VIETNAM NATIONAL UNIVERSITY HANOI

INTERNATIONAL SCHOOL

FINAL PROJECT

COURSE: MICROPROCESSORS AND MICROCONTROLLERS

DESIGN AND DEVELOPMENT OF A SMART CONVENIENT STORE MODEL USING RASPBERRY PI 4

Lecturer: Dr. Nguyen Dang Khoa

Major: Automation and Informatics

Group Number: 1

Student 1: Pham Van Phuong - 21070403

Student 2: Nguyen Khac Long - 21070906

Student 3: Vu Duc Hung - 21070084

HANOI - January, 2025

VIETNAM NATIONAL UNIVERSITY HANOI
INTERNATIONAL SCHOOL

FINAL PROJECT
COURSE: MICROPROCESSORS AND MICROCONTROLLERS

DESIGN AND DEVELOPMENT OF A SMART CONVENIENT STORE MODEL USING RASPBERRY PI 4

Major: Automation and Informatics

HANOI - January 2025

ACKNOWLEDGEMENTS

We would like to express our deepest gratitude to Dr. Nguyen Dang Khoa for his invaluable guidance and support throughout the development of our project titled "Design and Development of a Smart Convenience Store Model using Raspberry Pi 4." His expertise and encouragement have been instrumental in shaping our work.

We would also like to thank our group members for their hard work, collaboration, and dedication to the project. Together, we were able to overcome challenges and successfully implement both the hardware and software components required for the smart convenience store model.

Hanoi, January 01, 2025

Group Leader

Pham Van Phuong

CONTRIBUTION

Name	Role	Contribution (%)	Tasks
Pham Van Phuong	Leader	40%	Project overview, main programming in both frontend and backend, and assisting with report writing in Word.
Nguyen Khac Long	Member	40%	Hardware and database, report in LaTeX, slides, video editing
Vu Duc Hung	Member	20%	Assistance with hardware, slides, and report writing in Word.

Table 1. Contribution of Members

ABSTRACT

Abstract: This project focuses on the design and development of a smart convenience store model using Raspberry Pi 4 as the core microcomputer, integrated with RFID technology for access control and payment processing. The system features a database to manage transaction data, user information, and product inventory, facilitating efficient store operations. The dataflow between the RFID reader, Raspberry Pi 4, and the database is optimized to ensure seamless transactions and real-time updates. Additionally, a web server for management is implemented to allow remote monitoring and control of the system, enabling store administrators to oversee operations and analyze data. Although this is a proof of concept and not intended for large-scale deployment, the project demonstrates the feasibility of utilizing microcomputers and IoT solutions in retail environments to improve security, enhance customer experience, and support business management processes.

Keywords: *smart convenience store, Raspberry Pi 4, RFID, database, web server*

TABLE OF CONTENTS

ABSTRACT

Abstract: This project focuses on the design and development of a smart convenience store model using Raspberry Pi 4 as the core microcomputer, integrated with RFID technology for access control and payment processing. The system features a database to manage transaction data, user information, and product inventory, facilitating efficient store operations. The dataflow between the RFID reader, Raspberry Pi 4, and the database is optimized to ensure seamless transactions and real-time updates. Additionally, a web server for management is implemented to allow remote monitoring and control of the system, enabling store administrators to oversee operations and analyze data. Although this is a proof of concept and not intended for large-scale deployment, the project demonstrates the feasibility of utilizing microcomputers and IoT solutions in retail environments to improve security, enhance customer experience, and support business management processes.

Keywords: *smart convenience store, Raspberry Pi 4, RFID, database, web server*

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

Background

The rapid advancement of technology has become a defining feature of the global economy, driving significant changes across industries. In Vietnam, this trend is reflected in the country's active embrace of digital transformation. Over recent years, Vietnam has witnessed a steady shift from traditional business models to digital and e-commerce platforms. This transition has been fueled by increased internet penetration, widespread use of smartphones, and government policies encouraging digital transformation. The shift has not only reshaped consumer behavior but has also pushed businesses to adopt innovative solutions to remain competitive.

One of the key enablers of this transformation is the development of microcomputers, which offer compact, cost-effective, and powerful solutions for optimizing various processes. Microcomputers, when applied in settings such as supermarkets and restaurants, have proven to streamline operations like inventory management, customer service, and payment processing. These technological advancements enable businesses to enhance efficiency, reduce costs, and improve customer experiences.

Among the many microcomputers available, the Raspberry Pi 4 stands out as a versatile and accessible option. Known for its affordability and ease of use, the Raspberry Pi 4 offers robust computing capabilities suitable for a wide range of applications. Equipped with a quad-core processor, multiple connectivity options, and support for various operating systems, it provides a reliable platform for developing systems such as access control, automated billing, and data storage. Its adaptability makes it an ideal choice for small businesses looking to adopt digital solutions without incurring significant expenses. This project aims to leverage the capabilities of the Raspberry Pi 4 to design and develop a smart convenience store model. By integrating hardware and software components, the system seeks to demonstrate the potential of technology in modernizing traditional retail operations, particularly in the context of Vietnam's digital transformation.

Problem statement

In Vietnam, the retail industry is undergoing a significant transformation, with businesses shifting from traditional operations to digital and automated systems. However, many small and medium-sized retail businesses, including convenience stores, face challenges in adopting these advanced technologies due to high implementation costs, technical complexity, and limited access to reliable solutions.

Traditional convenience stores often rely on manual processes for access control, billing, and inventory management, leading to inefficiencies, human errors, and increased operational costs. Additionally, the lack of robust systems for tracking transactions and customer data limits opportunities for implementing loyalty programs and improving accounting accuracy.

Despite the availability of low-cost computing solutions like the Raspberry Pi 4, there is a lack of practical, scalable, and affordable models tailored for small-scale retail businesses in Vietnam. This gap presents an opportunity to develop a cost-effective and easily deployable smart convenience store model that can streamline operations, enhance security, and improve customer experiences.

This project addresses these challenges by designing and developing a prototype system using the Raspberry Pi 4 and low-cost peripherals. The goal is to demonstrate a proof-of-concept solution that integrates access control, payment processing, and transaction recording into a unified, user-friendly platform suitable for Vietnam's growing digital economy.

Objectives

The primary objective of this project is to design and develop a prototype smart convenience store model using the Raspberry Pi 4 that integrates essential retail operations into a cost-effective and efficient system. The specific objectives are as follows:

- To develop an integrated store model that has a web server to manage customers and transactions data with a centralized database.
- To optimize data handling by establishing a streamlined data flow to monitor store operations in real-time and enable data-driven decision-making.
- To develop a basic prototype that build a proof-of-concept model to demonstrate the feasibility of a smart retail system for small and medium-sized stores.

By achieving these objectives, the project aims to present a practical, and innovative solution for the challenges faced by traditional convenience stores in Vietnam's digital economy.

Scopes

This project focuses on the design and development of a smart convenience store model using Raspberry Pi 4, integrating software and hardware systems to optimize store operations. The scopes include:

- System integration to develop a prototype that combines hardware components (RFID reader, LCD, servo barrier, Raspberry Pi 4) with software for customers' data and transaction management.
- Access control to implement an RFID-based system to automate and secure customer entry and exit.
- Automated payment system that design a payment processing section capable of recording transactions and storing data in a centralized database.
- Data management that creates a real-time database to handle customers and products' information, customer transactions, and operational logs for streamlined management and reporting.
- Develop an web-based interface for registering customers' orders, managing store data.
- Prototype development that need to build and test a functional prototype to demonstrate the model's capabilities in real-world scenarios, focusing on scalability for larger retail environments.

Structure of report

This report is organized into six chapters, each detailing a specific aspect of the design and development of the Smart Convenience Store Model, as outlined below:

- **Introduction**: This section provides the background, problem statement, objectives, and an overview of the report's structure. It sets the context for the project and outlines the goals and approach taken to achieve them.

- **Chapter 1** provides a system overview, explaining the conceptual design and main components, including both hardware and software, and how they work together to form an integrated solution.

- **Chapter 2** delves into hardware design, discussing the selection of components such as Raspberry Pi 4, RFID readers, LCD, servo barriers, along with their roles in the system.

- **Chapter 3** covers the software design, detailing the development of the management software, its interaction with the hardware, and the database system for data storage and retrieval.

- **Chapter 4** outlines the system workflow, illustrating the interaction between hardware and software components and describing how the system operates in a typical use case scenario.

- **Chapter 5** presents the testing methodology, results, and evaluation of the system's performance, followed by a discussion of the findings and concluding remarks.

- **Conclusion and Discussion** summarizes the findings of the project, give discussions for limitations, improvements, and suggests potential future applications of the developed system.

CHAPTER 1
SYSTEM OVERVIEW

1.1. Overview of system architecture

The proposed system utilizes Radio Frequency Identification (RFID) technology to streamline customer identification and automate the payment process in a retail setting, such as a convenience store. The system integrates RFID scanning with a central database and a website, providing a seamless and efficient shopping experience. The architecture incorporates key components including RFID scanners, a microcontroller or Raspberry Pi 4, a database server, and an integrated web application. These components work together to identify customers, process payments, and control entry/exit barriers for enhanced convenience and security.

1.2. Key features of the convenient store model

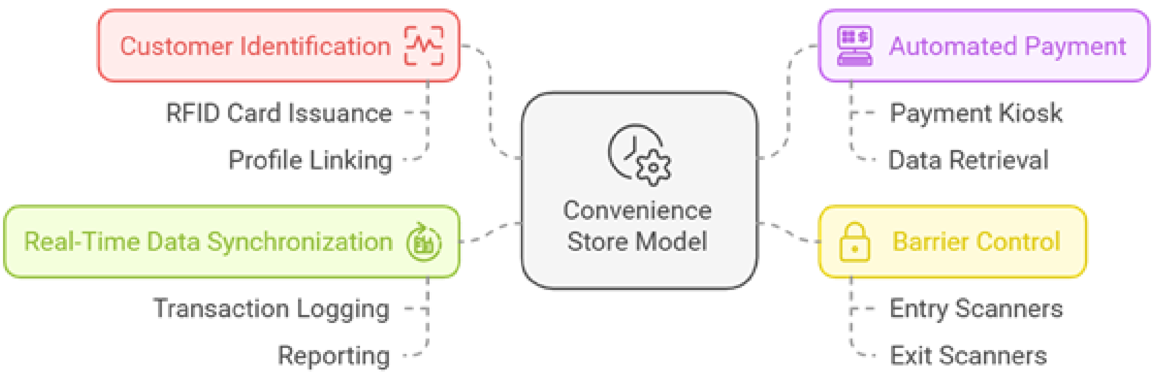

Figure 1.1. Key features of the convenient store model

- **Customer Identification:** Each customer is issued an RFID card that contains unique identification data linked to their profile in the central database. Upon entering the store, the system scans the RFID card to authenticate the customer.
- **Automated Payment:** After shopping, customers proceed to the payment kiosk, where their RFID card is scanned. The system retrieves the shopping cart data,

calculates the total amount, and processes the payment automatically using the stored payment details.

1.3. System block diagram

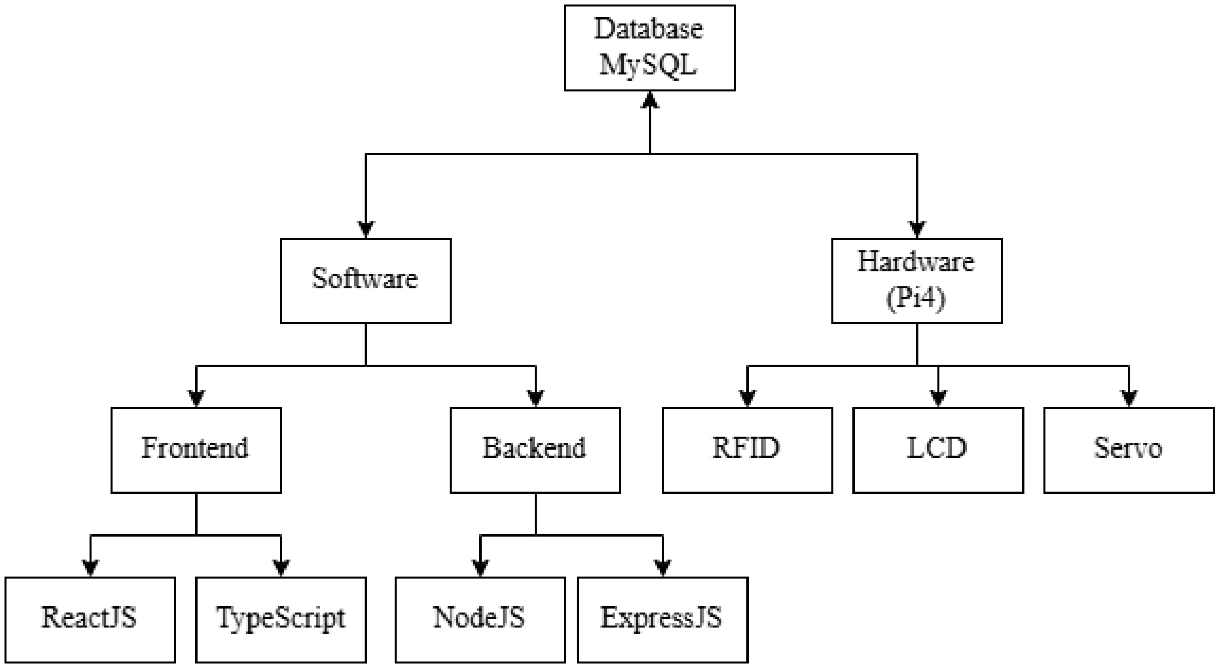

Figure 1.2. System block diagram

The system block diagram for the ***"Design and Development of a Smart Convenience Store Model using Raspberry Pi 4"*** is divided into three main components: Database, Software, and Hardware. These components work together to provide a seamless experience for managing and automating store operations.

