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

AITCS

Vol. 6 No. 1 (2025) 861-880
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

DriveEasy: Car Rental Management System

Tiviyarsiri A/P Gunasekaran[1], Siti Hajar Arbain[1]*

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

*Corresponding Author: sitihajara@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2025.06.01.048

Article Info

Received: 15 June 2024
Accepted: 19 June 2025
Available online: 30 June 2025

Keywords

booking system, online, car rental

Abstract

DriveEasy: Car Rental Management System aims to transform car rental operations and overcome the challenges of manual processes observed at Nadim Car Rental in Batu Pahat, Malaysia. The current system depends on inefficient paperwork, leading to errors, delays, and customer dissatisfaction. The primary objective of this study is to develop a web-based using an object-oriented approach and test it in alpha phases. The scope focuses on the specific needs of Nadim Car Rental with the involvement of internal stakeholders. This system is focused on different user roles and optimizes customer data, reservations, and maintenance. The methodology used to implement the car rental management system is the Rapid Application Development (RAD) model, encompassing phases such as Analysis, Quick Design, Prototype Cycles, Testing, and Implementation. The main results are improved operational efficiency, cost reductions, and a better customer experience. This abstract establishes a foundation for future innovations and improvements, suggesting a significant shift in car rental management practices.

1. Introduction

The car rental industry is an important aspect of the transportation sector, providing customers and business owners with short-term transportation alternatives. A car rental reservation system is a software solution that allows customers to reserve rental cars and make reservations online or offline. This project focuses on the development of a car rental management system to modernize and optimize the operations of Nadim Car Rental in Batu Pahat, Johor.

The current process relies on informal methods, especially the use of social media platforms such as WhatsApp to facilitate customer bookings. Ms. Zahrah, the owner, and Mr. Suhaimi, a single employee, manage the entire operation. The system is paper based, with manual forms and documents recording reservations and customer details. In addition, customers can either make their reservations in person at the rental station or speak directly to staff over the phone. This existing process has its limitations in terms of efficiency, data organization, and customer friendliness. The proposed car rental management system solves these problems by introducing a more structured approach to car rental operations.

The objective of this study is to design a car rental management system using an object-oriented approach and perform alpha testing. The study focuses on Nadim Car Rental in Batu Pahat, involving internal stakeholders including customers, administrative staff, and the owner. System modules of the Car Rental Management System, such as registration and login, vehicle inventory, booking and reservation, payment, reporting, and fleet maintenance, have been developed to streamline various functions. The expected results include greater operational efficiency, better customer experience, and a competitive advantage in the car rental business for

Nadim Car Rental. The significance of the project lies in its ability to improve operational efficiency, cost reduction, and overall service quality, not only for Nadim Car Rental but also for other companies facing similar issues in the market.

This paper is structured as follows: Section 2 reviews related work and existing systems; Section 3 details the methodology, specifically the Rapid Application Development (RAD) model used; Section 4 provides the analysis and design of the system; Section 5 discusses the implementation and testing phases; Section 6 presents the results and discussions; and Section 7 concludes the paper with suggestions for future work.

2. Related Work

A literature review is a method of collecting information from scholarly sources such as books, journals, articles, newspapers, and other sources appropriate for this study. The information collected is usually related to the title of the study being conducted.

2.1 Case Study: Car Rental Management System

The car rental industry operates both locally and globally [1]. Private companies are known as car rental companies and provide vehicles to their customers for a specific time with a fee. Car rental companies are very popular in Malaysia, especially among students on campuses and colleges, as the rented car is an affordable alternative for those who do not own a car and find public transportation inconvenient [2].

According to the statistics, the number of car rental companies and rented cars has significantly increased. The transition from traditional to digital car rental systems is evident and the number of car rental companies and rented cars has significantly increased. The existing research highlights the importance of car rental services in the tourism industry, influencing both international market capacity and tourist satisfaction [3].

This study aims to address the gaps and limitations of previous studies by analyzing the implementation of a DriveEasy: Car Rental Management System for Nadim Car Rental in Batu Pahat, Malaysia. This research aims to fill up the gaps and limitations found in previous studies. This research also provides valuable insights into the obstacles specific to the Malaysian context. Additionally, it will provide suggestions for improving the online car rental management system. This will eventually increase Nadim Car Rental's efficiency and contribute to a broader understanding of digital systems in the vehicle sector.

The central research question conducted in this study is: "How can the implementation of a Car Rental Management System enhance operational efficiency and customer satisfaction for Nadim Car Rental in Batu Pahat, Malaysia?" This investigation will explore the specific needs and challenges faced by Nadim Car Rental, offering solutions for the successful integration of an online system.

2.2 Technology

The "DriveEasy: Car Rental Management System" is a web-based application built using Visual Studio, HTML, CSS, JavaScript, PHP, and MySQL technological stack. The integrated programming environment is Visual Studio, and HTML, CSS, and JavaScript ensure a responsive and visually appealing user interface. PHP, a server-side scripting language, allows for dynamic content to be updated and interacted with in real time. MySQL offers reliable and structured data storage.

2.3 Study of Existing Related Systems

The study was conducted on three existing related systems on the background study. The three existing related systems are Wagons Car Rental Services [4], GoBig Travel & Tours [5], and ETT Car Travel [6]. In addition, this study was conducted to collect important information for the proposed system, named DriveEasy: Car Rental Management System.

2.4 Comparison with the Existing Systems

Based on the studies and comparisons that have been done, there are similarities and differences between the equivalent system and the proposed system. This comparison is made by evaluating the features found in the equivalent system and the proposed system. The features include the Registration and Login Module, Vehicle Inventory Module, Booking and Reservation Module, Payment Module, Report Module, Fleet Maintenance Module, and the system type.

Table 1 *Comparison of Features of the Existing Car System and the Proposed System*

Applied Information Technology and Computer Science Vol. 6 No. 1 (2025) p. 861-880

Features/System	Wagons Car Rental Services	GoBig Travel & Tours	ETT Car Rental System	DriveEasy: Car Rental Management System
Registration and Login Module	Yes	No	No	Yes
Vehicle Inventory Module	Yes	Yes	Yes	Yes
Booking and Reservation Module	Yes	Yes	Yes	Yes
Payment Module	Yes	Yes	Yes	Yes
Report Module	No	No	No	Yes
Fleet Maintenance Module	No	No	No	Yes
System Type	Web-based Application	Web-based Application	Web-based Application	Web-based Application

The existing car rental systems such as Wagons Car Rental Services, GoBig Travel & Tours, and ETT Car Rental share common features to increase operational efficiency and customer satisfaction. Wagons Car Rental Services provides modules for vehicle and customer registration, real-time reservations, and secure payment transactions. Similarly, ETT Car Rental offers a Vehicle Inventory module, a Reservation module for smooth booking processes, and a Payment module for convenient transactions. Then, GoBig Travel & Tours offers an online booking platform, traditional reservation methods, and a flexible payment system. The "DriveEasy: Car Rental Management System" combines and extends features from existing systems such as Wagons Car Rental Services, ETT Car Rental, and GoBig Travel & Tours to optimize the car rental process. In conclusion, the "DriveEasy: Car Rental Management System" expands on the strengths of existing systems, incorporating the functionalities with new features to optimize the car rental process, enhance user satisfaction, and maintain market competitiveness.

3. Methodology

Methodology is the intellectual framework within which research is carried out or the foundation upon which research is built [7]. A methodology is required since it assists others in understanding what this project or research is about, its significance and limitations, as well as its strengths. Rapid Application Development (RAD) is a software development process that aims to quickly create high-quality systems at lower costs compared to traditional methods. It compresses analysis, design, build, and testing into short iterative cycles, prioritizing prototyping, user feedback, and teamwork. RAD is flexible and adaptive to changing business needs, delivering the product incrementally. It integrates users throughout the process to ensure real-world needs are met [8].

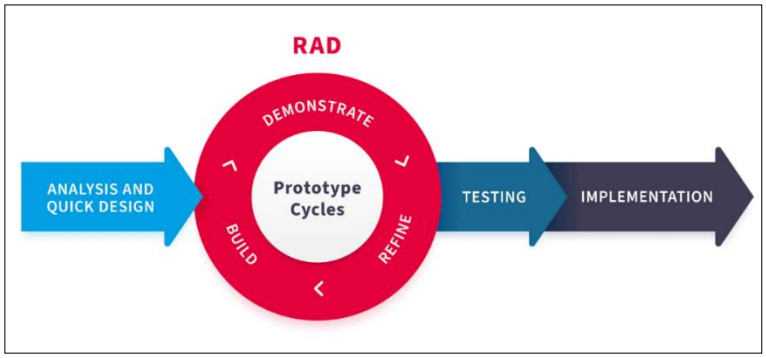


Fig. 1 *Rapid Application Development Process Model*

3.1 Analysis and Quick Design Phase

During the Analysis Phase of developing the DriveEasy Car Rental Management System, tasks include defining requirements and interfaces, conducting stakeholder interviews, analyzing business processes, and gathering system requirements through surveys and observations at Nadim Car Rental. Ms. Zahrah and Mr. Suhaimi collaborated to understand the needs of the rental company. The Quick Design phase uses gathered requirements to design the system architecture using wireframing, prototyping, and UML diagrams. Diagrams are created to represent system interactions, workflow, processes, and structure. A Data Dictionary is also developed to specify data items and connections. This phase lays the foundation for the project's following stages.

