Your Thesis Title

Your Name

March 13, 2024

Abstract

Contents

Chapter 1

Introduction

Chapter 2

Literature Review

2.1 Evolution of Restaurant Management Systems

The evolution of restaurant management systems has seen a shift from traditional paper-based methods to digital systems, driven by the need for improved efficiency and customer experience. Traditional systems, such as paper-based menu cards and manual order taking, have limitations in terms of time consumption, manual errors, and customer dissatisfaction 1.1. The adoption of digital systems, such as tablet food ordering and digital based ordering, has aimed to address these limitations by providing cost and time efficiency benefits for both management and customers 1.22.1. Factors driving the adoption of digital restaurant management systems include the need for faster services, reduced dependency on waiters, which increases productivity. 3.12.2. Additionally, the use of technology to replace some of the jobs done by human beings and make machines do the work has been a driving factor in the adoption of online food ordering management systems 4.1.

The adoption of digital systems has also been influenced by the increasing trend of consumers adopting a more tech-savvy approach to conducting business transactions and leisure activities, leading to the need for restaurant owners to keep up with technological advancements to attract and retain a broader customer base 5.1. Overall, the evolution of restaurant management systems from traditional to digital has been driven by the need for improved efficiency, reduced manual errors, and enhanced customer satisfaction, while also aligning with the changing technological landscape and consumer preferences.

2.2 Digital Ordering Systems

The implementation of digital ordering systems in restaurants has been shown to have a significant impact on customer experience and business outcomes. The use of QR codes for ordering has been designed to provide more advantages, including electronic payment of bills and entertainment facilities, and to reduce the time between ordering and delivering goods to customers 1.1. The system allows customers to place food orders by scanning the QR code on the restaurant table, which then directs them to a digital version of the restaurant's menu, enabling them to place orders directly from their phones 2.1. This automation of the ordering process has been found to reduce the time of order registry to delivery, improving customer satisfaction and business efficiency 1.2. The use of digital menus in restaurants has been shown to provide several benefits, including the elimination of traditional ordering stages, more favourable choices for customers, and the ability to pay bills digitally, which prevents the pollution of money exchange and has a significant effect on protecting the environment due to the reduced use of paper 1.3. The system also allows restaurant owners to have an insightful view of their business data, such as sales data, which can improve decision-making and forecasting demand using data analysis techniques 2.1.

Research has shown that the implementation of digital ordering systems has led to increased customer satisfaction, improved table turnover, and reduced labour and menu costs 3.1. Customers have reported that the QR menu ordering systems provide convenience, value, and enjoyment, leading to increased customer attraction and satisfaction 3.1. Furthermore, the system has been found to improve table turnover and reduce labour and menu costs, leading to improved business efficiency 3.1. In conclusion, the implementation of QR code ordering systems in restaurants has had a positive impact on customer experience and business outcomes, leading to improved customer satisfaction, business efficiency, and insightful views of business data for restaurant owners.

2.3 Customer Behavior and Technology Acceptance Towards Digitalisation

The Unified Theory of Acceptance and Use of Technology (UTAUT) model has been applied to understand customer willingness to use QR code ordering systems in luxury restaurants in Xi'an, China 1.1. The UTAUT model contains four independent variables: performance expectancy, effort expectancy, social influence, and facilitating conditions 1.2. Additionally, the UTAUT 2 model, an extended version of UTAUT, includes hedonic motivation, price value, trust, experience, and habit as independent variables, focusing more on the individual use of technology 1.2. The study found that performance expectancy, effort expectancy, social influence, facilitating conditions, hedonic motivation, price value, and trust positively affect customer intention to use QR menu ordering 1.3. However, effort expectancy and facilitating conditions negatively affect customer behavioural intention 1.3.

Factors influencing customer behaviour and acceptance include performance expectancy (the degree to which technology will benefit consumers performing certain activities), effort expectancy (the degree of ease associated with consumers' use of technology), social influence, facilitating conditions, hedonic motivation, price value, and trust 1.4. These factors have been found to significantly influence customers' behavioural intention to use QR menu ordering systems 1.51.3.

Moreover, the study revealed that trust is an important variable affecting customers' behavioural intention to use the QR menu, especially when they are asked to provide confidential information such as transaction codes or personal details 1.6. This highlights the significance of trust in influencing customer acceptance of QR code ordering systems.

