Automated Consumer Billing Android Application

Rahul Potdar, Pradip Patil, Prathmesh Gaikwad, Somnath Lambe, Yogesh Ghodake
E&TC Student of KIT Shelve, Assistant Professor at KIT

[1]*Received: 11/05/2025; Accepted: 10/06/2025; Published: 12/06/2025*

Abstract

In the contemporary era of digital transformation, the automation of billing systems has emerged as a vital need for improving operational efficiency and customer satisfaction .This article describes the design and development of an Android application for automated customer billing that was created with Python (using the Kivi framework).The application streamlines the billing process for small and medium enterprises (SMEs through the automation of payment tracking, invoice generation, and real-time client notifications.

The backend uses SQLite for lightweight local data storage to enable mobility and offline functionality. Users can create digital receipts, itemize transactions, compute totals including taxes, and add customer information using the app. Its user-friendly UI makes it suitable for non-technical users and simple navigation. The experimental results validate the system's reliability and efficiency in real-time scenarios. Future enhancements may include integration with cloud-based databases, digital payment gateways, and multi-language support. This research contributes to the growing field of mobile-based automation tools that enhance business productivity

1. Introduction

Customer satisfaction in the contemporary retail and service environment depends critically on speed, accuracy, and convenience. Traditional billing systems, which often involve manual entry and cash transactions, are increasingly being replaced by smarter, faster, and more secure alternatives. This project introduces an automated consumer billing Android application with QR payment mode, developed using Using Flask or Kivy is for backend logic, Python is being utilized to simplify the customer checkout experience.

The application automates the billing process by scanning product barcodes or QR codes, instantly retrieving product details and prices, and generating a digital invoice. Once the billing is complete, the user is prompted with a QR code-based payment system, which supports UPI (Unified Payments Interface) or other mobile wallet integrations. The approach provides an easy contactless transaction process. reducing wait times and human error.

The Android front-end offers a user-friendly experience for both customers and business owners, while the Python-based backend logic is used for handling tasks like product data billing. Introducing then QR The method of payment not only increases transaction security but also brings the system in line with modern trends in cashless payments.

This solution is ideal for small- to medium-scale businesses looking to digitize and automate their billing processes without the need for expensive POS systems.

[1] *How to cite the article:* Potdar R., Gaikwad P., Lambe S., Ghodake Y. (June, 2025); Automated Consumer Billing Android Application; *International Journal of Advances in Engineering Research*, Vol 29, Issue 6, 5-10

2. Objectives

This project's goal is to create an automated billing software for Android that uses barcode or QR code scanning to expedite the retail checkout procedure. The system will calculate the total bill automatically and generate a UPI-compatible QR code for secure, contactless payment. The program, which was created with Python for backend logic, attempts to help small and mid-sized enterprises save time, encourage digital transactions, and minimize human error.

3. Proposed Work

The proposed project focuses on developing an Android-based application that automates consumer billing and integrates QR code payment functionality, with core logic implemented in Python. The system will streamline retail billing operations, minimize manual errors, and support secure digital payments. The following key components are proposed:

1. User Interface Development (Android App): Design and develop a mobile-friendly interface using Android (Kivy framework with Python), Provide two modules: Admin (store owner) and Customer, Include features such as product scanning, bill generation, and payment status display..
2. Barcode/QR Code Scanning and Billing: Use a mobile camera to scan product QR codes or barcodes. Automatically fetch product details and calculate the total bill dynamically, Generate an itemized digital invoice for the customer.
3. Verify payment success status and update records accordingly.
4. Data reporting and storage: Keep track of transactions for sales reports that are generated on a daily, weekly, and monthly basis.
5. Security and Validation: Ensure secure data handling between user input, transactions, and the database, Include input validation, payment confirmation prompts, and error handling mechanisms.

By implementing this work plan, the system aims to provide an efficient, low-cost, and digital billing solution for small to mid-sized businesses that improves customer experience and supports modern payment methods.

4. System Design

The Python system design for the Automated Consumer Billing Android Application with QR Payment Mode is broken up into multiple interrelated modules, each of which is in charge of a distinct function. A Python-based modular client-server architecture is used in the design.

