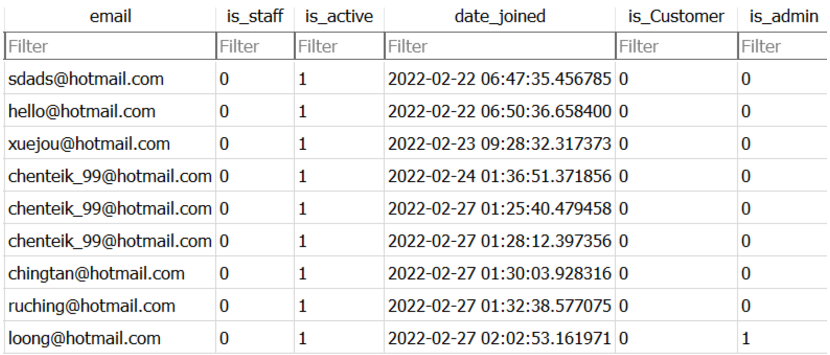
Test Case	1. Sign up as admin button
Objective	To test if the admin can navigate to signup.html successfully
Test Data	Navigation button
Expected Output	The admin will navigate to signup.html
Result	 **Sign UP** Username* Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only. Email address Password* • Your password can't be too similar to your other personal information. • Your password must contain at least 8 characters. • Your password can't be a commonly used password. • Your password can't be entirely numeric. Password confirmation* Enter the same password as before, for verification. Sign Up **Back**
Code	 `<button type="submit"><a href="/signup2">Sign up as admin</a></button>`
Conclusion	The admin navigates to signup.html successfully

Table 4.1 Test Case for Sign up as Admin Button

Test Case	2. Sign up button
Objective	To test if the admin can register as an admin successfully.
Test Data	Username, Email Address, Contact, Password, Password Confirmation, Sign Up button.
Expected Output	The admin can register as an admin and save inside database. The is_admin will represent as 1 inside the database if the admin is registered as an admin.
Result	

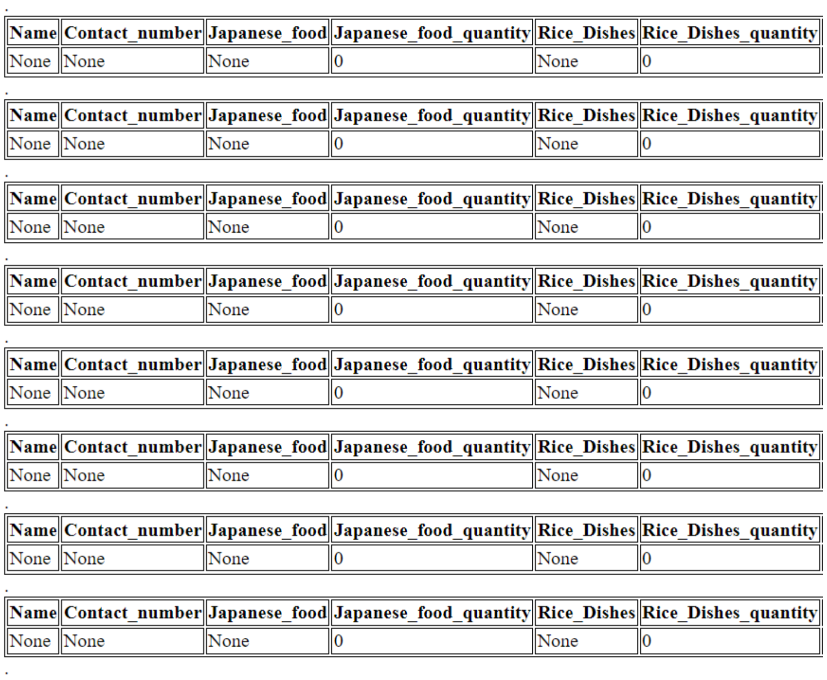
email	is_staff	is_active	date_joined	is_Customer	is_admin
Filter	Filter	Filter	Filter	Filter	Filter
sdads@hotmail.com	0	1	2022-02-22 06:47:35.456785	0	0
hello@hotmail.com	0	1	2022-02-22 06:50:36.658400	0	0
xuejou@hotmail.com	0	1	2022-02-23 09:28:32.317373	0	0
chenteik_99@hotmail.com	0	1	2022-02-24 01:36:51.371856	0	0
chenteik_99@hotmail.com	0	1	2022-02-27 01:25:40.479458	0	0
chenteik_99@hotmail.com	0	1	2022-02-27 01:28:12.397356	0	0
chingtan@hotmail.com	0	1	2022-02-27 01:30:03.928316	0	0
ruching@hotmail.com	0	1	2022-02-27 01:32:38.577075	0	0
loong@hotmail.com	0	1	2022-02-27 02:02:53.161971	0	1

Code

```
class AdminStudentRegisterView(CreateView):
    model = User
    form_class = AdminStudentRegistrationForm
    template_name = 'signup2.html'

    def form_valid(self, form):
        user = form.save()
        login(self.request, user)
        # return redirect('learner')
        return redirect('/signin')
```

Conclusion	The admin can register as an admin successfully.

Table 4.2 Test Case for Sign up button

<table>
<tr><td>Test Case</td><td>3. Sign In button</td></tr>
<tr><td>Objective</td><td>To test if the admin can sign in as an admin successfully and view the order.</td></tr>
<tr><td>Test Data</td><td>Username, Password, Sign In button.</td></tr>
<tr><td>Expected Output</td><td>The admin can sign as an admin and navigate to the showresult.html and view the order.</td></tr>
<tr><td>Result</td><td>

<table>
<tr><th>Name</th><th>Contact_number</th><th>Japanese_food</th><th>Japanese_food_quantity</th><th>Rice_Dishes</th><th>Rice_Dishes_quantity</th></tr>
<tr><td>None</td><td>None</td><td>None</td><td>0</td><td>None</td><td>0</td></tr>
</table>
<table>
<tr><th>Name</th><th>Contact_number</th><th>Japanese_food</th><th>Japanese_food_quantity</th><th>Rice_Dishes</th><th>Rice_Dishes_quantity</th></tr>
<tr><td>None</td><td>None</td><td>None</td><td>0</td><td>None</td><td>0</td></tr>
</table>
<table>
<tr><th>Name</th><th>Contact_number</th><th>Japanese_food</th><th>Japanese_food_quantity</th><th>Rice_Dishes</th><th>Rice_Dishes_quantity</th></tr>
<tr><td>None</td><td>None</td><td>None</td><td>0</td><td>None</td><td>0</td></tr>
</table>
<table>
<tr><th>Name</th><th>Contact_number</th><th>Japanese_food</th><th>Japanese_food_quantity</th><th>Rice_Dishes</th><th>Rice_Dishes_quantity</th></tr>
<tr><td>None</td><td>None</td><td>None</td><td>0</td><td>None</td><td>0</td></tr>
</table>
<table>
<tr><th>Name</th><th>Contact_number</th><th>Japanese_food</th><th>Japanese_food_quantity</th><th>Rice_Dishes</th><th>Rice_Dishes_quantity</th></tr>
<tr><td>None</td><td>None</td><td>None</td><td>0</td><td>None</td><td>0</td></tr>
</table>
<table>
<tr><th>Name</th><th>Contact_number</th><th>Japanese_food</th><th>Japanese_food_quantity</th><th>Rice_Dishes</th><th>Rice_Dishes_quantity</th></tr>
<tr><td>None</td><td>None</td><td>None</td><td>0</td><td>None</td><td>0</td></tr>
</table>
<table>
<tr><th>Name</th><th>Contact_number</th><th>Japanese_food</th><th>Japanese_food_quantity</th><th>Rice_Dishes</th><th>Rice_Dishes_quantity</th></tr>
<tr><td>None</td><td>None</td><td>None</td><td>0</td><td>None</td><td>0</td></tr>
</table>
<table>
<tr><th>Name</th><th>Contact_number</th><th>Japanese_food</th><th>Japanese_food_quantity</th><th>Rice_Dishes</th><th>Rice_Dishes_quantity</th></tr>
<tr><td>None</td><td>None</td><td>None</td><td>0</td><td>None</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>Code</td><td>
<pre>
def signin(request):
    if request.method=="POST":
        username=request.POST["username"]
        pass1=request.POST["pass1"]
        user=authenticate(username=username, password=pass1)
        if user is not None:
            auth.login(request, user)
            if user.is_admin or user.is_superuser:
                return redirect('/showresult')
            else:
</pre>
</td></tr>
</table>

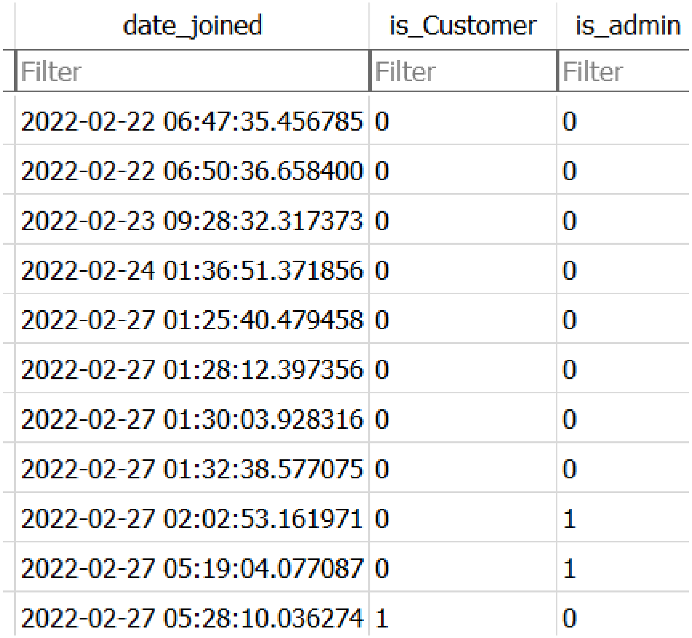
	`return redirect('/main_menu')`
	`else:`
	`messages.error(request,"Bad Credential!")`
	`return render(request, "signin.html")`
	`else:`
	`return render(request, "signin.html")`

Conclusion	The admin can sign in as an admin successfully

Table 4.3 Test Case for Sign In button

Test Case	4. Sign Up button
Objective	To test if the user can sign up successfully
Test Data	Username, Email, Contact, Password, Password confirmation, Sign Up button.
Expected Output	The user can register as a user and save inside database. The is_Customer will represent as 1 inside the database if the customer is registered as a user.

