Project Proposal

on

SmartServe: A Digital Food Ordering System

Tribhuvan University

Faculty of Humanities and Social Sciences

Bachelors in Computer Application

Submitted to

Department of Computer Application

Asian College Of Higher Studies

Ekantakuna, Lalitpur

Submitted by

Arin Shrestha

Shovit Bhatta

Contents

Introduction

SmartServe is a web-based platform designed to revolutionize the way food orders are managed in canteens. In the current digital age, traditional methods of ordering food—such as queuing, manual order taking, and cash transactions—are increasingly viewed as inefficient and outdated. These methods often lead to long waiting times, order inaccuracies, and operational bottlenecks, all of which can negatively impact the overall dining experience for customers and place additional stress on canteen staff.

SmartServe seeks to address these issues by providing a seamless, user-friendly interface that allows customers to place orders directly from their smartphones or other mobile devices. By scanning a QR code displayed within the canteen, customers can instantly access the menu, customize their orders, and submit them with just a few taps. This not only streamlines the ordering process but also minimizes the potential for errors, as orders are transmitted directly to the kitchen staff without the need for manual input.

Additionally, the digital nature of the system allows for easy tracking and management of orders, inventory, and sales data, providing canteen operators with valuable insights that can be used to optimize their service and menu offerings.

SmartServe is not just about improving the customer experience; it also offers significant benefits to canteen operators. By automating many of the manual processes traditionally associated with food ordering, the system reduces the workload on staff, allowing them to focus more on food preparation and customer service. Additionally, the system's data analytics capabilities enable operators to make informed decisions about inventory management, staffing, and menu planning, all of which contribute to a more efficient and profitable operation.

In summary, SmartServe is a comprehensive solution designed to modernize the food ordering process in canteens. By leveraging technology to streamline operations, reduce errors, and enhance customer satisfaction, this system represents a significant step forward in the management of canteen services.

Problem Statement

The traditional food ordering system in canteens present several challenges that hinder efficiency and customer satisfaction. One of the primary issues is the manual process of taking orders, which is prone to errors, such as miscommunication between customers and staff, leading to incorrect or delayed orders. This inefficiency not only frustrates customers but also increases the workload on canteen staff, particularly during peak hours.

Another significant problem is the lack of a centralized and easily updatable menu system. When new dishes are introduced or prices change, updating all physical menus across the canteen requires considerable time and expense. This often results in inconsistencies where customers may receive outdated information, leading to further confusion and dissatisfaction.

Furthermore, the traditional system does not provide real-time updates on order status, causing customers to wait without knowing the progress of their orders. This lack of transparency can lead to customer frustration, especially in a fast-paced environment like a canteen, where quick service is expected.

Objectives

- To develop a web-based platform that allows users to order food by scanning a QR code.
- To streamline the menu update process, allowing for easy modifications and additions without physical reprints.
- To reduce the time taken for customers to place orders.
- To provide track of customer orders.

Methodology

1. Requirement Identification

1.1 Literature Review

The digital food ordering system is a simple tool for users to provide a better dining experience. The implementation of food ordering systems has become an essential part of modern dining experiences, aiming to streamline the ordering process, reduce errors, and increase customer satisfaction. Various approaches to food ordering systems have been employed over the years, each with its own advantages and challenges. This literature study examines the drawbacks of traditional food ordering system, handheld device system, and Tablet Based ordering systems.

1.1.1 Traditional Food Ordering System

Traditional food ordering systems in canteens rely on manual processes, including taking orders by hand, communicating them to the kitchen, and handling cash transactions. This approach often results in long wait times, order inaccuracies, and inefficiencies in managing menu updates. Customers face delays and potential frustration, while staff members are burdened with repetitive tasks and high error rates.[1] The traditional ordering system brings inconvenience to both staffs and customer as it requires a lot of manual work. The manual work done by the staffs will cause some human errors such as the probability of paper lost is high and the kitchen"s can misinterpret the handwriting of order. Sometimes, when the staffs write in hurry

will make the handwriting difficult to understand. All these human errors will cause the customer dissatisfaction towards the cafeteria and gives the bad experience.[2]

1.1.2 Handheld Device Systems (e.g., Qorder)

Handheld devices, like those used in the Qorder system, allow waiters to input orders directly into a central system. While this method reduces some errors compared to paper-based methods, it still involves human intermediaries who can misinterpret or incorrectly input orders. Additionally, reliance on hardware for order management and the need for manual menu updates can lead to inconsistencies and delays.**[3]**

