



SOFTWARE ENGINEERING CLASS

PROJECT KAYSE EATS

Prepared by Group I :

Mohamed Kayse Hassan

Abdirahman Abdirizak Omer

Abdirizack Deeq Mahad



Phone: +252-67-234-5678

Email: info@kayseeats.so

Address: Hargeisa, Somaliland

DECLARATION

We, the undersigned, declare that this report titled "*Kayse Eats: Online Food Ordering System*" is our own work. It was developed as part of the **Software Engineering class at Gollis University**.

This report has not been submitted for any other degree, certificate, or professional qualification. It reflects the efforts and knowledge of our group members as part of our learning in the Software Engineering program.

Group Members:

1. Mohamed Kayse Hassan _____
2. Abdirahman Abdirizak Omer _____
3. Abdirizak Dek Mahad _____

Supervisor and Chairman:

1. Hussien Haybe

Date:

November 18, 2024

APPROVAL SHEET

This mini-project report titled "*Kayse Eats: Online Food Ordering System*" was prepared and submitted by Mohamed Kayse Hassan, Abdirahman Abdirizak Omer, and Abdirizak Dek Mahad. It was submitted as part of the requirements for the **Software Engineering class at Gollis University**.

This report was reviewed and approved by:

Supervisor and Chairman:

Hussien Haybe _____

Date of Review:

November 18, 2024

DEDICATION

We would like to dedicate this project to the people who have supported us throughout this journey.

To our families: Thank you for your love, patience, and constant encouragement. You have been our strength in challenging times and our motivation to keep moving forward.

To our supervisor, Mr. Hussien Haybe: Thank you for your guidance, valuable advice, and dedication to helping us succeed.

To our classmates and friends: Your support, feedback, and collaboration have made this experience both educational and enjoyable.

Finally, we dedicate this project to the **Software Engineering Department at Gollis University** for giving us the tools and knowledge to create this project.

ACKNOWLEDGEMENT

In the name of Allah, the Most Merciful and Compassionate, we express our deepest gratitude to all who helped us complete this mini-project.

We give special thanks to our teacher and supervisor, **Mr. Hussien Haybe**, for his guidance, patience, and valuable feedback throughout the project. His advice has been crucial in helping us achieve our goals.

We are grateful to the **Software Engineering Department at Gollis University** for teaching us the skills and providing us with the resources needed to build this system.

We also extend our thanks to the restaurant owners and customers who participated in our research. Their input helped us design a project that is practical and relevant to real-world problems.

Finally, we thank our families for their support and encouragement, and our classmates and friends for their collaboration, suggestions, and motivation.

ABSTRACT

Kayse Eats is a mini-project developed for the **Software Engineering class at Gollis University**. It is an online food ordering system that connects customers with local restaurants in Somaliland.

The system is designed to solve common problems in food delivery, such as delays, limited restaurant options, and wrong orders. Kayse Eats makes food ordering easier with features like:

- Real-time order tracking so customers know when their food will arrive.
- A simple and user-friendly website that allows customers to browse menus and place orders quickly.
- Secure payment methods, including mobile payments, for convenience.

This project is created using PHP, MySQL, and Bootstrap. It demonstrates the practical application of software engineering concepts. The system is also designed to help local restaurants by connecting them to more customers, increasing their sales and visibility.

By addressing the challenges of food delivery in Somaliland, Kayse Eats provides a modern, reliable, and easy-to-use solution for both customers and restaurants. This project is a small step towards improving food delivery services and supporting local businesses in the region.

TABLE OF CONTENTS

| | |
|---|---|
| 1. Project Title: Kayse Eats | 1 |
| 2. Introduction | 1 |
| 2.1 Overview of Kayse Eats | 1 |
| 2.2 Why It's Unique and New | 2 |
| 2.3 Examples of Restaurants | 2 |
| 2.4 Comparison with Existing Apps | 2 |
| 3. Background Problem | 3 |
| 3.1 Why Kayse Eats Is Needed | 3 |
| 3.2 Challenges with Existing Options | 3 |
| 3.3 How Kayse Eats Solves These Problems | 3 |
| 3.4 Impact on People's Lives | 4 |
| 4. Problem Statement | 4 |
| 4.1 Slow and Late Deliveries | 4 |
| 4.2 Unreliable Service in Bad Weather | 4 |
| 4.3 Incorrect Orders Due to Poor Address System | 4 |
| 4.4 Limited Restaurant Choices | 4 |
| 5. Objectives | 5 |
| 5.1 Fast and Reliable Delivery | 5 |
| 5.2 Connecting Many Restaurants | 5 |
| 5.3 Available All the Time (24/7) | 5 |
| 5.4 Simple Payment Methods | 5 |
| 5.5 Correct and Accurate Orders | 5 |
| 5.6 Support for Local Restaurants | 5 |
| 5.7 Easy-to-Use System | 5 |
| 6. Scope | 5 |
| 6.1 Customer Panel Features | 5 |
| 6.2 Admin Panel Features | 5 |
| 7. Methodology & Procedure | 5 |
| 7.1 Planning & Research (June) | 5 |
| 7.2 Design & Development (July) | 5 |
| 7.3 Testing (August) | 5 |
| 7.4 Launch & Presentation (September-October) | 5 |
| 8. Expected Outcome | 5 |
| 8.1 Easy & Quick Food Ordering | 5 |
| 8.2 Increased Sales for Restaurants | 5 |
| 8.3 Reliable & Trustworthy Service | 5 |
| 8.4 Happy Customers | 5 |
| 8.5 Support for Local Economies | 5 |
| 9. Schedule | 5 |
| 9.1 Detailed Timeline | 5 |

Project Title - "Kayse Eats"

Introduction: Kayse Eats is an online food ordering website that connects restaurants with customers in Somaliland. The name "Kayse" comes from a family name, making it feel local and familiar. It helps restaurants and hotels to get more customers, and it helps people to save time when ordering food. With Kayse Eats, customers can find available meals anytime and anywhere.

Why It's Unique and New: In Somaliland, there is no proper online food ordering system available. The only app that offers some food delivery is the Dhaweeye app, but it does not focus mainly on food. Kayse Eats is different because it will be the first complete online food ordering system that works well in Somaliland, connecting many restaurants in one place.

Examples of Restaurants: In Hargeisa, only a few hotels like Carro Edeg (in Suuqa Hoose), Zafaraan (in Jigjiga Yar), and Al Habasha (near Kaalinta Star) deliver food to their loyal customers. However, they do not have an online system. Kayse Eats will bring all these restaurants together, so customers can order from them easily using the website.

Comparison with Existing Apps: Unlike other apps, Kayse Eats will make it easy for anyone to find and order food. This makes it new and different in the Somali market.

Background Problem

Why Kayse Eats Is Needed: In Somaliland, people face many challenges when they want to order food. There is no complete online food ordering system available in our country. People usually have to call restaurants or physically go there to place their orders. This process takes time, is inconvenient, and makes it hard for people who are busy or do not want to leave their homes, especially in rainy weather.

Examples of Challenges:

- Limited Options for Online Food Ordering:** There are very few apps or websites for ordering food online. For example, the Dhaweeye app provides some food delivery services, but it does not cover all restaurants and does not offer a complete menu for customers. In Mogadishu, an app called "Food Chow" allows online food ordering, but it is not available in Somaliland. This means people in Hargeisa, Burco, and other cities do not have any proper online option for ordering food.

- Issues with Existing Apps:**

- Delayed Deliveries:** Many times, customers complain that their food takes too long to arrive. This happens because the current apps are not well-organized and do not have enough drivers.
- Wrong Orders:** Since Somaliland does not have a structured address system, food orders often get mixed up or are delivered to the wrong place. This causes frustration for both customers and restaurants.
- Not Working During Bad Weather:** When it rains, the existing apps become unavailable. This is because drivers do not want to work in bad weather, and the apps cannot manage the delivery process well.

 **Limited Restaurant Choices and Menus:** The few apps that exist only work with a small number of restaurants. This means customers have very few options to choose from, making the ordering process boring and not exciting.

How Kayse Eats Solves These Problems: Kayse Eats will solve all these challenges by providing a complete online ordering system that works all the time, even during bad weather. It will connect many restaurants, such as Carro Edeg, Maansoor, Beder, and more, giving customers a wide range of options. The website will be easy to use, and orders will be delivered quickly and accurately.

The Impact on People's Lives: By introducing Kayse Eats, students like me who rely on meals cooked at home can easily order from different restaurants without worrying about delays or wrong orders. Families will have a convenient way to get meals during busy days or weekends, and local restaurants will gain more customers, helping them grow their businesses.

Problem Statement

The Main Issues Kayse Eats Will Solve:

- A. **Slow and Late Deliveries:** Many existing apps or phone orders take a long time to deliver food to customers. This is frustrating and makes people not want to order food. Kayse Eats will have a fast delivery system, ensuring that food arrives on time, fresh, and hot.
- B. **Unreliable Service in Bad Weather:** The current apps and delivery services do not work when it's raining or during bad weather. Kayse Eats will have delivery drivers who are prepared to work in all weather conditions, making sure customers always get their food.
- C. **Incorrect Orders Due to Poor Address System:** In Somaliland, there is no proper address system, so many orders end up being delivered to the wrong location. Kayse Eats will include a clear way for customers to provide directions and contact numbers, making sure drivers reach the correct place every time.
- D. **Limited Restaurant Choices:** Existing apps only have a few restaurant options, and customers often get tired of the same food choices. Kayse Eats will include many restaurants, giving customers more options to choose from, so they can always find something they like.

Why Solving These Problems Is Important: Solving these problems will make it easier for everyone to order food, save time, and have more choices. Kayse Eats will be the best way to order food online, helping busy students, workers, families, and anyone who wants to enjoy a good meal without leaving home.

Objectives

Goals of Kayse Eats:

1. **Fast and Reliable Delivery:** Make sure food reaches customers quickly, without long waiting times. This will help people who want to eat on time, like students, families, and workers.
2. **Connect Many Restaurants:** Include many restaurants in the system, so customers have more options to choose from. This will make Kayse Eats the main place to order food online.
3. **Available All the Time (24/7):** Ensure that the Kayse Eats website is always available, even during holidays, weekends, and bad weather. Customers can order whenever they want, at any time of the day or night.
4. **Simple Payment Methods:** Allow customers to pay using mobile money services like EVC Plus or Zaad, and also provide an option for card payments. These are popular and easy-to-use payment methods in Somaliland.
5. **Correct and Accurate Orders:** Make sure every order is delivered correctly by having clear directions and communication with drivers. This will avoid mistakes and ensure that customers receive exactly what they ordered.
6. **Support for Local Restaurants:** Help small restaurants grow by giving them a platform to reach more customers. By including local restaurants, Kayse Eats will support the community and help these businesses succeed.
7. **Easy-to-Use System:** Ensure the website is simple, with clear menus and easy steps for ordering. Even people who are not used to technology will find it easy to place orders.

Why These Goals Are Important: These objectives will make Kayse Eats the most reliable and easy way to order food in Somaliland, helping people save time, enjoy their favorite meals, and support local restaurants.

Scope

Current Status:

- Kayse Eats is a project I am developing as a student. Right now, it runs only on my computer (localhost) because I am still testing and building it. It's not ready for everyone to use online yet.

Structure of the Kayse Eats Website:

1. Customer Panel:

- ❖ **User Registration Page:** Customers will fill in their name, email, phone number, and password to create an account. This is simple and quick.
- ❖ **Secure Login Page:** Customers can log in safely with a password and an OTP (One-Time Password) sent to their email.
- ❖ **Restaurants Pages:** Customers can view a list of available restaurants and their locations. This makes it easy to decide where to order from.

- ❖ **Menu Pages:** Each restaurant will show all the dishes they offer, including prices and descriptions. This helps customers make informed choices.
- ❖ **Order and Pay Page:** Customers can add items to their cart, check the total cost, and choose their payment method (credit card or cash).
- ❖ **My Orders Page:** Customers can see details of their past and current orders, including items, prices, and order status (e.g., dispatched, on the way, delivered).
- ❖ **Logout Feature:** Customers can log out safely after placing their orders.

2. Admin Panel:

- ❖ **Admin Dashboard:** The admin can view important information such as the total number of restaurants, dishes, users, total orders, and earnings.
- ❖ **Admin login:** with authentication and authorization.
- ❖ **User Management:** Admins can view, edit, or delete user accounts to ensure a smooth experience for all customers.
- ❖ **Restaurant Management:** Admins can add or update restaurant information and menus to keep the content current.
- ❖ **Order Management:** Admins can change order statuses and manage any issues, ensuring customers receive their food on time.

Future Plans:

- Once I finish testing, I plan to make Kayse Eats available online. This way, real people in Somaliland can use it to order food, and local restaurants can get more customers.

How It Will Handle More Users (Scalability):

- **Ready for More Users:** I designed Kayse Eats to handle more customers and restaurants as it grows. Even when many people use it at the same time, it will work smoothly without slowing down.
- **Regular Data Backups:** I will make sure all customer and restaurant data is backed up regularly to keep information safe and prevent data loss.
- **Improving Speed and Use:** I want to make sure that Kayse Eats is always fast and easy to use, even if many people use it at the same time. This is important in Somaliland, where internet speed might be slower for some people.
- **Listening to Feedback:** After launching it online, I plan to collect feedback from users and restaurant owners. This will help me improve the system and make it more useful for everyone.

Methodology & Procedure

How We Developed Kayse Eats:

June - Planning & Research:

We began by conducting research to understand the challenges people face in ordering food. This included talking to restaurant owners, customers, and students. We gathered information on what features people would like to see in an online food ordering system.

July - Designing & Building:

After research, we created a design for the website, making sure it was user-friendly and visually appealing. We used technologies such as PHP, MySQL, HTML, CSS, Bootstrap, and JavaScript to build the site. These tools allowed us to create a responsive website that works well on both computers and mobile devices.

August - Testing:

We tested each part of the website to find any bugs or issues. We made sure the registration process, order placing, and payment options worked perfectly. We also checked how the website performed with slow internet speeds, which is common in many areas of Somaliland.

September - Final Adjustments:

After testing, we collected feedback from users such as students, friends, some teachers. Based on their suggestions, we improved the website to make it even better. We ensured that the user experience was smooth and enjoyable.

October - Launch & Presentation:

The final step was to prepare for the official launch. We made the website live and set up everything for the presentation. This involved creating promotional materials and explaining how Kayse Eats works.

Expected Outcome

What We Hope to Achieve with Kayse Eats:

- Easy and Quick Food Ordering:** People in Somaliland will find it very easy to order food online using Kayse Eats. Even if they have never used a website before, they will understand how to place their orders.
- Increased Sales for Restaurants:** By connecting many restaurants to the platform, they will have more customers and earn more money. This will help local businesses grow and thrive.
- Reliable and Trustworthy Service:** Customers will trust Kayse Eats because it will be available all the time, even when other services don't work. This will encourage more people to use the service.

4. **Happy Customers:** With correct orders and timely deliveries, customers will be satisfied with their meals, leading to repeat orders and positive word-of-mouth.
5. **Support for Local Economies:** By helping local restaurants, Kayse Eats contributes to the economic growth of Somaliland, creating more job opportunities and helping businesses succeed.

Why This Matters: This project is important because it will transform how people in Somaliland order food. It will save time, increase choices, and support local restaurants, making life better for everyone.

Schedule

Date Activity

| | |
|-----------------------------|---|
| | Research & Planning: We gathered information about what people needed for food delivery in Somaliland. I looked at other services and identified what they were missing. I created a plan for how Kayse Eats would work. |
| June 13 - June 26 | |
| June 27 - July 10 | Design the Website: We designed simple layouts for the website. I planned how the customer and admin pages would look, ensuring they were easy to use. |
| July 11 - August 1 | Start Coding: We began writing the code for Kayse Eats. I set up the customer and admin pages and made sure the basic features, like signing up and logging in, were working. |
| August 2 - August 14 | Test Features: We checked each part of the website to ensure it worked correctly. I tested if customers could see restaurants, check menus, and place orders without any problems. |
| August 15 - August 28 | Fix Problems: We found and fixed any issues from testing. I also improved the design based on user feedback. |
| August 29 - September 12 | Final Testing: We tested the entire website again to make sure everything worked well and was ready for users. |
| September 13 - September 25 | Prepare for Launch: We are getting everything ready to show Kayse Eats to others. I am creating a simple presentation to explain the project and its features. |
| September 26 - October 3 | Final Changes: We will make any last-minute changes based on feedback. I will practice explaining how to use the website for my presentation. |
| October 10 | Project Presentation: We will present Kayse Eats, explaining its features and how it helps people order food in Somaliland. |

Why This Timeline Is Effective: This timeline clearly outlines the steps taken to develop Kayse Eats. Each phase was planned to ensure that everything was completed correctly and on time, leading to a successful project.

Project Overview for Kayse Eats

2.1 Background Information

Kayse Eats is an online food ordering service made for the people of Somaliland. This project aims to help customers easily order food from many different restaurants. Right now, when someone wants to eat from a restaurant, they usually have to call the restaurant or go there in person. This method can take a long time and can be very inconvenient, especially for busy people.

In Somaliland, sharing meals is very important in our culture. Families and friends love to gather and enjoy good food together, especially on weekends and during special events like weddings or holidays. However, the current ways of ordering food make this difficult. Many people are busy with work, school, or other responsibilities and do not always have time to cook. Because of this, they need a simple and fast way to get food delivered to their homes.

Kayse Eats will change how people order food. It will offer a simple and friendly website where customers can look at menus, choose their favorite dishes, and place orders from their homes. This service will save time and make ordering food more enjoyable for everyone. It will also help local restaurants reach more customers and grow their businesses.

As technology becomes more common in Somaliland, more people are using smartphones and the internet. Kayse Eats will help make it easier for people to enjoy their favorite foods while also supporting local restaurants. This project aims to improve daily life for many people in Somaliland by providing a modern way to order food, which can be a fun and social experience.

2.2 Current System or Domain Analysis

Right now, the main way people in Somaliland order food is by calling restaurants. This method has many problems:

1. **Slow Delivery:** When customers call a restaurant to place an order, it can take a long time for the food to be prepared and delivered. If the restaurant is busy, they might have to wait even longer than expected. Sometimes, customers are left wondering when their food will arrive. This can be very frustrating.
2. **Wrong Orders:** There are many times when food is delivered incorrectly. This often happens because there is no clear address system in many areas. Delivery drivers may struggle to find the right location, which leads to confusion. For example, if someone orders chicken and rice, but the driver delivers fish instead, it can upset the customer.
3. **Limited Choices:** Many restaurants do not offer delivery services at all. This limits the options for customers who want to order food. When there are only a few choices, people may not find what they want or get tired of eating the same meals over and over.
4. **Weather Problems:** During bad weather, such as heavy rain, deliveries can become even more difficult. Some drivers may refuse to work during rain or storms, which means customers might not get their food at all when they want it most.

5. **Dhaweeye App:** The **Dhaweeye** app provides some delivery services, but it does not focus only on food. It delivers many types of items, including groceries and packages. Because of this, it may not provide the best experience for customers looking to order food. Customers might find it hard to navigate the app to find the specific meals they want. This can make ordering food feel complicated and frustrating.

These challenges show that there is a clear need for a better solution. Kayse Eats will address these issues by creating an online platform that connects many restaurants, allowing customers to place orders quickly and easily. This will improve the overall food ordering experience in Somaliland, making it simpler and more enjoyable for everyone.

2.3 Comparing Similar Systems

When we look at successful food delivery systems in other countries, we see many examples like Uber Eats, Zomato, and Deliveroo. These platforms allow customers to order food from a variety of restaurants and have it delivered to their homes. They have some important features that make them popular:

- **User-Friendly Interface:** These platforms are designed to be easy to use. Customers can browse menus and select food with just a few clicks.
- **Track Orders:** Customers can see where their food is and when it will arrive. This feature helps reduce uncertainty and keeps customers informed about their orders.
- **Payment Options:** Many of these systems allow customers to pay online using credit cards or mobile payment methods. This makes the process quick and secure, so customers do not have to worry about handling cash.

