FOOD ORDERING APPLICATION

By

Chang Zi Hong

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONOURS)
DIGITAL ECONOMY TECHNOLOGY
Faculty of Information and Communication Technology
(Kampar Campus)

JUNE 2025

COPYRIGHT STATEMENT

This Final Year Project proposal is submitted in partial fulfillment of the requirements for the degree of Bachelor of Information Systems (Honours) Digital Economy Technology at Universiti Tunku Abdul Rahman (UTAR). This Final Year Project proposal represents the work of the author, except where due acknowledgment has been made in the text. No part of this Final Year Project proposal may be reproduced, stored, or transmitted in any form or by any means, whether electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the author or UTAR, in accordance with UTAR's Intellectual Property Policy.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisors, **Ts Dr Wong Pei Voon** who has given me this bright opportunity to engage in a food order system project. It is my first step to establish a career in food order field. A million thanks to you.

Finally, I must say thanks to my parents and my family for their love, support, and continuous encouragement throughout the course.

ABSTRACT

The demand for food ordering services has caused several problems in improving customer satisfaction and operational efficiency. This research presents a food ordering system that solves issues like food suggestion for people with allergies and specific dietary needs, delays in order processing, and the lack of ingredients in manage side. The system offers a tailored food ordering experience, focusing on user safety by providing allergy-sensitive meal suggestions, accurate preparation time estimates, and manage ingredients efficiently. The system can also improve kitchen workflows, reduce delays and improve customer satisfaction. Also, the integration of inventory management lets restaurant owner make early decisions about ingredients. This research helps develop smart, efficient, and clear food ordering systems, improving user experience without focusing on delivery and payment processes.

Area of Study (Maximum 2): Web Application

Keywords (Maximum 10): Allergy Management, Order Processing Optimization, Inventory Management, Chef Management, Customer Satisfaction

Table of Contents

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: Introduction

Introduction

In this chapter, it will present the background and motivation of the research, the contributions to the field, and an outline of the thesis. The chapter begins by introducing the core challenges faced by food ordering applications. It then outlines the research objectives and describes the scope and direction of the work undertaken.

1.1 Problem Statement and Motivation

Food delivery services have seen remarkable growth in recent years, driven by consumer demand for convenience, variety, and the ability to order meals from a diverse range of restaurants at the touch of a button. Not only that, due to their popularity, food ordering systems face some challenges that risk both customer satisfaction and operational effectiveness. These challenges include the lack of accurate food recommendations for individuals with allergies or specific food needs, delays in order processing, and limited access to food inventory management, which can potentially affect consumer health.

1.1.1 Challenges in Finding Safe Foods for Allergies and Diets

Consumers with food allergies sometimes have difficulty finding safe and suitable foods when using food ordering apps, as current systems usually offer general food choices that ignore important details such as allergies restrictions. According to Golding et al. [1], this lack may trigger health risks because food allergies can cause serious symptoms, and food choices are becoming increasingly important. There is a clear need for a feature that looks at user profiles and matches food ingredients to allergy information. This function would provide more secure and convenient food options, therefore enhancing the food ordering experience while increasing a healthy lifestyle.[8]

1.1.2 Order Processing Delays

Another major issue in food ordering systems is the delay in processing orders, particularly during peak hours. Many existing systems rely on manual processes, and inefficiencies in kitchen workflows often result in longer wait times for customers. Kumar [3] discusses how current food delivery systems frequently struggle with delays in processing orders due to a lack of predictive modelling to estimate kitchen workload. As food delivery becomes more mainstream, these delays may result in dissatisfied customers, lower retention rates, and loss of business. Research into efficient order management systems has demonstrated that utilizing predictive algorithms to estimate preparation times can significantly enhance the accuracy of delivery schedules and minimize delays. Efficient order management and real-time tracking of kitchen workflows can improve the overall user experience by providing timely updates on order status.

1.1.3 Ingredient Delays and Shortages in Online Ordering

Restaurant owners might encounter delays in ingredient accessibility, especially in fast online ordering settings. Without automated inventory management, there could be insufficient ingredients available to create current food items that may cause cancellations, wasted time, and unhappy customers. This problem results by poor inventory management and the absence of cohesive digital tools to effectively monitor ingredient usage. Creative smart inventory management systems can solve operational inefficiencies by automating updates in real time [5]. Moreover, the adoption of Industry 4.0 technologies in food supply chains enhances operational management and minimizes waste, aligning with the principles of a circular economy [11].

1.2 Research Objectives

The objective of this research is to create a food ordering application that uses modern methods to address the problems raised above. The specific research objectives are as below:

Objective 1: Food Detect for Allergies and Dietary Needs

The first objective is to develop a function that matches food ingredients to the user's allergy information and food preferences. The function will examine the ingredients of available food and compare them to the user's allergies and food preferences, ensuring that the foods do not cause allergic reactions or dietary inconsistencies. This leads to a safer and more satisfying choice of food experience. [2].

Objective 2: Efficient Order Processing and Estimation of Preparation Time

The second objective is to create a process that estimated food preparation times in real time, thus increasing operational efficiency with historical data, kitchen workload patterns, and real-time inputs, the method will provide customers with more accurate preparation time estimates, reducing wait times and increasing customer satisfaction. Kumar [3] states that estimation methods, when properly implemented, can help optimize kitchen operations by ensuring timely order accuracy, especially during peak hours.

Objective 3: Enhancing Restaurant Operations

The third objective is integrating an automated inventory management system into the food ordering application to combat ingredient delays and shortages. When ingredients are insufficient or unavailable, the related menu items are automatically marked as "sold out," ensuring customers only place orders that can be fulfilled. This increases trust, minimizes cancellations, and reduces wasted preparation time. Automated updates also enhance decision-making and operational efficiency in restaurant management [5]. Moreover, adopting Industry 4.0 technologies enables predictive stock monitoring, reduces food waste, and supports sustainable practices across the restaurant supply chain [11].

1.3 Project Scope and Direction

For this project, the key features will include an allergy check to ensure meal safety, an estimated preparation time for each dish, and the ability to view restaurant hygiene ratings. These features prioritize the safety and quality of the dining experience, ensuring users can make informed choices when ordering. Delivery and payment aspects will not be the main focus.

User Management Module

• Restaurant owner can register/login

• Customer can register/login

• All users can update their profile information

Allergy Check Module

• Restaurant owner can add ingredients details

• Customer can add allergy details

This module will check the ingredients details and allergy details, if match then warning the customer.

Restaurant Product Module

• Restaurant owner can add/edit/delete their own products

• Restaurant owner can upload product details:

- Ingredients
- Preparation time
- Hygiene certification images

  - Customer can view all available products

Order Management Module

• Customer can:

- Add products to cart
- Place orders
- View their order history
- View the product with allergy check

Sales Analytics Module

- Restaurant owners can view sales for their products
- Restaurant owners can track popular items
- Restaurant owners can view sales data

Chef Module

- Restaurant owners can add new staff
- Restaurant owners can manage their staff (Active/On leave/delete)

Inventory Management Module

- Restaurant owners can add stock by specific ingredients or foods
- Restaurant owners can check the delay of each ingredient

Admin Module

- Admin can approve the restaurant owner accounts
- Admin can view all the existing restaurant owners

Preparation Time Module

• Restaurant owner can:

- Set base preparation time per item
- Calculate the estimate preparation time

• Customer can:

- See estimate preparation time before ordering

1.4 Contributions

This research provides several valuable contributions to the fields of food technology, AI, and website development:

Allergy-Sensitive Food Warning:

The development of a recommendation system tailored specifically to user allergies and dietary restrictions is a novel approach in food order services. By analysing user profiles, the system will identify ingredients that may pose a risk to individuals with food allergies and provide warnings or suggestions accordingly. This system helps users make safer food choices, reducing the risk of allergic reactions. The importance of such allergy-sensitive meal suggestions has been highlighted by Gymnopoulos and Dimitropoulos [2], who emphasized that personalized nutrition solutions focused on dietary restrictions are essential for improving user safety and satisfaction in food ordering platforms.

Enhancing Operational Efficiency:

Through the estimating preparation times, this research will optimize the kitchen workflow and improve order management. As Kumar [3] discusses, predictive techniques can help streamline restaurant operations, thus reducing delays and enhancing the overall customer experience.

Easy restock and reduce the manual key in:

Food ordering system to automatically update menu availability and ingredients arrived time. By linking ingredient stock levels directly to ordering systems, restaurants can prevent cancellations caused by unavailable items, improve customer trust, and enhance operational efficiency. Furthermore, the approach leverages Industry 4.0 technologies to enable predictive stock monitoring, reduce food waste, and support sustainable restaurant supply chain practices.

1.5 Report Organization

The details of this research are presented in the following chapters. **Chapter 1** introduces the research background, the motivation for the study, the problem statements, and the objectives of the research, as well as the overall scope and contributions of the study. **Chapter 2** provides a review of the relevant literature, offering insights into existing systems and technologies related to food ordering applications, allergy-sensitive meal recommendations, order processing optimization, and manage ingredients. This chapter also identifies current gaps in the field and sets the stage for the proposed methodology. In **Chapter 3**, the system methodology and overall approach are discussed in detail, outlining the architectural style, development process, and key techniques used to address the challenges identified earlier, including allergy-sensitive detection, preparation-time prediction, and inventory management. **Chapter 4** presents the system design, including the principal modelling artefacts and diagrams—such as use-case diagrams, entity–relationship diagrams, and class diagrams—that define the structure and behaviour of the application. **Chapter 5** describes the implementation, detailing the software stack, core modules, data flow, and user interface components that realise the designed features in practice. **Chapter 6** reports the unit testing strategy and results, summarising the test cases, expected and actual outcomes, and the extent to which the implementation satisfies the stated objectives. Finally, **Chapter 7** concludes the thesis by summarising the findings, contributions, and significance of the research, and by offering recommendations for future work and potential directions for system improvement.

CHAPTER 2: Literature Review

2.1 FeedMe POS

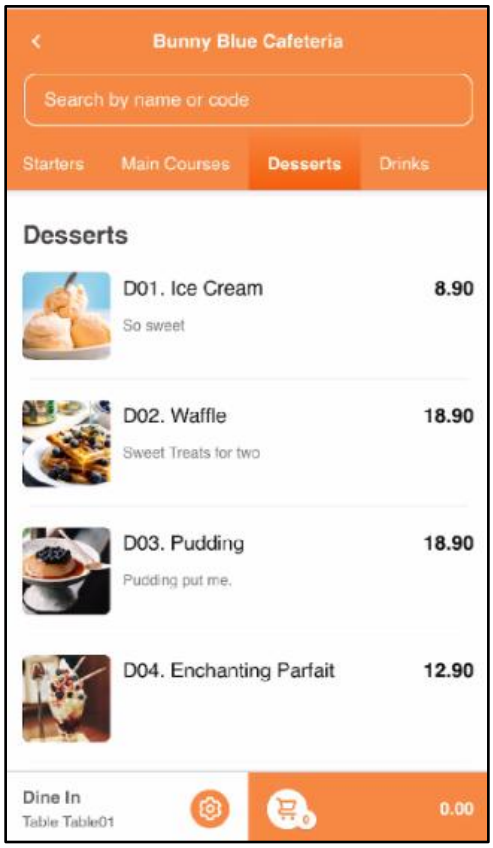


Figure 2. 1 FeedMe Food Ordering Interface

FeedMe POS, with its QR code ordering system, provides restaurants with a modern, digital ordering experience. Customers can use their mobile devices to browse the digital menu, place orders, and pay by scanning a QR code on their table [3]. Customers may view the food category, photos, and pricing of the item they want to order. They can add the food to their cart by clicking the cart button shown below.

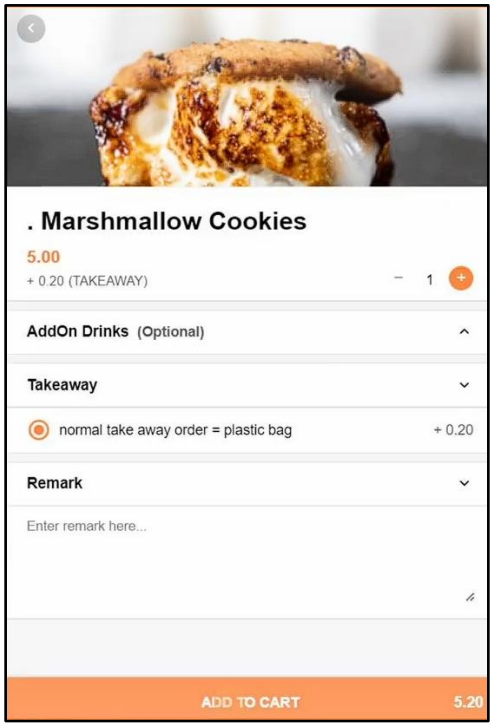


Figure 2. 2 FeedMe Meal Order Information

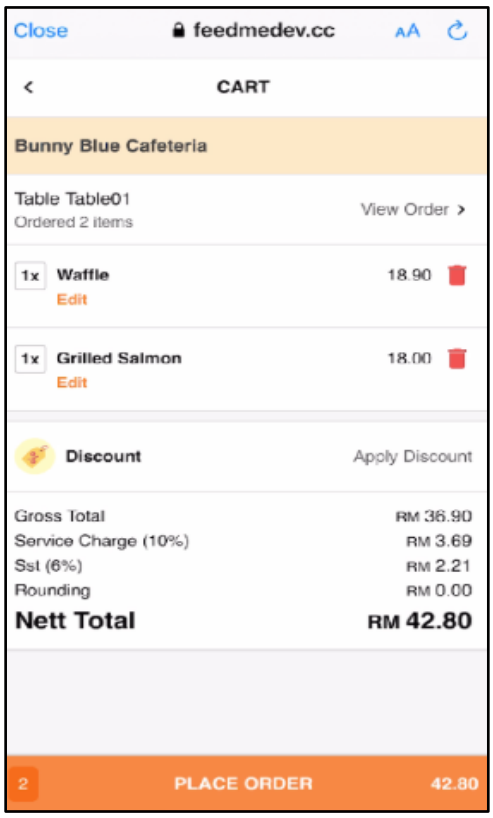


Figure 2. 3 FeedMe Cart

Customers can change the quantity of food and add a note in the FeedMe meal order information [1]. Customers can examine their order as well as price distribution details in the FeedMe basket. They can complete their orders by clicking the place order button [8]. Despite its impressive list of capabilities, FeedMe POS does not handle food allergies or dietary safety. The technology provides digital menus and ordering convenience, but meal ingredients are not matched to consumer allergy profiles, leaving health-conscious customers underserved [9]. Another issue is the lack of predictive preparation-time prediction; while FeedMe improves order flow with QR ordering and KDS connection [9], it does not calculate dynamic preparation times based on kitchen workload or past order patterns. This may still cause customer displeasure during peak hours.

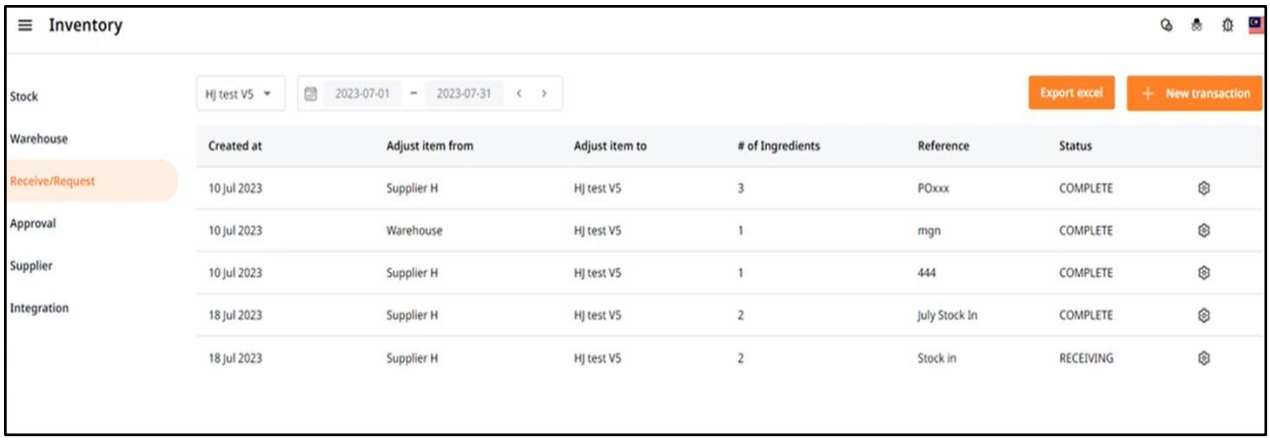


Figure 2. 4 FeedMe Inventory Management

FeedMe POS has a robust inventory management module that monitors product movement throughout the restaurant. The system enables restaurant managers to track ingredient levels in real time, including stock quantities, expiry dates, and days until shelf life [3]. The inventory operation includes sales (stock-out), purchases (stock-in), transfers, and modifications. While FeedMe offers ingredient tracking for employees, it does not appear to immediately disable menu items when related ingredients run out. This can result in circumstances where clients purchase dishes that cannot be supplied, causing cancellations and frustration.

2.2 StoreHub

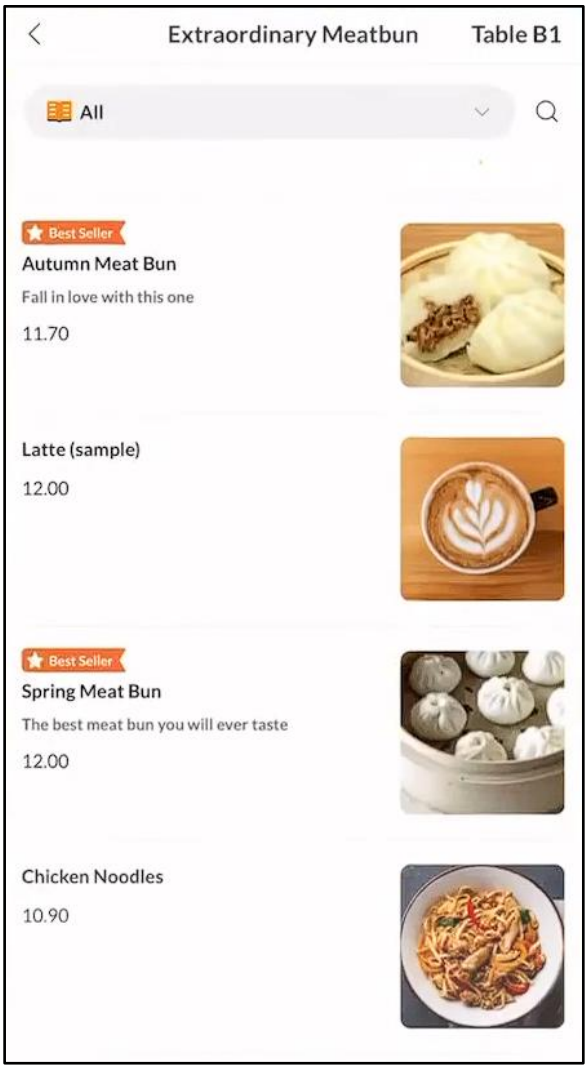


Figure 2. 5 StoreHub User Interface

StoreHub offers restaurants a modern food ordering functionality with its QR Order & Pay feature and Beep Delivery service [3]. Customers can use their mobile devices to scan a QR code at their table, view the restaurant's digital menu, place orders, and pay quickly. This reduces the reliance on waitstaff for order taking and increases table turnover.

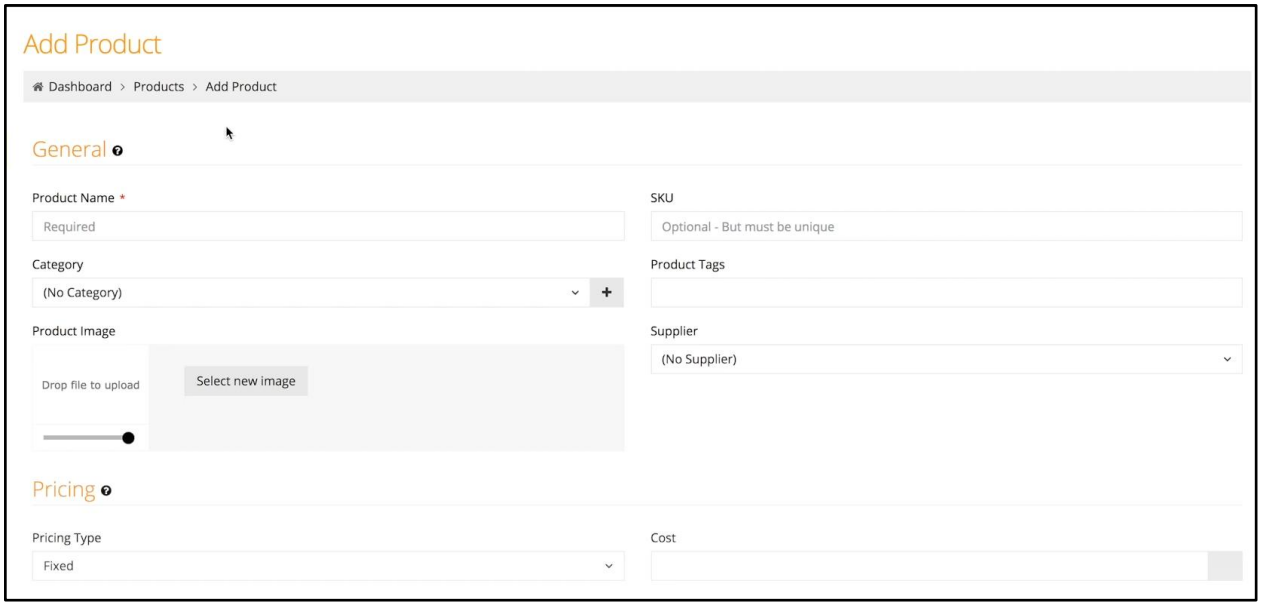


Figure 2. 6 StoreHub Inventory Management

StoreHub provides a fully integrated inventory management module with ingredient-level tracking, which is useful for food and beverage establishments [1,3,5]. The system tracks stock movements such as usage, receipt, transfer, and stock-taking [5,7]. It also delivers real-time information and notifications when specific ingredients are running low, giving restaurant owners insight into their operational demands [6].

Figure 2. 7 StoreHub Table Summary

Despite its impressive features, StoreHub has certain limits. Firstly, the system does not support food allergy or dietary safety detection, which is a major problem for clients who have health-related dietary limitations. Second, StoreHub lacks predictive preparation-time estimation, thus customers cannot receive correct wait time feedback during busy hours. Finally, while it tracks ingredients, it does not instantly disable menu items when the related ingredients run out, which might cause customer dissatisfaction when unavailable dishes are purchased.

