Bus Ticket Booking System using QR Code with AES Encryption

Sheng-Hui Chai[1], Lee-Ying Chong[2]*, Siew-Chin Chong[3], Pey-Yun Goh[4]

[1,2,3,4]*Faculty of Information Science and Technology, Multimedia University, Malaysia; E-mail: lychong@mmu.edu.my*

Abstracts: Most Malaysians prefer public transportation as their primary choice for working, returning to their hometown, or traveling from state to state. Due to the rapid development of the Internet, bus tickets can be bought online. However, a passenger who purchased the bus ticket still needs to queue up to get the physical bus ticket, which is time-consuming and increases the risk of virus infection during the pandemic. If the passenger loses his physical tickets, the information may be leaked and exploited. This paper is designed to allow a passenger to book tickets online, where the bus ticket is presented as a QR code. The QR code ticket is auto-generated after the booking is completed. Hence, the passenger can skip queueing at the counter to get the physical ticket, reduce the virus infection and save waiting time. Besides, a cryptographic mechanism is applied in the QR code ticket, which prevents the leakage of QR code information. When the passenger scans the QR code ticket, the system decrypts the protection layer of the QR code ticket and automatically records the passenger's presence in the proposed system.

Keywords: Ticket Booking System, QR Code, AES, Encryption, Decryption.

1. INTRODUCTION

As the population grows, the demand for vehicles also increases. People use transportation to get to work, travel, etc. Most Malaysians still travel to their desired destination by public transportation as their primary mode of transportation. Also, because of the lower cost of the tickets and the services offered, including trips to almost all Peninsular Malaysia's cities, public transportation services like bus express are always in high demand from passengers.

In 2023, most existing online bus ticket booking system provides e-tickets when passengers successfully book tickets, however passengers still need to show the e-tickets to the counter to replace them with physical tickets with QR code, and a printing fee is charged to the passengers. It also means that passengers need to go through an extra step to get their physical tickets instead of using the e-ticket directly as an official ticket.

Besides, the current ticket is in the paper, and all the information about the passenger is displayed, such as passenger name, departure, arrival location, date, seat number, bus operator, etc. Also, because it is on paper and if it is dropped, other people may pick it up, and they can pretend to be others taking the bus. Moreover, the ticket printout exposes the passenger information, which may make passenger information misused by others in some illegal activities.

Furthermore, in the normal process, passengers need to go to the counter to ask the bus working staff to verify the e-ticket and exchange the physical ticket for the passenger. As a result, passengers have to spend a lot of time queuing in order to exchange tickets. After exchanging tickets, they also need to ask the staff to check the tickets before they can board the bus, which is very time-consuming. In addition, queuing involved physical contact, which may increase the risk of infection.

In this paper, we propose a web application to allow passengers to book bus tickets and obtain a QR code ticket after booking is completed. This web application focus on QR code generation and cryptographic mechanism on the QR code. A security layer added to the QR code to ensure the confidentiality and integrity of passenger's information. After a passenger book a ticket, the system generates an encrypted QR code ticket through an encryption process. Besides that, the proposed system decrypts the QR code ticket when the passenger scans the QR code. Hence, the computer is required to have webcam for QR code scanning.

2. LITERATURE REVIEW

2.1. Bus Transportation System

In Malaysia, bus services continue to be one of the most widely utilised public services and are still a crucial part of Malaysia's land public transportation. There are several bus categories in Malaysia such as local bus, interstate bus, cross country bus, school bus, etc. Local bus services are available for citizens or visitors travelling in cities like Kuala Lumpur to help them get from one place to another place within the city as well as to some of the suburbs (Thanaditsayakun, 2023). Also, it is cheap to travel from one place to another within the city by local bus.

Interstate bus is a bus that travels from one state to another such as Johor to Malacca, Johor to Kuala Lumpur, etc. Interstate bus fares are usually between RM20 to RM50. Interstate bus are usually used for students who need to study in other states, or those who temporarily need to work in other states, etc. Besides that, cross country bus is mainly from workers from Malaysia to Singapore, which is an inexpensive and efficient way for them.

Moreover, the method of buying tickets is also changing gradually with the advancement of the times and the advancement of technology. For example, in the past, when taking a local bus, passengers need to put money into a cash box, and the driver prints the ticket to the passenger according to their destination. Also, in the past, when people needed to take an interstate bus, they had to queue up at the counter at the bus central to buy a ticket. Now, with the internet, people can buy electronic tickets from online bus ticketing platforms. After that, they just need to show the e-ticket at the counter to exchange the ticket. In addition, the form of the ticket has also been changed, from the original barcode to a QR code.

2.2. Cryptographic Study

Cryptography is a technique of protecting data and communications through use of mechanisms which only allow sender and receiver to view the messages. Cryptography has four main features. First is confidentiality, it ensures that data is only the intended receiver can understand the information. Therefore, to ensure data is only viewable by receiver, some of the encryption methods are used such as Advanced Encryption Standard (AES). Second is integrity. The information cannot be modified, tampered, or changed while it is in storage without the change being detected. Third is non-repudiation. The information send from sender cannot be deny after the being transmit whereby digital signatures is used to prevent entities from denying that they took an action. Last is authentication. The sender and receiver can prove each other’s identity (Richards, 2021).

2.3. Encryption and Decryption

A normal text or plaintext converting into meaningless message or ciphertext is called Encryption; whereby Decryption is the process of converting ciphertext back into its original form or normal text or plaintext. There are two types of encryptions which are symmetric and asymmetric encryption.

Symmetric encryption, also known as private-key cryptography, use a single private key, and sender and receiver share the key to encrypt and decrypt messages. If the private-key is disclosed, communications is compromised since both sender and receiver are equal. Example of symmetric encryption include Data Encryption Standard (DES), Triple Data Encryption Standard (Triple DES) and

Advanced Encryption Standard (AES). The most commonly used symmetric algorithms in nowadays is AES-256, AES-192 and AES-128.

Conversely, asymmetric encryption, also known as public-key cryptography, sender and receiver uses pair of keys to encrypt and decrypt messages, a public key and a private key. The public key may be known by everyone. Anyone can access it and encrypt messages by using the public key. The private key that only be known by originator/recipient to decrypt messages. Therefore, data can only be decrypted by using the corresponding private key. Example of asymmetric encryption include Rivest Shamir Adleman (RSA), Digital Signature Standard (DSS).

2.4. Plaintext and Ciphertext

Plaintext is the information that can be read by humans or a computer machine. Using an encryption algorithm and an encryption key, the plaintext is encrypted, also known as the process of encryption. Conversely, ciphertext is the output of encryption, which is a series of random numbers and letters that are not human-readable. Ciphertext can be used to protect integrity and confidentiality of data, because an unauthorised party intercept the messages, without the encryption algorithm and key, they still have difficulty and take time to break the ciphertext.

2.5. QR Code

In 1994, QR codes were invented for the first time. Denso Wave, a Toyota company in Japan, created the QR code to aid in the production process by assisting in the monitoring, managing, and tracking of automobiles and components (Schulfer, 2020). The term Quick Response code (QR code) comes from the fact that it was created to allow for quick decoding rates. Also, 2D (two-dimensional) QR codes were created in response to growing demands for storage capacity, usefulness, accuracy, and other variables, as well as the drawbacks associated with advances in bar code technology. Similar to the evolution of barcodes, QR codes have witnessed an increase in population, technological advancements to enhance storage capacity and usefulness, and so on since their inception.

2.6. The Existing Applications

To investigate and analyse existing systems with similar functionalities, three case studies were conducted on redBus, CatchThatBus, and BusOnlineTicket.com. These systems were examined to understand their functional components and gain valuable insights. The analysis of these systems enhances the comprehension of the overall system flow.

2.6.1. redBus

redBus is an online bus ticket booking system that offers the ability to buy tickets via its website, iOS, and Android mobile apps. redBus was founded in the year 2006 (redBus, 2023). redBus is headquartered in Bangalore, Karnataka, India. redBus brought together more than 3,500 bus operators that operate over 100,000 routes all over the world, revolutionising the online bus ticket booking industry. They have extended their presence to include India, Indonesia, Colombia, Malaysia, Singapore, and Peru. In Malaysia, they bring together over 150 bus operators. Their partner companies in Malaysia include Mayang Sari Express, StarMart Express, City Express, Causeway Link, etc.

redBus allows the passenger to book a ticket without login into the system. Users can select the departure and arrival location, onward date and return date as optional to find their desired buses. Also, redBus displays bus photos, ratings and reviews so that passengers can use them for comparison. redBus enable passengers to view or resend the booked ticket details by entering the ticket number and email used for booking. In addition, redBus not only allows users to book bus tickets but it also allows users to book ferry tickets on its platform.