Dhaweeye offers some delivery services, but it is not a dedicated food delivery app. Here are some differences between Kayse Eats and Dhaweeye:

- **Focus on Food:** Kayse Eats will focus only on food delivery, making it easier for customers to find meals. In contrast, Dhaweeye delivers many types of items, which can make it harder for customers to find the food they want.
- **Limited Restaurant Selection:** Dhaweeye does not list many local restaurants, which reduces the choices available to customers. This is a significant drawback for those who want to try different foods.
- **Less Specialized Service:** Since Dhaweeye serves many purposes, it may not provide the same quality of service for food ordering as Kayse Eats will offer. Customers may prefer a platform that specializes in food delivery, as it will be easier and more enjoyable to use.

Kayse Eats will have similar features to these platforms but will be tailored for Somaliland. It will focus on food delivery and ensure that customers can easily find what they want. This will help improve the food ordering experience for everyone.

2.4 Compare and Contrast the Existing System Technology with the Proposed Solution

The current way of ordering food in Somaliland relies mostly on phone calls. This method has many limitations:

- **No Central System:** Each restaurant manages its own orders separately. This means customers cannot see multiple menus in one place, which limits their choices.
- **Ineffective Communication:** Phone orders can lead to misunderstandings, which can result in wrong deliveries and unhappy customers.
- **No Order Tracking:** Customers have no way of knowing when their food will arrive. They often have to wait without any updates on their order status, which can be very frustrating.
- **Dhaweeye's Limitations:** While Dhaweeye offers some delivery services, it is not designed specifically for food. This can make it harder for customers to find meals and receive good service.

In contrast, Kayse Eats will introduce modern technology to improve the food ordering process:

1. **Online Ordering:** Customers can view all available restaurants and their menus on a single platform. They can easily choose what they want and place their orders online. This means no more waiting on the phone.
2. **Better Communication:** The platform will allow customers to provide specific instructions for their orders. For example, if someone wants extra spicy sauce or no onions, they can easily write that down. This helps reduce mistakes.
3. **Real-Time Tracking:** Kayse Eats will offer real-time tracking of food deliveries. Customers can see where their order is and when it is expected to arrive. This feature will help them know when to expect their meal and reduce waiting time.
4. **GPS Navigation:** The system will use GPS technology to help drivers find the correct delivery addresses. This will be especially useful in areas with unclear street names or addresses. This means food will be delivered faster and to the right place.
5. **Convenient Payment Options:** Kayse Eats will integrate with popular mobile payment systems like EVC Plus and Zaad. This will make it easy for customers to pay for their food securely and quickly, without needing cash.

By using these technologies, Kayse Eats will create a better experience for customers and restaurants alike. It will help solve many of the problems that the current system faces.

2.5 How the Proposed Solution Fits into the Overall Strategic Objectives of the Organization

Kayse Eats is designed to achieve important goals for both customers and local restaurants. For customers, the platform will offer several key benefits:

- **Convenience:** Customers will be able to order food quickly and easily from the comfort of their homes. This is particularly valuable for busy families or individuals who do not have time to cook.
- **Increased Choices:** By connecting multiple restaurants, Kayse Eats will provide a wider variety of food options. Customers will not be limited to just a few places. They will be able to explore different cuisines and try new dishes.
- **Reliable Service:** The system aims to improve delivery reliability through GPS navigation and better communication. This will lead to faster and more accurate deliveries, giving customers confidence in their orders.

For local restaurants, Kayse Eats offers several advantages:

- **Wider Reach:** Many small restaurants struggle to attract new customers. By joining Kayse Eats, they can advertise their offerings to a larger audience, increasing their visibility.
- **Higher Sales:** With more customers able to order from them, restaurants can expect an increase in sales. This growth is essential for sustaining their businesses.
- **Simplified Operations:** The platform will simplify the order management process for restaurants. They will have a clear way to track orders and communicate with customers, allowing them to focus more on cooking and providing excellent service.

Overall, Kayse Eats supports the goal of improving customer satisfaction and helping local businesses grow. It aligns with the aim of fostering economic growth in Somaliland and encouraging the use of technology in everyday life.

2.6 How Kayse Eats Will Work with Other Systems

To provide the best service, Kayse Eats will work closely with existing systems in Somaliland. Here are some ways the platform will connect with other services:

1. **Mobile Payment Systems:** Kayse Eats will partner with popular mobile payment services like EVC Plus and Zaad. These systems allow customers to pay for their orders easily and securely. This is important in a cash-based economy where many people prefer to use mobile money.
2. **Local Delivery Services:** Kayse Eats will collaborate with local drivers who know the area well. This will help ensure that deliveries are made quickly and accurately. The platform will use GPS to help drivers find the best routes to customers' homes, even in places where the roads may not be well marked.
3. **Communication Tools:** The platform will include features that allow customers to send messages about their orders. For example, if a customer has a special request or wants to change something, they can communicate that easily. This will help reduce mistakes and ensure that customers get their food as they ordered.

By working with these existing systems, Kayse Eats will create a smooth experience for users. This will make ordering food easier for customers and more efficient for restaurants, allowing everyone to benefit from the service.

2.7 Similarities of the Systems

Kayse Eats shares many features with other popular food delivery platforms. Here are some similarities:

- **Easy to Use:** Like Uber Eats and Zomato, Kayse Eats will have a simple and user-friendly interface. Customers will be able to browse through menus and place orders without any difficulty.
- **Real-Time Order Tracking:** Just like other food delivery apps, Kayse Eats will allow customers to track their orders in real-time. They will receive updates on where their food is and when it will arrive.
- **Variety of Restaurant Choices:** Kayse Eats will connect many restaurants, providing customers with a range of options to choose from, similar to what other delivery apps offer.
- **Multiple Payment Methods:** Kayse Eats will allow customers to pay using various methods, including mobile payments. This will make transactions convenient and secure.

While Kayse Eats shares these features with other platforms, it is specifically designed to meet the needs of the Somali market. By focusing on local challenges and conditions, Kayse Eats will provide a unique and valuable service that stands out from other food delivery systems.

CHAPTER 3

SYSTEMANALYSIS FOR KAYSE EATS

1. Existing System Description

Current Situation: Currently, food ordering in Somaliland is largely manual. Customers must either call restaurants or physically visit them to place an order. This traditional method of ordering presents several significant issues:

- **Delayed Deliveries:** Without an organized system, restaurants manually take orders over the phone, making it difficult to manage peak times effectively. This often leads to longer delivery times, especially during busy periods.
- **Incorrect Orders Due to Poor Address System:** Somaliland lacks a structured address system, making it challenging for drivers to locate delivery addresses precisely. As a result, orders are sometimes delivered to the wrong address, frustrating both customers and restaurant staff.
- **Limited Choices:** Many restaurants do not offer delivery services at all, limiting customer options and making it difficult for them to try different foods.
- **Weather-Related Problems:** In bad weather, such as heavy rain, delivery services become even less reliable. Some drivers may refuse to work during rain or storms, which means customers might not get their food at all when they want it most.

How Kayse Eats Will Improve This: Kayse Eats will introduce a fully online, streamlined food ordering system, allowing customers to view multiple restaurants in one place, order food easily, and track their orders in real-time. By providing a single platform for food ordering, Kayse Eats will:

- Ensure faster, more reliable deliveries.
- Use GPS and real-time tracking to reduce address errors.
- Offer more choices by partnering with a wide range of restaurants

2. System Requirements

This section defines what the Kayse Eats system needs to operate successfully. Requirements are divided into **Functional** and **Non-functional** categories to address both what the system must do and how it should perform.

2.1 Functional Requirements

The functional requirements describe what each user type—admin and customer—can accomplish within the Kayse Eats system.

Customer Panel Functional Requirements

The **Customer Panel** provides all the tools a user needs to explore restaurants, place orders, and track their delivery. Below are the functional requirements:

1. User Registration and Login:

- **User Registration Page:** Customers will fill in their name, email, phone number, and password to create an account. The page is designed for ease of use, allowing quick and straightforward registration.
- **Secure Login Page:** After registration, customers log in with a secure password. An additional OTP (One-Time Password) is sent to their email to verify identity, enhancing security.

2. Restaurant Discovery:

- **Restaurants Page:** Customers can view a list of available restaurants, with options to filter by location, cuisine, and rating. This organization helps customers decide where to order from quickly and easily.
- **Menu Pages:** Each restaurant page will display all available dishes, including descriptions, prices, and customization options (e.g., spice level or special requests). This detail allows customers to make informed choices.

3. Order Placement:

- **Order and Pay Page:** Customers add items to their cart, where they can review their order details. The total cost is calculated, and they can select a payment method (mobile payments like EVC Plus, Zaad, or credit card).
- **Customization Options:** Customers can customize orders by specifying preferences such as extra toppings or special instructions, helping reduce miscommunication and ensuring satisfaction.

4. Order Tracking and Status Updates:

- **Order Tracking:** Real-time order tracking with GPS allows customers to monitor their delivery, showing estimated arrival times and live driver location.
- **Order Status Notifications:** Customers receive notifications at key points—order confirmation, preparation start, out for delivery, and arrival—via email and app notifications.

5. Order History and Reordering:

- **My Orders Page:** Customers can access past and current orders, including details like order date, items, prices, and status (e.g., dispatched, on the way, delivered).
- **Quick Reordering:** Past orders can be reordered with a single click, simplifying the process for frequent users who prefer specific meals.

6. Account Management:

- **Profile Settings:** Customers can update their personal information, such as name, email, password, and payment details. This keeps the information up-to-date and provides flexibility for customers.
- **Logout Feature:** After placing orders, customers can log out safely, ensuring privacy and security.

7. Customer Support:

- **Support Chat:** Customers can connect with support staff through an in-app chat feature for any issues related to orders, delivery, or technical difficulties.

Admin Panel Functional Requirements

The **Admin Panel** is designed to help Kayse Eats administrators manage restaurants, monitor orders, and oversee customer accounts efficiently. Below are its functional requirements:

1. Admin Login and Access Control:

- **Admin Login:** Only authorized personnel can access the Admin Panel. Admins log in with secure credentials, and the system includes Two-Factor Authentication (2FA) to prevent unauthorized access.
- 2. Restaurant Management:**
- **Adding and Updating Restaurants:** Admins can add new restaurants and enter details like name, location, and contact information. Admins can also update restaurant details to keep the menu and restaurant information current.
 - **Menu Management:** Admins have control over restaurant menus, allowing them to add, modify, or remove items, ensuring customers always see accurate and up-to-date options.
- 3. Order Monitoring and Management:**
- **Order Dashboard:** The dashboard displays all ongoing orders in real-time, helping admins monitor status and detect any potential delays.
 - **Order Modification and Cancellation:** Admins can modify or cancel orders upon customer request or due to restaurant constraints, ensuring customer satisfaction and accurate service.
- 4. User Account Management:**
- **Customer and Driver Accounts:** Admins can view, edit, or delete customer and driver accounts to manage the user base effectively.
 - **Verification and Deactivation:** Admins verify account details as needed, especially for drivers, ensuring the platform maintains trusted users.
- 5. Analytics and Reporting:**
- **Sales and Order Reports:** The system generates daily, weekly, and monthly reports on total orders, revenue, top-performing restaurants, and delivery times, helping admins track growth and performance.
 - **Customer Feedback Analysis:** By reviewing customer feedback, admins can identify areas for improvement, enhancing the customer experience.
- 6. Customer Support:**
- **Issue Resolution:** A support dashboard helps admins monitor complaints and resolve issues efficiently.

2.2 Non-Functional Requirements

Non-functional requirements specify how the system should perform to provide a seamless and secure experience. Key areas include:

1. **Security:** The system should have strong data encryption to secure customer transactions, personal details, and payment information, meeting data privacy laws.
2. **Reliability:** Kayse Eats should be accessible 24/7, with minimal downtime to meet customer needs consistently.
3. **Performance:** Kayse Eats should load quickly and operate smoothly, even on slower internet connections, which are common in Somaliland.
4. **Scalability:** Kayse Eats should be able to handle growth in users and restaurants as Kayse Eats expands, without slowing down or experiencing technical issues.
5. **Usability:** Kayse Eats should be intuitive and user-friendly so that customers and admins can navigate it easily without requiring technical expertise.

3. Hardware and Software Requirements

3.2.1 Hardware Requirements

To run the Kayse Eats platform efficiently and provide a seamless user experience, the following hardware specifications are required:

1. Processor:

- **Recommended:** Intel® Core i5 or higher.
- **Purpose:** A powerful processor like Intel Core i5 or higher ensures fast and smooth processing of multiple customer requests, complex data queries, and real-time order updates without lag. This enables the system to handle high traffic and simultaneous transactions efficiently.

2. RAM:

- **Recommended:** 8 GB or more.
- **Purpose:** At least 8 GB of RAM is necessary to manage multiple tasks simultaneously, especially for processing data from various modules like customer accounts, restaurant menus, and order histories. Sufficient memory prevents the system from slowing down during peak usage times, offering a smooth experience for both customers and admins.

3. Storage:

- **Recommended:** Minimum 20 GB of free storage space.
- **Purpose:** Adequate storage is needed to save essential files such as system logs, backups, database files, and application updates. This storage allocation ensures that the platform can retain historical data and log files for troubleshooting, data recovery, and future analysis, which enhances the reliability and continuity of service.

4. Network:

- **Recommended:** High-speed, stable internet connection with low latency.
- **Purpose:** Reliable and fast internet is crucial for real-time features like order tracking and live notifications. A stable connection minimizes delays in data transmission, ensuring that customers receive timely updates and that the admin panel functions without interruptions.

5. Display:

- **Recommended:** A 14-inch or larger monitor with a minimum resolution of 1080p.
- **Purpose:** For administrators, a large display makes it easier to monitor system dashboards, manage user accounts, and oversee multiple restaurant orders simultaneously, which is particularly important for efficiency in high-volume environments.

3.2.2 Software Requirements

Kayse Eats also relies on a robust software stack to provide reliable functionality and compatibility across various devices. The following software components are essential:

1. Operating System:

- **Supported:** Windows, macOS, and Android.

- **Purpose:** The platform is compatible with all major operating systems to provide flexibility for development, deployment, and usage. Ensuring compatibility across OSs means that developers, admins, and end-users can access the system on different devices and environments without compatibility issues.

2. Database:

- **Recommended:** MySQL or PostgreSQL.
- **Purpose:** MySQL and PostgreSQL are reliable databases that store critical data, including customer information, restaurant details, menu items, and order histories. These databases offer high performance, scalability, and security, making them suitable for handling large volumes of transactions and ensuring data integrity over time.

3. Programming Languages:

- **Backend:** PHP for backend development.
- **Frontend:** HTML, CSS, and JavaScript for designing the user interface.
- **Purpose:** PHP is a popular, secure, and scalable language ideal for handling server-side operations and managing requests from the frontend. HTML, CSS, and JavaScript work together on the frontend to create an intuitive and interactive user experience, allowing customers to navigate menus and place orders effortlessly.

4. Framework:

- **Backend Framework:** Laravel.
- **Frontend Framework:** React.
- **Purpose:** Laravel is a robust framework that provides a structured and secure foundation for backend development, simplifying tasks like routing, authentication, and data management. React enhances the frontend with its component-based architecture, allowing for a responsive, dynamic, and user-friendly interface that adapts smoothly across devices.

5. Integrated Development Environment (IDE):

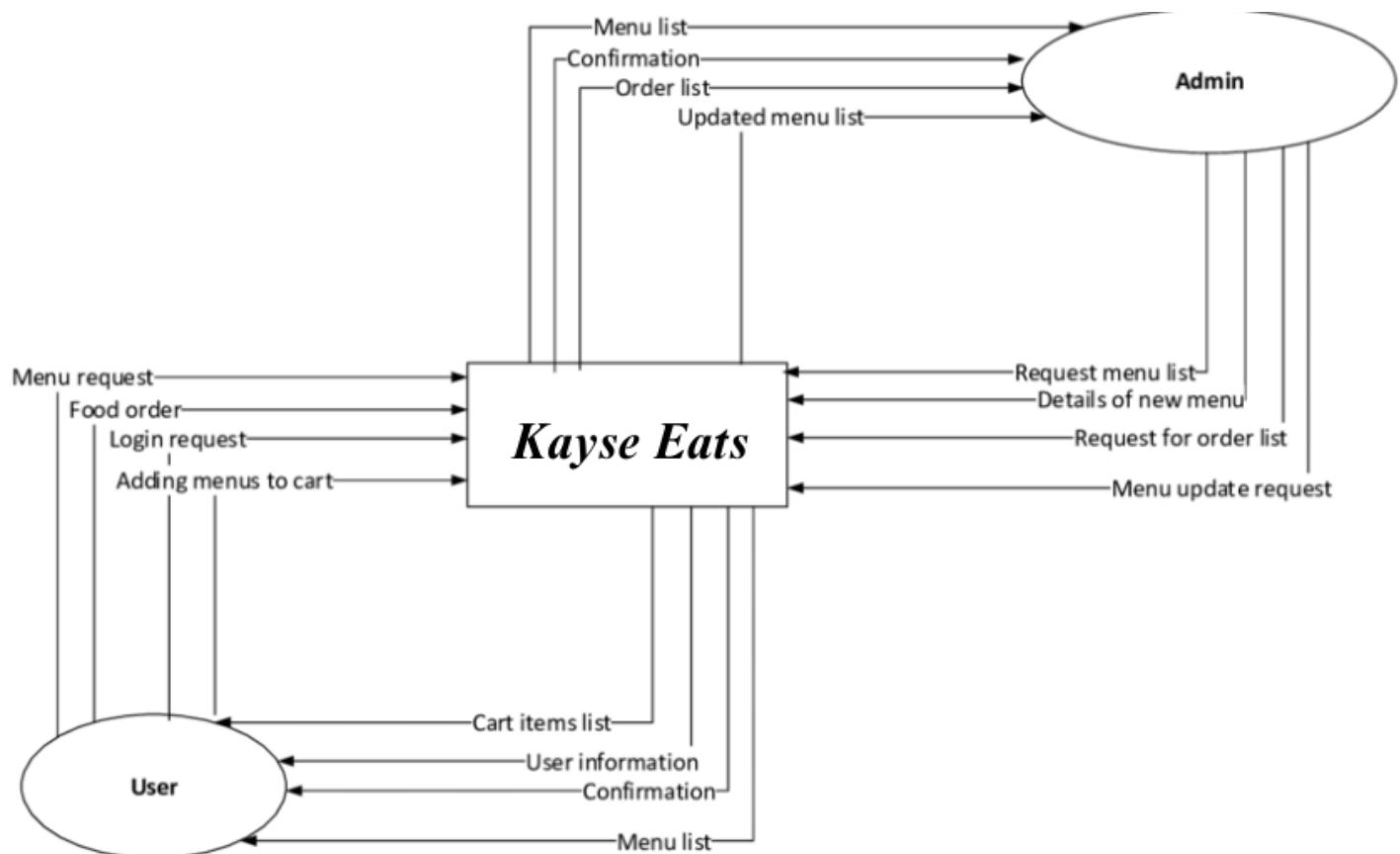
- **Recommended:** Visual Studio Code.
- **Purpose:** Visual Studio Code is a versatile and widely used IDE that helps developers write, test, and debug code efficiently. Its support for extensions and integrations with Git makes it ideal for managing large-scale projects like Kayse Eats, where version control and code quality are critical.

6. Browser Compatibility:

- **Supported Browsers:** Chrome, Firefox, Safari, and Edge.
- **Purpose:** Ensuring the website works smoothly on all major browsers allows customers to access the platform from their preferred browser without functionality issues. Consistent performance across browsers enhances accessibility and ensures a uniform user experience.