1.3.1. Database

The database acts as the central storage for all system data, including customer information, product inventory, and transaction logs. It is hosted on a server and accessed by both the frontend and backend software.

- **Purpose:** Stores product details, customer profiles, and transaction records for updates and secure data management.
- **Type:** A relational or NoSQL database such as MySQL and MongoDB is used to ensure scalability and performance.

1.3.2. Software

The software is divided into Frontend and Backend, ensuring a clear separation of user interaction and server-side logic.

Frontend

- Built using ReactJS and TypeScript, the frontend provides a user-friendly interface for customers and store operators.
- Features include browsing product information, viewing prices, and booking or purchasing items.
- Runs on web browsers or a local display connected to the Raspberry Pi.

Backend

- Developed using NodeJS and ExpressJS, the backend handles API requests, processes data, and communicates with the database.
- It serves as the middleware, managing tasks such as user authentication, product search, and purchase verification.
- The backend also interacts with the hardware components (via Raspberry Pi 4) to control devices like the servo barrier and RFID scanner.

1.3.3. Hardware

The hardware includes devices connected to the Raspberry Pi 4 to enable physical interactions and automation.

Raspberry Pi 4:

- The central control unit, running the software and coordinating data flow between the frontend, backend, and hardware devices.
- Acts as the interface between the software and external peripherals.

RFID Reader:

- Used to identify customers through RFID cards.
- Used for payments.

Servor Barrier:

- Acts as an automated gate, controlled by the Raspberry Pi.
- Opens or closes based on the transaction status, ensuring secure access to the store.

1.4. Functional modules

1.4.1. RFID scanning and data tracking

The RFID scanner reads the unique ID from the customer's card. The system cross-references this ID with the central database to fetch the customer's profile, shopping history, and payment preferences. During checkout, the RFID reader at the payment kiosk verifies the customer's identity and retrieves the shopping cart data.

1.4.2. Database and website operations

- *Database functions*: Stores customer profiles, including RFID ID, payment information, and shopping history. Logs every transaction, including time, items purchased, and payment details.
- *Website Functions:*: Allows customers to log in and view transaction history, remaining balances, and promotions. Enables administrators to manage inventory, monitor system status, and analyze sales trends.

1.4.3. Payment and barrier control

- *Payment processing:* The system automatically calculates the total bill based on the items scanned at checkout. It processes the payment using stored payment details linked to the customer's profile, ensuring a cashless and hassle-free transaction.
- *Barrier control:* The entry barrier opens upon successful RFID authentication. The exit barrier operates only after the payment is confirmed, ensuring that all purchases are accounted for.

CHAPTER 2
HARDWARE DESIGN

2.1. Hardware components

The hardware components in this project play crucial roles in creating a functional and efficient smart convenience store model. The Raspberry Pi 4 serves as the central processing unit, managing hardware interactions and executing control logic. The RFID card system enables secure and seamless identification of users or products, essential for payment and authentication processes. A 16x2 or 20x4 LCD display provides a user-friendly interface, showing prompts, product details, or transaction statuses. The servo barrier regulates physical access, opening or closing based on system commands, ensuring a controlled environment. Finally, a reliable power supply ensures all components operate smoothly by providing consistent voltage and current, guaranteeing system stability and functionality.

Table 2.1. Hardware Component List

Component	**Specification**	**Description**
Raspberry Pi 4	2GB RAM, Quad-core Cortex-A72, GPIO pins, HDMI, USB, Ethernet	Serves as the central processing unit, managing all connected hardware and executing control logic.
RFID Card	Frequency: 13.56 MHz, Standard: ISO/IEC 14443, Power: Passive	Used to scan and identify user or product information for authentication and payment systems.
LCD Display	Type: 16x2 or 20x4 characters, Controller: HD44780, Interface: I2C/SPI	Displays system information, such as user prompts, product details, or system status.
Servo Barrier	Torque: 2.5 kg·cm, Rotation: 180°, Power: 5V DC	Controls physical access by opening or closing based on user or system commands.
Power Supply	Input: 220V AC, Output: 5V/12V DC, Power: 50W-100W	Provides stable power to the Raspberry Pi, servo barrier, and other electronic components.

2.1.1. Raspberry Pi 4

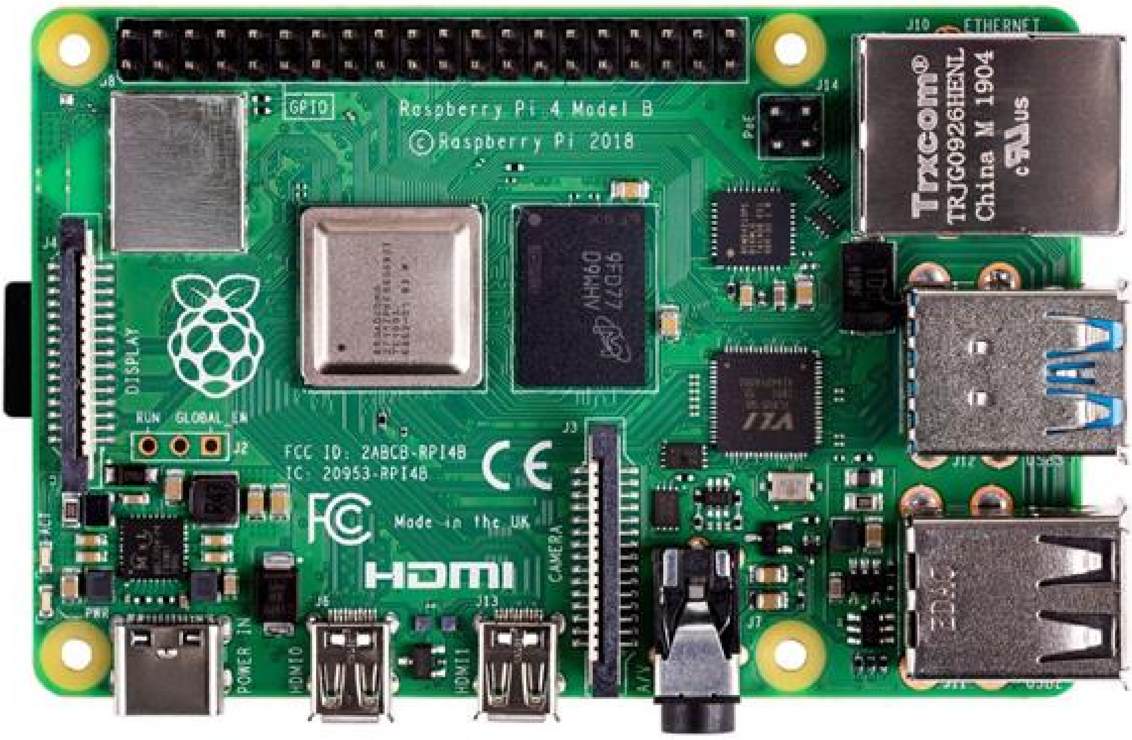

Figure 2.1. Raspberry Pi 4

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{IL}	Input low voltage[a]	VDD_IO = 3.3V	0	-	0.8	V
V_{IH}	Input high voltage[a]	VDD_IO = 3.3V	2.0	-	VDD_IO	V
I_{IL}	Input leakage current	TA = +85°C	-	-	10	μA
C_{IN}	Input capacitance	-	-	5	-	pF
V_{OL}	Output low voltage[b]	VDD_IO = 3.3V, IOL = -2mA	-	-	0.4	V
V_{OH}	Output high voltage[b]	VDD_IO = 3.3V, IOH = 2mA	VDD_IO - 0.4	-	-	V
I_{OL}	Output low current[c]	VDD_IO = 3.3V, VO = 0.4V	7	-	-	mA
I_{OH}	Output high current[c]	VDD_IO = 3.3V, VO = 2.3V	7	-	-	mA
R_{PU}	Pullup resistor	-	18	47	73	kΩ
R_{PD}	Pulldown resistor	-	18	47	73	kΩ

[a] Hysteresis enabled
[b] Default drive strength (8mA)
[c] Maximum drive strength (16mA)

Figure 2.2. DC Characteristics [1]

Pin Name	Symbol	Parameter	Minimum	Typical	Maximum	Unit
Digital outputs	t_{rise}	10-90% rise time[a]	-	TBD	-	ns
Digital outputs	t_{fall}	90-10% fall time[a]	-	TBD	-	ns

[a] Default drive strength, CL = 5pF, VDD_IO = 3.3V

Figure 2.3. Digital I/O Pin AC Characteristics[1]

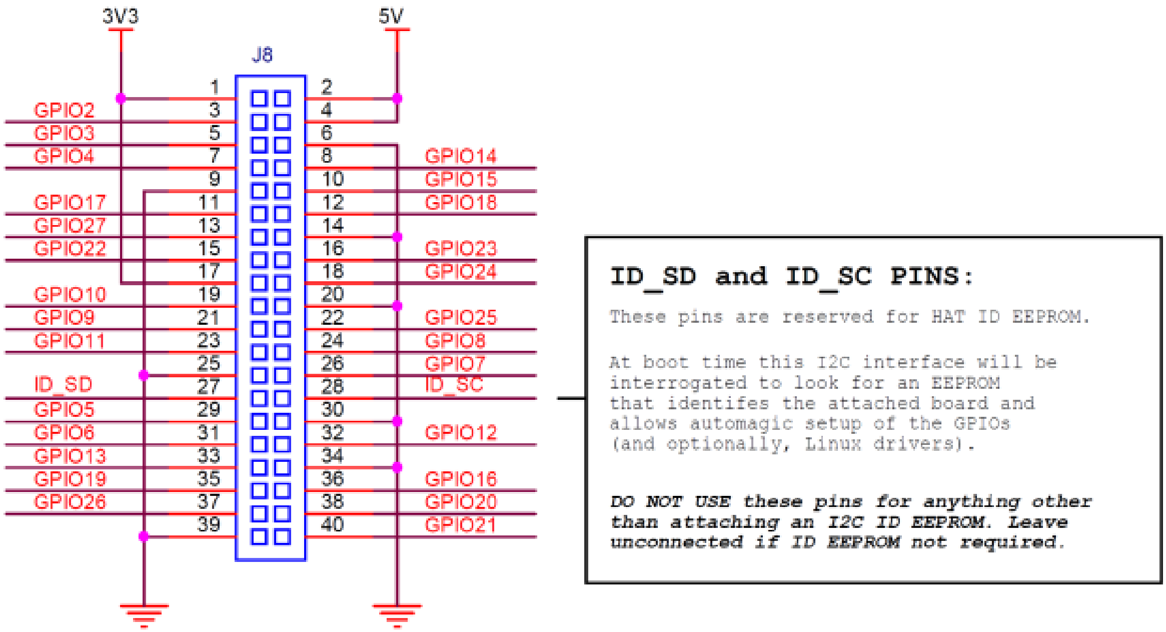

Figure 2.4. GPIO Connector Pinout [1]

The Raspberry Pi 4 is the central processing unit of the system. It handles the overall operation, including RFID tag reading, communication with the backend server, and display management. Here's how the Raspberry Pi 4 functions in this setup:

Role: The Raspberry Pi acts as the primary processing unit, receiving data from the RFID reader, querying the backend database for authentication, and displaying information on the LCD.

Processing Power: With its quad-core ARM Cortex-A72 CPU, the Raspberry Pi 4 provides sufficient processing power to handle tasks such as managing the RFID reader, interfacing with the server, and running the local software components.

Connectivity: The Raspberry Pi 4 supports both wired Ethernet and Wi-Fi, ensuring stable communication between the hardware and the backend system, as well as internet connectivity for additional functionalities (such as payment gateway integration).

GPIO Pins: The General Purpose Input/Output (GPIO) pins on the Raspberry Pi are used to interface with various sensors and actuators, such as the RFID reader, LCD, and Servo Motor.

