

JOMO KENYATTA UNIVERSITY OF AGRICULTURE AND TECHNOLOGY

BSc. COMPUTER SCIENCE

DESIGN AND IMPLEMENTATION OF COMPUTER APPLICATIONS

PROJECT REPORT: FLEET FRONTIER

GROUP MEMBERS

1. MACHARIA MAURICE - SCT211-0010/2022
2. JOSEPH NGURE - SCT211-0008/2022
3. PEACHES NJENGA - SCT211-0004/2022
4. HILDA MWANGI - SCT211-0026/2022
5. THEURI BONFACE KARUE - SCT211-0573/2022
6. KING'ORI FLORENCE WANGECHI - SCT211-0063/2022

INTRODUCTION:

Information on Fleet Frontier

Fleet Frontier emerges as a solution in response to the inherent inefficiencies plaguing traditional matatu management practices. Historically, matatu operators and fleet managers have relied on manual methods for storing and managing crucial operational records. This manual approach, characterised by paper-based systems and disjointed record-keeping methods, has often led to inefficiencies, errors, and suboptimal decision-making processes.

Purpose and Objectives of the Project

In light of these challenges, the primary objective of our project is to catalyse a paradigm shift in matatu management by transitioning from manual record-keeping to electronic data management through Fleet Frontier. By harnessing the power of technology, Fleet Frontier seeks to revolutionise the way matatu operations are managed, offering a centralised platform for storing, accessing, and analysing critical operational data.

Scope and Limitations

Fleet Frontier's scope extends far beyond the mere digitization of records; it aims to redefine the entire landscape of matatu management by providing comprehensive functionality for personnel management, vehicle management, trip scheduling and rental management. By automating previously manual processes, Fleet Frontier not only improves operational efficiency but also empowers matatu operators and fleet managers with actionable insights derived from real-time data.

While the transition from manual to electronic record-keeping represents a significant leap forward, it's essential to acknowledge the limitations inherent in this transformation. Challenges such as data migration, user adoption, and infrastructure requirements may pose initial hurdles. However, through strategic planning and iterative development, Fleet Frontier endeavours to overcome these challenges and pave the way for a more efficient and sustainable future in matatu management."

Here is a link to our Github project repository:

<https://github.com/Peachy-Njenga/Fleet-Frontier>

PROJECT OVERVIEW

Description of Fleet Frontier's concept

Fleet Frontier represents a pivotal advancement in matatu management systems, offering a comprehensive solution to address the diverse challenges encountered by matatu operators and fleet managers. At its essence, Fleet Frontier serves as a centralised hub for managing all aspects of matatu operations, spanning personnel management, vehicle management, trip scheduling and rental management.

Target audience

The target audience for Fleet Frontier comprises matatu operators and fleet managers seeking to optimise their operations. By offering a unified platform that caters to the unique needs of the matatu industry, Fleet Frontier aims to empower operators with the tools and insights necessary to streamline operations and enhance overall efficiency.

Justification for specific features and functionalities

In developing Fleet Frontier, careful consideration was given to the selection and implementation of features and functionalities to ensure that they align closely with the needs and challenges faced by matatu operators and fleet managers.

Personnel Management:

Efficient management of drivers and conductors is crucial for ensuring smooth operations and providing quality service to passengers. The personnel management feature in Fleet Frontier allows operators to store and manage employee details, including contact information. By centralising this information, Fleet Frontier simplifies the process of personnel management, enabling operators to track employee assignments and schedule shifts effectively.

Trip Scheduling and Management:

Efficient trip scheduling and management are critical for maximising revenue, minimising idle time, and providing reliable transportation services to passengers. Fleet Frontier's trip scheduling and management feature allows operators to plan and schedule trips dynamically based on demand, driver availability, and vehicle capacity. Additionally, operators can assign drivers and conductors to specific trips and track trip status in real-time. By automating trip scheduling and management processes, Fleet Frontier enables operators to optimise resource allocation, minimise wait times for passengers, and improve overall service quality.

