APPLIED INFORMATION TECHNOLOGY AND COMPUTER SCIENCE
e-ISSN: 2773-5141

AITCS

Vol. 6 No. 1 (2025) 979-998
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

Dhiya Az Zahra Umrah Management System

Adib Nawfal Yazid[1], Rabatul Aduni Sulaiman[1]*

[1] *Fakulti Sains Komputer dan Teknologi Maklumat,*
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author: rabatul@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2025.06.01.054

Article Info

Received: 15 June 2024
Accepted: 18 July 2024
Available online: 30 June 2025

Keywords

Umrah, Reservation, Management

Abstract

Dhiya Az Zahra is a company specializing in Umrah services, but the current process relies on WhatsApp, Telegram, and Email for communication, leading to inefficient operations and disorganized customer registrations. The project objectives involve designing the Dhiya Az Zahra Umrah Management System using object-oriented approach, developing the Dhiya Az Zahra Umrah Management System using a web-based approach, and testing the developed Dhiya Az Zahra Umrah Management System using system testing and user acceptance testing. The project uses iterative methodology as the software development lifecycle model. After system deployment, customers can make Umrah reservation, and administrators and staff can manage customer data through web browsers. The project completion should result in a web-based system for the Dhiya Az Zahra Umrah Management System that can assist in the Umrah reservation process and ensure efficient data management for Dhiya Az Zahra.

1. Introduction

The Umrah is an Islamic pilgrimage to holy cities of Saudi Arabia, Makkah and Medina and is performed by Muslims that can be undertaken at any time of the year, in contrast to the Haj which has specific dates according to the Islamic lunar calendar [1]. Umrah is an act of Sunnah (done by Prophet Mohammad, peace and blessings be upon him) of individual worship starting by Ihram from the Miqat, followed by visiting Masjid al-Haram for Tawaf (circling the Ka'aba seven times counter-clockwise) and Sai'e (travelling seven times between the as-Safa and al-Marwah hills), and ending with shaving off or cutting short of the hair [2]. More than 300,000 pilgrims from Malaysia have help performed the Umrah since the start of the Umrah season on July 30, 2022, and looking forward to the arrival of 31,600 pilgrims from Malaysia during the Haj season of 1444 Hijrah, said Saudi Arabia's Minister of Hajj and Umrah, Dr Tawfiq Fawzan Al-Rabiah [3].

Dhiya Az Zahra Travel & Tours Sdn Bhd is a company specializing in Umrah services. Since Dhiya Az Zahra does not have an automated system, four problems have been identified, including disorganization, time consumption, limited data management, and the risk of data loss. The existing manual process leads to inefficient operations, resulting in disorganized customer registrations. Manual customer reservation is time-consuming and prone to errors, especially with an increasing number of customers, causing processing delays, miscommunication, and potential data inaccuracies. Additionally, the lack of effective data management makes it challenging to track essential customer details. Relying on manual documentation poses a significant risk of data loss due to misplacement or damage of physical records, potentially disrupting the registration process.

The objectives of this project are to design the Dhiya Az Zahra Umrah Management System using object-oriented approach, to develop the Dhiya Az Zahra Umrah Management System using a web-based approach, and to test the developed Dhiya Az Zahra Umrah Management System using system testing and user acceptance testing. The implementation of the Dhiya Az Zahra Umrah Management System is crucial for handling the Umrah

reservation process and ensuring efficient data management for customers. This system is designed not only to simplify the reservation process but also to minimize staff workloads and significantly contribute to the overall success of Dhiya Az Zahra operations in the Umrah industry.

This paper is organized into five sections. Section 1 explains the project introduction. Section 2 discusses related work. Section 3 describes the methodology used in the project and discusses each of its phases. Section 4 describes the results and discussions. Lastly, Section 5 concludes the project.

2. Related Work

In this section, the domain background, Umrah management system, and result of the comparative analysis are discussed.

2.1 Domain Background

Dhiya Az Zahra's headquarters are located at No. 4, Jalan Beringin, Taman Beringin, 81400 Senai, Johor. As the current staff consists of family members, the office is primarily used for storing luggage and items related to Umrah. Customers who are interested in joining the Umrah trip can contact the owner using communication applications such as WhatsApp, and Telegram, or by directly calling the owner. Upon receiving the customer's message, the owner will provide the details about the Umrah packages including the date, trip, price, and any relevant information. If the customer agrees to register for the selected date's trip, the owner will request the required documents, including the identification card, passport, and payment for the trip.

Then, the customer will be invited to the Umrah group on Telegram. Upon payment for the Umrah, staff will prepare the digital receipt and send it to the customer. The customers information will then be inserted into an Excel file. Any updates or information about the trip will be informed within the group. Before the trip date, each customer will be provide with one luggage and one sling bag for use during the Umrah. Dhiya Az Zahra company not have any online system and only depends on communication applications for Umrah reservation.

2.2 Umrah Management System

According to a study by Chan et al. [4], there's a growing trend of Internet-based businesses worldwide, serving as a means for creating wealth and acquiring property. Seo et al. [5] also suggests that in today's economy, factors like skills, workplace management, information and communications technology (ICT), and knowledge play a crucial role in maintaining competitiveness. Additionally, Mahmad et al. [6] highlight the importance of using the latest applications such as WhatsApp, Telegram, and Facebook to promote Umrah packages and meet current demands.

The Umrah management system is a web-based system that enables customers to browse and make reservations for Umrah. The reservation process is completed by providing the necessary details and documents within the system. Staff can efficiently manage the customer data through the system. In summary, the Umrah management system improves the reservation process compared to manual methods.

2.3 Comparative Analysis

This section provides a comparative analysis of three existing applications which are AndalusiaKu [7], Az Zuha Travel [8], and Ikhlas Umrah [9]. Table 1 shows the comparison between the three existing systems and the proposed system. The compared features include customer registration, login, package management, Umrah management, account management, tracking management, generate report, and feedback management.

Table 1 *System's Comparison*

Features/System	AndalusiaKu	Az Zuha Travel	Ikhlas Umrah	Dhia Az Zahra Umrah Management System
Customer Registration	Yes	No	Yes	Yes
Login	Yes	No	Yes	Yes
Package Management	N/A	N/A	N/A	Yes
Umrah Management	Yes	No	Yes	Yes
Account Management	Yes	No	Yes	Yes
Tracking Management	No	No	No	Yes
Generate Report	N/A	N/A	N/A	Yes
Feedback Management	No	No	Yes	Yes

In summary, the proposed system offers a comprehensive set of features, including customer registration, login, package management, Umrah management, account management, tracking management, report generation, and feedback management. The comparison shows that the proposed system is improved from the existing systems.

3. Methodology

This chapter explained the methodology used to develop the Dhiya Az Zahra Umrah Management System. Iterative model is chosen for software development lifecycle (SDLC) of this project. According to Shylesh [10], in the iterative model, requirements aren't finalized at the beginning. The process starts with a small set of requirements. With each iteration, a module of the system is developed, and the cycle continues until the final version is achieved. The iterative process model begins implementation with a module of requirement specifications. Each iteration adds a new module, and the process continues until completion.

3.1 Planning Phase

In this phase the main goal is to clearly define what the project aims and needs to achieve. This involves conducting an initial meeting with stakeholders, proposing the project, and identifying the project scope and objectives to ensure a clear understanding of what needs to be accomplished. To achieve these tasks, requirements gathering from stakeholders is needed to ensure the stakeholders opinions and expectations are thoroughly considered. There are two methods of requirement gathering that have been conducted which are interview and observation.

3.2 Analysis Phase

In the analysis phase, the main goal is to carefully examine the requirements provided by stakeholders. This involves conducting interviews with stakeholders to analyze the requirements and define the modules of the system.

The modules identified in the system include:

- Customer Registration - Allows customers to register for the Umrah system.
- Login - Allows administrator, staff, and customers to login to the system.
- Package Management - Allows staff to customize or manage Umrah packages.
- Umrah Management - Allow customers to make Umrah reservations.
- Account Management - Allow users to manage their accounts and profiles.
- Tracking Management - Allow staff to track the current customer location during Umrah.
- Generate Report - Generates reports on customer data and Umrah reservations for administrator and staff.
- Feedback Management - Allow customers to provide feedback on the system.

3.3 Design Phase

In the design phase, the goal is to create a plan for how the project will be built. This includes designing the unified modelling language (UML) diagrams for the proposed system which include the overall structure, specifying how the system should work, and planning how different parts of the project will interact.

3.4 Implementation Phase

In the implementation phase, the main goal is to develop the proposed. This involves developing a module in each iteration of the implementation phase according to the system design of the proposed system, along with the system's database.

3.5 Testing Phase

In the testing phase, the main goal is to check and ensure that the system works well. This involves creating test cases. System modules are then tested based on these test cases, using both system testing and user acceptance testing to determine if the system is easy to use and meets user expectations.

3.6 Review Phase

In the review phase, the main goal is to review the functionality of each module and get feedback from stakeholders. The evaluation with stakeholders is conducted through meetings using Google Meet to discuss how the system is performing, what features are working well, and if there are any areas that need improvement.

