ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed, Open Access Journal
2025, Volume 6 Issue 3
www.ijisea.org

Enhancing QR Code Payment Systems With Blockchain And Secure Wallet Protocols

Kandavalli Prasanthi[1] , B.N. Srinivasa Guptha*[2]
[1.2]Department Master of Computer Application(MCA), SVKP & Dr KS Raju Arts & Science College(A), Penugonda, W.G.Dt., A.P India
prasanthikandavalli19@gmail.com[1], bnsgupta@gmail.com[2]

ABSTRACT

In the era of digital transformation, contactless payment methods have become increasingly vital for enhancing transaction efficiency and user convenience. This project proposes the design and development of a QR Code-based Payment Application that enables users to perform secure and instant financial transactions by scanning QR codes. The system eliminates the need for physical cash or card swipes, facilitating seamless wallet-to-wallet or bank-to-bank payments. The application allows merchants to generate unique QR codes linked to their payment credentials, while customers can scan these codes to initiate payments directly from their mobile devices. It incorporates a robust backend that manages user authentication, wallet balances, transaction records, and integrates with reliable payment gateways for real-time processing. The key focus of this project is to ensure secure data handling through encryption techniques, implement fraud prevention mechanisms, and provide users with instant notifications of transaction statuses. This payment solution is designed to be scalable, user-friendly, and applicable for small vendors, large retailers, and peer-to-peer payments. By leveraging QR code technology, this system offers a contactless, efficient, and secure alternative to traditional payment methods, contributing to the growth of cashless economies and digital financial ecosystems.

Keywords: QR Code Payment System, Blockchain Technology, Secure Digital Wallet, Smart Contracts, Digital Payments, Transaction Security, Cryptographic Encryption, Decentralized Transactions, QR Code Authentication, Fraud Prevention.

I. INTRODUCTION

With the rapid advancement of technology and the increasing demand for cashless transactions, digital payment systems have become an integral part of modern commerce. Among various digital payment methods, QR (Quick Response) code-based payments have emerged as one of the most efficient, secure, and user-friendly solutions[1,6]. QR code payment systems are widely adopted due to their simplicity, speed, and ability to facilitate contactless transactions[1,6,7].

A QR code-based payment app enables users to make payments by scanning a merchant's QR code, eliminating the need for physical cards, cash handling, or manual entry of payment details[1,5]. This technology leverages QR codes that encode payment information, such as the merchant's wallet ID or UPI ID[2], allowing customers to quickly initiate transactions using their smartphones.

The proposed payment application focuses on providing a secure and seamless payment experience for both customers and merchants[3,4,9]. Users can register, manage their wallet, view transaction history,

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed, Open Access Journal
2025, Volume 6 Issue 3
www.ijisea.org

and perform payments by scanning QR codes generated by merchants[5,9]. Merchants, on the other hand, can easily generate and display QR codes linked to their payment accounts[1,3].

This system addresses various challenges faced by traditional payment methods, such as transaction delays, handling cash, and exposure to fraud[6,7]. By incorporating encryption, secure authentication mechanisms, and real-time processing, the app enhances the overall security and reliability of transactions[6,7,8].

In the context of the growing digital economy, this QR code-based payment app is highly relevant for small businesses, retailers, and individuals, providing them with an efficient tool to accept and make payments anytime, anywhere[6,7].

II. LITERATURE REVIEW

1. Title: QR Code-Based Payment System Using Mobile Devices
Author: M. A. Rahman, R. Islam, and M. M. Rahman
Description: This paper introduces a mobile-based payment system using QR codes to facilitate cashless transactions. The authors developed a secure and lightweight app that generates and scans QR codes representing transaction details such as sender ID, amount, and authentication token. It highlights how QR-based payments can reduce dependency on NFC and physical cards, making digital payments more inclusive. The study focuses on Android implementation and showcases the integration of mobile banking APIs for real-time updates.