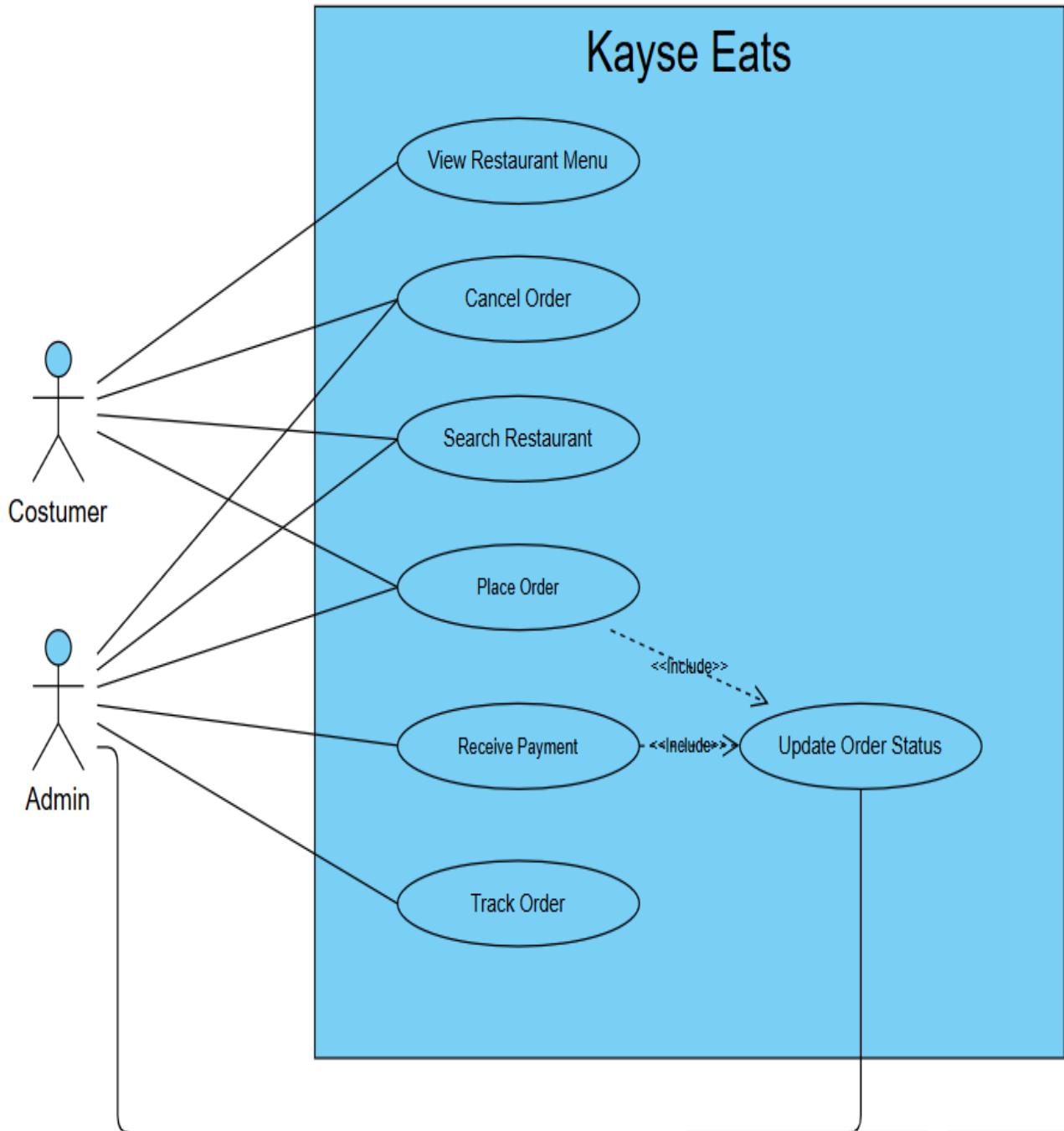
1. Context Diagram:

Context Diagram: A visual representation that shows the boundaries of a system, illustrating its interactions with external entities and the flow of information between them.



2. Use Case Diagram

- Represents primary system functions and user interactions for both customers and admins.
- Highlights essential actions like placing orders, tracking deliveries, and updating order status.
- Simplifies understanding of system capabilities and user roles.



3. Use Case Descriptions

1. Login

Describes how a user (customer or admin) securely logs into the Kayse Eats system to access their account and features.

2. View Restaurant Menu

Describes how a customer browses a selected restaurant's menu on Kayse Eats to view available items and prices.

3. Search Restaurant

Describes how a customer searches for restaurants within Kayse Eats using criteria such as location, cuisine type, or restaurant name.

4. Place Order

Describes how a customer selects items from a restaurant's menu on Kayse Eats and submits an order for processing.

5. Cancel Order

Describes how a customer cancels an order on Kayse Eats before it is processed by the restaurant.

6. Receive Payment

Describes how the Kayse Eats system securely processes payment for a customer's order.

7. Track Order

Describes how a customer views the current status of their order on Kayse Eats, including preparation and delivery stages.

8. Update Order Status

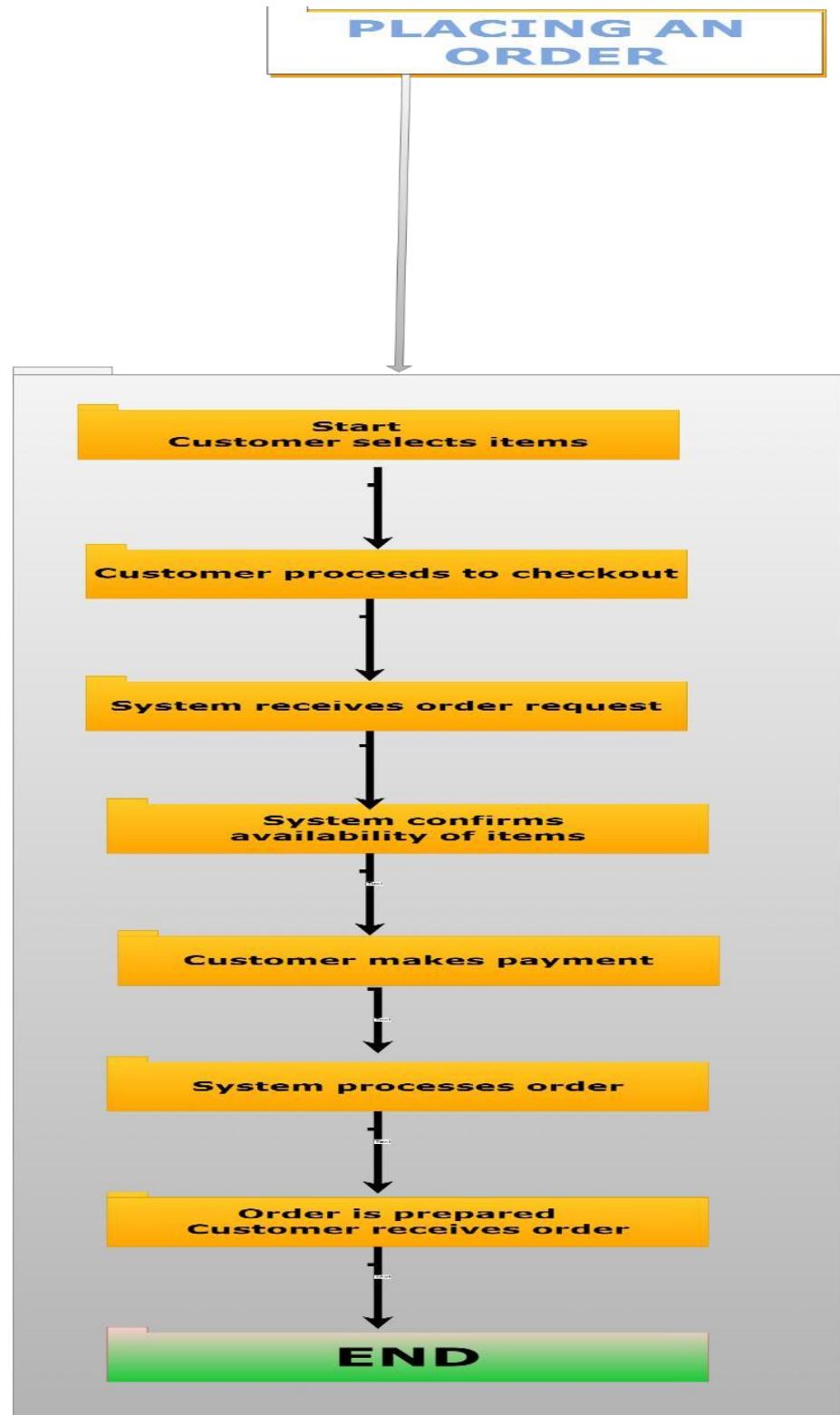
Describes how an admin on Kayse Eats updates the status of an order, such as marking it as "Preparing," "Out for Delivery," or "Delivered."

9. Logout

Describes how the Kayse Eats system securely logs the user out, ending their session.

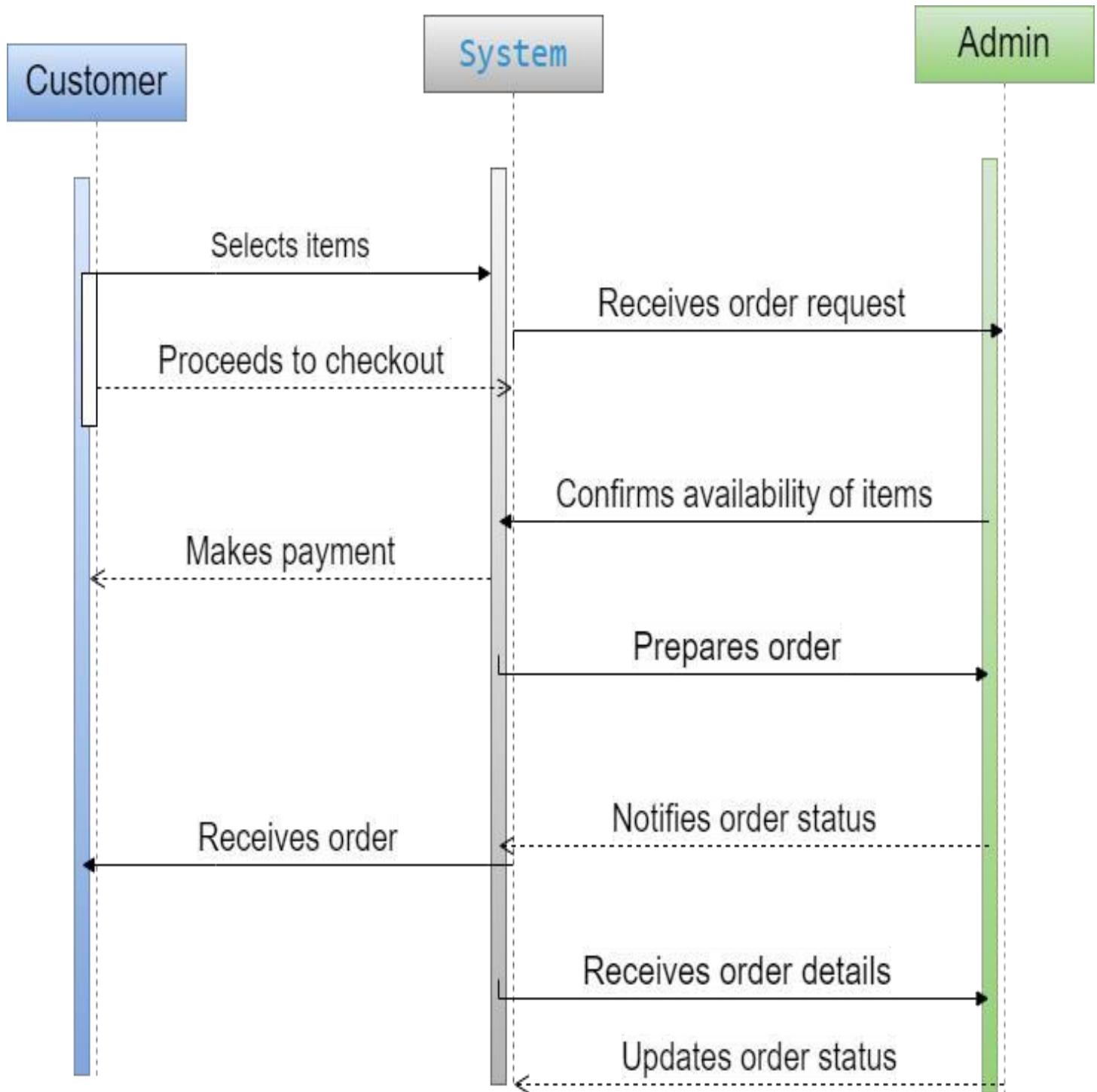
4. Activity Diagram

Represent the step-by-step process from order placement to delivery.



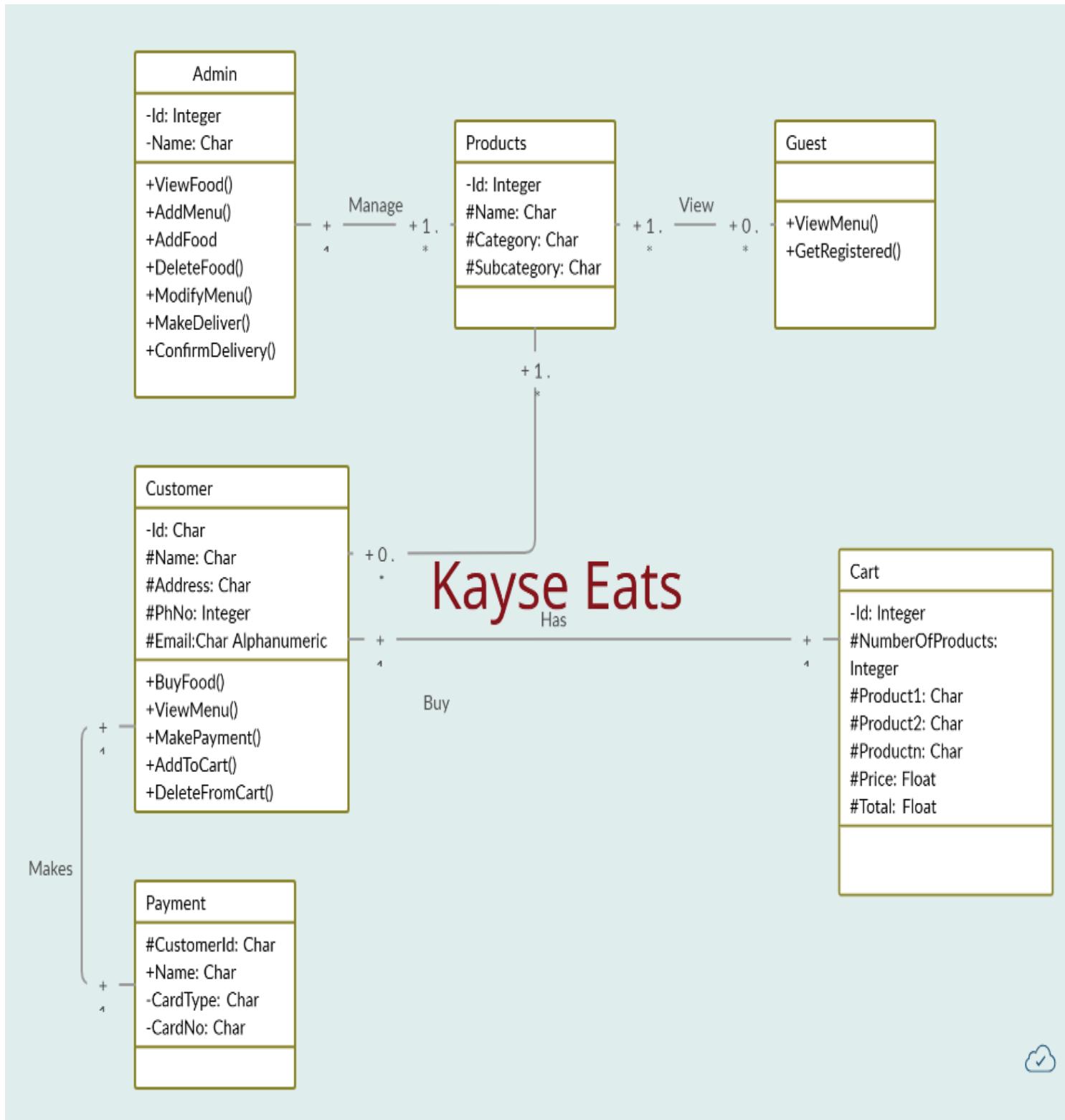
5.Sequence Diagram:

Map out interactions between system components during an order.



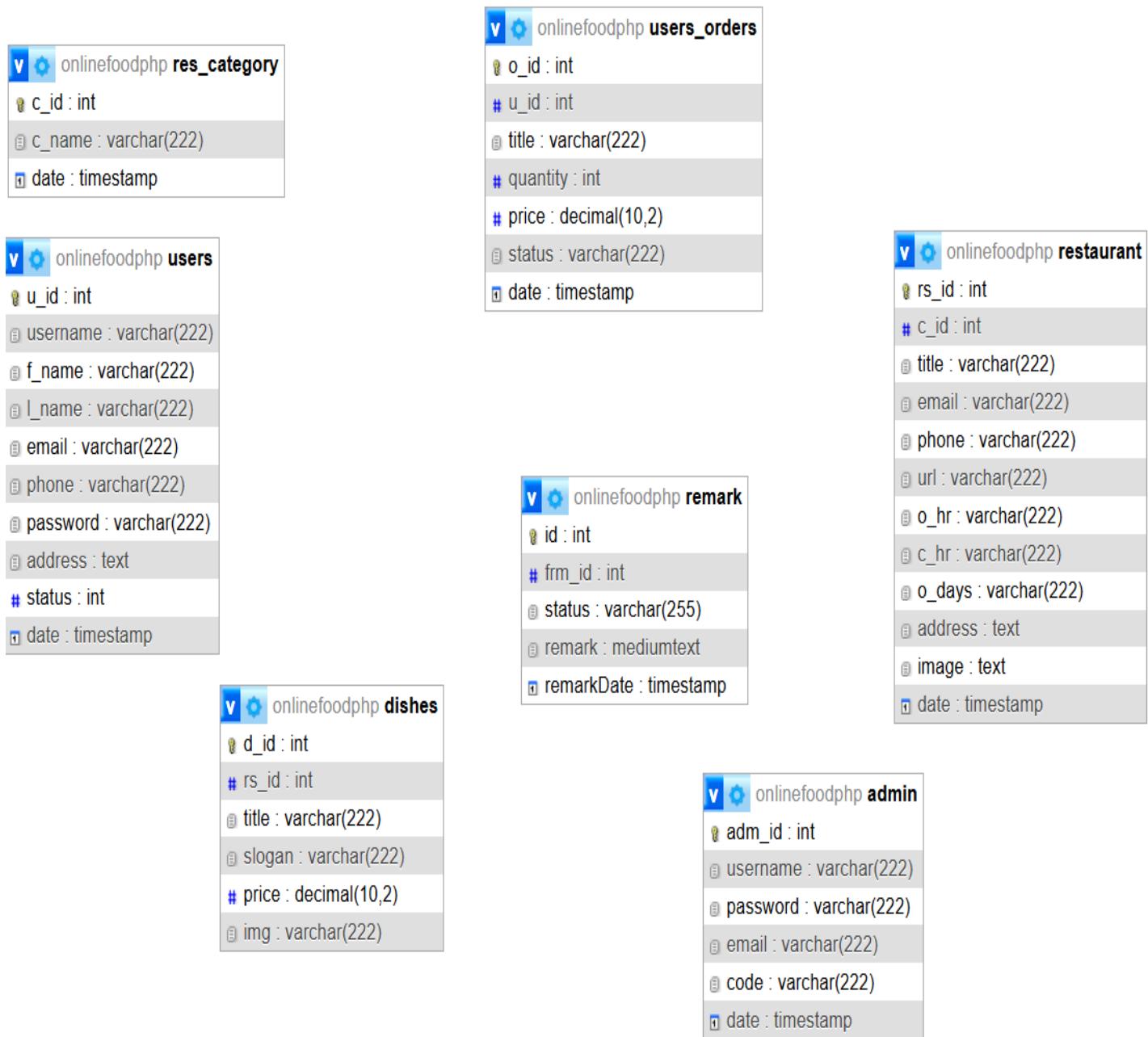
6. Class Diagram:

Display key classes, such as Customer, Order, cart, payment, and Admin.



7.ERD Diagram:

Visualize entities and their relationships within the database.