2.3 Eats365

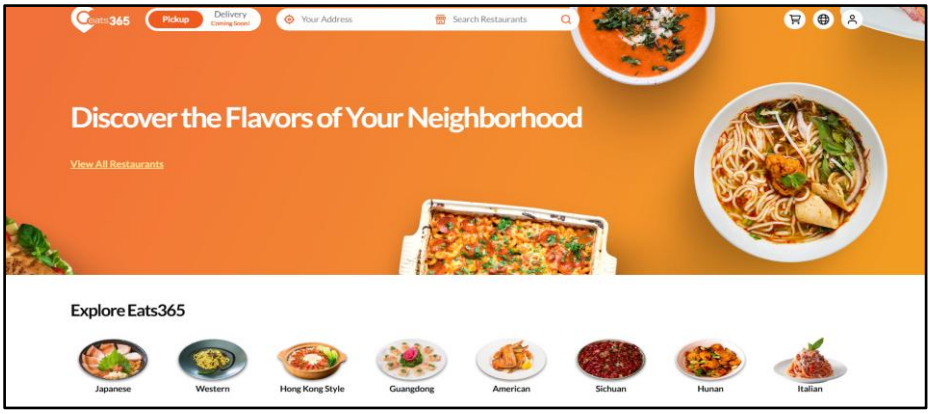


Figure 2. 8 Eats365 User Interface

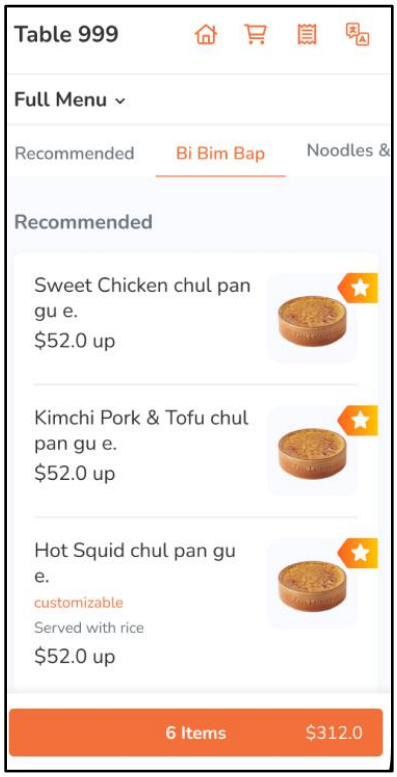


Figure 2. 9 Eats365 Food Ordering Interface

Eats365 positions itself as a whole restaurant ecosystem, combining POS operations with online ordering possibilities [11]. Customers may place orders seamlessly across multiple channels, including dine-in, takeaway, and delivery, and all orders are routed directly to the Kitchen Display System (KDS).

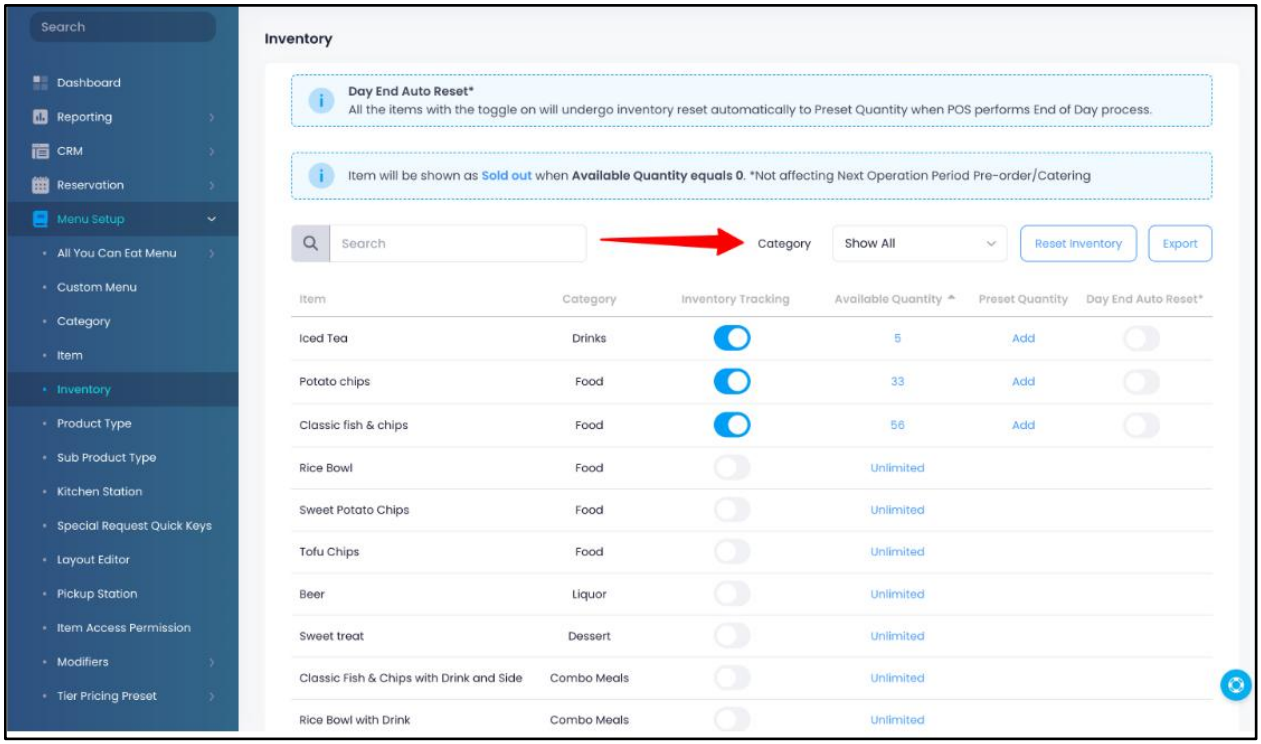


Figure 2. 10 Eats365 Inventory Management

The strength of Eats365 lies in its Back Office integration. It connects with third-party platforms such as Food Market Hub to manage inventory [1,5]. This integration enables automated vendor purchase orders, real-time ingredient tracking, and synchronisation between sales and stock levels. Each menu item can be linked directly to its required ingredients, ensuring that consumption is recorded accurately [2,4]. Reports generated from the back-end system help managers identify stock usage patterns and optimise restocking decisions [7].

Despite strong integration with third-party inventory platforms, Eats365 still faces shortcomings. The system does not support allergy or dietary detection, meaning customers with health requirements are still underserved. Additionally, Eats365 does not include predictive preparation-time features, instead focusing heavily on backend efficiency. Another issue is that the reliance on third-party integrations for stock tracking may delay real-time updates to customers, creating potential mismatches between inventory and menu visibility.

2.4 Slurp! POS

Figure 2. 11 Slurp! POS Food Ordering Interface

Slurp! POS emphasizes in self-service ordering using QR codes, an online ordering platform, and integration with delivery services [6]. Customers can access the menu digitally, place orders, and have them routed immediately to the Kitchen Display System (KDS) without the need for staff participation.

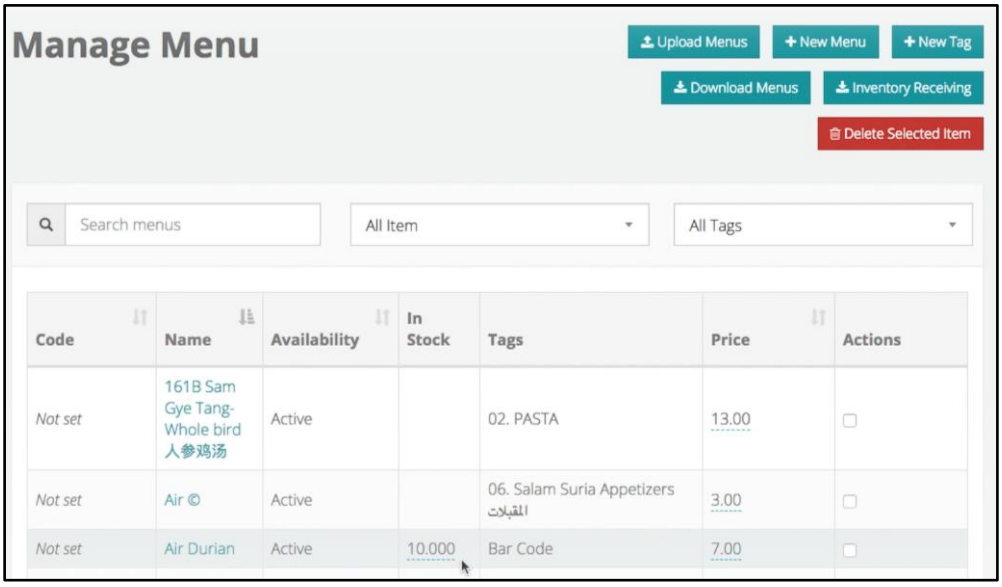


Figure 2. 12 Slurp! POS Menu Management

The system features an Inventory Management module, which allows restaurants to track stock usage in connection to sales. It tracks stock loss when menu items are sold and generates information that owners can use to estimate demand and purchase needs [11]. Slurp! assists owners of restaurants in reducing food waste and improving stock control by linking sales data to inventory operations [5,9].

While Slurp! provides useful fundamental functions, its drawbacks originate from a lack of security for customers and intelligence components. It does not detect allergies or food preferences, so consumers with limits must address these problems manually. The system also lacks predictive preparation-time estimation, so customers may have inaccurate expectations for food delivery or table service during peak hours. Lastly, when the inventory system runs out of ingredients, the unavailable menu options are not automatically disabled.

2.5 APICBASE

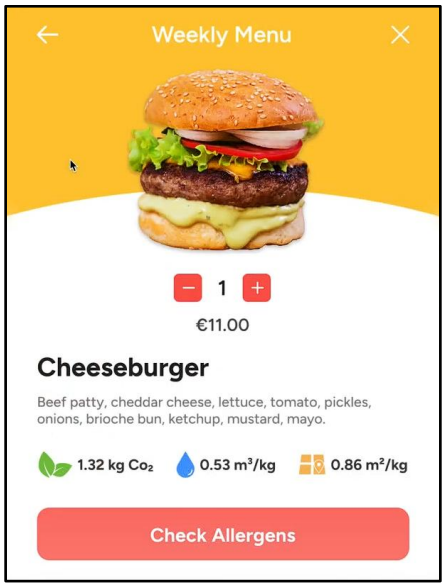


Figure 2. 13 APICBASE Food Allergen Management

By adding automatic allergy management to menu and recipe data, APICBASE makes it easier to order food by making sure that customers get clear and accurate information when choosing meals [5]. The application automatically generates and updates allergen information for the final dish when you combine ingredients into recipes. Each ingredient in the system is tagged with the right allergen attributes. This information may then be easily added to digital menus or online ordering systems, which lets customers find possible allergens or sort dishes by their dietary needs. By providing this level of transparency, APICBASE promotes better decisions for consumers with food allergies and contributes to higher food safety standards in the ordering environment [5,7].

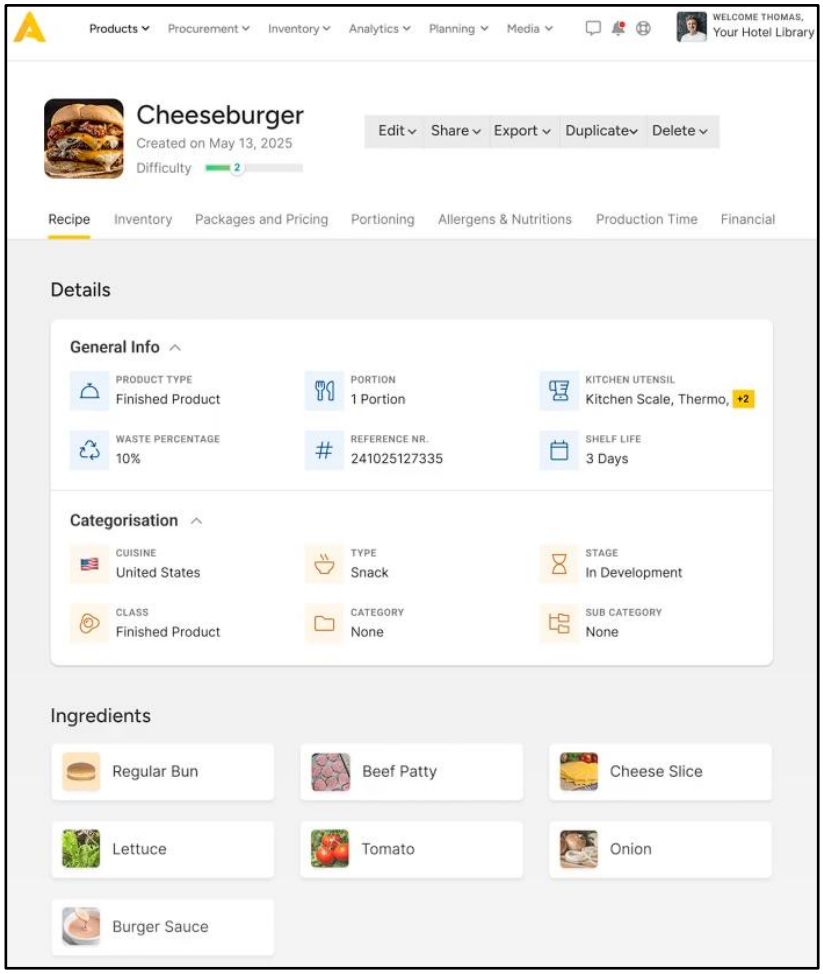


Figure 2. 14 APICBASE Inventory Management

APICBASE offers an advanced inventory management system that combines menu planning, restaurant owner management, and ingredient-level tracking. Restaurants can use the platform to forecast purchasing needs based on past consumption patterns, automatically deduct ingredients as orders are placed, and track stock levels in real time [7]. APICBASE reduces waste, avoids unplanned stockouts, and aids in cost control by precisely monitoring usage by connecting recipes to inventory data [5,8].

Despite its robust functionality, APICBASE presents certain limitations in the context of end-to-end food ordering. The system is primarily designed as a back-end management platform, meaning it does not operate as a standalone customer-facing ordering application. Instead, it requires integration with POS or online ordering systems to deliver allergen and inventory data to customers, which may increase implementation complexity and cost.

2.6 Comparative summary of existing system and proposed application

Feature	FeedMe pos	storehub	eats365	Slurp! pos	apicbase	proposed application
Allergen Information	No	No	No	No	Yes	Yes
Ingredient List	No	No	Yes	No	Yes	Yes
Preparation Time Estimate	No	No	Yes	No	No	Yes
Inventory management system	Yes	Yes	Yes	Yes	Yes	Yes

Table 2. 1 Summary comparison of each system

Based on the comparison of existing food ordering and management systems in Malaysia, it is evident that current platforms such as FeedMe POS, StoreHub, Eats365, Slurp! POS, and APICBASE offer basic ordering, menu setup, and inventory tracking functionalities, they fail to adequately address several critical challenges faced by both customers and restaurant owners. These challenges directly relate to the problem statements identified earlier, and the proposed system has been designed to bridge these gaps.

Firstly, in addressing the challenge of finding safe foods for allergies and dietary needs, most of the reviewed systems only provide general meal descriptions without the ability to filter or match meals against user-specific allergy profiles. As shown in the comparison table, there is only an existing system that provides allergen detection or detailed ingredient transparency, highlighting a significant gap in ensuring food safety for consumers with dietary restrictions. This limitation presents customers with special dietary needs or food allergies may be at risk for health problems as a result of this restriction. By adding a Food Detect feature that compares ingredient lists to the customer's dietary and allergy information, the suggested solution directly addresses this problem. This is in accordance with Objective 1, which is to improve food safety and transparency in order to boost user confidence and promote general health and wellbeing.

Second, current systems for order processing delays either don't estimate preparation times (APICBASE, FeedMe POS) or only use the restaurant owner's manual input (Eats365, Slurp! POS). This frequently leads to inefficiencies, particularly during times of high demand, which can result in unhappy customers and even business loss. By using historical data, current kitchen workload, and recurring order patterns, the suggested solution, on the other hand, fills this gap through predictive preparation time estimation. By doing this, the system achieves Objective 2, providing customers with more precise and trustworthy estimates of preparation times, cutting down on delays, and ultimately improving the experience of ordering food.

Finally, existing systems do not adequately address the problem of restaurant inventory management. Despite having inventory features, some platforms, like StoreHub and Eats365, are not completely integrated with the ordering process for food. This leads to order cancellations and wasted preparation time because menu items are still available for customers

even after the required ingredients are used up. By implementing a fully integrated inventory system that automatically updates the menu, the suggested solution fixes this problem.

CHAPTER 3: System methodology

This chapter explain the methodological approach used to design and implement the Food Ordering and Inventory Management System. The approach combines a prototype-driven development model with Scrum to validate important features early and to deliver incremental.

3.1 Rapid prototyping concept

Rapid prototyping means building a small working version quickly to test ideas and find problem early. The goal is to learn fast, fix issues while they are still small, and turn good trials into the final system.

Every cycle starts within a narrow, end to end slice. Basic database tables are created, a simple interface is built, and one operation is programmed to run from beginning to end such as presenting a product and saving an order. This keeps changes minimal and understandable.

Next, the cycle adds the parts with the most risk like allergy checks is matched to product ingredients, preparation time is estimated using a simple formula (base time plus kitchen load), stock levels are verified before checkout to prevent impossible orders, and hygiene images are shown only after approval

After each cycle and trial, quick feedback is gathered, mistakes are fixed, and temporary code is replaced with clear services and proper validation to ensure behaviour stays correct. When a trial meets its expected behaviours, the features become part of the products

3.2 Architecture (Laravel MVC)

The system uses a layered Laravel MVC design to separate concerns and keep business logic flexible. Browser clients render customer, restaurant owner and admin views while HTTP requests are routed to Laravel controllers and middleware for authentication and role verification. Controllers delegate to domain services such as allergy, ETA, inventory, and hygiene that encapsulate computations and rules, allowing for independent testing and reuse across endpoints.

Data storage is handled by Eloquent models, which are supported by a MySQL schema that matches the ERD like products link to ingredients via a pivot table to represent recipes, orders capture item quantities and status, chefs record availability for kitchen-load estimation, and restaurant owners store hygiene evidence with approval metadata. Queries are constructed to

be understandable and efficient, such as estimating per-product viable amounts from current stock and recipe needs or calculating active-chef counts to change estimated preparation time.

Static assets such as hygiene rating image are stored in the filesystem with database. Roles based access ensures that only approved materials are displayed correctly and that restaurant owners can view and manage only their own inventory and sales analytics.

3.3 Hardware & Software

Hardware

	Minimum	Recommended
Model	Any modern laptop/desktop	Any modern laptop/desktop
Processor	4-core 2.5 GHz	8-core 3.0 GHz
Operating system	Window 10/11, macOS 12+ or Ubuntu 22.04+	Window 11, macOS 13+ or Ubuntu 22.04 LTS
Memory	4GB ram	8GB ram
Storage	256GB SSD	512GB SSD

Table 3.1 Hardware Specifications

Software

Software	Purpose
Visual Studio Code	Used for writing and editing code for the food ordering system, providing syntax highlighting, debugging, and version control integration.
XAMPP Control Panel (Apache and MySQL)	Used for running a local server (Apache) and managing databases (MySQL) for development and testing of the food ordering system.
Laravel (PHP)	Used for server-side scripting to handle business logic, process orders, and interact with the database in the food ordering system.
Composer	Used for PHP dependency manager that installs Laravel packages for application classes.

HTML	Used for structuring the content and layout of the food ordering system's web pages.
CSS	Used for styling and designing the visual presentation of the food ordering system's web pages.

Table 3.2 Software Specifications

3.4 Scrum Methodology

The development of the Food Ordering and Inventory Management System adopts the Scrum methodology, an agile approach that emphasizes iterative progress, collaboration, and continuous feedback. Unlike traditional linear models, Scrum breaks the system into time-boxed development cycles called sprints, typically lasting 2–3 weeks. Each sprint delivers a working increment of the system that is reviewed, tested, and refined before moving to the next. This methodology providing transparency, adaptability, and early value delivery for this project, which ensuring that the system grows probably while meeting user needs.

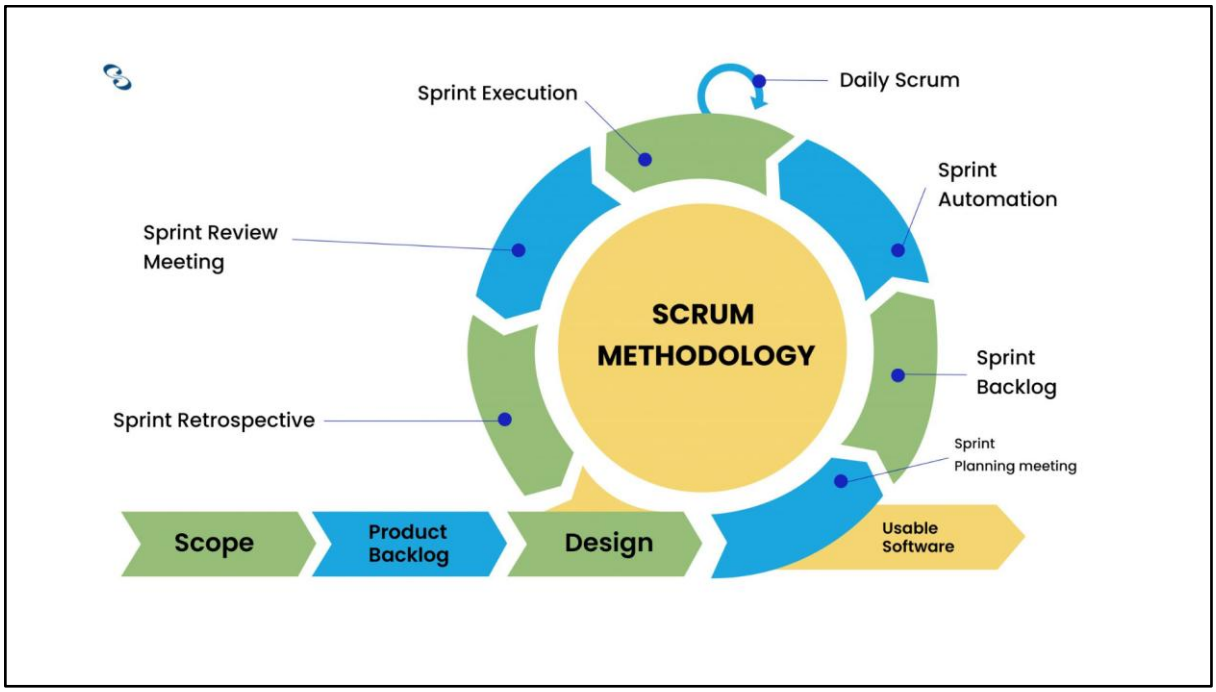


Figure 3. 1 Scrum Methodology Diagram

1. Product Backlog

To capture all of the platform's functional and non-functional requirements, a product backlog is established at the start of development. This backlog functions as a dynamic to-do list that is ranked according to the end users' value and importance. People like restaurant owners, chefs, kitchen managers, and patrons provide feedback on the backlog. Interviews, surveys, and observations of current processes are used to collect requirements.

