Applied Information Technology And Computer Science Vol. 4 No. 2 (2023) 83-102

AITCS

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

Web-Based Auction System with Dual-Authentication for Syarikat Perniagaan Fong Yuen Sdn. Bhd.

Chong Zhi Qi[1], Sapiee Jamel[1]*

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

DOI: https://doi.org/10.30880/aitcs.2023.04.02.006
Received 06 November 2023; Accepted 06 November 2023; Available online 30 November 2023

Abstract: Effective sales techniques are essential for business success, and auctions offer a quick and mutually beneficial approach to buying and selling. Syarikat Perniagaan Fong Yuen Sdn. Bhd., a furniture provider based in Kulai, Johor, Malaysia, faces challenges in their sales process. To address the issue of lengthy negotiations causing transaction delays, a web-based auction system is proposed. This system will enable the company to conduct product auctions, simplifying the negotiation process and reducing transaction delays. The system will feature easy registration, bidding, and payment processes for customers, improving the overall user experience. Following the prototyping model, the system will undergo iterative improvements based on user feedback. Security is a top priority, with dual-authentication implemented to ensure the safety of the system for administrators and bidders. The ultimate objective is to establish a secure and efficient platform that streamlines the sales process, benefiting both company and its customers.

Keywords: Auction system, Web-based application, Dual-authentication, Payment

1. Introduction

A web-based auction system with dual-authentication is being developed for Syarikat Perniagaan Fong Yuen Sdn. Bhd., a furniture provider established in 1996. The company offers a diverse range of furniture for various settings such as offices, hostels, schools, hospitals, and churches. The company's address is 19, Jalan SME 2, Kawasan Perindustrian SME, 81000 Kulai, Johor, Malaysia. In order to provide customers with a bargaining channel to purchase their products, a web-based auction system is proposed. This system will enable the company to set up product auctions and allow customers to easily register, bid, and make payments.

Currently, furniture sales are conducted through the company's website, social media platforms, and physical sales at their showroom. Customers can request product quotations by filling out a form on the

Corresponding author: sapiee@uthm.edu.my

website or through social media channels. The form requires customers to provide their name, email, contact number, and inquiry details. Company employees respond to customers via email, and once a price agreement is reached, an invoice is generated and either handed to the buyer in person or sent via email. The invoice includes the purchased items, the amount owed, and payment methods. Upon receiving the payment receipt, the company arranges furniture installation and delivery services to the specified address.

Based on their experience with various sales transactions, the company has identified several issues with the current sales process. Negotiating purchase terms often takes too long, resulting in potential delays. Additionally, the company website only allows customers to view products and request quotations, limiting customer engagement. To address these concerns and attract a larger customer base, the proposed auction system offers an efficient solution that enhances entertainment value and enables customers to purchase products at discounted prices. The system will be accessible online, allowing anyone in Malaysia with digital devices and internet access to participate. This will help the company reach a wider audience as the internet has become an integral part of daily life.

To ensure system security, dual authentication will be implemented, requiring customers to undergo One-Time Passcode verification during each login session. This measure ensures that only authorized individuals can access their online accounts. In addition, several security features will be implemented to create a secure environment for users to purchase their desired products, such as strong password management, security images and phrases, Captcha, and input validation.

This article is organized into six sections, covering various aspects of the project. It starts with the background, followed by an analysis of related work. The methodology section explains the project's approach and design process. Implementation details and testing results are discussed in the fourth section. The project conclusion highlights contributions, limitations, and suggestions for future improvements. The final section expresses gratitude and acknowledgments to those who supported the project.

2. Related Work

There are various sales techniques available for organizations to sell their products. One such technique is auctions, where potential buyers bid competitively for products or services in open or closed forms. Bidders compete with each other, with each subsequent bid being higher than the previous one, and the highest bidder wins the items. Auctions attract people due to their relatively low starting prices, providing an opportunity for buyers to purchase products at discounted prices. For sellers, auctions create a competitive environment that maximizes their bargaining power and allows them to achieve higher prices [1].

There are two types of auctions which are live auctions and online auctions. Live auctions require bidders to physically attend the auction at a specific time and location. The advantage of live auctions is that bidders have a preview period to inspect the items and assess their condition before bidding. However, this limits the participation of interested bidders who are located outside the auction's city or country and are unavailable on the specified date. On the other hand, online auctions have overcome this limitation by enabling goods to be sold on the Internet, allowing anyone to bid on products from anywhere and at any time. Online auctions typically utilize time-interval bidding, where bidding can take place over days or weeks but must be completed within a specified deadline [11].

In the case of Syarikat Perniagaan Fong Yuen Sdn. Bhd., the current sales technique involves customers requesting product quotes through the company's website or social media. This process

requires customers to fill out a form on the website, providing their names, emails, contact numbers, and inquiry information. However, this approach can be cumbersome and prone to errors, as customers may enter incorrect information. Such inaccuracies can lead to customers not receiving reply messages and the company potentially missing out on potential customers. To address these issues, a web-based auction system is proposed to change the company's sales technique.

The proposed system is a web-based application accessible via a website link, allowing users to access it from any digital device with an internet connection and web browser. This system offers the advantages of real-time information retrieval and greater availability of information for both the company and its users [2]. However, the use of information technology systems also poses risks, particularly in terms of cybersecurity. The digitization of data has significantly increased the prevalence of cybercrimes, and the lack of awareness and practice regarding cybersecurity has contributed to their rapid growth [3].

To ensure information security in this project, the focus is on maintaining the confidentiality, integrity, and availability of data. Confidentiality aims to prevent identity theft by keeping customers' personal information confidential [4]. All system information must be protected to prevent unauthorized alterations, and data availability ensures smooth company operations by allowing authorized entities to access and retrieve information when needed [5]. Several security features will be implemented in the proposed system, including the Advanced Encryption Standard (AES) for encrypting sensitive data, strong password management practices, Captcha to distinguish between humans and computer programs [9], security images and phrases to protect against phishing attacks [6], and dual authentication for additional account security [7]. Java programming will be utilized to develop the proposed system due to its "write once, run anywhere" capability, making it suitable for different platforms accessing the same web page on the Internet [8].

Today, there are several online auction systems that allow sellers to list their products for auction and enable buyers to bid on these products online. Therefore, conducting a review of existing systems becomes crucial as it helps in understanding the system architecture and provides guidelines for developing the project. Three existing systems have been reviewed and studied which are eBay, Lelong.my, and Hanamaru Auction Malaysia, as shown in Table 1.

Table 1: Comparison of existing auction systems with proposed system

System/ Specification	eBay	Lelong.my	Hanamaru Auction Malaysia	Web-based auction system
Types of auction system	Marketplace auction platform online	Marketplace auction platform online	Direct auction platform online	Direct auction platform online
Register and Login	Yes	Yes	Yes	Yes
Enforce users to create strong password	Yes	No	No	Yes
Dual authentication	Yes	No	No	Yes
Make payment through website	Yes	Yes	No	Yes
Change and forgot password feature	Yes	Yes	Yes	Yes
Hypertext Transfer Protocol Secure (HTTPS) Website	Yes	Yes	Yes	Yes

The comparison between the existing auction systems and the proposed web-based auction system in Table 1 reveals several key differences. eBay and Lelong.my are marketplace auction platforms, while Hanamaru Auction Malaysia and the proposed system are direct auction platforms. All systems offer registration and login functionalities. eBay and the proposed system enforce users to create strong passwords, whereas Lelong.my and Hanamaru Auction Malaysia do not have this feature. Only eBay and the proposed system includes dual authentication for added security. eBay and Lelong.my allow payments through their websites using various options such as credit and debit cards, FPX Online Banking, and E-Wallet. While Hanamaru Auction Malaysia does not support website-based payments.

