RSET
RAJAGIRI SCHOOL OF
ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

Project Phase I Report On

Advanced Supply and Trade Resource Optimisation

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science and Business Systems

By

Amel Chandra (U2109009)

Bharath S. (U2109018)

Joepaul Vilsan (U2109033)

Shane George Salphie (U2109063)

Under the guidance of

Dr. Nikhila T. Bhuvan

Department of Computer Science and Business Systems
Rajagiri School of Engineering & Technology (Autonomous)
(Parent University: APJ Abdul Kalam Technological University)
Rajagiri Valley, Kakkanad, Kochi, 682039
November 2024

CERTIFICATE

This is to certify that the project report entitled ***"Advanced Supply and Trade Resource Optimisation"*** *is a bonafide record of the work done by* ***Amel Chandra (U2109009), Bharath S (U2109018), Joepaul Vilsan (U2109033), Shane George Salphie (U2109063)****, submitted to the Rajagiri School of Engineering & Technology (RSET) (Autonomous) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in "Computer Science and Business Systems" during the academic year 2024-2025.*

Dr. Nikhila T. Bhuvan	Dr. Nikhila T. Bhuvan	Dr. Divya James
Project Guide	Project Co-ordinator	Head Of Department
Associate Professor	Associate Professor	Associate Professor
Dept. of CSBS	Dept. of CSBS	Dept. of CSBS
RSET	RSET	RSET

ACKNOWLEDGMENT

We wish to express our sincere gratitude towards **Rev. Dr. Jaison Paul Mulerikkal**, Principal of RSET, and **Dr. Divya James**, Head of the Department of Computer Science and Business Systems, for providing us with the opportunity to undertake our project, **"Advanced Supply and Trade Resource Optimisation"**.

We are highly indebted to our project coordinators, **Dr. Nikhila T. Bhuvan**, Associate Professor, Department of Computer Science and Business Systems, **Ms. Ancy C. A.**, Assistant Professor, Department of Computer Science and Business Systems, **Mr. Mahesh K. M.**, Assistant Professor, Department of Computer Science and Business Systems, for their valuable support.

It is indeed our pleasure and a moment of satisfaction for us to express our sincere gratitude to our project guide **Dr. Nikhila T. Bhuvan** for her patience and all the priceless advice and wisdom she has shared with us.

Last but not least, we would like to express our sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

Amel Chandra

Bharath S.

Joepaul Vilsan

Shane George Salphie

Abstract

Small-scale vendors face significant challenges in competing with large retail chains due to limited purchasing power, inefficient logistics, and restricted access to financial services. These barriers often hinder their ability to manage demand fluctuations and make informed business decisions, threatening their long-term sustainability. This project aims to address these issues by developing a comprehensive platform that empowers small vendors through collaborative purchasing, optimized logistics, and data-driven decision-making. By enabling vendors to combine orders, the platform facilitates bulk purchasing, unlocking competitive discounts and cost savings. Advanced algorithms optimize logistics by streamlining delivery routes, thereby reducing transportation costs and enhancing operational efficiency. These features empower vendors to make strategic decisions that align with market demands and their business goals. The ultimate goal is to level the playing field for small vendors, enabling them to compete effectively with larger retail chains while fostering sustainable growth in local economies. The deliverables of this project include a fully functional platform supporting vendor collaboration and an optimized logistics module to reduce operational costs. By promoting economic resilience, enhancing competitiveness, and driving sustainable development, this project seeks to strengthen the local vendor ecosystem and contribute to the overall development of communities.

Contents

List of Abbreviations

ASTRO - Advanced Supply and Trade Resource Optimisation

SDG - Sustainable Development Goals

ARIMA - Auto Regressive Integrated Moving Average

ACO - Ant Colony Optimisation

VAM - Vogel's Approximation Method

MACOA - Modified Ant Colony Optimisation Algorithm

MINLP - Modified Integer Non-Linear Programming

IMFO - Improved Moth-Flame Optimization

GA - Genetic Algorithm

PSO - Particle Swarm Optimization

GWO - Grey Wolf Optimizer

AHP - Analytic Hierarchy Process

MILP - Modified Integer Linear Programming

SME - Small to Medium-sized Enterprises

WOA - Whale Optimisation Algorithm

VRP - Vehicle Routing Problem

VRPTW - Vehicle Routing Problem with Time Windows

SIH - Solomn Insertion Heuristic

HGA - Hybrid Genetic Algorithm

CPLA - Co-operative Population Learning Algorithm

PITSH - Population-based Iterated Tabu Search Heuristic

API - Application Programming Interfaces

MOQ - Minimum Order Quantity

UML - Unified Modelling Language

OAS - Order Aggregation System

List of Figures

List of Tables

Chapter 1

Introduction

Small-scale vendors are essential contributors to local economies, offering unique products and personalized services that enrich communities. However, they face significant challenges competing against large retail chains, which benefit from greater purchasing power, efficient logistics, and easier access to financial resources. Due to limited resources, small vendors often struggle with high supply costs, fragmented logistics, and restricted access to credit, hindering their ability to grow and compete effectively. The ASTRO (Advanced Supply and Trade Resource Optimisation) platform addresses these challenges by providing a comprehensive solution that empowers small vendors through collaborative purchasing, logistics optimisation, and data-driven financial services. By allowing vendors to pool orders, ASTRO enables them to achieve bulk pricing similar to large retailers, reducing per-unit costs and enhancing profitability. The platform also incorporates advanced logistics optimisation algorithms, improving delivery efficiency and reducing operational expenses. Additionally, ASTRO offers financial services based on transaction data, enabling vendors to access credit and make informed decisions for business expansion.

ASTRO's functionality extends further with real-time demand forecasting tools, equipping vendors with predictive insights that help manage inventory and adapt to changing market demands. Through these features, ASTRO not only improves individual vendor competitiveness but also fosters economic resilience in local communities by supporting sustainable growth and operational efficiency. As a result, ASTRO is positioned as a vital resource for small vendors, helping them navigate a competitive retail landscape while contributing to the stability and diversity of local economies.

1.1 Background

Small-scale vendors are integral to the fabric of local economies, providing unique products, personalized services, and fostering vibrant community interactions that larger retail chains often cannot replicate. These vendors contribute significantly to economic diversity, cultural richness, and employment within their communities. However, despite their crucial role, small-scale vendors face persistent challenges that impede their ability to compete effectively with larger retail entities. These challenges include limited purchasing power, inefficient logistics, restricted access to financial services, and a lack of data-driven decision-making tools. In today's highly competitive retail landscape, large retail chains leverage economies of scale to negotiate bulk purchasing discounts, implement sophisticated logistics networks, and utilize advanced data analytics to optimize their operations and marketing strategies. In contrast, small-scale vendors typically operate with constrained resources, making it difficult for them to achieve similar efficiencies and cost-effectiveness. This disparity not only affects their profitability but also limits their capacity to respond to market fluctuations and evolving consumer demands. The ASTRO platform (Advanced Supply and Trade Resource Optimisation) is designed to address these disparities by providing small-scale vendors with tools and services that enhance their competitive edge. By enabling collaborative purchasing, optimizing logistics, and offering data-driven financial services, ASTRO empowers local vendors to achieve bulk pricing, streamline their operations, and make informed business decisions. This platform aims to bridge the gap between small vendors and large retail chains, fostering economic resilience and sustainable growth within local communities. Moreover, the advent of digital transformation has revolutionized various industries, including retail. The integration of technology into traditional business models is essential for small-scale vendors to remain relevant and competitive. ASTRO leverages modern technological advancements, such as machine learning algorithms for logistics optimisation and predictive analytics for demand forecasting, to deliver practical solutions tailored to the specific needs of small vendors. The significance of ASTRO extends beyond individual business success; it contributes to the overall economic health of communities by supporting the sustainability and growth of small-scale enterprises.

1.2 Problem Definition

Small-scale vendors face a significant challenge in today's competitive retail landscape, where large retail chains dominate with bulk pricing, advanced logistics, and extensive resources. This disparity puts small vendors at a disadvantage, as they often lack the economies of scale and infrastructure necessary to compete effectively. The problem is that, despite being crucial to local economies, small vendors struggle with high operational costs, inefficient logistics, and limited access to affordable credit, all of which hinder their ability to grow sustainably. This issue is exacerbated by the lack of platforms that cater to the specific needs of small-scale businesses, especially when it comes to aggregating demand, optimizing supply chains, and improving cash flow.

Without access to bulk purchasing and efficient logistics, small vendors pay higher prices for goods, limiting their profit margins and making it difficult to reinvest in their businesses. Furthermore, the absence of accurate demand forecasting and inventory management tools leads to overstocking or stockouts, reducing customer satisfaction and revenue. These issues underscore the need for a system that enables small vendors to pool their purchasing power, streamline their supply chains, and make data-driven decisions to improve their competitiveness.

The Advanced Supply and Trade Resource Optimisation (ASTRO) platform seeks to address this gap by providing small-scale vendors with the tools to compete with larger retail chains. ASTRO enables collaborative purchasing, allowing vendors to aggregate their orders and access bulk discounts, thus reducing product costs. Additionally, the platform offers real-time demand forecasting, helping vendors optimize inventory levels to avoid both overstock and stockouts.

By facilitating these services, ASTRO aims to foster economic resilience and sustainable growth in local communities. Its focus on empowering small vendors with advanced tools levels the playing field, promoting a more equitable and competitive market environment. This approach not only ensures that small vendors can thrive but also contributes to the overall health and diversity of the retail ecosystem.

