Journal of Informatics and Web Engineering
Vol. 5 No. 1 (February 2026) eISSN: 2821-370X

# A Unified Cloud-Based System for the Tyre Retail Industry: Design and Implementation of Tracktive

**William Wei Loon Theo[1], Siew-Chin Chong[2*], Kian-Ming Lim[3**], Lee-Ying Chong[4], Kuok-Kwee Wee[5]**

[1,2,4,5]Faculty of Information Science and Technology, Multimedia University, Jalan Ayer Keroh Lama, 75450 Bukit Beruang, Melaka, Malaysia.

[3]School of Computer Science, University of Nottingham Ningbo China, 199 Taikang East Road, Ningbo, 315100, China.

**corresponding author: chong.siew.chin@mmu.edu.my; ORCiD: 0000-0003-0421-4367)*

***corresponding author: Kian-Ming.Lim@nottingham.edu.cn; ORCiD: 0000-0003-1929-7978)*

*Abstract* - In the Malaysian concurrent tyre retailing industry, most of the business activities are carried out in the traditional labour-intensive ways. This manual process is prone to inefficiencies, increases operating costs and increases the likelihood of error. The current paper outlines Tracktive, a unified and cloud-based solution that was conceptualized and built to automate the industry by centralizing the ordering of tyres, inventory and order management to a single cloud-based platform. Tracktive is built on microservice architecture and event-driven architecture and deployed with the help of a simulated cloud environment. All the core functional modules are independent, scalable modules, each having its own management. The event-driven architecture is used in inter-module communication via Apache Kafka, which allows asynchronous communication between each microservices. A specific Application Programming Interface (API) Gateway is deployed to avoid unauthorized access and uses JSON Web Token (JWT) authentication and Bcrypt hashing. The system is finally tested with the help of LocalStack, which emulates Amazon Web Service (AWS) services, thus, proving the cloud-native design of the system cost-efficiently and effectively. Tracktive have the potential to improve the accuracy of operations, streamline the processes, and grant the business more agility by replacing the outdated manual processes with a centralized digital system, which could also be a driving force of digital transformation in the tyre retailing industry.

*Keywords— Cloud Computing, Microservices Architecture, Event Driven Architecture, Tyre Retail Industry, Digital Transformation, Security.*

*Received: 1 April 2025; Accepted: 27 July 2025; Published: 16 February 2026*



Journal of Informatics and Web Engineering
https://doi.org/10.33093/jiwe.2026.5.1.12

Published by MMU Press. URL: https://journals.mmupress.com/jiwe

## 1. INTRODUCTION

The current digitalization of the industries in Malaysia has left certain sectors stay behind. The tyre retailer industry in Malaysia is one of the significant examples [1], [2]. Although the government and the country have been trying to promote the use of digital tools, most small and medium sized enterprises (SMEs) within the industry opt to continue the application of traditional and manual processes in key activities like manual inventory management or order placement with phone calls [3]. This reliance on the old ways of handling business processes such as phone calls, emails and face-to-face sales representatives leads to inefficiencies, frequent miscommunications and poor user experience on both sides of the retailer-supplier relationship. To address these issues, this paper proposed Tracktive which a unified cloud-based system that is tailored to the tyre retail industry in Malaysia.

Tracktive is a system that is meant to digitalize and simplify the main business processes of tyre retailers and suppliers on a centralized platform. The system allows safe tyre ordering between retailers and suppliers, integrated inventory, order and payment management with a role-based access control. It is implemented as a microservices architecture where each functional part of the system including the product, order, user, authentication and inventory services is deployed separately as containerized services with the help of Docker. It is eventually orchestrated and tested in a cloud environment simulated with LocalStack. Security is a critical aspect of cloud computing, particularly for systems that manage sensitive business data. Recent studies have explored various methods to enhance cloud security, including hybrid cryptography approaches that leverage both symmetric and asymmetric encryption techniques to ensure data confidentiality and integrity [4]. Tracktive addresses similar concerns by implementing JSON Web Token (JWT) authentication traffic routing to secure the backend and Bcrypt encryption to encrypt sensitive data to ensure security standards [5]. Moreover, Kafka is incorporated to support asynchronous communication, which increases the responsiveness and scalability of the system.

There are three major objectives to be addressed by Tracktive. Firstly, Tracktive digitize and integrate several business processes between tyre shops and suppliers with a centralized system, replacing manual phone calls and varied processes to reduce reliance on human labour. Secondly, Tracktive aims to enhance the security of the systems through implementing an API gateway with JWT authentication in the system architecture to block unauthorized access to the system. Finally, Tracktive ensures the continuity of business through providing a scalable and fault-tolerant cloud-based solution. Its architecture embodies the goals, utilizing cloud-native principles like containerized deployment, event-driven communication and IaC to ensure high system availability and fault tolerance. These technical solutions directly solve the pain points that were discovered, including system downtimes, miscommunication, and order mismanagement, in addition to making users more confident in digital platforms.

Tracktive adopted microservices architecture in terms of implementation. Although monolithic architecture [6] is preferable in small-scale systems due to their cost-effectiveness, ease of management and the capability to offer real-time access to data, they have serious drawbacks as systems scale up. These are scalability limitations, the inability to isolate faults, and the lack of development agility as the number of users or the features set grows. On the other hand, microservices architecture offers key advantages that could help to address the issues and benefit in scalability, fault tolerance and maintainability. As each of the services in the microservices architecture can be developed and deployed independently, it enables the system to be able to react more freely to future expansion, changing needs, and localized failures; failure of one service would not impact the entyre platform. Since Tracktive is a product aimed at digitizing and unifying the tyre retail business in Malaysia by catering to a wide variety of stakeholders that encompasses retailers and suppliers, this microservices architecture is a strategic decision that would guarantee flexibility, performance, and resiliency of the system in the long run.

As a result, Tracktive offers a robust basis to change the tyre retail market in Malaysia, which is currently characterized by the lack of integration and manual processes, to an integrated and digital-first environment. It enables companies to work more effectively, react quicker to market fluctuations, and make informed choices by seeing data. Tracktive proves that cloud-based innovations can lead to real change in underserved industries by overcoming the historical obstacles to digitalization, which are scalability, fault tolerance, and cybersecurity.

## 2. LITERATURE REVIEW

### *2.1 Background Study*

Malaysian tyre retail industry is still mostly reliant on traditional approaches to conducting business, even though the country is trying to encourage digitalization among small and medium-sized enterprises. The market structure of this industry is highly fragmented, and the process of placing orders is highly manual as it is done over the phone, email, and in-person with sales representatives [1]. These practices lead to inefficiency, errors and breakdown of communication which end up increasing the cost of operations and making the business less responsive [1], [3].

