

FourGuys proposal to further improve the point-of-sale system: Checkout

Sven Veit
dept. of Economics
Kiel University of Applied Sciences
Kiel, Germany
svenveit@hanyang.ac.kr

Jiyong Lee
dept. of Educational Technology
Hanyang University
Seoul, Korea
jdragon2000@naver.com

Zabolotniaia Anna
dept. Business Administration
Hanyang University
Seoul, South Korea
elijaltayer@gmail.com

Aneesha Shaki
dept. of Engineering and Computer Science
University of New South Wales
Sydney, Australia
a.shaki@student.unsw.edu.au

Abstract—This document will define our proposal requirements, and explain our motivations and ongoing roles throughout the initial development process of the project and fulfil client requirements with an application to improve the POS system. This project aims to develop an enhanced POS (Point of Sale) system that leverages smartphone-based barcode scanning technology to enable customers to complete their purchases without the need to wait in checkout lines. By using our mobile application, users can simply scan product barcodes as they shop, automatically adding items to a virtual shopping cart. When ready, they can complete the payment through the app's secure and user-friendly interface, eliminating the need for traditional cashier interactions. This system not only reduces customer wait times but also minimizes staff workload and enhances the shopping experience by providing a seamless and efficient checkout process.

Index Terms—POS system, barcode scanning, mobile payment, contactless checkout

I. INTRODUCTION

A. Motivation

Amid the ongoing cost of living crisis and the rapid shift in consumer behavior, large retailers are increasingly focused on improving operational efficiency and cutting costs to meet rising customer expectations. Recent disruptions, including global supply chain issues and changing labor dynamics, have amplified the urgency for more innovative, cost-effective solutions in the retail industry. While technological advancements have already revolutionized the checkout process, most notably with the introduction of contactless kiosks as the benchmark for modern Point of Sale (POS) systems, there is still untapped potential for further improvement.

By capitalizing on the most universally accessible tool, the smartphone, we have the opportunity to not only reduce costs but also redefine the customer experience. The current retail landscape demands flexibility, and utilizing smartphones as mobile POS systems is a natural progression. This approach can significantly reduce infrastructure costs, eliminate the need for fixed kiosks, and decrease reliance on labor-intensive processes. Existing solutions allow QR and barcode scanning, but

this project goes beyond that, aiming to fully integrate smart devices into the shopping journey. By empowering consumers to scan and purchase items independently, we can deliver a frictionless, personalized retail experience that meets the evolving demands of today's shoppers, while future-proofing retail operations against future disruptions

B. Problem Statement

The current issues within the economy due to the current inflation globally has caused for a need to reduce costs within the retail industry to place less burden on the consumer with a lack of leeway within produce and goods adaptations to the current POS systems and how to increase the efficiency and costs for the business whilst simplifying the process for the consumer are concerns of leading industry giants.

The main issues are as follows:

1) *Long wait times and congestion*: After completing their shopping, customers often face long queues at the checkout, especially during peak hours. This not only causes discomfort for customers, but also forces stores to operate more checkout counters or hire additional staff to maintain efficiency. These measures lead to increased costs and reduced customer satisfaction.

2) *Increased costs*: Installing and maintaining traditional POS systems and kiosks incurs significant costs, which rise exponentially as the size of the store increases. Additionally, these systems require regular updates and maintenance, further driving up operational expenses.

3) *Lack of flexibility*: The current payment system relies on fixed checkout counters or kiosks, which makes it difficult to adapt to customer traffic flow or shopping patterns. This limitation hampers the store's ability to manage in-store congestion or offer more convenient options to customers.

4) *Limited customer experience*: Current payment systems do not provide customers with enough control over their

shopping experience, such as real-time price checking or managing their shopping list. This limits customers' ability to self-manage information related to their purchases.

These issues lead to customer dissatisfaction and increase operational costs. Our project aims to address these problems by introducing a mobile-based payment system that increases efficiency, reduces costs and improves the customer experience by allowing a streamlined seamless method of payment.

C. Solution

Our proposal is to solve this current problem through a mobile application that has the capabilities of functioning as a POS system within physical retail stores.

With this solution, customers can avoid traditional checkout lines, significantly reducing congestion and wait times, especially during peak shopping hours. Since there is no need for physical POS kiosks or additional staff assistance, retailers can drastically lower their infrastructure and labour costs. Moreover, this system allows retailers to move beyond the limitations of fixed checkout counters, adapting more flexibly to fluctuating customer flows.

Using smartphones also empowers customers to have more control over their shopping experience. They can access real-time information on prices, promotions, and their total cart value, easily add or remove items, and apply discounts automatically, completing transactions without physical interaction. This enhances convenience and safety, especially in the post-pandemic era where contactless solutions are highly valued.

