

Review Article

Open Access

Optimizing Transportation and Logistics Operations with Power Automate and .NET Full-Stack Development

Naga Lalitha Sree Thatavarthi

USA

ABSTRACT

The transport and logistics industry needs a system that will solve the current challenges faced in the industry, which are related to order management, fleet optimization, and inventory control. This paper delves deeper into how Power Automate and .NET full-stack applications can help optimize transportation and logistics operations. Power Automate is a robust platform that will help automate workflows between systems to ensure an optimized flow of processes. This paper will focus on the development of this application, its benefits, and its challenges. The challenges currently faced in the transport and logistics industry will be highlighted, as well as the desired features the .NET application should have.

*Corresponding author

Naga Lalitha Sree Thatavarthi, USA.

Received: February 12, 2024; Accepted: February 18, 2024, Published: February 25, 2024

Keywords: .NET Full Stack Application, Power Automate

Introduction

The transport and logistics sector are huge because it helps transport goods from manufacturers to consumers. This industry faces many challenges that deter it from performing its functions. The main issues faced in this industry include inefficient inventory management systems, problems with fleet management, issues with order processing management, and lastly, failure of efficient systems to integrate the different systems in the industry [1]. All these challenges have made it hard for the transport and logistics industry to perform its tasks as intended.

Technology is an essential tool that will make improvements in the transportation industry. Power Automate and .NET full-stack development in this sector will be very efficient because they are powerful enough to develop systems and ensure they run smoothly. The role of Power Automate is to provide the automatic integration of processes in the transportation and logistics industry. These processes include order processing, inventory management, and fleet management. The .NET full stack development involved the development of the client side and server site architectures and implementing tools that will enable both architectures to communicate and run smoothly [1].

These two technologies will address the industry's main challenges and ensure better systems for automated order processing, optimized fleet management, efficient inventory control, and smooth integration between the three systems. The system's effectiveness in terms of accuracy, timeliness, and overall smooth running will also improve customer satisfaction. This paper highlights the benefits of adopting the Power Automate service and .NET full-stack development.

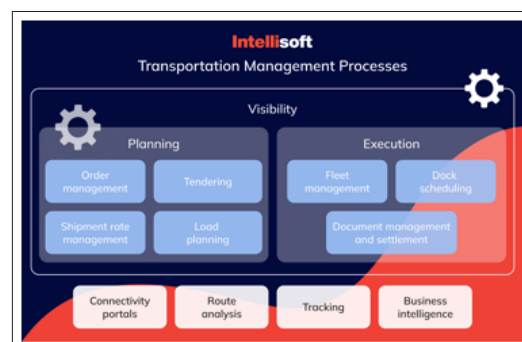


Figure 1: Shows the Transport Management Process

Current Challenges Faced

The transportation and logistics industry faces various challenges. The following outlines the main challenges that can be addressed by Power Automate and .NET full-stack development.

Inefficient Order Processing

The current order processing system is manual and prone to human errors, including misspellings. This manual system is very time-consuming, leading to delays in dispatching orders. There are no real-time updates when using this system; therefore, many inconsistencies in data will be faced, which usually takes a long time to resolve. This system cannot handle actions like cancellations or changes in order, which will cause dissatisfaction among the consumers [2]. Delays and inaccuracies often hurt and negatively impact the customers because they will not be fulfilled with their experience, losing their loyalty and trust in the company.

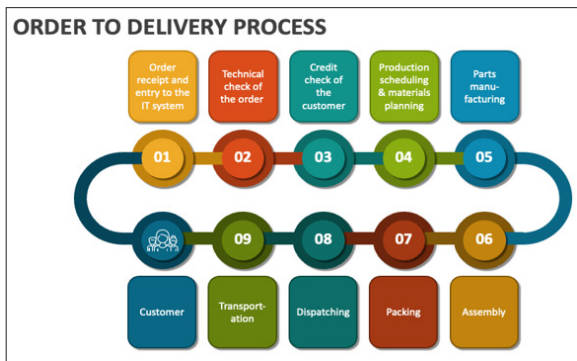


Figure 2: Shows the Correct Steps to Be Followed to Process

Unorganized Fleet Management Systems

Regarding fleet management, one of the main issues faced is an inefficient selection of routes to deliver orders. Roads with poor conditions or heavy traffic should be avoided because it will cause the delivery vehicle to take longer to give orders. The current system used in fleet management does not provide real-time data on the current state of the roads; this information is essential because it helps the delivery team know the streets that have been closed or when there is heavy traffic [3]. The transportation industry nowadays has vehicles in poor condition, hence experiencing breakdowns that cause delivery delays.

