Applied Information Technology And Computer Science Vol. 2 No. 2 (2021) 199-218

AITCS

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/aitcs
e-ISSN :2773-5141

Online Food Ordering System

Wong Chun Chuan, Chuah Chai Wen*

[1]Faculty of Computer Science & Information Technology,
Universiti Tun Hussein Onn Malaysia, Parit Raja, 86400, MALAYSIA

DOI: https://doi.org/10.30880/aitcs.2021.02.02.014
Received 15 June 2021; Accepted 09 September 2021; Available online 30 November 2021

Abstract: The online food ordering system is a web-based system that is developed for restaurant Shu Xiang Lou and its customers. The restaurant is a hot pot restaurant and they run their business physically in Kuala Lumpur. An interview is conducted with the restaurant's manager and had found that most of their customer needs to wait for a long time to collect the takeaway order. There is also the potential to lose customers when the phone line is engaged. Besides that, a fake order is an issue that may lead to the restaurant's financial risk as there are real cases that happened in the real world. Another problem found from the interview is that there is difficulty in providing appropriate and updated food information. Also, the manual calculation of the sales report in the restaurant is better to replace by the system and do it automatically. Hence, this project plan to develop a web-based system to handle the order process. The method uses to develop the system is object-oriented system development (OOSD). Java is the main programming language use to develop the system. The system is developed by using Model-View-Control (MVC) framework. A payment gateway is used in the system to prevent the fake order issue. The system also allows updating the restaurant menu easily. A sales report is also calculating automatically by the system. There is strong password management and encryption method to ensure system confidentiality.

Keywords: Online food ordering system, Order system, Online business, Restaurant, Payment gateway, Strong password, Advance Encryption Standard, AES

1. Introduction

E-commerce food and beverage has now become famous among consumers [1]. There are many online ordering delivery systems available now such as KFC, Pizza Hut, Foodpanda, Grabfood and so on. This online ordering delivery system has made the ordering and payment process easier [2]. The consumer only needs a device and internet connection to make the order. The payment process is also performed online.

Restaurant Shu Xiang Lou is a hot pot restaurant located at Kuala Lumpur, Malaysia. The menu of the restaurant includes four category which are royal soup base, vegetables, meats, and drink. The restaurant sells 8 royal soup base, 19 vegetables, 11 meats, and 8 drinks. For the example items in royal soup base are signature spicy hotpot, signature tom yam soup, and herbal soup. For the example items

*Corresponding author: cwchuah@uthm.edu.my

in vegetables category are Chinese cabbage, potatoes, and lotus root. For the example items in meats are australis premium beef slice, spicy beef, and crab meat stick.

Currently, the restaurant offers dine in, takeaway, and delivery services. For dine in service, the restaurant uses traditional method to record orders from customers. The traditional method is pen and piece of paper. Then, the staff key in the order into the order system at the restaurant counter. After finishing the foods, the food bill payment is performed at the restaurant counter. For takeaway service, the customer made order and payment directly at the restaurant counter. For delivery service that is limited to a 5 km radius from the restaurant, the customer orders via phone call. The payment is cash on delivery.

There are only two ways to perform an order in the restaurant. The first way is to order at the restaurant's counter. The second way is to order by phone call. The entire order process involves the restaurant's staff. For the customer order for takeaway service, they need to make long queues and wait for the others to perform their order especially during peak hours. Besides, they may need to wait near the counter until their order is ready for collection.

The restaurant is currently limited to only one phone line. If a phone line is engaged, it is mean that the phone line is already used by someone. The customer is unable to context the restaurant when the restaurant's phone line is engaged. The missed call from the customer does not record in the restaurant's phone. Therefore, there is a potential for the restaurant to lost customers.

For phone call orders, the restaurant prepares the food after receiving the order without paying. Customers pay for the order only when they pick up the food. If the customer does not pick up the food, the prepared food is not suitable for storage or sale to other customers. The restaurant needs to bear the cost. This is the financial risk of the restaurant. There is a real case about fake order is happened in Kuchai Lama, Kuala Lumpur, Malaysia [3]. The same issue occurred in the United States [4].

The fresh ingredients used to support the daily operations of the restaurant may vary according to market conditions. Therefore, in order to provide this unexpected information during the food ordering process, the restaurant's staff need to remember the availability of all foods and notify the customer at the beginning of the ordering process. For example, if the supplier has not supplied "Australia premium beef" recently. Restaurant staff needs to remember and inform the customer that all foods that contain "Australia premium beef" is currently not available. This is because the entire order process is performed by the restaurant's staff. But sometimes, the restaurant's staff may forget such unexpected information. Therefore, this may reduce consumer satisfaction after the decision is made, but in the end, the restaurant is not serving them accordingly.

The restaurant manually calculates the sales report every day. The staff needs to calculate by adding up the total price of all orders on the receipt. The recorded daily sales report is then used to calculate weekly and monthly sales. The sales report shows the restaurant's revenue to the restaurant manager. The daily, weekly, and monthly sales report use by the manager to plan further work. For example, the planning of business expansion may need to refer to the sales report.

There are three objectives for this project – to design, to develop, and to test the online food ordering system for restaurant Shu Xiang Lou. The scope of this project includes customer, admin, and staff.

The proposed system support for three order service, which are dine-in, takeaway, and delivery. Customers able to perform online orders with the system. Delivery and takeaway ordering services is offer in the system for customer. Only registered customers are allowed to place orders online. The ordering process includes three steps. First step, the existing customer log in to the system. Second step, the customer adds the product to the shopping cart. Finally, the customer pays the bill online. The customers may use credit card, debit card, or FPX to complete the payment online.

Only the staff and administration are able to perform dine-in order. This is to prevent the fake order issues. The restaurant does not know whether the customer is at the restaurant or not. If a customer made a dine-in order outside from the restaurant, the restaurant prepare the food order without know whether the order is fake order or not. This leads to financial risks for the restaurant.

Administrators and staff are able to modify menus online. The administrator is allowed to add, update, and delete food in the system. The proposed system allows administrators and staff to modify food status. The food status is changed to "sold out" for the unavailable food and "available" for the available food. The food status is shown in the customer's menu. Therefore, the latest menu is displayed to the customer. Staff may also refer to the system when ordering.

Sales report is calculated automatically by the proposed system. The proposed system calculates sales reports as daily, weekly, and monthly. The calculation is based on the order records in the database. After each order is completed, the sales report changes. The calculation is totally automatically calculated by the proposed system. The calculation by system may reduce human error [5].

Strong password management in the system is to forced users to use a strong password. The proposed system allows user to set their own password. The password must contain at least eight characters with a minimum of one upper-case character, a minimum of one lower-case character, a minimum of one number, and a minimum of one symbol. The combination of characters and integers to the password slows down the brute-force attack [6].

The login attempts in the system are limited to 5 times. Without restricting login attempts, an attacker may use a completely guessing method to perform a password cracking attack [6]. Brute force attacks and dictionary attacks are two examples of password cracking attacks. These two types of attacks work by testing different passwords multiple times until a successful login. Therefore, the proposed system limits the login attempts to prevent the system from password cracking attacks.

The encryption method used in the proposed system is the Advanced Encryption Standard (AES). AES is symmetric encryption that uses the same key for encrypt and decrypt. Encryption may prevent the intruder from misusing the database [7]. User's password is sensitive and private information. The proposed system encrypts the password of the users before storing it in the database. After encrypting, the database only shows ciphertext.

2. Literature Review

Literature review includes Advanced Encryption Standard, confidentiality for the system, Open Web Application Security Project (OWASP), authentication, authorization, Stripe payment gateway, and comparison of the system.