2.1.2. RFID card

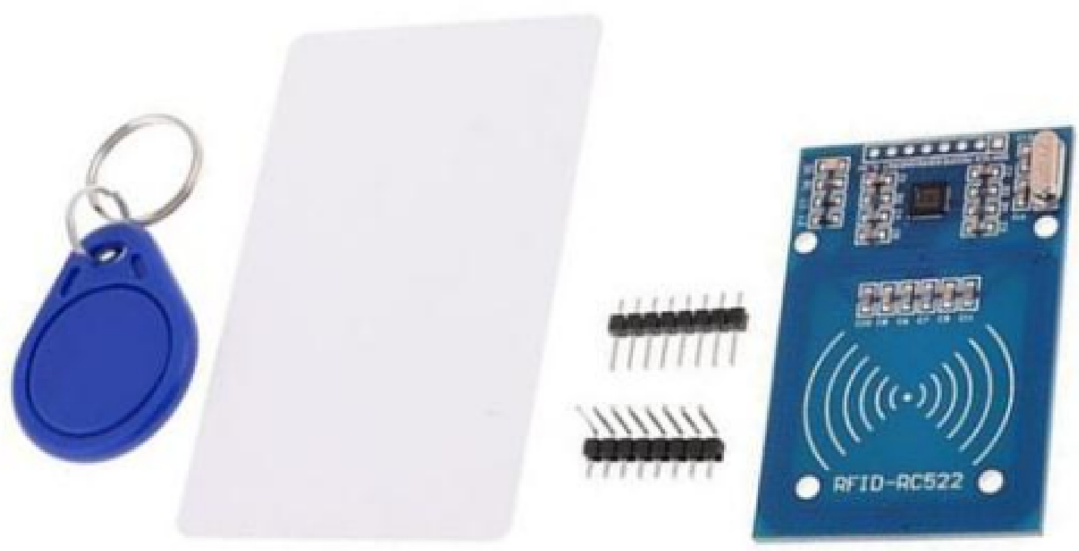

Figure 2.5. RC522 RFID Development Kit [2]

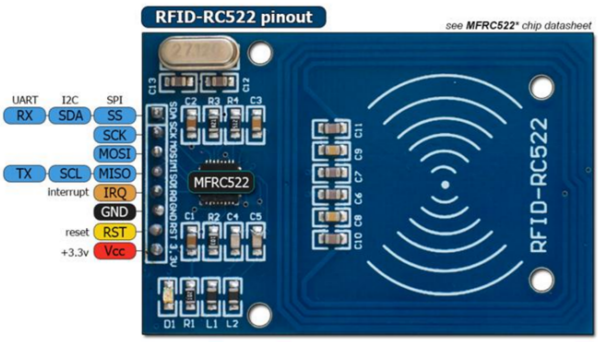

Figure 2.6. Interface Pins Function [2]

The RFID (Radio Frequency Identification) Card is used for secure authentication. The user's unique ID is embedded within the RFID card's chip, and it is read by the RFID reader.

Role: The RFID card stores a unique identifier (UID) that is used for user authentication when scanned. RFID Types: Common RFID types include HF (High Frequency) and LF (Low Frequency) cards. For this project, 13.56 MHz RFID cards (e.g., Mifare) are typically used for their compatibility with common RFID readers.

Communication: The RFID card communicates with the RFID reader using radio waves, transmitting its UID for further processing and validation by the backend system.

Security: The RFID tag is unique for each user, ensuring personalized and secure authentication. The system may also implement encryption or secure protocols to safeguard data.

2.1.3. LCD Display

Figure 2.7. I2C Serial Interface 1602 LCD Module [3]

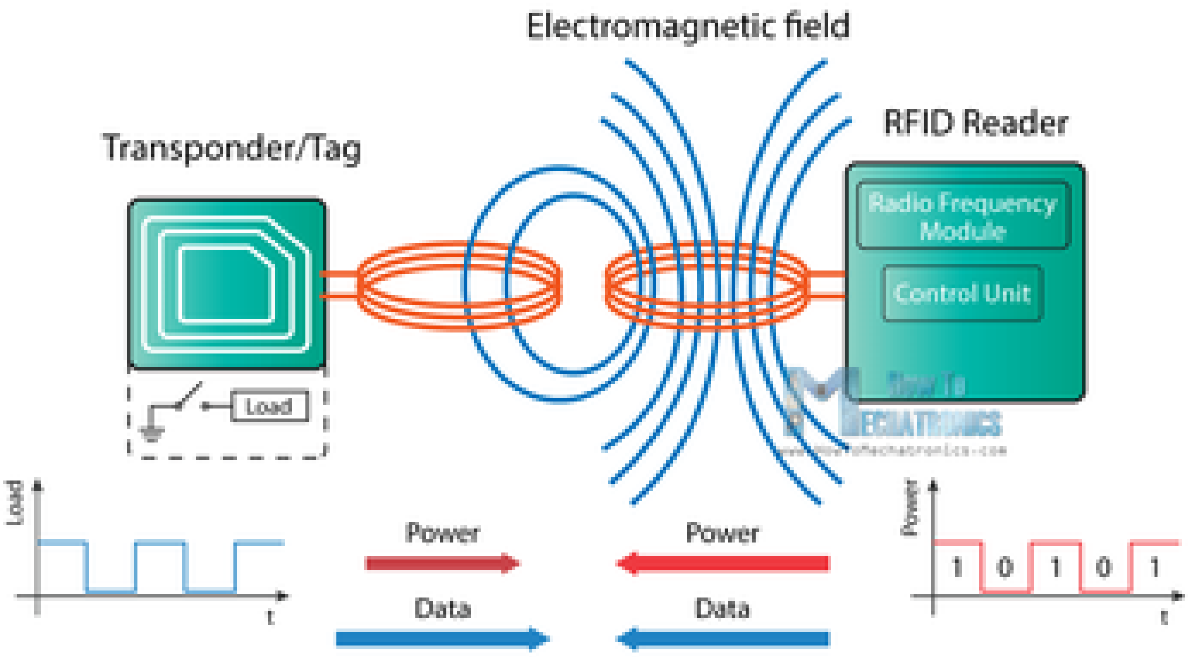

Figure 2.8. Working principles of RFID Tech

The LCD Display provides a visual interface for the user. It shows relevant information such as the system's status, product catalog, or personalized user data after authentication.

Type: A 16x2 LCD is commonly used for simple displays, offering enough space to display text messages such as the user's name, order status, or product information. More advanced displays like TFT screens can be used for a richer graphical interface.

Role: The LCD display is used toshow real-time status updates and instructions, helping the user navigate the system. It can display messages such as "Scan your RFID card" or "Welcome, [Customser's Name]!" Connection: The LCD connects to the Raspberry Pi via the I2C interface, making it easy to integrate into the system.

2.1.4. Servor Barrier

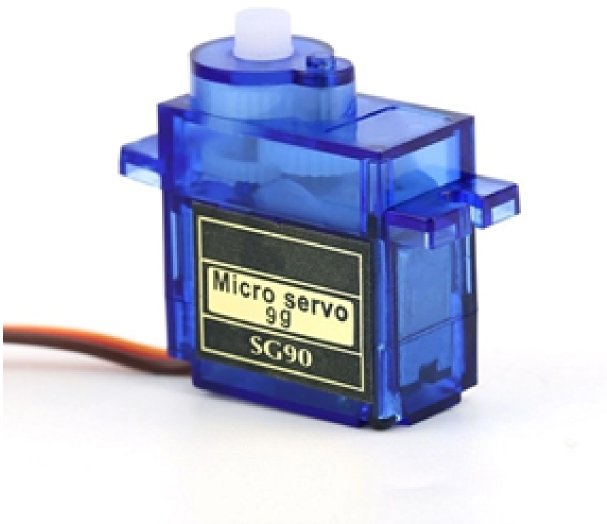

Figure 2.9. Servor Motor

The Servo Barrier is an essential hardware component used for access control. It provides a physical mechanism to allow or block entry based on RFID authentication.

Role: The servo motor controls the movement of a barrier, Upon successful authentication via RFID scanning, the Raspberry Pi sends a signal to activate the servo motor, allowing the barrier to open. If the user is not authenticated, the barrier remains closed.

Functionality: The servo motor moves the barrier to allow entry or prevent access. It operates in response to signals from the Raspberry Pi, which is triggered by successful RFID authentication.

Connection: The servo motor is connected to the Raspberry Pi through the GPIO pins. The system sends signals to control the servo's angle and position based on authentication status.

2.2. Circuit Diagram and Connection

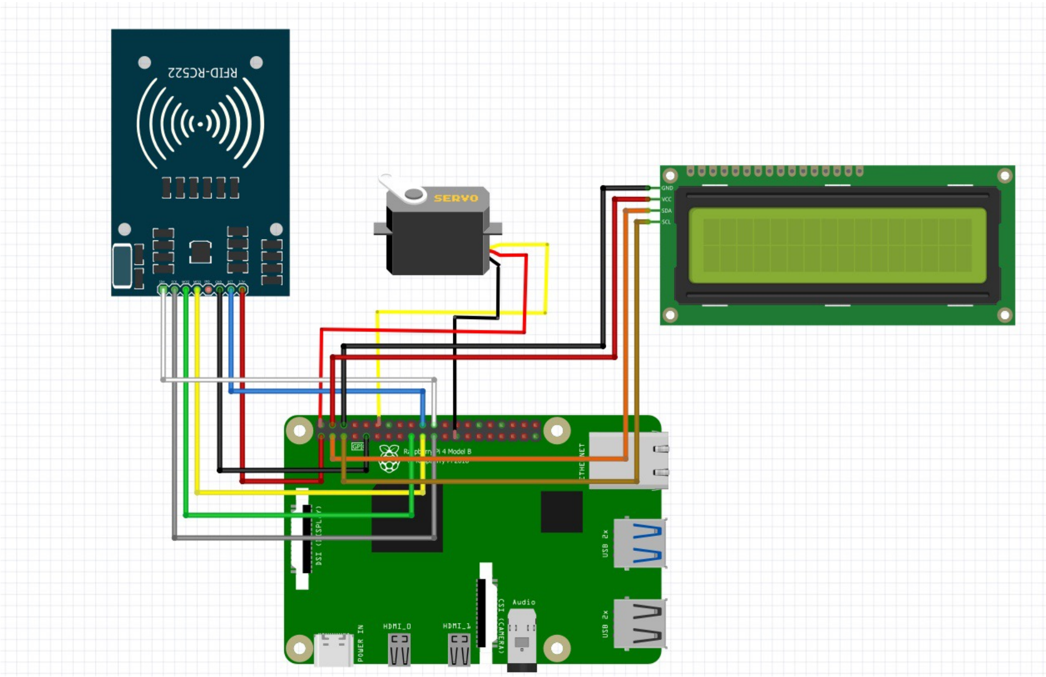

Figure 2.10. Circuit diagram of the system

CHAPTER 3
SOFTWARE DESIGN

3.1. Software architecture and Workflow

The architecture of the e-commerce website using Raspberry Pi integrates several components to ensure smooth functionality for both front-end and back-end processes.

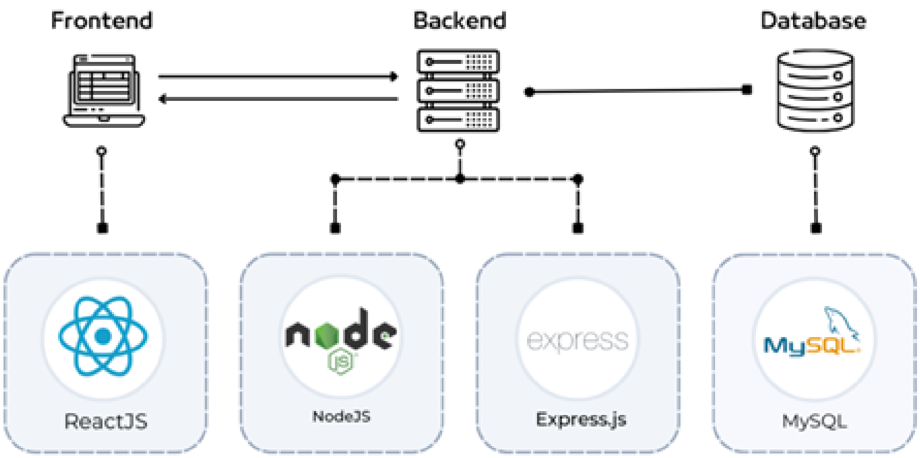

Figure 3.1. ReactJS, NodeJS, ExpressJs and MySQL working principle

Frontend: The front-end is built using ReactJS and TypeScript, ensuring type safety and easier maintainability of the codebase. The website is structured into components that handle the user interface for browsing products, adding items to the cart, managing user authentication, and completing purchases. The user experience is interactive, with a dynamic product catalog and a seamless checkout process.

Backend: The Node.js server with ExpressJS acts as the back-end framework for handling requests, processing user inputs, and interacting with the database. The back-end is responsible for managing sessions, processing payments, and providing product data from the MySQL database. ExpressJS routes handle product queries, order processing, and user authentication.

Database: The database is implemented using MySQL and is hosted on the Raspberry Pi. It consists of tables for storing information on customers, products, admin, and transaction records. ORM (Object-Relational Mapping) may be used in Node.js to interface with MySQL. RFID Integration: An RFID reader is connected to the Raspberry Pi to authenticate users based on RFID cards. When the user presents their RFID card, the system will match the card with stored user information, allowing them to access the checkout page or process a payment.

LCD Display: The LCD display serves as an interface for showing payment information, such as total price, payment status, and transaction results. The Raspberry Pi communicates with the LCD through GPIO pins, providing real-time updates for the user.

Servo: Servo working as a barrier to permit customer enter to the store for buying or after checking out, the barrier turn up and permit customer go out and ending shopping process.

