An Intelligent On-Demand Laundry Logistics Platform with Real-Time Location Tracking and Machine Learning-Based Customer Support

Muhammad Faris Rosli[1], Siti Zuraidah Ibrahim[2]*, Mohd Nazri A Karim[3],
Mohd Hafizuddin Mat[4], Faridah Hanim Mohd Noh[5], Tanakorn Inthasuth[6]
Faculty of Intelligent Computing (FKC), Universiti Malaysia Perlis (UniMAP), Perlis, Malaysia[1, 2]
Faculty of Electronic Engineering and Technology (FKTEN), Universiti Malaysia Perlis (UniMAP), Perlis, Malaysia[3, 4]
Department of Electrical Engineering Technology-Faculty of Engineering Technology,
Universiti Tun Hussein Onn Malaysia, Johor, Malaysia[5]
Department of Electrical Engineering-Faculty of Engineering,
Rajamangala University of Technology Srivijaya (RUTS), Songkhla, Thailand[6]

***Abstract*—Traditional on-demand laundry operations suffer from inefficient manual coordination, a lack of real-time delivery visibility, and delayed customer support handling. To address these operational challenges, this study presents LaundrOTrack, an intelligent on-demand laundry logistics platform connecting customers, drivers, and administrators within a unified digital ecosystem. Powered by Flutter for cross-platform access and Firebase Cloud Firestore for real-time data synchronization, the system incorporates Location-Based Services (LBS) via the Google Maps API to streamline last-mile delivery workflows. The platform's real-time location tracking system utilizes continuous GPS updates, route polyline visualization, dynamic Estimated Time of Arrival (ETA) calculation, and spatial arrival boundary detection to monitor pickup and return deliveries. Furthermore, machine learning-based customer support is integrated through an automated conversational agent (Laundra Ask). The underlying natural language processing pipeline employs text preprocessing, Term Frequency-Inverse Document Frequency (TF-IDF) feature extraction, and a Linear Support Vector Classifier (Linear SVC) model hosted on a Flask REST API to predict user query intents and deliver context-aware responses. System testing demonstrates seamless multi-role data synchronization, precise geolocation triggering, and high-accuracy query intent classification. The proposed platform significantly improves order transparency, reduces manual administrative overhead, and optimizes end-to-end service efficiency in smart service logistics.**

Keywords*—On-demand laundry logistics; Location-Based Services (LBS); linear SVC; intent classification; flutter; cloud firestore

I. Introduction

The growth of on-demand digital services has changed the way customers interact with service providers. Applications in transportation, food delivery, e-commerce, and home services have increased customer expectations for convenience, transparency, and real-time service updates. However, many small and medium-sized laundry businesses still depend on manual coordination through phone calls, messaging applications, handwritten records, and walk-in arrangements. This conventional approach can cause unclear order details, inconsistent pickup schedules, delayed status updates, and limited visibility of the laundry process.

In an on-demand laundry service, logistics coordination plays an important role because the service not only involves washing or processing, but it also involves managing pickup and return delivery. Customers often need to know when the driver will arrive, where the driver is currently located, and whether the order has reached the next service stage. Without real-time tracking, route visualization, estimated time of arrival (ETA), and arrival confirmation, customers may repeatedly contact the service provider to ask about delivery progress. At the same time, administrators may face difficulty monitoring active delivery tasks, checking driver movement, and ensuring that pickup and return deliveries are completed properly.

Several previous laundry management systems have introduced digital functions such as customer registration, online booking, service selection, order records, and status updates [1], [2], [3], [4]. Although these systems improve order handling and reduce dependency on manual records, many of them mainly focus on internal laundry operations rather than complete delivery coordination. Existing real-time tracking and geofencing studies have also demonstrated the usefulness of GPS-based monitoring in vehicle tracking, safety monitoring, and location-based alert systems [5], [6], [7]. However, these systems are not always designed for a service-oriented workflow that connects customers, delivery drivers, and administrators in a complete laundry logistics process. Therefore, there is still a need for an integrated platform that combines laundry order management with real-time delivery monitoring and location-based workflow support.

Customer support is another important requirement in laundry service operations. Customers may ask repetitive questions related to service types, price, payment, pickup procedures, delivery status, ETA, and return delivery. If these enquiries are handled manually, administrators may spend additional time responding to routine questions instead of focusing on operational issues. In customer-service environments, Natural Language Processing (NLP) and machine learning have been used to classify customer enquiries

*Corresponding author.

and support faster response handling [8], [9]. Intent classification is particularly useful because it identifies the purpose of a user query and maps the query to a suitable response or service action. Recent studies have also shown that text classification methods such as Term Frequency-Inverse Document Frequency (TF-IDF) with Support Vector Machine (SVM) remain relevant for domain-specific customer-support tasks, especially when the queries are short and the intent categories are clearly defined [10], [11].

Therefore, this study aims to design and evaluate LaundrOTrack, an intelligent on-demand laundry logistics platform with real-time location tracking and machine learning-based customer support. The objectives of this study are to develop a multi-role laundry logistics platform that connects customers, delivery drivers, and administrators within a single platform. Second, to implement real-time GPS-based tracking with route visualization, ETA display, and location-based arrival detection for pickup and return delivery. Third, to integrate a domain-specific customer-support chatbot that classifies laundry service enquiries using TF-IDF and Linear Support Vector Classification (Linear SVC).

The proposed platform consists of two main technical components. The first component focuses on laundry logistics tracking, where driver GPS coordinates are updated during active pickup and return delivery tasks and synchronized with customer and administrator interfaces. The system also displays route polyline, distance, and ETA information, while location-based arrival detection is used to determine whether the driver has reached the active destination. The second component is Laundra Ask, a machine learning-based customer-support chatbot. Laundra Ask processes user questions, transforms them into TF-IDF feature vectors, classifies the intent using Linear SVC, and returns a suitable response through a Flask-based application programming interface.

II. Related Work

A. Digital Laundry Service Systems

The digitalization of laundry services has encouraged the development of mobile and web-based systems for improving booking, service selection, order recording, and communication between customers and laundry providers. Yi Qing Tan and Nizam Omar developed the Smart Laundry System Malaysia, a mobile-based platform that allows customers and laundry owners to manage laundry transactions digitally. The system supports order handling and internal laundry management, but it does not provide a delivery module, real-time driver tracking, or location-based services [1].