3.7 Deployment Phase

In this phase, the completed system will be deployed on the live server. This involves finding a web server to host the system, deploying the system on the hosting service, and ensuring the system is functional after deployment.

4. Results and Discussion

This chapter explains the analysis and design of the proposed system, followed by the implementation and testing of the system. In Section 4.1, the general system architecture of the proposed system is presented. Section 4.2 discusses the system requirements analysis, including user, functional, and non-functional requirements. Section 4.3 covers the use case diagrams of the proposed system. The class diagram is discussed in Section 4.4. Section 4.5 shows the interface design of the system, while Section 4.6 focuses on the implementation of the system. Lastly, Section 4.7 presents the results of the testing.

4.1 General System Architecture

Fig. 1 shows the system architecture of the Dhiya Az Zahra Umrah Management System. The system is web-based and will be deployed on the web servers. There are three types of users for this system, which are administrator, staff, and customer.

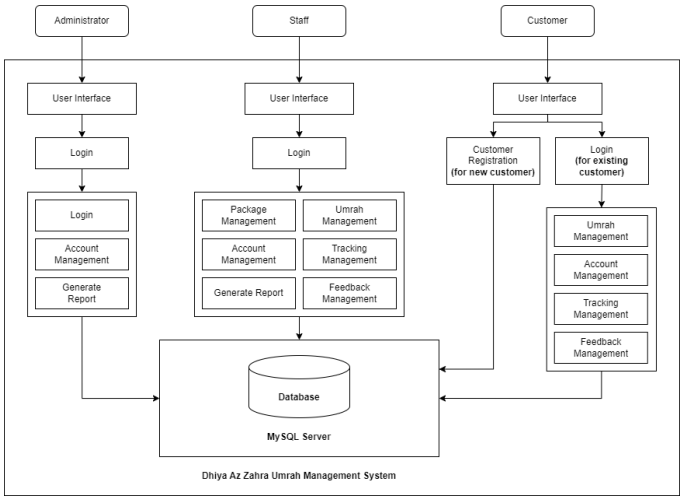


Fig. 1 *System Design Diagram of the Proposed System*

4.2 System Requirements Analysis

In this section, user requirements, functional requirements, and non-functional requirements will be presented and discussed. The system requirement analysis addresses the specific functionalities to be implemented in the proposed system. Table 2 shows the user requirements of the proposed system. Table 3 shows the functional requirements of the proposed system. Lastly, Table 4 shows the non-functional requirements of the proposed system.

Table 2 *User requirements of the proposed system*

User	Requirements
Administrator	Enable to login to the system, account profiles, create staff account, and generate reports.
Staff	Enable to login to the system, manage Umrah packages, manage Umrah reservations, manage account profiles, track the current customer location during Umrah, generate reports, and manage customer feedback.
Customer	Enable to register an account, login to the system, make Umrah reservations, manage account profiles, provide the current location, and give feedback to the system.

Table 3 *Functional requirements of the proposed system*

Modules	Requirements
Customer Registration	The system shall be able to allow customers to register an account.
Login	The system shall be able to allow users to login to the system and reset their account password.
Package Management	The system shall be able to allow staff to create, edit, and delete Umrah package.
Umrah Management	The system shall be able to allow staff to update and delete customer reservations. Additionally, the system shall be able to allow customers to make Umrah reservation and make cash or transfer payment for the Umrah reservation.
Account Management	The system shall be able to allow users to update their profile details and change their account password. Additionally, the system shall be able to allow administrator to create staff account and delete staff account.
Tracking Management	The system shall be able to allow staff to track the current customer location and receive the current customer location during Umrah.
Generate Report	The system shall be able to allow users to convert the selected report to PDF file and the system shall be able to create a downloadable PDF report file.
Feedback Management	The system shall be able to allow staff to reply and delete customer feedback and customer to give and delete the feedback.

Table 4 *Non-functional requirements of the proposed system*

Modules	Requirements
Operational	The system should be easy to use, work with any web browser, such as Google Chrome, Mozilla Firefox, or Microsoft Edge, and have a responsive design, ensuring optimal user experience on various devices.
Performance	The system should be always accessible and any interaction between the user and the system should take no longer than 3 seconds.
Security	User passwords should have a minimum length of 8 characters and include a combination of letters, numbers, and special characters and the system shall deny access to any user if the username and password input are incorrect. Additionally, customers must log into their accounts before making Umrah reservations.
Usability	The user interface and flow of the system should be easily understood by the user and the system should have a consistent layout and navigation structure across all pages.
Integrity	The password stored in the database should be encrypted.

4.3 Use Case Diagram

Fig. 2 shows the use case diagram of the proposed system. There are three types of users who are involved in the system and each of the users involved with some of the system use case. Administrator are involved in login, account management, and generate report use case. One the other hand, Staff are involved in login, package management, Umrah management, account management, tracking management, generate report, and feedback management. Lastly customers are involved in customer registration, login, Umrah management, account management, tracking management, and feedback management.

Fig. 2 *Use Case Diagram of the Proposed System*

4.4 Class Diagram

The class diagram illustrates the conceptual model in database modelling as shown in Fig. 3. The class includes the administrator, staff, customer, package, reservation, location, report, and feedback of the database tables. One administrator can generate one or many reports and one or many reports can be generated by one administrator. One administrator can manage one or many staff and one or many staff can be managed by one administrator. Many staff can generate one or many reports and one or many reports can be generated by many staff. Many staff can manage one or many packages and one or many packages can be managed by many staff. Many staff can manage one or many reservations and one or many reservations can be managed by many staff. Many staff can manage one or many feedback and one or many feedback can be managed by many staff. One customer can give one or many feedback and one or many feedback can be given by one customer. One customer can make one or many reservations and one or many reservations can be made by one customer. One customer has one location and one location is held by one customer. One reservation includes one package, and one package is included in one reservation.

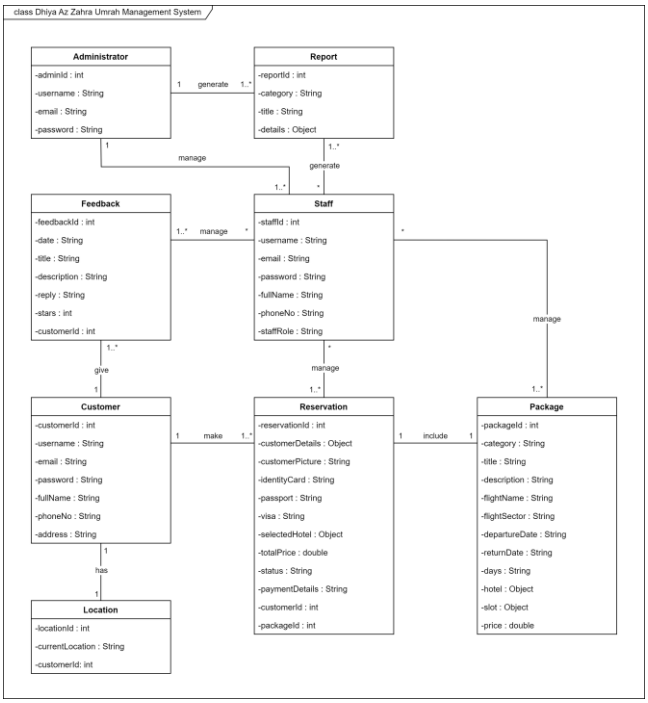


Fig. 3 *Class Diagram of the Proposed System*

Applied Information Technology and Computer Science Vol. 6 No. 1 (2025) p. 979-998

4.5 Interface Design

In this section, each interface of the system module will be shown. The interfaces shown in each system module are wireframes that represent the illustration of each page interface. Fig. 4 shows the interface for customer registration modules. Fig. 5 shows the interface for login modules. Fig. 6 shows the interface for package management modules. Fig. 7 shows the interface for Umrah management modules. Fig. 8 shows the interface for account management modules. Fig. 9 shows the interface for tracking management modules. Fig. 10 shows the generate report modules. Lastly, Fig. 11 shows the interface for feedback management modules.