2.6.2. CatchThatBus

CatchThatBus was founded in the year 2012. For each ticket sold, CatchThatBus receives a 10% to 15% commission. Since the ticket is only getting digitised during that time, many older drivers need to learn how to read it and determine its validity. Despite this, CatchThatBus has persuaded more bus operators and drivers to accept digital tickets. The company has persuaded 70 of 120 bus operators in Malaysia to publish their tickets on its website, selling 70,000 tickets each month (Noordin, 2017). CatchThatBus support different OS to allow the user book bus tickets. Also, CatchThatBus provides a currency converter function between Singapore and Malaysia for the convenience of its users. Besides, it also provides a form for the user to let them know if the user cannot find his trip.

2.6.3. BusOnlineTicket.com

BusOnlineTicket.com is an online bus ticket booking platform, was founded in the year 2008 by Winston Wong. It provides over 200 bus operators that can be booked, covering over 8,000+ routes within Malaysia and Singapore to over 140+ destinations (BusOnlineTicket Pte Ltd, 2023). BusOnlineTicket.com supports a full range of ticket booking services. In addition to bus tickets, it also provides trains and ferries. It even cooperates with Agoda to provide a combined search function for buses and hotels, making it easier and more convenient for users to find what they want. Likewise, it provides a currency converter function between Malaysia, Singapore, and Indonesia and a function to findusers' booking by enter their booking reference number and email. Table 1 shows the comparison of the existing application and the proposed system.

Table 1. Comparison of the existing applications and the proposed system

Comparison	Application			
	redBus	CatchThatBus	BusOnlineTicket.com	Proposed System
Cross State	Yes	Yes	Yes	Yes
QR code ticket is generated after booking is done	No	No	No	Yes
Ticket has to be exchanged manually	Yes	Yes	Yes	No
Automate the QR code ticket validationchecking.	No	No	No	Yes
Data protection inQR code	No	No	No	Yes
Programming Language	PHP	Angular 2.0	PhoneGap	Python

3. PROPOSED SOLUTION

The proposed system allows passengers to book bus tickets online and obtain a QR code ticket after booking is completed. The main focus of the system is QR code generation and cryptographic mechanism on the QR code. A security layer is added to the OR code to ensure the confidentiality and integrity of passenger's information. The system generates an encrypted QR code ticket through an encryption process once the ticket booking process is completed.

3.1. System Design

The context diagram illustrated in Fig. 1, shows that a passenger can submit a new account registration request and use the login information to login to the system. Passengers can request the route information and booking history information, and the system responds with the ticket information. On the other hand, the admin can submit the login request to access the system. Also, the admin is allowed to request access to the route and booking information.

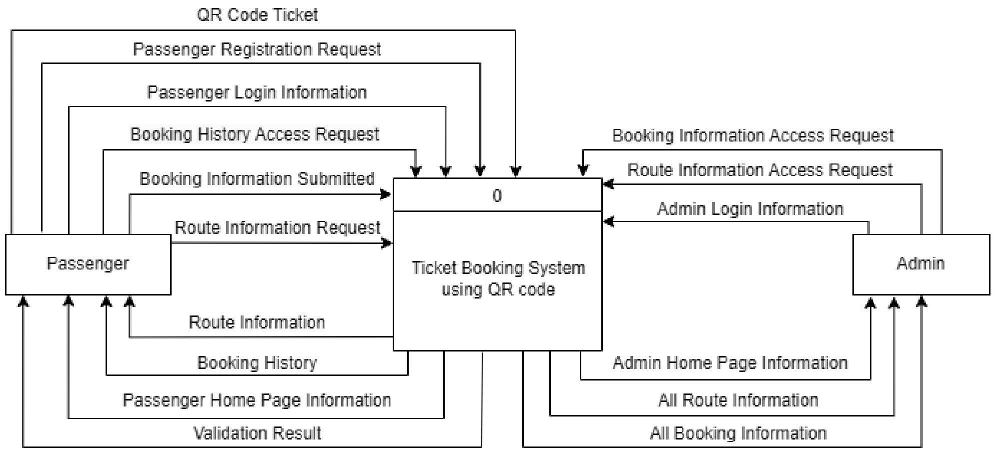

QR Code Ticket
Passenger Registration Request
Passenger Login Information
Booking History Access Request
Booking Information Submitted
Route Information Request
Passenger
0
Ticket Booking System using QR code
Booking Information Access Request
Route Information Access Request
Admin Login Information
Admin
Route Information
Booking History
Passenger Home Page Information
Validation Result
Admin Home Page Information
All Route Information
All Booking Information

Fig. 1: Context diagram of the proposed system

Fig. 2 shows the data flow diagram of the proposed system When the route information request received by the system and route information retrieved from the data source D3 (Route) and returned to the passenger. When passenger submit booking information, the system stores the information in data source D4 (Booking), QR code is generated and stored in same data source D4. On the other hand, the route or booking information is retrieved when admin send the request to the system. The process 5.0 (Generate QR code) and 9.0 (Check Validation) can be illustrated further to Data Flow Diagram Level- 2 in Fig. 3 and Fig. 4 respectively.

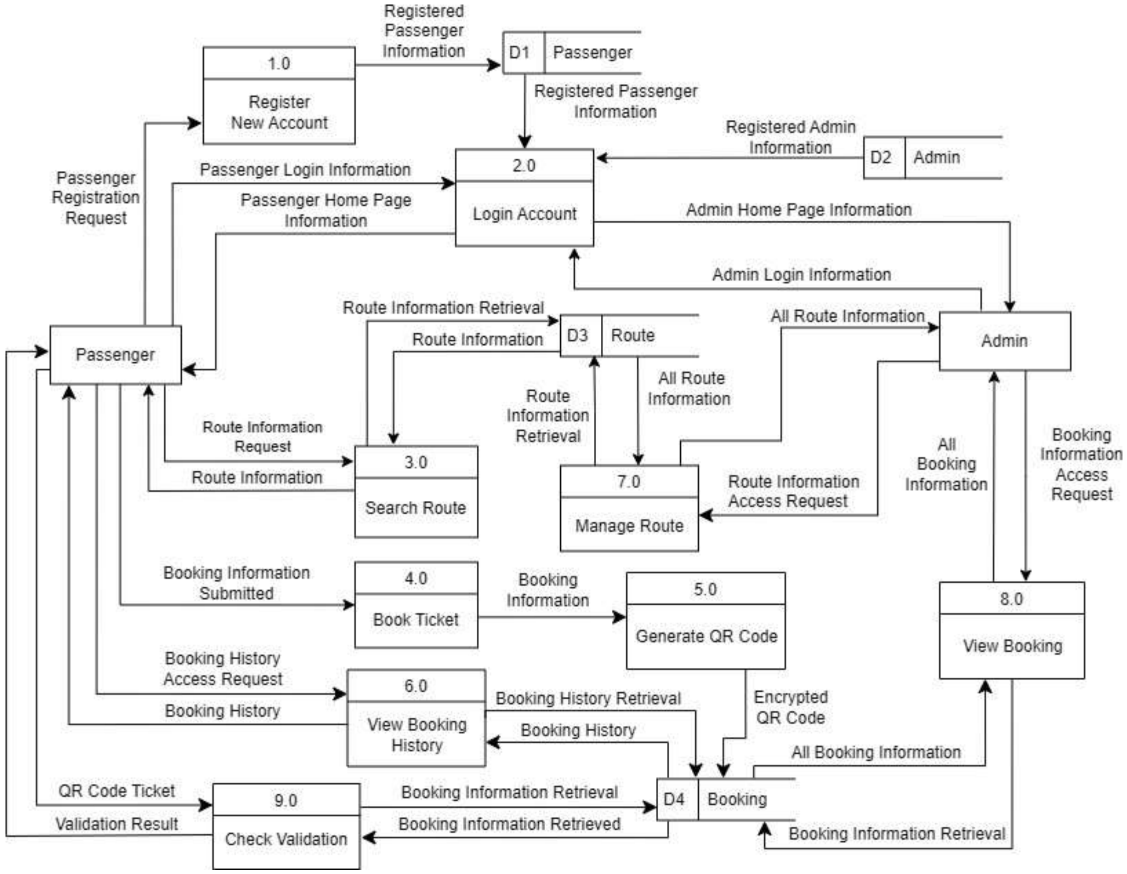

1.0
Register New Account
Registered Passenger Information
D1 Passenger
Registered Passenger Information
Registered Admin Information
D2 Admin
2.0
Login Account
Passenger Registration Request
Passenger Login Information
Passenger Home Page Information
Admin Home Page Information
Admin Login Information
Passenger
Route Information Retrieval
Route Information
D3 Route
All Route Information
Admin
Route Information Retrieval
All Route Information
Route Information Request
Route Information
3.0
Search Route
7.0
Manage Route
Route Information Access Request
All Booking Information
Booking Information Access Request
Booking Information Submitted
4.0
Book Ticket
Booking Information
5.0
Generate QR Code
8.0
View Booking
Booking History Access Request
Booking History
6.0
View Booking History
Booking History Retrieval
Booking History
Encrypted QR Code
All Booking Information
QR Code Ticket
Validation Result
9.0
Check Validation
Booking Information Retrieval
Booking Information Retrieved
D4 Booking
Booking Information Retrieval

