CartWave

Smart Cart System for Intelligent Shopping and Checkout Automation

Graduation Project Report

Presented to

Computer Engineering Department

Submitted for the Partial Fulfillment
of the Requirements for the Degree

Bachelor of Science in Computer Engineering

Team Members

Student Name	Student ID
Naif Hamed Dhawi Alghamdi	442000636
Anas Hassan Saleh Alghamdi	442000065

Project Advisor

Dr. Naif Abdullah M. Alzahrani

October, 2024

CartWave

Graduation Project

Submitted to

Computer Engineering Department

Approved:	Accepted:
Committee member 1 (signed)	Department Chair (signed)
Committee member 1 (printed) Date	Department Advisor (printed) Date
Committee member 2 (signed)	
Committee member 2 (printed) Date	
Committee member 3 (signed)	
Committee member 3 (printed) Date	

ABSTRACT

Improving customer experience is essential in digital business transformation projects. This project aims to revolutionize the traditional shopping process by making it interactive and automated. Utilizing a digital transformation lifecycle, we sought to enhance the shopping journey by addressing the passive and time-consuming nature of conventional shopping. To achieve this, we employed a design thinking process to collect requirements and develop initial prototypes [1], followed by an Agile-based methodology to design and implement the proposed solutions [2]. Our objective was to create a generic system applicable to any retail store.

The outcome of this transformation is CartWave, a smart shopping cart system that integrates cutting-edge technologies, including the Internet of Things (IoT) [3], computer vision [4], Bluetooth Low Energy (BLE) [5], and mobile application development [6]. IoT[3] enables connectivity and data sharing between devices, while computer vision automates product detection and recognition. BLE facilitates low-power, short-distance communication for location-based services. The mobile application enhances the shopping experience by providing personalized recommendations and account management.

CartWave comprises four integrated subsystems:

1. Raspberry Pi Core Module [7]: Acts as the central processing unit, integrating all other subsystems. It is installed on a shopping cart equipped with a camera module and display screen, providing features such as product details and total cost.

2. Computer Vision System: Automatically recognizes products picked up by customers using custom-developed algorithms.

3. BLE Location Detection System: Connects to sensors placed throughout the store, enabling location-based services like targeted advertisements on the cart's screen based on the store section.

4. Mobile Application: Allows customers to scan a personalized barcode to link their account, enhancing the shopping experience with personalized recommendations.

These subsystems work together to create an interactive shopping journey, eliminating the need for traditional checkout lines by triggering the payment process when the cart enters a designated payment zone. This provides a smoother, contactless checkout experience. Additionally, CartWave offers a scalable and efficient solution for retailers by reducing manual operations and enhancing overall shopping efficiency. The system also benefits retail management through real-time inventory tracking, sales monitoring, and digital inspection, streamlining stock management and reducing human error.

CartWave has the potential to significantly impact the retail industry by introducing automation and enhancing customer engagement, setting a new standard for the future of shopping.

ACKNOWLEDGEMENT

We extend our deepest gratitude to our parents for their unwavering support and encouragement throughout this journey. Their belief in us laid the foundation for our success.

We sincerely thank Assistant Professor Dr. Naif Abdullah M. Alzahrani of Electrical and Computer Engineering. His invaluable guidance, extensive knowledge, and constant encouragement were instrumental in shaping our project and personal growth.

We also express our appreciation to all the professors at the College of Engineering, Al-Baha University, whose comprehensive education and support equipped us with the skills necessary for this project. Their dedication to our academic growth laid the groundwork for our achievements.

Our heartfelt thanks go to our friends for their continuous encouragement and companionship during our university years. Their support has been invaluable.

Finally, we are grateful to Al-Baha University for providing the resources and opportunities that enabled us to pursue our studies and complete this project.

To everyone who guided and believed in us—thank you for being a source of inspiration and motivation.

TABLE OF CONTENTS

LIST OF FIGURES

A list of figures should be added with page number and captions.

LIST OF TABLES

A list of Tables should be added with page numbers and captions.

CHAPTER 1

INTRODUCTION

The retail industry is continuously evolving, driven by the need for enhanced customer experience and operational efficiency. As consumers demand more convenience and personalization, retailers are compelled to adopt innovative technologies that streamline shopping processes. The integration of smart solutions into retail environments has become essential to meet the expectations of modern consumers. Technologies such as the Internet of Things (IoT) [3], computer vision [4], and mobile applications are transforming traditional retail operations.

One of the key areas where technology has made a significant impact is the automation of the shopping experience. From self-checkout systems to mobile-based payment solutions, consumers are increasingly seeking convenient and seamless shopping options that save time and enhance their overall experience. Retailers are exploring ways to leverage these technologies to improve customer satisfaction and gain a competitive edge in the market.

The CartWave project introduces an innovative solution aimed at transforming the traditional shopping process by incorporating a smart shopping cart system. This system integrates technologies such as Bluetooth Low Energy (BLE) [5], computer vision [4], and mobile applications to automate and streamline the shopping experience for both customers and store owners. By utilizing BLE for zone-based detection, a camera for product recognition, and a mobile app for payment and account linking, CartWave seeks to reduce the need for manual checkout processes and improve overall shopping efficiency.

Due to the increasing reliance on digital solutions, the project also addresses challenges faced by retailers, such as managing inventory in real time, providing personalized advertisements, and enhancing customer engagement without relying on outdated paper-based flyers. The system is designed to be flexible and scalable, applicable to any retail environment, allowing store owners to improve their operational workflow while offering customers a smooth and efficient shopping journey.

This project focuses on leveraging the potential of the IoT [3] to revolutionize how consumers interact with the shopping environment. By integrating smart technologies, CartWave aims to create a future-oriented shopping experience, replacing traditional, inefficient methods with innovative solutions that benefit both customers and retailers alike.

1.1 **Problem Statement**

In today's retail environment, customers often face several challenges that hinder the overall shopping experience. Long checkout lines, inefficient inventory management, and reliance on outdated paper-based promotional materials diminish customer satisfaction and affect store operations. For example, a study by the National Retail Federation found that 40% of shoppers abandon purchases due to long wait times at checkout . Retailers struggle with ensuring accurate and up-to-date inventory, which can lead to either overstock or stock shortages, impacting sales and customer loyalty. Furthermore, the traditional shopping process, which involves manually scanning items at the checkout and dealing with printed invoices or paper receipts, is time-consuming and prone to errors.

As consumer expectations shift toward faster, more efficient, and personalized shopping experiences, there is a clear need for retailers to adopt smart technologies that address these inefficiencies. The inability to track inventory in real time, provide location-based personalized advertisements, or simplify checkout processes creates unnecessary friction in the shopping journey. This not only affects customer satisfaction but also prevents retailers from maximizing their operational efficiency and profit margins.

The CartWave project aims to address these critical issues by introducing an IoT-based smart shopping cart system [3] that automates various aspects of the shopping experience. Through the integration of BLE technology [5], computer vision [4], and a user-friendly mobile application, the system enables seamless shopping, real-time product recognition, and efficient checkout. Additionally, by providing real-time inventory updates and digital invoices, retailers can enhance inventory management and reduce reliance on manual operations.

Solving these problems is essential because it directly improves the customer experience, reduces operational costs, and positions retailers to compete more effectively in a fast-evolving market. In a world where consumers value efficiency and personalization, adopting such innovative solutions can result in increased customer loyalty, higher sales, and better resource management.

1.2 Project Key Objectives

The CartWave project aims to transform the traditional shopping experience by addressing the challenges identified in the retail environment through the integration of innovative technologies such as Bluetooth Low Energy (BLE) , computer vision [4], and the Internet of Things (IoT) [3]. The following key objectives have been established to guide the development and implementation of the project, ensuring a comprehensive solution that benefits both customers and retailers:

1. Automate the Shopping Process: Develop a smart shopping cart system capable of automatically recognizing and adding products to the customer's virtual cart using computer vision technology [4]. This objective seeks to streamline the shopping process by eliminating the need for manual scanning of products at checkout, thereby reducing wait times and enhancing customer convenience.

2. Enable Seamless Mobile Interaction: Integrate a user-friendly mobile application that allows customers to link their accounts to the shopping cart through a personalized barcode. This linkage will enable the cart to track purchases, display product details, provide personalized recommendations, and generate digital invoices, offering a smooth and personalized shopping experience.

3. Improve Inventory Management for Retailers: Provide store owners with a system that automatically updates inventory levels in real time based on customer purchases. This objective aims to help retailers maintain accurate stock levels, reduce the risk of overstocking or stock shortages, and enhance overall inventory management efficiency.

4. Implement BLE Zone-Based Advertisements: Utilize BLE sensors [5] strategically placed throughout the store to deliver personalized advertisements and promotions to customers based on their real-time location within the store. This reduces reliance on paper-based promotional materials and offers targeted

marketing solutions for retailers, enhancing customer engagement and increasing the effectiveness of marketing efforts.

5. Enhance the Checkout Experience: Simplify and automate the checkout process by enabling secure, contactless payment when the cart reaches a designated payment zone, recognized by the cart's BLE sensors . This objective aims to reduce checkout times, eliminate manual interactions at the point of sale, and improve overall customer satisfaction.

6. Provide Real-Time Data and Insights to Retailers: Equip store owners with access to real-time sales data, transaction histories, and detailed customer insights through a centralized management system. This will empower retailers to make informed decisions regarding product placements, stock replenishment, and promotional strategies, ultimately enhancing operational efficiency and profitability.

By achieving these objectives, the CartWave project intends to digitally transform the shopping journey for customers while offering retailers a modern, efficient, and scalable solution to optimize their operations. The successful implementation of these objectives is expected to result in measurable improvements in customer satisfaction, operational efficiency, and revenue growth, thereby positively impacting the retail industry and setting a new standard for the future of shopping.

1.3 Project Scope

The CartWave project focuses on developing a comprehensive smart shopping cart system that automates and streamlines the shopping experience for customers while providing retailers with advanced tools for inventory management, real-time data analytics, and personalized marketing. Although the system was developed and tested using a supermarket as a primary case study, it is designed to be versatile and adaptable to various retail environments. The architecture of CartWave allows for easy customization, making it suitable for implementation in other domains such as department stores, electronics outlets, and clothing stores. The features—including Bluetooth Low Energy (BLE)-based zone detection [5], product recognition using computer vision [4], and automated checkout—are universally applicable, providing flexibility for retailers across different sectors.

The core components of the CartWave project include the development of a functional smart shopping cart equipped with BLE sensors , a camera for product recognition [4], and an LCD screen for customer interaction. Additionally, a mobile application was developed to enhance the user experience for both customers and store owners. Customers can link their accounts, receive digital invoices, and track purchases, while store owners can view transaction records, monitor inventory, and manage stock levels in real time. The system integrates BLE zone detection [5], which allows for targeted advertisements based on the cart's location within the store and controls the checkout process by enabling payments only when the cart reaches a designated area.

An important feature of the system is product recognition via computer vision [4], allowing the cart to automatically identify products placed in front of its camera and add them to the customer's digital shopping cart. This is coupled with an automated checkout system that streamlines the payment process. Retailers benefit from real-time inventory management, where stock levels are automatically updated based on customer purchases, reducing the need for manual tracking and preventing issues such as overstocking or shortages.

While CartWave includes a range of advanced features, certain aspects are intentionally outside the scope of the project. Although the system was tested in a supermarket environment, specific features unique to supermarkets—such as managing fresh produce or integrating weigh stations—are not part of the project's core focus. Additionally, the project does not include integration with external payment gateways like PayPal or Stripe, as the focus is on developing an internal checkout system. Multi-language support is also not included in the current phase, with the system designed using a single-language interface. The project does not delve into advanced customer analytics, such as behavior prediction or loyalty programs, which could be considered in future expansions. Moreover, no architectural redesign of the physical store is included, aside from the placement of BLE sensors [5] and necessary hardware components.

Overall, the CartWave system is designed to enhance the shopping experience through automation and improved inventory management, focusing on the use of BLE , computer vision [4], and IoT technologies [3]. The Raspberry Pi [6] serves as the central hub for managing the various subsystems, ensuring seamless operation across the shopping cart's components. While future extensions of the project could include more advanced features, the current scope remains centered on delivering an efficient and seamless shopping journey for both customers and retailers.

1.4 Project Specification

The CartWave project combines advanced hardware and software technologies to develop a smart shopping cart system that enhances both customer experience and store operations. By leveraging machine learning for product recognition [4], Bluetooth Low Energy (BLE) for location-based services , and mobile applications for customer interaction, the system provides a seamless and automated shopping process.

1.4.1 Hardware Requirements and Specifications

The following hardware components form the foundation of the CartWave system:

1. Raspberry Pi 4 Model B [6]: Serves as the core processing unit, handling computations related to product recognition, BLE communication, and customer interactions. The Raspberry Pi's versatility and processing power make it ideal for integrating various functionalities in a compact form factor.

2. Raspberry Pi Camera Module[7]: Captures images of products placed in front of the camera. These images are then processed by the machine learning model to identify the products and add them to the digital shopping cart.

3. ESP32-C3 (BLE Zone Detection Sensors) [5]: These BLE sensors are strategically installed throughout the store to detect the cart's location. They enable location-based services such as displaying targeted advertisements on the cart's screen or enabling the checkout process when the cart reaches the designated payment area.

4. GeeekPi 10.1-Inch LCD Screen: Acts as the display interface for customers, providing real-time feedback on products added to the cart, total cost, and relevant promotions based on the cart’s location. The screen enhances customer engagement and ensures transparency throughout the shopping experience.

5. Shopping Cart**:** A traditional shopping cart retrofitted with the necessary hardware components to enable smart functionalities, including product recognition and customer interaction.

6. Power Bank**:** Provides a reliable power supply to all components, ensuring uninterrupted operation of the system throughout the shopping session.

1.4.2 Software Requirements and Specifications

The software stack behind CartWave is critical in powering the smart shopping cart operations. It consists of various platforms and technologies that enable product recognition, BLE communication, mobile interaction, inventory tracking, and web-based management features:

1. Teachable Machine [8]: Used to train a machine learning model for product recognition [4]. The trained model allows the cart's camera to accurately identify items placed in front of it and add them to the customer's digital shopping list. Teachable Machine provides an accessible platform for creating custom machine learning models without extensive coding.

2. Debian (Raspberry Pi OS)[9]: The operating system running on the Raspberry Pi, handling hardware interactions and serving as the foundation for the system's software components. Debian provides stability and compatibility with the required software tools.

3. Python[10]: Utilized for integrating the Teachable Machine model [8], enabling product recognition, and facilitating BLE communication for zone detection . Python scripts manage data flow between hardware components and the cart's interface, orchestrating the system's functionalities efficiently.

4. Arduino IDE[11]: Used to program the ESP32-C3 modules responsible for BLE signal detection and communication across different store zones [5]. This enables location-based services such as advertisement displays and checkout area recognition.

5. Flutter and Dart [12]: The framework and programming language used to develop mobile applications for both customers and store owners. Flutter allows for cross-platform development, ensuring compatibility with both Android and iOS devices. The mobile app enables customers to link their accounts to the cart, track purchases, and receive digital invoices, while store owners can manage sales data and monitor inventory levels.

6. PHP, HTML, CSS, and JavaScript[13][14][15]: These technologies were used to develop the web application that supports the cart's interface and backend services. The web application handles functionalities such as product display, database interactions, and communication with the system to manage product recognition and inventory updates.

7. Firebase [16]: A cloud-based platform used for data storage and management, including transaction histories, inventory levels, and customer profiles. Firebase ensures real-time synchronization of data between the cart, mobile application, and backend systems, providing accurate and up-to-date information.

8. Android APK and Achiever: Tools used to package and distribute mobile applications to customer devices, ensuring seamless installation and accessibility.

1.4.3 Anticipated Outcomes

The CartWave project delivers a fully automated shopping experience through its integration of hardware and software systems. The anticipated outcomes are as follows:

1. Automated Product Recognition and Cart Addition: Using the machine learning model trained via Teachable Machine [8], the cart's camera can automatically identify products placed in front of it. This feature eliminates the need for manual

barcode scanning, allowing for faster and more accurate addition of items to the customer’s digital shopping cart.

2. Seamless Mobile Integration: Customers can use the mobile application developed with Flutter [12] to scan a personalized barcode, linking their account to the cart. This enables real-time tracking of purchases, personalized recommendations, and receipt of digital invoices after checkout, enhancing the overall shopping experience.

3. Real-Time Inventory Management for Retailers: The system automatically updates inventory levels in real time based on customer purchases. Store owners can access sales data and monitor product availability through a centralized dashboard, facilitating efficient inventory management and reducing the risk of overstocking or shortages.

4. BLE-Based Location Services: BLE sensors [5] installed throughout the store detect the cart’s location, allowing the system to display targeted advertisements on the cart’s screen. This personalized marketing approach replaces traditional paper flyers, providing customers with relevant promotions as they navigate different sections of the store and increasing the effectiveness of marketing efforts.

5. Automated Checkout Process: Payment functionality is activated only when the cart reaches a designated payment zone, as detected by the BLE sensors . This ensures that customers cannot complete the checkout process until they are in the correct area, streamlining the payment process and eliminating the need for traditional cash registers or manual interactions at the point of sale.

6. Enhanced Customer Engagement and Satisfaction: By providing a user-friendly interface, personalized promotions, and a seamless shopping experience, Cart-Wave aims to increase customer satisfaction and encourage repeat business.

7. Improved Operational Efficiency for Retailers: The integration of real-time data analytics and automated processes reduces manual workload for staff, lowers operational costs, and allows retailers to make data-driven decisions to optimize store performance.

In summary, the CartWave project integrates advanced hardware and software components to deliver a cutting-edge smart shopping cart system that enhances the customer experience and optimizes retail operations. By leveraging IoT technologies [3], machine learning [4], BLE [5], and modern software development frameworks [12], CartWave provides a scalable and adaptable solution for the evolving needs of the retail industry. The system benefits both customers and retailers by streamlining the shopping process, offering personalized services, and improving operational efficiency, thereby setting a new standard for the future of shopping.

1.5 Development Methodology

The development of the CartWave project followed a structured and iterative approach, utilizing a variety of methodologies and emerging technologies to ensure a robust and scalable system. By adopting Agile practices [2], scenario-based analysis , and integrating both low-code/no-code frameworks alongside traditional coding platforms, Cart-Wave aimed to deliver an innovative solution that is flexible and adaptable to various retail environments.

1.5.1 Agile Methodology[2]

The Agile methodology was chosen as the primary development framework for this project due to its iterative and adaptable nature [2]. Agile allowed the team to develop, test, and refine the system in small, manageable increments, ensuring continuous feedback and improvement throughout the project lifecycle. This approach was particularly valuable as it enabled rapid responses to challenges or changing requirements, facilitating alignment with both technical and academic objectives.

Key aspects of the Agile approach included:

- Sprints**:** Short, focused development cycles where specific aspects of the project—such as Bluetooth Low Energy (BLE) integration [5], mobile app development, and product recognition—were addressed. Each sprint concluded with a review meeting to assess progress and plan subsequent steps.

- Continuous Feedback**:** Regular feedback sessions were held to gather input from all team members, refine features, and address issues early in the development cycle. This collaborative environment promoted transparency and ensured that the project remained on track.