Similarly, LaundryMama was introduced to digitize laundry-related tasks and improve interaction between customers and service providers. The system provides online service access and supports the internal management of laundry orders. However, its focus remains on laundry processing and customer transactions rather than the movement of orders during pickup and return delivery [2].

Vinodhini and Bhama proposed LAUNAPP, an Android application that allows customers to place laundry orders, monitor service progress, and receive basic service notifications. The application reduces reliance on manual order registration and face-to-face interaction. Nevertheless, it does not include driver assignment, pickup and delivery coordination, live location monitoring, or route information [3].

Sundar et al. developed the Super-dry Laundry Mobile Application to provide customers with a smartphone-based platform for placing laundry orders, selecting service types and receiving order-status notifications[12]. Similarly, Gupta et al. developed MY DOOR Laundry Application using Android Studio to support customer registration, service selection and digital order submission [13].

Uddin et al. developed IronMan, which combines an Android customer application with a web-based administrative platform. The system includes Google Maps for displaying nearby laundry locations and supporting rider assignment based on proximity [14]. Adekola et al. proposed an Online Laundry Management System that centralizes customer information, service records, and order-status management through a web interface [15]. Chimankar et al. similarly developed an Online Laundry Service that supports service selection, order submission, and order history access [16].

A delivery-oriented system was developed by Suhaimi M et al. for students at Universiti Tun Hussein Onn Malaysia. The web-based system introduced online laundry ordering together with a delivery component to improve accessibility for students who experience time and transportation constraints. Although delivery is included, the delivery process is managed without continuous GPS tracking, route visualization, ETA information, or automatic arrival detection [4].

B. Real-Time Location Tracking

Real-time location tracking has been widely studied in transportation monitoring and safety applications. Rudramurthy et al. developed a Real-Time Vehicle Tracking System for Smart Cities using GPS to provide continuous vehicle-location information. The system demonstrates the usefulness of real-time GPS data for monitoring moving objects and improving transportation awareness [5]. Mishra et al. extended conventional GPS tracking by incorporating geofencing-based event detection. In their system, vehicle coordinates are continuously transmitted to a backend server, and notifications are generated when a vehicle enters or exits a predefined geographical boundary [6].

Tsindeliani et al. investigate latency reduction in real-time GPS tracking across Android and web-based monitoring environments. Their study examined factors including network latency, location update frequency, data transmission, and server processing time. The work highlights that location update performance is important because delayed GPS information may cause the displayed position to differ from the actual movement of the tracked object. It focuses on technical performance [7].

Tanguturi et al. combined Location-Based Services and Google Maps in an Android application for real-time location monitoring and emergency assistance. The application retrieves GPS coordinates and displays user locations through a map-based interface. Although the study demonstrates mobile location acquisition and visualization, its application is directed toward safety and emergency response rather than pickup and return delivery management [17].

GPS measurements may also be affected by signal noise, environmental conditions, and temporary location fluctuations. Özdemir and Tuğrul investigated moving average and Kalman filter techniques to improve the accuracy of real-time GPS tracking [18].

C. Machine Learning-Based Customer Support

Automated customer support commonly applies natural language processing to identify the purpose of incoming customer enquiries. Intent classification converts an unstructured query into a predefined category that can be associated with a response, dialogue flow, or service action. Dong et al. described intent identification as an important first step in customer support because the predicted category can be used to route a request to a suitable specialist or retrieve a prepared response. Their work applied a semi-supervised multitask ALBERT model to e-commerce customer contact data and reported an improvement of approximately 20 points in average AUC-ROC over the baseline multiclass model [8].

Bonechi et al. studied automatic ticket classification in banking customer support using a real-world dataset containing 4243 chat-based requests distributed across ten classes. The authors compared BERT with a conventional pipeline that combined Term Frequency-Inverse Document Frequency (TF-IDF) and Support Vector Machine classification. Both approaches demonstrated potential for improving customer support efficiency [10]. This is particularly relevant because customer enquiries are frequently short, category-specific, and influenced by the presence of discriminative terms.

Rizou et al. developed a multilingual conversational agent for university administrative services. Their model combines Bidirectional Long Short-Term Memory and Conditional Random Fields to perform intent extraction and named entity recognition using the English and Greek UniWay dataset. The study reported that closed service domains can benefit from architectures that require fewer computational and memory resources while maintaining competitive performance [11].

Kulkarni et al. addressed conversational intent classification in an e-commerce chatbot using a semi-supervised approach. The study considered pre-purchase and post-purchase questions, including enquiries about product details, delivery time, exchanges and returns. It also highlighted practical challenges such as grammatical errors, code-mixed language and limited labelled data. Their proposed approach used a small, labelled corpus together with a larger collection of unlabeled queries and improved upon the supervised baseline [9].

Other researchers have explored different approaches for domain-specific assistance. Wang et al. developed a telecom customer-service assistant that combines information retrieval, word embeddings and multiclass intent prediction. The system retrieves and ranks candidate answers, then presents the five most relevant responses to customer-service staff [19].

Pawlik et al. further demonstrated that recognizing implicit intent and considering emotional information can improve actions performed by a customer-service bot [20]. In another application domain, Wang et al. developed a chatbot that identifies user intentions and provides context-aware COVID-19 information using NLP, machine learning and a knowledge graph [19].

III. METHODOLOGY

A. System Architecture

LaundrOTrack was developed as a multi-role platform comprising a customer mobile application, a driver mobile application and an administrator web dashboard. The customer application supports laundry ordering, order progress monitoring, real-time driver tracking, and access to the Laundra Ask customer-support assistant. The driver application manages pickup and return delivery tasks, status updates, route navigation and GPS location sharing. The administrator dashboard supports order monitoring, driver assignment and operational tracking.

The mobile applications were developed using Flutter and Dart, while Firebase was used as the central backend. Firebase Authentication manages user access and Cloud Firestore stores and synchronizes order information, driver locations and status updates between the three user modules. Google Maps services provide map visualization, route information, distance and estimated travel duration. In addition, the customer application communicates with a Flask REST API that hosts the Laundra Ask intent classification model. The overall architecture is illustrated in Fig. 1.