Rental management

Through Fleet Frontier, managers can easily view the availability of vehicles for rent in real-time. This enables them to quickly identify which vehicles are ready for rental, streamlining the booking process and minimising delays.

Additionally, Fleet Frontier tracks the return status of rentals, ensuring that vehicles are promptly returned.

Technologies and Tools Used for Development

Programming Languages:

Visual Basic .NET (VB.NET): Fleet Frontier was developed using the VB.NET programming language, which provided a powerful and versatile framework for creating Windows desktop applications.

Databases:

MySQL: The relational MySQL database management system was employed as the primary database for Fleet Frontier. MySQL offers robust features for data storage, retrieval, and

management, making it well-suited for handling the complex data structures inherent in matatu management systems.

Frameworks:

.NET Framework: Fleet Frontier utilised the .NET Framework, a comprehensive software development platform for building Windows-based applications. The .NET Framework provided a rich set of libraries, tools, and APIs that facilitated rapid application development and seamless integration with other Microsoft technologies.

Development Tools:

Visual Studio: Visual Studio served as the primary integrated development environment (IDE) for building Fleet Frontier. With its extensive features and intuitive interface, Visual Studio streamlined the development process, enabling developers to write, debug, and deploy applications with ease.

Azure Data Studio: Azure Data Studio was used for database development and management tasks. This lightweight, cross-platform tool offered a rich set of features for querying databases, exploring data, and performing administrative tasks.

Docker: Docker was employed for containerizing the MySQL database used in Fleet Frontier. By encapsulating the database environment within a Docker container, developers ensured consistency, portability, and scalability across different deployment environments.

Git and GitHub: Git version control system, in conjunction with GitHub, was used for source code management and collaboration. Git enabled developers to track changes, manage code branches, and merge contributions seamlessly, while GitHub provided a centralised platform for hosting repositories and facilitating collaboration among team members.

Development Process

Methodology

For the development of our project, we opted for the Agile methodology. The rationale behind this choice was to enable flexibility and adaptability in response to changing requirements and/or feedback.

Agile emphasises iterative development cycles, continuous collaboration with stakeholders, and the ability to quickly respond to changes. This approach was deemed suitable for our project due to its dynamic nature and the need for frequent feedback loops.

Roles and Responsibilities

Project Manager:

1. Peaches Njenga

Group leader - Oversaw the entire development process, including task assignment, progress tracking, and ensuring alignment with project objectives.

Designers:

1. Peaches Njenga - user interface design and user experience aspects of the project to enhance usability.

Developers(Frontend):

1. Florence Wangechi
2. Peaches Njenga
3. Hilda Mwangi

They were responsible for creating various user controls and functionalities of the different forms used in our project.

Developers(Backend):

1. Joseph Ngure
2. Maina Maurice
3. Theuri Bonface

They were responsible for implementing the database in MySQL, connecting the backend to the frontend and implementing functionalities of the project.

Testers:

All members

We ensured that the quality and functionality of the developed features went through rigorous testing.

Timeline and Milestones

The development process was divided into several phases, each with its set of milestones:

Planning Phase: Defined project scope, objectives, and requirements. (Duration: 2 days)

Design Phase: We created wireframes and mockups to help with the creation of the different forms used in the project. (Duration: 3 days)

Development Phase: Implemented features based on the design specifications. (Duration: 3 weeks). In here we used different use cases as a guide e.g. working on handling vehicles, working on staff and working on renting.

