QuikCart: A Mobile-Based Intelligent Real-Time Billing System for Smart Retail Automation with Administrative Verification

Dr. S.N. Sarda[1], Vishant Manwar[2], Chaitanya Mahajan[3], Simran Chaware[4], Shashank Jumade[5]
[1]*Assistant Professor (Corresponding/Supervising Author), Department of Information Technology, Prof. Ram Meghe Institute of Technology and Research, Badnera, Amravati, Maharashtra, India*
[2,3,4,5]*Department of Information Technology, Prof. Ram Meghe Institute of Technology and Research, Badnera, Amravati, Maharashtra, India*

***Abstract*—The traditional checkout systems used in the retail sector are no longer suitable to address the needs of the contemporary consumer who desires to have a short waiting period and an easy time with the process of transacting. The need of centralized billing counters creates systemic bottlenecks such as long queues, high staffing needs, and high susceptibility to manual scanning errors. This paper presents QuikCart, a mobile based smart cart system that allows customers to scan items in the shopping cart and keep a virtual cart running in store, in effect transforming the billing experience into a simultaneous shopping experience. The system runs on Android with Kotlin and connects to a central MySQL database to retrieve the live product information and store the records of transactions. This is a unique aspect of QuikCart: scanning performed by customers is supported by a two-level architecture, where the administrative verification stage is obligatory, whereby the contents of a shopper cart are physically validated by the store personnel before the final bill is approved and printed. This architecture mitigates a major weakness in all-automated self-checkout deployments (the lack of tamper detection) without compromising on the efficiency advantages of mobile-led billing. A simulated retail setting where QuikCart was empirically tested shows that the system shortens the average checkout period by about 67 per cent, decreases the rate of billing errors by more than 90 per cent, and greatly decreases the number of people waiting at payment counters. The modular framework supports the future addition of digital payment support, cloud implementation, and fraud detection with machine learning. The suggested system is a viable and cost-efficient way to smarter retailing.**

***Index Terms*—Real-Time Billing; Mobile Self-Checkout; Retail Automation; Android Application; Kotlin; MySQL; Administrative Verification; Smart Retail**

I. INTRODUCTION

The retail business has been struggling with the low efficiency of the traditional checkout processes. Although there have been huge improvements in inventory management, supply chain analytics, and customer relationship system, the last point-of-sale transaction is often limited by queues, manual labor and processes that are error prone. Industry estimates show that the delays caused by the checkout process represent a significant percentage of negative in-store customer experiences, which has a direct impact on retention and revenue.

The availability of mobile computing and ubiquitous internet connectivity has led to situations where customers can take active responsibility in the process of the transaction. The scan-and-go apps, theoretically, provide a stylish solution: instead of standing in a line and having a cashier scan the products at a fixed point, shoppers scan the goods as they put them in their physical cart. Billing is almost done by the moment they are ready to leave. The real-life implementation of these systems is however complicated by the twin demands of convenience and security. A completely unsupervised scan-and-go system is vulnerable to intentional abuse, e.g., scanning an item of lower cost and putting a more expensive one in the cart, or to unintentional misdemeanors caused by scanning errors.

The current commercial applications, like the cashier-less stores of Amazon Go, are dealing with this by using costly and computationally-intensive sensor fusion and computer vision systems. These strategies

are not economically viable to most of the retail outlets, especially the small and medium size stores located in the third world countries. What is required is a system that will maintain the performance benefits of self-scanning, but will add a lightweight human-in-the-loop verification step that will thwart abuse without creating a big penalty.

In this paper, QuikCart, an intelligent real-time billing system that is specifically intended in this context, is presented. The proposed platform allows the customers to scan the barcodes of the products with their smart phones, dynamically adds and removes the items in a virtual cart, and coordinates all the activities with a database on the back end. The system sends a verification request to an administrative interface before the finalization of the checkout, and store staff is able to verify the contents of the cart with minimum effort. On approval, the system will generate and save the final invoice.

The main contributions of this work are the following:

- Mobile billing structure based on lightweight, affordable, mobile technology, which can be deployed in the traditional retail store without getting specialized hardware.
- A hybrid automation system balancing customer-controlled efficiency and administrative control, which decreases checkout response and billing error.
- A quantitative assessment that shows quantifiable checkout time, error rates, and queues are reduced compared to the traditional billing methods.

The rest of this paper has the following structure. Section II is a literature survey. Section III outlines the system architecture and system methodology. Section IV discusses and analyzes the experimental findings. The implications and limitations are in Section V. Section VI provides the conclusion of the paper and the future work directions.

II. RELATED WORK

The literature on automated and self-service checkout systems is diverse in terms of the technologies and the deployment setting. The gap that QuikCart fills can be seen by a short overview of the representative previous work.

