Applied Information Technology And Computer Science Vol. 5 No. 1 (2024) 163-181

AITCS

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

Web Based Locker Booking System with Multifactor Authentication for Wajasakti Sdn Bhd

Radin Nur Haziqah Radin Abdul Aziz[1], Khairul Amin Mohamad Sukri[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.2024.05.01.010
Received 24 June 2023; Accepted 26 May 2024; Available online 30 August 2024

Abstract: E-Locker Booking System with Multifactor Authentication for Wajasakti Sdn Bhd is a system that allow their employee to book a locker to keep their belonging safe which located in their workspace. Since they only use logbooks to keep track of all the information, recordings occasionally stop existing and disappears and having several disorganised records which give time-consuming to search down the data of employees. This paper proposes a locker booking system that enables workers to reserve lockers while assisting administrators in managing reservations and workers' data. The Agile technique is being used for this project methodology to develop systems to be more effective because all needs and deliverables are better understood. As a result, the development of the locker booking system has been completed successfully by using HTML, CSS, JavaScript and PHP languages. The system has nine modules which include Login Module, Register Module, Booking Locker Module, Manage Booking Record Module, Manage Locker Module, Manage Admin Module, Manage User Module, Report Analysis Module, and Activity Log Module. The project is successful with that the security features of the system would be recognised for their efficacy. Thus, this system give effective management to the admin as it makes it possible for administrators to monitor locker availability, usage, and demand-based locker distribution. This reduces the possibility of conflict or double bookings and guarantees effective locker utilisation

Keywords: Multifactor Authentication, Booking System, E-Locker, One-Time Password (OTP), Email

1. Introduction

Wajasakti Sdn Bhd provides taxi and public transportation services, and its customers use them. They have supplied a locker for the workers to use the facilities there because of their vigorous physical activity. There are several lockers available for them that provide workers with an amazing variety of beneficial benefits. They can put their belongings such their supplies, backpack, wallet, and other items

*Corresponding author: khairulm@uthm.edu.my

in those practical locations while keeping them safe. In addition, it provides a key for the worker to lock and unlock the door, but the most crucial part is that they must guard the key carefully and hard to track.

This locker reservation system is essential since it might assist in the management of the production system. The employees and Wajasakti Sdn Bhd itself have also encountered certain issues. Among them include the fact that the department finds it difficult to monitor employees' data since they constantly use the locker without asking, waste employees' time, and keep a sizable number of disorganised records. Since they only use logbooks to record all the data, records periodically stop being retained and vanish. Name, IC number, phone number, date, and time are all included in the record. As a result, it is unable to maintain the data securely, making it accessible to the public or non-confidential.

The objectives of E-Locker Booking System with Multifactor Authentication for Wajasakti Sdn Bhd are to design a suitable booking system for workers and admin of Wajasakti Sdn Bhd , to develop and design an online booking system that can help worker to book, admin to insert, update and delete and also to produce a medium to facilitate Wajasakti Sdn Bhd on managing the services to be more systematic and to test the effectiveness level of the security of developed system to the worker’s data and also the security of admin sites.

This project divided the scope in several parts; the study domain is related to Wajasakti’s worker and admin only. It was developed using JavaScript, PHP and CSS language and for database using MySQL. This project focused on making booking and to manage the booking record to be more systematic with security applied (Multifactor authentication using one-time password). This study's findings will be helpful to both workers and Wajasakti.

The significance of this experiment is to demonstrate how crucial high-end booking security is to modern technology and advancement. The increasing demand for system development security serves as evidence of the need for better, more efficient management practices [1]. As was already said, one-time password (OTP) schemes are a useful tool that bridge usability and security holes and protect against known security attacks [2].

2. Related Work

2.1 Multifactor Authentication

"Authentication" is the use of one or more techniques to prove that you are who you claim you are. Access is authorised once the identity of the person or device has been confirmed. The most common type of authentication is one factor, however two factor authentication, which combines both aforementioned features, is regarded to be more secure and reliable. A user's identification and verification by the computer are controlled by an authentication system. The primary objective of an authentication system is to confirm the user's identity, or that they are who they claim to be [3].

2.2 Session Timeout and Session Destroy

Session timeout occurs when a user remains idle on a website for a predefined amount of time (determined by a web server). The event ends the user session, alerts the web server to end it, and sets the session's status to invalid (not used anymore). It notifies the web server to delete all the data it contains. Reducing the session timeout as much as feasible is a best practice for user session security in order to decrease the possibility that an attacker will be able to access the user account. By requiring sessions to expire after a predetermined amount of inactivity, users can restrict session lengths and improve security [4]. Thus, for this locker booking system expire the PHP session after 15 minutes.

2.3 Encryption (Salt Cryptography and Hashing)

To solve the challenge of protecting encrypted passwords, this system will use encryption using the Salt algorithm technique, which is composed of random bits added to each password instance prior to its

hashing. Salts produce unique passwords even when two users choose the same passwords. Salts help to mitigate hash table attacks by requiring attackers to recompute hash table attacks using the salts for each user. Not the original password, but the salt and the resulting hash value are then kept in a database. If the authentication data store is breached, hashing enables later authentication without storing or potentially disclosing the plaintext password [5]. There is no need to encrypt or store salts separately from the hashed password because they can withstand conventional attacks even if an attacker has access to the database containing the hash results and the salts.

2.4 Time-Based One-Time Password (Static Password)

OTPs do away with several the drawbacks of conventional (static) passwords. The biggest problem that OTPs solve is, unlike static passwords, they are not susceptible to replay attacks. Systems are safer with one-time password authentication than reusable password authentication. For instance, in order to gain remote access, the user normally needs to submit a password or passphrase. These data are typically sent across insecure networks without encryption. Because a password loses value after being used, one-time passwords reduce the probability of eavesdropping in. When implemented effectively, one-time password techniques are difficult for most attackers to guess and require active.

2.4 Activity Log for Security

Security event logging and monitoring is a methodology that most organizations use to examine system audit logs for indications that unauthorized security-related operations have been attempted or performed on a system or application that processes, transmits, or saves sensitive data. With the aid of security event logging and monitoring services, admin may quickly and affordably act on suspicious data from system and audit logs, leaving only the information that the organization needs to review and preserve [6]. Businesses may protect sensitive information with efficient logging and monitoring while also spotting critical flaws in their security management systems. By boosting productivity, resolving process problems, and optimizing employee time management, activity log analysis may optimize the amount of time spent on improving the user experience.

2.4 reCAPTCHA

The proposed reCAPTCHA controller approach stops automated attacks. The bulk of automated DDoS attempts are checked and rejected by the reCAPTCHA controller. Using an information theory-based metric, this technique is used to examine the variation in user requests in terms of unpredictability [7]. For the purpose on helping to protect websites from spam and abuse, Google provides reCAPTCHA as a free service. A "CAPTCHA" is a test that determines if a user is human or a robot. People can understand it easily, while "bots" and other dangerous software find it difficult to use.

2.4 Email Authentication

Multi-Factor Authentication (MFA), an extra security precaution, will help to secure any account. To enable multi-factor authentication, it will require a different device, such as personal phone and email, to identify and verify our identity. As a result, there is less chance that personal account will be compromised, and the data will always remain protected. It is also providing an additional layer of protection to the email address and password. If two-factor authentication is enabled, even if someone knows your password, they cannot access your account without your permission. Thus, when the employee registers for the account using their email, the interface design will be linked to it. A One-Time Password (OTP) that can only be used for a brief period will be sent to the email to maintain the security level. Simple Mail Transfer Protocol (SMTP) Server will be used to send the One- Time Password (OTP) through worker's email by using PHP Mailer.