2.4 Challenges and Solutions

The operational challenges faced by restaurants in implementing and managing web-based systems include difficulties in managing customer orders during peak hours, the need for efficient labour scheduling, and the management of customer reservations and waitlists 1.11.2.

Potential solutions to these challenges involve the implementation of a self-ordering System, which can automate the ordering process, provide real-time order tracking, and manage customer reservations efficiently 1.3.

Common technological barriers in implementing web-based systems include compatibility issues with existing systems, scalability concerns, and the need for secure data exchange between different systems 2.1.

These barriers can be overcome by using open-source technologies to maintain low costs, ensuring that the system is scalable to accommodate many users, and implementing secure data exchange protocols to protect customer and business data 2.22.3.

2.5 Current Implementations

There are some QR code ordering solutions reviewed, including solutions such as Toast, Square, Zuppler, TouchBistro, Bbot, Menufy, and Future Ordering. These solutions provide a a lot of functionalities, such as online ordering, and loyalty programs, to specialized tools for creating and managing QR code-based menus. To summarise it, each solution presents features such as real-time menu updates, and data analytics. These solutions highlights the importance of selecting a system that aligns with a restaurant's specific operational needs, budget, and customer engagement goals, emphasizing the role of digital technology in transforming the traditional dining model into a more interactive, convenient, and streamlined process.

Future Ordering is one of the most notable solutions in the UK as several big restaurant names such as KFC, Nandos and Burger King have implemented them. Future Ordering provides a digital ordering platform for food and beverage businesses, focusing on app, web, and kiosk channels. Their system supports various user journeys, including curbside, click'n'collect, and table ordering, and offers comprehensive management tools for digital channels. However, Future Ordering requires you to have their proprietary hardware which can be a barrier in terms of cost for smaller restaurants.

2.6 Research Gaps

While the adoption of digital ordering systems and the implementation of technologies like QR codes have greatly improved the efficiency of order-taking and the overall customer experience, there is a significant gap regarding personalized customer experiences through these digital systems. Most existing systems, as discussed, focus on streamlining operations, reducing errors, improving table turnover, and providing insights into business data. However, there's a scarcity of discussion around how these systems can leverage customer data to offer personalized dining experiences.

Personalization in restaurant management systems can significantly enhance customer satisfaction and loyalty by making customers feel valued and understood. By tracking previous orders and preferences, restaurants can offer tailored recommendations, which not only improves the customer experience but can save time during ordering as the recommendations are more likely to be accepted.

Another gap is the lack of solution for bill splitting. Bill splitting is a common issue in restaurants, especially when dining in groups. It can be a time-consuming and error-prone process for both customers and staff. A digital ordering system that can automate the bill splitting process can significantly improve the customer experience and reduce the workload on staff.

2.7 Conclusion

To conclude this literature review, the evolution of restaurant management systems has seen a shift from traditional paper-based methods to digital systems, driven by the need for improved efficiency and customer experience. The adoption of digital systems has been influenced by the increasing trend of consumers adopting a more tech-savvy approach to conducting business transactions and leisure activities, leading to the need for restaurant owners to keep up with technological advancements to attract and retain a broader customer base.

The implementation of digital ordering systems in restaurants has been shown to have a significant impact on customer experience and business outcomes, leading to improved customer satisfaction, business efficiency, and insightful views of business data for restaurant owners. The Unified Theory of Acceptance and Use of Technology (UTAUT) model has been applied to understand customer willingness to use QR code ordering systems in luxury restaurants in Xi'an, China. Factors influencing customer behaviour and acceptance include performance expectancy, effort expectancy, social influence, facilitating conditions, hedonic motivation, price value, and trust. These factors have been found to significantly influence customers' behavioural intention to use QR menu ordering systems.

The operational challenges faced by restaurants in implementing and managing web-based systems include difficulties in managing customer orders during peak hours, the need for efficient labour scheduling, and the management of customer reservations and waitlists. Common technological barriers in implementing web-based systems include compatibility issues with existing systems, scalability concerns, and the need for secure data exchange between different systems.

There are some QR code ordering solutions reviewed, including solutions such as Toast, Square, Zuppler, TouchBistro, Bbot, Menufy, and Future Ordering. These solutions provide a lot of functionalities, such as online ordering, and loyalty programs, to specialized tools for creating and managing QR code-based menus.