In this regard, digital transformation cannot be limited to e-commerce integration anymore but a comprehensive reorganization of internal processes with the help of scalable and secure platforms is required. Nevertheless, most of the tyre retailers in the industry especially small and medium-sized businesses, are still reluctant to implement such solutions because of cost and perceived technical complexity. As a result, the majority of tyre retailers in Malaysia cannot access real-time inventory information, efficient ordering processes, or regular communication channels which eventually causing common problems like stock management, lost sales, and order delays frequently happened [3].

The development of e-commerce systems in different parts of the world has proved that cloud-based solutions, along with user focus design, can transform the way companies communicate with their partners and customers. Cloud platforms eliminate the need for heavy infrastructure investment while offering scalability, accessibility, and enhanced data security [7], [8]. There are several similar solutions in the market. However, digital solutions like Tyre Shop Wizard and Tyre Experts are either too limited in scope or tailored for other regional markets. Besides, the tyre retail business in Malaysia still does not have a single digital platform that specifically serves the B2B processes between tyre retailers and suppliers.

Although the tyre retail industry in Malaysia presents a fragmented industry with many independent retailers and suppliers that do not coordinate their activities centrally, a microservices architecture is reasonable [9]. The framework provides unique benefits in terms of modularity and development agility, where each functional component, such as ordering, inventory, user management, and payment, and others, can be developed, deployed, and scaled separately. This kind of independence is essential in the case of future improvements, such as the integration of AI-based recommendation systems, the addition of delivery and order tracking features, or the introduction of new user roles, like end users who purchase tyre or logistic partners, which would require new services without interfering with the existing ones. Though monolithic systems are simpler and cheaper to put in place in the short run, they are less flexible to changes in the requirements and complexity of the system. Therefore, the choice of microservices-based approach in this scenario facilitates long-term maintainability, flexibility, and scalability that is consistent with the overall digital transformation goals in Malaysia.

Results obtained from the survey conducted confirm there is need to change. Most of the tyre retailers in Malaysia continue to operate in manual method as they still monitor inventory with spreadsheets or handwritten records and placing order through phone call. The results indicate that more than 97% of the stakeholders surveyed expressed a strong interest in using a cloud-based system that integrates centralized ordering, safe payment integration and inventory management to facilitate their business transactions [10]. Such expectations are consistent with the general trend in the world of digital-first platforms.

In short, technological solutions are available but they are not always able to solve the particular operational problems of the Malaysian tyre retail ecosystem. A customized cloud solution, which combines the most important capabilities such as ordering, inventory management, order management is one of the possible approaches to the modernization of this industry, increased transparency of the operations, and the possibility to make informed and data-driven decisions.

### *2.2. Existing Systems*

Several digital platforms have been developed all over the world to facilitate the business process of retailing and e-commerce. However, most of them are not directly related to the Business-to-Business (B2B) relationship in the Malaysian tyre retailing market. The Tyre Shop Wizard is one of the existing systems examples that is a cloud-based platform that intended to help tyre shops and service centres, with the main emphasis on internal inventory and point-of-sales systems [11]. It allows the retailers to control the tyre inventory via web interface, offering features like adding products, updating products, and tracking the inventory. Nevertheless, it is only focused on internal management and its functionality is very scarce in terms of supplier interaction or external order coordination.

Besides, another example of existing systems is the Tyre Experts which is a United Kingdom-based marketplace that is built around tyre retailers and consumers to buy tyres and get vehicle maintenance services [12]. It enables users to navigate products, add products to a shopping cart and book an appointment to have tyres installed. Although Tyre Experts is an effective B2C platform, it does not facilitate the backend business process like order management between the tyre retailers and the suppliers, thus it is not ideal to serve the wholesale-based tyre retailing system in Malaysia.

Another significant sample of existing systems that serve as a good case study is Shopee, which is the biggest general-purpose e-commerce platform in Southeast Asia [13]. It offers features like logistics tracking, secure payments, and business analytics through its seller centre. Despite its high versatility, Shopee is not a specialized solution to the tyre retailing and does not have the customization and real-time inventory management of tyre shops. It is an architecture-wise biased towards direct consumer sales as opposed to inter-business sales, which makes it less applicable to a B2B situation.

All of these platforms provide partial functionality applicable to the tyre retail industry, including inventory management, items listing, or analytics dashboard. However, none of them are provide a solution that is specifically designed to meet the needs of tyre retailers in Malaysia, who need to organize complex ordering and inventorying activities with various suppliers. These drawbacks highlight the necessity of a single, cloud-based solution that would directly respond to the business processes and operational pain point that are specific to the industry.

## 3. THE PROPOSED SOLUTION

The solution to this project is the creation of Tracktive, a single cloud-based service that would digitalize the essential business operations of the Malaysian tyre retail business and would cover major concerns like the security of the system, its scalability, and high availability. The implementation is categorized into two key parts which are system development and system deployment. The system frontend and backend are included in the system development phase. The frontend offers a user-friendly interface depending on the design needs of the system, whereas the backend is in charge of business logic implementation and the support of such features as tyre ordering and inventory management. The backend uses microservices architecture, and each service, including API Gateway, Authentication, Product, Order, Payment, and User, is independent and communicates through a message broker to guarantee modularity and fault isolation. The system deployment follows the cloud-native objectives of the project by using LocalStack to emulate Amazon Web Service (AWS) cloud services when developing the system. This will allow thoroughly testing the cloud functionality of the system in a controlled environment, without the complexity and cost of deploying to a live AWS infrastructure too early. The decision to adopt LocalStack was largely driven by the economic gains it brings in the development stage whereby developers are able to simulate AWS services without having to pay cloud-service costs. The term cost-effective, therefore, is used in a limited sense to mean the minimization of deployment costs in the development process; it is not used to mean any system performance characteristic like latency or downtime. Furthermore, as the implemented system builds cloud infrastructure using Infrastructure as Code (IaC), supporting deployment on both LocalStack and AWS. This enables reusability of the infrastructure when the system requires production-ready cloud deployment, though some configuration changes are still needed. This architecture ensures a smooth transition, as it is tested locally, proving that Tracktive is technically sound and practical.