Backlog items include features for restaurant owners like ingredient tracking, low-stock alerts, and hygiene ratings, as well as features for customers like menu browsing, allergen visibility, order placement, and preparation time estimates. Additional administrative needs include system monitoring and user management. The backlog is still adaptable and changes as the project progresses in response to comments and fresh information.

2. Sprint Planning (Design & Development Setup)

The top-priority items from the product backlog are chosen and added to the sprint backlog during sprint planning. Each sprint has a specific objective, such as creating the food ordering cart, turning on automatic inventory deduction, or implementing allergen detection and preparation time estimation. To make sure that every increment provides useful value, the sprint backlog is then divided into manageable tasks with clear acceptance criteria. The system design is kept modular to facilitate incremental delivery. Sprint 1 covers basic ordering and menu setup, while Sprint 2 adds inventory deduction, Sprint 3 allows management and restocking, and Sprint 4 incorporates more sophisticated features like allergen detection and preparation time estimates. Stability and testability are maintained while functionality is progressively added thanks to this structured approach.

3. Sprint Execution

To guarantee consistent progress toward the system's objectives, the development work is completed in brief, iterative steps during each sprint. The customary daily Scrum meeting is modified into a personal progress tracking habit because this project is being completed individually. This involves reviewing what was accomplished the day before, determining what needs to be done today, and highlighting any problems or difficulties that might prevent advancement. Consistently doing this makes it simpler to stay focused, prevent delays, and guarantee that planned activities are carried out without issues.

Each sprint's tasks are divided into small, realizable pieces so that significant progress can be made each day. For instance, the work is broken down into smaller tasks like creating the menu page, putting the cart in place, or connecting orders to the database rather than trying to construct the entire food ordering flow all at once. Early in the development cycle, completing these smaller tasks helps deliver value and builds momentum.

Each sprint concludes with a potentially shippable system increment that is ready for testing, demonstration, and improvement. In addition to ensuring that the system develops feature by feature, this iterative approach reduces the risk of integration issues, functional gaps, and technical difficulties. The system gradually develops into a comprehensive and dependable Food Ordering and Inventory Management platform as each sprint builds on the one before it.

4. Sprint Review

To illustrate and assess the finished increment, a sprint review is held at the conclusion of each sprint. This stage guarantees that the features created during the sprint are evaluated for usability and alignment with stakeholder needs in addition to technical aspects. A variety of testing levels are used, including performance testing to mimic peak-hour demand, integration testing to verify seamless interactions between modules like order placement and stock updates, and unit testing to validate computations like order pricing or ingredient deductions. Furthermore, User Acceptance Testing (UAT) is carried out to collect useful input, such as chefs confirming that kitchen alerts are sent on time or restaurant owners confirming that low-stock alerts function properly. This process offers a chance to modify the product backlog for upcoming development and confirms that the system increment offers genuine value.

5. Sprint Retrospective

A sprint retrospective is held to consider the development process itself following the sprint review. This reflection turns into a self-evaluation exercise in a solo project, where accomplishments, difficulties, and opportunities for development are meticulously recorded. For instance, it would be successful to duplicate in subsequent work if the allergen detection feature was successfully integrated. If stock-tracking queries revealed performance problems, they are noted as difficult to fix. Additionally, the retrospective finds enhancements to continue, like streamlining the database structure prior to the subsequent sprint. Continuous evaluation guarantees that every sprint not only improves the system but also the approach to development, which eventually results in increased productivity and better quality outputs.

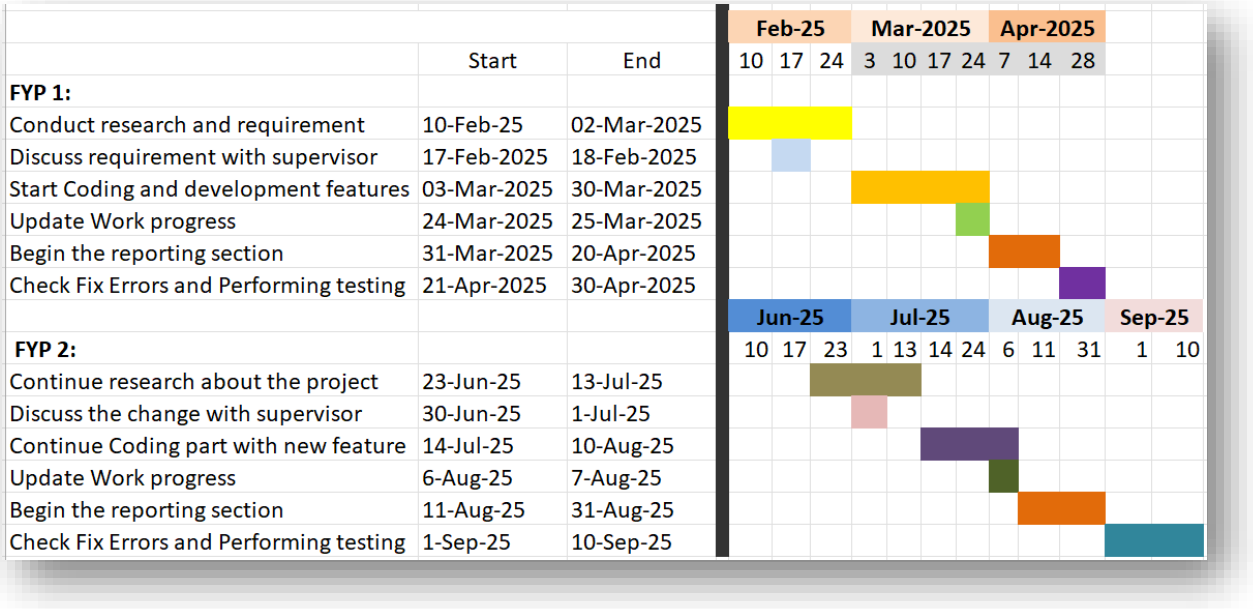
6. Increment

An increment, a working component of the system, is delivered every sprint. Before broader rollouts, these increments are first implemented in small pilot tests. While later increments handle complex workflows like restaurant owner restocking and allergy warnings, early increments will be tested with simple ordering scenarios. By the end of the sprint, every increment has been integrated into a comprehensive system that includes admin management, food ordering, management of inventory, allergy checks, and hygiene ratings. This guarantees that the project will gradually move closer to the complete solution while offering stakeholders early value.

7. Summary

Scrum is used to develop the Food Ordering System in iterative sprints that gradually produce functional features. Continuous stakeholder involvement guarantees early validation and system improvement for features like allergen information, preparation time estimates, and stock tracking. Scrum's flexibility enables the project to manage shifting needs, lower risks, and continue making steady progress. In the end, this strategy produces a dependable, user focused solution that raises customer satisfaction, ingredients management, and restaurant efficiency.

3.5 Project Timeline



	Start	End
FYP 1:		
Conduct research and requirement	10-Feb-25	02-Mar-2025
Discuss requirement with supervisor	17-Feb-2025	18-Feb-2025
Start Coding and development features	03-Mar-2025	30-Mar-2025
Update Work progress	24-Mar-2025	25-Mar-2025
Begin the reporting section	31-Mar-2025	20-Apr-2025
Check Fix Errors and Performing testing	21-Apr-2025	30-Apr-2025
FYP 2:		
Continue research about the project	23-Jun-25	13-Jul-25
Discuss the change with supervisor	30-Jun-25	1-Jul-25
Continue Coding part with new feature	14-Jul-25	10-Aug-25
Update Work progress	6-Aug-25	7-Aug-25
Begin the reporting section	11-Aug-25	31-Aug-25
Check Fix Errors and Performing testing	1-Sep-25	10-Sep-25

Figure 3. 2 Gantt Chart of the Project Progress

CHAPTER 4: System Design

4.1 Use Case Diagram

A use case diagram is a type of diagram in the UML which call as Unified Modelling Language that shows how users (called "actors") interact with a system.

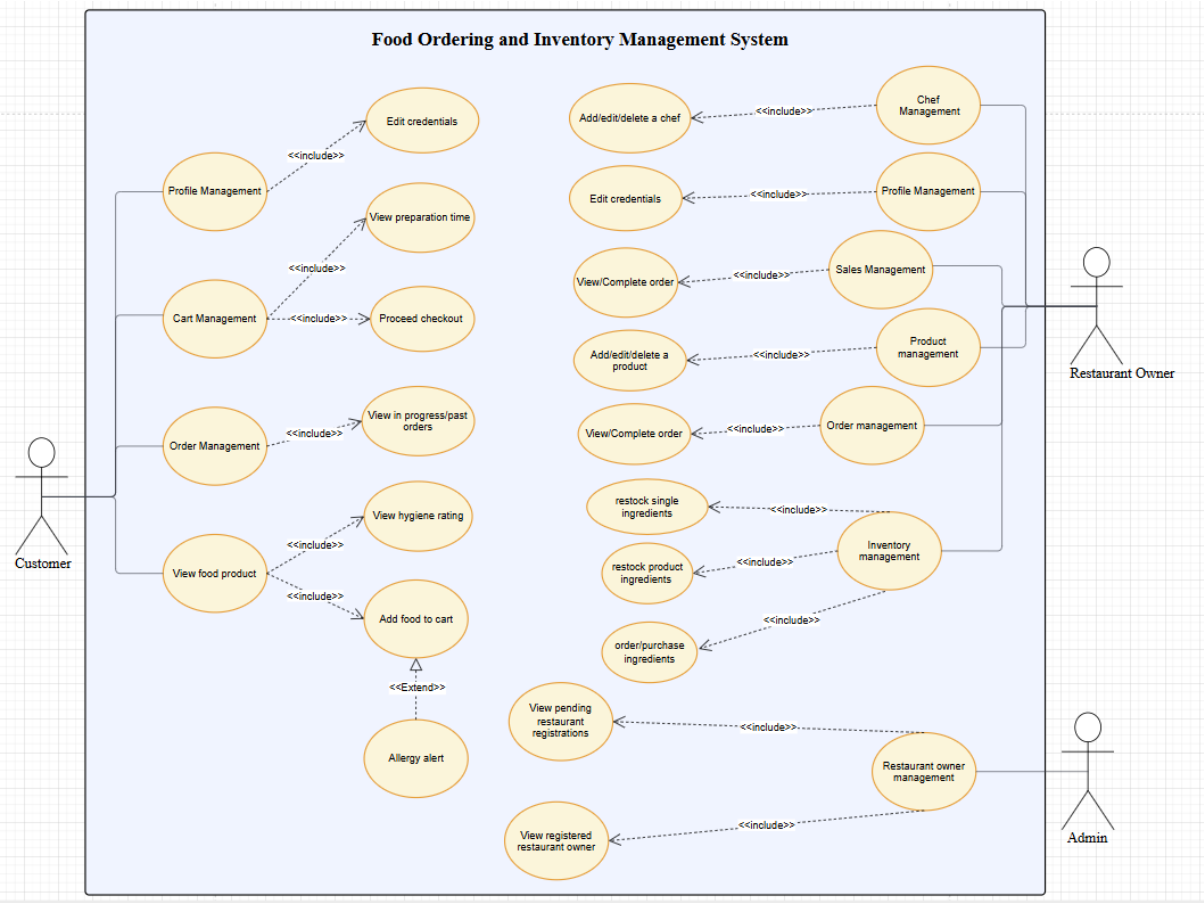


Figure 4. 1 Overview Use Case Diagram of Food Ordering and Inventory Management System

Figure 4.1 shows the overall operation of the Food Ordering System and the three primary users in this system are the customer, the restaurant owner, and the admin.

Customers can register, log in, and manage their accounts through the Profile Management module. Within the Cart Management module, users could add, edit, and remove products from their cart. The Order Management module allows customers to view both previous and pending orders. To help users make informed decisions before placing an order, the system includes a Hygiene Rating feature, which displays each restaurant's cleanliness standards. Additionally, the Allergy Alert feature notifies users if a selected product contains any known allergens (the alert is triggered when adding items to the cart). Customers can also browse and view food products with details. The system also shows preparation time so users can anticipate when their orders will be ready, helping users plan their time more effectively.

For the second actor, the restaurant owner, the system provides several modules to support daily operations. Through the Account Management module, restaurant owners can register, log in, and manage their profiles (including editing credentials). The Chef Management module allows owners to add, edit, or delete chef accounts. The Sales Management module enables them to monitor both historical and current orders and to view/complete orders for reconciliation. Using the Product Management module, restaurant owners can add new products, as well as edit or delete existing ones. The Food Preparation Time feature allows them to set estimated preparation times for each product, ensuring accurate delivery expectations. In the Ingredients Management module, restaurant owners can define the required ingredients for each product. The Inventory Management system tracks ingredient levels and notifies the restaurant owner when items are running low, either based on individual ingredients or by product requirements. Owners can restock single ingredients, restock the full set of ingredients required by a product, and create purchase orders for ingredients directly in the system.

Finally, the admin plays a supervisory role within the system. Through the Restaurant Owner Management module, the admin has the authority to approve or reject restaurant owner registrations, ensuring only verified businesses are allowed on the platform. Admins can also view pending restaurant registrations and review the list of registered restaurant owners for ongoing oversight

4.2 Class Diagram

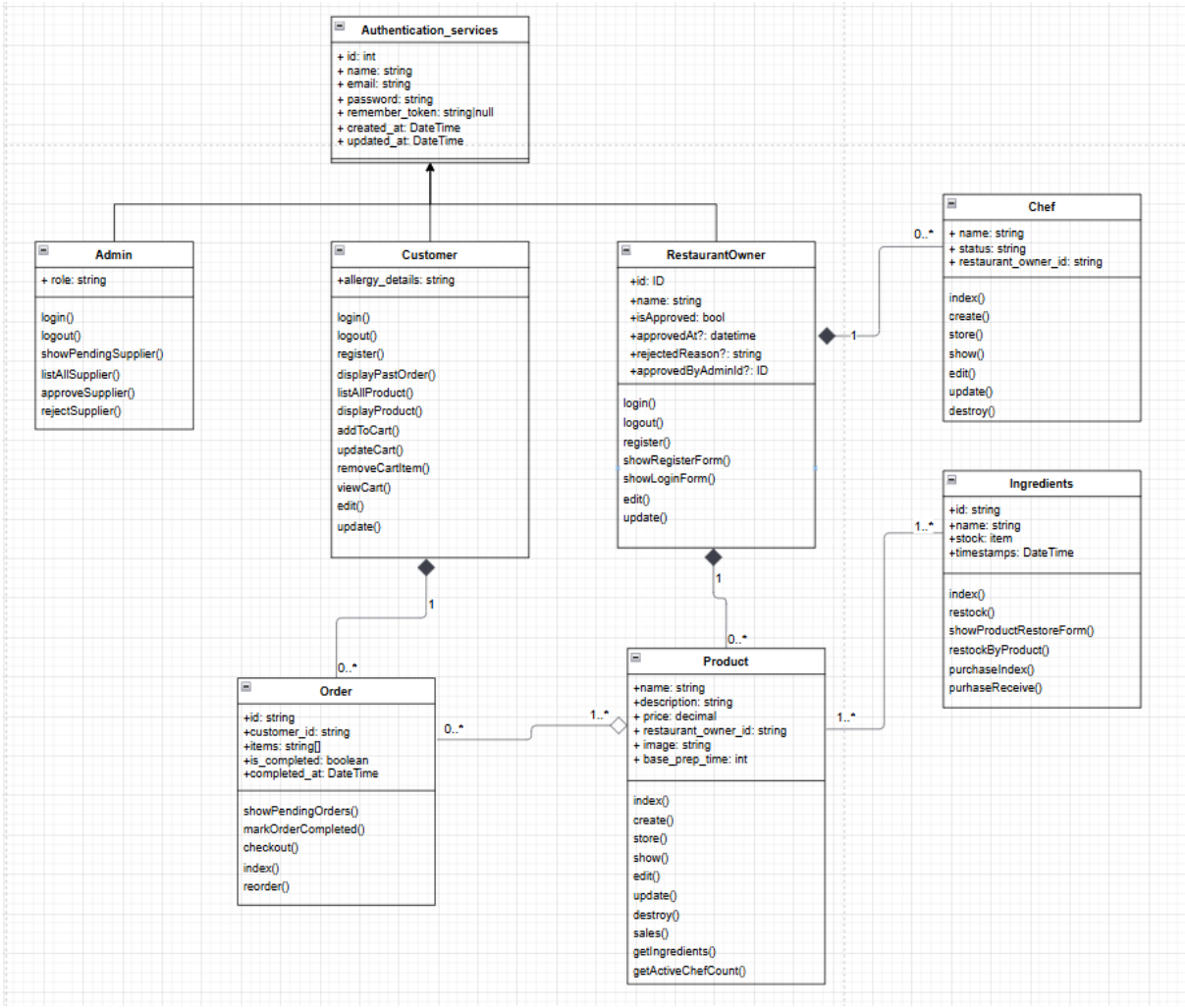


Figure 4. 2 Class Diagram of Food Ordering and Inventory Management System

1. Authenticatable Class:
    a. Attributes
        i. id: A unique identifier for each object
        ii. name: The name of each object
        iii. email: The email of each object
        iv. password: The password for each object to login
        v. created_at: The date and time at which the object is created
        vi. updated_at: The date and time at which the object is last updated
    b. Relationship
        i. Superclass for admin, customer and restaurant owner
2. Admin class

a. Attributes
    i. role: defines the role and security clearance it can get
b. Methods
    i. login(): Allows admin to login
    ii. logout(): Allows admin to logout
    iii. showPendingSuppliers(): Display all the restaurant owner that are pending for approval
    iv. listAllSuppliers(): Display all the restaurant owners
    v. approveSupplier(): Approve the registration request of restaurant owner
    vi. rejectSupplier(): Rejects the registration request of restaurant owner
c. Relationship
    i. sub-class of authenticatable super class, inheriting all attributes.

3. Customer class
    a. Attributes
        i. allergy_details: to note down what customer is allergic too
    b. Methods
        i. login(): Allows customer to login
        ii. logout(): Allows customer to logout
        iii. register(): Allows customer to register an account
        iv. checkout(): Allows customer to place order on all the product in cart
        v. displayPastOrder(): Display all the past order customer have
        vi. reorder(): Allows customer to place the same order as last time
        vii. listAllProduct(): Display all the products from each restaurant owner
        viii. displayProduct(): Display a product
        ix. addToCart(): Add a product to the cart
        x. updateCart(): Update changes of a product in cart
        xi. removeCartItem(): delete a product in the cart
        xii. viewCart(): Display all product in cart
        xiii. getActiveChefCount(): Display the current working chef in the restaurant (read from owner/chef status)
    c. Relationship
        i. sub-class of authenticatable super class, inheriting all attributes.
        ii. A Customer can have 0 or many Order (composition: an Order cannot exist without its Customer)

4. RestaurantOwner
    a. Attributes
        i. hygiene_rating_image: an image path to the image
        ii. is_approved: whether the owner account is approved by admin
        iii. approved_at: the timestamp when the account is approved
    b. Methods
        i. login(): Allows restaurant owner to login
        ii. logout(): Allows restaurant owner to logout
        iii. register(): Allows restaurant owner to register
        iv. edit(): Show profile edit form
        v. update(): Save profile updates
    c. Relationship
        i. RestaurantOwner can have 0 or many Chef (composition)
        ii. RestaurantOwner can have 0 or many Product (composition)
5. Order
    a. Attributes
        i. Id: The unique identifier for Order
        ii. customer_id: The customer id that places the order
        iii. items: An array of product
        iv. is_completed: The status of order if it is incomplete or complete
        v. completed_at: The timestamp at which the order is completed at
    b. Methods
        i. showPendingOrders(): Display all the order that are still pending
        ii. markOrderCompleted(): To Update the status of order to completed
        iii. checkout(): place order on the cart
        iv. index(): Display the product in cart
        v. reorder(): To place the same as previous order
    c. Relationship
        i. Order can have only 1 and only 1 Customer and it is composite, Order cannot exist without a Customer.
        ii. Order can have 1 or many product and it is aggregation, Order can exist even product is deleted
6. Product
    a. Attributes

i. name: Name of the product
ii. description: The description of the product
iii. price: The price of the product
iv. restaurant_owner_id: The owner of the product
v. image: the image path of the product
vi. base_prep_time: The time it requires to prepare the product

b. Methods
i. index(): Display all the product for the logged in restaurant owner
ii. create(): Fetches all the ingredients related to the restaurant owner
iii. store(): Create the new product
iv. edit(): Display the field for the product to be edited
v. update(): Update and save the product
vi. destroy(): To delete the product
vii. sales(): To display sales by product
viii. getIngredients(): Display ingredients based on the product
ix. getActiveChefCount(): Display the current working chef for the owner (helper for preparation ETA)

c. Relationship
i. Product can have 0 or many Order
ii. Product can have 1 and only 1 RestaurantOwner and is composite, Product cannot exist without RestaurantOwner
iii. Product can have 1 or many Ingredient and Ingredient can belong to 0 or many Product

7. Ingredient
a. Attributes
i. name: the name of the ingredient
ii. stock: the number of stock ingredient has

b. Methods
i. index(): Displays a list of ingredients associated with the restaurant owner products (through the formula).
ii. restock(): Adds stock to an individual ingredient, but only if it belongs to one of the restaurant owner's products.
iii. showProductRestockForm(): Prepares data for a view where the restaurant ownercan restock ingredients by product formula.

iv. restockByProduct(): Adds stock to all ingredients of a product based on its formula.
v. purchaseIndex(): Lists ingredient purchase orders
vi. purchaseCreate(): Shows the form to create a new ingredient purchase
vii. purchaseStore(): Saves a new purchase order
viii. purchaseShow(): Displays a specific purchase order
ix. purchaseReceive(): Marks a purchase as received and updates stock/lead time

c. Relationship:
   i. Ingredient can have 0 or many product

8. Chef
   a. Attributes
      i. name: name of the chef
      ii. status: the status of the current chef
      iii. restaurant_owner_id: the restaurant owner which hires the chef
   b. Methods
      i. index(): Lists all chefs that belong to the currently logged-in restaurant owner.
      ii. create(): Shows the form to add a new chef.
      iii. store(): Saves a new chef submitted from the create form.
      iv. show(): Shows the edit form for a specific chef.
      v. update(): Updates the name and status of an existing chef.
      vi. destroy(): Deletes a chef (only if owned by the restaurant owner).
   c. Relationship
      i. Chef can have 1 and only 1 restaurant owner and is composite, chef cannot exist without restaurant owner.