2.2 Existing System

A mobile application called E-Locker Book Secure Shared Bicycle Parking (system 1) was created to help communities manage their first-mile or last-mile public transportation difficulties [8]. Their target

audience includes people who enjoy driving, biking, and using public transportation. A variety of functions, including log in and registration, dashboard, locations, history, billing, help, and logout, will be included in the all-inclusive software. This project's major focus will be on the problem of renting a shared e-Locker by scheduling it 24 hours in advance using the "Dashboard" features. They want to make the e-Locker app a 24-hour notice for renting or booking the locker (choose the location, check for locker availability, pick the locker, choose the start and end periods, book the rental).

The second related project is Dr Locker, a locker or luggage storage facility in KL Sentral (system 2). For their target audience, which consists of customers who wish to keep their belongings safe, this Dr Locker system has operated a Luggage Storage or Locker [9]. Only two times per transaction will the locker door open, once to let the customer deposit their goods in the locker and once to retrieve them. Customers can start by going to the control panel and choosing "Deposit" in their selected language. The open, vacant locker that is now available on the screen must be selected as the next action. Lockers with red markings are occupied; those with green labels are free.

The third related project is Radical Storage (system 3), an application that allows simple online booking for handy luggage storage choices in several locations across the world [10]. After a successful reservation, the user will receive the address, allowing them to conveniently drop off their bags at the appointed time. Unlike airlines, Radical Storage does not increase the price for larger or heavier items, so you can prolong your stay for an additional seven days at the same affordable fee. Users must first register to utilize the services provided by this application or system. Their user must register using their email address to do this.

2.2 Comparison of the Existing System with the proposed System

Table 1 compares three current systems and the one that is being suggested. Its goal is to make sure that the suggested system can be enhanced or that the features described before can be implemented. Their current manual system has undergone certain modifications because of the existence of this E-Locker Booking process. Due to their different scopes, the systems also have various limitations.

Table 1 : Comparison of the Existing System with the proposed System

Functions	E-Locker Book Secure Shared Bicycle Parking	KL Sentral	Radical Storage	Proposed System
Platform	Application	System built and built-in camera	Application and Web base	Web base and E-mail
Dual authentication (Captcha security)	Not Available	Not Available	Not Available	Available
Multi-Factor used	Not Available	Face recognition technology (camera) and thumbprint	Not Available	One-Time Password (OTP) through E-mail
User Profile Update	Available	Not Available	Available	Available
Review of Booking data (date and time or payment)	Available	Not Available	Available	Available
List the Damage Locker	Not Available	Not Available	Not Available	Available
Showing Locker Availability	Yes	Yes	Yes	Yes
Showing Report Analysis of Locker used	No	No	No	Yes
Database	MySQL	MySQL	MySQL	MySQL
Activity Log	Not Mentioned	Not Mentioned	Not Mentioned	Yes

3. Methodology

Methodology examines the essential project management procedures and offers an overview of several project management [11]. The methodology used in this project is Agile model. In Agile model, it involves incremental and iterative software development techniques [12]. Agile can be characterised as a software development paradigm where a prototype is created, tested, and improved as necessary until a workable version is reached. Without particularly emphasizing long-term planning, agile project management strategies break work into smaller, more manageable iterations or components. The criteria and specifications for the project are determined before the development process even begins. Each repetition's quantity, duration, and scope are carefully thought out in advance. This prototype methodology comprises six phases, as shown in Figure 1, including requirements, design, develop, test and deploy the system.

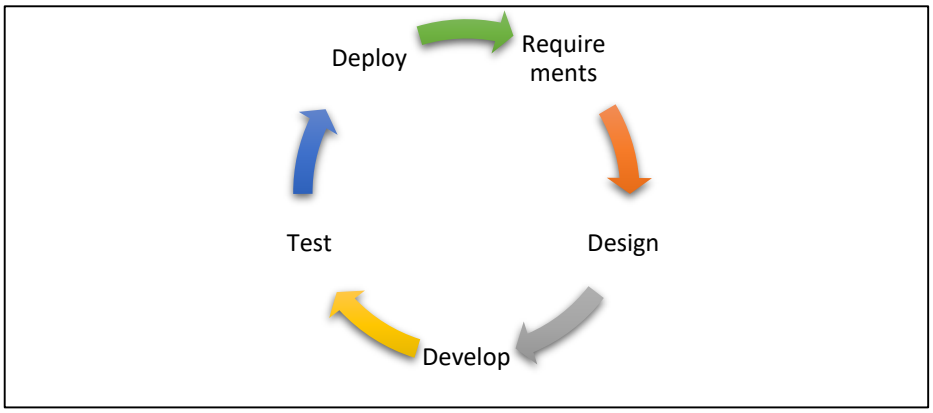


Figure 1 : Agile Methodology

3.1 Requirement Gathering and Analysis

The duration and itinerary of the system are planned and designed during this stage to make sure the system development process goes well and can be completed on time. The needs must now be made explicit. Outlining the project's potential (goals) and scheduling the time and resources required to complete it are essential. Based on these variables, the technical and financial viability can be assessed. A study of the problem statement regarding the lack of existing systems, identification of the system's objectives, and user scope are all under progress. At this stage, the input from the previous release is also taken into the project, and substantial improvements are categorized as new requirements.

Gathering requirements will also be a part of this phase. The locker reservation system's functional and non-functional needs will be established. While the functional requirements of the proposed system address the features and functionalities of each module, the non-functional requirements address the performance, usability, operational, and security needs.

3.2 Design

At this point, the interface and database will be designed in order to develop the conceptualized system. The locker booking system will design Data Flow Diagram Context Diagram (DFD CD), Data Flow Diagram (DFD) level 0 with their processes. The six-process node from the Data Flow Diagram (DFD) Level 0 is divided into sub-processes which consists of Register, Login, Book Locker, Manage Booking, Add Admin and Manage Locker. The diagram will require more data flows and data stores when new processes are introduced to connect them. The DFD visually represents the functions, or processes, that capture, manipulate, store, and distribute data between a system and its environment, as well as between system components. Because of the visual representation, it is a useful communication tool between the user and the system designer. The database for the locker booking system will then be built using the

entity relationship diagram (ERD) and class dictionary. The test strategy and user interface design will both be made during the project's design phase.

Figure 2 shows Context Diagram for the E-Locker Booking System with Multi-factor Authentication for Wajasakti Sdn Bhd. In this diagram, there are two entities, those involved are workers and admin. It offers a broad overview of the entire system or process under study or being modelled for this locker booking system. In what is meant to be an overview view, the system is shown as a single, high-level process along with its connections to external entities.