CHAPTER 4

SYSTEM DESIGN FOR KAYSE EATS

Table of Contents

1. **Introduction**
 - 1.1. Overview of the System
2. **System Architecture Design**
 - 2.1. Presentation Layer
 - 2.2. Application Layer
 - 2.3. Data Layer
3. **User Interface Design**
 - 3.1. **Customer Panel Pages**
 - 3.1.1. About
 - 3.1.2. Checkout
 - 3.1.3. Delete Orders
 - 3.1.4. Dishes
 - 3.1.5. FAQ
 - 3.1.6. Forgot Password
 - 3.1.7. Index
 - 3.1.8. Login
 - 3.1.9. Logout
 - 3.1.10. Map
 - 3.1.11. Product Action
 - 3.1.12. Registration
 - 3.1.13. Restaurants
 - 3.1.14. Your Orders
 - 3.2. **Admin Panel Pages**
 - 3.2.1. Admin Panel Dashboard
 - 3.2.2. Filter Food Items
 - 3.2.3. Filter Restaurants
 - 3.2.4. Order Foods
 - 3.2.5. Add to Cart System
 - 3.2.6. View Personal Orders
 - 3.2.7. Restaurant Management
 - 3.2.8. Food Category Management
 - 3.2.9. Food Menu Management
 - 3.2.10. Manage Food Orders
 - 3.2.11. Cancel Orders
 - 3.2.12. Customer Management
 - 3.2.13. Forgot Password
4. **Database Design**
 - 4.1. Overview of Database Structure
 - 4.2. Core Database Tables
 - 4.3. Relationships Between Tables
 - 4.4. Mapping Screens to Tables

4.1 Introduction

This chapter focuses on the design aspects of the **Kayse Eats** project, addressing its **System Architecture**, **User Interface Design**, and **Database Design**. The design ensures that all components are aligned, functional, and structured to meet the academic requirements and practical standards of an online food ordering system.

4.2 System Architecture Design

The system architecture for this project follows a **three-tier architecture** to ensure scalability, performance, and maintainability.

1. Presentation Tier (Frontend):

- This layer includes the **Customer Panel** and **Admin Panel**, which are the primary interfaces for users to interact with the system.
- Built using **HTML, CSS, JavaScript**, and **Bootstrap** for responsiveness.

2. Application Tier (Backend):

- Handles the business logic and connects the frontend with the database.
- All interactions, such as user authentication, order processing, and data updates, are executed here.
- Developed using **PHP**, utilizing REST principles for cleaner architecture.

3. Data Tier (Database):

- The **MySQL database** contains structured tables to store all critical data, including users, dishes, restaurants, and orders.
- Secure data operations are ensured through optimized queries and robust database management techniques.

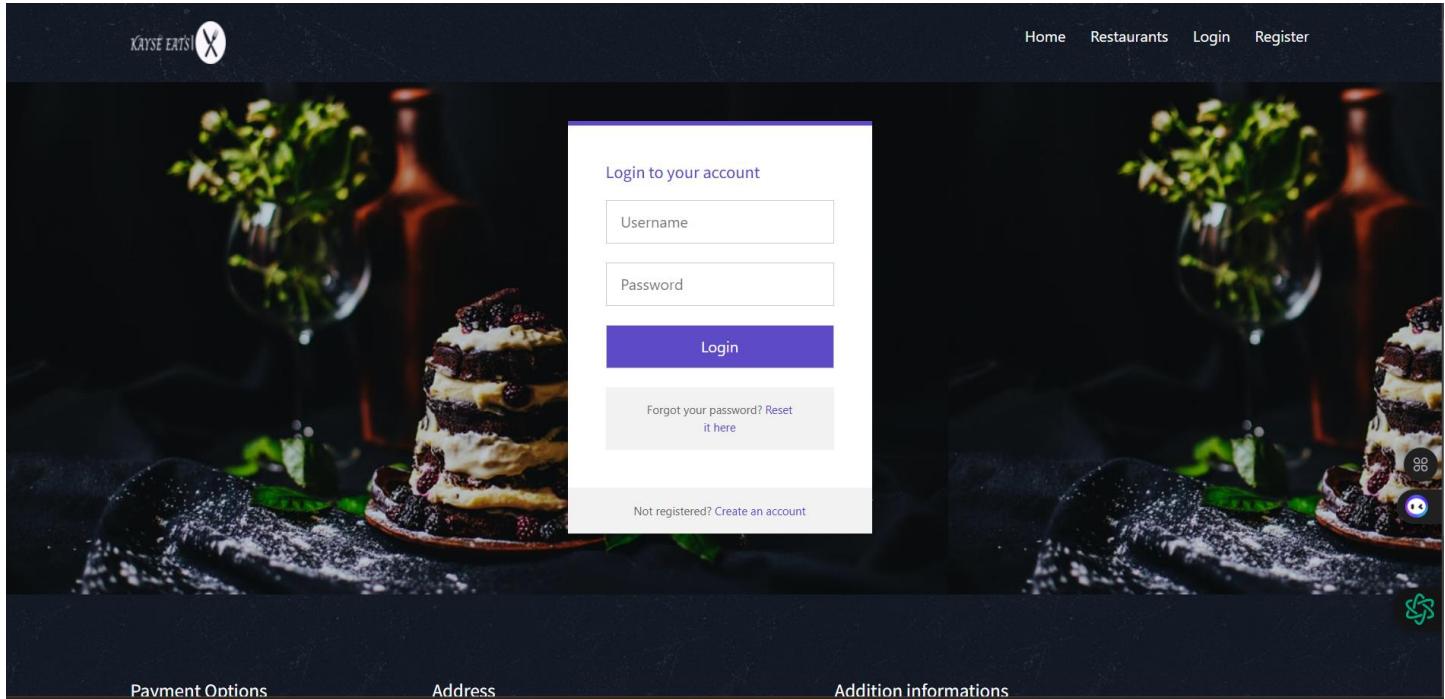
The architecture ensures modularity and allows for seamless future enhancements, like integrating APIs or adding new features.

4.3 User Interface Design

The user interfaces are designed to provide a seamless experience for both customers and administrators. Each interface aligns with the database tables and ensures functionality based on user roles.

4.3.1 Customer Panel Pages

Login: Secure login page for customers.



Index: The homepage with a welcoming interface and navigation options.



About: Provides information about the platform's purpose and features.

The screenshot shows the 'About' page of the Kayse Eats mobile application. At the top, there is a header with the Kayse Eats logo and a 'Call 8000' button. Below the header, the title 'About Kayse Eats' is displayed. A detailed description follows, explaining that Kayse Eats is an online food ordering platform developed by Mohamed Kayse and Abdirizak Dek, located in Hargeisa city, Somaliland. It connects food enthusiasts with local restaurants, offers a seamless dining experience, supports local businesses, and showcases the rich culinary diversity of Somaliland. The platform features real-time order tracking, secure payment options, and responsive customer support. The text concludes with an invitation to discover the vibrant flavors of Somaliland's food scene.

The screenshot shows the 'Our Services' section of the Kayse Eats mobile application. The title 'Our Services' is centered at the top. Below it, three services are listed in boxes: 'Menu Browsing' (Browse through a variety of restaurant menus, exploring different cuisines, dishes, and prices.), 'Order Placement' (Place orders directly through our website or mobile app, selecting your desired items and specifying any special requests.), and 'Delivery Services' (Enjoy prompt and reliable delivery to your doorstep, ensuring that your food arrives fresh and delicious.).

The screenshot shows the 'How It Works' section of the Kayse Eats mobile application. The title 'How It Works' is centered at the top. Below it, five steps are shown in a sequence: Step 1: Sign Up (Create an account on our platform to get started.), Step 2: Browse Menus (Browse through a variety of restaurant menus and select your desired items.), Step 3: Place Your Order (Place your order and specify any special requests.), Step 4: Track Your Order (Track the status of your order in real-time.), and Step 5: Enjoy Your Meal! (Receive your delicious meal and enjoy!).

The screenshot shows the 'Our Team' section of the Kayse Eats mobile application. The title 'Our Team' is centered at the top. Three team members are introduced with their portraits and roles: Mohamed Kayse Hassan (Co-founder & CEO), Abdirizak Dek Mahad (Lead Developer), and jees (Front-end Developer). Below each profile, a brief description of their roles and contributions is provided.

Mohamed Kayse Hassan
Co-founder & CEO
Mohamed is passionate about entrepreneurship and technology. With years of experience in business management, he leads our team with vision and dedication.

Abdirizak Dek Mahad
Lead Developer
Abdirizak is a coding ninja with a knack for problem-solving. He loves diving deep into complex algorithms and turning ideas into reality through elegant code.

jees
Front-end Developer
jees is a creative soul with a passion for design and user experience. She crafts beautiful and intuitive interfaces that make our projects stand out.

The screenshot shows the 'Contact Us' section of the Kayse Eats mobile application. The title 'Contact Us' is centered at the top. Below it, there are three contact points: Email (info@kayseats.com), Phone (+252 63 3936074), and Website (www.kayseats.com).

Map: Displays restaurant locations or tracks orders.

KAYSE EATS

Home Restaurants FAQ About Us Map

Jumeira Restaurant, Cafe & Lounge, Safari Hotel, Mouwana, Studio, Elm Secondary, Best Academy, MATCO branch 2, DHOOLDAHABLE, Xeradli Arab, Sheekhi Isaxaax, Hargeisa Stadium, etc.

© 2024 Kayse Eats. All Rights Reserved.

Privacy Policy | Terms of Service | Contact Us

Keyboard shortcuts: ; Map data ©2024 Imagery ©2024 Airbus, CNES / Airbus, Maxar Technologies

Registration: Sign-up page for new users.

KAYSE EATS

Home Restaurants Login Register

User-Name

First Name

Last Name

Email Address

Phone number

Password

Confirm password

Delivery Address

Register

FAQ: Answers common user queries.

The FAQ page features a dark background with a subtle image of food on the right side. It contains five white rectangular boxes, each with a question and a plus sign to its right. The questions are:

- What is Kayse Eats?
- How can I place an order?
- Is delivery available only in Hargeisa?
- What are the delivery hours?
- Can I track my order?

Each box has a small square icon in the top right corner.

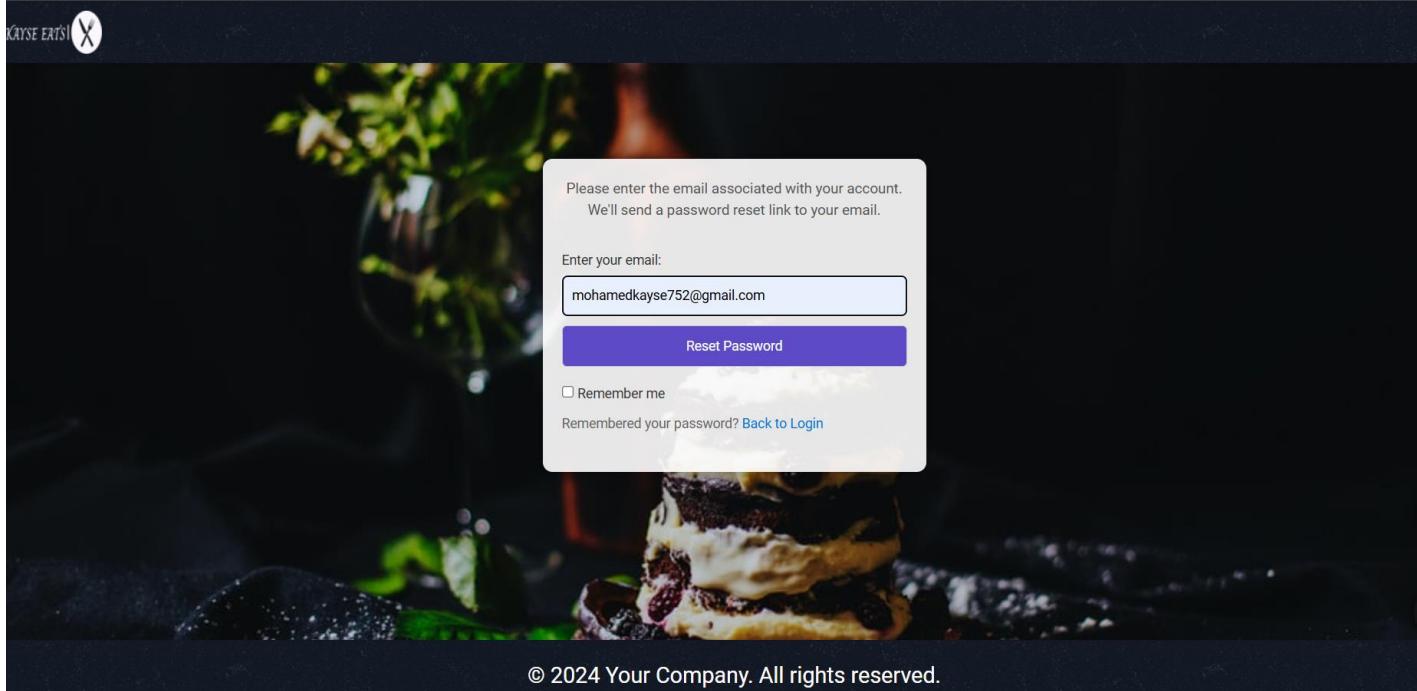
Restaurants: Displays a list of available restaurants.

The page has a dark header with the Kayse Eats logo and navigation links for Home, Restaurants, Login, and Register. Below the header, there are three numbered steps: 1. Choose Restaurant, 2. Pick Your favorite food, and 3. Order and Pay. A large image of a dessert is centered above the restaurant list.

| Restaurant Name | Location | Action |
|-----------------------------|--------------------|---------------------------|
| JEES Hotel | Shacab Area | View Menu |
| MAANSOOR HOTEL & RESTUARENT | H2FM+RFV, Hargeisa | View Menu |
| Damal Hotel | Siuna House | View Menu |

On the right side of the page, there are three circular icons with symbols: a grid, a person, and a gear.

Forgot Password: Allows users to reset their login credentials.



© 2024 Your Company. All rights reserved.

Checkout: Displays the cart summary and collects payment details for order completion.

A screenshot of a mobile application's checkout screen. The top navigation bar includes a logo, "Home", "Restaurants", "My Orders", and "Logout". The main content area is divided into three sections: "Choose Restaurant", "Pick Your favorite food", and "Order and Pay".

The "Order and Pay" section contains:

- Cart Summary**
- Cart Subtotal: \$140
- Delivery Charges: \$10
- Total: \$150**
- Payment options:
 - Pay on Zaad and my phone +252 63 3936074
 - Paypal
- A large green "Order Now" button.

Below the main section, there are three tabs: "Payment Options", "Address", and "Addition informations".

Product Action: Handles adding and managing cart items.

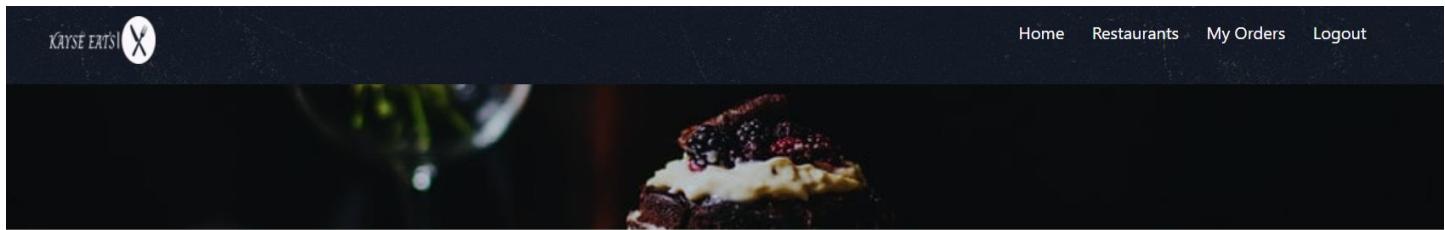
The screenshot displays a mobile application for food delivery. At the top, there is a navigation bar with the logo "KAYSE EATS" and a search icon. To the right are links for "Home", "Restaurants", "My Orders", and "Logout". Below the navigation, three circular icons represent steps: 1. Choose Restaurant, 2. Pick Your favorite food, and 3. Order and Pay.

The main content area features a large image of a restaurant named "Assod Hotel & Restuarent" located in "Sha'ab Area, Hargeisa". To the left of the image is a smaller thumbnail of the restaurant's exterior. To the right is a close-up image of a dish, likely spaghetti with meat sauce and basil.

On the left side of the screen, there is a "Your Cart" section. It lists a single item: "Hilib Xaniid" with a price of "\$10.00" and a quantity of "14". There is also a "Delete" icon next to the item name. Below this, the total amount is shown as "\$140" with the text "Cheap delivery!" underneath. A green "Checkout" button is at the bottom of this section.

On the right side, there is a "MENU" section. It shows a dish called "Hilib Xaniid" with a small image of the food. The description in English is: "Xaniid (Af Ingiris : beefsteak; Af Carabi : ar?) waa hilibka waslada marka lagu shiilo dab kali ah. Sidoo kale xaniidku waa nooc ka mid ah hilibka xoolaha, gaar ahaan hilibka laga jaro garabka, jeeniga, lugta, dhabarka iyo meelaha leh jiidh saafi ah. Sidoo kale wasladu waa hilibka laga gooyo muruqa xoolaha kaasi oo leh waxay oo baruur ah, laakiin inta badan jiidh saafi ah." The price is listed as "\$10.00" with a quantity input field containing the value "1". A blue "Add To Cart" button is located below the input field.

Your Orders: Lists customers' past and ongoing orders.



| Item | Quantity | Price | Status | Date | Action |
|--------------------|----------|---------|-----------------------|---------------------|--------------------|
| Hilib Xaniid | 14 | \$10.00 | Dispatch | 2024-11-18 12:16:43 | trash |
| Beer and Anjeero | 1 | \$5.00 | Dispatch | 2024-11-18 12:16:58 | trash |
| Baked Chicken Legs | 3 | \$10.00 | Dispatch | 2024-11-18 12:17:15 | trash |

Payment Options

A row of payment method icons including PayPal, Mastercard, Visa, American Express, and Stripe.

Address

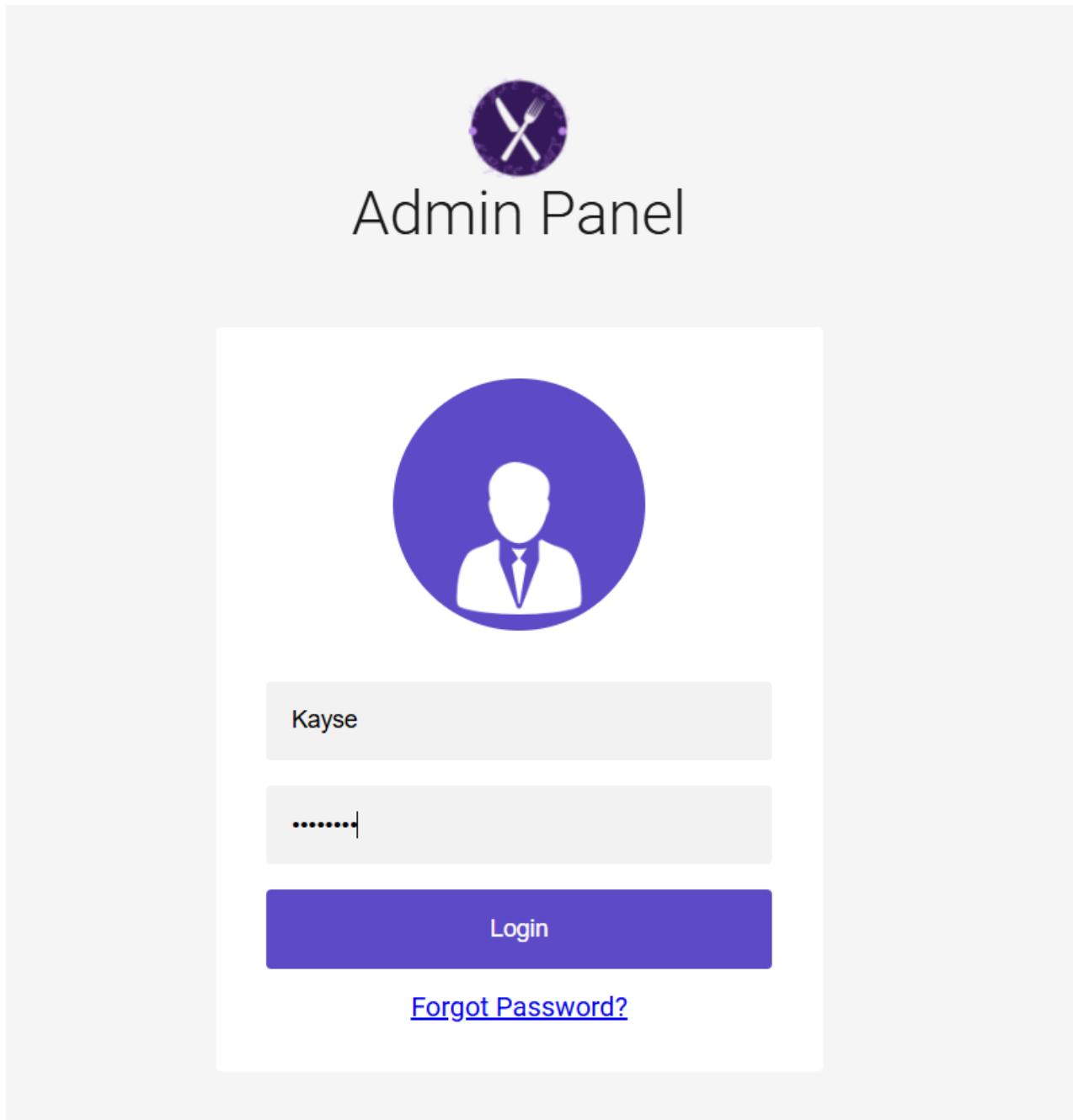
Wadada Madax-Tooyada ,Hargiesa TG JH, 09010

Addition informations

Join thousands of other restaurants who benefit from having partnered with us.