2.1 Advanced Encryption Standard (AES)

AES is a symmetric encryption block cipher algorithm established by Joan Daemen and Vincent Rijmen [8]. AES, which is essential for key data, is widely used to safeguard data confidentiality. The symmetric key (AES) 's high efficiency is suitable for encrypting long plain text [9].

The AES key length and data block length are changed as required. AES is a symmetric block cipher with a 128-, 192-, or 256-bits key length and 128-bits fixed block size [9]. The AES algorithm is applied to a 4×4 array of bytes (128-bit) block-sized array know as a state. According to the key size, using a 128-bit key, the algorithm has ten repetition cycles. For a 192-bit key, the algorithm has 12 cycles. For a 256-bit key, the algorithm has 14 cycles. Each cycle or round consists of four steps: substitute bytes, shift rows, mix columns, and add round key transformation [10].

AES is generally divided into four operating stages: key expansion, initial round, rounds, and final round. Rijndael's key schedule is used during the key extensions step to derive the round key from the

cipher key. Only one process is included in the initial rounds step, namely add round key. The Rounds step consists of four processes, namely sub bytes, shift row, mix columns, and add round key transformation. The last one is the final round step, which process sub bytes, shift row, and add round key transformation [8].

2.2 Confidentiality for the System

Confidentiality is the security principle of information access control. The primary goal of confidentiality is to ensure that only allow authorized persons to access sensitive information, and unauthorized persons are not allowed to access sensitive information [11].

For password-based authentication, a strong password is used to prevent unauthorized access. A password is used to login to an account. The admin account is private and should remain confidential. So, by using a strong password may reduce the risk of an unauthorized person from accessing. Good access control management using a strong password may protect sensitive data from disclosing to an unauthorized person.

Encryption of sensitive data in storage is one of the methods to ensure system confidentiality. Encryption of data in storage is to display unreadable messages in the database to prevent intruders from obtaining user identity data. The Advanced Encryption Standard (AES) is an effective encryption [12]. Section 2.3.2 discusses AES in more detail.

2.3 Open Web Application Security Project (OWASP)

An international non-profit organization dedicated to web application security is the Open Web Application Security Project or OWASP. One of OWASP's core principles is that all materials are freely available and easily accessible on its website. So, OWASP enables everyone to improve the security of their web applications. Documents, instruments, videos, and forums are included in the materials they provide [13].

When using a password for authentication, the key consideration is the strength of the password. "Strong" password policies make it difficult or even impossible for people to guess passwords manually or automatically. There are six characteristics in defining a strong password. These six characteristics include at least eight characters, minimum of one numeric character, a minimum of one lowercase letter, a minimum of one uppercase letter, a minimum of one unique keyboard character, and not the same as the username [14].

When using an authentication mechanism such as login, the application is always responded with an error message. The error messages, such as "Login failed, incorrect username" or "Login failed, incorrect password," are messages that inform the error part to the users. This type of response enables an attacker to distinguish between a wrong username and a wrong password. According to the authentication response guidelines in OWASP, such messages is not suitable to display in the security system [13]. The examples of the correct message are "Login failed, wrong username or password" or "Authentication failed". The two correct examples message is not telling which part is incorrect.

When a user types of passwords, a thief may steal the personal data (such as password) over their shoulder. This attack is called shoulder surfing. If the login form does not hide the password in the text box, the user's password is easily viewed by the thief. Hiding the password in the text box helps prevent this attack [14].

2.4 Authentication

Authentication is the process of confirming the required attributes of an entity. In most cases, entities use credential claims to confirm their identity [15]. The authentication process compares the information enter by the user with the data in the database. The user allows access to the security system only if the information matches the database information. There are two forms of password authentication methods, namely weak password authentication and strong password authentication. In the context of web applications, authentication is usually performed by submitting a username or ID and one or more private information (password) that only a given user knows [16].

2.5 Authorization

Authorization is the process by which specific operating permissions are granted to an entity. Authorization is a security mechanism used to determine levels of access or user privileges associated with system resources (like services, computer programs, data, files, and application functions). Authorization is the process by which access to network resources is allowed or disallowed. This process enables users to access various sources base on their identity [15]. For example, the proposed system only allow admin to add and delete staff. Also, staff unable to view the admin's information.

2.6 Stripe Payment Gateway

For web developers who want to use Stripe's robust API to integrate payment systems into their projects. By bypassing the traditional registration process, Stripe acts as a business account for its providers, handling all merchant approvals [17].

Stripe enforces HyperText Transfer Protocol Secure (HTTPS) for all services that use (Transport Layer Security) TLS (Secure Sockets Layer (SSL)), including its public website and dashboard. Stripe reviews the details of its implementation regularly, including the certificates of its services, the certificate authorities used by it, and the cyphers supported by it. To ensure that the browser only interacts with Stripe over HTTPS, Stripe utilizes HTTP Strict Transport Security (HSTS). The Stripe is also on Google Chrome and Mozilla Firefox's HSTS preload list [18].

AES-256 is used to encrypt all card numbers. The key for decryption is stored on a separate computer. Stripe's internal servers and daemons do not obtain the card number in plain text format but request that the card on the static whitelist be sent to the service provider. Stripe's card number storage, decryption, and transfer infrastructure operates in a different hosting environment and does not share any credentials with Stripe's primary services (API, website) [19].

2.7 Existing Food Order System

KFC delivery and Pizza Hut delivery are the two-existing system of online food order system.

2.7.1 KFC Delivery

The system allows customers to order food in two ways, namely delivery and self-collection. The error message when login does not directly indicate which data is incorrect. The user allows to show or hide the entered password in the text box. The system does not limit the number of passwords retries.

The KFC online ordering system requires users to enter their name, phone number, email address, and password. There is only one requirement for password management in the system. The requirement is to submit passwords with more than six characters and no more than fifteen characters.

The system allows users to pay by cash, debit card, credit card, online banking, and e-wallet. The cash payment is unavailable for contactless delivery and only for orders below RM300 [20].

2.7.2 Pizza Hut Delivery

Pizza Hut's online order system provides two ways to order. Pizza Hut Delivery provides guaranteed hot and fresh pizza within the promised delivery time, and the environment is comfortable, which bring customers more comfort [21]. Self-collection is for customers to order on the system, and the customers themselves pick-up the orders.

The Pizza Hut online order system does not show the text box's password. The entered password is hidden, so others person unable to see the screen's entered password when the user types the password.

The error message displayed when the user logs in with the wrong email or password does not directly indicate which part of the authentication data is incorrect. The message displays "Authentication failed". There is no limit to the number of password retries in this system. The system allows users to enter passwords continuously until they successfully log in.

The Pizza Hut online ordering system requires users to enter their username, email address, mobile phone number, password, and birth date. The system requires that the password contains at least eight characters and meets at least three given four conditions. The first condition is containing numbers zero to nine. Second, contain lowercase letters a to z in the password. Third, contain uppercase letters A to Z in the password. Last, contain non-alphanumeric characters in the password.

Pizza Hut's online ordering system allows its customers to pay through cash, credit cards, online banking, and e-wallets. For more information, cash payment should be made when the food arrives. Other payment methods should be made at the time of ordering.

2.7.3 Comparison of Existing System

Some concept of the proposed online food ordering system is like the existing online food ordering systems, such as the KFC Malaysia online delivery and self-collect system and Pizza Hut Malaysia online delivery and takeaway system. Table 1 shows the differences between these three systems in six aspects.