The proposed system facilitates payment through FPX Online Banking. All systems provide features for changing and recovering passwords. All systems use Hypertext Transfer Protocol Secure (HTTPS) for secure communication. Based on this comparison, the proposed web-based auction system incorporates desirable features such as strong password enforcement, dual authentication, and convenient payment options.

3. Methodology

This section provides an overview of the methodology and analysis employed in this project. The system development for the web-based auction system follows the prototyping model. It contains requirement analysis, Unified Modeling Language diagrams, database design, and general system architecture to ensure a thorough understanding of the system's structure and functionality.

3.1 System Development

Prototyping model is used in this project to develop the proposed web-based auction system because it have the potential to make a system that can meet the stakeholder’s requirements. The feedback of prototype provided by the users can help the developer to improve the prototype. The missing functions and errors of the system can be easily detected and the users will be more understanding of the system being developed [10]. The prototyping model includes six phases which are requirements phase, quick design phase, build prototype phase, user evaluation phase, refining prototype phase, implement and maintenance phase.

The system development process follows the prototyping model, which consists of several phases as shown in Table 2. In the requirements phase, the project requirements are collected, proposed, and analyzed using techniques like UML and class diagrams. The quick design phase focuses on designing the GUI, database, and functional modules of the system. In the build prototype phase, a functional prototype is designed and developed based on the quick design specifications. The user evaluation phase involves user testing, feedback collection, and documenting user acceptance testing results. The refining prototype phase focuses on modifying the prototype based on user feedback to create the final prototype. In the implement and maintain phase, the final prototype is implemented, system testing is conducted using test cases, the system is implemented, and ongoing maintenance and support are provided. This iterative approach allows for continuous improvement and ensures the system meets stakeholders' requirements.

Table 2: System Development activities and their tasks

Phase	Task	Output
Requirements	Collected the requirements, Proposed the project, Determine the project schedule, activities and output andAnalysis the requirements	• Proposal • Gantt Chart • UML Diagram • Class Diagram • ERD Diagram
Quick Design	Design GUI, Design database and Design module	• Database design • User interface design • Functional module design
Build prototype	Design prototype and Build the prototype	• Design specification • System architecture
User Evaluation	User testing	• User feedback • Test plan • User Acceptance Testing form
Refining prototype	Modified prototype	• Final prototype
Implement and maintain	Implement the final prototype, System testing, Implement the system and maintain the system	• Test cases • Completed system.

3.2 Requirement Analysis

Requirements analysis is the process of determining user expectations for a new or modified product. It involves collecting and analyzing system requirements, including functional and non-functional requirements, user requirements, and hardware and software requirements.

Table 3: System functional module

No	Module	Function	User
1.	User Registration Module	Allow user to register an account as a bidder to access the pages of the system	Bidder
2.	Login Module	Allow user and admin to login the system	Admin and bidder
3.	Admin Module	Allow the admin to manage and update the information of the admin, products, and bidder	Admin
4.	User Module	Allow the bidder to view all products, edit their profile information, bid products, view own bidding status and payment status	Bidder
5.	Bidding Module	To check whether the incremental amount entered by the bidder is equal to or more than the minimum incremental value, and check the highest bidder	Admin and bidder
6.	Payment Module	Allow the winning bidder to make payment by doing an online transfer and only last for the agreed period.	Winning bidder

Table 3 outlines the functional modules of the proposed web-based auction system. These modules cater to the specific needs of different users. The User Registration Module enables individuals to register as bidders, granting them access to relevant system pages. The Login Module ensures secure authentication for both users and admins. The Admin Module empowers admins to manage system information, including admin, product, and bidder details. The User Module provides bidders with various capabilities, such as viewing products, editing profiles, bidding, and monitoring payment and bidding statuses. The Bidding Module handles the bidding process, validating bid increments and

determining the highest bidder. The Payment Module facilitates online transfers for winning bidders within specified timeframes, ensuring smooth and secure transactions.

Table 4 summarizes the functional requirements of the proposed system. It outlines the specific functionality of each module, including user registration, login, admin module, user module, bidding, and payment. The table highlights the system's capabilities, such as input validation, error alerts, and enforcement of certain actions.

Table 4: Functional requirements for the proposed system

No	Module	Description
1.	User Registration Module	• Users register as bidders, with the system alerting for invalid input.
2.	Login Module	• Users input correct credentials and undergo One-Time Password verification, with the system alerting for invalid input and enforcing the first login admin to change their password.
3.	Admin Module	• Admins can register new admins, set up auction products, view auction progress and winners, monitor successful bidders' payment status.
4.	User Module	• Bidders can place bids on desired products, cancel their bids, edit their profile information, view their bidding and payment statuses
5.	Bidding Module	• The system verifies if the bid increment meets the minimum value, determines the highest bidder when the auction reaches its end date and time, and updates the bidding status accordingly.
6.	Payment Module	• The successful bidder enters shipping details, makes payment through online banking transfer within the agreed period, and the system updates the payment status accordingly.

Table 5 outlines the non-functional requirements of the proposed system, including performance, operational characteristics, and security measures. These requirements aim to ensure reliable system performance, dependency on internet connectivity, and robust security measures to protect user data and access.

Table 5: Non-functional requirement for the proposed system

No	Requirement	Description
1.	Performance	• The system should locate the correct session according to the user's authorization.
2.	Operational	• The system is only accessible when an internet connection is available.
3.	Security	• Users may access the system with the correct username, password, and One-Time Password. • The password is encrypted by using Advanced Encryption Standard (AES) in database. • The password must be more than 10 characters and combination of uppercase, lowercase, number, and special symbols. • A security image must be chosen and create a security phrase.

Table 6 outlines the user requirements for the proposed system, with separate categories for Admin and Bidder users. Admin users should be able to authenticate, manage admins and auction products, view transaction details, monitor bidder payments, and perform logout. Bidder users should have authentication, access to auction products, bidding capability, payment functionality, profile editing, viewing bidding and payment status, and logout.

Table 6: User requirements of the proposed system

No	User	Description
1.	Admin	Able to authenticate with a valid username and password, add/delete admin information, manage auction products, view transaction details, monitor bidder payment status, and perform logout functionality.
2.	Bidder	Able to authenticate with a valid username and password, view auction products, place bids, make payment after winning a bid, edit profile information, view their own bidding list and payment status, and perform logout functionality.