However, there is a significant gap regarding personalized customer experiences through these digital systems and the lack of solution for bill splitting. Personalization in restaurant management systems can significantly enhance customer satisfaction and loyalty by making customers feel valued and understood. By tracking previous orders and preferences, restaurants can offer tailored recommendations, which not only improves the customer experience but can save time during ordering as the recommendations are more likely to be accepted. A digital ordering system that can automate the bill splitting process can significantly improve the customer experience and reduce the workload on staff. This research aims to address these gaps by proposing a digital ordering system that leverages customer data to offer personalized dining experiences and automate the bill splitting process. The next chapter will discuss the research methodology used to achieve this goal.

Chapter 3

Product Specification

3.1 Empirical Exploration

This section describes the preparatory empirical work performed for this project. It aims to define the scope of the final solution by presenting a preliminary discussion with restaurant owners and staff about their current practices and needs. The goal is to understand the current state of the art in the restaurant industry and to identify the main challenges and opportunities for the development of a new system. This section also presents the results of a preliminary survey conducted with potential users of the system, which aims to identify the main requirements and expectations of the system.

3.1.1 Interviews with Restaurant Owners and Staff

The first step in the empirical exploration was to conduct interviews with restaurant owners and staff. The goal was to understand the current state of the art in the restaurant industry and to identify the main challenges and opportunities for the development of a new system. The interviews were conducted with the help of a semi-structured questionnaire, which was designed to guide the conversation and ensure that all relevant topics were covered. The questionnaire was divided into three main sections: the current practices and challenges of the restaurant, the potential benefits and opportunities of a new system, and the main requirements and expectations for the system. The interviews were conducted with a total of 2 restaurant owners, one is the owner of a small restaurant and the other is the owner of a multi-branch restaurant, and 6 restaurant staffs.

Chapter 4

Methodology

4.1 Overview

This chapter outlines the systematic methodology employed to design, develop, and evaluate a web-based restaurant management application. The application leverages modern web technologies, including a React.js frontend, to enhance the dining experience by enabling customers to place orders using QR codes, receive personalized menu recommendations, and choose flexible bill-sharing options. The development process is guided by user-centered design principles, aiming to improve customer satisfaction and operational efficiency for restaurants.

4.2 System Design

The web application is built using React.js frontend paired with Django backend, with a SQLite database. The frontend is responsible for rendering the user interface and handling user interactions, while the backend manages the business logic and data storage. The frontend and backend communicate via Djano RESTful APIs using Axios from React, enabling the application to be scalable and maintainable. The application can hosted on a cloud platform, such as AWS or Heroku or Railway.

4.2.1 Backend Architecture

- **Models**: Django models define the database schema. Each model corresponds to a database table and is used for creating, retrieving, updating, and deleting records. For instance, models like Place, Category, MenuItem, and Order represent the core entities of the restaurant management system.

- **Views**: Django views handle the business logic of the application. They receive web requests and return web responses. Views interact with models and return JSON responses (in the case of a REST API used by a React frontend).

- **URL Routing**: Django's URL routes incoming requests to the appropriate view based on the request URL. This is where you define URL patterns for your API endpoints.

- **Django REST Framework**: For integrating with a React frontend, Django REST Framework (DRF) is used to build a RESTful API. DRF provides serializers for converting complex data types to and from JSON and viewsets for handling common API operations.