1.1.3 Tablet-Based Ordering Systems

Tablet-based systems place a tablet at each table, allowing customers to place orders directly without waiter intervention. While this method reduces errors associated with manual order-taking, it requires significant investment in hardware and can be cumbersome to manage. These systems also often suffer from issues with menu updates and integration with existing kitchen workflows.**[4]**

1.1.4 Online Food System With Bluetooth

The smart ordering system is proposed with the use of a handheld tool which is used to make an order at the restaurant. It is proposed to solve the problems which are faced by the restaurant's entrepreneur in the attempt to organize the restaurant more efficiently skilled and capable. The system uses a small keyboard which is placed on each table for the customer to make orders. Order is made by inserting the menu code on the small keyboard. This code comes together with the menu. A signal will be sent to the order section by Bluetooth communication, and automatically will be displayed on a screen in the kitchen. The project will reduce time to be spent on making the orders and paying the bills, whereby the cost and man power also can be optimized.**[5]**

SmartServe differentiates itself by providing a web-based platform accessible via QR codes, eliminating the need for manual order-taking, reducing hassel of using keywords to order and reducing the potential for human error. Unlike traditional systems and handheld devices, our

4

system allows for real-time menu updates and integrates seamlessly with digital payment gateways, enhancing both accuracy and efficiency. It also avoids the need for expensive hardware investments by leveraging users' personal devices, thus reducing overall costs and simplifying deployment.

1.2 Requirement Collection

For SmartServe, requirements are identified through a combination of functional and non-functional aspects. This approach ensures the system meets both operational needs and performance standards.

1.2.1 Functional Requirements

For Customers:

a) Order Placement: The system must allow customers to place orders via their personal devices after scanning a QR code at their table.

b) Menu Access: Customers should be able to view an updated menu with item details.

For Restaurant Staff:

a) Order Management: The system should allow kitchen staff to receive and view orders in real-time, with details on item quantities and special requests.

b) Update Status: Staff should be able to update the status of orders.

c) Update Availability: Staff should be able to update menu items availability.

d) Menu Updates: Staff should be able to update menu items and prices.

e) Data Monitoring: Manager should be able to access and analyze order data and feedback to monitor system performance and user satisfaction.

1.2.2 Non-Functional Requirements

Security:

a) Access Control: The system will incorporate role-based access controls to ensure that sensitive functions, such as menu updates and user management, are only accessible to authorized personnel.

b) Data Protection: User data, including personal information and payment details, will be encrypted to prevent unauthorized access and ensure privacy.

Availability:

a) Service Uptime: The system will be designed to be operational 24/7, providing uninterrupted service to customers regardless of the time or day.

b) Redundancy: To minimize downtime, the system will include redundancy measures such as backup servers and data recovery procedures.

Performance:

a) Response Time: The system will be optimized to provide fast response times for user interactions, including order placement and menu navigation.

b) Scalability: The system should handle varying volumes of orders efficiently, accommodating both peak and off-peak times without performance degradation.

Reliability:

a) Error Handling: The system will include robust error handling mechanisms to manage both expected and unexpected issues, ensuring a smooth user experience.

b) Data Integrity: Measures will be in place to ensure the accuracy and completeness of data, preventing errors and ensuring reliable order processing.

2. Feasibility Study

2.1 Technical Feasibility

a) Programming Language & Tools: The system will utilize standard web technologies such as HTML, CSS, JavaScript and a backend i.e. Node.js. These are widely supported and have extensive libraries that are capable of delivering the required functionalities. Node.js is chosen for its ability to handle multiple requests concurrently with non-blocking I/O, making it highly suitable for real-time applications like Canteen ordering systems, where multiple users (customers, waitstaff, and kitchen staff) will be interacting with the system simultaneously. Additionally, Node.js allows for the use of a single programming language, JavaScript, on both the frontend and backend, streamlining development and improving maintainability.

b) Infrastructure: The current infrastructure of the canteen, including internet connectivity and user devices (smartphones or tablets), is sufficient to support the system. No additional hardware is required beyond what is already in place.