1.3 Scope and Motivation

The ASTRO platform is a multifaceted solution designed to support small-scale vendors by addressing key operational challenges through technological integration and collaborative strategies. The scope of ASTRO encompasses the development and implementation of several core functionalities:

1. **Collaborative Purchasing:** ASTRO facilitates group purchasing among small vendors, allowing them to pool their orders to achieve bulk pricing discounts. This feature enhances purchasing power, enabling vendors to reduce their cost of goods sold and improve profit margins.

2. **Logistics Optimisation:** Utilizing advanced algorithms, ASTRO optimizes delivery routes to reduce transportation costs and improve delivery efficiency. By streamlining logistics, the platform helps vendors achieve timely deliveries and minimize operational expenses.

3. **Real-Time Demand Forecasting:** Leveraging predictive analytics, ASTRO provides real-time demand forecasting tools that help vendors anticipate market demand and manage inventory levels effectively. Accurate demand forecasting reduces the risk of overstocking or stockouts, ensuring that vendors can meet customer needs efficiently.

4. **User-Friendly Interface:** ASTRO is designed with an intuitive user interface that ensures ease of use for vendors with varying levels of technical expertise. This feature maximizes user engagement and ensures that vendors can effectively utilize the platform's functionalities to enhance their business operations.

The motivation behind ASTRO stems from the urgent need to empower small-scale vendors in an increasingly competitive retail environment. As large retail chains continue to dominate the market through superior purchasing power and advanced logistics, small vendors risk marginalization and potential business failure. ASTRO addresses these challenges by providing a robust platform that enhances operational efficiency, reduces costs, and increases access to financial resources, thereby enabling small vendors to compete on a more equal footing.

Additionally, ASTRO is motivated by the broader goal of fostering economic resilience and sustainability within local communities. Small-scale vendors are pivotal to the economic diversity and vitality of local markets, contributing to job creation and community development. By supporting these vendors, ASTRO not only enhances individual business success but also strengthens the overall economic fabric of communities, promoting long-term sustainable growth and economic stability.

1.4 Objectives

The ASTRO project is guided by a set of clear and actionable objectives aimed at addressing the challenges faced by small-scale vendors. These objectives are designed to enhance operational efficiency, reduce costs, and provide financial support, thereby empowering vendors to compete effectively in the marketplace. The primary objectives of ASTRO include:

1. **Enable Collaborative Purchasing:**
    - **Objective:** Develop a robust system that allows small vendors to combine their orders, thereby achieving bulk purchasing discounts.
    - **Outcome:** Increased purchasing power, reduced cost of goods sold, and improved profit margins for vendors.
2. **Optimize Logistics:**
    - **Objective:** Implement advanced route optimisation algorithms to streamline delivery processes, reduce transportation costs, and enhance operational efficiency.
    - **Outcome:** Cost-effective logistics operations, timely deliveries, and reduced operational expenses.
3. **Provide Real-Time Demand Forecasting:**
    - **Objective:** Integrate predictive analytics tools to offer real-time demand forecasting, enabling vendors to anticipate market demand and manage inventory levels effectively.

- **Outcome:** Minimization of overstocking and stockouts, improved inventory management, and enhanced ability to meet customer demands.

4. **Promote Economic Resilience and Sustainability:**

   - **Objective:** Foster economic resilience by providing small vendors with the tools and resources needed to sustain and grow their businesses.
   - **Outcome:** Long-term sustainability, increased competitiveness, and strengthened economic stability within local communities.

5. **Enhance User Experience and Accessibility:**

   - **Objective:** Design a user-friendly interface that ensures easy access to the platform's features, regardless of vendors' technical expertise.
   - **Outcome:** High user engagement, effective utilization of platform functionalities, and improved overall user satisfaction.

6. **Support Community Development:**

   - **Objective:** Encourage collaboration and mutual support among vendors, fostering a sense of community and shared growth.
   - **Outcome:** Collective bargaining power, knowledge sharing, and resource optimisation, leading to a stronger and more cohesive vendor network.

By achieving these objectives, ASTRO aims to create a comprehensive solution that empowers small-scale vendors to overcome their operational challenges, enhance their competitiveness, and contribute to the sustainable growth of local economies. Each objective is strategically aligned to address specific pain points, ensuring that the platform delivers tangible benefits and drives meaningful improvements in vendors' business operations.

1.5 Relevance

The relevance of the ASTRO project is multifaceted, addressing critical needs within the retail ecosystem and contributing to broader economic and social goals. Key aspects of ASTRO's relevance include:

1. **Economic Empowerment of Small Vendors:**

    - **Impact:** ASTRO directly addresses the economic challenges faced by small-scale vendors, enhancing their purchasing power and optimizing logistics cost. This empowerment enables vendors to operate more efficiently, reduce costs, and increase profitability, thereby improving their economic standing and sustainability.

2. **Enhancing Local Economies:**

    - **Impact:** By supporting small vendors, ASTRO contributes to the vitality and resilience of local economies. Small businesses are integral to economic diversification and job creation, promoting the long-term sustainability of local markets.

3. **Technological Advancement and Digital Transformation:**

    - **Impact:** ASTRO exemplifies the role of technology in transforming traditional business models. By leveraging advanced algorithms for logistics optimisation, predictive analytics for demand forecasting, ASTRO introduces modern technological solutions to enhance the operational capabilities of small vendors. This digital transformation is crucial for small businesses to remain competitive in an increasingly digital and data-driven marketplace.

4. **Sustainable Growth and Environmental Impact:**

    - **Impact:** ASTRO's logistics optimisation not only reduces operational costs but also minimizes the environmental footprint of delivery processes. By streamlining routes and reducing transportation distances, ASTRO contributes to more sustainable business practices, aligning with global efforts to reduce carbon emissions and promote environmental sustainability.

5. **Competitive Parity:**

    - **Impact:** ASTRO helps small vendors achieve competitive parity with larger retail chains by providing tools that enhance their operational efficiency and cost-effectiveness. This parity is crucial for maintaining market diversity and

preventing the monopolistic dominance of large retailers, ensuring a healthy and competitive marketplace that benefits both vendors and consumers.

6. **Adaptation to Market Trends:**

   - **Impact:** In a rapidly evolving retail environment, the ability to adapt to changing market trends and consumer behaviors is essential. ASTRO's real-time demand forecasting and data-driven insights enable small vendors to stay ahead of market trends, respond promptly to consumer needs, and adapt their inventory and marketing strategies accordingly.

7. **Community Building and Collaboration:**

   - **Impact:** ASTRO fosters a collaborative ecosystem where small vendors can work together to achieve common goals. This sense of community and mutual support enhances collective bargaining power, knowledge sharing, and resource optimisation, contributing to the overall strength and cohesion of the local vendor network.

8. **Scalability and Replicability:**

   - **Impact:** ASTRO's platform is designed to be scalable, allowing it to be replicated in various regions and adapted to different market conditions. This scalability ensures that the platform can benefit a wide range of small-scale vendors across different industries, promoting widespread economic empowerment and community resilience.

9. **Support for Economic Diversity and Stability:**

   - **Impact:** By enabling small vendors to thrive, ASTRO supports economic diversity, which is essential for the stability and resilience of local economies. Diverse economic activities reduce dependence on a single sector or large corporations, making communities more adaptable to economic shocks and changes.

10. **Promoting Entrepreneurship and Innovation:**

    - **Impact:** ASTRO encourages entrepreneurship by lowering the barriers to entry for small vendors. By providing the necessary tools and resources, ASTRO

enables aspiring entrepreneurs to start and grow their businesses, fostering a culture of innovation and economic dynamism within local communities.

11. **Enhanced Customer Experience:**

    - **Impact:** By improving operational efficiency and reducing costs, ASTRO enables small vendors to offer better pricing and more reliable services to their customers. Enhanced customer experiences lead to increased customer loyalty and satisfaction, driving repeat business and positive word-of-mouth referrals.

12. **Economic Resilience and Crisis Management:**

    - **Impact:** ASTRO equips small vendors with the tools to better manage economic fluctuations and crises. By optimizing inventory levels, forecasting demand accurately, and accessing financial support, vendors are better prepared to navigate economic downturns, supply chain disruptions, and other unforeseen challenges.

13. **Alignment with Sustainable Development Goals (SDGs):**

    - **Impact:** ASTRO aligns with several United Nations Sustainable Development Goals, including Decent Work and Economic Growth (SDG 8), Industry, Innovation, and Infrastructure (SDG 9), and Reduced Inequalities (SDG 10). By empowering small vendors, ASTRO contributes to inclusive and sustainable economic growth, technological innovation, and the reduction of economic disparities.

14. **Enhancing Vendor Autonomy and Empowerment:**

    - **Impact:** ASTRO empowers small vendors by giving them greater control over their purchasing, logistics, and financial decisions. This autonomy fosters a sense of ownership and confidence among vendors, encouraging them to take proactive steps towards business improvement and growth.

Chapter 2

Literature Survey

[1]Taylor S. J., Letham B. , "Forecasting at Scale". PeerJ 5 Preprints: e3190v2,2017

The authors present a flexible and robust forecasting tool designed to handle complex business time-series data with ease and accuracy. Developed primarily for non-experts, Prophet aims to overcome the limitations of traditional forecasting models like ARIMA and exponential smoothing, which often require extensive domain knowledge and struggle with irregular data patterns. The motivation behind Prophet is to create a scalable and intuitive model that can adapt to diverse business forecasting needs, such as demand prediction, financial planning, and inventory management.

