Digital Transformation of Monitoring in The Customs System

Development of an App for Tracking Closed Vehicles at Land Border Crossings

Ernelda NOVAKU[1]
European University of Tirana

Aurora BINJAKU[2]
European University of Tirana

Abstract

The customs system in Albania faces significant challenges in monitoring sealed vehicles at land border points, where manual processes lead to delays, human errors, and a lack of transparency. These issues increase operational costs, reduce service efficiency, and hinder effective risk management due to the absence of advanced analytical tools. The research question addressed is: How can an application leveraging advanced technologies, such as Machine Learning (ML) and a Graphical User Interface (GUI), enhance the monitoring of sealed vehicles at Albanian land border points? This study tackles these gaps by developing a dedicated application designed to boost transparency, minimize operational errors, and provide risk analysis based on historical data. Drawing from international and local literature on

[1] Ernelda Novaku is currently working at the General Directorate of Customs of Albania within the Directorate of Risk Analysis and Monitoring. She graduated in "Information Technology" at the European University of Tirana and in "Business Administration" at the Faculty of Economy, University of Tirana. She is also a Lecturer at the Institute for Strategic Studies and Training (ISSAT).

[2] European University of Tirana, Faculty of Engineering, Informatics and Architecture, Department of Engineering, Tirana, Albania, aurora.binjaku@uet.edu.al

successful customs digitization, the research identifies ML and GUI as key tools for efficiency and security. The developed application includes a user-friendly interface for customs authorities, featuring functionalities like vehicle registration, statistical reporting, and risk prediction, supported by technologies such as PHP, MySQL, HTML, CSS, Bootstrap 5, Font Awesome, JavaScript with Chart.js, Python with scikit-learn, and Flask. The methodology combines a qualitative approach, based on professional observations at customs branches, with technical development. The application successfully automates monitoring processes, achieving a risk prediction accuracy of 92.3% through ML, while the GUI simplifies data entry and reporting for customs officers, enhancing operational workflow. This digital transformation offers a scalable solution to improve customs efficiency, reduce manual errors, and align with the Albanian Customs Development Strategy 2021-2025. It sets a foundation for further technological integration, potentially impacting regional customs modernization efforts.

Key words: *Customs Monitoring, Risk Prediction, Machine Learning, Random Forest Classifier, Digital Transformation, Web Application Development*

Introduction

The Albanian customs system faces significant challenges in monitoring sealed vehicles at land border points, as manual processes lead to delays, human errors, and a lack of transparency. These issues result in reduced efficiency of customs services and increased risk due to the inability to timely identify all vehicles during daily flows. The absence of an automated system for real-time tracking of sealed vehicles limits the ability to manage the flow of goods and ensure process security. This situation highlights the need for a digital solution that can improve transparency, accuracy, and speed of monitoring at border points, addressing the limitations of the current infrastructure and the technological capabilities of customs staff.

The main purpose of this work is the development of an application for real-time tracking of sealed vehicles at land border crossings in Albania, aimed at improving the efficiency, transparency, and security of customs processes through advanced analyses and a user-friendly interface.

The specific objectives include: (1) identifying user needs (customs authorities and other structures), (2) designing and testing an application prototype that uses Machine Learning (ML) for risk prediction and Graphical User Interface (GUI) for reporting, and (3) evaluating its impact on customs operations. The work aims to provide an integrated solution, relying on technologies such as PHP, MySQL,

Python, HTML, CSS, Bootstrap 5, Font Awesome, JavaScript with Chart.js, and Flask, with plans for further expansion based on the need for real-time tracking.

Methodology

The study employs a combined qualitative approach, technical development, and theoretical analysis based on professional experience as an employee in the Monitoring and Risk Analysis sector at the General Directorate of Albanian Customs. Data were collected through structured observations and document analysis, documenting the issues of manual processes, particularly at border crossings with high traffic volumes. The application development process follows a hybrid methodology, combining elements of the Software Development Life Cycle (SDLC) and utilizing technologies that include PHP for server-side logic, MySQL for the database, HTML for the user interface (GUI), Python with scikit-learn for training a Machine Learning model (Random Forest Classifier) for risk prediction, and Flask for ML API integration. The prototype is developed as a Minimum Viable Product (MVP), tested in local environments to evaluate the functionality of vehicle registration and risk analysis. For building an intelligent application for the analysis and prediction of risk levels for vehicles passing through border crossings, the Random Forest Classifier algorithm has been selected.

In the customs application, CSS is used to style the user interface (index. php, ndihma.php), defining colors, fonts, element positioning, and responsive design. CSS is integrated with Bootstrap 5, a well-known framework for creating responsive interfaces, and Font Awesome for adding visual icons.

Testing is supported by qualitative analyses of observations and collected data, with plans to expand with other technologies in the future for real-time tracking.