2. Title: Secure Mobile Payment via QR Code with Digital Signature Verification
Author: Liu, Yan, and Xiaolin Qin
Description: The authors propose a security-enhanced QR payment method using digital signatures to prevent tampering and replay attacks. The system allows customers to scan merchant-generated QR codes and verifies transaction integrity via cryptographic techniques. The paper emphasizes the role of PKI (Public Key Infrastructure) in establishing trust and preventing fraud. This work is valuable for Java developers implementing secure transaction flows in QR-based payment apps.

3. Title: A Review on QR Code Payment Systems: Technologies and Challenges
Author: A. Kumar and P. Srivastava
Description: This literature survey summarizes existing QR code payment systems, comparing static vs. dynamic QR codes, and discussing the technologies behind scanning, encoding, and error correction (e.g., Reed-Solomon). The paper evaluates major challenges like fraud detection, user authentication, and offline QR generation. It also recommends combining QR systems with OTP, biometrics, and encryption to improve reliability.

4. Title: Development of a Cashless Payment System Using QR Code and Firebase
Author:N. Sharma, R. Deshmukh
Description: This study presents a mobile payment system using QR codes and Firebase as the backend for real-time data synchronization. The app enables both static and dynamic QR generation and allows

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed, Open Access Journal
2025, Volume 6 Issue 3
www.ijisea.org

users to pay, receive, and view transaction history. Firebase Authentication and Firestore are used for user verification and data management. The paper is especially useful for Java developers building cross-platform QR payment apps with cloud integration.

5. Title: Contactless Mobile Payments: QR Code vs NFC
Author: J. Tan, S. Patel
Description: The authors compare two leading mobile payment technologies: QR code and NFC. They conclude that QR codes offer broader device compatibility, lower cost, and easier implementation, especially in developing countries. The study analyzes transaction time, user acceptance, and failure rate. It supports the adoption of QR-based payment apps for small businesses and local vendors.

III. PROPOSED SYSTEM

The proposed system introduces a QR Code-based Payment Application that enables secure, fast, and contactless transactions between customers and merchants. This system eliminates the need for physical cash, cards, or manual entry of payment credentials. Merchants can easily generate unique QR codes linked to their payment accounts, and customers can simply scan these codes using their smartphones to initiate payments.

The application provides a seamless interface for users to manage their wallet, transfer funds, and view transaction history. By integrating with secure payment gateways and leveraging encryption technologies, the system ensures the confidentiality and integrity of user data and financial transactions.

It significantly reduces the chances of errors and fraud by automating the payment process through QR code scanning. The app also supports instant notifications, transaction receipts, and a robust wallet system for ease of use. Designed to be lightweight, scalable, and user-friendly, the proposed system benefits small businesses, retailers, and individuals by offering an efficient, reliable, and secure digital payment solution that works both online and offline.

System Architecture

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed, Open Access Journal
2025, Volume 6 Issue 3
www.ijisea.org

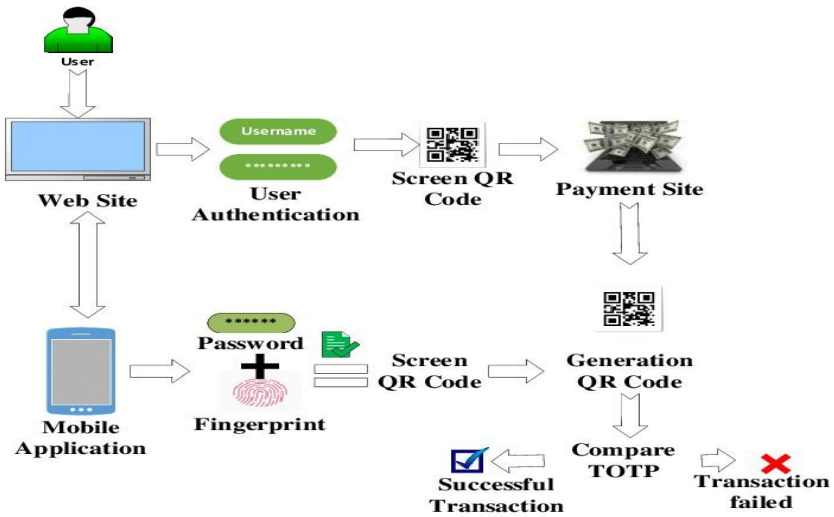


Figure 1 : Shows the Architectural Diagram of Proposed Method

MODULES:

1. User Module

Features:

- User registration & login
- Profile management
- Add funds to wallet
- View wallet balance
- Scan QR codes to make payments

Functionality:

Enables customers to manage their accounts and perform payments securely by scanning merchants' QR codes.