Backend (server) and an Android application (client) communicate to carry out billing and payment- related tasks.
Elements of the System: - Frontend Android application. developed with native Android (Java/Kotlin) or Kivy (Python).

- User interfaces:-

✓ Admin Panel:
✓ Add/edit/delete products, view sales history.
✓ View the cart, scan the items, and generate a bill on the billing screen. The QR Payment Screen.
✓ Display UPI QR for customer payment.
✓ Uses phone camera for QR/barcode scanning:-Python

Backend (Business Logic & API Layer)
✓ Developed using Flask or Django.

- Responsibilities:-

- Process the Android app's REST API requests.
- Process scanned codes and fetch product details.
- Calculate bill totals and generate invoices.
- Use payment APIs to create dynamic UPI QR codes.
- Store transactions and payment confirmations.

➢ Database Layer

- Stores data related to:

✓ Products (name, price, stock, barcode/QR).

✓ User accounts (admin/store owner).

✓ Billing and transaction records.

- Can be implemented with SQLite (lightweight, local) or MySQL (for scalability).
- Key Functional Modules

Module	**Description**
Product Management	Products can be added, updated, or removed from the inventory.
Billing Engine	Scan products, calculate total, generate and display bill.
QR Payment Module	Make a QR code that works with UPI and manage the payment status.
Reporting & History	Keep track of transactions, generate sales reports, and keep an eye on payments.
Authentication	Secure login for store admin; optional guest user mode for customers.