2.2 Operational Feasibility

a) User-Friendly Interface: The system is designed with a user-friendly interface, making it easy for customers to place orders and for staff to manage them. The use of QR codes simplifies the ordering process, reducing the need for training.

b) Integration with Existing Operations: The system integrates smoothly with existing canteen operations, enhancing order accuracy and reducing the dependency on

waitstaff. This will streamline the workflow and minimize errors associated with manual order-taking.

c) Ease of Implementation: The system can be implemented without major disruptions to the current operations. Staff training requirements are minimal due to the system's intuitive design.

2.3 Economic Feasibility

a) Minimal Additional Costs: The system is designed to run on existing hardware and infrastructure, eliminating the need for additional investments in new equipment.

b) Long-Term Savings: By reducing the reliance on waitstaff and improving order accuracy, the system can lead to long-term cost savings for the canteen. Operational efficiencies gained through the system will also contribute to a higher return on investment.
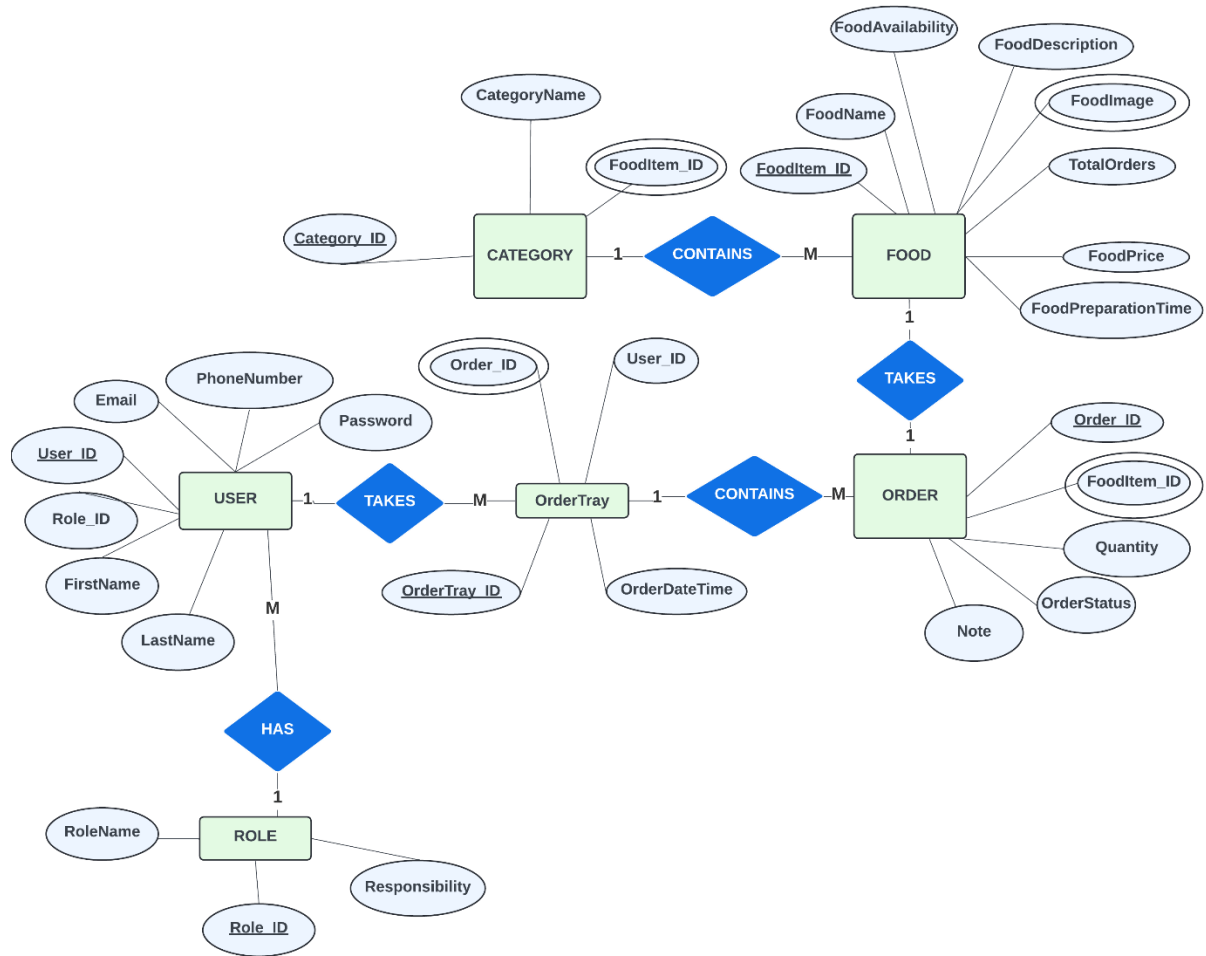
3. High Level Design of System