Table 1: Comparison between the existing system with the proposed system

	KFC	Pizza Hut	Proposed System
Order type	Delivery and self-collect	Delivery and self-collect	Delivery, self-collect, and dine-in
Password in the text box (login form)	Hide or show	Hide	Hide
Password requirement (number of character)	6 to 15 characters	At least eight characters	6 to 20 characters
Password requirement (condition)	No specific condition (allow to use any password with 6 to 15 characters)	Meets at least three of the four conditions following: • Contain numbers 0-9 • Contain lowercase (small) letters a-z • Contain uppercase (capital) letters A-Z • Contain non-alphanumeric characters (special characters)	Meets all five conditions following: • Contain numbers 0-9 • Contain lowercase (small) letters a-z • Contain uppercase (capital) letters A-Z • Contain special characters. • Not same as the username.

Table 1: (cont.)

	KFC	Pizza Hut	Proposed System
Login attempt	No limit	No limit	Three times
Payment method	- Cash on delivery - Credit card - Debit card - Online banking - E-wallet	- Cash on delivery - Credit card - Online banking - E-wallet	- Credit card - Debit card - FPX Online Banking

3. Methodology

Object-oriented System Development (OOSD) is the method use in developing the proposed system There are four phases in OOSD. The first phase is the object-oriented requirement analysis. The second phase is the object-oriented analysis. The third phase is the object-oriented design. The last phase is object-oriented programming and testing. Due to time constraints, object-oriented maintenance will not be implemented in this project.

The object-oriented requirement analysis phase includes the activity of planning and requirement analysis. The planning activity is design 10 interview questions and performs an interview with the manager of Shu Xiang Lou restaurant. The interview question is designed to understand how the restaurant running their business. To know what products, the restaurant sells. How the restaurant handles for the order and payment process. To know whether the restaurant face any fake order issue before. To know how the restaurant calculate for the sales report. To understand the restaurant's requirement about the system. To know who the main users of the system are. This information are collected to be analyzed at the analysis phase.

The requirement analysis activity is to analyze the collected information from the manager of the restaurant. Five main requirements for the proposed system are identified. The first requirement of the system is allowed the customer to perform online ordering for takeaway and delivery service. The second project requirement is to allow staff and administrator to perform dine-in order. The third project requirement is to prevent a fake order, and hence once the customers perform an order for takeaway and delivery service, they must complete the online payment. The fourth project requirement is allowed the staff and admin to modify the menu online. The fifth project requirement of the system is to calculate the sale report automatically.

Next, the objectives and the scopes of the proposed system are identified. There are three objectives in this project, which are to design, to develop and to test the proposed system. The project's scope includes three main users, and there are customer, admin, and staff of the restaurant. There are ten modules included. The modules are register, login, user profile, password reset, menu, cart, payment, payment receipt, sale report, and password management.

There are six software requirements and two hardware requirements for the proposed system. Table 3 has shown the requirement of software and hardware to develop this system.

In object-oriented requirement analysis, information that is collected in the object-oriented requirement analysis phase is analyzed. There are 20 functional requirements, and 8 non-functional requirements have been formulated and analyzed in this phase. The two requirements are required to identify the requirement of the proposed system.

Two existing systems which are KFC online order system and Pizza Hut online order system are studied to understand the concept and uses of the proposed system. The uses of these two existing systems are to allow the customer to perform online ordering. The two existing systems provide two services which are delivery and self-collect. The two existing system shows the availability of the food in the menu. The customer may use the system to order and perform payment easily.

In the object-oriented design phase, the complete architecture for the proposed system is designed. There are four designs in this phase which are database design, classes design, user interface design, and test plan.

For this online food ordering system, seven tables are designed to store the data. All the database tables contain a primary key. Next is the design of the classes. In this project, 21 classes, 10 interfaces, and 11 servlet classes are designed and link together using object-oriented programming. There are attribute and method inside the class.

There are 28 user interfaces (UI) designs in this phase. User interface design for the user to control and understand the system easily. There are 12 user interfaces for admin, four user interfaces for staff, eight user interfaces for the customer, and four other user interfaces.

The test plan also designs in this phase. A test plan is designed to test the food ordering process from customer order, payment order form customer, email verification, phone verification, user management from admin, category management from admin, item management from admin, test the login and register input validation.

In this implementation phase, the design is coded. The design of classes, database tables and user interface are implemented. All the classes and database tables are linked together to make sure that function well. For example, ItemDAOImpl class is linked to the user table. This allows the ItemDAOImpl class to access the data in the user table.

Besides that, user interfaces are linked to classes. For example, the item table is implemented together with the menu module. This allows item information from the item table to be retrieved and displayed in the menu module.

In the testing phase, the designed test plan ensures that the proposed system functions as expected. The test plan contains two categories. First is to test the system functionality. Second is to test the security requirement. If an error happens, the debugging process is taken to ensure that the system functions well. Penetration test also conducted to test the security of the system.

Once it is delivered to the client, the software may change. There may be many reasons why this change is happening. Because of some unexpected input values in the system, a change could happen. For example, the restaurant maybe needs to change its user interface design after a year. Another example is the version of the Application Programming Interface (API) in the source code. The API version is updated for a certain period. The source code needs to change to support by the new version. Due to the limitation of time, this step is not going to implement in this project.

4. System Analysis and Design

System analysis and design show the system architecture design, requirement analysis, use-case diagram, activity diagram, class diagram, entity relationship diagram (ERD), test plan, and user interface design.

4.1 System Architecture Design

The general system architecture includes the system design diagram. figure 1 shows the system design diagram of the proposed system. As the system is a web-based application, the user needs an internet connection to use the system. New admin and staff need to add by older admin, customer need to register at registration page. After the authentication is successful, admin may manage admin, staff, category, item, general setting, view customer, view sales report, and perform order. Staff may manage item, view sales report, and perform order. Customer may view menu, perform order, and payment.

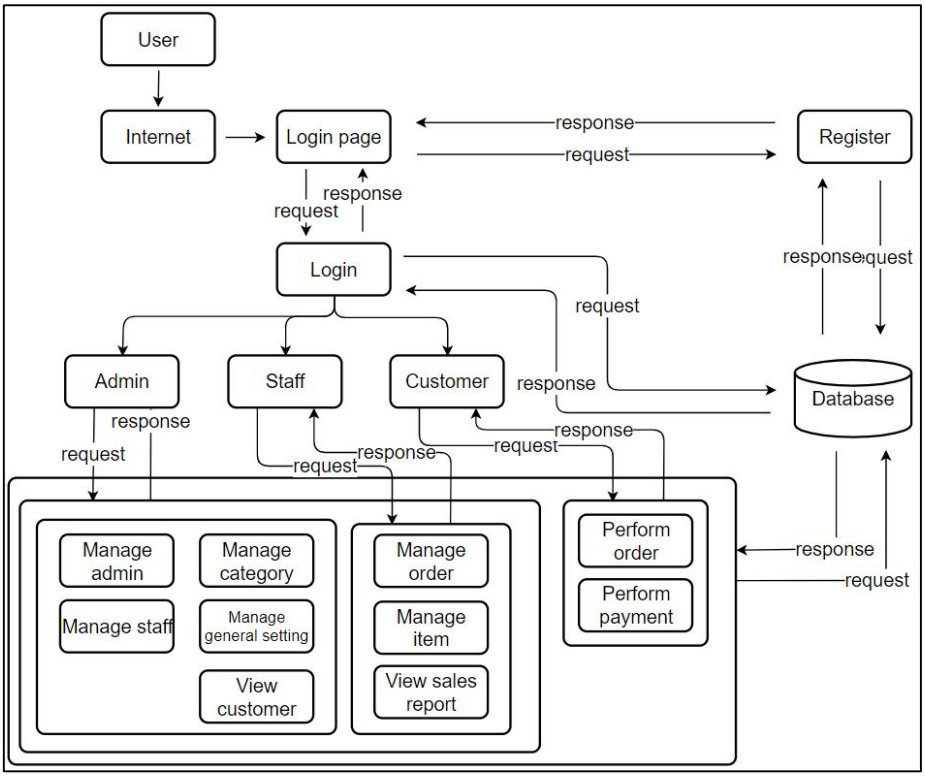


Figure 1: System design diagram

4.2 Requirement Analysis

Requirement analysis involves functional and non-functional requirement. There are 20 functional requirements for the proposed system. Table 2 shows the functional requirement analysis for the proposed system.