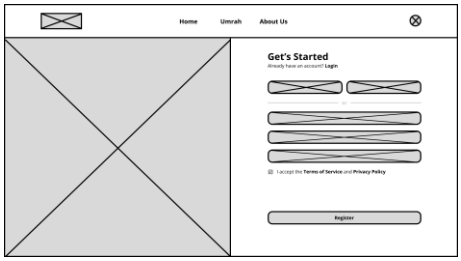
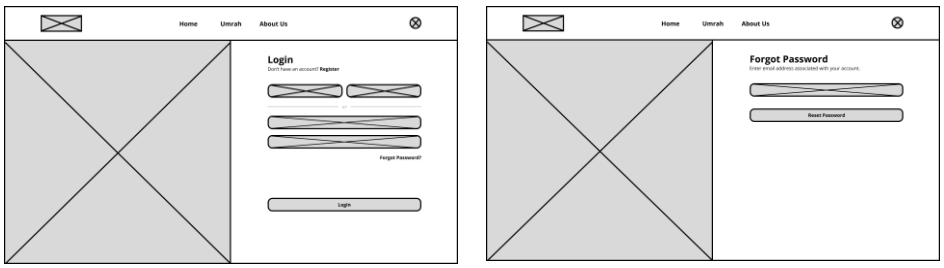
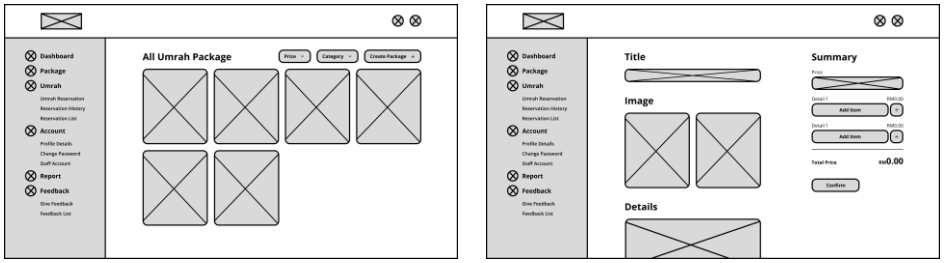


Fig. 4 *Interface for Customer Registration modules; Customer Registration*



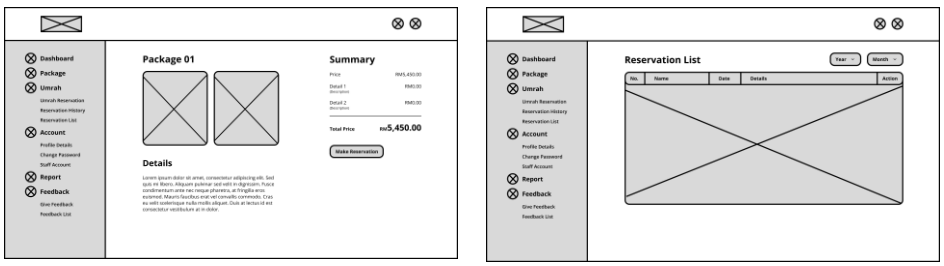
(a) **(b)**

Fig. 5 *Interface for Login modules (a) Login; (b) Forgot Password*



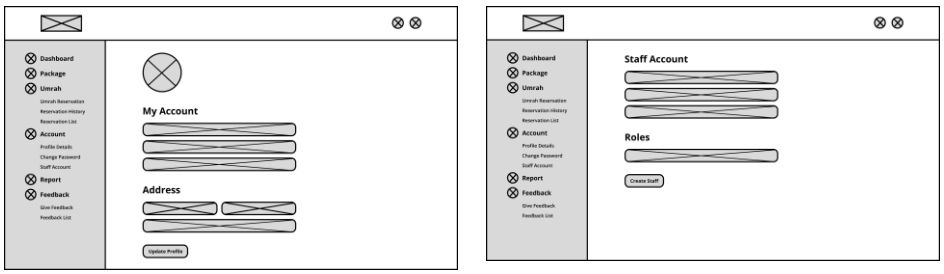
(a) **(b)**

Fig. 6 *Interface for Package Management modules (a) Package; (b) Create Package*



(a) **(b)**

Fig. 7 *Interface for Umrah Management modules (a) Umrah Package Details; (b) Reservation List*



(a) **(b)**

Fig. 8 *Interface for Account Management modules (a) Account; (b) Create Staff Account*

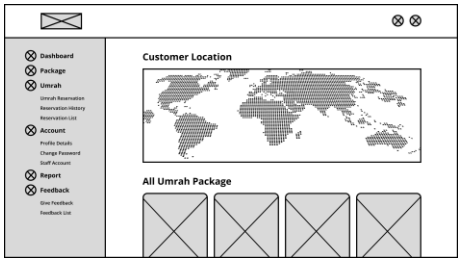


Fig. 9 *Interface for Tracking Management modules; Dashboard*

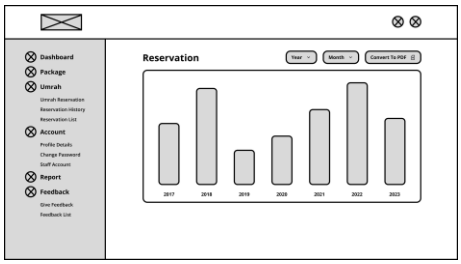
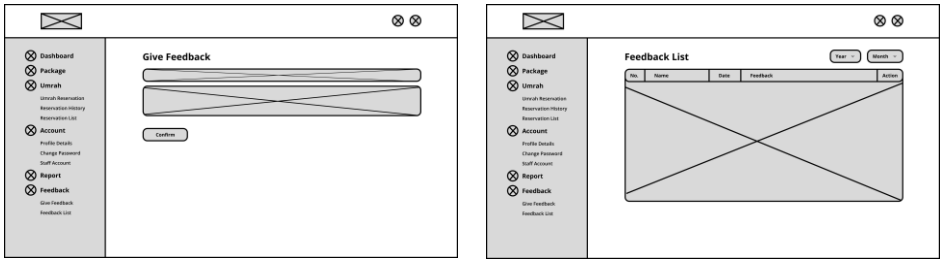


Fig. 10 *Interface for Generate Report modules; Report*



(a) **(b)**

Fig. 11 *Interface for Feedback Management modules (a) Give Feedback; (b) Feedback List*

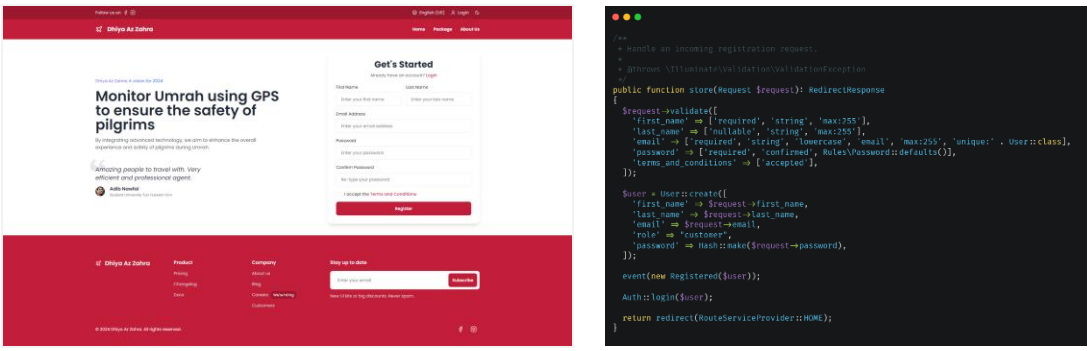
4.6 Implementation

In this section, the user interface and code segments of each module will be explained. There are a total of eight modules in the Dhiya Az Zahra Umrah Management System, which are customer registration, login, package management, Umrah management, account management, tracking management, generate report, and feedback management.

4.6.1 Customer Registration Module

In this module, customers can register an account to access the system. To register an account, customers need to go to the Registration Page and fill out the required details for the registration process. Fig. 12 shows the interface and code segment for Customer Registration modules. Customers need to input their first name, last name, email address, password, and confirm password to register an account. The system will validate the input

provided by the customers. If all of the inputs are valid, the system will store the customer registration details in the database and the system will redirect the customers to the Dashboard Page.

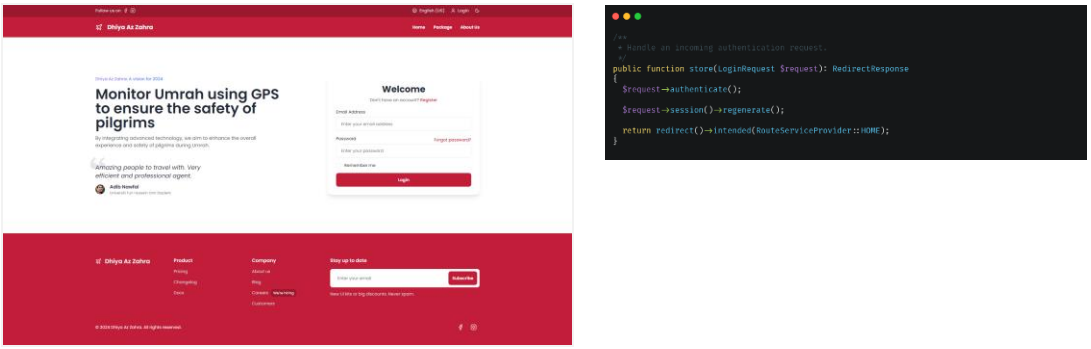


(a) **(b)**

Fig. 12 *Customer Registration modules (a) Interface; (b) Code Segment*

4.6.2 Login Module

In this module, administrators, staff, and customers can login to the system using their login credentials. Fig. 13 shows the Login interface and code segment. Administrators, staff, and customers need to input their email and password to be able to login to the system. The system will validate the login input provided by the administrator, staff, and customers. If all of the input is valid, the system will redirect the administrator, staff, and customer to the Dashboard Page.