### *3.1. Microservices Architecture*

The microservices architecture of the backend of Tracktive presented in Figure 1 is made to guarantee modularity, scalability, and easy maintenance. While the microservice design was primarily based on practical experience gained throughout development, it was partially inspired by architectural guidelines introduced in microservice development references such as the work of Newman [14], which provided foundational methodologies that helped guide the structuring and separation of services in the system. The services are independent of each other, and each service handle business processes on its own domain and was built using Java Spring Boot. The API Gateway, Authentication Service, User Service, Product Service, Order Service, and Payment Service are the basic services of the system backend. The API Gateway is built on top of the Spring Cloud Gateway and serves as a single-entry point to client requests and applies JWT-based authentication to protect sensitive endpoints. Tracktive's API Gateway employs JWT-based authentication to secure endpoints, a critical measure given the risks in cloud environments. Fathullah et al. identified security and compliance issues as prevalent in healthcare cloud projects, a concern equally applicable to

tyre retail data, which Tracktive mitigates through its security architecture [15]. The Authentication Service is concerned with user login; token generation and validation and the User Service intentionally decouples it to make sure that there is a separation of concerns. The User Service is responsible for managing user profiles and role assignments and exposes lifecycle events to other services. The Product Service is in charge of tyre listings, stock levels by supplier, and connects AWS S3 to store images. It also has the capability of multi-tenant product management. The Order Service is in charge of order lifecycle and orchestrates cart operations, order tracking, and order cancellation. The system architecture also adheres to Saga pattern of distributed transactions. The Payment Service handles payment processes and broadcasts corresponding events to trigger the order states. Consequently, the system architecture with the service decomposition design assists in the improvement of fault isolation, allows horizontal scaling, and supports continuous integration and deployment.

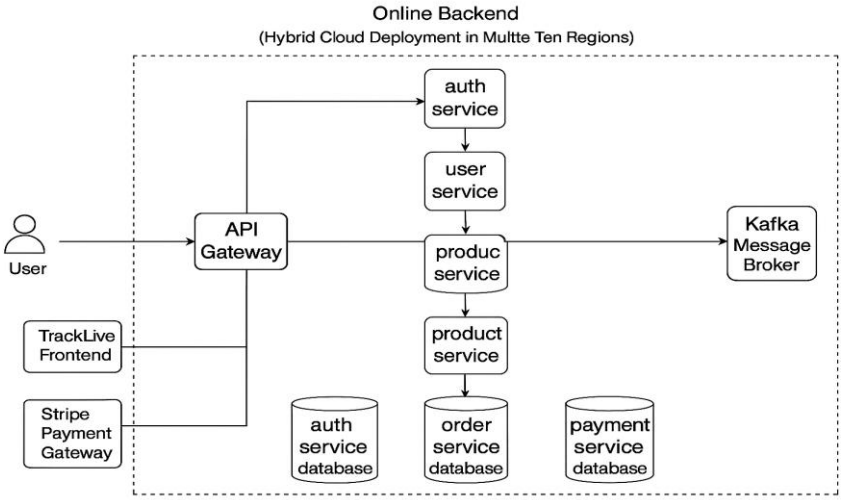


Figure 1. Tracktive's Microservices Architecture

### *3.2. Event-Driven Architecture*

As Figure 2 shows, Tracktive uses event-driven architecture where Apache Kafka is the main message broker to enable asynchronous communication among its microservices [16].

This type of architecture enables services to subscribe and publish domain-specific events without making direct API calls, which ultimately leads to better system resilience and scalability. An example is that the Order Service will publish an order-created event when an order is created which is consumed by the Product Service to reduce the inventory and by the Payment Service to start the payment process. In the event of successful or unsuccessful payment, the Payment Service will broadcast a payment-success or payment-failure event, which will be listened to by the Order Service, to update the order status to reflect this, or to compensate. The decoupled communication model is not only more performant and responsive but also enables future extensibility. For example, new services can be added to the system architecture without interfering with the current workflows. Moreover, Kafka message persistence guarantees that in case a service goes down, events are not lost and processed when the service is back. The application of event driven architecture further decoupling the dependencies between microservices and minimizes cascading failures to support the seamless addition of new services, making the system highly fault-tolerant and extensible.

### *3.3. Cloud Infrastructure Architecture*

In order to verify that the proposed solution is cloud-native and cost effective at the development stage, Tracktive is eventually run in a simulated cloud environment with LocalStack after development [17]. Such local simulation would simulate the core AWS services like S3 or RDS and allow end-to-end testing of cloud-native capabilities without any deployment expenses in the actual AWS cloud environment [8], [18-19].

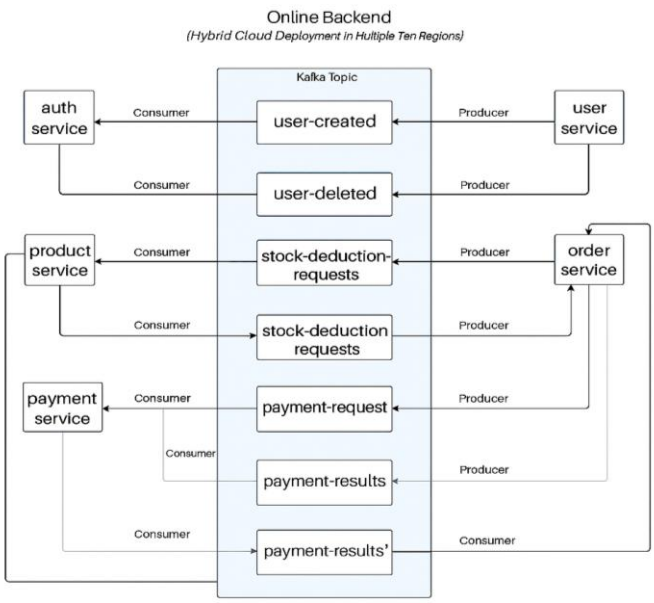


Figure 2. Tracktive's Event-Driven Architecture

Figure 3 shows the basic cloud infrastructure architecture of Tracktive simulated in LocalStack. All services are containerized with Docker, which offers the same environment in all development, testing, and future production. Kafka is also deployed in Docker containers to support event-based messaging. The frontend application, which was built using React, would communicate with backend services via the API Gateway. The system is built keeping in mind the infrastructure-as-code principles, which makes it easy to deploy to actual AWS environments when rolling out the production. This simulated infrastructure proves the compatibility of the platform with cloud deployment models and proves that Tracktive can scale reliably with high availability and performance in the real world.