- Collaboration Tools**:** Platforms like Miro [18] and ClickUp [19] were employed to manage tasks, facilitate collaboration, and maintain organization within the Agile workflow.

    - Miro**:** An online collaborative whiteboarding platform used for brainstorming, visualizing workflows, and mapping out project tasks. The team utilized Miro to sketch initial ideas, create diagrams, and develop workflows throughout the project development phase , enhancing communication and idea sharing.

    - ClickUp**:** A project management tool used to organize tasks, set deadlines, and track progress. ClickUp was essential for assigning tasks to team members, monitoring the overall project timeline, and ensuring that project milestones were met [19].

The adoption of Agile practices facilitated efficient project management, improved team coordination, and contributed to the timely delivery of project objectives.

1.5.2 Scenario-Based Analysis[17]

Scenario-based analysis was employed to ensure that the system could handle real-world shopping scenarios effectively . This approach allowed the team to analyze specific user interactions and edge cases, enhancing system reliability and user experience. Examples of scenarios considered include:

- A customer linking their mobile account to the shopping cart via barcode scanning.
- Transitioning between BLE zones [5] and triggering appropriate advertisements.
- Handling incorrect product placement or unrecognized items.
- Ensuring the checkout process is activated only in designated payment zones.

By simulating various shopping scenarios, the development team was able to design and optimize the system for a wide range of potential user behaviors, thereby improving overall performance and robustness.

1.5.3 Integration of Low-Code/No-Code Platforms

To accelerate development and reduce complexity, the project leveraged low-code/no-code platforms for certain components. For instance, Teachable Machine [8] was used to train the machine learning model for product recognition without requiring extensive custom code. This approach streamlined the development of the product recognition feature, allowing the team to focus on integration and testing.

Advantages of using low-code/no-code platforms included:

- Reduced Development Time: Enabled rapid creation of functional models without the need for in-depth programming expertise.
- Ease of Use: User-friendly interfaces facilitated quick iterations and adjustments.

However, the team balanced the use of low-code platforms with traditional programming to ensure flexibility and customization where necessary, particularly in integrating BLE and computer vision technologies [4].

1.5.4 Emerging Technologies

The CartWave project embraced a variety of emerging technologies to ensure the solution is both innovative and future-proof. Key technologies incorporated include:

- Internet of Things (IoT) [3]: The integration of BLE sensors [5] and Raspberry Pi devices [6] formed the backbone of the IoT infrastructure, enabling seamless communication between the cart, store environment, and mobile application. This IoT integration ensured real-time data flow, location-based services, and interaction with store inventory[3].

- Machine Learning [4]: Leveraging Teachable Machine [8] for product recognition, the system could identify specific items placed in front of the cart's camera, adding them automatically to the digital shopping cart. This technology provided a personalized and automated shopping experience, reducing human error and speeding up the process.
- Cloud Computing[20]: With the integration of Firebase [16], CartWave utilized cloud-based databases to store and manage real-time data such as transaction histories, product inventory, and customer information. This ensured that the system was scalable and accessible from anywhere, enabling seamless management of store operations.

The adoption of these technologies contributed to the project's innovation, enhanced functionality, and positioned CartWave as a forward-looking solution in the retail industry.

1.5.6 System Architecture and IoT Integration

As an IoT-based system [3], CartWave depended on the integration of multiple components—including BLE sensors, the Raspberry Pi platform [6], and mobile devices—all communicating in real time. Each cart interacted with BLE beacons placed in different zones of the store, allowing for location-specific services such as product advertisements and checkout availability.

The system's architecture was designed to ensure that hardware components (e.g., sensors, cameras, screens) communicated seamlessly with software components, including the machine learning model for product recognition and the mobile application for customer interaction. This IoT-driven approach enabled a highly dynamic and responsive environment, benefiting both customers and store owners through real-time updates and automation.

1.5.7 Benefits of Methodologies and Technologies

The combination of Agile methodology, scenario-based analysis, and emerging technologies resulted in several tangible benefits:

- Improved Team Efficiency: Agile practices and collaboration tools enhanced team coordination and productivity, leading to more efficient development cycles.

- Enhanced Product Quality: Continuous feedback and iterative development led to a refined and reliable system that meets user needs.

- Increased User Satisfaction: Scenario-based analysis ensured that the system provided a seamless shopping experience, catering to various user behaviors and preferences.

- Scalability and Adaptability: The use of IoT [3], machine learning [4], and cloud computing [20] provided a scalable infrastructure capable of adapting to future technological advancements.

In summary, the development methodology employed in the CartWave project was instrumental in achieving its objectives. By integrating Agile practices [2], leveraging low-code/no-code platforms , and embracing emerging technologies, the team delivered an innovative solution that addresses the challenges of the traditional shopping experience. The methodologies and technologies adopted not only facilitated efficient development but also ensured that the final product was robust, scalable, and aligned with the evolving needs of the retail industry.

CHAPTER 2

LITERATURE REVIEW

2.1 **Background**

Advancements in technology have significantly transformed the retail industry, leading to the adoption of innovative solutions that enhance customer experiences and optimize store operations. Technologies such as the Internet of Things (IoT)[3], machine learning[4], Bluetooth Low Energy (BLE)[5], and cloud computing[20] are increasingly being utilized to create dynamic, personalized, and efficient shopping environments. This section provides an overview of these technologies within the retail context, setting the foundation for understanding how it is utilized during the development of the CartWave project.

2.1.1 Internet of Things (IoT) in Retail

The Internet of Things (IoT) refers to a network of interconnected physical devices embedded with sensors, software, and other technologies to exchange data over the internet [3]. In the retail sector, IoT has enabled a multitude of applications[3], ranging from automated inventory management to enhanced customer engagement. For instance, smart shelves equipped with sensors can monitor stock levels in real time, reducing inventory discrepancies and enabling just-in-time restocking [21]. IoT devices such as beacons and smart carts can monitor customer movement within the store, providing valuable data for optimizing store layouts and offering personalized promotions [22]. By creating a more responsive and data-driven environment, IoT enhances operational efficiency and improves the overall shopping experience.

2.1.2 Machine Learning Applications in Retail

Machine learning, a subset of artificial intelligence, involves algorithms that learn from data to make predictions or decisions without being explicitly programmed [4]. In retail, machine learning has revolutionized areas such as demand forecasting, customer segmentation, and personalized recommendations. For example, retailers use machine learning models to analyze historical sales data and predict future demand, thereby optimizing inventory levels [23]. Additionally, machine learning enables the personalization of marketing efforts by analyzing customer behavior and preferences to deliver tailored product recommendations . These applications not only improve customer satisfaction but also increase sales and customer loyalty.

2.1.3 Bluetooth Low Energy (BLE) Technology[5]

Bluetooth Low Energy (BLE) is a wireless communication technology designed for short-range data exchange with minimal power consumption . BLE has gained prominence in retail for its role in enabling location-based services and proximity marketing. Retailers deploy BLE beacons throughout their stores to interact with customers' smartphones or smart carts, providing indoor navigation assistance, personalized promotions, and product information [22]. By leveraging BLE technology, retailers can enhance customer engagement, gather insights into shopping behaviors, and create a more interactive shopping environment.

2.1.4 Cloud Computing in Retail Operations[24]

Cloud computing involves delivering computing services—including servers, storage, databases, networking, software, and analytics—over the internet ("the cloud") to offer faster innovation and flexible resources . In the retail industry, cloud computing facilitates real-time data access, scalability, and centralized management of operations. Retailers utilize cloud services to store vast amounts of data, such as transaction histories, inventory levels, and customer profiles, which can be accessed and analyzed from any location. This enables efficient supply chain management, personalized customer experiences, and streamlined operations across multiple store locations.

2.1.5 Real-World Applications of Smart Retail Technologies

Several leading retailers have successfully implemented these technologies to enhance their operations and customer experiences. For example, Amazon Go stores utilize IoT sensors[3], computer vision, and machine learning algorithms to enable a checkout-free shopping experience [25]. Customers can enter the store, select products, and leave without waiting in line, as the system automatically detects the items taken and charges the customer's account. Similarly, retailers like Macy's and Target have implemented BLE[22] beacons to send personalized offers and product information to customers' smartphones as they navigate the store [26].

These implementations demonstrate the potential of integrating IoT[3], machine learning, BLE, and cloud computing to create innovative retail solutions. However, challenges such as high implementation costs, privacy concerns, and technological complexities remain. Addressing these challenges requires adaptable and scalable solutions that can be customized to different retail environments.

The main finding of literature review indicates that the integration of IoT[3], machine learning[4], BLE, and cloud computing technologies is reshaping the retail industry by enhancing customer experiences and optimizing operational efficiency. While significant advancements have been made, there is a need for versatile solutions that can overcome existing challenges and be adapted to various retail settings. The CartWave project builds upon this foundation by proposing a smart shopping cart system that leverages these technologies to provide a seamless and personalized shopping experience, which will be further detailed in subsequent sections.

2.2 Previous Work

The concept of smart shopping carts has gained significant traction in recent years as retailers seek innovative ways to enhance the shopping experience. Various companies have introduced systems that integrate advanced technologies like Bluetooth Low Energy (BLE) , computer vision [4], and the Internet of Things (IoT) [3] to create seamless, automated, and efficient retail environments. This section reviews notable previous work in smart shopping carts, including Amazon Fresh's Dash Cart [27] and the Caper Smart Cart [28], and compares them to the features and approach of the CartWave system.

Amazon Fresh Dash Cart

Amazon Fresh's Dash Cart is one of the most prominent smart cart systems currently in use. Designed to streamline the shopping process, the Dash Cart utilizes computer vision and sensor fusion to automatically detect items placed in the cart. Shoppers do not need to scan each item manually; instead, the cart recognizes products as they are added. Upon exiting the store through a designated lane, the system automatically charges the customer's account via the Amazon mobile application.

Key features of the Dash Cart include:

- Mobile Integration: Customers use the Amazon Fresh mobile app to scan a personalized barcode, linking the cart to their account. This integration allows for seamless tracking of purchases and automated payment processing.
- Product Recognition: The Dash Cart employs computer vision and sensor fusion technologies to detect and recognize items, adding them to the customer's virtual

cart. This eliminates the need for manual scanning and enhances the shopping experience by providing a faster checkout process.

- Automated Checkout: After completing their shopping, customers simply walk through a designated checkout lane, and the system processes the payment automatically. This contactless checkout experience reduces wait times and improves customer convenience.

While the Dash Cart effectively automates product recognition and checkout, it primarily relies on computer vision and does not utilize BLE technology for location-based services within the store. In contrast, CartWave combines BLE with computer vision, enabling real-time location tracking and location-based actions, such as displaying targeted promotions as customers move through different sections of the store [5].

Caper Smart Cart [28]

The Caper Smart Cart is another notable example of a smart shopping cart system. Similar to the Amazon Dash Cart, Caper uses computer vision to automatically detect and add products to a virtual shopping cart as they are placed in the cart. A distinctive feature of Caper is its touchscreen interface, which allows customers to view items added to their cart in real-time and track their total purchase cost. However, Caper does not leverage Bluetooth Low Energy (BLE) for location-based services[5], nor does it offer the same level of IoT integration found in CartWave[3]. The primary focus of Caper is on enhancing customer interaction through its touchscreen interface, improving the shopping experience by providing real-time feedback and recommendations.

Comparative Analysis:

While Amazon Fresh Dash Cart [27] and Caper Smart Cart share similarities with CartWave project in their use of computer vision for product recognition and their aim to simplify the checkout process, there are notable differences in the implementation approaches such as:

Integration of BLE Technology:

CartWave distinguishes itself by integrating BLE technology [5] to provide location-based services within the store. BLE sensors are strategically placed throughout the store to enable real-time location tracking, allowing the system to display targeted advertisements and promotions based on the customer's position. This adds a layer of personalization and enhances customer engagement, features not present in the other systems.

IoT Integration and Inventory Management:

CartWave incorporates IoT integration [3], offering real-time inventory management for store owners. This functionality enables retailers to monitor stock levels, track sales data, and manage inventory dynamically, optimizing store operations. Neither Amazon Fresh Dash Cart nor Caper Smart Cart provide this level of backend integration, focusing instead on front-end customer experience enhancements.

Comprehensive Solution:

By combining BLE, IoT[3], and computer vision [4], CartWave offers a more holistic solution that benefits both customers and retailers. It addresses not only the automation of the shopping and checkout process but also enhances operational efficiency and customer engagement through.

In summary, while existing smart shopping cart systems like Amazon Fresh Dash Cart and Caper Smart Cart offer innovative solutions to improve specific aspects of the shopping process, CartWave presents a more comprehensive approach. By integrating Bluetooth Low Energy [5], the Internet of Things [3], and computer vision [4], CartWave enhances the shopping experience through personalized, location-based services and provides valuable tools for retailers to manage operations effectively. This comprehensive integration addresses gaps in previous systems, offering an adaptable and scalable solution suitable for various retail environments.

The table below provides a comparative analysis of key features among the Amazon Fresh Dash Cart, Caper Smart Cart, and our proposed CartWave system. This comparison highlights the distinct functionalities and technological integrations that differentiate CartWave from existing smart shopping cart solutions.

Feature	Amazon Dash Cart	Caper Smart Cart	CartWave
IoT Integration	✔	✔	✔
BLE Tech	✖	✖	✔
Mobile integration	✔	✖	✔
Computer Vision	✔	✔	✔
Automated Checkout	✔	✔	✔
Location-Based Services	✔	✖	✔
Digital Invoices	✔	✔	✔

Table 1:Feature Comparison of Smart Shopping Carts

CHAPTER 3

REQUIREMENT ELICITATION

3.1 System Scenario

The system scenario provides an overview of how users are expected to interact with the CartWave system in a typical retail environment. It describes the anticipated user experience in a generic way, highlighting the system's functionalities from both the customer's and the retailer's perspectives. By understanding the flow of interactions, we can better identify and specify the system's requirements.

3.1.1 Generic System Usage

In a generic retail setting, the CartWave system aims to enhance the shopping experience through automation and personalization. The following scenario outlines how users are expected to use the proposed system:

1. Cart Acquisition and Account Linking:
    - Upon entering the store, a customer selects a smart shopping cart equipped with the CartWave system.
    - The customer opens the CartWave mobile application on their smartphone.
    - Within the app, the customer selects the option to link their account to the cart.
    - A unique QR code or barcode is generated by the app.

- The customer scans this code using the scanner or camera on the cart, establishing a secure connection between the cart and their account.

2. Product Selection and Recognition:

   - As the customer navigates the store, they select products they wish to purchase.
   - Before placing an item into the cart, the customer presents the product in front of the cart's camera.
   - The system uses computer vision and machine learning algorithms to recognize the product[4].
   - Once recognized, the product information (name, price, description) is displayed on the cart's screen.
   - The item is automatically added to the customer's digital shopping list, and the total cost is updated in real time.

3. Personalized Promotions and Recommendations:

   - The cart is equipped with Bluetooth Low Energy (BLE) sensors[5] that interact with beacons placed throughout the store[5].
   - As the customer moves through different store zones, the system detects their location.
   - The cart's display presents location-based promotions, advertisements, or product recommendations relevant to the customer's current aisle or section.

- This personalized content enhances the shopping experience by informing customers of relevant deals and new products.

4. Real-Time Cart Management:

    - The cart's interface allows customers to view their current shopping list, total cost, and any applicable discounts.
    - Customers can remove items from their digital cart via the interface if they change their mind.
    - The system ensures transparency and helps customers manage their budget and purchases effectively.

5. Automated Checkout Process:

    - Upon completing their shopping, the customer proceeds to the designated checkout area.
    - The cart's BLE sensors detect the proximity to the checkout zone.
    - The system prompts the customer to review their purchases and confirm the total amount.
    - Payment is processed automatically using the payment method linked to the customer's account (e.g., credit card, mobile payment service).
    - A digital receipt is sent to the customer's mobile application or email.

    - The customer can then exit the store without waiting in traditional checkout lines.

6. Inventory and Data Management for Retailers:

    - As products are added to and removed from carts, the store's inventory levels are updated in real time.

    - Retailers have access to sales data, inventory reports, and customer purchasing trends through a backend system.

    - This information aids in inventory management, restocking decisions, and strategic planning.

3.1.2 Project Implementation Case Study

While the CartWave system is designed to be a generic solution applicable to various retail environments, the implementation and testing of the system were conducted using a supermarket as the primary case study. The supermarket setting provides a comprehensive environment to evaluate the system's functionalities due to the diverse range of products and high customer traffic. The proposed implementation scenario in supermarket context as follows:

- Product Variety and Recognition:

    - The supermarket includes a wide array of products, from packaged goods to fresh produce.

    - The machine learning model was trained to recognize a selection of common items[4], allowing for effective testing of the product recognition feature.

- Challenges such as varying packaging designs and lighting conditions were addressed to improve recognition accuracy.

- BLE Zone Configuration[5]:
    - BLE beacons were strategically placed in different sections of the supermarket, such as dairy, bakery, and household goods.
    - The cart's BLE sensors interacted with these beacons to deliver location-based content.
    - Promotions specific to each section were displayed, enhancing the personalized shopping experience.
- User Interaction and Feedback:
    - Customers participating in the case study were provided with the CartWave mobile application.
    - Their interactions with the system were monitored to assess usability and satisfaction.
    - Feedback was collected to identify areas for improvement, such as interface design and response times.
- Retailer Benefits:
    - The supermarket's management accessed real-time inventory updates and sales analytics.
    - The data collected helped in optimizing stock levels and understanding customer preferences.

- The system demonstrated potential for reducing operational costs by automating inventory tracking and streamlining the checkout process.

The System Scenario outlines the intended use of the CartWave system, emphasizing the enhanced shopping experience for customers and the operational benefits for retailers. By implementing and testing the system in a supermarket case study, we were able to validate its functionalities and gather valuable insights for future deployments in various retail domains.

3.2 **Design Thinking Process**

The development of the CartWave project was guided by the Design Thinking methodology[1], which emphasizes a human-centered approach to problem-solving. This methodology allowed us to focus on understanding the needs and challenges of customers and retailers, ensuring that the final solution is both innovative and user-friendly. The process involved several key stages: framing a question, gathering inspiration, generating ideas, making ideas tangible, testing to learn, and sharing the story. Additionally, we applied the Blue Ocean Strategy to identify opportunities for value proposition in the retail domain. The following subsections illustrate the process of using design thinking during the requirement elicitation phase and all outcomes ideas that shape the final version of CartWave. We use the Miro application as a shared board to document all ideas and findings.

3.2.1 Developing the Problem Statement

We began by defining a clear problem statement, integrating elements such as the business model and the Blue Ocean Strategy[29]. Our focus was on creating a solution that centers around the user's experience rather than solely relying on technical specifications.