(a) **(b)**

Fig. 13 *Login modules (a) Interface; (b) Code Segment*

4.6.3 Package Management Module

In this module, administrators and staff can manage the Umrah packages. Fig. 14 shows the Package Management interface, which will display all the Umrah packages that have been created. Administrators and staff can add a new Umrah package by clicking the Add Package button, and the system will redirect the administrator and staff to the Add Umrah Package Page. Fig. 15 shows the code segment to Add Package and Update Package. In add package code segment, the system will validate the Umrah package details provided by the administrator and staff. If all of the input is valid, the system will store the Umrah package details in the database, and the system will redirect administrators and staff to the Package Management Page. In update package code segment, the system will validate the Umrah package details provided by the administrator and staff. If all of the input is valid, the system will update the Umrah package details in the database, and the system will redirect administrators and staff to the Package Management Page.

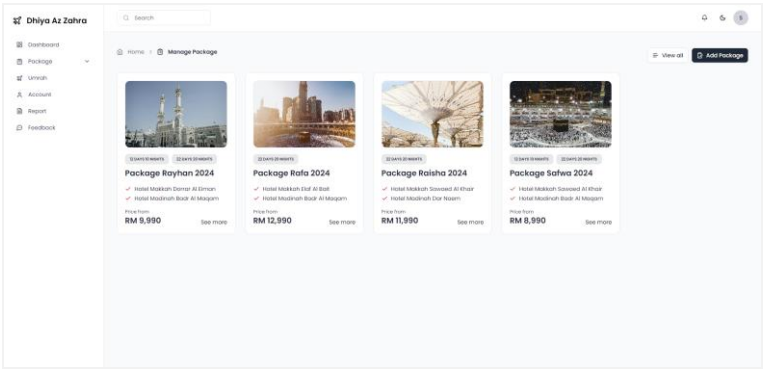


Fig. 14 *Interface for Package Management modules; Manage Package*

```php
/**
 * Submit add package form.
 */
public function postAddPackage(Request $request)
{
  $price_12_10 = new Price();
  $price_22_20 = new Price();
  $package = new Package();

  $this->validate($request, [
    'cover_img' => ['required', 'mimes:jpeg,jpg,png', 'max:10000'],
    'name' => ['required', 'string', 'max:255'],
    'year' => ['required', 'integer', 'min:1900'],
    'hotel_makkah' => ['required', 'exists:hotels,id'],
    'hotel_madinah' => ['required', 'exists:hotels,id'],
    'price_12_10_room_4_5' => ['nullable', 'numeric', 'min:1'],
    'price_12_10_room_3' => ['nullable', 'numeric', 'min:1'],
    'price_12_10_room_2' => ['nullable', 'numeric', 'min:1'],
    'price_22_20_room_4_5' => ['nullable', 'numeric', 'min:1'],
    'price_22_20_room_3' => ['nullable', 'numeric', 'min:1'],
    'price_22_20_room_2' => ['nullable', 'numeric', 'min:1'],
  ]);

  if (!$request->has('price_12_10_room_4_5') && !$request->has('price_22_20_room_4_5')) {
    // throws error to view
  }

  if ($request->filled('price_12_10_room_4_5')) {
    $price_12_10->package = '12 Days 10 Nights';
    $price_12_10->room_4_5 = $request['price_12_10_room_4_5'];
    $price_12_10->room_3 = $request['price_12_10_room_3'];
    $price_12_10->room_2 = $request['price_12_10_room_2'];
    $price_12_10->save();
    $package->package_12_10_id = $price_12_10->id;
  }

  if ($request->filled('price_22_20_room_4_5')) {
    $price_22_20->package = '22 Days 20 Nights';
    $price_22_20->room_4_5 = $request['price_22_20_room_4_5'];
    $price_22_20->room_3 = $request['price_22_20_room_3'];
    $price_22_20->room_2 = $request['price_22_20_room_2'];
    $price_22_20->save();
    $package->package_22_20_id = $price_22_20->id;
  }

  // Save profile picture
  if ($request->hasFile('cover_img')) {
    $image = $request->file('cover_img');
    $filename = 'package_' . time() . '.' . $image->getClientOriginalExtension();

    // Ensure the directory exists
    $path = public_path('images/packages');
    if (!File::isDirectory($path)) {
      File::makeDirectory($path, 0777, true, true);
    }

    Image::make($image)->save($path . '/' . $filename);
    $package->cover_img = $filename;
  }

  $package->name = $request['name'];
  $package->year = $request['year'];
  $package->hotel_makkah_id = $request['hotel_makkah'];
  $package->hotel_madinah_id = $request['hotel_madinah'];
  $package->save();

  return Redirect::route('package.manage-package')->with('status', 'package-submitted');
}
```

(a)

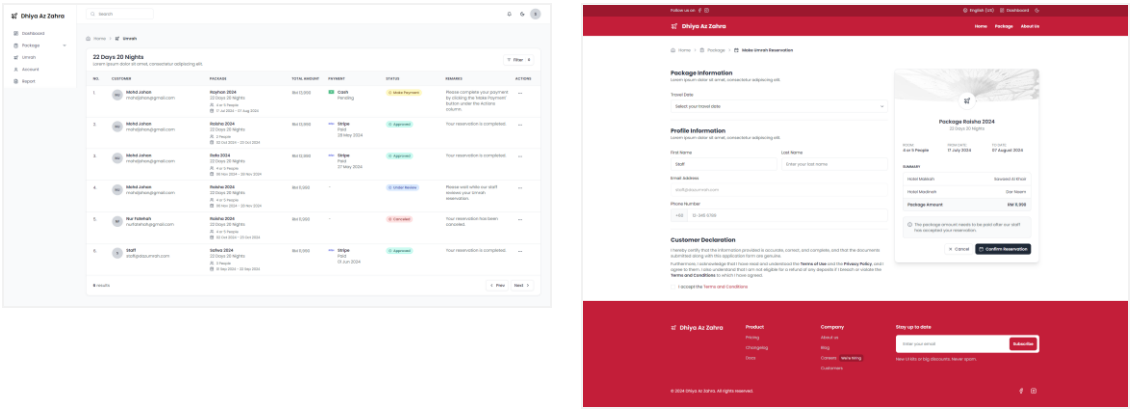
```php
/**
 * Submit update package form.
 */
public function patchUpdatePackage(Request $request, string $id)
{
  $package = Package::findOrFail($id);

  // Save profile picture
  if ($request->hasFile('cover_img')) {
    $image = $request->file('cover_img');
    $filename = 'package_' . time() . '.' . $image->getClientOriginalExtension();

    // Ensure the directory exists
    $path = public_path('images/packages');
    if (!File::isDirectory($path)) {
      File::makeDirectory($path, 0777, true, true);
    }

    Image::make($image)->save($path . '/' . $filename);
    $package->update(['cover_img' => $filename]);
  }

  $packageData = $request->validate([
    'name' => ['required', 'string', 'max:255'],
    'year' => ['required', 'integer', 'min:1900'],
    'hotel_makkah' => ['required', 'exists:hotels,id'],
    'hotel_madinah' => ['required', 'exists:hotels,id'],
  ]);

  $package->update($packageData);

  $this->validate($request, [
    'price_12_10_room_4_5' => ['nullable', 'numeric', 'min:1'],
    'price_12_10_room_3' => ['nullable', 'numeric', 'min:1'],
    'price_12_10_room_2' => ['nullable', 'numeric', 'min:1'],
    'price_22_20_room_4_5' => ['nullable', 'numeric', 'min:1'],
    'price_22_20_room_3' => ['nullable', 'numeric', 'min:1'],
    'price_22_20_room_2' => ['nullable', 'numeric', 'min:1'],
  ]);

  $package->package_12_10()->update([
    'room_4_5' => $request['price_12_10_room_4_5'],
    'room_3' => $request['price_12_10_room_3'],
    'room_2' => $request['price_12_10_room_2'],
  ]);

  $package->package_22_20()->update([
    'room_4_5' => $request['price_22_20_room_4_5'],
    'room_3' => $request['price_22_20_room_3'],
    'room_2' => $request['price_22_20_room_2'],
  ]);

  return Redirect::route('package.package-details', $id)->with('status', 'package-updated');
}
```

(b)

Fig. 15 *Code Segment for Management modules (a) Add Package; (b) Update Package*

4.6.4 Umrah Management Module

In this module, customers can make Umrah reservations, while administrators and staff can manage the Umrah reservations. Fig. 16 shows the interface for Umrah Management modules. In the reservation list interface, the system will show all the Umrah reservations that have been made by the customer. Customers can make Umrah reservations by clicking the Browse Umrah Package button or directly going to the Package Page to view all of the provided Umrah packages. Then the customer needs to select any of the packages and click the Reserve Now button. In the make reservation interface, customers need to select the travel date and accept the terms and conditions to be able to make an Umrah reservation. Fig. 17 shows the code segment for Make Umrah Reservation and Update Reservation Details. In make umrah reservation code segment, the system will validate the Umrah reservation details provided by the customer. If all of the input is valid, the system will store the Umrah reservation details in the database, and the system will redirect customers to the Umrah Reservation List Page. In update reservation details code segment, the system will validate the input details provided by the administrator and staff. If all of the input is valid, the system will update the Umrah reservation details in the database, and the system will redirect administrators and staff to the Umrah Reservation List Page.