The methodology behind Prophet is based on a decomposable time-series model comprising three core components: Trend, Seasonality, and Holiday Effects. The trend is modeled using either a piecewise linear or saturating growth function, allowing it to adapt to various growth behaviors. Seasonality is captured using Fourier series, which can handle multiple seasonal patterns effectively. Holiday effects account for custom, periodic events that impact the time series. Additionally, Prophet includes Automatic Changepoint Detection, which identifies shifts in trends using a sparse prior to ensure the model adjusts flexibly to changes in the data.

The model is highly interactive, allowing analysts to incorporate domain knowledge by tweaking trend changepoints, seasonality, and holiday components. This "analyst-in-the-loop" capability ensures that forecasts are not only data-driven but also aligned with business insights. Furthermore, Prophet's scalability enables it to handle large volumes of data, making it suitable for organizations with multiple time-series datasets.

Prophet was evaluated against traditional forecasting methods and demonstrated superior performance, especially in scenarios with complex seasonal patterns and irregularities. Its ability to handle missing data, outliers, and varying frequencies contributed to its

robustness. The model is also user-friendly, with parameters that are intuitive and easy to configure, making it accessible to both data scientists and business analysts without deep statistical expertise.

One of the significant contributions of this paper is its emphasis on flexibility and interpretability in forecasting. By providing a model that is both accurate and customizable, Prophet bridges the gap between sophisticated statistical tools and practical business applications. However, the authors acknowledge limitations such as the model's dependency on well-defined seasonalities and its difficulty in capturing highly non-linear patterns without additional input from the analyst.

The paper concludes by emphasizing Prophet's practical utility in real-world business environments. Future research directions include enhancing the model to handle multivariate time-series data and integrating it with machine learning techniques for greater predictive power. Overall, Prophet represents a significant advancement in time-series forecasting, providing a powerful yet accessible tool for data-driven decision-making.

[2]X. Wang and L. Wang, "Green Routing Optimization for Logistics Distribution with Path Flexibility and Service Time Window," 2018 15th International Conference on Service Systems and Service Management (ICSSSM), Hangzhou, China, 2018, pp. 1-5, doi: 10.1109/ICSSSM.2018.8464993.

The authors discuss the implementation of a time-dependent vehicle routing problem to optimize green logistics. The study emphasizes reducing fuel consumption and CO2 emissions by developing an algorithm that incorporates dynamic traffic conditions and customer service time windows. Unlike traditional routing models, the authors integrate real-world factors such as vehicle speed variations based on traffic, road conditions, and time of day, enabling a more practical application in urban logistics scenarios. The motivation for the research lies in enhancing sustainability by minimizing the carbon footprint of logistics operations while improving cost efficiency and customer satisfaction.

The methodology involves formulating an optimization model that minimizes both CO2 emissions and travel time. The routing problem includes constraints like vehicle capacity, service time windows, and varying traffic conditions. A unique feature of the model is its objective function, which calculates emissions based on vehicle speed, distance, and load. This approach incorporates a non-linear emission rate function that reflects the impact of speed variations on fuel consumption and emissions. Another innovative aspect

is the flexibility in path selection, allowing vehicles to choose optimal routes based on real-time traffic data.

To validate the model, the authors tested it on real-world logistics scenarios, comparing its performance against conventional methods. The results demonstrate significant reductions in both CO2 emissions and travel time, showcasing the effectiveness of considering dynamic traffic conditions. The model proved especially efficient in urban environments with high traffic variability, offering valuable insights for logistics managers seeking to balance cost savings with environmental responsibility.

One of the key contributions of this paper is its focus on sustainability in logistics. By addressing the dual objectives of environmental impact and operational efficiency, the study highlights the importance of adopting green logistics strategies in modern supply chains. The use of time-dependent speeds and flexible path selection represents a substantial improvement over static models, making the approach highly relevant for real-world applications. However, the authors also acknowledge limitations, such as the complexity of implementing such models at scale and the potential variability in traffic data accuracy.

The paper concludes by emphasizing the practical benefits of integrating environmental considerations into logistics planning. The proposed model not only achieves cost savings and emissions reductions but also aligns with the growing demand for sustainable business practices. Future research directions include extending the model to account for multi-modal transportation systems and exploring the integration of renewable energy-powered vehicles into the routing framework. Overall, this study provides a comprehensive solution for green logistics optimization, contributing significantly to the field of sustainable supply chain management.

[3]E. M. U. S. B. Ekanayake, S. P. C. Perera, W. B. Daundasekara, and Z. A. M. S. Juman, "A Modified Ant Colony Optimization Algorithm for Solving a Transportation Problem," Journal of Advances in Mathematics and Computer Science, vol. 35, no. 5, pp. 83-101, 2020

The authors present a novel approach to addressing logistics transportation issues using an enhanced version of the Ant Colony Optimization (ACO) algorithm. The study focuses on minimizing transportation costs for both balanced and unbalanced problems while improving computational efficiency. Unlike traditional methods such as Vogel's Approximation Method (VAM), the proposed Modified Ant Colony Optimization Algorithm

(MACOA) adapts pheromone updates and transition rules to provide superior solutions, particularly for large-scale logistics problems. The motivation behind the research stems from the increasing demand for cost-effective and time-efficient solutions in transportation logistics.

The methodology begins with converting unbalanced transportation problems into balanced ones by introducing dummy sources or destinations. Ants, which symbolize agents in the ACO framework, calculate probabilities based on a transition rule to allocate resources efficiently. The ants iteratively construct solutions by selecting routes with the highest probabilities, ensuring all demands and supplies are met. The pheromone update mechanism is customized to strengthen promising routes while discouraging suboptimal ones. By incorporating adjustments to the initialization and update phases, the algorithm enhances the traditional ACO approach, achieving faster convergence and improved solution quality.

The authors tested MACOA against benchmark transportation problems, comparing its performance to VAM and other established methods. Results indicate that MACOA consistently outperformed VAM in minimizing transportation costs while achieving comparable results to more computationally intensive techniques. Additionally, the algorithm demonstrated significant improvements in solving large-scale problems, offering time savings without compromising accuracy. The study highlights the algorithm's potential in real-world applications, particularly in industries where efficient logistics and supply chain management are critical.

The key contributions of this work include its ability to handle unbalanced transportation problems effectively and its computational efficiency for large datasets. By reducing the number of iterations needed to achieve optimal or near-optimal solutions, MACOA provides a practical alternative to traditional methods. However, the authors acknowledge that the algorithm, being heuristic, does not guarantee exact optimal solutions in all scenarios. They suggest future work to refine the method, aiming to balance efficiency with precision and scalability.

The study concludes that MACOA represents a significant advancement in transportation optimization, particularly for large-scale problems where computational resources are constrained. The proposed algorithm offers a blend of cost reduction and computational speed, making it an attractive option for logistics managers. Future research could explore

integrating MACOA with hybrid optimization techniques or adapting it to dynamic transportation scenarios involving real-time changes in supply and demand. Overall, the paper demonstrates the potential of modified ACO algorithms in addressing complex logistics challenges, contributing valuable insights to the field of transportation optimization.

[4]KAi, X., Yue, Y., Xu, H., and Deng, X. "Optimizing multi-supplier multi-item joint replenishment problem for non-instantaneous deteriorating items with quantity discounts," PLoS ONE, 16(2), e0246035, 2021

The authors address the challenges of managing inventory in supply chains dealing with perishable goods and multiple suppliers. The study focuses on creating a joint replenishment strategy to minimize costs while considering the perishability of items and supplier-specific quantity discounts. The motivation for this research arises from the inefficiencies faced by retailers when ordering independently, leading to higher costs and suboptimal inventory management. The proposed model seeks to balance replenishment timing and order quantities to reduce overall supply chain costs.

The methodology employs a Mixed-Integer Nonlinear Programming (MINLP) model to optimize total costs, including ordering, holding, and deterioration costs. The authors consider factors such as demand rate, preservation period, and deterioration rate for each item, along with supplier-specific discount schemes. To solve this complex optimization problem, the study uses an Improved Moth-Flame Optimization (IMFO) algorithm. This metaheuristic method enhances traditional optimization by avoiding local optima through techniques like hyperbolic spiral functions and Levy-flight strategies. The IMFO is designed to achieve faster convergence and higher accuracy compared to other algorithms such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Grey Wolf Optimizer (GWO).

The results highlight the effectiveness of the IMFO in solving joint replenishment problems. The algorithm demonstrated cost savings of 0.01% to 1.35% compared to traditional optimization methods, along with faster convergence rates. Sensitivity analysis revealed that demand rates significantly influence total costs, while supplier selection remained consistent across varying parameters. The model effectively optimized replenishment schedules for deteriorating items, ensuring cost-efficiency and better inventory turnover.

A key contribution of the study is its ability to handle multi-supplier and multi-

item scenarios with non-instantaneous deteriorating goods. By incorporating supplier-specific quantity discounts, the model provides a practical solution for retailers aiming to leverage economies of scale. Additionally, the use of IMFO ensures computational efficiency, making the approach suitable for large-scale problems. However, the authors acknowledge certain limitations, including the reliance on static demand rates and the challenges of applying the model to dynamic environments with real-time data.