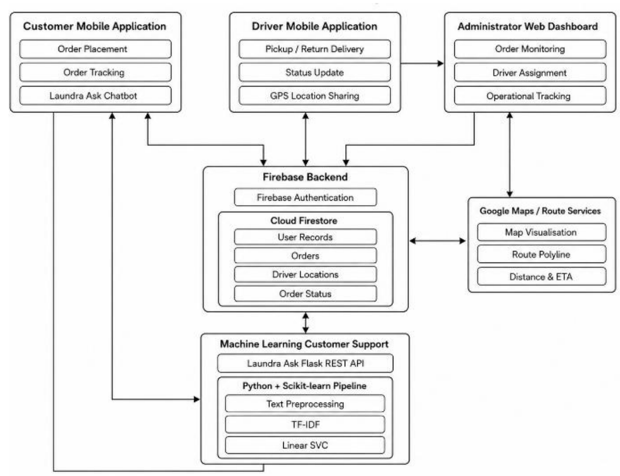


Fig. 1. Overall architecture of the proposed platform.

B. Laundry Logistics and Real-Time Tracking Workflow

The logistics workflow begins when a customer submits a laundry order containing the selected service and pickup and return addresses. The administrator reviews the order and assigns the corresponding task to a driver. Once the driver accepts the task and starts the journey, the order enters an active tracking stage.

During pickup, the customer pickup location becomes the active destination. The driver application obtains the current GPS coordinates from the device location service and updates the location data in Cloud Firestore. The customer application and administrator dashboard retrieve the latest coordinates and display the driver position on their respective tracking interface.

After the driver reaches the pickup location and confirms the collection, the order progresses through the laundry processing stages. Once processing is completed, the return delivery workflow is activated, and the customer return address becomes the new destination. The same tracking process continues until the driver confirms delivery completion. Tracking is activated only during movement-related stages and is stopped after the corresponding pickup or delivery task has been completed.

C. Route, ETA and Location-Based Arrival Detection

The route generation process uses the driver's latest coordinates as the origin and selects the destination according to the current order stage. During pickup, the destination is the customer pickup address, whereas during return delivery, the destination is the customer return address. The origin and destination coordinates are submitted to the route service, which returns encoded polyline data, travel distance and estimated duration.

The polyline is decoded and displayed on Google Maps together with markers representing the driver and active destination. The estimated arrival time is calculated as

$$T_{ETA} = T_{current} + T_{duration} \tag{1}$$

where, $T_{current}$ is the current system time and $T_{duration}$ is the travel duration returned by the route service. The route, distance and ETA can be updated when the driver's position or active destination changes.

Arrival detection is implemented using a distance-based condition rather than geofencing. The system compares the distance between the driver's current coordinate and the active destination against a predefined arrival threshold:

$$A = \begin{cases} 1, & d(P_d, P_t) \le r_a \\ 0, & d(P_d, P_t) > r_a \end{cases} \tag{2}$$

where, P_d represent the driver position, P_t represent the destination, r_a is the configured arrival threshold and A is the arrival state. When the driver remains outside the threshold, the arrival confirmation action remains restricted. Once the driver is sufficiently close to the destination, the application enables the next action, such as confirming pickup arrival and return delivery.

D. Laundra Ask Intent-Classification Pipeline

Laundra Ask was implemented as a closed-domain customer-support assistant for common laundry service enquiries. The prepared dataset contains example questions, intent labels and associated responses. The implemented categories include greeting, general help, service price, payment, order status, ETA request, pickup location, delivery issue, cancel order, contact support and unknown.

Before model training, the question text was cleaned and standardized to reduce variation in capitalization, punctuation and sentence structure. The processed text was then transformed into numerical features using TF-IDF. TF-IDF assigns greater importance to terms that are frequent within a particular question but less common across the complete dataset. Words such as price, payment, pickup, delivery, ETA and cancel therefore provide useful information for distinguishing between intent categories (Fig. 2).

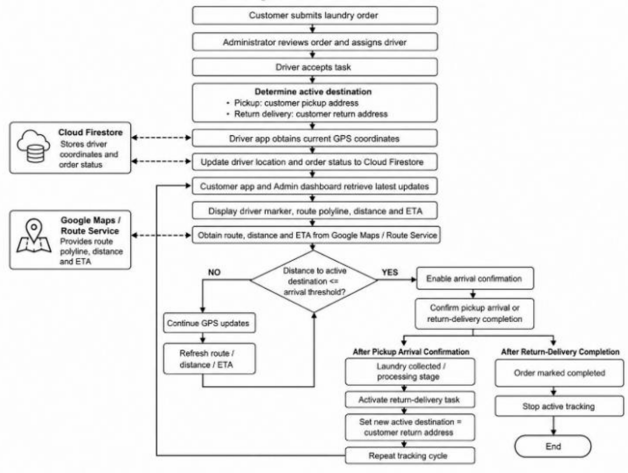


Fig. 2. Real-time tracking, route and arrival detection workflow.

Linear Support Vector Classification was used to classify the resulting TF-IDF feature vectors. Linear SVC was selected because it is suitable for sparse, high-dimensional text data and provides an efficient classification approach for a limited closed domain dataset. The TF-IDF vectorizer and Linear SVC classifier were combined within a Schikit-learn pipeline so that identical text transformation was applied during training and prediction.

The dataset was divided into training and testing subsets. The training subset was used to learn the relationship between customer question patterns and intent labels, while the testing subset was retained for evaluation using accuracy, precision, recall, and F1-Score.

E. Flask API Integration

After training, the complete TF-IDF and Linear SVC pipeline was stored as a reusable .pk1 model file. A Flask REST API was then implemented to connect the Python model with the Flutter customer application.

When a customer submits a question, the application sends the text to the API through an HTTP request. The API loads the saved pipeline, transforms the question into TF-IDF features, predicts the most suitable intent and retrieves the corresponding prepared response. The response is then returned to the Flutter application and displayed through the Laundra Ask interface.