<table>
<tr>
<td>Result</td>
<td>



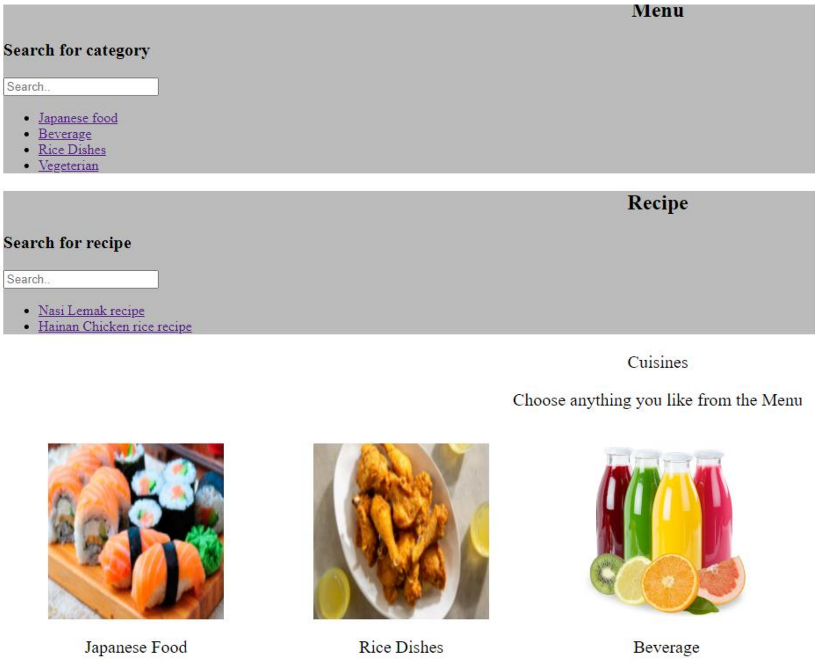
date_joined	is_Customer	is_admin
Filter	Filter	Filter
2022-02-22 06:47:35.456785	0	0
2022-02-22 06:50:36.658400	0	0
2022-02-23 09:28:32.317373	0	0
2022-02-24 01:36:51.371856	0	0
2022-02-27 01:25:40.479458	0	0
2022-02-27 01:28:12.397356	0	0
2022-02-27 01:30:03.928316	0	0
2022-02-27 01:32:38.577075	0	0
2022-02-27 02:02:53.161971	0	1
2022-02-27 05:19:04.077087	0	1
2022-02-27 05:28:10.036274	1	0

</td>
</tr>
<tr>
<td>Code</td>
<td>

```
def signin(request):
    if request.method=="POST":
        username=request.POST["username"]
        pass1=request.POST["pass1"]
        user=authenticate(username=username, password=pass1)
        if user is not None:
            auth.login(request, user)
            if user.is_admin or user.is_superuser:
                return redirect('/showresult')
            else:
                return redirect('/main_menu')
        else:
            messages.error(request,"Bad Credential!")
            return render(request, "signin.html")
    else:
        return render(request, "signin.html")
```

</td>
</tr>
<tr>
<td>Conclusion</td>
<td>The user can sign up as a user successfully</td>
</tr>
</table>

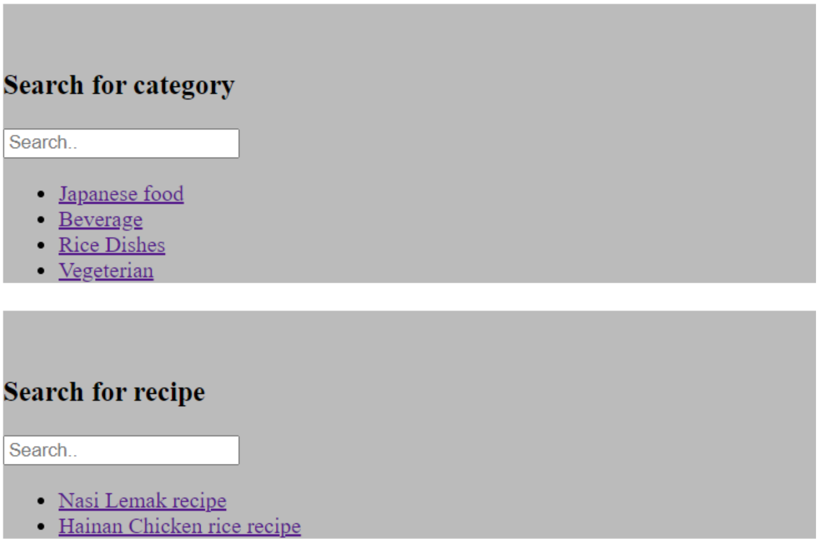
Table 4.4 Test Case for Sign up button

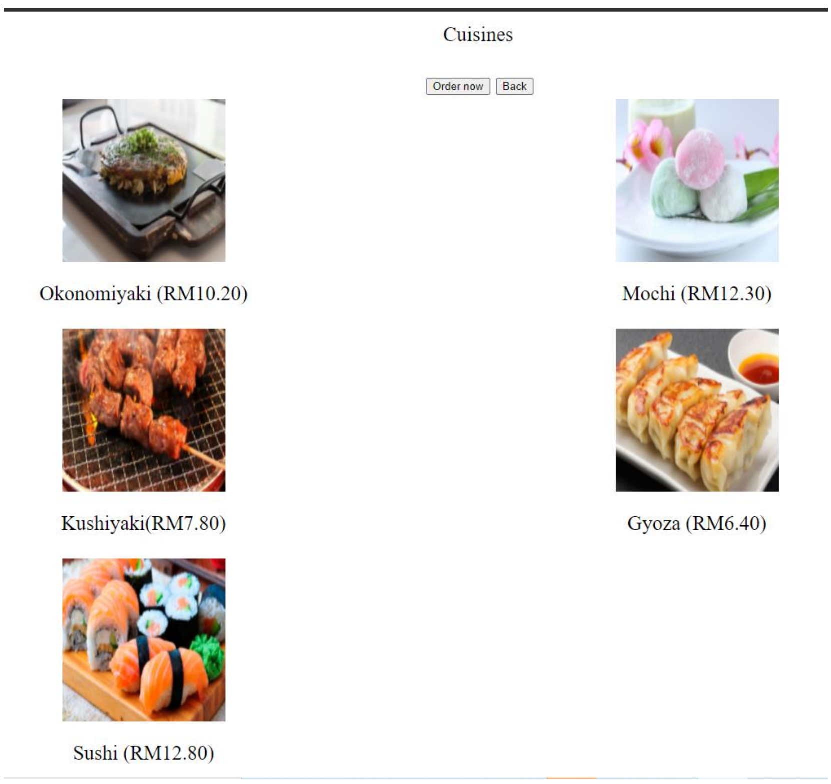
Test Case	5. Sign In button
Objective	To test if the user can sign in successfully
Test Data	Username, Password.
Expected Output	The user can sign in as a user and navigate to main_menu.html
Result	 Menu Search for category Search.. - Japanese food - Beverage - Rice Dishes - Vegeterian Recipe Search for recipe Search.. - Nasi Lemak recipe - Hainan Chicken rice recipe Cuisines Choose anything you like from the Menu Japanese Food Rice Dishes Beverage

ode	def signin(request):
	    if request.method=="POST":
	        username=request.POST["username"]
	        pass1=request.POST["pass1"]
	        user=authenticate(username=username, password=pass1)
	        if user is not None:
	            auth.login(request, user)
	            if user.is_admin or user.is_superuser:
	                return redirect('/showresult')
	            else:

<table>
<tr><td></td><td><pre>   return redirect('/main_menu')
    else:
        messages.error(request,"Bad Credential!")
        return render(request, "signin.html")
else:
    return render(request, "signin.html")</pre></td></tr>
<tr><td>Conclusion</td><td>The user can sign in as a user successfully</td></tr>
</table>

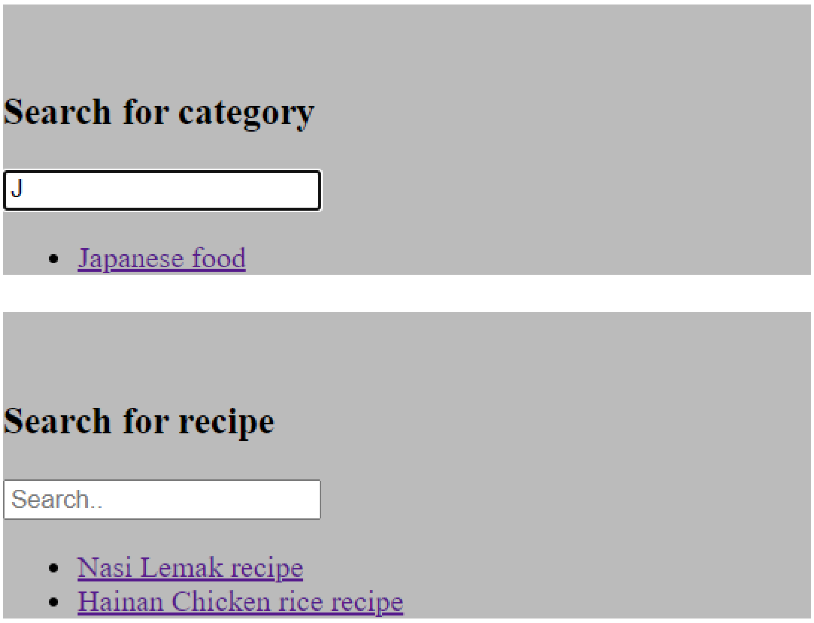
Table 4.5 Test Case for Sign In button

Test Case	6. Search function
Objective	To test if the user can search for category successfully
Test Data	Japanese food, Beverage, Rice Dishes, Vegetarian
Expected Output	The user can search for category successfully and navigate to the related page
Result	 **Search for category** Search.. • Japanese food • Beverage • Rice Dishes • Vegeterian **Search for recipe** Search.. • Nasi Lemak recipe • Hainan Chicken rice recipe

<table>
<tr><td></td><td>
</td></tr>
<tr><td>Code</td><td><pre>
 <h3>Search for category</h3>
   <input type="text" id="mySearch" onkeyup="myFunction()"
placeholder="Search.." title="Type in a category">
   <ul id="myMenu">
     <li><a href="Japanese_food">Japanese food</a></li>
     <li><a href="Beverage/">Beverage</a></li>
     <li><a href="Rice/">Rice Dishes</a></li>
     <li><a href="Vegetarian/">Vegeterian</a></li>