Fig. 2: Data flow diagram of the proposed system

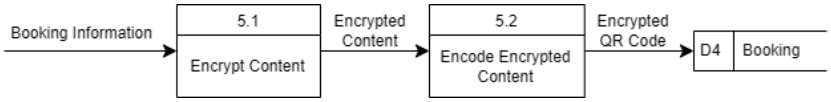


Fig. 3. Data flow diagram level 2 (process 5.0)

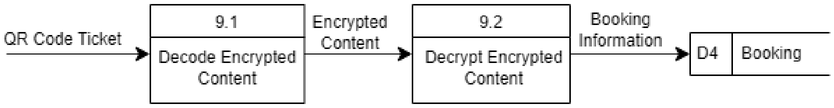


Fig. 4: Data flow diagram level 2 (process 9.0)

The passengers using the proposed system for the first time should register themselves on the registration page. Then, the passengers are required to provide a username and password as a login credential to access the system. After that, passengers can perform the search route, view route, book ticket, and view booking history, as well as check validation of the QR code ticket, as shown in Fig. 5.

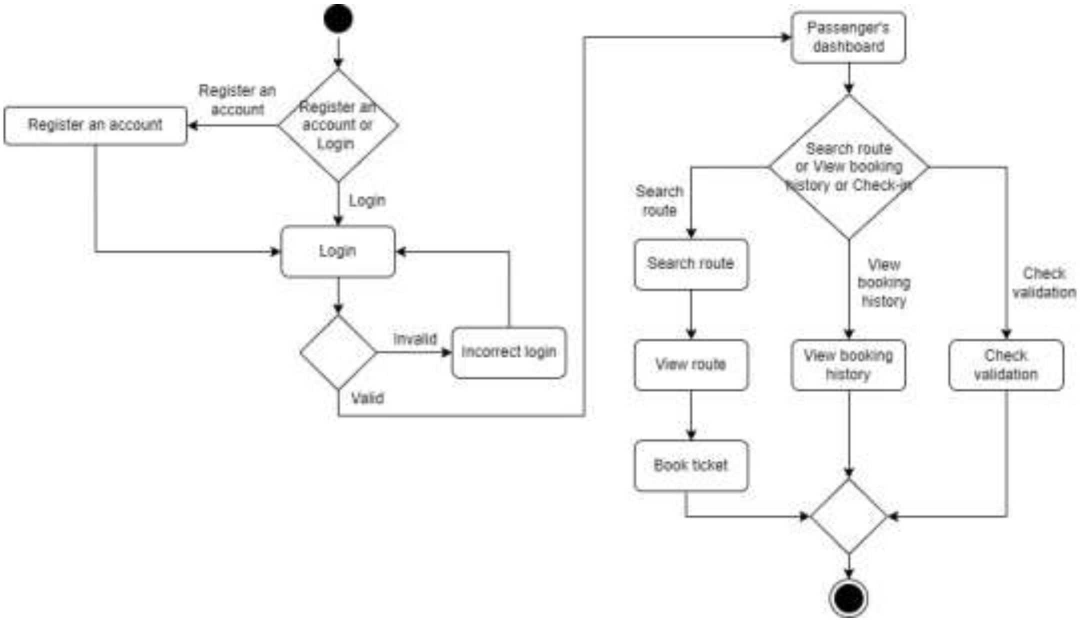


Fig. 5: Activity diagram (Passenger)

3.2. System Architecture

Fig. 6. illustrates the system architecture of the proposed system, consisting of frontend and backend parts.The frontend of the system is implemented by Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript (JS). The backend of the system is implemented by the Python Django framework and PostgreSQL database. The passenger submits the requests to the system through a web-based interface. The proposed system encrypts booking details and encodes them into a QR code. The QR code is stored in the database and presented on the system interface to the passenger. The admin submits the request to the system to access all the data stored in the database. The proposed system decrypts the data from the QR code and updates the booking details when the passenger scans the QR code for check-in.

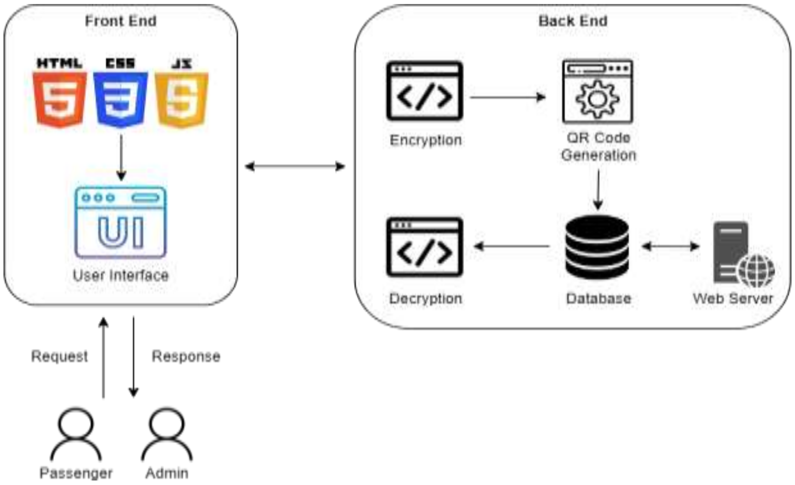


Fig. 6: System architecture of the proposed system

3.3. System Implementation

The proposed system generates the QR code which consists of the passenger's detail and the ticket's information. AES Encryption and Decryption is applied on the QR code to protect data in the OR code.

3.3.1. QR Code Generation

Flow chart of QR code generation process is illustrated in Fig. 7. After the proposed system receive the message, the message is converted into a byte string mixed with generic QR header information, error correction bytes, and a masking element. This altered byte string is then transformed into a two- dimensional matrix of 1s (white) and 0s (black), which can construct a QR code.

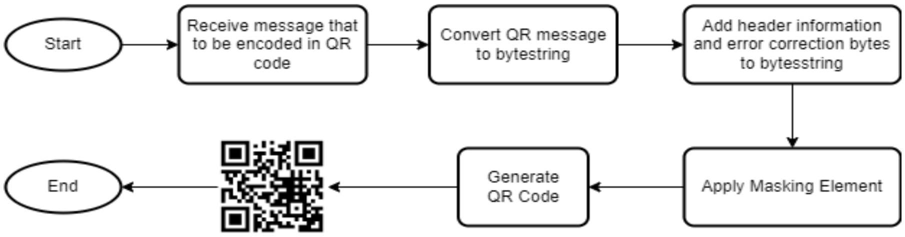

Start
Receive message that to be encoded in QR code
Convert QR message to bytestring
Add header information and error correction bytes to bytesstring
Apply Masking Element
Generate QR Code
End

Fig. 7: Flowchart of QR code generation

3.3.2. AES Encryption and Decryption

AES is a symmetric block cipher encryption. It is also known as the Rijndael algorithm. In 2000, NIST proposed Advanced Encryption Standard (AES) to replace DES and Triple DES. AES is based on a substitution-permutation network, with multiple rounds to encrypt data. AES uses longer key-length, including 10 rounds for 128-bit, 12 rounds for 192-bit and 14 rounds for 256-bit to encrypt and decrypt data (Bernstein & Cobb, 2021). The chosen key size determines the number of rounds. Therefore, with a longer key-length, it is practically unbreakable by brute-force attack based on current computing power.