Table 2: Functional requirement

No	Functional requirement
1	Customer register as a new customer to the system.
2	Customer able to verify email after registration is successful.
3	Customer able to verify phone number after registration is successful.
4	Systems alert for any invalid input.
5	User able to login with valid email and password.
6	System alert for any invalid input.
7	User able to view, and update profile.
8	System alert for any invalid input.
9	User able to reset password.
10	System alert for any invalid input.
11	Customer able to view menu.
12	Administrator able to view, add, update, and delete the item in the menu.
13	Staff able to view and update item status for the item in the menu.
14	Administrator able to view, add, update, and delete the category.
15	Customer able to view and manage the cart.
16	Customer able to perform payment order.
17	System generates payment receipt for any success order.
18	Customer able to view their own payment receipt.
19	Administrator able to view all payment receipt.
20	System shows the weekly, monthly, and total sales report to the administrator.

There are eight non-functional requirements for the proposed system. Table 3 shows the non-functional requirement analysis for the proposed system.

Table 3: Non-functional requirement

No	Non-functional requirement
1	The system should be in the correct session depend on user authorization.
2	Use NIST internet time for calculation of OTP time out.
3	Automatic log out for 20 minutes inactivates time out.
4	The system only available when there is an internet connection.
5	Users only allow to login into the system with the correct email and password.
6	Encrypt user password with the Advanced Encryption Standard (AES).
7	The user password should include at least eight characters, at least one capital letter, at least one small letter, at least one number, at least one symbol, and not the same as the username.

4.3 Class diagram

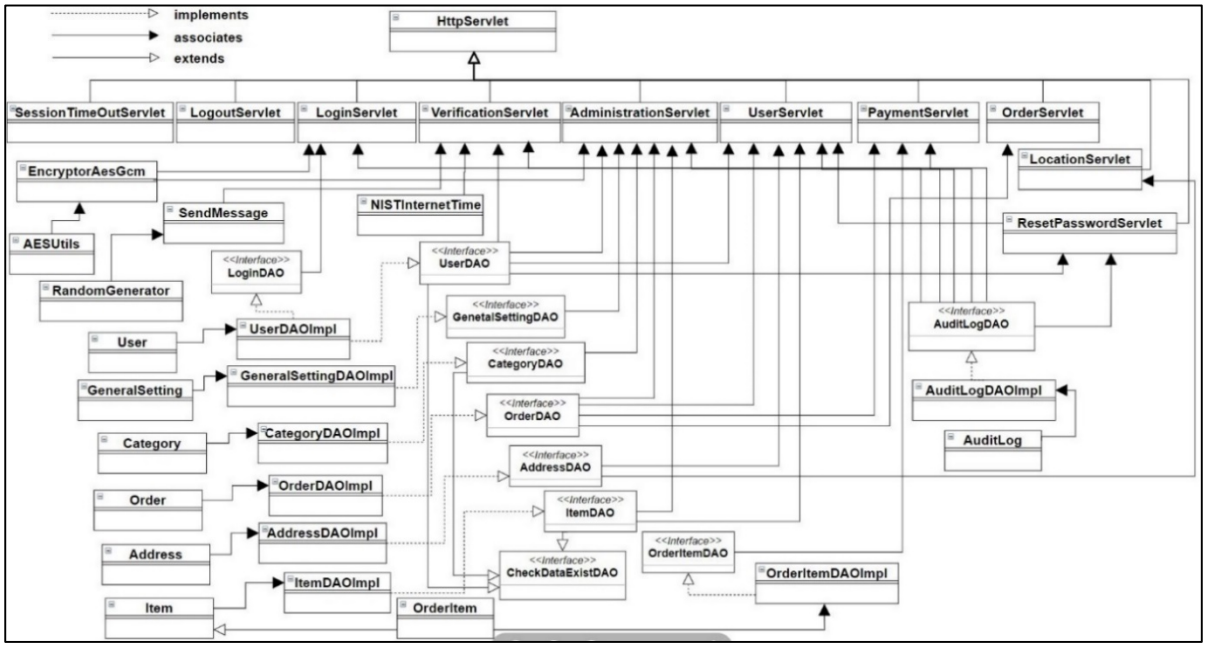


Figure 2: Class diagram for proposed system

Figure 2 shows the class design for the proposed system. The class diagram contains 11 servlets, 10 interfaces, and 21 classes. Data Access Object (DAO) is used to access the data in the database. Which is the class use to execute SQL comment.

The servlets are HttpServlet, SessionTimeOutServlet, LogoutServlet, LoginServler, VerificationServlet, AdministrationServlet, UserServlet, PaymentServlet, OrderServlet, LocationServlet, and ResetPasswordServlet. The servlets handle request from user and response to the user. Servlets contain two methods which are doGet() and doPost().

The 10 interfaces are LoginDAO, UserDAO, GeneralSettingDAO, CategoryDAO, OrderDAO, AddressDAO, ItemDAO, OrderItemDAO, AuditLogDAO, and CheckDataExistDAO. which contain 2 methods, 18 methods, 5 methods, 5 methods, 15 methods, 4 methods, 8 methods, 6 methods, 2 methods, and 1 method, respectively.

The eight DAO implementation classes are UserDAOImpl, GeneralSettingDAOImpl, CategoryDAOImpl, OrderDAOImpl, AddressDAOImpl, ItemDAOImpl, OrderItemDAOImpl, and

AuditLogDAOImpl which contain 19 methods, 5 methods, 6 methods, 16 methods, 4 methods, 9 methods, 6 methods, and 2 methods, respectively.

The five other classes are EncryptorAesGcm, AESUtils, NISTInternetTime, SendMessage, and RandomGenerator. EncryptorAesGcm have 4 attributes and NISTInternetTime have 1 attribute. AESUtils, EncryptorAesGcm, NISTInternetTime, Verification, and RandomGenerator contain 6 methods, 4 methods, 2 method, and 2 methods, respectively.

User, GeneralSetting, Category, Order, Address, Item, OrderItem, and AuditLog are the eight classes that perform the encapsulation concept. User, GeneralSetting, Category, Order, Address, Item, and OrderItem class have 19 attributes, 24 attributes, 7 attributes, 14 attributes, 5 attributes, 13 attributes, 4 attributes, and 8 attributes, respectively. The number of helper methods in each class is double of each attribute number.