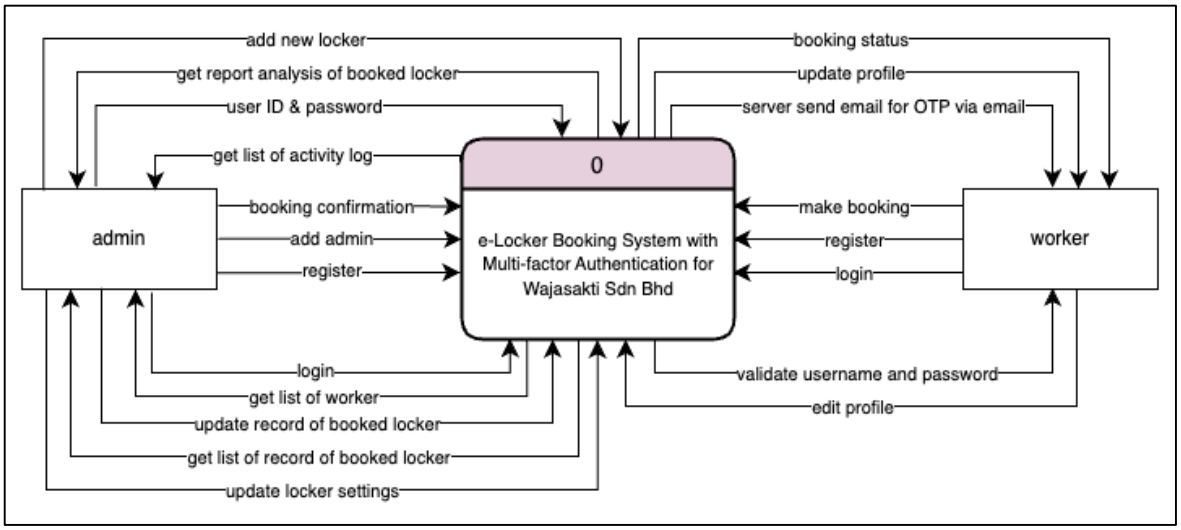


Figure 2 : Data Flow Diagram Context Diagram (DFD CD)

The Data Flow Diagram (DFD) level 0 shows in Appendix A is to illustrate delivery of data with greater thoroughness. Processes are displayed separately to give a more detailed and accurate insight. For the proposed system, there are six process which consists of Worker and Admin as entities while the data store consists of Worker, Admin, Record and Locker. First, worker need to register their account first before get access into the system. After getting verified user ID and password, user then can login into the system with the credentials and need to tick the reCAPCTHA first before get access to make booking for available locker displayed on the user homepage. However, user can request on reset password link which will be send to the registered email to change their password and required to login back with the new password. The book locker process 3.0 will let system shows the booking details which need to be fill before submitting the form to the admin webpage. Admin then can manage booking on process 4.0 by accepting it for confirmation. If the booking has been accepted, the booking status will automatically update on user webpage. Process 5.0 and 6.0 shows admin can manage admin setting and locker by adding, inserting, and deleting the data. On top of that all data will be save and update into the database.

The next design of this locker booking system is the system design. Figure 3 shows the architecture and the process for defining a system's modules, architecture, pieces, and data and interfaces. The worker or admin users can open the first user interface which is the “Login menu”. User also can use forgot password function to request email which redirect user to change their new password. For worker who not registered yet, they can do so on register page. This registering will connect PHPMailer with SMTP server to send One-Time Password to the registered email. The registration will include One-Time Password (OTP) which will be sending to their registered email as an authorization. The registering data then will be saved into database to let the user use their credentials to get access into the system. User also can request for forgot password so that the server will send the reset password’s link to their registered email. After get access, the worker can see homepage, book locker, edit their profile, and can see booked summary. While for admin, they can see dashboard, manage booking

record, locker, user, and admin setting also can see report analysis and activity log. All the data changes will be update into the database and synchronise with activity log module for better performance and to improve data security. Both users then can logout or the account will be automatically log out after 15 minutes of inactive.

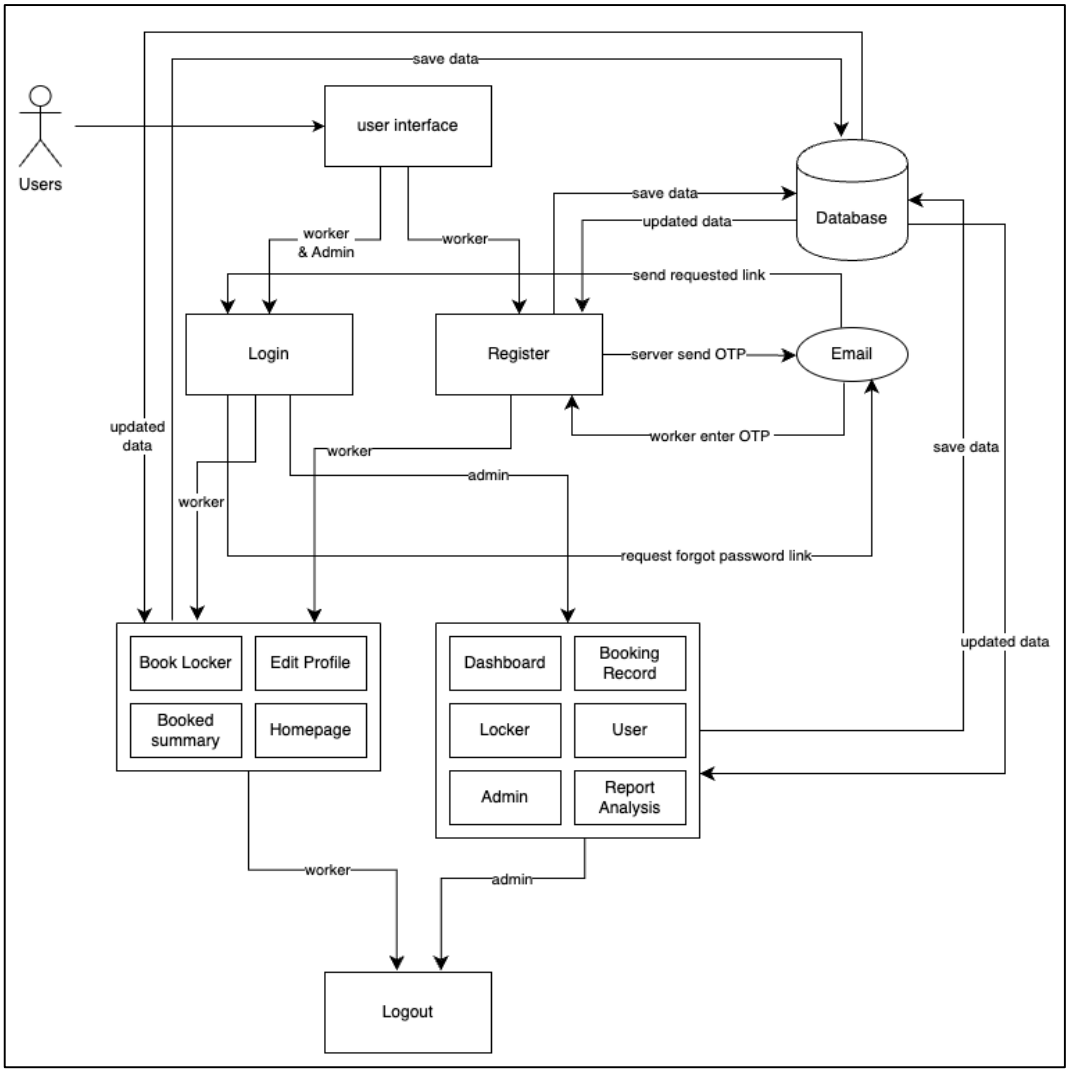


Figure 3 : Architecture and Flow of system

Databases are designed to connect the various entities, which aids in preserving the accuracy and integrity of the data throughout time. The four links between the entities involved worker, locker, admin, and log are depicted in Figure 4. Each entity has unique characteristics and cardinally. Many workers can book many available lockers and many available lockers can book by many workers. Therefore here, the associative entity which is the 'record' entity represents a relationship between two fundamental items which are worker and locker that may be many-to-many cardinally oriented or possess specific attributes. Other than that, one admin can approve many lockers and many admin account can view one list of logs. The Primary key (PK) and Foreign Key (FK) of record and log entity in this system are contained in the attribute. While relationships exemplify how different entities are interdependent on one another.