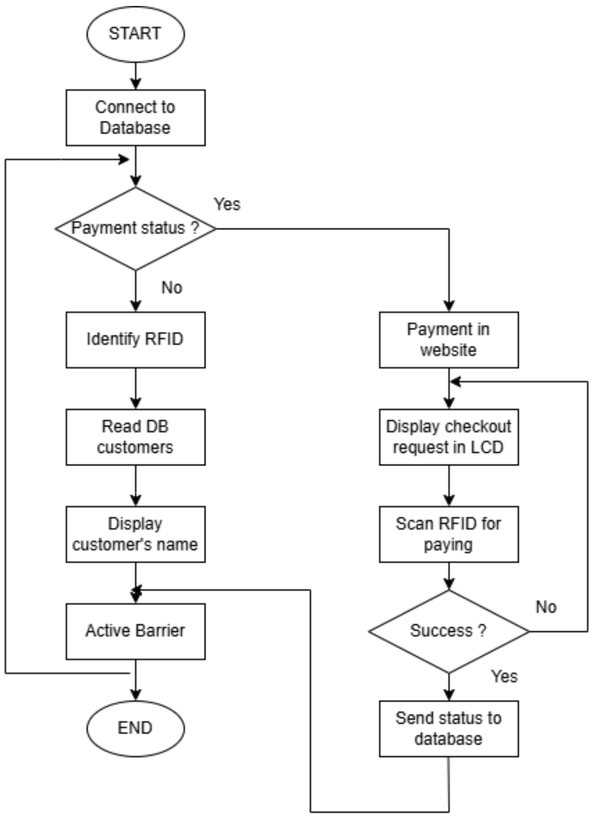

Figure 3.2. System workflow

3.2. Database design and Implementation

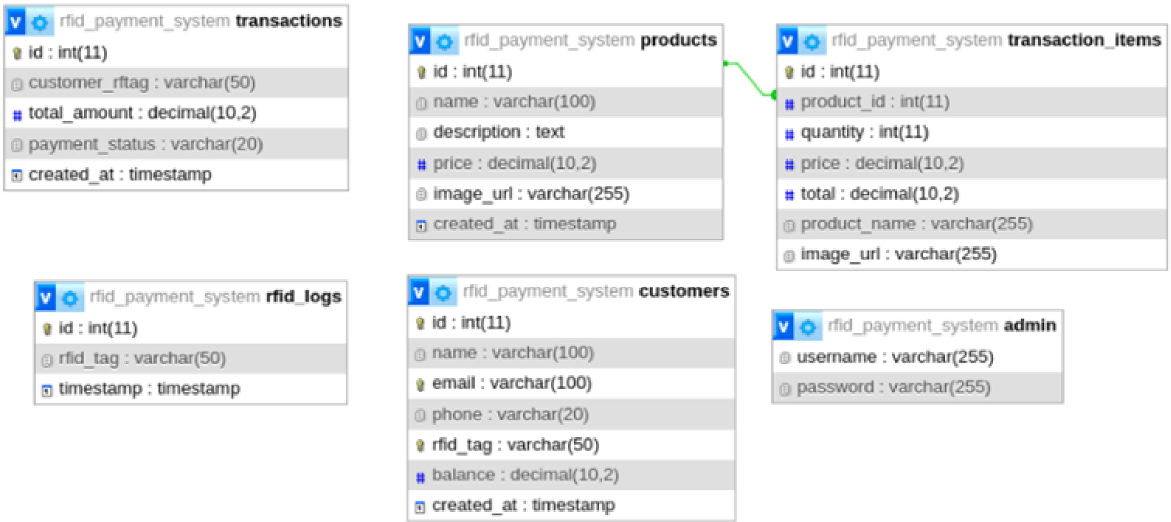

Figure 3.3. Database Diagram

a. transactions Table

Purpose: Stores information about each payment transaction.

Columns:

- **id** (int(11)): Primary key, uniquely identifies each transaction.
- **customer_rftag** (varchar(50)): Foreign key, references the `rfid_tag` from the customers table, identifying the customer involved in the transaction.
- **total_amount** (decimal(10,2)): The total value of the transaction.
- **payment_status** (varchar(20)): Indicates the current status of the transaction (e.g., "waiting", "success", "failed").
- **created_at** (timestamp): Timestamp indicating when the transaction was created.

Relationships:

- Linked to the `customers` table via `customer_rftag`.
- Linked to the `transaction_items` table.

b. products Table

Purpose: Contains details about the products available in the system.

Columns:

- **id** (int(11)): Primary key, uniquely identifies each product.
- **name** (varchar(100)): Name of the product.
- **description** (text): A textual description of the product.
- **price** (decimal(10,2)): Price of the product.
- **image_url** (varchar(255)): URL for the product image.
- **created_at** (timestamp): Timestamp indicating when the product information was added.

Relationships:

- Linked to the `transaction_items` table through `product_id`.

c. transaction_items Table

Purpose: Details which products were included in each transaction, along with their quantities and prices at the time of purchase.

Columns:

- **id** (int(11)): Primary key, uniquely identifies each transaction item.
- **product_id** (int(11)): Foreign key, references the `id` of the products table, identifying the product in the transaction.
- **quantity** (int(11)): Quantity of the product purchased in this specific transaction.
- **price** (decimal(10,2)): The price of the item at the time of the transaction (may differ from current product price in the product table).
- **total** (decimal(10,2)): The total price of all the units of an item (`quantity * price`).
- **product_name** (varchar(255)): The name of the product at the time of the transaction.

- **image_url** (varchar(255)): The image URL at the time of the transaction.

Relationships:

- Linked to `products` via `product_id`.
- Implicitly linked to `transactions` through the relation between `transactions` and `customers`.

d. customers Table

Purpose: Stores details about the customers using the system.

Columns:

- **id** (int(11)): Primary key, uniquely identifies each customer.
- **name** (varchar(100)): Customer's name.
- **email** (varchar(100)): Customer's email address.
- **phone** (varchar(20)): Customer's phone number.
- **rfid_tag** (varchar(50)): Unique identifier for the customer's RFID tag. This field acts as a foreign key and links with `transactions` table on `customer_rftag` column.
- **balance** (decimal(10,2)): Customer's current balance.
- **created_at** (timestamp): Timestamp indicating when the customer was registered.

Relationships:

- Linked to the `transactions` table via `rfid_tag`.

e. rfid_logs Table

Purpose: Keeps a log of RFID tag scans.

Columns:

- **id** (int(11)): Primary key, uniquely identifies each log entry.
- **rfid_tag** (varchar(50)): The RFID tag that was scanned. This could potentially be the tag of a customer or a product.

- **timestamp** (timestamp): Timestamp indicating when the RFID tag was scanned.

Relationships:

- Acts as an audit trail for RFID scans and does not have direct relationships with other tables.

f. admin Table

Purpose: Stores login credentials for the administrative users of the system.

Columns:

- **username** (varchar(255)): Admin username.

- **password** (varchar(255)): Admin password.

Relationships:

- Does not have direct relationships with other tables.

3.3. Booking website features

The convenience store website includes several essential features designed to provide a smooth and efficient shopping experience for users. These features cover everything from browsing items to completing purchases, ensuring that customers can easily access products and services.

Product catalog

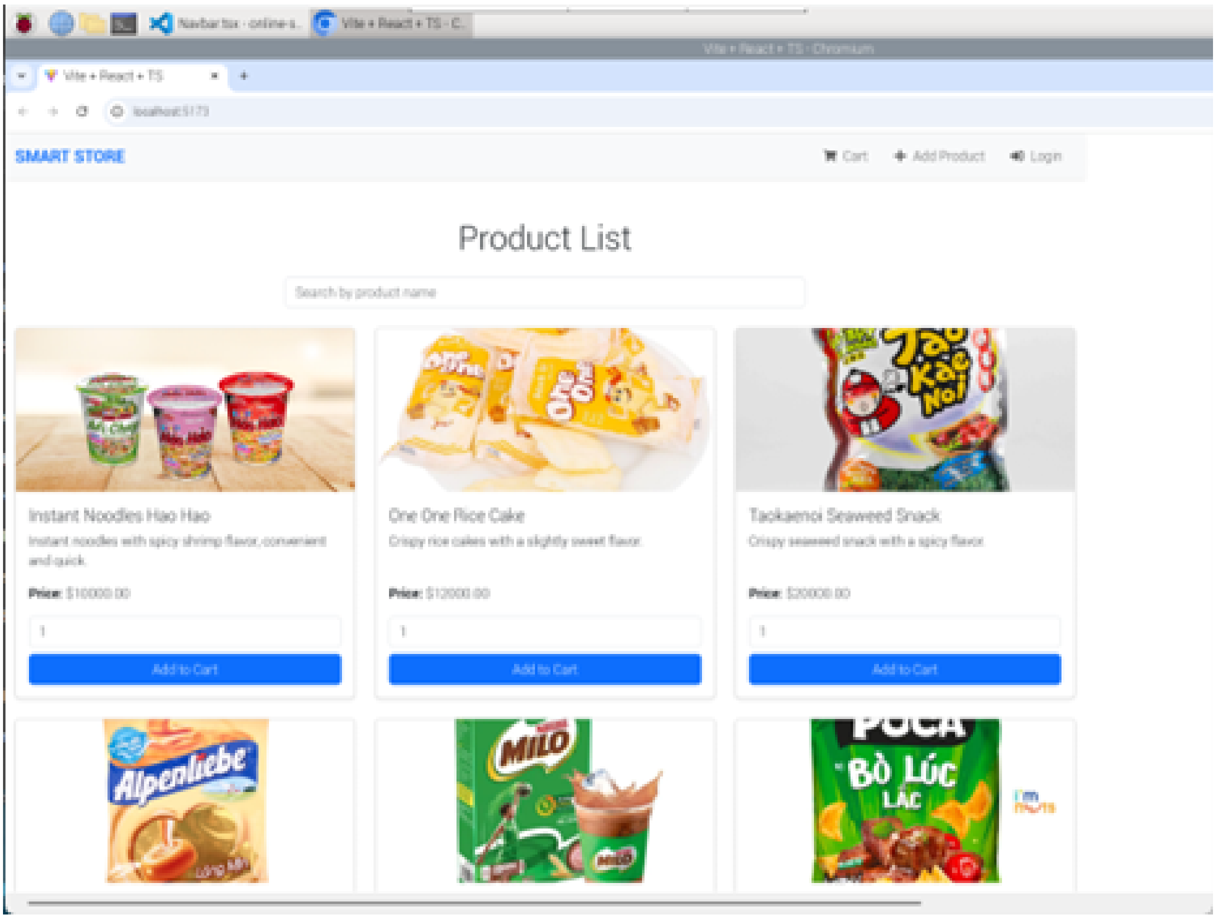

Figure 3.4. Product catalog

The product catalog is a key feature of the convenience store website. It allows users to browse through a variety of everyday items available for purchase. The catalog is structured to make it easy for users to find what they need quickly and efficiently.

Categories: Products are organized into different categories (e.g., Snacks, Beverages), making it easier for users to find specific types of products. Filters and Sorting: Users can filter products by different attributes such as price, popularity, or brand. Sorting options are also available to help users organize products based on their preferences.

Product Details: Each product page includes detailed information, such as price, quantity available, product description, and images. This helps customers make informed purchase decisions.

Search Functionality: A search bar is available, allowing users to search for specific items by name or keyword.

Shopping cart

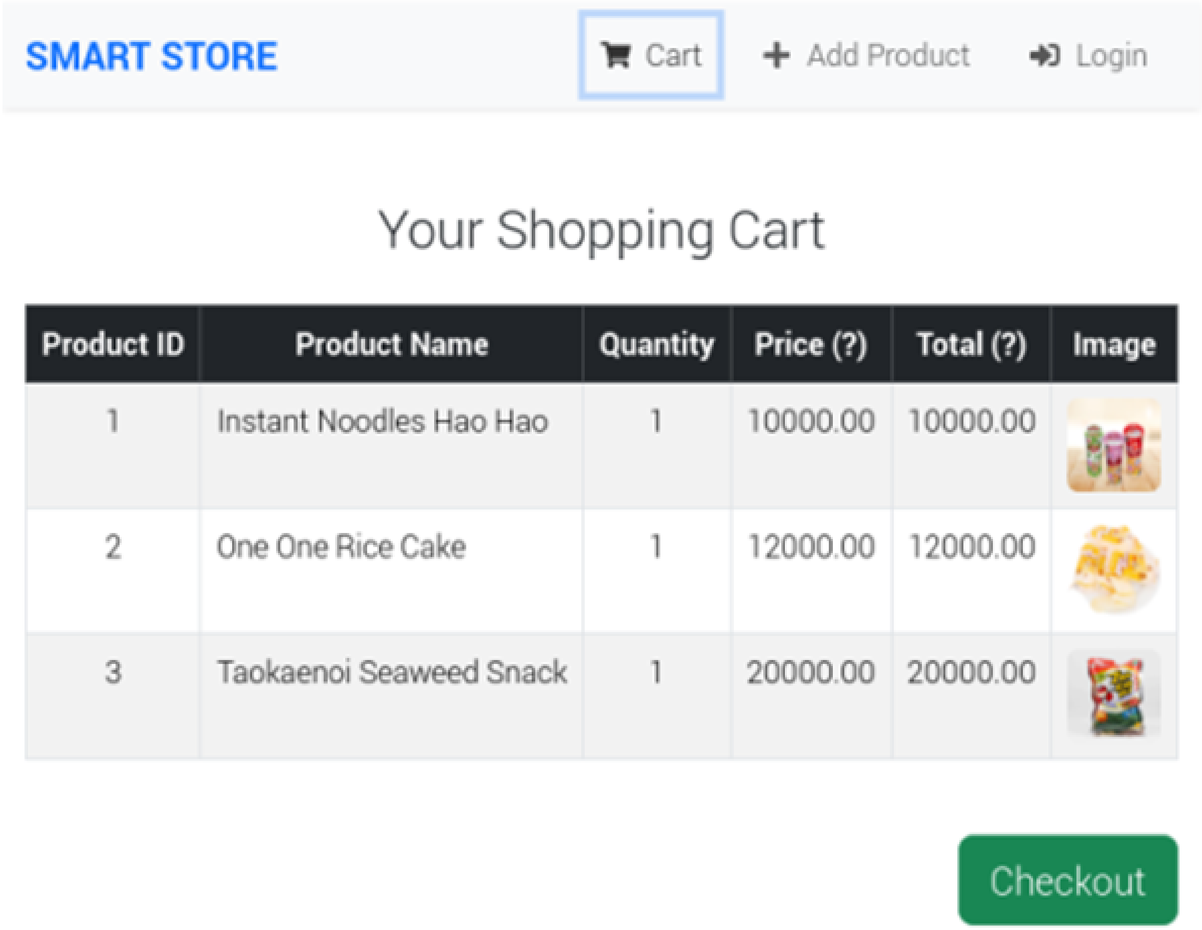

Figure 3.5. Shopping car

The shopping cart feature allows users to collect and manage their selected items before proceeding to checkout. Adding to Cart: Users can easily add products to their cart by clicking an "Add to Cart" button next to each item. The cart automatically updates to reflect the number of items and total cost.