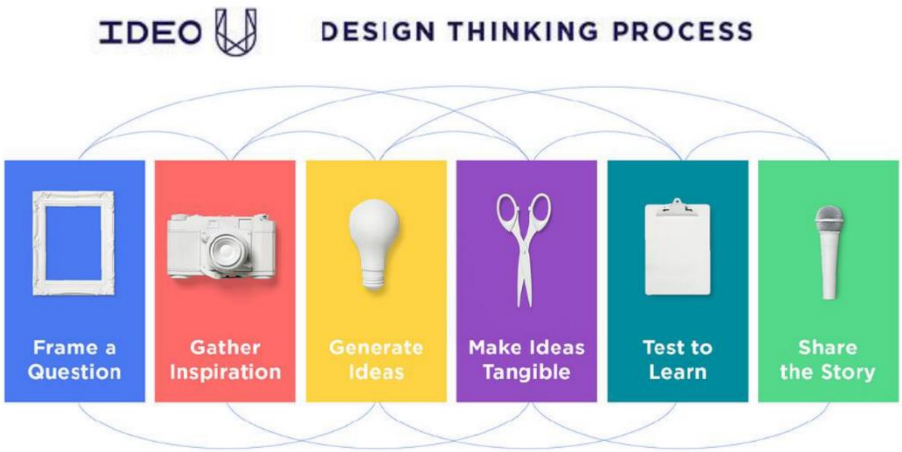

Figure 1:Design thinking process steps

3.2.2 Framing a Question

To initiate the process, we framed specific questions to guide our exploration:

- "How can we make the shopping journey more efficient and enjoyable for customers?"
- "How can we reduce shopping time and enhance convenience for shoppers?"
- "In what ways can we provide interactive assistance to improve the overall shopping experience?"

These questions helped us concentrate on core aspects of the shopping experience that could be enhanced through innovation. Figure 2 shows the finding of this phase

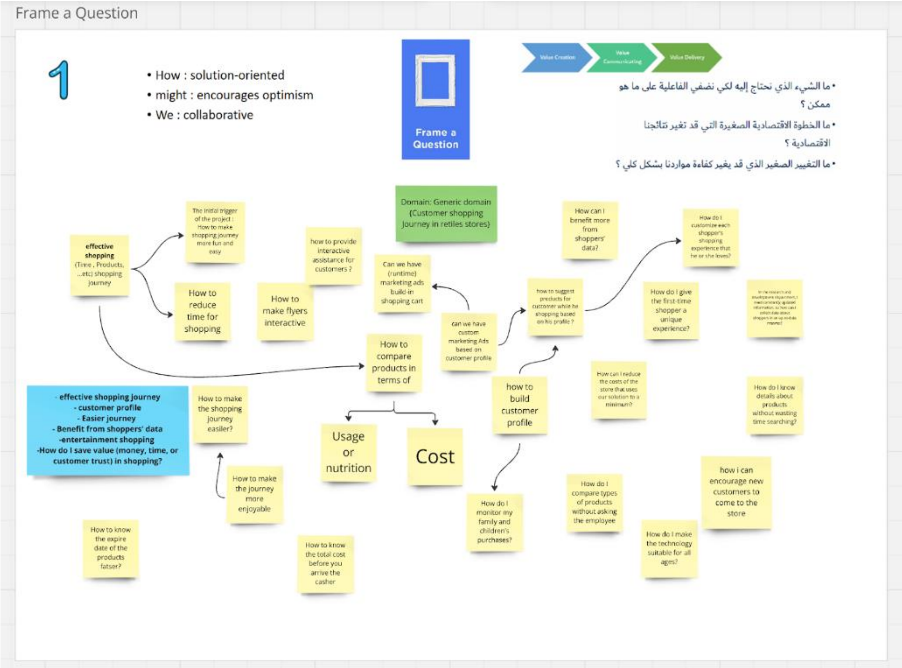

Figure 2:Framing Question.

3.2.3 Gathering Inspiration

In this stage, we sought to understand the pain points and challenges faced by customers during their shopping journey. Key issues identified included:

- Difficulty in finding detailed product information.
- Long queues at checkout counters causing delays.
- Insufficient visibility of product pricing and special offers.

We gathered inspiration from existing solutions, focusing on simplifying the customer experience through features such as easy payment options and providing accurate discount information. Figure 2 shows our finding in this step. For instance, we need to replace paper flyers with something interactive that allows customers to select the best offer based on their shopping profile. Moreover, the supermarket checkout queue is another challenge that needs to be addressed along with the number of cashiers. We notice opportunities in saving customer time as well as reducing cashiers' number by utilizing IoT technologies in automating checkout process[3].

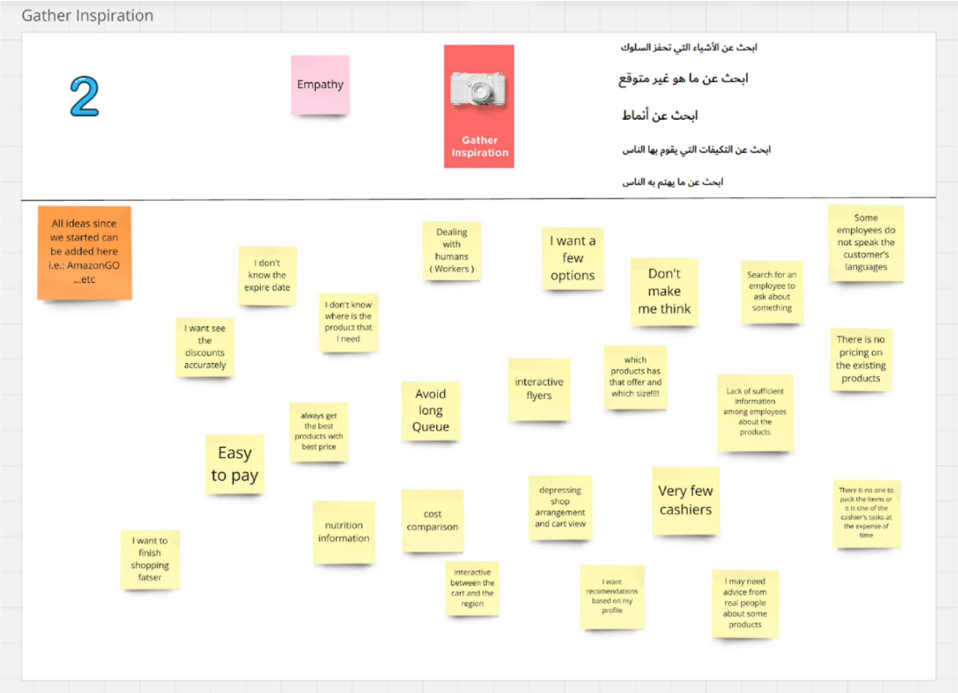

Figure 3:Gathering inspiration.

3.2.4 Generating Ideas

Building upon the insights gathered, we brainstormed a variety of ideas to address the identified challenges. Concepts generated included:

- Integrating technologies such as Radio-Frequency Identification (RFID)[30], Internet of Things (IoT) [3], and Virtual Reality (VR) to create a smart shopping cart system[31].
- Developing personalized marketing campaigns based on customer profiles and purchase history.
- Incorporating Artificial Intelligence (AI) and computer vision [4] into the shopping cart to enable features like automatic product recognition and interactive assistance.

This ideation phase allowed us to explore a wide range of possibilities before narrowing down to the most feasible and impactful solutions.

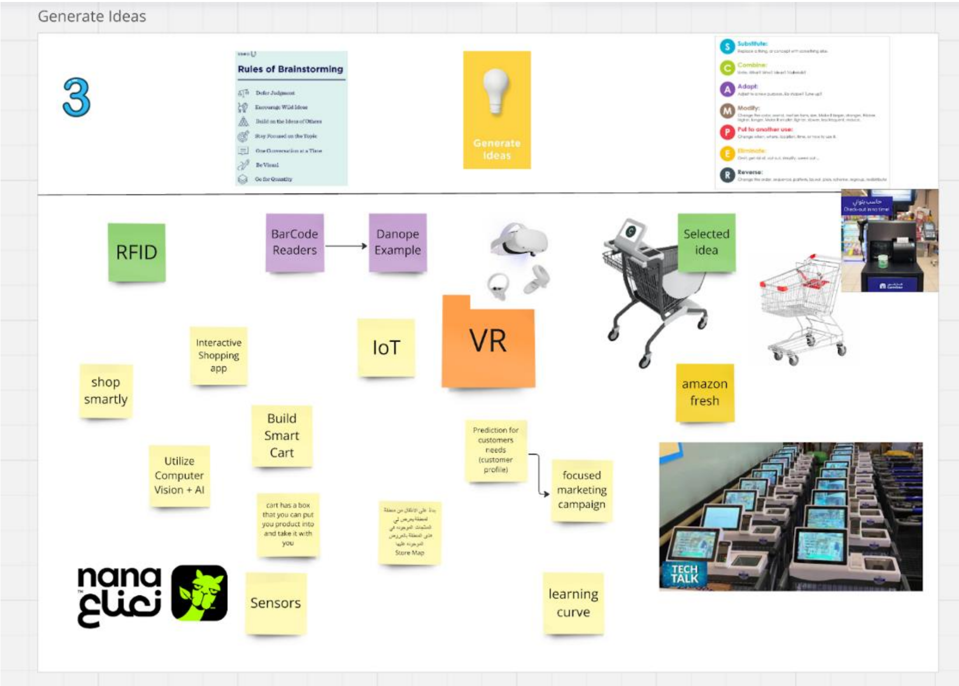

Figure 4:Generating Ideas.

3.2.5 Making Ideas Tangible

To transition from concepts to practical solutions, we focused on making our ideas tangible through prototyping:

- Hardware Selection**:** We selected appropriate hardware components, including Raspberry Pi [6], pi module camera [7], BLE sensors [5], and LCD screens, to build the smart shopping prototype.

- Software Development**:** Initial software prototypes were created to visualize the system's functionalities, including the mobile application interface and the product recognition algorithms. cart

Prototyping was essential to assess the feasibility of our ideas and to identify any technical challenges early in the development process.

3.2.6 Testing to Learn

Testing played a critical role in refining our solution:

- Functional Testing: We conducted tests on individual components, such as the BLE[5] sensors and the computer vision system[5][4], to ensure they operated correctly.

- User Acceptance Testing (UAT): Feedback was gathered from potential users to evaluate the usability and effectiveness of the system. This included assessing the intuitiveness of the user interface and the accuracy of product recognition.

- Unit Testing List: A comprehensive list of test cases was compiled to track the performance of various features and identify areas requiring improvement.

Through iterative testing, we enhanced the system's reliability and user satisfaction.

3.2.7 Sharing the Story

To communicate our journey and findings, we prepared materials for the final presentation:

- Reporting: Detailed documentation was created to outline the project's objectives, methodologies, and outcomes.

- Presentation Slides and Demonstrations: Visual aids and live demonstrations were developed to showcase the system's capabilities to stakeholders.

- Lessons Learned and Future Improvements**:** We documented insights gained throughout the project and proposed potential enhancements for future iterations of the smart shopping cart system.

3.3 **Blue Ocean Strategy**

We applied the Blue Ocean Strategy Canvas to identify opportunities for identifying value propositions[22]. The following is the key finding from applying such process.

Comparison with Traditional Shopping and Innovations: We analyzed current shopping experiences, including traditional retail and smart shopping solutions like Amazon Fresh.

- Identification of Key Competing Factors (KCFs): Factors such as product comparison, inventory management, customer service, and the use of e-payment systems were identified as critical elements influencing customer satisfaction.
- Strategic Focus: Our implementation approach aimed to eliminate unnecessary dependencies (e.g., long checkout lines, manual inventory tracking) and to maximize value innovation by offering enhanced features for both retailers and customers.

By aligning our solution with the Blue Ocean Strategy[29], we sought to differentiate CartWave from existing offerings and create a unique value proposition in the retail market . This figure presents a Blue Ocean Strategy Canvas comparing "Traditional Shopping" with a proposed "Smart Shopping" concept, likely related to an online platform like Amazon Fresh.

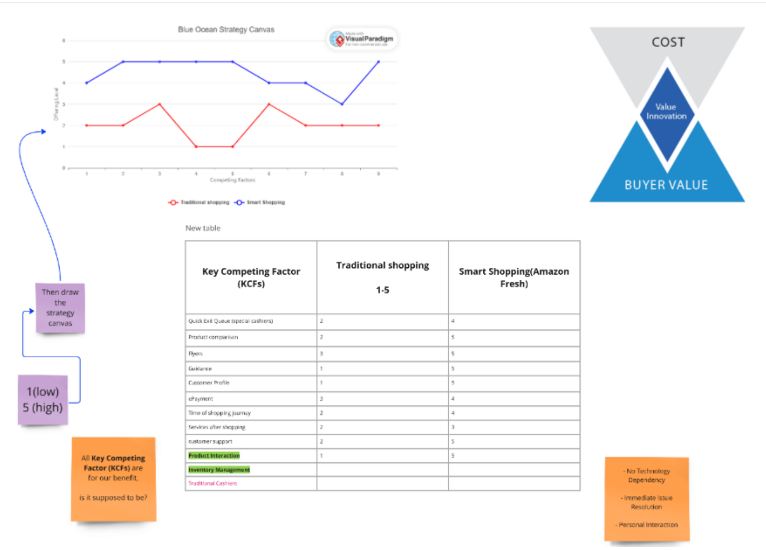

Figure 5:Blue Ocean Strategy Canvas for Smart Shopping .

This image represents a Blue Ocean Strategy Canvas, which compares key competing factors between traditional shopping and smart shopping (Amazon Fresh). The chart visually demonstrates the value differences between the two shopping methods, with smart shopping scoring higher on factors like product interaction and customer support. The table below highlights the evaluation of these competing factors, showing that smart shopping generally provides a better customer experience and innovation compared to traditional methods, especially in areas like ePayment, product comparison, and inventory management. The figure below illustrates an Eliminate-Reduce-Raise-Create (ERRC) grid, a tool used to systematically analyze and innovate upon existing industry factors. Building upon the Blue Ocean Strategy Canvas.

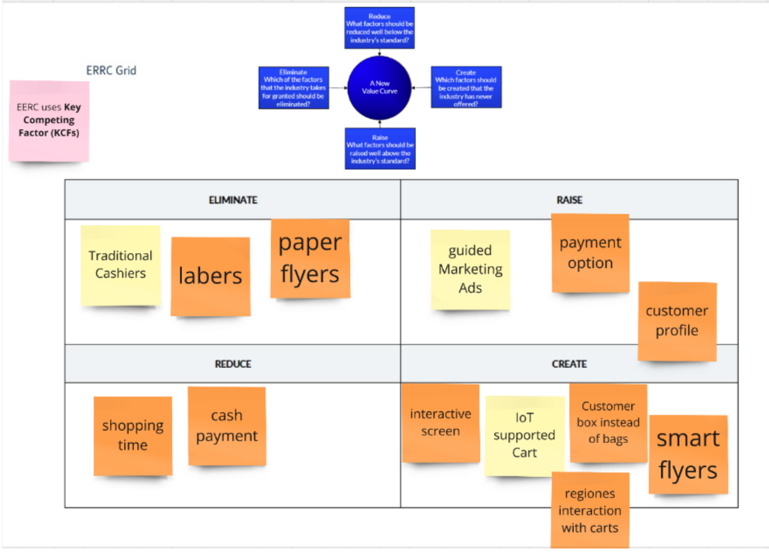

Figure 6:ERRC Grid for Smart Shopping Innovation.

This image illustrates the ERRC (Eliminate, Reduce, Raise, and Create) Grid, a tool used to develop new strategies by changing key competing factors in the industry. The grid highlights areas where traditional shopping can eliminate or reduce inefficiencies like cash payments and paper flyers while raising customer profile insights and creating IoT-supported carts. The findings suggest that by innovating through smart shopping technologies, companies can enhance customer experiences, streamline operations, and introduce value-added features like personalized marketing and smart flyers. Figure7 depicts a Value Proposition Canvas, a tool used to ensure a strong fit between a value proposition and customer needs. Building on the previous Blue Ocean Strategy and ERRC Grid analyses, this canvas details the customer profile and value map for the "Smart Shopping" concept.

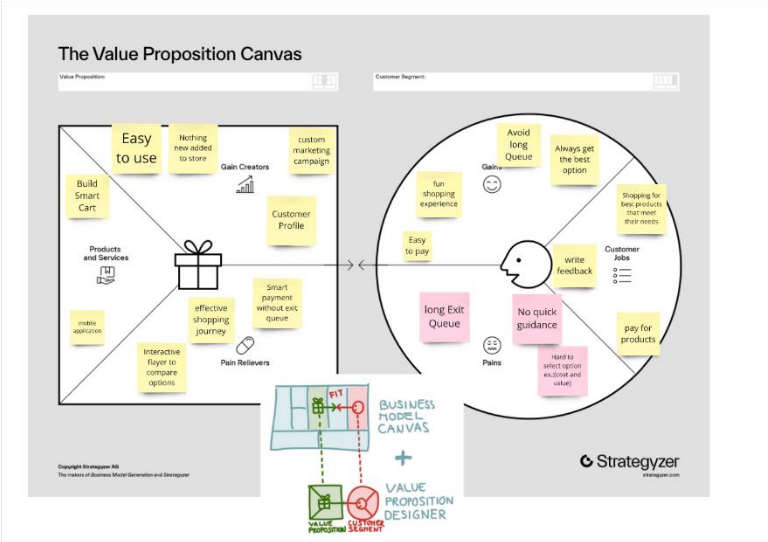

Figure 7:Value Proposition Canvas for Smart Shopping.

This image represents a Value Proposition Canvas, which connects customer needs and business offerings to ensure a better match between the product and customer expectations. On the right side, customer pains such as "long exit queue" and "no quick guidance" are listed, while gains like "easy to pay" and "fun shopping experience" are identified. On the left, the business provides solutions like "smart payment without exit queue" and "interactive product comparison," showcasing how the proposed smart cart system directly addresses customer concerns to enhance the overall shopping experience.

The Design Thinking process[1], complemented by the Blue Ocean Strategy[29], guided the development of the CartWave system from conceptualization to implementation. By maintaining a human-centered approach and focusing on innovation, we were able to design a generic solution adaptable to various retail environments. The process ensured that the system addresses real customer needs and provides tangible benefits to retailers, positioning CartWave as a forward-thinking solution in the smart retail domain.

3.4 **Requirement Specification**

3.4.1 Functional Requirements

The functional requirements specify the essential operations and activities that the CartWave system must perform to meet the needs of customers and store owners. We use UML use case modeling approach to capture core system functionality and align it with outcomes of design thinking and blue ocean as explained in the previous section of this report. These requirements are detailed through use case models that illustrate the interactions between actors and the system, ensuring that all functionalities are thoroughly documented and analyzed.

Actors:

1. Customer
    - The primary user who interacts with both the mobile application and the smart cart during the shopping process.
2. Mobile Application
    - The software platform used by the customer and the store owner/admin for various interactions, including account management and administrative tasks.
3. Smart Cart System
    - The hardware and software components integrated into the shopping cart, enabling smart functionalities like product recognition and location-based services.

4. Store Owner/Admin

   - The administrator who manages sales data, inventory, and system configurations through the mobile application.

Figure8 presents a Use Case Diagram for the "CartWave" system, visualizing the interactions between different actors (users) and the system's functionalities.