   </ul>
 </div>
</pre></td></tr>
<tr><td>Conclusion</td><td>The user can search successfully. The user can navigate to Japanese_food page after clicking the Japanese_food.</td></tr>
</table>

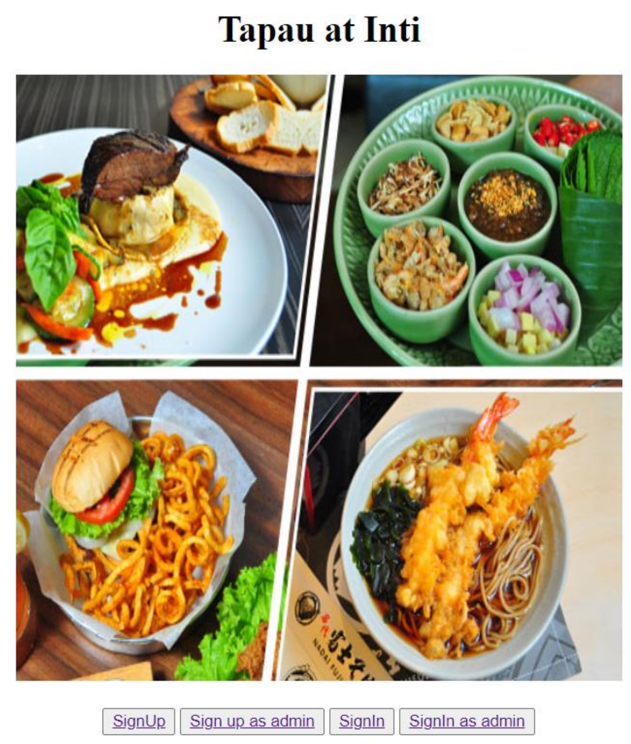
Table 4.6 Test Case for Search function

<table>
<tr><td>Test Case</td><td>7. Sort Function</td></tr>
<tr><td>Objective</td><td>To test if the user can sort the keyword of search for category successfully</td></tr>
<tr><td>Test Data</td><td>Japanese food, Beverage, Rice Dishes, Vegetarian</td></tr>
<tr><td>Expected Output</td><td>The user can sort the keyword</td></tr>
<tr><td>Result</td><td>
Search for category
J
- Japanese food
Search for recipe
Search..
- Nasi Lemak recipe
- Hainan Chicken rice recipe</td></tr>
<tr><td>Code</td><td><pre>
<script>
function myFunction() {
  var input, filter, ul, li, a, i;
  input = document.getElementById("mySearch");
  filter = input.value.toUpperCase();
  ul = document.getElementById("myMenu");
  li = ul.getElementsByTagName("li");
  for (i = 0; i < li.length; i++) {
    a = li[i].getElementsByTagName("a")[0];
    if (a.innerHTML.toUpperCase().indexOf(filter) > -1) {
      li[i].style.display = "";
    } else {
      li[i].style.display = "none";
</pre></td></tr>
</table>

	`    }`
	`  }`
	`}`

Conclusion	The user can sort the keyword successfully

Table 4.7 Test Case for Sort Function

Test Case	8. Log-out button
Objective	To test if the user can log-out successfully
Test Data	Navigation button
Expected Output	The user can log-out
Result	 Tapau at Inti

Code	`<a href="index/"> <h4>log-out</h4></a>`
Conclusion	The user can log-out successfully

Table 4.8 Test Case for log-out button

Test Case	9. Navigation Function
Objective	To test if the user can navigate successfully by clicking picture
Test Data	Navigation function
Expected Output	The user can navigate by clicking the picture

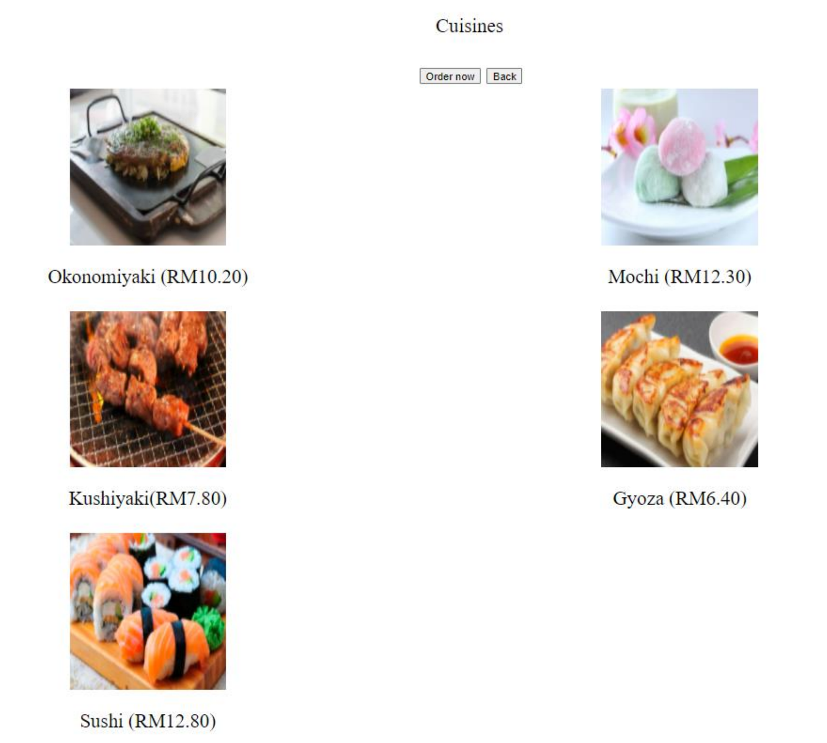
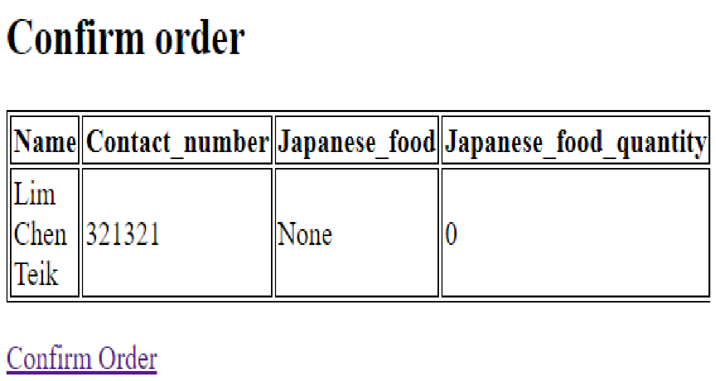
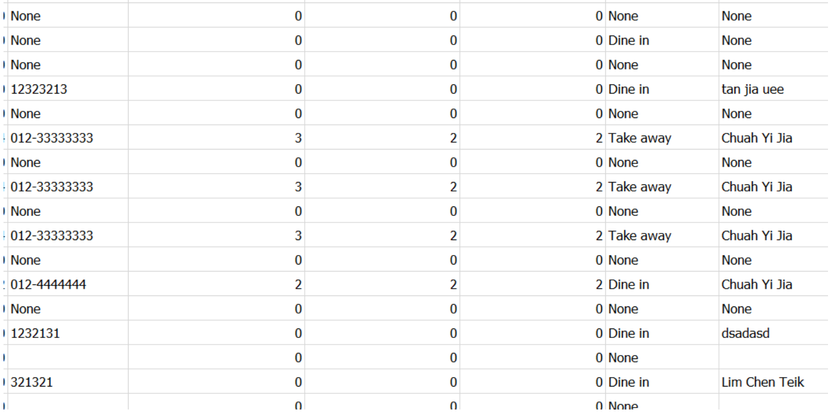
Result	 Cuisines Order now Back Okonomiyaki (RM10.20) Mochi (RM12.30) Kushiyaki(RM7.80) Gyoza (RM6.40) Sushi (RM12.80)
Code	`<a href="Japanese_food"><img src="https://cdn.tasteatlas.com//Images/Dishes/fde953e5aa504e8c83a84691582b0399.jpg?w=905&h=510 " width="200", height="200"></a>`
Conclusion	The user can navigate successfully by cling the picture.

Table 4.9 Test Case for Navigation Function

Test Case	10. Order now button
Objective	To test if the user can navigate to the showorder.html and order

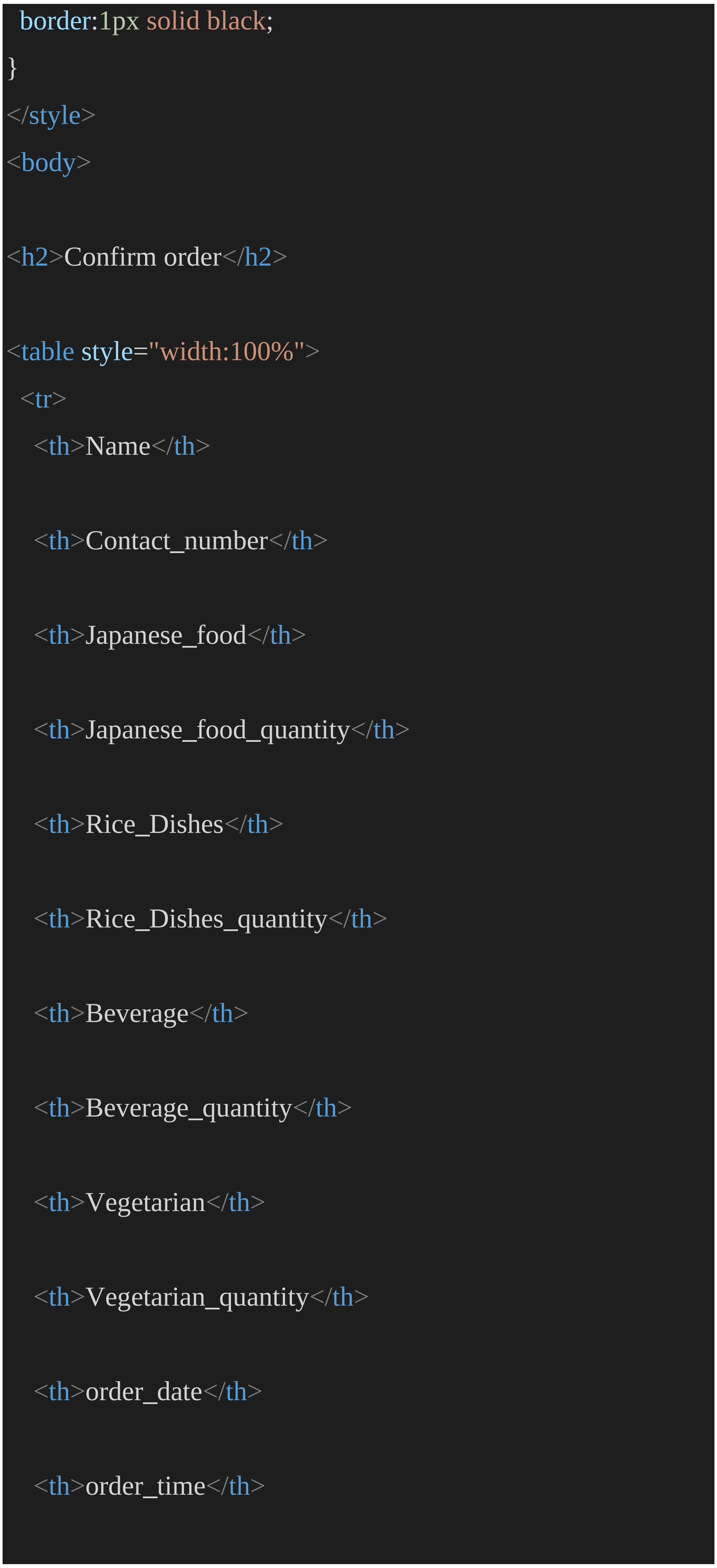
	record is saved in database.
Test Data	Name, Contact number, Japanese food, Japanese food quantity, Beverage, Beverage quantity, Rice Dishes, Rice Dishes quantity, Vegetarian, vegetarian quantity, Order date, order time, payment method and taking method.
Expected Output	The user can navigate by to showorder.html and save in database.
Result	 
Code	{% load crispy_forms_tags %} <!DOCTYPE html> <html> <style> table, th, td {