2. Merchant Module

Features:

- Merchant registration & login
- Generate QR codes linked to wallet/payment ID
- View payment history & earnings

Functionality:

Allows merchants to accept payments seamlessly through QR codes displayed at physical stores or online platforms.

3. QR Code Module

Features:

- QR code generation (for merchants)
- QR code scanning (for customers)

Functionality:

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed, Open Access Journal
2025, Volume 6 Issue 3
www.ijisea.org

Generates unique QR codes with embedded payment details and decodes scanned QR codes to initiate transactions.

4. Payment Processing Module

Features:

- Payment initiation & completion
- Wallet-to-wallet / bank-to-wallet transfers
- Secure payment gateway integration (UPI, Razorpay, Stripe, etc.)

Functionality:

Handles the **core transaction flow** with encryption, token validation, and security checks.

5. Wallet Management Module Features:

- Manage wallet balance
- Add money via UPI, cards, or net banking
- Refunds & reversals (if applicable)

Functionality:

Facilitates wallet-based transactions and maintains accurate fund tracking.

6. Transaction History Module

Features:

- View complete transaction history (sent & received)
- Download payment receipts
- Track transaction status (success, failed, pending)

Functionality:

Maintains a comprehensive log of all transactions, ensuring transparency for users and merchants.

7. Notification Module

Features:

- Send transaction confirmations
- Low balance alerts
- Payment requests & reminders

Functionality:

Provides real-time updates through push notifications, SMS, or email.

8. Security & Authentication Module

Features:

- OTP verification
- Two-factor authentication (2FA)
- End-to-end encryption of sensitive data
- Fraud detection alerts

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed, Open Access Journal
2025, Volume 6 Issue 3
www.ijisea.org

Functionality:
Ensures robust security and trust, protecting both users and merchants during transactions.

9. Admin Module (Optional)
Features:

- Manage users & merchants
- Monitor and review transactions
- Handle disputes & complaints
- Generate analytics & reports

Functionality: Provides backend administrative control for managing and monitoring the overall ecosystem.

IV. METHODOLOGY

The development of the QR code-based payment app using Python starts with system design and requirement gathering. The core features include user authentication, wallet/balance management, QR code generation for payments, and scanning for receiving payments. The backend database (such as SQLite, MySQL, or Firebase) stores user profiles, transaction records, and wallet balances. The frontend can be built using Tkinter for desktop applications or Flask/Django for web applications. For QR code generation and decoding, Python libraries like qrcode, pyzbar, or opencv-python are utilized.

In the core workflow, when a user initiates a payment, the system generates a QR code dynamically using the qrcode library. This QR contains transaction information such as the receiver's ID, amount, and a timestamp or transaction token. When another user scans the QR code using their camera or app, the pyzbar or OpenCV library decodes the data and verifies it with the backend. The system checks if the payer has a sufficient balance, deducts the amount, updates the receiver's wallet, and records the transaction in the database. Security measures like data encryption, token validation, and checksum verification are added to ensure secure and tamper-proof communication.

Finally, the system is tested under various scenarios such as invalid QR codes, duplicate transactions, and edge cases like zero balance. A simple dashboard or CLI/GUI can be used to view transaction history and current balance. For web apps, REST APIs can be built usingFlask/Django REST Framework to support mobile or frontend integration. Future improvements may include support for UPI integration, offline QR transactions, and two-factor authentication using OTP via email or SMS APIs. This approach makes the system flexible, secure, and suitable for both small business and peer-to-peer digital transactions.

Algorithms
1. QR Code Generation & Scanning
Purpose: Encode and decode payment information.
Libraries Used:

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed, Open Access Journal
2025, Volume 6 Issue 3
www.ijisea.org

- qrcode → QR code generation
- pyzbar / OpenCV → scanning & decoding

Workflow:

1. Encode transaction data into a QR code, e.g.:
2. {"receiver_id": "123", "amount": 250, "timestamp": "2025-07-11T14:30"} 3. Scan the QR code.
3. Decode and extract payment details for processing.