By integrating this mobile-based payment system, we aim to not only boost operational efficiency but also transform the overall shopping experience into a more flexible, cost-effective, and customer-centric retail environment.

D. Relate Software

1) *Amazon Price Comparison and Product Scanner*: Amazon's mobile app allows users to scan product barcodes and instantly access product information, reviews, and pricing within Amazon's marketplace. Available on both iOS and Android, this app uses barcode scanning to retrieve product details, enabling users to compare Amazon's prices with in-store prices. This functionality closely aligns with the goals of our project.

2) *ShopSavvy Barcode Scanner*: ShopSavvy is an app that allows users to scan barcodes to view pricing and availability across various online and local stores. Available on the App Store and Google Play Store, ShopSavvy's barcode scanning technology and its price comparison feature across multiple retailers align closely with the objectives of our app.

3) *Google Lens*: Google Lens can scan barcodes and provide detailed product information, including prices, from various online sources. Integrated within Google Search and Google Photos, Google Lens uses advanced image recognition and search capabilities to retrieve product information via barcodes, which mirrors the core functionality of our proposed app.

4) *Barcode Scanner Algorithm*: Algorithms like ZBar and pyzbar are widely used in Python for barcode detection and decoding in various apps and services. The pyzbar Python library, available at <https://pypi.org/project/pyzbar/>, offers an open-source solution for reading barcodes in images, making it a potential foundational component for implementing the barcode scanning functionality in our app.

These references illustrate existing solutions and technologies that provide similar barcode scanning and price comparison features. For our app, integrating similar barcode decoding libraries and developing a comparison feature across supermarkets would differentiate it by focusing on price comparisons specifically within the grocery retail sector

E. Role Assignments

TABLE I
ROLE ASSIGNMENTS

Role	Name	Task
Software Developer	Sven	Develop the fundamental back-end systems to allow fulfill the functional requirements.
Software Developer	Aneesa	Develop the fundamental backend systems to allow fulfill the functional requirements.
Front-end Developer	Jiyong	Develops and optimizes user interfaces (UI) for web and mobile applications, ensuring a seamless and responsive user experience (UX). Collaborates with designers to implement visually engaging and accessible components, focusing on usability and interactivity. Ensures cross-platform compatibility and addresses performance optimization for efficient rendering and responsiveness.
Graphics Designer	Anna	To construct a visual design for the user experience

II. REQUIREMENTS

A. Barcode Scanner Feature

1) *Barcode Scanning via Camera*: Users can conveniently scan product barcodes using their smartphone camera to retrieve detailed product information instantly. This feature supports fast and accurate product recognition, making it easy for users to view product descriptions, prices, and other relevant details without manually entering information. By scanning each item as they shop, users can quickly and seamlessly register products in their cart, enhancing the overall shopping efficiency.

2) *QR Code Scanning Support*: In addition to traditional barcodes, users can scan QR codes directly within the app to access special offers, discounts, or additional product information. This functionality allows for the automatic application of relevant coupons or promotional discounts, so users can effortlessly benefit from store promotions. The QR code scanning capability adds flexibility, enabling a smoother shopping experience where savings are automatically applied.

B. Shopping Cart Feature

1) *Add Products*: Each product scanned by the user is added to a virtual shopping cart in real time. The cart updates

with the latest price and quantity, allowing users to keep track of what they've added and monitor their spending. With each scan, product details such as name, price, and available quantity are immediately visible, making it simple for users to manage their shopping list and streamline the purchasing process.

2) *Modify and Remove Products:* Users can easily manage items in their shopping cart by adjusting quantities or removing products as needed. Each modification automatically recalculates the total price, reflecting real-time changes and ensuring users have a clear and accurate view of their spending. This flexibility helps users control their shopping experience, making it easy to review and make changes before finalizing a purchase.

3) *View total Payment Amount:* The system calculates the total cost of all items, including any applicable taxes, discounts, and promotions, in real time. This feature provides a transparent view of the final amount due, helping users make informed purchasing decisions and preventing unexpected charges at checkout. The clear summary of the total payment amount gives users a quick overview of their cart balance at any time during the shopping process.

C. Payment Feature

1) *Choose Payment Methods:* Users can register multiple payment methods, such as credit cards, mobile payment options, or digital wallets, within the app. Each saved method can be securely managed, allowing users to select or change their preferred payment option quickly and efficiently at checkout. By offering flexible payment choices, the app supports a variety of user preferences, making the checkout process faster and more convenient.

2) *Instant Payment:* With a single click, users can complete their purchase securely and quickly. The instant payment feature leverages secure transaction protocols, ensuring each payment is processed safely. This option is designed for efficiency, reducing wait times and enabling a seamless checkout experience that helps users finalize their purchases with confidence.