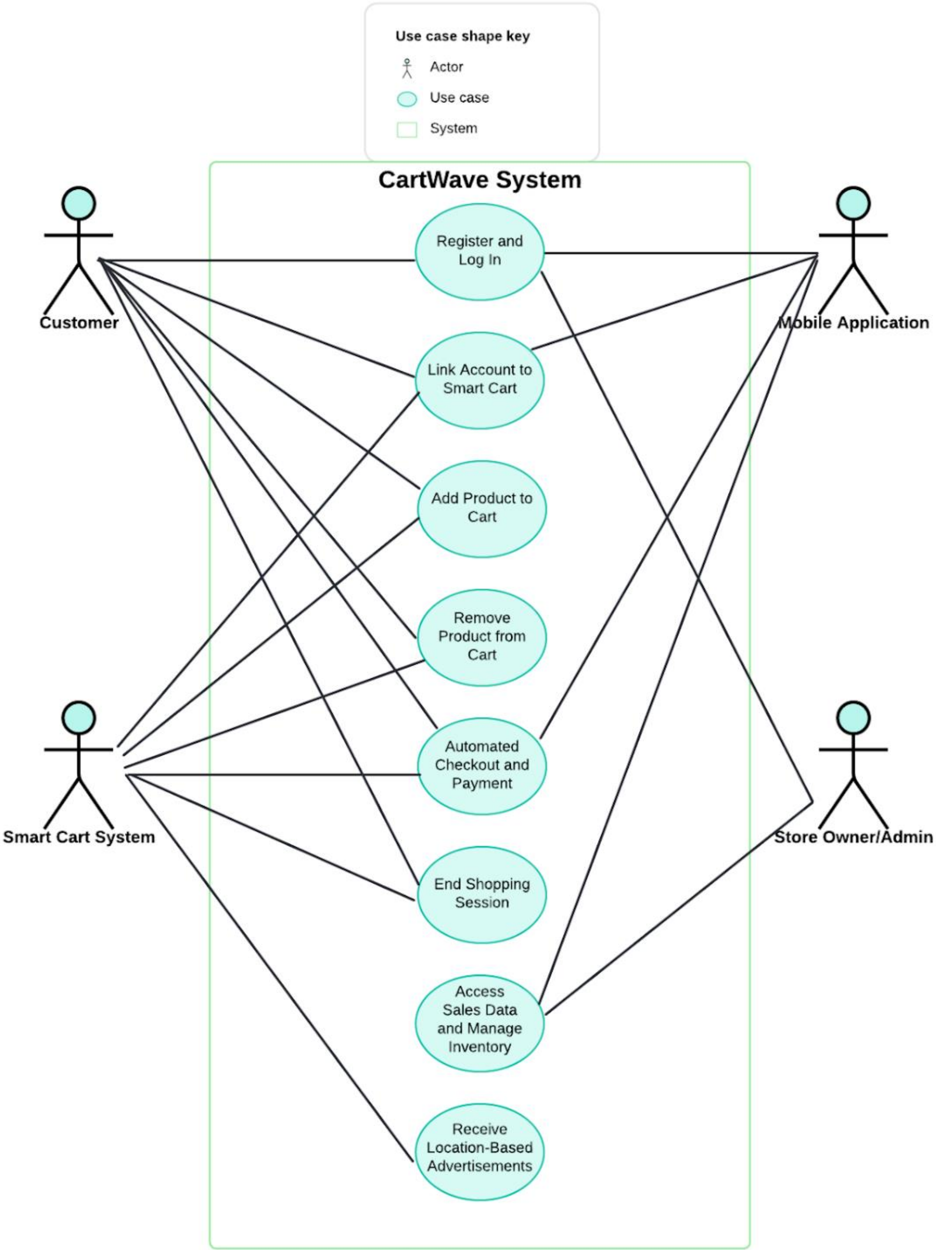

Figure 8:Use Case Diagram for the CartWave System.

This diagram represents a Use Case Diagram for the CartWave system, showing interactions between the system and its actors (Customer, Mobile Application, Smart Cart System, Store Owner/Admin). Each use case, such as "Register and Log In" or "Automated Checkout and Payment," is represented in an oval and shows how the actors interact with the system to perform specific tasks. The connections highlight the multiple ways that the system supports customer interactions, inventory management, and smart shopping functionalities, ensuring a seamless experience between different parts of the system and user needs

Core Use Case Specifications**:**

Below are the detailed use cases, presented in a structured format for clarity.

Use Case 1: User Registration and Login

- **Actor(s):** Customer, Mobile Application
- **Goal:** To register for an account and log in to access the mobile application.
- **Preconditions:** The customer has downloaded the CartWave mobile application.
- **Postconditions:** The customer is registered and/or logged in to the mobile application.

Main Success Scenario:

Step	Actor Action	System Response
1	Customer launches the mo-bile application.	System displays the welcome screen with "Log In" and "Sign Up" options
2	Customer selects "Sign Up" to create a new account.	System prompts for registration details (name, email, password).
3	Customer enters the re-quired information and submits the form.	System validates the input data and creates a new user ac-count.
4		System logs the customer into the app and displays the home screen.

Table 2:Use Case 1: User Registration and Login.

Alternate Scenario: Existing User Logs In

Step	Actor Action	System Response
1	Customer selects "Log In" on the welcome screen.	System prompts for login credentials (email, password).
2	Customer enters credentials and submits	System verifies credentials.
3		If valid, system logs the customer in and displays the home screen.
4		If invalid, system prompts for re-entry without a success message.

Table 3:Use Case 1:Alternate Scenario.

Use Case 2: Link Account to Smart Cart

- **Actor(s):** Customer, Mobile Application, Smart Cart System
- **Goal:** To connect the customer's mobile app account with the smart cart for a personalized shopping experience.
- **Preconditions:** The customer is logged in to the mobile application.
- **Postconditions:** The smart cart is paired with the customer's account.

Main Success Scenario:

Step	Actor Action	System Response
1	Customer presses the bar-code button in the mobile app.	Mobile application displays a unique barcode for pairing.
2	Customer places the mo-bile app screen in front of the cart's camera.	Smart Cart's camera captures the barcode from the mobile app.
3		System establishes a connec-tion between the mobile app and the smart cart.
4		Smart Cart's screen indicates that pairing is complete.

Table 4:Use Case 2: Link Account to Smart Cart.

Use Case 3: Add Product to Cart

- **Actor(s):** Customer, Smart Cart System
- **Goal:** To add a product to the customer's digital shopping cart using automatic product recognition.
- **Preconditions:** The customer's account is linked to the smart cart.
- **Postconditions:** The product is added to the digital shopping cart, and the total cost is updated.

Main Success Scenario:

Step	Actor Action	System Response
1	Customer selects a product from the shelf.	
2	Customer places the product's in front of the cart's camera.	Smart Cart's camera captures the product's image.
3		System recognizes the product using the machine learning model.
4		Product details (name, price,etc) are displayed on the cart's screen.
5		Product is added to the digital shopping cart; total cost is updated.
6	Customer places the product into the physical cart.	

Table 5:Use Case 3: Add Product to Cart.

Use Case 4: Remove Product from Cart

- **Actor(s):** Customer, Smart Cart System
- **Goal:** To remove a product from the customer's digital shopping cart.
- **Preconditions:** The product has been previously added to the digital shopping cart.

- **Postconditions:** The product is removed from the digital shopping cart, and the total cost is updated.

Main Success Scenario:

Step	Actor Action	System Response
1	Customer accesses the shopping list on the cart's screen.	System displays the list of products in the digital shopping cart.
2	Customer selects the product to be removed.	
3	Customer presses the "Remove" option next to the product	System removes the product from the digital shopping cart immediately.
4		Total cost is updated on the cart's screen.
5	Customer removes the physical product from the cart and returns it to the shelf if desired.	

Table 6:Use Case 4: Remove Product from Cart.

Use Case 5: Receive Location-Based Advertisements

- **Actor(s):** Customer, Smart Cart System
- **Goal:** To receive personalized advertisements and promotions based on the cart's location within the store.
- **Preconditions:** The customers enter a specific location detected by the beacon.

- **Postconditions:** The customer is informed of promotions relevant to their current location in the store.

Main Success Scenario:

Step	Actor Action	System Response
1	Customer moves to a different section of the store.	Smart Cart's System scan for BLE beacon of the current zone.
2	Customer arrived in specific section	Smart Cart's System detect the BLE beacon of the current zone.
3		System identifies the cart's current location within the store.
4		Cart's screen displays targeted advertisements and promotions relevant to the location.

Table 7:Use Case 5: Receive Location-Based Advertisements.

Use Case 6: Automated Checkout and Payment

- **Actor(s):** Customer, Smart Cart System, Mobile Application
- **Goal:** To complete the shopping session and process payment **automatically.**
- **Preconditions:** The customer has finished adding products and is near the designated payment zone.
- **Postconditions:** Payment is processed, a digital receipt is issued, and the shopping session is concluded.

Main Success Scenario:

Step	Actor Action	System Response
1	Customer approaches the check-out area.	
2		Smart Cart's System detect entry into the designated payment zone [5].
3		System activates the checkout process.
4		Cart's screen displays the total amount and prompts for confirmation.
5	The customer confirms the purchase on the shopping cart screen by pressing the payment button.	System shows payment methods to customer to choose
6	The customer chooses to pay from card or wallet.	System processes the payment using the customer's linked payment method.
7		Cart's screen displays a success message indicating payment completion.
8		Digital receipt is sent to the customer's mobile application.
9		Inventory levels are updated in real time.
10	Customer exits the store with their purchased items.	

Table 8:Use Case 6: Automated Checkout and Payment.

Use Case 7: End Shopping Session

- **Actor(s):** Customer, Smart Cart System
- **Goal:** To conclude the shopping session and reset the smart cart for the next user.
- **Preconditions:** Payment has been successfully processed.
- **Postconditions:** The cart is reset and ready for use by another customer.

Main Success Scenario:

Step	Actor Action	System Response
1	After checkout, customer confirms the end of the shopping session on the cart's screen if prompted.	
2		System clears the customer's data from the cart.
3		Cart resets to the initial state, displaying the default welcome screen.
4		No personal data remains on the cart.
5	Customer leaves the cart at the designated area for collection.	

Table 9:Use Case 7: End Shopping Session.

Use Case 8: Admin Access to Sales Data and Inventory Management

- **Actor(s):** Store Owner/Admin, Mobile Application
- **Goal:** To monitor real-time sales data, view invoices, and manage inventory levels effectively.
- **Preconditions:** The admin has the CartWave mobile application and has logged in using admin credentials.
- **Postconditions:** the admin has updated insights into sales performance and inventory status; inventory can be updated as needed.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin launches the CartWave mobile application.	
2	Admin selects "Log In" and enters admin email and password.	System verifies admin credentials.
3		Upon successful login, system displays the admin interface.
4	Admin navigates to "Sales Data" or "Invoices" section.	System displays a list of all invoices issued, including details of each transaction (customer ID, products purchased, totals).
5	Admin reviews transaction details as needed.	
6	Admin navigates to "Inventory Management" section.	System displays current stock levels of all products.
7	Admin updates inventory quantities or adds new stock as necessary.	System saves changes and updates inventory data in real time.

Table 10:Use Case 8: Admin Access to Sales Data and Inventory Management.

3.4.2 Non-Functional Requirements

Non-functional requirements define the quality attributes, performance criteria, and constraints that the CartWave system must meet to ensure a satisfactory user experience and efficient operation. These requirements are crucial for the system's overall success and user acceptance.

1. Usability
   - Ease of Use: The user interfaces on both the smart cart and the mobile application must be intuitive and straightforward, allowing users of varying technical proficiency to navigate the system without difficulty.
   - Consistency: Design elements should be consistent across the mobile app and smart cart interfaces to reduce learning time and prevent user errors.
   - Accessibility: Font sizes, colors, and button sizes should be designed for easy reading and interaction.
2. Performance
   - Response Time: System responses, including product recognition and screen updates, should occur within 4-5 seconds to maintain a seamless shopping experience.
   - Throughput: The system should handle multiple simultaneous users without degradation in performance, supporting peak shopping times.

3. Reliability

   - Availability: The system should be operational during store hours with minimal downtime. Scheduled maintenance should occur during off-hours.

   - Accuracy: Product recognition accuracy should be maintained at a high level, with continuous improvements to the machine learning model to reduce errors.

   - Data Integrity: All transactions and inventory updates must be accurately recorded without loss or corruption of data.

4. Scalability

   - Horizontal Scalability: The system architecture should support the addition of more smart carts and increased user load without significant redesign.

   - Adaptability: The system should be adaptable to different store sizes and types, allowing for customization based on specific retailer needs.

5. Compatibility

   - Platform Compatibility: The mobile application must be compatible with major mobile operating systems, including iOS and Android.

   - Integration: The system should integrate smoothly with existing store systems where applicable, such as inventory management and point-of-sale systems.

6. User Experience

- Aesthetics: The visual design should be professional and appealing to enhance user engagement.

The non-functional requirements are essential to ensure that the CartWave system not only performs its intended functions but also delivers a high-quality experience to users and operates efficiently within the retail environment. By adhering to these requirements, the system will meet the expectations of both customers and store owners, contributing to its success and adoption.

CHAPTER 4

System Analysis and Design

4.1 **System Analysis**

In this part, we delve into the structural and behavioral aspects of the CartWave system by identifying the core system classes and modeling their interactions. The analysis includes a Unified Modeling Language (UML) class diagram to represent the main classes and their relationships[32], as well as a UML sequence diagram to illustrate the dynamic flow of interactions between objects during a typical use case.

4.1.1 Class Diagram

The UML class diagram provides a static view of the CartWave system by showcasing the core classes, their attributes, methods, and the relationships between them. This diagram serves as a blueprint for the system's architecture, aiding in understanding how different components interact and depend on one another. During prototyping implementation, we follow the recommendations provided by the tools and we use this model base for building system components on selected low code framework [12].Figure 9 presents a Class Diagram for the "CartWave" system, providing a static view of the system's structure and the relationships between different classes.

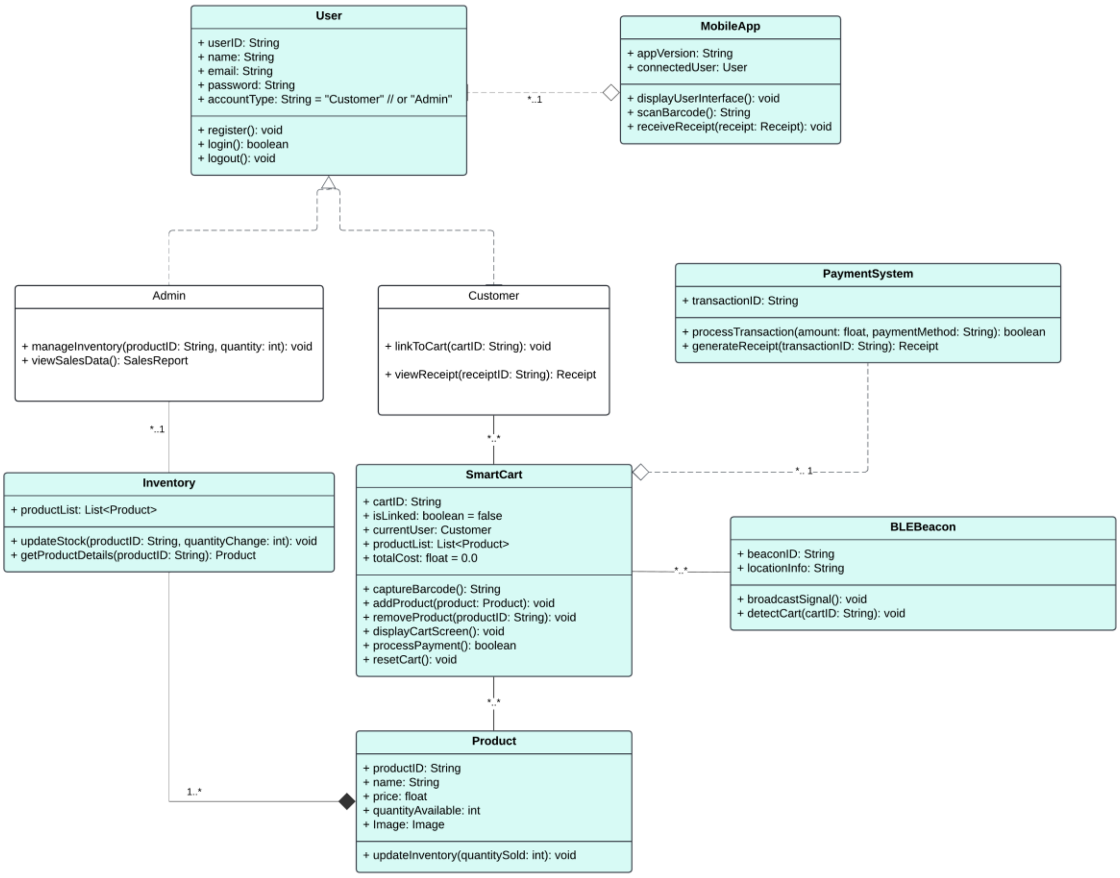

Figure 9:Class Diagram for the CartWave System.

4.1.2 Interaction Diagram[32]

The UML sequence diagram models the dynamic flow of interactions between objects in the CartWave system during a specific use case. It focuses on the sequence of messages exchanged between objects *to accomplish a particular functionality, providing insight into the system's behavior over time.* Figure 10 *shows the main sequence diagram detailing the interactions involved in adding a product to a shopping cart within the CartWave system. It illustrates the step-by-step flow of messages between the user, mobile application, smart cart system, and inventory.*

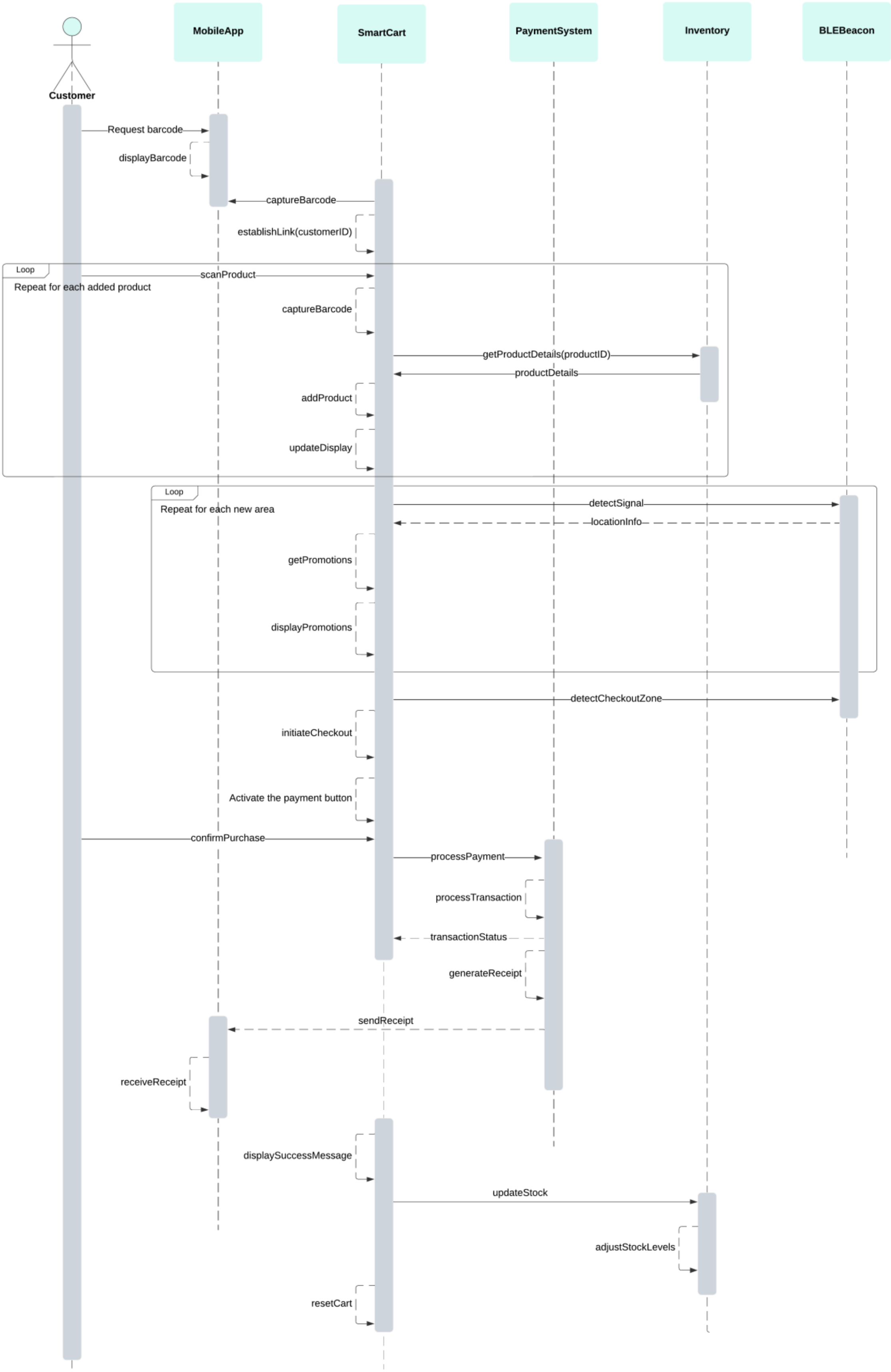

Figure 10:Sequence Diagram for CartWave System.