(a) **(b)**

Fig. 16 *Interface for Umrah Management modules (a) Reservation List; (b) Make Reservation*

```php
/**
 * Submit Umrah reservation form.
 */
public function postUmrahReservation(Request $request, int $packageId, int $priceId, string $room)
{
    $payment = new Payment();
    $reservation = new Reservation();
    $priceData = Price::find($priceId);
    $userId = auth()->user()->id;

    $this->validate($request, [
        'travel_date' => ['required', 'exists:travel_dates,id'],
        'first_name' => ['required', 'string', 'max:255'],
        'last_name' => ['nullable', 'string', 'max:255'],
        'phone_no' => ['nullable', 'string', 'max:255'],
        'terms_and_conditions' => ['accepted'],
    ]);

    $request->user()->update(['first_name' => $request['first_name']]);
    $request->user()->update(['last_name' => $request['last_name']]);
    $request->user()->update(['phone_no' => $request['phone_no']]);

    if ($room == '4 or 5 People') {
        $totalAmount = $priceData->room_4_5;
    } else if ($room == '3 People') {
        $totalAmount = $priceData->room_3;
    } else if ($room == '2 People') {
        $totalAmount = $priceData->room_2;
    }

    $travelDateData = TravelDate::find($request['travel_date']);

    $payment->package = $priceData->package;
    $payment->total_amount = $totalAmount;
    $payment->status = 'Pending';
    $payment->save();

    $reservation->user_id = $userId;
    $reservation->package_id = $packageId;
    $reservation->room = $room;
    $reservation->from_date = $travelDateData->from;
    $reservation->to_date = $travelDateData->to;
    $reservation->status = 'Under Review';
    $reservation->remarks = 'Please wait while our staff reviews your Umrah reservation.';
    $reservation->payment_id = $payment->id;
    $reservation->save();

    return Redirect::route('umrah.reservation-list')->with('status', 'reservation-submitted');
}
```

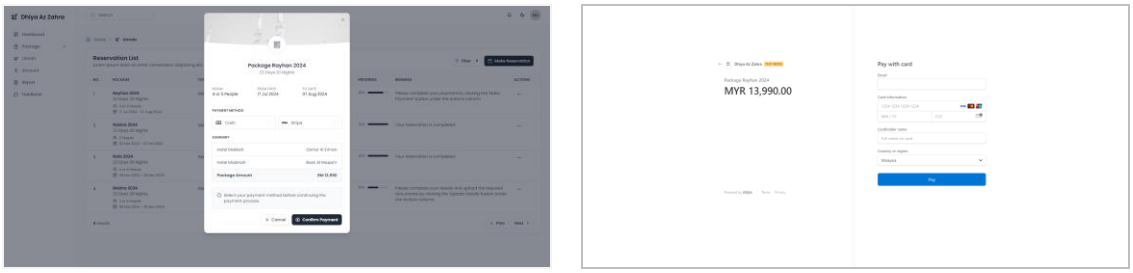
(a)

```php
/**
 * Submit update details form.
 */
public function putUpdateDetails(Request $request, string $id)
{
    $reservationData = Reservation::findOrFail($id);

    $this->validate($request, [
        'identity_card' => ['required', 'mimes:pdf,zip', 'max:10000'],
        'passport' => ['required', 'mimes:pdf,zip', 'max:10000'],
    ]);

    if ($request->hasFile('identity_card') && $request->hasFile('passport')) {
        $identityCard = $request->file('identity_card');
        $passport = $request->file('passport');

        $fileNameIdentityCard = 'identitycard_' . $request->user()->id . '_' . time() . '.' .
$identityCard->getClientOriginalExtension();
        $fileNamePassport = 'passport_' . $request->user()->id . '_' . time() . '.' . $passport-
>getClientOriginalExtension();

        // Ensure the directory exists
        $path = storage_path('app/files/umrah');
        if (!File::isDirectory($path)) {
            File::makeDirectory($path, 0777, true, true);
        }

        Storage::putFileAs('files/umrah', new NewFile($identityCard), $fileNameIdentityCard);
        Storage::putFileAs('files/umrah', new NewFile($passport), $fileNamePassport);

        $reservationData->identity_card = $fileNameIdentityCard;
        $reservationData->passport = $fileNamePassport;
        $reservationData->save();
    }

    return Redirect::route('umrah.reservation-list')->with('status', 'details-updated');
}
```

(b)

Fig. 17 *Code Segment for Umrah Management modules (a) Make Reservation; (b) Update Reservation Details*

The customer can make a reservation payment by clicking the Make Payment button in the Actions column. The system will show a summary of the Umrah reservation, and the customers are required to select the payment method between the Cash and Stripe methods. Fig. 18 shows the Umrah Reservation Summary interface to make the reservation payment and the Stripe Payment Interface. Fig. 19 shows the code segment for the Make Reservation Payment. The system will validate the payment process with the customer. If payment is successful, the system will update the payment status and method in the database, and the system will redirect customers to the Umrah Reservation List Page.



(a) **(b)**

Fig. 18 *Interface for Umrah Management modules (a) Reservation Summary; (b) Stripe Payment*

```php
/**
 * Submit make payment form.
 */
public function patchMakePayment(Request $request, string $id)
{
  $reservationData = Reservation::findOrFail($id);

  $this->validate($request, [
    'payment_method' => ['required', 'string', 'max:255'],
  ]);

  switch ($request['payment_method']) {
    case 'Cash':
      $reservationData->payment()->update([
        'status' => 'Pending',
        'method' => $request['payment_method'],
        'date_paid' => null,
      ]);
      return Redirect::route('umrah.reservation-list')->with('status', 'payment-cash');

    case 'Stripe':
      $reservationData = Reservation::findOrFail($id);
      Stripe::setApiKey(env('STRIPE_SECRET'));
      $session = Session::create([
        'line_items' => [
          [
            'price_data' => [
              'currency' => 'myr',
              'product_data' => [
                'name' => 'Package ' . $reservationData->package->name . ' ' . $reservationData->package->year,
              ],
              'unit_amount' => $reservationData->payment->total_amount * 100,
            ],
            'quantity' => 1,
          ],
        ],
        'mode' => 'payment',
        'success_url' => route('umrah.payment-success', $id),
        'cancel_url' => route('umrah.payment-failure'),
      ]);
      return redirect()->away($session->url);

    default:
      return Redirect::route('umrah.reservation-list')->with('status', 'payment-failure');
  }
}
```

Fig. 19 *Code Segment for Package Management modules; Make Reservation Payment*

4.6.5 Account Management Module

In this module, administrators, staff, and customers can update their account information, change their passwords, and delete their accounts. Fig. 20 shows the account management interface. To update the account information, administrators, staff, and customers need to provide their account information, and to change their account password, they need to provide the current password and new password to the system. Fig. 21 shows the code segment for the update account information and change account password. In update account information code segment, the system will validate the account information provided by the administrator, staff, and customers. If all of the input is valid, the system will update the account information in the database. In change password code segment, the system will validate the current password and new password provided by the administrator, staff, and customers. If the current password and new password are valid, the system will update the account password in the database.

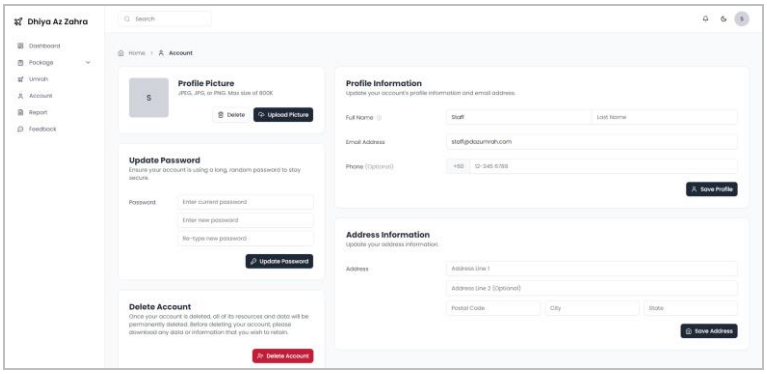


Fig. 20 *Interface for Account Management modules; Account Management*

```php
/**
 * Submit profile information form.
 */
public function patchProfileInformation(ProfileUpdateRequest $request): RedirectResponse
{
  $request->user()->fill($request->validated());

  if ($request->user()->isDirty('email')) {
    $request->user()->email_verified_at = null;
  }

  $request->user()->save();

  return Redirect::route('profile')->with('status', 'profile-updated');
}
```

```php
/**
 * Update the user's password.
 */
public function update(Request $request): RedirectResponse
{
  $validated = $request->validateWithBag('updatePassword', [
    'current_password' => ['required', 'current_password'],
    'password' => ['required', Password::defaults(), 'confirmed'],
  ]);

  $request->user()->update([
    'password' => Hash::make($validated['password']),
  ]);

  return back()->with('status', 'password-updated');
}
```

(a) **(b)**

Fig. 21 *Code Segment for Account Management modules (a) Update Account Information; (b) Change Password*

4.6.6 Tracking Management Module

In this module, staff can track the current customer location during Umrah. This is to ensure the safety of the customer during Umrah. In order to view the current customer location, staff need to go to the dashboard page, and the system will show the current customer location on a map. Fig. 22 shows the tracking management interface while Fig. 23 shows the store GPS and update customer location code segment. In store GPS code segment, the system will receive the current customer location from the GPS hardware provided to the customer. Then the system will store the current customer location in the database. In the update customer location code segment, the system will update and display the latest customer location on the map.