A. Self-Checkout Kiosk Fixed.

Retailers like Walmart, Tesco or Kroger have introduced self-checkout kiosks to enable customers to scan their items at a special terminal and pay without the involvement of a cashier [1]. Although these systems are less expensive in terms of labor, they lack the capability to be flexible in the way they are placed, and they also generate weight-check false alarms and require the supervision of the attendant. Most importantly, they do not remove the formation of queues since the customers still flock a predetermined amount of kiosk stations upon the completion of their shopping.

B. RFID-Based Systems

Radio-frequency identification (RFID) technologies involve placing passive tags onto every item and scanning them at exit gates of the stores to automatically record the items in the shopping cart of a customer [2]. This technique is fast and very precise however, it is costly in terms of tagging infrastructure and it is susceptible to shielding or interference. RFID is also not practical when the commodity goods have low margins, due to the cost per-unit tagging.

C. Computer Vision and Sensor Fusion.

The cashier-less idea behind Amazon Go will use ceiling cameras, shelf weight sensors, and deep learning algorithms to observe product interactions without customer involvement [3]. The resulting experience is frictionless, but the infrastructure cost, complexity of operation, and reliance on large labeled training datasets make this strategy unavailable to most retailers.

D. Mobile Scan-and-Go Applications.

A number of retail stores have installed companion mobile app that allows clients to scan barcodes and make payments through the app. Research has shown that the perceived convenience of these applications is enhanced, but questions remain regarding the compliance rate of scanning without verification and theft rates [4]. Bhatt et al. [5] showed that average checkout time was reduced by 4055% with mobile self-scanning in a grocery setting but reported that shrinkage increased by about 3-5 percent as compared to staffed checkout.

E. Positioning of QuikCart

The above systems surveyed could be broadly grouped on two axes i.e. (i) level of automation and (ii)

existence of human checks. Full-autonomic systems (e.g., Amazon Go) are the most convenient but prohibitively expensive and do not explicitly discourage the problem of fraud in case of less serious retail settings. Partially automated kiosk systems save human hand but does not decentralize the billing beyond the checkout counters. Mobile scan-and-go apps are also efficient, but generally based on probabilistic receiver auditing, as opposed to a systematic verification.

QuikCart is in a unique space, with its in-aisle, decentralized, billing using a smartphone camera and requiring a lightweight administrative verification process before bill generation. Such a hybrid model offers almost equivalent efficiency improvement over completely automated systems without being a significant impediment to both unintentional and intentional billing errors. These differences can be summarized in the table below.

Table I: Comparative Overview of Self-Checkout Systems

System	Technology	Admin Check	Real-Time DB	Self-Billing
Amazon Go	Computer vision, sensors	No	Yes	Yes
Walmart SCO	Fixed kiosk, barcode	Partial	Yes	Limited
RFID-based systems	RFID tags	No	Yes	Yes
Mobile POS apps	Mobile/barcode	No	Varies	Yes
QuikCart (Proposed)	Kotlin, MySQL, Camera	Yes	Yes	Yes

SCO = Self-Checkout; QuikCart row reflects the proposed system.

III. SYSTEM ARCHITECTURE AND METHODOLOGY

This is the section which provides the design rationale, architectural elements, development technologies and operational workflow of QuikCart at an adequate level to allow replication to be made.

A. Architectural Overview

QuikCart is based on a three-layer client-server model that consists of: (1) a customer facing mobile application (Android), (2) administrative verification panel, and (3) centralized MySQL relational database. The calls between the three tiers are by means of the RESTful API and all the product and transaction data are stored in the backend database to provide a single source of truth.

Customers: the customer application operates on any modern Android and reads product barcodes using the inbuilt camera. When a scan is found to be successful, the application sends a query to the database through the application server to obtain the name of the product, the category of the product and the price per unit and the stock of the product available. The item retrieved gets added to the session cart of the customer which is kept in the offline resilient local storage and synchronized with the backend. A notification is sent to the admin panel when a customer places an order of checking out and shows the contents of the cart to verify the order.

B. Technology Stack

Kotlin (Android): The mobile app is written using Kotlin, the default language to Android development. Kotlin is a null-safe language with correlated concurrency and terse syntax, which minimises runtime exceptions and code size compared to Java-implementations.

MySQL: A relational schema will store four key tables which are Products (product id, name, category, price, stock quantity), Users (user id, credentials, session token), Cart Sessions (session id, user id, product id, quantity, scan time), and Transactions (txn id, session id, total amount, status, time). Normalized schema design provides data integrity and high-performance queries.

Android SDK: The Android SDK uses the Camera2 API and the ZXing barcode decoding library which provided good barcode recognition in different lighting environments. Lifecycle management hooks maintain cart state through interruptions of applications.

