A SECURE TICKETING SYSTEM FOR IN-PERSON EVENTS

TEAM 17

FINAL REPORT

Safura Mohammed Arif (220548168)

Xiaoyue Zhang (220005373)

Prabhu Yogesh Vijayan (220611613)

Olomu Atinuke Praise (220588762)

Kailash Balachandiran (220243160)

Roshan Vadlapatla Kiran Naga Sai Satya (220486435)

Lanjian Huang (200879622)

CHAPTER NO.	CONTENTS	PAGE NO.
	ABSTRACT	**1**
1	**INTRODUCTION**	**2**
	1.1 Background	**2**
	1.2 Team Approach to The Design and Design Decisions	**2**
2	**SYSTEM ARCHITECTURE**	**3**
3	**IMPLEMENTATIONS**	**5**
	3.1 User End	**5**
	3.2 Ticket Checker End	**5**
	3.3 RSA For QR Code Security	**6**
	3.4 Keystore and Transport Security	**7**
	3.5 Additional Security Features	**7**
4	**EVALUATIONS**	**8**
	4.1 Performance Metrics	**8**
	4.2 Limitations of the System	**9**
	4.3 Threats and vulnerabilities	**9**
	4.4 Future or Possible Enhancements	**9**
5	**CONCLUSION**	**10**
6	**INDIVIDUAL CONTRIBUTION**	**10**
	APPENDIX	**12**

ABSTRACT

In-person events such as football matches, music events, etc. require digital ticketing applications, as these systems have become popular for providing users and organizers with a convenient ticketing system. Booking tickets and verifications can be done using smartphones, which improves customer satisfaction and the ticketing process. However, such systems could face some security issues, such as counterfeiting, theft, and fraud, which could be carried out by malicious insiders or hackers.

These types of systems reduce waiting time for booking and verifying tickets. A secure ticketing system has been designed and implemented for in-person events to avoid various security threats related to ticketing systems. The system allows users to register, choose an activity, enter their details, and generate a QR code quickly and easily. The Quick Response (QR) code, with additional security measures, is used to validate the generated tickets in the application.

The Android application is used to show the implementation of the security features. The Android Studio is used as the development environment, with Java programming language. The Cipher Block Chaining (CBC) mode in Advanced Encryption Standard (AES) with Rivest-Shamir-Adleman (RSA) is used in the system. Features like anti-screenshot functions have been implemented to prevent sharing the generated ticket. The Firebase Realtime Database is used to store the required data.

The valid ticket is then verified, and the ticketing system employs a cloud-based database powered by Google Fire store to provide secure storage for user data and ticket information.

1 INTRODUCTION

1.1 Background

Many of the current ticketing systems use paper-based or physical tickets, which are easily lost or duplicated. The manual ticket verification required by these systems at the entrance points can be time-consuming and liable to human error [1]. However, conventional ticketing systems do not have real-time tracking or reporting capabilities, which makes it challenging to manage attendance and spot fraudulent activity.

Many ticketing systems have implemented digital ticketing technologies like QR codes or NFC in response to the growing demand for contactless transactions and the requirement for increased security. Compared to conventional paper-based tickets, these solutions provide several benefits, such as better security, lower costs, and greater consumer convenience [2]. These systems do, however, have significant drawbacks, including the potential for usability and user adoption problems as well as vulnerability to hacking or counterfeiting.

Thus, there is a need for a ticketing system that is safer and more effective and that not only offers the advantages of digital ticketing but also tackles the problems and shortcomings of current systems. To overcome these issues and improve the entire ticketing experience, we have created a secure ticketing system based on QR codes using AES and RSA encryption [3].

1.2 Team Approach to The Design and Design Decisions

The team has chosen to develop an application that uses the Android operating system after studying multiple statistics provided by various organizations. The Android operating system has a major share in the smartphone market.

Conceptualization: The brainstorming sessions at different stages helped the team make suitable decisions to develop the application and achieve the objective.

Feasibility Assessment: The various requirements of the project have been discussed briefly to proceed further.

Design: The Figma user interface design tool is used to bring up the discussed designs during different sessions in the required format. Multiple designs were created, and the team worked

effortlessly to bring some into the final stage. The team dropped a few designs due to some limitations or security concerns that might affect the final deployed application. The team also discussed multiple countermeasures which have been implemented in the development stage.