The paper concludes by emphasizing the applicability of the proposed methodology to supply chains dealing with perishable goods. Future research could focus on integrating real-time data and dynamic demand forecasting into the optimization framework or exploring hybrid approaches to further enhance solution accuracy. Overall, the study provides valuable insights into joint replenishment strategies, contributing to the optimization of inventory management in complex supply chain systems.

[5]O. Triebe, H. Hewamalage, P. Pilyugina, N. Laptev, C. Bergmeir, and R. Rajagopal, "NeuralProphet: Explainable Forecasting at Scale" , arXiv preprint, Nov. 2021

The authors introduce an enhanced time-series forecasting model that builds upon Facebook's Prophet framework. NeuralProphet integrates neural networks and traditional statistical methods to improve accuracy, scalability, and explainability. The motivation behind this research is to address the limitations of traditional models like ARIMA and Prophet, which often struggle with capturing local dynamics, non-linear trends, and interactions in complex time-series data. NeuralProphet aims to bridge the gap by offering a more flexible, robust, and interpretable solution suitable for large-scale forecasting tasks.

The methodology centers on a modular model structure that decomposes time-series data into components, such as trend, seasonality, event effects, and auto-regression. Each component is modeled independently using both classical techniques and neural network-based extensions. The model's key innovation lies in its hybrid nature, combining additive and multiplicative effects for better flexibility. Features like auto-regression and lagged regressors enable the model to capture local patterns and dependencies in the data. The model also incorporates a deep-learning-powered changepoint detection mechanism to identify both abrupt and gradual shifts in trends effectively.

NeuralProphet was tested across diverse datasets and compared with other forecasting models, including ARIMA, Prophet, and modern neural network-based methods. The re-

sults showed significant improvements in accuracy, with NeuralProphet outperforming its predecessors by 55–92% on various metrics. The model demonstrated robustness in handling high-frequency and multi-seasonal data while retaining scalability for large datasets. Its explainability features allowed for a detailed understanding of each component's contribution to the forecast, enhancing its usability for analysts.

The study's main contribution lies in providing a forecasting tool that balances flexibility, accuracy, and interpretability. By combining the strengths of traditional and neural network-based approaches, NeuralProphet offers a practical solution for complex time-series forecasting tasks. Its ability to decompose forecasts into understandable components makes it particularly valuable for non-experts and domain specialists. However, the authors acknowledge that the model's performance depends on proper configuration and that further improvements in scalability and real-time forecasting capabilities are needed.

The paper concludes by emphasizing the potential of NeuralProphet in business and industry applications, particularly in scenarios requiring interpretable and scalable forecasting solutions. Future research could explore extending the model's capabilities to multi-variate and hierarchical time-series data or integrating external factors such as economic indicators. Overall, the study highlights NeuralProphet as a significant advancement in time-series forecasting, offering a powerful and user-friendly alternative to existing models.

[6] Ben Jouida, S., and Krichen, S. (2020). A genetic algorithm for supplier selection problem under collaboration opportunities. Journal of Experimental and Theoretical Artificial Intelligence, 34(1), 53–79.

The authors propose a framework to optimize order distribution and demand sharing across supply chain participants. The study addresses challenges such as limited information sharing, unequal benefit distribution, and inefficient coordination in collaborative supply chains. By leveraging a heuristic approach based on multi-criteria optimization, the authors aim to enhance collaboration, reduce inefficiencies, and achieve equitable decision-making among supply chain stakeholders. The motivation stems from the need to support demand-driven environments where real-time coordination is essential for supply chain effectiveness.

The methodology revolves around a central coordination system designed to manage

demand allocation and order distribution across a three-tier supply chain. The framework integrates the Analytic Hierarchy Process (AHP) to prioritize criteria influencing demand allocation and employs Genetic Algorithms (GA) for iterative optimization of order distribution. Collaboration rules are predefined to ensure equitable demand sharing and fairness among participants. The GA evaluates solutions based on criteria like cost efficiency, resource utilization, and adherence to the collaboration rules, refining them through evolutionary operations such as selection, crossover, and mutation.

The results demonstrate the framework's effectiveness in improving supply chain coordination and achieving balanced demand allocation. Using the heuristic approach, the system successfully optimized order distribution, leading to significant cost reductions and enhanced operational efficiency. The methodology's scalability was validated in real-world scenarios, proving its adaptability to various supply chain structures and demands. Additionally, the integration of AHP ensured that critical factors, such as fairness and resource prioritization, were consistently considered in the optimization process.

A key contribution of the paper is its demonstration of how heuristic methods can overcome the complexities of demand-driven collaborative supply chains. By emphasizing equitable benefit distribution and operational efficiency, the proposed framework fosters sustained cooperation among supply chain participants. However, the authors acknowledge certain limitations, such as the dependence on predefined collaboration rules and the computational demands of genetic algorithms for large-scale implementations.

The study concludes by highlighting the potential of heuristic methodologies to transform collaborative supply chains. The proposed framework offers a practical and scalable solution for real-world applications, particularly in environments requiring high levels of coordination and fairness. Future research directions include exploring the integration of real-time data analytics into the framework and expanding the model to accommodate dynamic changes in supply chain demands. Overall, the paper provides valuable insights into optimizing collaborative supply chains, contributing significantly to the field of demand-driven supply chain management.

[7]LSeifbarghy, M., Shoeib, M., and Pishva, D. "Coordination of a single-supplier multi-retailer supply chain via joint ordering policy considering incentives for retailers and utilizing economies of scale," International Journal of Production Economics, 151, 49–59, 2022

The authors of present a collaborative framework to optimize inventory management and reduce costs in supply chains involving small retailers. The research addresses challenges such as high individual ordering costs, inefficiencies in inventory management, and missed opportunities for economies of scale. By introducing a joint ordering policy, the study aims to enhance coordination between retailers and suppliers, leveraging bulk purchasing benefits while maintaining equitable cost-sharing.

The methodology focuses on developing a Mathematical Optimization Model to determine optimal joint ordering schedules and quantities. The model incorporates Mixed-Integer Linear Programming (MILP) to minimize total costs, including ordering, holding, and shortage costs, while considering constraints like demand rates, lead times, and supplier capacities. A central feature of the framework is its Incentive Mechanism, which distributes cost savings among retailers based on their contribution to the joint order. This encourages collaboration by ensuring fair benefit-sharing. The model also accounts for economies of scale by synchronizing ordering periods, enabling bulk discounts from suppliers and reducing logistics costs.

The results demonstrate that the joint ordering policy significantly reduces overall supply chain costs. Retailers benefit from lower procurement expenses due to volume discounts, while suppliers gain from streamlined production and delivery schedules. The proposed incentive mechanism effectively fosters collaboration, ensuring sustained participation from all retailers. Furthermore, the model improves inventory turnover rates and minimizes stockouts, enhancing supply chain efficiency and responsiveness.

One of the key contributions of this paper is its practical applicability for small and medium-sized enterprises (SMEs). The framework provides a scalable solution that enables smaller retailers to compete with larger counterparts by pooling resources and optimizing procurement strategies. However, the study notes limitations such as the reliance on static demand forecasts and the need for robust information sharing among participants for successful implementation.

The authors conclude by emphasizing the importance of collaborative strategies in modern supply chains. Future research could focus on incorporating dynamic demand forecasting and real-time data analytics to enhance the model's adaptability. Additionally, exploring applications in multi-tier supply chains or integrating sustainability metrics could further expand its utility. Overall, the study offers a significant contribution to sup-

ply chain optimization, providing a roadmap for achieving cost-efficiency and collaboration in single-supplier, multi-retailer systems.

[8]J. Zhou, "Research on Multi-objective Vehicle Path Optimization Based on Whale Algorithm," 2023 International Conference on Networking, Informatics and Computing (ICNETIC), Palermo, Italy, 2023

The authors of "Multi-objective Vehicle Path Optimization Based on Whale Algorithm" explore the use of a Modified Whale Optimization Algorithm (WOA) to address vehicle routing challenges by balancing cost, delivery efficiency, and vendor collaboration. This research focuses on optimizing logistics operations by improving delivery efficiency and ensuring coordination among vendors. The motivation behind this study stems from the growing demand for efficient logistics solutions that minimize operational costs while addressing the complexities of multi-vendor systems.

The methodology leverages the biological behavior of whales, particularly their hunting strategies, to solve multi-objective optimization problems. The algorithm employs two primary strategies: the Encircling Prey Strategy, which adjusts the whale's position based on proximity to the best-known solution, and the Spiral Movement Strategy, which simulates a helical path around the optimal solution. These strategies enable the algorithm to balance exploration (searching new solutions) and exploitation (refining current solutions). The authors further adapt the WOA for multi-objective scenarios by introducing parameters that optimize delivery cost and efficiency simultaneously.

The algorithm's performance was evaluated using standard benchmarks and compared with single-objective optimization methods. Results showed that the Modified WOA achieved superior outcomes, demonstrating reduced operational costs and faster convergence to optimal solutions. The multi-objective approach allowed the algorithm to account for trade-offs between competing goals, such as minimizing costs while ensuring timely deliveries. Additionally, the adaptive mechanisms of WOA improved its ability to handle diverse logistics scenarios, including those with dynamic constraints like fluctuating customer demands and delivery time windows.