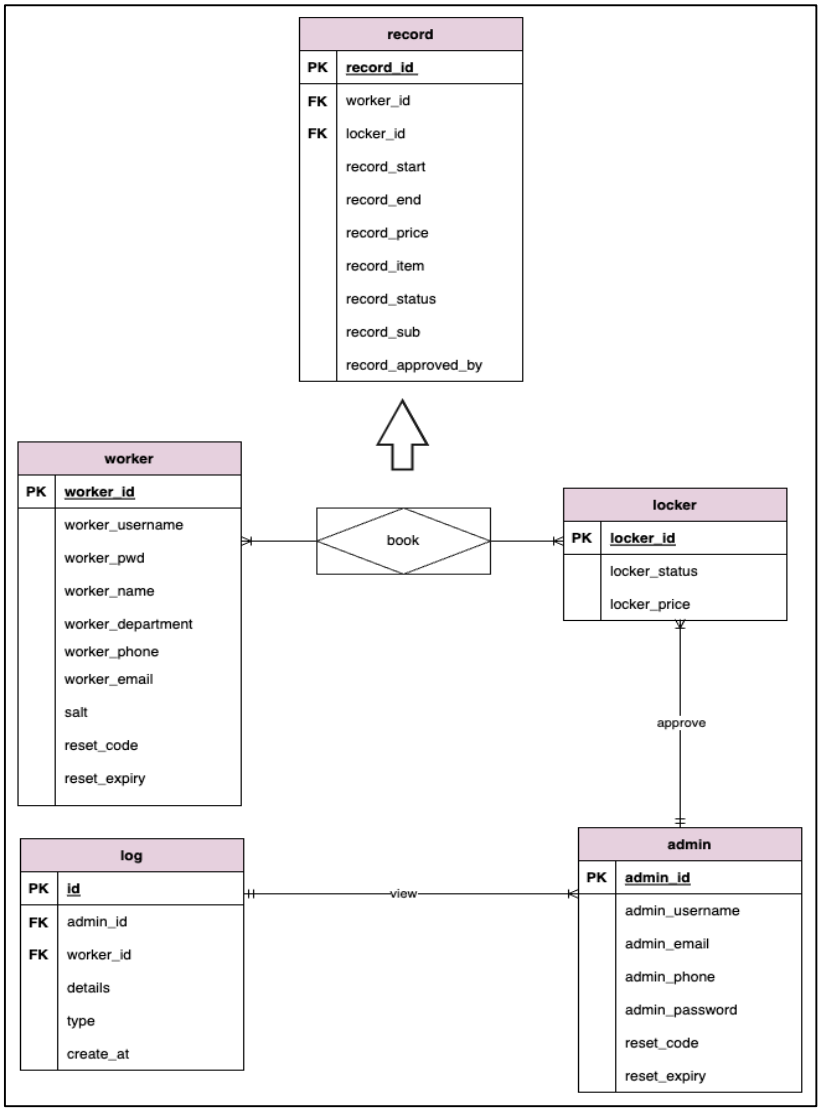


Figure 4 : ERD for E-Locker Booking System with Multifactor Authentication for Wajasakti Sdn Bhd

3.3 Develop

This stage of the e-locker booking system's real development is based on the system design's inputs. The coding and email verification phases will be put together in this phase to integrate the complete system based on the previously stated algorithms. In addition, each module, piece of code, and unit that will be created must be written and checked for errors in order for the e-Locker Booking System to operate properly between email authentications. Once the system has been designed in accordance with defined specifications, the system will begin to take shape during the design phase. During this stage, each system module that has been successfully integrated will be put together into a finished system.

3.3 Test

Throughout the testing process, the user will test the system to discover any potential supplementary requirements. When the system has been created, Wajasakti employees will initially test it. The compilation of all submitted ideas aims to improve this system. The administrator will then evaluate the system and offer the system developer with updated requirements. The system will be expanded to accommodate new requirements, after which users will rate it. Up till the system's testing phase is over, the phase will be repeated. To provide quality assurance, any flaws and problems detected during this step must be addressed.

3.3 Deploy

The main goal of the deployment phase is to successfully integrate the system created during the development phase before it is put into use by end users in a production environment. The fast delivery of post-live assistance throughout this phase ensures stability and quick problem-solving times. Several tasks can be completed at this phase, including making sure to do final tests on the production infrastructure, operating systems, and applications. These will guarantee that they are set up properly and on a secure website.

4. Results and Discussion

4.1 Results

In the results and discussion, the suggested system's implementation and testing results are discussed. The four security modules that are included in this application are Multifactor Authentication using One-Time Password through Email, reCAPTCHA, activity log, session destroy and password encryption. Each of these security modules is intended to reduce the security risks of the application. The implementation of the code is explained in the paragraphs that follow.

One- Time Password can be sent when a user clicks the ‘register’ button on register page, this section as shown in Figure 5 will appear where the user must enter the one-time password (OTP) sent to the registered email address in order to authenticate. The One-Time Password (OTP) given also has time limit to use which is only for 15 minutes. After user click on ‘Submit’ button, the system will automatically redirect user to login page so that they can access into the system by entering their username and password.

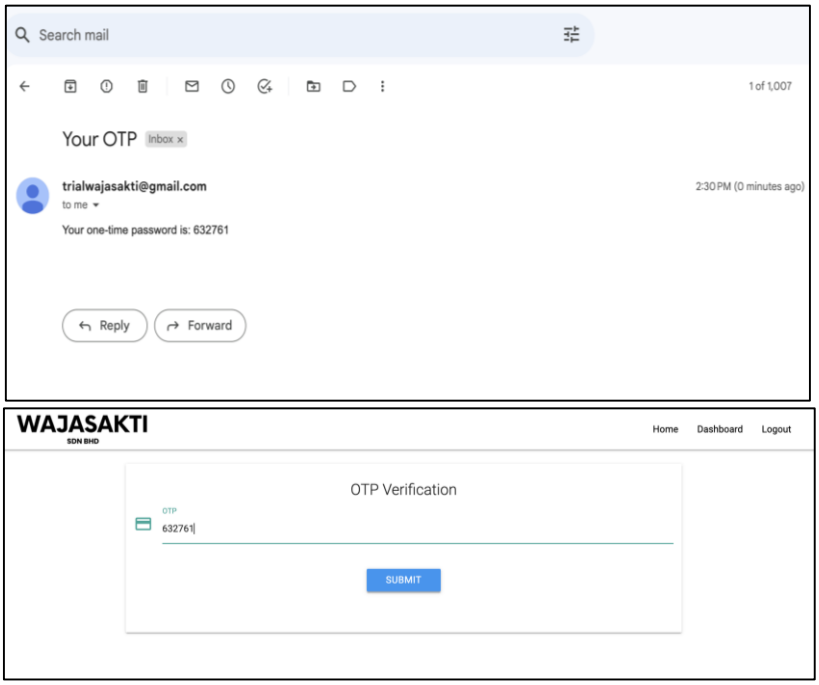


Figure 5 : OTP Verification

Figure 6 shows the segment code on connecting the PHPMailer with SMTP server. The TLS has been used to encrypting data sent between servers and web applications, such as when a web browser loads a website. It also uses to encrypt communications such email and messaging. Thus, this project must make sure to check the host, port, username, and password to avoid having the SMTP server reject credentials.

```php
else
{
    $mail = new PHPMailer;
    $mail->isSMTP();                                      // Set mailer to use SMTP
    $mail->Host = 'smtp.gmail.com';  // Specify main and backup SMTP servers
    $mail->Port = 587;                                    // TCP port to connect to
    $mail->SMTPAuth = true;                               // Enable SMTP authentication
    $mail->SMTPSecure = 'tls';                            // Enable TLS encryption, ssl also accepted
    // $mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS;
    $mail->SMTPDebug = '1';

    // $mail->Username = 'trialwajasakti@outlook.com';                 // SMTP username
    // $mail->Password = 'trail1010';                           // SMTP password

    $mail->Username = 'trialwajasakti@gmail.com';                 // SMTP username
    // $mail->Password = 'Trialwajasakti_';
    $mail->Password = 'actwzddtrwdxzpfv';


    $mail->setFrom('trialwajasakti@gmail.com');
    $mail->addAddress($email);

    $mail->Subject = 'Your OTP';
    $mail->Body = "Your one-time password is: $otp";

    if(!$mail->send())
    {
        echo 'Mailer Error: ' . $mail->ErrorInfo;
    }
    else
    {
        echo 'Message sent!';
        header('Location: verify-otp.php');
    }
}
}
```

