

# EasyService Car Service Booking System

Praveen Thiagarajan<sup>1</sup>, Mohamad Firdaus Ab Aziz<sup>1\*</sup>

<sup>1</sup> *Fakulti Sains Komputer dan Teknologi Maklumat,  
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA*

\*Corresponding Author: [mdfirdaus@uthm.edu.my](mailto:mdfirdaus@uthm.edu.my)  
DOI: <https://doi.org/10.30880/aitcs.2026.07.01.085>

## Article Info

Received: 12 June 2025

Accepted: 18 June 2026

Available online: 30 June 2026

## Keywords

Service , car , Booking

## Abstract

EasyService is a PHP-based car service booking system developed to address inefficiencies in booking processes, customer satisfaction, and resource management. The system aims to streamline service scheduling, enable real-time notifications, and enhance customer experience for small and medium-sized service centers. Built using PHP and MySQL, it includes modules for appointment booking, service history tracking, and feedback collection. Usability testing was conducted through user feedback, focusing on satisfaction, error rates, and task completion speed. Results showed improved booking efficiency and user satisfaction. Future improvements may include predictive maintenance, mobile support, and enhanced multilingual accessibility and innovation.

## 1. Introduction

EasyService is a web-based car service booking system developed for Day by Day Auto Parts to address booking inefficiencies and improve service operations. Traditional methods like phone bookings or walk-ins often lead to long waits, overbooking, and poor communication. EasyService allows customers to book services online, receive reminders, and track their service history. It also helps service providers manage appointments, resources, and customer records more efficiently [1]. By integrating automated features and centralized data, the system improves customer satisfaction and modernizes service management for small and medium-sized workshops.

## 2. Literature Reviews

Many car workshops still rely on manual service scheduling, leading to operational delays and missed maintenance. Recent studies show that digital platforms help streamline appointments, reduce errors, and enhance customer engagement[2]. Inspired by booking systems used in healthcare and e-commerce, EasyService integrates features such as real-time notifications, service history tracking, and centralized records. It simplifies booking, supports timely reminders, and enables better decision-making for workshop operators. These enhancements improve service quality and build stronger customer relationships [3].

### 2.1 Comparison with the Existing Systems

Existing systems like AutoFix, OpenBay, and YourMechanic offer useful features but lack full integration. EasyService improves them by combining booking, notifications, inventory management, and reminders into one platform. The table below highlights the key differences:

**Table 1** System comparison

| Features/System         | AutoFix              | OpenBay              | YourMechanic           | EasyService          |
|-------------------------|----------------------|----------------------|------------------------|----------------------|
| Log-in Process          | User ID & Password   | User ID & Password   | User ID & Password     | User ID & Password   |
| Service History         | √                    | √                    | √                      | √                    |
| Booking Flexibility     | Available time slots | Multi-provider model | Mechanical-to-location | Optimized scheduling |
| Real-time Notifications | √                    | X                    | √                      | √                    |
| Payment Integration     | X                    | √                    | X                      | √                    |
| Personalized Reminders  | X                    | X                    | X                      | √                    |

### 3. Methodology

The development of the EasyService system adopted the Agile methodology to ensure continuous improvement, user involvement, and adaptability throughout the project lifecycle. Agile is a flexible, iterative development approach that divides the project into smaller cycles known as sprints. Each sprint focuses on developing specific system features, allowing feedback from stakeholders such as mechanics, administrators, and customers from Day-by-Day Auto Parts to be gathered and incorporated before moving to the next phase. This cycle of design, development, testing, and review was repeated until the system met the desired usability and performance goals. The Agile model is particularly effective for web-based systems like EasyService, where user experience and rapid adjustments are critical. It also helps reduce development risks, ensures early detection of errors, and maintains alignment with user expectations. By emphasizing collaboration and adaptability, the Agile method supported the creation of a reliable, user-centric car service booking system. [4]

**Fig. 1** Agile Model

#### 3.1 System Development Workflow

There are five phases from the Agile model. As shown in Table 2, each phase has its own tasks and results that need to be produced during the entire project development. In addition, results were completed within a certain number of days.[5]

**Table 2** *Software development activities and their task*

| Phase          | Task                                                                                                                                                                | Output                                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| Planning       | <ul style="list-style-type: none"> <li>Define scope, set priorities</li> </ul>                                                                                      | <ul style="list-style-type: none"> <li>Project proposal, Gantt chart</li> </ul>                                                            |
| Analysis       | <ul style="list-style-type: none"> <li>Gather requirements, document functionalities</li> </ul>                                                                     | <ul style="list-style-type: none"> <li>DFD,</li> <li>ERD,</li> <li>flowchart</li> </ul>                                                    |
| Design         | <ul style="list-style-type: none"> <li>Design system architecture, UI/UX</li> </ul>                                                                                 | <ul style="list-style-type: none"> <li>system architecture</li> <li>database schema</li> <li>data dictionary</li> <li>UI design</li> </ul> |
| Implementation | <ul style="list-style-type: none"> <li>Develop frontend (HTML/CSS), backend (PHP) and database (XAMPP)</li> </ul>                                                   | <ul style="list-style-type: none"> <li>Functional modules</li> </ul>                                                                       |
| Testing        | <ul style="list-style-type: none"> <li>Conduct unit, integration, and UAT</li> </ul>                                                                                | <ul style="list-style-type: none"> <li>Tested and validated system</li> </ul>                                                              |
| Deploy Project | <ul style="list-style-type: none"> <li>System use by end users.</li> <li>Offer technical assistance and respond to enquiries or issues that may come up.</li> </ul> | <ul style="list-style-type: none"> <li>Verified firmware and test results.</li> </ul>                                                      |

#### 4. Analysis and Design

The EasyService System is an innovative platform aimed at simplifying vehicle service booking and management for customers and mechanics. This report details the system's functional and non-functional requirements, ensuring it effectively meets its objectives. By addressing the needs of all stakeholders, it provides a comprehensive understanding of the system's goals and capabilities.[8]

##### 4.1 Functional and Non-Functional Requirements

The Functional and non-functional requirements are important for defining what the EasyService System can do and how it will perform. Functional requirements describe the tasks the system should carry out, while non-functional requirements focus on how well the system should work in terms of speed, reliability, and other qualities. Both are essential for making sure the system works as expected and is efficient.

**Table 3** *Functional requirements.*

| Num | Category                | Functional Requirement                                                                                                                                                                                 |
|-----|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1   | Appointment Scheduling  | Users can book appointments online by selecting available time slots.<br>Prevent double booking or overbooking.<br>Allow users to reschedule or cancel appointments.                                   |
| 2   | Service Catalog         | Display a list of services with descriptions and prices.<br>Allow users to filter and search for specific services.<br>Update the catalog dynamically as new services are added.                       |
| 3   | Notification & Reminder | Send automated reminders for upcoming appointments.<br>Provide real-time updates on service status.                                                                                                    |
| 4   | Service History         | Maintain a detailed record of past services for each user.<br>Allow users to view their service history at any time.<br>Generate reports based on service history for user reference.                  |
| 5   | Reporting & Analytics   | Generate reports on booking trends, peak hours, and customer preferences.<br>Provide insights into inventory usage and service performance.<br>Offer dashboard views for administrators and mechanics. |

**Table 3 (cont)**

| Num | Category                     | Functional Requirement                                                                                                                                                                    |
|-----|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 6   | User Management              | Allow users to register, log in, and manage their profiles.<br>Support multiple roles: customers, mechanics, and administrators.<br>Securely store user data and maintain access control. |
| 7   | Feedback & Review            | Allow users to leave feedback.<br>Analyze feedback to improve services.                                                                                                                   |
| 8   | Customization and Promotions | Provide personalized service recommendations based on user history.<br>Enable administrators to create and manage promotions and discounts.                                               |
| 9   | Real-Time Communication      | Allow users to chat or message mechanics for inquiries.<br>Notify customers about delays or service completions in real-time                                                              |
| 10  | Geolocation & Navigation     | Provide a map to locate nearby service centers.                                                                                                                                           |
| 11  | Service Queue Management     | Enable users to join a virtual queue for walk-in appointments.                                                                                                                            |

The non-functional requirements for the EasyService system ensure a smooth and accessible user experience. The system must perform reliably, with consistent availability and page load times under one minute. A user-friendly interface should make navigation intuitive, while compatibility with major web browsers like Chrome, Firefox, and Safari ensures accessibility for all users. These requirements focus on reliability, efficiency, and ease of use to meet diverse user needs effectively.