3.1 ER Diagram

Fig 1.1: Entity-Relationship Diagram
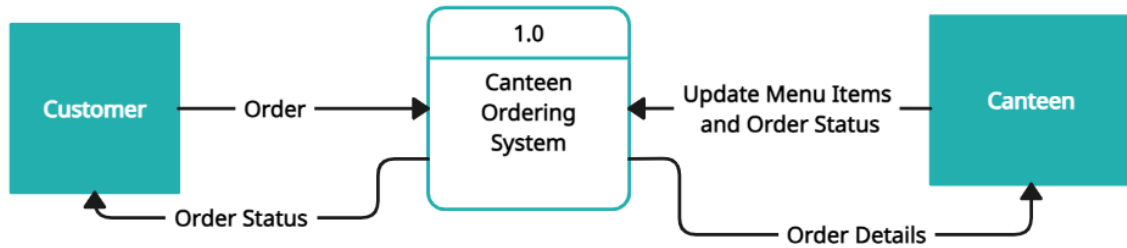
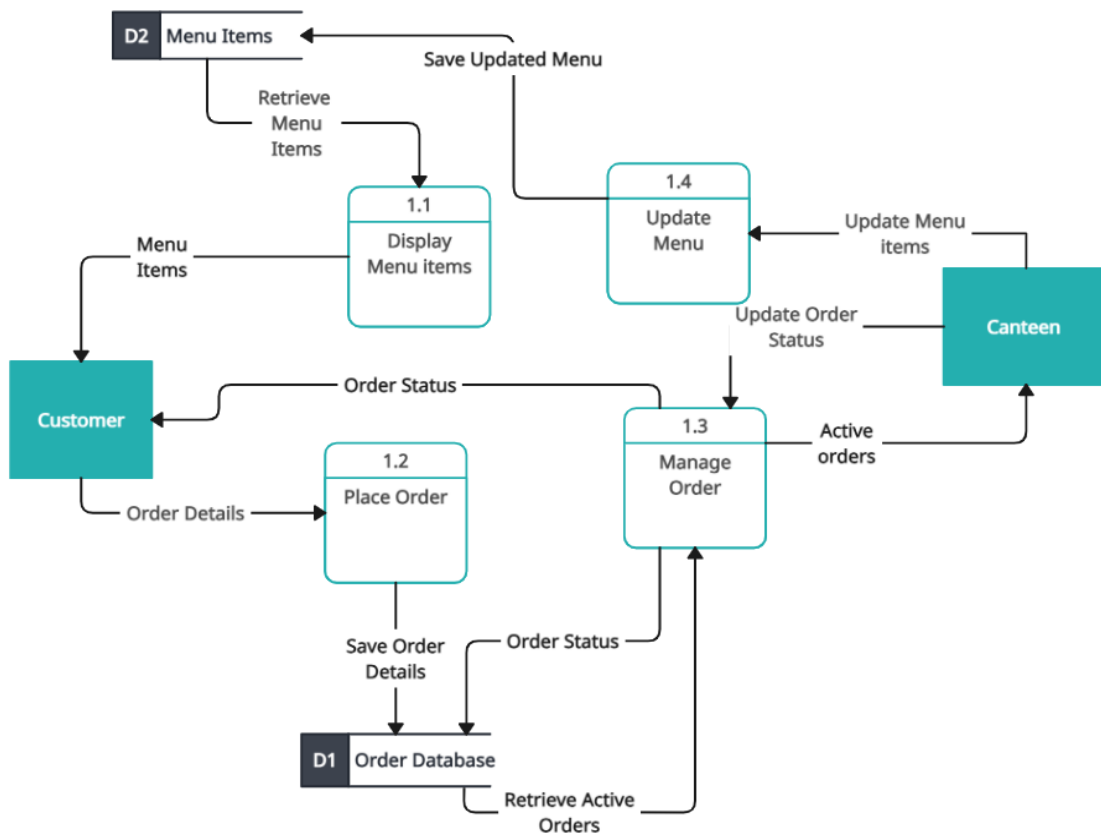## 3.2 Data Flow Diagram (DFD)

Fig 2.1: Level 0 DFD

Fig 2.2: Level 1 DFD

3.3 Methodology of proposed system

The methodology outlines the approach that will be taken in designing and developing SmartServe.