Figure 6 : Code Segment for OTP Verification

By entering their username and password or authenticating on the login screen, users can access any system developed. There are two different login side interface that has been develop which can be categorized as user login and admin login. Figure 7 show login page for user side which is for worker. For user login, there is a reCAPTCHA that need to be tick by user before proceeding on accessing into the system. The reCAPTCHA tokens expire after two minutes of use. Because the reCAPTCHA verification has a time limit, it is best to complete it last on a website user are reading.

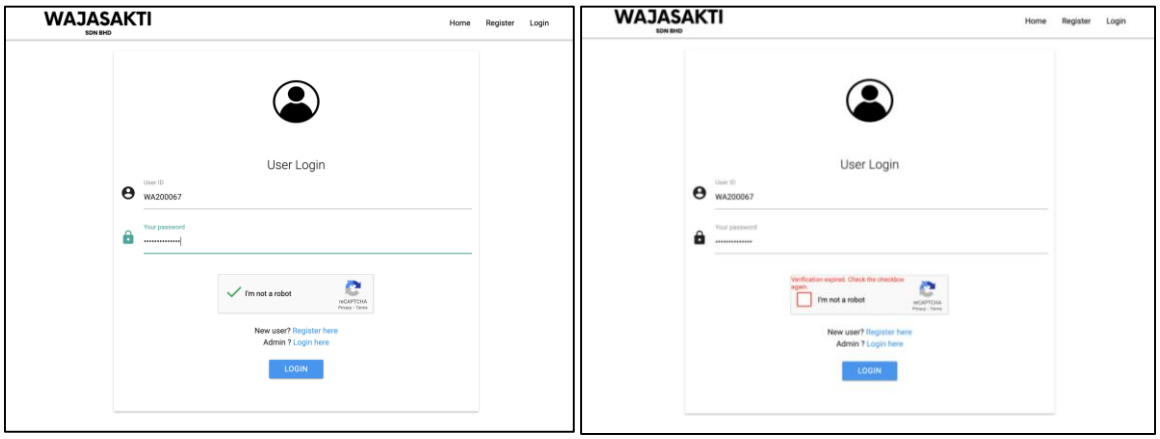


Figure 7 : reCAPCTHA in User Login

Figure 8 shows the code segment for login interface with complete validation. By using a username and password, you may verify a user's identity when they access a digital system. The user must input a username which is a unique identifier and a password which known as a secret to gain access. The system then cross-checks this information with its stored database to confirm the user's identification. The validation message will show if user enter not match username and password, not tick the reCAPTCHA and not insert any input.

```php
// Check if inputs are empty
if (!empty($id) && !empty($password)){
  // success
  $sql = "SELECT * FROM `worker` WHERE `worker_id`='$id'";
  $result = mysqli_query($conn, $sql);
  $resultCheck = mysqli_num_rows($result);
  $row = mysqli_fetch_assoc($result);

  if(isset($_POST['g-recaptcha-response']))
  {
    $captcha = $_POST['g-recaptcha-response'];
  }

  if(!$captcha)
  {
    $msg = "Please check the captcha form!";
    $msgClass = "red";
  }
  else
  {
    if ($resultCheck < 1)
    {
      // error, id not exist
      $msg = "Invalid user Id or password";
      $msgClass = "red";
    }

    else
    {
      // dehashing the password
      // $pwdCheck = password_verify($_POST['password'], $row['worker_pwd']);

      $password = $_POST['password'];
      $storedPassword = $row['worker_pwd'];
      $salt = $row['salt'];
      // Combine the provided password with the salt
      $combinedValue = $salt . $password;

      $pwdCheck =  password_verify($combinedValue, $storedPassword);

      if($pwdCheck == false)
      {
        $msg = "Invalid User Id or password";
        $msgClass = "red";
      }
```

Figure 8 : Code Segment for Login Authentication

This activity log shows description of all activity that worker and admin account activity which the analysis may optimise the amount of time spent on improving the user experience. The durations or start and end timings of user jobs and activities are also included as depict in Figure 9. Admin can choose category either Login, Update, Insert, Delete and Register to see the particular activity.

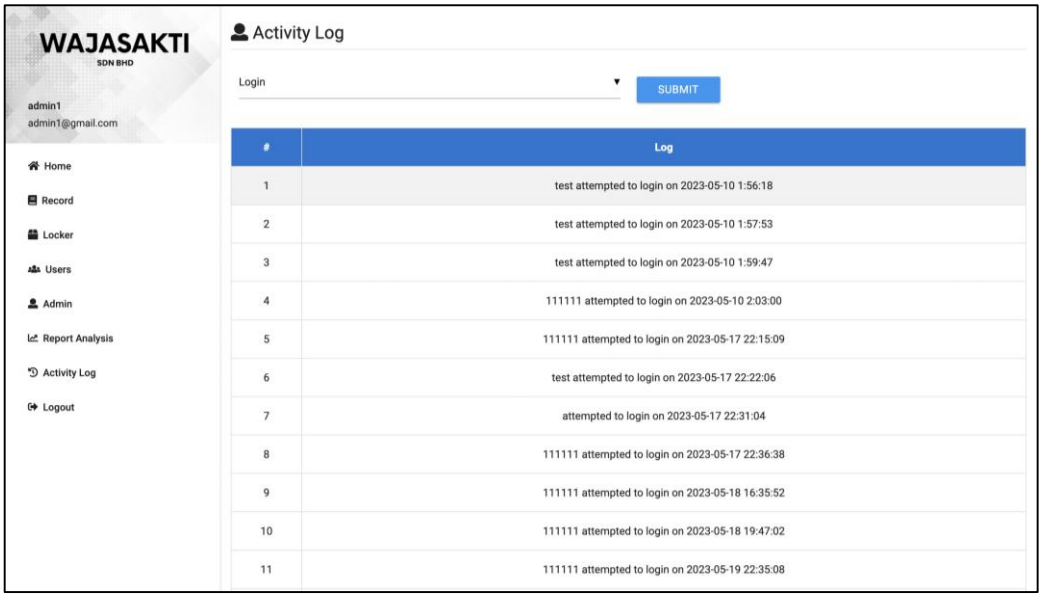


Figure 9 : Activity Log

Figure 10 shows code segment for log which the data get from all activity that has been done by user account. For selected category, the code has assigned ‘all’ as All, ‘1’ as Login, ‘2’ as Update, ‘3’ as Insert, ‘4’ as Delete and ‘5’ as Register and all of the category need to be same with the database.