Background of the Albanian Customs Monitoring System

The customs system in Albania is a central pillar of public administration and the national economy, regulating the import, export, and transit of goods across state borders. It plays a vital role in revenue collection and in safeguarding Albania's economic, health, and security interests. The Albanian Customs Administration, responsible for implementing customs policies, is organized into several units handling customs clearance, goods control, tax collection, and international cooperation.

Over the years, the system has undertaken several steps toward digitalization. One of the most notable innovations is the implementation of online customs procedures, which reduce the reliance on physical documentation and minimize

direct contact between customs officers and agents—thus decreasing administrative subjectivity and potential misuse.

Goods Control and Risk Management Framework

According to the *Customs Code of the Republic of Albania* (2014), customs controls rely on a structured risk-management approach. Authorities are empowered to verify goods, inspect documents, take samples, examine transport means, inspect baggage, and conduct investigations whenever deemed necessary. Controls are guided by national and international criteria and may include:

- verification of documentation authenticity and accuracy,
- examination of operators' accounts and records,
- physical inspection of vehicles and transported goods.

Risk management involves continuous collection, analysis, assessment, and monitoring of information. Customs administrations also exchange risk-related data with homologous agencies, enabling the establishment of common criteria and priority control areas. These areas cover specific customs regimes, categories of goods, transport modes, and routes identified as higher risk.

Travelers entering Albania benefit from duty exemptions for personal-use goods up to predefined thresholds (e.g., €300 for land routes and €430 for air or sea routes). Currently, vehicle selection for additional checks is conducted manually, which can lead to inconsistencies.

Monitoring Processes and Current Challenges

The customs monitoring system forms an essential component of administrative modernization and border management. It incorporates advanced systems for supervising the movement of goods and financial flows (Green Book, 2024). However, despite the adoption of several digital tools, the system faces operational and technological challenges.

Digital Monitoring Systems
The Albanian Customs Administration utilizes:

- **Asycuda World** for electronic declarations, document processing, and payment monitoring;
- **Oracle Customs Procedures System** for generating integrated reports from Asycuda data;

- **Camera-based monitoring systems (Milestone)** at border crossing points;
- **Joint monitoring initiatives with Kosovo**, including the use of cameras for cross-border surveillance.

These tools, although valuable, operate independently and lack a unified, real-time analytical framework.

Key Systemic Challenges
Several structural issues persist, including:

- **Cybersecurity vulnerabilities**, with Albania increasingly targeted by foreign cyberattacks.
- **Limited regional harmonization**, especially in trade operations with neighboring countries such as Kosovo.
- **Smuggling and tax evasion risks**, particularly in high-value vehicle categories.
- **Insufficient training and human capacities**, affecting the efficiency of customs procedures.

These challenges highlight the need for more advanced, secure, and integrated monitoring solutions.

Analysis of Current Manual Procedures and Their Consequences

Many core customs processes remain manual, including visual inspection of sealed vehicles via cameras, decisions regarding weighing or scanning, simplified declaration processing, and physical verification of goods. These manual procedures result in several drawbacks:

- **Time delays**, causing queues and slow vehicle processing;
- **Increased operational costs**, due to high dependence on human resources;
- **Higher business costs**, resulting from delays and extended storage times;
- **Subjective assessments**, leading to inconsistent or inaccurate decision-making;
- **Lack of standardization**, which undermines transparency and predictability.

These inefficiencies reduce the ability of customs authorities to manage risk effectively and hamper the flow of goods across borders.

Technologies Used in Developing the Proposed Application

In response to the identified challenges, the proposed application integrates modern web development tools with artificial intelligence to support automated risk prediction and streamlined customs operations.

Authentication and Access Control
The system uses secure login architecture with encrypted passwords (password_hash()) and session-based authentication. User roles are divided into **Admin** and **Operator**, ensuring structured access to system functions and enhanced data security.

Administrative and Operational Dashboards

- The **Admin Panel** provides statistical dashboards, interactive risk charts (Chart.js), user management tools, date-based filtering, and export options (Excel).
- The **Operator Panel** allows real-time vehicle registration through a dynamic form capturing details such as license plate, vehicle type, origin country, sealing status, and seal quality.

Registered data are stored in two central tables—**vehicles** and **controls**—to maintain clear inspection records.

Database and Application Architecture
The system is designed with modular components (db.php, controllogin.php, header.php, footer.php), enabling scalability and ease of maintenance. The database ensures data integrity through carefully defined relations and foreign keys.

Machine Learning Module
Python, combined with the Random Forest Classifier, forms the core of the risk prediction engine. The ML model:

- is trained using historical vehicle data,
- uses ensemble learning to enhance prediction accuracy,
- processes vehicle characteristics such as type, origin, sealing status, and seal condition,
- is deployed via a Flask API (api_predict.py), allowing real-time predictions.

This modular architecture separates ML processing from the main application, making the system more flexible and scalable.

User Interface Layer
The application employs:

- **Bootstrap 5** for responsive design,
- **Font Awesome** for visual elements,
- **Chart.js** for interactive data visualization.

