University of Sheffield

Implementation of payment module of flight booking system using RSA algorithm

Asad Masood

Supervisor: Professor Georg Struth

A report submitted in fulfilment of the requirements
for the degree of MSc in Advanced Computer Science

in the

Department of Computer Science

September 11, 2019

Declaration

All sentences or passages quoted in this report from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this project and the degree examination as a whole.

Name: Asad Masood

Signature: Asad Masood

Date: September 11, 2019

Abstract

Air Travelling is increasing day by day because people are so busy in their work routine which demands travelling. A lot of people use air travel for leisure and tourism. Another factor for the increase in air travelling is low fares offered by airlines. People prefer air travel as it saves a lot of time. Therefore, a system is needed which can help the people to book the flights and pay for them securely. Due to this, an online payment system is developed to make the payment for flights easy and efficient. Because of so many attacks online, the payment system developed is trustworthy so that the users can fully rely on it and would not hesitate in entering their confidential details online. The system developed is an interactive web-based application which provides the user with an easy to use interface and secure payment system that is achieved using the RSA algorithm.

Contents

List of Figures

Chapter 1

Introduction

1.1 Setting the Scene

As we know, it is an era of digitalization and people use mobile phones, laptop and the internet for almost everything. Over the last couple of years, air travelling has been increased. People tend to travel a lot whether it is for business or tourism. According to Eurostat, in 2017 there is an increase in the air travelling in the European Union by 7.3% compared with 2016 [2]. As air travelling is growing rapidly, there should be a system which allows the user to book flights and pay for them without any hesitation. It is because there are lots of online attacks and the customers does not feel safe while providing their account details online. People nowadays are too busy especially the frequent fliers who travel quite often for business. It is therefore important to provide the users with an easy to use interface so they can book a flight conveniently and efficiently.

1.2 Aims and Achievements

The aim of the project is to provide an online payment system to the users which is interactive as well as secure so that the users do not feel reluctant while providing their credentials online. Therefore, A system is developed through which the users can pay for the flights they have booked. There are a lot of websites online which asks the user for irrelevant information in addition to the debit or credit card details which results in data stealing. To make the users feel safe while providing their card details online, the system developed is made secure using cryptography. The RSA algorithm is used to encrypt and decrypt the user's credentials. It is considered one of the most effective algorithms for security because of its key generation procedure using large prime numbers.

When the users enter their credentials for the first time, they will be encrypted using the public key generated through the RSA algorithm. These credentials will be stored in a database. When the user logs in the second time and enters the credentials again, the system will check if it is the same user, then it will retrieve encrypted credentials from the database. The system will then decrypt those credentials using private key pair generated using the RSA algorithm. Lastly, the system will check if the decrypted credentials and entered credentials are the same. The payment will be considered successful if they are same and unsuccessful otherwise. SSL (Secure Socket Layer) is used for the secure connection so that no hacker can intervene in the payment process. SSL is used to establish a secure connection between the web browser and the server.

The security of the RSA algorithm lies in the multiplication of large prime numbers. To make the algorithm secure, prime numbers generated can be of 2048 bits in length. In the RSA algorithm, initially, we have to multiply the prime numbers which result in a composite number. Therefore, it will be difficult for the computer to factorize a composite number to get two prime numbers, which makes the RSA algorithm secure.

1.3 Overview of the Report

The following chapters highlight the content of the report.

Chapter 2 outlines the literature review carried out for this project. In this chapter, there is a detailed analysis of numerous articles which helps to analyse the online payment methods as well as its security. The algorithm and techniques to make the payment secure have also been explained.

Chapter 3 is aimed at defining the requirement of the project. It describes the functional and non-functional requirements of the project with the help of use-case diagrams.

Chapter 4 illustrates the design of the system. Mock-ups and final user interface of the system have been described along with the sequence diagram to show the workflow of the system.

Chapter 5 explains the detail implementation of the system using code snippets. Connection to the database, setting up the environment has also been discussed.

Chapter 6 describes the achievements of the project along with testing and results obtained. It also details the difficulties arise during the project and future work has also been discussed.

Finally, chapter 7 summarises the previous chapters and provides a conclusion.

Chapter 2

Literature Survey

E-commerce is increasing and expanding day by day. When you buy something from the internet, you are dealing with some kind of e-commerce. Payment is considered a chief ingredient of any online shopping system because without payment you cannot buy anything. Payment should be secure so the user can rely on the shipping website. This chapter details the payment systems which are currently in operation. The online payment system will be discussed along with the security of payment and overall web applications. The algorithms and methods used to implement the secure payment system will be explored and analysed.

2.1 Existing Systems

In this section, I will discuss the payment module of the airline reservation systems which are currently operational. I have studied some case studies and gone through the procedures of reservations of different airlines.

The airline alliance I have studied are OneWorld, Sky Team and Star Alliance and there are many different airlines. Some of them are American Airlines, Air Canada that operates in North America. KLM and British Airways operate in Europe. Skyscanner is another website which provides flights from different airlines [13].

Worldpay

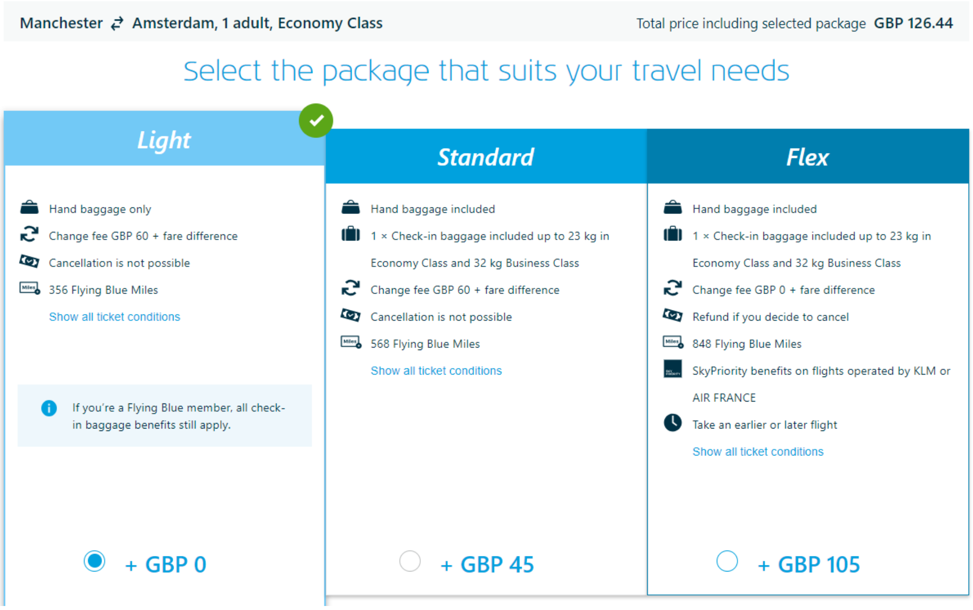
Worldpay is a payment processing company which provides the financial systems and end-user payment gateway to different companies. Some of the airlines use this gateway to process the payments. Emirates, British Airways, United Airline etc. are some of the examples that use this system [20].

Worldpay is famous because it provides 24/7 support to its customer. Whenever there is a problem in any payment procedure, the support team of Worldpay can be contacted to resolve any issue. Security is the most important aspect of any payment processed online or offline. Another reason due to which airline chose this payment provider is its advanced fraud protection. It provides a secure payment gateway to merchant and customer [20].

Now I will discuss the payment procedures of some of the airlines.

KLM

KLM is one of the leading airlines in the Europe region originating from the Netherlands and an alliance of SkyTeam [4]. The payment system of this airline is very well managed. After booking a flight, the user has been given a choice of selecting a package. Each package has its own benefits. For example, the Economy Flex has fewer benefits such as child fare is 75% of adult fare whereas Full Flex has more benefits which include a partial refund for cancellation after departure and stopovers are permitted [8]. There are many other benefits to different packages. Apart from credit card details user can enter vouchers or discount codes to process the payments.



Manchester ⇄ Amsterdam, 1 adult, Economy Class

Total price including selected package GBP 126.44

Select the package that suits your travel needs

✓ Light

Hand baggage only

Change fee GBP 60 + fare difference

Cancellation is not possible

356 Flying Blue Miles

Show all ticket conditions

If you're a Flying Blue member, all check-in baggage benefits still apply.

● + GBP 0

Standard

Hand baggage included

1 × Check-in baggage included up to 23 kg in Economy Class and 32 kg Business Class

Change fee GBP 60 + fare difference

Cancellation is not possible

568 Flying Blue Miles

Show all ticket conditions

○ + GBP 45

Flex

Hand baggage included

1 × Check-in baggage included up to 23 kg in Economy Class and 32 kg Business Class

Change fee GBP 0 + fare difference

Refund if you decide to cancel

848 Flying Blue Miles

SkyPriority benefits on flights operated by KLM or AIR FRANCE

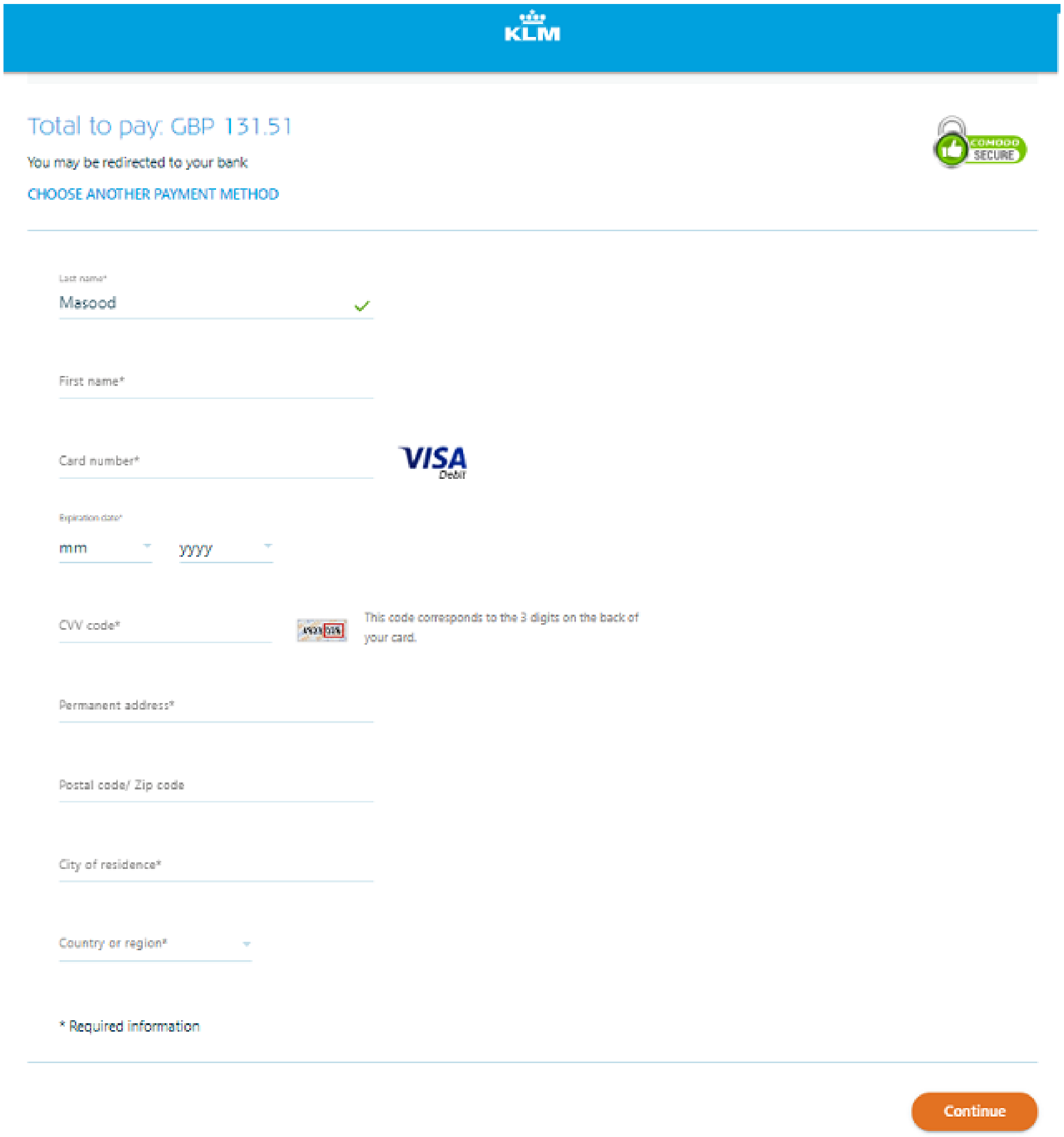
Take an earlier or later flight

Show all ticket conditions

○ + GBP 105

Figure 2.1: KLM Packages [11]

Some airlines also accept PayPal and other payment options. After selecting any package user will be directed to the payment window where he will select the payment option and add any voucher if available. Then user can enter his/her card details to process the payment which is then redirected, for a while, to the website of the bank of user and after confirmation redirected back to the airline website with the specific response.



KLM

Total to pay: GBP 131.51

You may be redirected to your bank

CHOOSE ANOTHER PAYMENT METHOD

Last name*

Masood ✓

First name*

Card number* VISA Debit

Expiration date*

mm yyyy

CVV code*

This code corresponds to the 3 digits on the back of your card.

Permanent address*

Postal code/ Zip code

City of residence*

Country or region*

* Required information

Continue

Figure 2.2: KLM payment details [11]

WestJet

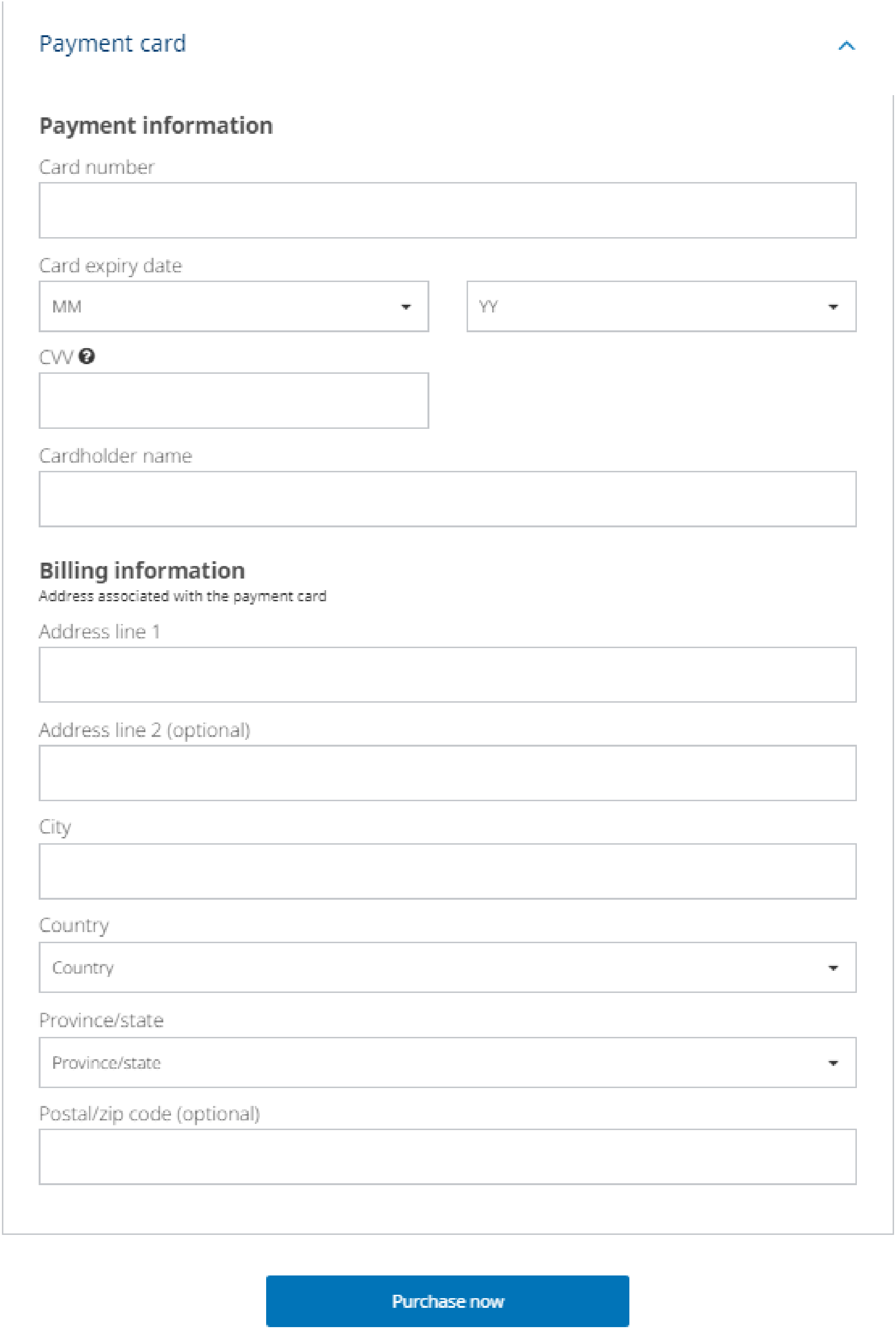


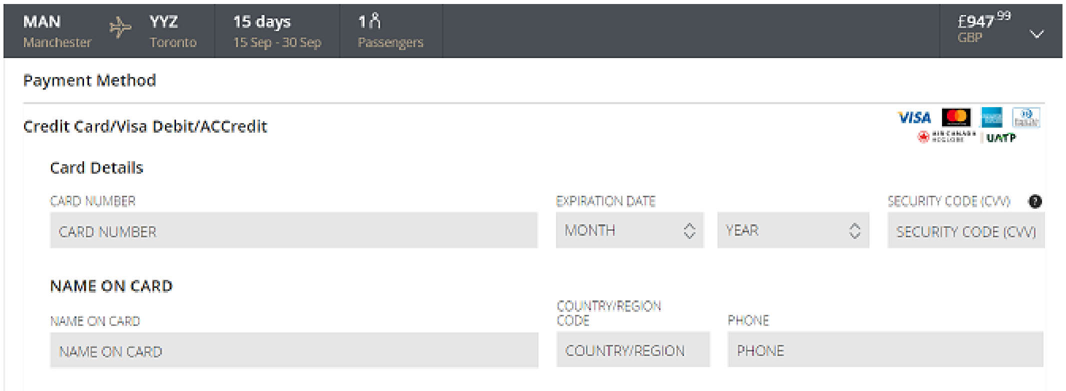
Figure 2.3: WestJet payment details [15]

WestJet is a 2nd biggest airline in Canada [27]. After selecting flight, user will be directed to the payment procedure platform where he selects payment plans and enters his card information along with the billing information.