**Table 4 Non-functional requirements**

| No | Requirement            | Description                                                      |
|----|------------------------|------------------------------------------------------------------|
| 1  | Performance            | The system should be always usable                               |
| 2  | Operational            | The loading time required for a website is no more than 1 minute |
| 3  | Security               | The system should be user friendly                               |
| 4  | Cultural and political | The system should be able to work on any web browser             |

## 4.2 User Requirement Analysis

User requirements outline the needs and expectations of the various stakeholders who will interact with the EasyService System. These requirements focus on the system's functionality and usability, ensuring it addresses the core challenges faced by customers, mechanics, and administrators.

**Table 5 User requirements.**

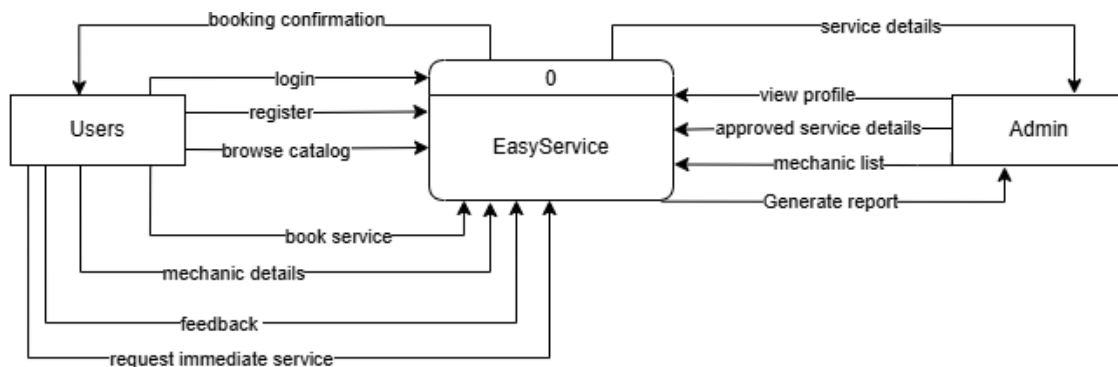
| No | Requirement                                                                                    |
|----|------------------------------------------------------------------------------------------------|
| 1. | All users must have an account with a valid ID and password to access the system.              |
| 2. | Customers should be able to book new appointments for car services online.                     |
| 3. | Customers should be able to edit or cancel their appointments within a specified time.         |
| 4. | Customers should be able to choose their preferred mechanics based on availability or ratings. |
| 5. | Mechanics should be able to update and manage their service records.                           |
| 6. | Administrators should be able to oversee the mechanical task schedules.                        |
| 7. | Administrators should be able to create and assign task lists for mechanics by date.           |
| 8. | The system should send automated reminders for upcoming service appointments to customers.     |
| 9. | Customers should be able to view and track their vehicle's service history.                    |

**Table 5 (Cont)**

| No. | Requirement                                                                                          |
|-----|------------------------------------------------------------------------------------------------------|
| 10. | Administrators should be able to create and assign task lists for mechanics by date.                 |
| 11. | Customers should be able to leave feedback or rate their service experience.                         |
| 12. | Feedback should be analyzed to improve services and display average ratings for transparency.        |
| 13. | Administrators should be able to create and manage promotions and discounts for customers.           |
| 14. | Customers should receive personalized service recommendations based on their service history.        |
| 15. | The system should enable real-time communication, allowing users to message mechanics for inquiries. |
| 16. | Customers should receive notifications and service completions in real time.                         |
| 17. | The system should offer navigation assistance.                                                       |
| 18. | Customers should be able to join a virtual queue for walk-in appointments.                           |
| 19. | Administrators should be able to generate reports on service usage and customer preferences.         |

### 4.3 Context Diagram

The EasyService car service booking system enables users to register, log in, browse services, book appointments, request urgent help, and give feedback, while the admin manages user accounts, updates services, and oversees mechanical details. This streamlined system ensures smooth user interactions and equips the admin with tools to maintain the platform, providing an efficient and user-friendly experience.

**Fig. 2** EasyService Context diagram

### 4.4 DFD Level 0

A Data Flow Diagram (DFD) visually represents the movement of data from an entity through a process, which produces output either to another entity or to data storage. It illustrates the inputs and outputs for each entity and process. Figure 3 displays the Level 0 Data Flow Diagram (DFD 0) of the developed system.[6]

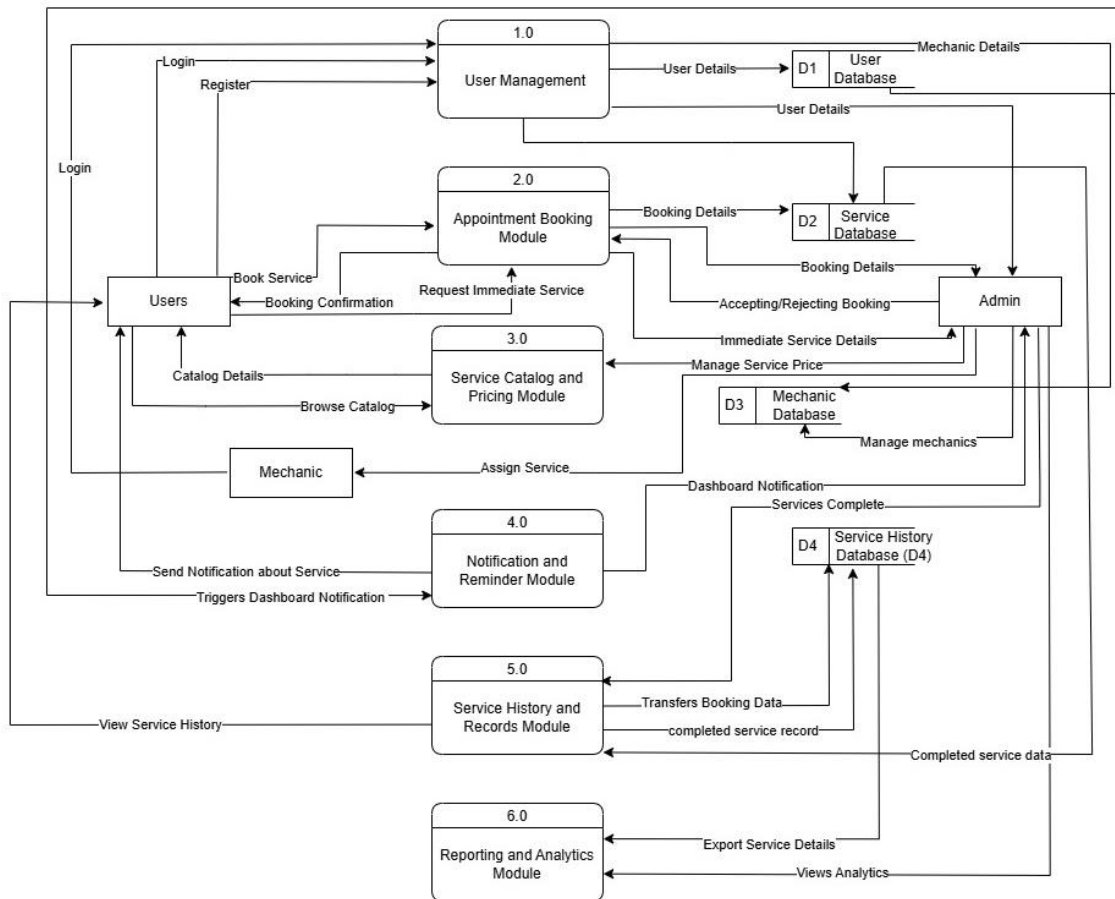


Fig. 3 DFD level 0

#### 4.5 Entity Relationship Diagram (ERD)

In the design of the EasyService car service booking system, an Entity Relationship Diagram (ERD) is used. The ERD, also called the ER Diagram or ER Model, is a diagram that helps in designing databases. It shows the different entities, their attributes, and how they are related to each other in the database. Figure 4 shows the ERD for the system.