The interface is designed to ensure intuitive navigation and efficient workflow for customs officers.

Literature Review

Digital transformation in customs systems has gained significant global attention, particularly as nations seek to improve the efficiency, transparency, and security of cross-border trade. Existing literature underscores that modern customs administrations increasingly rely on automation, artificial intelligence (AI), machine learning (ML), and integrated information systems to streamline procedures and strengthen risk assessment.

Digital Transformation in Customs
Studies such as Nenden Restu Hidayah et al. (2025) and Yajing Cai et al. (2024) highlight that digitalization in customs processes significantly improves efficiency, transparency, and international trade performance.

Additionally, institutional practices such as the TradeNet system developed by Singapore Customs (2026) demonstrate how integrated digital platforms can streamline declaration processes, risk management, and monitoring within a unified system. These systems represent best practices in e-customs implementation.

Machine Learning and AI in Risk Analysis
Leo Breiman (2001) provides the foundational framework for machine learning techniques such as Random Forests, which are widely used for classification and prediction tasks. In the context of customs, these techniques can support risk assessment by identifying high-risk shipments based on large datasets.

Furthermore, studies on data analytics in customs operations, such as M. Ramesh and Sruti Vijayakumar (2023), demonstrate the practical application of data-driven approaches in improving risk targeting and operational efficiency.

Digital Monitoring and IoT in Border Control

The application of IoT technologies in logistics and transport systems has been widely discussed in the literature. For instance, Hoang Phuong Nguyen et al. (2023) show how IoT solutions enhance monitoring and management in seaport environments, enabling real-time tracking and improved operational control.

Similarly, Andrew John Poulter and Simon J. Cox (2021) emphasize the importance of secure access and control mechanisms in IoT-based systems, which are essential for reliable monitoring in customs environments.

Digital Transformation, Industry 4.0 and Logistics

Research by Gustavo Araque González et al. (2022) highlights the role of big data and Industry 4.0 technologies in enhancing operational efficiency and decision-making processes.

Additionally, Nur Annisa Ezzaty Rusli et al. (2022) demonstrate how advanced analytical methods improve efficiency and productivity in logistics systems, which are closely related to customs operations.

Customs Performance and Competitiveness

Suzan Oğuz (2024) provides empirical evidence that efficient customs procedures significantly impact global competitiveness. This reinforces the importance of digital transformation as a key driver of performance in customs administrations.

Identified Gap in Literature

Although existing studies highlight the benefits of digital transformation, AI, and IoT in customs and related domains, there remains a gap in the literature regarding:

- Integrated AI-driven monitoring systems for sealed vehicles
- Real-time risk prediction embedded into operational workflows
- Unified platforms combining monitoring, registration, and automated risk assessment

This gap justifies the development of an integrated system that combines machine learning, IoT-based monitoring, and user-centered interfaces to enhance customs processes in Albania, aligning with recent national reforms (Kryeministria e Republikës së Shqipërisë, 2018; Drejtoria e Përgjithshme e Doganave, 2017).

Bulding and development of the application

Application Development

The platform is built using PHP for server-side logic, MySQL for the database, HTML and CSS for page structure and styling, Bootstrap for a clean design, and Font Awesome for icons. JavaScript along with Chart.js is used for chart visualization, while Python is utilized for data analysis and implementing Machine Learning algorithms. The combination of these technologies has enabled the creation of a secure, scalable system that meets the needs of a modern administrative-customs application.

The system begins with the login form, where users must authenticate by entering their username and password. At this stage, several important checks are performed: whether fields are empty, whether the user exists, whether the password is correct, and whether the account is active or passive. For authentication, the PHP $_SESSION session is used to store user information throughout their interaction with the system. If a user attempts to directly access the admin or operator panel without logging in, they are automatically redirected to the login form. Additionally, if an operator attempts to access the admin area, access is granted only if they have the appropriate access level. Security is a key element in this project, so passwords are encrypted using the password_hash() function, ensuring high protection against potential attacks.

After logging in, the user is redirected to the appropriate panel based on their role: Admin or Operator. The admin panel provides an interactive dashboard displaying general statistics, such as the total number of registered vehicles, the number of sealed vehicles, the number of high-risk vehicles, and the ratio of open to sealed vehicles over the last 7 days. Charts visualized with Chart.js offer an intuitive view of risk distribution and vehicle status. The admin also can manage users, view the list of performed controls, filter data by date, and export results in Excel format. The admin has full access to all system functionalities and can perform critical actions such as adding, modifying, and deleting users.

The operator panel is dedicated to registering vehicles entering the system. The dynamic form available to operators allows the input of detailed vehicle information, such as license plate, vehicle type, country of origin, sealing status, whether the seal is damaged or not, etc. After registering a vehicle, the application sends the data to the Flask API for automated risk assessment. This ML model, trained with historical data, analyzes the vehicle's characteristics and returns a risk level assessment: Low, Medium, or High. The registered data is stored in two

database tables: vehicles and controls ensuring a clear connection between the vehicle and the control performed on it.