The hardware requirements for the web-based auction system including a minimum processor speed of 2.10 GHz or higher, a minimum of 8.00 GB RAM, a stable internet connection of 100Mbps, and a minimum storage capacity of 256GB or higher. The software requirements for the web-based auction system include Eclipse IDE for interface creation and system function development, MySQL Workbench for database creation and management, and Microsoft Edge and Google Chrome for testing and running the system.

3.3 System Analysis

The use case diagram for the admin in Figure 1 illustrates the various actions they can perform, including logging in, registering new admins, setting up auction products, monitoring auction progress, and managing bidder payment status. Admins can also log out of the system when they are done. The use case diagram for the bidder in Figure 2 outlines the actions available to them, such as logging in, registering as new bidders, placing bids on products, making payments, entering shipping details, editing profile information, and monitoring bidding and payment statuses. Bidders can securely log out of the system as well.

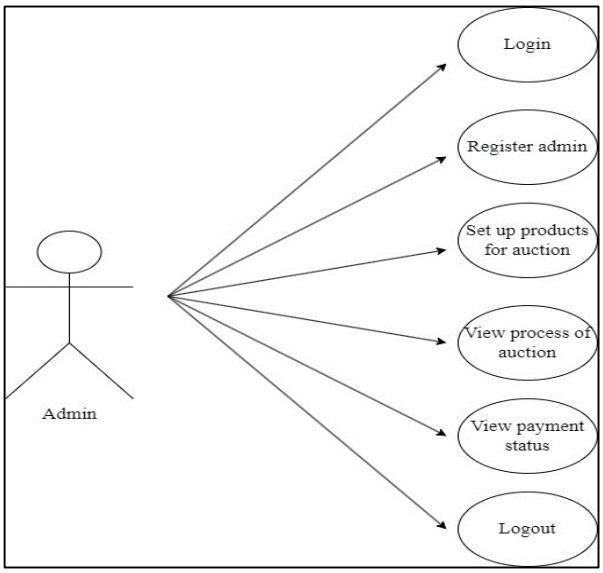

Figure 1: Use case diagram for admin

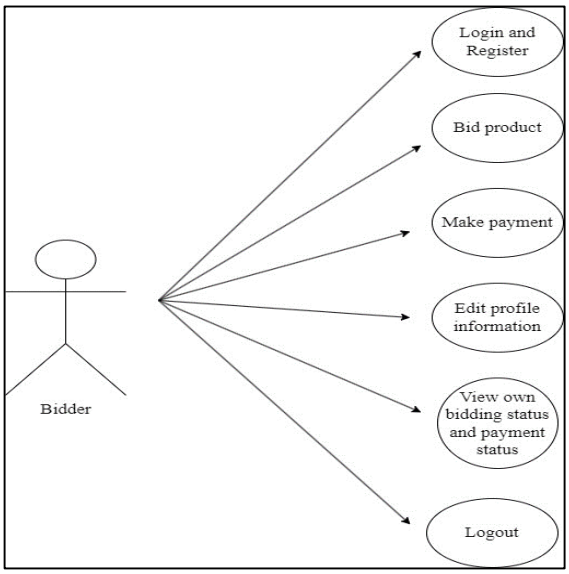

Figure 2: Use case diagram for bidder

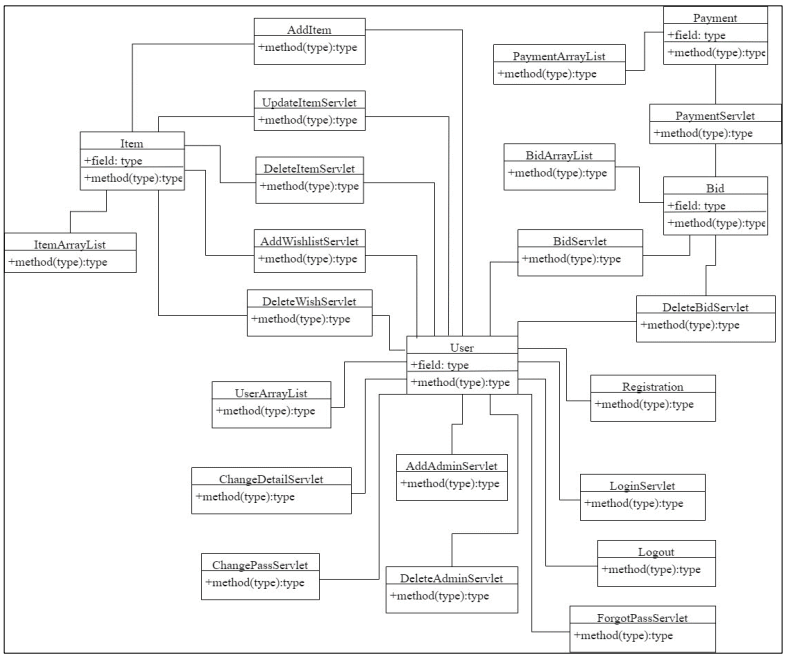

Figure 3: Class diagram of proposed system

Figure 3 presents the class diagram for the proposed system, comprising 24 classes with private attributes and public functions. The user, item, bid, and payment classes play vital roles in managing user information, item listings, bidding processes, and payment transactions. The diagram also includes supporting classes to enhance system functionality. These classes form the core of the system, enabling users to interact with items, place bids, and complete payments.

Figure 4 depicts the storage of registered admin details in the database. Admins log in using their email and password, with the login module performing verification and user identification. The setup product module allows admins to input auction product details, which are saved in the database. Admins can also view bidding process and results through the view bidding status module and monitor bidder payment status using the view payment status module. Figure 5 illustrates the registration of bidders in the system, with their details stored in the database. Bidders login using their email and password, and the login module verifies their credentials. Bidders can place bids on desired products, view their bidding status, make payments, and monitor their payment status.

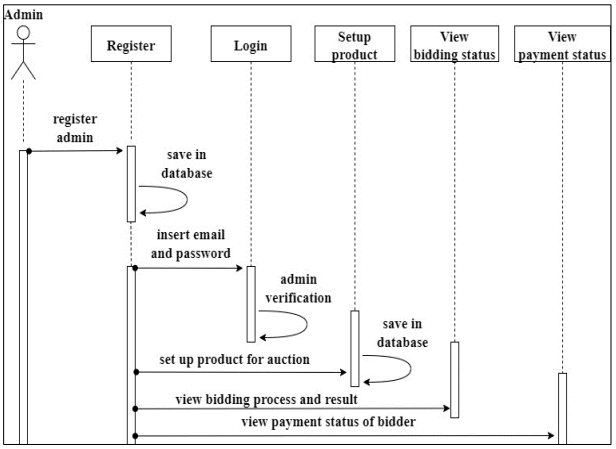

Figure 4: Sequence diagram for admin

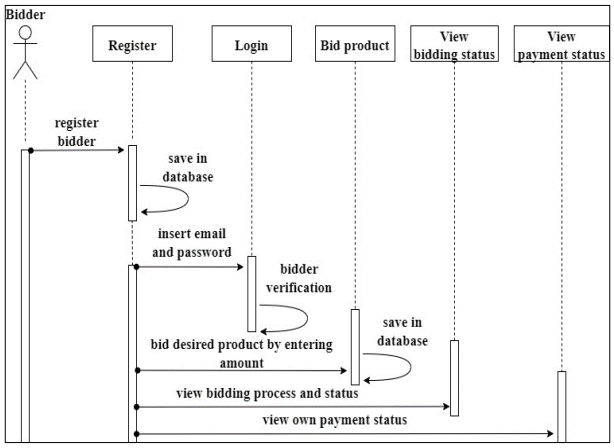

Figure 5: Sequence diagram for bidder

Figure 6 presents the entity relationship diagram (ERD) for the proposed system. It includes seven entities which are user, item, bid, address, security_images, wishlist, and payment. These entities represent different aspects of the system, and their relationships define how they are connected and interact with each other.