IV. Results and Discussion

The proposed platform was evaluated through multi-role user interface workflow validation, functional tracking and logistics testing, and machine learning intent classification analysis. The platform interface evaluation examined the operational workflow and usability across customer, driver, and administrator client modules. The logistics tracking evaluation examined GPS coordinate acquisition, Cloud Firestore synchronization, live tracking across the customer and administrator interfaces, route and ETA generation, active destination selection, location-based arrival detection, and tracking termination. Furthermore, Laundra Ask was evaluated using unseen testing data based on the standard classification metrics of accuracy, precision, recall, and F1-score.

A. Platform Implementation and User Interface Design

LaundrOTrack was realized across three interconnected client modules. The customer mobile application provides a streamlined end-to-end service workflow.

As shown in Fig. 3(a), the home dashboard serves as the central hub displaying user greetings, active order cards, and service shortcuts. When placing an order, users specify collection coordinates via the map-pinning pickup location interface in Fig. 3(b). The order summary interface in Fig. 3(c) consolidates selected packages, add-ons, and itemized fare breakdowns before payment. Post-checkout, the order history and progress timeline in Fig. 3(d) provide complete service visibility through historical records and a multi-stage milestone tracker.

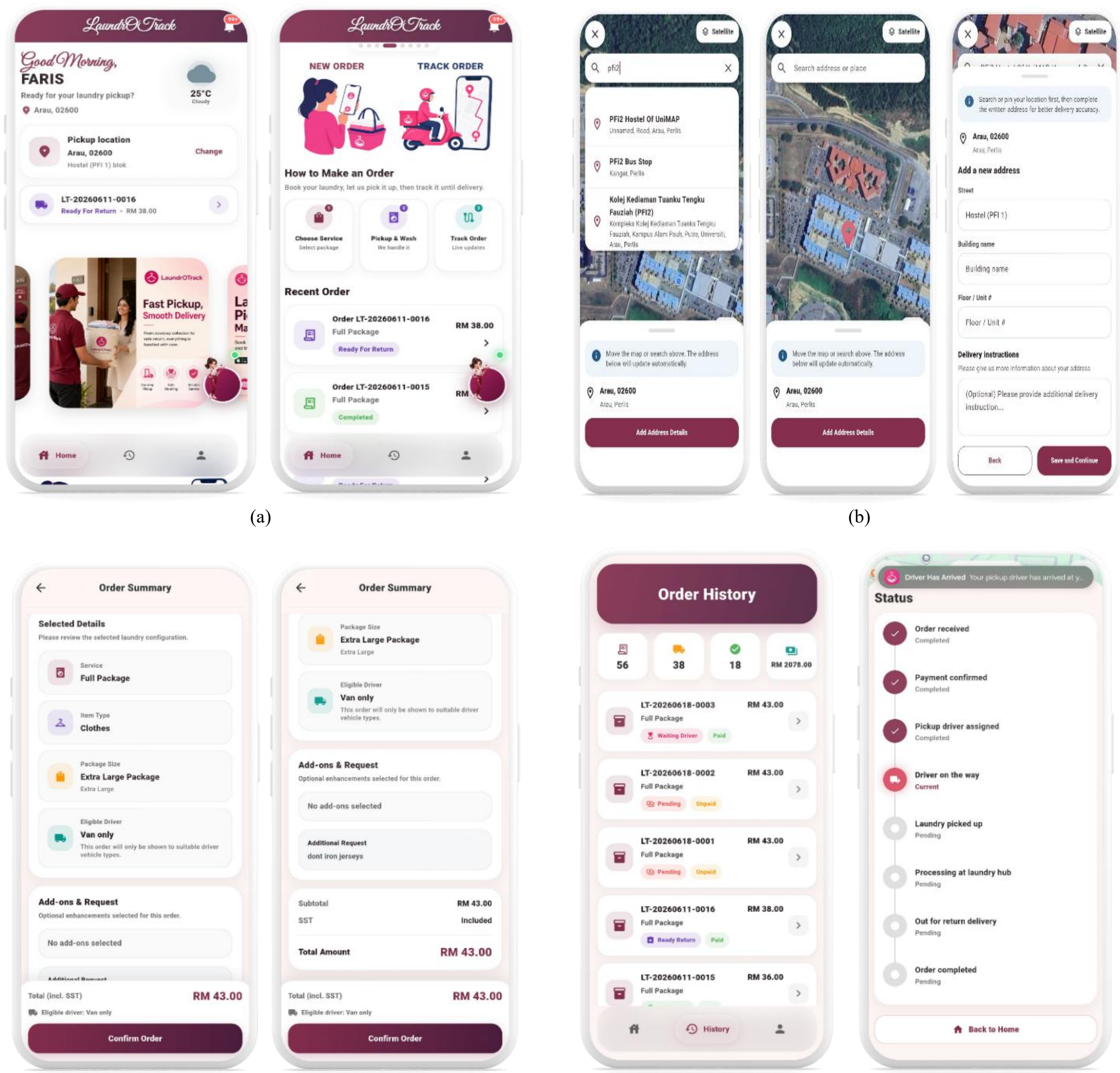
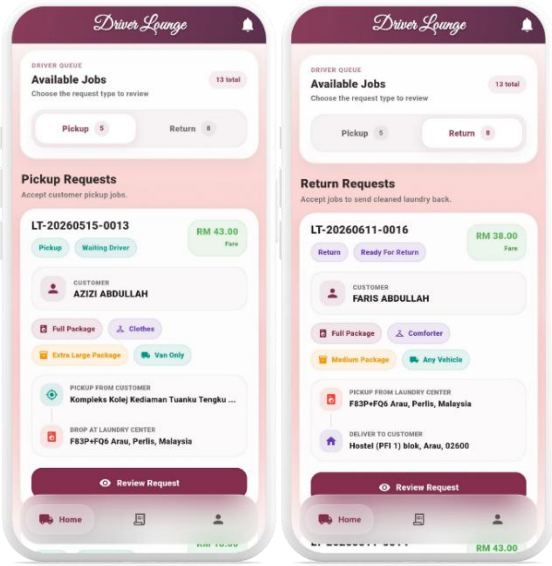
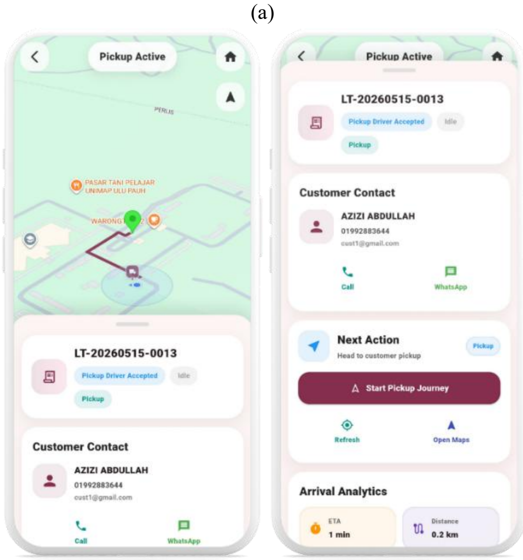


Fig. 3. Customer mobile application interfaces displaying: (a) customer home dashboard interface, (b) pickup location setup interface, (c) order summary interface, and (d) order history and progress timeline.