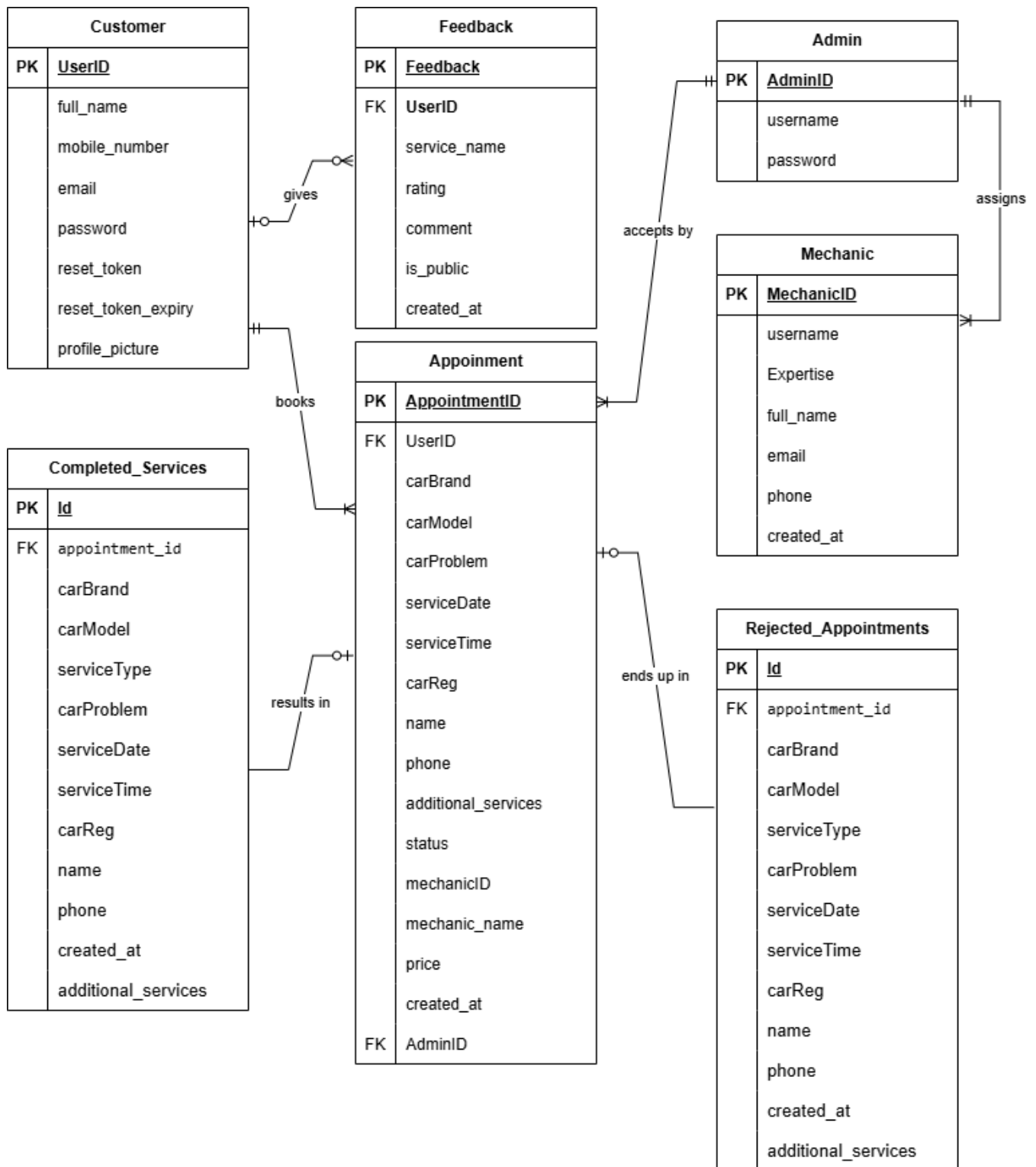
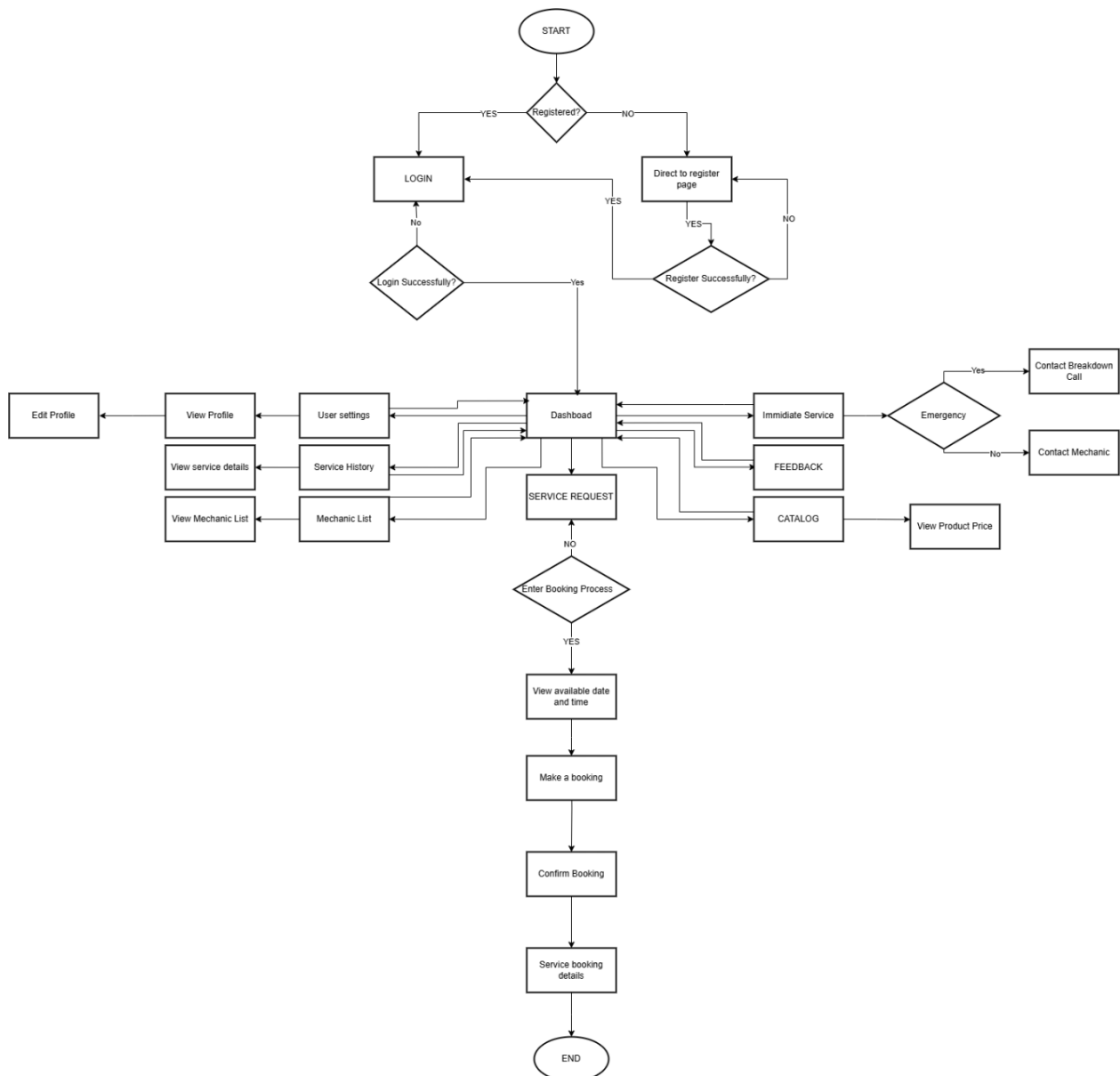