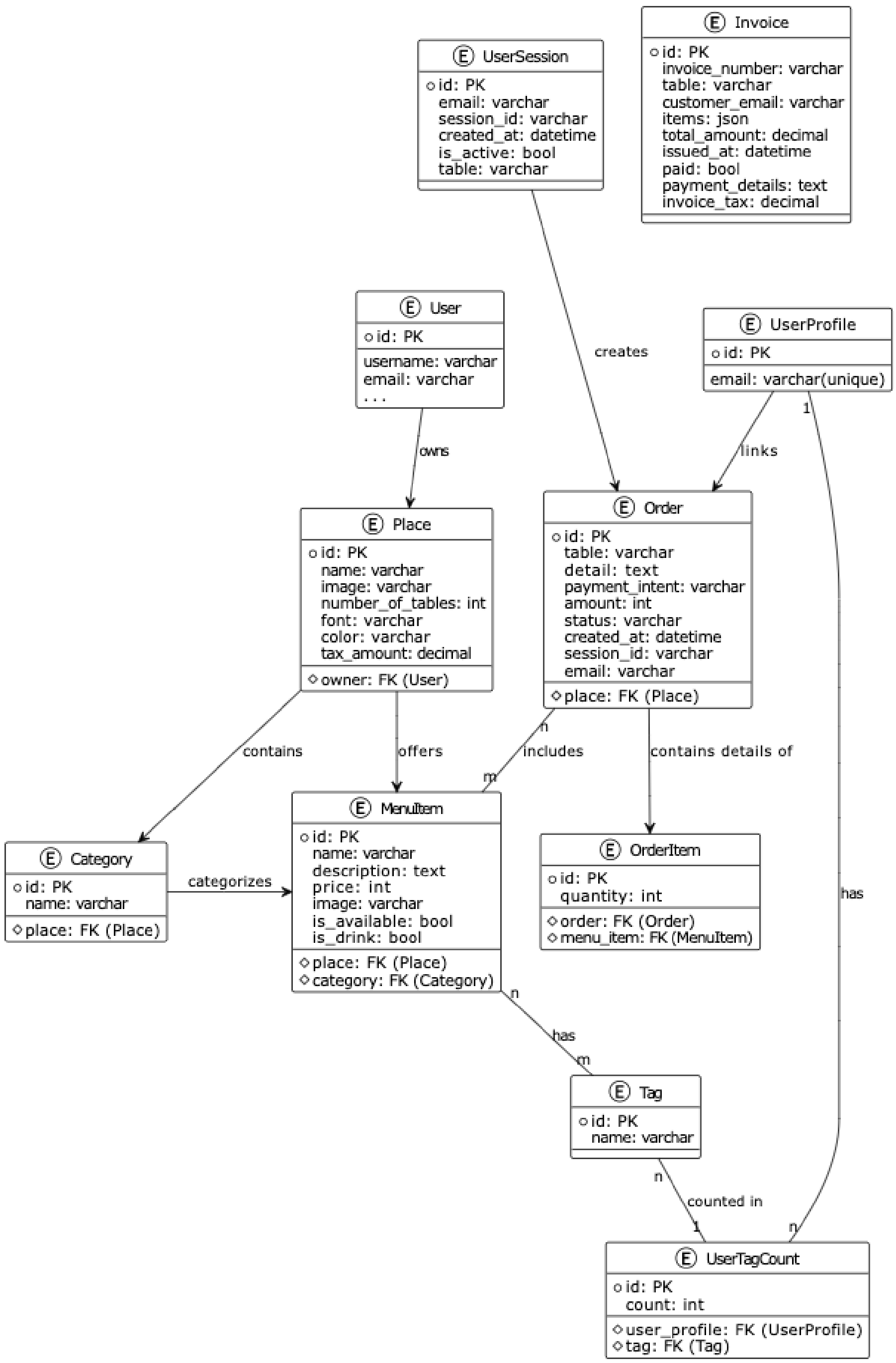
- **Authentication and Authorization**: Django comes with a built-in authentication system, and DRF offers additional mechanisms for API authentication, such as token authentication.

- **Database**: Django supports various databases like PostgreSQL, MySQL, SQLite, and Oracle. However, SQLite was chosen for its simplicity and ease of use.

4.2.2 Frontend Architecture

- **React Components**: The frontend is built using React.js. React components are used to create reusable UI elements, such as buttons, forms, and menus.
- **State Management**: React's state and context API are used to manage the application's state, such as user authentication, order details, and session information.
- **Routing**: React Router is used for client-side routing, allowing the application to navigate between different views without a page refresh.
- **API Integration**: Axios is used to make HTTP requests to the Django backend, enabling the frontend to fetch and send data to the server.
- **User Interface**: The frontend is designed to be responsive and user-friendly, with a focus on providing an intuitive and visually appealing experience for customers.
- **Bulma SCSS**: On top of using React Components, additional Bulma SCSS components were used to style the application, this allows the application to be responsive and have uniform styling.

4.3 Database Architecture



4.3.1 Place

Represents a restaurant or place that uses the app. It includes details like the owner (linked to the Django `User` model), name, image, number of tables, and aesthetic customizations (font and color). A new field, `tax_amount`, has been added to manage tax calculations.

4.3.2 Category

Organizes menu items into categories (e.g., appetizers, entrees, desserts) for each place. This helps in structuring the menu for easier navigation by the customers.

4.3.3 MenuItem

Details about each menu item, including the place it belongs to, its category, name, description, price, and availability. It also specifies whether the item is a drink and allows for tagging (e.g., vegan, spicy) through a many-to-many relationship with the `Tag` model.