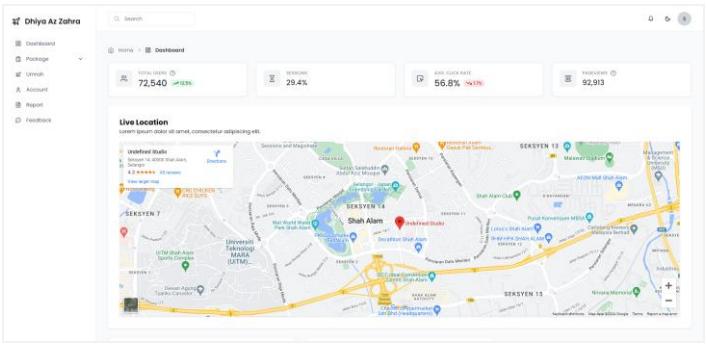


Fig. 22 *Interface for Tracking Management modules; Dashboard*

```php
/**
 * Store gps location.
 */
public function storeGPS(Request $request)
{
    $request->validate([
        'latitude' => ['required', 'numeric'],
        'longitude' => ['required', 'numeric'],
    ]);

    $gpsData = new GPS();
    $gpsData->latitude = $request->latitude;
    $gpsData->longitude = $request->longitude;
    $gpsData->save();

    return response()->json(['message' => 'GPS data saved successfully'], 200);
}
```

(a)

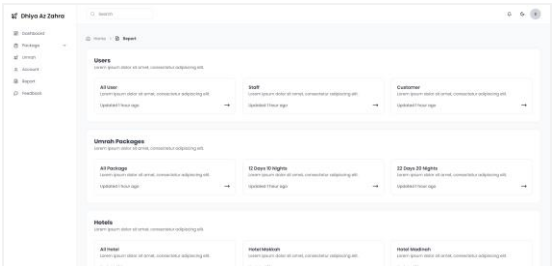
```php
/**
 * Get latest gps location.
 */
public function getLatestGPS()
{
    // Fetch the latest GPS data from the database
    $latestGPS = GPS::latest()->first();

    // Return the latest GPS data as JSON response
    return response()->json($latestGPS);
}
```

(b)

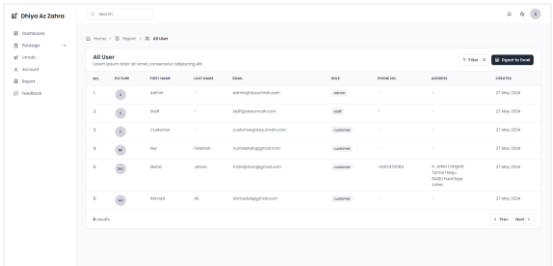
Fig. 23 *Code Segment for Tracking Management modules (a) Store GPS; (b) Update Customer Location*

4.6.7 Generate Report Module

In this module, administrators and staff can convert the report to Excel. Fig. 24 shows interface for generate report modules, which will show all the available reports. To convert the report to Excel, administrators and staff need to click the Convert to Excel button. Fig. 25 shows the exported report in Excel file while Fig. 26 shows the export report to an Excel code segment. The system will export all the report data to a downloadable Excel file, and administrators and staff can download the exported file to their devices.



(a)



(b)

Fig. 24 *Interface for Generate Report modules (a) All Report; (b) Export Report to Excel*

#	First Name	Last Name	Email	Role	Phone No.	Address Line 1	Address Line 2	Postal Code	City	State	Created
1	Admin		admin@dazumrah.com	admin							2024-05-27 02:07:03
2	Staff		staff@dazumrah.com	staff							2024-05-27 02:07:03
3	Customer		customer@dazumrah.com	customer							2024-05-27 02:07:03
4	Nur	Fatehah	nurfatehah@gmail.com	customer							2024-05-27 02:07:03
5	Mohd	Johan	mohdjohan@gmail.com	customer	174701159	4, Jalan Langsat	Taman Maju	81400	Parit Raja	Johor	2024-05-27 02:07:03
6	Ahmad	Ali	ahmadali@gmail.com	customer							2024-05-27 02:07:03

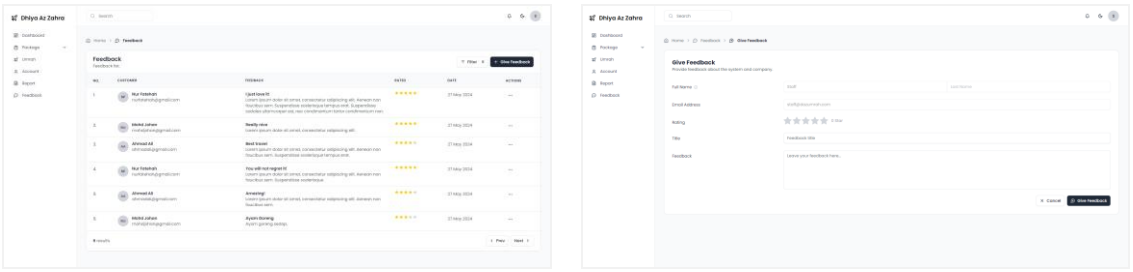
Fig. 25 *Exported Report in Excel File; All User*

```php
/**
 * Export all user data.
 */
public function exportAllUser()
{
  return Excel::download(new AllUserExport, 'all-user.xlsx');
}
```

Fig. 26 *Code Segment for Generate Report modules; Export All User*

4.6.8 Feedback Management Module

In this module, customers can give feedback, while administrators and staff can manage the feedback from the customers. Fig. 27 shows the interface for feedback management modules, which will list out all the feedback that has been made by the customer. Customers can give feedback by clicking the Give Feedback button. The system will redirect the customers to the Give Feedback page, and the customers need to provide the feedback details to be able to submit the feedback. Fig. 28 shows the give feedback code segment. The system will validate the feedback details provided by the customer. If all of the input is valid, the system will store the feedback details in the database, and the system will redirect customers to the Feedback List page.



(a) **(b)**

Fig. 27 *Interface for Feedback Management modules (a) Feedback Management; (b) Give Feedback*

```php
/**
 * Submit give feedback form.
 */
public function postGiveFeedback(Request $request)
{
  $feedback = new Feedback();
  $userId = auth()->user()->id;

  $this->validate($request, [
    'rating' => ['required', 'integer', 'min:0'],
    'title' => ['required', 'string', 'max:255'],
    'details' => ['required', 'string', 'max:255'],
  ]);

  $feedback->user_id = $userId;
  $feedback->rating = $request['rating'];
  $feedback->title = $request['title'];
  $feedback->details = $request['details'];
  $feedback->save();

  return Redirect::route('feedback')->with('status', 'feedback-submitted');
}
```

Fig. 28 *Code Segment for Feedback Management modules; Give Feedback*

4.7 Testing

The testing phase focuses on assessing the functionality of the system. This section involves two different testing methods, which are system testing and user acceptance testing.

4.7.1 System Testing

System testing is conducted to ensure that the developed system meets both functional and non-functional requirements. The completion of system testing is determined by the proper functionality of all modules, aligned with the specified functional and non-functional requirements. The list of test cases and results is presented in Table 5.

Table 5 *List of test cases*

Test Case ID	Requirement ID	Description	Test Status
TC_100	**REQ_100**	**Customer Registration**	

Table 5: *(cont).*

TC_100_01	REQ_101	The system shall be able to allow customers to register an account.	PASS
TC_100_02	REQ_102	The system shall be able to allow customers to input the registration details in the system.	PASS
TC_100_03	REQ_103	The system shall give choice to the customers to accept or decline the terms and conditions of the system.	PASS
TC_100_04	REQ_104	The system shall allow customers to submit their registration details.	PASS
TC_100_05	REQ_105	The system shall be able to verify the customer registration details.	PASS
TC_100_06	REQ_106	The system shall be able to store the customer details in the database.	PASS
TC_200	**REQ_200**	**Login**	
TC_200_01	REQ_201	The system shall be able to allow users to login to the system.	PASS
TC_200_02	REQ_202	The system shall be able to allow users to input the login details in the system.	PASS
TC_200_03	REQ_203	The system shall allow users to submit their login details.	PASS
TC_200_04	REQ_204	The system shall be able to verify the user login details.	PASS
TC_200_05	REQ_205	The system shall be able to allow users to recover their account password.	PASS
TC_200_06	REQ_206	The system shall be able to allow users to input their email address associated with their account.	PASS
TC_200_07	REQ_207	The system shall allow users to submit their reset password details.	PASS
TC_200_08	REQ_208	The system shall be able to verify the reset password email address.	PASS
TC_300	REQ_300	Package Management	
TC_300_01	REQ_301	The system shall be able to allow staff to create, edit, and delete Umrah package.	PASS
TC_300_02	REQ_302	The system shall be able to allow staff to input the Umrah package details.	PASS
TC_300_03	REQ_303	The system shall allow staff to submit the Umrah package details.	PASS
TC_300_04	REQ_304	The system shall be able to verify the Umrah package details.	PASS
TC_300_05	REQ_305	The system shall be able to store the Umrah package details in the database.	PASS
TC_300_06	REQ_306	The system shall be able to update the Umrah package details in the database.	PASS
TC_300_07	REQ_307	The system shall be able to remove the Umrah package from the database.	PASS
TC_400	**REQ_400**	**Umrah Management**	
TC_400_01	REQ_401	The system shall be able to allow staff to update and delete customer reservation.	PASS
TC_400_02	REQ_402	The system shall be able to allow staff to choose the reservation status.	PASS