Testing and Quality Assurance: Conducted testing to ensure functionality and resolve any issues. (Duration: 2days)

We carried out the following testing methods:

1. Unit Testing: This type of testing involves testing individual components or units of code in isolation to verify that they work as expected. It typically focuses on small, granular units of code, such as functions or methods. E.g. we tested to make sure that various buttons such as add vehicle and add staff are working as expected
2. Integration Testing: Integration testing evaluates the interactions and interfaces between different components or modules to ensure they work together seamlessly. It aims to identify any issues that may arise from the integration of individual units. Eg we made sure that the different user controls integrated well with other components of our system.
3. System Testing: System testing evaluates the entire system as a whole to verify that it meets specified requirements and functions correctly in its intended environment. It tests the system's behaviour and performance under various conditions. Once we completed the major requirements of our system we did a final test on the whole system to make sure that it met our requirements to the letter.

Milestones

1. Project Kickoff: This marked the official start of the project, including initial planning, team formation, and goal setting.
2. Completion of Requirements Gathering: This milestone signifies the completion of gathering and documenting project requirements, including functional and non-functional requirements.
3. Design Approval: The project's design, including wireframes, mockups, and prototypes, were finalised and approved.
4. Development Phase Initiation: This milestone marked the start of the development phase, where coding and implementation of features began based on the approved design.
5. Feature Complete: All major/important planned features and functionalities have been implemented, and the core functionality of the project is in place.
6. Alpha Release: This milestone marked the first internal release of the project for testing purposes within the development team.
7. Project Presentation: We eventually presented our project to our lecturer for grading.

Challenges and Strategies

Throughout the development process, several challenges were encountered:

Resource Constraints: Limited availability of certain resources, such as specialised skills considering we had to learn Visual Basic from scratch and start implementing what we've learnt from scratch.

Scope Creep: Additional features or requirements were introduced during the development, leading to scope expansion eg inclusion of a search bar

Technical Hurdles: Unexpected technical issues or limitations arose during implementation. One of them being performance bottlenecks since Visual Studio (the application we were using for development) was resource-intensive and would require us to sometimes clear storage or restart our devices from time to time.

To overcome these challenges, the following strategies were employed:

- a. Prioritisation: Focused on essential features and functionalities to ensure timely delivery.

These are the areas we prioritised:

1. Staff management
2. Vehicle management
3. Trip scheduling
4. Rental services

The areas we didn't get to implement due to time constraint include:

1. There arose a need to have the search bar but its functionality wasn't implemented.

- b. Communication:

Maintained open communication channels within the team.

We used tools such as Microsoft To-Do to help keep track of tasks due and those accomplished and Whatsapp to help with the communication.

- c. Adaptability:

Embraced changes and iterations, adjusting plans and priorities as needed to accommodate evolving requirements.

Changes During Development

Several changes were made during the development process based on feedback and evolving requirements:

Feature Adjustments:

Certain features were modified or reworked to better align with user needs and expectations.

Eg our homepage was changed from that of a vehicle's picture to depicting the current activities taking place.

Interface Enhancements:

User interface elements were refined based on usability testing and feedback from stakeholders.

Eg Initially the all front end developers did their designs separately and implemented them. However, there was a need for uniformity in our system and on how the forms looked which led to the creation of a template on which the frontend developers worked around on.

TECHNICAL DETAILS

Architecture of Fleet Frontier:

Fleet Frontier implements a robust client-server architecture to ensure efficient data management and seamless communication between matatu operators and fleet managers.

i. Client-Side:

The client-side application is developed using Visual Basic .NET (VB.NET) within the Microsoft .NET Framework. This choice offers several advantages, including a rich set of libraries for creating Windows-based desktop applications, a familiar programming environment for developers, and seamless integration with other Microsoft technologies.

ii. Server-Side:

On the server-side, Fleet Frontier leverages a combination of technologies to support the centralised database and business logic:

MySQL Database: Fleet Frontier utilises the MySQL relational database management system for storing and managing all essential data related to matatu operations such as vehicle, staff, renting details etc. MySQL is popular for its scalability, reliability, and performance, making it an ideal choice for handling the complex data structures inherent in matatu management systems.