4.3.2 Admin Panel Pages

Login: Secure login page for Admins.



Forgot Password: Admin-side credential recovery page.



mohamedkayse752@gmail.com

Reset Password

Password reset link has been sent to your email. Please check your inbox.

Remember your password? [Login](#)

Admin Dashboard: displaying system summaries and metrics.

The Admin Dashboard displays the following key metrics:

- Restaurants: 10
- Dishes: 19
- Users: 9
- Total Orders: 17
- Food Categories: 5
- Processing Orders: 2
- Delivered Orders: 8
- Cancelled Orders: 2
- Total Earnings: 110.00

Filter Restaurants: Provides a tool to find and manage restaurants.

| Category | Name | Email | Phone | Url | Open Hrs | Close Hrs | Open Days | Address | Image | Date | Action |
|----------------|-----------------------------|----------------------|-------------------|---|----------|-----------|-----------|--|-------|---------------------|--------------------------------------|
| Cunto Soomaali | Sky Hotel Hargeisa | sedrgmk@gmail.com | +252 633956074 | https://skyhotelhargeisa.com/ | 6am | 3am | 24hr-x7 | G39F+HX8, Hargeisa | | 2024-11-18 09:20:46 | trash edit |
| | Ambassador Hotel | ambasor123@gmail.com | +252 6339560745 | http://www.ambassadorhotelhargeisa.com/ | 6am | 3am | 24hr-x7 | G3CF+6XF, Airport Road, Hargeysa | | 2024-11-17 21:59:06 | trash edit |
| | Carro Edeg Hotel hargeisa | carroedeg@gmail.com | +252 633956074567 | https://www.carroedeghotel.com/ | 6am | 3am | 24hr-x7 | Wadada Madax-Tooyada Hargeisa TG JH, 09010 | | 2024-11-17 21:55:57 | trash edit |
| European Food | Hamdaan Hotel \$ Restaurant | hamdaan@gmail.com | +252 633956074 | http://www.hamdanhotel.com/ | 6am | 3am | 24hr-x7 | National Fuel Station(Indha Bir, Bada Cas Infront of, Jigiga-Yar Rd, Hargeysa) | | 2024-11-17 21:52:52 | trash edit |

Restaurant Management: Tools to add, edit, or delete restaurant records.

The screenshot shows the 'Add Restaurant' form. On the left is a sidebar with navigation links: HOME, Dashboard, LOG, Users, Restaurant (selected), All Restaurants, Add Category, Add Restaurant, Menu, and Orders. The main area has a purple header 'Add Restaurant'. It contains fields for Restaurant Name ('Sky Hotel Hargeisa'), Business E-mail ('abdiakalab@gmail.com'), Phone ('34569034578'), Website URL ('https://skyhotelhargeisa.com/'), Open Hours ('6am'), Close Hours ('3am'), Open Days ('24hr-x7'), Select Category ('--Select Category--'), and an Image field with a 'Choose File' button ('No file chosen'). Below these is a 'Restaurant Address' section with a text input field and a small image icon. At the bottom are 'Save' and 'Cancel' buttons.

Food Category Management: Handles the organization of menu categories.

The screenshot shows the 'Add Restaurant Category' form. The sidebar is identical to the previous one. The main area has a purple header 'Add Restaurant Category'. It contains a 'Category' input field ('Cunto Ethiopian') and 'Save' and 'Cancel' buttons. Below this is a table titled 'Listed Categories' with columns: ID, Category Name, Date, and Action. The table lists five categories:

| ID | Category Name | Date | Action |
|----|--------------------|---------------------|--------|
| 9 | European Food | 2024-11-17 21:09:32 | |
| 8 | Arabian Food | 2024-11-17 21:09:20 | |
| 7 | East African Food | 2024-11-17 21:09:09 | |
| 6 | Mediterranean Food | 2024-11-17 21:08:57 | |
| 5 | Cunto Soomaali | 2024-11-17 21:08:42 | |

Food Menu Management: Manage dishes by adding, updating, or removing items.

The screenshot shows the 'Add Menu' form. It includes fields for 'Dish Name' (Muqmad), 'Description' (Marinated boneless goat's meat cooked together with rice in spiced Indian masala gravy.), 'Price' (4), 'Image' (Choose File, No file chosen), and 'Select Restaurant' (MAANSOOR HOTEL & RESTUARENT). There are 'Save' and 'Cancel' buttons at the bottom. The sidebar on the left shows navigation links: HOME, Dashboard, LOG, Users, Restaurant, Menu (selected), All Menus, Add Menu, and Orders.

Filter Food Items: Allows admins to search and filter menu items.

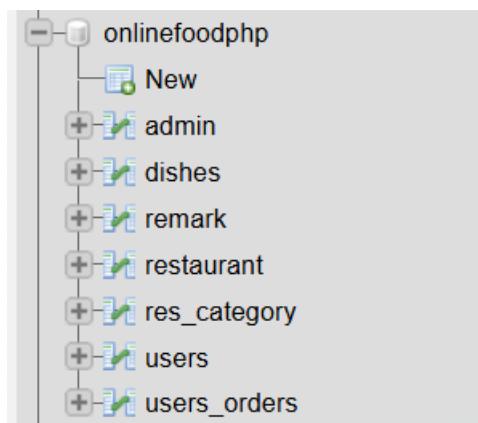
The screenshot shows the 'All Menu' page with a table of food items. The columns are: Restaurant, Dish, Description, Price, Image, and Action. The data includes:

| Restaurant | Dish | Description | Price | Image | Action |
|---------------------------|--------------------------|---|---------|-------|--------|
| Carro Edeg Hotel hargeisa | Cafe Mocha | Cafe Mocha Although a mocha is often interpreted differently across the world, the basis is that a shot of espresso is combined with chocolate powder or syrup, followed by milk or cream. It is a variant of a latte, in the sense that it is often 1/3 espresso and 2/3 steamed milk. | \$1.00 | | |
| Carro Edeg Hotel hargeisa | Apple Juice | Apple Juice | \$1.50 | | |
| Carro Edeg Hotel hargeisa | Hazelnut Latte | Hazelnut Latte is good drink | \$3.00 | | |
| Carro Edeg Hotel hargeisa | Goat Biryani | Marinated boneless goat's meat cooked together with rice in spiced Indian masala gravy. | \$13.60 | | |
| Carro Edeg Hotel hargeisa | Charcoal Grilled Chicken | Marinated then seasoned 3 chicken legs and deep fried until golden brown. | \$9.50 | | |
| Carro Edeg Hotel hargeisa | Chicken Tikka Masala | Cubes of chicken charcoal grilled finished in masala sauce. | \$8.00 | | |
| Carro Edeg Hotel hargeisa | Chilly Fish | Cubed fillet of fish sauteed with veggies, and infused with lime juice and complemented with sweet and sour chilli sauce. | \$20.00 | | |

4.4 Database Design

The **Kayse Eats** database is designed to support all functionalities of the system, with tables structured for efficiency and data integrity. The **MySQL database** schema is designed to support the functionalities of both panels. Each table aligns with specific screens, ensuring a smooth interaction between the frontend and the database.

4.4.1 Core Database Tables



| Table | Action | Rows | Type | Collation | Size | Overhead |
|--------------|--------|------|--------|--------------------|-----------|----------|
| admin | | 3 | InnoDB | latin1_swedish_ci | 16.0 KiB | - |
| dishes | | 19 | InnoDB | latin1_swedish_ci | 16.0 KiB | - |
| remark | | 16 | InnoDB | latin1_swedish_ci | 16.0 KiB | - |
| restaurant | | 10 | InnoDB | latin1_swedish_ci | 16.0 KiB | - |
| res_category | | 5 | InnoDB | latin1_swedish_ci | 16.0 KiB | - |
| users | | 9 | InnoDB | latin1_swedish_ci | 16.0 KiB | - |
| users_orders | | 17 | InnoDB | latin1_swedish_ci | 16.0 KiB | - |
| 7 tables | Sum | 79 | MyISAM | utf8mb4_0900_ai_ci | 112.0 KiB | 0 B |

1. **admin:** Stores admin account credentials and permissions.

| | adm_id | username | password | email | code | date |
|--|--------|----------|----------------------------------|---------------------|-------|---------------------|
| | 1 | Kayse | 9da8697858e6bbdeba9bcfea35cc607e | kayse@mail.com | | 2024-11-18 09:10:44 |
| | 2 | Deeq | a3d78b11c4c96a0360526f38e128117c | abdirizack@mail.com | 12 | 2024-11-18 09:13:59 |
| | 3 | Jees | 92936b24059feed692cc77cdd1a21780 | jees@mail.com | 67890 | 2024-11-18 09:14:45 |

- Connected Pages: Admin Login, Forgot Password, Admin Panel.
- **Purpose:** Stores admin credentials and permissions.

2. users: Manages customer account data.

- Fields: user_id (PK), name, email, password, phone, address.

| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | u_id | username | f_name | l_name | email | phone | password | address | status | date |
|--|--------------------------|-------------------------------|-------------------------------|---------------------------------|------|---------------|---------|----------|---------------------------|--------------------|----------------------------------|-----------|--------|---------------------|
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 7 | Hussien Haybe | ali | farah | mohamedkayse752@gmail.com | 2345678987 | 6fb42da0e32e07b61c9f0251fe627a9c | 150 | 1 | 2024-11-17 20:53:06 |
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 8 | Jama321 | jama | farah | sdfhjlvbn@gmail.com | +23456789f5678 | 6fb42da0e32e07b61c9f0251fe627a9c | 150 | 1 | 2024-11-17 21:00:16 |
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 9 | Ali321 | Ali | Muuse | sdgjkhjk@gmail.com | +2345678234679 | 6fb42da0e32e07b61c9f0251fe627a9c | 150 | 1 | 2024-11-17 21:01:12 |
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 10 | Faysal321 | ali | farah | sdhk@gmail.com | +56789023456789 | 6fb42da0e32e07b61c9f0251fe627a9c | 150 | 1 | 2024-11-17 21:01:48 |
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 11 | Suhail@ | Suhayb | Yuusuf | asdfgjk752@gmail.com | +2345694589 | 1cf126d20581502c3be21f524b40a9b9 | 150 | 1 | 2024-11-17 21:02:34 |
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 12 | Muuse Kaahin | Ali | Muuse | ajama0260@gmail.com | +252 633956074 | 6fb42da0e32e07b61c9f0251fe627a9c | Xero awr | 1 | 2024-11-17 21:04:34 |
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 13 | Deeq | Deeq | Ali Maal | abdialikalab@gmail.com | +252 633956079 | 6fb42da0e32e07b61c9f0251fe627a9c | Qudhayada | 1 | 2024-11-17 21:05:35 |
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 14 | Kayse | Xasan | Raabii | ajama02601@gmail.com | +252 6339560744567 | 6fb42da0e32e07b61c9f0251fe627a9c | Shulaac | 1 | 2024-11-17 21:06:31 |
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 15 | Nuur | Hussien | MKMK | sdfgujikope752@gmail.com | +252 633956074347 | 6fb42da0e32e07b61c9f0251fe627a9c | Shidhka | 1 | 2024-11-17 21:07:49 |

- Connected Pages: Registration, Login, Logout, Forgot Password, Customer Management.
- Purpose: Manages customer accounts.

3. dishes: Contains details of all available dishes.

- Fields: dish_id (PK), name, price, category_id, restaurant_id.

| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | d_id | rs_id | title | slogan | price | img |
|--|--------------------------|-------------------------------|-------------------------------|---------------------------------|------|-------|--------------------------|--|-------|--------------------|
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 17 | 12 | Cambaabur | Cambaabur is a Somali Eid bread, it is prepared mo... | 3.00 | 673a3d789f95d.jpg |
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 28 | 13 | Suqaar | Suqaar (pronounced sooh-car) is one of Somalia's m... | 4.00 | 673acf75c0cf0.webp |
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 29 | 6 | Hilib Geel | Camel meat has been described as raspberry red to ... | 7.00 | 673ad0409225f.webp |
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 30 | 11 | Beer and Anjeero | Beer, AKA liver goat, lamb, or cow, is one of Soma... | 5.00 | 673ad08c032b1.jpg |
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 31 | 10 | Hilib Xaniid | Xaniid (Af Ingiriis : beefsteak; Af Carabi : ar?) ... | 10.00 | 673ad0f2129f9.jpg |
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 32 | 12 | Biryani & Chicken | The Kolkata style of biryani belongs to West Benga... | 15.00 | 673ad1c9e5104.webp |
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 33 | 5 | Baked Chicken Legs | These chicken legs are golden on the outside, juic... | 10.00 | 673ad24edee79.webp |
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 34 | 9 | Kaluun | Easy, tender baked fish infused with garlic and ba... | 13.00 | 673ad2a4024c2.jpg |
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 35 | 8 | Surbiyaan Hilib Adhi | Lamb surbiyaan, this is one pot dish both deliciou... | 12.00 | 673ad36e09e0f.jpg |
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 36 | 9 | Somali Sambusa | Somali Sambusa is a stuffed triangular pastries th... | 1.50 | 673ad3b889a35.jpg |
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 37 | 12 | Somali Malawah | Malawah Somali Crepe these very thin cooked pancak... | 2.00 | 673ad4193303c.jpg |
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 38 | 12 | Muqmad | Laxooxo with finely diced beef. | 4.00 | 673ad4bd56703.webp |
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 39 | 12 | Chilly Fish | Cubed fillet of fish sauteed with veggies, and infuse... | 20.00 | 673ad55878b34.jpg |
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 40 | 12 | Chicken Tikka Masala | Cubes of chicken charcoal grilled finished in mas... | 8.00 | 673ad5e4db366.jpg |
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 41 | 12 | Charcoal Grilled Chicken | Marinated then seasoned 3 chicken legs and deep fr... | 9.50 | 673ad621ac678.webp |
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 42 | 12 | Goat Biryani | Marinated boneless goat's meat cooked together wit... | 13.60 | 673ad6a3a46d4.jpg |
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 43 | 12 | Hazelnut Latte | Hazelnut Latte is good drink | 3.00 | 673ad79d6b989.jpg |
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 44 | 12 | Apple Juice | Apple Juice | 1.50 | 673ad7d83f356.jpg |
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 45 | 12 | Cafe Mocha | Cafe MochaAlthough a mocha is often interpreted di... | 1.00 | 673ad85e9dac1.jpg |

- Connected Pages: Dishes, Food Menu Management, Filter Food Items.
- **Purpose:** Stores menu items.

4. **restaurant:** Stores restaurant information.

- Fields: `restaurant_id (PK)`, `name`, `location`, `contact`.

| | <input type="button" value="←"/> | <input type="button" value="→"/> | <input type="button" value="▼"/> | <code>rs_id</code> | <code>c_id</code> | <code>title</code> | <code>email</code> | <code>phone</code> | <code>url</code> | <code>o_hr</code> | <code>c_hr</code> | <code>o_days</code> | <code>address</code> | <code>image</code> |
|-------------------------------------|--|-------------------------------------|---------------------------------------|--------------------|-------------------|-----------------------------|---------------------|----------------------|---------------------------------|-------------------|-------------------|---------------------|---|--------------------|
| <input type="checkbox"/> | <input type="button" value="Edit"/> | <input type="button" value="Copy"/> | <input type="button" value="Delete"/> | 5 | 5 | JEES Hotel | jeesera52@gmail.com | +252 633956074 | www.jees.com | 6am | 3am | 24hr-x7 | Shacab Area | 673a337f764c9.webp |
| <input type="checkbox"/> | <input type="button" value="Edit"/> | <input type="button" value="Copy"/> | <input type="button" value="Delete"/> | 6 | 0 | MAANSOOR HOTEL & RESTAURENT | maansoor@gmail.com | +252 63395607412 | www.Maansoor.com | 6am | 8pm | 24hr-x7 | H2FM+RFV, Hargeisa | 673a3a28e8286.jpg |
| <input type="checkbox"/> | <input type="button" value="Edit"/> | <input type="button" value="Copy"/> | <input type="button" value="Delete"/> | 7 | 0 | Damal Hotel | www.Damal.com | +252 633956074 | www.Damal.com | 6am | 3am | 24hr-x7 | Suuqa Hoose | 673a367f22012.jpg |
| <input type="checkbox"/> | <input type="button" value="Edit"/> | <input type="button" value="Copy"/> | <input type="button" value="Delete"/> | 8 | 0 | Beder Hotel | beder@gmail.com | +252 633956074 | www.Beder.com | 6am | 8pm | 24hr-x7 | H366+W6J Wadada Timacade Hargeysa TG JH, 09010 | 673a3aa053ff2.jpg |
| <input type="checkbox"/> | <input type="button" value="Edit"/> | <input type="button" value="Copy"/> | <input type="button" value="Delete"/> | 9 | 0 | Maamuus Hotel & Restuarent | maamuus@gmail.com | +252 633956074 | www.maamuus.com | 6am | 3am | 24hr-x7 | Maguugo Street, Rd Number 1, Hargeisa | 673a3874bcd6f.jpg |
| <input type="checkbox"/> | <input type="button" value="Edit"/> | <input type="button" value="Copy"/> | <input type="button" value="Delete"/> | 10 | 0 | Assod Hotel & Restuarent | assod@Gmail.com | +45623456789 | www.aasod.com | 6am | 3am | 24hr-x7 | Sha'ab Area, Hargeisa | 673a38d2dd1e1.jpg |
| <input type="checkbox"/> | <input type="button" value="Edit"/> | <input type="button" value="Copy"/> | <input type="button" value="Delete"/> | 11 | 9 | Hamdaan Hotel \$ Restuarent | hamdaan@gmail.com | +252 633956074 | http://www.hamdanhotel.com/ | 6am | 3am | 24hr-x7 | National Fuel Station(Indha Bir, Bada Cas Infront ... | 673a3b849bcc4.jpg |
| <input type="checkbox"/> | <input type="button" value="Edit"/> | <input type="button" value="Copy"/> | <input type="button" value="Delete"/> | 12 | 0 | Carro Eddeg Hotel hargeisa | carroedeg@gmail.com | +252 633956074567 | https://www.carroedeghotel.com/ | 6am | 3am | 24hr-x7 | Wadada Madax-Toyada Hargeisa TG JH, 09010 | 673a3c3cf3f31.jpg |
| <input checked="" type="checkbox"/> | <input type="button" value="Console"/> | | | | | | | | | | | | | G3CF+6XF, |

- Connected Pages: Restaurants, Filter Restaurants, Restaurant Management.
- **Purpose:** Stores restaurant details.

5. **res_category:** Organizes restaurants into categories.