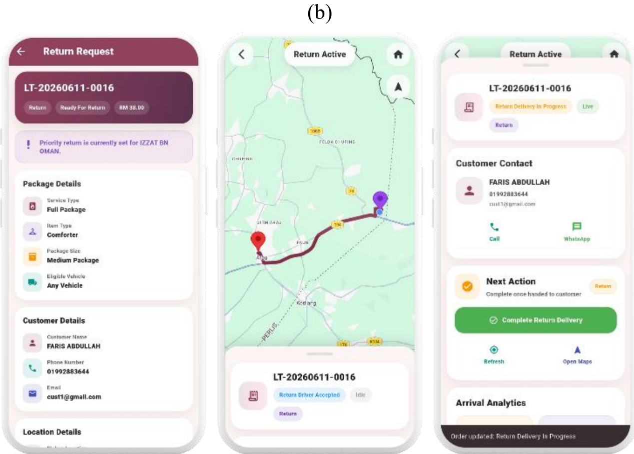
The driver mobile application focuses on an intuitive layout optimized for task execution across both delivery phases. As depicted in Fig. 4(a), the pickup and return delivery request interface organizes incoming jobs into structured queues, enabling drivers to inspect order specifications, fare amounts, and customer locations prior to task acceptance. Upon task acceptance, the driver active pickup order screen shown in Fig. 4(b) provides an operational tracking layout with integrated map routes, direct customer communication shortcuts, and arrival confirmation controls. Following laundry processing, the return delivery handling process interface in Fig. 4(c) transitions the driver into the final delivery stage, providing dedicated controls to manage and confirm the return of processed items to the customer.



(a)

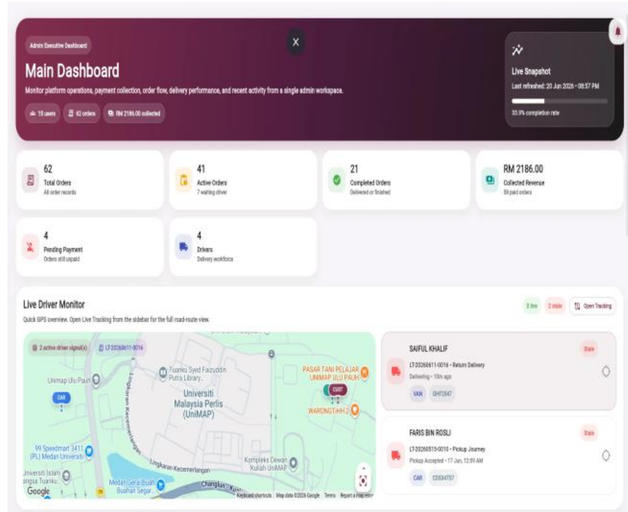


(b)

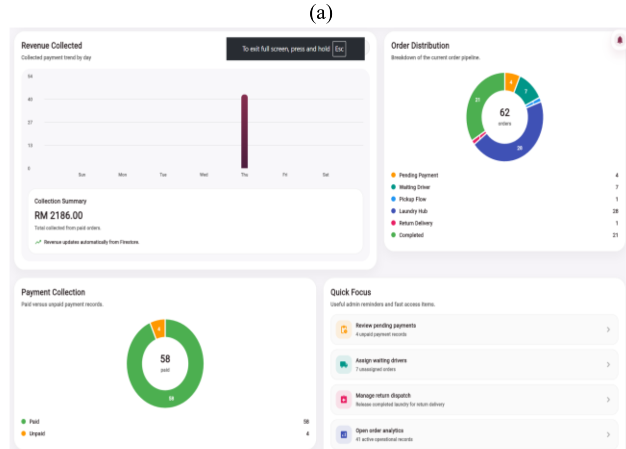


(c)

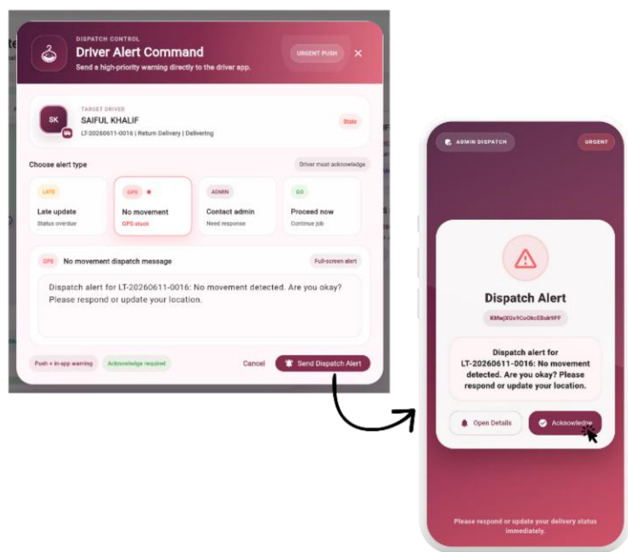
Fig. 4. Driver mobile application interfaces displaying: (a) pickup and return delivery request interface, (b) driver active pickup order screen, and (c) return delivery handling process interface



(a)



(b)



(c)

Fig. 5. Administrator web management console displaying: (a) administrator main dashboard interface, (b) analytical section of the administrator main dashboard, and (c) driver alert command and dispatch notification interface.