4.3 Activity Diagram

4.3.1 Customer Activity Diagram

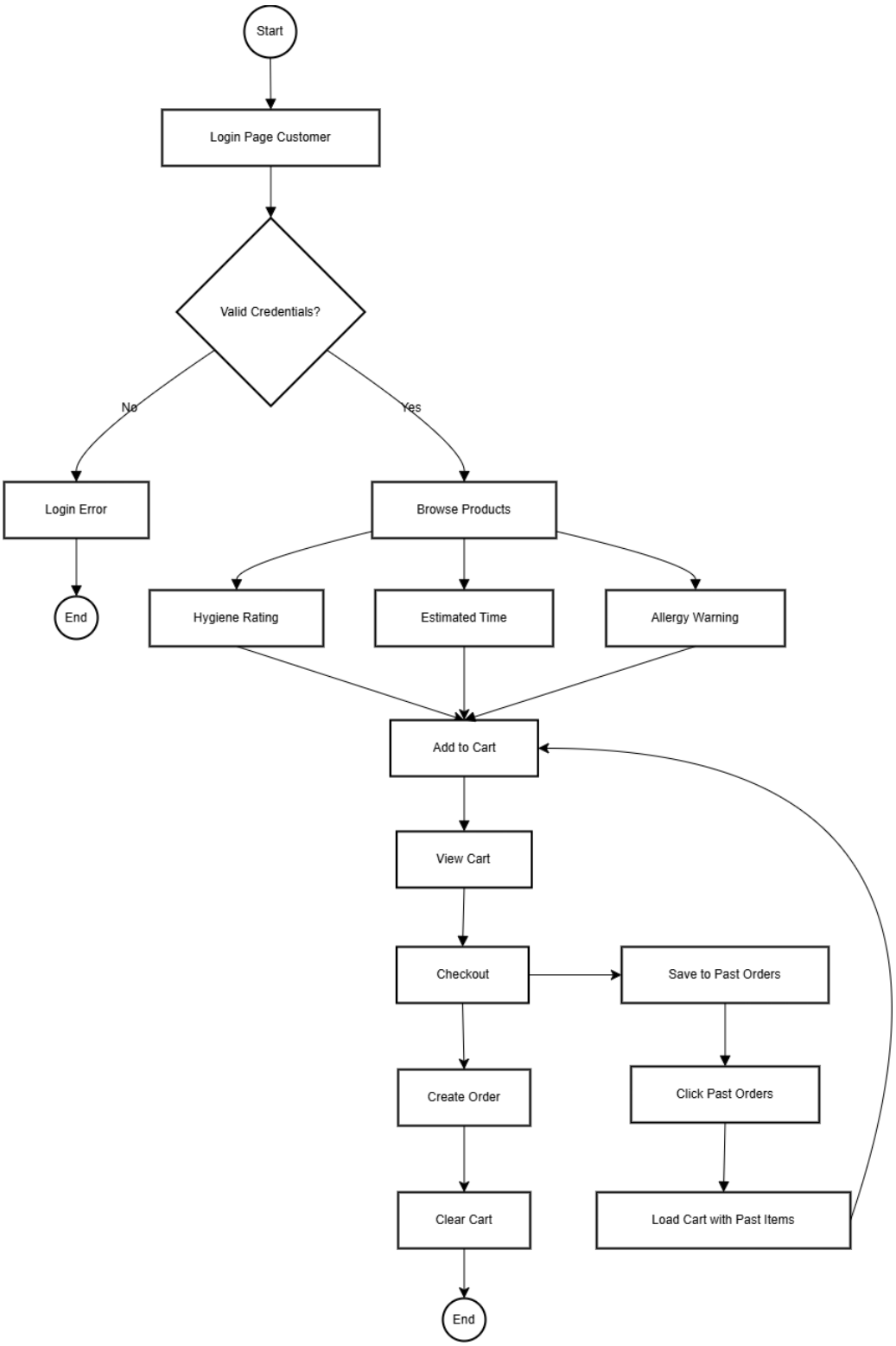


Figure 4. 3 Customer Activity Diagram

The customer activity diagram (Figure 4.3) begins when the customer logs into their account and browses the available products. After selecting the desired item and adding it to the shopping cart, the system performs a rigorous security check to look for any registered food allergies. If an allergy exists, the system automatically cross-checks the contents of the product with the customer's known allergens. So, when conflicting information about potential allergens is found, the system can quickly display a warning notice prompting the customer to confirm before placing an order. Exactly, this is important for safety checkpoints which help prevent accidental exposure to allergens.

If there are no allergy concerns, the process moves smoothly to the checkout stage. The estimated preparation time and the associated hygiene rating are two crucial pieces of information that the system computes and provides to guarantee total transparency. Customers can easily make intelligent choices about their orders because this information is clearly shown during the checkout process. While hygiene ratings guarantee food safety, estimated lead times assist in handling customer requests so they can decide for themselves whether it is appropriate to order this food right away.

In the end, before completing their order submission, customers can receive a comprehensive order summary that includes all of the foods they have chosen, any applicable allergy warnings, estimated lead times, and hygiene instructions. Last-minute validations and necessary modifications are supported by this integrated view. The system only processes the entire order once all the information has been confirmed to be accurate, guaranteeing that all necessary safety checks and disclosures have been completed. The complete process shows how stringent food safety regulations and operational openness can be deftly incorporated into a quick, user-friendly ordering procedure.

4.3.2 Restaurant Owner Activity Diagram

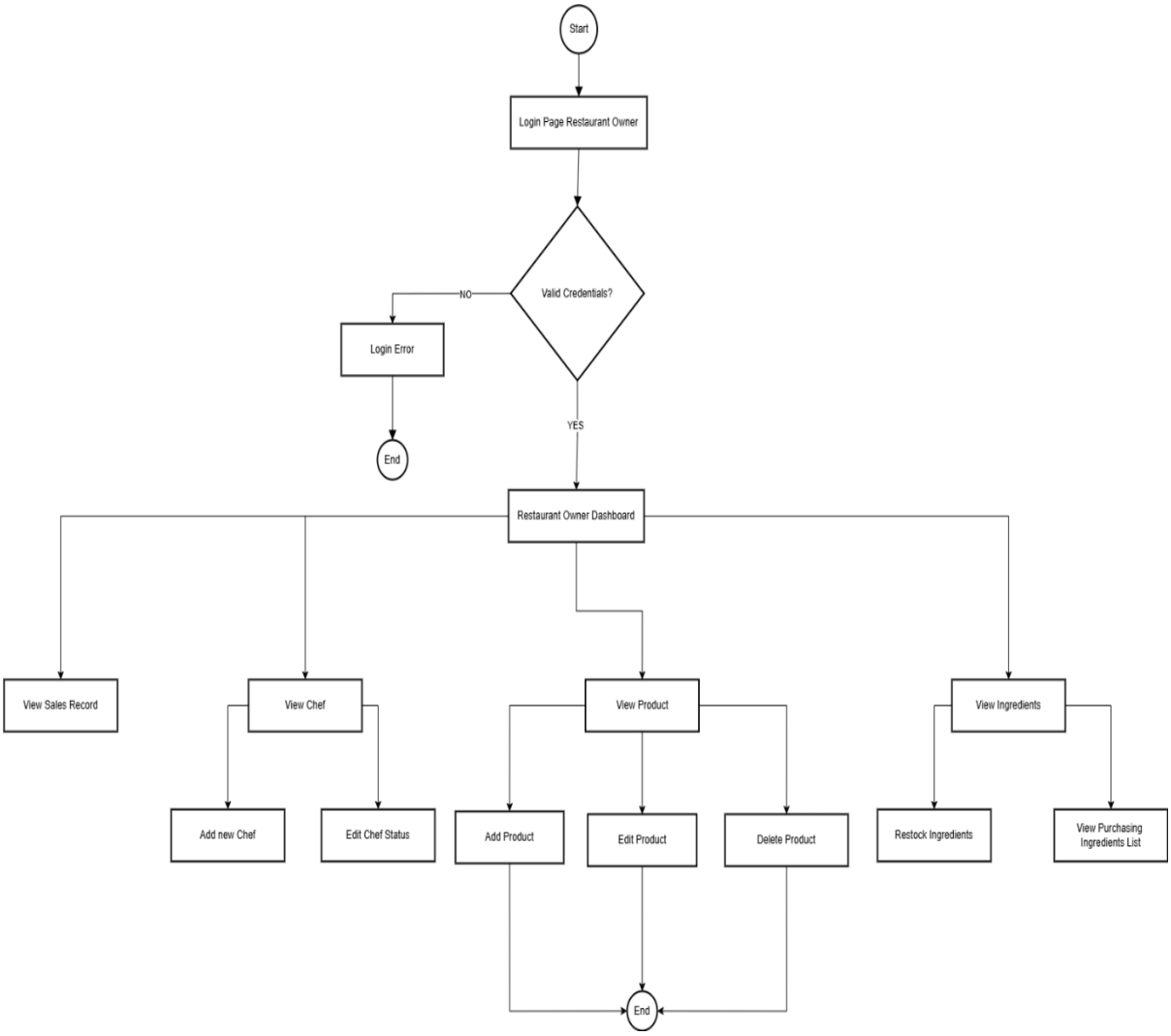


Figure 4. 4 Restaurant Owner Activity Diagram

The system's error actions and login process workflow are depicted in the restaurant owner activity diagram (Figure 4.4). The restaurant owner accesses the login page to start the process, and the system then confirms the credentials they entered. A login error message is shown and the procedure is terminated if the credentials are invalid. If the credentials are valid, the restaurant owner is granted access to the dashboard, where several management functions are available. From the dashboard, the owner can view sales records, manage chefs by adding new chefs or editing their status, and manage products by adding, editing, or deleting items. Additionally, the owner can view and manage ingredients, including restocking supplies or reviewing the purchasing ingredients list. This diagram totally highlights the decision making process at the login stage and stated the structured set of activities that what a restaurant owner can perform upon successful authentication.

4.3.3 Admin Activity Diagram

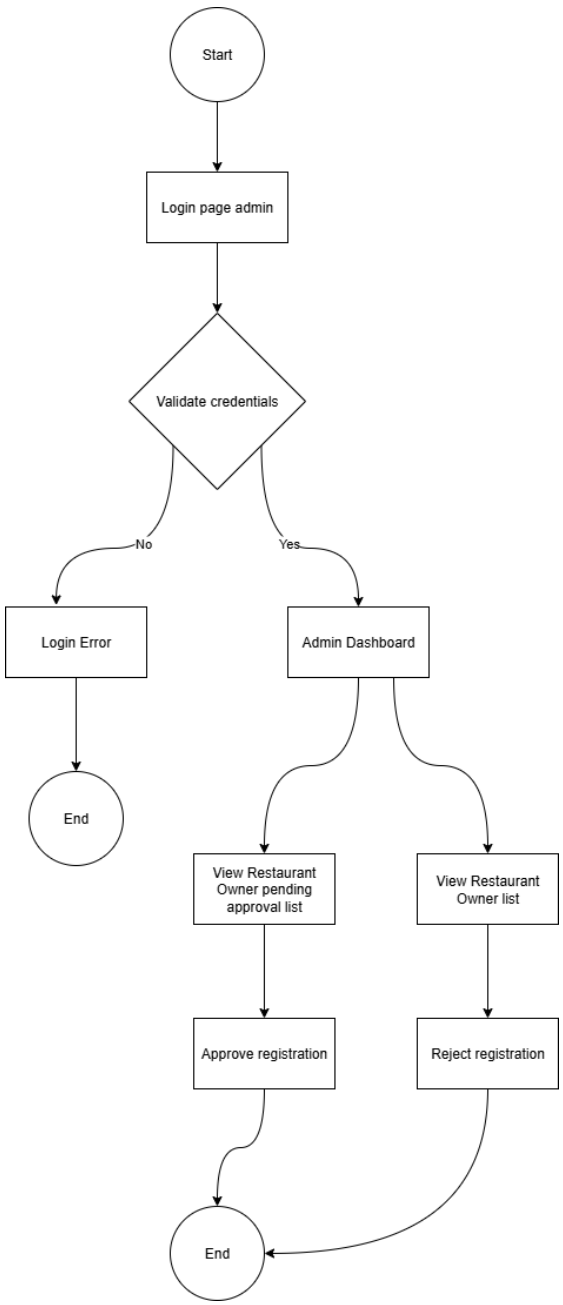


Figure 4. 5 Admin Activity Diagram

The activity begins at the Start point, where the administrator is about to log into the system. The first action the system presents is the admin login page, where the administrator is required to enter their login credentials. Once the credentials are submitted, the system performs a check to validate the credentials.

At this decision point, there are two possible outcomes. If the credentials are invalid, the system displays a login error message. In this case, the process immediately terminates, as the administrator is unable to proceed further. On the other hand, if the credentials are valid, the administrator successfully logs in and is directed to the Admin Dashboard.

From the Admin Dashboard, the administrator has two main options. The first option is to view the Restaurant Owner pending approval list, which contains new registration requests from restaurant owners who are waiting for approval. After reviewing this list, the administrator may choose to approve a registration, and once this action is completed, the process comes to an end.

The second option available on the Admin Dashboard is to view the Restaurant Owner list, which displays all existing registered owners. From this list, the administrator has the ability to reject a registration, possibly meaning to deny a pending owner or revoke access for an existing owner. Once the rejection is performed, the process also comes to an end.

4.4 Entity Relationship Diagram (ERD)

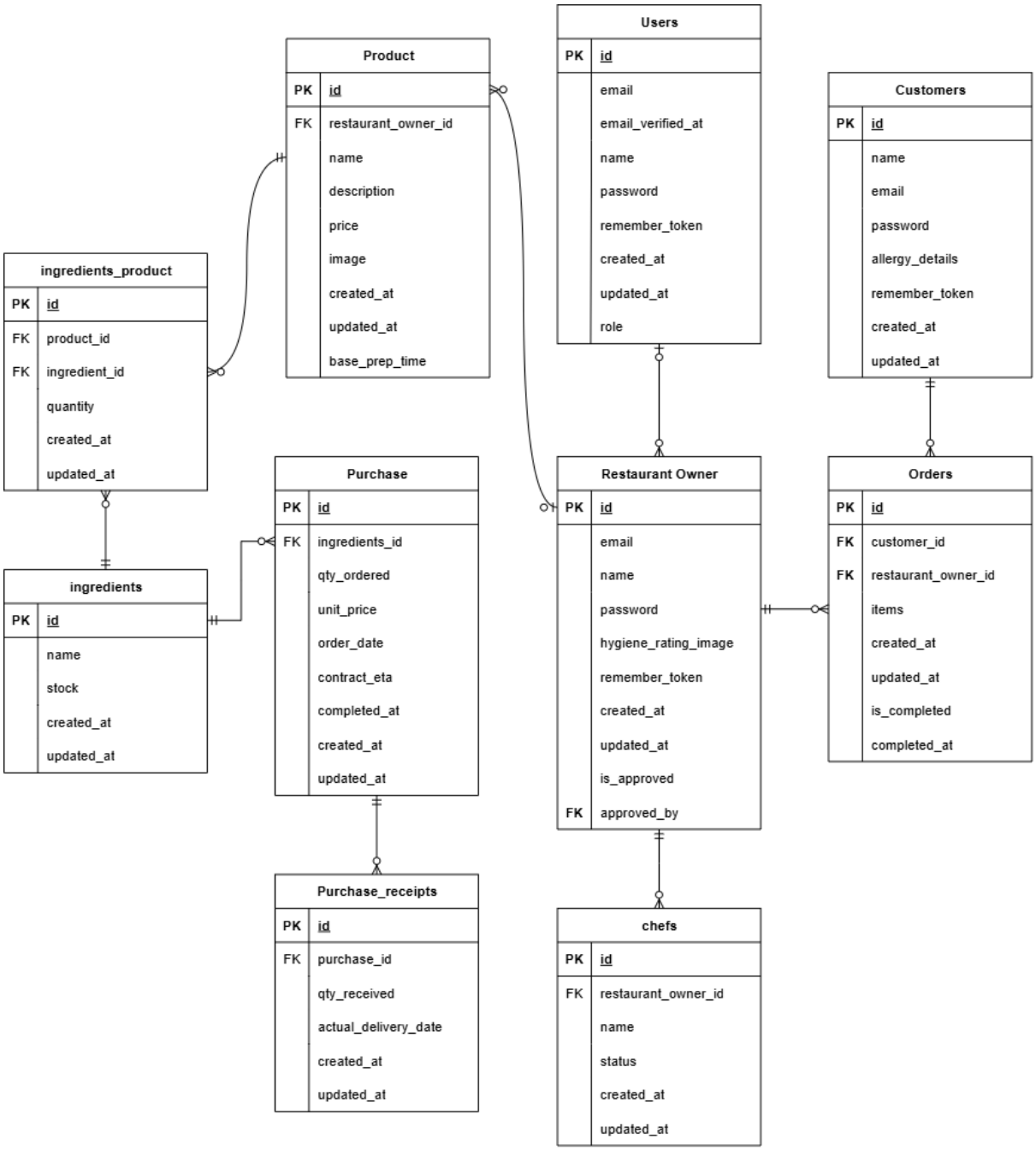


Figure 4. 6 ERD Diagram of Food Ordering and Inventory Management System

1. Restaurant Owner Table

The Restaurant Owner table represents vendors who provide products and manage operations like chefs and inventory. Each owner has a unique identity and can upload their hygiene rating image, along with credentials like name, email, and password. The table also stores an approval flag and the admin who approved the account. A restaurant owner has one-to-many relationships with the products, chefs, and orders tables. This table is central to the system and connects with multiple others.

2. Products Table

The products table stores details about items that a restaurant owner offers, including the product's name, description, price, image, and preparation time. Each product belongs to one restaurant owner, as indicated by the restaurant_owner_id foreign key. Additionally, products are connected to multiple ingredients through a many-to-many relationship, managed by the ingredients_product pivot table. This allows each product to have its own formula or recipe composed of various ingredients and their quantities.

3. Ingredients Table

The ingredients table keeps track of all the individual ingredients used across products. Each ingredient has a name and a current stock level. Ingredients participate in a many-to-many relationship with products through the ingredients_product pivot table. This design enables an ingredient to be shared across multiple products, each potentially using a different quantity. It's essential for inventory and recipe management.

4. Ingredients_product Table (Pivot Table)

The ingredients_product table acts as a junction table that connects products and ingredients. It records not only the relationship between a product and its ingredients but also includes an additional field: quantity, which specifies how much of each ingredient is required for the product. This structure supports product formulas, enabling dynamic and precise inventory management. It includes created_at and updated_at fields for tracking changes over time.

5. Chefs Table

The chefs table stores information about the chefs working under each restaurant owner. Each chef has a name, a status (such as “Active” or “On Leave”), and a foreign key restaurant_owner_id linking them to the owner they work for. This creates a one-to-many relationship where each restaurant owner can have multiple chefs. This table is useful for tracking personnel responsible for preparing products.

6. Orders Table

The orders table tracks customer orders, including the items ordered, their completion status, and timestamps. Each order has two foreign keys: customer_id and restaurant_owner_id, creating many-to-one relationships with both the customers and restaurant owner tables. This means each order is placed by one customer and fulfilled by one restaurant owner. The items field stores details of the purchased products. The is_completed and completed_at fields help in order tracking and status management.

7. Customers Table

The customers table stores personal information for customers, including their name, email, password, and any allergy_details. A customer can place multiple orders, forming a one-to-many relationship with the orders table. This allows customers to manage their order history and enables the system to personalize product offerings based on allergy information.

8. Users Table

The users table stores information for system administrators (and roles). It includes login credentials and audit fields. The table has a one-to-many relationship with Restaurant Owner via the approved_by foreign key on the owner side, which records which admin approved an owner’s registration.

9. Purchase Table

The purchase table records procurement of individual ingredients. Each record references a single ingredient (ingredients_id) and captures qty_ordered, unit_price, order_date, contract_eta, and completed_at along with timestamps. This supports tracking planned vs. actual lead times and costs for restocking.

10. Purchase_receipts Table

The purchase_receipts table logs receipts against a specific purchase (via purchase_id). It records qty_received and actual_delivery_date with timestamps. Multiple receipts can be tied to one purchase to support partial deliveries and accurate stock updates.