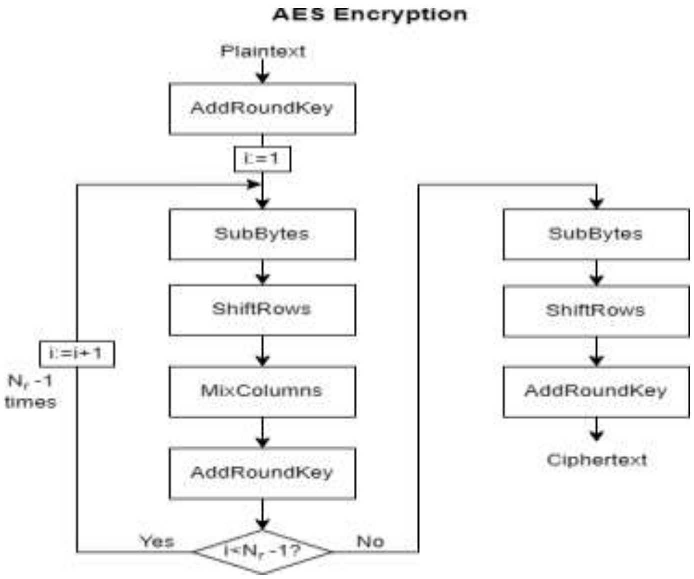


Fig. 8: AES encryption process

As shown in Fig. 8 above, 128-bit key is selected to encrypt the message, therefore 10 times of iterations is carried out. AES encryption consist of few fundamental operations: SubBytes, ShiftRows, MixColumns, and AddRoundKey. Before the encryption process start, the plaintext (original data) is divided into a 4 x 4 matrix of 16 bytes since AES is a block cipher. Then, by using Rijndael's key schedule, the AES algorithm generates round keys from the initial key during the AddRoundKey process (Awati, 2022). The initial key is indicated as a secret key or cipher key, it can be a random number, data, or a password.

In the SubBytes process, the bytes of data are substituted based on a table called Rijndael S-box to generate a substitute of new data. Then, the ShiftRows process the substitute of new data by shifts the bytes' place except for the first row. The second row's bytes are moved one position to the left, the third row's bytes are moved two positions to the left, and the last row’s bytes are moved three positions to the left.

The MixColumns process takes a column of data multiplied by a pre-established matrix to generate a new column for the subsequent state array. After that, the AddRoundKey process takes the mix column of the block to perform the XOR operation with the respective round key. The encryption key size affects the number of cipher rounds, N_r. Therefore, if a 128-bit key size is used, the process from SubBytes to AddRoundkey is repeated for 9 rounds, with the final round only going through the processe of SubBytes, ShiftRows, and AddRoundkey. Finally, the ciphertext (encrypted data) with random characters is generated after completing all rounds (Mustafeez, 2022).

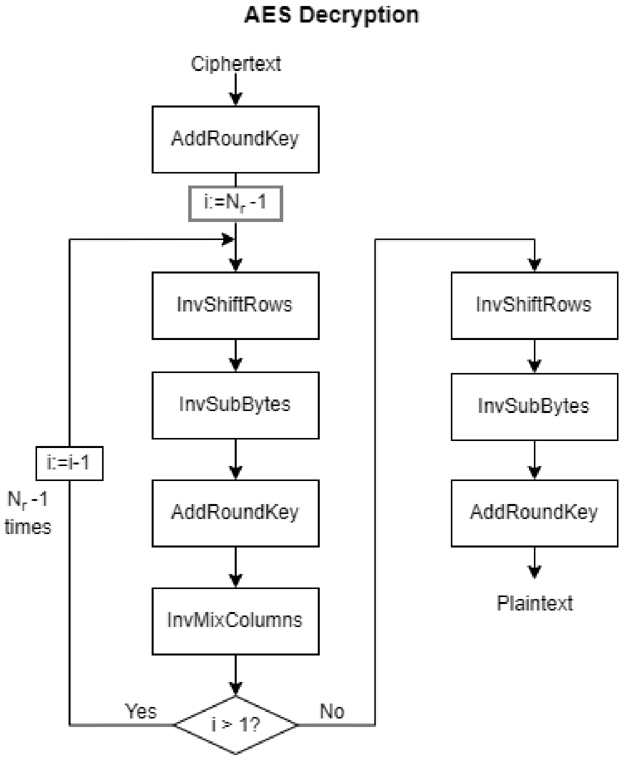


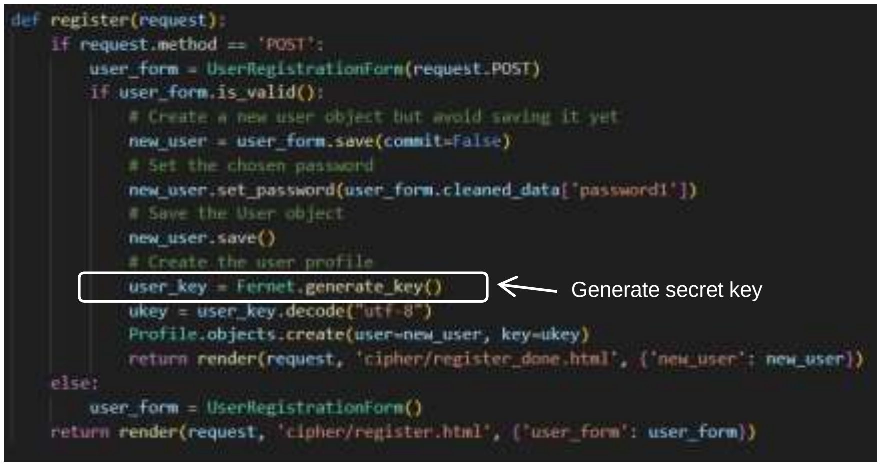
Fig. 9: AES decryption process

The flow of AES decryption is illustrated in Fig. 9. The fundamental distinction is that every basic decryption operation is substituted with its corresponding encryption inverse operation. The AES ciphertext can be returned to its initial state (plaintext) using inverse encryption (Rimkienė, 2022). As exhibited in Fig.

9, all the process is reversed. and final iteration is excluding the InvMixColumns process. Ten iterations are performed to get back the plaintext (original data) from the ciphertext.

3.3.3. AES Encryption and Decryption in the QR Code

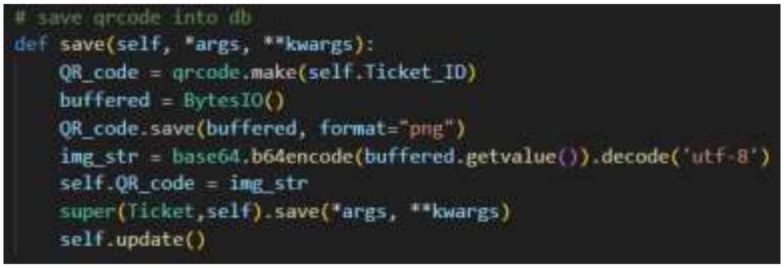
In the proposed system, the AES encryption mechanism is used to secure the data in the QR code, preventing the data from being misused by others. Therefore, when the passenger registers an account for the first time, the system assigns a secret key for the passenger, as illustrated in Fig. 10. The system generates a corresponding QR code after the passenger successfully booked a ticket. The QR code generation process is exhibited in Fig. 11.



```
def register(request):
    if request.method == 'POST':
        user_form = UserRegistrationForm(request.POST)
        if user_form.is_valid():
            # Create a new user object but avoid saving it yet
            new_user = user_form.save(commit=False)
            # Set the chosen password
            new_user.set_password(user_form.cleaned_data['password1'])
            # Save the User object
            new_user.save()
            # Create the user profile
            user_key = Fernet.generate_key()
            ukey = user_key.decode("utf-8")
            Profile.objects.create(user=new_user, key=ukey)
            return render(request, 'cipher/register_done.html', {'new_user': new_user})
    else:
        user_form = UserRegistrationForm()
    return render(request, 'cipher/register.html', {'user_form': user_form})
```