WestJet also provides the Universal Air Travel Plan (UATP) [16]. It has a lot of benefits such as security by controlling online frauds. It also provides savings for its customers.

Air Canada

Air Canada is the largest airline of Canada both internationally and for domestic flights as well. The payment procedure of this airline is also similar to that of discussed earlier.



MAN Manchester → YYZ Toronto | 15 days 15 Sep - 30 Sep | 1 Passengers | £947.99 GBP

Payment Method

Credit Card/Visa Debit/ACCredit

Card Details

CARD NUMBER: CARD NUMBER

EXPIRATION DATE: MONTH, YEAR

SECURITY CODE (CVV): SECURITY CODE (CVV)

NAME ON CARD

NAME ON CARD: NAME ON CARD

COUNTRY/REGION CODE: COUNTRY/REGION

PHONE: PHONE

Figure 2.4: Air Canada payment [1]

After selecting outbound or return flight, it provides the user with a lot of options from different packages. Then the user selects a particular package and proceeds to payment. Now the user will enter his card details. The user interface of Air Canadas payment page is easy to use.

2.2 Online Payment Methods

In the past two decades, online shopping has increased rapidly. As it is an internet era, people are preferring online shopping from the ease of home instead of going out. Due to the increased demand for online shopping, there are a lot of the businesses that are growing at a great speed and many new businesses have been introduced in the market.

According to Statista, retail e-commerce sales worldwide has grown from US$ 1.3 trillion in 2014 to US$ 2.8 trillion in 2018 and is expected to reach US$ 4.9 trillion in 2021 [29].

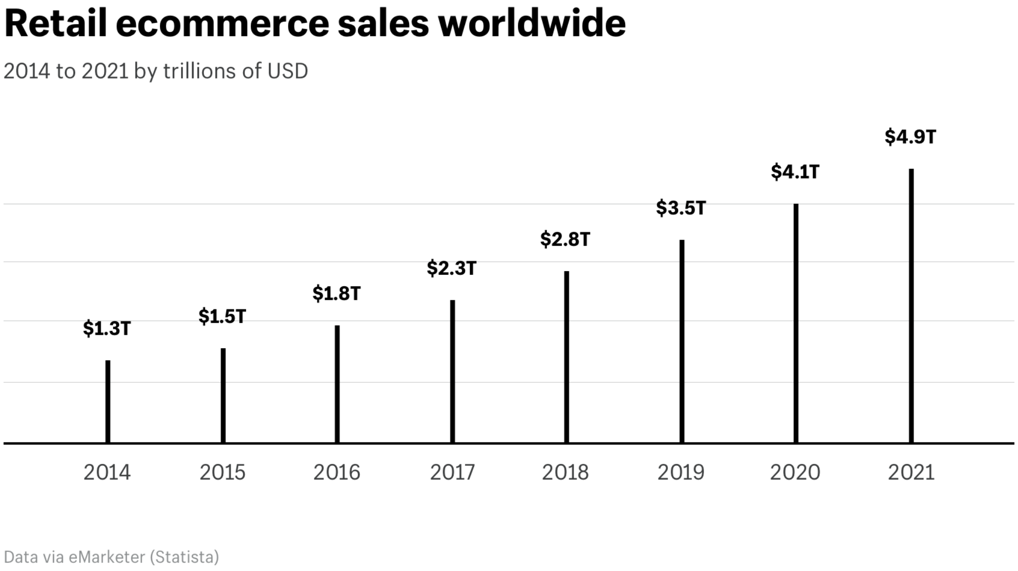


Figure 2.5: Retail E-commerce Sales worldwide [29]

Online Payment plays a pivotal role in online shopping. As online shopping is increasing day by day, it is the responsibility of the online business owner to provide customer satisfaction. The things customers really want when they are shopping online is the product of good quality and a secure method of payment. The Payment procedure must meet the satisfaction of the customers. So, the payment method should be easy to use and must be secure. If the payment system used on the website is secured and according to customers needs, the business will grow rapidly because customers can rely on that and would not hesitate to provide their payment details to the website.

Debit and Credit Cards

There are a lot of payment methods available online. The most popular of them is payment through debit and credit cards. Some other payment methods that are also used these days include bank transfers, smart cards and e-wallets etc. Previously, there are offline payment methods in which the customers hand over their cards to the seller and he then uses the card machine to process the payment. But now its the era of contact-less payments which has its advantages and disadvantages.

Now I will discuss how the online payment works.

After filling up a cart, the customer will be redirected to the payment page where he will pay for the products he bought. Usually, there is a form on the payment page which asks the user for the card number, name on card, card expiry date and three-digit code on the back of the card which is known as Card Security Code (CSC) or sometimes called Card Verification Value or CVV.

The user will provide his card details on the form. These details are then passed to the merchant either through payment gateway or any other payment system. This information is then passed to the acquirer which then asks for the confirmation from the respective card provider (for e.g. VISA or MasterCard). The card provider then submits this information to the issuing bank which confirms the card details and transaction details and sends the response back to the merchant or retailer. The process can be seen in the image below.

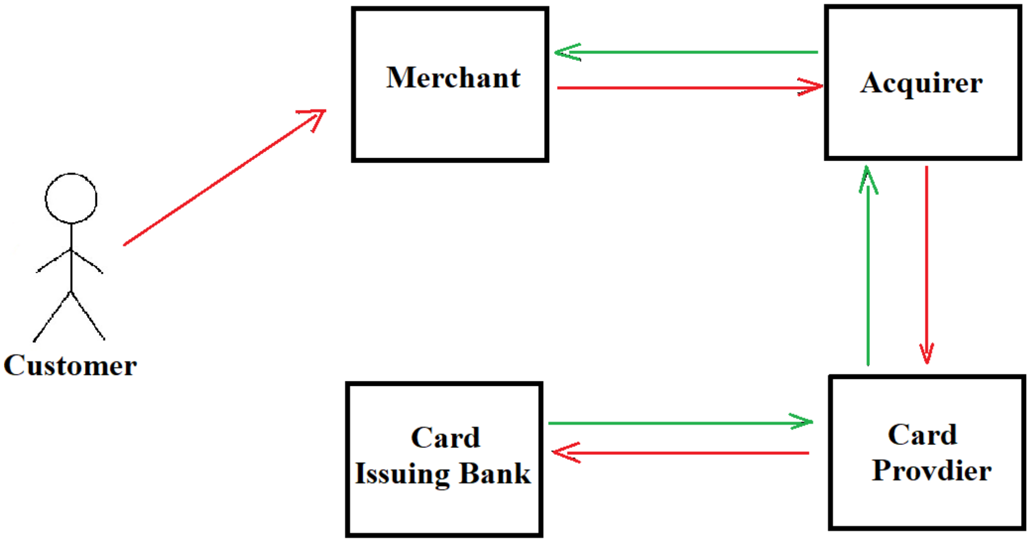


Figure 2.6: Card Payment Process

Stripe

There are a lot of payment gateways available in the market which different companies used in their businesses. Payment gateways are very famous these days for any business online because of their acceptance of multiple types of cards, security, billing systems etc. One of the most famous payment gateways is Stripe [14].

It provides API which can be used in any online websites to make the payment module of the system. After implementing this API, it will handle everything on its own related to payment. There are a lot of companies around the world which uses Stripe for their business. Some of them are amazon, booking.com, Expedia, uber etc. [14].

PayPal

Paypal is another payment gateway in the market which is similar to Stripe and used widely around the world. It offers a lot of options for the seller as well as the buyer. There 24/7 support is available for the business. Advantages of Paypal is that it provides all forms of payment which includes credit or debit cards, contactless payments and Paypal [12].

As a buyer, the users can set up their account which acts as a wallet. The users can transfer the money from their bank account to Paypal because it is very secure and has a lot of advantages for both the buyer and the seller which also includes cashback offers. With the help of Paypal, the users can transfer money to and from their bank accounts. Paypal also provides a debit card which can be used to make real purchases. Paypal is accepted all over the world and a lot of currency are available with it.

The alternative airline is a site which provides airline tickets all around the world using Paypal and Paypal credit. This website accepts Paypal on domestic and international flights from over 650 airlines. 26 different currencies are available to make payment using Paypal and 3 currencies for Paypal credit. There are a lot of coupons and vouchers available when the user checks out with Paypal [10].

Paypal also provides APIs for the developer to set up the sandbox account which they can use for their product while it is in the development stage. The main advantage of Paypal is that the user does not have to worry about carrying cards. As Paypal is all digital, payments can be made at any time and at any place.

Apart from these payment gateways, there are a lot of other payment methods for different types of goods. There is a payment system that can be used for buying information items over the internet. Flash disks are used for this system. This system is useful for micropayments. The protocol for flash payments is based on centralized accounts in a trusted third party server. This protocol can be integrated with any ecommerce website [26].

2.3 Security of the Web

As discussed earlier, online shopping has been increased rapidly in the past decades. Security of online payment must be considered when developing a business online. This is the only way to expand the business so that the customers can trust the payment methods used for the business. If the payment method used is not secure, there are a lot of opportunities for cybercriminals. Online transactions and payment details should be kept secure so that any hacker who tries to intercede the transaction can not be able to get hold of the payment and transaction details of the customer.

In every industry, whether it is online shopping, hotel or airline, security plays a major role. For example, when the customers books a flight using any online site, they enter a lot of sensitive information before payment as well as during payment. Before payment, they add their passport details, contact details and during payment, their card details to make payment. If the website is not following any secure protocols, there are a lot of cyber-attacks which can take place. If they are using debit or credit card for the payment, their card information can be lost which can be used by cybercriminals for their on use. Similarly, if they are using some kind of wallet to make the payment, it can also be used in the same way by cybercriminals. When the customers enter their contact details as well as passport details, a cyber attacker could use these details to blackmail him. And in some other scenario, they can use this information for other organized crimes which can include terrorism.

Hackers use malicious software to for a lot of cybercrimes. Identity theft is one of the most popular cybercrimes in which hacker uses someone elses identity to commit crime which can be stealing money, blackmailing or some other information stealing. For example, when booking a flight, the customers have to enter their personal information which includes name, address, ID numbers etc. The hackers can intervene in the system and can steal this information unless the system is very secure. This stolen information can be used by hackers for stealing information, frauds etc. According to the new Javelin strategy and research study, 16.7 million people became the victim of identity fraud in the United States [7].

Therefore, it is necessary for the business to invest in the security of online payment so that the users can use the system without hesitation. It is also important for the customers to not provide their sensitive information unless they are very sure about the security of the system.

There are some other types of attacks that are carried out by the hackers which are discussed below.

1. SQL Injections

Almost all of the website uses some kind of database. The database is used to store data related to the application such as user information, payment information and all other kinds of entities that are needed for the website to run properly. SQL queries are used for any action that is needed to perform on these entities. For example, when the users enter their credentials to log in, an SQL query is run at the back end which checks if the provided credentials are available in the database or not. Consider the database has a table with the name 'user' and has two columns, 'username', 'password'. The general syntax of the query is something like this to check if the username and password are correct, ''SELECT * FROM user WHERE username = 'Asad' AND password = 'abc123' ''. If the database has the above credentials, the user will log into the system.

Here SQL injection attacks come into play. If an attacker wants to log in to this system, he will enter some kind of statement. As a result, he might get access to all the usernames and passwords. One of the examples of SQL injection is 1=1. When the attacker tries to log in to the system, the statement he will enter will be something like this ''attacker OR 1=1'' for username and ''password OR 1=1'' for the password. The SQL query of this statement will become ''SELECT * FROM user WHERE username = 'attacker OR 1=1' AND password = 'password OR 1=1' ''. As we know that 1=1 will always be true, hence, the query will return all rows which will give the attacker access to the database. In short, SQL injection is a type of attack which execute malicious SQL statements [19].

SQL injection attacks must be prevented and the only way to protect a website from SQL injection attacks is to use SQL parameters. These are the values which are added to the SQL query at runtime. As a result, only those parameters were added to the SQL which really belongs to the column. For example, by adding the SQL parameter, SQL query will become ''SELECT * FROM user WHERE username =? AND password =?''. After that query will be executed with the values from the correct column. Prepared Statements are also used along with this to prevent SQL injection.

2. Cross-Site Scripting

Cross-Site Scripting or XSS is another type of attack that is mostly found in web applications. It enables the attacker to inject some kind of script into a web page. The script is injected at a client-side which is a users' web browser. The attacker can inject a malicious script which is generally a JavaScript or HTML code into the client's browser and can carry out any task he wants. Whenever the web page is loaded, this malicious script will also run and attacks the system. An example of cross-site scripting is when a user received a mail and it contains a link which in reality is a malicious JavaScript. When the user clicks on that link, an HTTP request will be generated from the victim's web browser and will be sent to vulnerable web application [5].

Cross-site scripting can be very dangerous. It can access users location, webcam and even some files from the users' computer [6]. To avoid cross-site scripting, there are a lot of methods. Some of them are discussed below.

The first rule is not to put the data you don't trust in HTML. There are some sanitization methods that can be used to prevent cross-site scripting. With the help of sanitizer, the HTML will be cleaned before executing. No script can run as it will remove all the tags before executing any HTML code. Escaping is also used to prevent XSS attacks. Escaping the user input will change the JavaScript code that the hacker has intentionally added in the HTML. JavaScript contains a lot of symbols such as angle brackets ⟨ ⟩, round brackets ( ), slashes. To prevent HTML from executing these type of JavaScript, escaping is used which escape all these characters of JavaScript and makes the HTML safe.

By avoiding these attacks, the web application will become a lot safer and trustworthy. As a result, the customers will not hesitate in providing their sensitive information such as credit card details.

Preventive Methods

To prevent frauds and cyber attacks, there are a lot of methods available in the market which I am going to discuss below.

1.Multi-Factor Authentication

The method that can be used to prevent cyber attacks is multi-factor authentication. Multi-factor authentication is a technique in which the user has to authenticate himself through a system which considers multiple factors. These factors can be a physical object, fingerprints, secret code, any place or combination of these. Two-factor authentication is a type of multi-factor authentication that is used to confirm users identity. An example of this type of authentication is withdrawing money from the ATM where the user has an ATM card which is a factor and its PIN which is another factor. The user should have both things to withdraw money from the ATM [9].

This authentication is currently used by a lot of systems and is progressing. According to new European Union regulation, Strong Customer Authentication will come into force in September 2019. According to this report, only 1-2% online transactions require the cardholder authentication which is expected to rise to 25% [25].

2.Trusted Emails

A research study describes the fraud detection technique which is classified into two techniques, Supervised and Unsupervised technique. It also explains the fraud prevention techniques [22]. One technique is address verification service where the merchant compares the billing address of the customer and the address of card issuer from the bank. Another technique is checking CVV which is the three-digit code on the back of the card. The fraudster can obtain card number and all other card details from the internet but he cannot have the CVV of the card unless he has the physical card of the person. Last but not least, another technique is proposed which is the extension of the address verification scheme. Along with confirming the address of the customers, it also verifies their email. E-mail should be the same as that of stored against the customer in a bank [22].

Web Security

Now I will discuss how to make payment secure from the developer perspective. According to Head First Servlets and JSP, there are four types of security domains in servlets [24].

1. Authentication
2. Authorization
3. Data Integrity
4. Confidentiality

Authentication

Authentication is a technique which is used to make the web application secure. There is some communication between the browser and the web container of the web application. When the browser receives a request from the container, it checks the header for any authentication and asks the user to enter a username and password. If there is no authentication defined then the response of 401 is sent to the container which means unauthorized to view the page. And if the header has some kind of authentication defined and the user enters a username and password, then the browser sends a positive response with a respective JSP page.

There are 4 types of authentication that can be added in the deployment descriptor of the application.

1.BASIC

This type of authentication transmits the login information in encoded form. Although it uses Base64 encoded scheme, yet this type of authentication is weak.

2.DIGEST

This type of authentication transmits the information in more secure way and use advanced encryption.

3.CLIENT-CERT

It is one of the most secured type of authentication as it transmits the login information in most secured way using Public Key Certificates (PKC). This type of authentication is mainly used for the business and client must have some certificates to use this authentication type.

4.FORM

It is considered one of the least secured authentication as it lets the developers to create their own HTML form where the customers can enter their sensitive information.

Authorization

Authorization is also used to make the web application secured. In this type of security only authorized users are allowed to access some web content. The administrator of the website maintains the data by specifying username, passwords and roles to the users.

The first step of authorization is to define roles. Container will determine who is authorized for certain request and which container and servlet to invoke. Roles can be defined in the deployment descriptor of the web application. Second step is defining resource or method constraints which is also done in the deployment descriptor which specifies about the method invocation (GET or POST) and URL pattern.

When the customers enters their payment information, only certain bodies are allowed to access that information which includes card provider, acquirer etc to authenticate the payment.

Data Integrity

Data should be accurate throughout its lifecycle. There must not be any change in data along its way. There are certain steps which must be considered to ensure data integrity.

Confidentiality

Data confidentiality plays a very important role in web security. Data must be secured and should not be seen by anybody along the way. Certain data should be visible to certain people. When the customers enters their payment details, it should be kept confidential.

So, to make the online application secure these steps must be considered. Other general tips to make the payment secure is to use strong passwords. It means use password which is a combination of small, capital letters, digits and special characters. Public WiFi should not be used when making online payments.

Now I will discuss the encryption method which is used by a lot of online payment systems.

2.4 RSA Algorithm

In this section, I will describe the working of the RSA algorithm which is an encryption algorithm that is used in a lot of online payment systems as a base algorithm.

2.4.1 Public Key Cryptography

Cryptography is a combination of two words 'crypt' which means hidden and 'graphy' means writing. So, cryptography means hidden writing. In computer science, cryptography means to secure information based on some mathematical concepts in such a way that it will become hard to decrypt or decipher that information [18].

I will discuss two types of cryptography:

1. Symmetric Cryptography
2. Asymmetric Cryptography

In cryptography, a message that we want to secure and then transferred to some other place is called Plain Text. In symmetric cryptography, plain text is converted to ciphertext using a key and then transferred to someplace else and decrypted using the same key to retrieve the original message.