4.3.4 Tag

Used to label menu items with specific attributes, enhancing the ability to offer personalized recommendations based on customer preferences.

4.3.5 UserSession

Tracks active sessions by customers, including their email, session ID, and table number. It supports functionalities like order tracking and session management.

4.3.6 Order

Records details of customer orders, including the items ordered (linked through `OrderItem`), the table and place, payment information, and status. It also tracks the session and customer email.

4.3.7 OrderItem

A through model for the many-to-many relationship between `Order` and `MenuItem`, capturing the quantity of each menu item ordered.

4.3.8 Invoice

Manages billing information, including the items ordered, total amount, and payment status. It introduces the `items` field as a JSON structure to store ordered items, allowing for flexible representation of order details.

4.3.9 UserProfile

Represents a user's profile with a unique email. It is linked to `Tag` through `UserTagCount` to track preference metrics.

4.3.10 UserTagCount

Tracks the count of tags associated with a user's orders, facilitating the personalization engine to recommend items based on preferred tags.

4.4 Digital Menu Implementation

QR codes placed on each table serve as the entry point for customers to access the menu and place orders. Upon scanning the QR code with a smartphone, customers are directed to a web interface where they can browse the menu, make selections, and specify their order preferences.

4.4.1 Menu Display

The menu is organized into categories, with each item displaying its name, description, price, and availability. The user interface is designed to be visually appealing and easy to navigate, with a responsive layout that adapts to different screen sizes.

4.4.2 Order Placement

Customers can add items to their order by clicking on the menu items and specifying the quantity. The system tracks the items ordered and the customer's session, allowing for order tracking and billing.

4.4.3 Personalized Recommendations

The application uses the customer's order history and preferences to recommend menu items that match their taste. This is achieved through methods in the `UserProfile` model, which calculate preferences based on past orders and tag popularity. The algorithm takes into account the most ordered tags and preferred price range to recommend items that align with the customer's preferences.

4.5 Recommendation Algorithm

The recommendation algorithm embedded within the `UserProfile` model of our restaurant management application is designed to offer personalized menu suggestions to users. It analyzes users' past orders, preferences, and price sensitivity to recommend items that align with their tastes.

4.5.1 User Preferences and Order History Analysis

- The algorithm commences by evaluating the user's previous orders to deduce preferences. This involves:
  - Calculating the average price of the items ordered to establish a preferred price range.
  - Identifying the most ordered tags to understand the user's food preferences.

4.5.2 Tag-Based Recommendations

- Users are linked to tags via the `UserTagCount` model, reflecting their taste preferences.
- Menu items are scored based on their relevance to the user's preferred tags, with higher scores for better-matched items.

4.5.3 Price Sensitivity

- The algorithm adjusts scores for items within the user's preferred price range, favoring those that align with past spending habits.

4.5.4 Personalized Recommendations

- For food items: Excludes drinks, scoring food items based on tag preferences and price sensitivity.
- For drink items: Similar process, but focuses solely on beverages, tailoring recommendations accordingly.

4.5.5 Combining Recommendations for Combo Meals

- Creates combo meal suggestions by pairing the top food and drink recommendations.

4.5.6 Example Usage

Step 1: Determining the Preferred Price Range

Assume the user has ordered the following items:

- Pizza: $12 (Quantity: 2)
- Burger: $9 (Quantity: 1)
- Salad: $7 (Quantity: 3)

Calculation

```
Total spent on Pizza = $12 * 2 = $24
Total spent on Burger = $9 * 1 = $9
Total spent on Salad = $7 * 3 = $21

Total spent = $24 + $9 + $21 = $54
Total quantity of items ordered = 2 + 1 + 3 = 6

Average price per item = Total spent / Total quantity = $54 / 6 = $9
```

Preferred price range is calculated as ± 20 percent of the average price:

```
Lower bound = $9 * 0.8 = $7.20
Upper bound = $9 * 1.2 = $10.80
```

Step 2: Identifying Most Ordered Tags

Given tags for each item:

- Pizza: ["Italian", "Cheese"]
- Burger: ["American", "Beef"]
- Salad: ["Vegetarian", "Healthy"]

The most ordered tags based on quantity are "Vegetarian" and "Healthy".