The administrator web management console provides centralized control over the system's operational and dispatch operations. As shown in Fig. 5(a), the administrator main dashboard interface presents a high-level operational overview through active order metrics, financial performance cards, recent transactions, and live driver monitoring panels. Complementing the operational layout, the analytical section of the administrator main dashboard in Fig. 5(b) converts raw platform data into visual charts detailing revenue collections, order status distributions, and payment status insights for strategic decision-making. Furthermore, the driver alert command and dispatch notification interface depicted in Fig. 5(c) provides direct operational intervention capabilities, allowing administrators to send task alerts, dispatch updates, and status reminders directly to assigned drivers during live order management.

B. Real-Time Tracking and Logistics Results

The functional evaluation verified nine tracking-related scenarios, including location permission handling, GPS coordinate acquisition, Firestore location updates, customer and administrator tracking displays, status-based tracking activation, tracking termination, route and ETA output and location-based arrival detection, as shown in Table I. The actual results showed that all evaluated functions produced the expected behavior.

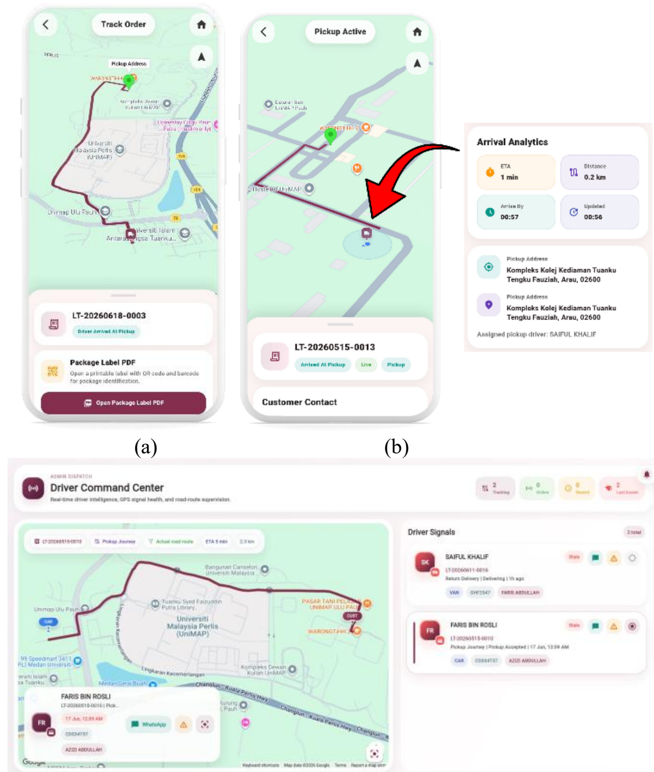
Fig. 6 presents the tracking output from the customer, driver and administrator perspectives. Fig. 6(a) shows the customer tracking interface, where the route between the driver and the active destination is displayed together with the current order information. The route polyline provides a clearer indication of the driver's travel direction than a location marker alone.

TABLE I. FUNCTIONAL EVALUATION OF THE TRACKING COMPONENT

Evaluated Function	Expected Result	Outcome
GPS coordinate acquisition	Driver application obtains the current location	Passed
Firestore location update	Latest driver coordinates are stored and updated	Passed
Customer live tracking	Customer can view the driver location	Passed
Administrator monitoring	Administrator can monitor driver movement	Passed
Tracking activation	Tracking begins during an active pickup or return delivery task	Passed
Tracking termination	Tracking stops after the movement stage is completed	Passed
Route and ETA	Route, distance and ETA are displayed	Passed
Arrival detection	Arrival confirmation is enabled near the active destination	Passed

Fig. 6(b) shows the driver tracking interface and arrival analytics during an active pickup task. The interface displays the current driver position, destination, remaining distance, ETA, expected arrival time and last updated information. This output allows the driver to monitor the current journey while providing measurable progress information for the active task.

Fig. 6(c) presents the administrator live tracking dashboard. The dashboard provides a centralized map view together with active driver information, allowing the administrator to monitor driver movement and identify tasks that may require attention. The customer and administrator outputs demonstrate that the driver's location can be viewed across different system roles according to their operational requirements.



(a) (b)

(c)

Fig. 6. Real-time logistics tracking output: (a) customer tracking interface; (b) driver route and arrival analytics; (c) administrator live tracking dashboard.

Location-based arrival detection was also verified during pickup and return delivery testing. The system compared the driver's current position with the active destination. When the driver remained outside the configured arrival boundary, tracking continued and the arrival-confirmation action remained unavailable. Once the required proximity was reached, the application enabled the next relevant confirmation action.

C. Laundra Ask Training and Classification Results

The Laundra Ask dataset contained 1100 labelled questions distributed across eleven intent categories: cancel_order, contact_support, delivery_issue, eta_request, general_help, greeting, order_status, payment, pickup_location, service_price and unknown. The dataset included English, Malay and informal mixed-language questions to represent different customer writing styles.

The dataset was divided using a stratified 80:20 training-testing split with a fixed random state of 42. Consequently, 880 samples were used for training, and 220 unseen samples were retained for testing. Each intent category contributed 20 testing samples. The Scikit-learn pipeline applied lowercase TF-IDF feature extraction with unigram and bigram features, followed by Linear Support Vector Classification.

After training, the classifier achieved an overall accuracy of 99.55%, corresponding to 219 correct predictions from 220 testing samples. This result indicates that the TF-IDF and Linear SVC pipeline successfully learned the discriminative language patterns associated with the prepared laundry service intent categories.

TABLE II. LAUNDRA ASK INTENT CLASSIFICATION RESULTS

Intent	Precision	Recall	F1-score	Support
Cancel order	1.00	1.00	1.00	20
Contact support	1.00	0.95	0.97	20
Delivery issue	1.00	1.00	1.00	20
ETA request	1.00	1.00	1.00	20
General help	0.95	1.00	0.98	20
Greeting	1.00	1.00	1.00	20
Order status	1.00	1.00	1.00	20
Payment	1.00	1.00	1.00	20
Pickup location	1.00	1.00	1.00	20
Service price	1.00	1.00	1.00	20
Uknown	1.00	1.00	1.00	20

Table II shows that nine of the eleven intent categories obtained precision, recall and F1-score values of 1.00; the contact_support category recorded a recall of 0.95 and an F1-score of 0.97, while general_help recorded a precision of 0.95 and an F1-score of 0.98. The macro and weighted averages were displayed as 1.00 in the original classification report because the values were rounded to two decimal places.