Ciphers are used to transmit the data from one place to another securely. There are a lot of techniques to encrypt the data. In symmetric encryption, there is a type of cipher which is known as a monoalphabetic cipher in which a single alphabet is used in place of the alphabet that is needed to encrypt. For example, if I want to encrypt the message, 'My name is Asad'. To encrypt 'a', I can use any letter such as 'k' which makes the message 'My nkme is kskd'. Similarly, there is another type of cipher which is known as a polyalphabetic cipher. It is similar to the monoalphabetic cipher. The only difference is instead of a single letter to represent another letter, we can use many different letter.

Caesar cipher is another example of symmetric encryption. It is also known as shift cipher because in this cipher each letter is replaced by a letter which is some shifts after the alphabet. For example, after a shift of 5, G would become B [3]. It can be the right shift or left shift depending upon the situation. For example, we have a plain text, ' HELLO WORLD' and we have to shift it right with a shift of 3. After ciphering the plain text, it will become, 'KHOOR ZRUOG'.

These are the ciphers which are based on asymmetric encryption. However, these ciphers are not secure and considered weakest of all ciphers because they are vulnerable to brute force attacks.

In Asymmetric cryptography, plain text is converted to a ciphertext and then this ciphertext is transferred to someplace else and decrypted using some other key to retrieve the original message.

In short, for asymmetric cryptosystem, we have the same key for both encryption and decryption while in an asymmetric cryptosystem, we have a different key for both encryption and decryption. There are two types of keys public and private keys. The public key is visible to everyone whereas the private key is only visible to one person who has the ciphertext and wants to decipher it.

Lets understand the concept of asymmetric cryptosystem using a simple example.

Let us assume there are two persons Chris and Dave. Chris wants to send some secure message to Dave. So, both Chris and Dave have their own pair of public and private keys. They both will exchange their public keys. Then Chris will encrypt his message using Daves public key and send it to Dave. When Dave receives the encrypted message, he will decrypt the message using his private key and can send the reply to Chris using the same method. If an attacker wants to interfere in the communication between Chris and Dave, he must have to have private keys of both Chris and Dave to decrypt the message but since Chris and Dave only exchange their public keys, an attacker cannot do anything with the encrypted message.

The asymmetric cryptosystem is also known as public-key cryptography. This type of cryptography is used at a lot of places such as to make the web connection secure SSL. There are a lot of algorithms available on the internet which uses one of these two cryptographies. These include Advanced Encryption Standard (AES), Diffie-Hellman key exchange, Rivest Shamir Adi (or RSA) algorithm etc. Now I am going to discuss RSA algorithm.

2.4.2 RSA Algorithm

In 1977, three professors of MIT, Rivest, Shamir and Adelman proposed an algorithm to encrypt and decrypt messages which use the asymmetric or public key cryptosystem [30]. According to this algorithm, the messages can be encrypted and decrypted using different keys. The public key is used to encrypt the message whereas the private key is used to decrypt the message. It follows the same principle of public-key cryptography or asymmetric cryptography where different keys are used for encryption and decryption. The basis of this algorithm is that it is very hard for a computer to factor a large composite number to retrieve two prime numbers. This algorithm is used in a lot of application such as emails, chat applications and virtual private network or VPN. It is also used in digital signatures to make the signature secure. Although the RSA algorithm is very secure, it is not used to encrypt the whole files because it is rather slow. The reason this algorithm is slow because of the huge calculation performed for encryption and decryption. In real-world cases, the bit length of the prime numbers used should be at least 1024 bits to make the algorithm secure. These days some applications are considering to increase the bit length to be 2048 or even more which results in double the size of number due to multiplication. The RSA algorithm works in 3 major steps.

Algorithm for key generation

In this step, two keys are generated which is done step by step.

1. The first step is to select two random prime numbers 'p' and 'q'. Prime numbers chosen must be very large. Prime numbers are necessary for this cryptography method because it takes very less time to multiply two large prime numbers and very hard and takes a lot of time to do the reverse. In simple terms, to break a very large number into prime factors is a very intensive task for the computer. This step is the basis of the RSA algorithm that it is almost impossible for the computer to factorize a composite number to retrieve two prime numbers.

2. After selecting two prime numbers, multiply these two numbers p * q to obtain a number 'n'.

3. In this step, multiply p and q after subtracting 1 from each, i.e. (p - 1) * (q - 1) which is a number 'phi'.

4. Now choose a value for 'e'. 'e' should fulfil two conditions, one is e must be greater than 1 and less than phi and the other is the greatest common divisor or gcd of e and phi should be 1. In mathematical terms, 1<e<phi and gcd(e, phi)=1. This also means e and phi should be co prime. In real-world problems, e is sometimes chosen to be 65537.

5. In this step 'd' is calculated which is calculated using the extended euclidean algorithm. In simpler terms, d is nothing but a multiplicative inverse of e mod phi. The mathematical formula to calculate d is $d = e^{-1}$ mod phi. If somebody has a public key, one cannot easily determine the private key.

6. Public Key is generated in this step which is the pair of e and n. Public Key {e, n}

7. Private Key is generated in this step which is the pair of d and n. Private Key {d, n}

So, the key generation consists of 7 steps. Now encryption and decryption will be performed in the other two steps.

Encryption

In this step, encryption is performed on the plain text, 'M', to convert it into ciphertext, 'C'. The formula to obtain ciphertext is $C = M^e \ mod \ n$

Decryption

In this step, encryption is performed on the ciphertext, 'C', to convert it into plain text, 'M'. The formula to obtain ciphertext is $M = C^d \ mod \ n$

Now let's see the basic example of the RSA algorithm. Consider Plain Text to be a number M = 31 to make the algorithm simpler.

1. The first step is to select two prime numbers, let's say p = 11 and q = 5.

2. Calculating n = p*q, n = 11 * 5, n = 55.

3. Calculating phi = (p - 1) * (q - 1) = (11-1) * (5-1) = 40.

4. Calculating e. e = 7 so that 1 <e = 7 <40 and gcd (7, 40) = 1.

5. Calculating d. $d = 7^{-1} mod$ 20 . d = 23

6. Public Key = {7, 40}

7. Private Key = {23, 40}

Encryption: $C = P^e \ mod \ n = 30^7$ mod 55 = 35

Decryption: $P = C^d \ mod \ n = 35^{23}$ mod 55 = 30

The example shown above is a very basic working of the RSA algorithm. In real-world, the numbers are huge to make the algorithm secure.

2.4.3 Advanced Concepts

Although the RSA algorithm is widely used over the internet, a lot of research is going on this algorithm which results in some additions and advancements to the algorithm.

As we know that the RSA algorithm is slow due to the fact that it uses very large prime numbers which results in huge multiplication. Three new variants of the RSA were proposed in a paper to speed up the RSA decryption [23]. One of them is Batch RSA. In this version of RSA, instead of using one large component for a public key, use two small public components with the help of which we can decrypt two ciphertexts for the cost of one. This improvement in RSA will speed up the calculation of RSA [23].

Another type of RSA is Multi-factor RSA. In this version, the structure of the RSA modulus is modified. In simple RSA, for decryption, we use a modulus of n. In this version, the modulus will change to N = pqr and $N = p^2q$. First one is called multi-prime modulus and the second one is multi-power. After implementing this version of RSA, there is a speedup of approximately 2.25 over standard RSA [23].

Last but not least, balanced RSA is implemented. In this version of RSA, most of the work is shifted to the encryption part of the RSA. It is because in encryption, less work is done or we can say the implementation of RSA encryption is less intensive as compared to decryption. This version of RSA gives a large speedup but is limited to the browser because of some issues with the certificates [23].

In another research study, an improved version of the RSA scheme is proposed. According to this study, the RSA algorithm depends on the linear algebraic group over the ring of integer number mod a composite modulus n which is a product of two prime numbers. Square matrices are used to represent both plain text and ciphertext. The advantage of this scheme is more flexible as compared to the standard version of RSA because of the use of the matrices. This proposed scheme can be used along with some other cyphers to obtain strong encryption [21].

2.5 Security using SSL

Internet must be secure to make the most out of any business you are running online. To make the website secure, there are many ways which include implementing secure algorithms when retrieving user's payment information or any other secure information. Another way to make the website secure is to use HTTPS instead of HTTP.

HTTP stands for HyperText Transfer Protocol. It is a web protocol that is used to transmit data from one place to another over the world wide web. If the website uses HTTP, the data transferred is not secure. Any hacker can intervene in the transmission of the data. The data transferred using HTTP is written in plain text so the hackers can easily understand the information and can use it for their benefits. There can be phishing attacks on a website or receive from email. These attacks are like a clickbait when it wants the user to click a certain link. When the user clicks a link, he will land on a site which looks similar to his desire site but in actual its a scam site. The information that the user can enter on this site can be used against him or used to steal the information of the user. So there must be some methods to make the transmission of data from the browser to the server secure and also to avoid any phishing attacks. Here HTTPS comes into play. S in HTTPS means secure. Any website that uses HTTPS instead of HTTP is considered secure but there are a lot of other factors that must be kept into the account before providing any sensitive information such as card details.

SSL is used to make HTTPS secure. SSL is Secure Socket Layer that is used to make a secure connection between the web browser and the server by generating an SSL certificate. SSL certificate is a digital certificate of a web server that is issued by a Certification Authority. It verifies the identity of the webserver.

Let's consider a simple example which shows how SSL works. A user named Bob decides to book a flight through an online website which is very famous, let's say https://airline.com. When Bob enters this link, it means Bob's browser is asking for the secure page from the webserver of this website. The website sends its SSL certificate along with its public key to Bob's browser. This SSL certificate is digitally signed by a third party which is Certificate Authority (CA). When the browser receives the SSL certificate and a public key from the webserver of the site, it will check the digital signature of the issuer and compare these certificate. If the certificate matches, it's all good and a green padlock will appear at the URL which means the public key belongs to the webserver.

According to Nexcess, there are a lot of advantages of SSL. Identity Verification is one of them. When SSL is generated and matches with the certificate of a webserver, it guarantees that the information transmitted will reach the right place without any attack. Data integrity is also observed when using SSL. An attacker is always sitting in between when there is a transmission of data from one place to another place. SSL certificates confirm that the attacker cannot understand the data because of strong encryption used by SSL certificates. When the green padlock appears in the URL, it makes the website trustworthy to be used by the user [28].

These are all the methods that are used by the modern internet to make the internet secure. As e-commercing is growing rapidly, modern payment method, as well as security of these methods, should be the top priority of the one who owns the business. Information like credit or debit card details, passwords, social security numbers are very private information and must be kept secure. Whenever the users visit a site, they should have to be sure before providing their details. The users can only trust a website if it has modern methods implemented to make the payment secure which can also be judged from its URL if it is secure or not. The reputation of the website should also be good which can be known by its ratings by different users.

Chapter 3

Requirement Analysis

This chapter details the functional and non-functional requirements of the project and analyses the requirements using diagrams. The use case diagram is used to show the relationship between the user and the system. A workflow diagram is also shown which describes the workflow of the system step by step and also describes the pre-conditions and post conditions of the system. This chapter also outlines ethical, professional and legal issues associated with the project.

3.1 Functional Requirements

3.1.1 Payment

Payment is one of the most important aspects of online shopping. Whether you are buying grocery, clothing or movie tickets, payment plays a pivotal role. The airline also deals with the payment procedure very efficiently. All the payments must be very secure so the user can rely on the payment methods and on the airline.

In this application instead of the bank, the system will need to acts on its own. The system is needed to encrypts the card details of the user and decrypts is whenever it is necessary. Therefore, an encryption algorithm is required to process the payment so that the information the user will enter can stay hidden from everyone. When the users selects a flight, they sometimes feel hesitant while entering their card information because of the online attacks which could hack their credit or debit cards. So, an algorithm is needed which encrypts the card details.

This image below shows the basic use case diagram of how payment works in online shopping.

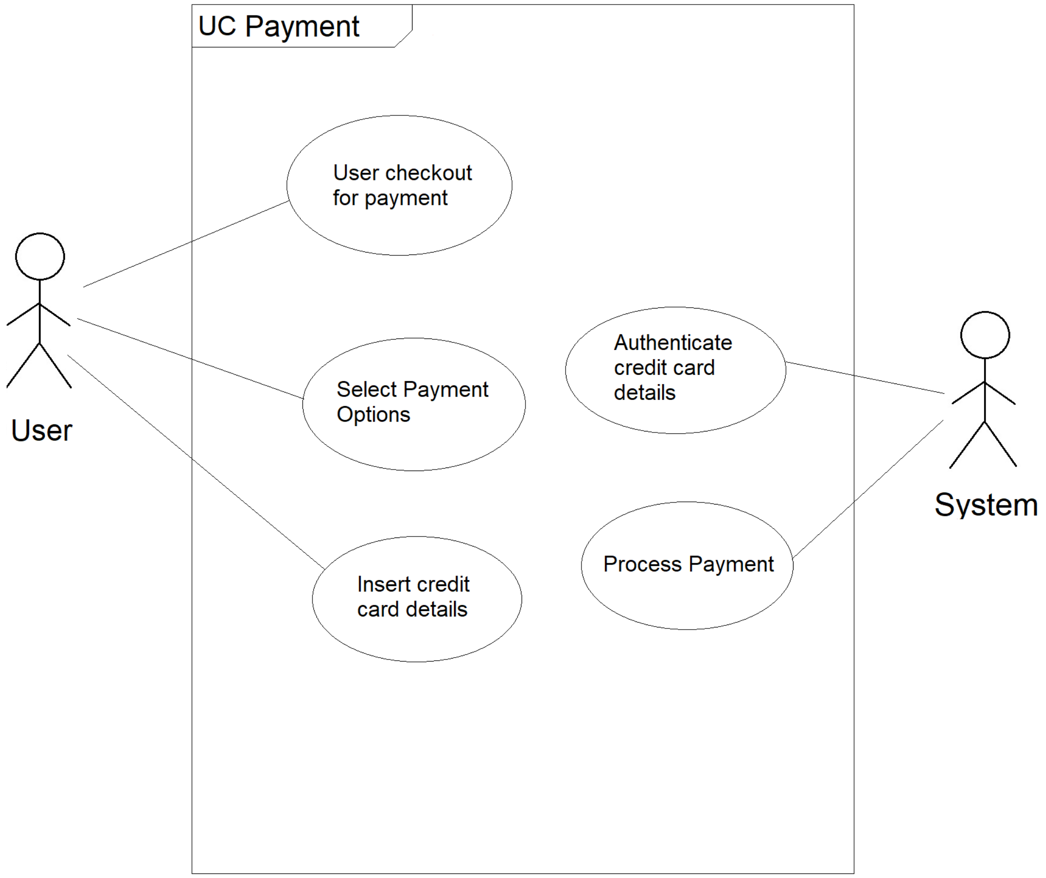
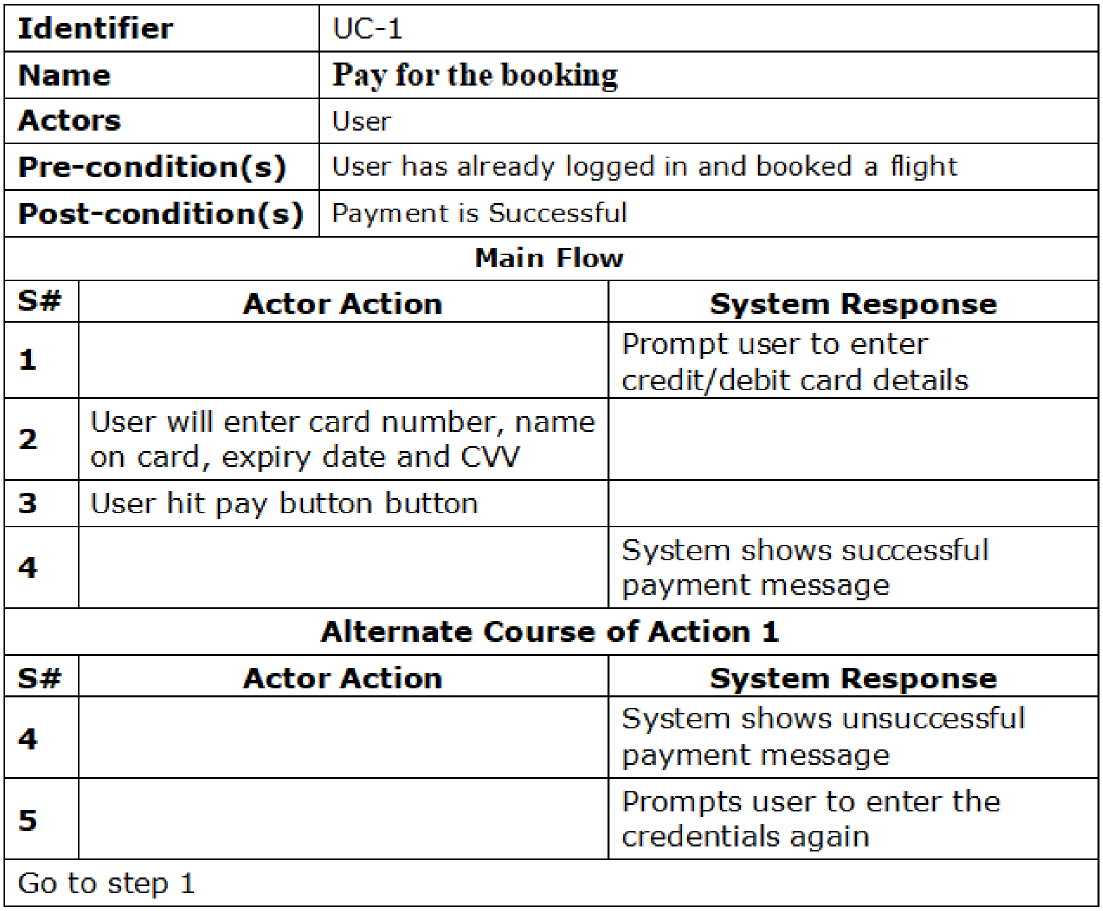

UC Payment
User
System
User checkout for payment
Select Payment Options
Insert credit card details
Authenticate credit card details
Process Payment

Figure 3.1: Payment Use case

Security of the payment is the most important part of any online purchase. Customers should fee safe while providing their valuable details. All banks are connected with the payment systems used by the airlines. There are also many payment providers which provide the services of payment security to the airlines.

In this system, the implementation of the payment system is made secured using various technologies which includes SQL injection, cross-site scripting. The application developed is quite secure as HTTPS is used instead of HTTP which establishes a secure connection between the web browser and the server.

The diagram below shows the use case diagram which describes the workflow of the system.