4.5 Data Dictionary

4.5.1 Chef table

PK/FK	Field Name	Description	Data Type	Field Size
PK	id	Unique id for chef	bigint	20
FK	restaurant_owner_id	An unique id for restaurant_owner	bigint	20
	name	Name of the chef	varchar	255
	status	The status of the chef	enum	Active, On Leave
	created _at	The date and time chef is created at	timestamp	NA
	updated_at	The data and time chef is last updated	timestamp	NA

Table 4. 1Chef Data Dictionary

4.5.2 Customers table

PK/FK	Field Name	Description	Data Type	Field Size
PK	id	Unique id for customer	bigint	20
	name	Name of the customer	varchar	255
	email	The unique email for customer	varchar	255

	password	The password for customer to login	varchar	255
	allergy_details	The allergen customers are allergic to	text	255
	created _at	The date and time customer is created at	timestamp	NA
	updated_at	The data and time customer is last updated	timestamp	NA

Table 4. 2 Customer Data Dictionary

PK/FK	Field Name	Description	Data Type	Field Size
PK	id	Unique id for ingredients	bigint	20
	name	Name of the ingredients	varchar	255
	stock	The number of stock an ingredient has	int	10
	created _at	The date and time ingredients is created at	timestamp	NA
	updated_at	The data and time ingredients is last updated	timestamp	NA

Table 4. 3 Ingredients Data Dictionary

PK/FK	Field Name	Description	Data Type	Field Size
PK	id	Unique id for ingredient_product	bigint	20
FK	product_id	The unique id for product	bigint	20
FK	ingredient_id	The unique id for ingredient_id	bigint	20
	quantity	The number of ingredient use	int	10
	created _at	The date and time ingredient_product is created at	timestamp	NA
	updated_at	The data and time ingredient_product is last updated	timestamp	NA

Table 4. 4 Ingredients product Data Dictionary

PK/FK	Field Name	Description	Data Type	Field Size
PK	id	Unique id for orders	bigint	20
FK	customer_id	The unique id for customer	bigint	20
FK	restaurant_owner_id	The unique id for restaurant_owner	bigint	20
	items	The products that are added	longtext	255
	created _at	The date and time orders is created at	timestamp	NA
	updated_at	The data and time orders is last updated	timestamp	NA
	is_completed	The status of order if it is still pending or completed	tinyint	1
	completed_at	The time which the order is completed at	timestamp	NA

Table 4. 5 Orders Data Dictionary

PK/FK	Field Name	Description	Data Type	Field Size
PK	id	Unique id for products	bigint	20
FK	restaurant_owner_id	The unique id for restaurant_owner	bigint	20
	name	The name for the product	varchar	255
	description	The description of the product	text	255
	price	The price of the product	decimal	(10,2)
	image	The path of the image	varchar	255
	created _at	The date and time products is created at	timestamp	NA
	updated_at	The data and time products is last updated	timestamp	NA
	base_prep_time	The estimate time to complete the product	int	11

Table 4. 6 Products Data Dictionary

PK/FK	Field Name	Description	Data Type	Field Size
PK	id	Unique id for restaurant_owners	bigint	20
	name	Name of the restaurant_owners	varchar	255
	email	The unique email for restaurant_owners	varchar	255
	password	The password for restaurant_owners to login	varchar	255
	Hygiene_rating_image	The path of the image of cleanliness rating	varchar	255
	created_at	The date and time restaurant_owner is created at	timestamp	NA
	updated_at	The data and time restaurant_owner is last updated	timestamp	NA
	is_approved	The status of restaurant_owner if	tinyint	1

		it is approved		
FK	approved_by	Unique id for users	bigint	20

Table 4. 7 Restaurant Owner Data Dictionary

PK/FK	Field Name	Description	Data Type	Field Size
PK	id	Unique id for user	bigint	20
	name	Name of the user	varchar	255
	email	The unique email for user	varchar	255
	password	The password for user to login	varchar	255
	created_at	The date and time users is created at	timestamp	NA
	updated_at	The data and time users is last updated	timestamp	NA
	is_approved	The status of users if it is approved	tinyint	1
	role	The role of the user	varchar	255

Table 4. 8 Users Data Dictionary

Chapter 5: System Implementation

5.1 Customer Side

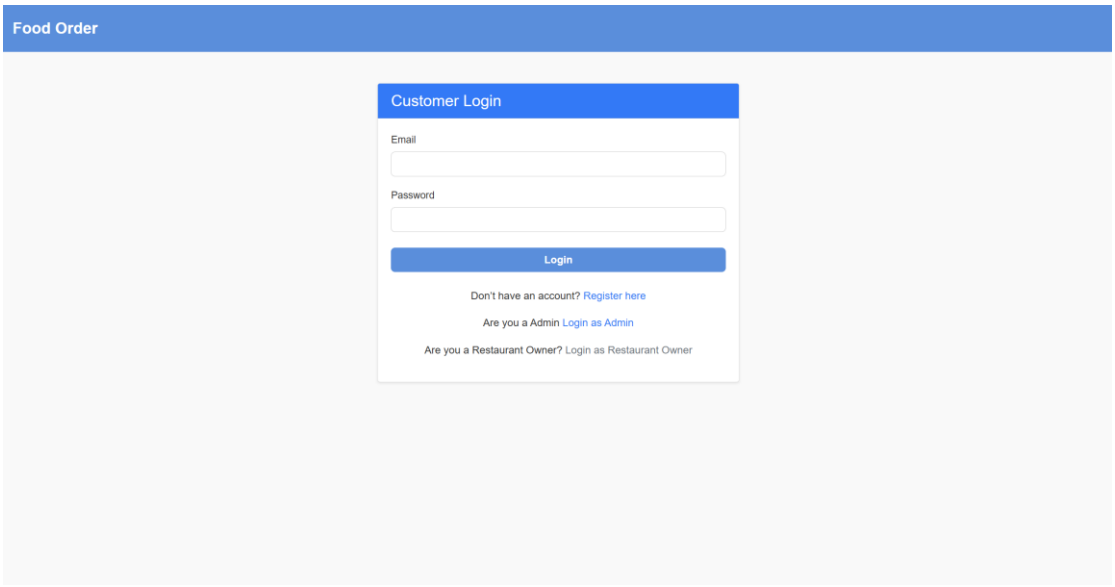


Figure 5. 1 Customer Login Page

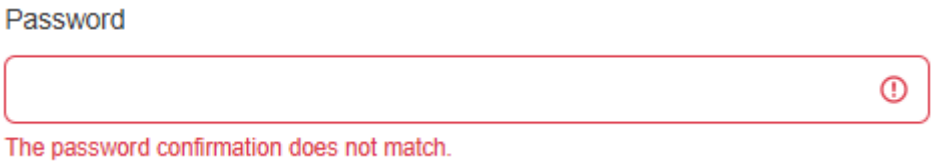


Figure 5. 2 Password error detect

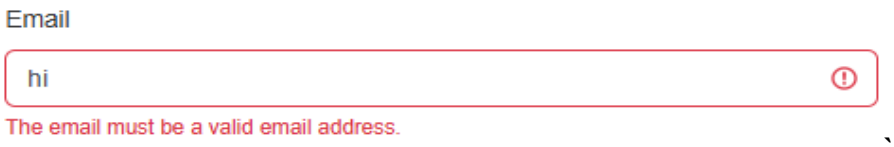


Figure 5. 3 Email valid detect

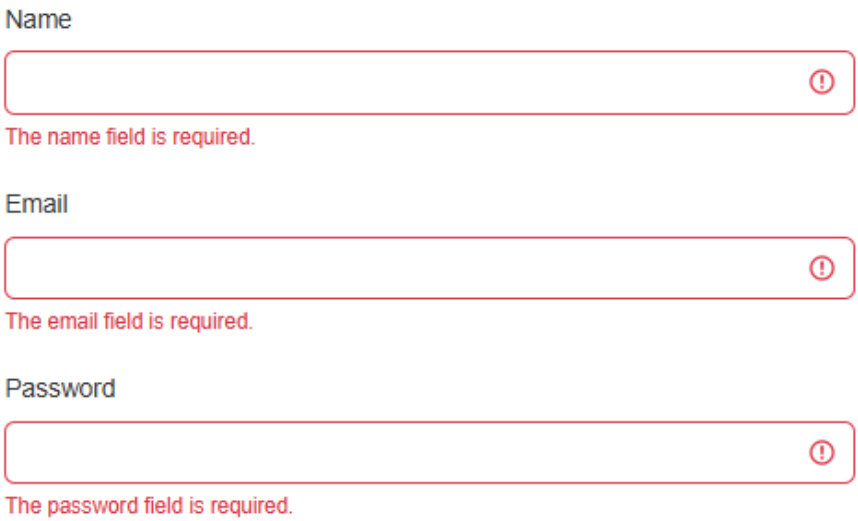


Figure 5. 4 Empty detect

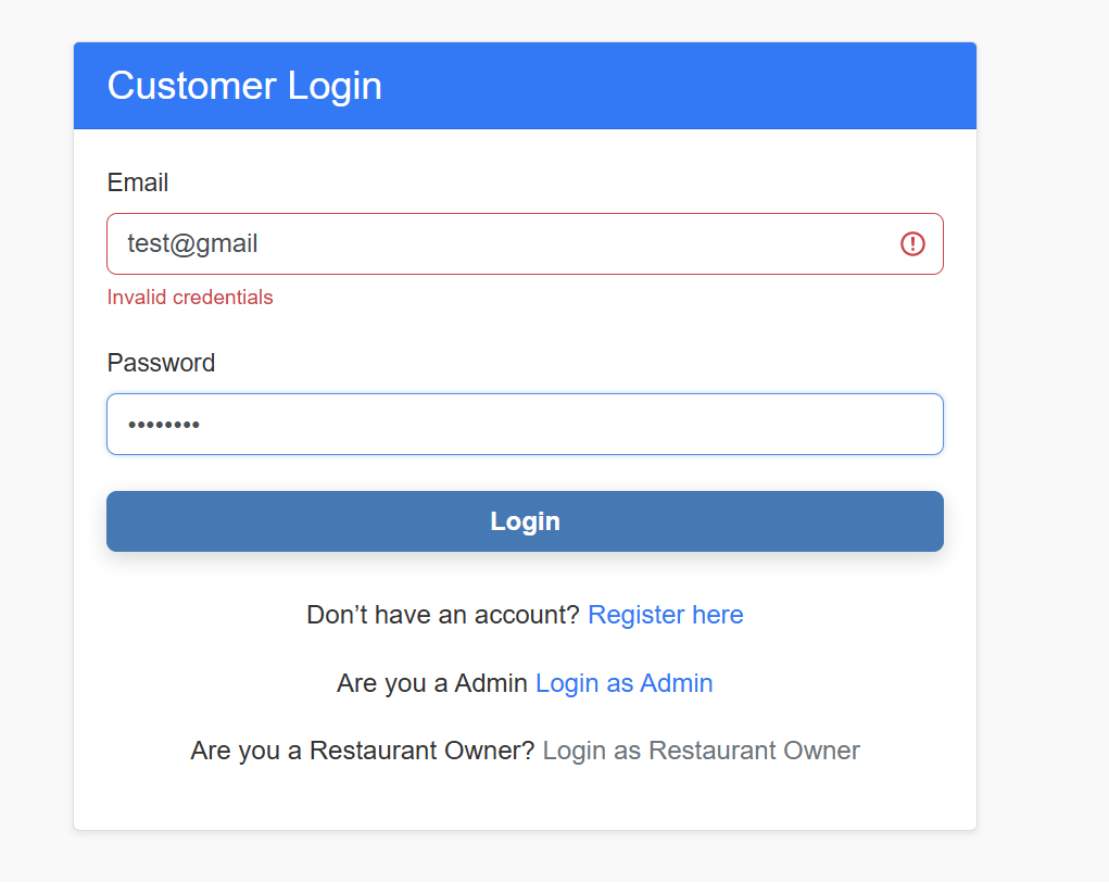


Figure 5. 5 Invalid login detect

Figure 5.1 shows the customer login page with fields for email and password. In Figure 5.2, an error appears when the password confirmation does not match, indicating that the user must correct it. Figure 5.3 shows an email validation error, where the system prompts the user that the email must be a valid email address. Figure 5.4 illustrates the case where the name, email, and password fields are empty, and the system requires these fields to be filled in before submission. Finally, Figure 5.5 shows an invalid login detection, where the system notifies the user of incorrect login details, prompting them to re-enter their credentials.

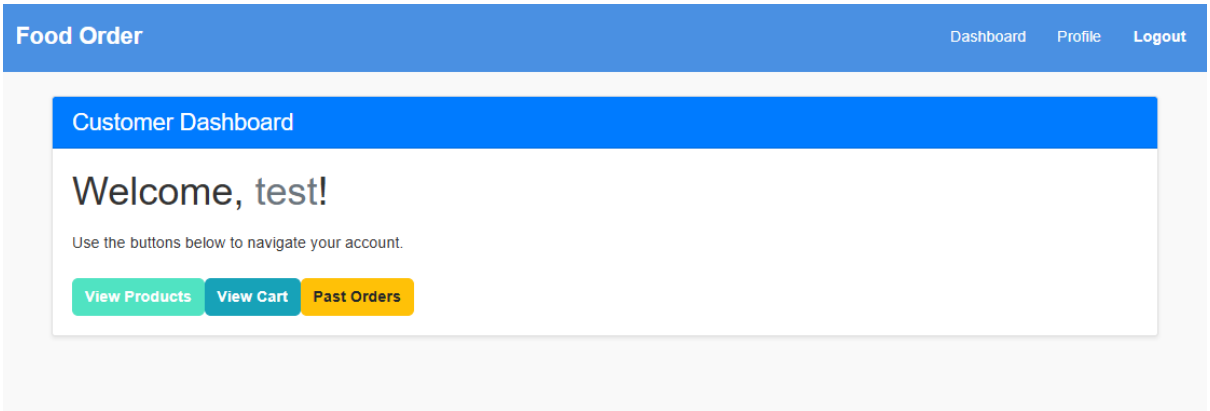


Figure 5. 6 Dashboard of Customer

Figure 5.6 shows the customer dashboard page that appears after successful login. This page greets the customer with a welcome message and presents several buttons, including "View Products," "View Cart," and "Past Orders," allowing the user to easily navigate through their shopping options. In addition, the customer can also quickly access their account settings and log out by clicking on the "Logout" and "Profile" links in the navigation bar at the top of the page. The customer can view previous purchases, manage their shopping experience, and access their cart or product listings all from this dashboard.

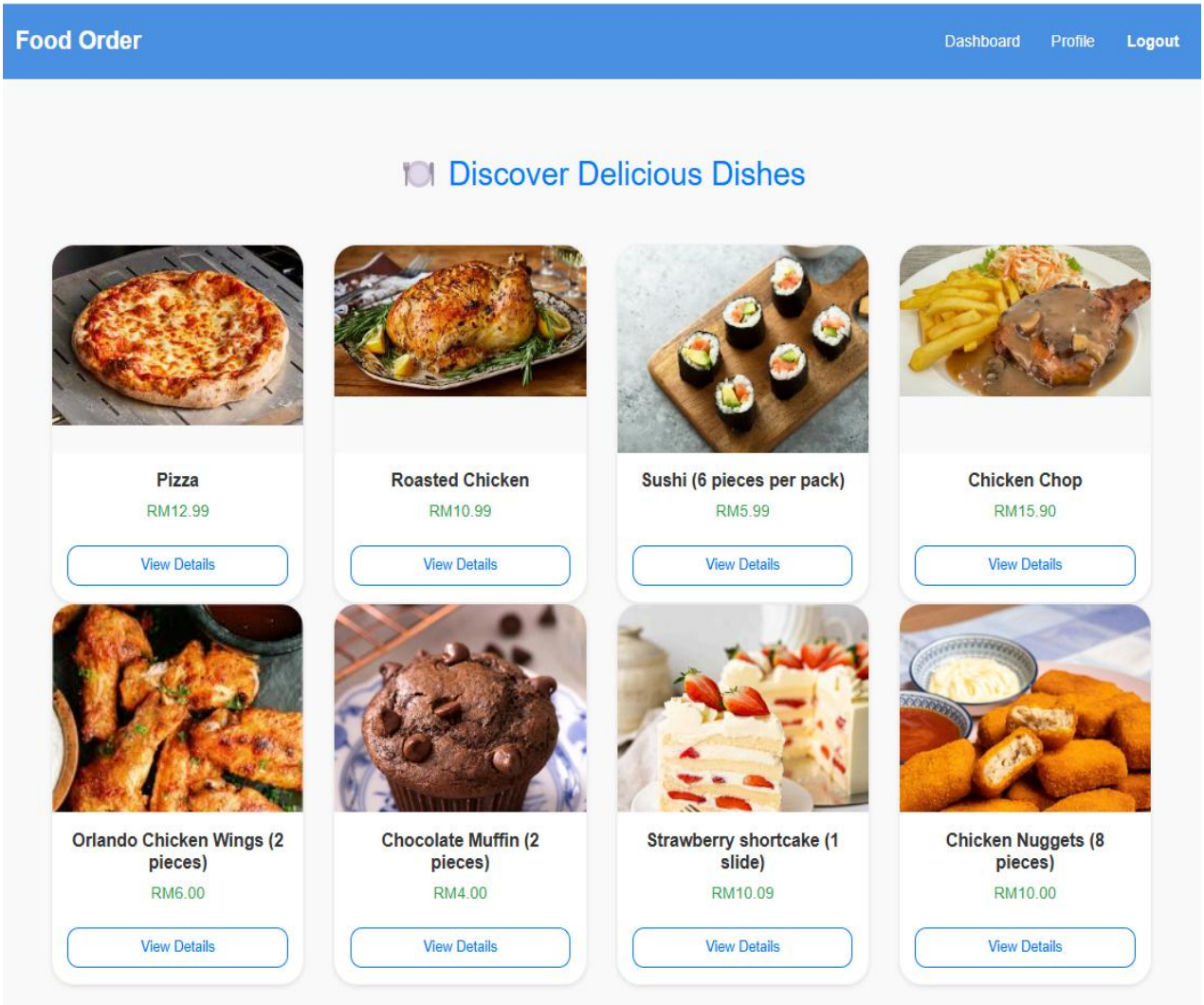


Figure 5. 7 Product Page of Customer

Figure 5.7 illustrates the food product page for the customer which displaying a variety of products from different restaurant owner. Each product is displayed with image and product name which allowing the customer to browse options from various restaurant owner. If the customer is interested in ordering a product, they can click on the "View Details" button to check the product's full details, such as specifications, price, availability, and other relevant information. This feature ensures that the customer has all the necessary information before making a purchase decision.

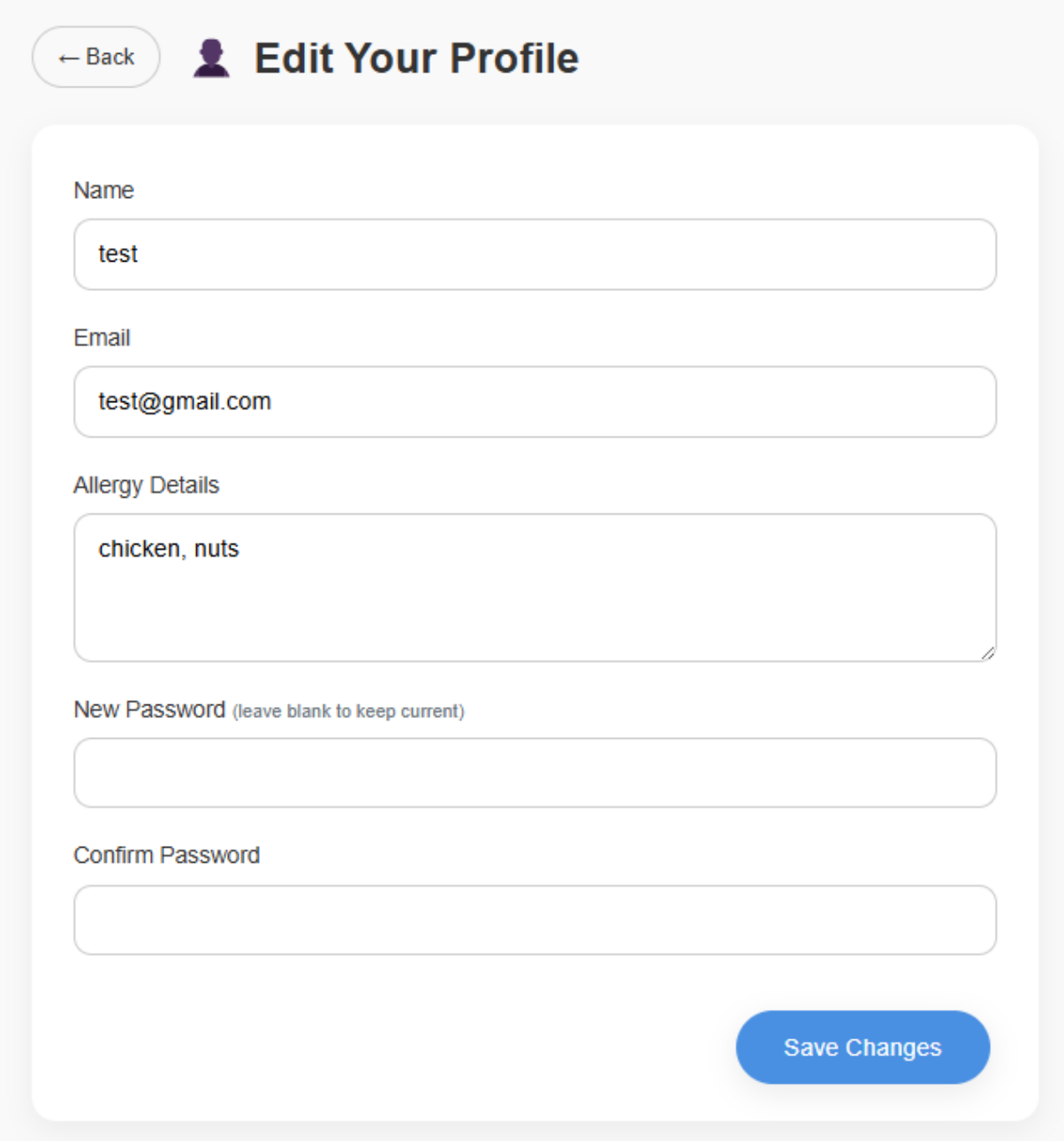


Figure 5. 8 User Profile

Figure 5.8 shows the user profile page, where customers can update their personal information. They can change details such as their name, update their latest allergy information, and modify their password. However, the email address is not editable, ensuring that it remains consistent for account verification and communication purposes. This feature allows customers to keep their profile current with any changes, including the most up to date allergy details, while maintaining key information secure.