RESTful Application Server: This is a slender server layer, which acts to connect the mobile application with the MySQL database, including authentication, input validation, and query parameterization, to forestall injection attacks.

C. Operational Workflow

The overall operation cycle includes six stages, which are explained in the following paragraphs and represented in the conceptual design of the system (Fig. 1).

1. User Authentication: When the customer starts using the application, he or she logs in with the credentials provided at the time of store registration. Session management (Tokens) -... This type of session management maintains authenticated state during the shopping session without having to input their credential again.
2. Product Scanning: The customer opens the in-app scanner and scans the camera on a product barcode. The decoded product ID will be sent to the application server that will send product details with a target latency of less than 500 ms. The product is screened to confirm and then it is committed to cart.
3. Cart Management: Virtual cart interface enables the customers to check the items added, to change the quantities or to delete the products. Cart changes are instantly reflected back in the back-end session record and the running total is updated after every change.
4. Checkout Request Submission: Once the customer finishes with his/her shopping, a checkout request is made. The system adds all the cart entries and sends them to the administration module along with the session time and customer identifier.
5. Administrative Verification: The checkout notification is sent to a store associate on the administrative panel and each of the scanned products is shown on it with quantity. The associate does a cursory physical examination of the basket to ensure the physical contents are what are listed. The verification process is expected to take a customer less than 90 seconds to verify a cart of 10-15 products. On confirmation, the associate approves the request.
6. Bill Generation and Transaction Closure: After the approval process, the system calculates the final invoice with tax added to it, records the transaction as closed in the database and transfers the bill to the customer application in a format of an on-screen overview as well as a receipt that can be downloaded.

D. Fraud and Security Prevention.

One of the common complaints raised about self-scanning systems is that they are prone to intentional under- scanning, product substitution, and misuse of scanning anomalies. QuikCart controls these risks in a number of ways. The administrative approval requirement is a physical barrier which makes systematic fraud impractical in the absence of collusion. Duplicate scan incidents are identified by comparing product ID and scan timestamp pairs in a session, a duplicate in a small-time interval will raise an alert in the admin interface. Session tokens have a time limit that runs upon idle time and thus a customer would not be able to scan the items in more than one session.

Admittedly, the step of the verification of the administration does not imply cryptographic enforcement, but rather depends on the care of the store staff. As new technology will be investigated in the future, the use of computer vision to facilitate this step without losing the deterrent effect will be discussed.

V. RESULTS

QuikCart was tested under a controlled environment which was designed to represent a mid-size retail store, including an estimated 200 different product SKU. A sample of twenty test subjects of different levels of smartphone proficiency was recruited to go through the standardized shopping processes in the traditional check out process and use of QuikCart application. All the participants were given five trials in each condition and this translated into 100 observations on a metric per condition.

A. Checkout Time

Mean check out time - the time taken since the final item was added to a basket to the time when a receipt was issued was decreased in the mean in the traditional condition of 9.4 minutes to 3.1 minutes with QuikCart, a 67% decrease. The difference in time at checkout was also significantly reduced in QuikCart (SD = 0.8 min) than in the traditional one (SD = 2.3 min) indicating a higher predictability of the customer experience. The length of the queue at maximum simulation intervals went down to 1.4 customers as opposed to 7.2 customers on average.

B. Billing Accuracy

Any difference between items in a basket of a

customer and those on the generated bill was counted as billing discrepancy. When the store was run on the old cashier system the variance rate stood at 4.1%. In QuikCart, the discrepancy rate was 0.3, and the rest resulted due to the breakdown of camera recognition with spoilt barcodes. The administrative check point blocked all deliberate misalignments that were made as controlled fraud simulation.

C. Scan Latency

The time spent to retrieve the product information between scan events and the display on the screen was used as a measure of product information retrieval time over 500 scan operations. The median latency was 320 ms; the 95th percentile was 480 ms; the maximum latency recorded was 640 ms with artificially increased load of simultaneous users. This shows that the system is within the sub-500 ms range of the median case and can also sustain high lead levels.

D. User Experience

According to post-trial questionnaires, the QuikCart interface was quite popular: 85% of respondents rated the app as easy or very easy to use; 78% said they preferred using QuikCart to future shop; 91% felt that the saving process was quicker; and 74% felt that the administrative verification process was not obtrusive, indicating that the security overhead did not have a significant negative effect on

Table II: Performance Comparison QuikCart vs. Traditional Billing System

Metric	Traditional System	QuikCart	Improvement
Avg. Checkout Time	8–12 min	2–4 min	~67% reduction
Billing Errors	3–5%	< 0.5%	~90% reduction
Queue Length	6–10 customers	0–2 customers	Significant
Admin Overhead	High	Low	Automated
Scan Latency	N/A	< 500 ms	Real-time

Values represent mean measurements from controlled evaluation trials (n = 100 per metric).