3.2 Prototype Cycles Phase

The prototype Cycle phase is iterative, focusing on developing, demonstrating, and refining prototypes until the system meets the stakeholders' requirements. This phase involves the following sub-activities:

3.2.1 Develop Phase

In the Development phase of DriveEasy: Car Rental Management System, the high-level design is converted into code, with sub-activities like detailed design, unit testing, integration, and coding. In the first cycle, core components like Registration and Login are developed. The second cycle focuses on Booking, and Payment modules. The third cycle includes Fleet Maintenance and Reporting modules, managing maintenance schedules, and generating reports like booking summaries. Technologies like PHP, HTML, CSS, JavaScript, and MySQL are used in Visual Studio Code for development.

3.2.2 Demonstrate Phase

During the demonstration phase of RAD, the built prototype is shown to stakeholders, specifically end users, for feedback and validation. Activities include demos, walkthroughs, and gathering user feedback. User interviews provide qualitative insights into the system. Demonstrations help stakeholders understand system functionality, usability, and effectiveness. In Cycle 1, the Registration, Login, and Vehicle Inventory modules are demonstrated. Cycle 2 includes Booking, Reservation, and Payment modules, presented to stakeholders for feedback on integration and satisfaction, while Cycle 3 ends with a review of new Reporting and Fleet Maintenance modules, gathering stakeholder feedback.

3.2.3 Refine Phase

During the RAD process for the DriveEasy: Car Rental Management System, the Refine phase is crucial for adjusting the program based on user input from demonstrations. Changes are made to improve system operations, address flaws, and update the database. Cycle 1 focuses on refining the Registration, Login, and Vehicle Inventory modules. Cycle 2 enhances Booking, Reservation, and Payment modules while integrating new features with existing ones. Cycle 3 finalizes improvements to Reporting and Fleet Maintenance, and resolves all remaining issues, ensuring prototypes represent the final system accurately.

3.3 Testing Phase

During the Testing phase, executable code modules are tested independently, and integrated with external components, and defects are identified and fixed before deployment. Sub-activities include unit, system, and acceptance testing. Internal testing corrects faults, with beta testing involving Nadim Car Rental employees for feedback. The phase includes functional, performance, security, and usability testing to ensure a high-quality car rental management system.

3.4 Implementation Phase

During the implementation phase, software is installed in the client's environment after the requirements have been verified. Goals include installing, configuring, and making the system available to end users. User training, data migration, and system installation are key activities. User training sessions are held for Nadim Car Rental employees, along with data migration and system setup. User acceptance testing ensures the system meets expectations for successful live operation.

4. Analysis and Design

This section will explain the analysis and design of the DriveEasy: Car Rental System based on a web-based application. All the diagrams are built based on object-oriented.

4.1 System Requirement Analysis

In system requirements analysis for the Car Rental Management System, there are two categories of requirements which are functional requirements and non-functional requirements. The functional and non-functional requirements of the DriveEasy: Car Rental Management System are listed in Table 2 and Table 3.

Table 2 *Functional Requirements for the System*

Functional Requirements	Description	Users
Registration and Login Module	Users, including customers and staff, should be able to register for an account. The system must authenticate users through a login process.	Customers, Staff, Owner
Vehicle Inventory Module	Users should be able to view vehicle details and check availability. Staff should be able to add, modify, and remove the vehicle information.	Customers, Staff, Owner
Booking and Reservation Module	Customers should be able to book vehicles for specific dates and times.	Customers
Payment Module	Customers should be able to book vehicles for specific dates and times.	Customers
Report Module	Staff and owners should have access to reporting tools for monitoring system performance.	Staff, Owner
Fleet Maintenance Module	Staff and owners should have access to reporting tools for monitoring system performance and generating financial reports.	Staff, Owner

Table 3 *Non-functional Requirements for the System*

Non-functional Requirements	Description
Performance	The system must be responsive and efficient, ensuring quick response times for user interactions
Usability	The system should be user-friendly. Navigation and interaction should be straightforward
Operation	The system can be used only in web-based. The system will have a database to store all the data

4.2 User Requirements Analysis

The user requirements are explained in Table 4. The users of the system consist of customers, staff, and owners.

Table 4 *User Requirements Analysis*

User Requirements
Customers should be able to input their username and password to log into the system.
There should be an option for new customers to register.
Customers should be able to view detailed information about each car (e.g., images, model, type, price).
Customers should be able to reserve a car for specific dates.
Customers should be able to make payments via cash or QR code payment methods.
Customers should be able to update their personal information (name, contact details, address, etc.).

Table 4 *User Requirements Analysis (continued)*

Customers should be able to access support via chat, email, or phone.
Customers should be able to log out of the system.
Customers should be able to input their username and password to log into the system.
Staff and owner should be able to log into the system with their username and password.
Staff and owners should be able to add, update, and delete vehicles in the system.
Staff and owners should be able to view and manage all customer reservations.
Staff and owners should be able to confirm and modify reservations.
Staff and owners should be able to manage customer information.
Staff and owners should be able to record and update maintenance logs for each vehicle.
Staff and owners should be able to record incoming and outgoing financial transactions.
Staff and owners should be able to generate reports related to income, expense, and booking.
Staff and owners should be able to log out of the system.

4.3 System Analysis

Systems analysis is the process of studying a procedure to identify its goals and purposes and create systems and procedures that will efficiently achieve them.

The Use Case Diagram for DriveEasy: Car Rental Management System shown in Fig. 2, shows interactions among Customers, Staff, and Owners with eight key use cases. The system includes use cases such as registration, login, booking and reservation, vehicle information management, booking management, payment, fleet maintenance, and report generation. Registration allows customers to create accounts, while Login provides access for customers, owners, and staff. Book and Reserve allows customers to browse and reserve vehicles. Managing Vehicle Information allows employees and owners to update details. Managing Booking helps you change or cancel reservations. Payment monitors financial activity. Fleet Maintenance tracks maintenance tasks. Generate Report creates detailed reports for analysis.

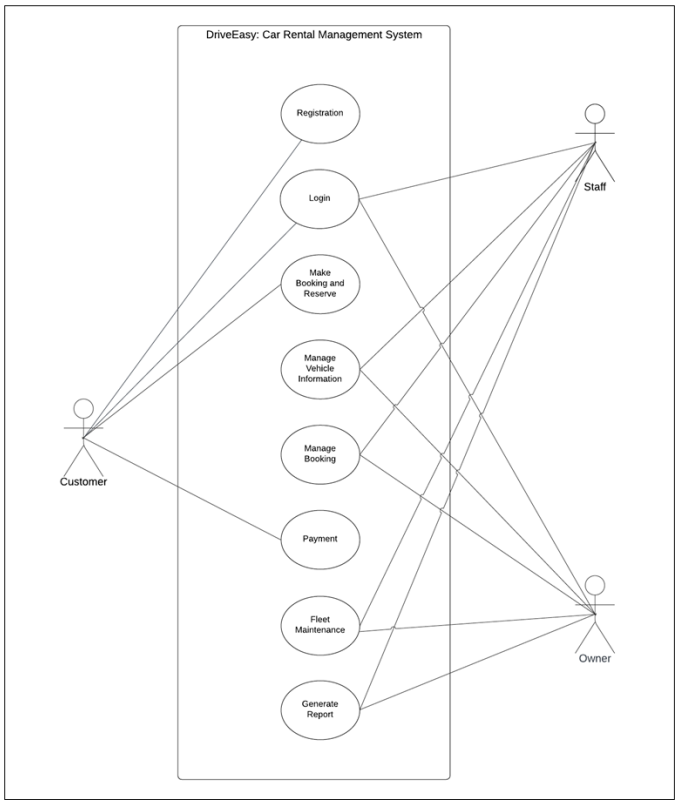


Fig. 2 *Use Case Diagram for DriveEasy System*

The class diagram for the DriveEasy system as shown in Fig. 3, represents the system's core classes and their interactions. The major classes include User, Admin, Booking, Vehicle, Maintenance, Road Tax, and Expense. The User class represents individuals who rent vehicles, whereas the Admin class represents system staff members and owners who oversee rental operations. The Vehicle class contains information about the available vehicles. The Booking class contains information about rental information that connects clients to specific vehicles. The Maintenance class oversees tracking and managing the fleet's maintenance activities, while the RoadTax class manages the road tax details for each vehicle. Vehicle financial transactions are handled via the Expense class.