Waterfall Model: A waterfall development approach will be adopted because the project requirements are clear, well-understood, and unlikely to change during the development process.

Modular Design: The system will be divided into modules like user management, order processing, payment processing and menu management. This modular approach will make the system easier to develop, maintain, and test.

Database Design: A SQL database schema will be designed based on the ERD, ensuring that data storage is efficient and supports all necessary relationships.

Security Considerations: Security protocols will be integrated, such as secure login mechanisms and access controls to protect sensitive data.

3.4 Working Mechanism of the Proposed System

3.4.1 Customer Interaction:

1. The customer will access the system via a web interface, either by scanning a QR code at the canteen or through a direct link.
2. The customer will register or log in, browse the menu, select items, and place an order.

3.4.2 Order Processing:

1. Once the order is placed, it will be sent to the kitchen interface. The kitchen staff will receive the order details in real-time, prepare the order, and update the order status.
2. The customer will be notified of the order status (e.g., preparation started, ready for pickup).

3.4.3 Canteen Staff Interfaces:

1. The canteen staff will use a dedicated interface to manage and update order statuses.

2. Manager will be able to oversee the entire process, manage menu items, view order histories, and generate reports.
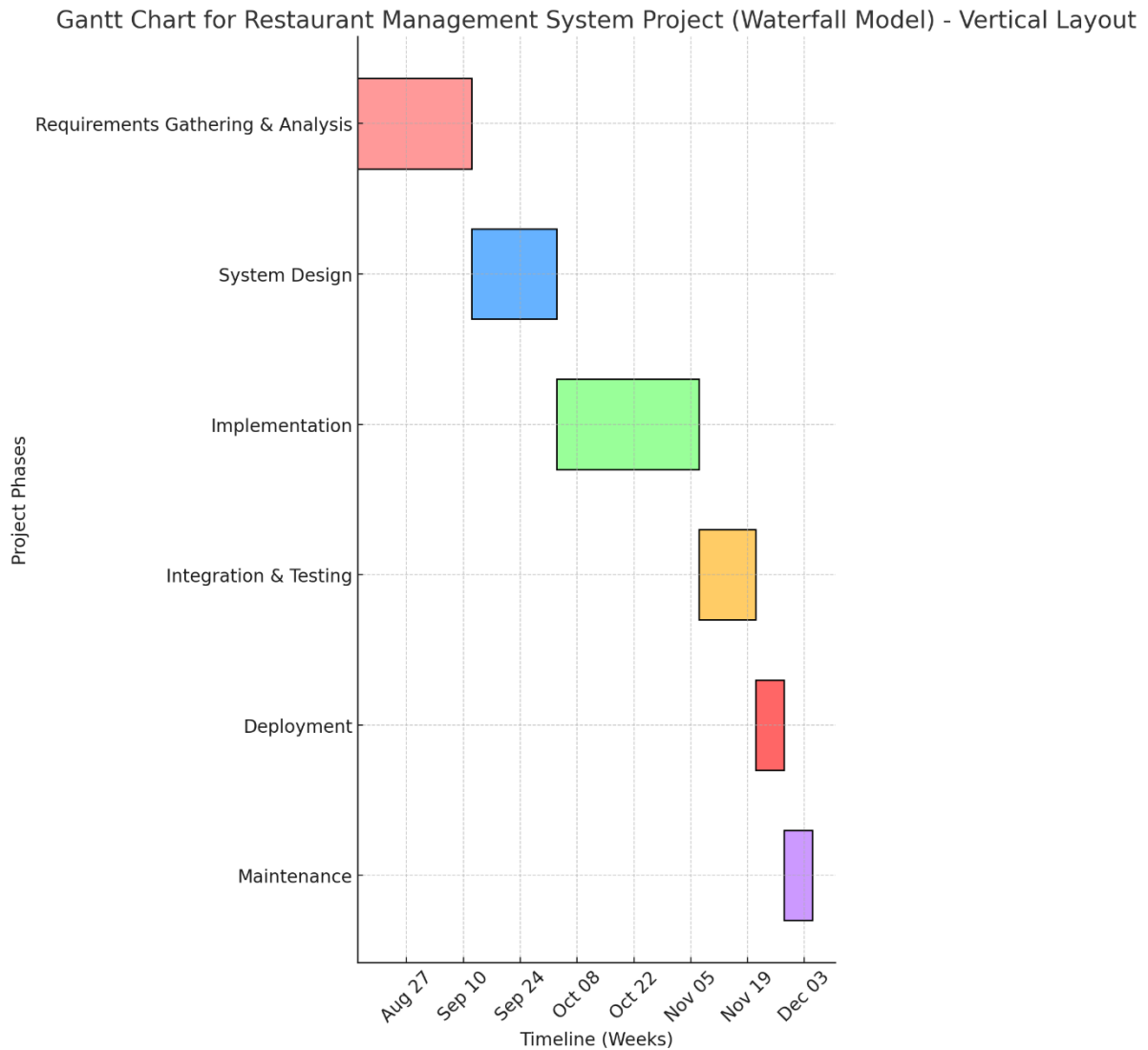
Gantt Chart