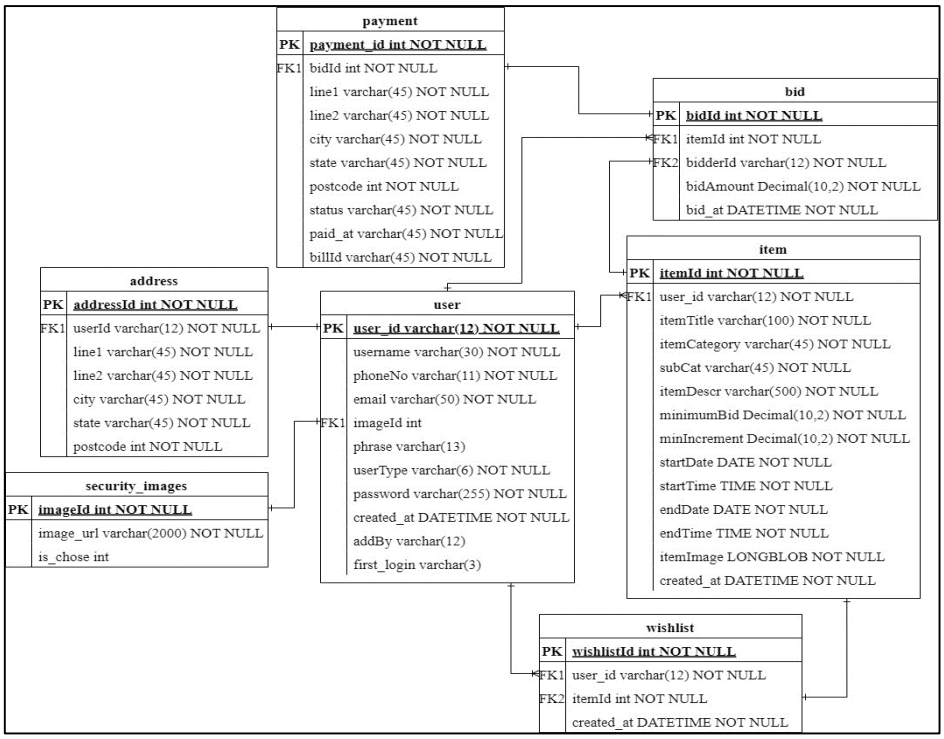

Figure 6: Entity Relationship Diagram (ERD) for the proposed system

Figure 7 illustrates the architecture design of the proposed system. The design includes registration for bidders, login functionality for both admins and bidders with One-Time Password verification, admin management features, auction product setup, monitoring of bidding and payment statuses, and the ability for bidders to view, bid on, and make payments for auctioned products. The system ensures secure access and provides a comprehensive platform for managing auctions and facilitating successful transactions between bidders and admins.

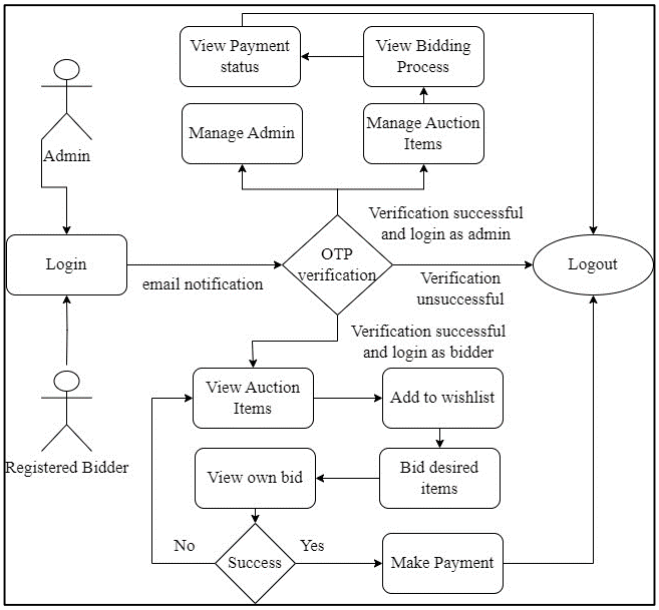

Figure 7: System architecture design of the proposed system

4. Implementation and Testing

This section focuses on system implementation and testing for the web-based auction system. It covers the implementation of security modules and individual module functionalities. Additionally, it discusses the comprehensive functional testing conducted to assess the system's performance and adherence to desired standards.

4.1 Implementation of Security Module

Figure 8 showcases the code snippet used to verify the user's session. The code checks if the "id" attribute is present in the user's session, and if not, redirects the user to the homepage. If the "id" attribute exists, indicating a valid session, the code further examines the current servlet path. If the path corresponds to the "checkSession.jsp" page, which is deemed redundant, the user is redirected back to the homepage.

```
<%
if(session.getAttribute("id")==null){
    response.sendRedirect("home.jsp");
}else {
      if (request.getServletPath().equals("/checkSession.jsp")) {
          response.sendRedirect("home.jsp");
        }
}
%>
```

Figure 8: Code that Checks the User's Session.

The code in Figure 9 verifies the user's session and checks if they have admin privileges. If the "type" attribute in the session is not null, indicating a non-admin user, and they attempt to access an admin-only page, they are redirected to the "home.jsp" page.

```jsp
<%
if(session.getAttribute("id")==null || session.getAttribute("type") != null){
    response.sendRedirect("home.jsp");
}else {
      if (request.getServletPath().equals("/checkSessionAdmin.jsp")) {
          response.sendRedirect("home.jsp");
        }
}
%>
```

Figure 9: Code for user session and admin type validation.