.NET Framework: Fleet Frontier utilises the .NET Framework for implementing the server-side logic and facilitating communication between the client and server components.

The .NET Framework provides a robust development platform with extensive libraries, tools, and APIs, enabling developers to build scalable and efficient server-side applications.

Docker: Docker containers are employed to containerize the MySQL database, ensuring consistency, portability, and scalability across different deployment environments. Docker simplifies database deployment and management tasks, offering benefits such as isolation, resource efficiency, and easy scaling to meet growing demand.

Overview of Programming Languages, Frameworks, and Libraries Used:

i. Programming Languages:

Visual Basic .NET (VB.NET): Used for client-side application development, VB.NET offers a versatile framework for creating intuitive and feature-rich Windows desktop applications. Its integration with the .NET Framework simplifies development and enhances the user experience.

ii. Frameworks and Libraries:

.NET Framework: Fleet Frontier relies on the .NET Framework for both client-side and server-side development. With its comprehensive set of libraries and tools, the .NET Framework accelerates application development, improves productivity, and ensures compatibility with other Microsoft technologies.

MySQL Connector/.NET: Used for establishing connections between the Fleet Frontier application and the MySQL database, the MySQL Connector/.NET library simplifies data access and manipulation tasks, ensuring seamless communication between the client and server components.

Entity Framework: Fleet Frontier leverages Entity Framework, an ORM framework within the .NET Framework, to abstract the underlying database schema and streamline data access operations. Entity Framework simplifies CRUD (Create, Read, Update, Delete) operations, enhances code maintainability, and improves developer productivity.

Database Design and Implementation for Managing Matatu Data:

Fleet Frontier's database design was implemented using Azure Data Studio. It focuses on efficiency, scalability, and data integrity to meet the complex requirements of matatu management:

Tables: The database comprises multiple tables, each representing a specific entity or aspect of matatu operations, such as employee, vehicles, trips and rentals. This modular approach enhances data organisation and facilitates efficient data retrieval and manipulation.

Relationships: Tables are linked through relationships to establish associations between different entities, enabling efficient data retrieval and integrity constraints enforcement.

Foreign key constraints ensure referential integrity, maintaining consistency and accuracy across related data tables.

Indexing: Indexes are applied to key columns to optimise query performance and facilitate faster data retrieval, especially for frequently accessed data. Indexes improve database efficiency by reducing the time required to locate and retrieve specific records, enhancing overall system responsiveness.

Constraints: The database employs various constraints, including primary keys, foreign keys, unique constraints, and check constraints, to maintain data integrity and enforce business rules. Constraints prevent data inconsistencies, ensure data accuracy, and uphold the integrity of the matatu management system.

User Interface Design Considerations for Matatu Operators and Fleet Managers:

Fleet Frontier's user interface design prioritises usability, intuitiveness, and efficiency to meet the specific needs of matatu operators and fleet managers:

Dashboard: The dashboard provides a centralised overview of key metrics, such as the vehicle status, staff, activity status and the calendar . This intuitive dashboard enables users to quickly assess the current state of operations and make informed decisions.

Navigation: The user interface features intuitive navigation menus and controls, allowing users to easily access different features and functionalities within Fleet Frontier. These are situated on the navigation menu which allow users to choose the different forms they want to interact with. Clear navigation paths enhance user experience and streamline workflow navigation.

Forms and Input Controls: Forms and input controls are designed to be user-friendly and responsive, simplifying data entry, retrieval, and manipulation tasks such as adding, deleting and updating details of different entities such as vehicles for operators and managers. Intuitive input controls and validation mechanisms eg the log in page that restricts unauthorised access. This ensures data accuracy and consistency, reducing the risk of errors.

Features and Functionalities

Key Features

1. **Vehicle Management:** Fleet Frontier provides comprehensive vehicle management functionalities, allowing matatu operators to track and manage their fleet efficiently. This includes features such as adding, updating, and deleting vehicles from the system, as well as monitoring vehicle availability.
2. **Trip Management:** Matatu operators can plan and manage trips seamlessly using Fleet Frontier. The system enables users to schedule trips, assign drivers and conductors, and track trip details such as departure and arrival times and trip routes.