Table 5: *(cont).*

TC_400_03	REQ_403	The system shall be able to allow customers to make Umrah reservation.	PASS
TC_400_04	REQ_404	The system shall allow customers to make cash or transfer payment for the Umrah reservation.	PASS
TC_400_05	REQ_405	The system shall be able to allow customers to input the Umrah reservation details.	PASS
TC_400_06	REQ_406	The system shall be able to allow customers to input the reservation payment details for transfer payment method.	PASS
TC_400_07	REQ_407	The system shall allow customers to submit the Umrah reservation.	PASS
TC_400_08	REQ_408	The system shall allow customers to submit the reservation payment.	PASS
TC_400_09	REQ_409	The system shall be able to verify the Umrah reservation details.	PASS
TC_400_10	REQ_410	The system shall be able to verify the reservation payment details.	PASS
TC_400_11	REQ_411	The system shall be able to update the customer reservation status in the database.	PASS
TC_400_12	REQ_412	The system shall be able to remove the customer reservation from the database.	PASS
TC_400_13	REQ_413	The system shall be able to store the customer reservation in the database.	PASS
TC_400_14	REQ_414	The system shall be able to update the customer reservation payment details in the database.	PASS
TC_500	**REQ_500**	**Account Management**	
TC_500_01	REQ_501	The system shall be able to allow users to update their profile details.	PASS
TC_500_02	REQ_502	The system shall be able to allow users to change their account password.	PASS
TC_500_03	REQ_503	The system shall allow users to submit the updated profile details.	PASS
TC_500_04	REQ_504	The system shall allow users to submit the change account password.	PASS
TC_500_05	REQ_505	The system shall be able to verify the updated profile details.	PASS
TC_500_06	REQ_506	The system shall be able to verify the change account password input.	PASS
TC_500_07	REQ_507	The system shall be able to allow administrator to create staff account.	PASS
TC_500_08	REQ_508	The system shall be able to allow administrator to delete staff account.	PASS
TC_500_09	REQ_509	The system shall allow administrator to submit the create staff account.	PASS
TC_500_10	REQ_510	The system shall allow administrator to confim to delete the staff account.	PASS
TC_500_11	REQ_511	The system shall be able to verify the new staff account.	PASS
TC_500_12	REQ_512	The system shall be able to update the profile details in the database.	PASS

Table 5: *(cont).*

TC_500_13	REQ_513	The system shall be able to update the user password in the database.	PASS
TC_500_14	REQ_514	The system shall be able to store the staff details in the database.	PASS
TC_500_15	REQ_515	The system shall be able to remove the staff from the database.	PASS
TC_600	**REQ_600**	**Tracking Management**	
TC_600_01	REQ_601	The system shall be able to allow staff to track the current customer location during Umrah.	PASS
TC_600_02	REQ_602	The system shall be able to receive the current customer location during Umrah.	PASS
TC_600_03	REQ_603	The system shall be able to retrieve the current customer location from the database.	PASS
TC_600_04	REQ_604	The system shall be able to store the current customer location in the database.	PASS
TC_700	**REQ_700**	**Generate Report**	
TC_700_01	REQ_701	The system shall be able to allow users to convert the selected report to PDF file.	PASS
TC_700_02	REQ_702	The system shall be able to create a downloadable PDF report file.	PASS
TC_800	**REQ_800**	**Feedback Management**	
TC_800_01	REQ_801	The system shall be able to allow staff to reply and delete customer feedback.	PASS
TC_800_02	REQ_802	The system shall be able to allow customer to give and delete the feedback.	PASS
TC_800_03	REQ_803	The system shall allow staff to input the feedback reply.	PASS

4.7.2 Overall Result

In this section, the summary of the results for the system testing will be shown and discussed. There are a total of eight modules with sixty seven test cases that have been carried out to test the system. Table 6 shows the overall result of test cases.

Table 6 *Overall result of test cases*

Test Case ID	Total Test Cases	Total Success	Total Failed
TC_100	6	6	-
TC_200	8	8	-
TC_300	7	7	-
TC_400	14	14	-
TC_500	15	15	-
TC_600	4	4	-
TC_700	2	2	-
TC_800	11	11	-
Total	67	67	-

4.7.3 User Acceptance Testing

User Acceptance Testing (UAT) is a crucial stage in software engineering where the software is evaluated from the standpoint of end-users and stakeholders. Its primary goal is to assess whether the software aligns with user expectations, fulfills the specified requirements, and performs effectively. The UAT form is attached in Appendix

A. Fig. 29 shows the results of the UAT for Section A: Users Satisfaction on System Functionality, indicating very positive feedback. Most users strongly agree that the Dhiya Az Zahra Umrah Management System works well, meets their needs, reduces mistakes in data entry, makes Umrah reservation easier, and completes tasks quickly without major delays. In general, users are very happy with how the system functions.

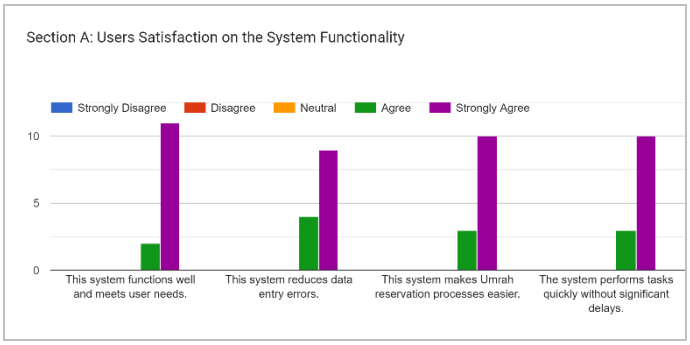


Fig. 29 *Users Satisfaction on the System Functionality*

Fig. 30 shows the UAT for Section B: Users Satisfaction on the System Usability, indicating a highly positive reception. Most users strongly agree that the system is easy to understand and use, with many users finding it simple to navigate to other pages. The majority of users also strongly agree that the information provided on the website is sufficient. Additionally, most users strongly agree that they can complete tasks without needing extensive support. Overall, users are very satisfied with the system's usability.

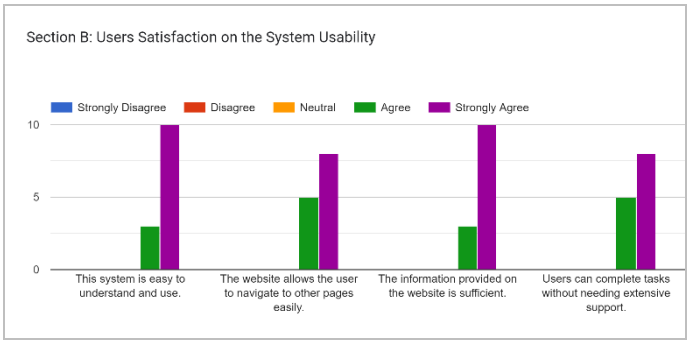


Fig. 30 *Users Satisfaction on the System Usability*

Fig. 31 shows the UAT for Section C: Users Satisfaction on the System User Interface Design, indicating a highly positive reception. Most users strongly agree that the system uses themes and colors consistently, making the user interface attractive. The majority also strongly agree that the size and type of fonts are easy to read, and the layout of the system is clear and well-organized. Overall, users are very satisfied with the design of the system's user interface.

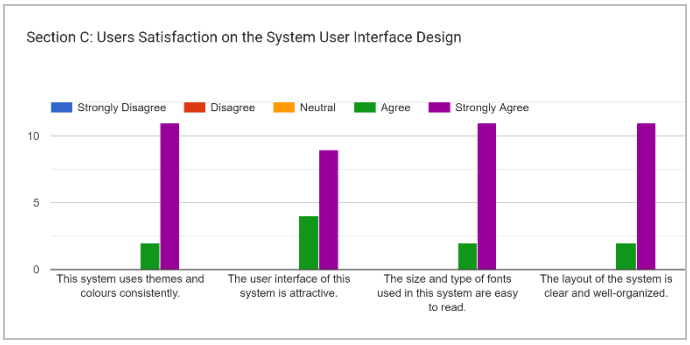


Fig. 31 *Users Satisfaction on the System User Interface Design*

Fig. 32 shows the additional comments, suggestions, and feedback on the Dhiya Az Zahra Umrah Management System. Overall, the responses are very positive. Users like the attractive and neat website, finding it easy to understand and navigate. The system is user-friendly, efficient, and much better than the old way of making Umrah reservations. Some users suggest adding a multi-language feature and providing tooltips to help new users. Overall, people are very happy with the system design and how well it works.