The confusion matrix in Fig. 7 provides a more detailed representation of the classification result. All intent categories recorded 20 correct predictions along the main diagonal except contact_support, which recorded 19 correct predictions. One contact_support question was classified as general_help. This overlap is understandable because broad requests for assistance may use wording similar to enquiries asking how to contact the customer support service.

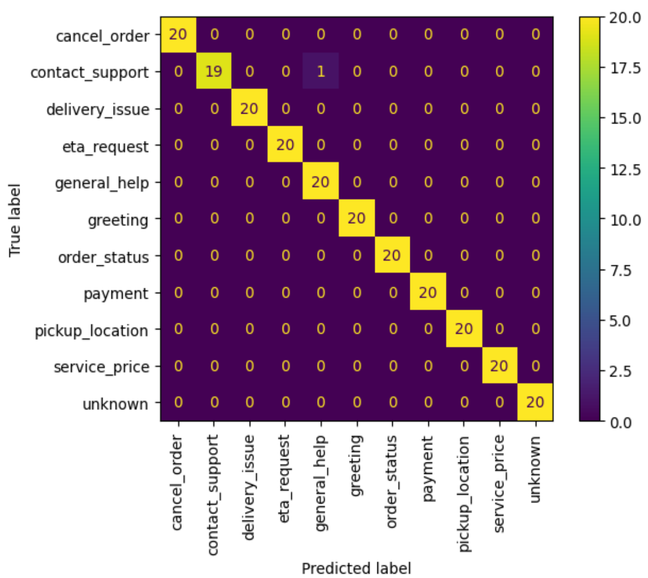


Fig. 7. Confusion matrix of the laundra ask intent classifier.

Additional prediction tests were conducted using informal and mixed-language customer questions. These enquiries involved order location, driver movement, refund requests, delayed delivery, customer-support contact and pick-up-address changes. The model predicted relevant categories such as order_status, eta_request, delivery_issue, contact_support and pickup_location.

D. Laundra Ask API and Application Integration Results

After evaluation, the complete TF-IDF and Linear SVC pipeline was saved as a .pk1 file. Since the vectorizer and classifier were stored within the same Scikit-learn pipeline, the deployed backend could perform text transformation and intent prediction through a single saved model.

The Flask API was successfully deployed and made accessible through an online endpoint. When a customer submitted a question from the Flutter application, the API received the input, loaded the saved pipeline, predicted the intent and returned the corresponding response. The operational output confirmed that the deployed server was running and accessible by the customer application.

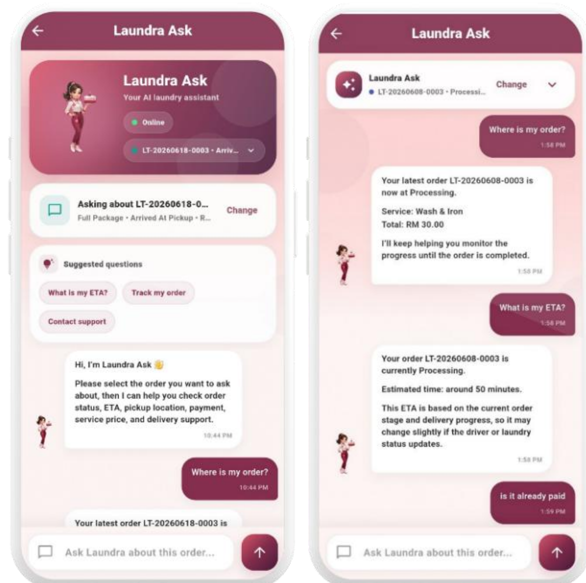


Fig. 8. Laundra Ask chatbot interface.

Fig. 8 shows the final Laundra Ask interface integrated into the customer application. The output demonstrates that customers can submit questions related to order status, ETA, payment and other laundry service enquiries and receive responses through a chat-based interface. This confirms that the trained classifier was successfully connected to the operational application rather than being evaluated only as a standalone machine learning model. While internal cross-validation confirmed model stability on the curated dataset, we acknowledge that this limits out-of-vocabulary generalizability. Real-world queries containing slang or typos may degrade performance, which will be addressed in future online retraining.

V. Limitation and Future Work

The tracking component depends on GPS accuracy and stable internet connectivity. Location information may become less accurate indoors, in weak signal areas, or near tall buildings. The current tracking evaluation verified system integration and state transitions under synthetic conditions. Physical GPS drift, network latency, and dynamic ETA accuracy were not quantitatively measured and are reserved for live field deployment. Network instability may delay Firestore synchronization, route updates and arrival detection. The tracking evaluation was also based on functional scenarios rather than controlled measurements of positioning error and end-to-end latency.

Laundra Ask is limited by the size and coverage of the prepared dataset. Although English, Malay and mixed-language questions were included, substantially different expressions may produce incorrect predictions. This work focuses on alpha-stage technical feasibility. Formal end-user studies assessing usability, learnability (e.g., via the System Usability Scale), and operational efficiency will be conducted in future field trials. Future work should include real customer data collection, broader multilingual testing, confidence-based fallback handling, GPS accuracy and latency measurement and evaluation in larger operational environments.

VI. Conclusion

This study presented LaundrOTrack, an intelligent on-demand laundry logistics platform that integrates real-time location tracking and machine learning-based customer support. The platform connects customers, drivers and administrators through a Flutter application, a web dashboard and Firebase backend services. Its logistics component provides GPS-based driver monitoring, route visualization, distance and ETA information and location-based arrival detection for pickup and return delivery. Laundra Ask applies TF-IDF and Linear SVC to classify eleven categories of laundry service enquiries and communicate with the customer application through a Flask REST API. Functional testing confirmed the expected operation of the tracking workflow, while the intent classifier achieved 99.55% accuracy on 220 testing samples. These results demonstrate the feasibility of combining delivery visibility and automated first-level support within a unified laundry-service environment. Further field evaluation with real users, broader language variations and controlled GPS-performance measurements is required before commercial deployment.