4.1.3 Database Architecture

In this section, we present an overview of the system's architecture, focusing on the database structure and the subsystem decomposition. These architectural models provide a high-level understanding of how the CartWave system is organized and how its components interact.

The CartWave system utilizes Firebase Firestore[16], a cloud-based NoSQL database, to efficiently store and manage data. Firestore's flexible, scalable document model is well-suited for applications requiring real-time updates and offline support.

The database is structured into three main collections:

1. Users
2. Products
3. Transactions

Each collection contains documents that represent entities within the system. This organization allows for straightforward data retrieval and management, facilitating seamless interactions between users and the system. As shown in figure 11 illustrates the proposed database architecture for the CartWave project, outlining the organization of data into various tables and their relationships. The diagram employs a notation that appears to be inspired by Entity-Relationship Diagrams (ERDs) with some modifications.

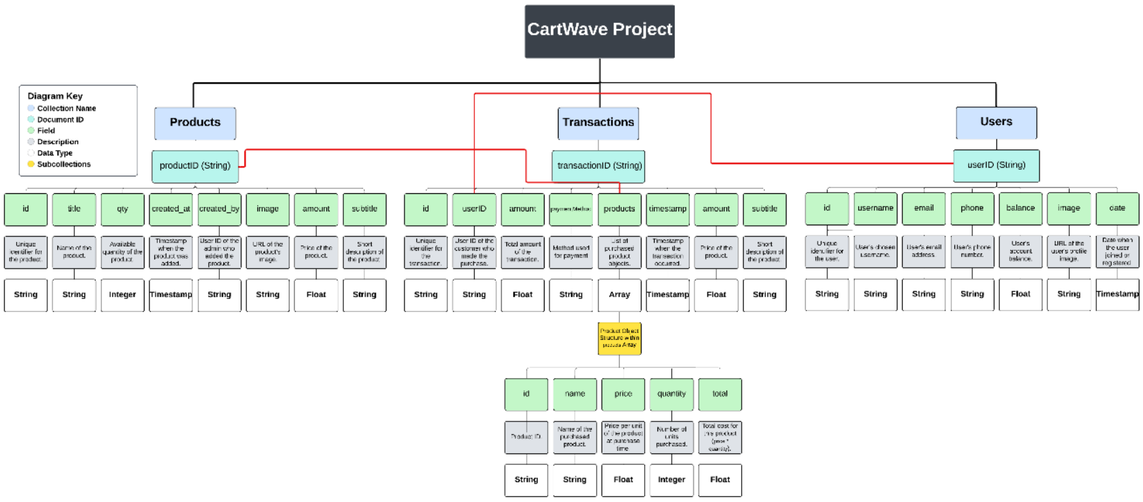

Figure 11:Database Architecture for the CartWave Project.

4.3 **Business Process Models**

In this section, we present the core business process models of the CartWave system using Business Process Model and Notation (BPMN)[32]. These models aim to represent the main scenarios in a clear visual manner that reflects how users interact with the system and illustrates the process flows smoothly. By using these models, we strive to highlight the key steps in each process, making them easier to understand and contributing to the enhancement and development of the system.

Use Case 1: User Registration and Linking Account to Smart Cart

This diagram illustrates the process by which a customer registers for an account using the mobile application and then links their account to a smart cart. It demonstrates how the system enables new users to create an account and pair their mobile app with a smart cart for a personalized shopping experience.

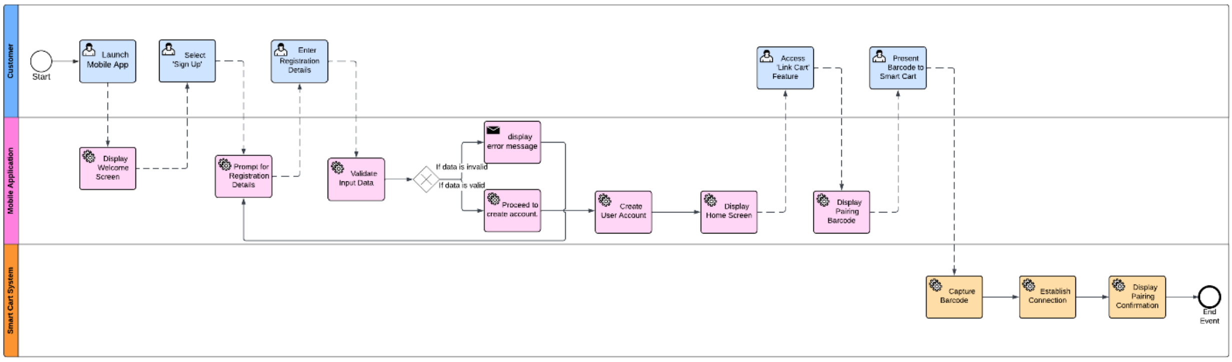

Figure 12:BPMN Diagram for User Registration and Linking Account to Smart Cart.

By completing this process, the customer successfully registers and is ready to interact with the smart cart system, allowing for a seamless and personalized shopping experience.

Use Case 2: Adding and Removing Product from Cart

This diagram depicts the steps involved when a customer adds products to their digital shopping cart using the smart cart's product recognition feature, as well as how they can remove products from the cart. It shows the interaction between the customer and the smart cart system during the shopping process. Through this process, customers can effectively manage their shopping cart, with real-time updates to product selections and total cost, enhancing convenience and efficiency during shopping.

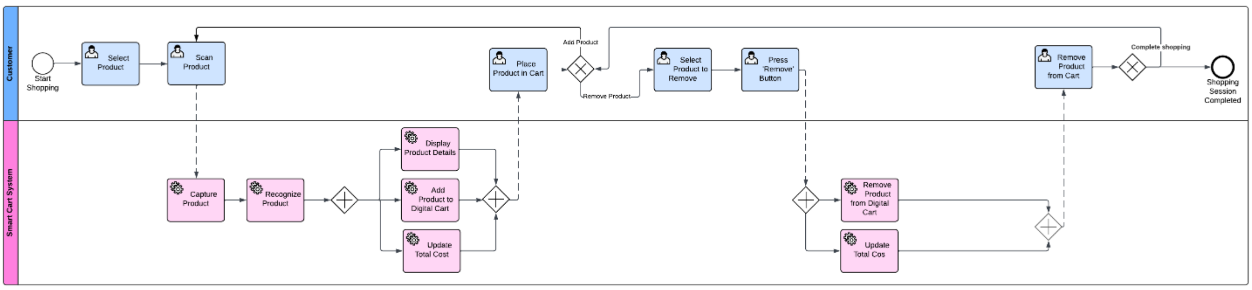

Figure 13:BPMN Diagram for Adding and Removing Products from Cart.

Use Case 3: Receive Location-Based Advertisements

This diagram illustrates how the smart cart system detects the customer's location within the store using BLE beacons and provides personalized advertisements based on that location[5]. It highlights how the system enhances customer engagement by delivering relevant promotions as they move through different areas of the store.

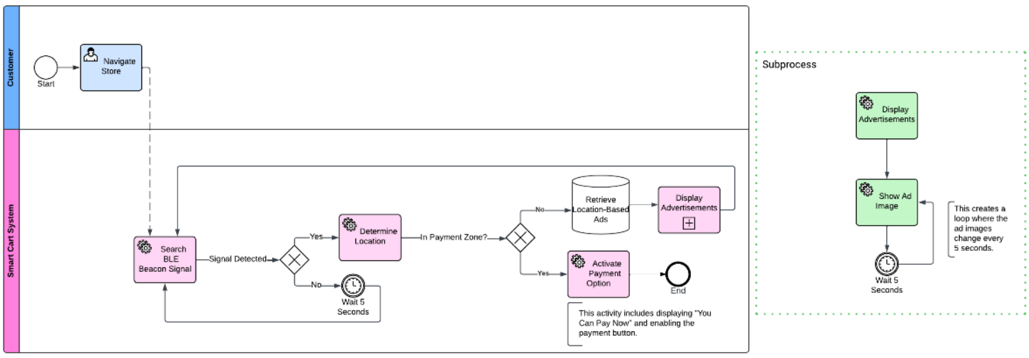

Figure 14:BPMN Diagram for Receiving Location-Based.

By receiving location-based advertisements, customers are informed of promotions relevant to their current location, which can enhance their shopping experience and influence purchasing decisions.

Use Case 4: Automated Checkout and Payment

This diagram outlines the automated checkout and payment process that occurs when a customer completes their shopping and approaches the designated

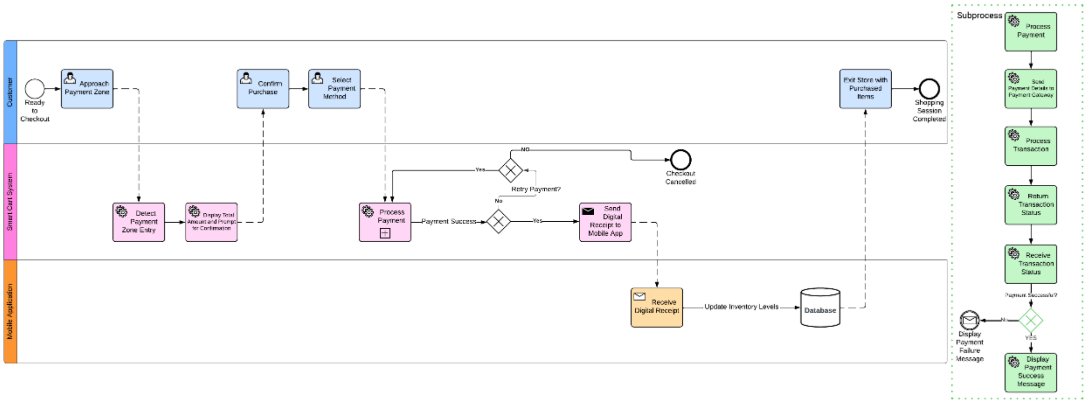

Figure 15:BPMN Diagram for Automated Checkout and Payment.

payment zone. It details the steps taken by the smart cart system to process payment and finalize the transaction.

The successful completion of this process results in an efficient checkout experience, where payment is processed automatically, and the customer can exit the store with their purchased items without delay.

CHAPTER 5

IMPLEMENTATION

5.1 Case study description

- Housing: Designed and fabricated enclosure this chapter, we delve into the implementation of the CartWave system within the retail supermarket domain. The CartWave project is designed to revolutionize the traditional shopping experience by integrating smart technology into shopping carts, thereby providing customers with an interactive and seamless way to shop.

The selected domain for this case study is the retail supermarket industry, a sector characterized by large product assortments and high customer footfall. Supermarkets present an ideal environment for implementing CartWave due to the need for efficient shopping experiences and the potential for enhancing customer engagement through technology.

The CartWave system leverages a combination of hardware and software components:

- Smart Carts equipped with cameras and BLE (Bluetooth Low Energy) technology[5].
- A Mobile Application that customers use for account management and pairing with smart carts.
- A Cloud-Based Database (Firebase Firestore) for real-time data storage and retrieval[16].

By integrating these components, CartWave enables several key functionalities:

- Automatic Product Recognition: Customers can add products to their digital cart simply by scanning them with the smart cart's camera[4].
- Location-Based Advertisements: BLE technology allows the system to detect the cart's location within the store and display relevant promotions.
- Automated Checkout and Payment: The system facilitates a seamless checkout process by processing payments automatically when customers enter the designated payment zone.

The implementation in the supermarket domain addresses common challenges such as long checkout lines, lack of personalized shopping experiences, and inefficiencies in inventory management.

5.2 **Hardware Implementation**

In this section, we discuss the hardware components utilized in the development of the CartWave system and how they were integrated to create a functional smart cart.

5.2.1 Hardware Components

The primary hardware components used in the CartWave system include:

- Raspberry Pi 4 Model B[6]: Serves as the main processing unit for the smart cart.

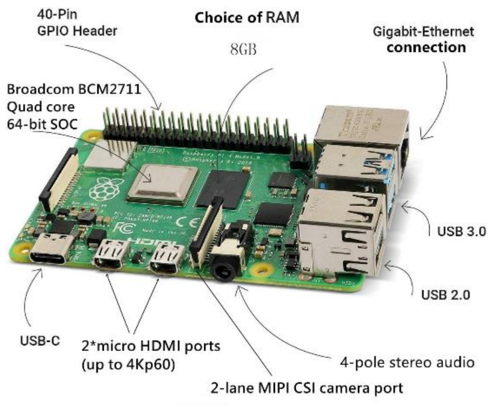

Figure 16:Raspberry Pi 4 Model B Microcontroller.

- Pi Camera Module[7]: Enables image capture for product recognition.

Figure 17:Raspberry Pi Camera Module.

- Bluetooth Low Energy (BLE) Beacons[5]: Used for indoor positioning to detect the cart's location within the store.

Figure 18:ESP32-C3 BLE [5]Beacon.

- Touchscreen Display: Provides an interactive interface for customers.

Figure 19 10.1Touchscreen Display for Raspberry Pi.

- Power Supply Unit: Supplies power to all hardware components.

- Cart Structure: A modified shopping cart designed to house all electronic components securely.

5.2.2 Integration of Hardware Components

Raspberry Pi 4 Model B[6]:

The Raspberry Pi 4 was selected for its balance of performance and cost-effectiveness. It handles image processing tasks and communicates with the mobile application and cloud services.

- Setup: Installed Raspbian OS and configured the necessary drivers.

- Connectivity: Enabled Wi-Fi and Bluetooth for communication with the mobile application and BLE beacons[5].

Pi Camera Module[7]:

The Pi Camera Module is attached to the Raspberry Pi to capture images of products placed in front of it.

- Mounting: Secured on the cart at an angle optimal for scanning product images.
- Configuration: Calibrated for appropriate focus and exposure settings to ensure clear image capture.

BLE Beacons[5]:

BLE beacons are strategically placed throughout the store to facilitate indoor positioning.

- Deployment: Installed at various locations within the store to create a grid for location detection.
- Calibration: Adjusted signal strength and intervals to optimize accuracy and reduce interference.

Touchscreen Display:

A touchscreen display is connected to the Raspberry Pi to serve as the user interface on the smart cart.

- Installation: Mounted securely on the cart handle for easy customer access.
- Configuration: Set up to display the graphical user interface developed for the smart cart.

Power Supply Unit:

A reliable power supply is essential to ensure uninterrupted operation.

- Battery Selection: Chose a rechargeable lithium-ion battery with sufficient capacity to power all components for the duration of a shopping session.
- Safety Measures: Implemented voltage regulation and overcharge protection circuits.

Cart Structure Modifications:

Modifications were made to a standard shopping cart to accommodate the hardware components.

- es to protect electronic components from damage and tampering.
- Cable Management: Organized wiring to prevent tangling and maintain a neat appearance.

5.3 **Software Development**

This section outlines the software development aspects of the CartWave system, including the tools and technologies used.

5.3.1 Technology Stack

The software components were developed using the following technologies:

- Programming Languages:

- *Python[10]*: Used for developing applications on the Raspberry Pi, including the trained machine learning model and the Bluetooth communication system.
- *Dart[12]*: For mobile application development using Flutter.
- *PHP, HTML, CSS, JavaScript[13][14][15]*: For developing the cart application interface displayed on the touchscreen.

- Frameworks and Libraries:
    - *Teachable Machine[8]*: Used to train machine learning models on product images.
    - *Flutter [12]*: For cross-platform mobile app development, ensuring compatibility with both iOS and Android platforms.
    - *Firebase SDKs[33]* : For authentication, real-time database operations, and cloud functions.

5.3.2 Development Tools

During implementation we use set of Integrated Development Environments (IDEs) such as :

- Visual Studio[34]: Served as the primary IDE for developing the cart application and other components.
- Arduino IDE[11]: Used for programming the Bluetooth modules and related hardware components.

- Android Studio: Utilized for testing the mobile application, including emulation and debugging.

- Raspberry Pi OS (Debian)[9]: The operating system used on the Raspberry Pi, providing a stable environment for development in Python.

Testing Tools

- Visual Studio Debugger: Used for debugging and testing the cart application and other codebases.

- Flutter Testing Tools: Employed for unit and widget testing within the Flutter framework.

- Android Emulator: Provided by Android Studio, used to test the mobile application on virtual devices.

5.3.3 Software Components

Developing Mobile Application was one of core task in this project. We develop it using Flutter and Dart to create a cross-platform application compatible with both iOS and Android devices[12]. This platform has set of features such as :

- User registration and authentication.

- Account management.

- Display of digital receipts and transaction history.

- Barcode generation for pairing with the smart cart.

- Integration with Firebase:
    - Implemented Firebase Authentication for secure login and sign-up processes.
    - Used Firebase Firestore for storing user data, transaction records, and syncing information with the smart cart.

Another core application in this project is Cart Application which was

Developed using PHP, HTML, CSS, and JavaScript, the cart application runs on the smart cart's touchscreen display[13][14][15].

- Features:
    - Interactive user interface for displaying product information, cart contents, and promotional materials.
    - Facilitates user interactions such as adding or removing products, and initiating payment processes.
    - Communicates with the backend systems to update cart status and synchronize with the mobile application.

The Smart Cart Software transform normal shopping cart to interactive cart since it , runs on the Raspberry Pi, utilizing Python for various functionalities[10]. The following shows key features:

- Operating System:
    - Raspberry Pi OS (Debian)[9]: Provides a Linux-based environment optimized for the Raspberry Pi hardware.