Generate secret key

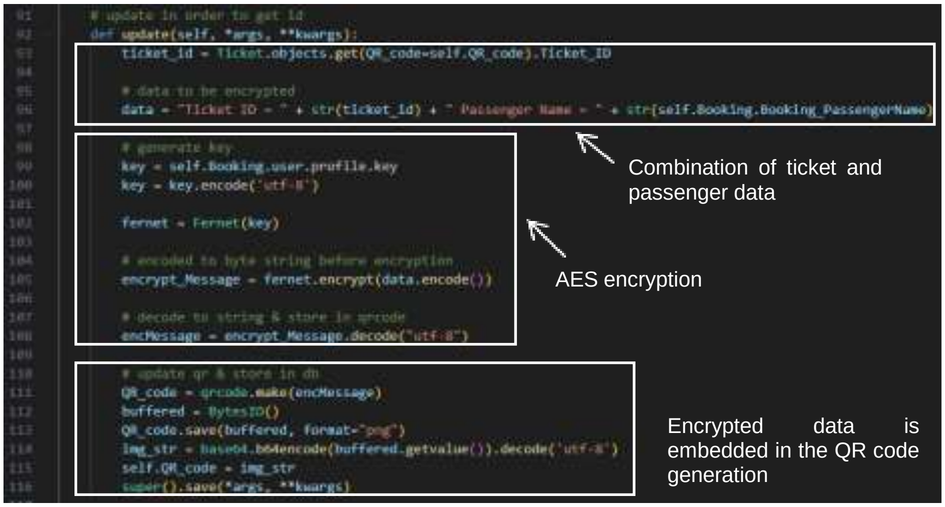
Fig. 10: Secret key generation



```
# save qrcode into db
def save(self, *args, **kwargs):
    QR_code = qrcode.make(self.Ticket_ID)
    buffered = BytesIO()
    QR_code.save(buffered, format="png")
    img_str = base64.b64encode(buffered.getvalue()).decode('utf-8')
    self.QR_code = img_str
    super(Ticket,self).save(*args, **kwargs)
    self.update()
```

Fig. 11: Process of QR code generation

The QR code library incorporates the ticket and passenger data into QR codes, while the base64 module converts the image object into a base64 image. It is crucial to highlight that the system specifically stores the base64 encoded QR code image rather than the entire image. This approach reduces the number of image requests, thus alleviating the server's workload. The base64 image stored in the database is in a byte string format. It is noted that the QR code only contains the ticket data without the passenger data. Hence, an update function is necessary to store both the ticket data and the passenger data into the QR code. The update function which involves AES encryption, is displayed in Fig. 12.



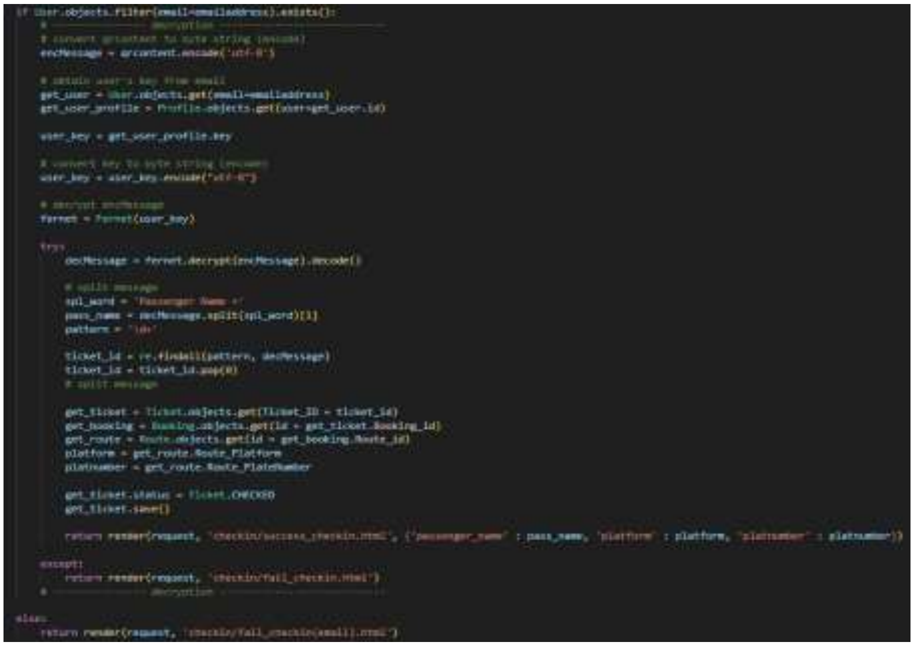
```
    # update in order to get id
    def update(self, *args, **kwargs):
        ticket_id = Ticket.objects.get(QR_code=self.QR_code).Ticket_ID

        # data to be encrypted
        data = "Ticket ID = " + str(ticket_id) + " Passenger Name = " + str(self.Booking.Booking_PassengerName)

        # generate key
        key = self.Booking.user.profile.key
        key = key.encode('utf-8')

        fernet = Fernet(key)

        # encoded to byte string before encryption
        encrypt_Message = fernet.encrypt(data.encode())

        # decode to string & store in qrcode
        encMessage = encrypt_Message.decode("utf-8")

        # update qr & store in db
        QR_code = qrcode.make(encMessage)
        buffered = BytesIO()
        QR_code.save(buffered, format="png")
        img_str = base64.b64encode(buffered.getvalue()).decode('utf-8')
        self.QR_code = img_str
        super().save(*args, **kwargs)
```

Combination of ticket and passenger data
AES encryption
Encrypted data is embedded in the QR code generation

Fig. 12: Process of AES encryption

In Fig. 12, the ticket and passenger data are acquired and merged. The combined data undergoes the AES encryption process. The system retrieves the passenger's secret key from the database and applies encryption to the content. Subsequently, the encrypted content is transformed into a QR code. The QR code is then converted into a base64 image and stores in the database.

When passengers prepare to check in, they must show the QR code and provide their registered email address. The system searches for the corresponding secret key based on the email address entered by the passenger and decrypts the encrypted content in the QR code. The decryption of the QR code process in the code view is displayed in Fig. 13.



```
if User.objects.filter(email=emailaddress).exists():
    # ------------------- decryption ---------------------------
    # convert qrcontent to byte string (encode)
    encMessage = qrcontent.encode('utf-8')

    # obtain user's key from email
    get_user = User.objects.get(email=emailaddress)
    get_user_profile = Profile.objects.get(user=get_user.id)

    user_key = get_user_profile.key

    # convert key to byte string (encode)
    user_key = user_key.encode("utf-8")

    # decrypt encMessage
    fernet = Fernet(user_key)

    try:
        decMessage = fernet.decrypt(encMessage).decode()

        # split message
        spl_word = 'Passenger Name ='
        pass_name = decMessage.split(spl_word)[1]
        pattern = '\d+'

        ticket_id = re.findall(pattern, decMessage)
        ticket_id = ticket_id.pop(0)
        # split message

        get_ticket = Ticket.objects.get(Ticket_ID = ticket_id)
        get_booking = Booking.objects.get(id = get_ticket.Booking_id)
        get_route = Route.objects.get(id = get_booking.Route_id)
        platform = get_route.Route_Platform
        platnumber = get_route.Route_PlateNumber

        get_ticket.status = Ticket.CHECKED
        get_ticket.save()

        return render(request, 'checkin/success_checkin.html', {'passenger_name' : pass_name, 'platform' : platform, 'platnumber' : platnumber})

    except:
        return render(request, 'checkin/fail_checkin.html')
    # ------------------- decryption ---------------------------

else:
    return render(request, 'checkin/fail_checkin(email).html')
```

Fig. 13: Process of AES decryption

Based on the Fig. 13, after the passenger clicks the "Check In" button:

a. The system checks the email address provided by the passenger to ensure it exits in the database.

b. If the email address exists, the system searches for the passenger’s corresponding secret key based on the email address and decrypts the encrypted content in the QR code.

c. The system extracts passengers’ details and ticket information from the decrypted and displays ticket information such as platform, bus plate number, and passenger name to passengers.

d. The system updates the QR code status from "booked" to "checked".

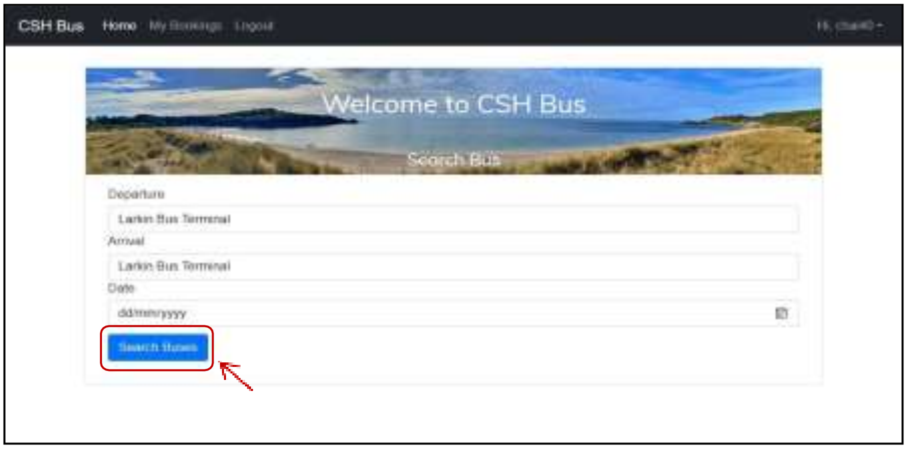
e. If the passenger enters an email address that is not registered in the system, the system displays the result of “This email address was not found” to the passenger. If the passenger enters the wrong email address or QR code ticket, the system cannot decrypt the encrypted content, and it displays the result of “Email address and QR code doesn’t match” to the passenger.