Quantity Adjustments: Users can adjust the quantity of items in their cart or remove unwanted products. This flexibility allows for precise control over their purchases.

Temporary Storage: The cart can be stored in the user's session or database, ensuring that items remain in the cart even if the user navigates to other pages or logs out.

Cart Preview: Users can preview their cart on every page, displaying the items selected, quantities, and total cost, so they can easily review before checkout.

Checkout

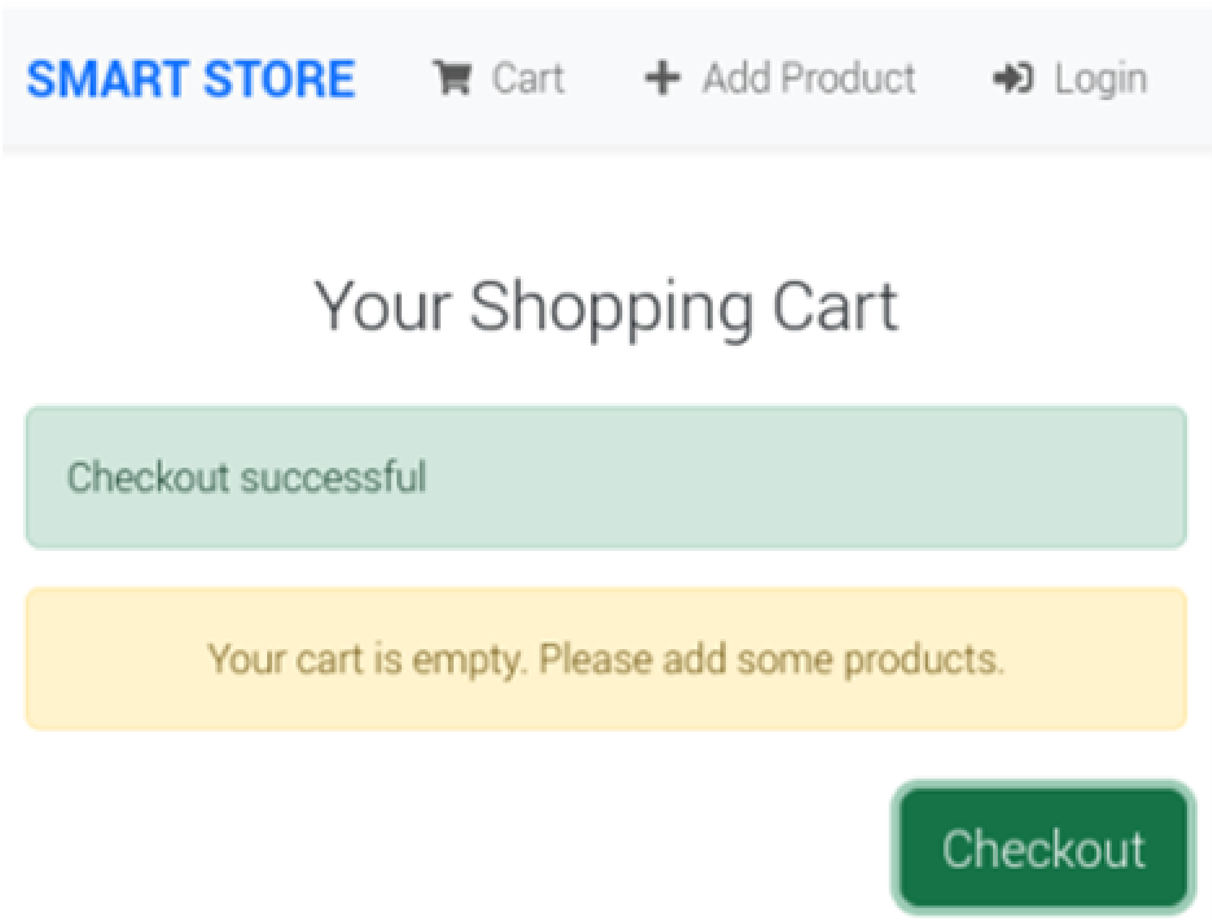

Figure 3.6. Checkout

The checkout process is designed to be straightforward, guiding users through the final steps of completing their purchase.

Cart Review: Users are presented with a summary of the products in their cart, along with the total cost, before proceeding to checkout. They can modify their selections if necessary. Shipping Information: Users must provide shipping details, including name, address, and contact information. The website can store this information for future purchases, speeding up the checkout process for returning customers.

Payment Method: Users can choose from various payment options, including credit/debit cards and other payment methods. The payment gateway securely processes payments.

Confirmation: After submitting the payment, users receive a confirmation page showing their order details and payment status. A confirmation email is also sent for record-keeping.

Order history

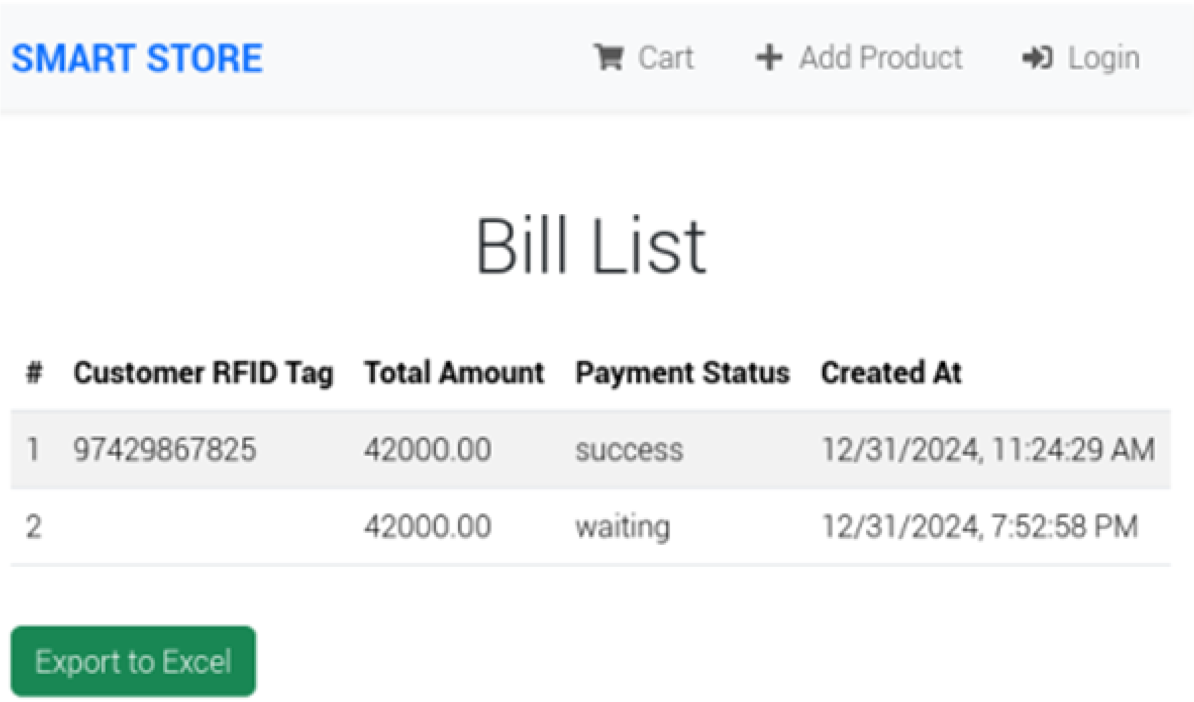

Figure 3.7. Order records

The order history feature allows admins to track and review their previous purchases.

View Past Orders: Admins can access a history of all their past orders, including product details, quantities, prices, and payment status.

Order Status: Each order is labeled with its current status (e.g., processing, shipped, delivered). This helps users keep track of their order's progress.

Reorder Items: Customers can easily reorder products they have purchased previously, which is convenient for those who frequently buy the same items. Downloadable Receipts: Each order has a downloadable receipt that contains detailed transaction information, useful for record-keeping or returns.

3.4. Integration of RFID and software

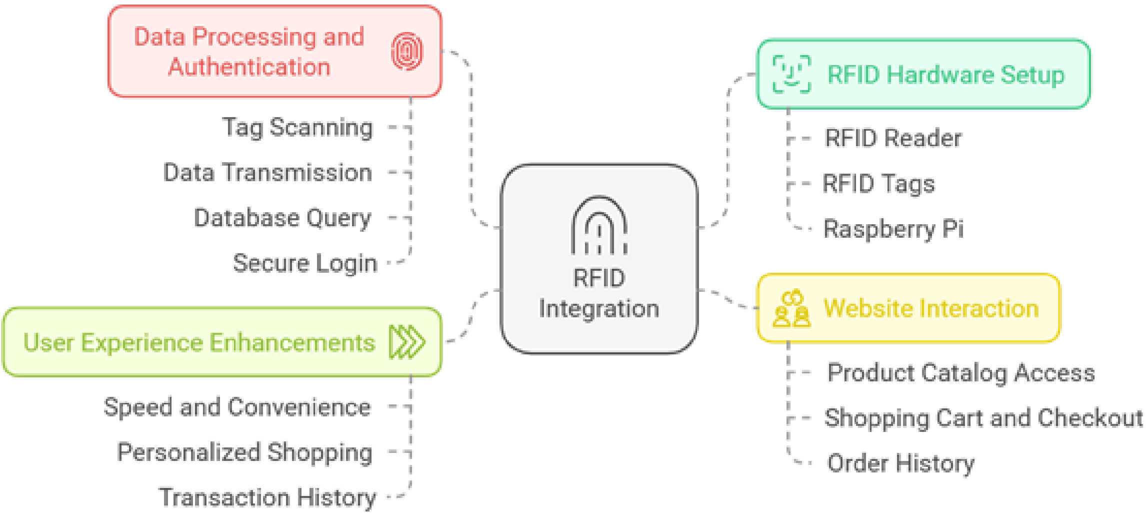

Figure 3.8. RFID Integratiom

- RFID Hardware Setup

The process begins with the RFID hardware, which includes an RFID reader connected to the Raspberry Pi. The RFID reader is responsible for scanning RFID tags, which are embedded in physical cards or other objects, each containing a unique identifier. The setup involves the following key components:

- **RFID Reader:** A device capable of reading RFID tags, such as the RC522 module, connected to the Raspberry Pi via GPIO pins.
- **RFID Tags:** Physical cards or key fobs embedded with a unique RFID code, typically in the form of an ID number.
- **Raspberry Pi:** A small, low-cost, single-board computer that acts as the central processing unit for handling the RFID data, querying the database, and interacting with the website's backend.

- RFID Data Processing and Authentication

When an RFID tag is scanned, the following sequence of operations takes place:

- **Tag Scanning:** The user scans their RFID tag by holding it near the RFID reader. The reader captures the unique ID embedded in the RFID tag and sends this information to the Raspberry Pi for processing.

- **Data Transmission:** The Raspberry Pi receives the RFID tag ID from the reader. This ID is sent to the backend server running the Node.js application for further processing.

- **Database Query:** The Python server queries the MySQL database to find a match for the scanned RFID tag ID. Each customer's RFID ID is stored in the database, associated with their account information such as their username, profile, and purchase history. If the RFID tag matches an existing entry in the database, the customer is entered or paid. If there is no match, the system denies access and prompts the customer to try again.

- **Secure Login:** Rather than relying on a traditional username and password, the RFID tag serves as a secure login method. The RFID tag ID is used as a unique key to identify the user and authenticate them for various website functions. This is especially useful for users who prefer a quicker login process or wish to avoid remembering multiple passwords.

- Interaction with the Website

Once authenticated, the user is granted access to various features on the convenience store website. These features include:

- **Product Catalog Access:** The user can browse through the catalog of available products and add items to their shopping cart.

- **Shopping Cart and Checkout:** Users can add products to their cart, proceed to checkout, and select a payment method. RFID authentication helps streamline the checkout process by remembering user preferences and payment methods.

- **Order History:** Once logged in, users can view their previous orders and transaction details, making it easy to reorder items they purchase frequently.

- **Secure Access:** The RFID system also adds a layer of security to the process. Since the RFID card is linked to the individual user, it prevents unauthorized access to personal data and financial transactions.

- User Experience Enhancements

The integration of RFID technology not only provides secure access but also significantly enhances the overall user experience by reducing friction during interactions with the website:

- **Speed and Convenience:** Users no longer need to remember passwords or usernames. A simple RFID tag scan allows them to log in quickly, especially in environments like stores or kiosks where speed and efficiency are crucial.
- **Personalized Shopping:** Once logged in, the website can offer personalized recommendations, show relevant products based on previous purchases, and pre-fill shipping information, enhancing the shopping experience.
- **Transaction History:** The user's transaction and order history can be easily retrieved and displayed based on the authenticated RFID tag, enabling quick access to past orders without requiring manual searching.