The activity log gets from all modules develop in the system to be display in this page and sychronised with the database for better performance and to improve data security.

```
<h5><i class="fas fa-user"></i> Activity Log</h5>
<div class="divider"></div>
<br>
<form method="POST">
  <div class="row">
    <div class="col s12 m6">
      <select name="action">
          <option value="all" <?php if($selectedAction === 'all') echo 'selected'; ?>>All</option>
          <option value="1" <?php if($selectedAction === '1') echo 'selected'; ?>>Login</option>
          <option value="2" <?php if($selectedAction === '2') echo 'selected'; ?>>Update</option>
          <option value="3" <?php if($selectedAction === '3') echo 'selected'; ?>>Insert</option>
          <option value="4" <?php if($selectedAction === '4') echo 'selected'; ?>>Delete</option>
          <option value="5" <?php if($selectedAction === '5') echo 'selected'; ?>>Register</option>
          <option value="6" <?php if($selectedAction === '6') echo 'selected'; ?>>Booking</option>
      </select>
    </div>
    <div class="col s12 m6">
      <div class="input-field">
        <button type="submit" class="waves-effect waves-light btn blue" name="submit">Submit</button>
      </div>
    </div>
  </div>
</form>
<table id="myTable" class="responsive-table highlight centered">
  <thead class="blue darken-2 white-text">
    <tr class="myHead">
      <th>#</th>
      <th>Log</th>
    </tr>
  </thead>
  <tbody>
    <?php
      $i = 1;
      if (isset($result)) {
        while ($row = mysqli_fetch_array($result)):
    ?>
          <tr>
            <td><?php echo $i; $i++; ?></td>
            <?php if($row['admin_id']): ?>
              <td><?php echo $row['admin_id'] . ' ' . $row['details']. ' ' . $row['created_at']; ?></td>
            <?php else: ?>
              <td><?php echo $row['worker_id'] . ' ' . $row['details']. ' ' . $row['created_at']; ?></td>
            <?php endif; ?>
          </tr>
    <?php endwhile; } ?>
```

Figure 10 : Code Segment for Activity Log

As shown in Figure 11, the code in line 41 describes how the code gets from the user activity and will be save into log table in the database. The data of activity get from line 32 which from user session where it temporarily storing data and making it accessible across all webpages so that it will display the user data on log page. This code applicable to all modules which record the activity of user accounts.

```
    if ($resultCheck < 1) {
      // error, id not exist
      $msg = "Invalid Admin id or password";
      $msgClass = "red";
    } else {
      // dehashing the password
      $pwdCheck = password_verify($_POST['password'], $row['admin_password']);

      if($pwdCheck == false) {
        $msg = "Invalid password";
        $msgClass = "red";
      } elseif ($pwdCheck == true) {
        $_SESSION['admin_id'] = $row['admin_id'];
        $_SESSION['admin_uname'] = $row['admin_username'];
        $_SESSION['admin_email'] = $row['admin_email'];

        $admin_id = mysqli_real_escape_string($conn, $row['admin_id']);
        $text = "attempted to login on ";
        date_default_timezone_set('Asia/Kuala_Lumpur');
        $postingDate = date('Y-m-d G:i:s ', strtotime("now"));

        $sql = "INSERT INTO `log` (`admin_id`, `details`, `type`, `created_at`)
        VALUES ('$admin_id', '$text', '1','$postingDate')";

        if (mysqli_query($conn, $sql))
        {
          // Success
          $_SESSION['admin_user_type'] = 'admin';
          header("location: index.php");
        }
        else
        {
          $msg = "Something Went Wrong";
          $msgClass = "red";
        }
      }
    }
  } else {
    // failed ouput an error
    $msg = "Please fill in all fields";
    $msgClass = "red";
  }

  mysqli_close($conn);
}
```

Figure 11 : Code Segment on Getting Data and Log

A strong password should be made up of a mixture of 8 character which consists of lowercase, capital, numeric, and special characters when registering locker booking system's account. The

presence of the required characters in the password is checked using regular expressions (Regex). Encryption use in Figure 12 code segment is using hashing and salt function. A password is transformed into a hash of fixed-length characters using Bcrypt. Bcrypt adds the salt before hashing a password. The combine password with salt function will be stored into the database so that the encryption will work on it. Security of user credentials in a database depends on password encryption. Without password encryption, anyone with access to a user database on a company's servers, including hackers, might simply view any saved passwords.

```php
if ($con_password != $password)
{
    $msg = "Confirm Password Doesn't Match";
    $msgClass = "red";
}
elseif (!preg_match('/^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*\W).{8,}$/', $password))
{
    $msg = "Password must include uppercase letters, lowercase letters, numerals, special characters, and at least 8 characters.";
    $msgClass = "red";
}
else
{

    $salt = password_hash(random_bytes(16), PASSWORD_DEFAULT);
    // Combine the password with the salt
    $combinedValue = $salt . $password;

    // Hash the combined value using bcrypt
    $hashedPassword = password_hash($combinedValue, PASSWORD_DEFAULT);
```

Figure 12 : Code Segment for Encryption Password

Session Timeout for this locker booking system has been set to 15 minutes which convert into 900 seconds as shown in Figure 13. The session destroys then use to will destroy all the session data for that user while the session unset then will unset the session variables. In brief, it will wipe out all information related to the active session.

```php
<?php
error_reporting(E_ALL);
ini_set('display_errors', 1);
if (session_status() == PHP_SESSION_NONE) {
    session_start();
}
require_once 'model/db.php';

$inactive_period = 900; // 15 minutes of inactivity

if (isset($_SESSION['last_activity']) && (time() - $_SESSION['last_activity']) > $inactive_period)
{
    session_unset();
    session_destroy();
    header("Location: login.php");
    exit;
}
else
{
    $_SESSION['last_activity'] = time();
}

if (!isset($_SESSION['s_id']))
{
    header("Location: login.php");
    exit;
}

?>
```

Figure 13 : Code Segment for Session Timeout and Session Destroy

4.2 Testing

Such a test case can verify that a certain graphical user interface (GUI) component appears and performs as expected. As shown in Table 2, the test case has been done while clear and concise description has been listed as shown in expected results followed with the modules. All test case in content in each module are successfully meet the functional requirement for both admin and user as the result pass for all.

Table 2 : Testing Case with Result

No.	Test cases	Expected Result	Status
TEST_01_LOGIN MODULE			
1.	TEST_01_01	Worker and Admin enter correct ID and password.	Pass
2.	TEST_01_02	The reCAPTCHA works well and can be use excellently.	Pass
3.	TEST_01_03	The reset password link sent after request on forgot password.	Pass
4.	TEST_01_04	The error message shown when not tick on reCAPTCHA.	Pass
5.	TEST_01_05	All data updated success saves in the database.	Pass
TEST_02_USER REGISTRATION MODULE			
1.	TEST_02_01	Valid user ID, full name, IC Number, telephone number, password and match password needed.	Pass
2.	TEST_02_02	The validation messages are shown in all field if user enter wrong input.	Pass
3.	TEST_02_03	The One Time-Password sent to registered email.	Pass
4.	TEST_02_04	New worker success registers for new account.	Pass
3.	TEST_01_05	All data success saves in the database.	Pass
TEST_03_BOOKING LOCKER MODULE			
1.	TEST_03_01	User access the list of Homepage.	Pass
2.	TEST_03_02	Users choose desire locker and date.	Pass
3.	TEST_03_03	Users fill in all details and success to make the reservation.	Pass
4.	TEST_03_04	Employees' dashboard automatically updates the locker booking status from 'pending', 'approved' and 'expired'.	Pass
5.	TEST_03_05	All data success saves in the database.	Pass
TEST_04_MANAGE BOOKING MODULE			
1.	TEST_04_01	Admin view list of booking details	Pass
2.	TEST_04_02	Admin able to accept the booking.	Pass
3.	TEST_04_03	Admins able to delete the booking.	Pass
4.	TEST_04_04	The status of locker automatically changes from 'pending', 'approved' and 'expired'.	Pass
5.	TEST_04_05	All data success saves in the database.	Pass
TEST_05_MANAGE LOCKER MODULE			
1.	TEST_05_01	Admin view list of available locker.	Pass
2.	TEST_05_02	Admin able to insert new locker.	Pass
3.	TEST_05_03	Admin able to edit locker status.	Pass
4.	TEST_05_04	Admin able to delete locker.	Pass
5.	TEST_05_05	All data success saves in the database.	Pass
TEST_06_MANAGE ADMIN MODULE			
1.	TEST_06_01	Admin view the admin list.	Pass
2.	TEST_06_02	Admin able to insert new admin.	Pass
3.	TEST_06_03	Admin able to edit admin profile.	Pass
4.	TEST_06_04	Admin able to delete other admin.	Pass
5.	TEST_06_05	All updated data success saves in the database.	Pass
TEST_07_MANAGE USER MODULE			
1.	TEST_07_01	Admin view a list of user list.	Pass
2.	TEST_07_02	Admin able to edit user profile.	Pass
3.	TEST_07_03	Admin able to delete user account.	Pass
4.	TEST_07_04	All updated data success saves in the database.	Pass
TEST_08_REPORT ANALYSIS MODULE			
1.	TEST_08_01	Admin view the report of locker in chart to visualize the data followed by their categorized.	Pass