The database structure is carefully designed to ensure data integrity and efficiency in common queries. The main tables are:

- users for user management,
- vehicles for vehicle data,
- products for the list of transported products,
- mjete_produkt for linking vehicles and products,
- kontrollet for recording customs inspections,
- flamujt_risku for cases where a vehicle is flagged as suspicious.

Relationships between tables are implemented using foreign keys, ensuring organized and easily accessible data.

To ensure well-structured and maintainable code, the application uses shared files such as db.php for database connection, kontrollologin.php for session control, header.php and footer.php for common page components. This modular structure reduces code duplication, simplifies maintenance, and enables easy system expansion in the future. Specific scripts for each functionality, such as perdorues.php for user management, mjete.php for listing vehicles, mrisk.php for high-risk vehicles, mmbyllura.php for sealed vehicles, mdyshuara.php for sealed vehicles with high risk, and midentifikuara.php for sealed vehicles with damaged seals, provide an intuitive and functional interface for users.

The Machine Learning module is a critical component of this project. The training_model.py file trains the risk prediction model using historical data from the database. Using the Random Forest Classifier, the model identifies relationships between vehicle characteristics and risk levels. Then, the api_predict.py file provides a Flask endpoint that accepts new data in JSON format and returns an automated risk prediction. This module makes the application intelligent and enables data-driven decision-making, increasing the efficiency of customs processes.

Overall, this project is an advanced customs system that combines traditional web development with modern artificial intelligence technologies. It provides a secure, user-friendly, and scalable platform that can easily adapt to various administrative requirements. Through the integration of different components, the system automates critical processes, analyzes data in real-time, and offers valuable support for fast and accurate decision-making in customs environments.

The development of the project prototype was based on creating an intelligent web application for managing customs processes, combining modern web development technologies with concepts of Artificial Intelligence (AI) and Machine Learning (ML).

The development process began with the creation of the database structure in MySQL, where core tables such as users, mjetet, kontrollet, produktet, mjete_produkt, and flamujt_risku were designed. Relationships between tables were ensured using foreign keys to maintain referential data integrity. The db.php file was used for database connectivity and included in every necessary page.

Main components of the application

Login & Authentication
The homepage is the login form. The system checks:

If fields are empty (empty()).
If the user exists or not.
If the password is correct or not.
Account status: active or passive.
Roles: Admin, Operator.
Sessions ($_SESSION) are used for authentication and storing user information.
The kontrollologin.php file ensures access only for authorized users.

Modular Application Structure
The application uses shared files to avoid code duplication and facilitate maintenance:

db.php: Database connection to MySQL.
kontrollologin.php: Manages sessions and user access.
header.php: Common page structure (HTML head + header).
footer.php: Page footer (JavaScript, Bootstrap scripts).

Main Database Tables
users: User management (username, password hash, status, access level).
mjetet: Data about vehicles entering the system (license plate, type, country of origin, risk level, etc.).
produktet: List of transported products.
mjete_produkt: Linking table between vehicles and products.
kontrollet: Records of customs inspections.
flamujt_risku: Cases where a vehicle is flagged as suspicious.
Relationships between tables are based on foreign keys to ensure data integrity.

Database Documentation:
Exported from phpMyAdmin (127_0_0_1.sql). This documentation focuses on the logical structure of the dogana database, including tables, relationships, field descriptions, and their functions in the context of an intelligent customs system.

Database Name
dogana Used to manage vehicles, transported products, customs controls, and risk data.

Structure of Main Tables

MJETET *TABLE 3: Information about vehicles entering the customs territory*

Field	Type	Description
id	int (PK)	Unique ID for the vehicle
targa	varchar(20)	Vehicle license plate
lloji_mjetit	varchar(50)	Truck, van, trailer, etc.
vendi_origjines	varchar(50)	Country of origin
i_mbyllur	tinyint(1)	1 = sealed, 0 = unsealed
niveli_riskut	enum	Low, Medium, High – calculated by ML
koha_hyrjes	datetime	Time of entry registration
plumbçja_demtim	tinyint(1)	1 = damaged seal, 0 = intact seal

PRODUKTET *TABLE 4: List of products that can be transported*

Field	Type	Description
id	int (PK)	Product ID
emri	varchar(100)	Product name
kategoria	varchar(50)	E.g., Food, Electronics
risku_produktit	enum	Low, Medium, High

MJETE_PRODUKT *TABLE 5: Linking table for vehicles and transported products*

Field	Type	Description
id	int (PK)	ID
mjeti_id	int (FK)	Reference to mjetet.id
produkti_id	int (FK)	Reference to produktet.id
sasia	int	Quantity of the product in the vehicle