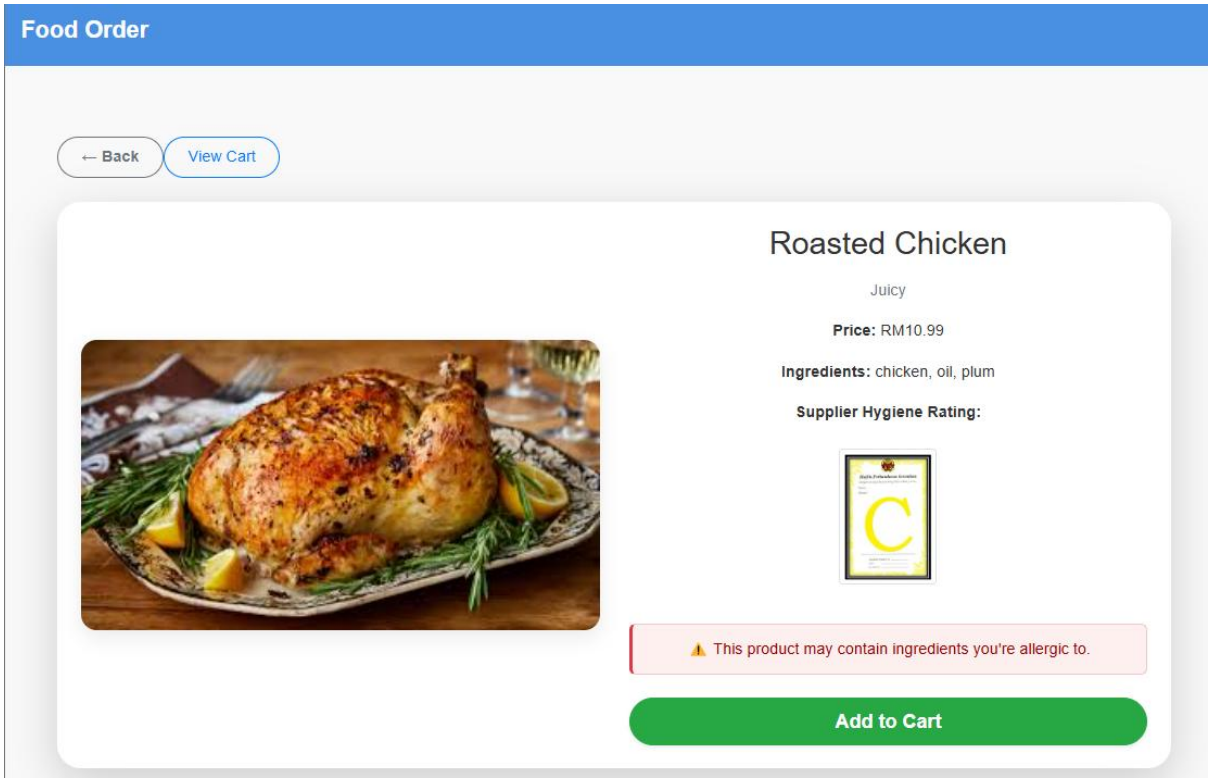


Figure 5. 9 Allergy detect (1)

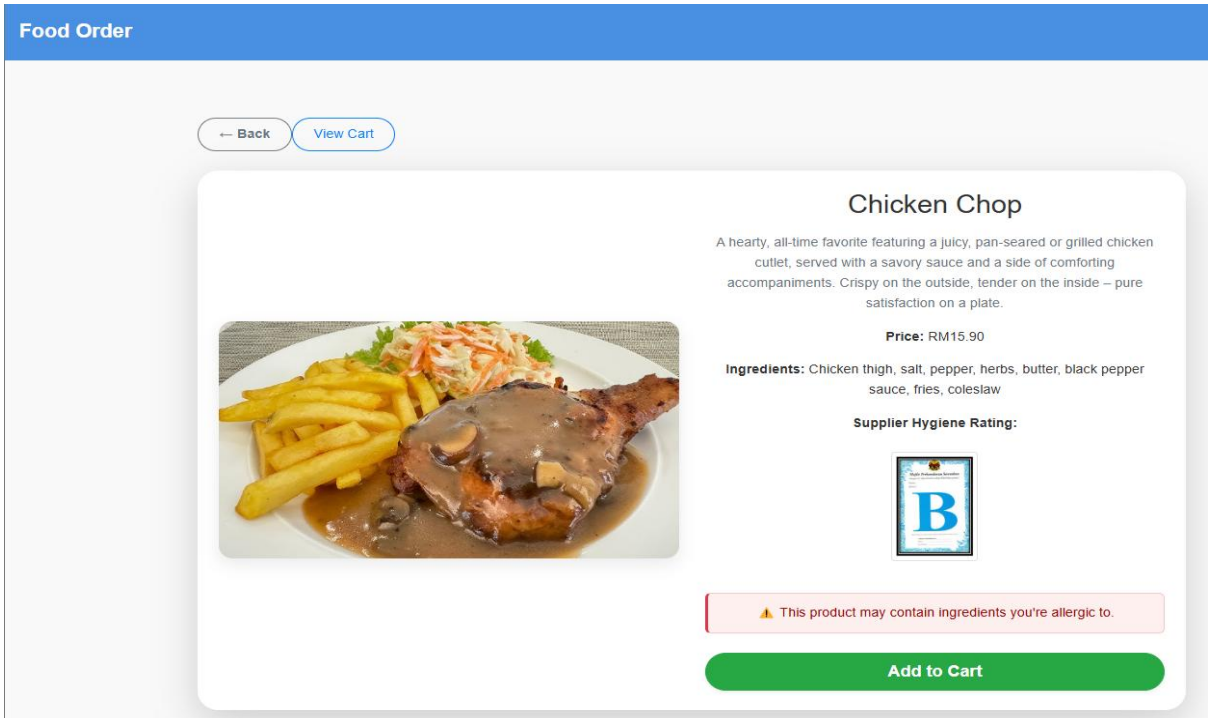


Figure 5. 10 Allergy detect (2)

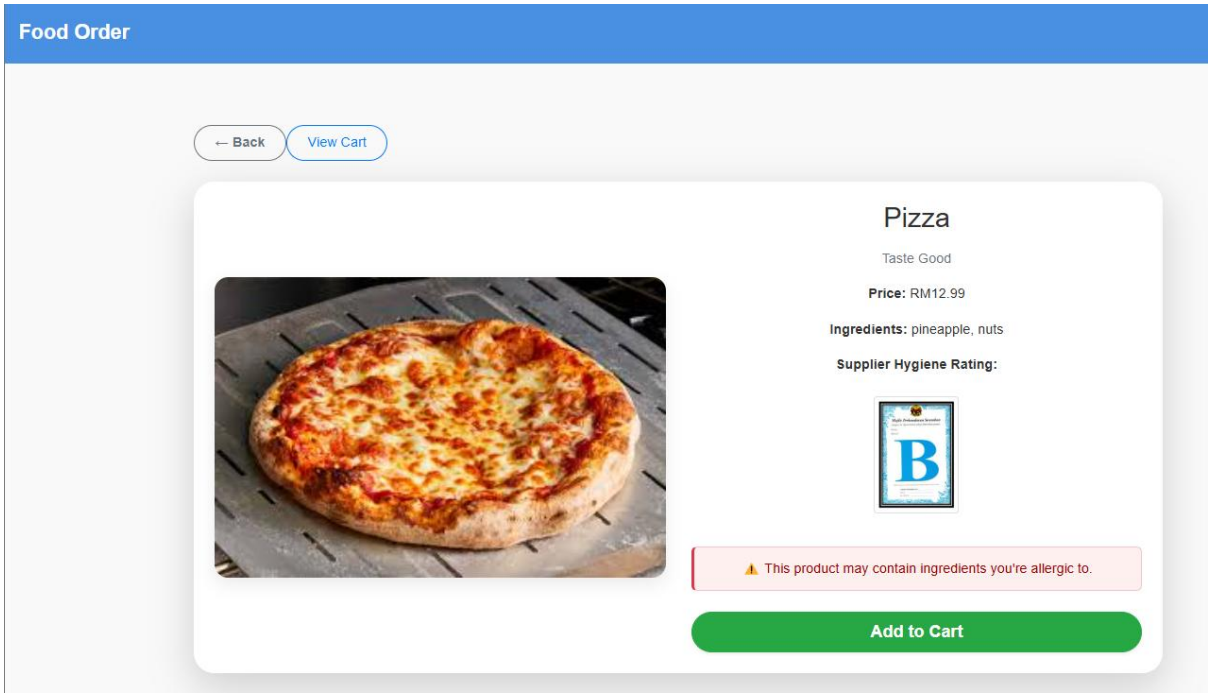


Figure 5. 11 Allergy detect (3)

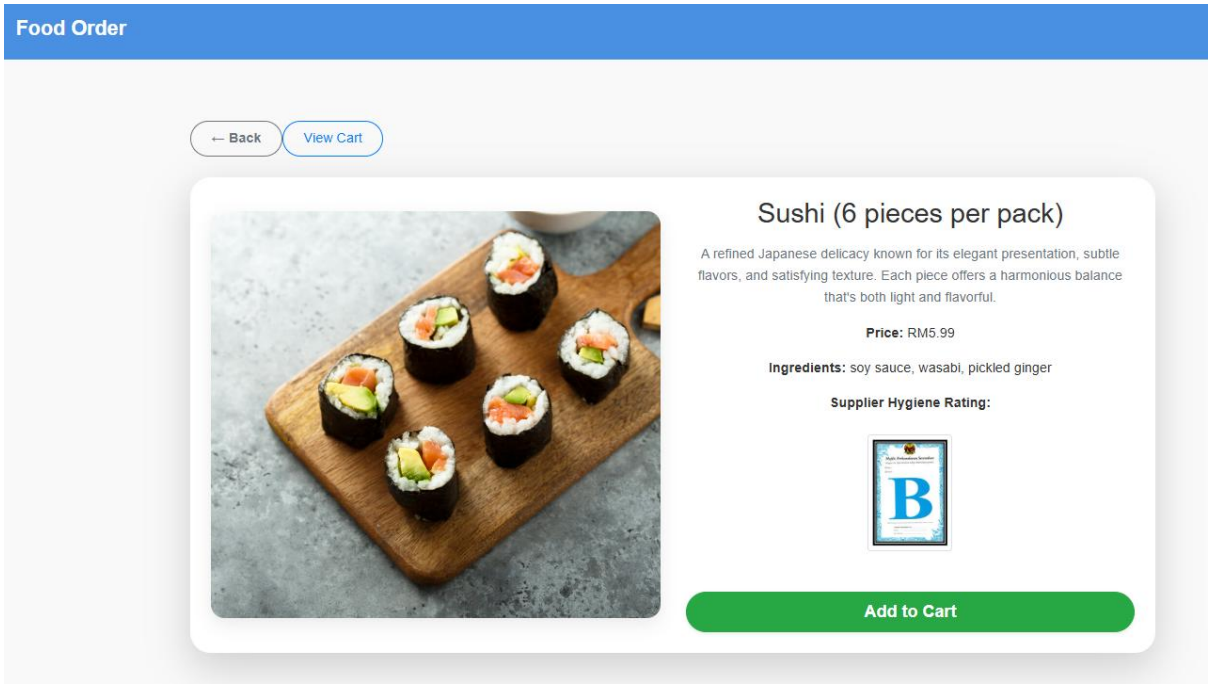


Figure 5. 12 Ingredients that do not contain allergy

Figure 5.9, Figure 5.10, and Figure 5.11 demonstrate the system's allergy detection feature. In Figure 5.9, the system identifies that the product contains chicken, which matches the customer's allergy details in Figure 5.8. Similarly, in Figure 5.10, the chicken chop contains "chicken thigh," which can also be detected as an allergen based on the customer's profile. In Figure 5.11, the system detects that the product contains nuts, which matches the user's allergy details. If the product does not contain any ingredients that match the user's allergies, Figure 5.12 shows that the system will indicate no allergy-related concerns with the product, confirming that it is safe for the customer to purchase. This allergy detection feature helps customers make informed choices based on their specific allergy information.

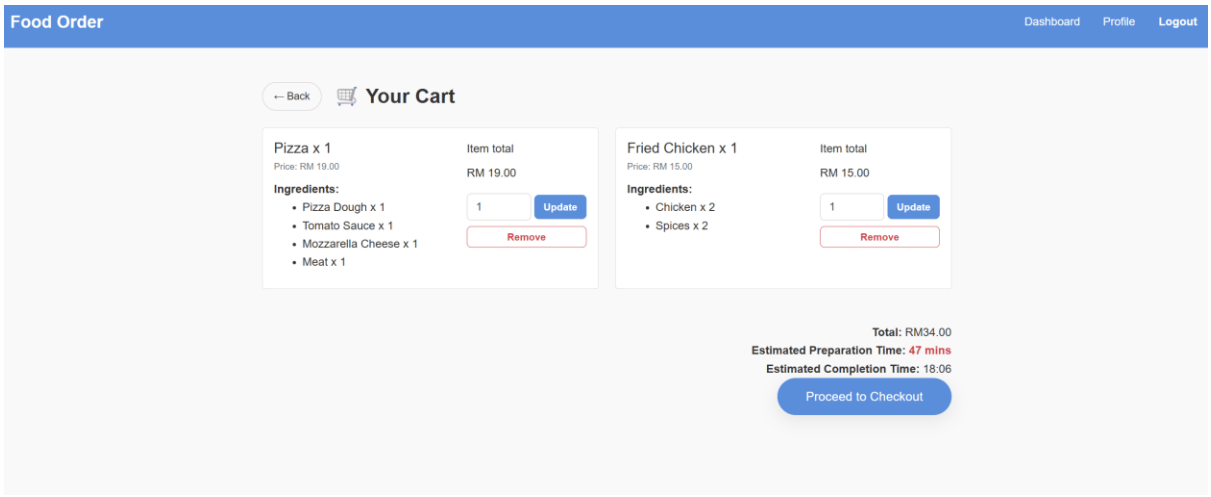


Figure 5. 13Product cart page

Figure 5. 14 Confirmation of checkout

Figure 5.13 shows the product cart page, where customers can view the products, they have added to their cart. With an option to add or remove items and change quantities, the cart makes product management simple. The customer can go to checkout as soon as they're prepared to complete the transaction. The system asks the customer to confirm their order before

completing the purchase by displaying a checkout confirmation message, as shown in Figure 5.14. This guarantees that the client can check their order before it is finished.

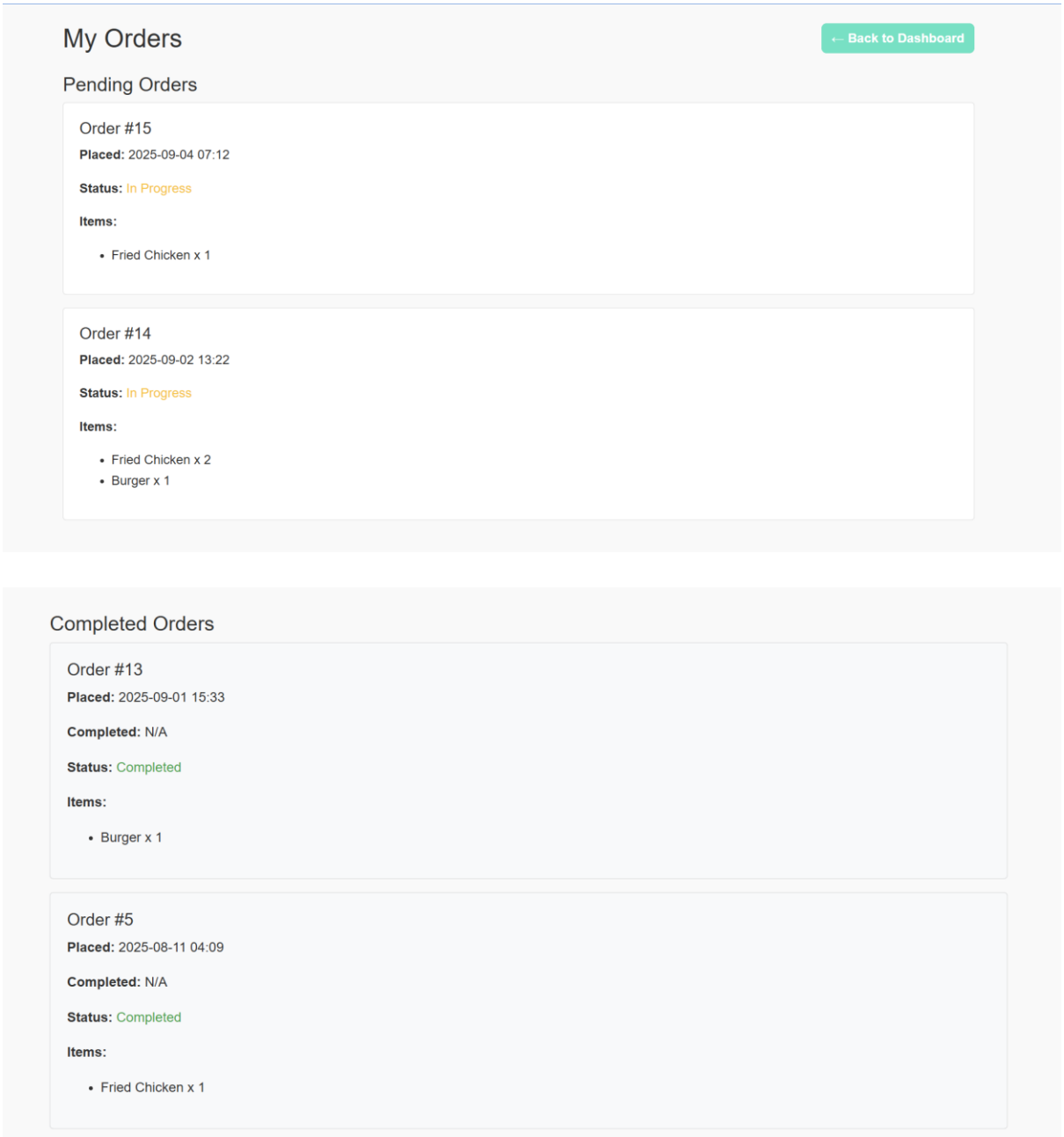


Figure 5. 15 Past Order Page

The past order page, which clients can view after finishing their checkout, appears in Figure 5.15. Customers can see information about their past orders, including the products they purchased, their costs, and the total amount spent, on this page. Customers can also choose to copy products from previous orders straight into their cart, which makes it simpler to place

new orders for the same items later. Customers can easily duplicate their past meal orders thanks to this feature, which improves convenience.

5.2 Restaurant Owner Side

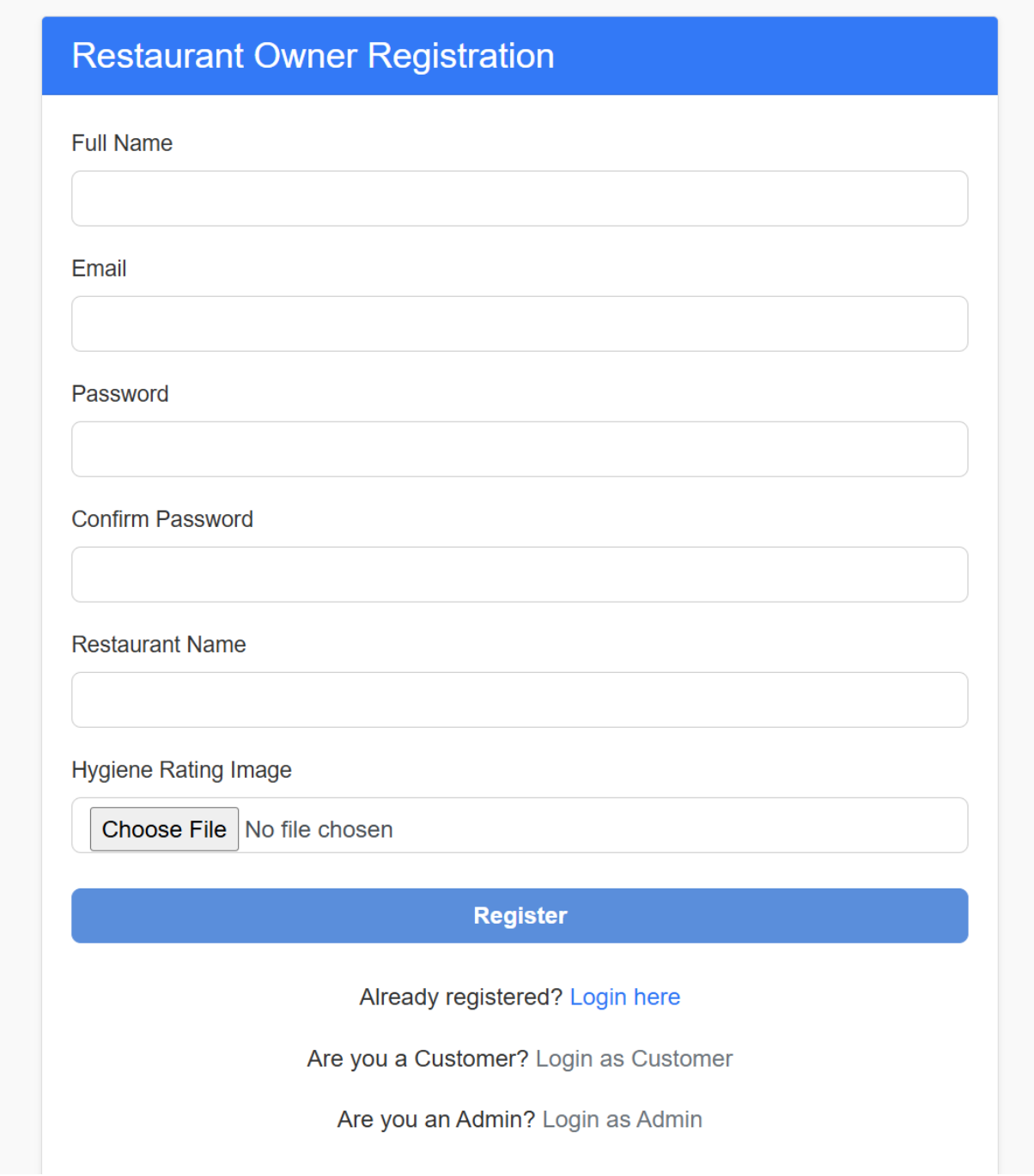


Figure 5. 16 Restaurant owner registration page

The restaurant owner registration page is displayed in Figure 5.16. Here, owners can create an account by providing their name, email address, password, and password confirmation. They are also required to upload a hygiene rating image, ensuring their registration includes this

essential information. Once completed, they can click the "Register" button to create their account and then proceed to the login page.

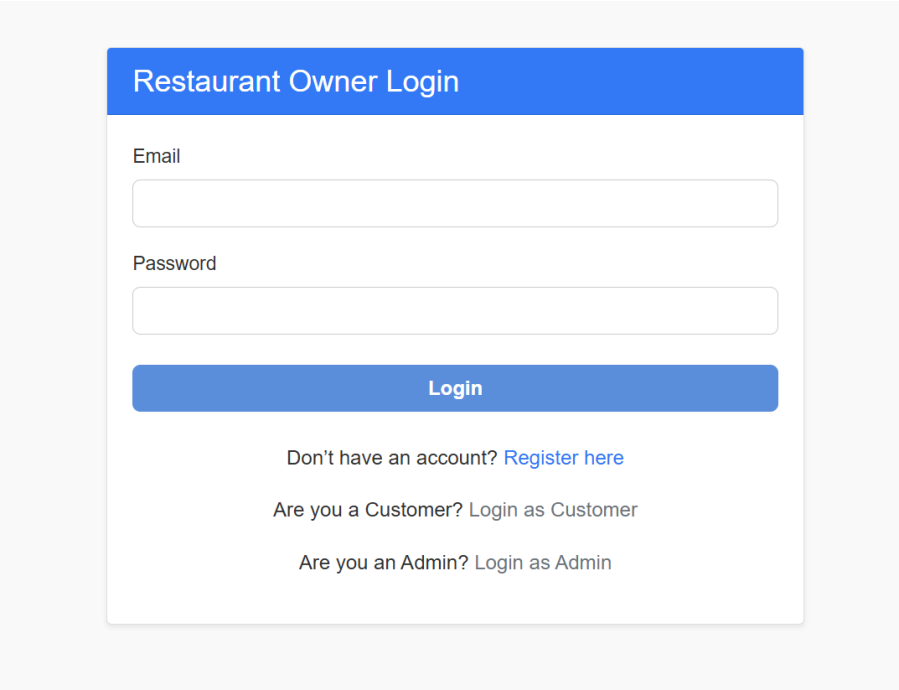


Figure 5. 17 Restaurant owner login page

Figure 5.17 shows the restaurant owner login page, where owners enter their registered Email and Password to access the system. The page provides a simple form with a prominent Login button. For new owners, a Register here link is provided to create an account before logging in. To streamline navigation, the page also includes quick links to Login as Customer and Login as Admin. Upon successful authentication, restaurant owners are redirected to their dashboard to manage products, orders, inventory, and chefs.