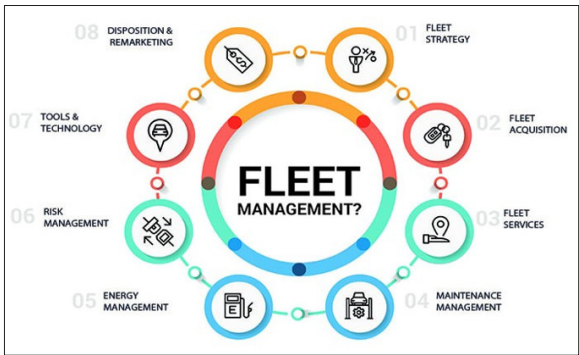


Figure 3: Shows A Fleet Management System

Inefficient Inventory Management

Several inaccuracies are needed in inventory management. These errors include stock discrepancies, whereby the number of stock on the records differs from the number of items in stock. This inconsistency often leads to stockouts and overstocking. There needs to be a more consistent supply chain because the current system cannot automatically coordinate with the suppliers to know when to replenish the stock. This system often causes customer dissatisfaction because the items they want to order may always be out of stock [4].



Figure 4: Shows the Correct Process of Inventory Management

Poor Integration Between Systems

The current manual system has poor data integration systems because there is isolated data whereby the systems put in place operate separately, leading to the creation of data silos and inconsistencies. This separation of systems often makes it hard for managers to decide on the system's efficiency. The lack of real-time data often causes errors in the process of data trading because it is done manually. This problem often causes delayed updates because of manual data transfer, making the system unreliable.

Technical Framework

Power Automate

This cloud-based cloud-based framework is responsible for automating workflows in systems. It allows its consumers to define repetitive tasks in the system and enable its triggers so that the framework identifies one task; it is automatically done. This framework comes with built-in connectors that help the system connect to other systems such as MySQL, ERP, and many more. This framework can be used in order processing, whereby it will automate actions such as verification, payment, and shipment.

This framework can be applied to inventory management systems, whereby it performs inventory tracking and updates to ensure real-time data throughout the system. This framework can be connected to suppliers to notify them and replenish the stock whenever it is low. Power Automate can be used in fleet management to send vehicles to deliver goods to clients; this system is crucial in ensuring fleet optimization because it will select the most suitable route for the cars to use to save time [5]. This framework is also responsible for sending the clients updates on the delivery status of the items they ordered to ensure that the client is informed.

The .NET Full-Stack Development

Various .NET platforms were used in the development of this application. One of the .NET applications used in this case is the ASP.NET core, which is used in the backend development of the system. This application is reliable, scalable, and high-performing high-performing, suitable for serverside architecture use. A system is needed to manage the database used; therefore, Entity Framework core is used to manage the databases used in this case. Blazor is another application that can be used to develop the front end of the application. It uses C#, which is efficient and high-performing and can be easily integrated with the backend to ensure smooth running [6].

Various benefits come with the development of these applications, some of them being that they are scalable, meaning the system will still work efficiently if it processes large volumes of data. These very secure applications ensure data privacy and confidentiality, ensuring the client's data is safe. It applied various steps like authorization, verification, and reporting to ensure that data is accessible to the right person only. These applications ensure cross-platform development where the programmer can use the application to develop the system on any operating system [6]. These flexible applications can seamlessly integrate into other systems using Azure services and RESTful APIs.

System Implementation

Overall System Design

This system comprises frontend, backend, and API gateways that connect the front and backend.