3) *View Payment History:* After each transaction, users have access to detailed payment histories, including itemized receipts and comprehensive records of previous purchases. This feature allows users to track their spending, review past orders, and easily access receipts for future reference. Payment history is stored in the user's profile, ensuring that all past transactions are conveniently available for review.

D. Promotions and Coupon Management

1) *Register Discount Codes/Coupons:* Users can enter discount codes or coupons directly within the app, which are then stored in their account for future use. The system remembers these codes, automatically applying them to eligible purchases during checkout. This feature makes it easy for users to keep track of their savings and ensures that no discounts are overlooked when shopping.

2) *Automatic Coupon Application:* When a product is scanned, the system automatically checks for any applicable promotions or discounts. Relevant offers are applied in real time, with the discounted prices immediately visible in the shopping cart. This seamless integration of promotions enhances the user experience, as they can see their savings instantly without additional steps.

3) *Reward Programs:* Users can accumulate rewards or loyalty points from store purchases, which can be redeemed or applied to future transactions. The system tracks each user's reward status, making it easy for users to benefit from loyalty programs and store credits that encourage continued shopping. This feature adds value by promoting customer retention and offering users more ways to save.

E. Notifications and Push Features

1) *Promotion Notifications:* Users receive real-time push notifications about discounts, special offers, or promotions that relate to products they've scanned or shown interest in. These timely updates ensure that users never miss an opportunity to save and can stay informed of new deals without actively searching for them. The notifications are customizable, allowing users to control which alerts they receive.

2) *Payment and Order Notifications:* Once a transaction is completed, users are instantly notified of successful or failed payments, along with real-time updates on the status of their orders. This feature provides assurance and transparency, as users are kept informed of their transaction status and any changes to their order's processing or delivery timeline.

F. User Profile Management

1) *Manage Personal Information:* Users can update and manage their personal information, such as name, contact details, and address, within the app. Keeping this information accurate ensures a seamless and personalized shopping experience. The app prioritizes data security, so users can trust that their information is protected while they make updates as needed.

2) *Manage Payment Information:* Users have full control over their registered payment options, allowing them to add new payment methods or edit existing details whenever necessary. This flexibility enables users to keep their payment preferences up-to-date, making checkout smooth and preventing any interruptions due to outdated information.

III. DEVELOPMENT ENVIRONMENT

A. Development Platform

Web application platform is suitable for our software due to its accessibility, scalability and cross platform compatibility. Unlike traditional desktop applications, a web application can be accessed from any device with internet access, allowing users to interact with the software in real time without requiring specific hardware or operating system configurations. This flexibility significantly broadens the user base, as it eliminates the need for installations or updates on individual devices. Additionally, web applications are naturally scalable, making

it easy to accommodate growing user demands and adapt to changing requirements with no requirement to accommodate to the OS system of a device reducing the complexity of the project. Development frameworks for web applications also offer extensive libraries and tools that accelerate development and provide built-in security features, which are crucial for protecting sensitive data. Altogether, these advantages make a web application an optimal platform for delivering robust, user-friendly, and secure software solutions.

B. Coding Languages

1) *JavaScript*: JavaScript is a coding language influenced by Java with syntax derived from C. It is enriched by a vast ecosystem of libraries and frameworks that enhance its capabilities in barcode scanning, data handling, and API integration. Libraries like QuaggaJS, jsBarcode, and Dynamsoft are specifically designed for barcode scanning, enabling web applications to process barcode data directly through the browser. For data handling, libraries such as Lodash and D3.js offer powerful tools for data manipulation, visualization, and transformation, making it easier to process large datasets and present them in interactive, user-friendly formats. JavaScript also offers comprehensive support for API integration, with libraries like Axios and Fetch simplifying asynchronous data requests and responses. For handling real-time data, libraries such as Socket.IO facilitate WebSocket connections, allowing developers to create applications that update dynamically without page reloads. Additionally, popular frameworks like React, Angular, and Vue streamline the development of user interfaces, providing reusable component architectures and state management tools, while libraries such as jQuery simplify DOM manipulation and event handling. These frameworks and libraries, along with JavaScript's cross-platform support, make it an essential language for building feature-rich, responsive, and data-driven web applications. Compatible software development frameworks include Node.js and React Native which offer an extensive library for building and developing software and a range of tools.

2) *Typescript*: Typescript is an extension of JavaScript that implements static typing, classes and interfaces. This allows for optimal code organisation and object-oriented programming techniques. Typescript allows for improved error detection with additional tooling support that enhances the IDE experience and ensures proper API documentation.

3) *CSS*: CSS frameworks and libraries complement JavaScript, offering pre-designed, responsive UI components that enhance the styling and layout of web applications. Libraries like Bootstrap, Tailwind CSS, and Bulma streamline styling by providing ready-made classes for layouts, typography, buttons, and more, allowing developers to quickly achieve consistent, visually appealing designs. Additionally, CSS animation libraries, like Animate.css and GreenSock (GSAP), enable smooth transitions and dynamic visual effects that bring interactivity to user interfaces.