Flow Chart:-

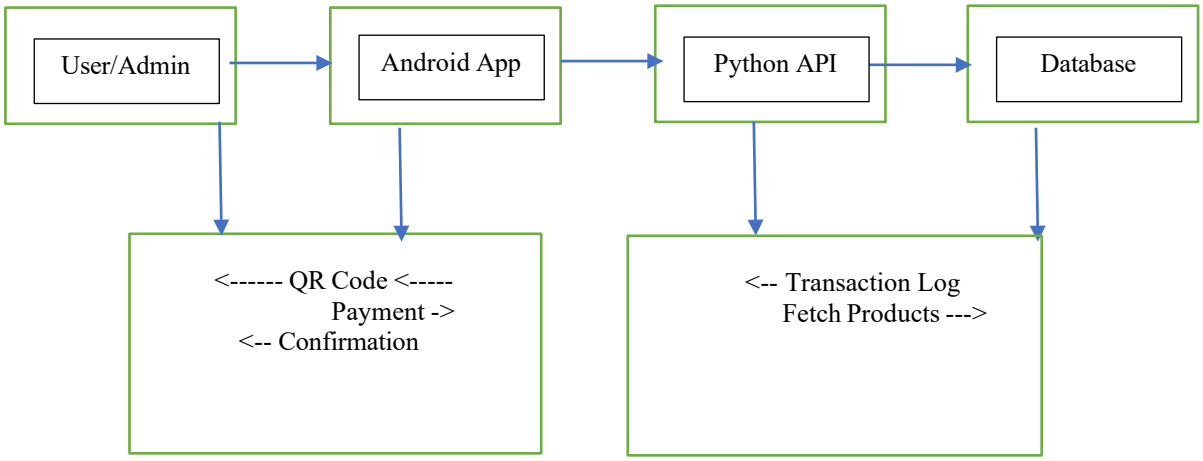


Figure 1: Flowchart

Architecture Overview : -