One of the key contributions of the paper is the demonstration of WOA's applicability to real-world logistics challenges. Unlike traditional optimization techniques, which often focus on single objectives, this algorithm provides a comprehensive solution that addresses multiple logistics goals simultaneously. The study also highlights the algorithm's scala-

bility and computational efficiency, making it suitable for large-scale problems in supply chain management. However, the authors note that while WOA effectively balances multiple objectives, its performance may be influenced by parameter tuning, necessitating further exploration to improve robustness.

The paper concludes by emphasizing the practical benefits of using Modified WOA for vehicle routing problems. The algorithm's ability to optimize cost, efficiency, and collaboration offers a competitive edge in logistics management. Future work could focus on integrating WOA with other optimization methods, such as hybrid algorithms, or applying it to emerging areas like drone-based logistics and autonomous vehicle routing. Overall, the study showcases the potential of bio-inspired algorithms in solving complex multi-objective problems, contributing to advancements in efficient and sustainable logistics operations.

[9]Mamoun, K.A., Hammadi, L., Ballouti, A.E., Novaes, A.G.N.. and Cursi, E.S.D. Vehicle Routing Optimization Algorithms for Pharmaceutical Supply Chain: A Systematic Comparison. Transport and Telecommunication Journal, Sciendo, Vol. 25 (Issue 2), (2024) pp. 161-173.

The authors of "Vehicle Routing Optimization Algorithms for Pharmaceutical Supply Chain: A Systematic Comparison" conduct a comprehensive evaluation of multiple algorithms to address the Vehicle Routing Problem (VRP) within the pharmaceutical supply chain. The focus of the study lies in comparing the efficiency and scalability of various optimization methods, including Branch and Cut, Clarke and Wright's Savings Algorithm, Tabu Search, and Simulated Annealing, to minimize delivery costs and improve logistical performance. The motivation stems from the critical nature of pharmaceutical supply chains, where timely and cost-effective deliveries are essential for ensuring consistent medication availability.

The methodology involves analyzing the input parameters of VRP, such as customer locations, delivery demands, vehicle capacity, and distance matrices, to evaluate the performance of the selected algorithms. Branch and Cut is employed as an exact algorithm to generate optimal solutions for smaller problems. Meanwhile, heuristic methods like Clarke and Wright's Savings Algorithm and metaheuristic approaches such as Tabu Search and Simulated Annealing are applied to solve larger and more complex problems. The study assesses the algorithms' performance based on metrics like computational time, cost min-

imization, and solution accuracy.

The results highlight the strengths and weaknesses of the various algorithms. Branch and Cut delivers optimal solutions with a minimal gap percentage but struggles with scalability due to its computational intensity. Clarke and Wright's heuristic is fast and simple but less effective for large-scale problems. In contrast, metaheuristic methods, such as Tabu Search and Simulated Annealing, efficiently handle complex scenarios, achieving near-optimal solutions while maintaining computational feasibility. The study demonstrates that Tabu Search, in particular, provides a balanced trade-off between solution quality and scalability, making it well-suited for large-scale pharmaceutical supply chains.

One of the key contributions of this paper is its systematic comparison of different algorithms, offering valuable insights into their applicability for specific scenarios within the pharmaceutical sector. The research underscores the importance of metaheuristic methods in addressing large and complex VRPs, where exact algorithms may fall short due to their computational demands. However, the study also acknowledges the limitations of metaheuristics, such as the need for parameter tuning and the potential for local optima.

The authors conclude by emphasizing the practical relevance of their findings. The choice of an algorithm should align with the specific requirements of the supply chain, such as problem size and the need for precision. Future research could explore hybrid approaches that combine the strengths of exact and heuristic methods or investigate real-time optimization techniques to address dynamic routing problems. Overall, this study makes a significant contribution to optimizing pharmaceutical supply chains by providing a detailed analysis of algorithmic performance, offering a guide for practitioners in selecting the most suitable routing solution.

[10] A. Maroof, B. Ayvaz and K. Naeem, "Logistics Optimization Using Hybrid Genetic Algorithm (HGA): A Solution to the Vehicle Routing Problem With Time Windows (VRPTW)," IEEE Access, vol. 12, 2024

The authors explore the use of an enhanced Genetic Algorithm (GA) to solve the Vehicle Routing Problem with Time Windows (VRPTW). This study introduces a hybrid approach combining GA with Solomon Insertion Heuristic (SIH) to optimize logistics operations by minimizing route distances and improving vehicle utilization. The motivation stems from the complexity of VRPTW, where traditional methods struggle to balance constraints like delivery time windows, cost-efficiency, and vehicle capacity. The authors

aim to develop a solution that is computationally efficient while ensuring optimal logistics performance.

The methodology begins with population initialization, where a diverse set of high-quality initial routes is generated using SIH. This step ensures that the initial solutions are closer to optimal. The algorithm then iteratively refines the routes through GA operations, including selection, crossover, and mutation. Fitness evaluation is a key step where routes are assessed based on criteria such as travel distance, adherence to delivery time windows, and overall cost. The hybridization with SIH enhances the algorithm's capability to explore the solution space effectively, allowing for faster convergence to optimal or near-optimal solutions.

The performance of the Hybrid Genetic Algorithm was benchmarked against other algorithms, including CPLA and PITSH. Results demonstrated that HGA consistently outperformed its counterparts, achieving better route optimization and reduced computational time. The integration of SIH proved particularly effective in generating initial solutions that required fewer iterations to refine. Additionally, HGA showed scalability, maintaining its efficiency and accuracy even as the problem size increased.

One of the paper's significant contributions is the development of a practical solution tailored for small and medium-sized logistics operations. By combining the exploratory power of GA with the problem-specific refinement capabilities of SIH, the authors present a method that is not only robust but also adaptable to various real-world scenarios. However, the study also notes certain limitations, such as the reliance on fine-tuning algorithm parameters and the potential for suboptimal solutions in highly dynamic environments.

The paper concludes by highlighting the potential of HGA as a versatile tool for solving complex logistics problems. The hybrid approach offers an effective balance between cost efficiency and computational feasibility, making it well-suited for small-scale vendors and localized supply chains. Future research could explore extending this approach to dynamic VRPTW scenarios or integrating it with other metaheuristic methods to improve adaptability. Overall, the study provides a substantial contribution to logistics optimization, demonstrating the effectiveness of hybrid algorithms in enhancing operational efficiency.

Chapter 3

System Architecture

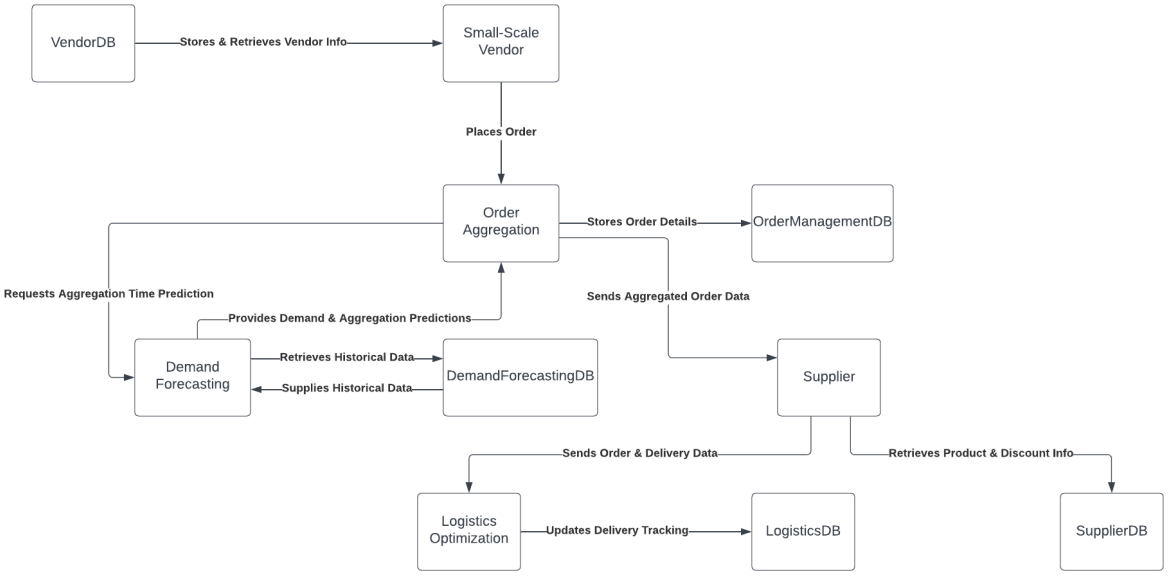


Figure 3.1: System Architecture

Figure 3.1 illustrates the system architecture for the ASTRO Platform, designed to support small-scale vendors by facilitating bulk order discounts, demand forecasting, and logistics optimisation. This architecture provides a clear overview of the platform's components and how they work together to streamline the entire process from order initiation to optimized delivery. Here's a breakdown of the key aspects and importance of this architecture:

1. **High-Level Overview:** The architecture serves as a roadmap, showing the main stages of the platform's workflow. From initial product viewing and order placement by vendors, through demand forecasting, order aggregation, and logistics optimisation, this high-level overview provides all stakeholders—including vendors, suppliers, and platform administrators—with a clear understanding of the system's core

functionality.

2. **Clarifies Data Flow:** The architecture clarifies the flow of data within the system, beginning with the acquisition of order and sales data. This data undergoes preprocessing and feature extraction, followed by demand forecasting with the Neural Prophet model. The forecast data then flows into the Order Aggregation module, which combines vendor orders to meet bulk purchase thresholds. Finally, optimized delivery routes are created through the Logistics Optimisation module, using Green Routing techniques. This sequential data flow enhances system transparency and makes it easy to identify potential bottlenecks.