Identifier	UC-1	
Name	**Pay for the booking**	
Actors	User	
Pre-condition(s)	User has already logged in and booked a flight	
Post-condition(s)	Payment is Successful	
Main Flow		
S#	**Actor Action**	**System Response**
1		Prompt user to enter credit/debit card details
2	User will enter card number, name on card, expiry date and CVV	
3	User hit pay button button	
4		System shows successful payment message
Alternate Course of Action 1		
S#	**Actor Action**	**System Response**
4		System shows unsuccessful payment message
5		Prompts user to enter the credentials again
Go to step 1		

Figure 3.2: Payment Workflow

3.2 Non-Functional Requirements

This section outlines the non-functional requirements of the airline reservation system.

3.2.1 Usability

The interface of this system must be easy to learn and very interactive. Users can easily navigate on our website. Easy-to-understand help text and error messages must be provided. The user might not need any extra abilities to understand or navigate on our website.

3.2.2 Security

To make the payment secure, SSL should be used. The system should be protected against all SQL injections and any kinds of hacks.

3.2.3 Compatibility

This system must be compatible to all famous web browsers.

3.2.4 Extensibility

In future, I might include login functionality and other payment gateways. This system must be opened to incorporate new features, it would not be difficult to add new features.

3.3 Ethical, Professional and Legal Issues

3.3.1 Ethical Issues

The ethical review of the project has been clearly sought out. There are no potential issues in the project as most of the data, credit/debit card details, used in this system is dummy data. Data confidentiality would be a great ethical issue if we are dealing with the original data of airlines as these airlines hold personal data of the individuals. The customers are relying on the airline system when they were entering their personal information so it is important for the airline to deal with the users data confidentially and privately.

3.3.2 Professional and Legal Issues

The research carried out is properly used and cited. Data confidentiality of different airlines as described in ethical issues. It can be a professional issue as well because it is against the law to share the data of different airlines if the airline has forbidden it. To avoid this issue, I have tried to make the website secure. It will follow all the standards of BCS.

Chapter 4

Design

This chapter illustrates the overall design of the project starting from the overview. Mockups of the website are shown which helped in implementing the final user interface of the website afterwards. This section also describes the workflow of the system.

4.1 Design Overview

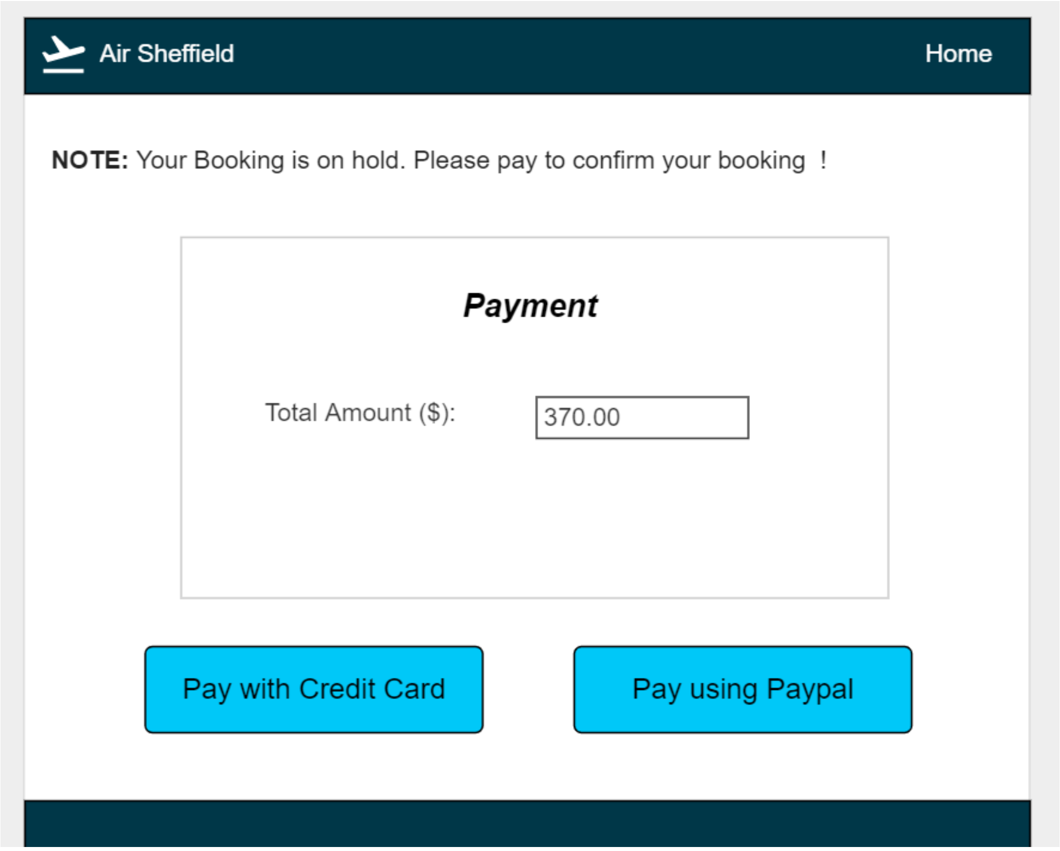
The overall design of the website is quite simple. The process of the design took some time to make it user-friendly and easy to use. After gathering all the requirements, the mockups were designed to represent the basic functionality of the system. As a part of mockups, a total of four screens was designed. An online tool is used to design the mockups of the website which is known as, moqups.com [ref]. After designing the mockups, they were reviewed and improved before implementing the final user interface.

All principles of Human-Computer Interaction or HCI were kept in mind while implementing the design of the website. The design of the system is kept minimal and has not extended where there is no need. The one-touch feature is used to design the system which means the users does not have to go through the long menus or a lot of clicks to reach their desired field. The colour scheme used is quite aesthetic and appealing. The user can easily navigate through the website and does not require any special knowledge to use the website. Informative messages were shown wherever it is necessary along with the alerts to make the system interactive.

Now I am going to discuss the mockups of the system that were designed to represent the overall functionality of the website.

4.2 Mock-ups

In this section, mockups of the website were illustrated along with a brief description of each screen.



NOTE: Your Booking is on hold. Please pay to confirm your booking !

Payment

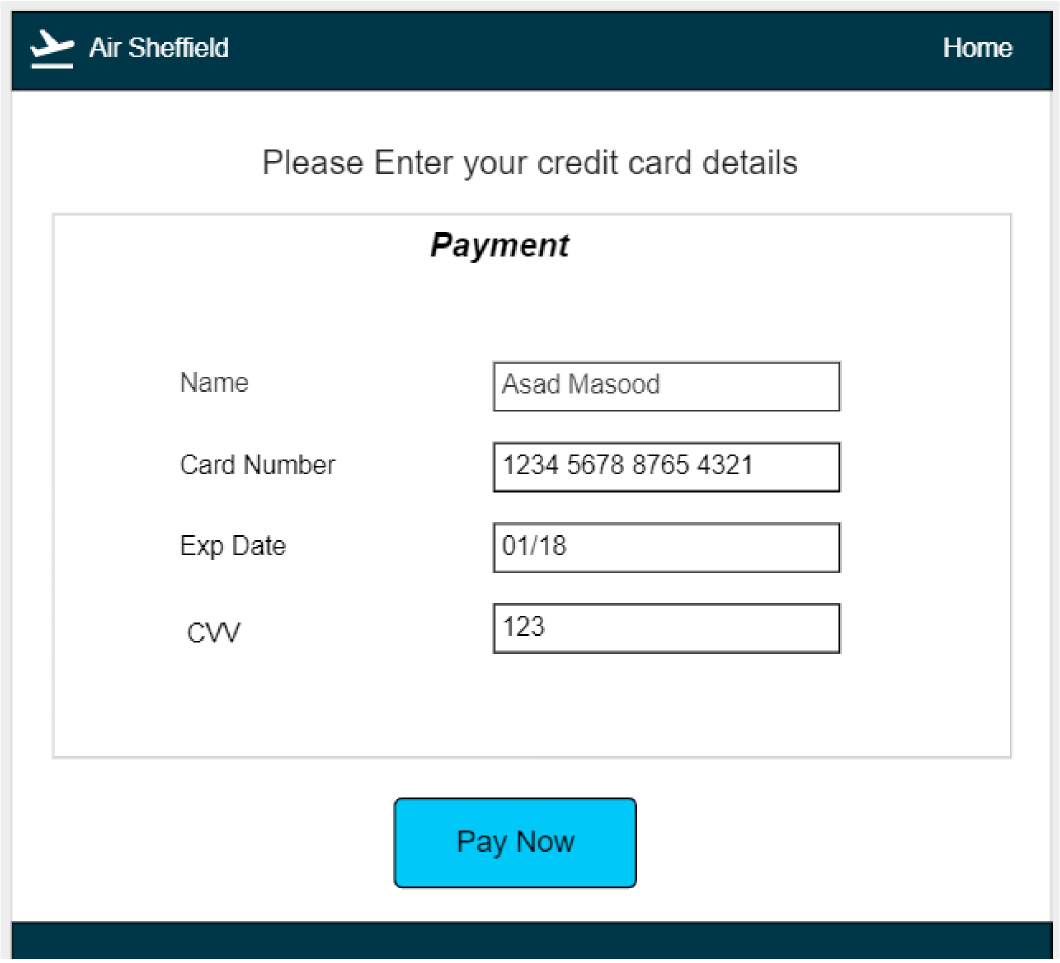
Total Amount ($): 370.00

Figure 4.1: Mock-up of main page

Figure 4.1 shows the main page of the website. The page is designed to make a look-alike of the airline website. Header and footer are shown. There's a navbar which shows the name of the website along with an icon. The navbar also contains the link to the homepage of the system. After that, a note is shown which states any announcement that the system wants the user to see. In this case, it is the announcement that the booking of the user has been held and will be confirmed when the user pays the amount.

The main body of the screen has a rectangular box which contains a heading of payment to let the user know about the payment page. Along with that, a label of the total amount is shown and a text field which states the total payment to be made.

Below the rectangular box, there are two buttons which state the payment method. One button states that the user can pay the amount using debit or credit card. The other button states that the user can pay the amount using PayPal. The user will just have to click any of these buttons on this screen to proceed.



Please Enter your credit card details

Name: Asad Masood

Card Number: 1234 5678 8765 4321

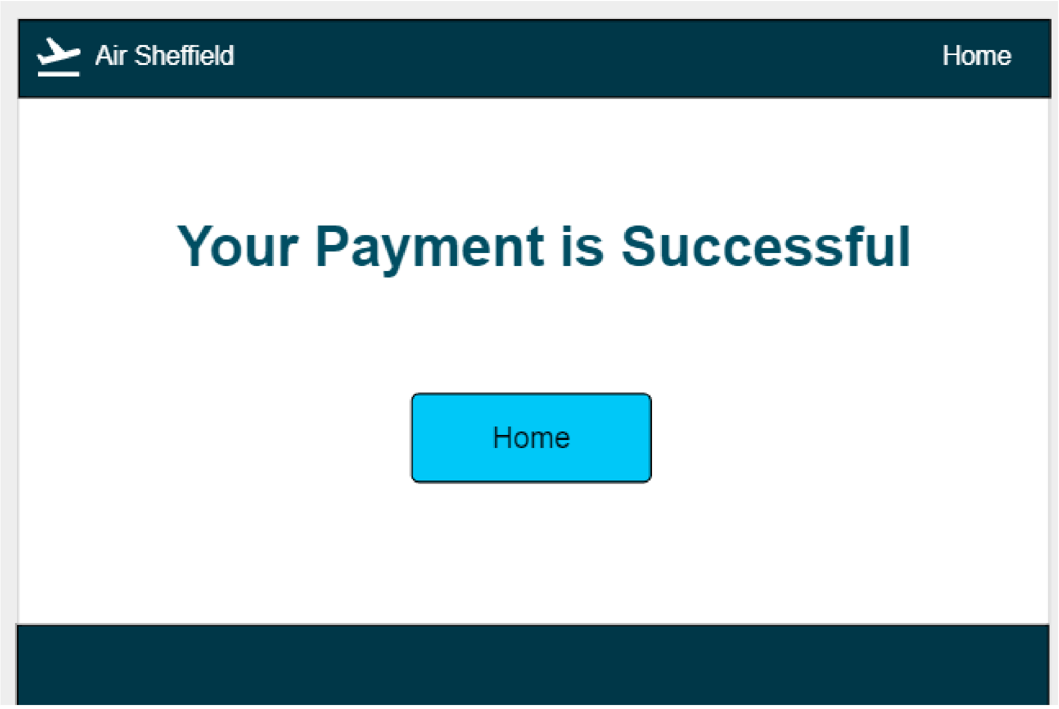
Exp Date: 01/18

CVV: 123

Figure 4.2: Mock-up of payment page

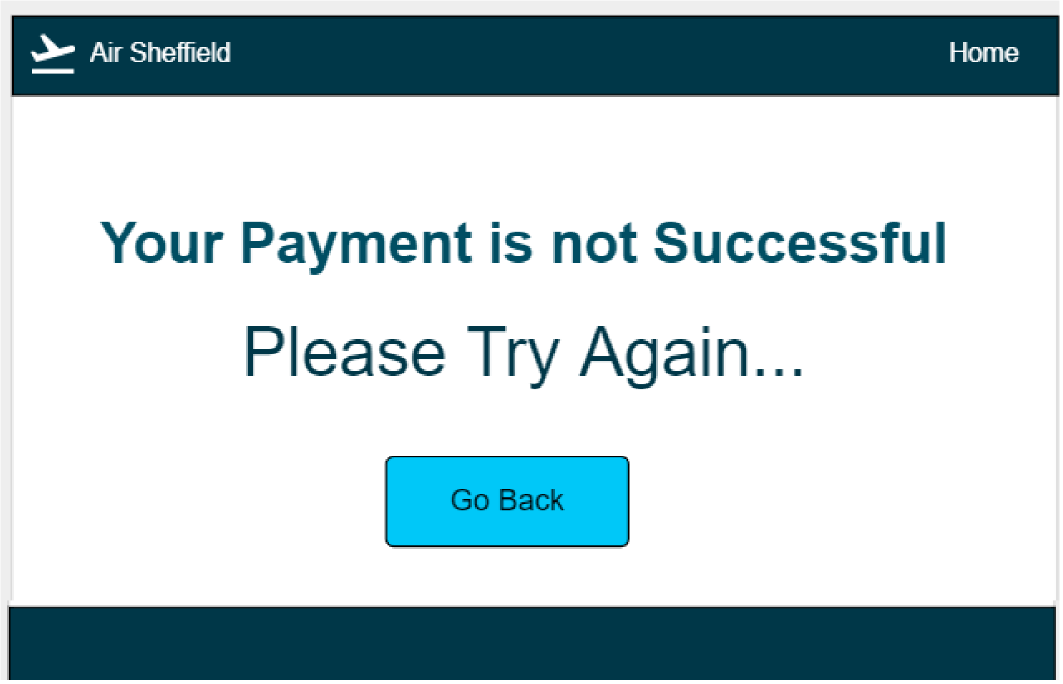
After selecting the payment criteria, the user will see another screen, Fig 4.2, which is the main functionality of the system. It shows that the user has selected to pay with credit or debit card. So, the screen will show the fields related to credit and debit card. The header and footer are the same as of the previous page. A navbar stating the name of the site along with an icon and a link to the Homepage. The main body of the screen has a label which informs the user to enter the credit or debit card details.

The page has a rectangular box which shows the fields. The box is made to differ the information and fields on a page. The input fields are illustrated in this box. There are four labels and four input fields shown in the box. The input fields are for cardholder name, card number, the expiry date of the card and the CVV. The user will enter his credentials in these input fields and press 'pay now' button to proceed.



Your Payment is Successful

Figure 4.3: Mock-up of successful message page



Your Payment is not Successful

Please Try Again...

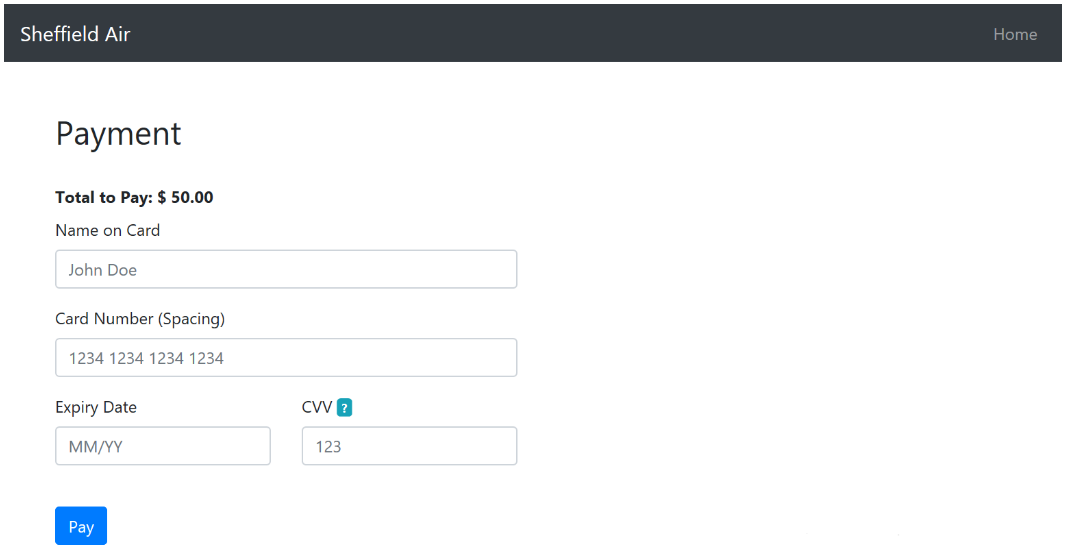
Figure 4.4: Mock-up of unsuccessful message page

Figure 4.3 and 4.4 shows the message page which the user will see after clicking pay now button. If the credentials, he entered are correct, the user will see Payment Successful message and Payment Unsuccessful message otherwise.

4.3 Final User Interface

This section illustrates the final user interface developed for the website. There is a total of three pages designed. The main page, successful message page and an unsuccessful message page. Now I am going to discuss the functionality of each page.

HTML, CSS and JavaScript is used to design the front end of the system. Bootstrap is used to make the website responsive.