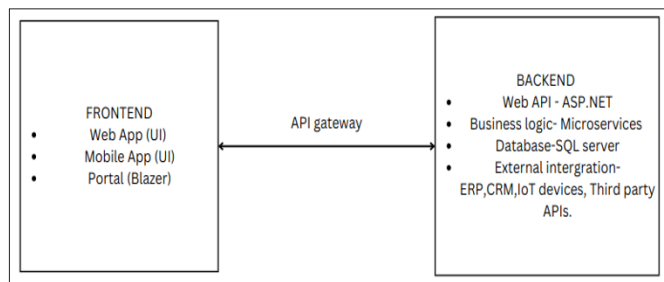


Figure 5: Represents the Overall System Design

The front end is the interface that the client can use. It is designed to work as a web application and mobile application so that the needs of the consumers of this system are met regardless of the device they are using. The backend application handles all the integrations and the business logic. It also contains third-party APIs that connect the system to other frameworks and systems like ERP and CRM [7]. The backend connects to a database, which stores data related to the systems, such as the inventory level, customer details, and many more. Lastly, this system contains APIs that connect the backend to the front end to ensure seamless communication between the two architectures.

Frontend Architecture

1. ASP.NET core and Angular are used to develop the front end because both work well to ensure high performance in handling all system operations. ASP.NET core offers various security measures to protect the user's data. This technology also works seamlessly with other .NET frameworks, making it a recommended choice for development [8].
2. Angular is a robust framework for developing the web app UI because it offers data binding and various tools for building the application, testing it, and deploying it to consumers who need this technology. This framework is simple and easy to use, so it is recommended for building large applications.
3. Xamarin is used to develop mobile applications because it combines C# and .NET applications. This technology ensures cross-platform compatibility so all users' needs can be met. This technology is compelling because it uses native code, therefore giving the programmer a native experience in their interaction with the code [8]. It offers a single code base for various platforms, ensuring that there is no confusion when it comes to using the code.

UI Considerations

1. Responsiveness: The application should be able to resize depending on the size of the device being used. This can be achieved by using Bootstrap for the styling of the application. Media queries are very useful when designing an application; they help to ensure that the styling considerations fit the characteristics of the devices being used [8].

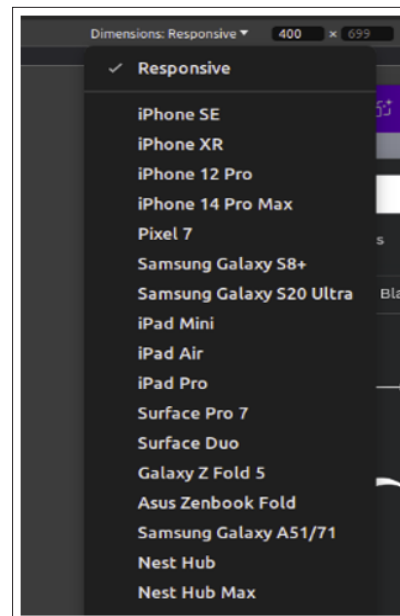


Figure 6: Shows the Different Dimensions of the Devices Used

2. Usability: The application should be simple and user-friendly; it should have straightforward navigation that requires minimal assistance from the user. The application should have features that will meet the needs of the users as well as other features that ensure the application is running smoothly.
3. Performance: both the web and mobile applications should have high performance in terms of the amount they take to load components and the overall user experience. This can be achieved through lazy loading, a technique that allows only loading the application's primary components; the other secondary components will only load when needed [8]. This can also be achieved through modulation, where the code is broken down into smaller components, allowing it to load independently to save time.

Backend Architecture

1. ASP.NET: this technology is used in developing RESTful APIs, which are useful in maintaining the state of the application and making it easy to integrate with other systems. The .NET framework performs routing services to map all the user requests to the correct action, ensuring smooth application navigation. Jason Web Tokens(JWT) are often used to handle the authentication of the users; this allows only authorized personnel to use the system [9].
2. Database access: The backend uses Entity Framework Core (EF core), an ORM that connects the backend to the database to ensure that the programmer can interact with the database to perform the Create, Read, Undo, Delete (CRUD) operations and create queries on the data.