3. **Enables Collaborative Order Aggregation:** The architecture highlights how order aggregation enables small-scale vendors to achieve bulk discounts. By aggregating orders from multiple vendors, the platform ensures Minimum Order Quantities (MOQs) are met, enabling vendors to access bulk discounts typically reserved for larger chains. This aspect of the system is crucial for enhancing the competitiveness of small vendors.

4. **Supports Logistics Optimisation through Green Routing:** The architecture emphasizes the logistics optimisation approach used in the platform. The Green Routing component incorporates path flexibility and service time windows to create efficient and environmentally friendly delivery routes. This optimisation minimizes transportation costs and enhances delivery reliability for vendors, contributing to a streamlined and cost-effective supply chain.

5. **Enables Real-Time Notifications and Updates:** The system architecture includes a notification mechanism that provides real-time updates to vendors on order status, inventory levels, and logistics. These notifications foster transparency, enabling vendors to make informed decisions about inventory and delivery schedules.

6. **Facilitates Collaboration and Communication Among Stakeholders:** The system architecture acts as a shared framework, providing a common language for platform users, developers, and suppliers. It allows stakeholders to better understand the interactions between components and collaborate effectively toward im-

proving platform functionality. This collaborative framework is essential for ensuring smooth operations and adapting to vendor needs.

Overall, the system architecture is crucial for understanding the ASTRO Platform. It provides a clear visual representation of the platform's workflow, highlighting key features such as order aggregation, demand forecasting, logistics optimisation, and real-time notifications. These components work together to enhance vendor competitiveness by offering cost-saving opportunities, efficient delivery solutions, and improved collaboration with suppliers.

3.1 Dataset

Purpose: This dataset supports research in sales forecasting, specifically at the item and store levels. The objective is to predict the next three months of sales for individual items at various store locations, applicable for demand forecasting, inventory management, and sales optimisation.

Data Collection: This dataset was collected from multiple store locations, representing real-world sales data with variations in store performance and item demand. It provides diverse daily sales records for individual items across different stores, enabling demand pattern analysis.

Data Fields:

date: Date of each sale record. There are no adjustments for holidays or store closures, simplifying the analysis but possibly requiring external data for event-based effects.

store: A unique identifier for each store, allowing for store-level analysis and capturing unique characteristics and item demand differences.

item: A unique identifier for each item sold, enabling item-level demand forecasting and cross-store analysis.

sales: The number of items sold at a particular store on a given date, which is the target variable for forecasting models.

File Descriptions:

train.csv: Contains historical sales data used for training forecasting models.

test.csv: Provides the test data where future sales predictions are needed, with a time-based split for realistic scenario testing.

sample_submission.csv: A sample submission file showing the required format for submitting sales forecasts.

Usage in Research: This dataset supports the development and validation of time series forecasting models. It allows researchers to explore seasonality, trends, and external influences on sales. Models like Neural Prophet or ARIMA can be applied, along with machine learning techniques for enhanced forecasting accuracy.

Challenges and Opportunities: Although the dataset lacks explicit holiday or store event information, researchers may incorporate external data to improve forecasts. This provides an opportunity to explore techniques that account for seasonality and trend modeling, such as Fourier series for seasonality or external regressors.

Access: This dataset is easily accessible, enabling collaborative studies on sales forecasting and providing a resource for testing and improving demand forecasting algorithms.

3.2 Preprocessing Steps

To ensure the data is prepared for accurate demand forecasting and that the model interprets the features effectively, the following preprocessing steps are applied:

Tasks

Handling Missing Values: Missing data may exist in historical datasets due to reporting errors or system downtime. The Neural Prophet model employs a data imputation mechanism to avoid excessive data loss when working with incomplete data. Missing events are filled in with zeros, indicating their absence. For other real-valued variables, including the time series itself (when autoregression is enabled), the following imputation procedure is applied:

1. **First Step:** Missing values are approximated using bi-directional linear interpolation. The missing values are filled by interpolating between the last known value before and after the missing data. This process is applied to a maximum of 5 missing values in each direction.

2. **Second Step:** Remaining missing values are imputed using a centered rolling average. A window of 30 is used, and up to 20 consecutive missing values can be filled.

3. **Third Step:** If more than 30 consecutive missing values remain, these data points are dropped from the dataset.

This imputation process ensures minimal loss of data while maintaining the integrity of the time series.

Scaling and Normalization: To improve model efficiency, numerical features, particularly sales data, are normalized. The Neural Prophet model offers various normalization options, with the default being the soft method. This approach scales the minimum value to 0.0 and the 95th quantile to 1.0, ensuring that the time series values are appropriately transformed for the model.

The formula for soft normalization is as follows:

$$x' = \frac{x - \min(x)}{Q_{95} - \min(x)}$$

where Q_{95} represents the 95th quantile of the time series values.

Time-Based Feature Engineering: To capture temporal and seasonal patterns in the data, key features like Day of the Week, Month, and a binary IsWeekend indicator are created. These features help the model understand weekly sales patterns, broader seasonal trends (e.g., holiday spikes), and the impact of weekends on sales. The IsWeekend feature is binary, set to 1 if the date is a Saturday or Sunday and 0 otherwise.

Lagged Sales Features: Since past sales are directly predictive of future demand, lagged sales features are created for the last 7 and 30 days to capture both short-term and longer-term patterns. Additionally, rolling statistics such as the rolling mean and rolling standard deviation over the past 30 days are computed. These features smooth out fluctuations and help the model capture trends more accurately. The rolling mean is calculated as:

$$\text{Rolling Mean}_{30} = \frac{1}{30} \sum_{i=t-30}^{t-1} \text{Sales}(i)$$

and the rolling standard deviation as:

$$\text{Rolling Std}_{30} = \sqrt{\frac{1}{30} \sum_{i=t-30}^{t-1} (\text{Sales}(i) - \text{Rolling Mean}_{30})^2}$$

Output

The result of these preprocessing steps is a clean, normalized dataset with imputed missing values and encoded categorical features. The dataset is now ready for model training, ensuring that the features are aligned and consistent, helping the model make accurate demand forecasts.

3.3 Feature Extraction

Feature extraction is a critical process in the ASTRO platform as it transforms raw sales data into meaningful representations, enabling accurate demand forecasting and efficient operational management. By identifying and engineering relevant features from the dataset, we capture essential patterns, trends, and relationships that enhance the performance of predictive models.

In the context of the dataset, which contains store-level and item-level sales information over time, feature extraction is designed to uncover temporal patterns, historical dependencies, and interactions between stores and items. The extracted features are categorized into two major types: time-based features and lagged variables, both of which are integral to understanding and predicting demand fluctuations.

3.3.1 Time-Based Features

Time-based features play a crucial role in capturing cyclical and seasonal patterns inherent in sales data. By decomposing the date field into granular components, these features allow the model to understand how sales vary across different times of the week, month, and year. For instance, the day of the week feature represents weekdays as integers, enabling the model to differentiate between regular weekday and weekend shopping behaviors. Similarly, the month feature identifies seasonal effects that often influence demand, such as increased sales during holiday months or specific seasonal trends for particular items.

Another important time-based feature is the week of the year, which highlights demand patterns recurring annually, such as back-to-school shopping or end-of-year festive sales. Additionally, a binary is weekend feature isolates the typically higher weekend sales from weekday trends, ensuring that the model appropriately weighs these differences when forecasting.

These features collectively form the temporal backbone of the forecasting model, allowing it to detect repeating cycles and shifts in demand driven by the calendar.

3.3.2 Lagged Variables

Lagged variables focus on incorporating historical sales data to enhance the model's ability to forecast future demand. These features represent the past performance of sales, which is often predictive of future trends. For example, a 1-day lag feature captures the sales volume from the previous day, providing immediate short-term context for the current demand. Similarly, a 7-day lag feature reflects weekly demand patterns by showing the sales from the same day of the prior week, which is especially useful for capturing weekly seasonality.

To understand longer-term patterns, a 30-day lag feature is introduced, representing sales from one month earlier. This helps in identifying recurring monthly trends or evaluating the effectiveness of past promotions. In addition to specific lags, rolling statistical features such as the rolling average (7-day) and rolling standard deviation (7-day) are calculated. These features smooth short-term fluctuations and provide insights into the stability or variability of demand over a week.

A cumulative sales feature further extends this approach by summing all sales up to the current date for a given store-item combination. This feature captures overall sales momentum and growth trends, offering a long-term perspective on performance.

Lagged variables are particularly effective in making the model aware of historical dependencies, enabling it to account for demand inertia, recent trends, and periodic changes.

The combination of time-based features and lagged variables provides a comprehensive representation of the dataset, balancing temporal patterns with historical sales behavior. These features enable the demand forecasting models to detect and learn from intricate relationships within the data, ensuring robust predictions and informed decision-making for vendors. Through this customized feature extraction approach, the platform empowers

small-scale vendors with insights that drive collaborative efficiency and competitiveness.

3.4 Project Modules and Their Integration

ASTRO platform comprises three core modules: Demand Forecasting, Order Aggregation, and Path Optimisation. These modules work in harmony to streamline vendor operations, optimize costs, and enhance efficiency in the supply chain. By integrating these modules, the platform empowers vendors to achieve bulk order discounts, coordinate deliveries, and reduce logistics expenses, creating a collaborative ecosystem that benefits all participants.