Confirm order

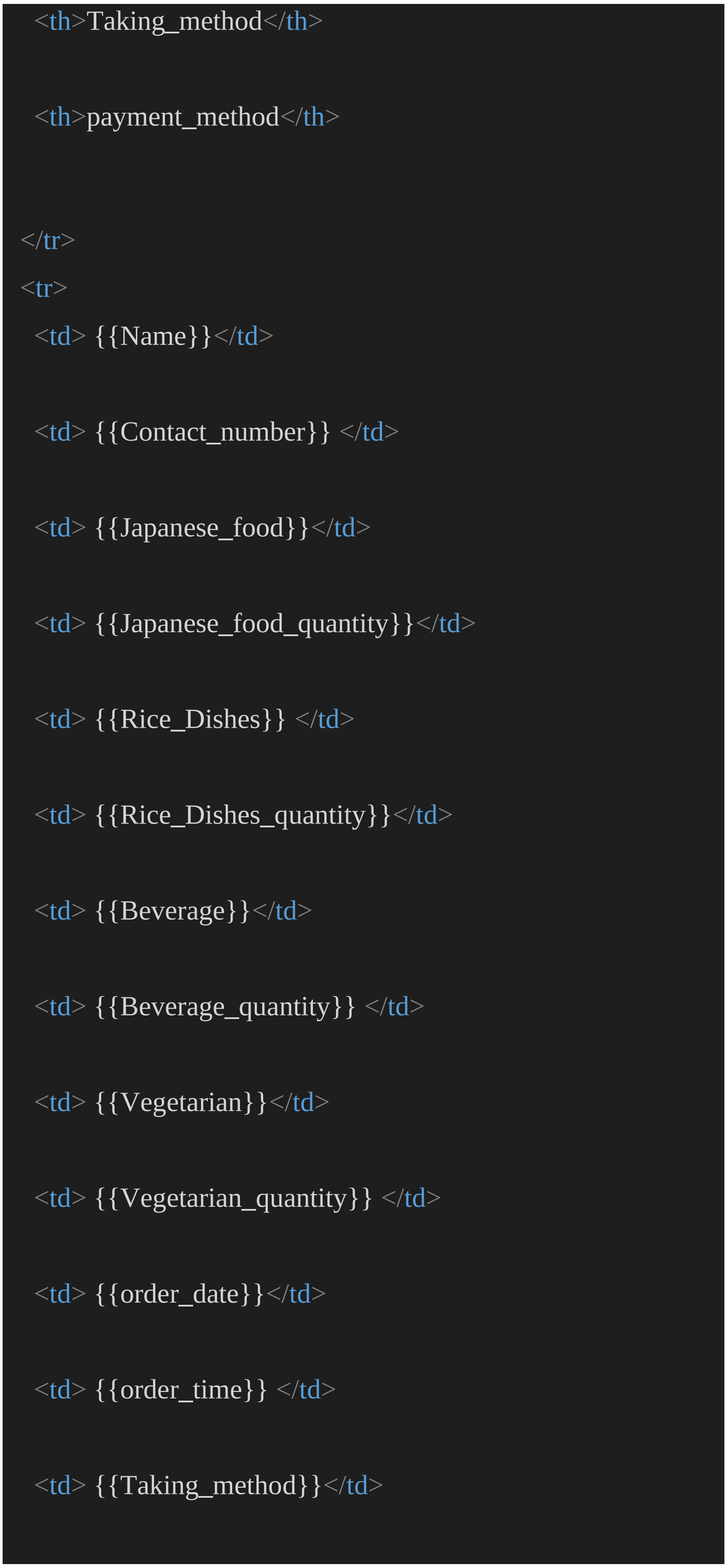
Name	Contact_number	Japanese_food	Japanese_food_quantity
Lim Chen Teik	321321	None	0

Confirm Order

None	0	0	0	None	None
None	0	0	0	Dine in	None
None	0	0	0	None	None
12323213	0	0	0	Dine in	tan jia uee
None	0	0	0	None	None
012-33333333	3	2	2	Take away	Chuah Yi Jia
None	0	0	0	None	None
012-33333333	3	2	2	Take away	Chuah Yi Jia
None	0	0	0	None	None
012-33333333	3	2	2	Take away	Chuah Yi Jia
None	0	0	0	None	None
012-4444444	2	2	2	Dine in	Chuah Yi Jia
None	0	0	0	None	None
1232131	0	0	0	Dine in	dsadasd
	0	0	0	None	
321321	0	0	0	Dine in	Lim Chen Teik
	0	0	0	None	



```
  border:1px solid black;
}
</style>
<body>

<h2>Confirm order</h2>

<table style="width:100%">
  <tr>
    <th>Name</th>

    <th>Contact_number</th>

    <th>Japanese_food</th>

    <th>Japanese_food_quantity</th>

    <th>Rice_Dishes</th>

    <th>Rice_Dishes_quantity</th>

    <th>Beverage</th>

    <th>Beverage_quantity</th>

    <th>Vegetarian</th>

    <th>Vegetarian_quantity</th>

    <th>order_date</th>

    <th>order_time</th>

```



```
  <th>Taking_method</th>

  <th>payment_method</th>


</tr>
<tr>
  <td> {{Name}}</td>

  <td> {{Contact_number}} </td>

  <td> {{Japanese_food}}</td>

  <td> {{Japanese_food_quantity}}</td>

  <td> {{Rice_Dishes}} </td>

  <td> {{Rice_Dishes_quantity}}</td>

  <td> {{Beverage}}</td>

  <td> {{Beverage_quantity}} </td>

  <td> {{Vegetarian}}</td>

  <td> {{Vegetarian_quantity}} </td>

  <td> {{order_date}}</td>

  <td> {{order_time}} </td>

  <td> {{Taking_method}}</td>
```

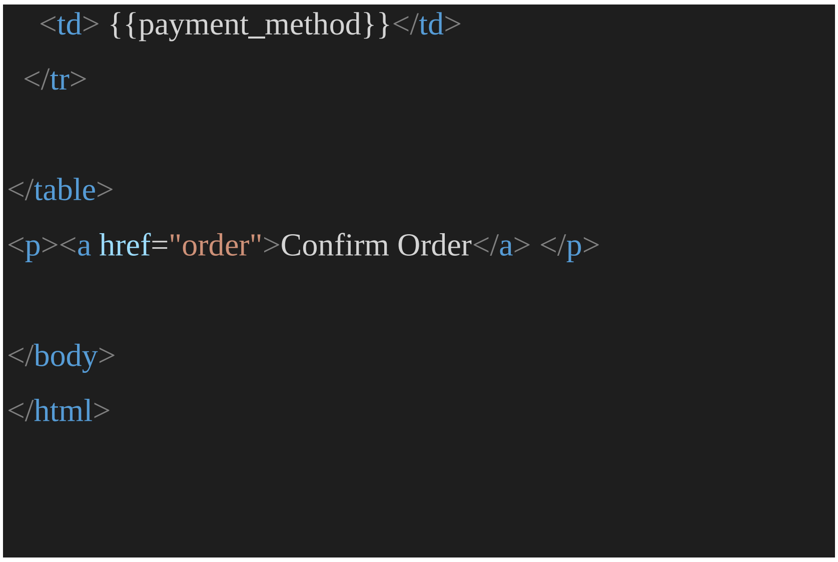
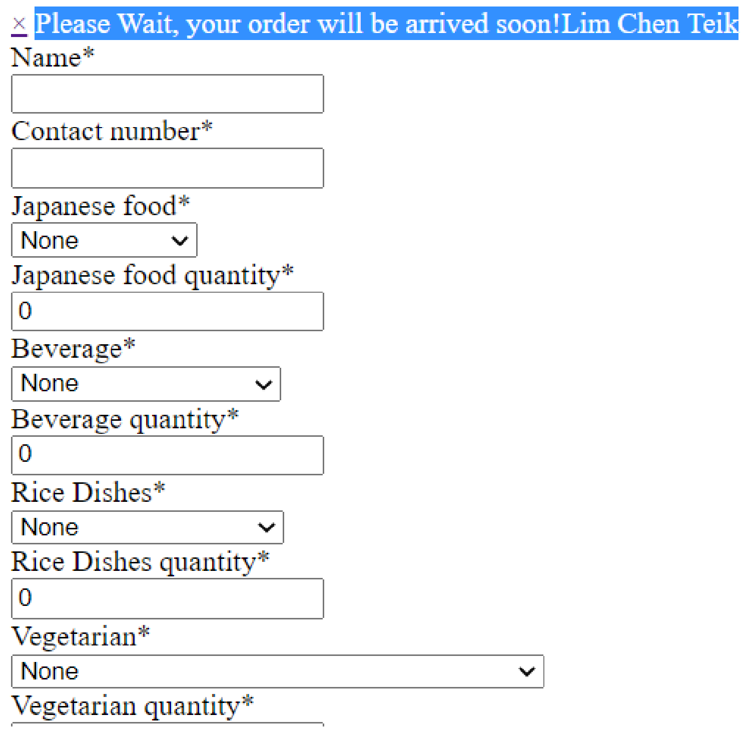
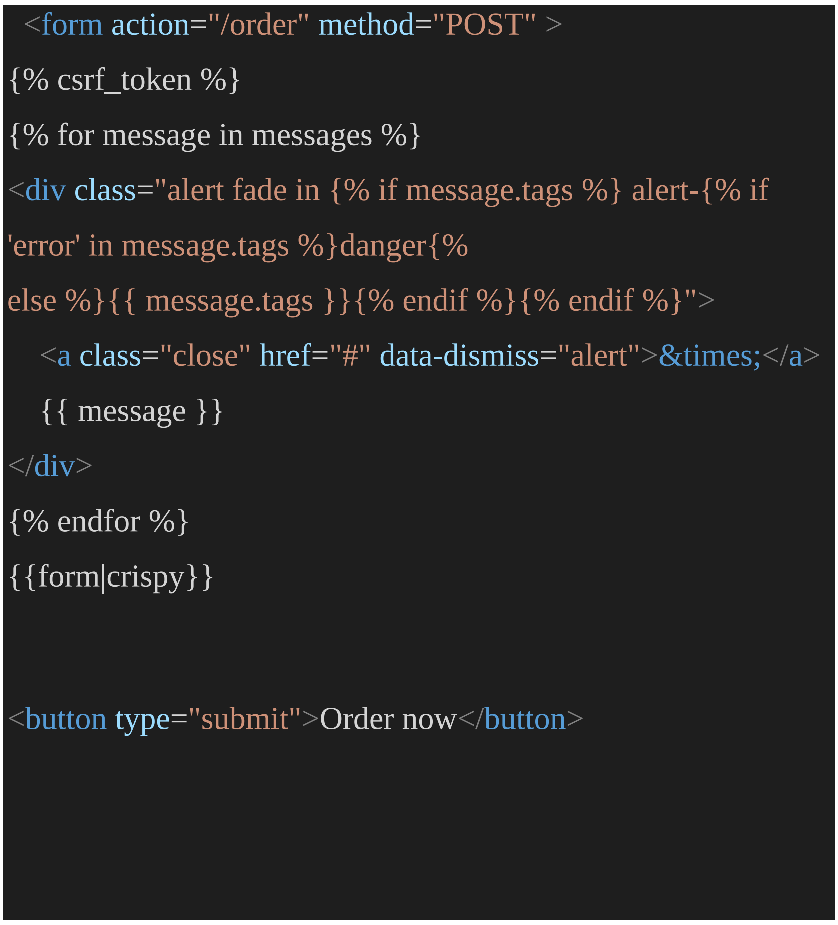
	 <td> {{payment_method}}</td> </tr> </table> <p><a href="order">Confirm Order</a> </p> </body> </html>
Conclusion	The user can navigate to showorder.html successfully and record is saved inside the database