Acknowledgment

The authors would like to express their sincere gratitude to Universiti Malaysia Perlis (UniMAP) for providing the necessary facilities and funding to support this work. The authors also extend their appreciation to Dina Suci Teguh Enterprise, known as SM Dobi, a laundry service provider located in Sungai Petani, Kedah, Malaysia, for serving as an industry collaboration partner by contributing valuable domain inputs and practical requirements during system development.

References

[1] T. Yi Qing and M. Nizam Omar, "Smart Laundry System" *Multidisciplinary Applied Research and Innovation*, vol. 4, no. 1, pp. 216–223, 2023, doi: 10.30880/mari.2023.04.01.040.

[2] L. Y. Mei, K. N. F. Ku Azir, S. Z. Ibrahim, and S. N. Azemi, "LaundryMama: Humanising Laundry Tasks using Laundry Management System and Laundry-On-Demand Mobile Applications," in *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, Mar. 2020. doi: 10.1088/1757-899X/767/1/012061.

[3] Vinodhini and Bhama, "LAUNAPP – An Android application for Laundry Services," European Alliance for Innovation, Jan. 2022. doi: 10.4108/eai.7-12-2021.2314592.

[4] M. Suhaimi *et al.*, "Delivery Laundry System: The Development of Laundry Web-Based System for UTHM Student to Make Laundry More Convenient," *Multidisciplinary Applied Research and Innovation*, vol. 4, no. 3, pp. 124–129, 2023, doi: 10.30880/mari.2023.04.03.017.

[5] M. S. Rudramurthy *et al.*, "Real Time Vehicle Tracking System for Smart Cities," in *2023 International Conference on Data Science and Network Security, ICDSNS 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ICDSNS58469.2023.10245776.

[6] R. Mishra, A. K. Tiwari, A. Singh, K. Gupta, K. Srivastava, and A. Srivastava, "Real-time Vehicle Tracking System Using Geofencing," in *Lecture Notes in Networks and Systems*, Springer Science and Business Media Deutschland GmbH, 2024, pp. 269–277. doi: 10.1007/978-981-97-6318-4_20.

[7] D. Tsindeliani, Y. Povstyana, N. Lishchyna, and A. Yashchuk, "Latency Reduction in Real-time GPS tracking in Android and the Web-based GPS Monitoring System," in *Proceedings of the 2022 IEEE 12th International Conference on Dependable Systems, Services and Technologies, DESSERT 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/DESSERT58054.2022.10018609.

[8] L. Dong, M. C. Spencer, and A. B. Amazon, "A Semi-supervised Multi-task Learning Approach to Classify Customer Contact Intents," in *Proceedings of the 4th Workshop on e-Commerce and NLP, 2021 Aug,* pp. 49-57.

[9] M. Kulkarni, K. Kim, N. Garera, and A. Trivedi, "Label efficient semi-supervised conversational intent classification." in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, 2023, pp. 96-102.

[10] S. Bonechi, G. Palma, M. Caronna, M. Ugolini, A. Massaro, and A. Rizzo, "Enhancing Customer Support in Banking: Leveraging AI for Efficient Ticket Classification," in *Procedia Computer Science*, Elsevier B.V., 2024, pp. 128–137. doi: 10.1016/j.procs.2024.09.235.

[11] S. Rizou *et al.*, "Efficient intent classification and entity recognition for university administrative services employing deep learning models," *Intelligent Systems with Applications*, vol. 19, Sep. 2023, doi: 10.1016/j.iswa.2023.200247.

[12] S. Sundar, S. Keerthivasan, S. Sachin Ramsangu, L. E. Vikram, and R. L. Deepthi, "Implementation of Mobile App for Laundry 'Super-Dry,'" in *Proceedings of the 2nd International Conference on Electronics and Sustainable Communication Systems, ICESC 2021*, Institute of Electrical and Electronics Engineers Inc., Aug. 2021, pp. 1837–1842. doi: 10.1109/ICESC51422.2021.9532904.

[13] A. Gupta, D. Kumar Panda, and M. Pande, "Development of Mobile Application for Laundry Services Using Android Studio," 2018. [Online]. Available: http://www.ripublication.com

[14] M. M. Uddin, R. Roy, S. A. Miduri, and R. M. Rahman, "IronMan: An Android-Web Based Application for Laundry Services," in *2022 IEEE International IOT, Electronics and Mechatronics Conference, IEMTRONICS 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/IEMTRONICS55184.2022.9795823.

[15] O. D. Adekola *et al.*, "Online Laundry Management System," *International Journal of Computer*, 2021, [Online]. Available: http://ijcjournal.org/

[16] D. Chimankar, G. Khairkar, H. Maheshram, K. Gayakwad, and M. Shahzad, "ONLINE LAUNDRY SERVICE," *www.irjmets.com @International Research Journal of Modernization in Engineering*, vol. 6144, Apr. 2023, doi: 10.56726/IRJMETS37554.

[17] R. C. Tanguturi, V. Dankan Gowda, S. K. H. Kauser, N. Vyshnavi, K. Hephzibah, and T. Aswani, "Design and Implementation of an Android App for Real-Time Location Monitoring and Emergency Assistance," in *7th International Conference on Electronics, Communication and Aerospace Technology, ICECA 2023 - Proceedings*, Institute of Electrical

and Electronics Engineers Inc., 2023, pp. 1561–1567. doi: 10.1109/ICECA58529.2023.10394685.

[18] Zeynep ÖZDEMİR and Bülent TUĞRUL, "Geofencing on the Real-Time GPS Tracking System and Improving GPS Accuracy with Moving Average, Kalman Filter and Logistic Regression Analysis", *2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*. IEEE, 2019, pp. 1-6.

[19] X. Wang *et al.*, "Artificial Intelligence-Empowered Chatbot for Effective COVID-19 Information Delivery to Older Adults," *International Journal of E-Health and Medical Communications*, vol. 12, no. 6, 2021, doi: 10.4018/IJEHMC.293285.

[20] Ł. Pawlik, M. Płaza, S. Deniziak, and E. Boksa, "A method for improving bot effectiveness by recognising implicit customer intent in contact centre conversations," *Speech Commun.*, vol. 143, pp. 33–45, Sep. 2022, doi: 10.1016/j.specom.2022.07.003.