3.4.1 Demand Forecasting Module

The Demand Forecasting Module lies at the heart of the platform, enabling vendors to predict future sales and order cycles. By analyzing historical sales data, the module leverages the Neural Prophet model to identify trends, seasonal patterns, and demand fluctuations. The predictions provide vendors with actionable insights into when and how much to order, ensuring they maintain optimal inventory levels while avoiding overstocking or stockouts.

Additionally, this module facilitates order cycle prediction, which is critical for synchronizing vendor orders. When a vendor places an order with a supplier, the platform uses the demand forecasting output to identify similar demand cycles for other vendors. This allows the platform to notify these vendors about the opportunity to join the order, achieving the Minimum Order Quantity (MOQ) threshold required for bulk discounts. By doing so, the demand forecasting module not only supports individual vendors but also sets the foundation for collective collaboration.

3.4.2 Order Aggregation Module

The Order Aggregation Module builds on the insights provided by the demand forecasting module. Once multiple vendors agree to participate in a joint order, this module combines their requirements into a single bulk order. The process ensures that the MOQ thresholds are met, unlocking discounts from suppliers that would be unattainable for individual vendors.

The aggregation process also categorizes and organizes the joint order to ensure that

products are distributed efficiently. Each vendor's share of the bulk order is clearly delineated, and the aggregated data serves as input for the subsequent logistics operations.

Notifications play a key role in this module: vendors are informed about the opportunity to join an order and, once confirmed, receive updates about the order status and expected delivery timelines. The module ensures transparency and coordination among vendors, simplifying the process of collective purchasing.

3.4.3 Path Optimisation Module

The Path Optimisation Module is responsible for ensuring efficient delivery of the aggregated orders. After the order is finalized and fulfilled by the supplier, this module uses Green Route Optimisation algorithms to design an optimal delivery route.

Since the joint order often involves multiple vendors located in different areas, the path optimisation module focuses on clustering delivery locations and minimizing logistics costs. The Path Flexibility and Service Time Window feature ensures that deliveries are scheduled efficiently while adhering to time constraints.

The module optimizes the route for a single vehicle tasked with delivering products to multiple vendors. Factors such as load capacity, distance between vendors, and delivery time windows are considered to achieve the most cost-effective and eco-friendly route. By reducing unnecessary travel and fuel consumption, this module contributes to both financial savings and environmental sustainability.

3.4.4 Integration of Modules

The three modules—demand forecasting, order aggregation, and path optimization—work together in a seamless, integrated manner to ensure efficient operations:

- **Demand Forecasting to Order Aggregation**: The demand forecasting module predicts when vendors are likely to need specific products. When a vendor places an order, the platform cross-references the forecast data to identify other vendors with overlapping demand cycles. Notifications are sent to these vendors, encouraging them to join the order and achieve bulk discounts.

- **Order Aggregation to Path Optimisation**: Once vendors confirm their participation, the order aggregation module compiles their requests into a single bulk

order. The aggregated data, including vendor locations and product quantities, is passed to the path optimisation module.

- **Path Optimisation for Delivery**: The path optimisation module designs a delivery route to ensure that all vendors receive their products efficiently. The vehicle carrying the bulk order is routed through clustered vendor locations, minimizing travel distance and logistics costs.

The seamless integration of demand forecasting, order aggregation, and path optimisation modules creates a powerful, collaborative platform that addresses the challenges faced by small-scale vendors. The demand forecasting module drives proactive order management, the order aggregation module facilitates cost-effective purchasing, and the path optimisation module ensures efficient delivery. Together, these modules enable vendors to compete with large retail chains by reducing costs, improving operations, and fostering a cooperative ecosystem.

3.5 Demand Forecasting with Neural Prophet

ASTRO platform, demand forecasting is essential for streamlining inventory management, optimizing order cycles, and enabling vendor collaboration. Using Neural Prophet, we forecast future sales for each store-item combination. This helps vendors make timely decisions about inventory and orders while facilitating joint purchasing to achieve bulk discounts. The model's capability to handle trends and seasonality makes it ideal for predicting sales patterns critical to the platform's success.

3.5.1 Inputs for Neural Prophet

The forecasting process begins with preparing the data. The key inputs used in the project include:

Column Name	Description	Format
`ds`	Date of sales	Datetime (`YYYY-MM-DD`)
`y`	Sales value for the given date	Numeric

Table 3.1: Key Inputs for Neural Prophet

The `ds` column represents the timeline for sales data, while `y` contains the actual number of items sold at a given store for a particular item. Both columns are preprocessed to ensure consistency. Missing sales values (`y`) are interpolated to maintain data continuity, and the `ds` column is checked for valid date formatting.

3.5.2 Neural Prophet Structure

Neural Prophet decomposes the sales data into the following components to identify patterns: The NeuralProphet model consists of multiple modules, each contributing an additive component to the forecast. The forecasted value $\hat{y}_t$ is the sum of the following components:

$$\hat{y}_t = T(t) + S(t) + E(t) + F(t) + A(t) + L(t) \tag{3.1}$$

Where:

- $T(t)$ = Trend at time t: This captures long-term growth or decline in the sales data.
- $S(t)$ = Seasonal effects at time t: This models recurring patterns in sales, such as seasonal peaks.
- $E(t)$ = Event and holiday effects at time t: This component adjusts for external events or holidays that affect demand.
- $F(t)$ = Regression effects for future-known exogenous variables at time t: This accounts for external factors, such as planned promotions or events, that are known in advance.
- $A(t)$ = Auto-regression effects based on past observations at time t: This models the relationship between past sales data and current demand.
- $L(t)$ = Regression effects for lagged observations of exogenous variables at time t: This captures the delayed impact of external variables on sales.

Each component is modeled independently, and their outputs are summed to generate the final forecast $\hat{y}_t$. The NeuralProphet framework is flexible, allowing each of these components to be switched on or off depending on the data and requirements of the project.

3.5.3 Outputs

The model generates forecasts over a predefined time horizon, typically ranging from 30 to 60 days based on vendor needs. The outputs include:

- **Predicted Sales** (`yhat1`): Projected daily sales for the forecasted period.
- **Smoothed Sales Trend** (`yhat1_trend`): Rolling averages (e.g., 30-day mean) applied to predicted values to highlight trends and reduce noise.

Date (ds)	Predicted Sales (yhat1)	Smoothed Sales (yhat1_trend)
2024-12-01	100	98
2024-12-02	105	99
2024-12-03	110	101

Table 3.2: Example of Forecasted Sales and Smoothed Trend

3.5.4 Integration with Platform Workflow

The demand forecasting module integrates seamlessly with other parts of the platform, creating a cohesive system that enhances vendor operations:

- **Inventory Management**: Forecasted sales help vendors prepare for upcoming demand, ensuring stock levels are optimized.
- **Order Aggregation**: Predictions trigger notifications to vendors nearing their replenishment cycle. These notifications encourage vendors to place joint orders, maximizing cost savings.
- **Logistics Optimisation**: Predicted delivery schedules align with logistics planning, enabling route optimisation and efficient load distribution.

By integrating Neural Prophet's forecasts into the platform, vendors can proactively manage inventory, coordinate joint purchases, and reduce costs. The model's ability to deliver actionable insights ensures the platform's operational efficiency and supports the growth of small-scale vendors.

3.6 Order Aggregation Using Genetic Algorithm

Order aggregation is a crucial module in the platform, enabling small-scale vendors to achieve bulk discounts by combining their orders with others. This process ensures that the Minimum Order Quantities (MOQ) required by suppliers are met, leading to cost savings and operational efficiency. The Genetic Algorithm (GA) is employed to optimize the selection of vendors for aggregation, ensuring that the combined order meets the MOQ while considering constraints like geographical proximity, order compatibility, and delivery logistics.

3.6.1 Process Workflow

When a vendor initiates an order for a product from a supplier, the platform identifies other vendors who might need the same product. Notifications are sent to these vendors, inviting them to join the order. The challenge lies in selecting the optimal subset of vendors whose combined orders meet or exceed the MOQ while minimizing costs like logistics and delivery time.

The Genetic Algorithm simulates the process of natural selection to arrive at the optimal solution. The steps include:

1. **Initialization**: A population of potential vendor groupings (chromosomes) is generated. Each chromosome represents a subset of vendors, encoded as binary strings, where '1' indicates inclusion and '0' exclusion from the group. For example:

$$\text{Chromosome } 1010: \begin{array}{l}\text{Vendor 1 and Vendor 3 are included,}\\ \text{while Vendor 2 and Vendor 4 are excluded.}\end{array}$$

2. **Fitness Evaluation**: The fitness of each chromosome is calculated based on its ability to meet the MOQ and minimize associated costs. The fitness function F can be expressed as:

$$F = \text{Revenue from bulk discounts} - (\text{Logistics cost} + \text{Order deviation penalty})$$

   - **Revenue from bulk discounts**: Rewards solutions that exceed the MOQ.
   - **Logistics cost**: Penalizes solutions with high transportation expenses.

- **Order deviation penalty**: Accounts for significant mismatches in vendor order requirements.

3. **Selection**: Chromosomes with higher fitness are selected for reproduction. Techniques like tournament selection or roulette wheel selection are used to ensure diversity while favoring fitter solutions.

4. **Crossover**: Selected chromosomes undergo crossover, where segments of two parent chromosomes are swapped to produce offspring. This operation introduces new combinations of vendors and explores alternative aggregation solutions.