Payment

Total to Pay: $ 50.00

Name on Card: John Doe

Card Number (Spacing): 1234 1234 1234 1234

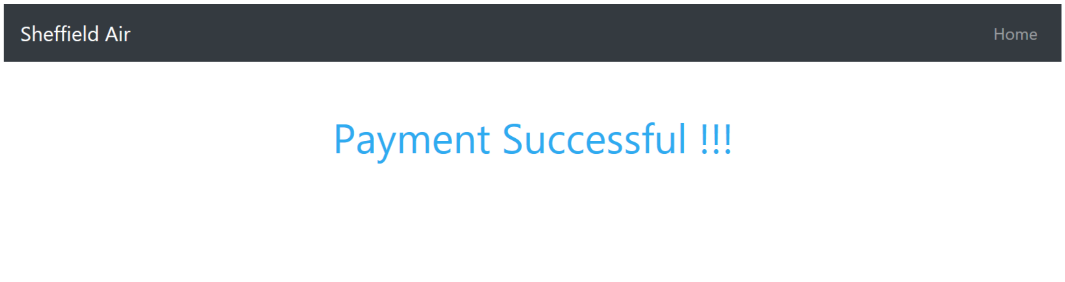
Expiry Date: MM/YY

CVV: 123

Figure 4.5: Main Page

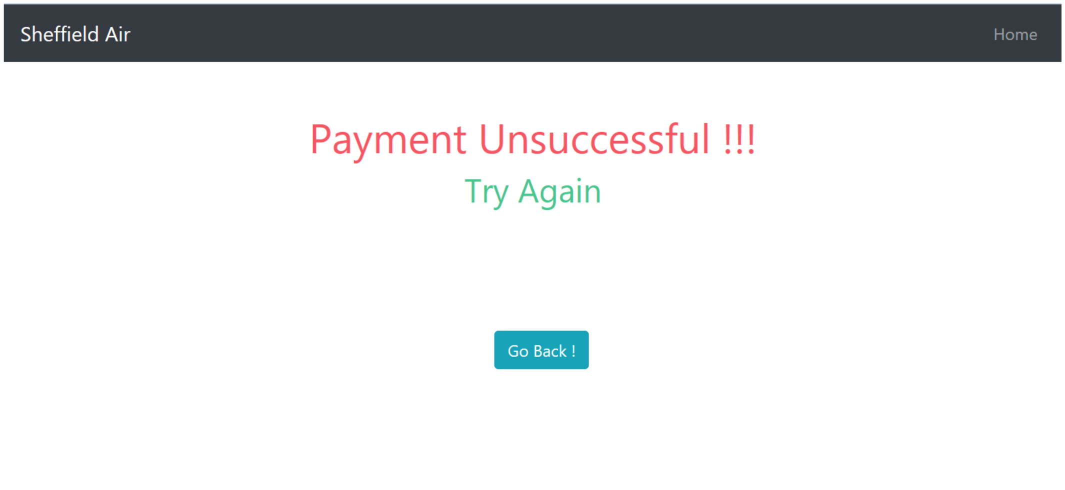
Figure 4.5 shows the main functionality of the page. This is the page where the user will land after selecting a flight to pay for the flight. For this project, this is the main page of the system. It has a navbar with the name of the website i.e. 'Sheffield Air' and a link to the Homepage of the website. Then the main body of the system comes into play. It has four input fields to let the user enter his card details. The labels have been provided along with the fields to make the system interactive. Placeholders were also added to the input fields so that the user enters correct credentials. The page is designed similar to the payment pages that are used by most of the shopping websites.

After entering their details, the users will land on another page. If the credentials are correct, they will be shown a successful message on a page and an unsuccessful message otherwise. This page also contains a button to go back and try again in case the payment is not successful.



Payment Successful !!!

Figure 4.6: successful message page



Payment Unsuccessful !!!
Try Again

Figure 4.7: Unsuccessful message page

Figure 4.6 and 4.7 shows the successful and unsuccessful message page which the user will see after entering the card details.

I have tried to make these pages keeping in mind the principles of HCI. For example, red color is used to make the unsuccessful message and blue color is used to show the successful message.

The input fields are validated and all the checks have been implemented so that when the users tries to enter incorrect information or empty information, it wont let them do it.

4.4 Design Workflow

This section describes the overall workflow of the system with the help of a sequence diagram also known as message sequence chart.

The diagram below shows a sequence diagram of the online payment system and a part of the airline reservation system to show the working of the system.

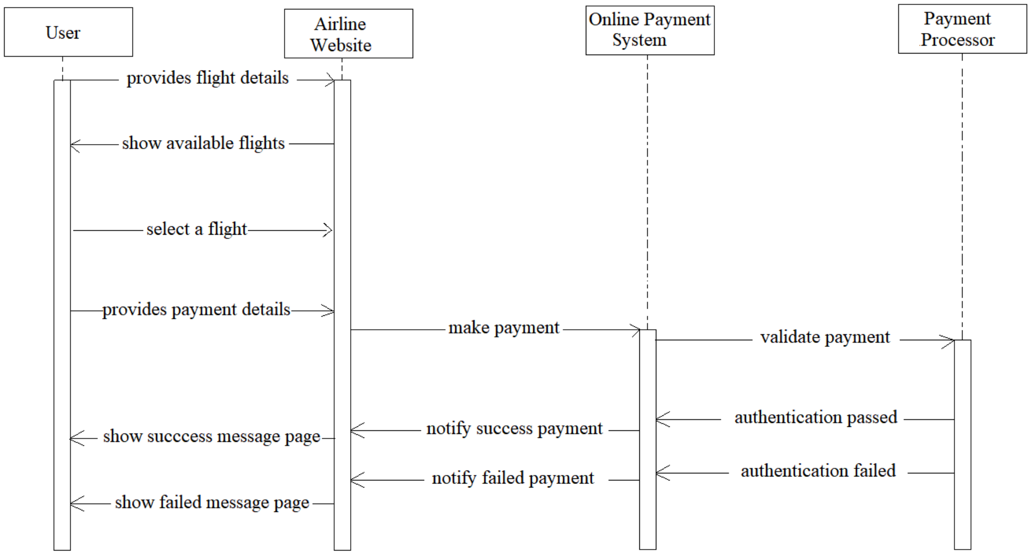


Figure 4.8: Sequence Diagram of Payment

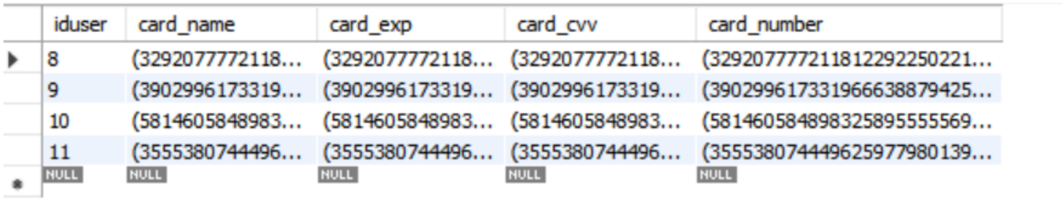
When the application runs for the first time, all the entities are inactive. The user entity will become active when he provides the detail of the flight to the airline website. These details include source and destination cities, departure and return date (in case the user has select a return flight). After providing the details, the airline website will show the user list of available flights according to the user input. After that, the user will select a flight and will enter his contact details.

The next step is the website will be redirected to the payment page where the user will enter his credit or debit card details. These details include the name on card, card number, its expiry date and security code or CVV. After entering the card details, airline website will make payment through an online payment system which is the main functionality of my project. The card details will be validated using the RSA algorithm and will send success and unsuccess message depending on the card information provided.

In this project, it is assumed that the user will enter the correct details for the first time to store them in a database. It is because only an actual bank can authorize and authenticate the card details. So, when the user enters the details for the first time, the details will be encrypted and will be stored in a database. When the user enters the details some other time to make the payment again, the details will be checked again against the details that are stored in a database after decryption.

The database design of the system is very simple. There is only one table of user where his card details are stored. The iduser column is of integer type and is the primary key of the table whereas all other columns are of type varchar.

The database table can be seen in figure 4.8.



	iduser	card_name	card_exp	card_cvv	card_number
▶	8	(3292077772118...	(3292077772118...	(3292077772118...	(329207777211812292250221...
	9	(3902996173319...	(3902996173319...	(3902996173319...	(390299617331966638879425...
	10	(5814605848983...	(5814605848983...	(5814605848983...	(581460584898325895555569...
	11	(3555380744496...	(3555380744496...	(3555380744496...	(355538074449625977980139...
*	NULL	NULL	NULL	NULL	NULL

Figure 4.9: Database table

Chapter 5

Implementation

This chapter describes the detail implementation of the system. The structure of the system was explained along with the code snippets to outline the working of the system. It also describes the technologies, languages and the frameworks used in developing the system.

5.1 Technologies and Framework

IDE

Eclipse IDE (Integrated Development Environment) for Java EE (Enterprise Edition) is used to develop the whole system. To develop a JavaEE application, Eclipse is considered one of the best IDE because of its easy-to-use features. It provides an interactive interface to the user so that the user can easily develop an application. It also provides errors and warnings which are shown in separate tabs to differentiate them.

Server

For this project, I am using Apache Tomcat as a server. Apache is used as a whole web server whereas Apache tomcat is used as a server for java web application. I am using this server because it is one of the most famous server used for the web application that are build using java. It is because tomcat is open source and flexible server. It contains a server.xml. We can change that file according to our needs. For example, to make the server secure, we can introduce https by adding few lines of codes in server.xml.

Database

MySQL is used as a database for this application. MySQL is one of the most famous database that is used for web applications. It provides data security as well as high performance while processing queries.

In this project, the name of schema is airline and the table is called 'users1'. There are five columns in the database which I will describe later in this section.

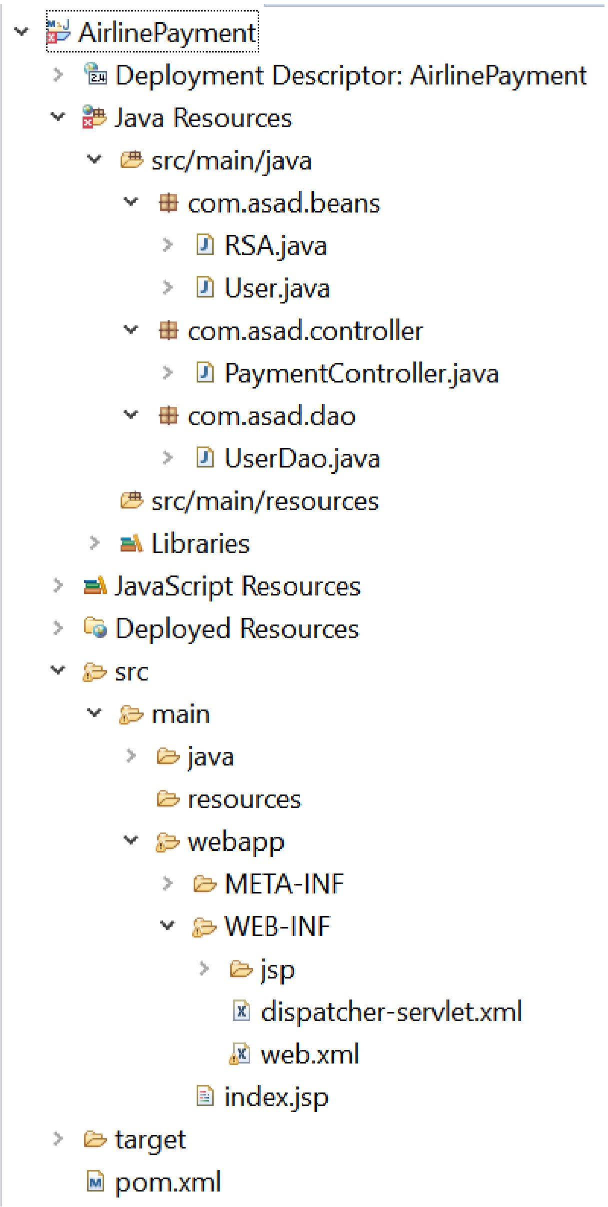
Framework

The framework is very important for any website. For this system, I am using spring framework, more specifically spring MVC. Spring framework is an application framework used for the java platform and this framework is opensource [1]. It is a widely used framework for java web applications.

The application is following a Model View Controller (MVC) structure. In this type of web application structure, business logic resides in the model. The interface of the application which is JSP pages resides in the view part of MVC and the controller is used as a middle man between view and model. However, In Spring MVC, it is slightly different. There is a front controller which is a dispatcher servlet that receives every request and passes onto the controller which creates a model and return that to front controller which further passes that to the view. In the end, view return the response as a JSP page to the front controller.

5.2 Directory Structure

This system uses the standard directory structure of the spring java. The directory structure of the system can be seen in the image below.



```
AirlinePayment
  Deployment Descriptor: AirlinePayment
  Java Resources
    src/main/java
      com.asad.beans
        RSA.java
        User.java
      com.asad.controller
        PaymentController.java
      com.asad.dao
        UserDao.java
    src/main/resources
    Libraries
  JavaScript Resources
  Deployed Resources
  src
    main
      java
      resources
      webapp
        META-INF
        WEB-INF
          jsp
          dispatcher-servlet.xml
          web.xml
        index.jsp
  target
  pom.xml
```

Figure 5.1: Application directory structure

From figure 5.1, it is clear that all of the main code resides inside the src directory. The src directory contains 3 main folders or directories namely java, resources and webapp. The java folder contains all the java files and packages that are used in the system. It contains three packages for beans, controller and dao or data access object.

Beans package contains the model class of the user and the main business logic of the system which is the RSA algorithm. The controller package contains the payment controller which is a java class that is used to communicate between the model and the view. The dao package contains the DAO class which is responsible for communicating with the database The resource folder can have any images or public files that are used in the project. Finally, the webapp folder contains all the interfaces file or JSPs along with the configuration file of the project.

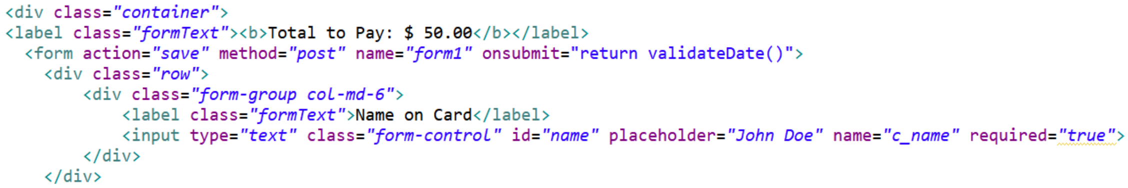
Apart from these main directories, there is another file called pom.xml which contains all the dependencies that are needed for the spring framework to work. This is the standard directory structure that is used by almost all of the web applications.

5.3 Implementations

In this section, I will discuss the detail implementation of the project with the help of code snippets.

Interface

For the interface, I have designed JSP pages which consist of HTML, CSS and JavaScript code. HTML is used to develop the basic layout of the website which includes navbar, the main body i.e. input fields, label and buttons. CSS is used to design the layout and components of HTML and JavaScript is used for event listener. Bootstrap is used to make the page look responsive. The image below shows the div that acts as an input field.



```
<div class="container">
<label class="formText"><b>Total to Pay: $ 50.00</b></label>
  <form action="save" method="post" name="form1" onsubmit="return validateDate()">
    <div class="row">
        <div class="form-group col-md-6">
            <label class="formText">Name on Card</label>
            <input type="text" class="form-control" id="name" placeholder="John Doe" name="c_name" required="true">
        </div>
    </div>
```

Figure 5.2: Input field div

The container is a basic bootstrap element that is used to layout the grid of the page. It gives the page certain padding which defines the style of the component. Input field is designed using a class of form-control. It has a placeholder to let the user know what kind of text should be entered in this field. The attribute required specifies that this field is required and should be entered. The name attribute is used in controller later.

Bean/Model

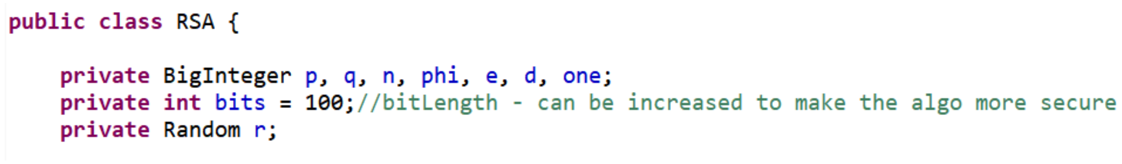
The java folder contains a package called beans which contains the model classes or sometimes called bean which has the code for business logic. In this application, there are two java classes, one for the user and another for the RSA algorithm. JavaBeans are classes which encapsulates a lot of objects into a single object. The User class is a JavaBean class whereas the RSA class is simple java class which contains the business logic.

User Class

The user class contains the attributes of the user that are needed for the payment. This includes card name, card number, card expiry date and CVV. This class of User also contains getters and setters that are needed for the computations.

RSA Algorithm

Now I am going to discuss the code of RSA algorithm.