- Product Recognition Module:
  - Teachable Machine: Used to train machine learning models on product images[8].
  - The trained model is deployed on the Raspberry Pi to recognize products through images captured by the Pi Camera[7].
  - Python scripts handle image processing and interaction with the machine learning model.
- Bluetooth Communication:
  - Python and Arduino IDE[10][11]: Used to program the Bluetooth modules.
  - Facilitates communication between the smart cart and other devices, including BLE[5] modules for location detection and the customer's mobile application.
- User Interface:
  - Developed using web technologies (PHP, HTML, CSS, JavaScript[13][14][15]) displayed on the cart's touchscreen.
  - Provides an interactive experience for customers, allowing them to view cart contents, receive location-based advertisements, and proceed to checkout.
- Cloud Services and Database
  - Firebase Firestore: Serves as the cloud-hosted NoSQL database for storing and synchronizing data in real-time[16].

- Firebase Authentication: Manages user authentication securely.
- Firebase Cloud Functions: Used for backend logic, such as processing payments and updating inventory levels.

5.3.4 Challenges and Solutions

During implementation we face a set of challenges. The following points summarize these challenges and our proposed solutions

- Machine Learning Model Deployment:
  - *Challenge*: Integrating the model trained with Teachable Machine into the Raspberry Pi environment[8].
  - *Solution*: Exported the model in a compatible format and used Python libraries to run the model locally on the Raspberry Pi.
- Bluetooth Communication:
  - *Challenge*: Ensuring reliable communication between the smart cart and BLE[5]modules or other devices.
  - *Solution*: Used the Arduino IDE[11] to program the Bluetooth modules effectively, and implemented error-handling routines in Python.
- Cross-Platform Mobile Application Development:
  - *Challenge*: Delivering a consistent user experience across both iOS and Android devices.

- *Solution*: Utilized Flutter's cross-platform capabilities and thoroughly tested the application using Android Studio's emulator and, where possible, on actual devices.

- Real-Time Synchronization:
    - *Challenge*: Maintaining real-time data synchronization between the mobile application, cart application, and cloud database.
    - *Solution*: Leveraged Firebase's real-time database features to ensure data consistency and implemented mechanisms to handle conflicts[16].
- User Interface Development for Cart Application:
    - *Challenge*: Creating an intuitive and responsive interface on the cart's touchscreen display.
    - *Solution*: Developed the interface using web technologies (PHP, HTML, CSS, JavaScript[13][14][15]) optimized for the hardware specifications of the touchscreen display.
- Development Environment Consistency:
    - *Challenge*: Managing different development environments and tools.
    - *Solution*: Standardized on Visual Studio as the primary IDE where possible and ensured that team members had consistent setups[34].

5.4 **Main Algorithms**

In this section, we describe the main algorithms implemented in the CartWave system, focusing on the key components that enable product recognition and location detection. We provide an overview of how these algorithms function and how they can be extended to accommodate additional products or zones.

5.4.1 Product Recognition Algorithm

The product recognition algorithm enables the smart cart to identify products placed in front of its camera. This is achieved using a machine learning model trained to recognize specific product images[4]. The algorithm involves capturing an image of the product, processing it, and then classifying it using the trained model.

To train models we utilized Teachable Machine[8], a web-based tool by Google, to train a convolutional neural network (CNN) model on images of the products. The steps involved in training the model are:

1. Data Collection: Gather multiple images of each product from different angles and under various lighting conditions to enhance the model's robustness.

2. Model Training: Upload the images to Teachable Machine, label them accordingly, and train the model using its built-in training pipeline.

3. Exporting the Model: After training, export the model in a TensorFlow Lite format compatible with deployment on the Raspberry Pi.

The trained model is deployed on the Raspberry Pi, where it runs locally to perform real-time product recognition.

- Programming Language: Python[10].

- Libraries Used:
    - TensorFlow Lite: For running the machine learning model[35].
    - Keras: For handling the model structure.
    - Pillow (PIL): For image processing.
    - NumPy: For numerical computations.

Algorithm high level execution steps are summarized in the following

1. Image Capture: When a product is placed in front of the camera, an image is captured using the Pi Camera module.
2. Preprocessing:
    - The image is converted to RGB format.
    - Resized to the input shape required by the model (e.g., 224x224 pixels).
    - Normalized to match the data distribution the model was trained on.
3. Prediction:
    - The preprocessed image is fed into the model.
    - The model outputs probabilities for each class (product).
    - The class with the highest probability above a certain threshold is selected as the prediction.
4. Result Handling:
    - If a product is recognized, it is added to the digital cart.

- The system avoids duplicate detections by keeping track of the last recognized product.

Figure20 presents a pseudocode example for an algorithm designed to recognize products within the CartWave system.

Pseudocode Example:

```
# Load the trained model
model = load_model('trained_model.h5')
class_names = ['ProductA', 'ProductB', 'ProductC']  # Example product classes
threshold = 0.7  # Confidence threshold

def recognize_product(image):
    # Preprocess the image
    image = preprocess_image(image)

    # Expand dimensions to match model input
    image_array = np.expand_dims(image, axis=0)

    # Make prediction
    predictions = model.predict(image_array)
    confidence_scores = predictions[0]

    # Get the class with the highest probability
    class_index = np.argmax(confidence_scores)
    confidence_score = confidence_scores[class_index]

    # Check if confidence is above threshold
    if confidence_score >= threshold:
        class_name = class_names[class_index]
        return class_name, confidence_score
    else:
        return None, None
```

Figure 20:Product Recognition Algorithm Pseudocode Example.

To add a new product to the recognition system:

1. Collect Images: Capture multiple images of the new product from various angles.
2. Retrain the Model:
    - Use Teachable Machine to include the new product images[8].
    - Retrain the model with the expanded dataset.
3. Deploy the Updated Model:
    - Export the updated model.

    - Replace the existing model on the Raspberry Pi with the new one.

This approach ensures the system can be continuously updated with new products as needed.

5.4.2 Location Detection Algorithm Using Bluetooth Low Energy (BLE)[5]

The location detection algorithm utilizes Bluetooth Low Energy (BLE)[5] technology to determine the smart cart's zone within the store. BLE modules placed in different zones broadcast unique identifiers, which the smart cart detects to ascertain its current location.

Implementation:

- Programming Languages:
    - Python for the scanning script[10].
    - Arduino IDE[11] for programming BLE modules.
- Libraries Used:
    - Bleak: An asynchronous Python library for BLE device scanning.
    - Asyncio: For handling asynchronous operations.

Algorithm Steps:

1. Scanning for BLE Devices:
    - The smart cart continuously scans for nearby BLE devices.
    - Scanning is performed asynchronously to avoid blocking other operations.
2. Device Identification:

- Each BLE module broadcasts a unique UUID corresponding to a specific zone (e.g., Zone A, Zone B).

3. Zone Determination:
   - Upon detecting a BLE module, the algorithm checks the UUID against known zone identifiers.
   - The latest detected zone is stored as the cart's current location.
4. Periodic Scanning:
   - Scanning occurs at regular intervals (e.g., every 5 seconds) to update the cart's location as it moves.

Figure21 below provides a code snippet illustrating an algorithm for detecting a user's location within the CartWave system using Bluetooth Low Energy (BLE) beacons.

Code Snippet Example:

```python
import asyncio
from bleak import BleakScanner

latest_zone = None  # Global variable to store the latest detected zone

async def scan_for_beacons():
    global latest_zone
    while True:
        devices = await BleakScanner.discover()
        for device in devices:
            uuids = device.metadata.get('uuids', [])
            if 'UUID_ZONE_A' in uuids:
                latest_zone = 'A'
            elif 'UUID_ZONE_B' in uuids:
                latest_zone = 'B'
            elif 'UUID_ZONE_C' in uuids:
                latest_zone = 'C'
        await asyncio.sleep(5)  # Wait 5 seconds before the next scan

# Start the scanning loop
asyncio.run(scan_for_beacons())
```

Figure 21 :Location Detection Algorithm Code Snippet Example.

Integration with the System:

- Advertisements and Promotions:
    - Based on the detected zone, the system retrieves relevant advertisements to display to the customer.
- Payment Zone Detection:
    - When the cart enters the payment zone, the system initiates the checkout process.

5.4.3 Avoiding Duplicate Detections

To prevent duplicate product entries, we apply the following steps:

- State Tracking:
    - The system keeps track of the last detected product and its confidence score.
- Thresholding:
    - A confidence threshold (e.g., 0.7) is set to ensure only confident predictions are considered.
- Comparison:
    - If the current detection matches the last one and occurs within a short time frame, it is ignored.

Figure22 presents a code snippet demonstrating a mechanism to avoid duplicate product detections in the CartWave system. This is crucial to prevent the same product from being added to the cart multiple times when it remains in view of the camera.

Code Snippet Example:

```
last_detection = None  # Store last detection to avoid duplicates

def predict_product(image):
    global last_detection
    class_name, confidence_score = recognize_product(image)
    if class_name and confidence_score:
        if last_detection != (class_name, confidence_score):
            last_detection = (class_name, confidence_score)
            # Add product to cart
            return class_name, confidence_score
    # Ignore duplicate detection
    return None, None
```

Figure 22:Avoiding Duplicate Detections Code Snippet Example.

5.4.4 Error Handling and Robustness

- Exception Handling:
    - Try-except blocks are used to catch and handle errors during image processing and BLE scanning.
- Asynchronous Operations:
    - Asynchronous programming ensures that scanning for BLE devices does not hinder the performance of other tasks.

This figure shows a code snippet demonstrating error handling within the CartWave system, specifically during the product prediction process.

Example:

```
try:
    # Attempt to process image
    class_name, confidence = predict_product(image)
except Exception as e:
    print(f"Error during product prediction: {e}")
```

Figure 23:Error Handling and Robustness Code Snippet Example.

5.4.5 Example Usage Scenario

When a customer places a product in front of the smart cart's camera:

1. Image Capture:
    - The camera captures the image of the product.
2. Product Recognition:
    - The image is processed, and the product is recognized using the machine learning model[8].
3. Cart Update:
    - The product is added to the digital cart displayed on the touchscreen.
4. Zone Detection:
    - As the customer moves, the cart detects BLE[5] modules to determine its location.
5. Personalized Content:
    - The system displays location-based advertisements relevant to the current zone.
6. Checkout Process:
    - Upon entering the payment zone, the cart initiates the automated checkout process.

5.4.6 Extensibility

The algorithms are designed to be extensible:

- Adding Products:
  - New products can be added by retraining the model with additional images, following the same training process.
- Adding Zones:
  - Additional zones can be included by deploying more BLE modules and updating the UUIDs in the code.

Steps to Add a New Zone:

1. Deploy a New BLE Module:
   - Program the BLE module with a unique UUID using the Arduino IDE.
2. Update the Code:
   - Add the new UUID and corresponding zone label to the scanning script.
3. Test the Integration:
   - Ensure the system correctly detects the new zone and triggers appropriate actions.

By implementing these algorithms, the CartWave system achieves efficient product recognition and accurate location detection, enhancing the overall shopping experience for customers. The modularity and extensibility of the algorithms ensure that the system can adapt to new products and store configurations as needed.

5.5 **Prototype Deployment**

In this section, we detail how the CartWave system prototype was deployed, integrating both hardware and software components to create a functional smart cart. The deployment process follows the BPMN models and incorporates the main algorithms discussed earlier.

5.5.1 System Usage Overview

The CartWave system enhances the shopping experience by allowing customers to interact with a smart cart equipped with advanced features. The system operates as follows:

1. **User Registration and Cart Pairing**: Customers register through the mobile application and pair their account with a smart cart using a generated barcode. Figure24 shows a screenshot of the CartWave mobile application interface during the user registration and cart pairing process.

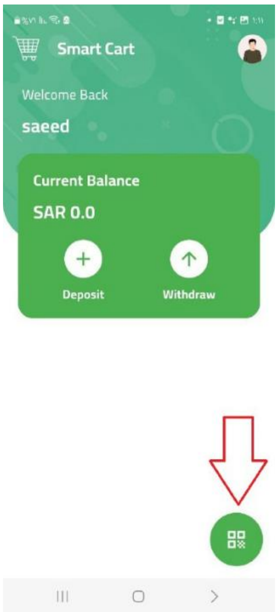

Figure 24:User Registration and Cart Pairing in CartWave.

2. **Product Recognition and Cart Management**: Customers add products to their cart by scanning them with the smart cart's camera. The system recognizes the products

using the trained machine learning model and updates the digital cart accordingly[4].Figure25 shows a screenshot of the CartWave interface demonstrating product recognition and cart management features.

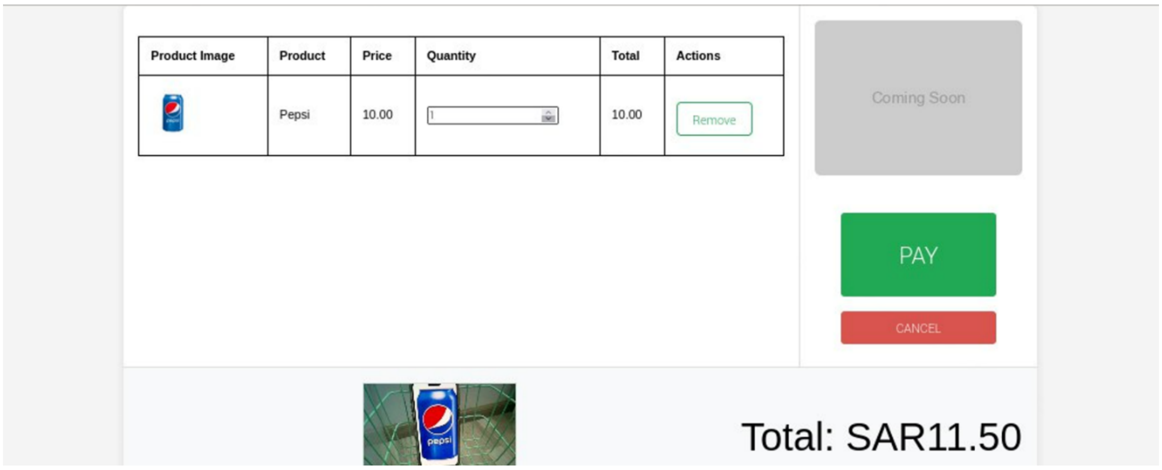

Figure 25:Product Recognition and Cart Management in CartWave.

3. **Location-Based Advertisements**: As customers navigate the store, the smart cart detects BLE[5] beacons to determine its location and displays relevant advertisements on the touchscreen interface. Figure26 shows another screenshot of the CartWave, this time highlighting the display of a location-based advertisement.

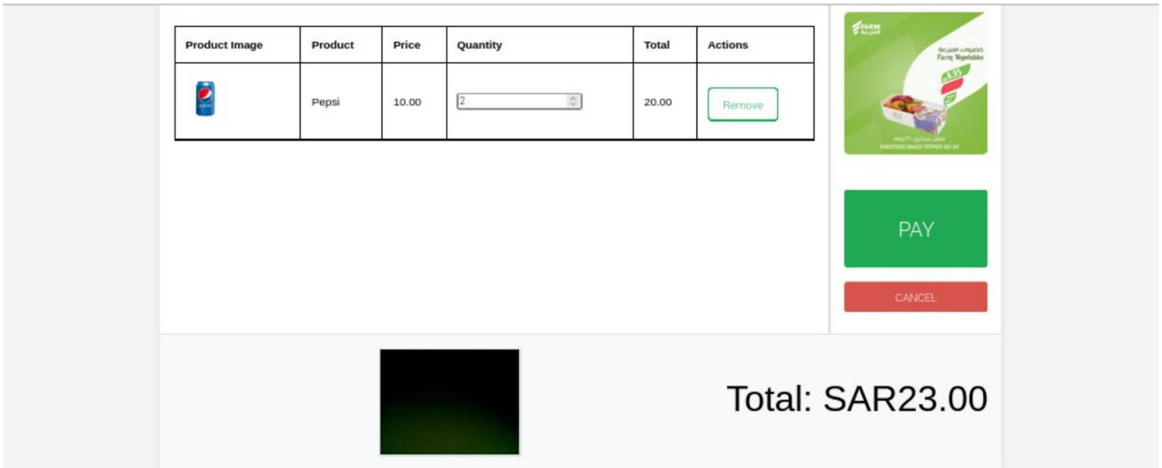

Figure 26:Location-Based Advertisements in CartWave

4. **Automated Checkout and Payment**: When customers enter the payment zone, the system initiates the checkout process, allowing for seamless payment through the mobile application.The figure below shows the CartWave mobile application interface during the automated checkout and payment process.

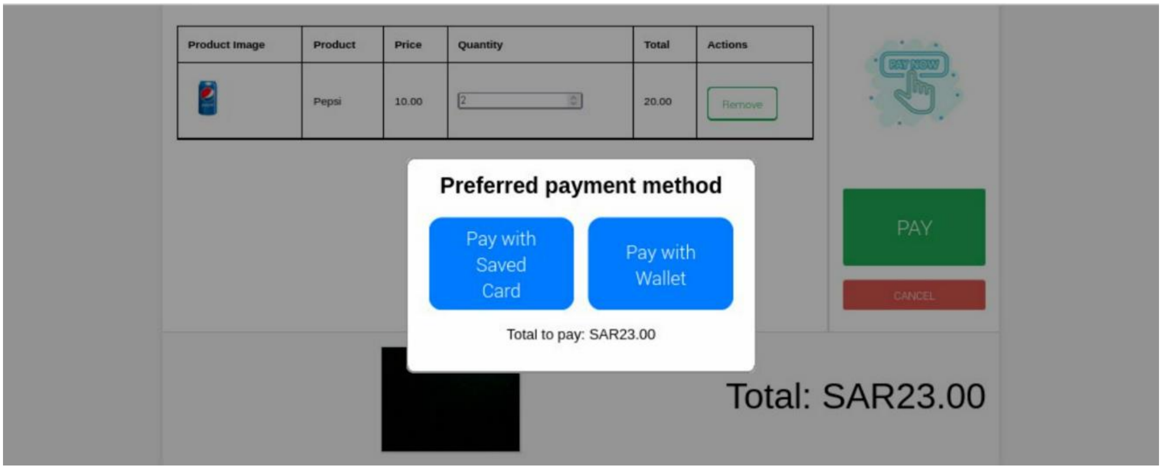

Figure 27:Automated Checkout and Payment in CartWave.

5.5.2 Hardware Components Deployment

The hardware components were assembled and integrated into the shopping cart as per the design specifications. We use the following components

- Raspberry Pi[6] Integration

The Raspberry Pi 4 Model B was mounted securely on the cart, connected to the Pi Camera module and the touchscreen display. *Figure28 shows the integration of a Raspberry Pi with other hardware components in the CartWave system, likely mounted on a shopping cart.*

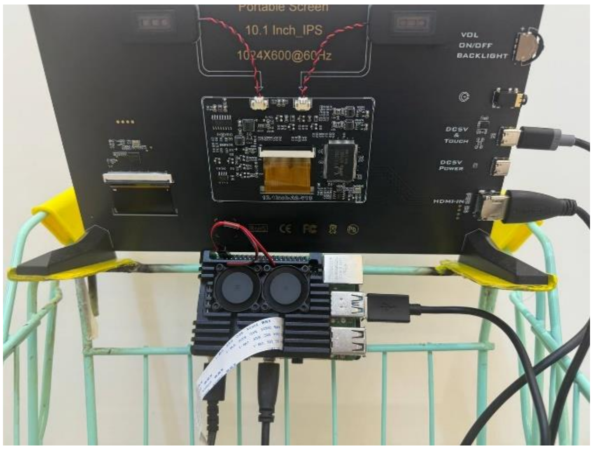

Figure 28:Raspberry Pi Integration with CartWave Hardware.