No.	Test cases	Expected Result	Status
TEST_09_ACTIVITY LOG MODULE			
1.	TEST_09_01	Admin view activity list	Pass
2.	TEST_09_02	Admin choose activity which categorized as login, update, insert, delete, register and booking.	Pass

The user acceptance result is getting from one respondent which is Mr. Abdul Aziz Bin Razali where he is the manager of Wajasakti Sdn Bhd. He has test both side of user which is worker side and admin side before respond on the user testing form. He also gives comment on adding printing function to make improvement for this locker booking system.

Table 3 provides a summary of the Security Checklist's findings from the User Acceptance Test. The Security Checklist's findings are based on the feedback provided by the user after used the locker booking system.

Table 3 : Security Testing Result

	Security Requirements	Pass	Fail
1.	A validation message is shown if the username or the password field is left blank when login.	✓	0
2.	Validation messages will show up if user not fill in all field provided in all module.	✓	0
3.	Make sure the improper data authentication is not specified in the error message. For instance, the words "Wrong password" or "Wrong worker ID" shouldn't appear in an error message.	✓	0
4.	Enforce the password's complexity. The password must 8 characters long, for instance, be a combination of digits, small letters, capital letters, and special characters.	✓	0
5.	Passwords in the textbox need to be hidden.	✓	0
6.	One Time Passwords (OTPs) are sent to registered emails and have a 15-minute usage limit.	✓	0
7.	Worker needs to tick on reCAPTCHA before login into the system. The error message will display if not tick on it.	✓	0
8.	An error message will be shown if password is not match with confirm password.	✓	0
9.	The system will be automatically logout after 15 minutes of inactive.	✓	0
11.	Forgot password link works with one minute duration and server send back the link through email upon request.	✓	0
10.	Logout modules end all the session in the system.	✓	0

The user acceptance result is getting from one respondent which is Mr. Abdul Aziz Bin Razali where he is the manager of Wajasakti Sdn Bhd. He has test both side of user which is worker side and admin side before respond on the user testing form. He also gives comment on adding printing function to make improvement for this locker booking system

A single test must be run in accordance with the specification of the inputs, execution circumstances, testing process, and anticipated outcomes to obtain. There are 42 test cases to ensure the proper operation of the system functions. Table 4 shows the contains the system's operational status which then produce a graph for overall result of testing of security and user acceptance. The result shows all of modules has passed the test cases with 100% results as it satisfied with all 42 test cases.

Table 4 : System Operational Status

Test Cases	Total of Pass Test Cases	Passed (100%)
Registration Module	5/5	100%
Log Module	5/5	100%
Booking Locker Module	5/5	100%
Manage Booking Module	5/5	100%
Manage Locker Module	5/5	100%
Manage Admin Module	5/5	100%
Manage User Module	4/4	100%
Report Analysis Module	1/1	100%
Activity Log Module	2/2	100%

The overall result of the test case as shown in Figure 13 taken from System Operational Status which each module then classified into a graph to get more specific and logic details.

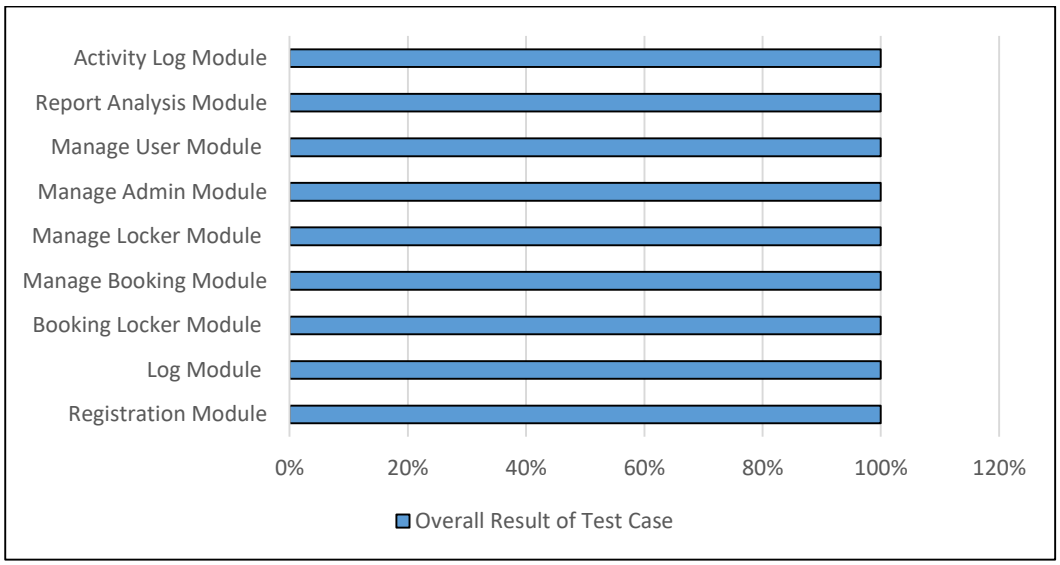


Figure 14 : Overall Result of Operational Test Cases

5. Conclusion

Both users and administrators can benefit greatly from a locker reservation system. It simplifies locker reservations, enhances security, increases user comfort, and provides management with relevant data. By using a locker booking system, Wajasakti Sdn Bhd may effectively manage their locker spaces, enhance the user experience, and streamline their operations. Overall, E-Locker Booking System with Multifactor Authentication for Wajasakti Sdn Bhd is a practical instrument that accelerates, organises, and ups security in locker management. By using an online system, users can instantly check the availability of lockers, make reservations in advance, and receive real-time updates on their bookings. Because of this, there is less administrative work to be done, and there are fewer chances for conflicts or double bookings.