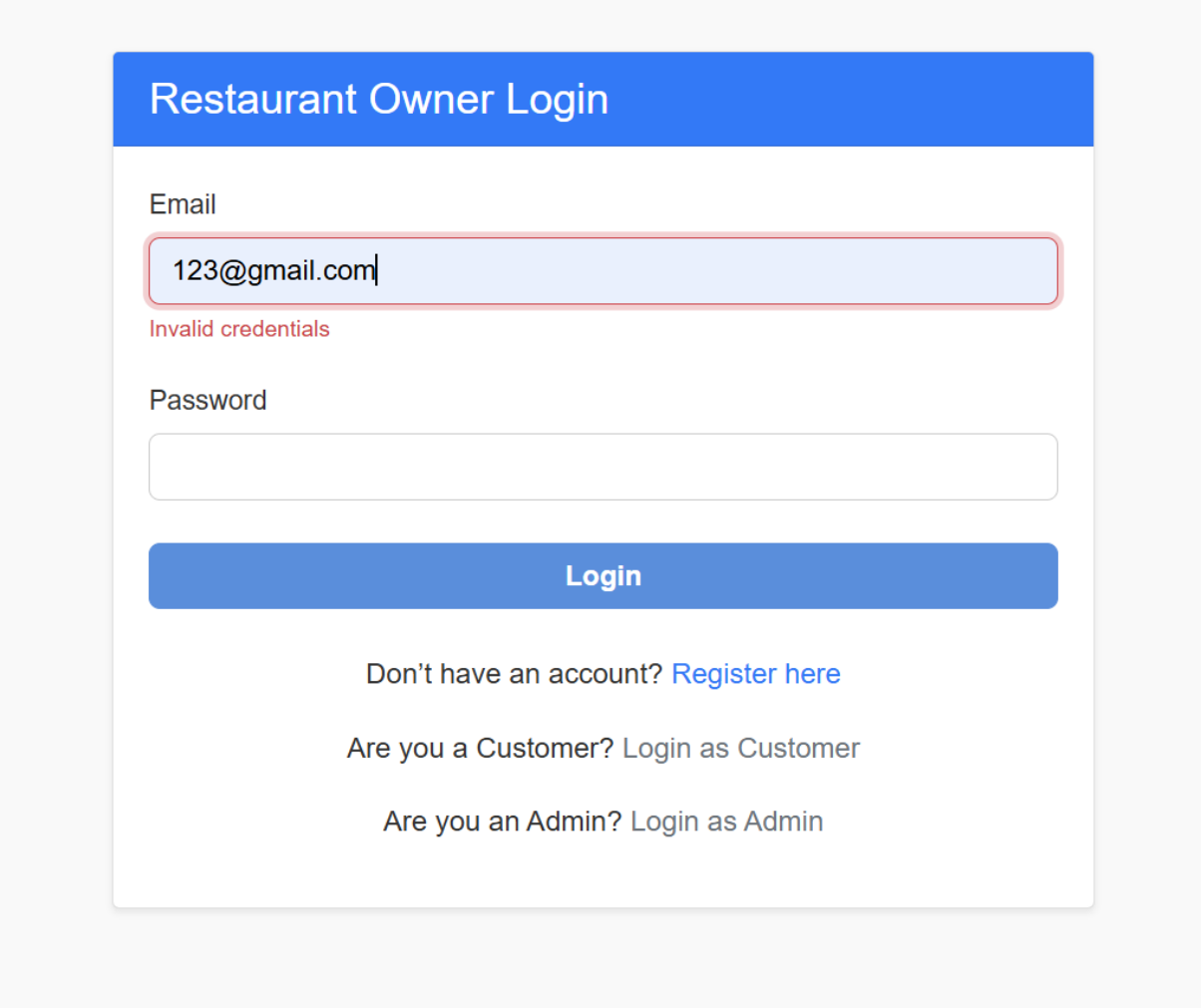


Figure 5. 18 Invalid Login page of Restaurant owner

Figure 5.18 depicts the invalid login page for restaurant owners. If the restaurant owner enters incorrect credentials, such as an invalid email or password, the system displays an error message, prompting them to try again**.**

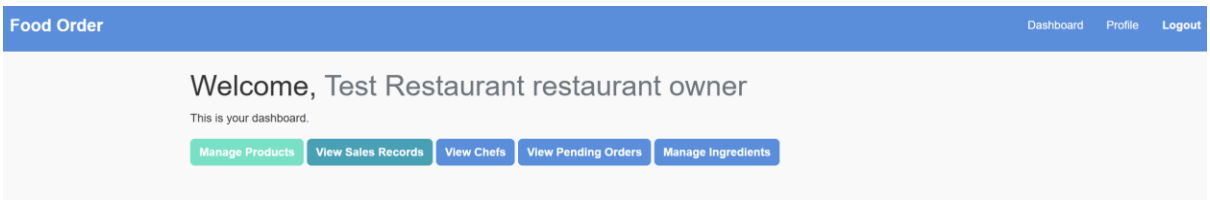


Figure 5. 19 Dashboard of Restaurant owner

Figure 5.19 shows the restaurant owner dashboard page, which appears after the restaurant owner logs in. The page welcomes the restaurant owner by name and provides two main options: "Manage Products" and "View Sales Records." These buttons allow the restaurant owner to

manage their product listings and review their sales history. The navigation bar at the top provides easy access to the restaurant owner's profile and the option to log out. This dashboard serves as the central hub for restaurant owners to oversee and manage their business activities.

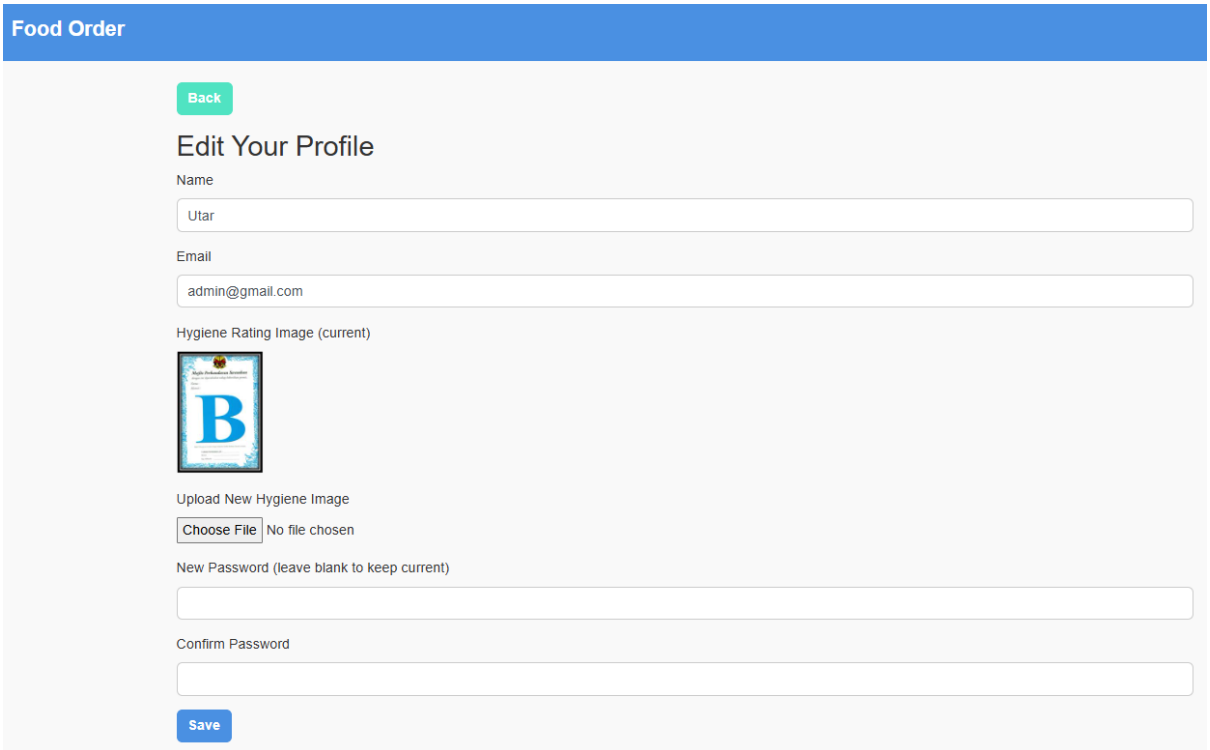


Figure 5. 20 Restaurant owner profile page

Figure 5.20 shows the restaurant owner profile page, where restaurant owners can edit their profile information. The page allows the restaurant owner to change their name, upload a new hygiene rating image, and update their password. However, the email address field is not editable, ensuring that the restaurant owner's email remains unchanged for consistency and communication purposes. Once the restaurant owner has made the necessary updates, they can click "Save" to apply the changes. This page ensures that restaurant owners can maintain accurate details while keeping key information, such as their email, secure.

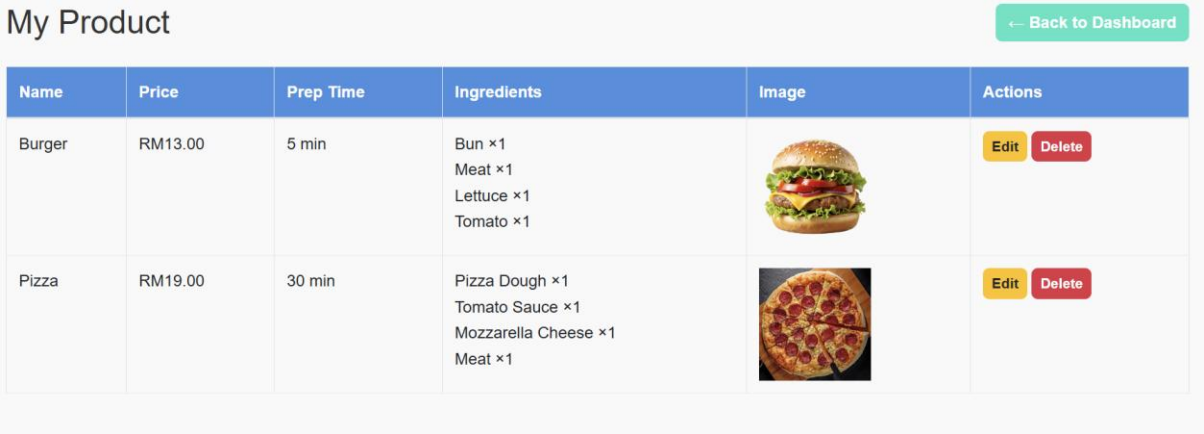


Figure 5. 21 Restaurant owner product page

Figure 5.21 shows the restaurant owner product page, where restaurant owners can view a list of their products. Each product includes the product name, price, ingredients, an image, and action buttons to either edit or delete the product. This page provides a clear overview of the restaurant owner's product offerings and allows for easy management.

Figure 5. 22 Delete confirmation

Figure 5.22 shows the confirmation prompt when a restaurant owner attempts to delete a product. The system asks the restaurant owner to confirm the deletion of the product, with options to either proceed by clicking "OK" or cancel the action by clicking "Cancel." This ensures that restaurant owners are aware of the action they are taking before it is finalized.

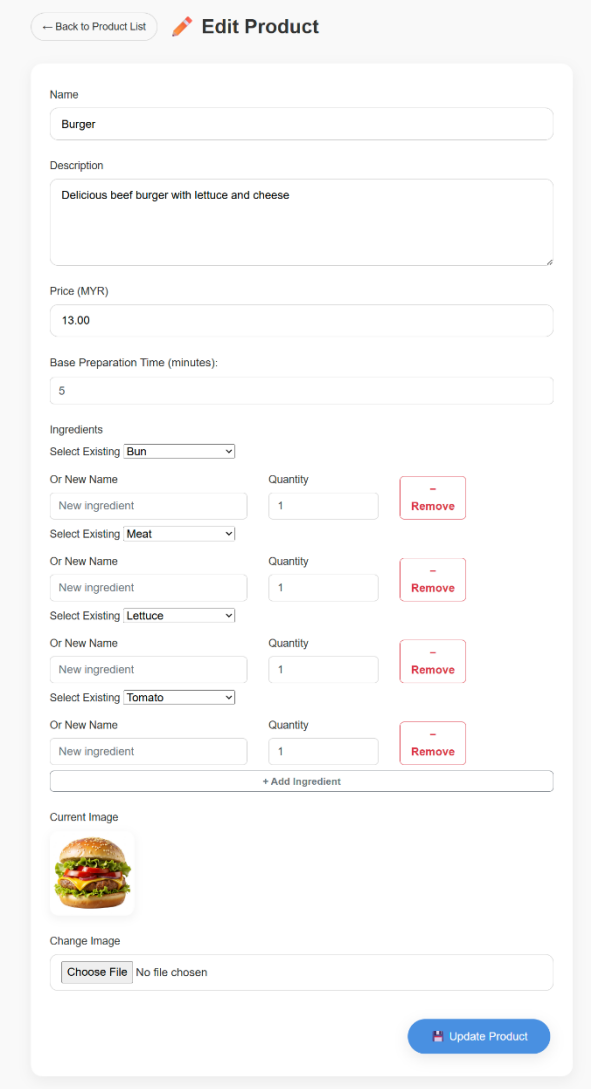


Figure 5. 23 Product edit page

Figure 5.23 shows the product edit page for restaurant owners. The form is pre-filled with the current details and allows the owner to update the Name, Description, Price (MYR), and Base Preparation Time (minutes). In the Ingredients section, each row lets the owner either select an existing ingredient or enter a new name, specify the Quantity, and Remove a row if needed; additional rows can be added using + Add Ingredient to adjust the product's recipe. The page also displays the Current Image preview and provides a Change Image file picker to replace it. A Back to Product List link supports quick navigation. Once all changes are made, the owner clicks Update Product to save the updates.

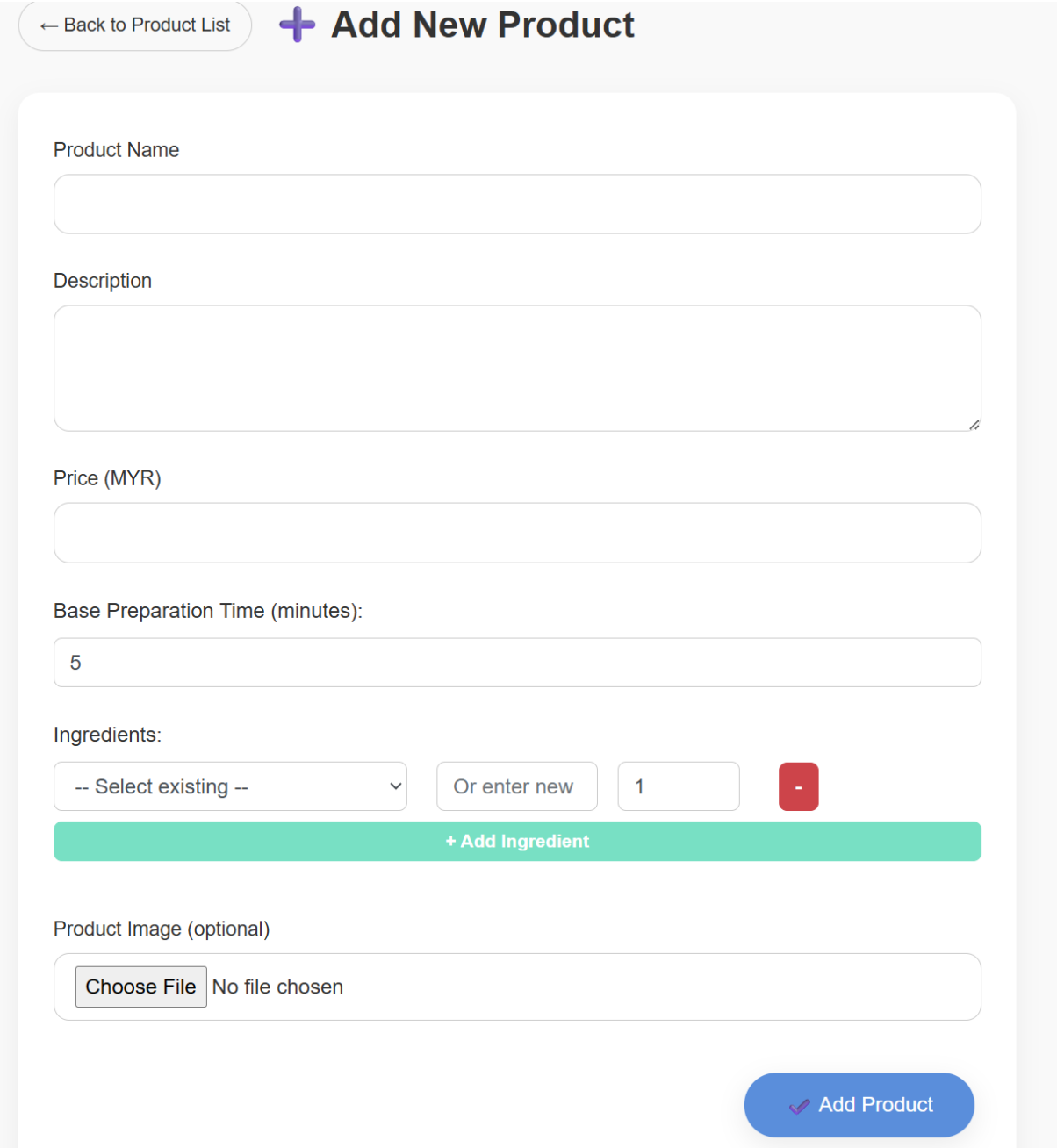


Figure 5. 24 Add product page

Figure 5.24 shows the "Add New Product" page, where restaurant owners can add a new product to their listing. The restaurant owner is prompted to enter the product's name, description, price (in MYR), and ingredients. Additionally, they can upload a product image, although this is optional. Once all the necessary details are filled out, the restaurant owner can click the "Add Product" button to save and add the product to their inventory. This page allows restaurant owners to easily expand their product offerings and ensure all relevant information is provided for each item.

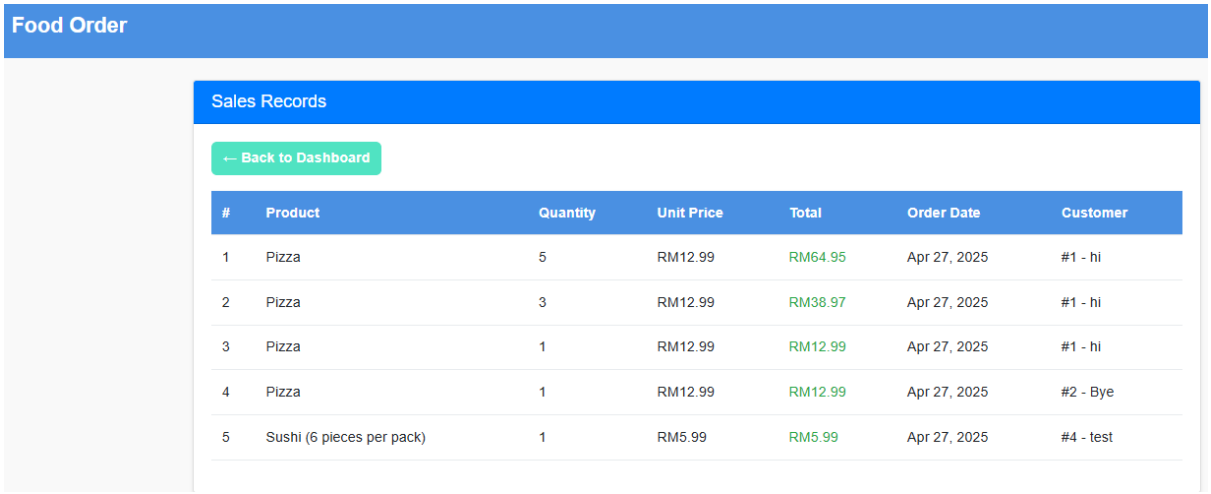


Figure 5. 25 Sales record page

Figure 5.25 shows the sales record page, where restaurant owners can view the sales history of the products they have created. The page displays key details, including the product name, quantity sold, unit price, total amount, order date, and the customer who made the purchase. Each restaurant owner can only view the sales records of their own products, ensuring that they have access to only relevant data for managing their inventory and sales. This page helps restaurant owners track their sales and monitor product performance

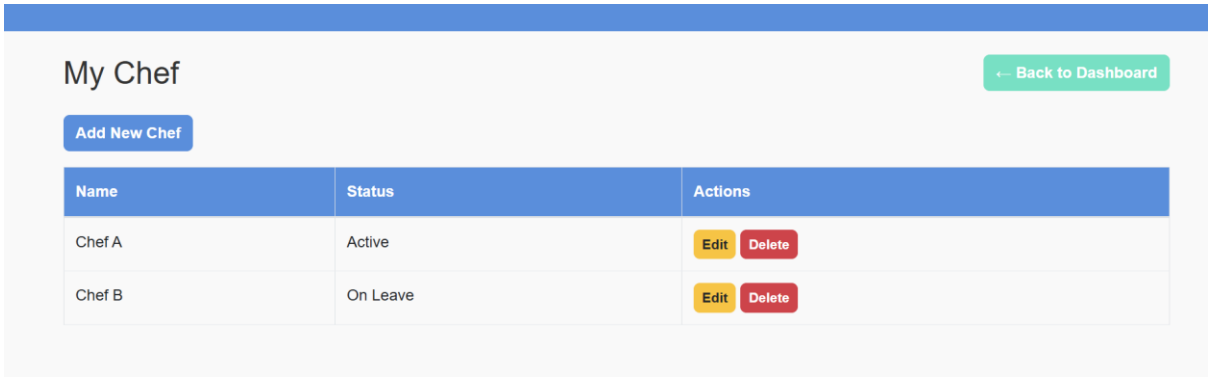


Figure 5. 26 Chef management page

Figure 5.26 shows the chef management page, where the restaurant owner can view and manage all chefs under their restaurant. The page lists each chef's Name and current Status (e.g., *Active*, *On Leave*) with action buttons to Edit or Delete a record. An Add New Chef button allows owners to register additional chefs, while a Back to Dashboard link provides quick navigation. This page centralizes staffing control so owners can keep their kitchen workforce up to date.

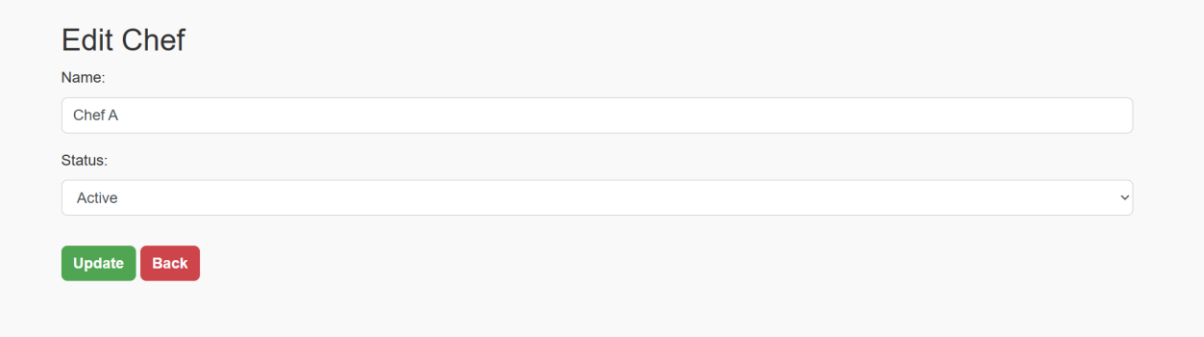


Figure 5. 27 Edit chef page

Figure 5.27 shows the edit chef page, which the owner accesses by clicking Edit from the list. The form allows updating the chef's Name and selecting a Status from a dropdown. The owner clicks Update to save changes or Back to return without modification. Updating the status immediately reflects on the chef list to support accurate staffing and order preparation planning.