4.4 Entity Relationship Diagram (ERD)

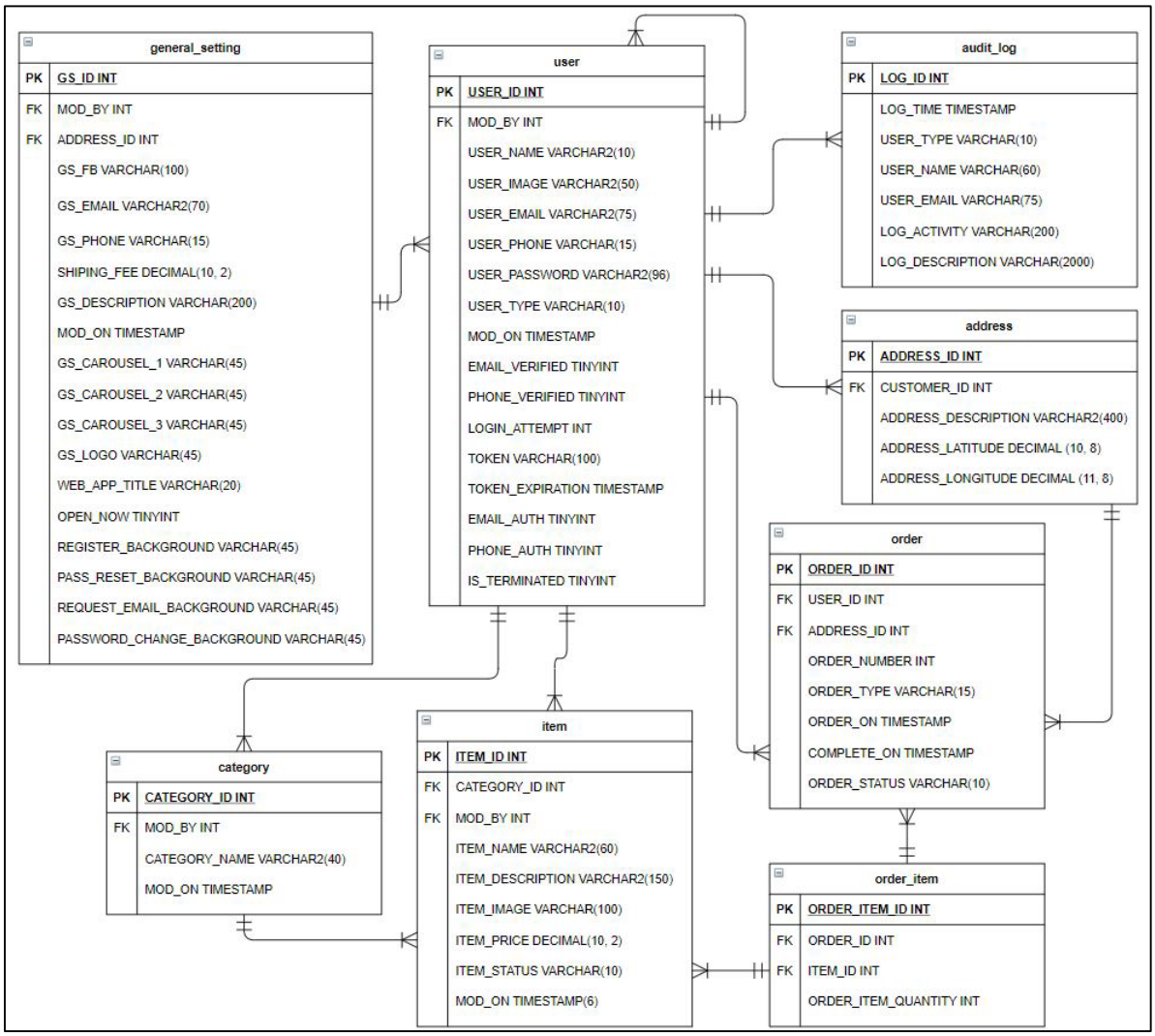


Figure 3: Entity Relationship Diagram (ERD) for proposed system

Figure 3 shows the entity relationship diagram. The diagram shows eight tables. The tables are user, general_setting, category, item, order, order_item, address, and audit_log. Each table contain a primary key. Table general_setting, category, user, and address contain one foreign key. Table item, order, order_item contain two foreign keys.

5. Implementation

This section examines the implementation of the proposed system. The system is implemented based on the design in design phase.

5.1 Implementation of system security module

This section discusses the implementation of security module in the proposed system.

5.1.1 Implementation of encryption

Figure 4 shows the code for AES key generation. The key is generated by using the hash value of the plaintext of the user password.

```
// AES secret key
public SecretKey getAESKey(String key) throws NoSuchAlgorithmException {
    byte[] sha256byte = DigestUtils.sha256(key);
    SecretKey secretKey = new SecretKeySpec(sha256byte, 0, sha256byte.length, "AES");
    return secretKey;
```

Figure 4: Code for AES key generation

Figure 5 shows the code for implement AES encryption for the password. The key that is generated, is passes to the encryption function together with password and generate the ciphertext. Then the function is calling the encryption mode for AES algorithm and use the key that generated form figure 4 to encrypt the password.

```
public byte[] encrypt(byte[] pText, SecretKey secret, byte[] iv) throws Exception {

    Cipher cipher = Cipher.getInstance(ENCRYPT_ALGO);
    cipher.init(Cipher.ENCRYPT_MODE, secret, new GCMParameterSpec(TAG_LENGTH_BIT, iv));
    byte[] encryptedText = cipher.doFinal(pText);
    return encryptedText;
```

Figure 5: Code for implement AES encryption.

5.1.2 Implementation of strong password

Figure 6 shows the code for implementing strong password in the system. The system validates the input password from the user and return error if the input password does not match the strong password policy. The minimum length and maximum length are set to 8 characters and 20 characters, respectively. The password needs to match the regex which is include number, capital letter, small letter, and symbol. The password is not allowed to same as the username and not allow space. If the error return is not null, the system will not consider the input password and the user need to create a new password to match the password policy.

```
String regex = "^(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])(?=."
        + "*[@#$%^&+=_!~<>,;:?*+.\"()\\|\\[\\]\\-\\^\\/])(?=\\S+$).{8,20}$";
Pattern pattern = Pattern.compile(regex);
Matcher matcher = pattern.matcher(password);
if (!matcher.matches()) {
}
if (password == null || "".equals(password)) {
    error = "Empty password is not allow";
} else if (password.length() < 8) {
    error = "Password must be at least 8 characters";
} else if (password.length() > 20) {
    error = "Password can not more than 20 characters";
} else if (!matcher.matches()) {
    error = "Password must contain<br> number(0-9),<br> small letter(a-z),<br> "
          + "capital letter(A-Z), and<br> symbol(@#$%^&+=_!~<>,;:?*+.\"()|[]-^/)";
} else if (userName.equals(password)) {
    error = "Password can not same as username";
} else if (password.length() > 8 && password.length() < 20) {
    for (int i=0; i < password.length(); i++) {
        if (password.charAt(i) == ' ') {
            error = "Password not allow space";
```

Figure 6: Code for validation of password.

5.1.3 Implementation of login attempt

Figure 7 shows the code to add the number of login attempts. The number of login attempt is added when an existing user login with a wrong password. The user is not able to login to the system and need to reset password when the number of login attempt reach five. The number of login attempt is set to zero when the user is successfully login to the system or after reset password.

```
if (!decryptPassword.equals(password) || decryptPassword.equals("")) {
    user.setLoginAttempt(user.getLoginAttempt() + 1);
    loginDAO.updateLoginAttempt(user);
    out.write("Authentication Failed");
```

Figure 7: Code for adding login attempt.