The code in Figure 10 validates the user's session and ensures they have the bidder type. If the "type" attribute is null, indicating a non-bidder user, and they try to access a bidder-only page using the correct URL, they are redirected to the "home.jsp" page.

```jsp
<%
if(session.getAttribute("id")==null || session.getAttribute("type") == null){
    response.sendRedirect("home.jsp");
}else {
      if (request.getServletPath().equals("/checkSessionBidder.jsp")) {
          response.sendRedirect("home.jsp");
        }
}
%>
```

Figure 10: Code for user session and bidder type validation

```java
private static final String SECRET_KEY = "my_super_secret_key";
public static String encrypt(String strToEncrypt) {
    try {
            byte[] salt = generateSalt();
            byte[] iv = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
            IvParameterSpec ivspec = new IvParameterSpec(iv);

            SecretKeyFactory factory = SecretKeyFactory.getInstance("PBKDF2WithHmacSHA256");
            KeySpec spec = new PBEKeySpec(SECRET_KEY.toCharArray(), salt, 65536, 256);
            SecretKey tmp = factory.generateSecret(spec);
            SecretKeySpec secretKey = new SecretKeySpec(tmp.getEncoded(), "AES");

            Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
            cipher.init(Cipher.ENCRYPT_MODE, secretKey, ivspec);
            byte[] encryptedBytes = cipher.doFinal(strToEncrypt.getBytes(StandardCharsets.UTF_8));

            byte[] combinedBytes = new byte[salt.length + encryptedBytes.length];
            System.arraycopy(salt, 0, combinedBytes, 0, salt.length);
            System.arraycopy(encryptedBytes, 0, combinedBytes, salt.length, encryptedBytes.length);

            return Base64.getEncoder().encodeToString(combinedBytes);
        } catch (Exception e) {
            System.out.println("Error while encrypting: " + e.toString());
        }
    return null;
}
```

Figure 11: Code to Encrypt Password

Figure 11 demonstrates the encryption process used in the system, utilizing AES encryption with a 256-bit key. The secret key is generated using the PBKDF2WithHmacSHA256 algorithm, combining it with a random salt and initialization vector (IV) to ensure unique encryption results. The resulting ciphertext is encoded using Base64 for storage in the database. This approach enhances security by preventing attackers from extracting information or detecting patterns from repeated ciphertexts.

```javascript
function validatePass(){
    var pass = document.getElementById('inputPassword').value;
    var passinput = document.getElementById('inputPassword');

    if(pass.length == 0){
        passinput.classList.remove('valid-input');
        passinput.classList.add('error-input');
        passError.innerHTML='Password is required';
        return false;
    }

    else if(!pass.match("^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-9])(?=.*?[#?!@$%^&*-]).{10,}$")){
        passinput.classList.remove('valid-input');
        passinput.classList.add('error-input');
        passError.innerHTML='Password should follow the guidelines';
        return false;
    }
    passinput.classList.remove('error-input');
    passinput.classList.add('valid-input');
    passError.innerHTML = '<i class="fas fa-check-circle"></i>';
    return true;
}
```

Figure 12: Function to Validate the Password Format

The code in Figure 12 validates user-input passwords, checking for empty fields and enforcing specific password guidelines such as length, uppercase, lowercase, digit, and special character requirements.

```javascript
if(recaptchaResponse.length == 0){
        recaptchaError.innerHTML = 'Please complete the reCaptcha verification.';
        return false;
}
```

Figure 13: Code to Validate the reCaptcha verification.

The code presented in Figure 13 is responsible for validating whether the user has successfully completed the reCaptcha verification before submitting the register and login form.

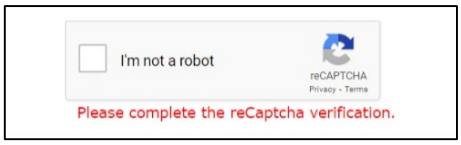

Figure 14: Implementation of reCaptcha

Figure 14 demonstrates the integration of the reCaptcha feature, specifically utilizing Google Captcha, to prevent bot attacks and enhance security.

```java
String email = request.getParameter("email");

UserArrayList userArrayList = new UserArrayList();
ArrayList<User> image = userArrayList.getImageandPhrase(email);
    if(image != null){
        for(User imageList:image){
                String url = imageList.getImageUrl();
                String phrase = imageList.getPhrase();
                response.setContentType("text/plain");
                response.setCharacterEncoding("UTF-8");
                response.getWriter().write(url+'|'+phrase);
            }
    }
    if(image.size()==0){
        String ALPHA_NUMERIC_CHARS = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";
        Random random = new Random();
        StringBuilder sb = new StringBuilder(9);

        for (int i = 0; i < 9; i++) {
             int index = random.nextInt(ALPHA_NUMERIC_CHARS.length());
             char randomChar = ALPHA_NUMERIC_CHARS.charAt(index);
             sb.append(randomChar);
         }
        String phrase = sb.toString();
        ArrayList<User> randomPhoto = userArrayList.getRandomImage();
        if(randomPhoto != null){
            for(User randomImage:randomPhoto){
                String url = randomImage.getImageUrl();
                response.setContentType("text/plain");
                response.setCharacterEncoding("UTF-8");
                response.getWriter().write(url+'|'+phrase);
            }
        }
    }
```

Figure 15: Code to Display the Correct Security Image and Phrase

Figure 15 demonstrates the code implementation to display the correct security image and phrase after the user enters their registered email. It retrieves the associated images and phrases using the provided email. If matching images and phrases are found, they are written to the response. If no matching images and phrases are found, a random phrase is generated and a random image URL is retrieved from a list of random images, which are then written to the response.

Figure 16: Security Image and Phrase According User's Account

In Figure 16, the system demonstrates the successful display of the correct security image and phrase based on the email provided by the user.

```
1  function validateForm() {
2    if (!validateIC() || !validateUsername() || !validatePass() || !validatePassC() || !validateEmail() || !validatePhone()) {
3      submitError.style.display = 'block';
4      submitError.innerHTML = 'Please fix errors to submit';
5      setTimeout(function () {
6        submitError.style.display = 'none';
7      }, 5000);
8      return false;
9    }
10   checkRepeat();
11   function checkRepeat() {
12     var ic = document.getElementById('inputIC').value;
13     var email = document.getElementById('inputEmail').value;
14     var phone = document.getElementById('inputMobile').value;
15     $.ajax({
16       url: 'CheckRepeatServlet',
17       type: 'POST',
18       method: 'POST',
19       data: { ic: ic, email: email, phone: phone },
20       success: function (response) {
21         var check = response;
22
23         if (check == 'repeat') {
24           submitError.style.display = 'block';
25           submitError.innerHTML = 'This account is already registered. (Identity card number/Mobile number/Email has been used)';
26           return false;
27         }
28         showModal();
29       },
30       error: function (xhr, status, error) {
31         console.error(error);
32       }
33     });
34   }
35   return false;
36 }
```

Figure 17: Code to Validate Register Form.

The code in Figure 17 validates entered values using validation functions in line 2. If incorrect formats are detected, the registration process is rejected with an error message. If all values have the correct format, the code checks for duplicate entries in the database. If any duplicates are found, an error message is displayed. Otherwise, an One-Time Password modal dialog is shown to verify the user's email before proceeding with registration.

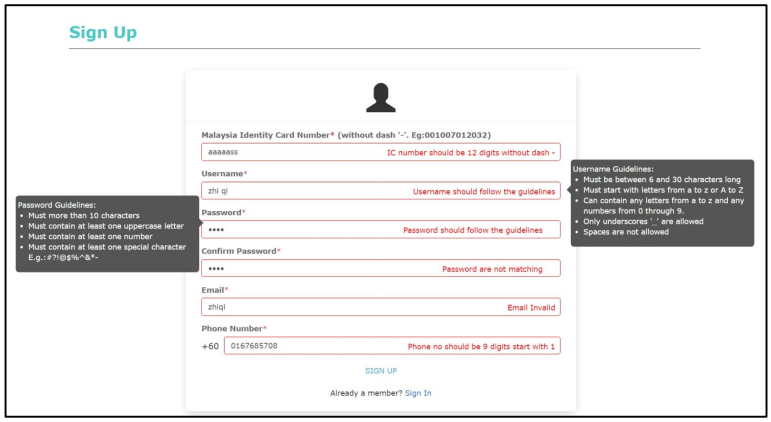

Figure 18: Input Validation in Register Form

Figure 18 demonstrates input validation in the register form, which displays error messages for incorrect data formats and prevents form submission until errors are resolved.