3. **Employee Management:** Fleet Frontier facilitates the management of employees, including drivers and conductors. Users can add, update, and remove employee records, assign them to specific trips, and track their availability.
4. **Rental Management:** The platform offers rental management capabilities, allowing users to handle vehicle rentals efficiently. This includes features such as adding new rental records, updating existing rentals, tracking customer information, and managing rental statuses.
5. **Availability Tracking:** Fleet Frontier provides real-time tracking of vehicle and employee availability. Users can monitor the availability status of vehicles, drivers, and conductors, enabling better resource allocation and scheduling.

Features to be included in next versions:

1. **Report Generation:** Enhance the reporting capabilities of Fleet Frontier to allow users to generate various types of reports, such as trip summaries, vehicle utilisation metrics, employee performance analytics, and financial reports. These reports can provide valuable insights into the operational efficiency and performance of the fleet, helping operators make informed decisions and identify areas for improvement.
2. **Searching and Filtering:** Implement advanced search and filtering functionalities within Fleet Frontier to enable users to quickly find and access relevant information. This can include the ability to search for specific vehicles, trips, employees, or rental records based on various criteria such as date, location, vehicle type, or customer name. Advanced filtering options can help users narrow down search results and retrieve the information they need more efficiently.
3. **Maintenance Scheduling:** Introduce a maintenance scheduling module to help operators manage and track vehicle maintenance tasks effectively. This feature can allow users to schedule routine maintenance activities, set reminders for upcoming service appointments, and track maintenance history for each vehicle in the fleet. Integrating maintenance scheduling with other parts of the system can ensure that vehicles remain in optimal condition, minimising downtime and reducing the risk of unexpected breakdowns.

User Stories

1. As a fleet manager, I want to easily add new vehicles to my fleet so that I can keep track of all vehicles.
2. As a fleet manager, I want to schedule trips efficiently by assigning drivers and conductors to specific routes and monitoring trip details in real-time.

3. As a matatu operator, I want to maintain accurate employee records, including driver and conductor details, to ensure smooth operations and compliance with regulations.
4. As a rental manager, I want to manage vehicle rentals effectively by updating rental information, tracking customer details, and monitoring rental statuses.
5. As a fleet administrator, I want to track vehicle and employee availability to optimise resource allocation and ensure timely service delivery.
6. As a fleet manager, I want to easily view available vehicles for hire to plan trips efficiently.
7. As a fleet manager, I want to update the availability status of vehicles to reflect their operational status accurately.
8. As a fleet manager, I want to create new trips with detailed information such as origin, destination, and vehicle assignment.
9. As a driver or conductor, I want to view my assigned trips and their details for better planning and execution.
10. As a fleet manager, I want to assign drivers and conductors to trips based on their availability and expertise.
11. As a customer, I want to rent a vehicle by providing my name and contact information for a specific duration.
12. As a rental manager, I want to update the rental status of vehicles (returned/not returned) to manage availability effectively.
13. As a user, I want to navigate between different features of the application easily using intuitive controls.
14. As a user, I want to see clear and concise information displayed on the interface for quick decision-making.
15. As a manager, I want the application interface to be responsive and accessible across various devices for flexibility in usage.
16. As a user, I want to receive informative error messages if I input invalid data or encounter errors during operation.