- Pi Camera[7] Installation

The Pi Camera was positioned to have a clear view of products placed in front of it for scanning. Figure29 provides a close-up view of the Raspberry Pi Camera Module installation on the CartWave cart.

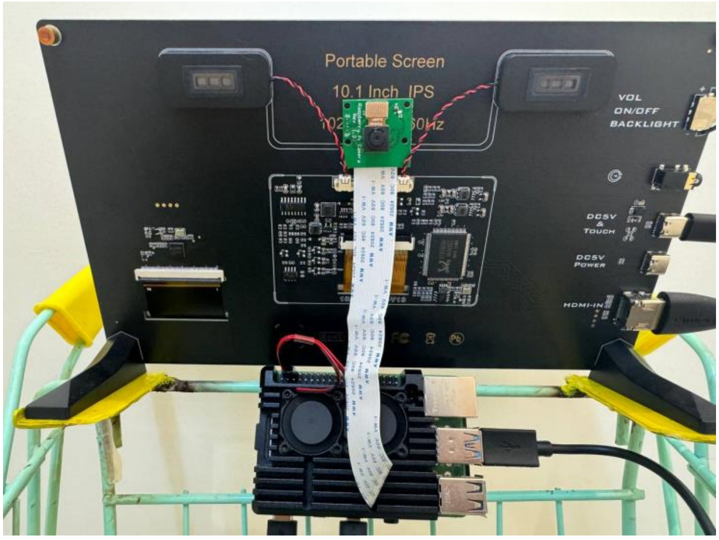

Figure 29 :Pi Camera Installation on CartWave Display.

- Touchscreen Display Setup

The touchscreen display was installed on the cart handle, providing easy access for customers. Figure30 shows the 10.1-inch touchscreen display installed and operational on a CartWave smart cart.

Figure 30:Touchscreen Display Setup on CartWave Smart Cart.

- BLE[5] Modules Deployment

BLE modules were programmed using the Arduino IDE and installed in various store zones for location detection. Figure 31 shows the deployment of multiple BLE (Bluetooth Low Energy) modules, ESP32-C3 beacons, for the CartWave system.

Figure 31:BLE[5] Modules Deployment for CartWave.

5.5.3 Software Components Deployment

The mobile application developed using Flutter and Dart[12] was deployed to both Android and iOS devices, enabling customers to register, pair with the cart, and manage their accounts. Figure32 provides a glimpse into the development environment and user interface of the CartWave mobile application.

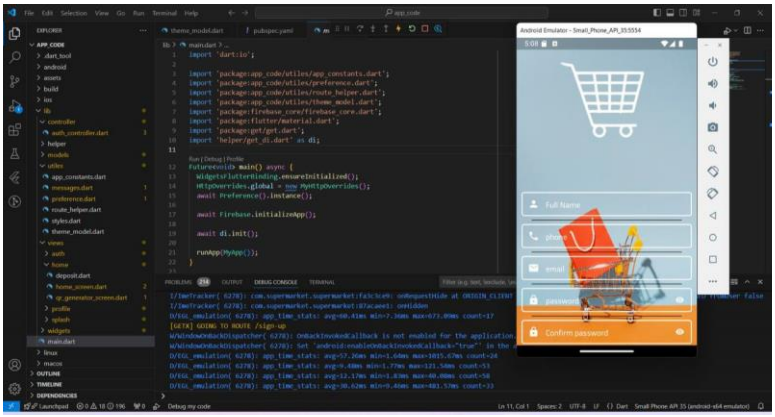

Figure 32:CartWave Mobile Application Development

The cart interface, developed using PHP[13], HTML[14], CSS[14], and JavaScript[15], was deployed on the Raspberry Pi and displayed on the touchscreen. Figure33 below showcases the user interface of the CartWave cart application, responsible for displaying and managing the shopping cart contents on the touchscreen display.

Figure 33: CartWave Cart Application Interface.

The trained machine learning model was deployed on the Raspberry Pi to perform real-time product recognition[4]. Figure34 shows the learning model for product recognition in the CartWave system.

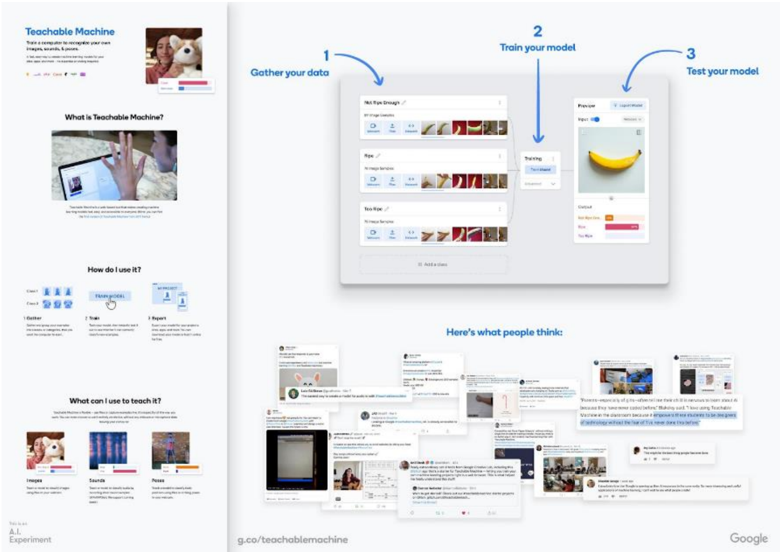

Figure 34: teachable machine.[8]

Firebase services [16] were configured to handle authentication, real-time database operations, and cloud functions. Figure35 shows the Firebase console, highlighting the integration of Firebase services into the CartWave system.

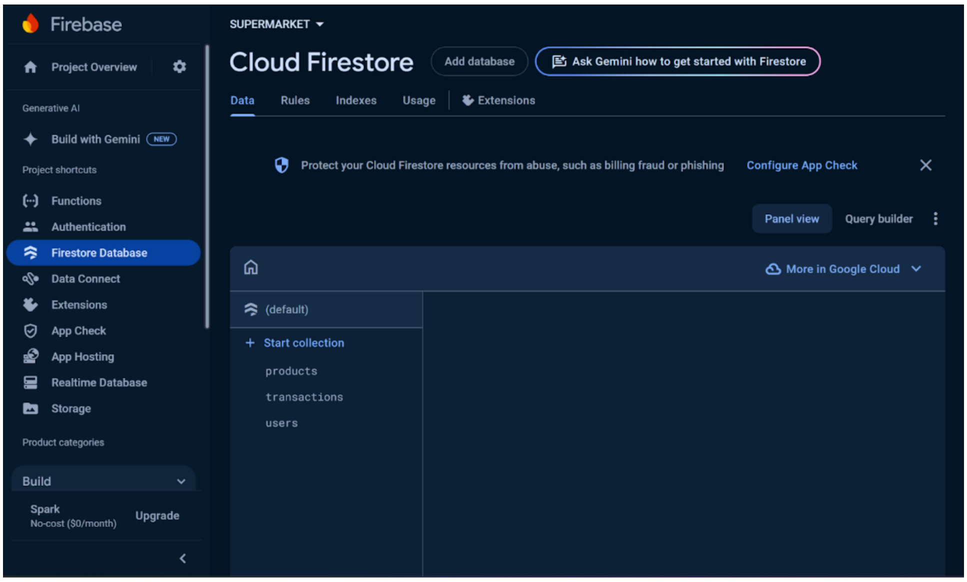

Figure 35: Integration with Firebase for CartWave.

5.5.4 Implementation of BPMN[32] Models and Algorithms

The deployment closely followed the BPMN models designed earlier, ensuring that the system's processes mirrored the planned workflows.

- **Use Case 1**: User registration and cart pairing were implemented as per the BPMN diagram, utilizing QR code scanning and account linking algorithms.
- **Use Case 2**: The product recognition algorithm was deployed to allow adding and removing products from the cart, following the designed process flow.
- **Use Case 3**: Location-based advertisements were enabled through the BLE[5] detection algorithm, displaying relevant content as customers moved through different zones.
- **Use Case 4**: The automated checkout process was implemented, initiating payment when the cart entered the payment zone, in line with the BPMN model.

By adhering to the BPMN models and utilizing the algorithms effectively, the prototype demonstrates the intended functionalities of the CartWave system.

5.6 Testing and Validation

This section focuses on testing the implemented use cases and validating the system's performance according to the designed steps. Screenshots and images illustrate how the system operates in real scenarios.

Testing Use Case 1: User Registration and Cart Pairing

- **Testing Steps**:
    1. Open the mobile application and navigate to the registration screen. Figure36 shows User Registration interface and user required information for registration.

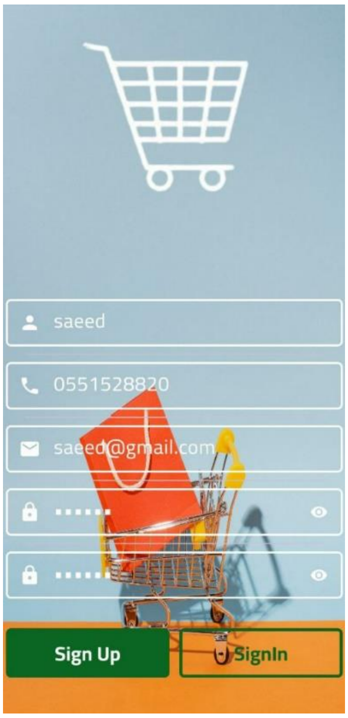

Figure 36:User Registration and Cart Pairing .

2. Generate the barcode for cart pairing. Figurs37 shows User welcome interface and features like withdraw and deposit and displays a QR code, used for pairing a user's Cart-Wave account with a specific smart cart in the store and

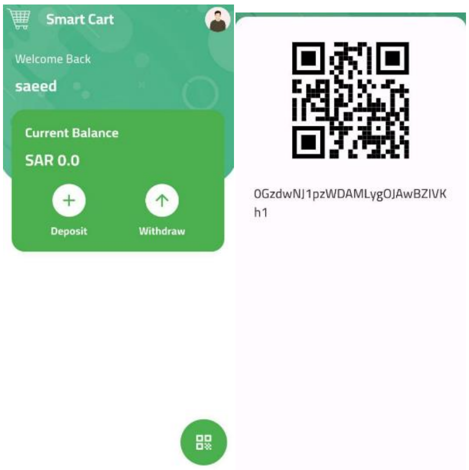

Figure 37:QR Code for Smart Cart Pairing.

3. Scan the barcode using the smart cart's scanner to link the account. Figure 38 shows the CartWave mobile app interface during the process of scanning a QR code to pair with a smart cart.

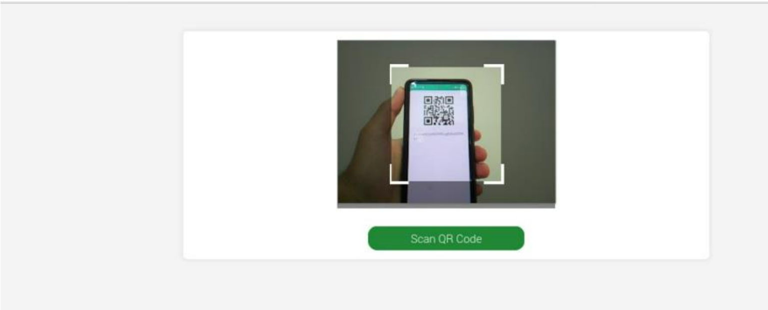

Figure 38 :QR Code Scanning for Cart Pairing in the CartWave App.

- **Validation**: The system successfully linked the user's account to the smart cart, allowing personalized interactions.

Use Case 2: Adding and Removing Products

- **Adding a Product**:

    1. Place a product in front of the smart cart's camera and the system recognizes the product and adds it to the digital cart. Figure39 shows the CartWave cart application interface demonstrating the successful recognition and addition of a product to the digital shopping cart.

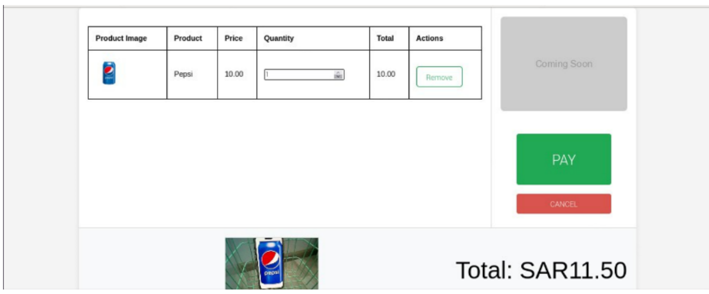

Figure 39: Product Recognition and Addition to Cart in CartWave.

- **Removing a Product**:

    1. Navigate to the cart contents on the touchscreen interface and select the product to remove and confirm the action. Figure40,41 shows the Cart-Wave cart application interface during the process of removing a product from the digital shopping cart and shows the CartWave cart application interface after all products have been removed from the digital shopping cart.

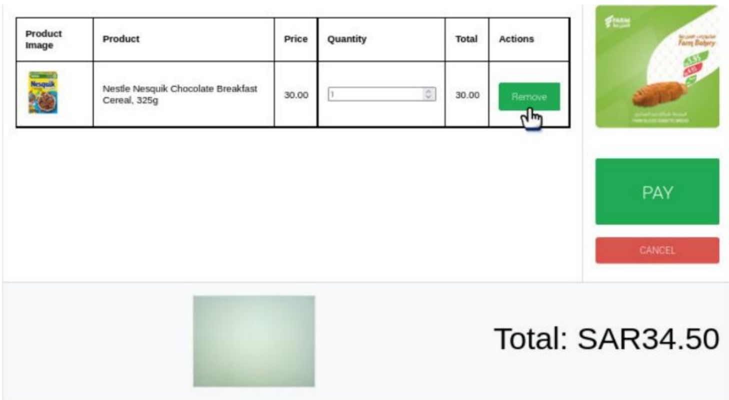

Figure 40: Removing a Product from the Cart in CartWave.

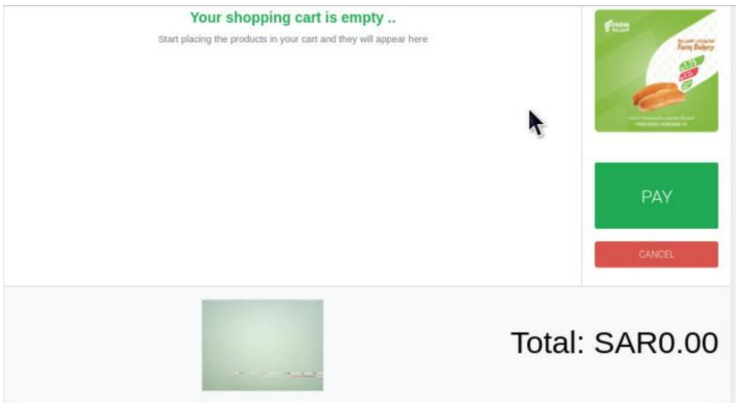

Figure 41: Empty Cart State in CartWave.

Validation: Products were accurately recognized and added or removed from the cart as expected.

Testing Use Case 3: Receiving Location-Based Advertisements

- **Testing Steps**:
    1. Move the smart cart into a designated zone with a BLE[5] beacon. Figure 42 shows the CartWave cart application interface in a state where it appears to be ready to display a location-based advertisement.

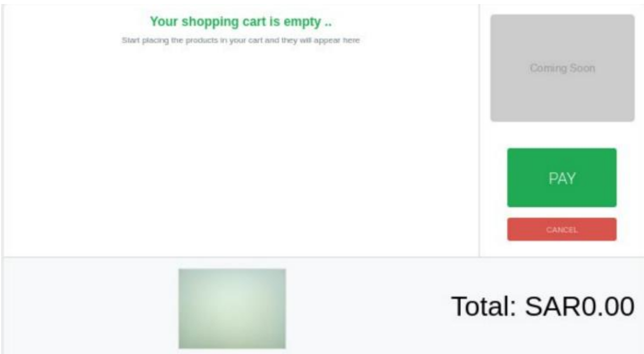

Figure 42: CartWave Awaiting Location-Based Advertisement.

2. The system detects the zone change and displays relevant advertisements. Figure 43 shows the CartWave cart application interface with a location-based advertisement displayed.

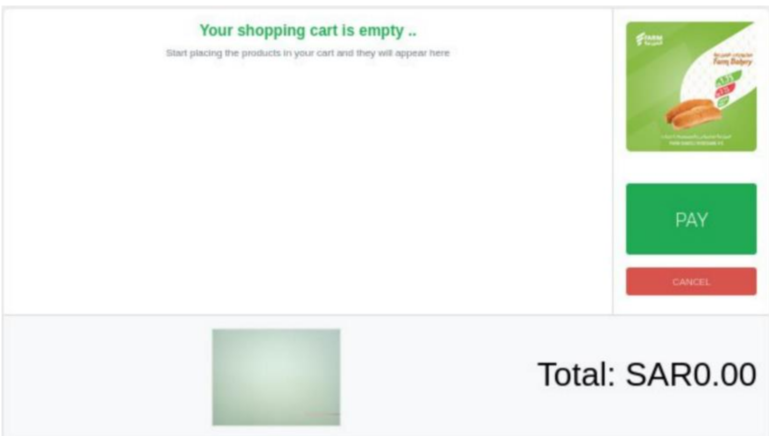

Figure 43 : Location-Based Advertisement Display in CartWave.

- **Validation**: The system correctly detected location changes and presented appropriate advertisements.

Use Case 4: Automated Checkout and Payment

- **Testing Steps**:
    1. Enter the payment zone with the smart cart. Figure44 shows the CartWave cart application interface as the user enters the designated payment zone within the store , now the pay button works and the add photo change to be pay now.

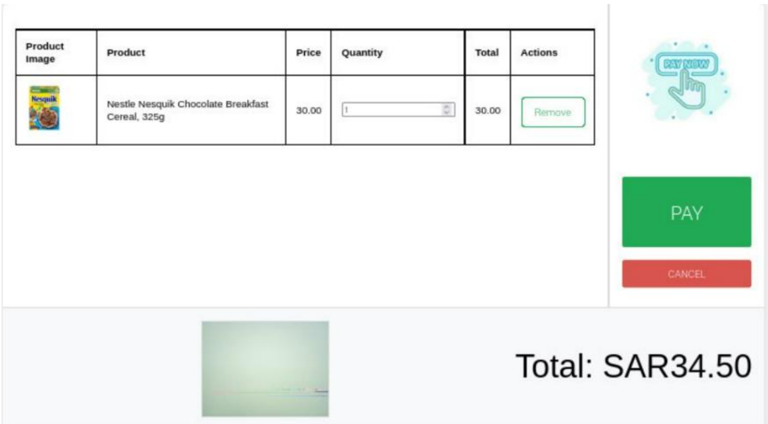

Figure 44: Entering the Payment Zone in CartWave

2. The cart interface prompts for payment. Figure45,46 shows the CartWave cart application interface prompting the user to select their preferred payment method and shows the CartWave cart application interface confirming a successful payment.