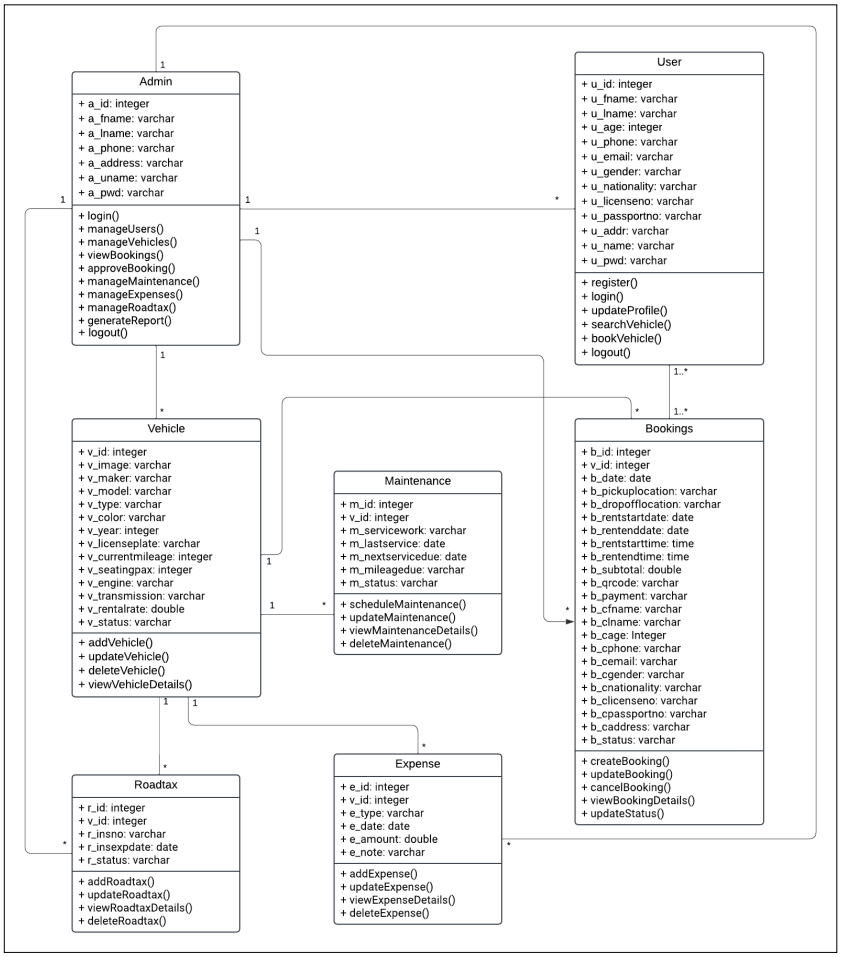


Fig.3 *Class Diagram of DriveEasy System*

4.4 System Design

The system architecture of the DriveEasy: Car Rental Management System is shown in Fig. 4, which is a Multi-Tier Architecture. The client, Front-End Server, Back-End Server, and database work together to perform functions. The Front-End Server uses HTML, CSS, JavaScript, and Bootstrap, while the Back-End Server uses PHP to handle logic and data storage with MySQL. This architecture allows for independent updates to different system components.

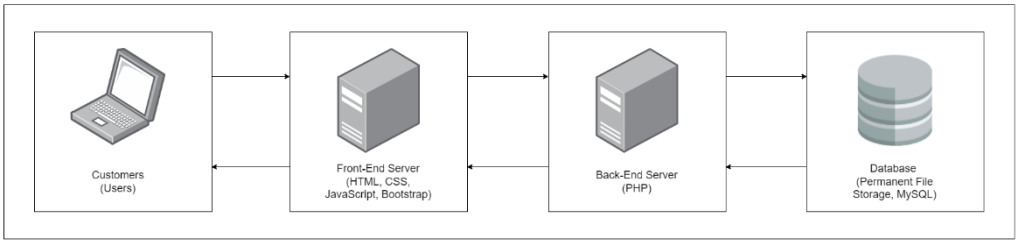


Fig.4 *Multi-Tier Architecture*

4.5 Database Design

The purpose of the database is to store and manage the data of the system. The database structure represents the entities that hold different types of data within the database.

(i) **User** (u_id, u_fname, u_lname,u_age, u_phone, u_email, u_gender, u_nationality, u_licenseno, u_passportno, u_addr, u_name, u_pwd)

(ii) **Admin** (a_id, a_fname, a_lname, a_phone, a_address, a_uname, a_pwd)

(iii) **Vehicle** (v_id, v_image, v_maker, v_model, v_type, v_color, v_year, v_licenseplate, v_currentmileage, v_seatingpax, v_engine, v_transmission, v_rentalrate, v_status)

(iv) **Bookings** (b_id, v_id, b_date, b_pickuplocation, b_dropofflocation, b_rentstartdate, b_rentenddate, b_rentstarttime, b_rentendtime, b_subtotal, b_qrcode, b_payment, b_cfname, b_cage, b_cphone, bcemail, b_cgender, b_nationality, b_clicenseno, b_cpassportno, b_caddress, b_status)

(v) **Maintenance** (m_id, v_id, m_servicework, m_nextservicedue, m_mileagedue, m_status)

(vi) **RoadTax** (r_id, v_id,r_inspo, r_insexpdate, r_status)

(vii) **Expense** (e_id, v_id, e_type, e_date, e_amount, e_note)

5. Results and Discussion

The Results and Discussion section presents the findings of the study or project, interpreting them within the context of the research questions or objectives. It discusses the significance of these results, and their implications for the field, and suggests potential avenues for further research or application.

5.1 Implementation

The system implementation combines HTML, CSS, JavaScript, and Bootstrap for the front end, with PHP for the back end. TCPDF, ApexCharts, and jQuery are also utilized. HTML, CSS, JavaScript, and Bootstrap are used to create responsive user interfaces, while PHP handles server-side functionality. Bootstrap offers pre-made CSS classes for quick development. TCPDF generates PDFs, ApexCharts creates interactive charts, and jQuery simplifies HTML manipulation. This approach enables efficient development and easier maintenance due to a clear code structure.

5.1.1 Registration and Login

The interface for registering a new customer is shown in Fig. 5. The user enters various details to start the process: first name, last name, age, phone number, email, gender, nationality, license number, passport number, address, username and password. The system checks the information entered, registers successful entries, and issues error messages if validation fails.

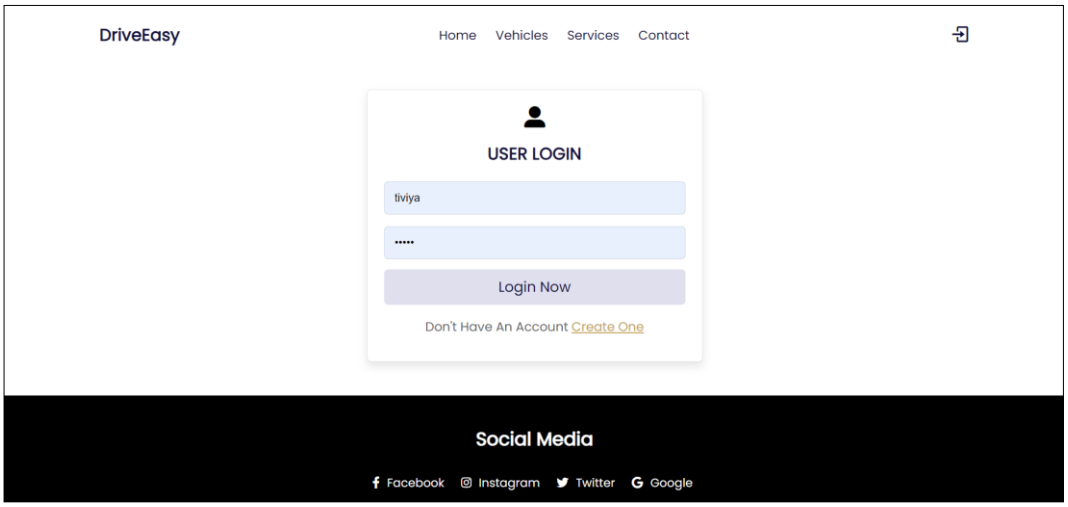


Fig. 5 *User Registration Interface*

The PHP script for handling a user registration form submission is shown in Fig. 6(a). It sanitizes and escapes input data to prevent SQL injection and other security issues, checks if the username already exists in the database, and then inserts the new user into the database if the passwords match. Upon successful registration, it redirects the user to the login page with a success message, while Fig. 6 (b) shows the HTML code for the design of the registration form. The registration form contains fields for personal details such as name, age, phone number, email, gender, nationality, address, username and password. It contains conditional input fields for license and password numbers, which are displayed depending on the nationality selected. The form also contains validation requirements, such as required fields and appropriate input types, as well as a link to the login page for existing accounts.

(a) **(b)**

Fig. 6 *(a) PHP Code Segment for Registration (b) HTML Code Segment for Registration*

The login interface for the customers is shown in Fig. 7. Users need to enter a username and password into the input fields. If the username and password are valid, the users will be allowed to log in to the system. If they are not valid, the system will display a message indicating that the username and password are incorrect.