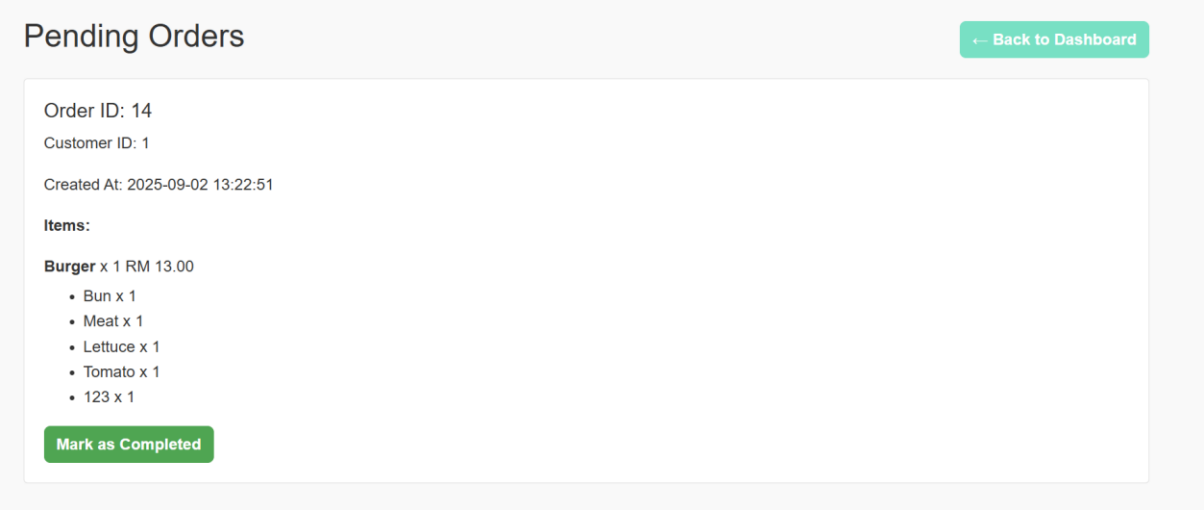


Figure 5. 28 Restaurant Owner pending orders page

Figure 5.28 shows the restaurant owner's pending orders page. Each card displays the Order ID, Customer ID, and Created At timestamp, followed by an itemized list that includes the product name, quantity, price, and a breakdown of its ingredients to guide kitchen preparation. The owner can click Mark as Completed to finalize the order, after which it is removed from the pending list and its completion time is recorded. A Back to Dashboard button is provided for quick navigation.

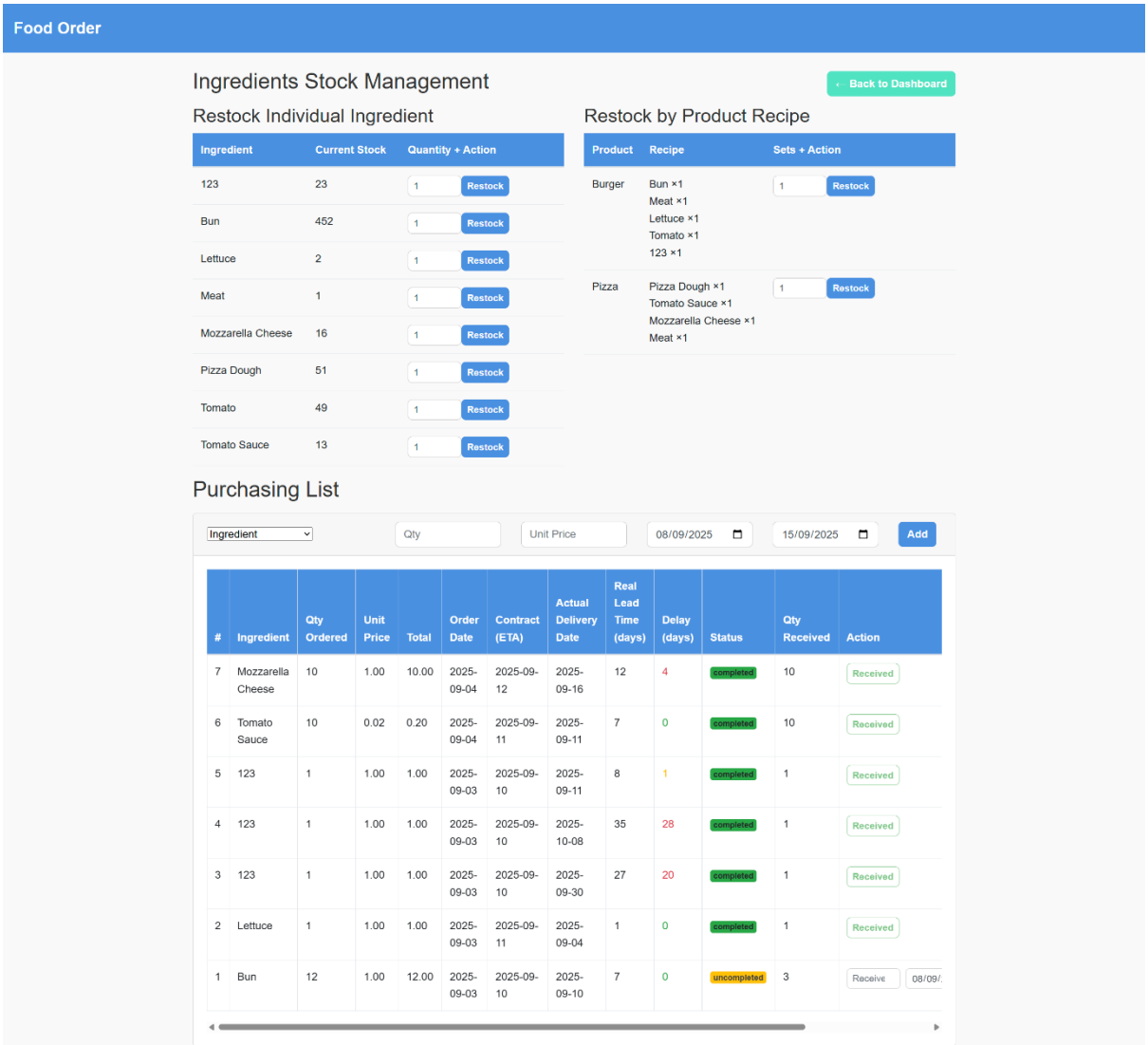


Figure 5. 29 Ingredients stock management and purchasing list page

Figure 5.29 shows the ingredients stock management page for restaurant owners. The page provides two quick restock options and a purchasing list to track orders and receipts. Restock Individual Ingredient lists each ingredient with its Current Stock. The owner enters a Quantity and clicks Restock to add stock for that single item and Restock by Product Recipe displays each product with its recipe (the list of required ingredients). The owner specifies the number of Sets and clicks Restock to add stock for all ingredients in that recipe in one action—useful for batch preparation.

The lower section is the Purchasing List, where the owner can create and monitor ingredient purchase orders. There is a 'Add' list allows selecting an Ingredient, entering Qty and Unit Price, setting a date range, and clicking Add to record a new purchase. The table then tracksevery order with columns for Ingredient, Qty Ordered, Unit Price, Total, Order Date,

Contract (ETA), Actual Delivery Date, Real Lead Time (days), Delay (days), Status (*completed* / *uncompleted*), Qty Received, and Action. For uncompleted rows, the owner can click Receive to confirm delivery (including partial receipts), which updates stock and calculates lead time and delay, completed rows display Received as disabled.

5.3 Admin Side

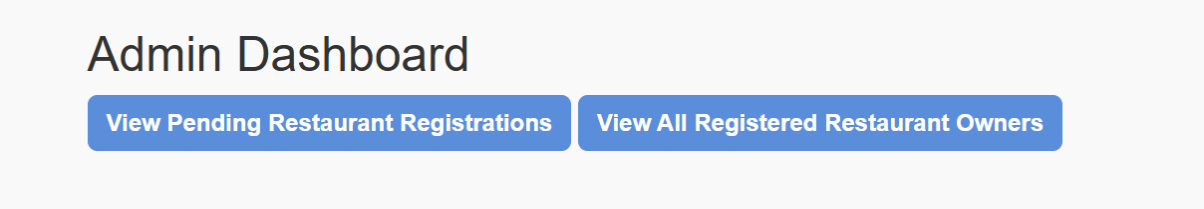


Figure 5. 30 Admin dashboard page

Figure 5.30 shows the admin dashboard, which provides two primary actions for platform oversight. View Pending Restaurant Registrations takes the admin to a list of newly submitted owner accounts awaiting approval, where each application can be reviewed and decided. View All Registered Restaurant Owners opens the full registry of approved owners for ongoing monitoring and auditing. This dashboard streamlines the admin's workflow by placing verification and oversight shortcuts in one place.

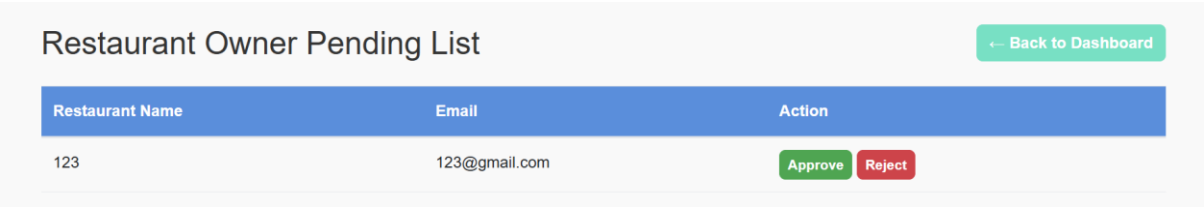


Figure 5. 31 Restaurant owner pending list page

Figure 5.31 shows the pending list page for admin review. When there are no new applications, the table displays a message indicating that no pending restaurant owners are found. When applications exist, each row lists the Restaurant Name and Email, with action buttons to Approve or Reject the registration. A Back to Dashboard button is provided for quick navigation.

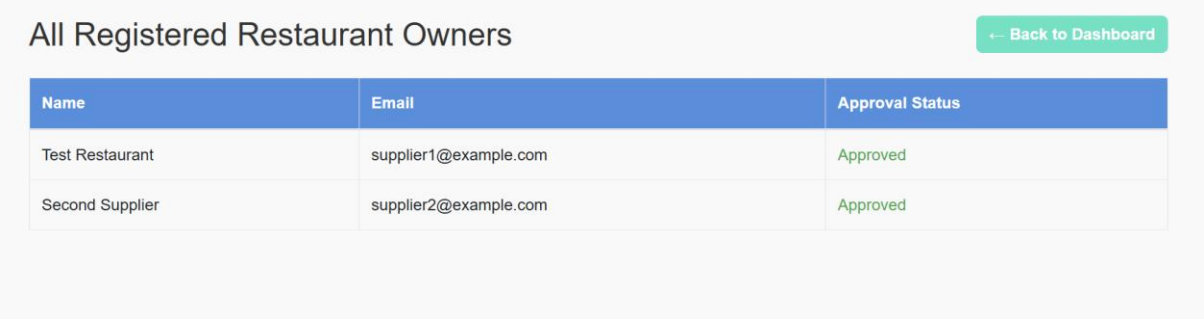


Figure 5. 32 All registered restaurant owner's page

Figure 5.32 shows the list of approved restaurant owners. The table displays each owner's Name, Email, and Approval Status for ongoing oversight. This page enables the admin to verify that only approved businesses have access to the platform and to audit registrations as needed.

CHAPTER 6: Unit testing

No	Test Case	Function Name & Description	Inputs	Expected Output	Actual Output	Action/ Remark
1	Add product to cart	addToCart(productid, qty) *Note:* This function enables adding an item to the cart	Valid product, qty > 0, stock available	Item added; cart total updated; message "Added to cart"	Able to click add button and alert	PASS
2	Block add when stock not enough	checkInventoryFeasibility(productId, qty) *Note:* This function prevents impossible orders	Product with limited stock, qty exceeds feasible	Error "Insufficient stock", product shows "Sold Out"	Error banner "Insufficient stock" shown, product marked "Sold Out", add to cart disabled	PASS
3	Allergy warning shown	checkAllergy(allergies, productId) *Note:* This function check allergies to ingredients	allergies "nuts, chicken", product contains conflict	Warning banner with offending ingredients, checkout	Warning banner "Contains nuts or chicken" displayed	PASS
4	No allergy detected	checkAllergy(allergies, productId) *Note:* This function check allergies to ingredients	allergies empty or non-conflicting	"No allergy detected", proceed to checkou	No allergy detected shown	PASS

5	ETA basic calculation	estimateEta(cart, activeChefs, pendingItems) *Note*: this function calculates estimate preparation time	base time present, activeChefs ≥ 1, pendingItems ≥ 0	ETA =base + factor times pending items divided by active chefs plus	ETA displayed, updates when cart changes, no errors observed	PASS
6	Place order successfully	placeOrder(cart, payment) *Note*: This function enables user create an order	Valid cart and payment	Order created with ID, status set to In Progress	Order ID shown, order summary saved, status set to In Progress	PASS
7	Inventory deduction on order	deductStock(orderId) *Note*: This function reduce stock by recipe time quantity	New order with items	Each ingredient stock reduced correctly, never negative	Stocks decrease based on recipe quantity, no negative values, database values updated	PASS
8	View past orders	getPastOrders(customerId) *Note*: This function is getting past order history list	Valid customerId	Past orders listed	History page shows orders in descendin g date	PASS

					order, details match database	
9	Reorder from history	reorder(orderId) *Note*: This function copy items to cart for customer reorder	Existing completed order	Items copied, allergy and stock checks run again, ETA updated	Items duplicated into cart, allergy check runs, , ETA recalculated	PASS
10	Validate restock fields	validatePurchase(input) *Note*: This function ensure the quantitiy, unit price, date for purchase	qty, unit price, dates for purchase	qty > 0, unit price > 0, contract date ≥ order date, received qty ≤ remaining,	Valid inputs pass with no errors, invalid inputs show field messages for quantity, price, dates, and received quantity	PASS
11.	Pending orders visible to correct restaurant owner	showPendingOrders() *Note*: This function show restaurant_owner pending order list	Two restaurant owner with different products and orders	restaurant owner see only their orders		PASS

12.	Sales summary by custome	sales?filter=customer *Note*: This function filter report view	Orders exist for multiple customers	Table groups totals by customer correctly	Report shows correct totals per customer and correct order counts	PASS
13.	Restock by product sets	restockByProduct(productId, sets) *Note*: This function enable restaurant owner restock ingredients by product set	sets equals 3, recipe uses X equals 2 per set	Stock of X increases by 6 and summary message correct	Stock updated by formula, success summary shows total increase 6	PASS
14.	Receive quantity not beyond remaining	purchasesReceive() Note: This function is to validate received quantity always greater than remaining	received quantity greater than remaining	Validation error and no stock change	Error message shown for received quantity, stock and status unchanged	PASS
15.	Add chef validation	ChefController@store Note: This function is to validate chef name and status is valid information	Missing name or invalid status	Validation error shown and no record created	Form returns with errors, database has no new chef	PASS

16.	External ingredients search success		Query returns ingredient list	200 with ingredients string	Response 200 with ingredients list string displayed	PASS

Table 6. 1 Unit Testing

CHAPTER 7: Conclusion

7.1 Conclusion

This research focuses on improving the food ordering system with the goal of making it safer for customers and more efficient for restaurant owners. The major issues like food allergies and order processing delays are addressed by the system. Customers can make safer and better-informed decisions by using it to suggest food options based on their allergies and to get better estimates of how long it will take to prepare the food. In addition, it assists restaurant owners by enhancing order and inventory management, offering resources for replenishing ingredients, tracking purchase lists, keeping tabs on sales, and gaining knowledge about the most popular products. The ordering process is made easier for both parties by additional features that allow restaurant owners to add and update product details and a feature that displays previous orders to customers. In conclusion, this research emphasizes how technology can improve food ordering by putting the safety of customers first, facilitating effective restaurant operations, maintain inventory control, and enhancing the overall dining experience for customers and restaurant owners.

7.2 Recommendation

According to the research's findings, it is recommended that restaurants implement the suggested food ordering system in order to improve customer satisfaction and operational effectiveness. The system's integration of allergy-based food recommendations guarantees that customers choose safer meals, and precise preparation time estimates help control expectations and lessen customer annoyance brought on by delays. The system offers restaurant owners useful tools for order management, sales tracking, product analysis, and inventory control through purchasing and restocking lists. These features enable improved resource management and decision-making in addition to streamlining daily operations. In order to fully utilize the system's capabilities, particularly with regard to product updates and inventory management, it is also advised that restaurants regularly train their employees. To increase efficiency and customer satisfaction, future enhancements might involve adding data analytics for demand forecasting and customer behavior analysis to the system. All things considered, implementing this system will help customers choose safer foods and restaurant owners manage their operations more skillfully.

REFERENCES

[1] M. A. Golding et al., "An investigation of a novel allergy-sensitive food recommendation system," Frontiers in Allergy, vol. 5, 2024. [DOI: 10.3389/falgy.2024.1301834]. [Online]. Available: https://doi.org/10.3389/falgy.2024.1301834

[2] L. P. Gymnopoulos, K. Dimitropoulos, "Artificial Intelligence and Machine Learning Technologies for Personalized Nutrition: A Review," Informatics, vol. 11, no. 3, p. 62, 2024. [Online]. Available: https://www.mdpi.com/2227-9709/11/3/62

[3] M. Kumar et al., "Prediction of Kitchen Preparation Time in Food Delivery," Unpublished, Indian Institute of Technology Delhi, 2023. [Online]. Available: https://www.cse.iitd.ac.in/~cs5180411/Prediction%20of%20Kitchen%20Preparation%20Time%20in%20Food%20Delivery.pdf

[4] A. D. Starke, J. Dierkes, G. A. Lied, G. A. B. Kasangu, and C. Trattner, “Supporting healthier food choices through AI-tailored advice: A research agenda,” PEC Innovation, vol. 6, p. 100372, Jan. 2025, doi: https://doi.org/10.1016/j.pecinn.2025.100372.

[5] A. Halim, H. Hashim, A. M. Sahilah, and M. A. Ghani, "Food safety regulations implementation and their impact on food security level in Malaysia: A review," ResearchGate, 2024. [Online]. Available: https://www.researchgate.net/publication/378967337_Food_safety_regulations_implementation_and_their_impact_on_food_security_level_in_Malaysia_A_review

[6] Z. Wang, C. Meng, S. Ji, T. Li, and Y. Zheng, “Food package suggestion system based on multi-objective optimization: A case study on a real-world restaurant,” Applied Soft Computing, vol. 93, p. 106369, Aug. 2020, doi: https://doi.org/10.1016/j.asoc.2020.106369.

[7] A. Bryan Endres, R. Endres, and M. K. Nižić, “Restaurant disclosure of food allergens: Analysis and economic implications,” Tourism and Hospitality Research, vol. 21, no. 2, pp. 202–215, Dec. 2020, doi: https://doi.org/10.1177/1467358420977576.

[8] B. C. Remington et al., “Allergen quantitative risk assessment within food operations: Concepts towards development of practical guidance based on an ILSI Europe workshop,” Food Control, vol. 138, p. 108917, Aug. 2022, doi: https://doi.org/10.1016/j.foodcont.2022.108917.

[9] M. A. Flanagan and J. M. Soon-Sinclair, “Consumers’ perceptions of regulatory food hygiene inspections of restaurants and takeaways,” British Food Journal, Dec. 2024, doi: https://doi.org/10.1108/bfj-07-2024-0764.

[10] C. G. Awuchi, “HACCP, quality, and food safety management in food and agricultural systems,” Cogent Food & Agriculture, vol. 9, no. 1, 2023, doi: https://doi.org/10.1080/23311932.2023.2176280.

[11] N. Recuero-Virto and C. Valilla-Arróspide, “Forecasting the next revolution: food technology’s impact on consumers’ acceptance and satisfaction,” British Food Journal, Jan. 2022, doi: https://doi.org/10.1108/bfj-07-2021-0803.

POSTER

UTAR
UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION COMMUNICATION AND TECHNOLOGY

FOOD ORDER APPLICATION

INTRODUCTION

RISING DEMAND FOR FOOD ORDERING SERVICES DRIVEN BY CONVENIENCE, BUT CHALLENGES PERSIST IN ALLERGEN MANAGEMENT, ORDER DELAYS, AND LACK OF INGREDIENTS MANAGE.

GOAL: DEVELOP A FOOD ORDERING SYSTEM THAT PRIORITIZES SAFETY, EFFICIENCY, AND TRANSPARENCY.

OBJECTIVE

1. ALLERGY & DIETARY DETECTION: SAFE MEAL SUGGESTIONS BASED ON ALLERGY DETAILS.
2. EFFICIENT ORDER PROCESSING: REAL-TIME PREPARATION TIME ESTIMATES FOR BETTER KITCHEN MANAGEMENT.
3. BETTER INVENTORY MANAGE: VERIFIED HYGIENE RATINGS TO GUIDE SAFE FOOD CHOICES.

PROPOSED METHOD

FOCUS ON ALLERGY CHECKS, PREPARATION TIME ESTIMATES, AND INVENTORY MANAGEMENT.

SCRUM METHODOLOGY METHOD: STEP-BY-STEP APPROACH FROM DESIGN TO TESTING.

WHY PROPOSED SYSTEM?

1) CURRENT SYSTEMS FAIL TO PROVIDE CRITICAL SAFETY AND EFFICIENCY FEATURES.

2) OUR SYSTEM PROVIDE RESTAURANT OWNERS TO EASILY MANAGE ORDERS, TRACK SALES, UPDATE PRODUCTS, AND CONTROL INVENTORY IN ONE PLATFORM.

CONCLUSION

- SAFER, MORE EFFICIENT FOOD ORDERING WITH PERSONALIZED FEATURES.
- EMPOWERING CUSTOMERS WITH ALLERGY-FREE OPTIONS AND HYGIENE INFO.
- OPTIMIZING RESTAURANT OPERATIONS TO IMPROVE CUSTOMER SATISFACTION.
- PROVIDE ONE SOLUTION PLATFORM FOR RESTAURANT OWNER

PORJECT DEVELOPER: CHANG ZI HONG
PROJECT SUPERVISOR: TS DR WONG PEI VOON