5.1.4 Implementation of captcha

Figure 8 shows the code for reCAPTCHA validation at the server side. The code is implemented to communicate with the google reCAPTCHA API to validate the captcha. The response of true is for valid captcha and response of false is invalid captcha.

```
URL verifyUrl = new URL(SITE_VERIFY_URL);

// Open a Connection to URL above.
HttpsURLConnection conn = (HttpsURLConnection) verifyUrl.openConnection();

// Add the Header informations to the Request to prepare send to the server.
conn.setRequestMethod("POST");
conn.setRequestProperty("User-Agent", USER_AGENT);
conn.setRequestProperty("Accept-Language", "en-US,en;q=0.5");

// Data will be sent to the server.
String postParams = "secret=" + SECRET_KEY //
        + "&response=" + gRecaptchaResponse;

// Send Request
conn.setDoOutput(true);

// Get the output stream of Connection.
// Write data in this stream, which means to send data to Server.
OutputStream outStream = conn.getOutputStream();
outStream.write(postParams.getBytes());

outStream.flush();
outStream.close();

// Response code return from Server.
int responseCode = conn.getResponseCode();
System.out.println("responseCode=" + responseCode);

// Get the Input Stream of Connection to read data sent from the Server.
InputStream is = conn.getInputStream();

JsonReader jsonReader = Json.createReader(is);
JsonObject jsonObject = jsonReader.readObject();
jsonReader.close();

// ==> {"success": true}
System.out.println("Response: " + jsonObject);

boolean success = jsonObject.getBoolean("success");
return success;
```

Figure 8: Code for reCAPTCHA validation

Figure 9 shows the code for calling the function to validate reCAPTCHA. The system shows the message of invalid captcha if the return value is false.

```
String gRecaptchaResponse = request.getParameter("recaptcha");

boolean verify = VerifyRecaptcha.verify(gRecaptchaResponse);
if (!verify) {
    out.write("Invalid captcha");
    return;
```

Figure 9: Code for calling function to validate reCAPTCHA.

Figure 10 shows the code for reCAPTCHA at the front end. The code is implemented in jsp file to show the iframe of the reCAPTCHA on the website. The site-key is getting from the google API.

```
<!-- reCAPTCHA -->
<div id="g-recaptcha" class="g-recaptcha" data-sitekey="6LczFBMbAAAAAOk7j5fDeY_g1q2OJYhi8x8SfFi7"></div>
```

Figure 10: Code for reCAPTCHA in jsp

5.1.5 Implementation of One-Time-Passcode (OTP)

Figure 11 shows the code for generate random integer. The function receives an integer parameter of the length needed of the random integer. A 'for' loop is used to generate the random integer base on the number of lengths received. The final result is then return as an OTP.

```
public String generateRandomInt(int length) {
    Random rand = new Random();
    String OTP = "";
    for (int i = 0; i < length; i++) {
        int intRandom = rand.nextInt(10);
        OTP += Integer.toString(intRandom);
    }
    return OTP;
```

Figure 11: Code for generate random integer.

Figure 12 shows the code for set the OTP expiration time. From the code, the function receives a parameter of the minute that need to be add on the current time. The function getNISTIternetTime() that call in the coding is to get the current time. Calendar is then used to add the received minute to the current time and generate a new timestamp. The function finally returns the new timestamp as the expiration time of the OTP.

```
public Timestamp getFutureTime(int minute) throws IOException {
    // get current timestamp
    Date currentTime = this.getNISTInternetTime();
    // add 90 seconds to current timestamp
    Calendar calendar = Calendar.getInstance();
    calendar.setTimeInMillis(currentTime.getTime());
    calendar.add(Calendar.MINUTE, minute);
    // get timestamp added 300 seconds
    Timestamp newTimestamp = new Timestamp(calendar.getTime().getTime());

    return newTimestamp;
```

Figure 12: Code for set OTP expiration time.

5.1.6 Implementation of secure header

Figure 13 shows the implementation of secure header. From the code, the header set for ten types of headers. The header set in the system are Content-Security-Policy (CSP), X-XSS-Protection, Strict-Transport-Security, X-Content-Type-Options, Cache-control, X-Frame-Options, Cache-Control, Pragma, Expires, and Referrer-Policy.

```
HttpServletResponse httpServletResponse = (HttpServletResponse) response;
httpServletResponse.setHeader("Content-Security-Policy", AddHeaderFilter.POLICY);
httpServletResponse.setHeader("X-XSS-Protection", "1; mode=block");
httpServletResponse.setHeader("Strict-Transport-Security", "max-age=31536000; includeSubDomains; preload");
httpServletResponse.setHeader("X-Content-Type-Options", "nosniff");
httpServletResponse.setHeader("Cache-control", "no-store, no-cache");
httpServletResponse.setHeader("X-Frame-Options", "DENY");
httpServletResponse.setHeader("Set-Cookie", "SameSite=None; Path=/; Secure; HttpOnly");
httpServletResponse.setHeader("Cache-Control", "no-cache,no-store,max-age=0,must-revalidate");
httpServletResponse.setHeader("Pragma", "no-cache");
httpServletResponse.setHeader("Expires", "-1");
httpServletResponse.setHeader("Referrer-Policy", "strict-origin-when-cross-origin");
```

Figure 13: Code for implementing secure header.

6. Result and Discussion

This section discusses the testing result of the proposed system. The testing is based on the check list that designed in design phase. Penetration test also conducted to test the security of the proposed system. The user acceptance test form is sent to the target user and the result is analyzed using the graph form.

6.1 Security Test Plan Result

Security test plan is to test whether the security feature of the developed system is function as the expectation. Table 4 show the result of the security test plan.

Table 4: Security test plan result

No	Check List	Actual Result
1	Ensure the error message not direct indicate which part of the authentication data incorrect. For example, error message should not show “incorrect password” or “incorrect username”.	Pass
2	Enforce the password length inside the policy. For example, minimum eight characters and maximum twenty characters.	Pass
3	Enforce the complexity of the password. For example, requiring use of alphabetic and numeric as well as special character to create password.	Pass
4	Password should be obscured in the text box.	Pass
5	The account is locked, after retrying the wrong password more than five times.	Pass

6.2 Penetration Test Result

The penetration tools use for the test are ImmuniWeb and SiteCheck. ImmuniWeb provides free penetration testing for SSL security testing, web security testing, mobile application security testing and phishing testing [22]. SiteCheck checks any sites on the Internet for malware, spam, blacklists, and other security issues, such as .htaccess redirects and hidden eval codes [23].

Based on the result from ImmuniWeb in figure 14, the system has implemented the HTTP secure header. The secure header involved are Content-Security-Policy (CSP), X-XSS-Protection, Strict-Transport-Security, X-Content-Type-Options, X-Frame-Options, and Referrer-Policy. Content Security Policy is an effective measure to protect the site from XSS attacks. By whitelisting sources of approved content, help to protect browser from loading malicious assets. X-XSS-Protection sets the configuration for the XSS Auditor built into older browsers. HTTP Strict Transport Security is an excellent feature to support on the site and strengthens the implementation of TLS by getting the User Agent to enforce the use of HTTPS. X-Content-Type-Options stops a browser from trying to MIME-sniff the content type and forces it to stick with the declared content-type. X-Frame-Options tells the

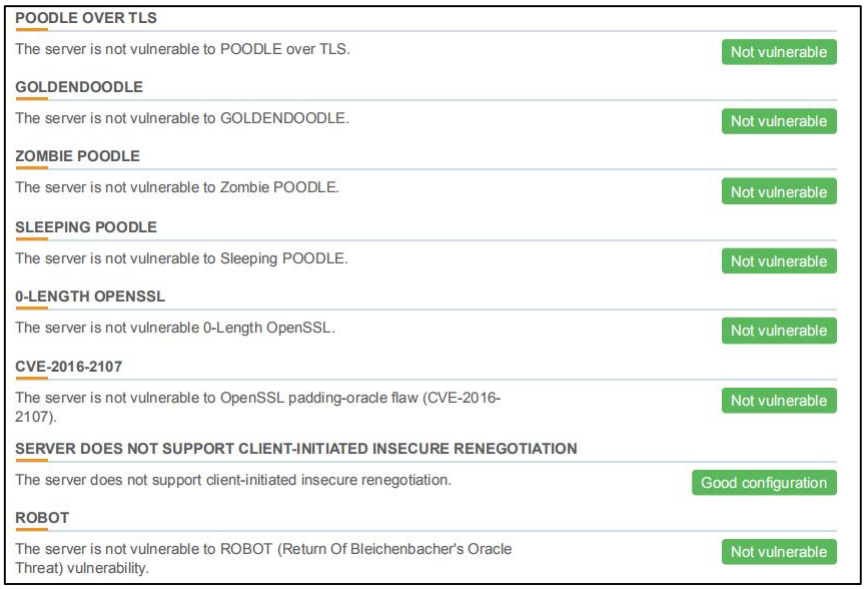
browser whether you want to allow your site to be framed or not. By preventing a browser from framing the site to defend against attacks like clickjacking. Referrer Policy is a new header that allows a site to control how much information the browser includes with navigations away from a document should be set by all sites.