- Fields: `category_id (PK)`, `category_name`.

| | <input type="button" value="←"/> | <input type="button" value="→"/> | <input type="button" value="▼"/> | <code>c_id</code> | <code>c_name</code> | <code>date</code> |
|--------------------------|-------------------------------------|-------------------------------------|---------------------------------------|-------------------|---------------------|---------------------|
| <input type="checkbox"/> | <input type="button" value="Edit"/> | <input type="button" value="Copy"/> | <input type="button" value="Delete"/> | 5 | Cunto Soomaali | 2024-11-17 21:08:42 |
| <input type="checkbox"/> | <input type="button" value="Edit"/> | <input type="button" value="Copy"/> | <input type="button" value="Delete"/> | 6 | Mediteranean Food | 2024-11-17 21:08:57 |
| <input type="checkbox"/> | <input type="button" value="Edit"/> | <input type="button" value="Copy"/> | <input type="button" value="Delete"/> | 7 | East African Food | 2024-11-17 21:09:09 |
| <input type="checkbox"/> | <input type="button" value="Edit"/> | <input type="button" value="Copy"/> | <input type="button" value="Delete"/> | 8 | Arabian Food | 2024-11-17 21:09:20 |
| <input type="checkbox"/> | <input type="button" value="Edit"/> | <input type="button" value="Copy"/> | <input type="button" value="Delete"/> | 9 | European Food | 2024-11-17 21:09:32 |

- Connected Pages: Filter Restaurants, Restaurant Management.
- **Purpose:** Defines restaurant categories.

6. users_orders: Tracks customer orders.

- Fields: o_id (PK), u_id (FK), dish_id (FK), order_date, status.

| | <input type="button" value="Edit"/> | <input type="button" value="Copy"/> | <input type="button" value="Delete"/> | o_id | u_id | title | quantity | price | status | date |
|--------------------------|-------------------------------------|-------------------------------------|---------------------------------------|------|------|--------------------|----------|-------|------------|---------------------|
| <input type="checkbox"/> | <input type="button" value="Edit"/> | <input type="button" value="Copy"/> | <input type="button" value="Delete"/> | 10 | 13 | Cambaabur | 1 | 3.00 | closed | 2024-11-18 09:04:21 |
| <input type="checkbox"/> | <input type="button" value="Edit"/> | <input type="button" value="Copy"/> | <input type="button" value="Delete"/> | 11 | 13 | Biryani & Chicken | 1 | 15.00 | closed | 2024-11-18 09:05:08 |
| <input type="checkbox"/> | <input type="button" value="Edit"/> | <input type="button" value="Copy"/> | <input type="button" value="Delete"/> | 12 | 13 | Somali Malawah | 2 | 2.00 | closed | 2024-11-18 09:05:38 |
| <input type="checkbox"/> | <input type="button" value="Edit"/> | <input type="button" value="Copy"/> | <input type="button" value="Delete"/> | 13 | 13 | Muqmad | 3 | 4.00 | NULL | 2024-11-18 09:03:47 |
| <input type="checkbox"/> | <input type="button" value="Edit"/> | <input type="button" value="Copy"/> | <input type="button" value="Delete"/> | 14 | 13 | Goat Biryani | 1 | 13.60 | in process | 2024-11-18 09:05:57 |
| <input type="checkbox"/> | <input type="button" value="Edit"/> | <input type="button" value="Copy"/> | <input type="button" value="Delete"/> | 15 | 14 | Hilib Xaniid | 14 | 10.00 | NULL | 2024-11-18 12:16:43 |
| <input type="checkbox"/> | <input type="button" value="Edit"/> | <input type="button" value="Copy"/> | <input type="button" value="Delete"/> | 16 | 14 | Beer and Anjeero | 1 | 5.00 | NULL | 2024-11-18 12:16:58 |
| <input type="checkbox"/> | <input type="button" value="Edit"/> | <input type="button" value="Copy"/> | <input type="button" value="Delete"/> | 17 | 14 | Baked Chicken Legs | 3 | 10.00 | NULL | 2024-11-18 12:17:15 |

- Connected Pages: Your Orders, Manage Food Orders, Delete Orders, Cancel Orders.
- Purpose: Tracks customer orders.

7. remark: Logs customer reviews and admin remarks.

- Fields: remark_id (PK), user_id (FK), dish_id (FK), comment.

| | <input type="button" value="Edit"/> | <input type="button" value="Copy"/> | <input type="button" value="Delete"/> | id | frm_id | status | remark | remarkDate |
|--------------------------|-------------------------------------|-------------------------------------|---------------------------------------|----|--------|------------|-----------|---------------------|
| <input type="checkbox"/> | <input type="button" value="Edit"/> | <input type="button" value="Copy"/> | <input type="button" value="Delete"/> | 13 | 10 | closed | waad | 2024-11-18 09:04:21 |
| <input type="checkbox"/> | <input type="button" value="Edit"/> | <input type="button" value="Copy"/> | <input type="button" value="Delete"/> | 14 | 11 | closed | ghjkl | 2024-11-18 09:05:08 |
| <input type="checkbox"/> | <input type="button" value="Edit"/> | <input type="button" value="Copy"/> | <input type="button" value="Delete"/> | 15 | 12 | closed | asdfghjkl | 2024-11-18 09:05:38 |
| <input type="checkbox"/> | <input type="button" value="Edit"/> | <input type="button" value="Copy"/> | <input type="button" value="Delete"/> | 16 | 14 | in process | dfghjk | 2024-11-18 09:05:57 |

- Connected Pages: Customer Feedback, Product Action.
- Purpose: Stores customer reviews and admin remarks.

4.4.2 Relationships Between Tables

- **users ↔ users_orders:** Links customers to their orders.
- **users_orders ↔ remark:** Allows orders to have feedback.
- **restaurant ↔ dishes:** Associates restaurants with their menus.
- **restaurant ↔ res_category:** Groups restaurants by categories.
- **dishes ↔ users_orders:** Tracks dishes within specific orders.

4.4.3 Mapping Pages to Tables

Each database table is mapped to one or more interface screens to provide full functionality:

| Screen | Database Table(s) | Purpose |
|-----------------------|--------------------|--|
| Home | restaurant, dishes | Displays featured restaurants and dishes. |
| Login | users | Authenticates customer accounts. |
| Registration | users | Adds new customers to the database. |
| Dishes | dishes, restaurant | Lists menu items for browsing. |
| Your Orders | users_orders | Shows customer order history. |
| Restaurant Management | restaurant | Allows management of restaurant details. |
| Food Menu Management | dishes | Enables admin to add or update menu items. |
| Manage Food Orders | users_orders | Tracks customer orders and updates statuses. |

CHAPTER 5

KAYSE EATS IMPLEMENTATION AND TESTING

TABLE OF CONTENTS

1. Introduction
 - 1.1. Overview of the System
2. Implemented Codes for Interfaces
 - 2.1. Customer Panel Interfaces
 - 2.1.1. Registration Page
 - 2.1.2. Login Page
 - 2.1.3. View Orders Page
 - 2.2. Admin Panel Interfaces
 - 2.2.1. Admin Login Page
 - 2.2.2. Manage Restaurants Page
 - 2.2.3. Manage Food Menu
3. Implemented Codes for Database
 - 3.1. Admin Table
 - 3.2. Users Table
 - 3.3. Dishes Table
 - 3.4. Restaurant Table
 - 3.5. Categories Table
 - 3.6. Users Orders Table
 - 3.7. Remarks Table
4. Testing
 - 4.1. Unit Testing
 - 4.2. Integration Testing
 - 4.3. System Testing
 - 4.4. Acceptance Testing

5.1 Introduction

As a robust and scalable online food ordering system, Kayse Eats was designed to provide seamless interactions for customers and administrators. It integrates two main panels: the Customer Panel for users to browse dishes, place orders, and track their orders, and the Admin Panel, which enables administrators to manage food items, categories, restaurants, and customer orders. This chapter covers the technical implementation of the interfaces, database schema, and comprehensive testing of the system. It demonstrates the development process and highlights the steps taken to ensure the system meets high standards of functionality, usability, and reliability.

The database schema consists of seven interrelated tables designed to support all features of the system. Passwords are stored securely using the MD5 hashing algorithm to protect user and admin credentials. This chapter reflects the detailed work and effort put into developing Kayse Eats and validates its readiness for deployment.

5.2 Implemented Codes for Interfaces

The implementation of the interfaces for both the Customer Panel and Admin Panel ensures user-friendly and efficient navigation throughout the system.

Customer Panel Interfaces

1. Registration Page

This interface allows customers to create their accounts by providing basic information. The password entered is hashed using MD5 before storage to enhance security.

```
2. <!DOCTYPE html>
3. <html lang="en">
4. <?php
5.
6. session_start();
7. error_reporting(0);
8. include("connection/connect.php");
9. if(isset($_POST['submit']) )
10. {
11.     if(empty($_POST['firstname']) ||
12.         empty($_POST['lastname']) ||
13.         empty($_POST['email']) ||
14.         empty($_POST['phone']) ||
15.         empty($_POST['password']) ||
16.         empty($_POST['cpassword']) ||
17.         empty($_POST['cpassword'])) {
18.             $message = "All fields must be Required!";
19.         }
20.     }
21. else
22. {
23.
24.     $check_username= mysqli_query($db, "SELECT username FROM users where username
= '".$_POST['username']."' " );
```

```

25.      $check_email = mysqli_query($db, "SELECT email FROM users where email =
26.      '".$_POST['email']."' ");
27.
28.
29.      if($_POST['password'] != $_POST['cpassword']){
30.
31.          echo "<script>alert('Password not match');</script>";
32.      }
33.      elseif(strlen($_POST['password']) < 6)
34.      {
35.          echo "<script>alert('Password Must be >=6');</script>";
36.      }
37.      elseif(strlen($_POST['phone']) < 10)
38.      {
39.          echo "<script>alert('Invalid phone number!');</script>";
40.      }
41.
42.      elseif (!filter_var($_POST['email'], FILTER_VALIDATE_EMAIL))
43.      {
44.          echo "<script>alert('Invalid email address please type a valid
45.          email!');</script>";
46.      }
47.      elseif(mysqli_num_rows($check_username) > 0)
48.      {
49.          echo "<script>alert('Username Already exists!');</script>";
50.      }
51.      elseif(mysqli_num_rows($check_email) > 0)
52.      {
53.          echo "<script>alert('Email Already exists!');</script>";
54.      }
55.
56.
57.      $mql = "INSERT INTO
58.      users(username,f_name,l_name,email,phone,password,address)
59.      VALUES('".$_POST['username']."' , '".$_POST['firstname']."' , '".$_POST['lastname']."' ,
60.      '".$_POST['email']."' , '".$_POST['phone']."' , '".$md5($_POST['password'])."' ,
61.      '".$_POST['address']."' )";
62.
63.      mysqli_query($db, $mql);
64.
65.      header("refresh:0.1;url=login.php");
66.  }

```

```

62.      }
63.      }
64.      }
65.

66.      ?>
67.

68.      <head>
69.          <meta charset="utf-8">
70.          <meta http-equiv="X-UA-Compatible" content="IE=edge">
71.          <meta name="viewport" content="width=device-width, initial-scale=1, shrink-
    to-fit=no">
72.          <meta name="description" content="">
73.          <meta name="author" content="">
74.          <link rel="icon" href="#">
75.          <title>Registration</title>
76.          <link href="css/bootstrap.min.css" rel="stylesheet">
77.          <link href="css/font-awesome.min.css" rel="stylesheet">
78.          <link href="css/animsition.min.css" rel="stylesheet">
79.          <link href="css/animate.css" rel="stylesheet">
80.          <link href="css/style.css" rel="stylesheet"> </head>
81.      <body>
82.          <div style=" background-image: url('images/img/pimg.jpg');">
83.              <header id="header" class="header-scroll top-header headrom">
84.                  <nav class="navbar navbar-dark">
85.                      <div class="container">
86.                          <button class="navbar-toggler hidden-lg-up" type="button"
    data-toggle="collapse" data-target="#mainNavbarCollapse">#9776;</button>
87.                          <a class="navbar-brand" href="index.php">  </a>
88.                          <div class="collapse navbar-toggleable-md float-lg-right"
    id="mainNavbarCollapse">
89.                              <ul class="nav navbar-nav">
90.                                  <li class="nav-item"> <a class="nav-link active"
    href="index.php">Home <span class="sr-only">(current)</span></a> </li>
91.                                  <li class="nav-item"> <a class="nav-link active"
    href="restaurants.php">Restaurants <span class="sr-only"></span></a> </li>
92.
93.                                  <?php
94.                                  if(empty($_SESSION["user_id"]))
95.                                  {

```

```

96.           echo '<li class="nav-item"><a href="login.php"'
  class="nav-link active">Login</a> </li>
97.           echo '<li class="nav-item"><a href="registration.php"'
  class="nav-link active">Register</a> </li>';
98.       }
99.   else
100.   {
101.
102.
103.           echo '<li class="nav-item"><a'
  href="your_orders.php" class="nav-link active">My Orders</a> </li>';
104.           echo '<li class="nav-item"><a href="logout.php"'
  class="nav-link active">Logout</a> </li>';
105.       }
106.
107.   ?>
108.
109.           </ul>
110.           </div>
111.           </div>
112.           </nav>
113.           </header>
114.           <div class="page-wrapper">
115.
116.           <div class="container">
117.             <ul>
118.
119.
120.             </ul>
121.           </div>
122.
123.           <section class="contact-page inner-page">
124.             <div class="container " >
125.               <div class="row " >
126.                 <div class="col-md-12" >
127.                   <div class="widget" >
128.                     <div class="widget-body" >
129.
130.                     <form action="" method="post" >
131.                       <div class="row" >
132.                         <div class="form-group col-sm-12" >
133.                           <label for="exampleInputEmail1" >User-
Name</label>

```

```

134.                                     <input class="form-control" type="text"
    name="username" id="example-text-input">
135.                                     </div>
136.                                     <div class="form-group col-sm-6">
137.                                         <label for="exampleInputEmail1">First
    Name</label>
138.                                     <input class="form-control" type="text"
    name="firstname" id="example-text-input">
139.                                     </div>
140.                                     <div class="form-group col-sm-6">
141.                                         <label for="exampleInputEmail1">Last
    Name</label>
142.                                     <input class="form-control" type="text"
    name="lastname" id="example-text-input-2">
143.                                     </div>
144.                                     <div class="form-group col-sm-6">
145.                                         <label for="exampleInputEmail1">Email
    Address</label>
146.                                     <input type="text" class="form-control"
    name="email" id="exampleInputEmail1" aria-describedby="emailHelp">
147.                                     </div>
148.                                     <div class="form-group col-sm-6">
149.                                         <label for="exampleInputEmail1">Phone
    number</label>
150.                                     <input class="form-control" type="text"
    name="phone" id="example-tel-input-3">
151.                                     </div>
152.                                     <div class="form-group col-sm-6">
153.                                         <label
    for="exampleInputPassword1">Password</label>
154.                                         <input type="password" class="form-
    control" name="password" id="exampleInputPassword1">
155.                                         </div>
156.                                         <div class="form-group col-sm-6">
157.                                             <label
    for="exampleInputPassword1">Confirm password</label>
158.                                             <input type="password" class="form-
    control" name="cpassword" id="exampleInputPassword2">
159.                                             </div>
160.                                         <div class="form-group col-sm-12">
161.                                             <label for="exampleTextarea">Delivery
    Address</label>
162.                                             <textarea class="form-control"
    id="exampleTextarea" name="address" rows="3"></textarea>

```

```

163.                                     </div>
164.                                     </div>
165.                                     </div>
166.                                     <div class="row">
167.                                         <div class="col-sm-4">
168.                                             <p> <input type="submit" value="Register"
169.                                                 name="submit" class="btn theme-btn"> </p>
170.                                         </div>
171.                                     </div>
172.                                     </form>
173.
174.                                     </div>
175.
176.                                     </div>
177.
178.                                     </div>
179.
180.                                     </div>
181.                                     </div>
182.                                     </section>
183.
184.
185.                                     <footer class="footer">
186.                                         <div class="container">
187.
188.                                         <div class="row bottom-footer">
189.                                             <div class="container">
190.                                                 <div class="row">
191.                                                     <div class="col-xs-12 col-sm-3 payment-options color-
192.                                                         gray">
193.                                                 <h5>Payment Options</h5>
194.                                                 <ul>
195.                                                     <li>
196.                                                         <a href="#">  </a>
198.                                                     </li>
199.                                                     <li>
200.                                                         <a href="#"> <img

```

```

201.                                     <a href="#">  </a>
202.                                     </li>
203.                                     <li>
204.                                         <a href="#">  </a>
205.                                         </li>
206.                                         <li>
207.                                             <a href="#">  </a>
208.                                             </li>
209.                                         </ul>
210.                                         </div>
211.                                         <div class="col-xs-12 col-sm-4 address color-gray">
212.                                             <h5>Address</h5>
213.                                             <p>Wadada Madax-Tooyada ,Hargiesa
214.                                               TG JH, 09010</p>
215.                                             <h5>Phone: +252 63 3936074</a></h5> </div>
216.                                         <div class="col-xs-12 col-sm-5 additional-info
    color-gray">
217.                                             <h5>Addition informations</h5>
218.                                             <p>Join thousands of other restaurants who
    benefit from having partnered with us.</p>
219.                                         </div>
220.                                         </div>
221.                                         </div>
222.                                         </div>
223.
224.                                         </div>
225.                                         </footer>
226.
227.                                         </div>
228.
229.                                         <script src="js/jquery.min.js"></script>
230.                                         <script src="js/tether.min.js"></script>
231.                                         <script src="js/bootstrap.min.js"></script>
232.                                         <script src="js/animstition.min.js"></script>
233.                                         <script src="js/bootstrap-slider.min.js"></script>
234.                                         <script src="js/jquery.isotope.min.js"></script>
235.                                         <script src="js/headroom.js"></script>
236.                                         <script src="js/foodpicky.min.js"></script>
237.                                         </body>
238.