Development: The team has chosen Android Studio as the development environment and used the Java programming language. Security features have been implemented as countermeasures to avoid various security concerns. The application has been developed and installed on physical Android smartphones, and an emulator is used in some cases. The team dropped a few decisions at this stage due to some limitations.

Testing: The developed ticketing system has been tested for multiple iterations, and positive results have been seen after testing for multiple iterations. The application is able to implement countermeasures and avoid security concerns as discussed. The testing was initially conducted using the emulator.

Deployment: The application is installed on the physical smartphone, and the implemented features are tested.

2 SYSTEM ARCHITECTURE

Introduced an encrypted QR code ticket that is safer than a normal QR code. In this system, users register themselves and enter all the details required to obtain a ticket. After entering all the required data and completing the payment transaction successfully, the user will receive an encrypted QR code. To encrypt the data, we used AES and RSA encryption algorithms. All attendees' data are stored in a database. At the ticket checker, we ensure that the user scans an encrypted QR code. Once decrypted, information about the user can be viewed and verified.

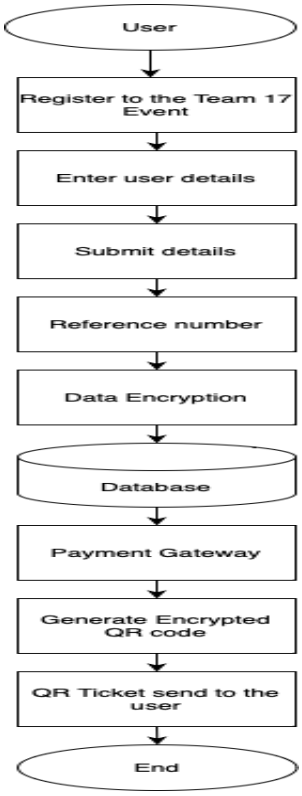


Fig 1 System Architecture of User's end

The generated QR is end-to-end protected and encrypted. All attendee's data are stored in a database. Complete user information is stored in the server's database. To communicate with the user, the server must first establish a connection. Each user's activity is updated in the server's database. Before a user can access an application, the server verifies the user's identity and keeps track of all user information in its database, verifying it as needed.

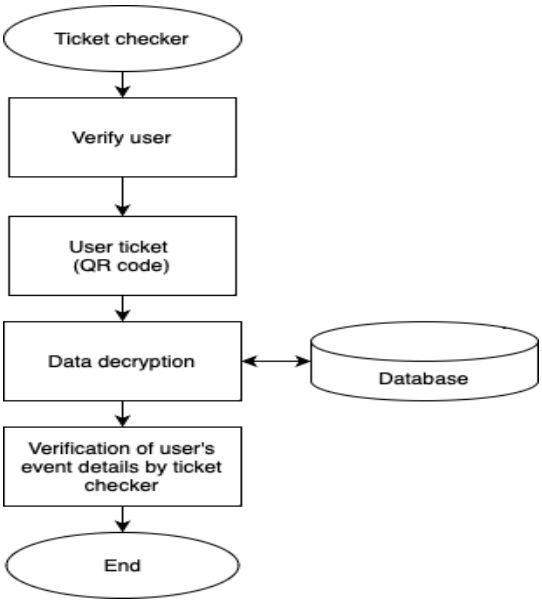


Fig 2 System Architecture of Ticket checker's end

3 IMPLEMENTATION

3.1 User End

The AES and RSA algorithm is used for encryption. User data is stored in the database in encrypted form. Decryption happens at the receiver's end. On the Ticket Examiner page. In addition to ticket creation, ticket validation is also very important. As a result, digital ticket controls are very useful for 100% ticket validation. Users register on the client side and the scanning or verification happens entirely on the ticket checker side. QR code generation is performed on the user side, and QR code verification is performed on the ticket checker side to ensure security during data transmission. The Anti-screenshot feature has been added in addition to prevent users to take screenshot and sharing those images.

The user interface allows the user to enter their data, select an event, and request a ticket. The user's data is sent to the Firebase Realtime Database for storage.

The Firebase Realtime Database stores the user's encrypted data, including their name, email, event selection, and ticket ID. A secure random number generator is used to create the ticket ID, which is then encrypted with AES before being saved in the database.

QR code library called QR Gencoder and a scanning library called ZXing are used in conjunction with the ticket ID to create an encrypted QR code ticket. Finally, the user's device receives the QR code.