4. IMPLEMENTATION RESULTS

The system interface of the proposed system that can be used by the registered passenger and the admin is developed. Functions of the proposed system includes bus searching, ticket booking, history of booking, passenger check-in and etc.

4.1. Search Buses Page

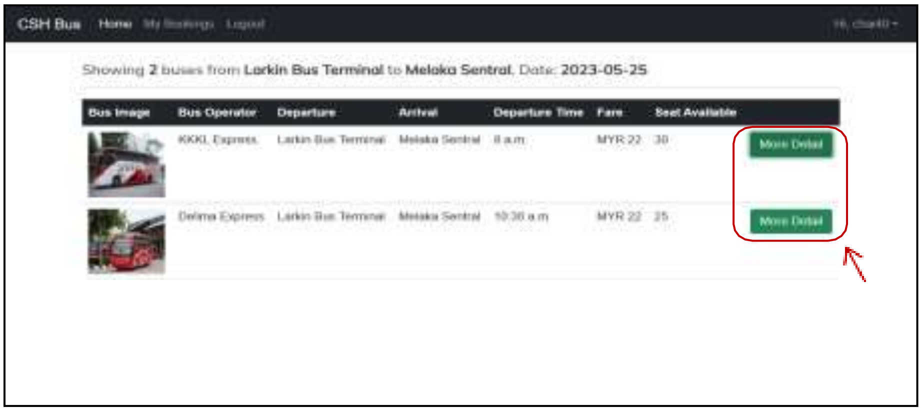
The main page of the proposed system once passengers log in is displayed in Fig. 14. This page allows passengers to search their desired bus by selecting departure location, arrival location, and departure date. Fig. 15 shows the search result after passenger pressed the “Search Buses”.



Departure
Larkin Bus Terminal
Arrival
Larkin Bus Terminal
Date
dd/mm/yyyy

Fig. 14: Search buses page

This search result contains information about the route such as bus image, bus operator, departure, arrival, departure time, fare, and seat available. Also, they can view the details of the route after click the “More Detail” button as shown in Fig. 15.



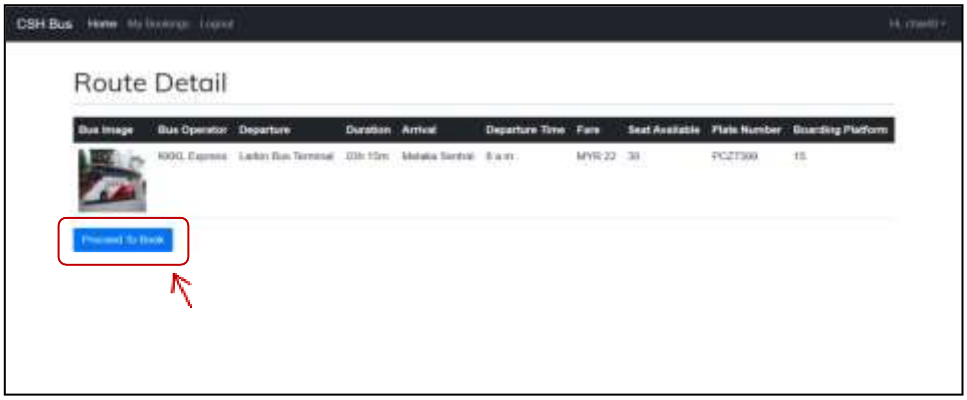
Showing 2 buses from Larkin Bus Terminal to Melaka Sentral. Date: 2023-05-25

Bus Image	Bus Operator	Departure	Arrival	Departure Time	Fare	Seat Available
	KKKL Express	Larkin Bus Terminal	Melaka Sentral	8 a.m	MYR 22	30
	Delima Express	Larkin Bus Terminal	Melaka Sentral	10:30 a.m	MYR 22	25

Fig. 15: Search result page

4.2. Ticket Booking Page

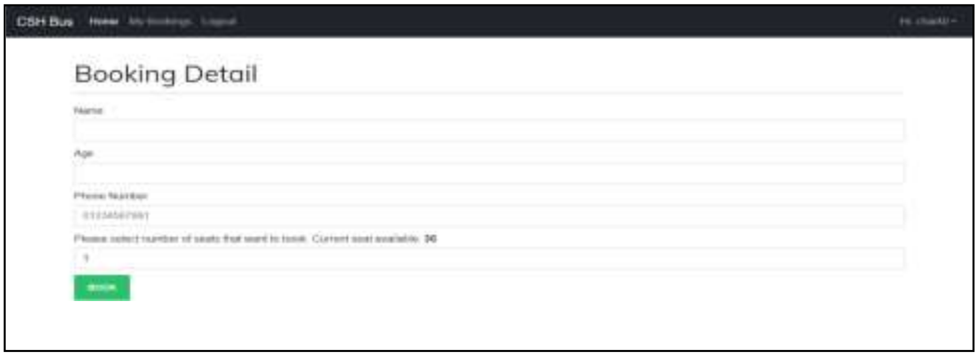
The Route Detail page appears after pressing “More Detail” from the search result page, as displayed in Fig. 16. When the “Proceed To Book” button from the Route Detail page is clicked, another page appears,which allows the passenger to book their ticket by filling in their information, as exhibits in Fig. 17. The system displays the booking history page after ticket booking is success.



Route Detail

Bus Image	Bus Operator	Departure	Duration	Arrival	Departure Time	Fare	Seat Available	Plate Number	Boarding Platform
	KKKL Express	Larkin Bus Terminal	03h 15m	Melaka Sentral	8 a.m	MYR 22	30	PCZ7309	15

Fig. 16: Route detail page



Booking Detail

Name

Age

Phone Number

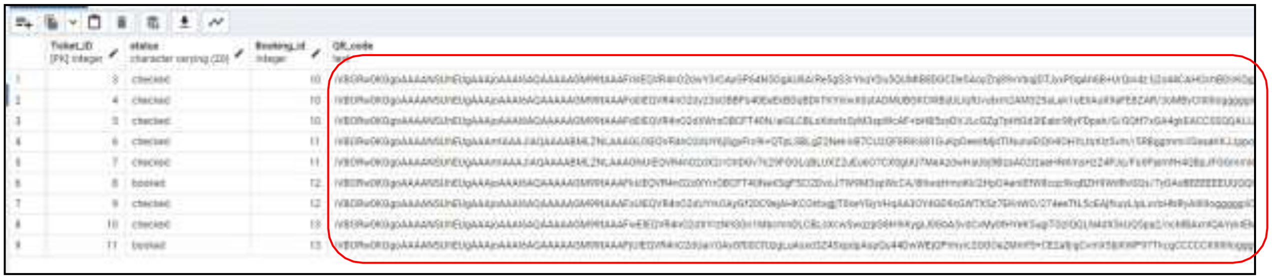
01234567891

Please select number of seats that want to book. Current seat available: **30**

1

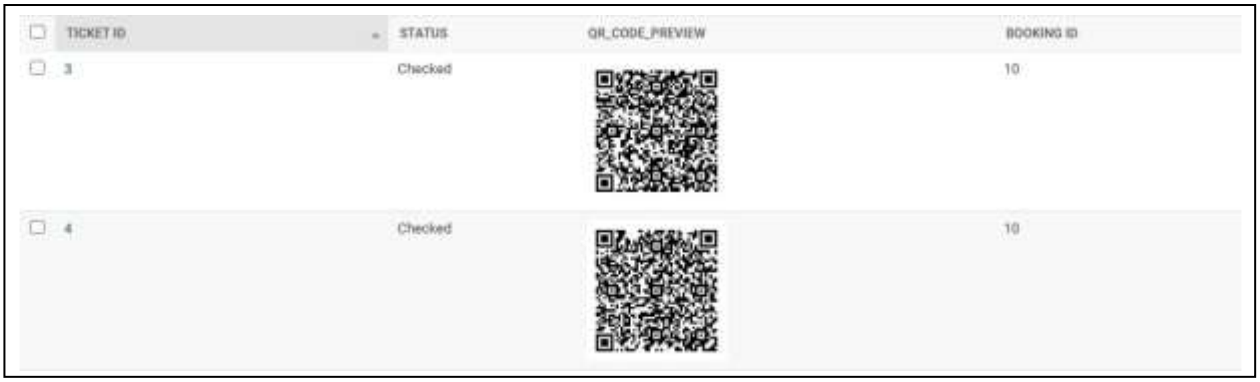
Fig. 17: Ticket booking page

Once the booking is performed, the AES encryption is applied, and the data is encrypted in the QR code. The encrypted QR code is converted to a base64 image and stored in the database. Fig. 18 shows the QR code as stored in the base64 string in the database. To allow the admin and passenger to view the QR code in the front-end system, it only needs to indicate the source of the image and define it as base64 encoded form. Fig. 19 shows the QR code displayed in the admin interface in the front-end system view.If using an online QR code decoder to decode the QR code, the content inside the QR code is encrypted instead of displaying the ticket and passenger data, which means the system is encrypted successfully, as shown in Fig. 20.