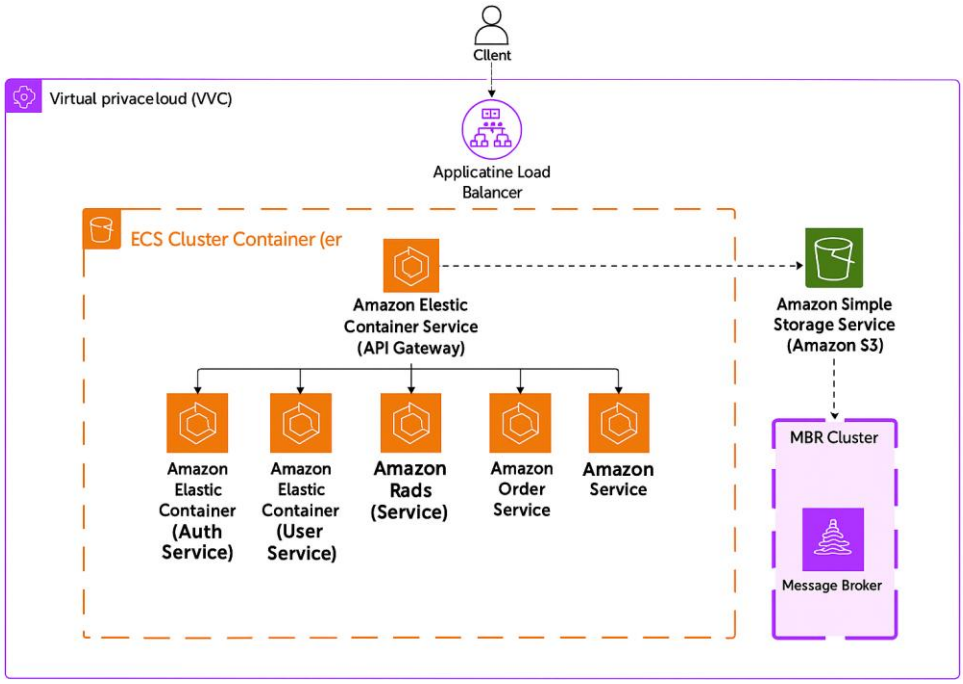


Figure 3. Tracktive's Cloud Infrastructure Architecture (Simulated through LocalStack)

## 4. IMPLEMENTED RESULT

Tracktive, the proposed solution has been conceived and operated as a web-based application that is specifically established to revolutionize the tyre retailing market in Malaysia. Focusing on the drawback of inefficiency of traditional manual practice, the system digitalized and centralized the fundamental business activities of tyre purchasing, order management, and inventory management. The three main categories of users that the platform will support include system administrators, tyre retailers, and tyre suppliers. Specific functions and features will be provided to each of the users based on their role through role-based access control. The retailers are given the ability to explore and order tyre products of various suppliers in one and centralized marketplace. On the other hand, the suppliers can manage their own stocks and product listing. Administrators, in their turn, control system-wide processes, such as managing system users and managing products within the system. The solution also contains secure authentication, real-time update, and dynamic user interface based on a latest-generation technology stack. Key features which have been released in Tracktive are covered below. Every block presents a particular system module, describes its main capabilities and gives subsidiary interface screenshots that show this module usage and design. All these functions together make up a complete and cloud-enabled platform that helps Malaysian tyre companies shift to a well-organized, scalable, and digitally interconnected environment.

### *4.1. Centralized Market*

Tracktive introduces a centralized marketplace feature designed specifically for tyre retailers, enabling them to browse and place orders with multiple suppliers through a single, unified interface. As depicted in Figure 4, the centralized market page provides retailers with a large inventory of tyre products carried by a pool of suppliers where they can contact suppliers with no need of manual requests or discontinuous method of communication. The interface includes convenient search and filter options which allow choosing tyres according to basic parameters, such as brand, size, width, aspect ratio, and rim diameter. Being able to filter products based on suppliers puts retailers in a position to determine which ones suit them most in terms of availability, preference of brand name or even prices. This central ordering would reduce the complexity of ordering, and at the same time make the market more transparent, such that retailers would make buying decisions based on comparisons of several different suppliers' offerings in real time. Every product page has all the basic information such as the quantity of the stock, the price per unit and product specifications as well as the option of adding the product to the cart and placing the order at the moment. Tracktive saves a lot of time in supplier communications and makes operations of tyre retailers more efficient by all means since all the relationships are centred around digital environment.

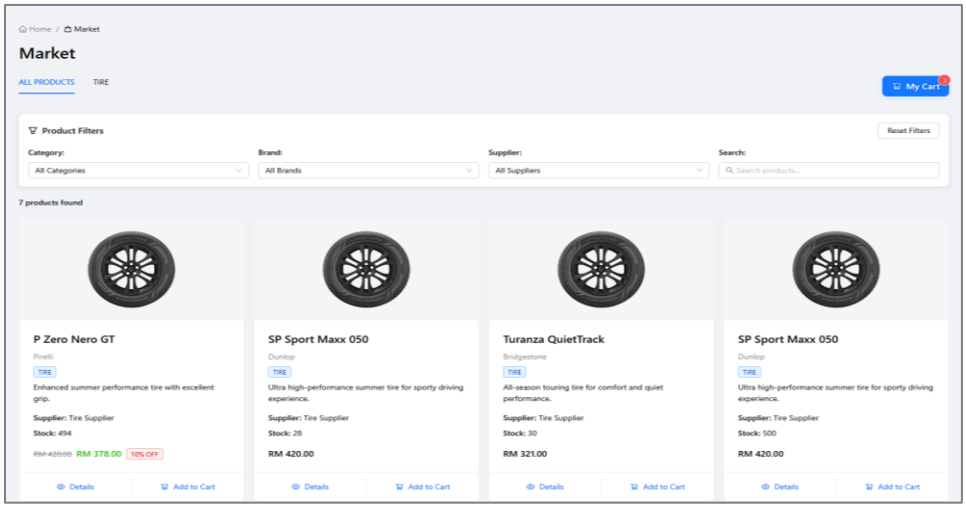


Figure 4. Centralized Market Page

### *4.2. Inventory Management*

Tracktive provides tyre retailers with an inventory-management system that helps the professionals to track and manage the inventory on hand effectively. Considering the nature of the industry that requires a wide range of tyre

variants and standards in terms of size, brand, and specification, an efficient and real-time by the moment inventory field is critical in ensuring that the products remain in line and possible to bring out when customer orders have been received. The interface displayed in Figure 5 can be used by the users and offers an opportunity to see all the tyres with the warehouse-stock tyres, their details including name, quantity, and other characteristics.