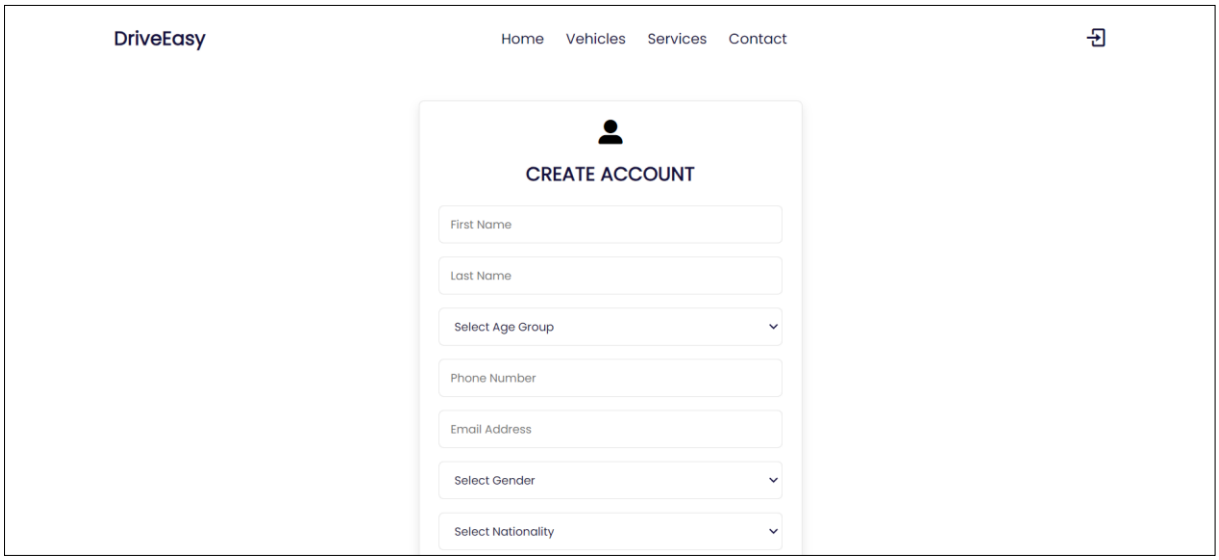


Fig. 7 *User Registration Interface*

The PHP script for handling a login form submission is shown in Fig 8(a). It sanitizes and hashes the provided username and password, compares them to a MySQL database, and, if a match is discovered, it initiates a session and redirects to index.php. If no match is found, it displays an error message indicating an incorrect username or password, while Fig. 8 (b) shows the HTML code for the design of the login form. It includes input fields for a username and password, a submit button labeled "login now," and a prompt with a link to a registration page for users who do not have an account.

```php
<?php include('../config/constants.php');

if (isset($_POST['submit']))
{
   $filter_u_name = filter_var($_POST['u_name'], FILTER_SANITIZE_STRING);
   $u_name = mysqli_real_escape_string($conn, $filter_u_name);
   $filter_u_pwd = filter_var($_POST['u_pwd'], FILTER_SANITIZE_STRING);
   $u_pwd = mysqli_real_escape_string($conn, md5($filter_u_pwd));

   $select_users = mysqli_query($conn, "SELECT * FROM `user` WHERE u_name = '$u_name' AND u_pwd = '$u_pwd'")
   or die('query failed');

   if(mysqli_num_rows($select_users) > 0){
      $row = mysqli_fetch_assoc($select_users);
      if($row['u_name'] == $u_name){
         $_SESSION['u_name'] = $row['u_name'];
         header('location:index.php');
      }
      else{
         $message[] = 'No User Found!';
      }
   }
   else{
      $message[] = 'Incorrect Username or Password!';
   }
}
include('../usr/sections/headerLogin.php'); ?>
```

(a)

```html
<!-- login form -->
<div class="register-form-wrapper">
<center>
   <section class="register-form-container">
      <form action="" method="post" name="login">
         <div class="fas fa-user-alt" style='font-size:25px'></div><br><br>
         <h3>user login</h3>
         <input type="text" name="u_name" placeholder="Username" class="box" required>
         <input type="password" name="u_pwd" placeholder="Password" class="box" required>
         <input type="submit" name="submit" value="login now" class="btn">
         <p>don't have an account <a href="registration.php">create one</a></p>
      </form>
   </section>
</center>
</div>
```

(b)

Fig. 8 *(a) PHP Code Segment for Login (b) HTML Code Segment for Login*

5.1.2 Make Booking and Reservation

The users interacting with a user-friendly interface is shown in Fig. 9, enabling easy navigation through a catalog of vehicles. This interface provides details such as vehicle models, vehicle type, rental rates, year, current mileage, and any relevant specifications. Users can browse for the available cars in the fleet and check the prices for each vehicle. The interface aims to improve user experience by presenting vehicle details in a visually appealing and structured manner, to assist customers in making informed decisions when renting a car,

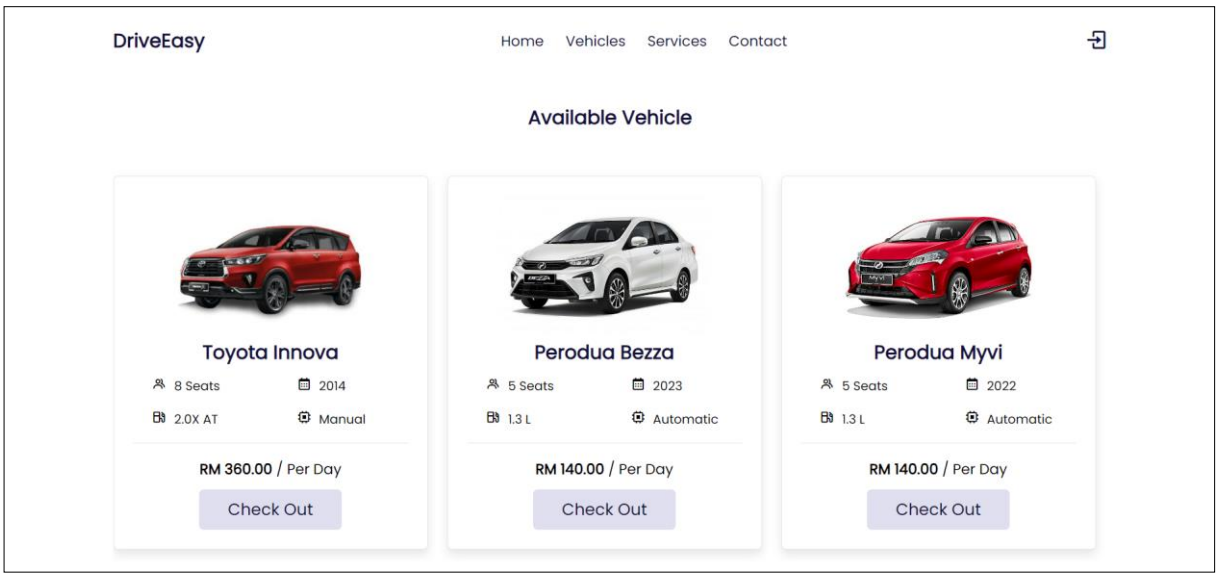


Fig. 9 *Available Vehicle Interface*

The booking and reservation interface is shown in Fig. 10. The users can initiate, view, and manage reservations. This interface will allow users to check vehicle availability and make reservations. Users need to input details such as pick-up and drop-off location, the desired rental, period, time, and any additional preferences. Upon submission, the system validates the information, checking for availability and ensuring compatibility with the specified criteria. After bookings are confirmed, users will receive booking ID details where can track the booking status of the booked vehicle.

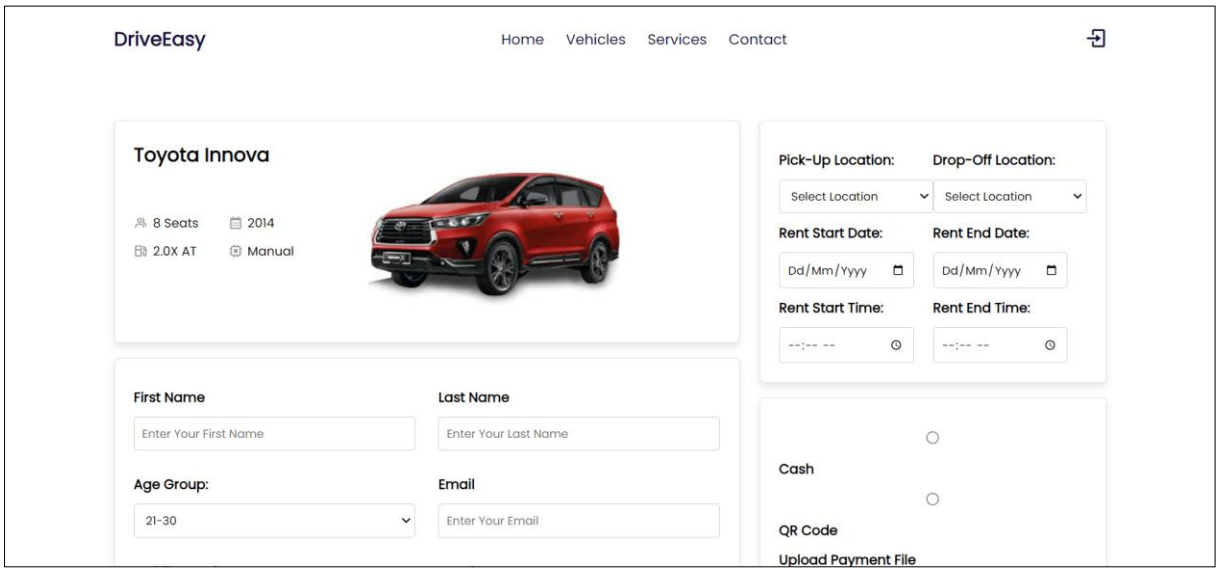


Fig. 10 *Make Booking and Reservation Interface*

5.1.3 Manage Vehicle Information

The manage vehicle information interface is shown in Fig. 11. The staff or owner can efficiently update details related to the fleet of vehicles. This interface provides a centralized platform where staff or owners can add, modify, or remove information such as vehicle models, registration details, and availability status. Additionally, it will allow for the uploading of vehicle images and documents.