```

239. </html>

2. Login Page

Customers can log in using their credentials. The provided password is hashed and compared with the stored hash.

```
<!DOCTYPE html>
<html lang="en" >

<head>
    <meta charset="UTF-8">
    <title>Login</title>

        <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/meyer-
reset/2.0/reset.min.css">

        <link rel='stylesheet prefetch'
href='https://fonts.googleapis.com/css?family=Roboto:400,100,300,500,700,900|RobotoDraft:400,
100,300,500,700,900'>
<link rel='stylesheet prefetch' href='https://maxcdn.bootstrapcdn.com/font-
awesome/4.3.0/css/font-awesome.min.css'>

        <link rel="stylesheet" href="css/login.css">

        <style type="text/css">
#buttn{
    color:#fff;
    background-color: #5c4ac7;
}
</style>

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<link href="css/bootstrap.min.css" rel="stylesheet">
<link href="css/font-awesome.min.css" rel="stylesheet">
<link href="css/animsition.min.css" rel="stylesheet">
<link href="css/animate.css" rel="stylesheet">
<link href="css/style.css" rel="stylesheet">

</head>

<body>
<header id="header" class="header-scroll top-header headrom">
    <nav class="navbar navbar-dark">
        <div class="container">
            <button class="navbar-toggler hidden-lg-up" type="button" data-
toggle="collapse" data-target="#mainNavbarCollapse">#9776;</button>
            <a class="navbar-brand" href="index.php">  </a>
```

```

<div class="collapse navbar-toggleable-md float-lg-right"
id="mainNavbarCollapse">
    <ul class="nav navbar-nav">
        <li class="nav-item"> <a class="nav-link active"
href="index.php">Home <span class="sr-only">(current)</span></a> </li>
        <li class="nav-item"> <a class="nav-link active"
href="restaurants.php">Restaurants <span class="sr-only"></span></a> </li>

        <?php
        if(empty($_SESSION["user_id"]))
        {
            echo '<li class="nav-item"><a href="login.php" class="nav-
link active">Login</a> </li>
            <li class="nav-item"><a href="registration.php" class="nav-link
active">Register</a> </li>';
            }
        else
        {

            echo ' <li class="nav-item"><a href="your_orders.php"
class="nav-link active">My Orders</a> </li>';
            echo ' <li class="nav-item"><a href="logout.php"
class="nav-link active">Logout</a> </li>';
            }

        ?>

        </ul>
    </div>
</div>
</header>
<div style=" background-image: url('images/img/pimg.jpg');">

<?php
include("connection/connect.php");
error_reporting(0);
session_start();
if(isset($_POST['submit']))
{
    $username = $_POST['username'];
    $password = $_POST['password'];

    if(!empty($_POST["submit"]))
    {

```

```

$loginquery ="SELECT * FROM users WHERE username='$username' &&
password='".md5($password)."'"; //selecting matching records
$result=mysqli_query($db, $loginquery); //executing
$row=mysqli_fetch_array($result);

        if(is_array($row))
        {
            $_SESSION["user_id"] = $row['u_id'];
            header("refresh:1;url=index.php");
        }
        else
        {
            $message = "Invalid Username or Password!";
        }
    }

}

?>

<div class="pen-title">
<
</div>

<div class="module form-module">
<div class="toggle">

</div>
<div class="form">
<h2>Login to your account</h2>
<span style="color:red;"><?php echo $message; ?></span>
<span style="color:green;"><?php echo $success; ?></span>
<form action="" method="post">
    <input type="text" placeholder="Username" name="username"/>
    <input type="password" placeholder="Password" name="password"/>
    <input type="submit" id="buttn" name="submit" value="Login" />
    <!-- Add this line within your form module -->
<div class="cta">Forgot your password? <a href="forgot_password.php"
style="color:#5c4ac7;">Reset it here</a></div>

    </form>
</div>

<div class="cta">Not registered?<a href="registration.php" style="color:#5c4ac7;"> Create
an account</a></div>

```

```

</div>
<script src='http://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.3/jquery.min.js'></script>

<div class="container-fluid pt-3">
    <p></p>
</div>

<footer class="footer">
    <div class="container">

        <div class="bottom-footer">
            <div class="row">
                <div class="col-xs-12 col-sm-3 payment-options color-gray">
                    <h5>Payment Options</h5>
                    <ul>
                        <li>
                            <a href="#"> 
</a>
                        </li>
                        <li>
                            <a href="#">  </a>
                        </li>
                        <li>
                            <a href="#"> 
</a>
                        </li>
                        <li>
                            <a href="#"> 
</a>
                        </li>
                        <li>
                            <a href="#"> 
</a>
                        </li>
                    </ul>
                </div>
                <div class="col-xs-12 col-sm-4 address color-gray">
                    <h5>Address</h5>
                    <p>Wadada Madax-Tooyada ,Hargiesa

```

```

TG JH, 09010</p>
<h5>Phone: +252 63 3936074</a></h5> </div>
<div class="col-xs-12 col-sm-5 additional-info color-gray">
    <h5>Addition informations</h5>
    <p>Join thousands of other restaurants who benefit from having
partnered with us.</p>
    </div>
</div>
</div>

</div>
</footer>

</body>

</html>

```

4-Your Orders Page

Displays the customer's order history with options for cancellation.

```

<!DOCTYPE html>
<html lang="en">
<?php
include("connection/connect.php");
error_reporting(0);
session_start();

if(empty($_SESSION['user_id']))
{
    header('location:login.php');
}
else
{
?>
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <link rel="icon" href="#">
    <title>My Orders</title>
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <link href="css/font-awesome.min.css" rel="stylesheet">

```

```

<link href="css/animsition.min.css" rel="stylesheet">
<link href="css/animate.css" rel="stylesheet">
<link href="css/style.css" rel="stylesheet">
<style type="text/css" rel="stylesheet">

.indent-small {
  margin-left: 5px;
}
.form-group.internal {
  margin-bottom: 0;
}
.dialog-panel {
  margin: 10px;
}
.datepicker-dropdown {
  z-index: 200 !important;
}
.panel-body {
  background: #e5e5e5;
  /* Old browsers */
  background: -moz-radial-gradient(center, ellipse cover, #e5e5e5 0%, #ffffff 100%);
  /* FF3.6+ */
  background: -webkit-gradient(radial, center center, 0px, center center, 100%, color-stop(0%, #e5e5e5), color-stop(100%, #ffffff));
  /* Chrome,Safari4+ */
  background: -webkit-radial-gradient(center, ellipse cover, #e5e5e5 0%, #ffffff 100%);
  /* Chrome10+,Safari5.1+ */
  background: -o-radial-gradient(center, ellipse cover, #e5e5e5 0%, #ffffff 100%);
  /* Opera 12+ */
  background: -ms-radial-gradient(center, ellipse cover, #e5e5e5 0%, #ffffff 100%);
  /* IE10+ */
  background: radial-gradient(ellipse at center, #e5e5e5 0%, #ffffff 100%);
  /* W3C */
  filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#e5e5e5',
endColorstr='#ffffff', GradientType=1);
  font: 600 15px "Open Sans", Arial, sans-serif;
}
label.control-label {
  font-weight: 600;
  color: #777;
}

/*
table {
  width: 750px;

```

```

border-collapse: collapse;
margin: auto;

}

/* Zebra striping */
/* tr:nth-of-type(odd) {
background: #eee;
}

th {
background: #404040;
color: white;
font-weight: bold;

}

td, th {
padding: 10px;
border: 1px solid #ccc;
text-align: left;
font-size: 14px;

} */ */

@media
only screen and (max-width: 760px),
(min-device-width: 768px) and (max-device-width: 1024px)  {

/* table {
width: 100%;
}

table, thead, tbody, th, td, tr {
display: block;
} */

/* thead tr {
position: absolute;
top: -9999px;
left: -9999px;
}

```

```

tr { border: 1px solid #ccc; } */

/* td {

    border: none;
    border-bottom: 1px solid #eee;
    position: relative;
    padding-left: 50%;

}

td:before {

    position: absolute;

    top: 6px;
    left: 6px;
    width: 45%;
    padding-right: 10px;
    white-space: nowrap;

    content: attr(data-column);

    color: #000;
    font-weight: bold;
} */

}

</style>

</head>

<body>

<header id="header" class="header-scroll top-header headrom">

    <nav class="navbar navbar-dark">
        <div class="container">

```

```

        <button class="navbar-toggler hidden-lg-up" type="button" data-
toggle="collapse" data-target="#mainNavbarCollapse">&#9776;</button>
        <a class="navbar-brand" href="index.php">  </a>
        <div class="collapse navbar-toggleable-md float-lg-right"
id="mainNavbarCollapse">
            <ul class="nav navbar-nav">
                <li class="nav-item"> <a class="nav-link active"
href="index.php">Home <span class="sr-only">(current)</span></a> </li>
                <li class="nav-item"> <a class="nav-link active"
href="restaurants.php">Restaurants <span class="sr-only"></span></a> </li>

                <?php
                if(empty($_SESSION["user_id"]))
                {
                    echo '<li class="nav-item"><a href="login.php" class="nav-
link active">Login</a> </li>
                    <li class="nav-item"><a href="registration.php" class="nav-link
active">Register</a> </li>';
                }
                else
                {

                    echo ' <li class="nav-item"><a href="your_orders.php"
class="nav-link active">My Orders</a> </li>';
                    echo ' <li class="nav-item"><a href="logout.php"
class="nav-link active">Logout</a> </li>';
                }
            }

            ?>

        </ul>
    </div>
</div>
</nav>

</header>
<div class="page-wrapper">

    <div class="inner-page-hero bg-image" data-image-src="images/img/pimg.jpg">
        <div class="container"> </div>

    </div>

```

```

<div class="result-show">
    <div class="container">
        <div class="row">

            </div>
        </div>
    </div>

    <section class="restaurants-page">
        <div class="container">
            <div class="row">
                <div class="col-xs-12">
                    </div>
                <div class="col-xs-12">
                    <div class="bg-gray">
                        <div class="row">

                            <table class="table table-bordered table-hover">
                                <thead style = "background: #404040; color:white;">
                                    <tr>

                                        <th>Item</th>
                                        <th>Quantity</th>
                                        <th>Price</th>
                                        <th>Status</th>
                                        <th>Date</th>
                                        <th>Action</th>

                                    </tr>
                                </thead>
                                <tbody>

                                    <?php

$(query_res= mysqli_query($db,"select * from users_orders where
u_id='".$SESSION['user_id']."'"));

if(!mysqli_num_rows($query_res) > 0 )
{
    echo '<td colspan="6"><center>You
have No orders Placed yet. </center></td>';
}
else
{



}

```

```

while($row=mysqli_fetch_array($query_res))
{
    ?>
    <tr>
        <td data-column="Item"> <?php echo
$row['title']; ?></td>
        <td data-column="Quantity"> <?php
echo $row['quantity']; ?></td>
        <td data-column="price">$<?php echo
$row['price']; ?></td>
        <td data-column="status">
<?php
    $status=$row['sta
tus'];
    if($status=="" or
$status=="NULL")
    {
        ?>
        <button
type="button" class="btn btn-info"><span class="fa fa-bars"  aria-hidden="true" ></span>
Dispatch</button>
        <?php
    }
    if($status=="i
n process")
    {
        ?>
        <button
type="button" class="btn btn-warning"><span class="fa fa-cog fa-spin"  aria-hidden="true"
></span> On The Way!</button>
        <?php
    }
    if($status=="clos
ed")
    {
        ?>
        <button
type="button" class="btn btn-success" ><span  class="fa fa-check-circle" aria-
hidden="true"></span> Delivered</button>
        <?php
    }
    ?>
    <?php
    if($status=="reje
cted")
    {

```

```

        ?>
        <button
type="button" class="btn btn-danger"> <i class="fa fa-close"></i> Cancelled</button>
        <?php
    }
?>

                </td>
                <td data-column="Date"> <?php echo
$row[ 'date']; ?></td>
                <td data-column="Action"> <a
href="delete_orders.php?order_del=<?php echo $row[ 'o_id'];?>" onclick="return confirm('Are
you sure you want to cancel your order?');" class="btn btn-danger btn-flat btn-addon btn-xs
m-b-10"><i class="fa fa-trash-o" style="font-size:16px"></i></a>
                </td>

            </tr>

        <?php } } ?>

                </tbody>
            </table>

        </div>

    </div>

    </div>

```

```

        </div>
    </section>

<footer class="footer">
    <div class="row bottom-footer">
        <div class="container">
            <div class="row">
                <div class="col-xs-12 col-sm-3 payment-options color-gray">
                    <h5>Payment Options</h5>
                    <ul>
                        <li>
                            <a href="#">  </a>
                        </li>
                        <li>
                            <a href="#">  </a>
                        </li>
                    </ul>
                </div>
                <div class="col-xs-12 col-sm-4 address color-gray">
                    <h5>Address</h5>
                    <p>Wadada Madax-Tooyada ,Hargiesa
TG JH, 09010</p>
                    <h5>Phone: +252 63 3936074</a></h5> </div>
                <div class="col-xs-12 col-sm-5 additional-info color-gray">
                    <h5>Addition informations</h5>
                    <p>Join thousands of other restaurants who benefit from
having partnered with us.</p>
                </div>
            </div>
        </div>
    </div>

```

```

        </div>
    </footer>

</div>

<script src="js/jquery.min.js"></script>
<script src="js/tether.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/animstion.min.js"></script>
<script src="js/bootstrap-slider.min.js"></script>
<script src="js/jquery.isotope.min.js"></script>
<script src="js/headroom.js"></script>
<script src="js/foodpicky.min.js"></script>
</body>

</html>
<?php
}
?>
```

2. Admin Panel Pages

Admin Panel - Main dashboard for overall admin functions.

```

<!DOCTYPE html>
<html lang="en">
<?php
include("../connection/connect.php");
error_reporting(0);
session_start();
if(empty($_SESSION["adm_id"]))
{
    header('location:index.php');
}
else
{
?>
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">
    <title>Admin Panel</title>
    <link href="css/lib/bootstrap/bootstrap.min.css" rel="stylesheet">
    <link href="css/helper.css" rel="stylesheet">
```

```

<link href="css/style.css" rel="stylesheet">
</head>

<body class="fix-header">

    <div class="preloader">
        <svg class="circular" viewBox="25 25 50 50">
            <circle class="path" cx="50" cy="50" r="20" fill="none" stroke-width="2" stroke-miterlimit="10" />
        </svg>
    </div>

    <div id="main-wrapper">

        <div class="header">
            <nav class="navbar top-navbar navbar-expand-md navbar-light">

                <div class="navbar-header">
                    <a class="navbar-brand" href="dashboard.php">

                        <span></span>
                    </a>
                </div>

                <div class="navbar-collapse">
                    <ul class="navbar-nav mr-auto mt-md-0">
                        </ul>

                        <li class="nav-item dropdown">
                            <a class="nav-link dropdown-toggle text-muted" href="#" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false"></a>
                            <div class="dropdown-menu dropdown-menu-right animated zoomIn">
                                <ul class="dropdown-user">
                                    <li><a href="logout.php"><i class="fa fa-power-off"></i> Logout</a></li>
                                </ul>
                            </div>
                        </li>
                    </ul>
                </div>
            </nav>
        </div>
    
```

```

<div class="left-sidebar">

    <div class="scroll-sidebar">

        <nav class="sidebar-nav">
            <ul id="sidebarnav">
                <li class="nav-devider"></li>
                <li class="nav-label">Home</li>
                <li> <a href="dashboard.php"><i class="fa fa-tachometer"></i><span>Dashboard</span></a>
                    </li>
                    <li class="nav-label">Log</li>
                    <li> <a href="all_users.php"> <span><i class="fa fa-user f-s-20"></i></span><span>Users</span></a></li>
                        <li> <a class="has-arrow " href="#" aria-expanded="false"><i class="fa fa-archive f-s-20 color-warning"></i><span class="hide-menu">Restaurant</span></a>
                            <ul aria-expanded="false" class="collapse">
                                <li><a href="all_restaurant.php">All Restaurant</a></li>
                                <li><a href="add_category.php">Add Category</a></li>
                                <li><a href="add_restaurant.php">Add Restaurant</a></li>
                            </ul>
                        </li>
                        <li> <a class="has-arrow " href="#" aria-expanded="false"><i class="fa fa-cutlery" aria-hidden="true"></i><span class="hide-menu">Menu</span></a>
                            <ul aria-expanded="false" class="collapse">
                                <li><a href="all_menu.php">All Menues</a></li>
                                <li><a href="add_menu.php">Add Menu</a></li>
                            </ul>
                        </li>
                        <li> <a href="all_orders.php"><i class="fa fa-shopping-cart" aria-hidden="true"></i><span>Orders</span></a></li>
                    </ul>
                </li>
            </ul>
        </nav>

    </div>

<div class="page-wrapper">

```

```

<div class="container-fluid">
<div class="col-lg-12">
    <div class="card card-outline-primary">
        <div class="card-header">
            <h4 class="m-b-0 text-white">Admin Dashboard</h4>
        </div>
    <div class="row">

        <div class="col-md-3">
            <div class="card p-30">
                <div class="media">
                    <div class="media-left meida media-middle">
                        <span><i class="fa fa-building" style="color:#357ae8;font-size: 2.5em;"></i></span>
                    </div>
                    <div class="media-body media-text-right">
                        <h2><?php $sql="select * from restaurant"; $result=mysqli_query($db,$sql); $rws=mysqli_num_rows($result); echo $rws;?></h2>
                        <p class="m-b-0">Restaurants</p>
                    </div>
                </div>
            </div>
        </div>
    </div>

    <div class="col-md-3">
        <div class="card p-30">
            <div class="media">
                <div class="media-left meida media-middle">
                    <span><i class="fa fa-cutlery f-s-40" aria-hidden="true" style="color:#17a2b8; font-size: 2.5em;"></i></span>
                </div>
                <div class="media-body media-text-right">
                    <h2><?php $sql="select * from dishes"; $result=mysqli_query($db,$sql); $rws=mysqli_num_rows($result); echo $rws;?></h2>
                    <p class="m-b-0">Dishes</p>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

<div class="col-md-3">
    <div class="card p-30">
        <div class="media">
            <div class="media-left meida media-middle">
                <span><i class="fa fa-users f-s-40" style="color:#d8ad2e;font-size: 2.5em;"></i></span>
            </div>
            <div class="media-body media-text-right">
                <h2><?php $sql="select * from users";
                    $result=mysqli_query($db,$sql);
                    $rws=mysqli_num_rows($result);

                    echo $rws;?></h2>
                <p class="m-b-0">Users</p>
            </div>
        </div>
    </div>
</div>

<div class="col-md-3">
    <div class="card p-30">
        <div class="media">
            <div class="media-left meida media-middle">
                <span><i class="fa fa-shopping-cart f-s-40" aria-hidden="true" style="color:#9466de; font-size: 2.5em;"></i></span>
            </div>
            <div class="media-body media-text-right">
                <h2><?php $sql="select * from users_orders";
                    $result=mysqli_query($db,$sql);
                    $rws=mysqli_num_rows($result);

                    echo $rws;?></h2>
                <p class="m-b-0">Total Orders</p>
            </div>
        </div>
    </div>
</div>

<div class="row">
    <div class="col-md-4">
        <div class="card p-30">
            <div class="media">
                <div class="media-left meida media-middle">

```

```



## <?php $sql="select * from res_category"; $result=mysqli_query($db,$sql); $rws=mysqli_num_rows($result); echo $rws;?></h2> Food Categories



## <?php $sql="select * from users_orders WHERE status = 'in process' ";



Processing Orders



## <?php $sql="select * from users_orders WHERE status = 'closed' ";


```

```

        $rws=mysqli_num_rows($result);

        echo $rws;?></h2>
        <p class="m-b-0">Delivered Orders</p>
    </div>
</div>
</div>
</div>
</div>

<div class="row">
    <div class="col-md-4">
        <div class="card p-30">
            <div class="media">
                <div class="media-left meida media-middle">
                    <span><i class="fa fa-times-circle" style="color:#dc3545; font-size: 2.5em;" aria-hidden="true"></i></span>
                </div>
                <div class="media-body media-text-right">
                    <h2><?php $sql="select * from users_orders WHERE status = 'rejected' ";

                    $result=mysqli_query($db,$sql);
                    $rws=mysqli_num_rows($result);

                    echo $rws;?></h2>
                    <p class="m-b-0">Cancelled Orders</p>
                </div>
            </div>
        </div>
    </div>
</div>

<div class="col-md-4">
    <div class="card p-30">
        <div class="media">
            <div class="media-left meida media-middle">
                <span><i class="fa fa-usd f-s-40" aria-hidden="true"></i></span>
            </div>
            <div class="media-body media-text-right">
                <h2><?php
                    $result = mysqli_query($db, 'SELECT SUM(price) AS value_sum FROM users_orders WHERE status = "closed"');
                    $row = mysqli_fetch_assoc($result);
                    $sum = $row['value_sum'];
                    echo $sum;
                ?></h2>
            </div>
        </div>
    </div>
</div>

```

```

                    <p class="m-b-0">Total Earnings</p>
                </div>
            </div>
        </div>
    </div>
</div>

<script src="js/lib/jquery/jquery.min.js"></script>
<script src="js/lib/bootstrap/js/popper.min.js"></script>
<script src="js/lib/bootstrap/js/bootstrap.min.js"></script>
<script src="js/jquery.slimscroll.js"></script>
<script src="js/sidebarmenu.js"></script>
<script src="js/lib/sticky-kit-master/dist/sticky-kit.min.js"></script>
<script src="js/custom.min.js"></script>
<style>

@import url("https://fonts.googleapis.com/css?family=Poppins:400,500,700");

* {
    outline: none;
}

html,
body {
    height: 100%;
    width: 100%;
    font-family: "Poppins", sans-serif;
    color: #222;
}

.navbar {
    padding: 0.8rem;
}

.navbar-brand {
    font-family: Verdana, Geneva, Tahoma, sans-serif;
    font-weight: bold;
    font-size: 1.5rem;
    color: #6648b1 !important;
}

.navbar li {
    padding-right: 20px;
}

.nav-link {
    font-size: 1.1em !important;
}