KONTROLLET *TABLE 6: Records customs controls performed on vehicles*

Field	Type	Description
id	int (PK)	Unique ID
mjeti_id	int (FK)	Reference to mjetet.id
emer_kontrolluesi	varchar(100)	Name of the inspector
koha_kontrollit	datetime	Time of the control
verejtje	text	Inspector's comments
rezultati	enum	Passed, Failed, Flagged

FLAMUJT_RISKU *TABLE 7: For cases where a vehicle is flagged for potential risk*

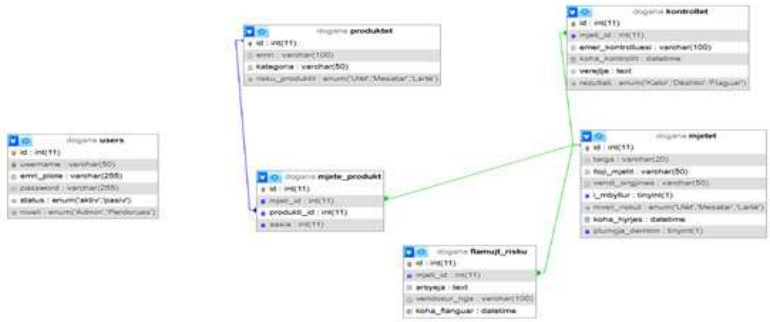
Field	Type	Description
id	int (PK)	ID
mjeti_id	int (FK)	Flagged vehicle
arsyeja	text	Reason for flagging
vendosur_nga	varchar(100)	Who flagged the vehicle
koha_flanguar	datetime	Time of flagging

Administrator Panel
The admin dashboard provides:
General statistics (total number of vehicles, sealed vehicles, high-risk vehicles, etc.).
Visualization charts (Chart.js) for risk distribution and open/sealed vehicle ratio.

Other Functionalities:
User management (add, edit, delete).
Listing of controls and registered vehicles.

Operator Panel
Dynamic form for vehicle registration:
Input: license plate, vehicle type, country of origin, sealing status, etc.
Integration with API for risk prediction (ML).
Data entry into two tables: mjetet and kontrollet.
Based on input, the application sends data to the Flask API for automated risk assessment.

FIGURE 1: The schema of the relationships between entities in the database



Machine Learning Module
Contains two Python files:
training_model.py:
Used for training the ML model.
Uses Random Forest Classifier.
Converts textual data into numerical format.
Saves the trained model in a .pkl file.
api_predict.py:
Flask API providing the /predict endpoint.
Accepts JSON input with vehicle characteristics.
Predicts risk level (Low, Medium, High).
Returns output in JSON format for the PHP application.

System Security
Password Hashing: Uses password_hash() with the PASSWORD_DEFAULT algorithm.
Session Control: Only logged-in users can access sensitive panels.
Logout: Full session termination and redirection to the login form.

FIGURE 2: Application Homepage

FIGURE 3: Login Form

FIGURE 4: Logical Algorithm for the Login Process

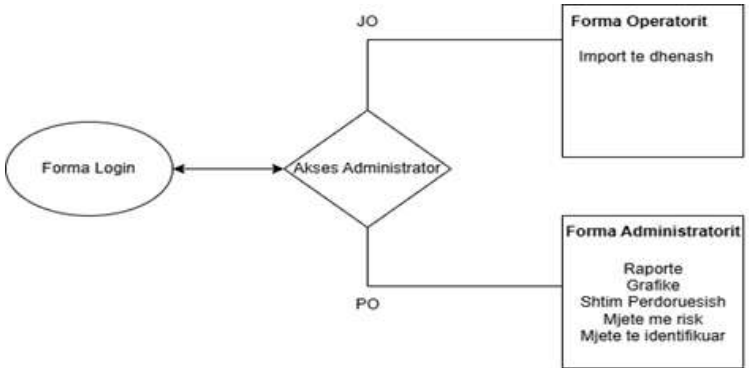
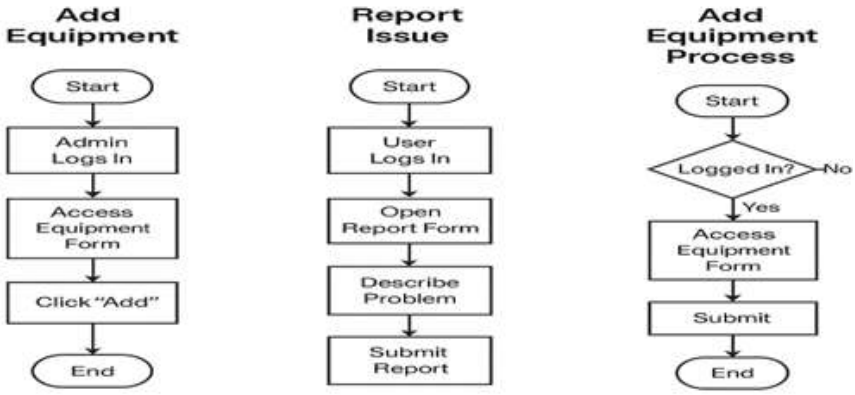