3.5. LCD display

The LCD display serves as an interface for showing payment information, customer's information such as customer's name, payment status, and transaction results. The Raspberry Pi communicates with the LCD through GPIO pins, providing real-time updates for the user.

CHAPTER 4
SYSTEM WORKFLOW

4.1. RFID authentication and Daa display

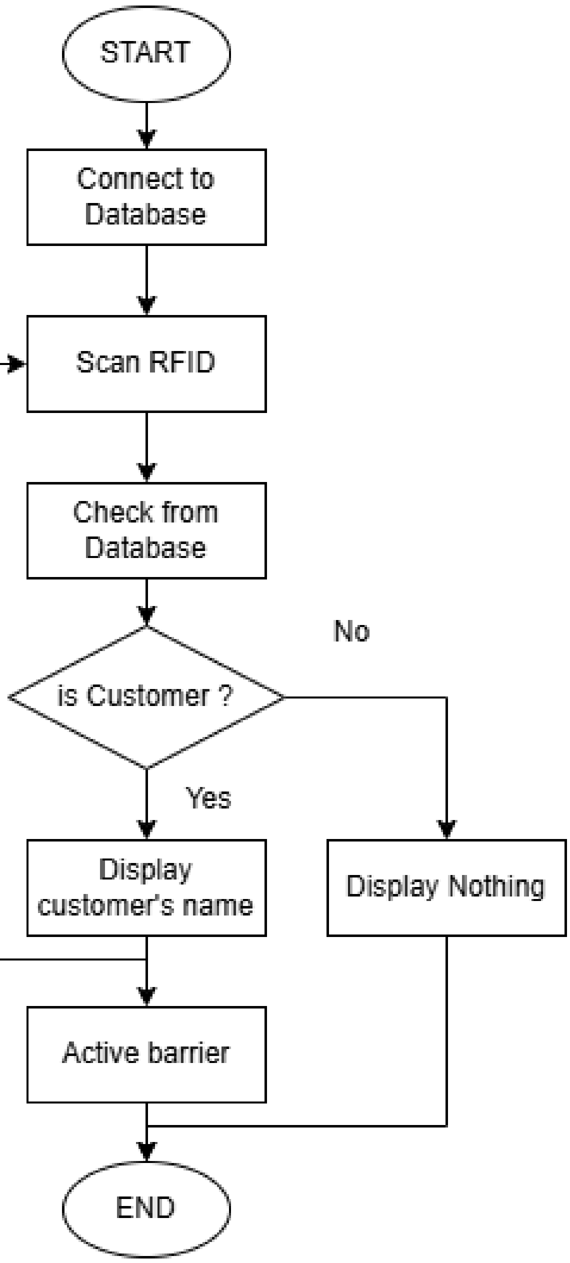

Figure 4.1. Entry workflow

The integration of RFID technology into the authentication process offers customers a seamless, secure, and efficient method to access the website without the hassle of remembering passwords or entering manual credentials. This process not only saves time for the users but also enhances the overall user experience, especially in high-traffic environments like a convenience store or booking system. The workflow is extended as follows:

a) RFID Tag Scanning

The first step in the RFID authentication process is the detection of the RFID tag, which acts as a unique identifier for the user. Here is a more detailed breakdown of the scanning process:

- **Tag Scanning:** The customer taps or holds their RFID card near the RFID reader, which is connected to the Raspberry Pi via the GPIO pins or USB interface. The RFID reader emits a signal that activates the RFID tag.
- The RFID tag sends back a unique identifier (ID), typically a 12-digit code or alphanumeric string, which is stored in the card's memory.
- **Data Collection:** The RFID reader processes the signal, extracts the unique identifier (UID) from the tag, and prepares it for transmission.
- The Raspberry Pi collects the UID from the RFID reader and prepares it for communication with the backend system.

b) Data Transmission to the Server

Once the RFID tag is scanned, the Raspberry Pi forwards the data to the backend server. This step involves multiple processes to ensure the secure transmission of the RFID tag's unique ID to the server for further processing.

- **Data Packaging:** The Raspberry Pi takes the UID from the RFID tag and packages it into a structured format, usually a JSON object, that includes the unique ID, timestamp, and device information.
- **HTTP Request:** The Raspberry Pi sends an HTTP request to the backend server powered by Python. This request can be made through various protocols such as REST or WebSocket, depending on the system configuration.

- The transmission may occur over a secure connection (HTTPS) to protect sensitive data, especially if the network involves public or shared channels.

- **Network Communication:** The server receives the request through the designated API endpoint. If the system is operating in a local network, the communication is fast and low-latency. However, in a cloud-based setup, secure data transmission protocols ensure that data remains encrypted and protected.

c) Database Query

After receiving the RFID tag's unique ID, the backend server processes it by querying the MySQL database to validate the user and fetch relevant information. The workflow includes the following steps:

- **Query Execution:** The backend system uses the unique RFID ID to query the MySQL database. A SQL query (e.g., `SELECT * FROM customers WHERE rfid_tag = 'UID'`) is executed to search for an entry that matches the scanned RFID tag.

- **User Lookup:** The MySQL database returns a record matching the RFID tag, which includes essential user data such as their name, contact information, order history, and any specific preferences or settings associated with their profile.

- **Optimized Query Performance:** To minimize latency, the system uses optimized indexing (e.g., indexing the `rfid_tag` field) to quickly locate the corresponding user record. In addition, caching strategies can be applied to reduce database load for frequent queries.

d) Customer Authentication

Once the server retrieves the user data from the database, it performs authentication to verify the legitimacy of the user and grant access to the system. This step ensures that only authorized users can interact with the system and proceed to the next stages of product selection and checkout.

- **Authentication Check:** If a matching entry is found in the database, the system authenticates the user by cross-referencing the UID against the stored record. The user is deemed authorized and can proceed.

- If the RFID tag does not match any entry in the database, the user is considered unauthorized, and the system may respond with an error message or prompt the user to register.

- **Authentication Failure Handling:** If authentication fails, the system alerts the user and gives them options to retry scanning, register a new RFID tag, or contact customer support for assistance.

e) Data Display

The backend server retrieves user-specific data such as order history, saved preferences, shipping address, and previously browsed items. This information is presented dynamically based on the user's account.

- **Fetching User Data:** The backend server retrieves user-specific data such as order history, saved preferences, shipping address, and previously browsed items. This information is presented dynamically based on the user's account.

- **Displaying Personalized Content:** The ReactJS frontend displays personalized content on the user's homepage or dashboard. For example, the homepage might show tailored product recommendations, loyalty points, or a personalized greeting ("Welcome, [User Name]!").

- **Order History and Preferences:** Users can also access their previous orders and view the status of current or past purchases. A dedicated section may list recent transactions with options to reorder items or track delivery statuses.

- **Enhanced User Experience:** The frontend dynamically adjusts the display based on the fetched data. For example, the system may highlight offers based on the user's past shopping patterns, or suggest related products based on their browsing history. This can improve engagement and drive sales.

- **Real-Time Updates:** If the user's session includes dynamic data, the frontend ReactJS components can update in real time to reflect any changes in product availability or new discounts that are applicable.

f) Continuous Data Synchronization

Throughout the entire process, there is a continuous synchronization between the hardware (Raspberry Pi, RFID reader) and software (backend server, MySQL database, ReactJS frontend). This ensures that all user interactions, from scanning the RFID tag to product selection and checkout, are seamless and synchronized in real time.

- **Session Management:** After successful authentication, the system manages user sessions, ensuring that all subsequent interactions, such as adding products to the cart or viewing order details, are properly linked to the user's profile.

- **Real-Time Communication:** If the system incorporates features such as live stock updates or instant order processing, the frontend and backend communicate in real time to ensure that users always have the most up-to-date information.

- **Security and Data Integrity:** Secure protocols are used throughout the process to ensure that all data transferred between the Raspberry Pi, backend server, and database remains secure and intact. The use of HTTPS, encryption algorithms, and secure authentication methods prevents unauthorized access and data breaches.

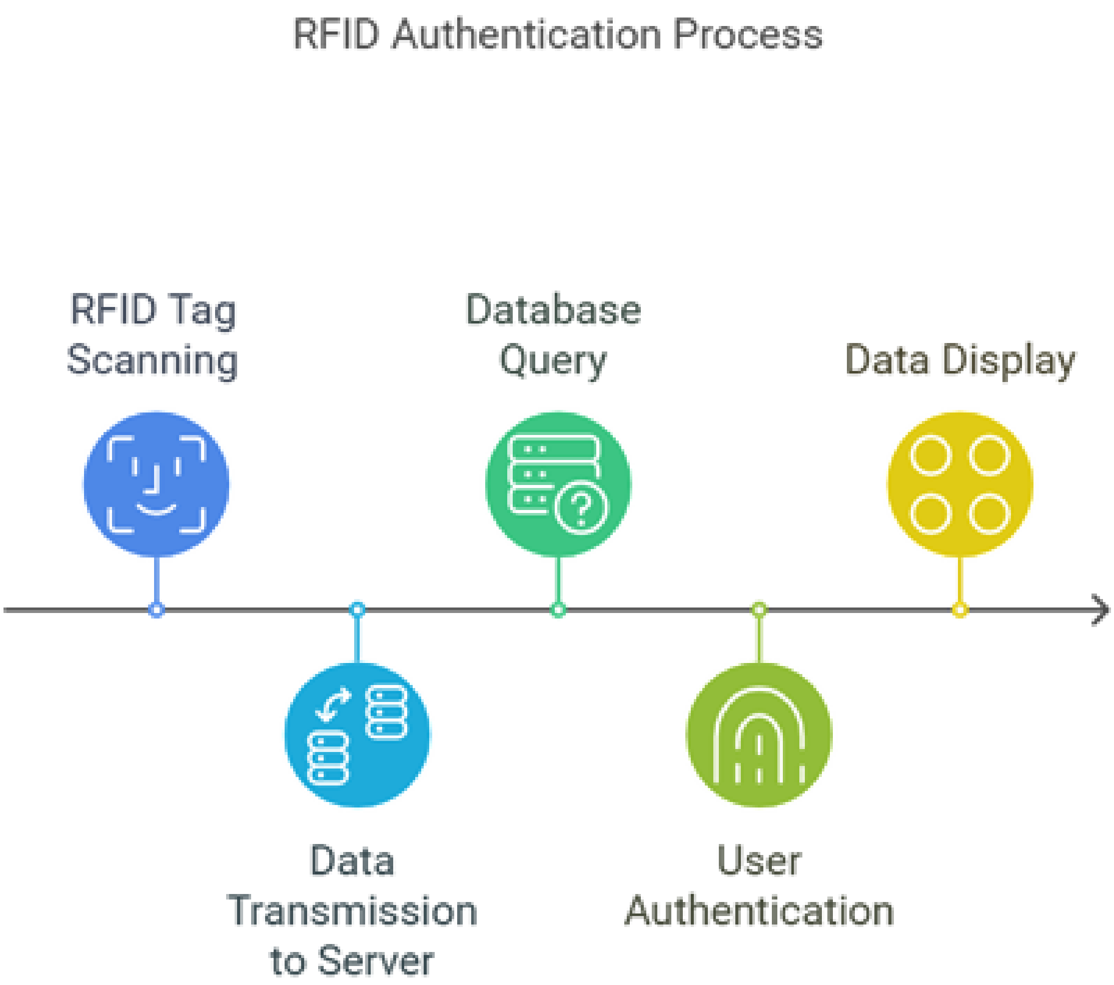

Figure 4.2. RFID Authentication process

4.2. Product selection and admin website

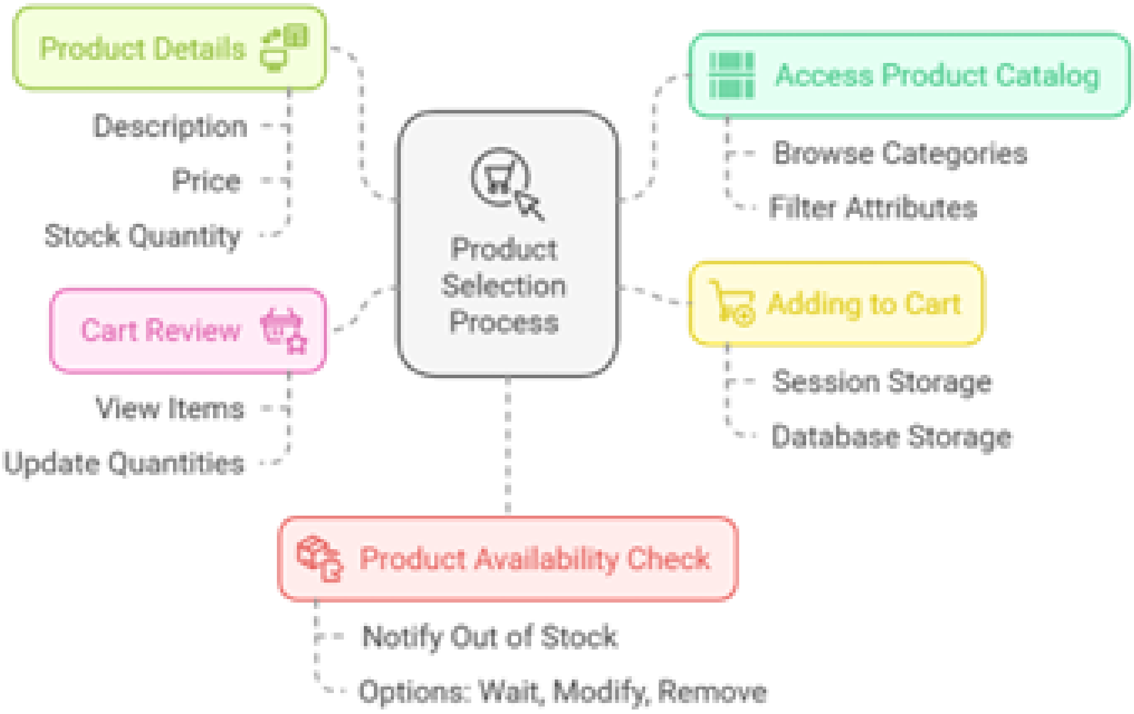

Figure 4.3. Product Selection via Booking Website

After successful RFID authentication, the customer can choose items to purchase. This part of the workflow involves browsing, selecting, and adding products to the cart:

- **Access Product Catalog:** The cashier accesses the website's product catalog, which is dynamically fetched from the MySQL database. Products are organized by categories, and users can filter them based on various attributes such as price, popularity, and ratings.
- **Product Details:** When the cashier selects a product, detailed information (such as description, price, stock quantity) is displayed. The user can then decide whether to add the product to their cart.
- **Adding to Cart:** Cashiers can add selected products to their shopping cart. The cart is temporarily stored either in a session or the database, allowing users to review or modify their selections later.