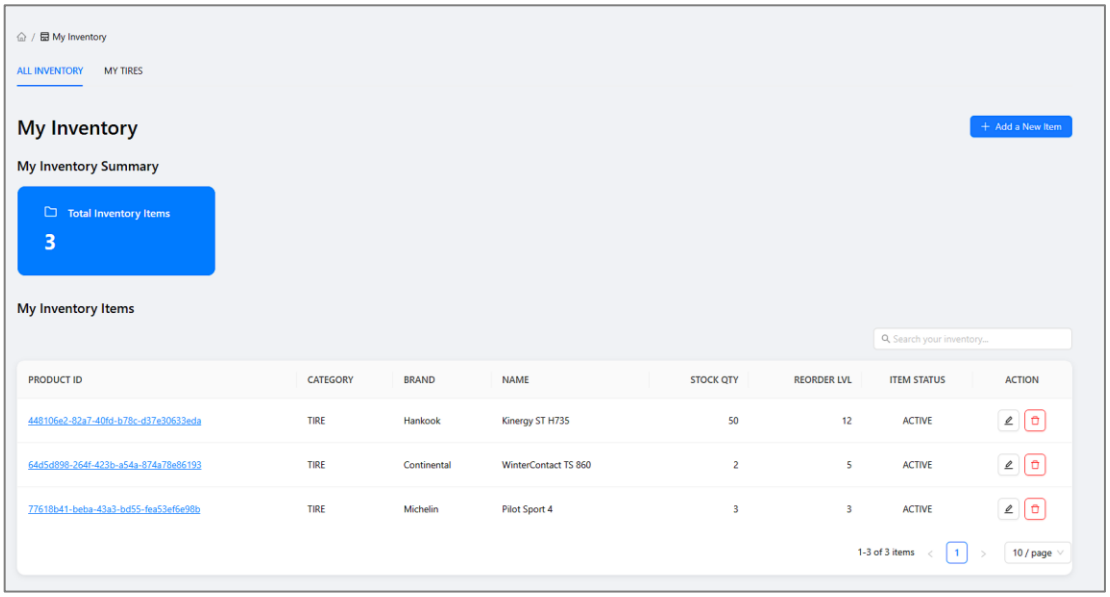


Figure 5. Inventory Management Page

*4.3 Order Management*

Tracktive has a robust Order Management module that allows tyre retailers and suppliers to observe, inspect and control the orders in a neat and transparent interface. The module enables a fully transparent view of the transaction history, the status of the order as well as the details of the counterpart and allows user to effectively monitor their current state of the business transactions. Since, as shown in Figure 6, the Order Management page shows a list of all the related orders, Order ID, name of the counterpart (supplier or retailer), total amount, the current status (e.g. pending, confirmed, completed), and time stamps are all necessary information. Tracktive improves the coordination of the company and its B2B partners, minimizes miscommunication, and facilitates easier relations between the companies within the tyre retail environment by offering both parties a transparent and unified way of accessing data on orders.

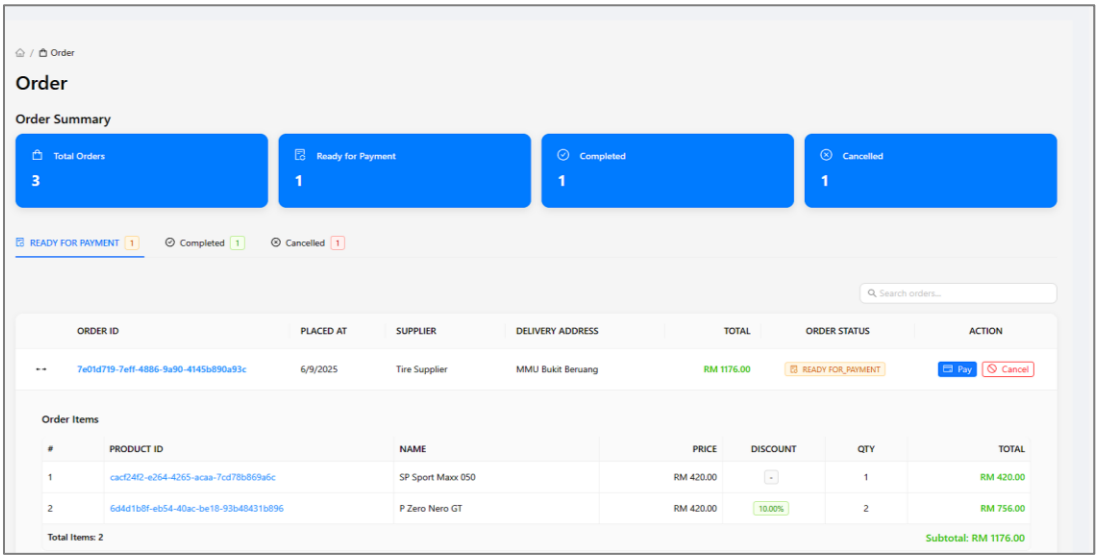


Figure 6. Order Management Page

### *4.4 Payment Management*

With the help of the Payment Management Module in Tracktive, the retailer is given a detailed view and control of payment records associated with tyre orders. Figure 7 indicates that retailers have access to the special Payment History page, where they can find the record of the successfully completed transactions, which include the reference to their invoices and identifiers of their orders, the amount of money and the exact time of being paid.

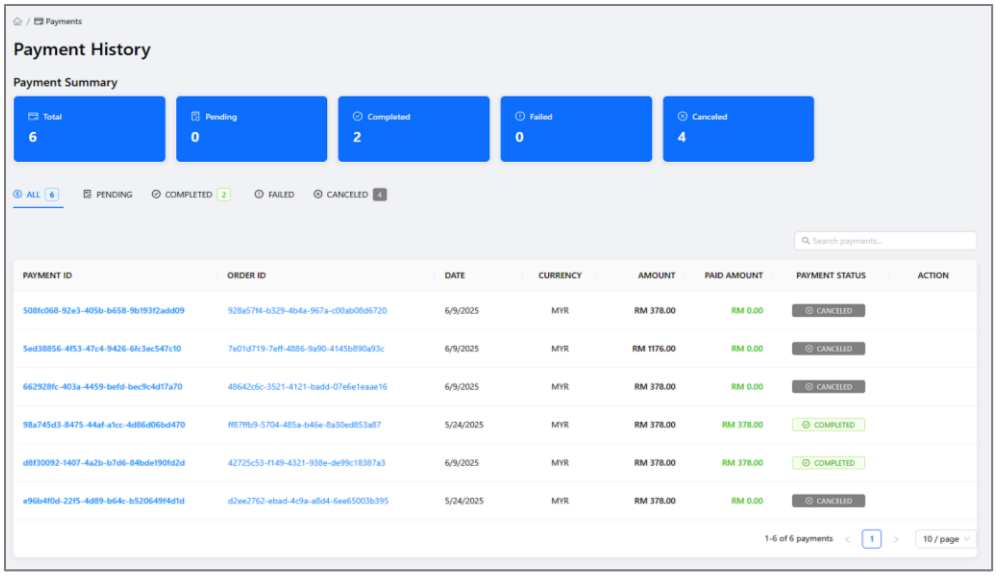


Figure 7. Payment Management Page

### *4.5 Payment Integration*

In a bid to permit the transparent and safe online payment, Tracktive has introduced the Stripe payment system in its checkout process [20]. Stripe is a well-known Payment Card Industry (PCI)-compliant payment platform which will make it possible to process the payments in real-time, with the help of credit or debit card. After an order is confirmed, a retailer is directed to a safe Stripe-hosted checkout page as shown in Figure 8 where they may finalize the transaction with the payment method of their choice.