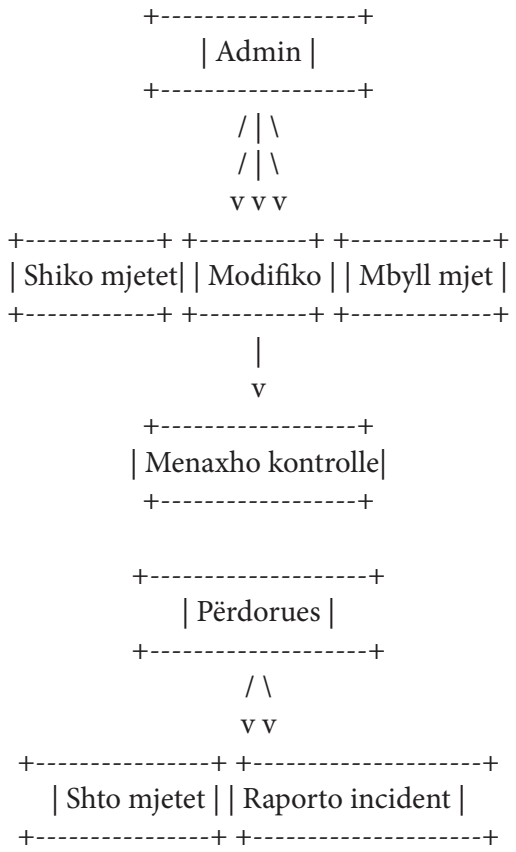
FIGURE 5: Workflow Diagram



Main Actors

Admin: Has full access to the system. Can add,
modify, and close vehicles or controls.
User: Can log in, view vehicles, and report defects or incidents.

DIAGRAM 1: Relationships Between Actors and System Functions



Detailed Use Case Descriptions
UC1: Add Vehicle
Actors: User
Description: The user adds a new vehicle with relevant data (name, type, status).
Normal Flow:
User opens the add form.
Fills in vehicle details.
Clicks the "Add" button.
Vehicle is registered in the database.

UC2: Modify Vehicle
Actors: Admin
Description: Admin updates data for an existing vehicle.
Normal Flow:
Admin selects the vehicle from the list.
Modifies the data.
Clicks the "Save Changes" button.
UC3: View Vehicles
Actors: User
Description: The user can view the list of vehicles and their status.
Normal Flow:
User logs in.
Opens the homepage.
The vehicle list is displayed.
UC4: Report Incident/Control
Actors: User
Description: The user identifies a problem and reports it.
Normal Flow:
Opens the reporting form.
Describes the problem.
Submits the report.
Conditions
If the user does not exist:
A message is displayed indicating that the user does not exist.
If fields are empty:
A message is displayed prompting to fill all fields.

Machine Learning Module for Risk Prediction
Collection of Vehicle Data Customs operators enter vehicle data through the web interface form (index.php). The data includes variables such as:
lloji_mjetit (e.g., Truck, Van),
vendi_origjines (e.g., Albania, Turkey, Greece),
i_mbyllur (0 for unsealed, 1 for sealed),
plumbçja_demtim (0 for no damage, 1 for potential damage). These data are initially stored in the mjetet table (dogana.sql) and prepared for submission to the ML model (Brown, 2020).

Sending Data to the ML Model The entered data are automatically sent from the PHP interface (index.php) to an API built with Flask (api_predict.py) via an HTTP request using cURL. The API loads the trained Random Forest model (saved with joblib from train_model.py) and processes the data to generate

a risk level prediction. For example, a vehicle with vendi_origjines Turkey and plumbçja_demtim 1 may be classified as "High" based on patterns learned from historical data.

Prediction and Result Display the ML model returns a prediction (Low, Medium, High), which is sent back to the PHP interface (index.php) and displayed in the admin panel with a visual notification (e.g., alert alert-success or alert alert-danger from Bootstrap 5). The result is also stored in the niveli_riskut field of the mjetet table, enabling decision tracking and subsequent statistics. This process takes a few seconds, making it suitable for real-time use at border crossings.

Integration with the Database The vehicle data and the predicted risk level are stored in dogana.sql to create a complete record of controls. The kontrollet table is linked to mjetet via foreign keys (kontrollet_ibfk_1), enabling tracking of inspection history and linking with flamujt_risku for sensitive goods. This structure ensures transparency and compliance with the documentation requirements of the Customs Code (Law No. 102/2014).

In this system, two Python files are used to build and serve a machine learning model that predicts the risk level for vehicles passing through customs.

training_model.py – Training Model

This file is responsible for:

Connecting to the MySQL database (dogana) to retrieve historical vehicle data.
Selecting columns: lloji_mjetit, vendi_origjines, i_mbyllur, and niveli_riskut.
Transforming the data:
niveli_riskut is converted into numerical values (0 = Low, 1 = Medium, 2 = High).
Textual columns are converted into numerical format using get_dummies for use by the model.
Training a Random Forest Classifier model based on existing data.
Saving the trained model along with the trained columns in a .pkl file named model_dogana.pkl.