3.2 Ticket Checker End

A smartphone application is created to scan the encrypted QR code at the Ticket Checker's end. Java programming is used to design the android application in Android Studio. The attendee details can be seen and verified after decryption. Both the checker and the customer will receive notification that the ticket has been examined and verified. In this way, the ticket will be confirmed by the ticket checker. This method makes sure that the large number of tickets are inspected as quickly as possible.

The major objective of this project is to offer 100% ticket checking in, as well as an innovation for quick, efficient ticket checking with minimal disruption and at the user's

convenience. Ticket Checker is has a QR scanner device which has access to database it scans the QR code ticket shown by the attendee after scanning data is decrypted at the backend and then verification of attendee event details by the Ticket Checker.

3.3 RSA For QR Code Security

Symmetric algorithms such as AES are suitable for encrypting QR code images when the QR code needs to be scanned (sent) by multiple people. On the other hand, if a QR code is protected by an asymmetric algorithm, each user has its own dedicated private key, and thus must be encrypted exclusively for each recipient. It is faster and less computationally intensive than asymmetric algorithms., However, existing symmetric key algorithm implementations for image security [4], [6], [7] lack security of the shared key. The shared secret key must be secured strongly enough to prevent successful attacks on the key. Therefore, a model where the QR code is encrypted and decrypted using AES algorithm [4] and its shared key, is secured by RSA algorithm as shown in Fig 3 [7]