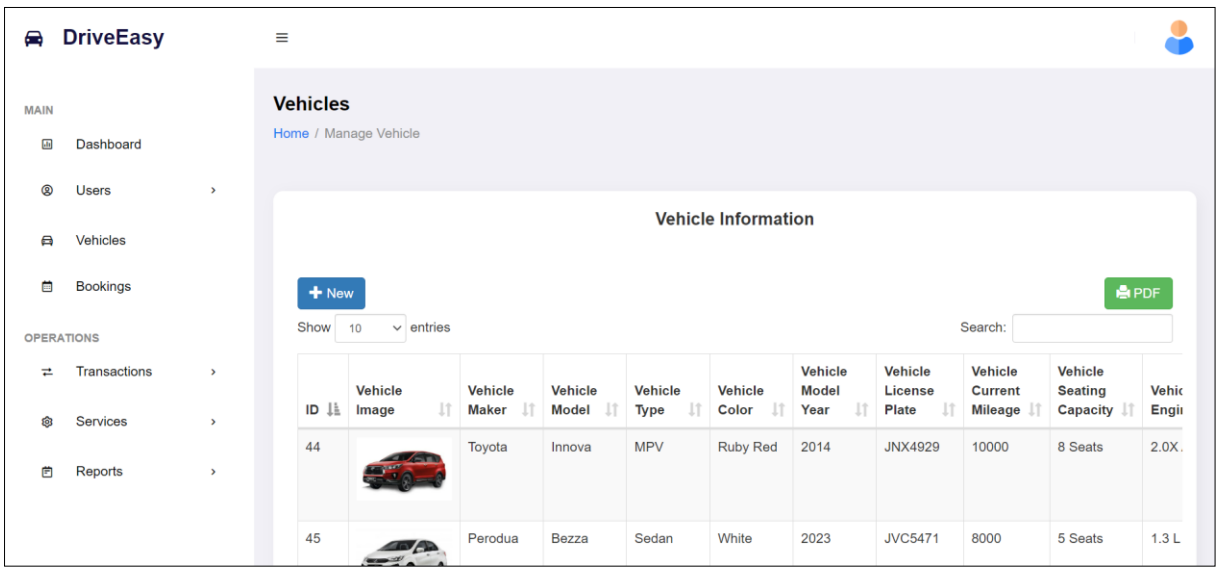


Fig. 11 *Manage Vehicle Information Interface*

5.1.4 Manage Booking

The manage booking interface is shown in Fig. 12. The staff or owners access a dedicated interface to oversee and manage bookings in the Manage Booking Interface. This interface provides a comprehensive overview of current reservations, allowing authorized personnel to create, update, or cancel bookings as needed. It includes functionalities like viewing customer details, checking vehicle availability, and managing rental schedules.

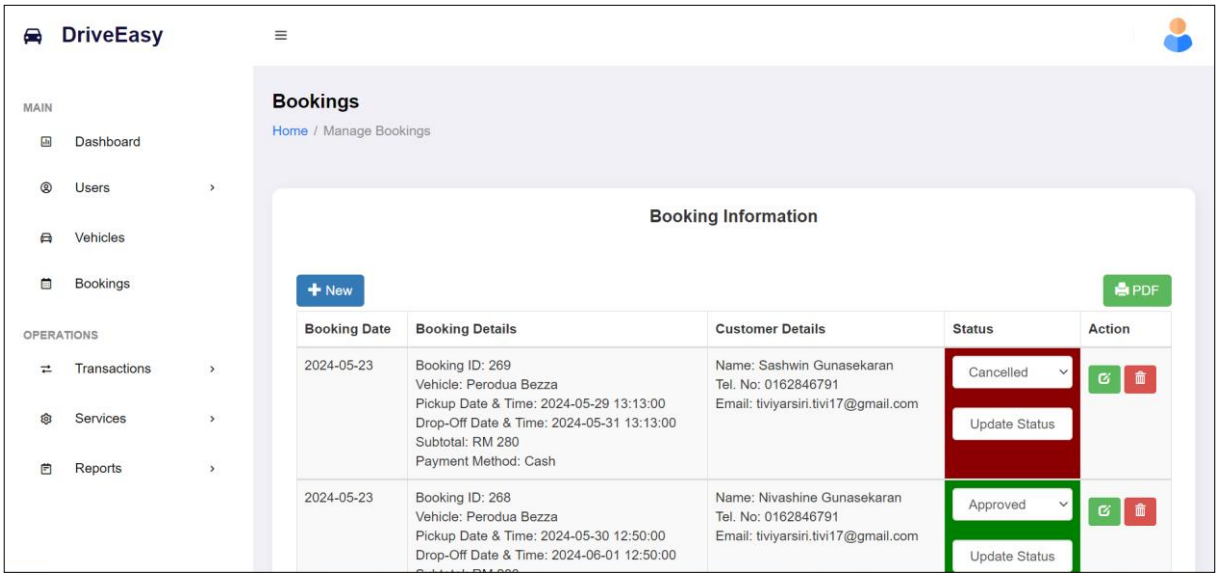


Fig. 12 *Manage Customer's Booking Interface*

5.1.5 Manage Fleet Maintenance

The manage fleet maintenance interface The manage booking interface is shown in Fig. 13. The authorized personnel, including staff and owners, can access this interface for managing and maintaining the rental fleet. These interfaces facilitate tasks such as scheduling routine maintenance, tracking vehicle inspections, managing repairs, and monitoring the overall fleet,

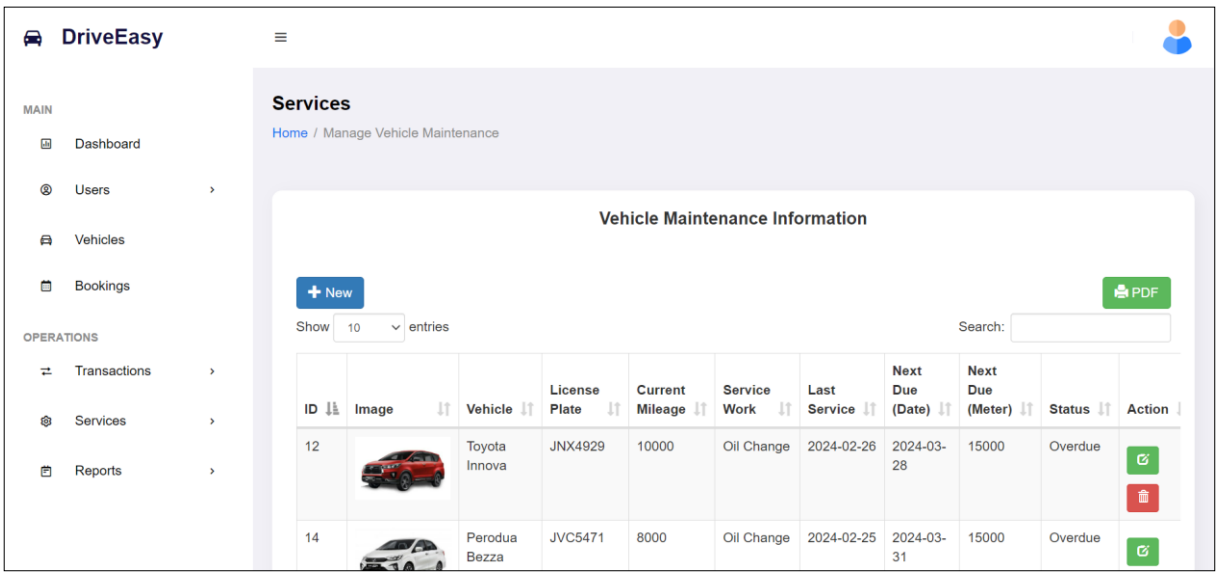


Fig. 13 *Vehicle Maintenance Interface*

The manage road tax interface is shown in Fig. 14. The staff or owners access a dedicated interface to oversee and manage road tax services. This interface provides a comprehensive overview of the vehicle's road tax details, allowing authorized personnel to create, update, or delete road tax details as needed.