Raw Headers

HTTP/1.1	200
Date	Fri, 04 Jun 2021 14:28:49 GMT
Content-Type	text/html;charset=ISO-8859-1
Transfer-Encoding	chunked
Connection	keep-alive
Server	Apache/2.4.46 (Amazon) OpenSSL/1.0.2k-fips
Content-Security-Policy	**default-src** *'self'* https: ; **style-src** * *'unsafe-inline'*; **img-src** * data: *'unsafe-inline'*; **font-src** * data:; **script-src** *'self'* https: *'unsafe-inline'*;
X-XSS-Protection	**1**; **mode**=block
Strict-Transport-Security	**max-age**=31536000; **includeSubDomains**; **preload**
X-Content-Type-Options	**nosniff**
Cache-control	no-cache,no-store,max-age=0,must-revalidate
X-Frame-Options	**DENY**
Pragma	no-cache
Expires	-1
Referrer-Policy	**strict-origin-when-cross-origin**
Set-Cookie	**SameSite**=None; Path=/; **Secure**; **HttpOnly**
Set-Cookie	JSESSIONID=4B74853101DF8D891FEC317F65E32C6D; Path=/; **Secure**; **HttpOnly**
Vary	Accept-Encoding

Figure 14: Raw header penetration test result

The result in figure 15 states that all the certificates provided by the server are trusted. The server is not vulnerable to POODLE over TLS, GOLDENDOODLE, Zombie POODLE, Sleeping POODLE, 0-Length OpenSSL, OpenSSL padding-oracle flaw (CVE-2016- Not vulnerable 2107), ROBOT (Return of Bleichenbacher's Oracle Not Vulnerable Threat) vulnerability, and Heartbleed attack.



POODLE OVER TLS

The server is not vulnerable to POODLE over TLS. — Not vulnerable

GOLDENDOODLE

The server is not vulnerable to GOLDENDOODLE. — Not vulnerable

ZOMBIE POODLE

The server is not vulnerable to Zombie POODLE. — Not vulnerable

SLEEPING POODLE

The server is not vulnerable to Sleeping POODLE. — Not vulnerable

0-LENGTH OPENSSL

The server is not vulnerable 0-Length OpenSSL. — Not vulnerable

CVE-2016-2107

The server is not vulnerable to OpenSSL padding-oracle flaw (CVE-2016-2107). — Not vulnerable

SERVER DOES NOT SUPPORT CLIENT-INITIATED INSECURE RENEGOTIATION

The server does not support client-initiated insecure renegotiation. — Good configuration

ROBOT

The server is not vulnerable to ROBOT (Return Of Bleichenbacher's Oracle Threat) vulnerability. — Not vulnerable

Figure 15: Vulnerability penetration test result

There are six good configuration results in the test in figure 16. The server enforces cipher suites preference. The HTTP version of the website redirects to the HTTPS version. The server provides HTTP Strict Transport Security for more than 6 months. The server does not support client-initiated secure renegotiation. The server supports secure server-initiated renegotiation. TLS compression is not supported by the server. The server supports TLS_FALLBACK_SCSV extension for protocol downgrade attack prevention.

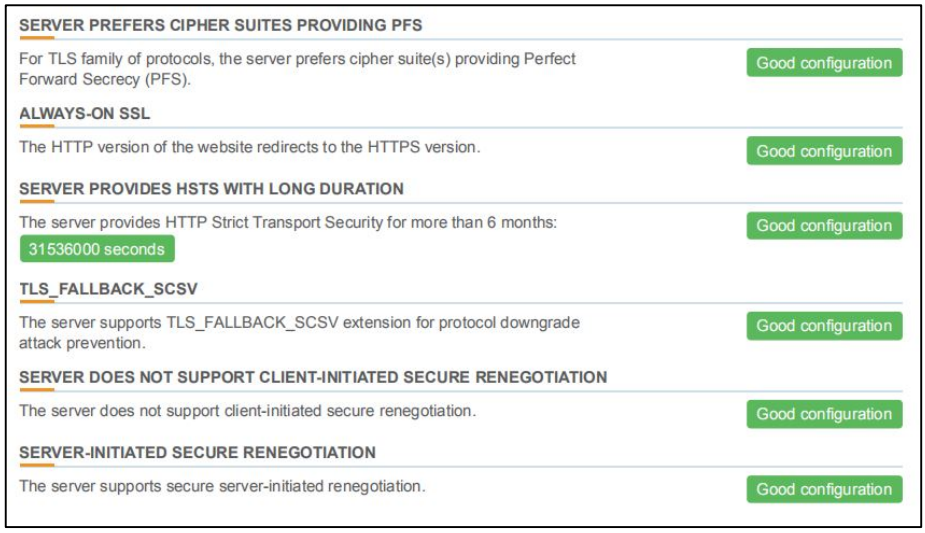


Figure 16: Configuration penetration test result

Figure 17 shows the result from SiteCheck. The result state that there is no malware detected by scan, no injected spam detected, no defacements detected. The website is not a suspected phishing page as the site does not appear to be a forgery or imitation of another website. The website also not a suspected malware provider as website does not appear to contain malicious code. The website is not suspected of unwanted software as the website does not appear to be attempting to install unwanted software.

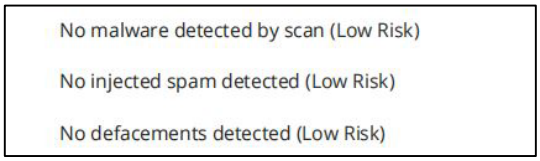


Figure 17: Website malware and security penetration test result

6.2 User Acceptance Result

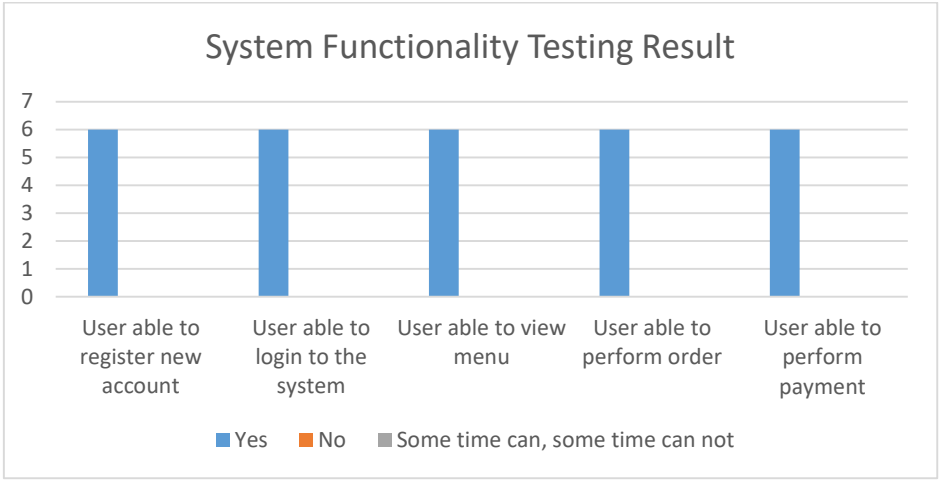


Figure 18: System functionality testing result

Figure 18 shows the result of the system functionality testing. All user able to use the function without error. The function of register, login, menu, order, and payment are test without error and able to perform well.