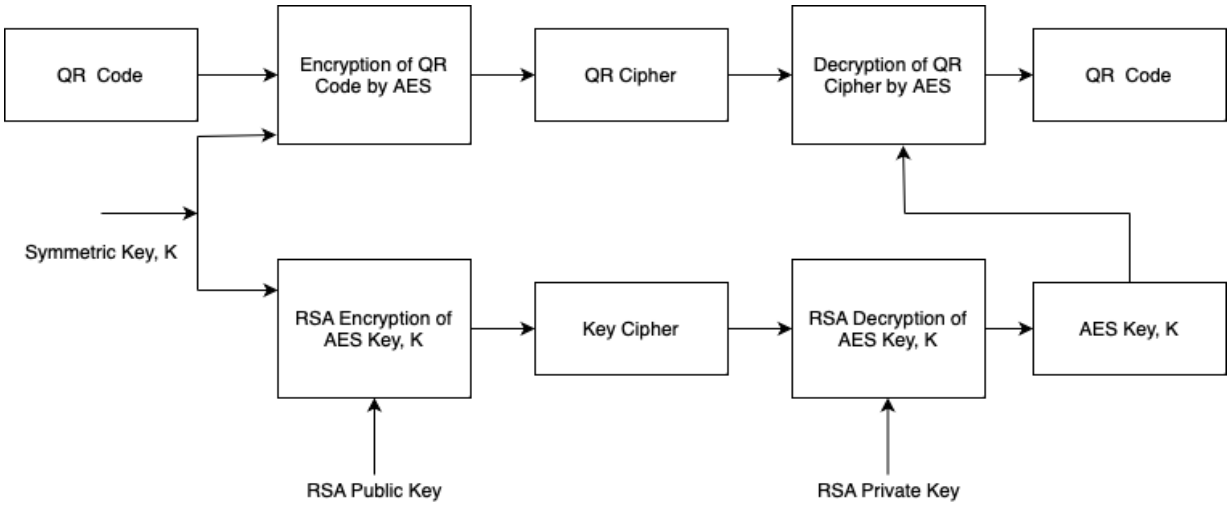


Fig 3 Security of QR code and its shared key

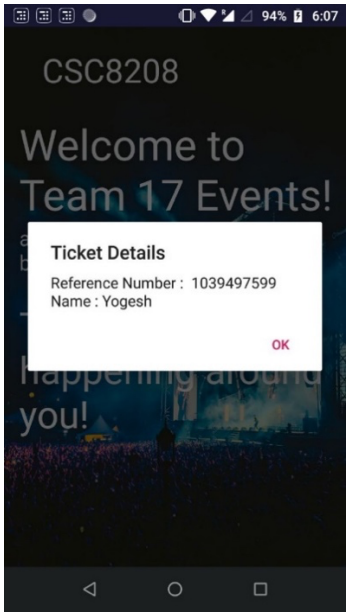


Image 1

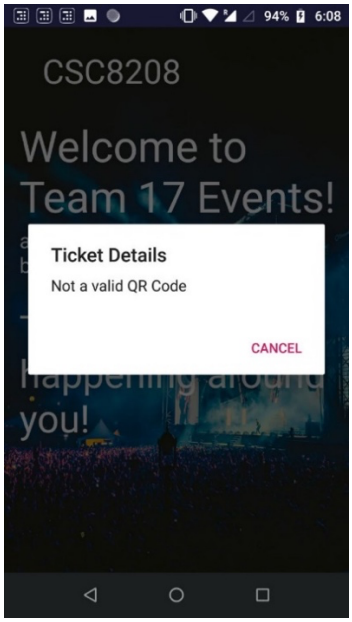


Image 2

The images 1 and 2 are the outputs of the implementation in A secure Ticketing system for In-person events developed using the Android studio.

3.4 Keystore and Transport Security

Our application uses several keys as a part of the AES and RSA encryptions. These keys need to store securely to prevent any unauthorized use. Hence, we are using the Android KeyStore system to safely store our cryptographic keys.

It is also crucial to ensure that the data is not intercepted while its being send from the application to the database. This would prevent the man-in-the-middle attacks on the user data that is being sent. Hence Fire store protects our data as it travels over the Internet during read and write operations, through the use of Transport Layer Security (TLS). This ensures any eavesdropping or altered communication between the client and server.

3.5 Additional Security Features

One of the ways to steal event tickets is by taking the screenshots of the QR code which have been mistakenly shared. Hence allowing people other than the ticket holder to enter the event. Hence, we have implemented a feature that will prevent the user from screenshotting the QR

code. Hence the QR code will only be displayed when the user logs and view his event ticket. An example of the feature is show in the image below.

Image 3

4 EVALUATIONS

4.1 Performance Metrics

Key Performance Indicators	Time Taken in Seconds	Testing Iterations	Results	Justifications
QR Code Generation Time	< 1	25	Success	The testing has been carried out with different bandwidth and positive results has been achieved
Ticket Verification Time	< 1		Success	

Some of the assets in this Application which include the User, Event, Ticket, Application data. The Threat agents in this scenario are the Malicious Insiders, Fraudsters, Social Engineers, Third-party Resellers, Hackers. The threat Event Frequency is High if occurred for single time. The Threat capability here is low as the countermeasures has been implemented and the threat event consequence is critical.

4.2 Limitations of the System

There are several limitations of the ticketing system implemented which includes;

The dependence on smartphones as the system relies on camera features and access to internet to function it is highly required for a smartphone to be used in accessing the application. The system is also only compatible with only Android devices as it built using Android studio hence cannot be accessible with other operating systems such as IOS. It also required for the application to be installed on user's device in other to buy and scan tickets.

4.3 Threats and vulnerabilities

Denial of service: hackers could launch denial of service attacks on the system by overloading the system with requests above its threshold, thereby preventing users from accessing the system.

Privilege escalation: a user can escalate their privilege from accessing the buying tickets portal to gain access to admin portal in other to verify false tickets or gain access to database.

Social engineering attacks: social engineering could be done on users to decode their login details and gain unauthorised access to application.

Spoofing attacks: this can occur when attacker's direct users to a malicious website posing as the ticketing website to gain user data, attacker could also pose as a trusted user to trick ticketing system into granting access to unauthorised events and data. It can be prevented by implementing authentication and access control protocols.

4.4 Future or Possible Enhancements

- Compatible to other Operating Systems such as IOS, etc.
- Implementing Access control policies.
- Dynamic QR code feature
- Enhancing Authentication Features by implementing Biometric, Multi-factor Authentication, etc.
- QR codes can be further enhanced with anti-counterfeiting measures such as holograms, watermarks, or other security features. This can help prevent the creation and use of fake tickets.

5 CONCLUSION:

The use of smartphones for booking tickets and verification helps ease the ticketing process and enhance customer satisfaction. However, this technology also presents potential security challenges such as counterfeiting, fraud, and theft. These risks could arise from malicious insiders or hackers. This report shows an overview of the implementation process for a secure ticketing system for in-person events which attempts to deal with the security challenges faced in event ticketing system, it is implemented with Quick Response (QR) code functionality and additional security measures which involves encoding an AES public key with RSA private key and decrypting when QR code is scanned with the RSA private key to validate the generated Tickets in the Application. The application was able to detect the fake tickets that are not contained in the database and prevent unauthorised access to events.