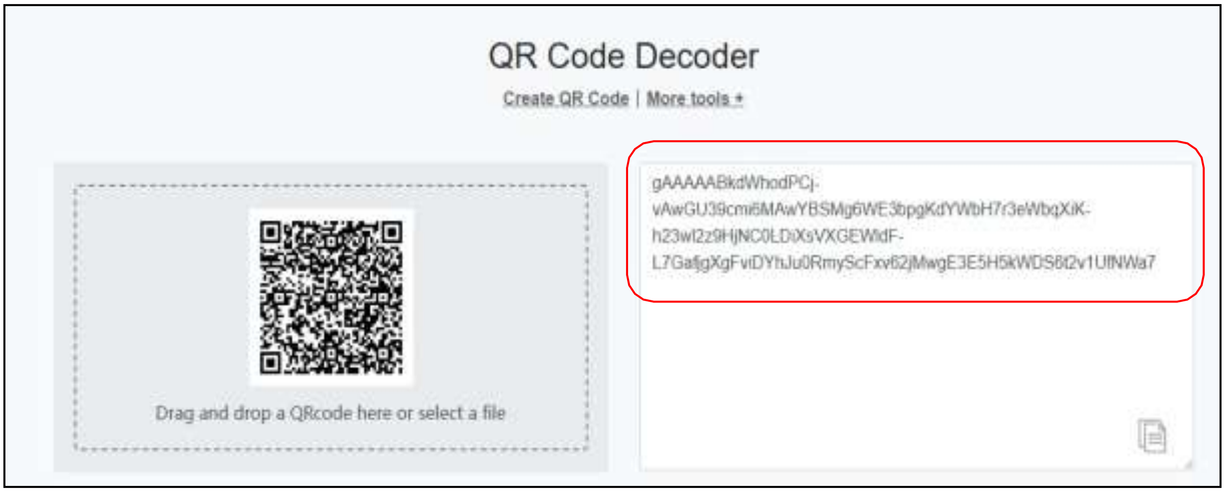
	Ticket_ID [PK] integer	status character varying (20)	Booking_id integer	QR_code text
1	3	checked	10	[illegible]
2	4	checked	10	[illegible]
3	5	checked	10	[illegible]
4	6	checked	11	[illegible]
5	7	checked	11	[illegible]
6	8	booked	12	[illegible]
7	9	checked	12	[illegible]
8	10	checked	13	[illegible]
9	11	booked	13	[illegible]

Fig. 18: Encrypted QR code in base64



☐	TICKET ID	STATUS	QR_CODE_PREVIEW	BOOKING ID
☐	3	Checked		10
☐	4	Checked		10

Fig. 19: QR code in admin page



QR Code Decoder

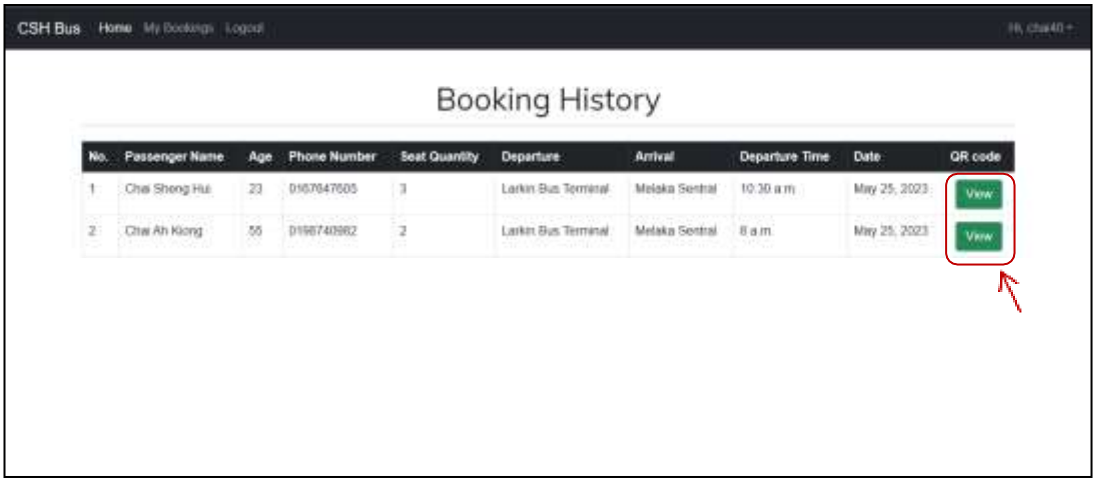
Create QR Code | More tools +
Drag and drop a QRcode here or select a file

gAAAAABkdWhodPCj-
vAwGU39cmi6MAwYBSMg6WE3bpgKdYWbH7r3eWbqXiK-
h23wl2z9HjNC0LDiXsVXGEWldF-
L7GafjgXgFviDYhJu0RmyScFxv62jMwgE3E5H5kWDS6t2v1UfNWa7

Fig. 20: Ciphertext in QR code

4.3. Booking History Page

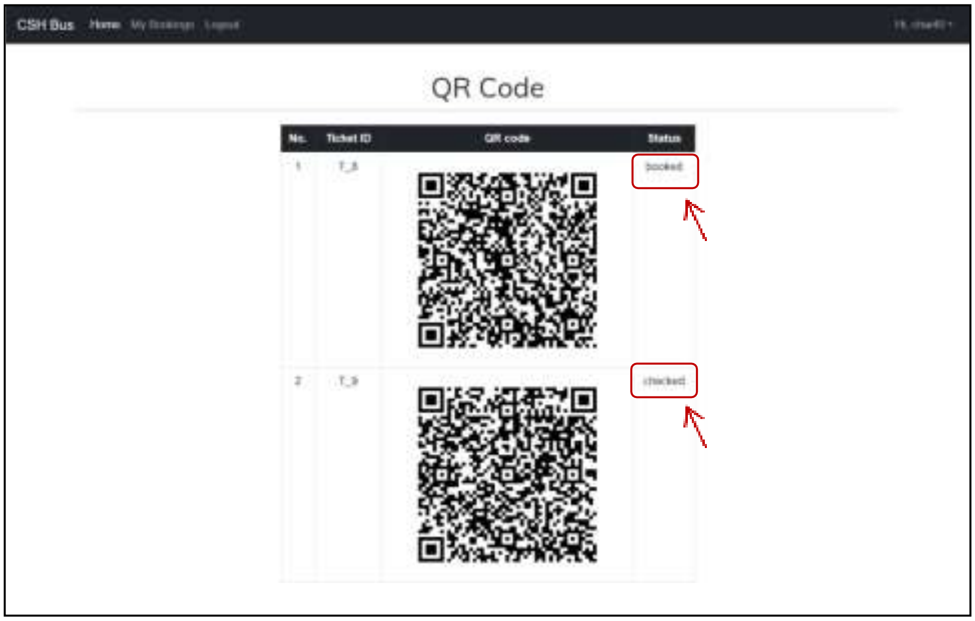
All the booked ticket record is listed at the booking history page as illustrated in Fig. 21. Passengers can press the view button at the right side to view their ticket in the QR code format as shown in Fig. 22. Passenger uses this QR code as ticket to check-in the bus. In addition, this page also shows the status of each QR code ticket. After the booking is completed, the system automatically save it as “booked”. Whena passenger successfully checks in for the trip, the system changes the status from “booked” to “checked”.