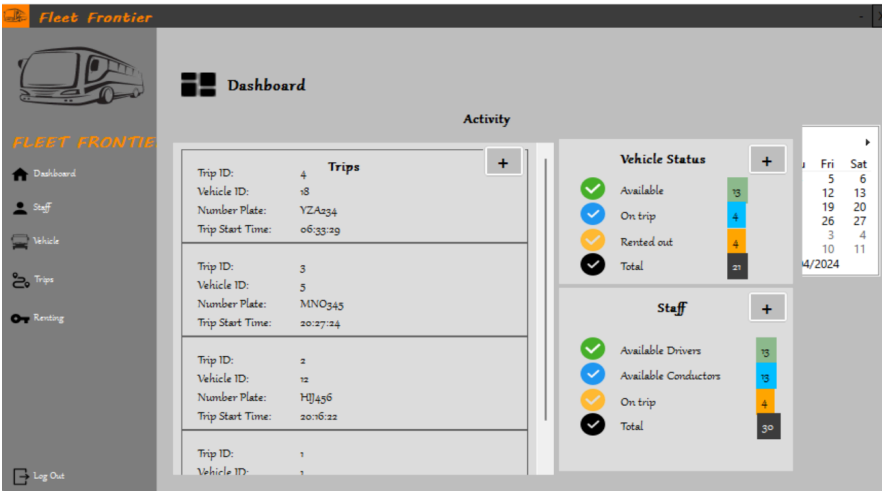
17. As a manager, I want to ensure that all data entered into the system is accurate and validated to maintain data integrity.

18. As a developer, I want to implement robust error handling mechanisms to handle unexpected situations gracefully and enhance user experience.

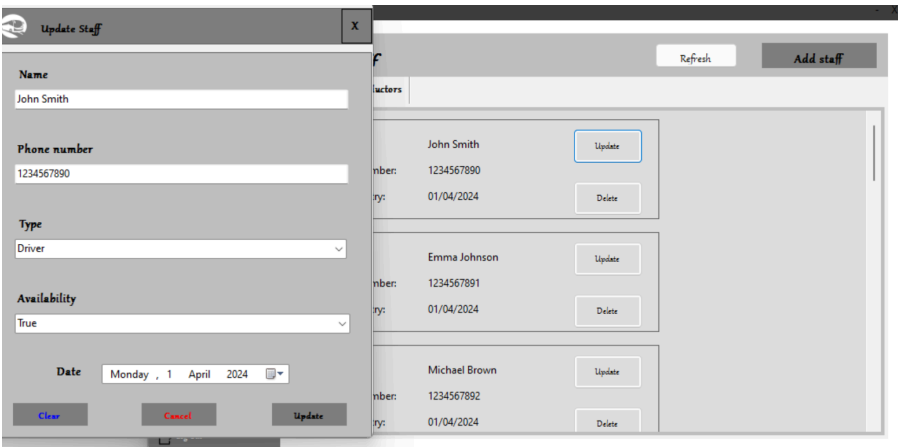
These features and user stories demonstrate how Fleet Frontier streamlines matatu management processes, enhancing operational efficiency and customer satisfaction.