Fig. 32 *Comments, Suggestions, and Feedback about the Dhiya Az Zahra Umrah Management System*

5. Conclusion

The Dhiya Az Zahra Umrah Management System represents a significant advancement in the management and organization of Umrah services, effectively addressing the needs of administrators, staff, and customers. The system has successfully achieved its primary objectives, delivering a robust platform that enhances efficiency, security, and customer satisfaction in the Umrah registration and management process.

Key advantages of the system include secure user authentication, flexible management of Umrah packages, streamlined reservation processes, and comprehensive account management features. Additionally, the ability to track customer locations during Umrah and generate detailed reports contributes to a safer and more informed service delivery. The system also encourages continuous improvement through customer feedback and review management.

However, the system does have areas that require further development. The limitations in payment methods, the need for a more engaging user interface, and the lack of multilingual support highlight opportunities for enhancement. Addressing these issues will make the system more accessible and user-friendly, thereby broadening its appeal and usability.

In conclusion, the Dhiya Az Zahra Umrah Management System marks a substantial improvement in the administration of Umrah services. While it already offers numerous benefits, future improvements will further enhance its functionality and user experience. This system is poised to greatly enhance the efficiency and effectiveness of Umrah management, making the process more convenient and satisfactory for all users involved.

Acknowledgement

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

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

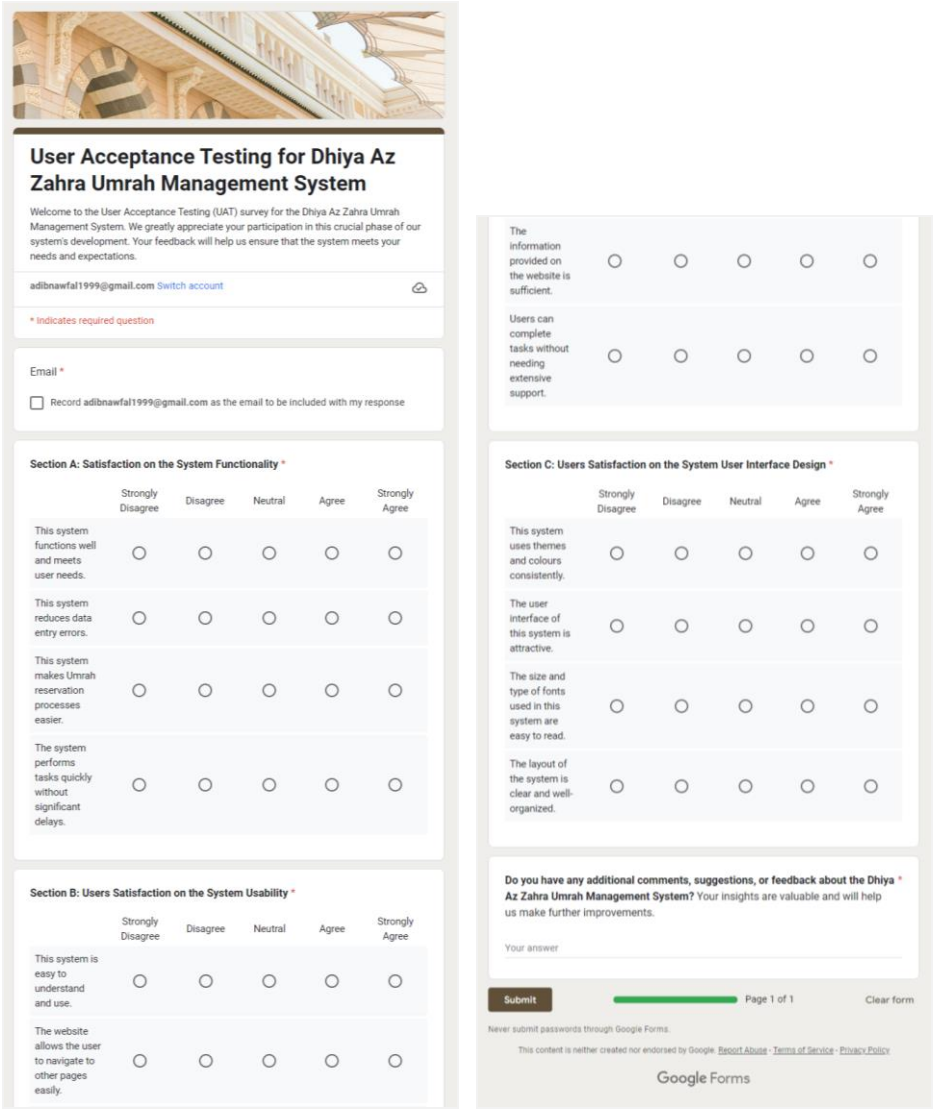
The authors confirm contribution to the paper as follows: **study conception and design:** Adib Nawfal Yazid, Rabatul Aduni Sulaiman; **data collection:** Adib Nawfal Yazid, Rabatul Aduni Sulaiman; **analysis and interpretation of results:** Adib Nawfal Yazid, Rabatul Aduni Sulaiman; **draft manuscript preparation:** Adib Nawfal Yazid, Rabatul Aduni Sulaiman. All authors reviewed the results and approved the final version of the manuscript.

References

[1] Bashir, H., Beigh, R., Rumysa, Bashir, S., Bazaz, A., Beigh, A., & Farooq, R. (2017). Experience of Umrah Trip to Saudi Arabia from Kashmir, India: Model Study on Management and Facilities to Pilgrims by Custodian of

Holy Mosques Kingdom of Saudi Arabia. *Journal of Tourism & Hospitality, 06*(06). https://doi.org/10.4172/2167-0269.1000328
[2] Murad, M. (2006). *The Book of Hajj & Umrah*. Rabwah: The Islamic Propagation Office.
[3] Bernama. (2023). Saudi Arabia reScords arrival of over 300,000 Malaysian Umrah pilgrims in just 8 months. *New Straits Times*. https://api.nst.com.my/news/nation/2023/03/885188/saudi-arabia-records-arrival-over-300000-malaysian-umrah-pilgrims-just-8
[4] Chan, K. L. G., Selvadurai, S., Dato, B., & Nurdin, G. (2022). Women Online Entrepreneurship and Social Environment in Malaysia: Some Preliminary Findings.
[5] Seo, H.-J., Lee, Y., Hur, J.-J., & Kim, J. (2012). The impact of information and communication technology on skilled labor and organization types. Information Systems Frontiers, 14. https://doi.org/10.1007/s10796-010-9276-7
[6] Mahmad, M. A., Ismail, N. H., Rofie, K. H., Ahmad, S., & Islieh, A. R. I. S. (2021). Handling Of Umrah Packages In Kedah: An Early Review. *Journal of Tianjin University Science and Technology, 54*(12), 1-9. https://oarep.usim.edu.my/jspui/handle/123456789/14247
[7] AndalusiaKu. (n.d.). *Andalusia Travel & Tours Sdn Bhd.* Retrieved November 18, 2023, from https://www.umrahandalusia.com/
[8] Azzuha Group Travel & Tours. (n.d.). *Az Zuha Group Travel & Tours Sdn Bhd.* Retrieved November 19, 2023, from https://www.azzuhatravel.com/
[9] Ikhlas Umrah. (n.d.). *Ikhlas Com Travel Sdn Bhd.* Retrieved November 19, 2023, from https://www.ikhlas.com/umrah/ms/my/
[10] Shylesh, S. (2017). A Study of Software Development Life Cycle Process Models. *Social Science Research Network.* http://dx.doi.org/10.2139/ssrn.2988291

Appendix A: User Acceptance Testing (UAT) Form



User Acceptance Testing for Dhiya Az Zahra Umrah Management System

Welcome to the User Acceptance Testing (UAT) survey for the Dhiya Az Zahra Umrah Management System. We greatly appreciate your participation in this crucial phase of our system's development. Your feedback will help us ensure that the system meets your needs and expectations.

adibnawfal1999@gmail.com Switch account

* Indicates required question

Email *

Record adibnawfal1999@gmail.com as the email to be included with my response

Section A: Satisfaction on the System Functionality *

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
This system functions well and meets user needs.					
This system reduces data entry errors.					
This system makes Umrah reservation processes easier.					
The system performs tasks quickly without significant delays.					

Section B: Users Satisfaction on the System Usability *

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
This system is easy to understand and use.					
The website allows the user to navigate to other pages easily.					
The information provided on the website is sufficient.					
Users can complete tasks without needing extensive support.					

Section C: Users Satisfaction on the System User Interface Design *

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
This system uses themes and colours consistently.					
The user interface of this system is attractive.					
The size and type of fonts used in this system are easy to read.					
The layout of the system is clear and well-organized.					

Do you have any additional comments, suggestions, or feedback about the Dhiya Az Zahra Umrah Management System? Your insights are valuable and will help us make further improvements.

Your answer

Submit Page 1 of 1 Clear form

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. Report Abuse - Terms of Service - Privacy Policy

Google Forms