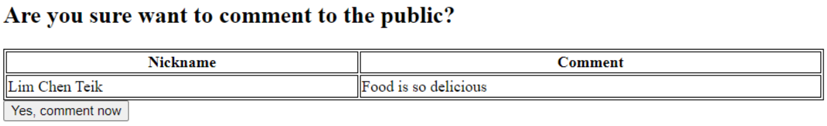
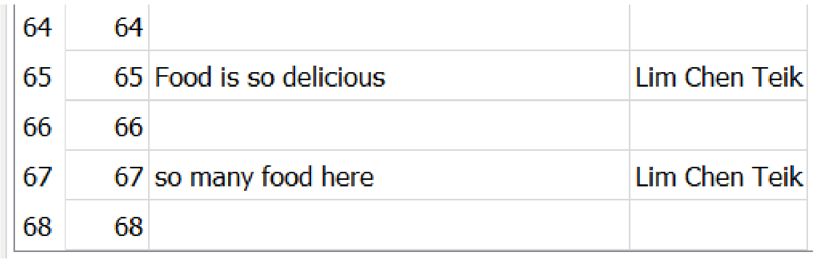
Table 4.10 Test Case for order now button

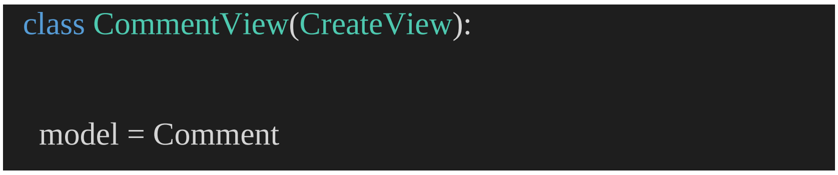
Test Case	11. Confirm Order button
Objective	To test if the messages will be showed when the Confirm Order button is pressed
Test Data	Confirm Order button
Expected Output	Messages will show “Please Wait, your order will be arrived soon! Name”

Result	 × Please Wait, your order will be arrived soon!Lim Chen Teik Name* Contact number* Japanese food* None Japanese food quantity* 0 Beverage* None Beverage quantity* 0 Rice Dishes* None Rice Dishes quantity* 0 Vegetarian* None Vegetarian quantity*
Code	 <form action="/order" method="POST" > {% csrf_token %} {% for message in messages %} <div class="alert fade in {% if message.tags %} alert-{% if 'error' in message.tags %}danger{% else %}{{ message.tags }}{% endif %}{% endif %}"> <a class="close" href="#" data-dismiss="alert">&times;</a> {{ message }} </div> {% endfor %} {{form|crispy}} <button type="submit">Order now</button>

Conclusion	Messages show “Please Wait, your order will be arrived soon! Name” successfully.

Table 4.11 Test Case for Confirm Order Button

Test Case	12. Comment to public button
Objective	To test if the nickname and comment of user is saved and displayed in table.
Test Data	Nickname, Comment
Expected Output	the nickname and comment of user is saved and displayed in table.
Result	 **Are you sure want to comment to the public?** Nickname | Comment Lim Chen Teik | Food is so delicious  64 64 65 65 Food is so delicious Lim Chen Teik 66 66 67 67 so many food here Lim Chen Teik 68 68

Code	
	class CommentView(CreateView):
	
	model = Comment

<table>
<tr><td></td><td><pre>
    form_class = CommentForm
    template_name = 'comment.html'

    def form_valid(self, form):
        comment = form.save()
        # return redirect('learner')
        Nickname=comment.Nickname
        Comment=comment.Comment
        messages.success(self.request, '%s, your comment is posted '%comment.Nickname)
        return render(self.request, "showcomment.html",{'Nickname':Nickname, "Comment":Comment})
</pre></td></tr>
<tr><td>Conclusion</td><td>The nickname and comment of user is saved and displayed in table.</td></tr>
</table>

Table 4.12 Test Case for Comment to Public Button

Test Case	13. “Yes, comment now” button
Objective	To test if the messages will be showed after click the “Yes, comment now” button.
Test Data	Yes, comment now button
Expected Output	Message which is “user nickname, your comment is posted” will be showed.

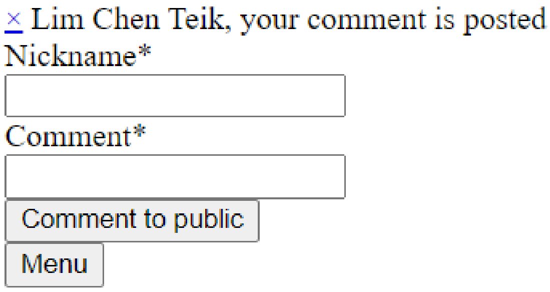
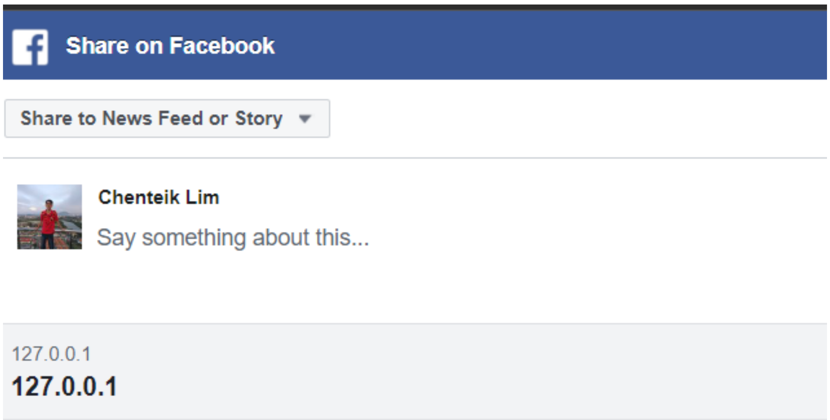
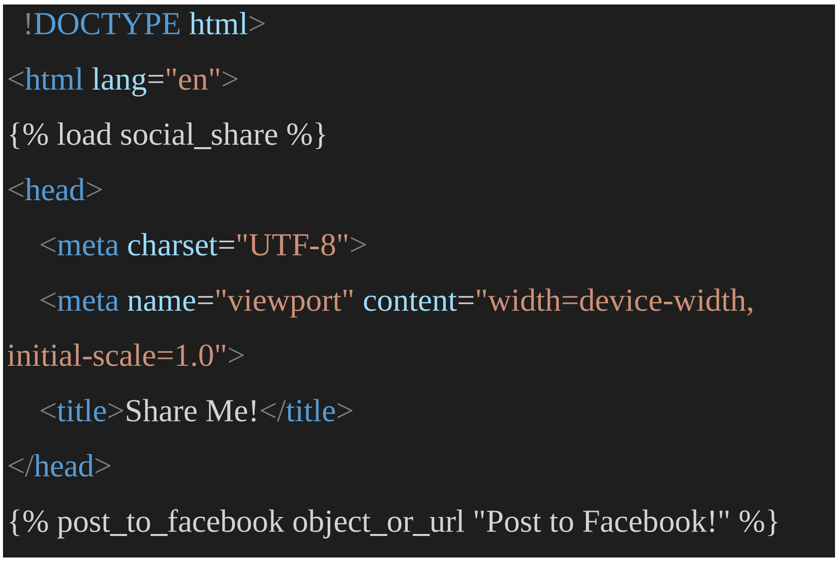
<table>
<tr><td>Result</td><td>
× Lim Chen Teik, your comment is posted
Nickname*
Comment*
 </td></tr>
<tr><td>Code</td><td><pre>
 def form_valid(self, form):
        comment = form.save()
        # return redirect('learner')
        Nickname=comment.Nickname
        Comment=comment.Comment
        messages.success(self.request, '%s, your comment is
posted '%comment.Nickname)
        return render(self.request,
"showcomment.html",{'Nickname':Nickname,
"Comment":Comment})
</pre></td></tr>
<tr><td>Conclusion</td><td>Message which is “user nickname, your comment is posted” will be showed.</td></tr>
</table>

Table 4.13 Test Case for “Yes, comment now” Button

Test Case	14. Post to Facebook button
Objective	To test if the user will come to Facebook page when click this button
Test Data	Post to Facebook button
Expected Output	User will navigate to Facebook pages.
Result	 Share on Facebook Share to News Feed or Story ▾ Chenteik Lim Say something about this... 127.0.0.1 127.0.0.1

Code	
	!DOCTYPE html>
	<html lang="en">
	{% load social_share %}
	<head>
	    <meta charset="UTF-8">
	    <meta name="viewport" content="width=device-width, initial-scale=1.0">
	    <title>Share Me!</title>
	</head>
	{% post_to_facebook object_or_url "Post to Facebook!" %}

	{% post_to_whatsapp object_or_url "Share To WA" %} {% post_to_reddit object_or_url "Share To Reddit" %} <body> </body> </html>
Conclusion	User navigate to Facebook pages successfully when clicking the button.