6 INDIVIDUAL CONTRIBUTION

The all members of the Team have contributed equally at all stages of the coursework. The entire team was able to achieve the objective of the coursework through effective teamwork and communication.

S.NO	Student Number	Name	Course	Contribution In Percentage
1	**220548168**	Safura Mohammed Arif	MSc Cyber Security	14.29 %
2	**220005373**	Xiaoyue Zhang	MSc Advanced Computer Science	14.29 %
3	**220611613**	Prabhu Yogesh Vijayan	MSc Advanced Computer Science	14.29 %
4	**220588762**	Olomu Atinuke Praise	MSc Advanced Computer Science	14.29 %
5	**220243160**	Kailash Balachandiran	MSc Advanced Computer Science	14.29 %
6	**220486435**	Roshan Vadlapatla Kiran Naga Sai Satya	MSc Advanced Computer Science	14.29 %
7	**200879622**	Lanjian Huang	MSc Advanced Computer Science	14.29 %
			Total	100 %

References:

[1] Pappas, C. (2019). The Pros and Cons of QR Codes. [online] Business.com. Available at: https://www.business.com/articles/qr-codes-pros-cons/.

[2] Yu, J. and Yang, X. (2020). The Application of NFC Technology in Ticketing System. Journal of Physics: Conference Series, [online] 1641(1), p.012087. Available at: https://iopscience.iop.org/article/10.1088/1742-6596/1641/1/012087/meta.

[3] Fattah, A. and Ismail, M.A. (2020). Enhancing the Security of NFC Ticketing System through Secure Element. Journal of Physics: Conference Series, [online] 1529(1), p.012020. Available at: https://iopscience.iop.org/article/10.1088/1742-6596/1529/1/012020/meta

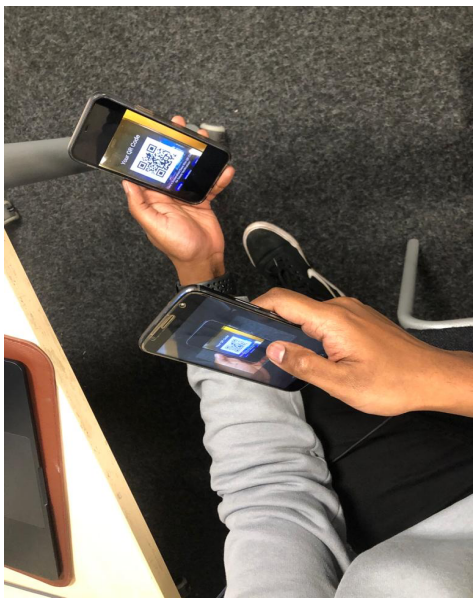
[4] Paweł Chodowiec and Kris Gaj. Very compact fpga implementationof the aes algorithm. In International Workshop on CryptographicHardware and Embedded Systems, pages 319–333. Springer, 2003. Available at: https://link.springer.com/chapter/10.1007/978-3-540-45238-6_26

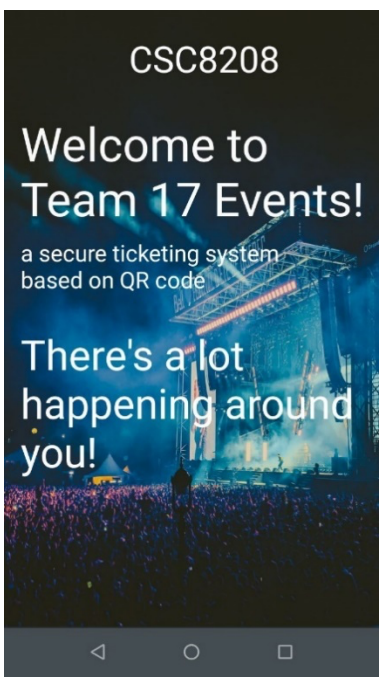
[5] Anish Goel and Kaustubh Chaudhari. Fpga implementation of anovel technique for selective image encryption. In Frontiers of SignalProcessing (ICFSP), International Conference on, pages 15–19. IEEE,2016. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7802949