- **Cart Review:** The cashier can view all items in their cart, including product details, quantities, and total price. The cart is updated in real time as the user adds or removes items.

- **Product Availability Check:** The website can check stock availability for selected products. If an item is out of stock, the user will be notified with options to wait, modify the quantity, or remove the item.

4.3. Billing and Payment processing

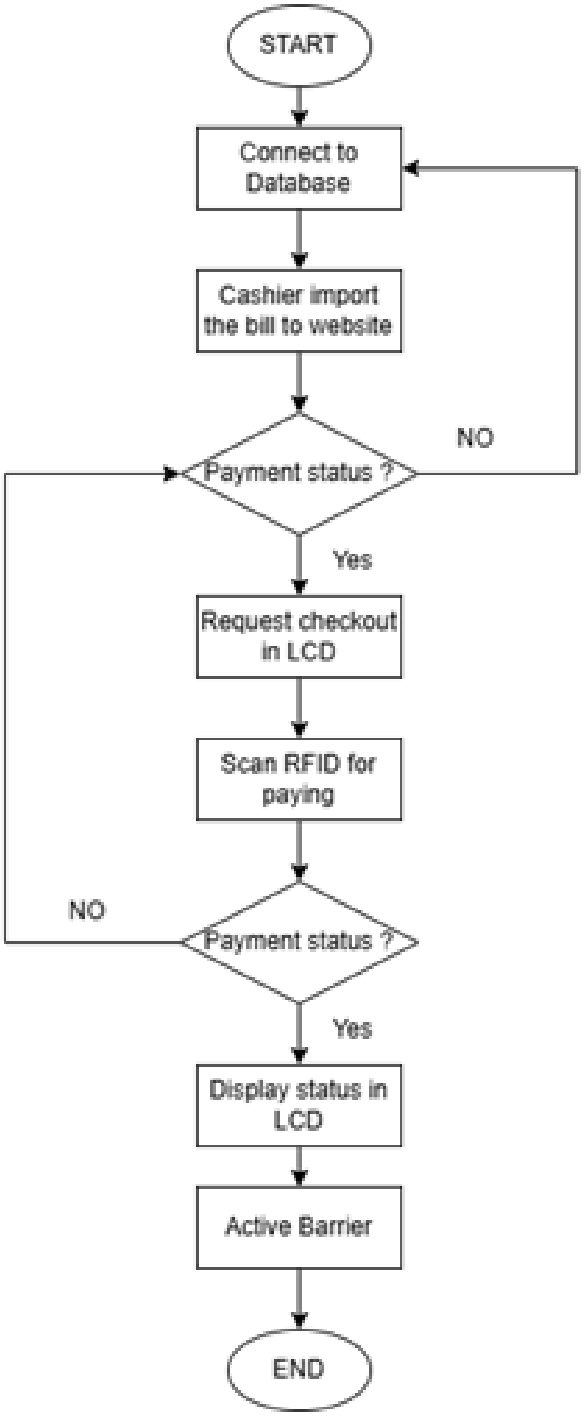

Figure 4.4. Payment process

Once the customer has selected products and is ready to check out, the system moves to the billing and payment processing stage:

- **Checkout Process:** The customer proceeds to checkout, where they can review their order, modify quantities, and choose their shipping method if applicable. They can also apply any discount codes or loyalty points that might be linked to their RFID tag.

- **Order Storage in Database:** The system stores the order details in the MySQL database under the customer's account, allowing them to view their purchase history in the future.

4.4. Hardware and Software Synchronization

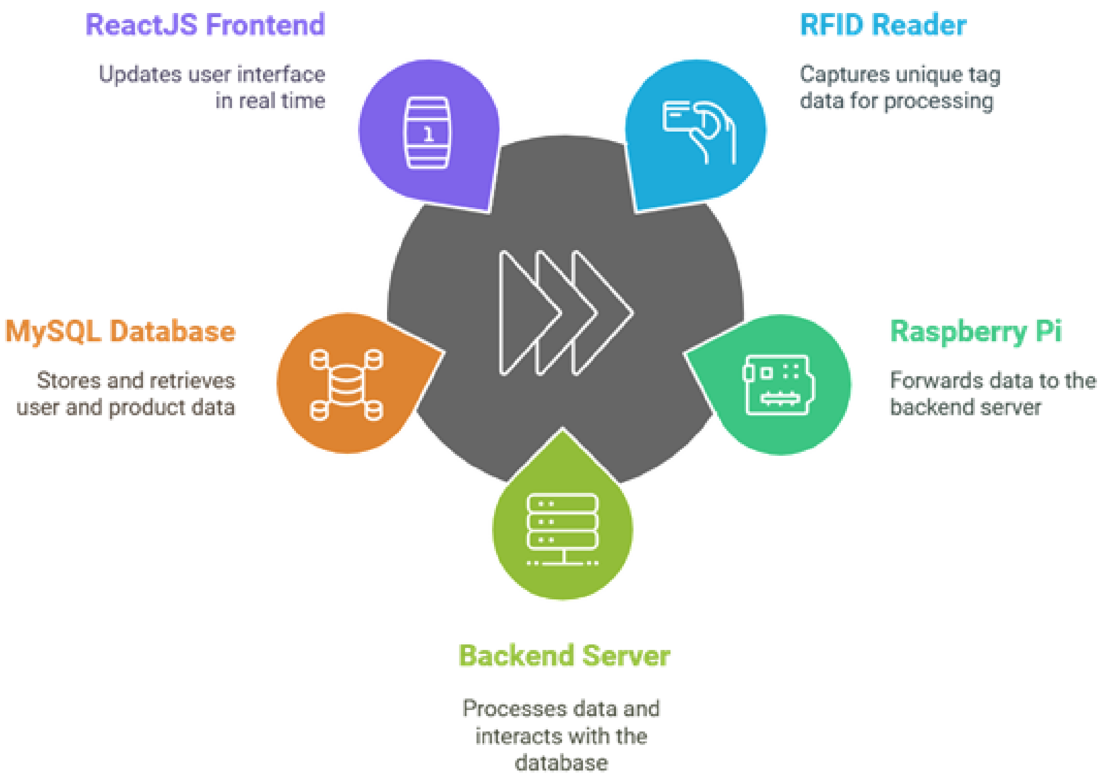

Figure 4.5. Hardware and Software Synchronization

The integration of the hardware (Raspberry Pi and RFID reader) with the software (booking website) ensures that the system runs smoothly and efficiently, providing a seamless experience for the user. The synchronization process involves continuous communication between the hardware components and the software backend:

Data Flow Between Hardware and Software

- The RFID reader (hardware) sends the unique RFID tag data to the Raspberry Pi, which forwards the information to the backend server via an API or HTTP request.

- The backend server (Python, Node.js and ExpressJS) interacts with the MySQL database to validate the user's credentials and retrieve relevant data (user profile, product catalog, etc.).

Real-Time Data Updates

- Any changes to the cart or order information are updated in real time both in the frontend (ReactJS) and the backend database. This ensures that product availability, pricing, and user data are always up to date.

- For instance, when a user adds a product to their cart, the cart data is immediately updated in the session or the database to ensure accuracy.

Seamless Transition Between Phases

- The system ensures smooth transitions between authentication, product selection, checkout, and payment. After RFID authentication, the user can immediately browse products, add them to the cart, and proceed to checkout. Payment is processed, and the order is confirmed, all within the same system workflow, without requiring any manual intervention.

Order History and Customer Data

- After a successful transaction, the user's purchase details are stored in the database. The user can view their past orders at any time by simply scanning their RFID tag again. This seamless integration between hardware (RFID) and software (MySQL database) ensures that user data is consistent and accessible across devices and sessions.

CHAPTER 5
TESTING AND RESULT

5.1. Testing methodology

Hardware Testing

The hardware components, including RFID scanners, microcontrollers, barriers, and displays, are thoroughly examined for functionality and reliability:

- **RFID Scanners:** Tested for accuracy in detecting RFID tags, sensitivity to various orientations, and resistance to signal interference.
- **Microcontrollers:** Evaluated for processing speed, data handling capacity, and communication with peripherals.
- **Barriers:** Checked for responsiveness and smooth operation, ensuring seamless access control.
- **Displays:** Verified for clear and prompt updates, ensuring they provide accurate feedback during transactions.

Software Evaluation

The system's software, including backend logic, databases, and user interfaces, is rigorously tested for performance and reliability:

- **Database:** Assessed for query efficiency, data integrity, and real-time synchronization with hardware inputs.
- **Backend Logic:** Tested for handling data from RFID scans, processing payments, and updating the system's status accurately.
- **User Interface:** Checked for ease of use, responsiveness, and compatibility across devices.

Process Validation (Identification and Payment)

The end-to-end process, from customer identification to payment and access control, is validated to ensure accuracy and efficiency:

- **RFID-Based Identification:** Tested for speed and accuracy in retrieving customer data from the database upon card scanning.

- **Payment Processing:** Assessed for reliability in calculating totals, charging linked accounts, and updating transaction logs.

- **Access Control:** Verified to ensure barriers operate correctly after successful payments, preventing unauthorized access or incomplete transactions.