Fig. 4 ERD EasyService

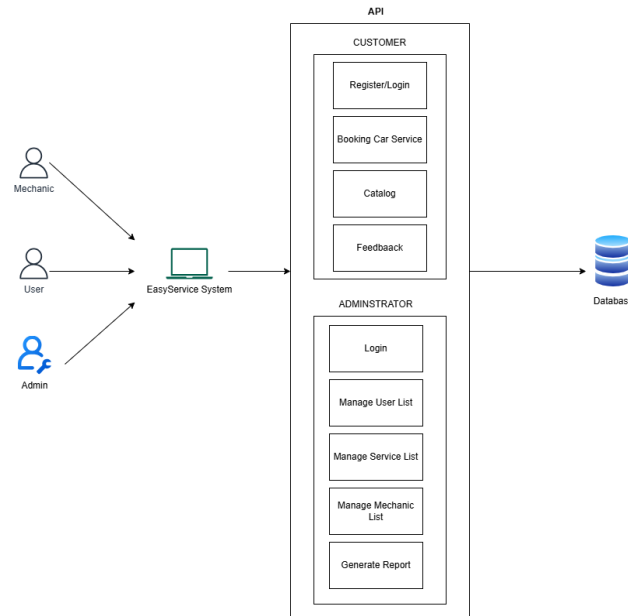
#### 4.6 Flowchart

A flowchart is a diagram that shows the steps in a process. It's used to visualize, analyze, and communicate processes. Figure 5 shows a flowchart for the developed system.

**Fig. 5** Flowchart

#### 4.7 System Architecture

The proposed EasyService system is designed to facilitate smooth and secure transactions for customers while offering comprehensive management tools for administrators. Users can sign up, log in, browse services, book appointments, and provide feedback, while mechanics carry out car service tasks. Administrators manage user accounts, update services, handle mechanical details, and generate reports. The system connects all roles to a centralized database via an API, ensuring effective management of user, service, mechanic, and feedback data, resulting in smooth and efficient operation.



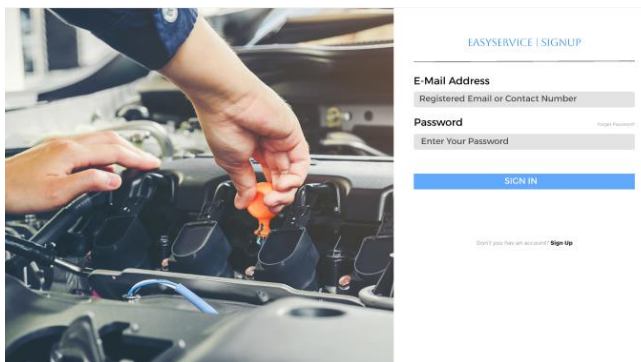
**Fig. 6** System architecture design

## 4.8 Interface Design

The interfaces for each process have been designed using Figma, a collaborative web application that focuses on user interface and user experience design. Figma also offers offline capabilities through desktop applications for macOS and Windows. Its feature set is tailored to create and refining interfaces, ensuring a seamless design experience for users.

### 4.8.1 Login interface design

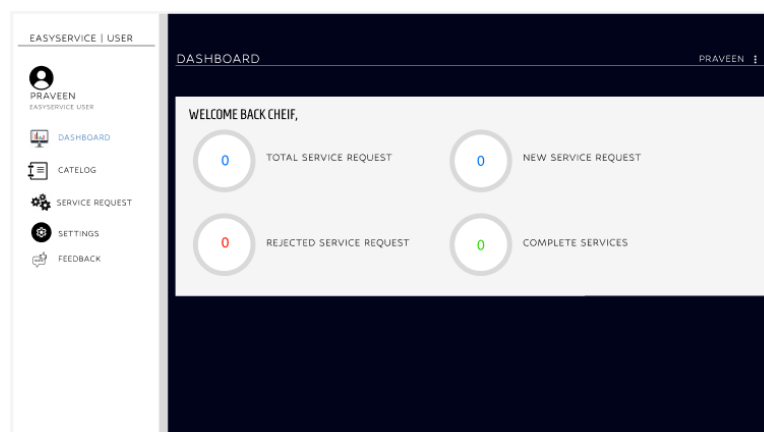
The login interface design, shown in Figure 7, is used by both customers and administrators to access the system by entering their correct username/email and password. The system validates the credentials, and if both are correct, the user is redirected to the main page. If the credentials are incorrect, an error message is displayed, prompting the user to re-enter the correct username and password.



**Fig. 7** Login interface design

### 4.8.2 Dashboard interface design

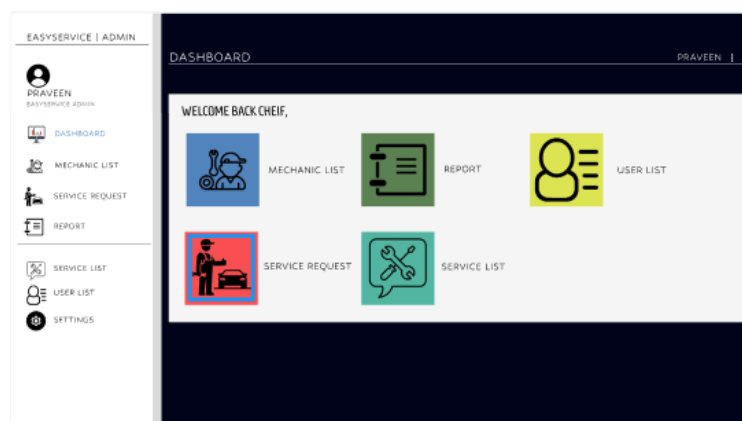
Figure 8 shows the User Dashboard, which users see after logging in. This screen serves as the central hub, displaying key information such as recent activities, notifications, and updates. It also offers quick links or buttons to navigate to various parts of the system, such as the user's profile, service requests, and other important features. The dashboard is designed to be simple and user-friendly, ensuring users can easily access the information and functions they need without confusion.



**Fig. 8** Dashboard interface design

### 4.8.3 Admin interface design

Figure 9, the Admin Dashboard, is a dedicated interface for system administrators to manage and monitor the system. It offers tools for managing user accounts, including adding new users, updating details, and removing inactive accounts. Administrators can track system activity, view reports, and oversee ongoing tasks or issues. The dashboard also displays key metrics, such as user statistics and performance data, for quick decision-making. Designed to be efficient and user-friendly, it helps administrators keep the system organized and functioning smoothly.