Screenshots illustrating functionality

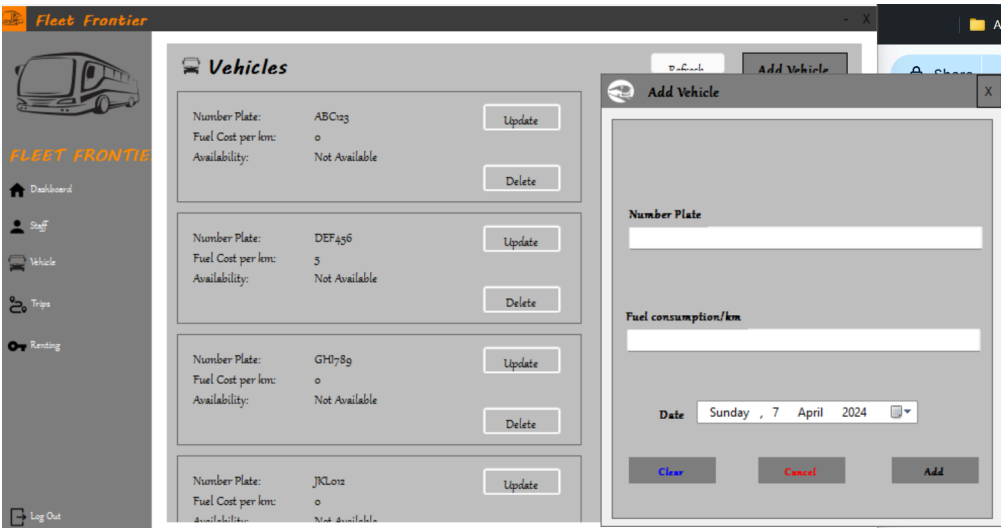
1. User Dashboard



2. Staff Management



3. Vehicle Management



4. Trip Management

The screenshot shows the 'Trips' management page in the Fleet Frontier application. The left sidebar contains navigation links: Dashboard, Staff, Vehicle, Trips (selected), Renting, and Log Out. The main content area is titled 'Trips' and features a 'Refresh' button and a 'Start Trip' button. It displays two trip entries, each with a form containing fields for From, To, Start Time, Driver Id, Driver Name, Conductor ID, Donor's Name, Vehicle, and Status. The first trip is 'Complete' and the second is 'In transit'. Each entry has 'Update' and 'Delete' buttons.

From	To	Start Time	Driver Id	Driver Name	Conductor ID	Donor's Name	Vehicle	Status	Update	Delete
Nairobi	Thika	20:15:54	1	John Smith	18	Richard Johnson	ABC123	Complete	Update	Delete
Thika	Juja	20:16:22	3	Michael Brown	17	Lisa Davis	HJJ456	In transit	Update	Delete

5. Rental Management

The screenshot shows the 'Rent out' management page in the Fleet Frontier application. The left sidebar is identical to the previous screenshot. The main content area is titled 'Rent out' and features a 'Refresh' button and a 'New Rental' button. It displays four rental entries, each with a form containing fields for Customer Name, Phone Number, Vehicle Number, and Rental Status. Each entry has 'Update' and 'Delete' buttons. A confirmation dialog box is open in the foreground, asking 'Are you sure you want to delete this rental?' with 'OK' and 'Cancel' buttons.

Customer Name	Phone Number	Vehicle Number	Rental Status	Update	Delete
Kevin	0981967	ABC123	Not Returned	Update	Delete
Mike	09817266	GHD789	Not Returned	Update	Delete
Boryface	0993732	JKL012	Not Returned	Update	Delete
Hilda	0972814	PQR5678	Not Returned	Update	Delete

Testing and Quality Assurance

Testing Methodologies Employed:

The testing phase of the Matatu Fleet Management System project followed a comprehensive approach, incorporating various methodologies to ensure the reliability and performance of the software solution. The testing methodologies employed included:

1. **Unit Testing:** Individual components and modules of the system were tested in isolation to verify their functionality and behaviour according to design specifications. For example: Test Case: Verify that the login function authenticates users with valid credentials. Test Steps:
 - a. Enter valid username and password.
 - b. Click on the "Login" button.
 - c. Verify that the user is redirected to the dashboard page.
 - d. Confirm that the user's session is active.
2. **Integration Testing:** Tested the integration of different modules and subsystems to ensure seamless communication and data flow between components. For example: Test Case: Verify that data from the vehicle tracking module accurately synchronises as vehicles update availability in the central database. Test Steps:
 - a. Simulate vehicles updating their availability status (e.g., becoming available or unavailable).
 - b. Trigger synchronisation process.
 - c. Query the database to verify that the availability status of the vehicles is updated correctly.
 - d. Compare the synchronised data with the original input.
3. **System Testing:** The entire system was tested as a whole to validate its functionality, usability, and performance in a real-world environment. For example: Test Case: Verify that the system dashboard displays the list of available vehicles. Test Steps:
 - a. Access the dashboard interface.
 - b. Observe the list of available vehicles.
 - c. Confirm that the list displays accurate information about vehicle availability.