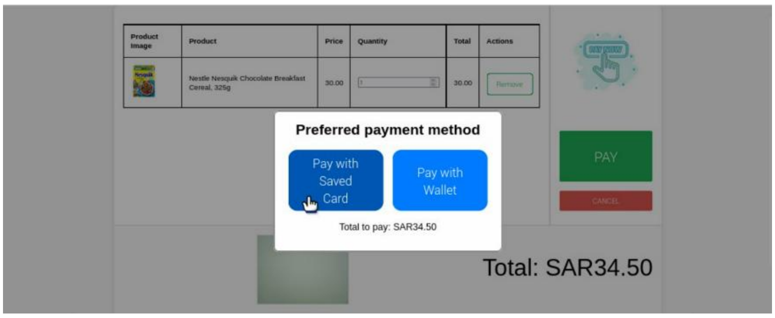

Figure 45: Payment Prompt in CartWave.

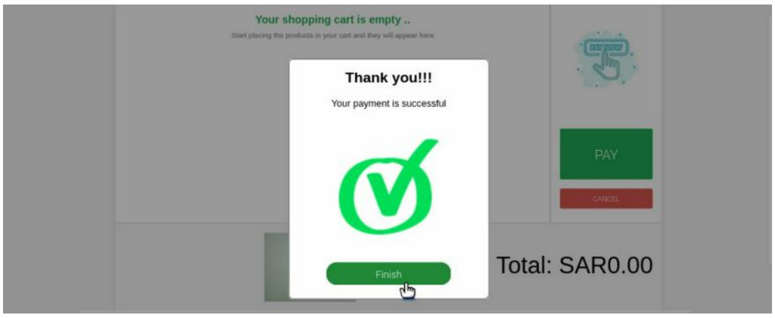

Figure 46: Successful Payment Confirmation in CartWave.

3. View bill from mobile app. Figures 47 shows the CartWave mobile app interface displaying a bill or transaction summary after a purchase has been completed and another view of the CartWave mobile app interface displaying a bill or transaction details.

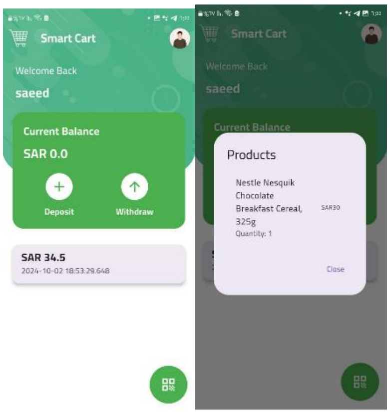

Figure 47: Viewing the Bill in the CartWave Mobile App.

- **Validation**: The checkout process was initiated automatically, and payment was processed successfully.

Use Case 5: Admin Access to Sales Data and Inventory Management

Testing Steps:

Logging in as Admin:

- Open the mobile application on a device.
- Navigate to the login screen.

- Enter the admin email and password.

Figure48 shows the CartWave mobile app interface with the login screen, likely accessed by an administrator to manage sales data and inventory

Figure 48: Admin Login in CartWave.

Accessing Admin Dashboard:

- *Upon successful login, the application recognizes the admin credentials and displays the admin dashboard instead of the regular user interface.* Figure49 showcases the CartWave admin dashboard, which is displayed after a successful admin login.

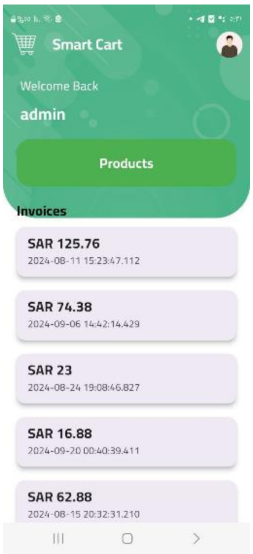

Figure 49: CartWave Admin Dashboard.

Viewing Invoices:

- Select an invoice to view detailed information. Figure50 provides a closer look at the CartWave admin dashboard, specifically the detailed view of an invoice

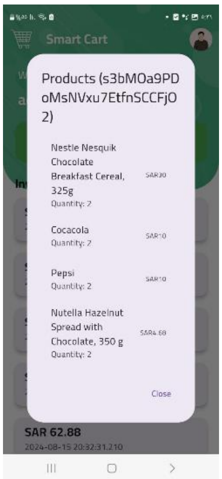

Figure 50: Viewing Invoice Details in CartWave Admin Dashboard.

Managing Inventory:

- *Navigate to the 'Inventory Management' section, View current stock levels and product details.* Figure51 showcases the "Inventory Management" section within the CartWave admin dashboard, where administrators can view and manage product stock levels.

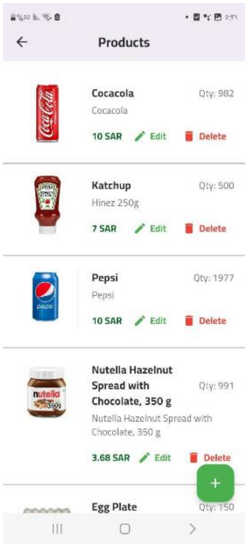

Figure 51: Inventory Management in CartWave Admin Dashboard.

Editing Inventory Items:

- *Select a product to edit.* Figure52 shows the interface for editing product information within the CartWave admin dashboard.

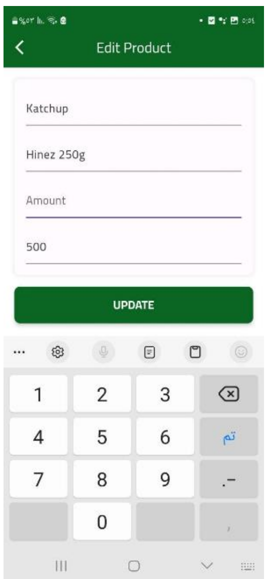

Figure 52: Editing Product Details in CartWave Admin Dashboard.

- *Update product information such as price, stock quantity, or description.*
- *Save changes.*

Figures 53 shows the "Edit Product" screen within the CartWave admin dashboard, specifically for modifying the price of "Ketchup" before the change is made also shows the "Products" screen within the CartWave admin dashboard, specifically after the price of "Ketchup" has been successfully updated.

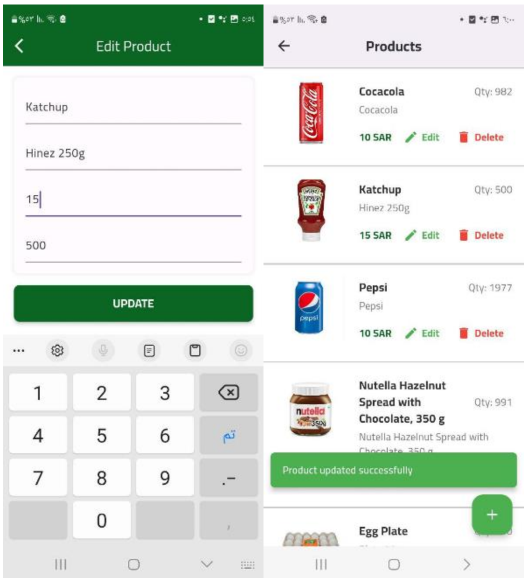

Figure 53: Editing and Updating "Ketchup" Price in CartWave Admin Dashboard

Adding a New Product:

- *Tap on the 'Add New Product' button.* Figure54 shows the interface for adding a new product to the CartWave inventory within the admin dashboard.

Figure 54: Adding a New Product in CartWave Admin Dashboard

- *Fill in product details and upload images.*
- *Submit the form.*

Figure55 shows the "Products" screen within the CartWave admin dashboard after the new product ("Kinder Bueno 2 Bars") has been successfully added to the inventory.

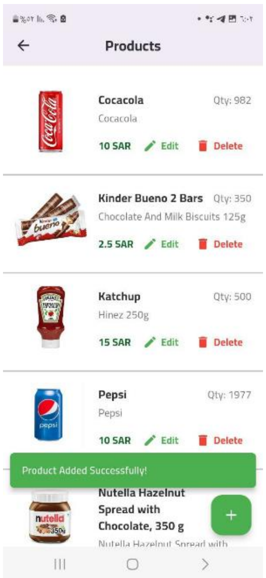

Figure 55: New Product Added to Inventory in CartWave Admin Dashboard.

The testing and validation phase demonstrated that the CartWave system functions as intended for the implemented use cases. The system effectively integrates hardware and software components to provide an enhanced shopping experience.

CHAPTER 6

CONCLUSION

6.1 **Summary of Finding**

The CartWave system was developed to enhance the shopping experience in retail supermarkets by integrating smart technology into traditional shopping carts. The project successfully combined hardware components like the Raspberry Pi[6], Pi Camera[7], and BLE[5] modules with software applications developed using Python[10], Flutter[12], and web technologies. The main achievements include:

- Efficient Product Recognition: Implemented a machine learning model trained with Teachable Machine to accurately identify products using image recognition on the Raspberry Pi[4][8].

- Seamless Customer Interaction: Developed a user-friendly mobile application and cart interface that allow customers to manage their shopping experience effortlessly.

- Location-Based Services: Utilized BLE technology to detect the cart's location within the store, enabling the display of relevant advertisements and promotions.

- Automated Checkout Process: Streamlined the payment process by initiating checkout when customers enter the payment zone, reducing wait times and improving convenience.

Overall, the project demonstrated that integrating smart technology into shopping carts can significantly enhance the retail shopping experience by providing personalized services and increasing operational efficiency.

6.2 **Limitations**

Despite the successful implementation, the project faced several limitations:

- **Hardware Constraints**: The Raspberry Pi's processing power limited the speed of image recognition, affecting real-time performance.
- **Model Scalability**: The need to retrain the machine learning model for each new product added challenges in maintaining and scaling the system.
- **BLE Interference**: Signal interference in the store environment sometimes led to inaccurate location detection.
- **Limited Testing Environment**: Due to resource constraints, testing was conducted in a simulated environment rather than a fully operational store, which may not capture all real-world variables.

These limitations highlight areas where further refinement and optimization are necessary to improve system performance and reliability.

6.3 Lesson learned

Throughout the development of the CartWave system, several valuable lessons were learned:

- **Importance of Modular Design**: Designing the system with modular components enhanced flexibility and made it easier to update individual parts without affecting the whole system.

- **User-Centered Development**: Prioritizing the user experience in both the mobile application and cart interface was crucial for ensuring customer satisfaction and adoption.

- **Challenges of Hardware-Software Integration**: Balancing the capabilities of hardware components with software requirements underscored the need for careful planning and testing.

- **Agile Development Benefits**: Adopting an Agile methodology allowed for iterative improvements and quick responses to issues that arose during development.

These lessons will inform future projects and contribute to more efficient and effective development processes.

6.4 **Future Work**

To build upon the foundation established by the CartWave system, several avenues for future work are proposed:

- **Enhancing Product Recognition**: Implement more advanced machine learning models and optimize image processing to improve speed and accuracy.

- **Dynamic Model Updates**: Develop a system that allows the machine learning model to update dynamically, enabling the addition of new products without extensive retraining.

- **Improving Location Accuracy**: Explore alternative technologies like Wi-Fi triangulation or RFID to enhance indoor positioning accuracy.

- **Expanded Testing**: Conduct extensive field tests in actual supermarket environments to gather real-world data and user feedback.

- **Scalability and Deployment**: Work on scaling the system for deployment in multiple stores, addressing challenges related to network infrastructure and data management.
- **Security Enhancements**: Strengthen security measures to protect user data and ensure compliance with privacy regulations.

By addressing these areas, the CartWave system can be further developed to provide a more robust, efficient, and user-friendly solution that meets the evolving needs of the retail industry.

7. References

[1] DESIGN thinking framework, Innovation & Methodology. IDEO U. (n.d.b).https://www.ideou.com/pages/designthinking?srsltid=AfmBOopZdOWZyJttZLyDchha1XdzB2anak5dmQMcuvVNBPH9HmxMzO8f

[2] RUBIN, Kenneth S. Essential Scrum: A Practical Guide to the Most Popular Agile Process. Addison-Wesley, 2013.

[3] "Find the Best IOT Solutions." IoT For All Solutions Marketplace - IoT Smart Solutions Platform, marketplace.iotforall.com/.

[4] Tim. "Object and Animal Recognition with Raspberry Pi and Opencv – TutorialAustralia."CoreElectronics,16Feb.2023,coreelectronics.com.au/guides/object-identify-raspberry-pi/.

[5] Afaneh, Mohammad. "Bluetooth Low Energy (BLE): A Complete Guide." Novel Bits, 30 Mar. 2023, novelbits.io/bluetooth-low-energy-ble-complete-guide/.

[6] RaspberryPi Documentation-Getting Started www.raspberrypi.com/documentation/computers/getting-started.html. [1] [1] [1] [1]

[7]Raspberry Pi Documentation-Camera, www.raspberrypi.com/documentation/accessories/camera.html.

[8] Google, Google, teachablemachine.withgoogle.com/.

[9] "Documentation." Debian, www.debian.org/doc/.

[10]"The Python LanguageReference."PythonDocumentation, docs.python.org/3/reference/index.html.

[11]"Arduino Documentation." Docs.Arduino.Cc, docs.arduino.cc/.

[12] "Build Apps with Flutter | GOOGLE for Developers." Google, Google, developers.google.com/learn/pathways/intro-to-flutter.

[13] "Hypertext Preprocessor." Php, www.php.net/docs.php.

[14] Harta Ciptajaya. "Documentation of HTML and CSS." The freeCodeCampForum,6Sept.2022,forum.freecodecamp.org/t/documentation-of-html-and-css/551354.

[15]DevDocs, devdocs.io/javascript/.

[16}*Google*, Google, firebase.google.com/?gad_source=1&gclid=CjwKCAjw6JS3BhBAEiwAO9waF-p97kiyauuY-h7P4FU4vunq ZSfVRkZk8ef4qz1M-ZIdMgvL1gO-RhoCUQkQAvD_BwE&gclsrc=aw.ds.

[17]Scenario Analysis: Definition, Importance, Examples and Faqs | Indeed.Com,www.indeed.com/careeradvice/careerdevelopment/scenario-analysis.

[18] "Miro: The Innovation Workspace." Https://Miro.Com/, miro.com/.

[19] ClickUpTM | One App to Replace Them All, clickup.com.

[20] Introduction to Cloud Computing, www.dialogic.com/~/media/products/docs/whitepapers/12023-cloud-computing-wp.pdf .

[21] Tamer, Ozgur, and Tunca Koklu. "A Smart Shelf Design for Retail Store Real Time Inventory Management Automation." Review of Computer Engineering Research 8.2 (2021): 96-102

[22] Pangriya, Ruchita. "Beacon Technology the Future of Retail: A Review of the Literature and SWOT Analysis." A Journal of Management 1 (2023): 11

[23] (PDF) Inventory Optimization Using Machine Learning: Creating a System That Utilizes Machine Learning Algorithms to Analyze Sales Data, Predict Future Demand, and Optimize Inventory Levels, Reducing CostsandImprovingEfficiencyforSmallBusinesses,www.researchgate.net/publication/378175939.

[24] Ali, Mohammed Maqsood, and Mohammad Haseebuddin. "Cloud computing for retailing industry: An overview." International Journal of Computer Trends and Technology 19.1 (2015): 51-56

[25] Shopping at an Amazon Go Storewww.amazon.com/gp/help/customer/display.html?nodeId=GQKJHZZQDJBQN2QF.

[26] "How Macy's Maximized in-Store Traffic with Beacons ." Retail Dive, www.retaildive.com/ex/mobilecommercedaily/how-macys-maximized-in-store-traffic-through-beacons-during-black-friday.

[27] Amazon Fresh Groceries – Learn More, www.azon.com/fmc/m/30003175?almBrandId=QW1hem9uIEZyZXNo.

[28] "Smart Shopping Carts: Ai Shopping Cart Technology." Caper, www.caper.ai/.

[29] "What Is Blue Ocean Strategy: About Blue Ocean Strategy." Blue Ocean Strategy, 1 Mar. 2024, www.blueoceanstrategy.com/what-is-blue-ocean-strategy/.

[30] "RFID Verification: What It Is and How It Works." Regula, regulaforensics.com/blog/rfid-verification/.

[31] Yasar, Kinza, and Robert Sheldon. "What Is Virtual Reality? How It's Used and How It Will Evolve." WhatIs, TechTarget, www.techtarget.com/whatis/definition/virtual-reality.

[32] Bernd Bruegge and Allen Dutoit, "Object-Oriented Software Engineering: Using UML, Patterns and Java", Third Edition, Prentice Hall 2009.

[33] Firebase APIR eference." Google,Google,firebase.google.com /docs/reference.

[34] Ghogen. "Visual Studio Documentation." Microsoft Learn, learn.microsoft.com/en-us/visualstudio/windows/?view=vs-2022.

[35] "API Documentation :Tensorflow V2.16.1." TensorFlow, www.tensorflow.org/api_docs.

Appendix A

System Architecture Overview

A.1 Hardware Components

- **Raspberry Pi 4 Model B**: Serves as the main processing unit for the smart cart.
- **Pi Camera Module**: Captures images for product recognition.
- **BLE[5] Modules**: Facilitate indoor positioning and communication.
- **Touchscreen Display**: Provides an interactive interface for customers.
- **Power Supply Unit**: Powers all hardware components.
- **Modified Cart Structure**: Houses and protects all integrated hardware securely.

A.2 Software Components

- **Mobile Application (Flutter and Dart)**: Allows user registration, account management, and cart pairing.
- **Cart Application (PHP, HTML, CSS, JavaScript)**: Runs on the Raspberry Pi to manage cart interactions.
- **Machine Learning Model (Teachable Machine, Python)**: Recognizes products using the Pi Camera[7].
- **Database and Cloud Services (Firebase)**: Handle authentication, data storage, and synchronization

Appendix B

Key Algorithms Overview

B.1 Product Recognition Algorithm

An algorithm utilizing a machine learning model trained with Teachable Machine [8]to recognize products through images captured by the Pi Camera. It processes images, predicts product classes, and updates the digital cart accordingly.

B.2 Location Detection Algorithm

A BLE-based algorithm that determines the cart's location within the store by detecting signals from BLE modules placed in different zones. It enables location-based services like targeted advertisements.

Appendix C

Testing and Validation Data

C.1 Product Recognition Accuracy

- **Total Test Cases**: 100
- **Successful Recognitions**: 90%
- **Summary**: The product recognition system showed high accuracy for trained products, indicating effective model training and implementation.

C.2 BLE Location Detection Performance

- **Zones Tested**: 3
- **Average Detection Accuracy**: 90%
- **Summary**: The location detection system reliably identified the cart's zone, with minor inaccuracies due to signal interference.

Appendix D

User Feedback Summary

D.1 User Satisfaction Highlights

- **Ease of Use**: Rated highly by users for its intuitive interface and straightforward interactions.
- **System Responsiveness**: Positive feedback on the promptness of product recognition and updates.
- **Areas for Improvement**: Suggestions included expanding the product database and enhancing personalization features.

Appendix E

Acronyms and Abbreviations

- **API**: Application Programming Interface
- **BLE[5]**: Bluetooth Low Energy
- **BPMN**: Business Process Model and Notation
- **IDE**: Integrated Development Environment
- **UI/UX**: User Interface/User Experience

By focusing on key information and avoiding repetition, this chapter concisely presents supplementary materials that support the main content of the report.