**Fig. 9** Admin interface design

## 5. Implementation and Testing

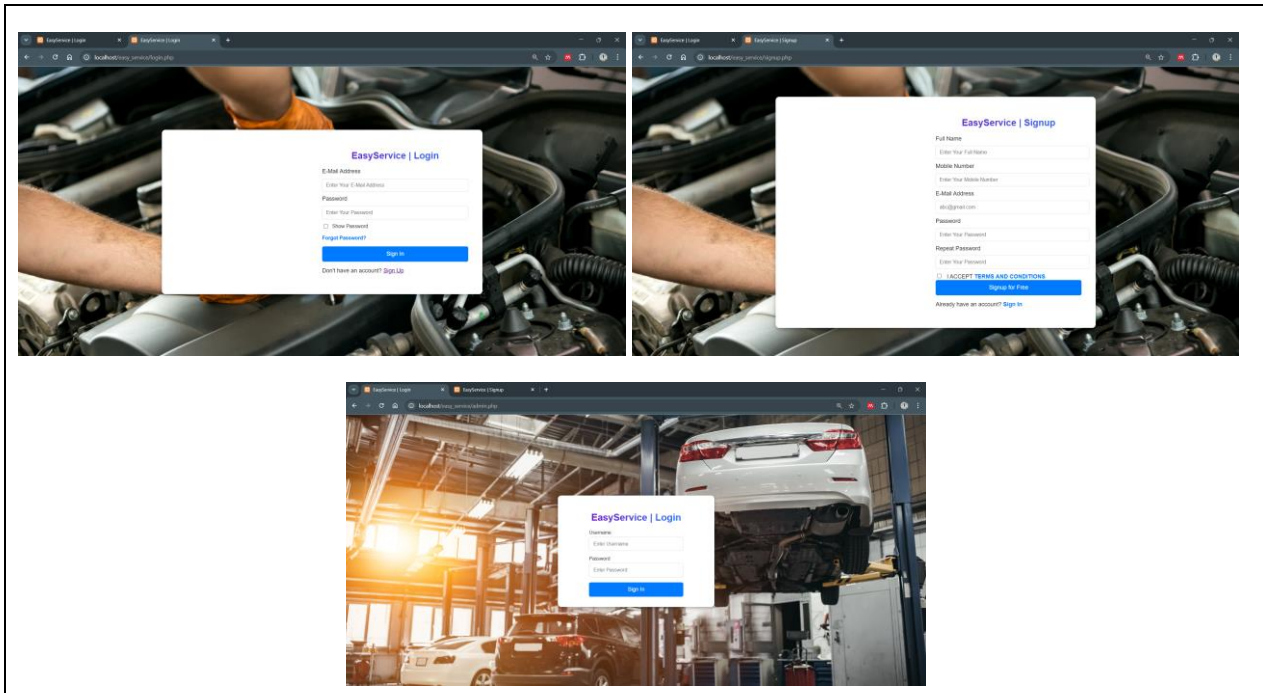
The EasyService system was developed as a web application using PHP for the backend and MySQL for the database to manage car service bookings efficiently. The front end uses HTML, CSS, and JavaScript to provide a clean and responsive user interface. Development and testing were done using Visual Studio Code and XAMPP. Key modules include User Registration and Login, Service Booking, Mechanic Assignment, Price Management, Notification System, Feedback, and Report Generation. Each feature was tested locally to ensure smooth performance and meet user needs effectively.

### 5.1 Interface Design

The interface design for the EasyService Car Service Management Application was implemented using React Native, ensuring a responsive and user-friendly experience. The design emphasizes simple navigation, accessibility, and consistent functionality for customers, mechanics, and admins. Each interface is crafted to deliver smooth and efficient user experience while supporting the application's core features and operations. The following sections highlight the design and functionality of each module, showcasing how the application provides a seamless experience for all users. [7]

### 5.1.1 Sign Up and Login Page

The Sign Up and Login interface enables user authentication, allowing users to either log in or create a new account. The login screen includes fields for entering an email address and password, along with a "Forgot Password?" link for account recovery. The sign-up screen allows new users to enter their name, email, phone number, password, and confirm their password. Both screens feature a clean user interface with a blue and white theme.



**Fig. 10** Login interface Page

### 5.1.2 Appointment and Booking

The EasyService web application streamlines car service bookings through a user-friendly interface. Users can fill in the car and booking details (top-left), then receive confirmation with appointment tracking (top-right). Admins manage incoming service requests (middle-left) and assign them to mechanics via the accepted requests panel (middle-right). Mechanics access a personalized dashboard to view, and update assigned jobs (bottom-left). Additionally, an emergency contact feature (bottom-right) enhances user safety by allowing quick access during urgent situations.

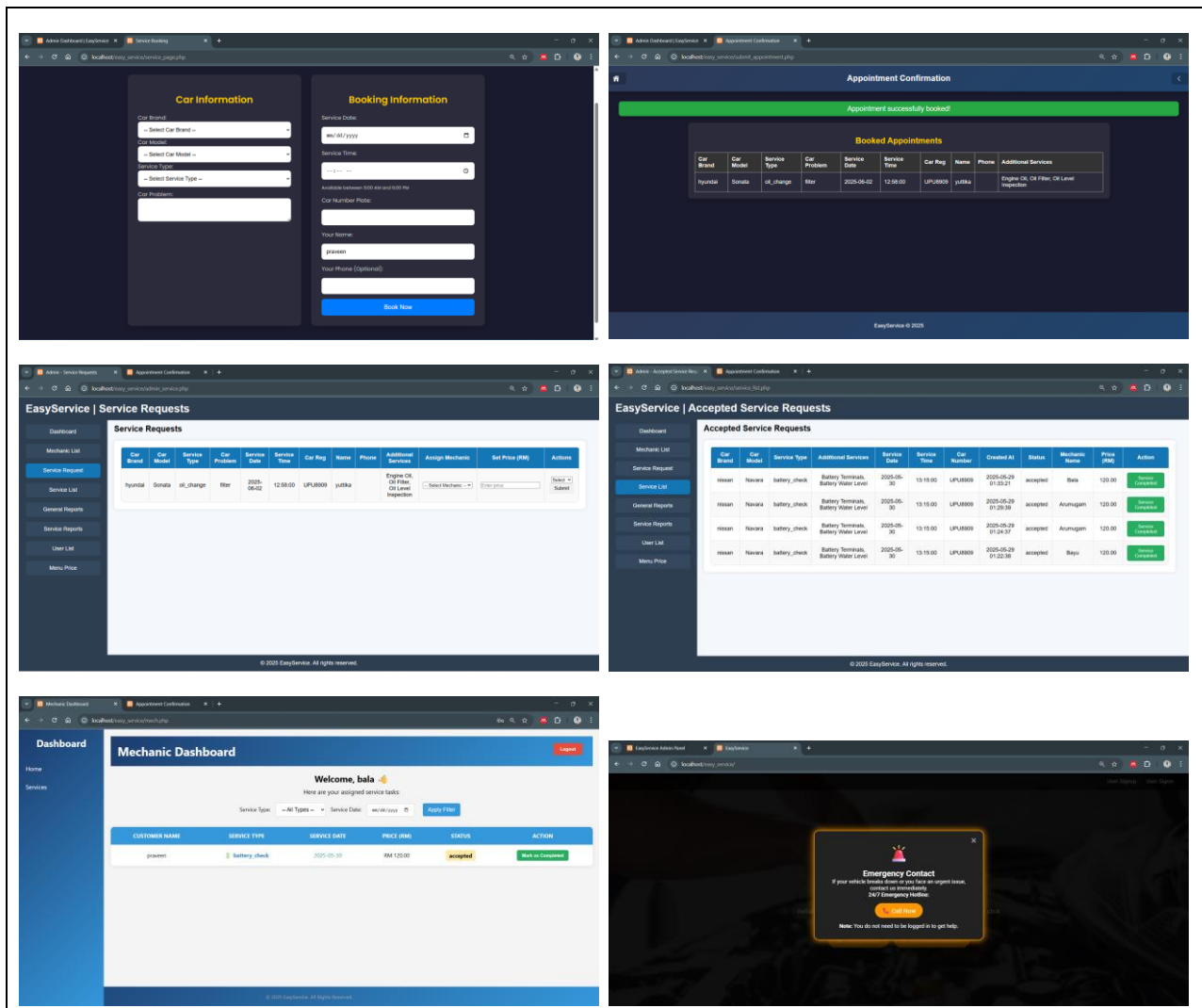


Fig. 11 Service Catalog and Pricing Interface Page

### 5.1.3 Service Catalog and Pricing

The Service Catalog and Pricing Module enables admins to manage the list of available car services in the system. It includes features for adding, updating, and displaying services along with details such as service name, category, and price range. Figure 12 illustrate the interface where the admin can view and manage all listed services. The module presents an organized table that includes options to edit or delete each service. Users can also view the pricing information for all services.