Results of Testing Phases:

During the testing phases, several bugs and issues were identified and addressed to enhance the quality and reliability of the system. Some of the key findings include:

1. **Data Synchronisation Issues:** In the integration testing phase, discrepancies were observed in data synchronisation between the central database and the mobile application. This issue was resolved by optimising data transfer protocols and implementing synchronisation checks.

2. **User Interface Glitches:** User acceptance testing uncovered minor user interface glitches and inconsistencies across different devices and screen resolutions. These issues were rectified through UI refinements and compatibility testing.

Quality Assurance Measures Implemented:

To ensure the reliability and performance of the system , the following quality assurance measures were implemented throughout the development and testing process:

1. **Code Reviews:** Regular code reviews were conducted to identify and rectify coding errors, ensure adherence to coding standards, and promote best practices.
2. **Continuous Testing:** continuous test suites were conducted to execute regression tests and ensure the stability of new features and enhancements.
3. **Performance Monitoring:** Real-time performance monitoring techniques were integrated into the system to monitor system metrics and identify performance issues proactively.

By implementing these quality assurance measures, the system project aimed to deliver a robust, reliable, and high-performance solution that meets the needs of its users and stakeholders.

Conclusion

The development of Fleet Frontier marks a significant milestone in the evolution of matatu management systems, offering a comprehensive solution to address the longstanding challenges faced by operators and fleet managers. Through meticulous planning, agile development methodologies, and collaborative efforts, our team has successfully crafted a robust and feature-rich platform designed to streamline operations, enhance efficiency, and empower stakeholders with actionable insights derived from real-world observations.

Key Findings and Outcomes:

Throughout the development journey, we encountered various challenges, ranging from resource constraints to technical hurdles. However, through strategic prioritisation, effective communication, and adaptability, we were able to overcome these challenges and deliver a solution that meets the diverse needs of the matatu industry.

The implementation of Fleet Frontier has yielded several positive outcomes, including:

- **Efficient Vehicle and Trip Management:** Fleet Frontier empowers operators to manage their fleets and schedule trips dynamically, resulting in improved resource allocation and enhanced service quality.
- **Streamlined Personnel Management:** The platform simplifies the management of drivers and conductors, enabling operators to track assignments and schedules effectively.

- **Effective Rental Management:** Fleet Frontier facilitates seamless handling of vehicle rentals, optimising the booking process and minimising delays.
- **Enhanced User Experience:** The intuitive user interface and responsive design of Fleet Frontier enhance usability and accessibility, ensuring a smooth experience for operators and managers.

Lessons Learned:

The development and deployment of Fleet Frontier have taught us valuable lessons that will guide us in future endeavours. Some of the key takeaways include the importance of effective project management, the value of continuous communication and collaboration, and the significance of adaptability in navigating challenges and changes. These lessons will serve as valuable assets as we continue to innovate and develop solutions to address real-world problems.

Reflection on the Project Experience:

The journey of developing Fleet Frontier has been both challenging and rewarding. We have learned valuable lessons about teamwork, communication, and adaptability, which have contributed to our personal and professional growth. Moreover, the project has provided us with the opportunity to aim to make a meaningful impact on the matatu industry by introducing a transformative solution that addresses critical pain points and drives operational efficiency.

Acknowledgments:

We extend our sincere gratitude to all team members, mentors, and stakeholders who contributed to the success of this project. Their dedication, expertise, and support were instrumental in overcoming obstacles and achieving our goals.

Additionally, we would like to thank our lecturer for providing guidance and encouragement throughout the development process.

In conclusion, Fleet Frontier represents not only a technological advancement in matatu management but also shows the impact of collaboration and innovation. As we look towards the future, we remain committed to refining and enhancing Fleet Frontier to further meet the evolving needs of the matatu industry and contribute to its continued growth and sustainability.