Table 4.14 Test Case for Post to Facebook Button

4.4 Evaluation

After the test case has been carried out, the admin navigates to signup.html successfully. The admin can register as an admin successfully. The admin can sign in as an admin successfully. The user can sign up as a user successfully. The user can sign in as a user successfully. The user can search successfully. The user can navigate to Japanese_food page after clicking the Japanese_food. The user can sort the keyword successfully. The user can log-out successfully. The user can navigate successfully by cling the picture. The user can navigate to showorder.html successfully and record is saved inside the database. Messages show “Please Wait, your order will be arrived soon! Name” successfully. The nickname and comment of user is saved and displayed in table. Message which is “user nickname, your comment is posted” will be showed. User navigate to Facebook pages successfully when clicking the button. In summary, all the requirements in Tapau at inti are met.

4.5 Summary

In conclusion, in this testing phase all the requirements are checked to determine if the button or feature can work perfectly. Test case is carried out to check the requirement in detail. After the test case is done all the functions can work properly. So, we can conclude that all the requirements are met.

CHAPTER 5
CONCLUSION AND FUTURE WORK

5.1 Conclusion

During this pandemic, it is compulsory for students and college staffs to adhere Standard Operating Procedure (SOP) to avoid from infected by COVID-19. Students or staffs are expected to return back to campus soon and students or staffs will be in a difficult situation when they need to make order for food in canteen. It will be hassle for them to wait for queue while following the SOP. Tapau at Inti is designed and implemented to overcome this problem. A web-based food ordering system which is implementing can ensure the students can buy the food at everywhere and every time. It is more convenient compared to traditional food ordering system. The main feature of Tapau at Inti is it is a food ordering system which can saved the data of the customer or student inside a database which is sqlite3 and able to be retrieved by the admin or staff. It is a very convenient tool to store the data which makes the customer can order by using the laptop or mobile phone. This is a web-based application which is provided with a lot of attractive images with advance tools which is interactive. For example, student can make comment to the public. The user can share the recipe to their friend. Hopefully this application can help the students in the future.

5.2 Future work.

The table of Tapau at Inti is not well organized because of time constraint. Later if more time is given, the design of table which is viewed by the admin will be more nicely organized. The comment table will be able to show to the admin. Some designed part is needed to be strengthen and more useful and attractive feature will be added.

Reference:

Square. (2020). Everything You Need to Know About Setting Up an Online Food Ordering System

https://squareup.com/au/en/townsquare/what-is-an-online-food-ordering-system#:~:text=An%20online%20food%20ordering%20system%20generally%20has%20two%20components%20%E2%80%93%20a,receive%20and%20fulfil%20customer%20orders.

Mayurkumar, P. (2015). Online Food Order System for Restaurants. *Technical Library. 219.* https://scholarworks.gvsu.edu/cgi/viewcontent.cgi?article=1222&context=cistechlib

McCombes, S. (2021). How to write a literature review. Scribbr https://www.scribbr.com/dissertation/literature-review/

Trustpilot. (n.d.). foodpanda https://www.trustpilot.com/review/justeat.in

Alanna, T. (2020). I Was Today-Years-Old When I Realised These 7 Things You Can Do on GrabFood. *Vulcan Post.* https://vulcanpost.com/691293/grabfood-cool-features-singapore/

Truspilot (n.d.). Grab https://www.trustpilot.com/review/grab.com

Joycelyn, T. (2021) With ShopeeFood M'sia launched, here's the big-name food delivery app showdown of 2021. *Vulcan Post.* https://vulcanpost.com/764944/shopeefood-review-comparison-food-delivery-klang-valley/

Mr.X. (2021) Airasia Food Delivery Review. *Maggi Goreng.* https://maggigoreng.xyz/airasia-app-review/

G2. (n.d.). Deliveroo Reviews & Product Details. https://www.g2.com/products/deliveroo/reviews#survey-response-5076196

Full Stack Python. (n.d.). Web Framework. https://www.fullstackpython.com/web-frameworks.html

Benchmark. (n.d.). Let’s Understand the Pros and Cons of Using Django. https://www.benchmarkit.solutions/lets-understand-the-pros-and-cons-of-using-django/#:~:text=The%20benefits%20of%20Django%20are,inability%20to%20create%20smaller%20projects.

DDI Development. (2020). Best Python Framework for Web Development in 2020. https://ddi-dev.com/blog/programming/best-python-frameworks-for-web-development-in-2020/

Derry, M. (2021). What Is flask and How Do Developer Use it? A Quick Guide. https://careerfoundry.com/en/blog/web-development/what-is-flask/

Null. (2019). What is Captcha. https://www.pandasecurity.com/en/mediacenter/panda-security/what-is-captcha/

wallarm. (n.d.). What is Captcha? Types and Examples. https://www.wallarm.com/what/what-is-captcha-types-and-examples#:~:text=Text%2Dbased%2C%20picture%2Dbased,types%20of%20CAPTCHAs%20available%20today.

Kulkami, S., Fadewar, H. (2018). Audio CAPTCHA techniques: A Review. *Springer Link* https://link.springer.com/chapter/10.1007/978-981-10-8228-3_33#:~:text=Audio%20CAPTCHA%20is%20a%20reverse,of%20CAPTCHA%20for%20web%20security.

Slant. (n.d.) web2py Review. https://www.slant.co/options/1397/~web2py-review

Kumar, M. (2019). Strength and weakness of Various CAPTCHA Schemes http://proceeding.conferenceworld.in/IIMT_April_2019/48NQSxVoqTxeNN116.pdf

Altvater, A. (2020). What Is SDLC? Understand the Software Development Life Cycle. Stacktify
https://stackify.com/what-is-sdlc/

Rigamon, D. (2019). Agile vs Waterfall What are the differences? *Project Cubicle*
https://www.projectcubicle.com/similarities-and-differences-between-agile-vs-waterfall/

360Logical (n.d.) Agile Development – Advantages, Disadvantages and when to use it. *Logica.*
https://www.360logica.com/blog/agile-development-advantages-disadvantages-and-when-to-use-it/

tutorials point. (n.d.) SDLC-waterfall Model

McCombes, S. (2021). How to write a literature review. Scribbr
https://www.scribbr.com/dissertation/literature-review/

Zulqadar, A. (2019). SDLC Waterfall Model: The 6 phases you need to know about. Rezoid
https://rezaid.co.uk/sdlc-waterfall-model/

tenstep. (n.d.). 420.0 Design Phase
https://www.lifecyclestep.com/open/420.0DESIGNPHASE.htm

IBM, (n.d). Use-case diagram.
https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagrams-use-case

Upadhyay, I. (2020). What is a Modular Design? Everything You want to Know in 8 Easy Answers. *Jigsaw*
https://www.jigsawacademy.com/blogs/product-management/modular-design/

Lucid Content Team (n.d.). What the Waterfall Project Management Methodology Can (and Can't) Do for You https://www.lucidchart.com/blog/waterfall-project-management-methodology#:~:text=Simply%20put%2C%20waterfall%20project%20management,linear%20process%20of%20project%20management.&text=No%20phase%20begins%20until%20the,start%20over%20at%20phase%20one.

Tutorialspoint. (n.d.) Software Testing-Quick Guide. https://www.tutorialspoint.com/software_testing/software_testing_quick_guide.htm

Explorance. (2013) 5 Key Reasons Your Institution Needs to Conduct Program Evaluations.

https://explorance.com/blog/5-key-reasons-to-conduct-program-evaluations/

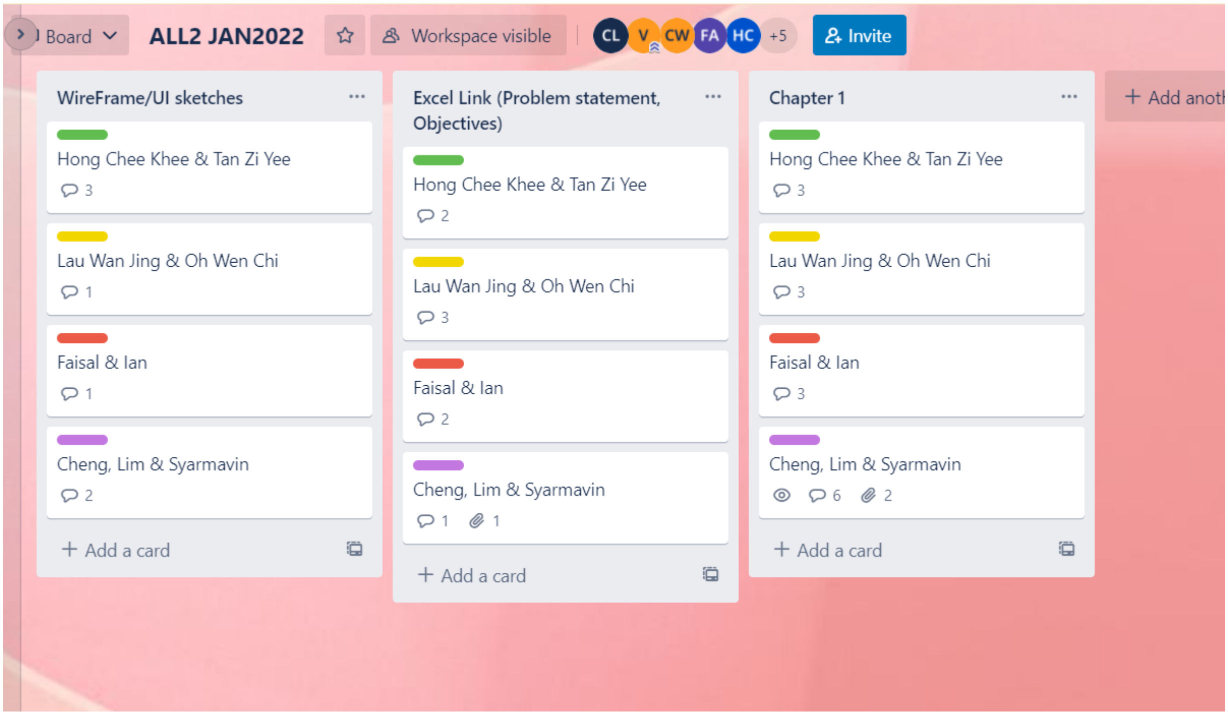
Meera. (n.d.). Evaluation: What is it and why do it? https://meera.snre.umich.edu/evaluation-what-it-and-why-do-it