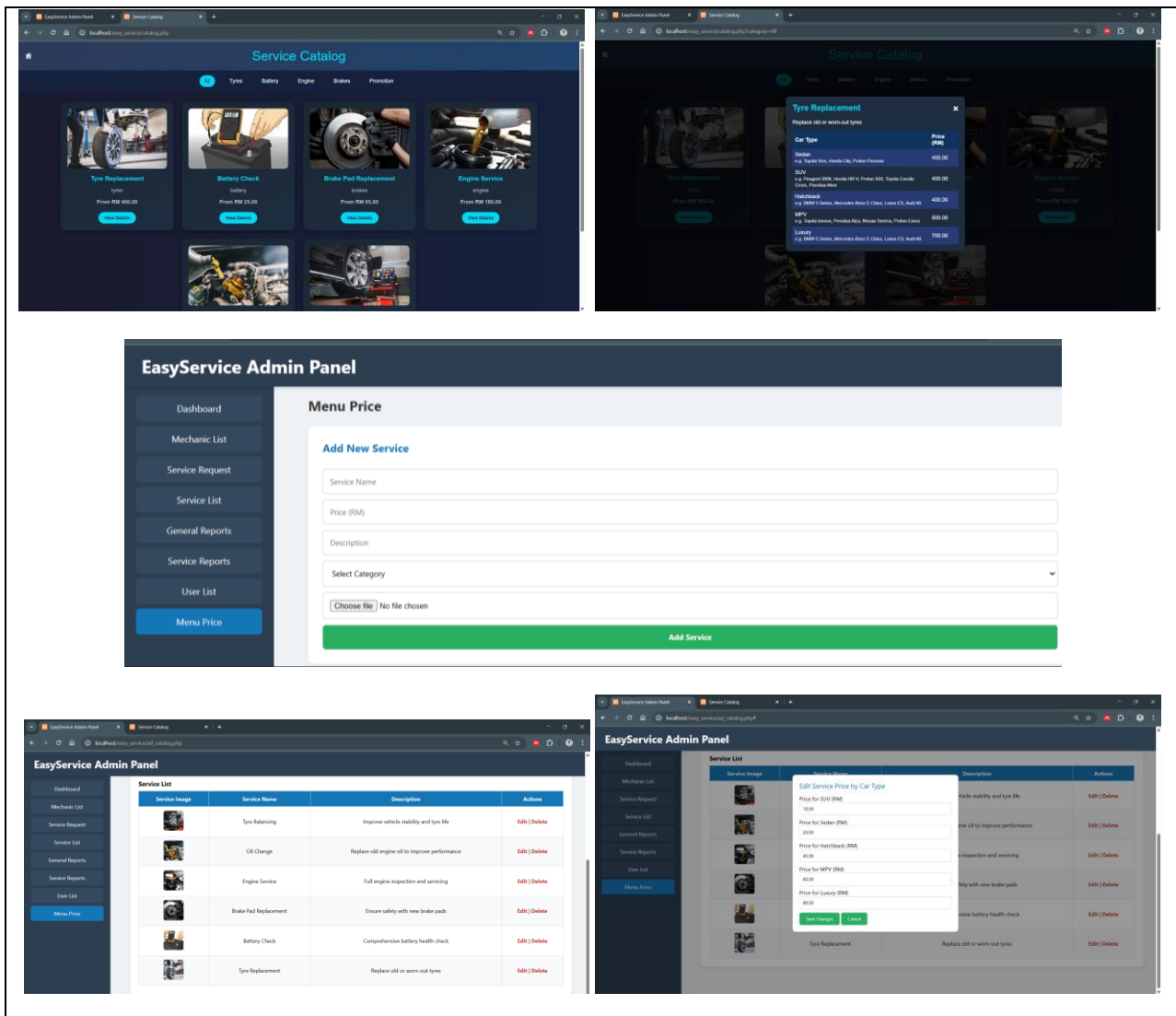


Fig. 12 Service Catalog and Pricing Interface Page

#### 5.1.4 Notification and Reminder

The Notification and Reminder feature in EasyService ensures users and admins stay informed about service statuses and upcoming appointments. The admin dashboard displays a summary of past and upcoming bookings, helping manage schedules efficiently. Additionally, email notifications are sent automatically to users when their service requests are accepted, or rejected, keeping them updated in real time.

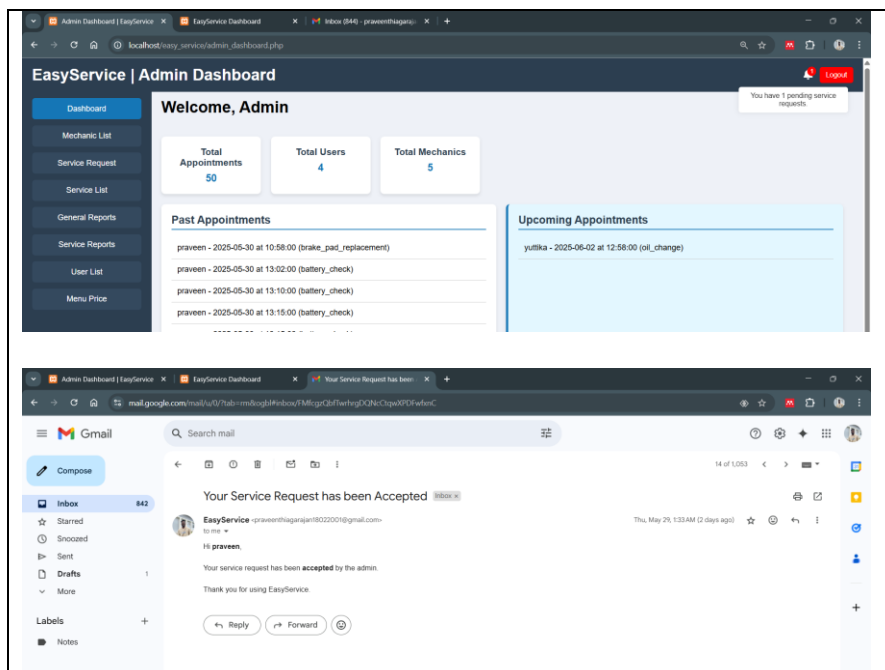


Fig. 13 Notification and Reminder Interface Page

### 5.1.5 Service History and Records

The Service History and Records module provides both users and admins with a detailed view of all completed services. Users can view their service requests along with specific details such as service type, date, mechanic name, and price. Admins can access a comprehensive service report with filtering options and PDF export functionality, ensuring effective record management and transparency.

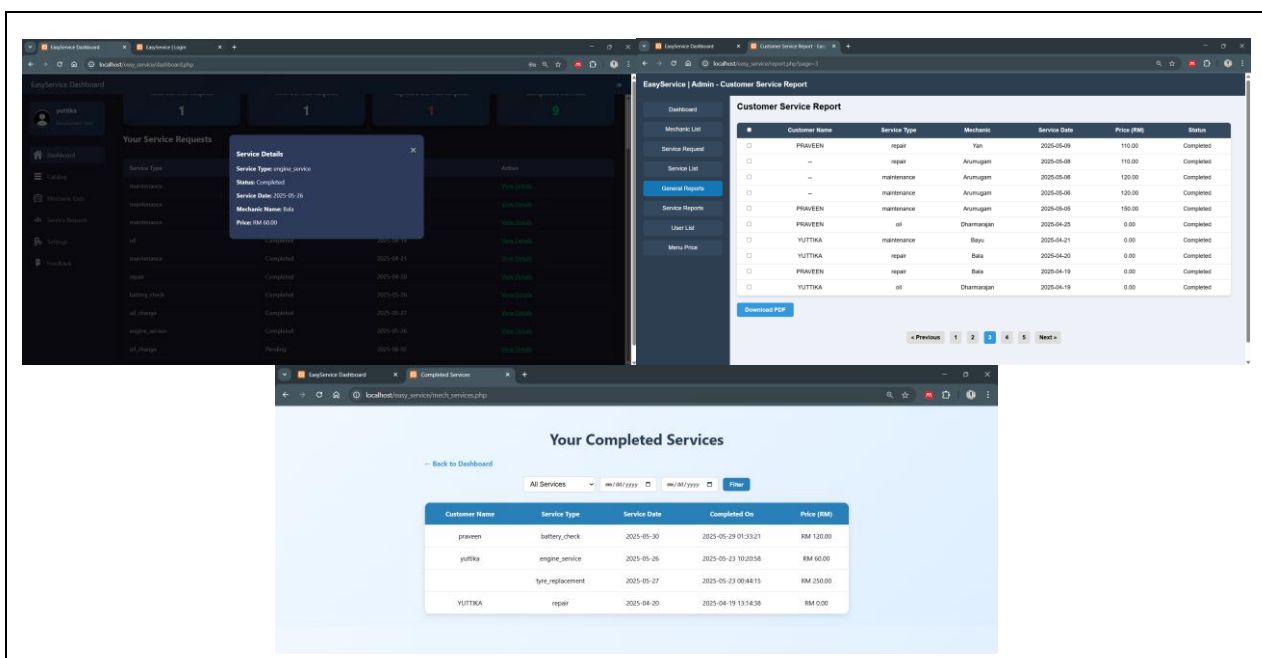
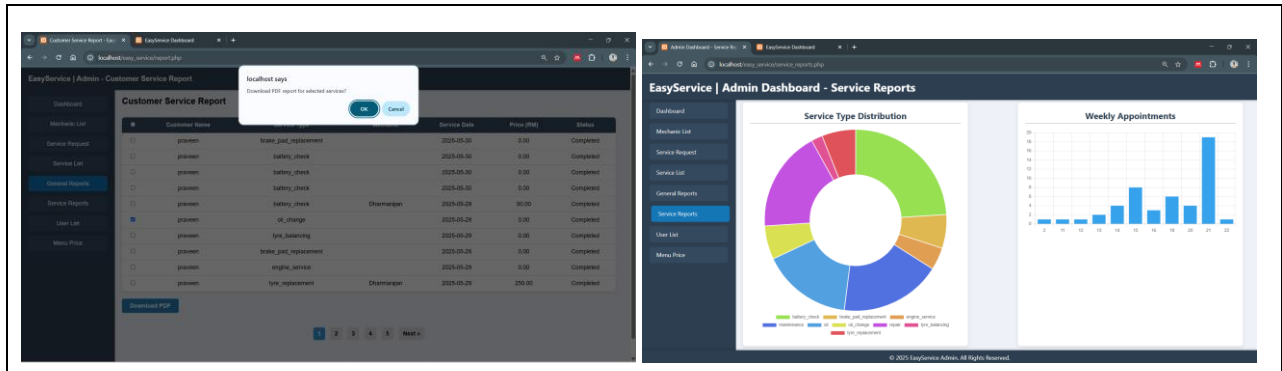