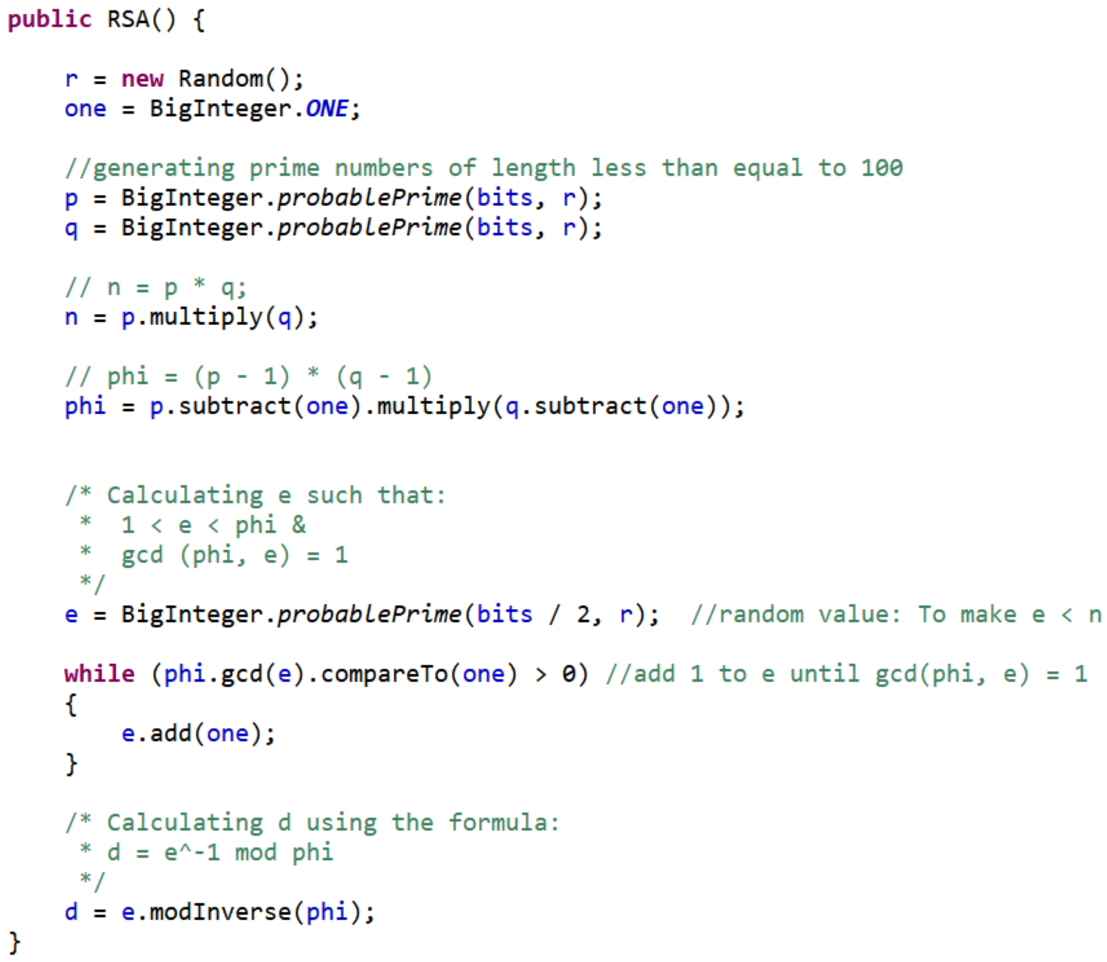
```
public class RSA {

    private BigInteger p, q, n, phi, e, d, one;
    private int bits = 100;//bitLength - can be increased to make the algo more secure
    private Random r;
```

Figure 5.3: RSA attributes

First of all, a public class is created named RSA which contains attributes used for the RSA algorithm. The main attributes are p, q, n, phi, e and d. They all are BigInteger. BigInteger is a datatype used in java for the computation of huge integers. Random is used to have random prime numbers to make the algorithm secure so that every time the code runs for encryption, different prime numbers must be generated. Another attribute is bits which is initialized to 100 to make the calculation easier for the time being. This is actually the bit length of the prime number that will be generated. In real-world cases, the bit length can be 1024 or even 2048 to make the algorithm secure. The more the bit length, the secure the algorithm will be.

After initialization of the attributes, a constructor is created. The computations for calculating private and public key will be done here.

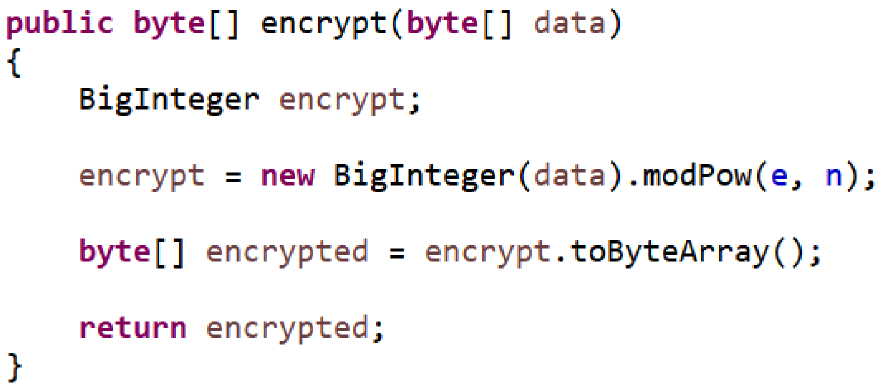
The image below shows the constructor.



```
public RSA() {

    r = new Random();
    one = BigInteger.ONE;

    //generating prime numbers of length less than equal to 100
    p = BigInteger.probablePrime(bits, r);
    q = BigInteger.probablePrime(bits, r);

    // n = p * q;
    n = p.multiply(q);

    // phi = (p - 1) * (q - 1)
    phi = p.subtract(one).multiply(q.subtract(one));


    /* Calculating e such that:
     *  1 < e < phi &
     *  gcd (phi, e) = 1
     */
    e = BigInteger.probablePrime(bits / 2, r);  //random value: To make e < n

    while (phi.gcd(e).compareTo(one) > 0) //add 1 to e until gcd(phi, e) = 1
    {
        e.add(one);
    }

    /* Calculating d using the formula:
     * d = e^-1 mod phi
     */
    d = e.modInverse(phi);
}
```

Figure 5.4: RSA constructor

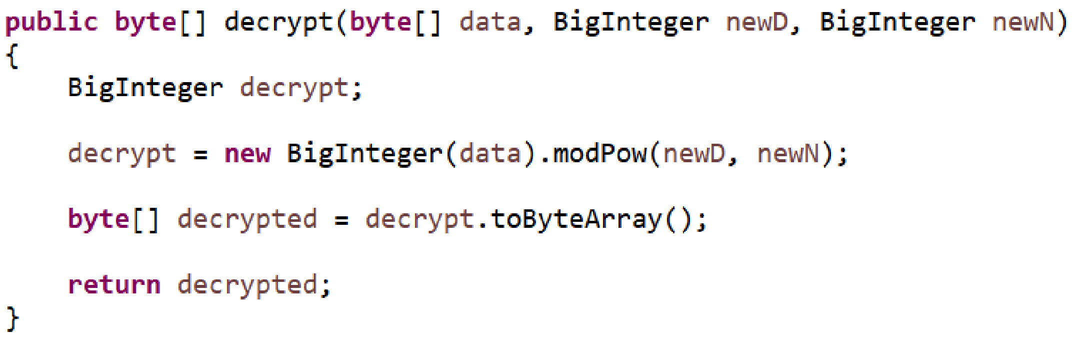
The first step of the constructor is to initialize the random variable. Then two prime numbers are generated of the length less than or equal to 100. BigInteger has a function called probablePrime which will return a prime number of specified bit length. The next step is to multiply these two number. As these are BigIntegers and not simple java integer, therefore, we can not use asterisk operator to multiply these two numbers. Fortunately, BigInteger has a multiplication function which multiplies these two prime numbers. The next step is to calculate 'phi' which can be calculated using the formula shown in the code snippet above. Now we need to calculate 'e' and 'd' which are the main parts of public and private keys respectively. A random prime number will be generated which is half of the length of the bits to make it less than 'n'. While looping over 'e', I am adding one until greatest common divisor or gcd of 'e' and 'phi' is one. To calculate 'd', I am using a formula which can be implemented using the BigInteger function modInverse.



```
public byte[] encrypt(byte[] data)
{
    BigInteger encrypt;

    encrypt = new BigInteger(data).modPow(e, n);

    byte[] encrypted = encrypt.toByteArray();

    return encrypted;
}
```

Figure 5.5: Code for encryption

Figure 5.5 shows the code for encryption. A function is made which accepts a byte [ ]. As we are dealing with the ASCII for the input the user is going to enter, therefore, byte [ ] is used instead of string because the data can be changed if string is used and it takes a lot of time to compute. A BigInteger is created which is used to encrypt the data. Encryption is done using the BigInteger function modPow. It accepts the public key pair and encrypts the data. The encrypted data is then converted back to byte [ ] and will be returned to the main.

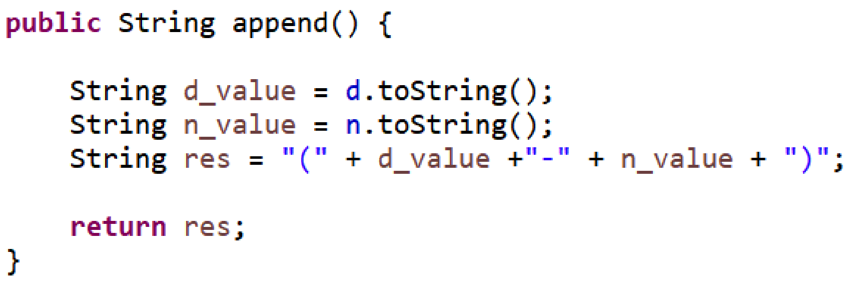


```
public byte[] decrypt(byte[] data, BigInteger newD, BigInteger newN)
{
    BigInteger decrypt;

    decrypt = new BigInteger(data).modPow(newD, newN);

    byte[] decrypted = decrypt.toByteArray();

    return decrypted;
}
```

Figure 5.6: Code for decryption

Figure 5.6 shows the code for decryption. Similar to encryption, a function is made which accepts a byte [ ]. A BigInteger is created which is used to decrypt the data. For decryption, a BigInteger function of modPow is used which accepts the private key pair. The decrypted data is then converted back to byte [ ] and will be returned to the main.

For the time being, the private key pair is saved in the database to make the system simple. The private key pair includes 'd' and 'n'. These values are appended to the encrypted string before storing the string to database.



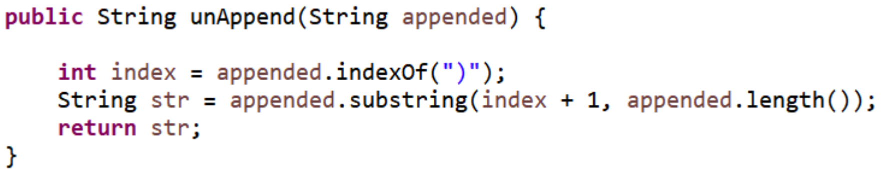
```
public String append() {

    String d_value = d.toString();
    String n_value = n.toString();
    String res = "(" + d_value +"-" + n_value + ")";

    return res;
}
```

Figure 5.7: Value of n and d appended

This code shows the append function in which the value of 'd' and 'n' are converted to a string and then appended to the encrypted string later using this function call. Round brackets and a hyphen is used to append the values at the start of the encrypted string. The start of the string has opening brackets followed by the value of 'd' then hyphen followed by the value of 'n' and closing bracket and then the rest of the encrypted string.

The appended string following the format "(d-n)encryptedstring" will look something like this: "(123-456)abcxyz".

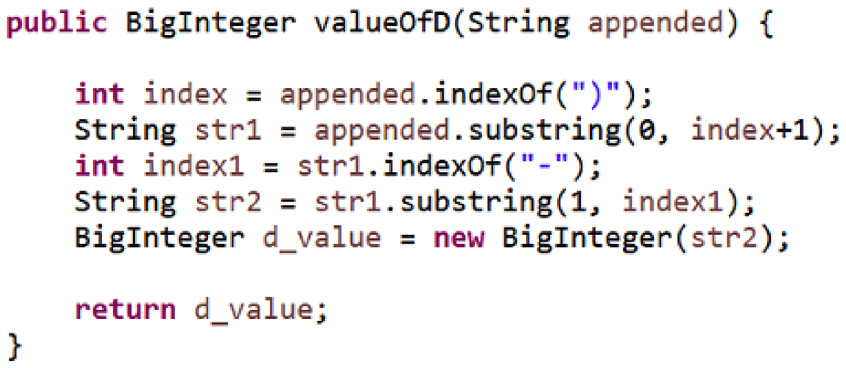
Now the encrypted string has the value of 'd' and 'n'. For decryption, we will need the original encrypted string so we need to unappend the values of 'd' and 'n' from the encrypted string. Below function shows the code to unappend these values from the string. It will return the original encrypted string.



```
public String unAppend(String appended) {

    int index = appended.indexOf(")");
    String str = appended.substring(index + 1, appended.length());
    return str;
}
```

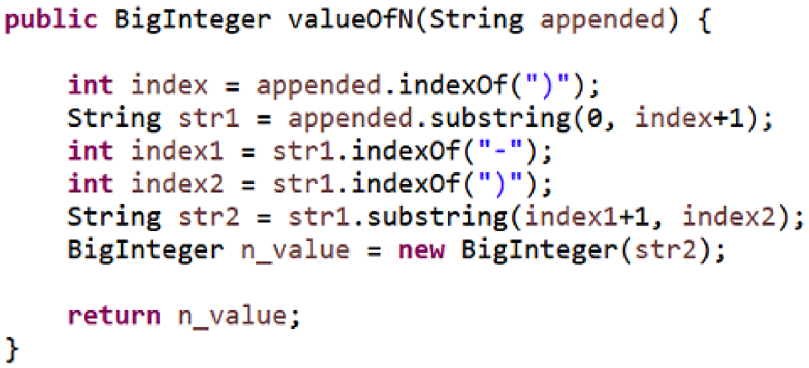
Figure 5.8: Value of n and d un-appended

Figure 5.9 and 5.10 below shows the method to unappend the values of 'd' and 'n' respectively from the encrypted string. Java's method of substring is used to calculate the value for both 'd' and 'n'. Java's indexOf method is used to locate the position of brackets and hyphen.



```
public BigInteger valueOfD(String appended) {

    int index = appended.indexOf(")");
    String str1 = appended.substring(0, index+1);
    int index1 = str1.indexOf("-");
    String str2 = str1.substring(1, index1);
    BigInteger d_value = new BigInteger(str2);

    return d_value;
}
```

Figure 5.9: Calculating value of d



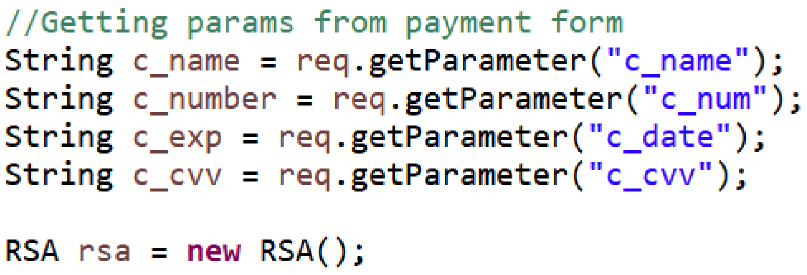
```
public BigInteger valueOfN(String appended) {

    int index = appended.indexOf(")");
    String str1 = appended.substring(0, index+1);
    int index1 = str1.indexOf("-");
    int index2 = str1.indexOf(")");
    String str2 = str1.substring(index1+1, index2);
    BigInteger n_value = new BigInteger(str2);

    return n_value;
}
```

Figure 5.10: Calculating value of n

Controller

This section is used to describe the implementation of the controller which is used to communicate between the view and the model or bean. Two main functionalities are taking place in the controller i.e. encryption and decryption. In real-world scenarios, bank authenticate the user credentials but in this application I am assuming that the user enters his credentials correctly when entering them for the first time. So, at this moment encryption part of the controller will called which will encrypts his credentials and will store them in the database. When the user enters the credentials second time, the decryption part of the controller will be called. The credentials will be decrypted and then check against the value user enters this time. If they are same, successful payment will be shown to the user and unsuccessful otherwise.

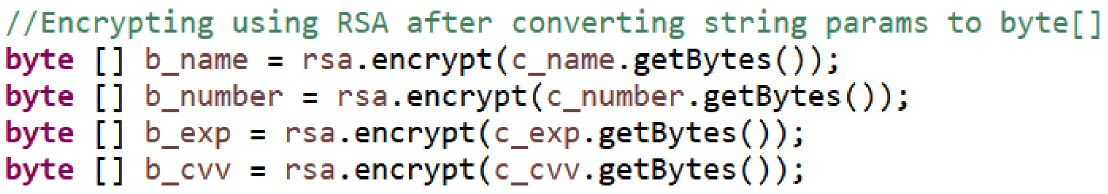
Encryption



```
//Getting params from payment form
String c_name = req.getParameter("c_name");
String c_number = req.getParameter("c_num");
String c_exp = req.getParameter("c_date");
String c_cvv = req.getParameter("c_cvv");

RSA rsa = new RSA();
```

Figure 5.11: Getting parameters from the form

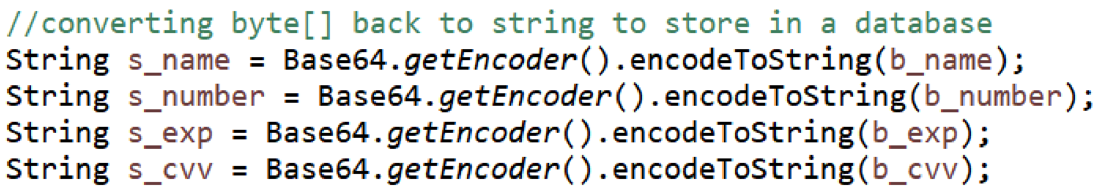
When the user enters the credentials which include the name on card, card number, expiry date and CVV, the parameters will be saved in a string. The code above shows getting parameters from the form using request.getParameter() method. These parameters will be stored in a string.



```
//Encrypting using RSA after converting string params to byte[]
byte [] b_name = rsa.encrypt(c_name.getBytes());
byte [] b_number = rsa.encrypt(c_number.getBytes());
byte [] b_exp = rsa.encrypt(c_exp.getBytes());
byte [] b_cvv = rsa.encrypt(c_cvv.getBytes());
```

Figure 5.12: Encryption code in Controller

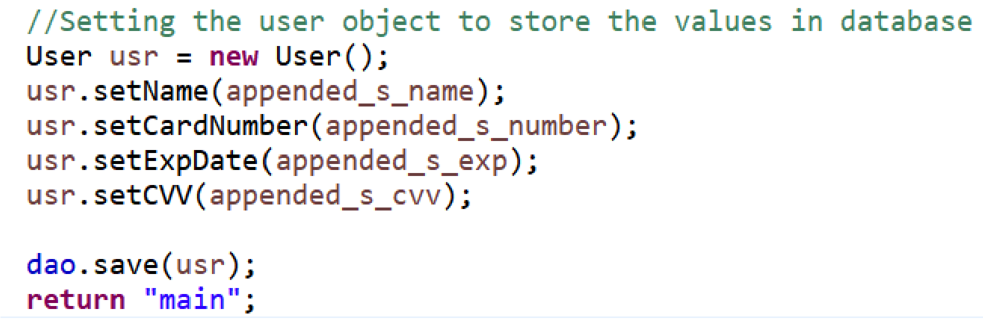
Figure 5.12 shows the simple function call to encrypt function of RSA class. The credentials will be encrypted and stored in a byte [ ].



```
//converting byte[] back to string to store in a database
String s_name = Base64.getEncoder().encodeToString(b_name);
String s_number = Base64.getEncoder().encodeToString(b_number);
String s_exp = Base64.getEncoder().encodeToString(b_exp);
String s_cvv = Base64.getEncoder().encodeToString(b_cvv);
```

Figure 5.13: Conversion from byte [ ] to String

This code above shows the conversion from byte [ ] to string to append the values of 'd' and 'n' and storing them in a database. Base64 encoder function is used to convert byte [ ] to string so that any data does not loss as it is very sensitive information of the user.