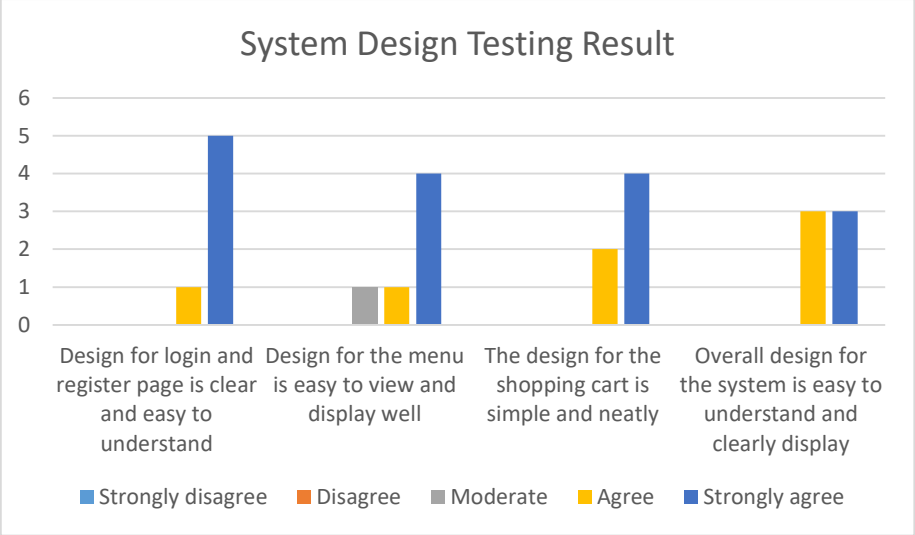


Figure 19: System design testing result

Figure 19 shows the result of the system design testing. Five respondents strongly agree, and one respondent agree that the design for login and register page is clear and easy to understand. Four respondents strongly agree, one respondent agree, and one respondent feel moderate that the design for the menu is easy to view and display well. Four respondents strongly agree, and two respondents agree that the design for the shopping cart is simple and neatly. Three respondents strongly agree, and three respondents agree that the overall design for the system is easy to understand and clearly display.

7. Conclusion

The online food ordering system allow users to perform order, perform payment, and manage the order online. The system has strong password policy to force user to create a strong password. The system also secures the password of the user using AES encryption. The system allows the user to switch on and off the two-factor authentication for login. The captcha is using to prevent automatic account creation.

The system has eight advantages. The system has strong password policy to force user to use a strong password. The system encrypts the user password using Advanced Encryption Standard (AES). The system has captcha to prevent automatic create account. The system has restrictions on login attempts to prevent attackers from trying different passwords multiple times. The system allows the user to switch on the two-factor authentication by receive OTP when login to the system. The system provides a platform for the user to order food online. The system allows the user to perform payment online.

However, the system also has two disadvantages. The system does not have rating feature for the customer to rate the items. The system only has credit card, debit card, and FPX payment method.

Since the web-based system does not have rating feature for the customer to rate the items and the system only has credit card, debit card, and FPX payment method. For future implementation, the system should add the rating function and implement more payment method like e-wallet and PayPal payment method.

Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, University Tun Hussein Onn Malaysia for its support and encouragement throughout the process of conducting this study.

References

[1] R. Shahjee, "THE IMPACT OF ELECTRONIC COMMERCE ON BUSINESS ORGANIZATION," Scholarly Research Journal's, vol. 4, no. 27, pp. 3130-3140, 2016.

[2] S. Hatim, N. A. Mohamad Zamani, L. M. Abdul Latif, N. Kamaruddin, n. ahmad and . m. kardri, "E-FoodCart: An Online Food Ordering Service," International Journal of Innovative Technology and Exploring Engineering (IJITEE), 2019.

[3] Sheralyn, "M'sian Receives Fake Order Of 20 Burgers By Customer Who Claims She's "Ginnyboy"," WORLD OF BUZZ, 27 April 2020.

[4] M. Burke, "Man accused of scamming pizza restaurants with fake large orders for police," NBC News, p. 1, 12 April 2020.

[5] J. M. Haight and R. G. Caringi, "Automation vs. human intervention: What is the best mix for optimum system performance? A case study," International Journal of Risk Assessment and Management, 2007.

[6] A. L.-F. Han, D. F. Wong and L. S. Chao, "Password Cracking and Countermeasures in Computer Security: A Survey," 2014.

[7] P. Singh and K. Kaur, "Database security using encryption," 2015 1st International Conference on Futuristic Trends in Computational Analysis and Knowledge Management, ABLAZE 2015, pp. 353-358, 2015.

[8] M. J. Dworkin, E. B. Barker, J. R. Nechvatal, J. Foti, L. E. Bassham, E. Roback and J. F. Dray Jr., "Advanced Encryption Standard (AES)," Federal Inf. Process. Stds. (NIST FIPS) - 197, 2001 November 2001.

[9] N. Mathur and R. Bansode, "AES Based Text Encryption Using 12 Roundswith Dynamic Key Selection," Procedia Computer Science, 2016.

[10] M. . E. Hameed, . M. . M. Ibrahim and N. . A. Manap, "Review on Improvement of Advanced Encryption Standard (AES) Algorithm based on Time Execution, Differential Cryptanalysis and Level of Security," Journal of Telecommunication, Electronic and Computer Engineering, 2018.

[11] B. Lundgren and . N. Möller, "Defining Information Security," Science and Engineering Ethics, 2019.

[12] J. George and T. Bhila, "Security, Confidentiality and Privacy in Health of Healthcare Data," International Journal of Trend in Scientific Research and Development., 2019.

[13] O. Baker and Q. Nguyen, "A Novel Approach to Secure Microservice Architecture from OWASP vulnerabilities," in Proceedings of the 10th Annua CITRENZ Conference (2019), New Zealand, Nelson, 2019, pp. 54-59.

[14] G. Daniel, G. Franco, J. Danilo and S. Manuel, "Implementation of techniques and OWASP security recommendations to avoid SQL and XSS attacks using J2EE and WS-Security," pp. 1-7, 2017.

[15] M. Trnka, T. Cerny and . N. Stickney, "Survey of Authentication and Authorization for the Internet of Things," Security and Communication Networks, 2018.

[16] M. Farik, N. A. Lal and S. Prasad, "A Review Of Authentication Methods," International Journal of Scientific & Technology Research, pp. 246-249, 2016.

[17] N. M, "E-commerce: Recommended Online Payment Method - PayPal," International Journal of Computer Science and Mobile Computing, vol. 3, no. 7, pp. 669-679, 2014.

[18] K. Nair, "An approach to authenticate magnetic strip bank card transactions at point-of-sale terminals," 2015.

[19] S. Supriyati and E. Nurfiqo, "Effectiveness of Payment Gateway in E-Commerce," 2019.

[20] KFC Malaysia, dinein.kfc.com.my, 2020. [Online]. Available: https://dinein.kfc.com.my/faq. [Accessed 12 Nov 2020].

[21] PizzaHut, pizzahut.com.my, 2020. [Online]. Available: https://www.pizzahut.com.my/aboutus/my. [Accessed 12 Nov 2020].

[22] ImmuniWeb, 2021. [Online]. Available: https://www.immuniweb.com/. [Accessed 5 Jun 2021].

[23] "Sitecheck," 2021. [Online]. Available: https://sitecheck.sucuri.net/. [Accessed 5 Jun 2021].