VI. DISCUSSION

Those findings confirm the central design hypothesis of QuikCart: moving the scanning operation to the shopping aisle and implementing a short administrative checkpoint yields significant efficiency improvements without any loss of billing integrity.

A. The worth of Hybrid Oversight.

The conflict between the effectiveness of automation and loss prevention is always cited in literature as the critical unsolvable issue of self-checkout systems [4, 5]. QuikCart does not solve this tension but restructures it. Instead of the bulk scanning labor which is the work of a cashier, a store associate does a short confirmatory scan on an already assembled cart. Such redistribution of labor is qualitatively different: a cognitive burden to bear by the associate is verification, not execution, which is quicker, less prone to error and can be easily scaled. Stores will be able to reassign most of their checkout counters to serve more QuikCart customers on the floor by employing them as floor-based verifiers.

B. Deployment Considerations

QuikCart is not based on any specific hardware, only the smartphones that customers already have. The backend infrastructure (MySQL database and a simple application server) may be deployed on a commodity cloud or on-premise infrastructure. The product onboarding process. This would only involve the conversion of existing

inventory records into the schema of the QuikCart database which is not complicated when the stores have electronic point-of-sale (ePOS) systems.

C. Limitations

It has a number of limitations. The test was done in a controlled simulation and not on an actual retail deployment; performance in the real life may be different since network variations, device heterogeneity and the customer behaviour is quite unpredictable. The administrative verification step assumes that the level of staffing is sufficient: understaffed stores may have a higher verification latency during peak hours. Also, there is no support at the moment of products which are not machine-readable barcodes (e.g. loose produce).

D. Scalability

The horizontal scaling is enabled by the modular isolation of the mobile application, application server and database. Database model replicas can be used to handle more query loads without changing the logic of the application; application server may be containerized and dynamically coordinated. Capacity management would be made even easier with a shift to cloud-hosted MySQL (e.g. Amazon RDS or Google Cloud SQL).

VII. CONCLUSION

In this paper, QuikCart which is an innovative mobile billing system has been discussed in which we have reconceptualized the retail check out system by facilitating continuous in-aisle scanning, real time cart control and administrative signature before the final billing. The proposed architecture shows that the important efficiency levels such as the decrease of the size of a checkout process by 67 and billing errors by 90 can be achieved with the help of the combination of mobile technology and systematic human control without attempting to involve the technology, which is prohibitively expensive to install.

The greatest contribution of QuikCart is that through its hybrid verification paradigm, it has solved the main weakness of currently deployed scan-and-go systems; the lack of an authoritative fraud deterrent. The system preserves the quality of customer experience by making the administrative verification a light verification point instead of a bottleneck, which offers significant verification of billing integrity.

Further growth is to be integrated with digital payment gateways (UPI, NFC-based contactless payment, and card processing), real-time inventory optimization, machine learning-based fraud detection, and a cloud-native deployment architecture to support multiple stores chains.

ACKNOWLEDGEMENTS

The authors would like to give their heart-felt thanks to Dr. S.N. Sarda, Assistant Professor, Dean of Information Technology, Prof. Ram Meghe Institute of Technology and Research, Badnera, Amravati as the mentor of this study and the corresponding author. The success of QuikCart and the creation of this manuscript was primarily due to his professional guidance, feedback in a timely manner, and unremitting encouragement.

The authors also acknowledge the faculty and staff of the Department of Information Technology PRMIT&R as having provided the required resources and a conducive environment of the research.

REFERENCES

[1] M. Braun and T. Fischer, “Self-service checkout systems in supermarkets: An empirical analysis of user acceptance and operational impact,” *J. Retailing Consumer Services*, vol. 48, pp. 107–116, 2019.

[2] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication*, 3rd ed. Chichester, U.K.: Wiley, 2010.

[3] Avery and S. Nguyen, “Cashier-less retail: A technical and economic assessment of sensor-based autonomous checkout,” *IEEE Trans. Consumer Electron. *, vol. 67, no. 2, pp. 145–153, May 2021.

[4] R. T. Hogan and A. Patel, “Mobile scan-and-go: Consumer adoption, behavioral patterns, and retail outcomes,” *Int. J. Retail Distrib. Manag. *, vol. 49, no. 3, pp. 390–408, 2021.

[5] S. Bhatt, P. Sharma, and V. Desai, “Quantifying shrinkage in mobile self-scanning retail deployments,” in *Proc. IEEE Int. Conf. Industrial Engineering and Engineering Management (IEEM)*, Singapore, 2020, pp. 832–836.