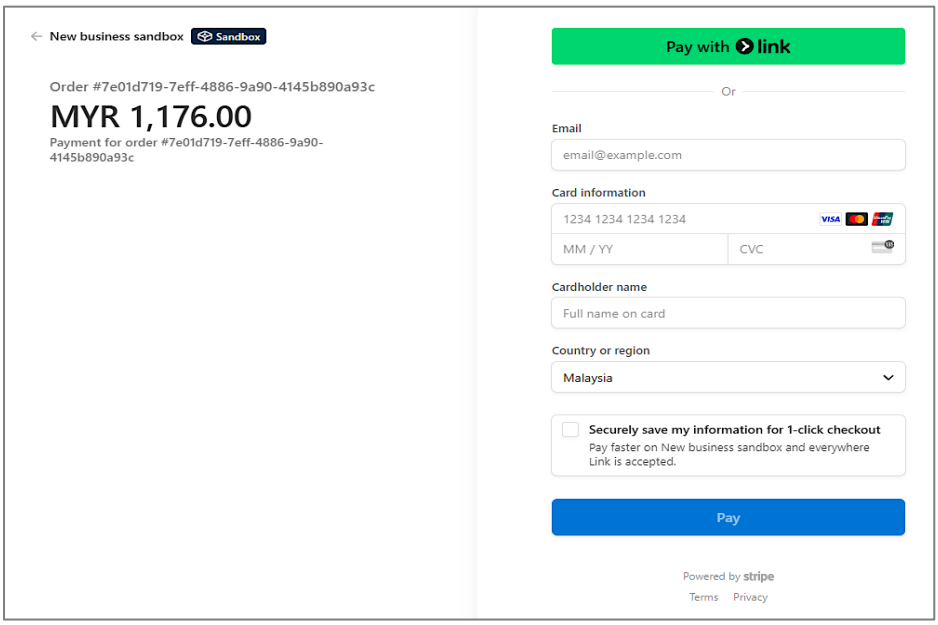


Figure 8. Payment Gateway Page

Integration guarantees that sensitive financial data will be processed securely but it will not be stored on the servers of Tracktive. Such configuration gives an opportunity to assure transaction stability and convenient payment process to tyre retailers as well as to give suppliers and administrators direct access to the transaction status. When the payment is made, Stripe sends an indication to the system in a webhook that goes to affect an update in the status of the order and to record a secure back-end transaction log.

To further strengthen the justification for the deployment of Tracktive, Table 1 presents a comparison of the proposed system with existing platforms currently used in the industry. Unlike general-purpose or consumer-focused systems, Tracktive is specifically designed to address the challenges faced by tyre retailers and suppliers in Malaysia by integrating core B2B functionalities such as centralized ordering, real-time inventory tracking, secure payment processing, and scalable cloud-based deployment.

Table 1. Comparison of Tracktive with Existing Systems

| Feature | Tyre Shop Wizard | Tyre Experts | Shopee | Tracktive (Proposed System) |
|---|---|---|---|---|
| **Target Users** | Tyre retailers | Consumers | General public | Tyre retailers & suppliers (B2B) |
| **Order Management** | Basic | Yes (consumer orders) | Yes | Centralized B2B order system |
| **Inventory Management** | Yes | No | Yes (seller only) | Yes (for both retailer & supplier) |
| **Logistics Tracking** | No | Limited | Yes | Integrated delivery tracking |
| **Payment Integration** | Basic | Yes | ShopeePay, online methods | Secure gateway (Stripe, encrypted) |
| **Security Features** | Limited | Basic | Advanced | JWT Auth, encryption, role-based |
| **Cloud-based** | Yes | Yes | Yes | Yes (cloud-native, scalable) |
| **Industry Fit** | Tyre retail (internal use) | Consumer tyre services | General marketplace | Tailored for Malaysia's tyre industry |

## 5. TESTING

Functional testing is a crucial stage in the software development cycle, which is used to verify that a created system is functioning as per the design it is supposed to be, and as per the requirements it is supposed to be. This is to ensure that each of the functions works well under various conditions, thus evaluating both the components and the interactions. Table 2 to Table 11 demonstrate the results of various test cases performed of each functional module and provide an overall summary of the results, stating whether the system meets its functional requirements.

Table 2. Authentication Module Testing

| Action | Role | Input | Expected Output | Result |
|---|---|---|---|---|
| User login without credentials | Anonymous | User clicks login without entering email and password | Form validation errors: "Please input your email!" and "Please input your password!" | Successful |
| User login with Invalid credentials | Anonymous | User enters incorrect email or password | Toast: "Invalid email or password", HTTP 401 | Successful |
| User login with Valid credentials as Admin | Anonymous | Correct admin email/password | Redirect to /admin/user-management, JWT token stored | Successful |
| User login with Valid credentials as Supplier | Anonymous | Correct supplier email/password | Redirect to /supplier/my-products, JWT token stored | Successful |
| User login with Valid credentials as Retailer | Anonymous | Correct retailer email/password | Redirect to /retailer/my-inventory, JWT token stored | Successful |
| User logout | Admin/Supplier /Retailer | Clicks logout | Session cleared, redirect to login | Successful |

Table 3. User Module Testing

| Action | Role | Input | Expected Output | Result |
|---|---|---|---|---|
| View all users | Admin | Access user management | User table with pagination and search | Successful |
| Create new user | Admin | Add user with valid data | Success message, list refreshed | Successful |
| Create user with Duplicate email | Admin | Email already exists | Error: "This email address is already in use" | Successful |
| Create user with Duplicate SSM | Admin | SSM already exists | Error: "SSM Registration Number has been registered" | Successful |
| Edit user details | Admin | Update user info | Changes saved and reflected | Successful |
| Delete user | Admin | Confirm deletion | User removed from list | Successful |
| Search users | Admin | Search term entered | Filtered user list | Successful |
| View user details | Admin | Click user detail | Modal shows full user info | Successful |

Table 4. Product Module Testing

| Action | Role | Input | Expected Output | Result |
|---|---|---|---|---|
| Add new product | Admin | Add new product | New product added to the system | Successful |
| View all products | Admin | Access product page | Table of all products with supplier info | Successful |
| Edit any product | Admin | Edit product | Changes saved system-wide | Successful |
| Delete any product | Admin | Delete product | Product and related inventory removed | Successful |