4) *HTML*: HTML is the core language for structuring and presenting content on the web. It serves as the foundation of

web pages, defining elements like headings, paragraphs, images, links, and forms, enabling browsers to interpret and display content in a coherent layout. The latest version, HTML5, introduced new semantic tags such as header, footer, article, and section, which help organize content more meaningfully while improving accessibility and search engine optimization (SEO). HTML5 also brought powerful features for embedding multimedia through audio and video elements, along with APIs that enable functionalities like local data storage and geolocation, enhancing interactivity and user experience in web applications. Working together with CSS for styling and JavaScript for interactivity, HTML is essential for building accessible, structured, and user-friendly websites and applications.

C. Cost Estimation

TABLE II
COST ESTIMATION TABLE

Expense attribute	Cost per annum	reasoning
Web Hosting and Server Costs	\$1200-2400	Platforms such as AWS, Google Cloud or Heroku are required to host the website for use to deliver reliable, scalable and high-performance solutions.
Database Hosting	\$240-600	Database hosting provides infrastructure needed to store, manage and access databases over the internet. Examples of database host include Firebase, MongoDB Atlas or MySQL.
Outsourced API	\$600-1200	In the case of using a third party API to access barcode scanning libraries to reduce code development and focus on the websites functionalities.
Domain Name/registration	\$0-20	The costs associated to secure a domain name.
SSL	\$0-249	A digital authentication of the websites identity and enables encrypted connection for security purposes and needed to accept online payments.

D. Software in Use

1) *VSCode*: VSCode (Visual Studio Code) is our chosen code editor due to its flexibility and extensive support for various programming languages and extensions. It plays an important role in our project, which involves creating an application that allows users to scan barcodes with their phones for payment. VSCode allows us to customize our workspace and integrates well with Git, enabling us to efficiently track code changes. Additionally, it provides features like live sharing and remote development, enhancing collaboration and productivity among team members.

2) *Git and Gitlab*: Git serves as our core version control system, essential for tracking and managing code changes systematically. GitLab complements Git by providing a self-hosted platform with robust DevOps tools, allowing us to manage our barcode payment app project in one integrated environment. GitLab's CI/CD (Continuous Integration and Continuous Deployment) features are especially valuable for automating testing and deployment processes, which helps

streamline our development pipeline. It also offers comprehensive project management tools, enabling us to oversee the entire lifecycle of the project in one place, from issue tracking to deployment. GitLab's integrated approach enhances our ability to handle tasks efficiently and keep the development process organized.

3) *Github*: While GitLab serves as our primary development platform, GitHub complements our workflow by providing a strong collaborative environment for code review and open-source visibility. GitHub's large community and social coding features, such as pull requests, make it ideal for reviewing and managing contributions from multiple team members. GitHub's user-friendly interface and extensive integrations also make it an excellent choice for code sharing and feedback. Unlike GitLab, which focuses heavily on integrated DevOps workflows, GitHub excels in fostering collaboration and is widely used by developers for code management. By using GitHub, we ensure code quality and organization, which aligns well with our project's focus on creating a robust barcode payment app.

IV. SPECIFICATIONS

A. *Login page*

A login feature is needed to store the user's payment methods and purchase history. When the user enters a correct username and password and presses the login button, a login POST API request is sent to the server.

B. *Register page*

The username must be 2 to 20 characters long, using only English letters and numbers. The password must be a combination of English letters and numbers, with a length between 8 and 15 characters, and it should meet additional conditions such as an option to include special characters. Once registration is complete, the user's username and password are stored in the database.

C. *User information page*

On the user information page, users can change their username or password and navigate to the card and account registration page or the purchase history page.

D. *Card or account registration page*

On the card or account registration page, users can link a card or account to enable instant payment when scanning a barcode. Users can register multiple cards or accounts and select one of them at the time of payment. When a user registers a card or account, the information is saved in the database. For future payments, the payment gateway API is integrated to send payment requests and receive approval or rejection responses. When storing card numbers or account information, encryption technology is always used to protect sensitive information, and security is reinforced for login sessions and API requests.

E. *Purchase history page*

On the purchase history page, users can view the items they have purchased so far, stored in the database. This information can be used to recommend items that may be of interest to the user.

F. *Barcode scanning page*

On the barcode scanning page, the application connects to the phone's camera, allowing users to take photos. Based on the photo of the barcode, it accesses the market database to retrieve the item's price, quantity, and type, which are then added to the shopping cart. When the user presses the checkout button in the cart, the payment is processed through a pre-registered card or account.