Database Design

1. Schema design: Regarding ma design, various actions such as normalization. This ensures that there is no repetition in the database. The table in the database should be normalized to up to the 3 Normal Form (3NF). The tables in the database should be represented using Entity Relation (ER) diagrams so that the components of the table and their relationships between the diagrams can be seen clearly [10].
2. Data management: Constraints such as primary and foreign keys should be used to ensure data integrity in the database. A reliable backup plan should ensure that the data in the database is not lost in case of system failure or theft. Security measures, such as encryption, should be implemented to deter data theft. Archiving should be implemented to improve the database's navigation by hiding less necessary information.

Automation with Power Automate

Automation of the Order Management

1. Making an order: The customer can order an item to be delivered to them through the mobile or web application. The details of the order are sent by the ASP.NET core RESTful APIs to the backend to be processed, the stock level adjusted, and the order delivered to the customer [5].
2. Validation: Powe Automate triggers endpoints that handle order validation to check the customer's details to ensure they are correct. After the verification, a notification is sent to the customer, confirming if the order validation was successful.
3. Inventory management: Power automation triggers actions that check if the item the client has ordered is in stock; if not, the customer is notified through the application.
4. Confirmation: After the validation and inventory management are done, the customer is notified that the order placed has been confirmed. This is done through email or SMS.
5. Delivery: The customer's order is then sent for delivery, and the customer is updated on the status of the item to be delivered.

Automation of the Fleet Management

1. Real-time tracking: GPS tools should be placed in the delivery vehicle to monitor its movement and inform the customer if there are any delays.
2. Route Optimization: GPS models such as Google Maps should be integrated with Power Automate to determine the route to deliver items [5]. The route should be selected based on the shortest amount of time it takes to deliver the item and the road with good conditions.
3. Maintenance: Power Automate should notify the transportation and logistics company's management and drivers when the delivery vehicles are due for maintenance. It should also be able to schedule the maintenance of the vehicles automatically.

Automation of the Inventory Management

1. Stock level management: Power Automate should be able to manage stock levels and send real-time data to management to ensure no stockouts or overstocking.
2. Reports: The system should be able to provide a detailed report of all the orders made and suggest which items should be ordered more because they are in high demand. This report is crucial when it comes to demand forecasting because it gives the management an idea of which items to order more [5].
3. Stock replenishment: Power Automate should be able to trigger stock replenishment by automatically contacting the suppliers to deliver the items that need to be stocked.

Implementation of the System

Various steps are carried out in this phase. The development process used in this case is Agile methodology to ensure continuous system development.

1. Requirements gathering: This is the phase where the programmer interacts with the stakeholders and the management, discussing the needs that have to be met in the application and also getting an idea of what the application should look like [11].
2. Feasibility test: This ensures the application is feasible in terms of technology, cost, operation, and timeliness.
3. The architecture used: In this phase, the programmer and the stakeholders discuss the architecture that should be used in the application's development.
4. Design: The programmers create a conceptual design of the application showing the database, the UI, and the flow of processes. A prototype of a real-world system is then created to determine if the system is operational in the real world. Once the application is made, Testing Testing is done to see the areas that need to be improved.
5. System development: This phase involves the actual coding of the real-world system. During the development of the system, modularization is done to ensure the components can be run and tested independently. This is also done to ease the process of debugging because the errors can be easily identified [11].
6. Testing: The developed application undergoes various tests, such as unit testing, functional testing, integration testing, security testing, and performance testing. These tests debug the system and ensure it runs smoothly once it is deployed.
7. Implementation and maintenance: This phase involves deploying the application into the real world. The programmers are then tasked with training the users on how to use the system. The system should undergo continuous maintenance to ensure it is running smoothly and still meets the needs of its users.