Table 5. Supplier Product Module Testing

| Action | Role | Input | Expected Output | Result |
|---|---|---|---|---|
| View supplier products | Supplier | Access "My Products" | Table of supplier's products | Successful |
| View available products | Supplier | Browse system products | View-only table of other supplier products | Successful |
| Add New Product | Supplier | Select available product, fill with required information | Supplier Product added to the system | Successful |
| Update product price | Supplier | Modify price | Price updated | Successful |
| Update stock quantity | Supplier | Adjust stock | Stock updated | Successful |
| Delete own product | Supplier | Delete product | Product removed from catalogue | Successful |

Table 6. Inventory Module Testing

| Action | Role | Input | Expected Output | Result |
|---|---|---|---|---|
| View inventory | Retailer | Access inventory | Table of current stock | Successful |
| Add system product to inventory | Retailer | Select product | Product added with stock quantity | Successful |
| Add tyre to inventory | Retailer | Select tyre | Tyre with specs added | Successful |
| Update inventory item | Retailer | Edit stock/reorder level | Item updated | Successful |
| Delete inventory item | Retailer | Remove item | Item removed from inventory | Successful |
| View available system products | Retailer | Browse products | View of all products | Successful |
| View available system tyres | Retailer | Browse tyres | View of all tyres with specs | Successful |

Table 7. Market and Cart Module Testing

| Action | Role | Input | Expected Output | Result |
|---|---|---|---|---|
| Browse market | Retailer | Access market | Grid view of products | Successful |
| Browse tyre market | Retailer | Access tyre market | Tyres shown with filters | Successful |
| Filter products by brand | Retailer | Select brand | Brand-filtered list | Successful |
| Filter products by supplier | Retailer | Select supplier | Supplier-filtered list | Successful |
| Filter tyres by specifications | Retailer | Use tyre filters | Filtered tyre list | Successful |
| Reset filters | Retailer | Click reset | All filters cleared | Successful |
| View product details | Retailer | Click product | Modal with full details | Successful |
| Add product to cart | Retailer | Add product with quantity | Product added, cart count updated | Successful |
| Add product to cart - Out of stock | Retailer | Add unavailable product | Error: "Out of stock" | Successful |
| View cart | Retailer | Open cart | Cart modal with items and total | Successful |
| Update cart quantity | Retailer | Change quantity | Quantity and total updated | Successful |
| Remove from cart | Retailer | Remove item | Cart updated | Successful |
| Clear Item from cart | Retailer | Click clear | Cart updated with no items left | Successful |

Table 8. Order Module Testing

| Action | Role | Input | Expected Output | Result |
|---|---|---|---|---|
| Place order | Retailer | Place order | Order status: "PENDING", cart cleared | Successful |
| Place order with Empty cart | Retailer | No items in cart | Error: "Cart is empty" | Successful |
| View retailer orders | Retailer | Access orders | Table of own orders | Successful |
| View supplier orders | Supplier | Access orders | Orders related to supplier products | Successful |
| View order details | Retailer/ Supplier | Click order | Modal with full details | Successful |
| Cancel order | Retailer | Cancel before payment | Status: "CANCELLED" | Successful |

Table 9. Payment Module Testing

| Action | Role | Input | Expected Output | Result |
|---|---|---|---|---|
| Initiate payment | Retailer | Click "Make Payment" | Redirect to Stripe | Successful |
| Payment with Invalid order status | Retailer | Pay wrong status order | Error: "Order not payable" | Successful |
| Complete payment | Retailer | Finish payment | Payment completed, order is placed | Successful |
| Cancel payment | Retailer | Cancel payment flow | Return to order page | Successful |
| View payment history | Retailer | View payments | Table with payment transactions | Successful |

Table 10. My Account Module Management

| Action | Role | Input | Expected Output | Result |
|---|---|---|---|---|
| View my account | All roles | Access account page | Display profile info | Successful |
| Update account | All roles | Update profile info | Info updated successfully | Successful |
| Change password | All roles | Enter current + new password | Password updated | Successful |
| Change password - Invalid current password | All roles | Wrong current password | Error: "Incorrect current password" | Successful |

Table 11. Security Testing (Authentication and Authorization)

| Action | Role | Input | Expected Output | Result |
|---|---|---|---|---|
| Access protected route (no token) | All roles | Request without JWT | Error: "Unauthorized access" (HTTP 401) | Successful |
| Access protected route (invalid token) | All roles | Request with expired or tampered JWT | Error: "Invalid token" (HTTP 401) | Successful |
| Access protected route (valid token) | All roles | Request with valid JWT | Access granted to protected resource | Successful |
| JWT validation by API Gateway | All roles | API call with JWT | Gateway validates and forwards request if token is valid | Successful |
| Password encryption check | System Admin | Check user table in DB | Password stored as hash, not in plain text | Successful |
| Role-based access control | All roles | Try accessing other roles resource | Error: "Access denied" (HTTP 403) | Successful |
| Logout session | All roles | Click logout | Token invalidated; access to protected routes denied | Successful |

## 6. CONCLUSION

This article presents Tracktive, an integrated cloud-based solution that has been designed to address the existing inefficiencies in the tyre retail market in Malaysia by replacing the traditional, labour-intensive processes with a scalable and secure digital alternative. With the integration of the core operational capabilities, which are tyre ordering, inventory management, and order processing, into a centralized setting, Tracktive provides a concrete example of digitalization in the industry that has long relied on manual operations. The system has been confirmed to be ready to be deployed in the cloud on a large scale after a proof-of-concept simulation was performed in an AWS environment with the help of LocalStack. The features that have been implemented such as a centralized marketplace, role-based interface, Stripe-integrated payment processing, and transparent order management provide a modern and easy-to-use experience to administrative users, suppliers, and retailers. In the future, Tracktive offers a good starting point to make improvements like the integration of analytics to support demand forecasting, real-time tracking of delivery, and multilingual user interface support. Finally, the platform shows how cloud-native solutions can serve as modernization drivers in the underserved industry, enhancing efficiency, traceability, and collaboration along the supply chain.

## ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their constructive comments and suggestions to improve this article.

## FUNDING STATEMENT

The authors received no funding from any party for the research and publication of this article.

## AUTHOR CONTRIBUTIONS

William Wei Loon Theo: Conceptualization, Methodology, Validation, Writing – Original Draft Preparation;
Siew-Chin Chong: Project Administration, Supervision, Writing – Review & Editing;
Kian-Ming Lim: Methodology, Validation;
Lee-Ying Chong: Project Administration, Writing – Review & Editing;
Kuok-Kwee Wee: Data Curation, Methodology, Review.