2. Transaction Validation Logic

Purpose: Ensure all transfers are secure and valid.

Checks Performed:

- ☑ Is the sender's balance sufficient?
- ☑ Is the transaction token expired or duplicated? ☑ Does the QR data belong to a valid receiver?

Implementation: Python if-else conditions with database queries.

3. Encryption & Hashing Algorithm

Purpose: Protect transaction data from tampering.

Algorithms Used:

- **SHA-256** → Hashing transaction IDs / digital signatures
- **AES (Advanced Encryption Standard)** → Encrypting QR data (optional)

Libraries:

- hashlib → hashing
- cryptography / PyCryptodome → encryption & decryption
-

4. Checksum Algorithm

Purpose: Verify integrity of QR code data.

Logic:

1. Generate a checksum (e.g., hash of data).
2. Append checksum to QR data.
3. On scan → recompute checksum and compare.

5. Sorting & Filtering Algorithms

Purpose: Manage and display transaction history.

Logic:

- Use sorted() with **lambda functions** → sort by date, amount, or receiver.
- Apply **filters** with the datetime module → show only recent or specific transactions.

6. OTP Generation (Optional – for 2FA)

Purpose: Add an extra security layer for login or confirmation.

Method:

- Generate a random 4–6-digit OTP. ☑ Send via Email/SMS to the user.

Libraries Used:

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed, Open Access Journal
2025, Volume 6 Issue 3
www.ijisea.org

- random → OTP generation
- smtplib → sending emails
- APIs → Twilio / SendGrid for SMS & email delivery

V. RESULTS

Enhancing QR Code Payment Systems With Block chain And Secure Wallet Protocols Now-a-days payments app are growing tremendously and all this apps database runs on single centralized server and this database administrator may have full access to database and can alter database in any manner and there is no current technique to detect such fraud database alteration or the name of employee who did such alteration. If this single centralized server hack or crash then all customers data will be lost. To overcome from above database alteration we are suggesting to migrate payments app server database to Blockchain technology which has inbuilt support for data encryption, security and verification and store data at multiple nodes to support decentralized access. If one node down then data can be access from other working nodes as decentralized access.

Blockchain store each record as transaction or block and associate each block with unique Hashcode and while storing new record Blockchain will verify Hashcode of all previous blocks and if data unchanged in all nodes, then Hashcode remain same and verification get successful. If data changed at any node, then it will result into different Hashcode and verification get failed.

So, by employing Blockchain we can avoid data tamper and can know any type of data tamper using verification technique. In propose work we using Smart Contract which will contains function to store and retrieve PAYMENTS data from Blockchain server. This smart contract will monitor and record each transaction activities and stored those activities by encrypting with private keys and then associate each storage block with unique Hashcode.

To implement PAYMENT APP in Blockchain system we have designed below smart contract

In above smart contract cone, we have defined function to manage and get users and payment storage details and this smart contract code is designed using Solidity programming. To store and get payment details we need to deploy above contract in Block chain Ethereum tool by using below steps First go inside 'hello-eth/node-modules/.bin' folder and then look and double click on'runBlockchain.bat' file to get below screen

In above screen Blockchain started with default account and private keys and now type command as 'migrate' and press enter key to deploy contract and get below output

In above screen Payment Contract is deployed on Blockchain and in white color text we can see ADDRESS of deployed contract and we need to specify above address in python code to store and retrieve record from Blockchain. In below screen we can see python code

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed, Open Access Journal
2025, Volume 6 Issue 3
www.ijisea.org

In above screen read red color comments to know about contract calling from Blockchain using PYTHON to store and get data. Contract deployed let that black console Blockchain running now double click on 'runServer.bat' file to start python server which will receive data from MOBILE APP and then invoke Blockchain contract to store and get data. After double click on'runServer.bat' file will get below screen