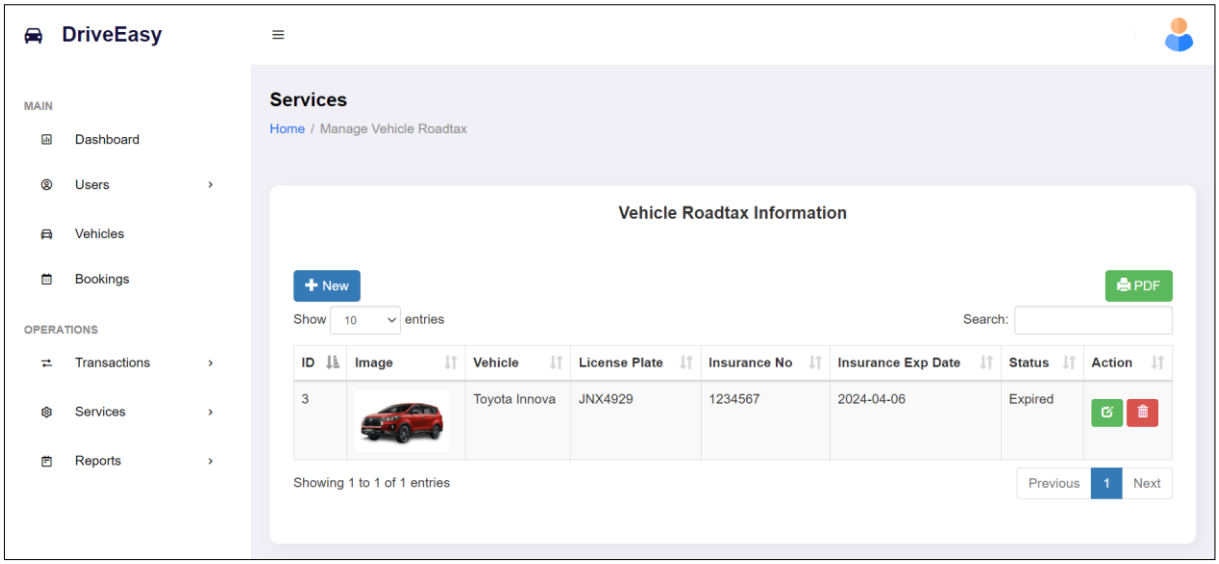


Fig. 14 *Manage Vehicle Road Tax Interface*

5.1.6 Generate Report

The generated income report interface, shown in Fig. 15, provides user-friendly drop-down filters for selecting the year and month. These filters allow for dynamic updates to the pie and bar charts that visually represent the income data for the specified period. With this functionality, employees or business owners can easily access and analyze relevant financial information. By using these drop-down filters, users can quickly change the time frame of the data they want to review, ensuring they can focus on specific months or years if required. The charts provide a clear and immediate visual summary of revenue, making it easier to identify trends, patterns, and anomalies in the company's financial performance.

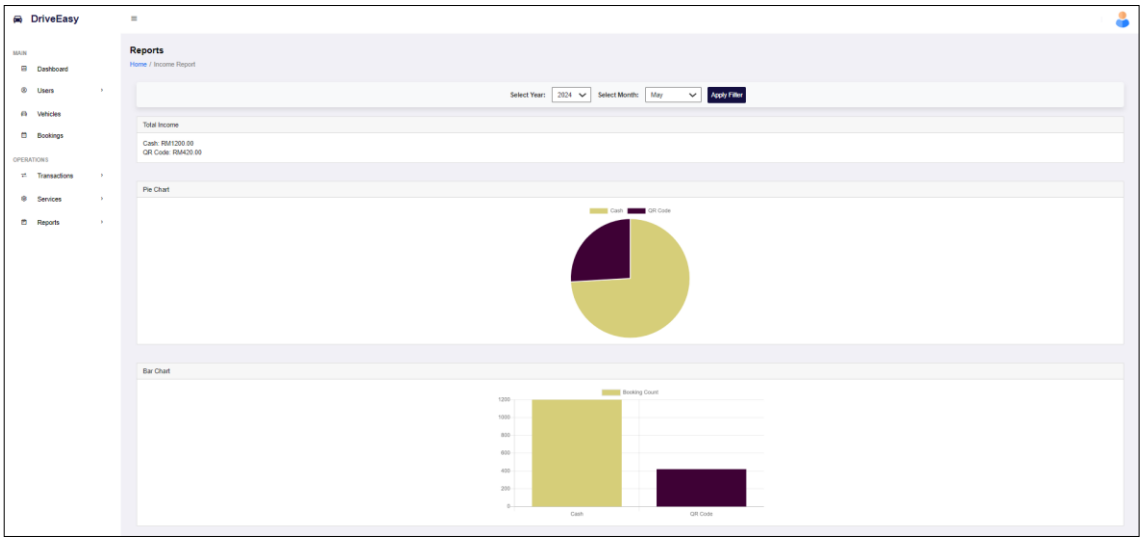


Fig. 15 *Income Report Interface*

The generated income report interface, shown in Fig. 16, provides user-friendly drop-down filters for selecting the year and month. These filters allow for dynamic updates to the pie and bar charts that visually represent the income data for the specified period. With this functionality, employees or business owners can easily access and analyze relevant financial information. By using these drop-down filters, users can quickly change the time frame of the data they want to review, ensuring they can focus on specific months or years if required. The charts provide a clear and immediate visual summary of revenue, making it easier to identify trends, patterns, and anomalies in the company's financial performance.

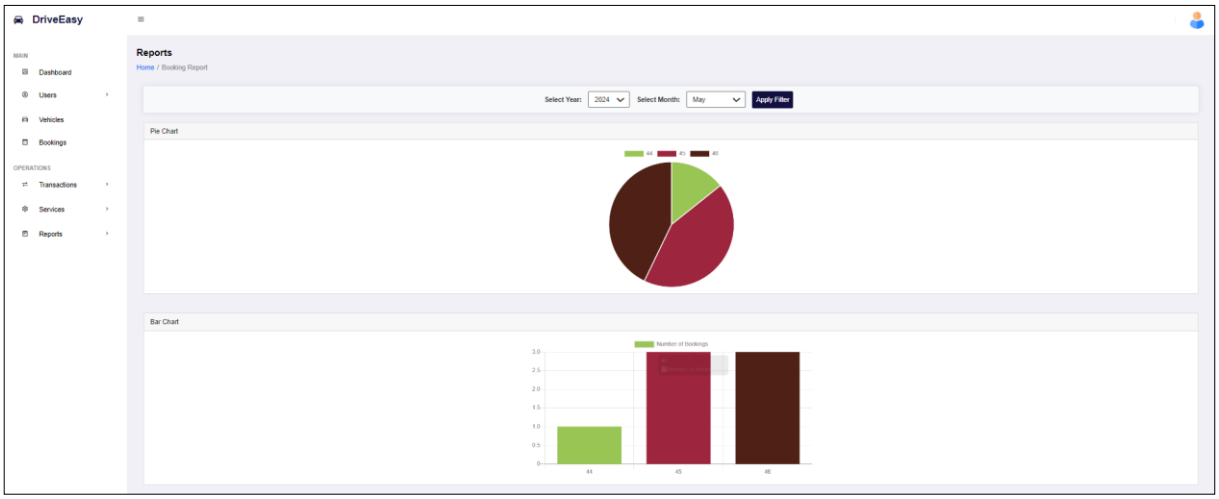


Fig. 16 *Booking Report Interface*

The generated expense report interface shown in Fig. 16 offers user-friendly drop-down filters for selecting year and month. These filters allow for dynamic updates to pie charts and bar graphs that visually represent the expense data for the specified period. With this functionality, employees or business owners can easily access and analyze relevant financial information. By using these drop-down filters, users can quickly change the time frame of the data they want to review, ensuring they can focus on specific months or years if required. The charts provide a clear and immediate visual summary of expenditure, making it easier to identify trends, patterns, and anomalies in the company's financial performance.

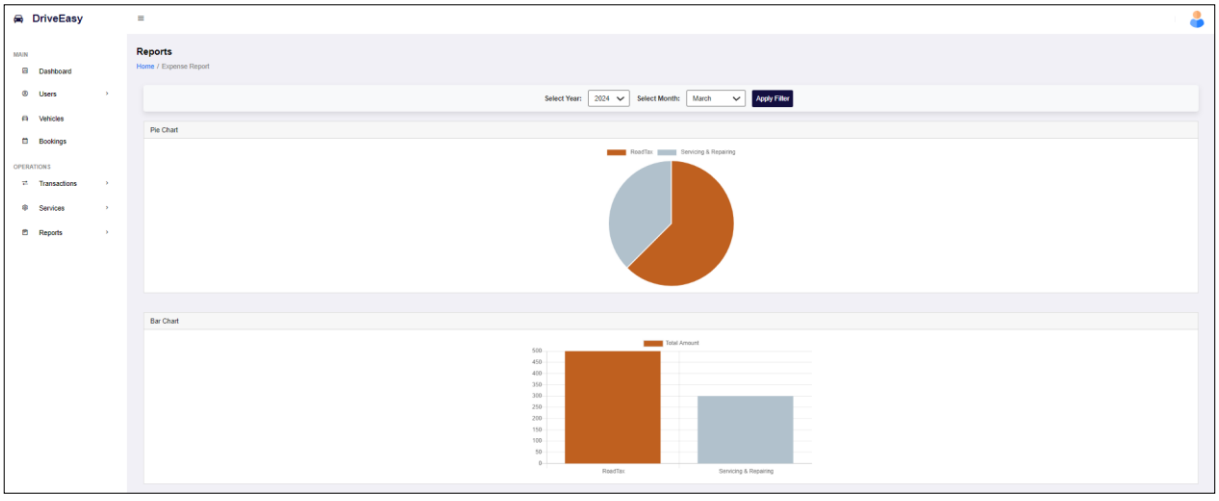


Fig. 17 *Expense Report Interface*

5.2 Testing

The testing phase is an important process for evaluating the performance and quality of the system. It involves functional and user acceptance testing to provide a comprehensive evaluation [9]. The main objective of system testing is to improve the reliability and usability of the system while minimizing risks and meeting all user needs.