The advantages of this locker booking system is convenience to use because it allows users to reserve them in advance. Then, the systems use security features like unique authentication. By adding authorization to the reserved lockers to authorised staff, this improves the security of individual belongings and reduces the likelihood of theft or tampering. Other than that, it give effective management to the admin as It makes it possible for administrators to monitor locker availability, usage,

and demand-based locker distribution. This reduces the possibility of conflict or double bookings and guarantees effective locker utilisation

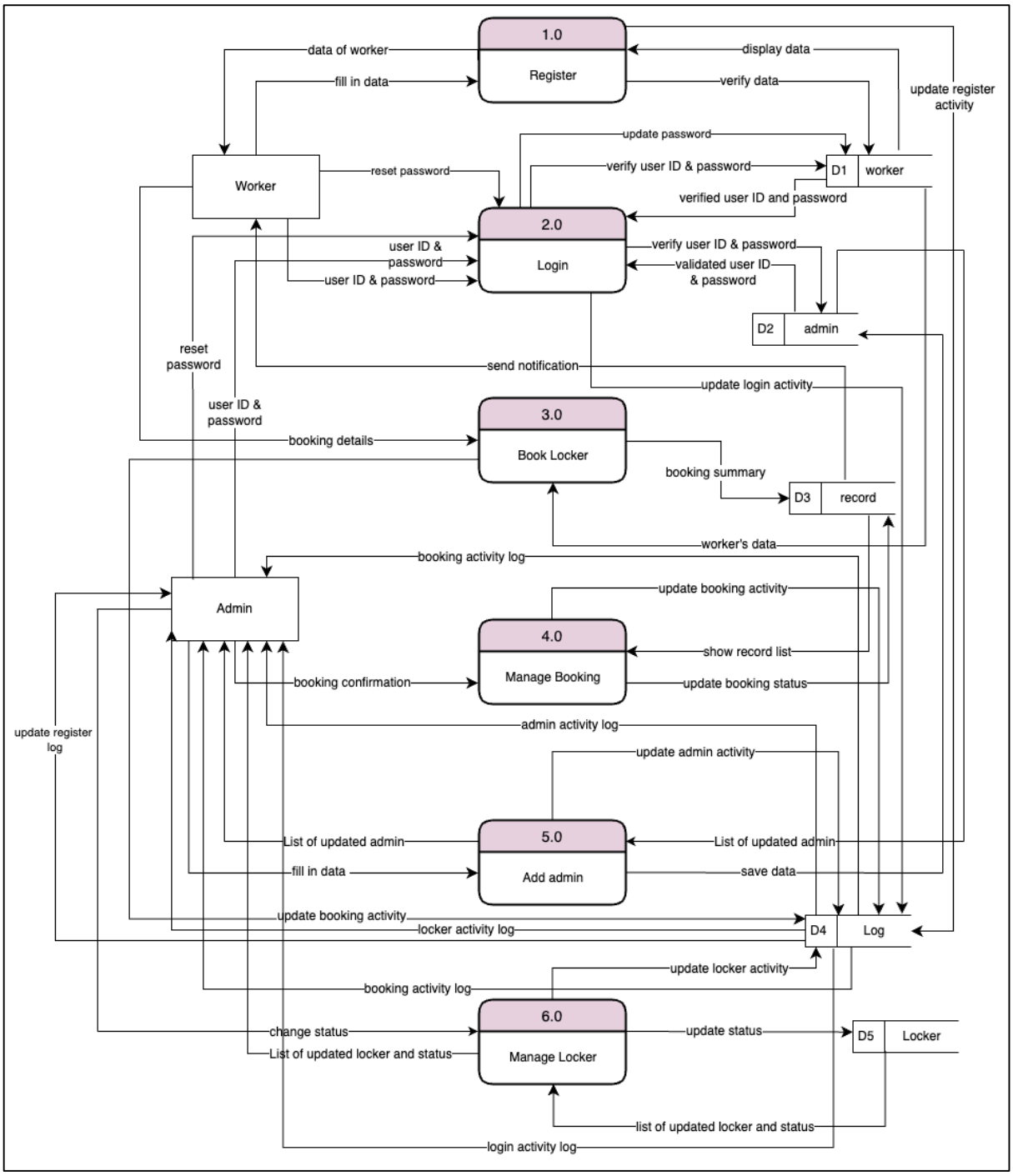
One of the disadvantages of this locker booking system is it cannot be used without internet connection. The second is the payment is separate from the system which need to be manually pay in person and the last one is there is no video which provides navigation or step to use for user to interact with the system.

In future work, this locker booking system can use online payment to that offers a secure and dependable payment integration. This makes it easier for users to pay for their reserved lockers and promotes the validity of reservations. On top of that, by adding security measures in place to prevent unauthorised access. Features like Internet of Things (IoT) such automation of unlock and lock locker, private access codes, and face ID mechanisms and last one by adding user feedback which is by adding printing mechanism to help the admin do auditing or reporting.

Acknowledgement

The authors are grateful to University Tun Hussein Onn Malaysia's Faculty of Computer Science and Information Technology also to supervisor for their assistance.

Appendix A



Data Flow Diagram Level 0 (DFD Level 0)

References

[1] Dickinger^, A., & Mazanec^, J. (n.d.). Consumers' Preferred Criteria for Hotel Online Booking. Retrieved October 24, 2022, from https://link.springer.com/chapter/10.1007/978-3-211-77280-5_22

[2] Paterson, K. G., & Stebila, D. (2010). One-time-password-authenticated key exchange. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 6168 LNCS, 264–281. https://doi.org/10.1007/978-3-642-14081-5_17

[3] Aloul, F. A., El-Hajj, W., Aloul, F., Zahidi, S., & El-Hajj, W. (2009). Multi Factor Authentication Using Mobile Phones Wireless Security View project OMA: Opinion Mining in Arabic View project Multi Factor Authentication Using Mobile Phones. In *International Journal of Mathematics and Computer Science* (Vol. 4, Issue 2). https://www.researchgate.net/publication/228972704

[4] Baykara, S. (2021, December 13). *PCI DSS*. Retrieved from Session Timeout Requirements: https://www.pcidssguide.com/pci-dss-session-timeout-requirements/#:~:text=A%20session%20timeout%20represents%20an,defined%20by%20a%20web%20server.

[5] Anderson, R., & Safari, an O. M. Company. (n.d.). *Security Engineering, 3rd Edition*.

[6] Ávila, R., Khoury, R., Khoury, R., & Petrillo, F. (2021). Use of Security Logs for Data Leak Detection: A Systematic Literature Review. In *Security and Communication Networks* (Vol. 2021). Hindawi Limited. https://doi.org/10.1155/2021/6615899

[7] Poongodi, M., Vijayakumar, V., Al-Turjman, F., Hamdi, M., & Ma, M. (2019). Intrusion Prevention System for DDoS Attack on VANET with reCAPTCHA Controller Using Information Based Metrics. *IEEE Access*, *7*, 158481–158491. https://doi.org/10.1109/ACCESS.2019.2945682

[8] Patwa, S. (2021). eLocker | Book secure shared bicycle parking. Retrieved from SHIVANG PATWA: https://www.shivangpatwa.com/projects/elocker-book-secure-shared-bicycle-parking#Prototype

[9] Sentral, K. (12 May, 2021). *KL SENTRAL LUGGAGE STORAGE / LOCKER*. Retrieved from KL Sentral: https://www.klsentral.info/kl-sentral-luggage-storage-locker/

[10] Storage, R. (2021). *Radical Storage*. Retrieved from https://radicalstorage.com/?ac=532&subac=c:gadto:enct:brlt:ml&gclid=Cj0KCQiAgribBhDkARIsAASA5buAOC7lYEKgI4praTaLAiwdsz5zDWNE_g6mwikfEB73cFQNIL4-GJYaAgMnEALw_wcB&gclsrc=aw.ds

[11] Špundak, M. (2014). Mixed Agile/Traditional Project Management Methodology – Reality or Illusion? Procedia - Social and Behavioral Sciences, 119, 939–948. https://doi.org/10.1016/j.sbspro.2014.03.105

[12] Kumar, G., Kumar Bhatia, P., & Jambheshwar, G. (2012). Impact of Agile Methodology on Software Development Process Software Cost Estimation View project Algorithms View project. In International Journal of Computer Technology and Electronics Engineering (IJCTEE (Vol. 2, Issue 4). https://www.researchgate.net/publication/255707851