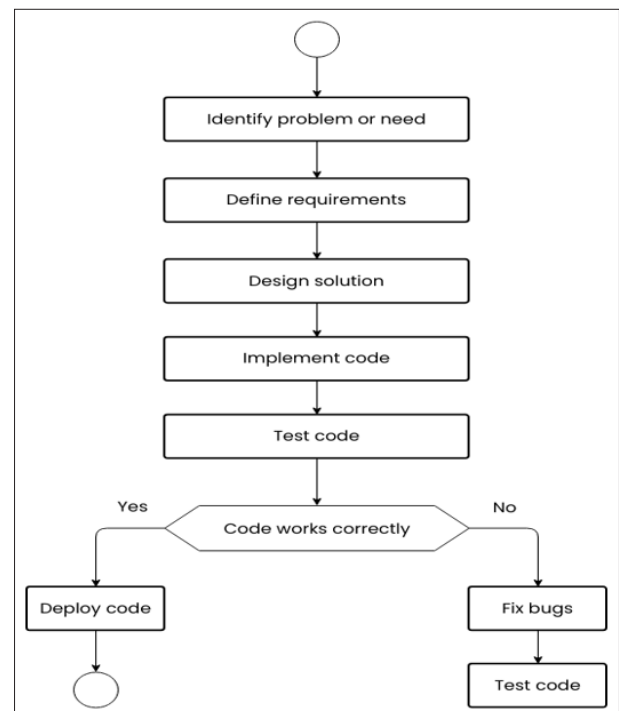


Figure 7: Shows the System Development Phases

Benefits of the System

Various benefits come with creating a power automated and .NET complete stack application, one of them being reduced operational costs since manual labor will not be needed. There will be an improvement in customer satisfaction because there will be fewer erroneous mistakes, and the process of placing an order will be fast and accurate. There will also be communication with the customer regarding the delivery of items; this will improve the customer's trust in the brand. The system will be able to handle processes where there is high demand because it is scalable and very flexible [12]. This system will help management with better demand forecasting strategies by providing a comprehensive report of orders made within a predefined period.

Conclusion

This paper comprehensively optimizes sports and logistics operations using Power Automate and .NET full-stack development. It analyzes the current challenges faced in order processing, fleet management, inventory management, and integration processes in the application created for the transport and logistics industry. The technical framework of power automation and .NET complete stack applications are then discussed, showing their capabilities and how they will help develop the full stack application. The paper then discusses the technology and architecture used to create the backend, frontend, and third-party APIs and systems. This paper further delves deeper into the development process of a power automate and .NET full-stack application and states the steps to be followed in developing the application. Lastly, this paper highlights the benefits of developing this system. Power automation and .NET full stack applications are potent tools that develop scalable, reliable, high-performing applications.

References

1. Chung SH (2021) Applications of intelligent technologies in logistics and transport: A review. *Transportation Research Part E: Logistics and Transportation Review* 153: 102455.
2. Toan TT (2023) Opportunities and challenges for the quality of human resources in the public sector of Vietnam's logistics industry. *International journal of public sector performance management* 10.
3. Liu W, Liang Y, Bao X, Qin J, Lim MK (2022) China's logistics development trends in the post-COVID-19 era. *International Journal of Logistics Research and Applications* 25: 965-976.
4. Mangla SK, Sharma YK, Patil PP, Yadav G, Xu J (2019) Logistics and distribution challenges to managing operations for corporate sustainability: a study on leading Indian dairy organizations. *Journal of Cleaner Production* 238: 117620.
5. Ray S, Villa A, Rashid N, Vincent P, Guttridge K, et al. (2021). Magic quadrant for robotic process automation <https://www.gartner.com/doc/reprints?id=1-26Q65VFT&ct=210706&st=sb>.
6. Almeida F (2018) Visual C# .NET: Console Applications and Windows Forms. *Tutorials Teacher*.
7. Meyers DC, Durlak JA, Wandersman A (2012) The quality implementation framework synthesizes critical steps in the implementation process. *American journal of community psychology* 50: 462-480.
8. Andersson V, Roll E (2020) Frontend study and application of modern web-app technologies to improve an existing system. Bachelor's thesis, NTNU.
9. Zilahi R (2023) Mobile App & Backend Integration for Streamlined Library Operations: Approach with Cutting-Edge Technology <https://www.theseus.fi/handle/10024/844459>.
10. Foster E, Godbole S (2022) Database systems: a pragmatic approach. Auerbach Publications <https://link.springer.com/book/10.1007/978-1-4842-1191-5>.
11. Lu Y, Xu X, Wang L (2020) Innovative manufacturing process and system automation—a critical review of the standards and envisioned scenarios. *Journal of Manufacturing Systems* 56: 312-325.
12. Chung SH (2021) Applications of intelligent technologies in logistics and transport: A review. *Transportation Research Part E: Logistics and Transportation Review* 153: 102455.