Functional testing refers to the process of testing the functionality of a system to ensure it works as expected and meets user requirements. It focuses on testing each module and function within the system. Functional testing uses test cases to evaluate the system based on functional requirements. It aims to identify errors in functional requirements, such as incorrect calculations, incomplete system features, or inconsistent user interfaces. Table 5 shows test cases for all modules.

Table 5 *List of Test Cases*

Test Case	Description	Test Result (Pass/Fail)
TEST_100		
TEST_100_001	Verify that the user can access the registration page.	PASS
TEST_100_002	Verify that a customer can successfully register an account.	PASS
TEST_100_003	Verify that the system prompts an error when required fields are missing.	PASS
TEST_100_004	Verify that the system displays an error message if the password and password confirmation do not match.	PASS
TEST_100_005	Verify that the system displays an error message if the password and password confirmation do not match.	PASS
TEST_100_006	Verify that the input fields accept valid data and reject invalid data.	PASS
TEST_100_007	Verify that the system redirects the user to the Home page after successful registration.	PASS
	TEST_200	
TEST_200_001	Verify that a customer can log in successfully.	PASS
TEST_200_002	Verify that a staff member can log in successfully.	PASS
TEST_200_003	Verify that the owner can log in successfully.	PASS
TEST_200_004	Verify that the system displays an error message for a non-existent account.	PASS
TEST_200_005	Verify that the system displays an error message for an incorrect password.	PASS
TEST_200_006	Verify that the system displays a pop-up message in case of a system error during login.	PASS
TEST_200_007	Verify that the login page is accessible.	PASS
	TEST_300	
TEST_300_001	Verify that a customer can successfully make a booking with all valid details.	PASS
TEST_300_002	Verify that the system does not allow booking if the customer provides incomplete information.	PASS
TEST_300_003	Verify that the system does not allow booking if the selected rental date or time is invalid.	PASS
TEST_300_004	Verify that the system handles errors during the booking process and returns to its original state.	PASS
TEST_300_005	Verify that the customer can input all required booking details and complete the reservation.	PASS
TEST_300_006	Verify that the customer can track the booking status after receiving the Booking ID.	PASS

Table 5 *List of Test Cases (continued)*

	TEST_400	
TEST_400_001	Verify that a new vehicle can be added with all required details.	PASS
TEST_400_002	Verify that mandatory fields are validated when adding new vehicle information.	PASS
TEST_400_003	Verify that existing vehicle information can be updated.	PASS
TEST_400_004	Verify that existing vehicle information can be deleted.	PASS
TEST_400_005	Verify that vehicle information can be searched using various criteria.	PASS
TEST_400_006	Verify the system behavior when adding new vehicle information fails.	PASS
TEST_400_007	Verify the system behavior when updating vehicle information fails.	PASS
TEST_400_008	Verify the system behavior when deleting vehicle information fails.	PASS
TEST_400_009	Verify the system behavior when no matching vehicle information is found during a search.	PASS
	TEST_500	
TEST_500_001	Verify that staff can navigate to the Manage Booking module.	PASS
TEST_500_002	Verify that staff can view a list of current bookings.	PASS
TEST_500_003	Verify that staff can confirm a booking.	PASS
TEST_500_004	Verify that staff can modify booking details.	PASS
TEST_500_005	Verify that staff can cancel a booking.	PASS
TEST_500_006	Verify that booking information can be searched using various criteria.	FAIL
TEST_500_007	Verify that the customer is notified when the booking status changes.	PASS
TEST_500_008	Verify the system behavior when adding new booking information fails.	PASS
TEST_500_009	Verify the system behavior when updating booking information fails.	PASS
TEST_500_010	Verify the system behavior when deleting booking information fails.	PASS
TEST_500_011	Verify the system behavior when no matching booking information is found during a search.	FAIL
TEST_500_012	Verify that the system handles errors during booking updates gracefully.	PASS
	TEST_600	
TEST_600_001	Verify that the customer can access the payment section.	PASS
TEST_600_002	Verify that the customer can select the cash payment method.	PASS
TEST_600_003	Verify that the customer can select the QR code payment method.	PASS
TEST_600_004	Verify that the customer can change the payment method.	PASS
TEST_600_005	Verify the system behavior when the payment file upload fails.	PASS
	TEST_700	
TEST_700_001	Verify the staff can navigate to the Manage Fleet Maintenance module.	PASS
TEST_700_002	Verify the staff can create a maintenance task for a vehicle.	PASS
TEST_700_003	Verify the system automatically updates the maintenance status based on the Next Due Date.	PASS

Table 5 *List of Test Cases (continued)*

TEST_700_004	Verify that staff can modify maintenance details.	PASS
TEST_700_005	Verify that staff can cancel a maintenance record.	PASS
TEST_700_006	Verify the system behavior when adding new maintenance information fails.	PASS
TEST_700_007	Verify the system behavior when updating maintenance information fails.	PASS
TEST_700_008	Verify the system behavior when deleting maintenance information fails.	PASS
TEST_700_009	Verify the system behavior when no matching maintenance information is found during a search.	PASS
TEST_700_010	Verify that only authorized staff or owners can access the fleet maintenance management page.	PASS
	TEST_800	
TEST_800_001	Verify the staff or owner can access the report generation module.	PASS
TEST_800_002	Verify the system handles errors gracefully during report generation.	PASS
TEST_800_003	Verify the staff can use various filtering options to customize the report.	PASS

The overall results of the test cases are shown in Table 6, indicating the number of test cases and the number of passed tests, while Fig. 18 shows a pie chart indicating the percentage of passed and failed tests for the entire test case. All test cases for registration, login, booking, vehicle info management, payment, fleet maintenance, and report generation (TEST_100, TEST_200, TEST_300, TEST_400, TEST_600, TEST_700, TEST_800) had a 100% pass rate. Manage Booking (TEST_500) had an 83% pass rate, highlighting areas for improvement. Overall, with one small exception, the system performs well and ensures stable functionality.

Table 6 *Overall Test Case Result*

Test Case ID	Total Test Cases	Total Passed
TEST_100	7	7
TEST_200	7	7
TEST_300	6	6
TEST_400	9	9
TEST_500	12	10
TEST_600	5	5
TEST_700	10	10
TEST_800	3	3

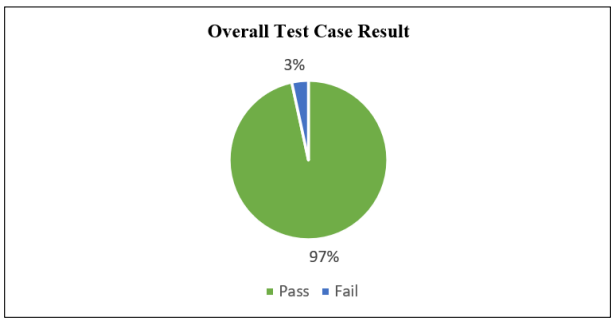
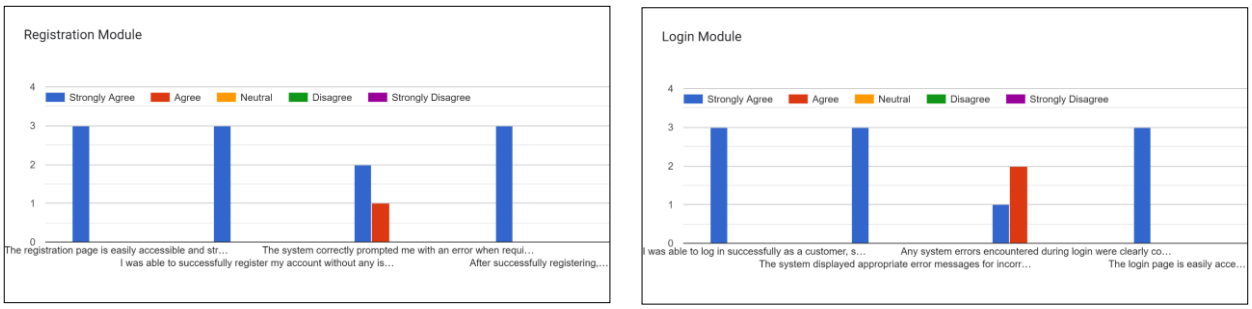


Fig. 18 *Overall Test Case Result*

User acceptance testing is an important phase of system testing in which users verify the compatibility and functionality of the system before it is deployed on a large scale. Three users, including the owner, employees, and the customer, tested the system and provided feedback via a Google form. Eight features require ratings on a scale from "strongly disagree" to "strongly agree" to assess satisfaction with the system's performance and ease of use.