Booking History

No.	Passenger Name	Age	Phone Number	Seat Quantity	Departure	Arrival	Departure Time	Date	QR code
1	Chai Shong Hui	23	0167847605	3	Larkin Bus Terminal	Melaka Sentral	10:30 a.m.	May 25, 2023	View
2	Chai Ah Kiong	55	0198740982	2	Larkin Bus Terminal	Melaka Sentral	8 a.m.	May 25, 2023	View

Fig. 21: Booking history page



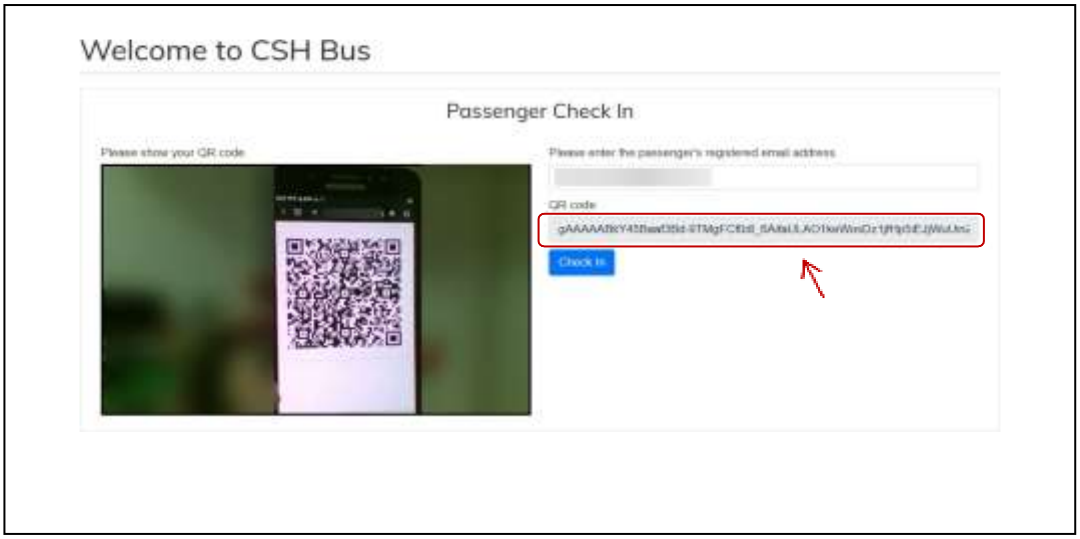
QR Code

No.	Ticket ID	QR code	Status
1	T_8		booked
2	T_9		checked

Fig. 22: QR code page

4.4. Passenger Check-In Page

Passengers are required to show the QR code in front of the webcam of a device at the bus station. Meanwhile, they have to fill in their registered email address to perform the check-in, as shown in Fig. 23. When the device scans the QR code, the encrypted data (ciphertext) have been captured and display in theQR code field. This ciphertext is a protective layer for the passenger's details and ticket information in theQR code. If an unauthorised individual attempts to scan the QR code, they are not able to view thecontent of the passenger's details and ticket information, as these data are already safeguarded.



Welcome to CSH Bus

Passenger Check In

Please show your QR code

Please enter the passenger's registered email address

[redacted]

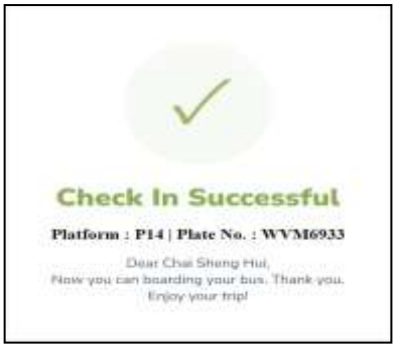
QR code

gAAAAABkY45Baw35bl-9TMgFC6bl_SAbkLAO1keWmDz1jRfqSEJjWuLkn

Fig. 23: Passenger check-in page

After passengers click “Check In” button, the system searches for the corresponding secret key based on the email address entered by the passenger and decrypts the encrypted content in the QR code. Then, a validation result is displayed. It is important to emphasize that the entire check-in process for passengersis fully automated, with no external intervention apart from the passenger themselves.

If the passenger enters the correct email address and QR code ticket, the system decrypts the encrypted content, and extract the passenger’s details and ticket information, and display the ticket information such as the platform, bus plate number and passenger name to the passenger, as shown in Fig. 24.



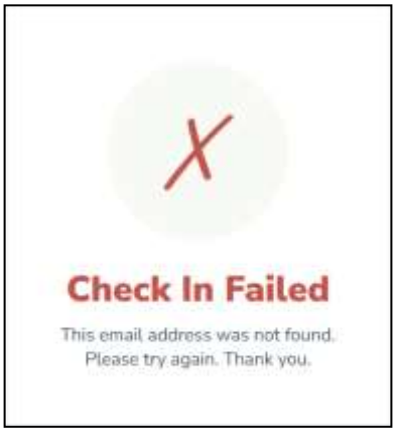
Check In Successful

Platform : P14 | Plate No. : WVM6933

Dear Chai Sheng Hui,
Now you can boarding your bus. Thank you.
Enjoy your trip!

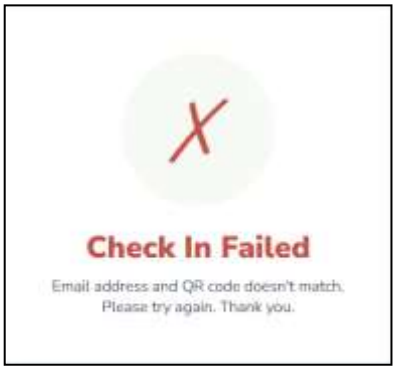
Fig. 24: Check-in successfully

At the same time, the system updates the status from “booked” to “checked” and passenger can view and confirm the status of the QR code ticket from the QR code page after check in is completed, as shown in Fig. 22. Besides, if the passenger enters the wrong email address or QR code ticket, the system cannot decrypt the encrypted content, and display the result of “Email address and QR code doesn’t match” to the passenger, as shown in Fig. 25. Also, Fig. 26 displays the result of “This email address was not found” when the passenger enters an email address that is not registered in the system.


Check In Failed

This email address was not found.
Please try again. Thank you.

Fig. 25: Check-in failed (Email address and QR code does not match)


Check In Failed

Email address and QR code doesn't match.
Please try again. Thank you.

Fig. 26: Check-in failed (Email address was not found)

CONCLUSION

This paper proposes a system that allow passengers to book tickets and obtain QR code ticket. The purpose of this system is to make it easier and more convenient for passengers to obtain their QR code tickets after booking is complete without wasting any time. Besides, due to the situation of the pandemic Covid-19, queuing up for ticket exchange may increase the physical contact between people and the risk of infection. Therefore, this system can help to reduce this problem. Moreover, the QR code ticket adopts the AES encryption mechanism to ensure that the passenger information is protected from being used by others.

REFERENCES

[1] Awati, R. (2022). Rijndael. Available at: https://www.techtarget.com/searchsecurity/definition/Rijndael (Accessed at 7 Nov 2022)

[2] Bernstein, C. and Cobb, M. (2021). Advanced Encryption Standard (AES). Available at: https://www.techtarget.com/searchsecurity/definition/Advanced-Encryption-Standard (Accessed at 11 Nov 2022).

[3] BusOnlineTicket Pte Ltd (2023). Get bus tickets online in Malaysia & Singapore at BusOnlineTicket.com. Available at: https://busonlineticket.com/ (Accessed at 5 May 2023).

[4] Mustafeez, A.Z. (2022). What is the AES algorithm? Available at: https://www.educative.io/answers/what-is-the-aes-algorithm (Accessed at 3

Dec 2022).

[5] Noordin, K.A. (2017). Start-up >: Buying bus tickets in the digital age. Available at: https://www.theedgemarkets.com/article/startup-buying-bus-tickets-digital-age (Accessed at 19 Oct 2022).

[6] redBus. (2023). Bus Tickets Online | Best Bus Ticket Booking Platform - redBus Malaysia. redBus Malaysia. Retrieved February 8, 2023, from https://www.redbus.my/

[7] Richards, K. (2021). cryptography. Available at: https://www.techtarget.com/searchsecurity/definition/cryptography (Accessed at 22 Dec 2022).

[8] Khan, S., Jam, F. A., Shahbaz, M., & Mamun, M. A. (2018). Electricity consumption, economic growth and trade openness in Kazakhstan: evidence from cointegration and causality. OPEC Energy Review, 42(3), 224-243.

[9] Rimkienė, R. (2022). What is AES encryption and how does it work? Available at: https://cybernews.com/resources/what-is-aes-encryption/ (Accessed at 4 Jan 2023).

[10] Schulfer, S. (2020). Who Invented the QR Code? QR Code History & Creator. Available at: https://www.sproutqr.com/blog/qr-code-history (Accessed at 13 Feb 2023).

[11] Thanaditsayakun, K. (2023). Travel by Bus in Malaysia. Available at: https://bangkokattractions.com/travel-by-bus-in-malaysia/ (Accessed at 2 May 2023).

DOI: https://doi.org/10.15379/ijmst.v10i3.1846