Jansen, D. (2020). What (Exactly) Is Research Methodology? *GradCoach* https://gradcoach.com/what-is-research-methodology/

Types of evaluation. (n.d.). What are the most common types of evaluation? https://www.cdc.gov/std/program/pupestd/types%20of%20evaluation.pdf

Appendices

Trello



WireFrame/UI sketches

- Hong Chee Khee & Tan Zi Yee — 3
- Lau Wan Jing & Oh Wen Chi — 1
- Faisal & Ian — 1
- Cheng, Lim & Syarmavin — 2

+ Add a card

Excel Link (Problem statement, Objectives)

- Hong Chee Khee & Tan Zi Yee — 2
- Lau Wan Jing & Oh Wen Chi — 3
- Faisal & Ian — 2
- Cheng, Lim & Syarmavin — 1, 1

+ Add a card

Chapter 1

- Hong Chee Khee & Tan Zi Yee — 3
- Lau Wan Jing & Oh Wen Chi — 3
- Faisal & Ian — 3
- Cheng, Lim & Syarmavin — 6, 2

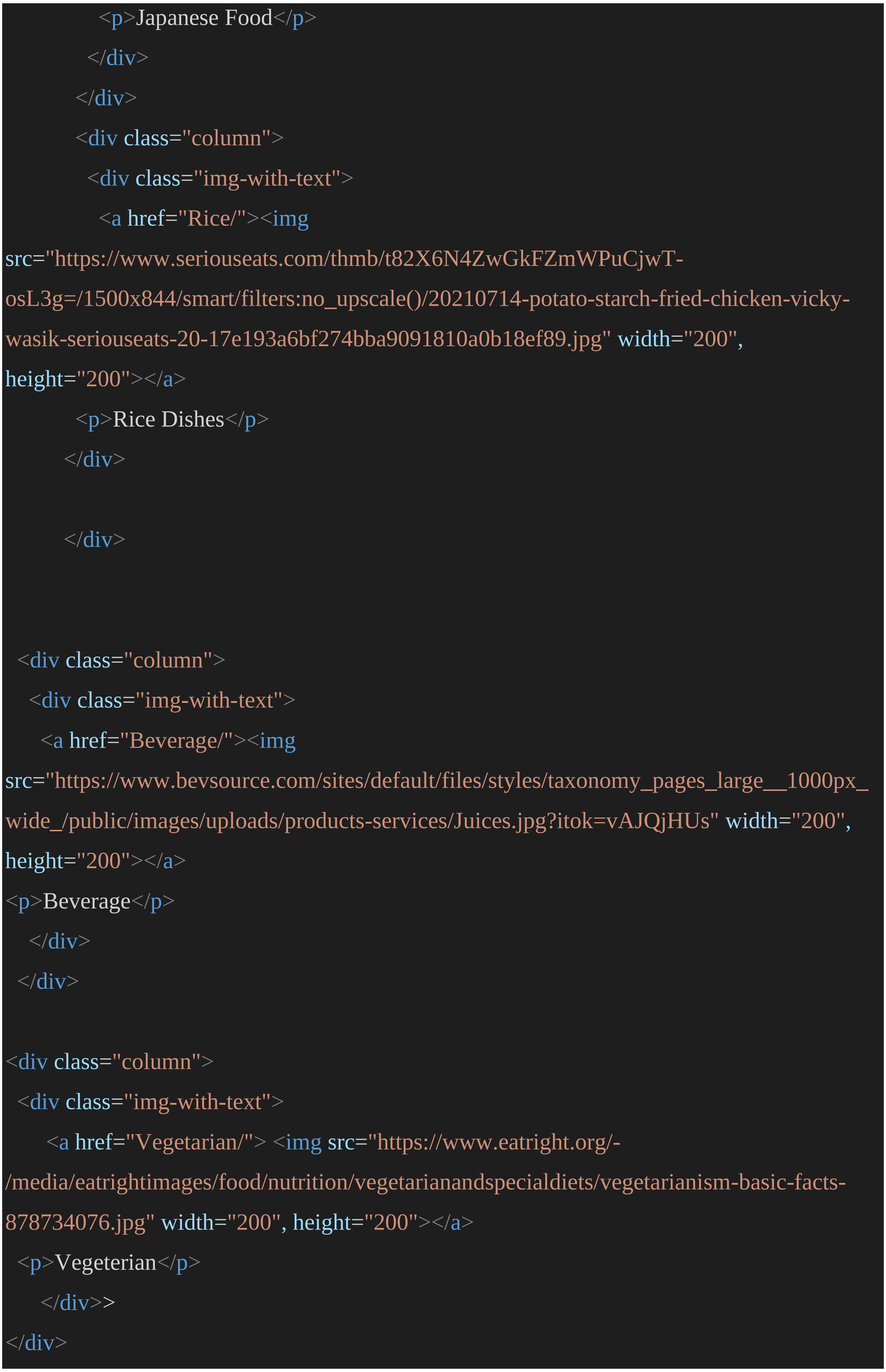
+ Add a card

Main_menu.html

```
<!DOCTYPE html>

<title>Food ordering System </title>
<head>
   <meta charset="UTF-8">
   <meta name="viewport"
    content="width=device-width, initial-scale=1.0">


</head>

<body>
```

```
  <a href="index/"> <h4>log-out</h4></a>

  <header>

    <h1>Tapau at Inti</h1>

    <h2>Dine in or Take away</h2>

  </header>
  <hr>

  <h2>Search Menu</h2>
<p>Start to type for a specific category inside the search bar to "filter" the search options.</p>

<div class="row">
 <div class="left" style="background-color:#bbb;">
  <h2>Menu</h2>
  <h3>Search for category</h3>
  <input type="text" id="mySearch" onkeyup="myFunction()" placeholder="Search.."
title="Type in a category">
  <ul id="myMenu">
   <li><a href="Japanese_food/">Japanese food</a></li>
   <li><a href="Beverage/">Beverage</a></li>
   <li><a href="Rice/">Rice Dishes</a></li>
   <li><a href="Vegetarian/">Vegeterian</a></li>

  </ul>
 </div>


 <div class="left" style="background-color:#bbb;">
  <h2>Recipe</h2>
  <h3>Search for recipe</h3>
```

```
    <input type="text" id="mySearch2" onkeyup="myFunction2()" placeholder="Search.."
title="Type in a category">
    <ul id="myMenu2">
      <li><a href="recipe">Nasi Lemak recipe</a></li>
      <li><a href="recipe2">Hainan Chicken rice recipe</a></li>
    </ul>
  </div>





<script>
function myFunction() {
  var input, filter, ul, li, a, i;
  input = document.getElementById("mySearch");
  filter = input.value.toUpperCase();
  ul = document.getElementById("myMenu");
  li = ul.getElementsByTagName("li");
  for (i = 0; i < li.length; i++) {
    a = li[i].getElementsByTagName("a")[0];
    if (a.innerHTML.toUpperCase().indexOf(filter) > -1) {
      li[i].style.display = "";
    } else {
      li[i].style.display = "none";
    }
  }
}

function myFunction2() {
  var input, filter, ul, li, a, i;
  input = document.getElementById("mySearch2");
  filter = input.value.toUpperCase();
  ul = document.getElementById("myMenu2");
  li = ul.getElementsByTagName("li");
  for (i = 0; i < li.length; i++) {
    a = li[i].getElementsByTagName("a")[0];
    if (a.innerHTML.toUpperCase().indexOf(filter) > -1) {
      li[i].style.display = "";
    } else {
      li[i].style.display = "none";
    }
  }
}
</script>




            <p>Cuisines</p>
            <div class="label">
            <p1>Choose anything you like from the Menu</p1>
          </div>
        <br><br>

        <div class="row2">
            <div class=" column">

              <div class="img-with-text">
                <a href="Japanese_food/"><img
src="https://cdn.tasteatlas.com//Images/Dishes/fde953e5aa504e8c83a84691582b0399.jpg?w
=905&h=510 " width="200", height="200"></a>
```



```
        <p>Japanese Food</p>
       </div>
      </div>
      <div class="column">
       <div class="img-with-text">
        <a href="Rice/"><img
src="https://www.seriouseats.com/thmb/t82X6N4ZwGkFZmWPuCjwT-
osL3g=/1500x844/smart/filters:no_upscale()/20210714-potato-starch-fried-chicken-vicky-
wasik-seriouseats-20-17e193a6bf274bba9091810a0b18ef89.jpg" width="200",
height="200"></a>
      <p>Rice Dishes</p>
     </div>

     </div>



 <div class="column">
  <div class="img-with-text">
   <a href="Beverage/"><img
src="https://www.bevsource.com/sites/default/files/styles/taxonomy_pages_large__1000px_
wide_/public/images/uploads/products-services/Juices.jpg?itok=vAJQjHUs" width="200",
height="200"></a>
<p>Beverage</p>
  </div>
 </div>

<div class="column">
 <div class="img-with-text">
    <a href="Vegetarian/"> <img src="https://www.eatright.org/-
/media/eatrightimages/food/nutrition/vegetarianandspecialdiets/vegetarianism-basic-facts-
878734076.jpg" width="200", height="200"></a>
 <p>Vegeterian</p>
   </div>>
</div>
```