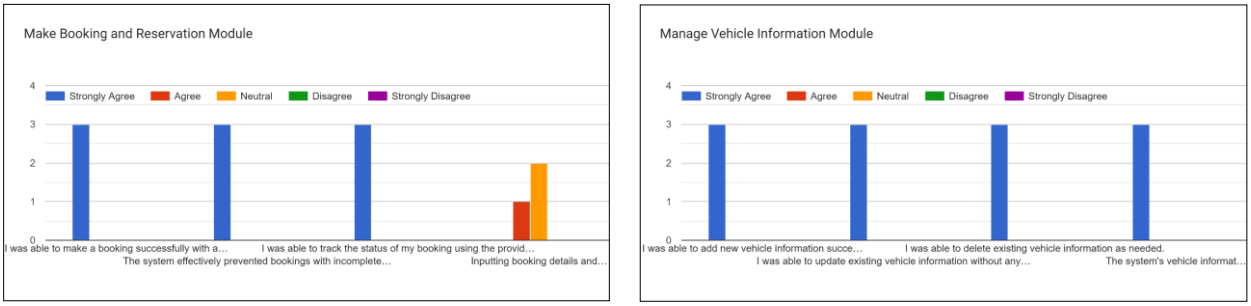
The feedback for the registration module in Fig. 18(a) indicates that most users strongly agree that it functions well. Users found the registration page accessible and easy to use, registering without issues and being redirected to the home page. In Fig. 18(b), feedback for the login module shows that most users strongly agree, and it meets their expectations. Users found the login process straightforward, with appropriate error messages for incorrect logins. Some users rated "Agree" instead of "Strongly Agree" for system error communication.



(a) **(b)**

Fig, 19 *(a) Feedback for the Registration Module (b) Feedback for the Login Module*

Most users think that the booking module works well and allows them to successfully make and track bookings in Fig. 19(a). They appreciate the system's ability to prevent incomplete or invalid bookings. There are concerns about the ease of entering details and completing the process. In Fig.19(b), users agree with the statement that the vehicle information management module is effective and allows them to easily perform tasks such as adding, updating, and deleting vehicle information. The search function is also praised for its effectiveness in searching for vehicle information.

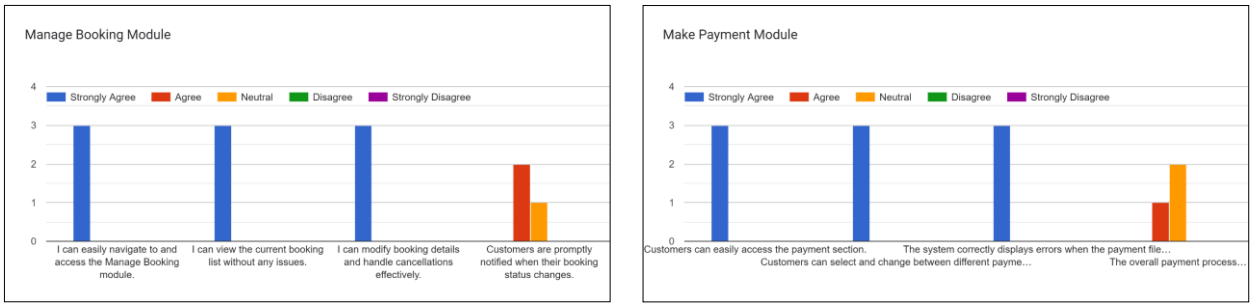


(a) **(b)**

Fig, 20 *(a) Feedback for the Make Reservation Module (b) Feedback for the Manage Vehicle Information Module*

The majority of users strongly agree that the booking module meets their expectations and offers easy navigation and effective booking changes as shown in Fig. 20(a). However, a few users point out that the notification of booking status changes should be improved. Fig. 20(b) shows that most users are very satisfied with the functionality and ease of use of the payment module, although some users express concerns about the smooth and user-friendly payment process.

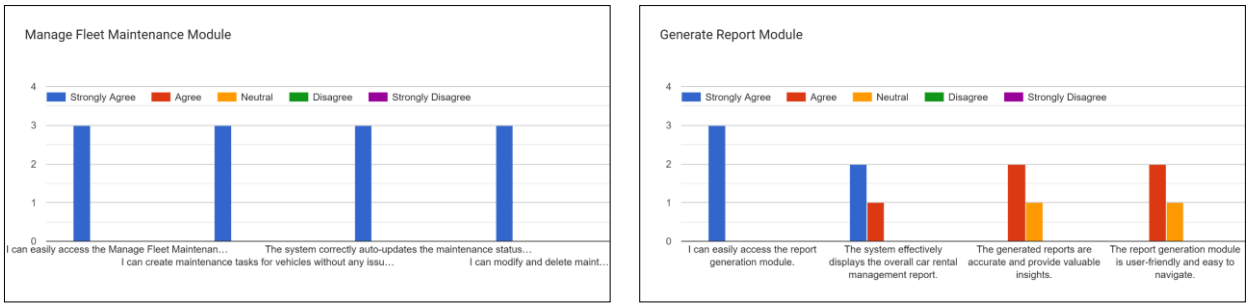
Applied Information Technology and Computer Science Vol. 6 No. 1 (2025) p. 861-880



(a) **(b)**

Fig, 20 *(a) Feedback for the Manage Booking Module (b) Feedback for the Make Payment Module*

The feedback on the fleet maintenance management module in Fig. 21(a) shows that users think it works well and meets expectations. They find it easy to use, create, update, and delete maintenance tasks. Users appreciate the automatic update function of the system based on the next due date. In Fig. 21(b), feedback on the Generate Report module shows that users think it works well and is easy to use. Users are satisfied with the car rental management report overall but have mixed opinions on its accuracy and ease of use.



(a) **(b)**

Fig, 21 *(a) Feedback for the Manage Fleet Maintenance Module (b) Feedback for the Generate Report Module*

6. Conclusion

The DriveEasy Car Rental Management System achieved its goals using an object-oriented approach to modularity, scalability, and maintainability. Front-end and back-end technologies were used to create an appealing user interface. User acceptance testing was conducted to ensure reliability and user satisfaction and to demonstrate the system's reliability, ease of use, and user-centered design.

The developed web-based car rental management system offers various advantages, including improved operational efficiency, higher customer satisfaction, and comprehensive reporting modules for better decision-making. However, there are also drawbacks, such as potential security risks associated with remaining logged in, manual input of payment details leading to inaccuracies, and limitations in real-time vehicle tracking. Future improvements to the system include integration with online payment gateways, a feedback mechanism for customer reviews, and the development of a mobile application for reservations and bookings. These improvements aim to optimize the payment process, collect customer feedback, and improve accessibility for customers using mobile platforms. Overall, the web-based car rental management system is functional and beneficial, and work will continue to optimize its features and performance in the future.

In summary, the implementation of DriveEasy: Car Rental Management System at Nadim Car Rental has opened up significant potential to improve operational efficiency and customer experience. With its key features and emphasis on automation, the proposed system is well-positioned to deliver positive results. To stimulate further research, the studies should investigate the impact of implementing a car rental system in practice.

Acknowledgment

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:** Tiviyarsiri A/P Gunasekaran, Siti Hajar Binti Arbain; **data collection:** Tiviyarsiri A/P Gunasekaran, Siti Hajar Binti Arbain**; analysis and interpretation of results:** Tiviyarsiri A/P Gunasekaran, Siti Hajar Binti Arbain; **draft manuscript preparation:** Tiviyarsiri A/P Gunasekaran, Siti Hajar Binti Arbain. All authors reviewed the results and approved the final version of the manuscript.

References

[1] Lazov, "Profit management of car rental companies," European Journal of Operational Research, vol. 258, no. 1, pp. 307–314, 2017. https://doi.org/10.1016/j.ejor.2016.08.064.
[2] M. N. Osman, N. Md. Zain, Z. Paidi, K. A. Sedek, M. N. Yusoff, and M. Maghribi, "Online Car Rental System using Web-Based and SMS Technology", JCRINN, vol. 2, no. 3, pp. 33–38, Sep. 2017.
[3] E. H. Ekiz and A. Bavik, "Scale development process: service quality in car rental services," Electronic Journal of Business Research Methods, vol. 6, no. 2, pp. 161-174, 2008.
[4] Wagons Car Rental Services, "Wagons Car Rental Services," 2023. [Online]. Available: https://kuchingcarsrental.com/.
[5] Go Big Travel & Tours, "Go Big Travel & Tours," Gobigtravel.my, 2015. [Online]. Available: https://gobigtravel.my/index.php.
[6] ETT EXCELLENT SERVICES, "ETT Car Rental," 2019. [Online]. Available: https://www.ettcarrental.com/. [Accessed: 24-Nov-2023].
[7] R. B. Brown and E. Corporation, Doing your dissertation in business and management: the reality of researching and writing. London: Sage Publications Ltd, 2006.
[8] H. Berger and P. Beynon-Davies, "The utility of rapid application development in large-scale, complex projects," Information Systems Journal, vol. 19, no. 6, pp. 549–570, Nov. 2009, doi: https://doi.org/10.1111/j.1365-2575.2009.00329.x.
[9] T. Jindal, "Importance of Testing in SDLC," International Journal of Engineering and Applied Computer Science, vol. 01, no. 02, pp. 54–56, Dec. 2016, doi: https://doi.org/10.24032/ijeacs/0102/05.