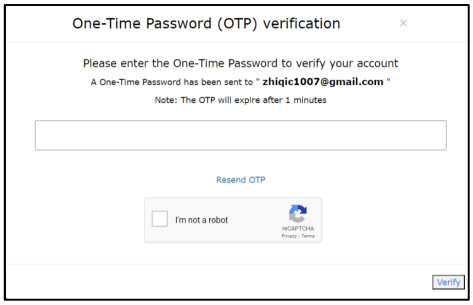

Figure 19: One-Time Password Verification Modal

In Figure 19, the one-time password (OTP) modal is presented as an additional layer of authentication to verify the user's email account. This modal is utilized in various processes such as registration, login, password recovery, changing email addresses, and changing passwords.

```
function validateOTPforLogin(){
    var recaptchaResponse = grecaptcha.getResponse();
    var inputOTP = document.getElementById("otp").value;
    var checkotp = giveOtp.innerHTML;
    var checkTime = new Date(otpTime.innerHTML);
    var inputTime = new Date();
    var timeDifference = inputTime.getTime() - checkTime.getTime();
    var minutesDifference = Math.floor(timeDifference / 60000);
    console.log(inputOTP);
    console.log(inputTime);
    console.log(checkotp);
    console.log(checkTime);
    console.log(minutesDifference);
    if(inputOTP != checkotp || minutesDifference >= 1){
        otpError.style.display = 'block';
        otpError.innerHTML = 'One-Time Password is wrong';
        setTimeout(function () {
          otpError.style.display = 'none';
        }, 5000);
        return false;
    }
    if(recaptchaResponse.length == 0){
        recaptchaError.innerHTML = 'Please complete the reCaptcha verification.';
        return false;
    }
    recaptchaError.innerHTML = ' ';
     document.getElementById("loginForm").submit();
}
```

Figure 20: Code to Validate the One-Time Password

Figure 20 demonstrates the OTP validation function, which verifies the accuracy and validity of the entered One-Time Password (OTP). If the OTP is incorrect or has expired, an error message is displayed.

4.2 Implementation of Module

Figure 21 showcases the bid page for a specific item, offering details like the current prices and remaining auction time. Once a bidder places a bid on the item by entering a bid price, the data is passed to the bid servlet for further processing.

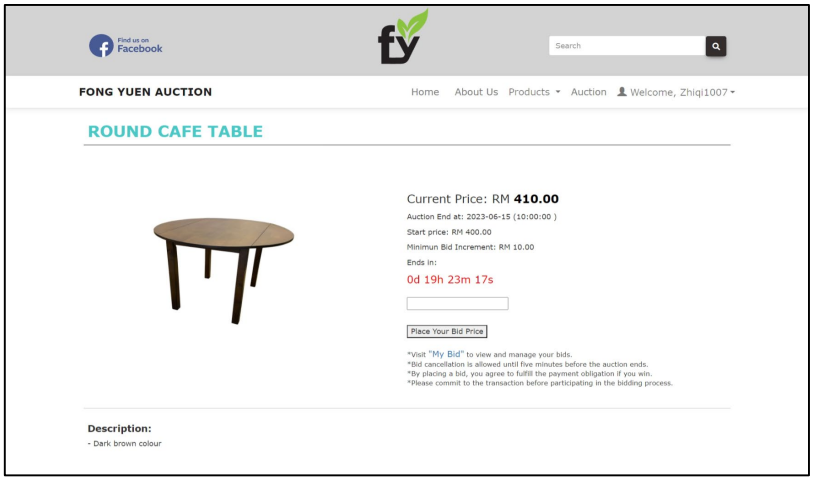

Figure 21: Bid Page

```
try {
    PreparedStatement check = conn.prepareStatement("select * from bid where bidderId=? and itemId=?");
    check.setString(1, bidderId);
    check.setInt(2, itemId);
    ResultSet rs = check.executeQuery();
    if (rs.next()) {
        PreparedStatement highest = conn.prepareStatement("select case when b.bidderId=? and b.bidAmount=m.max_bid then 1 else 0"
                                                          + " end as is_highest"
                                                          + " from bid b"
                                                          + " inner join (select max(bidAmount) as max_bid"
                                                          + " from bid where itemId = ?) m on b.bidAmount = m.max_bid"
                                                          + " where b.itemId = ?"
                                                          + " and b.bidAmount = (select max(bidAmount) from bid where itemId = ?)");
        highest.setString(1, bidderId);
        highest.setInt(2, itemId);
        highest.setInt(3, itemId);
        highest.setInt(4, itemId);
        ResultSet checkHighest = highest.executeQuery();
        if(checkHighest.next()) {
            int high = checkHighest.getInt("is_highest");
            if(high != 0) {
                    request.getSession().setAttribute("message", "highest");
                    response.sendRedirect("bid.jsp?id="+itemId+"");
            }else {
                    PreparedStatement pstbid = conn.prepareStatement("update bid set bidAmount=?,bid_at=? where bidderId=? and itemId=?");
                    pstbid.setBigDecimal(1, bidAmo);
                    pstbid.setString(2, bidTime);
                    pstbid.setString(3, bidderId);
                    pstbid.setInt(4, itemId);
                    int rowCountBid = pstbid.executeUpdate();
                    if(rowCountBid > 0) {
                        request.getSession().setAttribute("message", "bidSuccess");
                        response.sendRedirect("bid.jsp?id="+itemId+"");
                    }else {
                        request.getSession().setAttribute("message", "bidUnsuccess");
                        response.sendRedirect("bid.jsp?id="+itemId+"");
                    }
            }
        }else {
            request.getSession().setAttribute("message", "bidUnsuccess");
            response.sendRedirect("bid.jsp?id="+itemId+"");
        }
    }else {
        PreparedStatement pst = conn.prepareStatement("insert into bid(itemId,bidderId,bidAmount) values(?,?,?)");
        pst.setInt(1, itemId);
        pst.setString(2, bidderId);
        pst.setBigDecimal(3, bidAmo);
        int rowCount = pst.executeUpdate();
        if(rowCount > 0) {
            request.getSession().setAttribute("message", "bidSuccess");
            response.sendRedirect("bid.jsp?id="+itemId+"");
        }else {
            request.getSession().setAttribute("message", "bidUnsuccess");
            response.sendRedirect("bid.jsp?id="+itemId+"");
        }
    }
}
```