5.2. Hardware functionality tests

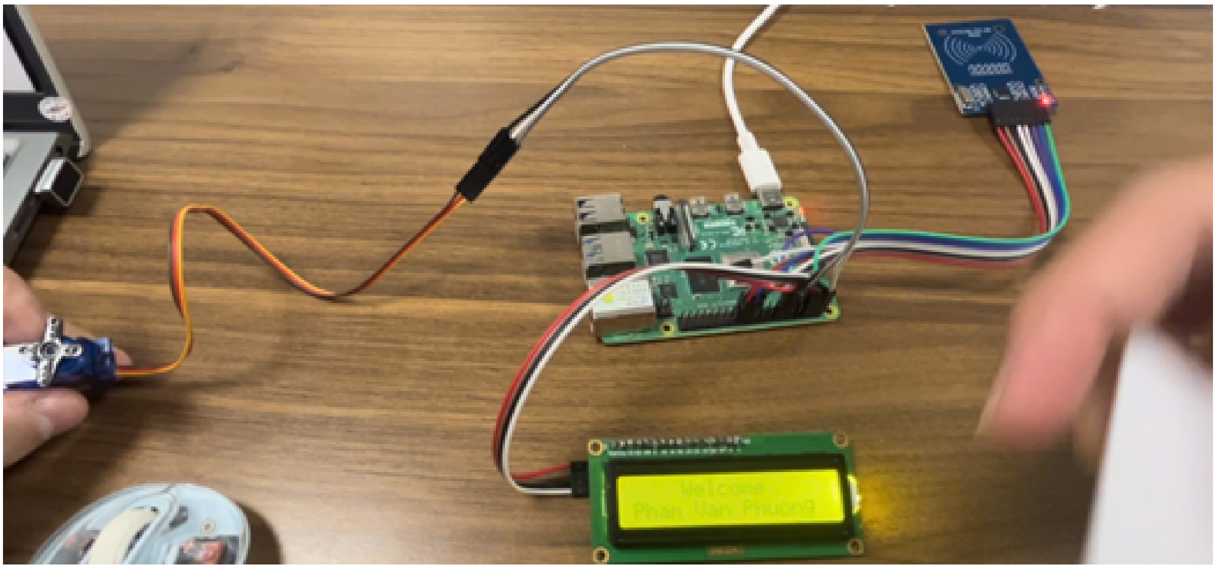

Figure 5.1. Check in via customer Pham Van Phuong

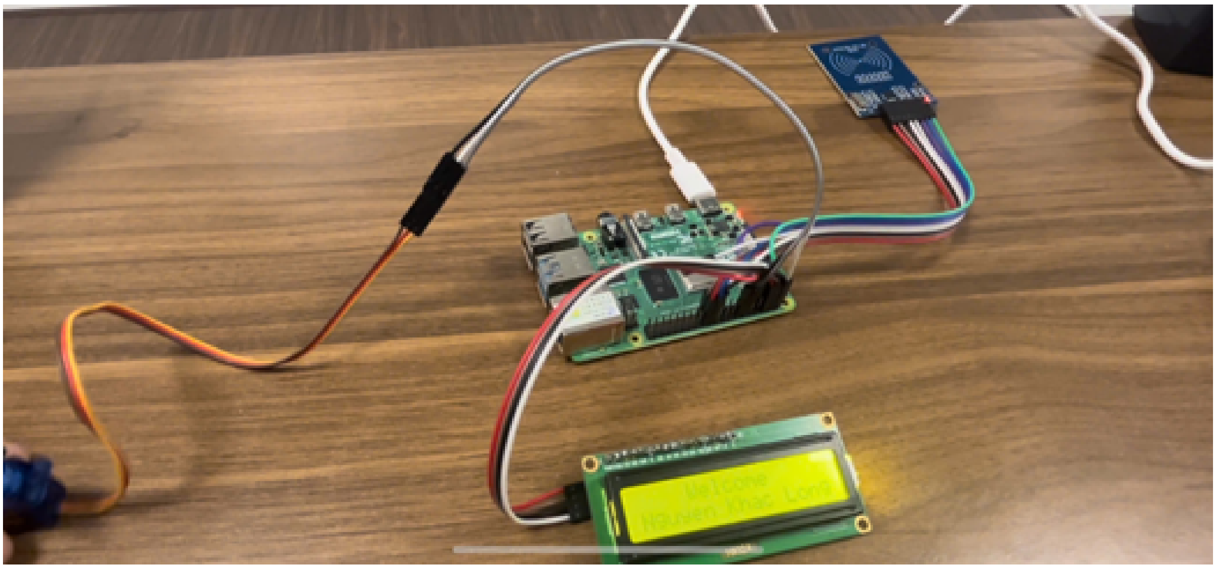

Figure 5.2. Check in via customer Nguyen Khac Long

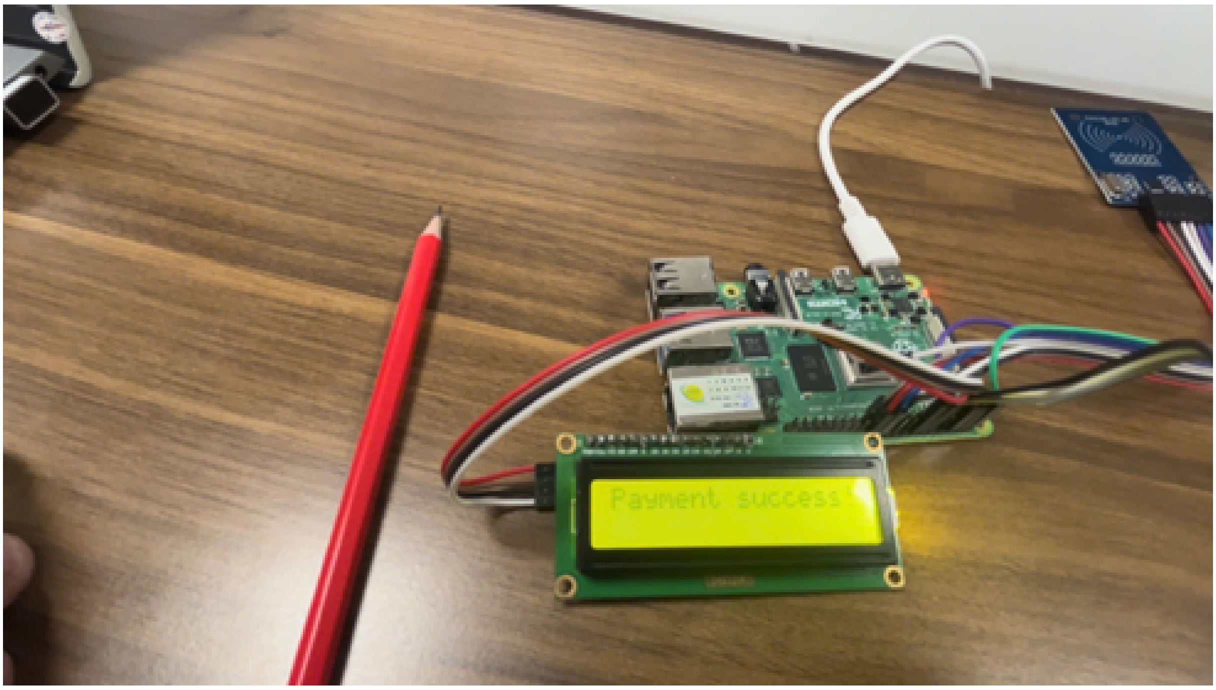

Figure 5.3. Payment success status

5.3. Integration Testing Results

The integration of hardware and software is tested to ensure seamless communication between modules. Key scenarios include:

- RFID card scanning triggering accurate customer identification and database retrieval.

- Real-time synchronization between database updates and display outputs. Smooth operation of barriers controlled by payment and identification confirmation.

- End-to-end functionality tests from customer entry, shopping, payment, to exit.

Results demonstrate that all modules communicate effectively, with response times within acceptable thresholds and minimal failure rates.

5.4. Screenshots and Systems demonstration

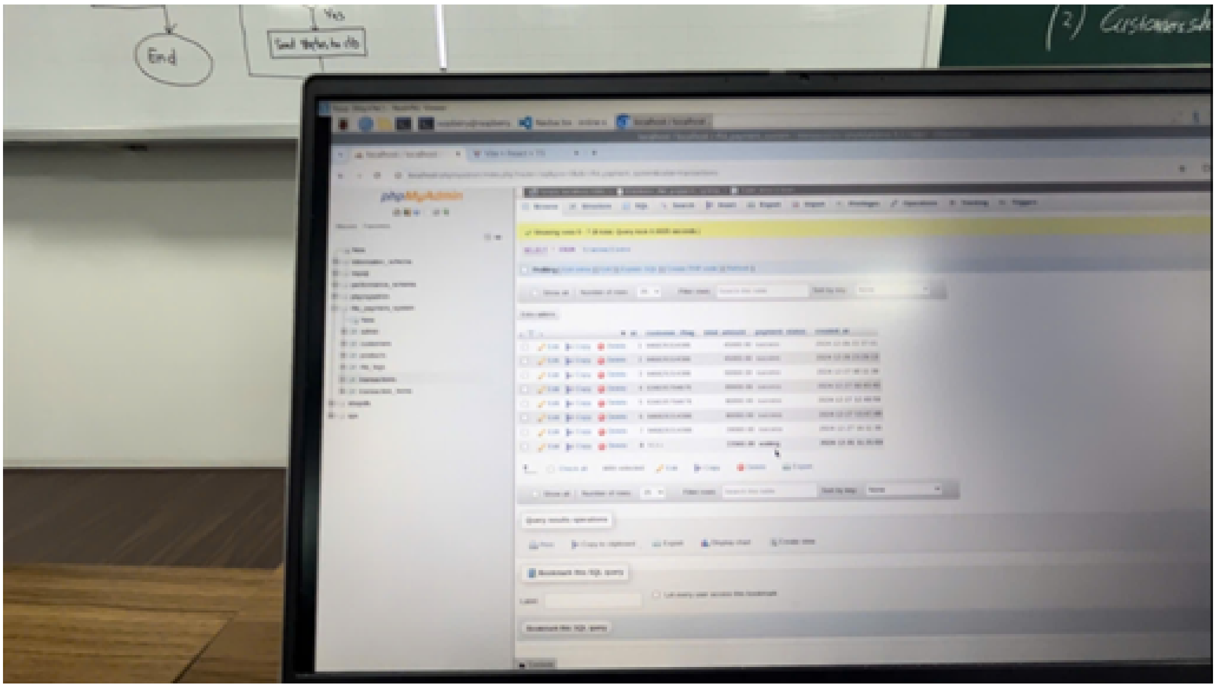

Figure 5.4. Waiting status sent to database

Simulation Video: `https://drive.google.com/file/d/1WfGedNuDOvaQmCwx` `view?usp=sharing`

Github: `https://github.com/LongNk0702/RaspberryPi4---Smart` `-convenient-store---fullstack`

CONCLUSION AND DISCUSSION

1. Key achievements

In this project, we successfully integrated RFID technology with a Raspberry Pi-based system to streamline the shopping experience in a smart convenience store. Key achievements include:

- **RFID Authentication:** The system enables customers to authenticate themselves quickly and securely via RFID tags, eliminating the need for traditional login methods such as usernames and passwords.
- **Seamless Shopping Process:** The integration between hardware (RFID readers, Raspberry Pi) and software (backend server, MySQL database, ReactJS frontend) allows customers to browse products, add them to the cart, and proceed with checkout efficiently.
- **Real-Time Updates:** The real-time synchronization between the backend and frontend ensures accurate product availability, pricing, and user data, enhancing the user experience.
- **Order History and Data Retrieval:** Customers can easily access their order history and other personalized details, improving the overall service and convenience of the shopping experience.

2. Challenges and Solutions

Several challenges were encountered during the implementation of the RFID-based smart store system, including:

- **Signal Interference:** RFID scanners sometimes faced interference in certain environments, leading to scanning errors.
- **Solution:** We optimized the placement of RFID readers and used more powerful antenna configurations to reduce errors and improve detection accuracy.

- **Database Synchronization:** Ensuring real-time data synchronization between the MySQL database and the frontend ReactJS application was a challenge.
- **Solution:** We implemented WebSocket-based communication to ensure fast, real-time updates between the backend and the frontend, particularly for cart and order data.
- **User Experience (UX) Design:** Initially, the user interface was not intuitive, leading to some confusion during product selection and checkout.
- **Solution:** The UX was improved by simplifying navigation and ensuring that key actions, such as adding items to the cart or completing checkout, were clearly highlighted and easy to follow.

3. Strengths

The system exhibits several strengths that make it a reliable and efficient solution:

- **Speed and Efficiency:** The RFID-based login and checkout process significantly speed up the user's interaction with the system, minimizing wait times and ensuring a smooth customer experience.
- **Security:** Using RFID tags as secure login identifiers ensures a higher level of protection for user accounts, reducing the risk of unauthorized access.
- **Scalability:** The modularity of the system allows for easy scaling, whether for adding more RFID tags, expanding product catalogs, or integrating additional features such as mobile app compatibility.
- **Personalization:** The ability to retrieve user-specific information and order history enhances the customer experience by providing tailored product recommendations and personalized services.

4. Limitations and Improvements

Despite its strengths, the system also has several limitations:

- **Limited Coverage:** RFID systems typically work best in close-range environments, which could limit their use in larger or more complex store layouts.

- **Solution:** To address this limitation, additional RFID readers can be strategically placed in large stores to ensure complete coverage.

- **Dependence on Hardware:** The system's reliance on hardware such as RFID readers and Raspberry Pi limits its ability to function in environments without these specific devices.

- **Solution:** Future iterations could focus on making the system more hardware-agnostic by using more widely available hardware or even developing a mobile app that supports RFID scanning.

- **Data Privacy Concerns:** Storing sensitive user data, including purchase history and personal information, raises potential privacy issues.

- **Solution:** We recommend implementing advanced encryption techniques for both data storage and transmission to ensure privacy and security.

5. Future Development Possibilities

There are several areas for future development that could further enhance the capabilities of the smart store system:

- **Mobile App Integration:** Developing a mobile app to complement the RFID system could allow customers to perform actions such as checking their order history or browsing the product catalog without needing to be physically present in the store.

- **Advanced Analytics:** The integration of machine learning algorithms could enable predictive analytics to offer personalized product recommendations based on shopping history and preferences.

- **Integration with IoT Devices:** Future versions of the system could be integrated with IoT devices to enable features like automatic restocking or real-time monitoring of inventory levels.

- **Expanded Payment Options:** Supporting a wider range of payment options, including mobile wallets or cryptocurrency, could attract a broader customer base.

6. Final Reflections and Takeaways

In conclusion, the development and implementation of the RFID-based smart store system has been a highly successful project, with significant contributions to improving customer experience and store efficiency. The use of RFID technology for seamless authentication and transaction processing has proven to be both reliable and secure, while the system's integration with a robust backend database and frontend interface offers scalability and real-time updates.

Key takeaways from this project include:

- **The Power of RFID:** RFID technology can significantly enhance the efficiency of everyday processes such as shopping, providing a more streamlined and user-friendly experience.
- **System Integration:** Successful integration of hardware and software is crucial for creating a functional and cohesive system that meets both user and business needs.
- **User-Centered Design:** The success of the system heavily relies on focusing on the user experience, ensuring the interface is simple, intuitive, and responsive.

The project serves as a foundation for future advancements in the smart retail space and demonstrates the potential of integrating emerging technologies to create more efficient, secure, and personalized shopping experiences.

REFERENCE

[1] Raspberry Pi Ltd, "Raspberry Pi 4 Model B", *Raspberry Pi (Trading Ltd)*, 2024, [Online]. Available: `https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf`.

[2] Handson Technology, "RC522 RFID Development Kit", 2024, [Online]. Available: `https://www.handsontec.com/dataspecs/RC522.pdf`.

[3] Handson Technology, "I2C Serial Interface 1602 LCD Module", 2024, [Online]. Available: `https://www.handsontec.com/dataspecs/module/I2C_1602_LCD.pdf`.