Fig. 14 Service History Interface Page

### 5.1.6 Reporting and Analysis

The Reporting and Analysis module provides comprehensive functionality for viewing, generating, and downloading detailed service reports. It allows admins to access organized data such as service dates, mechanic assignments, and service categories, all presented in a clear and structured format. The system also includes features to export these reports, typically in PDF format, making it easier for admins to maintain accurate

records and conduct performance analysis. By automating the reporting process, this module enhances efficiency and supports informed decision-making through accessible and well-documented service information.



**Fig. 15** Reporting and Analysis Interface Page

## 5.2 Testing

In this section, a test will be carried out to assess the functionality of each module. A User Acceptance Test (UAT) method is utilized to perform testing.

**Table 6** Test criteria

| Test Case ID | Description                                                                                                                    | Expected Result                                                                                                                | Actual                                                                               | Pass/Fail |
|--------------|--------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|-----------|
| M1           | Login and User Management                                                                                                      |                                                                                                                                |                                                                                      |           |
| M1-1         | To check whether administrator, mechanic and user can log in into the system The user should be able to log in into the system | To check whether administrator, mechanic and user can log in into the system The user should be able to log in into the system | The user has successfully logged in into the system                                  | Pass      |
| M1-2         | To check whether the administrator could create accounts for other mechanics                                                   | The administrator should be able to create accounts for another mechanics                                                      | The administrator has successfully created account for another mechanics             | Pass      |
| M1-3         | To check whether the system will restrict login whenever a wrong credential is entered                                         | The system should restrict login when incorrect credentials have been entered                                                  | The system restricted the login when an incorrect or no credentials has been entered | Pass      |
| M1-4         | To check whether the user could change their password if they forgot it                                                        | The user should be able to change their password by E-mail                                                                     | The user has successfully changed their password                                     | Pass      |
| M1-5         | To check whether the system could display the users' information                                                               | The system should be able to display the users' information                                                                    | The system successfully displayed the users' information                             | Pass      |

**Table 6 (cont)**

| Test Case ID | Description                                      | Expected Result                                                   | Actual                                                                           | Pass/Fail |
|--------------|--------------------------------------------------|-------------------------------------------------------------------|----------------------------------------------------------------------------------|-----------|
| M2           | Appointment Booking Module                       |                                                                   |                                                                                  |           |
| M2-1         | Book appointment with all valid details          | Appointment is saved and confirmation shown                       | Appointment saved successfully; confirmation message displayed                   | Pass      |
| M2-2         | Submit appointment form with missing data        | Error message shown prompting required fields                     | Validation error displayed: "Please fill in all required fields"                 | Pass      |
| M2-3         | Book appointment on already occupied slot        | Error or unavailability message is shown                          | Error message shown: "Slot unavailable. Please choose another."                  | Pass      |
| M2-4         | Appointment date selected is in the past         | Error message shown: "Invalid date. Please select a future date." | Error message displayed when past date selected: "Invalid date"                  | Pass      |
| M2-5         | User views a list of their upcoming appointments | List of booked appointments is displayed                          | User dashboard displays list of their scheduled appointments                     | Pass      |
| M2-6         | Admin views all upcoming appointments            | Admin dashboard shows all appointment details                     | Admin panel lists all upcoming appointments with relevant user and mechanic info | Pass      |
| M2-7         | Mechanic views their assigned appointments       | Assigned appointments are displayed correctly to the mechanic     | Mechanic dashboard correctly shows assigned appointments                         | Pass      |
| M2-6         | Admin views all upcoming appointments            | Admin dashboard shows all appointment details                     | Admin panel lists all upcoming appointments with relevant user and mechanic info | Pass      |
| M2-7         | Mechanic views their assigned appointments       | Assigned appointments are displayed correctly to the mechanic     | Mechanic dashboard correctly shows assigned appointments                         | Pass      |
| M3           | Service Catalog and Pricing Module               |                                                                   |                                                                                  |           |
| M3-1         | View all available services                      | Services with name, category, and price shown                     | Services displayed correctly with all relevant details                           | Pass      |
| M3-2         | Admin adds new service                           | Service is added and displayed in catalog                         | New service added successfully and appears in catalog list                       | Pass      |
| M3-3         | Admin edits existing service                     | Updated details are reflected immediately                         | Edited service information updated and visible instantly                         | Pass      |
| M3-4         | Admin deletes an existing service                | Service is removed from the catalog and no longer visible         | Service successfully deleted and no longer appears in the catalog                | Pass      |
| M3-5         | Admin adds a service with missing/invalid data   | Error message prompts for required or correct input               | Form validation triggered; error message shown for missing fields                | Pass      |
| M3-6         | User views services filtered by category         | Only services under the selected category are displayed           | Services filtered correctly by category selection                                | Pass      |