```

```

.dropdown:hover .dropdown-menu {
  display: block;
}
.dropdown-menu {
  margin-top: 0;
}
.card {
  background: #ffffff none repeat scroll 0 0;
  margin: 15px 0;
  padding: 20px;
  border: 0 solid #484848;
  border-radius: 5px;
  box-shadow: 0 5px 20px rgb(107 107 107 / 23%);
}
.media {
  display: -webkit-box;
  display: -ms-flexbox;
  display: flex;
  -webkit-box-align: start;
  -ms-flex-align: start;
  align-items: flex-start;
}
.media-text-right {
  text-align: right;
}
.media-body {
  vertical-align: middle;
}
.media-body p {
  color: #99abb4;
  line-height: 15px;
}
#iarchive {
  color: yellow;
  font-size: 1.5em;
}
@media (max-width: 768px) {
  .card {
    display: inline-block;
    width: 100%;
  }
}

```

```
.radius {
  border-radius: 4px;
}

.img-responsive {
  width: 100%;
  height: auto;
  display: inline-block;
}

.shadow-container {
  border-radius: 4px;
  padding: 30px;
  box-shadow: 0 0 10px #333;
}

/* for login page */
.form-container {
  padding: 30px;
  border-radius: 10px;
  box-shadow: 0 0 10px 0 #000;
  width: 50%;
}
.form-control {
  border-radius: 10px;
}
.form-control:focus {
  box-shadow: none;
  border: 1.5px solid #0779e4;
}
input[placeholder] {
  font-weight: 500;
}
/* for form-validation */
.form-group .success {
  border-color: #2ecc71;
}

/* for number fields in forms */
input::-webkit-outer-spin-button,
input::-webkit-inner-spin-button {
  -webkit-appearance: none;
  margin: 0;
}
```

```
</style>
```

```
</body>
```

```
</html>
```

```
<?php
```

```
}
```

```
?>dff
```

Login - Customer login page.

```
<!DOCTYPE html>
<html lang="en" >
<?php
include("../connection/connect.php");
error_reporting(0);
session_start();
if(isset($_POST['submit']))
{
    $username = $_POST['username'];
    $password = $_POST['password'];

    if(!empty($_POST["submit"]))
    {
        $loginquery ="SELECT * FROM admin WHERE username='".$username' &&
password='".$_.md5($password)."'";
        $result=mysqli_query($db, $loginquery);
        $row=mysqli_fetch_array($result);

        if(is_array($row))
        {
            $_SESSION["adm_id"] = $row['adm_id'];
            header("refresh:1;url=dashboard.php");
        }
    }
    else
    {
        echo "<script>alert('Invalid Username or Password!');</script>";
    }
}

?>
```

```

<head>
    <meta charset="UTF-8">
    <title>Admin Login</title>

        <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/meyer-reset/2.0/reset.min.css">

        <link rel='stylesheet prefetch'
        href='https://fonts.googleapis.com/css?family=Roboto:400,100,300,500,700,900'>
<link rel='stylesheet prefetch'
href='https://fonts.googleapis.com/css?family=Montserrat:400,700'>
<link rel='stylesheet prefetch' href='https://maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-awesome.min.css'>

        <link rel="stylesheet" href="css/login.css">

</head>

<body>

<div class="container">
    <div class="info">
        
        <h1>Admin Panel </h1>
    </div>
</div>
<div class="form">
    <div class="thumbnail"></div>
    <span style="color:red;"><?php echo $message; ?></span>
    <span style="color:green;"><?php echo $success; ?></span>
    <form class="login-form" action="index.php" method="post">
        <input type="text" placeholder="Username" name="username"/>
        <input type="password" placeholder="Password" name="password"/>
        <input type="submit" name="submit" value="Login" />
    <a href="forgot_password.php">Forgot Password?</a>
    </form>
</div>
    <script src='http://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.3/jquery.min.js'></script>
    <script src='js/index.js'></script>

```

```
</body>
```

```
</html>
```

Filter Restaurants - Search and filter restaurants in the system.

```
<!DOCTYPE html>
<html lang="en">
<?php
include("../connection/connect.php");
error_reporting(0);
session_start();

?>
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">
    <link rel="icon" type="image/png" sizes="16x16" href="images/favicon.png">
    <title>All Restaurants</title>
    <link href="css/lib/bootstrap/bootstrap.min.css" rel="stylesheet">
    <link href="css/helper.css" rel="stylesheet">
    <link href="css/style.css" rel="stylesheet">
</head>

<body class="fix-header fix-sidebar">

    <div class="preloader">
        <svg class="circular" viewBox="25 25 50 50">
            <circle class="path" cx="50" cy="50" r="20" fill="none" stroke-width="2" stroke-miterlimit="10" />
        </svg>
    </div>

    <div id="main-wrapper">

        <div class="header">
            <nav class="navbar top-navbar navbar-expand-md navbar-light">
                <div class="navbar-header">
                    <a class="navbar-brand" href="dashboard.php">
                        <span></span>
                    </a>
                </div>
            </nav>
        </div>
    </div>
</body>
```

```

<div class="navbar-collapse">

    <ul class="navbar-nav mr-auto mt-md-0">

        </ul>

    <ul class="navbar-nav my-lg-0">

        <li class="nav-item dropdown">

            <div class="dropdown-menu dropdown-menu-right mailbox animated zoomIn">
                <ul>
                    <li>
                        <div class="drop-title">Notifications</div>
                    </li>

                    <li>
                        <a class="nav-link text-center"
                            href="javascript:void(0);"> <strong>Check all notifications</strong> <i class="fa fa-angle-right"></i> </a>
                        </li>
                </ul>
            </div>
        </li>

        <li class="nav-item dropdown">
            <a class="nav-link dropdown-toggle text-muted" href="#" data-
            toggle="dropdown" aria-haspopup="true" aria-expanded="false"></a>
            <div class="dropdown-menu dropdown-menu-right animated zoomIn">
                <ul class="dropdown-user">

                    <li><a href="logout.php"><i class="fa fa-power-off"></i>
Logout</a></li>
                </ul>
            </div>
        </li>
    </ul>
</div>
</nav>

```

```

        </div>

<div class="left-sidebar">

    <div class="scroll-sidebar">

        <nav class="sidebar-nav">
            <ul id="sidebarnav">
                <li class="nav-devider"></li>
                <li class="nav-label">Home</li>
                <li> <a href="dashboard.php"><i class="fa fa-tachometer"></i><span>Dashboard</span></a></li>
                    <li class="nav-label">Log</li>
                    <li> <a href="all_users.php"> <span><i class="fa fa-user f-s-20"></i></span><span>Users</span></a></li>
                        <li> <a class="has-arrow " href="#" aria-expanded="false"><i class="fa fa-archive f-s-20 color-warning"></i><span class="hide-menu">Restaurants</span></a>
                            <ul aria-expanded="false" class="collapse">
                                <li><a href="all_restaurant.php">All Restaurant</a></li>
                                <li><a href="add_category.php">Add Category</a></li>
                                <li><a href="add_restaurant.php">Add Restaurant</a></li>

                            </ul>
                        </li>
                    <li> <a class="has-arrow " href="#" aria-expanded="false"><i class="fa fa-cutlery" aria-hidden="true"></i><span class="hide-menu">Menu</span></a>
                            <ul aria-expanded="false" class="collapse">
                                <li><a href="all_menu.php">All Menues</a></li>
                                <li><a href="add_menu.php">Add Menu</a></li>

                            </ul>
                        </li>
                    <li> <a href="all_orders.php"><i class="fa fa-shopping-cart" aria-hidden="true"></i><span>Orders</span></a></li>

                </ul>
            </li>
        </nav>

    </div>

<div class="page-wrapper">

```

```

<div class="container-fluid">

    <div class="row">
        <div class="col-12">

            <div class="col-lg-12">
                <div class="card card-outline-primary">
                    <div class="card-header">
                        <h4 class="m-b-0 text-white">All Restaurant</h4>
                    </div>

                    <div class="table-responsive m-t-40">
                        <table id="example23" class="display nowrap table table-hover table-striped table-bordered" cellspacing="0" width="100%">
                            <thead class="thead-dark">
                                <tr>
                                    <th>Category</th>
                                    <th>Name</th>
                                    <th>Email</th>
                                    <th>Phone</th>
                                    <th>Url</th>
                                    <th>Open Hrs</th>
                                    <th>Close Hrs</th>
                                    <th>Open Days</th>
                                    <th>Address</th>
                                    <th>Image</th>
                                    <th>Date</th>
                                    <th>Action</th>
                                </tr>
                            </thead>

                            <tbody>

                                <?php
                                    $sql="SELECT * FROM restaurant order by rs_id
desc";
                                    $query=mysqli_query($db,$sql);

                                    if(!mysqli_num_rows($query) > 0 )

```

```

        {
            echo '<td colspan="11"><center>No
Restaurants</center></td>';
        }
    else
    {
        while($rows=mysqli_fetch_
array($query))
        {

            $mql="SELECT
* FROM res_category where c_id='".$rows['c_id']."'";

            $res=mysqli_
query($db,$mql);
            $row=mysqli_
fetch_array($res);

            echo '
<tr><td>'.$row['c_name'].'</td>
<td>'.$rows['title'].'</td>
<td>'.$rows['email'].'</td>
<td>'.$rows['phone'].'</td>
<td>'.$rows['url'].'</td>

<td>'.$rows['o_hr'].'</td>
<td>'.$rows['c_hr'].'</td>
<td>'.$rows['o_days'].'</td>

<td>'.$rows['address'].'</td>

<td><div class="col-md-3 col-lg-8 m-b-10">

```

```

<center></center>

</div></td>

<td>' . $rows['date'] . '</td>

    <td><a href="delete_restaurant.php?res_del=' . $rows['rs_id'] . '" class="btn btn-danger btn-flat btn-addon btn-xs m-b-10"><i class="fa fa-trash-o" style="font-size:16px"></i></a>

        <a href="update_restaurant.php?res_upd=' . $rows['rs_id'] . '" class="btn btn-info btn-flat btn-addon btn-sm m-b-10 m-l-5"><i class="fa fa-edit"></i></a>

    </td></tr>';

}

}

?>

                </tbody>
            </table>
        </div>
    </div>
</div>

```

```
        </div>

            </div>
        </div>
    </div>
</div>

</div>

<footer class="footer"> © 2024 - KAYSE EATS </footer>

</div>

</div>

<script src="js/lib/jquery/jquery.min.js"></script>
<script src="js/lib/bootstrap/js/popper.min.js"></script>
<script src="js/lib/bootstrap/js/bootstrap.min.js"></script>
<script src="js/jquery.slimscroll.js"></script>
<script src="js/sidebarmenu.js"></script>
<script src="js/lib/sticky-kit-master/dist/sticky-kit.min.js"></script>
<script src="js/custom.min.js"></script>
<script src="js/lib/datatables/datatables.min.js"></script>
<script
src="js/lib/datatables/cdn.datatables.net/buttons/1.2.2/js/dataTables.buttons.min.js"></script>
<script
src="js/lib/datatables/cdn.datatables.net/buttons/1.2.2/js/buttons.flash.min.js"></script>
<script
src="js/lib/datatables/cdnjs.cloudflare.com/ajax/libs/jszip/2.5.0/jszip.min.js"></script>
<script
src="js/lib/datatables/cdn.rawgit.com/bpampuch/pdfmake/0.1.18/build/pdfmake.min.js"></script>
<script
src="js/lib/datatables/cdn.rawgit.com/bpampuch/pdfmake/0.1.18/build/vfs_fonts.js"></script>
<script
src="js/lib/datatables/cdn.datatables.net/buttons/1.2.2/js/buttons.html5.min.js"></script>
<script src="js/lib/datatables/datatables-init.js"></script>
</body>
```

```
</html>
<?php
include("../connection/connect.php");
error_reporting(0);
session_start();

mysqli_query($db,"DELETE FROM users WHERE u_id = '".$_GET['user_del']."' ");
header("location:all_users.php");

?>

<?php
include("../connection/connect.php");
error_reporting(0);
session_start();

mysqli_query($db,"DELETE FROM res_category WHERE c_id = '".$_GET['cat_del']."' ");
header("location:add_category.php");

?>

<?php
include("../connection/connect.php");
error_reporting(0);
session_start();

mysqli_query($db,"DELETE FROM res_category WHERE c_id = '".$_GET['cat_del']."' ");
header("location:add_category.php");

?>
```

3. Implemented Database Codes

The database consists of seven tables:

1. Admin Table

```
CREATE TABLE admin (
    admin_id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) NOT NULL,
    password VARCHAR(255) NOT NULL,
    email VARCHAR(100) NOT NULL
);
```

2. Users Table

```
CREATE TABLE users (
    user_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    password VARCHAR(255) NOT NULL,
    phone VARCHAR(15),
    address TEXT
);
```

3. Dishes Table

```
CREATE TABLE dishes (
    dish_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    price DECIMAL(10, 2) NOT NULL,
    category_id INT,
    restaurant_id INT,
    FOREIGN KEY (category_id) REFERENCES res_category(category_id),
    FOREIGN KEY (restaurant_id) REFERENCES restaurant(restaurant_id)
);
```

4. Restaurant Table

```
CREATE TABLE restaurant (
    restaurant_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    location TEXT NOT NULL,
    contact VARCHAR(50) NOT NULL
```

```
) ;
```

5. Categories Table

```
CREATE TABLE res_category (
    category_id INT AUTO_INCREMENT PRIMARY KEY,
    category_name VARCHAR(100) NOT NULL
);
```

6. Users Orders Table

```
CREATE TABLE users_orders (
    order_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    dish_id INT,
    quantity INT NOT NULL,
    total_price DECIMAL(10, 2) NOT NULL,
    status ENUM('pending', 'in progress', 'canceled', 'delivered') DEFAULT
    'pending',
    order_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    FOREIGN KEY (dish_id) REFERENCES dishes(dish_id)
);
```

7. Remarks Table

```
CREATE TABLE remark (
    remark_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    dish_id INT,
    comment TEXT NOT NULL,
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    FOREIGN KEY (dish_id) REFERENCES dishes(dish_id)
);
```

4. Testing

Testing is a critical part of the development process for any software project. For *Kayse Eats*, the testing process was designed to ensure that the system functions as expected, is easy to use, and can handle all the required tasks without errors. The tests focused on verifying individual components, how they work together, and the overall usability of the system. Below are the types of tests conducted for *Kayse Eats* and their detailed explanations.

4.1 Unit Testing

Unit testing involves checking each small part of the system to ensure it works correctly on its own. For *Kayse Eats*, the following areas were tested:

- **Customer Registration:**

Verified that all required fields (name, email, password, phone, address) are correctly filled in before the account is created.

Example Test Case:

- Input: Empty "email" field.
- Expected Result: Error message prompts the user to fill in the missing information.
- Actual Result: Passed.

- **Login Functionality:**

Tested the login system for both customers and administrators. Passwords entered are hashed and checked against the database.

Example Test Case:

- Input: Correct email but incorrect password.
- Expected Result: Error message "Invalid login credentials."
- Actual Result: Passed.

- **Order Placement:**

Confirmed that customers could add dishes to their cart and place an order successfully.

Example Test Case:

- Input: Selected two dishes and proceeded to checkout.
- Expected Result: Order is placed, and the database records the order details.
- Actual Result: Passed.

4.2 Integration Testing

Integration testing focuses on how different parts of the system work together. This ensures that when one feature relies on another, they function seamlessly.

- **Order System Integration:**

Tested how orders are processed from the customer panel to the admin panel.

Example Test Case:

- Input: Customer places an order. Admin views the order in their panel and updates the status to "Delivered."
- Expected Result: The order status changes and reflects correctly in both the admin and customer panels.
- Actual Result: Passed.

- **Restaurant and Menu Management:**

Verified that adding a restaurant in the admin panel updates the list of available restaurants in the customer panel. Similarly, adding new dishes under a specific restaurant correctly displays them for customers.

Example Test Case:

- Input: Admin adds "Pizza House" and assigns it a menu of dishes.
- Expected Result: "Pizza House" and its dishes appear under the restaurant list for customers.
- Actual Result: Passed.

4.3 System Testing

System testing ensures the entire application works as a whole. Every feature was tested together to confirm the system functions as expected in real-world scenarios.

- **End-to-End Order Process:**

Example Test Case:

- Input: Customer registers, logs in, browses dish, adds items to the cart, and places an order.
- Expected Result: The admin sees the order, updates its status, and the customer can track the order until delivery.
- Actual Result: Passed without errors.

- **Error Handling:**

Tested how the system behaves when something goes wrong, such as entering invalid data or attempting unauthorized actions.

Example Test Case:

- Input: Customer tries to access the admin panel URL without logging in as an admin.
- Expected Result: Redirect to the login page with an error message.
- Actual Result: Passed.

4.4 Acceptance Testing

Acceptance testing is conducted with real users to ensure the system meets their expectations and requirements. For *Kayse Eats*, this involved asking potential customers and administrators to use the system and provide feedback.

- **Ease of Use:**

Tested whether users found the system intuitive and easy to navigate.

Feedback:

- Some My freinds liked the clean design of the customer panel.

- **Performance Testing:**

Checked how the system handles multiple users at the same time.

Example Test Case:

- Simulated 5 persons placing orders simultaneously.
- Expected Result: System remains responsive without crashing or slowing down.
- Actual Result: Passed

CHAPTER 6

SUMMARY AND CONCLUSION

Chapter 6: Summary and Conclusion

6.1 Introduction

The *Kayse Eats* project was developed to address the challenges faced in food ordering and delivery within Somaliland. By creating a practical and user-friendly solution, it bridges the gap between customers and restaurants, ensuring convenience and efficiency. This chapter summarizes the project outcomes, reflects on lessons learned during development, highlights its limitations, and outlines future plans. The project showcases the application of software development skills to solve real-world problems in the local context.

6.2 Factors Facilitating Project Completion

Several factors contributed to the successful development of *Kayse Eats*:

1. **Relevance to Somaliland:** The lack of an efficient online food ordering system in Somaliland motivated the creation of a platform tailored to local needs.
2. **Technological Accessibility:** Using widely available tools like PHP, MySQL, and Bootstrap made the development process straightforward and resource-efficient.
3. **Structured Workflow:** A systematic approach, from planning to testing, ensured steady progress.
4. **Step-by-Step Planning:** Following a structured workflow, including requirement analysis, system design, implementation, and testing, ensured smooth progress.

6.3 Lessons Learned During the Project

The project provided valuable insights and skill-building opportunities:

- **Understanding User Needs:** Learning to analyze local challenges, such as the absence of structured address systems, and designing solutions accordingly.
- **Software Development Skills:** Strengthening knowledge in programming, database management, and UI design.
- **Problem Solving:** Tackling challenges like integrating payment methods and creating user-friendly interfaces.
- **Time Management:** Balancing development milestones and ensuring timely completion.

6.4 Limitations of the Project

While the project achieved its primary goals, a few limitations remain:

1. **Internet Dependency:** The system relies on stable internet connectivity, which can limit its reach in areas with poor access.
2. **Limited Payment Options:** The system supports local mobile money services but does not yet integrate global payment gateways.
3. **Still in Local Server:** Not published to the net.

6.5 Future Plans for the Project

To ensure greater impact and usability, the following enhancements are planned:

1. **Mobile Application Development:** Expanding the platform to mobile apps for wider accessibility in Somaliland.
2. **GPS Integration:** Implementing real-time tracking to improve delivery accuracy and efficiency.
3. **Global Payment Support:** Adding options like Visa and PayPal to attract a broader user base, including international customers.
4. **AI-Powered Recommendations:** Using artificial intelligence to provide personalized restaurant and dish suggestions based on user preferences.
5. **Regional Expansion:** Scaling the platform to include other cities and regions in Somaliland, boosting its impact and user base.

Conclusion

The *Kayse Eats* project successfully addresses a pressing need in Somaliland by creating a functional online food ordering system. It reflects a deep understanding of local challenges and the application of technical skills to solve them effectively. With planned future enhancements, *Kayse Eats* has the potential to transform food ordering and delivery, benefiting both customers and restaurants across Somaliland.

References

1. Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2011). *Systems Analysis and Design in a Changing World* (5th Edition).
2. Rosenblatt, H. J. (2014). *System Analysis and Design* (10th Edition).
3. Kendall, K. E., & Kendall, J. E. (2008). *Systems Analysis and Design* (3rd Edition).