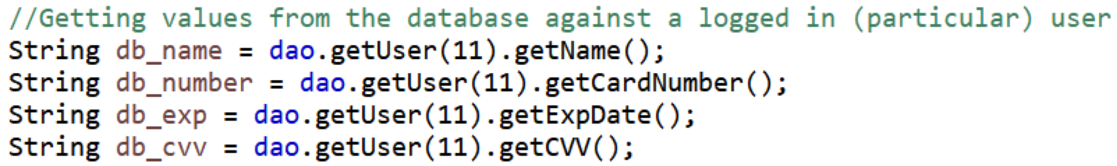
```
//Setting the user object to store the values in database
User usr = new User();
usr.setName(appended_s_name);
usr.setCardNumber(appended_s_number);
usr.setExpDate(appended_s_exp);
usr.setCVV(appended_s_cvv);

dao.save(usr);
return "main";
```

Figure 5.14: Saving the encrypted credentials in a database

After converting the values to string, the value of 'd' and 'n' are appended to the encryption string. For this, a function call to append function of RSA class will happened.

After that, a new User will be created and all the credentials he entered will be added using the setters method of the User class and will be saved in a database by calling save function of UserDao class.

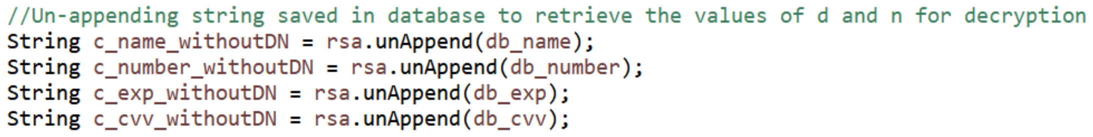
Decryption



```
//Getting values from the database against a logged in (particular) user
String db_name = dao.getUser(11).getName();
String db_number = dao.getUser(11).getCardNumber();
String db_exp = dao.getUser(11).getExpDate();
String db_cvv = dao.getUser(11).getCVV();
```

Figure 5.15: Getting the values of the user from database

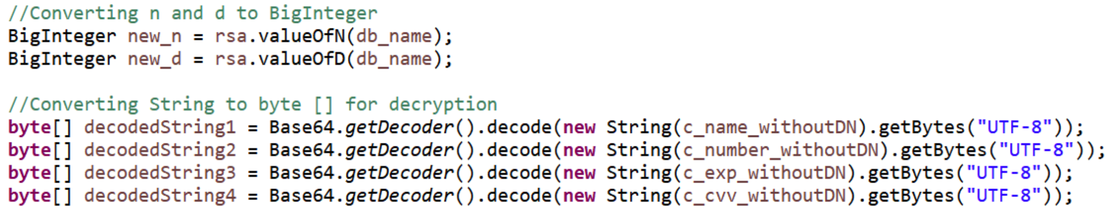
The first step in decryption is to get the value from the database that is stored against the logged-in user. Since login functionality is not part of this application, I am using hard code value for the user ID. Dao object is used to call the getUser function which accepts an integer and returns the row against that ID. The value we get from this step will be used in decryption.



```
//Un-appending string saved in database to retrieve the values of d and n for decryption
String c_name_withoutDN = rsa.unAppend(db_name);
String c_number_withoutDN = rsa.unAppend(db_number);
String c_exp_withoutDN = rsa.unAppend(db_exp);
String c_cvv_withoutDN = rsa.unAppend(db_cvv);
```

Figure 5.16: Unappending the values of d and n

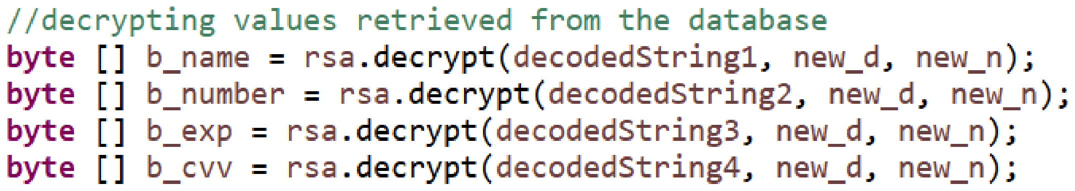
As the value from the database are encrypted and has 'd' and 'n' append, we need to unappend these values. Figure 5.16 shows the code to unappend the values which is nothing but a simple function call to unAppend fucntion of RSA class.



```
//Converting n and d to BigInteger
BigInteger new_n = rsa.valueOfN(db_name);
BigInteger new_d = rsa.valueOfD(db_name);

//Converting String to byte [] for decryption
byte[] decodedString1 = Base64.getDecoder().decode(new String(c_name_withoutDN).getBytes("UTF-8"));
byte[] decodedString2 = Base64.getDecoder().decode(new String(c_number_withoutDN).getBytes("UTF-8"));
byte[] decodedString3 = Base64.getDecoder().decode(new String(c_exp_withoutDN).getBytes("UTF-8"));
byte[] decodedString4 = Base64.getDecoder().decode(new String(c_cvv_withoutDN).getBytes("UTF-8"));
```

Figure 5.17: Conversion of string to byte [ ]

Now we have the values of 'd' and 'n' separated from the encrypted string. In this step, we need to convert the encrypted string to byte [ ] to perform encryption. For this, Base64 is used to again so that the data does not loose. Base64 decode function is used to convert string to byte [ ]. UTF-8 is specified because in java the strings are usually in UTF-8. So, to avoid any error in conversion, UTF-8 is specified and Base64 decoder is used to avoid loss of data or any alteration in a string that could occur.

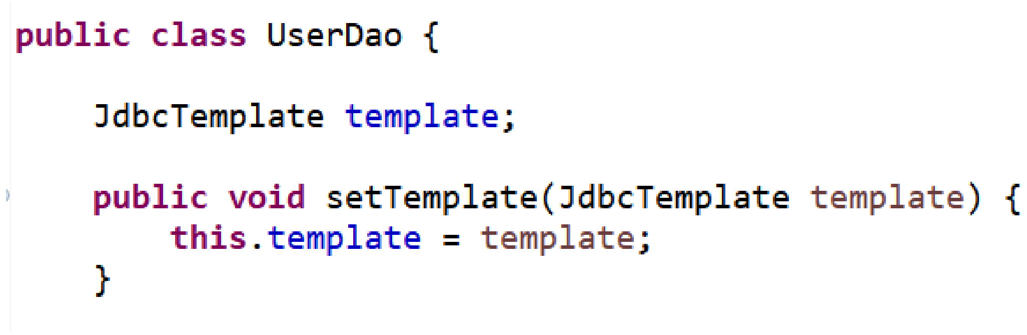


```
//decrypting values retrieved from the database
byte [] b_name = rsa.decrypt(decodedString1, new_d, new_n);
byte [] b_number = rsa.decrypt(decodedString2, new_d, new_n);
byte [] b_exp = rsa.decrypt(decodedString3, new_d, new_n);
byte [] b_cvv = rsa.decrypt(decodedString4, new_d, new_n);
```

Figure 5.18: Decryption code in Controller

Figure 5.18 shows the simple function call to decrypt function of RSA class. The value from the database will be retrieved, decrypted and stored in a byte [ ]. The final step of the application is to check these credentials. If the decrypted credentials and the one entered by the user are same, the payment will be successful, otherwise, an error message will be shown to the user.

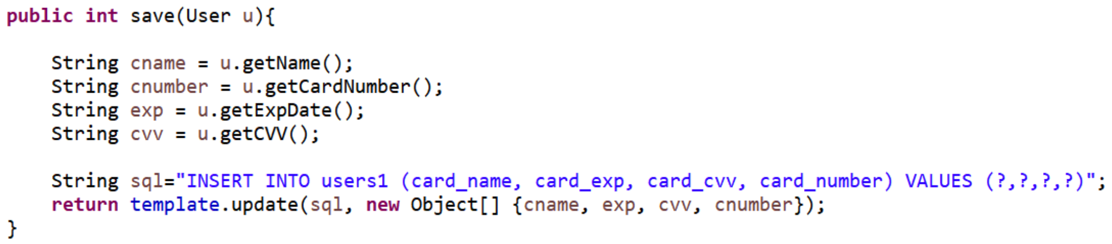
DAO (Data Access Object)



```
public class UserDao {

    JdbcTemplate template;

    public void setTemplate(JdbcTemplate template) {
        this.template = template;
    }
```

Figure 5.19: UserDao class

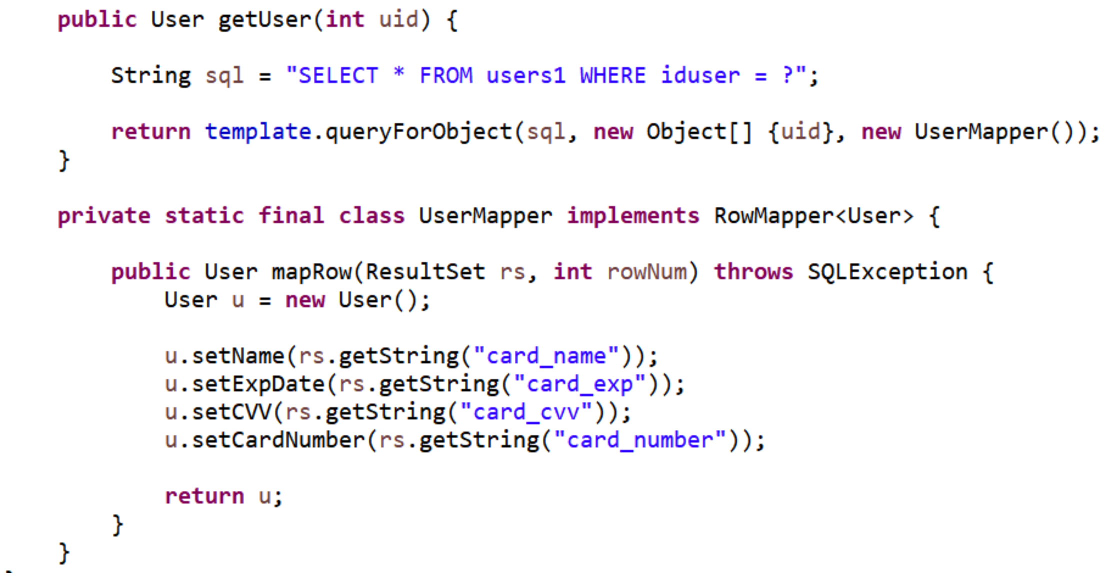
Figure 5.19 shows the UserDao class. JDBC template is used to interact with the database. JDBC template is used to execute the SQL queries. This is specifically used for spring. With the help of the JDBC template, we not need any prepared statements or result sets because it takes care of all these functionalities by itself. The code shows the constructor of the UserDao class where the JDBC template is just setting up.



```
public int save(User u){

    String cname = u.getName();
    String cnumber = u.getCardNumber();
    String exp = u.getExpDate();
    String cvv = u.getCVV();

    String sql="INSERT INTO users1 (card_name, card_exp, card_cvv, card_number) VALUES (?,?,?,?)";
    return template.update(sql, new Object[] {cname, exp, cvv, cnumber});
}
```

Figure 5.20: Saving the credentials of the user in database

The code above shows the function of the UserDao class where the credentials of the user are storing in the database using the INSERT query of SQL. At first, getters are used to get the attributes of the user from the User bean class. Then an SQL INSERT query is fired. '?' is used to stay safe against SQL injection. There can be hackers which can attack the system when communicating from browser to server or another way round. They can get access to the system by using the famous '1 = 1' statement. So, instead of a simple query, it will be a statement which will be safe to any SQL injections. In simple MVC, the prepared statement is used against SQL injections but in this case, the JDBC template internally handles prepared statement.

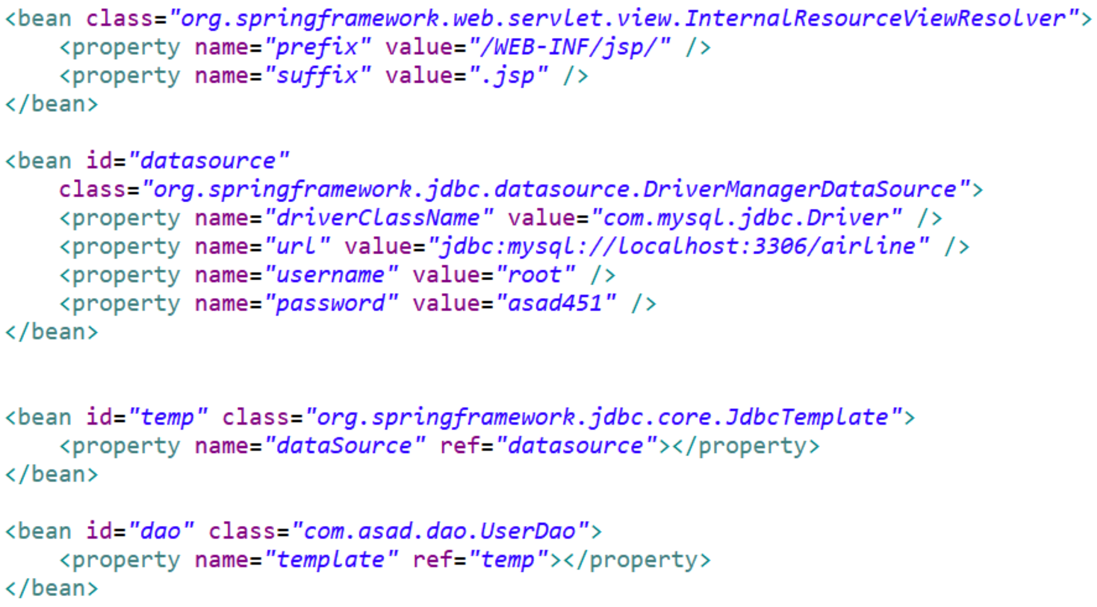


```
public User getUser(int uid) {

    String sql = "SELECT * FROM users1 WHERE iduser = ?";

    return template.queryForObject(sql, new Object[] {uid}, new UserMapper());
}

private static final class UserMapper implements RowMapper<User> {

    public User mapRow(ResultSet rs, int rowNum) throws SQLException {
        User u = new User();

        u.setName(rs.getString("card_name"));
        u.setExpDate(rs.getString("card_exp"));
        u.setCVV(rs.getString("card_cvv"));
        u.setCardNumber(rs.getString("card_number"));

        return u;
    }
}
```

Figure 5.21: Getting the credentials of the user from database

For decryption, we need the saved encrypted credentials. Figure 5.21 shows the code for getting the user credentials from the database. A SELECT query is used to get the values. An inner class of RowMapper is created so that the values we get from the database can be mapped to the User bean. RowMapper class has a function mapRow which accepts a result set and used to set the user.

Apart from the code snippets of java code, a lot of configurations are needed for the application to run properly. This include configuration of files such as web.xml, pom.xml and dispatcher-servlet.xml. Dispatcher Servlet acts as a front controller in spring MVC. It contains the code for connection to database, dao and views. The figure below shows the code of dispatcher-servlet.xml.



```
<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <property name="prefix" value="/WEB-INF/jsp/" />
    <property name="suffix" value=".jsp" />
</bean>

<bean id="datasource"
    class="org.springframework.jdbc.datasource.DriverManagerDataSource">
    <property name="driverClassName" value="com.mysql.jdbc.Driver" />
    <property name="url" value="jdbc:mysql://localhost:3306/airline" />
    <property name="username" value="root" />
    <property name="password" value="asad451" />
</bean>


<bean id="temp" class="org.springframework.jdbc.core.JdbcTemplate">
    <property name="dataSource" ref="datasource"></property>
</bean>

<bean id="dao" class="com.asad.dao.UserDao">
    <property name="template" ref="temp"></property>
</bean>
```

Figure 5.22: Code of dispatcher-servlet.xml

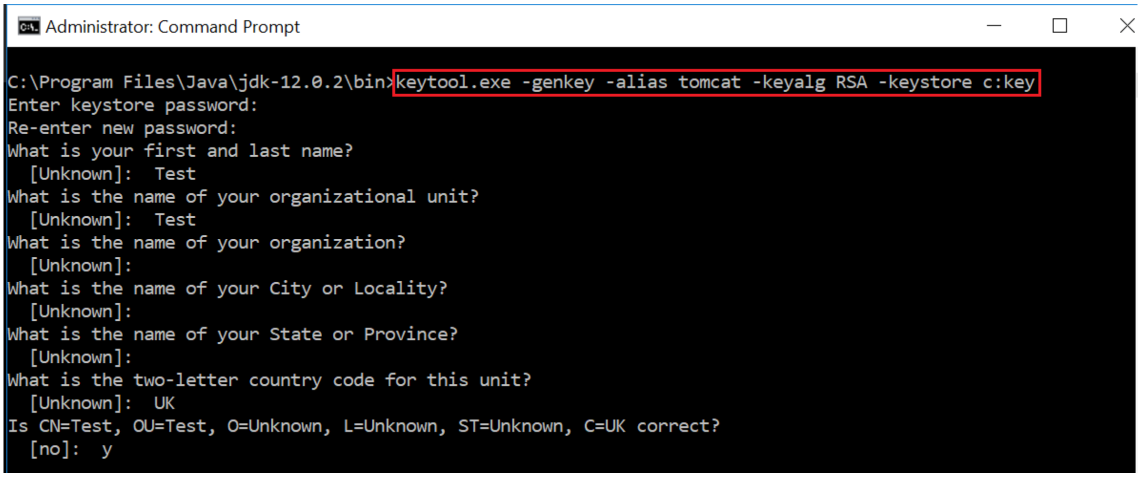
SSL Implementation

Now I will talk about the implementation of SSL that is used to make the connection secure.

Secure Socket Layer or SSL is used to establish a secure connection between the web browser and the server. After implementing the SSL, we will get an HTTPS protocol instead of HTTP in the URL which shows that the connection is secure.

To implement SSL, certificates need to be generated. In real-world, this certificate is generated and signed by some authentic authorization which usually charges a fee. As this project is on the local environment, a certificate is generated through the local computer.