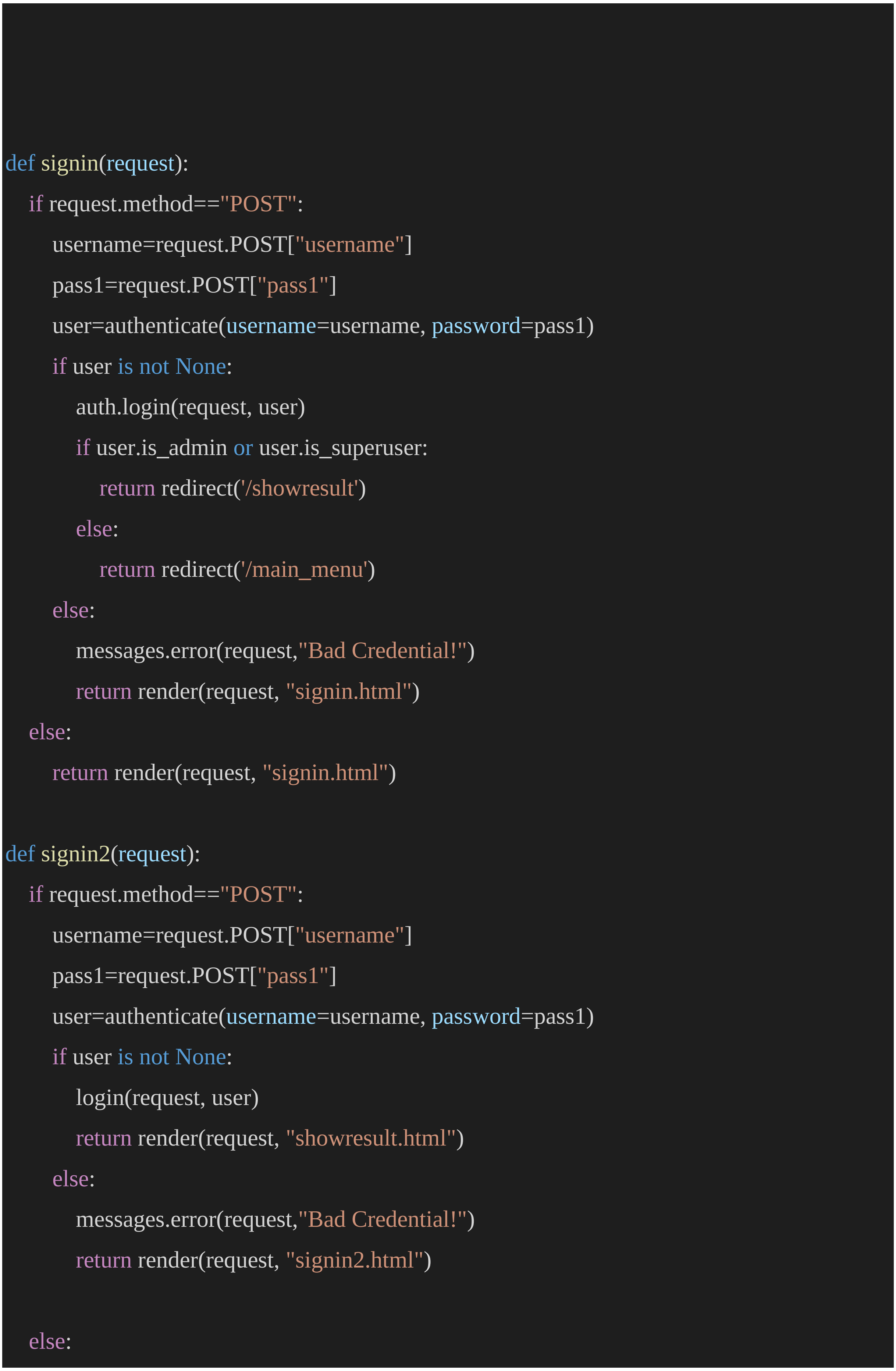
```
  </div>



 <div class="row3">
  <div class="Contact">
   <br>
  <br><br><br>


  <br>
 <p>Contact us:012-4820649  </p>
  <p>email: chenteik_99@hotmail.com </p>
  </div>
 </div>





<style>
 .row1 {
   box-sizing: border-box;
  }

  .column {
   float: left;
   width: 20%;
   padding: 2px;
  }
  .row2 {
   box-sizing: border-box;
  }
```

```
        .column {
          float: left;
          width: 20%;
          padding: 2px;
        }

        .img-with-text {
    text-align: center;
    width: [width of img];
    font-size: 100px;
}




.img-with-text img {
    display: block;
    margin: 0 auto;
}

h1{
  text-align: center;
}
h2{
  text-align: center;
}
h3{
  text-align: left;
  font-size: 20px;
}

h4{
  text-align: right;
  font-size: 20px;
```

```
}
p{
  font-size: 20px;
  text-align: center;
}

.label{
  font-size: 20px;
  text-align: center;
}



/* width */
::-webkit-scrollbar {
  width: 10px;
}

/* Track */
::-webkit-scrollbar-track {
  background: #f1f1f1;
}

/* Handle */
::-webkit-scrollbar-thumb {
  background: #888;
}

/* Handle on hover */
::-webkit-scrollbar-thumb:hover {
  background: #555;
}

body {
  padding: 20px;
  width: 1490px;
  height: 1800px;
  resize: both;
  overflow: auto;
}

.Contact {
          float: left;
          width: 100%;
          padding: 2px;
        }




        </style>

</body>
</html>
```

Views.py

```
from django.shortcuts import render
from django.conf import settings
from django.http import FileResponse, HttpRequest, HttpResponse
from django.views.decorators.cache import cache_control
from django.views.decorators.http import require_GET
from logging import exception
from multiprocessing import AuthenticationError
from django import forms
from django.contrib import auth
from django.shortcuts import render,redirect
from django.contrib.auth.forms import UserCreationForm,AuthenticationForm
from django.contrib.auth.models import User
from .models import Order
from django.contrib.auth import authenticate,login
from django.contrib import messages
from django.views.generic import CreateView
from .forms import RegistrationForm, AdminStudentRegistrationForm
from .forms import OrderForm
from .forms import CommentForm
from bootstrap_datepicker_plus.widgets import DatePickerInput
from .models import Order
from .models import Comment




def home(request):
    return render(request, "index.html")


class RegistrationView(CreateView):
    model = User
    form_class = RegistrationForm
    template_name = 'signup.html'


    def form_valid(self, form):
        user = form.save()
        login(self.request, user)
        # return redirect('learner')
        return redirect('/signin')

class AdminStudentRegisterView(CreateView):
    model = User
    form_class = AdminStudentRegistrationForm
    template_name = 'signup2.html'

    def form_valid(self, form):
        user = form.save()
        login(self.request, user)
        # return redirect('learner')
        return redirect('/signin')

class OrderView(CreateView):

    model = Order
    form_class = OrderForm
    template_name = 'order.html'



    def form_valid(self, form):
        order = form.save()
        # return redirect('learner')
        messages.success(self.request, 'Please Wait, your order will be arrived soon!%s  '%order.Name)
        Name=order.Name
        Contact_number=order.Contact_number
        Japanese_food=order.Japanese_food
        Japanese_food_quantity=order.Japanese_food_quantity
        Rice_Dishes=order.Rice_Dishes
        Rice_Dishes_quantity=order.Rice_Dishes_quantity
        Beverage=order.Beverage
        Beverage_quantity=order.Beverage_quantity
        Vegetarian=order.Vegetarian
        Vegetarian_quantity=order.Vegetarian_quantity
        order_date=order.order_date
        order_time=order.order_time
        Taking_method=order.Taking_method
        payment_method=order.payment_method
        return render(self.request, "showorder.html",{'Name':Name,
"Contact_number":Contact_number, "Japanese_food":Japanese_food,
"Japanese_food_quantity":Japanese_food_quantity, "Rice_Dishes":Rice_Dishes,
"Rice_Dishes_quantity":Rice_Dishes_quantity, "Beverage": Beverage,
"Beverage_quantity":Beverage_quantity, "Vegetarian":Vegetarian,
"Vegetarian_quantity":Vegetarian_quantity, "order_date":order_date,
"order_time":order_time, "Taking_method":Taking_method,
"payment_method":payment_method})


def showresult(request):
    context={}
    context["dataset"] = Order.objects.all()
    return render(request, "showresult.html", context)



class CommentView(CreateView):

    model = Comment
    form_class = CommentForm
    template_name = 'comment.html'


    def form_valid(self, form):
        comment = form.save()
        # return redirect('learner')
        Nickname=comment.Nickname
        Comment=comment.Comment
        messages.success(self.request, '%s, your comment is posted  '%comment.Nickname)
        return render(self.request, "showcomment.html",{'Nickname':Nickname,
"Comment":Comment})
```



```
def signin(request):
    if request.method=="POST":
        username=request.POST["username"]
        pass1=request.POST["pass1"]
        user=authenticate(username=username, password=pass1)
        if user is not None:
            auth.login(request, user)
            if user.is_admin or user.is_superuser:
                return redirect('/showresult')
            else:
                return redirect('/main_menu')
        else:
            messages.error(request,"Bad Credential!")
            return render(request, "signin.html")
    else:
        return render(request, "signin.html")

def signin2(request):
    if request.method=="POST":
        username=request.POST["username"]
        pass1=request.POST["pass1"]
        user=authenticate(username=username, password=pass1)
        if user is not None:
            login(request, user)
            return render(request, "showresult.html")
        else:
            messages.error(request,"Bad Credential!")
            return render(request, "signin2.html")

    else:
        return render(request, "signin2.html")

def signout(request):
    return render(request, "index.html")


def menu(request):
    return render(request, 'main_menu.html')
def Japanese_food(request):

    return render(request, 'Japanese_food.html')

def Rice(request):

    return render(request, 'Rice.html')
def Beverage(request):

    return render(request, 'Beverage.html')

def Vegetarian(request):
    return render (request, 'Vegetarian.html')

def rating(request):
    return render(request,'comment.html')

def recipe(request):
    return render(request,'recipe.html')

def recipe2(request):
    return render(request,'recipe2.html')

def share(request):
    return render(request, "share.html")
```

[redacted]

Gantt Chart

Task Name	Day								
	12Jan	19Jan	26Jan	2Feb	9feb	16feb	23Feb	28Feb	Submitted date
Gathering information	-								-
Design		-							-
Implementation			-	-					-
testing					-	-			-
Follow up							-	-	-