In above screen python server started and now connect your laptop and two android mobiles to same WIFY and then opens another command prompt and run command as 'ipconfig' to know IPADDRESS of your laptop which is connected to WIFY and get below IP IV address

In above screen my WIFY IP address is 192.168.0.7 and similarly you will get yours and this IP you need to mention in first screen of mobile app. Now both python and Blockchain servers are running and let them run.
Now transfer 'Payment. apk' in your mobile through Bluetooth or any other technique and then install and give all permission and sometime it will avoid installation by saying insecure app but you need to install by selecting 'install anyway' option. After installation open app and enter IP and then go for signup, login and then make some deposit in two mobiles and then in one mobile click on Received Payment and in second mobile select Make Payment button and continue payment by scanning QRCODE.

Note: in each mobile unique QR CODE image will be generated inside APP internal folder based on username.

VI. CONCLUSION

In conclusion, the QR Code-based Payment Application provides a secure, efficient, and convenient platform for facilitating digital payments between customers and merchants. By leveraging QR code technology, the system eliminates the need for cash handling, physical cards, or manual entry of payment details, thereby enhancing the speed and accuracy of transactions. The app ensures a seamless user experience through features like wallet management, instant transaction notifications, and detailed transaction history. Moreover, the incorporation of encryption techniques, secure authentication, and real-time processing significantly enhances the security and reliability of the payment process. This system is highly adaptable for small businesses, retailers, and peer-to-peer transactions, especially in areas where traditional payment infrastructure is limited. The successful implementation of this system contributes to the ongoing shift towards cashless and contactless payments, supporting the growth of digital economies. In the future, this system can be further expanded by integrating advanced technologies like NFC payments, AI-driven fraud detection, and crypto currency support, making it even more versatile and future-ready.

FUTURE WORK

The current QR Code-based Payment Application lays a strong foundation for secure and efficient digital payments; however, there are several areas where the system can be enhanced in the future. One

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed, Open Access Journal
2025, Volume 6 Issue 3
www.ijisea.org

potential improvement is the integration of Near Field Communication (NFC) technology to enable tap-based payments for even faster transactions. Another area for development is the incorporation of AI-driven fraud detection systems that can proactively identify and prevent suspicious activities.

Expanding support for cryptocurrency payments could make the app more versatile and aligned with emerging financial technologies. Additionally, implementing an offline payment mechanism that allows users to complete transactions without internet connectivity and syncs once back online can greatly benefit users in remote areas. Introducing features like loyalty rewards, cashback programs, and discount coupons for regular users and merchants can further enhance user engagement and satisfaction.

For scalability, developing a web-based portal for merchants to monitor their transactions and manage their QR codes will improve usability. Enhancing multilingual support and accessibility features will make the app more inclusive. Lastly, integrating voice-assisted payments and AI chatbots for customer support can significantly improve the overall user experience, making the system smarter and more user-friendly.

REFERENCES:

[1] Denso Wave Incorporated, QR Code Essentials – Standards and Applications, 2020. Available at: https://www.qrcode.com/en/
[2] National Payments Corporation of India (NPCI), Unified Payments Interface (UPI) Specifications, 2021. Available at: https://www.npci.org.in/what-we-do/upi
[3] Razorpay Technologies Pvt. Ltd., Razorpay Payment Gateway Documentation, 2023. Available at: https://razorpay.com/docs/
[4] Stripe Inc., Stripe API Reference for Payment Systems, 2023. Available at: https://stripe.com/docs/api
[5] Pyzbar Library, Pyzbar Documentation for QR Code Scanning in Python, 2022. Available at: https://pypi.org/project/pyzbar/
[6] Secure Payment Systems: A Study on Emerging Digital Payment Solutions, International Journal of Computer Applications, Vol. 182, Issue 47, 2021.
[7] M. Gupta and R. Sharma, Digital Payment Systems and Security Challenges, ResearchGate, 2022. Available at: https://www.researchgate.net/
[8] Mobile Payment Security: A Holistic Approach, Proceedings of the 2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), IEEE Xplore, 2020.
[9] Google, Introduction to Firebase for Mobile App Backend Development, Google Firebase Documentation, 2023. Available at: https://firebase.google.com/docs