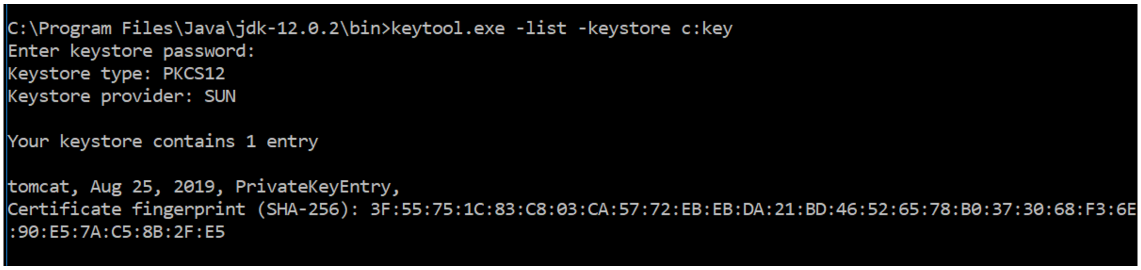
To generate a certificate, key tool is need to be installed. Keytool is used to generate and manage certificates. Figure 5.23 shows a statement on the command line to generate a file which is used to implement SSL.



```
C:\Program Files\Java\jdk-12.0.2\bin>keytool.exe -genkey -alias tomcat -keyalg RSA -keystore c:key
Enter keystore password:
Re-enter new password:
What is your first and last name?
  [Unknown]:  Test
What is the name of your organizational unit?
  [Unknown]:  Test
What is the name of your organization?
  [Unknown]:
What is the name of your City or Locality?
  [Unknown]:
What is the name of your State or Province?
  [Unknown]:
What is the two-letter country code for this unit?
  [Unknown]:  UK
Is CN=Test, OU=Test, O=Unknown, L=Unknown, ST=Unknown, C=UK correct?
  [no]:  y
```

Figure 5.23: Generating key for SSL

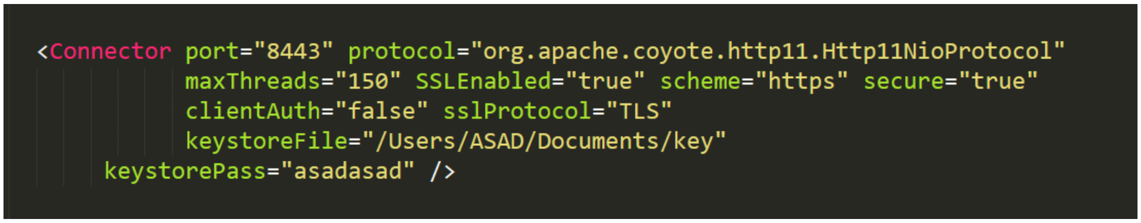
After entering the command, a 2048 bit key is generated which has a validity of 90 days because it is a self-signed certificate that is not generated by an authorized company. The image below shows the certificate details which is usually called a certificate fingerprint.



```
C:\Program Files\Java\jdk-12.0.2\bin>keytool.exe -list -keystore c:key
Enter keystore password:
Keystore type: PKCS12
Keystore provider: SUN

Your keystore contains 1 entry

tomcat, Aug 25, 2019, PrivateKeyEntry,
Certificate fingerprint (SHA-256): 3F:55:75:1C:83:C8:03:CA:57:72:EB:EB:DA:21:BD:46:52:65:78:B0:37:30:68:F3:6E
:90:E5:7A:C5:8B:2F:E5
```

Figure 5.24: SSL key details

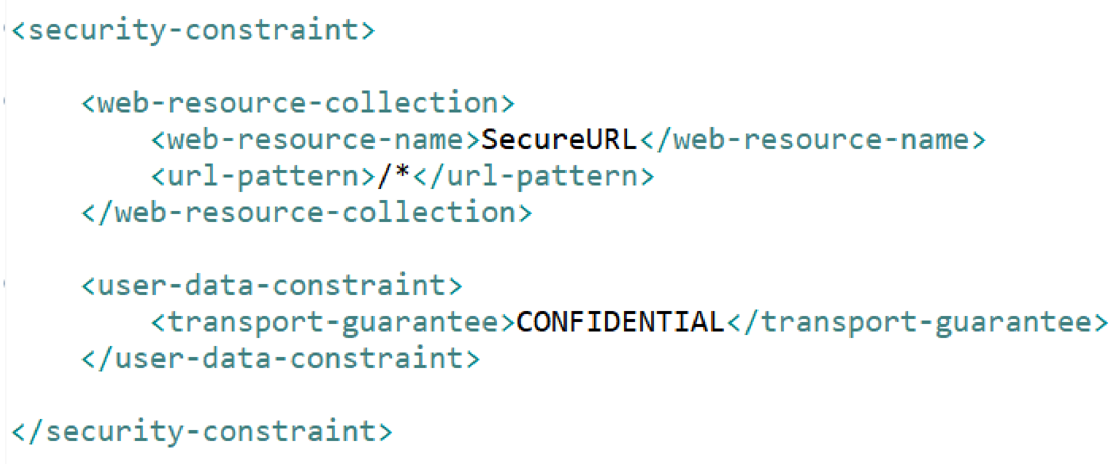
To make the SSL certificate key pair work, we need to configure the server.xml file of the tomcat server. 8443 port is used by default for HTTPS. Therefore, a connector needs to be implemented for this port which has certain attributes such as port, protocol, SSL enabled flag, location of key etc. The code of the server.xml can be seen in figure 5.25 below.



```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
           maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
           clientAuth="false" sslProtocol="TLS"
           keystoreFile="/Users/ASAD/Documents/key"
     keystorePass="asadasad" />
```

Figure 5.25: server.xml code for HTTPS

After generating a certificate, some code needs to be added in the web.xml file of the project. Figure 5.26 shows the code for SSL implementation.



```
<security-constraint>

    <web-resource-collection>
        <web-resource-name>SecureURL</web-resource-name>
        <url-pattern>/*</url-pattern>
    </web-resource-collection>

    <user-data-constraint>
        <transport-guarantee>CONFIDENTIAL</transport-guarantee>
    </user-data-constraint>

</security-constraint>
```

Figure 5.26: web.xml code for HTTPS

From figure 5.26, it is clear that a tag of security constraint is needed which has further sub-tag of web resource collection that is used to describe the web resource name which can be anything and a URL pattern. If https is needed throughout the website, we use / and *. If we need the https for certain page, we need to mention that page name in the URL pattern. Then there is a user data constraint which has a sub tag of transport guarantee which is set to CONFIDENTIAL to make the website secure and confidential from all the thrid party hacks.

Chapter 6

Results and Discussion

This section includes the testing performed on the system to point out any errors or defects in the system. It also explains the difficulties faced during the designing and implementation phase of the project in the discussion section.

6.1 Testing

Testing is very important for the systems success. Sometimes due to human error, some process or functionality can be ignored unintentionally by the developer. So, it is important for both the developer and the tester to perform different types of testing on the system. In this section, I will discuss the testing performs on the system.

Unit Testing

Unit testing is used to test this system. In this type of testing, a single unit of the system is tested to check whether the unit performed as designed. As the system of processing payment is a single unit, therefore, unit testing is performed.

As we are assuming the credentials user will enter for the first time will be authenticated because we are not dealing with the bank, therefore, unit testing is performed for the second time when the user enters the credentials again. At this moment, the credentials will be authenticated after performing decryption.

Below are the test cases that are designed to check the validity of the system along with the results.

Test Case 1: Submitting empty field(s).

Test Result: The system will not accept empty fields and advice the user to enter the credentials.

Passed

Test Case 2: Entering alphabets in card number fields.

Test Result: The system will not accept alphabets.

Passed

Test Case 3: Entering alphabets in CVV fields.

Test Result: The system will not accept alphabets.

Passed

Test Case 4: Entering more than 3 digits in CVV fields.

Test Result: The system will not accept more than 3 digits in CVV fields.

Passed

Test Case 5: Entering more than 16 digits in card number fields excluding spaces.

Test Result: The system will not accept more than 16 digits in card number field.

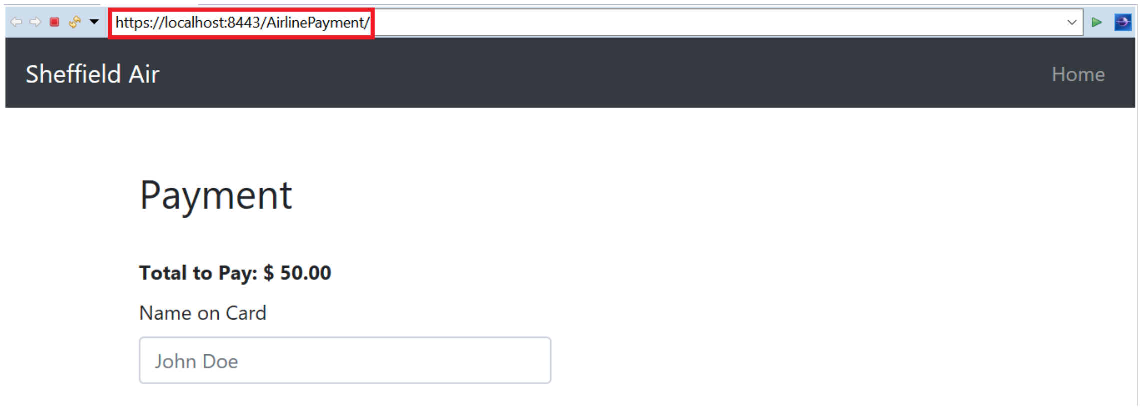
Passed

This shows that all of the test cases made for the functionality has been passed.

Security Testing

Security Testing is important for the website, especially which involves credit card payments, selling and buying so that no third party can intervene in the payment process and hack the user's card details. In Security Testing, we check if the person is authorised to use the system. This can be done by assigning roles as discussed in chapter 2 of this report. Sessions must be checked and should be killed after the user is inactive for a long period of time. If SSL is implemented, the web pages should be redirected to the HTTPS pages instead of HTTP pages.

In this project, I have tested the website for SSL. When I run the code which is based on HTTP, it will be redirected to HTTPS version which shows that SSL is implemented as it is shown in figure 6.1 below. The red boundary shows the URL of the site which is HTTPS. So, the security test for SSL has been passed.



https://localhost:8443/AirlinePayment/

Sheffield Air

Home

Payment

Total to Pay: $ 50.00

Name on Card

John Doe

Figure 6.1: HTTPS in the URL

6.2 Discussion

All of the requirements of the project has been completed whether they are functional or non-functional. All of the requirements have been carefully designed with quality implementation and testing. This indicates that the aim of the project has been achieved which is to provide the user with an interactive web application for secure payment of airline flights. In addition to that, SSL functionality is added to the system to make the website secure. SSL is used to establish a secure connection between the web browser and the server.

To develop the project, I have to learn new technologies from scratch. This includes the spring framework, SSL, some new datatypes of Java. Spring framework is used to develop a web application. It is lightweight and open source java platform and is used by almost all modern websites that use Java. The concept of dependency injection (or DI) of spring framework was difficult to understand although it is based on Inversion of Control or IoC but it was worth it as it helps a lot in the development. The setting of the project took more time than expected because of this new technology. The connection of the project to the database took a lot of time because spring uses JDBC template which I had to learn to implement it. Setting up the secure server was another issue because of some dependency and libraries issues with the server, apache tomcat.

Instead of using the RSA algorithm from the internet, I had decided to implement the algorithm from scratch for understanding. The initial approach towards developing the RSA algorithm was quite simple. I had used simple datatypes of Java to understand how RSA algorithm work. Instead of using BigInteger, I had used integer type to calculate the public and private key pair for encryption and decryption and only single-digit or alphabet is used for this purpose. The algorithm worked fine for small bit length and single input but as I increased the number of bits and the length of the message to be encrypted, the algorithm becomes really slow. All of the functions that are used to calculate the private and public keys was developed manually after studying the algorithm. Then I decided to use BigInteger because this datatype was designed for huge numbers. For encryption, I need to convert the string to ASCII character and vice versa. For this, I have developed a function but as I progressed I realized that I do not require it, which wasted my time. Another problem arises during the encryption and decryption of the data is that I had to convert string to byte and vice versa. At this moment, I came across that the Java function of toString() and toByteArray() does not work properly for conversion. It works well sometimes but most of the time throws an error. So, it takes time to find the solution but in the end, I had found the solution which was to use Base64 encoding.

For the dissertation, I have to put a substantial amount of time in research of the project. As this project was part of the group project, the research carried out previously was for the group project as a whole which includes different modules. But in this project, I had to do research for the payment systems, algorithms and security of the websites from the scratch which consumed most of the time.

Although it was very hard to achieve the projects requirement on time because of the new technologies and limited time, yet I managed to complete all of it with the constant guidance of my supervisor.

6.3 Future Work

There are many ideas and functionalities which can be added to the project to expand it. Some of these are described below.

- Login functionality can be added to the system so that when the user logs in, it will be known if he is entering the card details for the first time or not.
- With the help of login functionality, the user session can be maintained so that the user could not hold the booking for more than a specific time. It can be done using cookies.
- Different payment gateways can be added to the system.
  - Stripe is a payment gateway which can be added to the system. It handles all the credit and debit card payment on its own. It provides an API to the developer which can be added to any website [14].
  - Similar to Stripe, PayPal can also be added to the system. It will help the user if the user does not have a physical card at some time. PayPal provides

a sandbox account which can be used in the development environment [12].

- To make the application secure, a digital signature can also be added to the system. The digital signature can be implemented using hashing. It used to transfer the message in a secure way similar to the RSA algorithm [17].
- An e-mail functionality can also be added to the system. When the user pays for a flight, an automatic e-mail will be generated and sent to the user as a confirmation.
- A mobile app of the system can be developed in future to provide interactivity to the user as they are faster. Push notifications can also be provided to the user through the mobile app.
- As the security of the system is very important, in future more algorithm can be implemented and tested to check which algorithm performs well in less time and which algorithm is more secure.

Chapter 7

Conclusion

As air travelling is growing rapidly, an online system is required which helps the users in booking flights and paying for them. The main aim of the project is to implement an online payment system where the users can pay for their booked flights. To prevent the system from online attacks, different techniques were used such as enabling SSL to make the connection secure between the browser and the server. This way no third party can intercede in the process. Another technique was used to avert the system from the SQL injections. After implementing all the methods and procedures, a web-based application was built which helps the users in paying for their flights securely. The interface of the system is easy to use and interactive so that the user does not feel any difficulty while operating it.

This report outlines the background research carried out on different airline reservation systems that are currently operational. Literature survey includes a detailed analysis of the payment packages and methods used by these systems. To understand the security in an online world, various articles were read and analyzed which described the algorithms and methods used to make the online payment secure. Apart from that, all functional and non-functional requirements of the project have been discussed in detail. The design of the system is developed very carefully to make the user feel confident when using the payment system. All principles of Human-Computer Interaction were kept in mind while designing the system. The detailed implementation is discussed with the help of code snippets which makes it easier to understand the system.

After implementing the system, unit testing was conducted to verify the working of the application. As a result, all tests were passed successfully. The future work has also been discussed which outlines the functionalities that can be added later to extend the project. This includes implementing several payment gateways, designing and implementing different cryptography algorithm, comparing the results of these algorithms and choosing the best one.

Bibliography

[1] Air canada - the official website. https://www.aircanada.com/uk/en/aco/home.html.

[2] Air transport statistics - statistics explained. https://ec.europa.eu/eurostat/statistics-explained/index.php/Air_transport_statistics.

[3] Caesar cipher. https://www.101computing.net/caesar-cipher/.

[4] Company profile - klm corporate. https://www.klm.com/corporate/en/about-klm/profile/index.html.

[5] Cross-site scripting (xss) cheat sheet. https://www.veracode.com/security/xss.

[6] How the cross-site scripting (xss) attack works. https://mountainhomemarketing.com/cross-site-scripting-xss-attack-how-it-works/.

[7] Identity fraud hits all time high with 16.7 million u.s. victims in 2017, according to new javelin strategy & research study. https://www.javelinstrategy.com/press-release/identity-fraud-hits-all-time-high-167-million-us-victims-2017-according-new-javel

[8] Klm - fare categories and conditions. https://www.klm.com/travel/sd_en/plan_and_book/ticket_information/ticket_conditions/fare_categories_and_conditions.htm#p4.

[9] Multi-factor authentication. https://en.wikipedia.org/wiki/Multi-factor_authentication.

[10] Pay for flights with paypal & paypal credit. https://www.alternativeairlines.com/paypal.

[11] Payment methods. https://www.klm.com/travel/gb_en/plan_and_book/booking/payment/index.htm.

[12] Paypal business account - payment methods. 12.https://www.paypal.com/uk/webapps/mpp/merchant.

[13] Skyscanner - find the cheapest flights fast: save time, save money! https://www.skyscanner.net/.

[14] Stripe - online payment processing for internet business. https://stripe.com/gb.

[15] Westjet - payment options - travel info. https://www.westjet.com/en-gb/travel-info/payment/index.

[16] Westjet uatp program. https://www.westjet.com/en-gb/travel-info/payment/uatp.

[17] What is a digital signature? https://www.instantssl.com/digital-signature.

[18] What is cryptography? https://searchsecurity.techtarget.com/definition/cryptography.

[19] What is sql injection (sqli) and how to prevent it. https://www.acunetix.com/websitesecurity/sql-injection/.

[20] Worldpay - home. https://www.worldpay.com/global/business-types/airlines.

[21] Aboud, Sattar, A. M. A. M., and Jabbar, H. An efficient rsa public key encryption scheme.

[22] Al-Furiah, S., and Al-Braheem, L. Comprehensive study on methods of fraud prevention in credit card e-payment system.

[23] Boneh, D., and Shacham, H. Fast variants of rsa.

[24] Bryan Basham, K. S., and Bates, B. *Head First Servlets and JSP 2nd Edition*. O'Riely, 2008.

[25] Fedorenka, S. New eu regulation will require a two-factor payment authentication to protect retailers: Mastercard. https://internetretailing.net/themes/themes/new-eu-regulation-will-require-a-two-factor-payment-authentication-to-protect-ret

[26] Javan, M. S., and Shajari, M. Flash payment: Payment using flash disks.

[27] Laporte, J. Topic: Westjet. https://www.statista.com/topics/5154/westjet/.

[28] Nexcess. The pros and cons of implementing ssl or https. `https://blog.nexcess.net/2014/09/03/the-pros-and-cons-of-implementing-ssl-https/`.

[29] Orendorff, A. Global ecommerce statistics and trends to launch your business beyond borders. `https://www.shopify.com/enterprise/global-ecommerce-statistics`.

[30] Rivest, R. L., Shamir, A., and Adleman, L. A method for obtaining digital signatures and public-key cryptosystems.