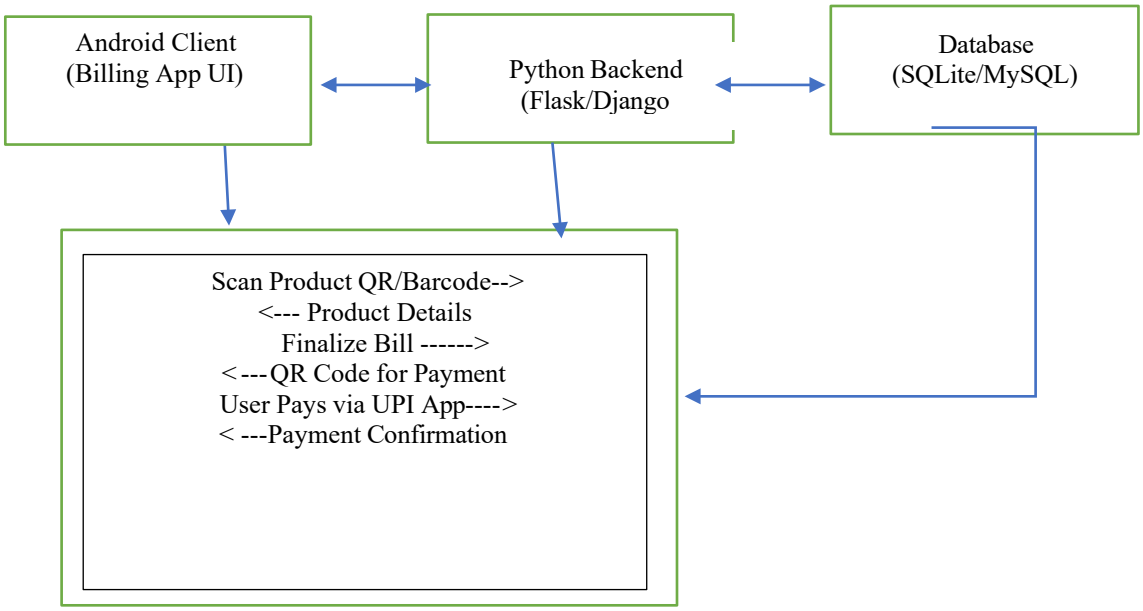
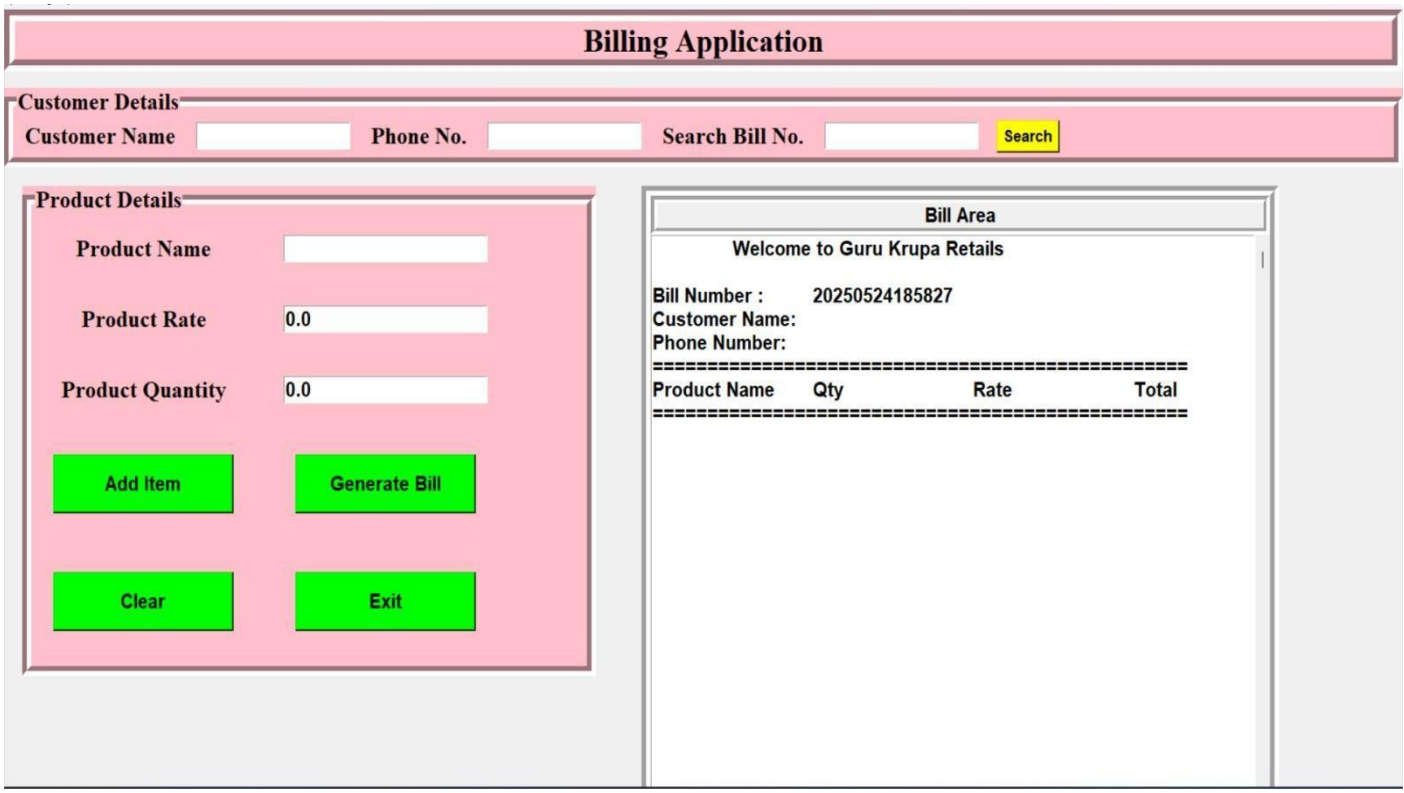
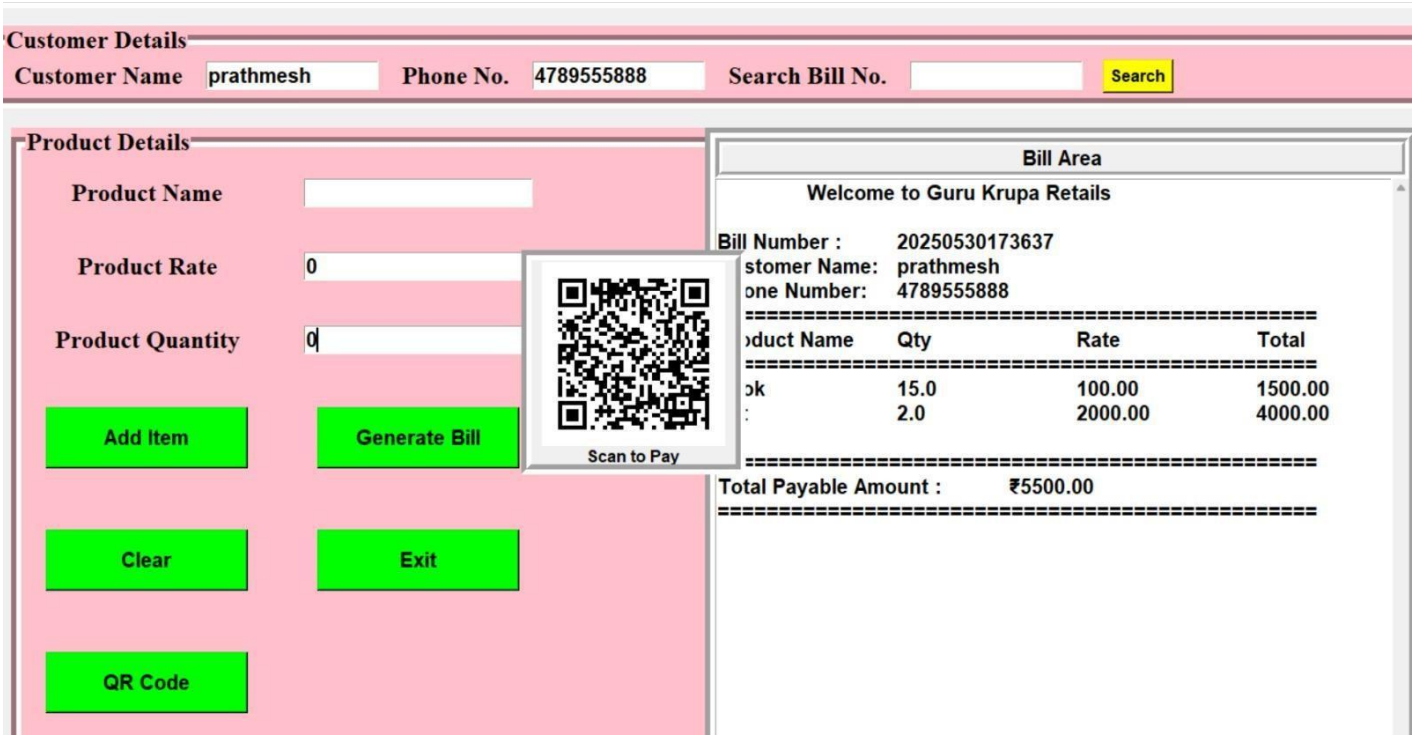