Figure 22: Bid Servlet

Figure 22 represents the Bid Servlet, which is utilized by bidders to place bids on items. The servlet retrieves the item ID and user ID, which are used to validate and process the bid. The servlet begins by retrieving the item ID and user ID from the bidding user. Using the item ID and user ID, the servlet executes a query in line 2 to check if the bid data already exists in the bid table of the database. This is done to determine if the bidder has already placed a bid on the item. If the bid data exists, the servlet proceeds to check if the bidder is the highest bidder for the item. This is achieved by executing a query in lines 7 to 13, which compares the bid amount with the current highest bid amount for the item. If the bidder is determined to be the highest bidder, a message is displayed to inform them that they are currently the highest bidder and cannot place a higher bid until another bidder outbids them. If the bidder is not the highest bidder, the servlet updates the bid amount for the item in the bid table by executing a query in line 25. This ensures that the bid amount is correctly updated and recorded. If the user ID and item ID do not exist in the bid table, indicating that the bidder has not placed a bid on the item before, the servlet inserts the bid data into the bid table by executing a query in line 44.

```
1  try {
2      PreparedStatement check = conn.prepareStatement("select b.bidId,b.bidderId, b.bidAmount"
3                                                      + " from bid b"
4                                                      + " inner join("
5                                                      + " select itemId, MAX(bidAmount) as maxBidAmount"
6                                                      + " from bid"
7                                                      + " where itemId = ?"
8                                                      + " group by itemId"
9                                                      + " ) m on b.itemId = m. itemId and b.bidAmount = m.maxBidAmount"
10                                                     + " where b.itemId = ?");
11     check.setInt(1, itemId);
12     check.setInt(2, itemId);
13     rs = check.executeQuery();
14     if(rs.next()) {
15         String highest = rs.getString("bidderId");
16         if(highest.equals(userId)) {
17             PreparedStatement pay = conn.prepareStatement("select * from payment where bidId=?");
18             pay.setInt(1, bidId);
19             payrs = pay.executeQuery();
20             if(!payrs.next()) {
21                 PreparedStatement get = conn.prepareStatement("insert into payment(bidId,status) values(?,?)");
22                 get.setInt(1, bidId);
23                 get.setString(2, "Pending");
24                 get.executeUpdate();
25                 String winStatus = "Winning";
26                 response.setContentType("text/plain");
27                 response.setCharacterEncoding("UTF-8");
28                 response.getWriter().write(winStatus);
29             }else {
30                 String status = payrs.getString("status");
31                 if(status.equals("Done")) {
32                     String doneStatus = "Win";
33                     response.setContentType("text/plain");
34                     response.setCharacterEncoding("UTF-8");
35                     response.getWriter().write(doneStatus);
36                 }else {
37                     String winStatus = "Winning";
38                     response.setContentType("text/plain");

39                      response.setCharacterEncoding("UTF-8");
40                      response.getWriter().write(winStatus);
41                  }
42              }
43         }else {
44             String lostStatus = "Lost";
45             response.setContentType("text/plain");
46             response.setCharacterEncoding("UTF-8");
47             response.getWriter().write(lostStatus);
48         }
49     }else {
50         response.sendError(HttpServletResponse.SC_NOT_FOUND);
51     }
52 }
```

Figure 23: Get Winner Servlet

Figure 23 represents the Get Winner Servlet, which is used to determine the highest bidder for an item when the auction end date and time are reached. The servlet performs the necessary queries to identify the winner and update the bidding status accordingly. The servlet begins by executing a query in lines 2 to 10 using the retrieved item ID. This query is used to obtain the highest bidder ID for the item. If the retrieved user ID matches the highest bidder ID, it indicates that the bidder has won the auction. In this case, the servlet executes the query in line 21 to insert the relevant data into the payment table of the database. Additionally, the response is set to "win" to indicate the winning status to the bidder. If the retrieved user ID does not match the highest bidder ID, it signifies that the bidder has lost the auction. The response is set to "lost" and the bidding status is displayed accordingly for the bidder with the corresponding user id.

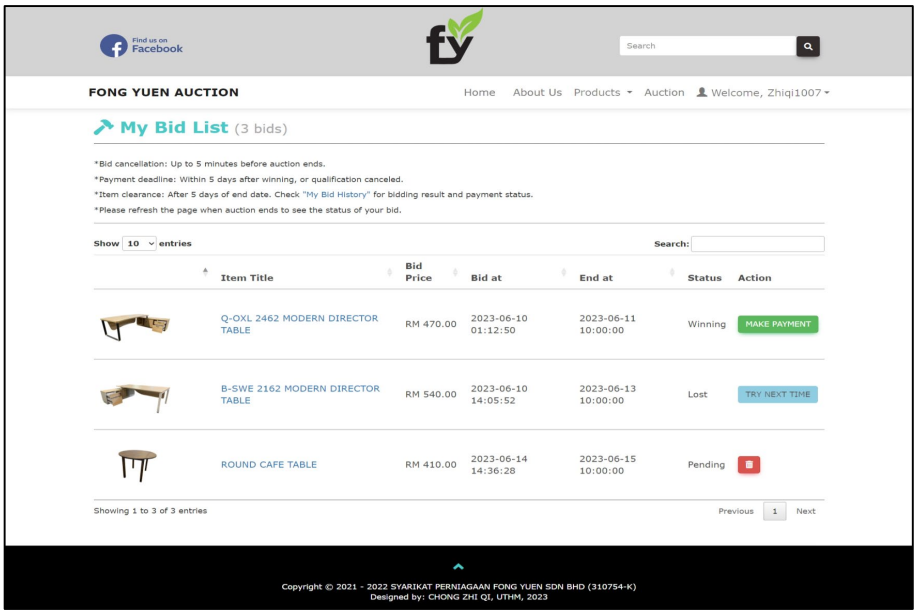

Figure 24: Bid Status According User's Account

In Figure 24, the bid status list presents bidders with an organized table displaying their bids and the associated bid status. The status is indicated as "Pending" for ongoing auctions, "Winning" for auctions won, and "Lost" for auctions lost. If a bidder wins an auction, a "Make Payment" option is provided to facilitate the payment process and finalize the bidding transaction.

```
1  try {
2      PreparedStatement pst = conn.prepareStatement("update payment set line1=?,line2=?,city=?,state=?,postcode=? where bidId=?");
3      pst.setString(1, line1);
4      pst.setString(2, line2);
5      pst.setString(3, city);
6      pst.setString(4, state);
7      pst.setInt(5, postcode);
8      pst.setInt(6, bidId);
9
10     int rowCount = pst.executeUpdate();
11     if(rowCount > 0) {
12         Billplz billplz = new Billplz();
13         BigDecimal total = bidAmount.multiply(new BigDecimal(100));
14         String amount = total.toString();
15         String urlBill = billplz.addBill(winnerEmail, winnerName, winnerMobile, amount, bidId);
16         response.sendRedirect(urlBill);
17     }else {
18         request.getSession().setAttribute("message", "saveUnsuccess");
19         response.sendRedirect("GetAddressReceipt?bidId="+bidId);
20     }
21 }
```

Figure 25: Payment Servlet

In Figure 25, the Payment Servlet is shown, which handles the updating of shipping addresses in the payment table and the generation of a bill in the Billplz payment gateway. The servlet is triggered when the highest bidder provides their shipping address and clicks the "PAY" button. It updates the payment table with the shipping address and then generates a bill in the Billplz payment gateway for the bidder to complete the payment process. Figure 26 displays the "Make Payment" page, where the winning bidder is required to enter their shipping address before proceeding with the payment. Figure 27 depicts the bill page, where the auction winner can choose their preferred bank for the online transfer to complete the payment process.