Application testing and functionality analysis

Testing this project was a crucial phase in the application's development, aimed at ensuring that all system components function correctly before the application is put into use. The primary goal was to identify and fix any errors or malfunctions in the server-side logic, database, Machine Learning module, and user interface. The testing was conducted in a planned and systematic manner, including integration testing, user testing, and performance testing.

Initially, the focus was on testing the login process, which is one of the most critical components of the application. Verification was performed to ensure that

checks for empty fields were correctly implemented, preventing users from logging in without entering a username and password. Additionally, testing was conducted on error messages for incorrect credentials or inactive accounts, ensuring they display according to the defined rules. Access levels were rigorously evaluated operators should not have access to the admin panel, while administrators should be the only ones authorized to manage data. The kontrollologin.php file is a key element for security, where sessions and access levels are checked at every step of the process.

Next, the modular structure of the application was tested. Since the code is organized into shared files such as db.php, header.php, footer.php, and kontrollologin.php, it was verified that all pages correctly include these files and that the database connection remains stable across all pages. This modular structure facilitates code maintenance and avoids duplication but requires continuous testing to ensure that changes in one file are properly reflected across all pages that use it.

The next planned phase was testing the forms and vehicle registration actions. The dynamic vehicle registration form was tested to ensure it accepts accurate and complete data, including information such as license plate, vehicle type, country of origin, sealing status, and whether the seal is damaged or not. It was verified whether the data is correctly sent to the database and stored in the kontrollet table. Additionally, the integration with the Python API for risk prediction was tested to ensure it receives input from the form and returns an automated risk level assessment, with the new data accurately stored in the niveli_riskut field.

Regarding database testing, the relationships between tables were analyzed to ensure data integrity and consistency. It was confirmed that foreign keys function correctly, particularly in the relationships between mjetet.id and kontrollet.mjeti_id, mjete_produkt.mjeti_id, etc. CRUD operations (Create, Read, Update, Delete) were tested on each table, including adding, modifying, and deleting users from the admin panel.

The Machine Learning module was also evaluated. It was verified whether the model trained with the training_model.py file can produce reliable results based on historical data. Subsequently, the Flask API in the api_predict.py file was tested to ensure that the /predict endpoint accepts JSON data and delivers accurate risk predictions. Multiple tests with various input combinations were conducted to confirm that the Random Forest Classifier algorithm operates reliably and consistently.

Finally, the application's visual interface was tested for good user experience. Pages such as the admin dashboard, vehicle list, interactive charts with Chart.js, Bootstrap forms, and Excel export buttons were checked for responsiveness, accurate styling, and stable functionality across different devices. It was verified whether date filters function correctly on all relevant pages, such as mjete.php,

kontrollet.php, mrisk.php, etc., and whether Excel exports produce readable and accurate files.

These tests ensure that the application is not only functional but also secure, accurate, and intuitive for users. Testing is an ongoing process where every code or database structure change is accompanied by renewed tests to ensure the overall system quality. This process was supported by detailed documentation of the platforms and technologies used, such as PHP for server-side logic, MySQL for data management, Python for data analysis and machine learning, HTML/CSS/Bootstrap for the user interface, and JavaScript/Chart.js for graphical visualization.

Analysis of results and discussion

Application Performance and Impact on Customs Processes

The developed application for customs processes is an intelligent system that combines modern web development technologies with Artificial Intelligence (AI) and Machine Learning (ML). Based on the conducted tests and performance evaluations, the application successfully meets the designed objectives and has significant potential to optimize customs operations.

The development of this application has a profound impact on modernizing customs processes by automating tasks that were previously manual or semi-automated. For example, vehicle registration is now performed dynamically and automatically, reducing human errors and increasing operational efficiency. The integration of the Machine Learning module for predicting vehicle risk is a revolutionary feature, as it enables the identification of suspicious vehicles before physical inspections begin, saving time and resources.

The admin panel, with its interactive dashboard and visual charts, provides the ability to conduct detailed statistical data analysis, enabling fact-based decision-making. This is particularly useful for strategic-level managers who need to monitor trends, assess the effectiveness of controls, and plan resources more efficiently.

On the other hand, the operator panel is designed to facilitate the operational tasks of customs staff. The dynamic vehicle registration form, combined with the integration of the ML API, ensures accurate workflow, with all information stored in the database in real-time. Additionally, the options for exporting data in Excel format are valuable for further reporting and offline analysis by field experts.

The modular structure of the application, which includes shared files such as db.php, header.php, footer.php, and kontrollologin.php, ensures well-structured, maintainable, and scalable code. This means the application can easily adapt to

future customs system requirements without significant changes to the existing codebase.