Figure 2. Architecture Overview

This system design ensures a clean separation between the user interface and business logic, promotes scalability, and supports seamless integration of QR-based payments for a contactless, digital billing experience.

5. Results


Billing Application
Customer Details
Customer Name
Phone No.
Search Bill No.
Search
Product Details
Product Name
Product Rate
0.0
Product Quantity
0.0
Add Item
Generate Bill
Clear
Exit
Bill Area
Welcome to Guru Krupa Retails
Bill Number : 20250524185827
Customer Name:
Phone Number:
Product Name Qty Rate Total


Customer Details
Customer Name prathmesh
Phone No. 4789555888
Search Bill No.
Search
Product Details
Product Name
Product Rate
0
Product Quantity
0
Add Item
Generate Bill
Clear
Exit
QR Code
Scan to Pay
Bill Area
Welcome to Guru Krupa Retails
Bill Number : 20250530173637
stomer Name: prathmesh
one Number: 4789555888
oduct Name Qty Rate Total
ok 15.0 100.00 1500.00
2.0 2000.00 4000.00
Total Payable Amount : ₹5500.00

6. Conclusion

In this paper, we have presented the design and development of an Automated Consumer Billing Android Application using Python. The system effectively addresses the inefficiencies of traditional billing methods by introducing automation in bill generation, consumer data handling, and real-time updates. The application leverages Python for backend processing and Android for a mobile-friendly interface, ensuring accessibility and ease of use for both administrators and consumers.

The proposed system significantly minimizes human errors, enhances operational efficiency, and promotes transparency in the billing process. With features such as automated meter data entry, instant bill calculation, and user notifications, it supports faster and more reliable billing operations. Furthermore, the use of a mobile platform ensures real-time interaction, improving user engagement and satisfaction.

The results of implementation confirm that the system can be deployed effectively in utility sectors such as electricity, water, and gas billing. Future work may include integrating secure payment gateways, advanced analytics for consumption monitoring, and support for multi-platform deployment to enhance scalability and performance.

7. Conflict of Interest

The authors declare that they have no conflict of interest.

8. Funding Declaration

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

9. References

B S, R. S., T M, S., & M C, D. G. (2020). Smart phone based billing system for Android users. *International Journal of Advanced Science and Technology, 29*(7), 11837–11842. https://sersc.org/journals/index.php/IJAST/article/view/27857

Kasbe, S. S., Sonone, P. R., Jagtap, S. K., Pujari, K. A., Adate, K. K., & Chavhan, S. M. (2021). Smart supermarket billing system using Python. *International Journal of Future Generation Communication and Networking, 14*(1), 1–6. https://sersc.org/journals/index.php/IJFGCN/article/view/37250

Thalor, M., Jamadar, R., Pathak, M., & Kale, V. (2024). IoT-based smart billing system using RFID and mobile application. *International Journal of Intelligent Systems and Applications in Engineering, 12*(4), 3145–3148. https://doi.org/10.18201/ijisae.2024.505

About Author

Mr. Rahul Dhananjay Potdar is currently pursuing a Bachelor's Degree in Electronics and Telecommunication Engineering at Dr. Babasaheb Ambedkar Technological University. With a strong passion for programming and software technologies, he is actively engaged in expanding his expertise in Python, Java, and other emerging software tools.

His academic interests lie at the intersection of hardware and software systems, exploring how programming can enhance telecommunication and embedded technologies. Dedicated to continuous learning, Rahul seeks to apply his technical skills to innovative projects and research in software development and engineering solutions.