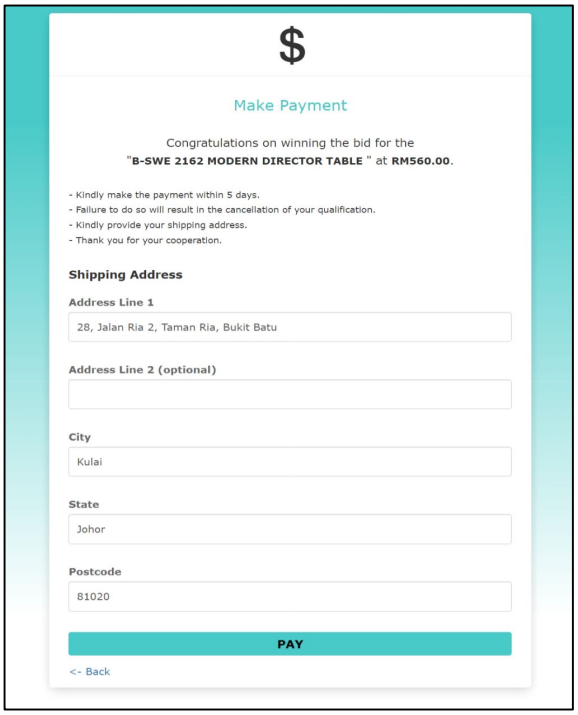

Figure 26: Make Payment Page

Figure 27: Bill Page for Winner Make Payment

4.3 Testing Result

After the implementation of the module, the system underwent functionality testing and user acceptance testing to evaluate its performance. Functionality testing verified the system's functionality and security, ensuring it met requirements and operated as intended. User acceptance testing assessed the system from

the perspective of end users, confirming that it met their requirements, expectations, and usability standards.

To evaluate functionality, three test categories were conducted which are user information storage and manipulation, auction product data handling, and bidding list data management. These tests comprehensively analyzed the system's performance and its adherence to predefined requirements. A security checklist was also used to assess the implemented security measures.

The results of these tests demonstrated that the proposed system successfully passed all functionality tests and security checks. This indicates that the system aligns with the specified requirements, functions as intended, and effectively handles user information, auction products, and bidding lists.

Furthermore, the user acceptance test conducted by the employees of Syarikat Perniagaan Fong Yuen Sdn. Bhd. confirmed that the proposed system met their requirements and satisfaction. The feedback received indicated that the system fulfilled their needs, met their expectations, and they were satisfied with its performance and usability. This provided validation of the system's effectiveness and suitability for the organization.

5. Conclusion

The project successfully accomplished its objectives by developing a web-based auction system for Syarikat Perniagaan Fong Yuen Sdn. Bhd. The system underwent thorough testing and demonstrated its functional capabilities. It provides a secure platform for online auctions, allowing the company to showcase products and customers to participate in bidding activities.

Key contributions of the project include the implementation of dual authentication for enhanced security, access control mechanisms for regulating user privileges, comprehensive input validation techniques to mitigate risks, AES encryption for password storage, and a CAPTCHA feature to prevent DDoS attacks. These security measures ensure the integrity, confidentiality, and availability of the system.

However, there are limitations to the project. The payment gateway is currently in test mode and lacks necessary document verification, limiting its ability to handle actual financial transactions. The system only supports email OTP verification, and there is no notification system to inform users about auction results. Additionally, the system's design lacks responsiveness, leading to display issues on different devices.

Future improvements can be made to address these limitations. Completing the document verification process for the payment gateway, incorporating alternative OTP verification options, implementing a notification system for auction outcomes, and optimizing the design for responsiveness will enhance the system's functionality, user experience, and security.

In conclusion, the project has successfully developed a secure and feature-rich web-based auction system. The web-based auction system significantly benefits Syarikat Perniagaan Fong Yuen Sdn. Bhd by expanding its reach, improving customer engagement, creating a competitive advantage, generating revenue, optimizing inventory management, providing customer insights, and streamlining the sales process. Future improvements will further enhance its capabilities and address the identified limitations, providing a more robust and user-friendly platform for online auctions.

Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

References

[1] Khan, S. (2022). *Advanced and Secure Online Web-Based Auction System An Efficient Android-Based Application and Tool Development to Trace Smartphones.* View project Simulation of 3-Phase 17-level Inverter Using two Cascaded Square Wave Bridge View project Advanced and Secure Online Web-Based Auction System. International Journal of Computer. Available: http://ijcjournal.org/. [Accessed Oct 30, 2022]

[2] Dissanayake, N. R., & Dias, G. K. A. (2018). *Rich Web-based Applications: An Umbrella Term with a Definition and Taxonomies for Development Techniques and Technologies.* International Journal of Future Computer and Communication, 7(1), 14–20. doi: 10.18178/ijfcc.2018.7.1.513

[3] Kabanda, S., Tanner, M., & Kent, C. (2018). *Exploring SME cybersecurity practices in developing countries. Journal of Organizational Computing and Electronic Commerce*, 28(3), 269–282. doi: 10.1080/10919392.2018.1484598

[4] Yehya, D., & Joudi, M. (2020). *AES Encryption: Study & Evaluation Operating Systems View project Cryptography & Network Security Papers View project.* Available: https://www.researchgate.net/publication/346446212. [Accessed 10 Nov, 2022]

[5] Hatzivasilis, G. (2020). *Password management: How secure is your login process? Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 12512 LNCS, 157–177*. doi: 10.1007/978-3-030-62433-0_10

[6] Lee, J., & Bauer, L. (2018). *Studying the Effectiveness of Security Images in Internet Banking*. Available: http://www.ancbank.com?id=15213. [Accessed 11 Nov, 2022]

[7] Shirvanian, M., & Agrawal, S. (2021). *2D-2FA: A New Dimension in Two-Factor Authentication.* Available: http://arxiv.org/abs/2110.15872. [Accessed 20 Nov, 2022]

[8] Jabri, M. (2021). *Java Programming Language Report*. doi: 13140/RG.2.2.33652.48005

[9] Gao, Y., Gao, H., Luo, S., Zi, Y., Zhang, S., Mao, W., Wang, P., Shen, Y., & Yan, J. (2021). *Research on the Security of Visual Reasoning CAPTCHA. USENIX Security Symposium.* Available: https://www.usenix.org/system/files/sec21fall-gao.pdf. [Accessed 10 Dec, 2022]

[10] Matthew Martin. (2022). *Prototype Model in Software Engineering*. Available: https://www.guru99.com/software-engineering-prototyping-model.html#:~:text=Prototyping%20Model%20is%20a%20software,are%20not%20known%20in%20detail. [Accessed 15 Dec, 2022]

[11] Régis, B., & Marie, B. (2021). *The three stages of an auction: how do the bid dynamics influence auction prices? Evidence from live art auctions*. Available: http://ifs.u-strasbg.fr/large/publications/2021/2021-10.pdf. [Accessed on 20 Nov, 2022]