Regarding technical performance, the MySQL database connection was evaluated as stable and fast, even when tested with large volumes of data. The Flask API for risk prediction was reliable, returning results in less than a second for each request, making it suitable for real-time use within the PHP application.

Conclusions

The results obtained from testing the system and analyzing its implemented functionalities clearly validate the initial hypothesis. The developed application demonstrates substantial improvements in the organization, automation, and acceleration of customs processes, directly addressing inefficiencies associated with manual procedures. By significantly reducing the need for human intervention, the system minimizes operational errors and enhances the overall reliability of the data generated within customs workflows.

Furthermore, the application establishes a solid technological foundation for future advancements, particularly concerning the integration of more sophisticated artificial intelligence modules for risk assessment and predictive analytics. This positions the system not only as a functional solution for current needs but also as a scalable platform capable of supporting long-term digital transformation within the customs administration.

In its current form, the application has fully achieved its intended objectives. It enables structured and systematic registration and identification of vehicles entering customs control, ensuring that all relevant information is captured accurately and efficiently. The use of a stable MySQL database guarantees secure and consistent data storage, supporting both operational activities and managerial decision-making. The interface has been designed to be intuitive and user-friendly, allowing customs officers to navigate the system with ease and execute their tasks without technical barriers. Additionally, the inclusion of secure login and logout modules strengthens system access management, ensuring that user authentication and data confidentiality are always maintained.

Overall, the application represents a meaningful step forward in modernizing customs operations, enhancing transparency, efficiency, and technological readiness for future expansions.

Bibliography

1. Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324
2. Cai, Y., Zhang, W., Kong, L., Yang, Y., & Yang, Z. (2024). Impact of digital transformation on international trade. *Transactions on Economics, Business and Management Research*, 9, 434–437. https://doi.org/10.62051/xwmyf659
3. Drejtoria e Përgjithshme e Doganave. (2017, June 1). *Risitë e kodit të ri doganor dhe dispozitat zbatuese: Biznesi takim konsultues me doganat*.
4. González, G. A., Suárez Hernández, A., Gómez Vásquez, M., Vélez Uribe, J., & Bernal Avellaneda, A. (2022). Sustainable manufacturing in the fourth industrial revolution: A big data application proposal in the textile industry. *Journal of Industrial Engineering and Management*, 15(4), 614–636.
5. Hidayah, N. R., Susena, K. C., & Rahmadania, D. (2025). Digital transformation of customs in Indonesia's customs system. *SOC*, 1(2). https://doi.org/10.70963/soc.v1i2.185
6. Investment Council Secretariat. (2018). *Annual activity report: January–December 2017—For a "systemic change" in the Albanian investment climate*. Investment Council Secretariat.
7. Khan, M. K., & Ullah, A. (2024). Implication of IoT and its impact on library services: An overview. *Inverge Journal of Social Sciences*, 3(2). https://doi.org/10.63544/ijss.v3i2.94
8. Kosovo Customs. (2024). *Modernization of Kosovo Customs Service*. https://dogana.rks-gov.net
9. Kryeministria e Republikës së Shqipërisë. (2018, January 31). *Sistemi online për procedurat doganore: Transformim historik i doganës*. https://kryeministria.al
10. Law No. 102/2014. (2014). *Customs code of the Republic of Albania*. Official Publishing House. https://www.qbz.gov.al/eli/ligj/2014/102
11. Ministry of Finance. (2024). *Green book – customs*. Publishing House of the Ministry of Finance. https://dogana.gov.al/dokument/5951/libri-jeshil-2025
12. Nguyen, H. P., Nguyen, P. Q. P., Nguyen, D. K. P., Bui, V. D., & Nguyen, D. T. (2023). Application of IoT technologies in seaport management. *International Journal on Informatics Visualization*, 7(1), 228–240.
13. Oğuz, S. (2024). The impact of customs procedures on global competitiveness: An investigation with structural equation modelling. *Fiscaoeconomia*, 8(3), 1264–1278. https://doi.org/10.25295/fsecon.1458266
14. Poulter, A. J., & Cox, S. J. (2021). Enabling secure guest access for command-and-control of Internet of Things devices. *IoT*, 2(2), 236–248. https://doi.org/10.3390/iot2020013
15. Ramesh, M., & Vijayakumar, S. (2023, October 18). How Indian customs leverages data analytics for efficient risk management. *WCO News*, (102). https://www.wcoomd.org
16. Rusli, N. A. E., Ramli, N. A., Shariff, S. R., Zahid, Z., & Hussin, S. A. S. (2022). Evaluating the efficiency and productivity of Malaysian logistics companies using epsilon-based measure and Malmquist index during the COVID-19 pandemic. *Journal of Industrial Engineering and Management*, 15(4), 521–537.
17. Singapore Customs. (2026, February 19). *What you need to know about TradeNet*. https://www.customs.gov.sg
18. Yin, R. K. (2016). *Qualitative research from start to finish* (2nd ed.). The Guilford Press.