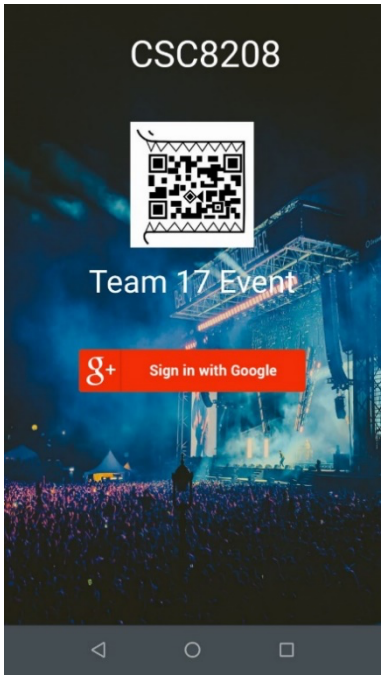
[6] Medien Zeghid, Mohsen Machhout, Lazhar Khriji, Adel Baganne, andRached Tourki. A modified aes based algorithm for image encryption.International Journal of Computer Science and Engineering, 1(1):70–75, 2007. Available at: https://www.researchgate.net/publication/337022861_A_Modified_AES_Based_Algorithm_for_Image_Encryption

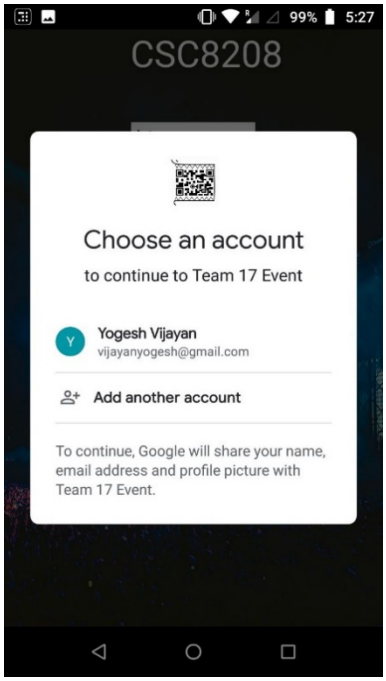
[7] Priyanka Gupta, Sandeep Saini and Kusum Lata. Securing qr codes by rsa on fpga. 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI) Available at: https://www.researchgate.net/publication/321509936_Securing_qr_codes_by_rsa_on_fpga [accessed Mar 23 2023].

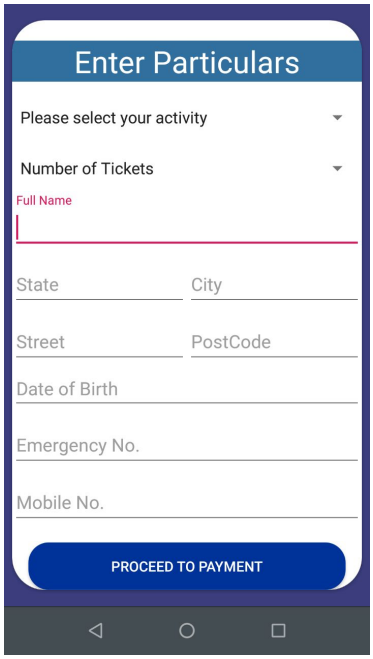
APPENDIX

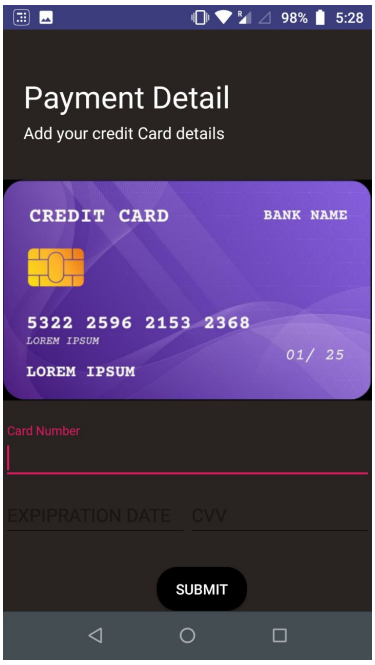












1. The above images are the outputs of the implementation in A secure Ticketing system for In-person events developed using the Android studio.
2. The Save QR option where the QR code is only saved in the wallet in the Application.
3. The above images are the outputs of the implementation in A secure Ticketing system for In-person events developed using the Android studio.
4. The Save QR option where the QR code is only saved in the wallet in the Application.