Step 3: Scoring and Recommending Menu Items

New menu items to recommend:

- Spaghetti: $8 ["Italian", "Vegetarian"]
- Chicken Wrap: $10 ["Healthy", "Chicken"]
- Fish Tacos: $11 ["Seafood"]
- Veggie Burger: $9 ["Vegetarian", "American"]

Scoring

Spaghetti scores high for matching "Vegetarian" and being within the price range. Chicken Wrap scores for "Healthy" and being within the price range. Fish Tacos score lower due to being outside the price range and no matching tags. Veggie Burger scores for "Vegetarian" and being within the price range.

Results

Based on the scoring, the recommended items in order are:

1. Spaghetti
2. Chicken Wrap
3. Veggie Burger

4.6 Flexible Bill Sharing

This section outlines the systematic steps taken to design, implement, and evaluate this bill splitting implementation.

4.6.1 Design Considerations

- **User Experience**: A primary focus was to ensure an intuitive and hassle-free interaction with the bill-sharing feature, allowing customers to easily choose between individual payments or sharing the bill with others.
- **Accuracy and Flexibility**: The feature needed to accurately calculate and split the bill according to the specific orders and preferences of the diners, providing flexibility in how the bill could be shared.
- **Session Management**: The system has to manage individual ordering sessions and group sessions, allowing customers to join and leave sessions as needed, while ensuring accurate tracking of orders and payments.

4.6.2 Implementation

- **Order Placement via individual Sessions**: When customers scan the QR code at their table, they have the option to create an individual ordering session by entering their email address. This process ensures that each diner's selections are tracked separately, even though they are physically sitting at the same table.
- **Joining Sessions for Shared Billing**: The application provides an option for customers to join a session created by another diner at the same table. This feature is particularly useful for groups who want to combine their orders into a single bill, allowing easy payment splitting without the need for manual calculations or multiple transactions.
- **Separate Payments or Single Payment**: At the end of the meal, the group can decide whether to pay separately or together. If they opt for individual payments, each diner pays only for their ordered items. For a shared bill, the total cost is evenly divided among the participants in the joined session, or the payment can be split according to the specific items each person ordered, depending on the application's capabilities and settings.

4.7 Admin Panel

4.7.1 Features

- **Menu Management**: The adminstrator would be able to manage the menu, including adding, updating, and deleting menu items and categories. The menu management system is where the admin can add new items, update existing items, and delete items that are no longer available. The admin can also create new categories and assign items to specific categories. This is where the admin would specify the Tags for each item which would be used for the recommendation algorithm.

- **Order Tracking**: Provides real-time tracking of customer orders, allowing staff to monitor and manage orders efficiently. This feature allows the adminstrator to keep track of which tables are occupied, the time since the last order, and the status of each order.

- **Billing and Invoice Management**: The adminstrator can manage billing and invoices, for each table and would be able to see all the separate sessions that are on that table. This feature also allows the adminstrator to view and manage the billing and invoices for each table, including the ability to split bills and track payments.

- **Sales Analysis**: The adminstrator can view sales data such as daily, weekly and monthly sales. The adminstrator can also view the most popular items and the least popular items. This feature also displays the peak hours of the day which along side with the popular items can be used to make decisions on inventory.

- **Global Settings**: The adminstrator can manage global settings such as tax rates, menu themes and fonts. They can also change the name of the restaurant and the logo.

4.8 Kitchen Display System

4.8.1 Features

- **Real-time Order Display**: The kitchen display system provides a real-time display of incoming orders, allowing kitchen staff to prepare and manage orders. This feature allows the kitchen staff to view the orders as they come in, and mark them as in progress or completed.

- **Separate Displays**: The system can display orders for different stations on separate screens, allowing staff to focus on their specific tasks. This feature allows the kitchen staff to view the orders that are relevant to their station, and manage them accordingly. Such as the bar which would only display drink orders.

4.9 Floor Staff Display

4.9.1 Features

- **Order Tracking**: The floor staff would be able to see the food and drink orders that have been completed and show them where to deliver the orders. This feature allows the floor staff to view the orders that are ready to be delivered, and manage them accordingly. Once the order is served, the floor staff can mark the order as served.

4.10 Order Tracking

Chapter 5

Evaluation

5.1 Usability Evaluation

5.2 Performance Evaluation

5.3 Security Evaluation

5.4 Comparative Evaluation

Chapter 6

Discussion

Chapter 7

Conclusion

Bibliography