Fig 3: Gantt Chart

As shown in the Gantt chart the process of the development of the project will be as follows:

1. Requirement Gathering and Planning

Tasks:

- Document all requirements, including ordering, and admin functionalities.

- Create a Requirements Specification Document (RSD).
- Perform a feasibility Study for the project.

2. System Design

Tasks:

- Create the high-level architecture of the system (e.g., Client-Server model).
- Design data flow diagrams (DFDs) and entity-relationship diagrams (ERDs).
- Design the database schema, user interfaces, and define the APIs.

3. Implementation

Tasks:

- Set up the development environment for frontend and backend.
- Develop the frontend interface using HTML, CSS, and JavaScript.
- Build the backend using Node.js, implementing business logic and connecting it to the database.
- Integrate the frontend with the backend through APIs.

4. Integrating and Testing

Tasks:

- Conduct unit testing on individual modules.
- Perform integration testing to ensure modules work together smoothly.
- Conduct system testing to validate the system as a whole.
- Fix any bugs or issues discovered during testing.

5. Deployment

Tasks:

- Deploy the system
- Conduct user training sessions, providing documentation and support materials.

Expected Outcome

Increased Efficiency: Orders will be processed faster as customers place them directly through their devices, reducing the time spent on manual order-taking.

Reduced Errors: The system will minimize order inaccuracies by eliminating the need for waitstaff to manually relay orders, leading to greater customer satisfaction and less food wastage.

Enhanced Customer Experience: A streamlined ordering process with digital menus will provide a modern and convenient dining experience.

References:

[1] POSist, "From Pen-Paper to Point of Sale Software - The Evolution of Restaurant Tech," Medium, Nov. 2016. [Online]. Available: https://medium.com/@POSist/from-pen-paper-to-point-of-sale-software-the-evolution-of-restaurant-tech-c777ac56f21c. [Accessed: Aug. 22, 2024].

[2] International Journal of Scientific Research in Science, Engineering and Technology, IJSRSET, "Cafeteria Food Ordering System," IJSRSET Journals, vol. 10, issue 2, pp 127-163, March 27 2023. [Online]. Available: https://ijsrset.com/paper/8824.pdf. [Accessed: Aug 20, 2022].

[3] Toast, "The True Value of Handheld POS Systems in Restaurants," Toast Blog, [Online]. Available: https://pos.toasttab.com/blog/on-the-line/restaurant-handheld-pos. [Accessed: Aug. 20, 2024].

[4] IRE Journals, "Digital Food Ordering System for Restaurants," IRE Journals, vol. 5, no. 9, pp. 56-61, 2022. [Online]. Available: https://www.irejournals.com/formatedpaper/17022151.pdf. [Accessed: Aug. 20, 2024].

[5] International Journal of Computer Trends and Technology (IJCTT), "Smart Ordering System via Bluetooth", IJCTT Journal, vol. 4, no. 7, pp. 2253, 2013, [Online]. Available: https://ijcttjournal.org/Volume4/issue-7/IJCTT-V4I7P151.pdf. [Accessed: Aug, 20, 2024]