## CONFLICT OF INTERESTS

No conflict of interests were disclosed.

## ETHICS STATEMENTS

The paper follows The Committee of Publication Ethics (COPE) guideline. https://publicationethics.org/.

## DATA AVAILABILITY

The data that support the findings of this study are available on request from the corresponding author. The data, which contain information that could compromise the privacy of research participants, are not publicly available due to certain restrictions.

## REFERENCES

[1] MarkNtel Advisors, "Malaysia Tyre Market | Size, Trends and Forecast Report, 2027," 2022. [Online]. Available: https://www.marknteladvisors.com/research-library/malaysia-tyre-market-analysis.html.

[2] S. Sharma, "Digital transformation strategies for small and medium enterprises," *Journal of Business Research*, vol. 112, pp. 45–53, 2020, doi: 10.1016/j.jbusres.2020.02.015.

[3] TM ONE, "Taking Stock of Malaysia's Digital Retail Revolution," 2021. [Online]. Available: https://www.tmone.com.my/think-tank/taking-stock-of-malaysias-digital-retail-revolution/.

[4] J.-F. Lai and S.-H. Heng, "Secure file storage on cloud using hybrid cryptography", *Journal of Informatics and Web Engineering*, vol. 1, no. 2, pp. 1–18, 2022, doi: 10.33093/jiwe.2022.1.2.1.

[5] Okta, "Understanding JSON Web Tokens (JWT)," 2022. [Online]. Available: https://developer.okta.com/docs/guides/validate-jwt-tokens/.

[6] D. d. S. Krug, R. Chanin, and A. Sales, "Exploring the pros and cons of monolithic applications versus microservices", *Proceedings of the 26th International Conference on Enterprise Information Systems*, pp. 256-263, 2024, doi: 10.5220/0012703300003690.

[7] M. A. A. Altemimi, and A. H. H. Alasadi, "Ecommerce based on cloud computing: The art of state," *European Journal of Information Technologies and Computer Science*, vol. 2, no. 4, pp. 7–13, 2022, doi: 10.24018/compute.2022.2.4.59.

[8] Amazon, "AWS," 2024. [Online]. Available: https://aws.amazon.com/.

[9] Q. Xiang, X. Peng, C. He, H. Wang, T. Xie, D. Liu *et al.*, "No free lunch: Microservice practices reconsidered in industry", *arXiv*, 2021, doi: 10.48550/arxiv.2106.07321.

[10] W. Theo, "Survey on IT system adoption in the tyre retail industry in Malaysia," 2025. [Unpublished survey]. Google Forms.

[11] Tyre Inventory Solutions, "Tyre Shop Wizard," 2024. [Online]. Available: https://tyreinventorysolutions.com/.

[12] Autogarage Network, "Tyre Experts (Version 1.0)," 2023. [Online]. Available: https://www.tyreexperts.co.uk/.

[13] Shopee, "Shopee (Version 3.42.24)," 2015. [Online]. Available: https://shopee.com.my.

[14] S. Newman, *Building Microservices: Designing Fine-Grained Systems*, 2nd ed., O'Reilly Media, 2021.

[15] M. A. Fathullah, A. Subbarao, and S. Muthaiyah, "A systematic review: Risk management of cloud computing projects in healthcare," *International Journal of Management, Finance and Accounting*, vol. 4, no. 2, pp. 83–115, 2023, doi: 10.33093/ijomfa.2023.4.2.5.

[16] G. Hohpe, and B. Woolf, *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*, Addison-Wesley, 2003.

[17] LocalStack, "LocalStack Documentation," 2023. [Online]. Available: https://docs.localstack.cloud/.

[18] Amazon Web Services, "Amazon S3," 2024. [Online]. Available: https://aws.amazon.com/s3/.

[19] Amazon Web Services, "Amazon RDS," 2024. [Online]. Available: https://aws.amazon.com/rds/.

[20] Stripe, "Stripe API Documentation," 2024. [Online]. Available: https://stripe.com/docs/api.

## BIOGRAPHIES OF AUTHORS

| | |
|---|---|
|  | **William Theo Wei Loon** is a final-year Information Technology student at Multimedia University, specializing in Security Technology. Fluent in English and Mandarin, he excels in Java, Spring Boot, and web development. His projects include a unified cloud-based system for the tyre retail industry and the "Requirement Navigation and Coordination" project during his Java Backend Internship at Ant International. Passionate about problem-solving and innovation, William seeks opportunities to create impactful tech solutions. He can be contacted at 1211103037.willstudy@gmail.com. |
|  | **Siew-Chin Chong,** an IEEE Senior Member, holds a B.IT, M.Sc., and Ph.D. in Information Technology from Multimedia University. She is currently Deputy Dean of Student Experience & Alumni at the Faculty of Information Science and Technology, MMU. Her research focuses on machine learning, biometric security, and mobile app development, with numerous publications. She has also served on journal editorial boards and as Technical Chair for international conferences. Contact: chong.siew.chin@mmu.edu.my. |
|  | **Kian-Ming Lim,** an Associate Professor at the University of Nottingham Ningbo China and IEEE Senior Member, was recognized among the World's Top 2% Scientists by Stanford University in 2025. With over 100 publications and extensive editorial and review experience in AI and ML, he has served as Deputy Head of the Artificial Intelligence and Optimization Research Group (2024–2025). His research focuses on machine learning, deep learning, computer vision, few-shot learning, federated learning, generative AI, and NLP, and he has successfully supervised over 10 postgraduate research students. He can be contacted at Kian-Ming.Lim@nottingham.edu.cn. |

| | |
|---|---|
|  | **Lee-Ying Chong** received her B.IT (Hons.) in Information System Engineering (2003), M.Sc. in Information Technology (2007), and Ph.D. in Information Technology (2018) from Multimedia University. Her research interests include biometric authentication, computer vision, and machine learning. She has been an IEEE Senior Member since 2013. Currently, she is a senior lecturer at the Faculty of Information Science and Technology, Multimedia University, Malaysia. She can be contacted at lychong@mmu.edu.my. |
|  | **Kuok-Kwee Wee** holds a BSc in Computer Science and an MSc in Networking from Universiti Putra Malaysia, and a PhD in Engineering from Multimedia University. He is an Associate Professor at the Faculty of Information Science and Technology, MMU. He serves as an editorial board member, IEEE Senior Member, and reviewer for international journals and conferences. His research interests include quality of service, wireless access, networking, and mobile communication. Email: wee.kuok.kwee@mmu.edu.my. |