**Table 6 (cont)**

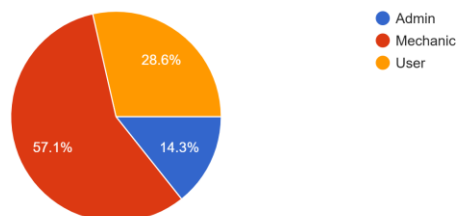
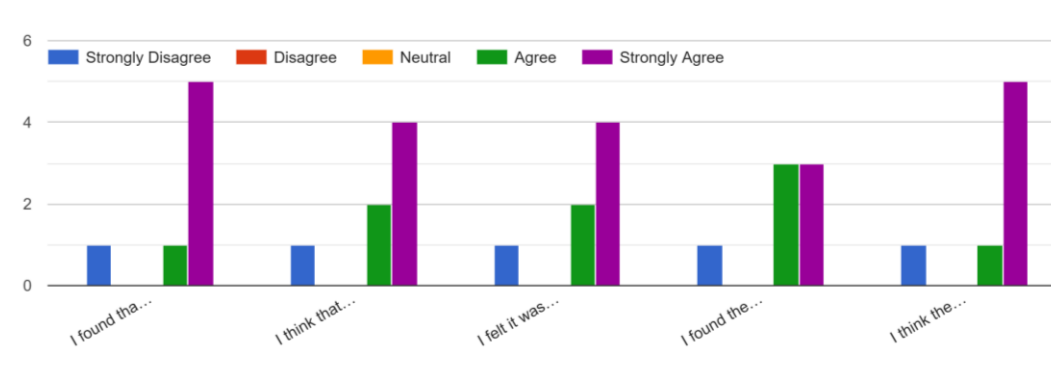
| Test Case ID | Description                                        | Expected Result                                                      | Actual                                                                                                  | Pass/Fail |
|--------------|----------------------------------------------------|----------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|-----------|
| M3-7         | Admin sets or updates service price                | New price is saved and reflected in the service list/catalog         | Admin successfully updated the price; change visible to users                                           | Pass      |
| M4           | Notification and Reminder Module                   |                                                                      |                                                                                                         |           |
| M4-1         | User receives notification after booking           | Notification appears in dashboard or via email                       | Notification displayed on user dashboard after successful booking                                       | Pass      |
| M4-2         | Mechanic receives service assignment alert         | Mechanic dashboard shows new task assigned                           | Mechanic sees newly assigned service listed in dashboard with alert icon                                | Pass      |
| M4-3         | Notification not shown to unrelated user           | Notification only shown to intended recipient                        | Only the relevant customer or mechanic sees the notification                                            | Pass      |
| M4-4         | Notifications are persistent                       | Notifications remain visible until explicitly dismissed or read      | Notifications remain until user marks them as read or dismisses manually                                | Pass      |
| M4-5         | Notifications handle simultaneous multiple events  | Multiple notifications appear separately without overlapping or loss | All triggered notifications displayed correctly without overlap; each event shown as a distinct message | Pass      |
| M5           | Service History and Records Module                 |                                                                      |                                                                                                         |           |
| M5-1         | View all completed services for user               | List of past services displayed with details                         | Completed service records displayed correctly on the user dashboard                                     | Pass      |
| M5-2         | Filter service history by date                     | Only services within date range are shown                            | Filtering works as expected; only services in the selected range are shown                              | Pass      |
| M5-3         | View service details from history                  | Full-service report opens with download option                       | Service details displayed; PDF download button functional                                               | Pass      |
| M5-4         | Unauthorized access to another user's history      | Access denied message displayed                                      | System restricts access and shows "Access Denied" or redirects to login page                            | Pass      |
| M5-5         | Download completed service report as PDF           | PDF files are generated and downloaded                               | PDF download works properly and contains accurate service details                                       | Pass      |
| M5-6         | Search service history by service type or mechanic | Results match selected filter criteria                               | Services filtered correctly by service type or mechanic name                                            | Pass      |

**Table 6 (cont)**

| Test Case ID | Description                                | Expected Result                                            | Actual                                                                   | Pass/Fail |
|--------------|--------------------------------------------|------------------------------------------------------------|--------------------------------------------------------------------------|-----------|
| M6           | Reporting and Analytics Module             |                                                            |                                                                          |           |
| M6-1         | Admin views monthly service report         | Report chart or table with summary shown                   | Monthly report displayed with accurate summary and visual chart          | Pass      |
| M6-2         | Export report to PDF                       | PDF file generated and downloaded                          | PDF report generated successfully and downloaded with correct formatting | Pass      |
| M6-3         | Filter report by date range                | Only records within the selected date range are shown      | Reports filtered accurately based on selected date range                 | Pass      |
| M6-4         | Generation report on specific service type | Report includes only entries for the selected service type | Report successfully generated for chosen service type                    | Pass      |

### 5.2.1 User Acceptance Testing

Roles  
7 responses

**Fig. 16 UAT results 1****Fig. 17 UAT results 2**

The UAT included questions about system ease of use, interface clarity, booking functionality, and overall satisfaction. Most respondents rated the system positively, with most selecting “Agree” or “Strongly Agree” for all usability statements. This feedback confirms that EasyService is user-friendly, efficient, and well-suited for managing car service appointments, supporting the system’s goal of improving service operations at *Day-by-Day Auto Parts*.

## 6. Conclusion

The EasyService Car Service Booking System effectively addresses key challenges in the car service industry, including inefficient booking processes, low customer satisfaction, and resource mismanagement. Developed using PHP and MySQL, the system streamlines service scheduling, enables real-time customer feedback, and improves overall operational efficiency. User testing showed notable time savings, increased user satisfaction, and smoother coordination between customers, mechanics, and administrators. Despite its benefits, the current system has limitations, such as the absence of mobile application support and limited analytics for performance tracking. Future improvements could include predictive maintenance, mobile app integration, automated notifications, and advanced reporting features to enhance system functionality and user engagement. The modern, user-friendly interface of EasyService highlights its potential to transform car service management and support the long-term digital transformation of the automotive service sector [9].

## 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 interest regarding the publication of the paper.

## Author Contribution

*The authors confirm contribution to the paper as follows: **study conception and design:** Praveen Thiagarajan, Mohamad Firdaus Ab Aziz; **data collection:** Praveen Thiagarajan, Mohamad Firdaus Ab Aziz; **analysis and interpretation of results:** Praveen Thiagarajan, Mohamad Firdaus Ab Aziz; **draft manuscript preparation:** Praveen Thiagarajan, Mohamad Firdaus Ab Aziz. All authors reviewed the results and approved the final version of the manuscript.*

## References

- [1] A. A. Y. Irwanto, A. Solehudin, and A. Voutama, "Perancangan pembuatan aplikasi rental kendaraan berbasis website," *\*Elkom: Jurnal Elektronika dan Komputer\**, vol. 15, no. 1, 2022, doi: 10.51903/elkom.v15i1.621.
- [2] C. S. Reddy and P. Savant, "A survey on car service slot booking system," *\*International Journal of Research in Applied Science and Engineering Technology\**, vol. 10, no. 4, 2022, doi: 10.22214/ijraset.2022.41448.
- [3] M. N. M. Zaki and N. Azizan, "The planning process for USIM students' car booking," *\*Malaysian Journal of Science Health & Technology\**, vol. 9, no. 1, 2023, doi: 10.33102/mjosht.v9i1.332.
- [4] D. Dani and E. S. Eriana, "Metode rapid application development dalam perancangan sistem informasi booking service kendaraan," *\*Sainstech: Jurnal Penelitian dan Pengkajian Sains dan Teknologi\**, vol. 32, no. 3, 2022, doi: 10.37277/stch.v32i3.1358.
- [5] M. F. M. Sam, "The factors which influencing users' behavioral intention towards using online booking system for car service at car service centre in Malacca," *\*COJ Electronics & Communications\**, vol. 1, no. 4, 2018, doi: 10.31031/cojec.2018.01.000517.
- [6] S. M. Almalki and A. R. Sulaiman, "A cloud-based car service management system," *\*International Journal of Advanced Computer Science and Applications\**, vol. 10, no. 7, pp. 345–350, 2019, doi: 10.14569/IJACSA.2019.0100744.
- [7] P. R. Rani and V. R. Uday, "A web-based platform for automobile service management," *\*International Journal of Engineering and Techniques\**, vol. 5, no. 2, pp. 21–25, 2019.
- [8] K. K. Patel and R. Sharma, "Development of online vehicle service booking system using PHP and MySQL," *\*International Journal of Computer Applications\**, vol. 162, no. 9, pp. 12–15, 2017, doi: 10.5120/ijca2017913700.
- [9] M. S. Zakaria, R. Yusof, and M. F. A. Ghani, "Mobile application for vehicle service booking using agile methodology," *\*International Journal of Advanced Trends in Computer Science and Engineering\**, vol. 9, no. 3, pp. 3595–3601, 2020, doi: 10.30534/ijatcse/2020/78932020.
- [10] M. H. Lee and S. R. Lin, "Improving the usability of vehicle maintenance service booking systems: A UX-based approach," *\*Journal of Theoretical and Applied Information Technology\**, vol. 97, no. 11, pp. 3156–3164, 2019.