5. **Mutation**: Random changes are introduced in some offspring to prevent premature convergence and explore less obvious solutions. For instance, flipping a bit in the chromosome (e.g., 1010 becomes 1110) can add or remove a vendor.

6. **Termination**: The algorithm iterates through several generations, refining the population until an optimal or near-optimal solution is found. Termination occurs when a predefined number of generations is reached or when the fitness stabilizes.

3.6.2 Output and Implementation

The output of the Genetic Algorithm is the optimal subset of vendors to participate in the aggregated order. This grouping ensures that:

- The MOQ is met, unlocking bulk discounts.
- Costs are minimized by considering logistics and order alignment.

Vendor	**Order Quantity**	**Selected (1/0)**
Vendor A	50	1
Vendor B	30	1
Vendor C	10	0
Vendor D	20	1

Table 3.3: Example of Vendor Selection in Order Aggregation

In this example, Vendors A, B, and D are selected, contributing a total order of 100 units, meeting the MOQ.

3.6.3 Integration with the Platform

Once the optimal vendor subset is determined, the platform finalizes the bulk order and notifies all participating vendors. The order details are then passed to the logistics module for route optimisation and delivery scheduling, ensuring seamless fulfillment. By leveraging the Genetic Algorithm, the platform provides an efficient, scalable, and automated solution for order aggregation, empowering vendors to compete effectively with larger retailers.

3.7 Logistics Using Green Route Optimisation Algorithm

Logistics optimisation is a key component of platform, ensuring efficient delivery of aggregated orders to vendors while minimizing transportation costs and environmental impact. In the project, we employ the Green Route Optimisation Algorithm, tailored to optimize the route for a single delivery vehicle. This algorithm determines the most efficient path to deliver products to different vendors included in an aggregated order, focusing on reducing fuel consumption, delivery time, and carbon emissions.

3.7.1 Mechanics of Logistics Optimisation

Once the order aggregation process is complete, the platform identifies the vendors participating in the bulk order and their respective delivery locations. The challenge is to determine the optimal route for a single delivery vehicle to service all vendors while adhering to constraints such as delivery time windows, vehicle capacity, and geographic proximity.

The Green Route Optimisation Algorithm is designed to achieve this by incorporating concepts of path flexibility and environmental considerations. Here's how it works:

Input Data: The algorithm takes the following inputs:

- Vendor locations (latitude and longitude).
- Delivery quantities for each vendor (ensuring vehicle capacity constraints are met).
- Service time windows for each vendor (if applicable).
- Vehicle starting point (typically the supplier's location).

Route Representation: The delivery route is represented as a sequence of vendor locations, starting and ending at the supplier's location. For example:

Route: Supplier → Vendor A → Vendor B → Vendor C → Supplier

Objective Function: The objective is to minimize the total distance traveled by the vehicle while considering additional factors like delivery priority and environmental impact. The optimisation function is:

$$\text{Minimize: } C = \sum_{i=1}^{n-1} d(p_i, p_{i+1}) + \alpha \cdot e(p_i, p_{i+1})$$

Where:

- $d(p_i, p_{i+1})$: Distance between points p_i and p_{i+1}.
- $e(p_i, p_{i+1})$: Emission cost for traveling between points p_i and p_{i+1}.
- α: Weighting factor for environmental impact.

Optimisation Process:

- The algorithm evaluates various route permutations using heuristic techniques like the Travelling Salesman Problem (TSP) approach combined with green routing principles.
- It calculates the cost for each route and iteratively refines it to find the most efficient path.
- Constraints like vehicle load capacity and vendor service time windows are enforced to ensure feasibility.

Output: The algorithm provides the optimized delivery route, detailing the order in which vendors should be serviced. For example:

Optimized Route: Supplier → Vendor D → Vendor A → Vendor C → Supplier

3.7.2 Advantages of Using Green Route Optimization Algorithms

The Green Route Optimisation Algorithm is specifically chosen for the project because it aligns with the platform's goals of cost efficiency, environmental sustainability, and scalability. Key reasons for its selection include:

- **Single Vehicle Focus:** Unlike fleet-based algorithms, it is optimized for scenarios where only one vehicle is used for deliveries, making it ideal for the project.
- **Environmental Considerations:** By factoring in emission costs, the algorithm promotes eco-friendly logistics, reducing the carbon footprint of deliveries.
- **Flexibility with Constraints:** It handles dynamic vendor locations, varying delivery quantities, and service time windows, ensuring practical applicability.
- **Efficiency and Cost Savings:** By minimizing the total distance traveled, the algorithm reduces fuel consumption and operational costs.

3.7.3 Integration with the Platform

After the aggregated order is finalized, the vendor delivery details are fed into the logistics module. The Green Route Optimisation Algorithm computes the optimal delivery path and provides the route to the delivery driver. The platform ensures that real-time updates, such as traffic conditions or vendor availability changes, are seamlessly integrated into the routing process if necessary.

This integration ensures timely and efficient delivery of products, enhancing vendor satisfaction while supporting the platform's commitment to sustainability and cost-effective operations.

3.8 System Integration and Notifications

The System Integration and Notifications module unifies all components in the platform, ensuring seamless data flow and robust communication across users. This module facilitates the synchronization of forecasting, order aggregation, and logistics, enabling vendors to make informed decisions and manage operations efficiently. The system's architecture

has been designed to optimize user interaction and data processing, allowing for real-time updates and notifications.

3.8.1 System Integration

System integration connects each component in the platform, enabling a smooth exchange of information necessary for coordinated operations. This integration links the demand forecasting, order aggregation, and logistics optimisation modules, ensuring that all components work in harmony. Key integration processes include:

- **Data Pipeline Management**: The platform manages a centralized data pipeline, which allows information such as sales forecasts, order details, and logistics schedules to flow seamlessly. Data from the forecasting model directly informs order aggregation, which in turn updates logistics routing.
- **Real-Time Inventory and Order Status Updates**: Inventory levels and order statuses are dynamically updated as transactions occur, keeping all users informed of current stock and order progress. When vendors place an order, the system automatically checks available stock, order quantities, and logistics parameters before initiating further processes.
- **Cross-Module Communication**: Communication protocols enable real-time interaction among components, with APIs connecting external data sources or integrating third-party logistics providers if needed. This setup enables efficient responses to fluctuations in demand or changes in supply chain schedules.

3.8.2 Notifications

The Notifications feature is integral to user engagement, keeping vendors updated on critical aspects of order processing, inventory status, and delivery schedules. Notifications are triggered by specific events within the platform, ensuring timely and relevant information delivery to users. The key notifications implemented in this system include:

- **Order Status Updates**: Vendors receive updates on order processing, confirmation of joint orders, and shipment tracking. This transparency allows vendors to plan their inventory management and sales strategy effectively.

- **Inventory Level Alerts**: Low inventory alerts notify vendors when stock reaches a specified threshold, ensuring they can replenish products before stockouts occur. High-demand items are monitored, and vendors are alerted to prevent missed sales opportunities.

- **Delivery Schedule Notifications**: Notifications include estimated delivery times and routing updates, assisting vendors in planning for incoming shipments and scheduling resources for unloading or further distribution.

Notifications are configured to reach users through multiple channels, such as in-app alerts, emails, and SMS, allowing vendors to remain updated even when away from the platform.

Table 3.4: Notification Types and Triggers

Notification Type	**Trigger Event**	**Delivery Channel**
Order Status Update	Order Confirmation, Shipment Dispatch	In-App, Email
Inventory Level Alert	Low Stock Threshold Reached	In-App, SMS
Delivery Schedule	Route Finalized, Estimated Delivery Time	In-App, Email

This chapter has outlined the system's architecture and described each component's role in creating a cohesive platform for demand forecasting, order aggregation, and logistics management. Through effective system integration and real-time notifications, the platform provides a unified experience that supports vendors in optimizing their operations, managing inventory, and coordinating logistics. The collaborative approach facilitated by this platform empowers vendors to maintain competitive advantages, meeting demand efficiently and enhancing overall supply chain resilience.

Chapter 4

DESIGN AND MODELING

Design and modeling are essential phases in the software development lifecycle, playing a crucial role in the creation of effective and wellstructured systems. These phases involve the conceptualization, planning, and representation of a software solution before its actual implementation.

4.1 Activity Diagram

An activity diagram is a type of UML (Unified Modeling Language) diagram that visually represents the flow of activities in a system or business process. It is commonly used in software development to illustrate the dynamic aspects of a system and describe how different activities or tasks interact with each other. Activity diagrams are particularly useful for modeling the workflow within a system and for understanding the sequence of actions that take place.

4.1.1 Key elements of Activity Diagrams

- **Activity:** Represented by rounded rectangles, activities are the specific tasks or actions that are performed within the system. These can range from simple operations to complex processes.
- **Transitions:** Arrows connecting activities indicate the flow or transition from one activity to another. The direction of the arrow shows the order of execution.
- **Decision Nodes:** Diamonds are used to represent decision points in the workflow. Depending on certain conditions, the process may take different paths.
- **Fork and Join Nodes:** Fork nodes (split) and join nodes (merge) show parallel or concurrent activities. Forks represent the divergence of multiple flows, while joins

represent their convergence.

- **Initial and Final Nodes:** Circles are used to denote the start (initial node) and end (final node) of the activity diagram. The initial node represents the beginning of the process, and the final node indicates the conclusion.

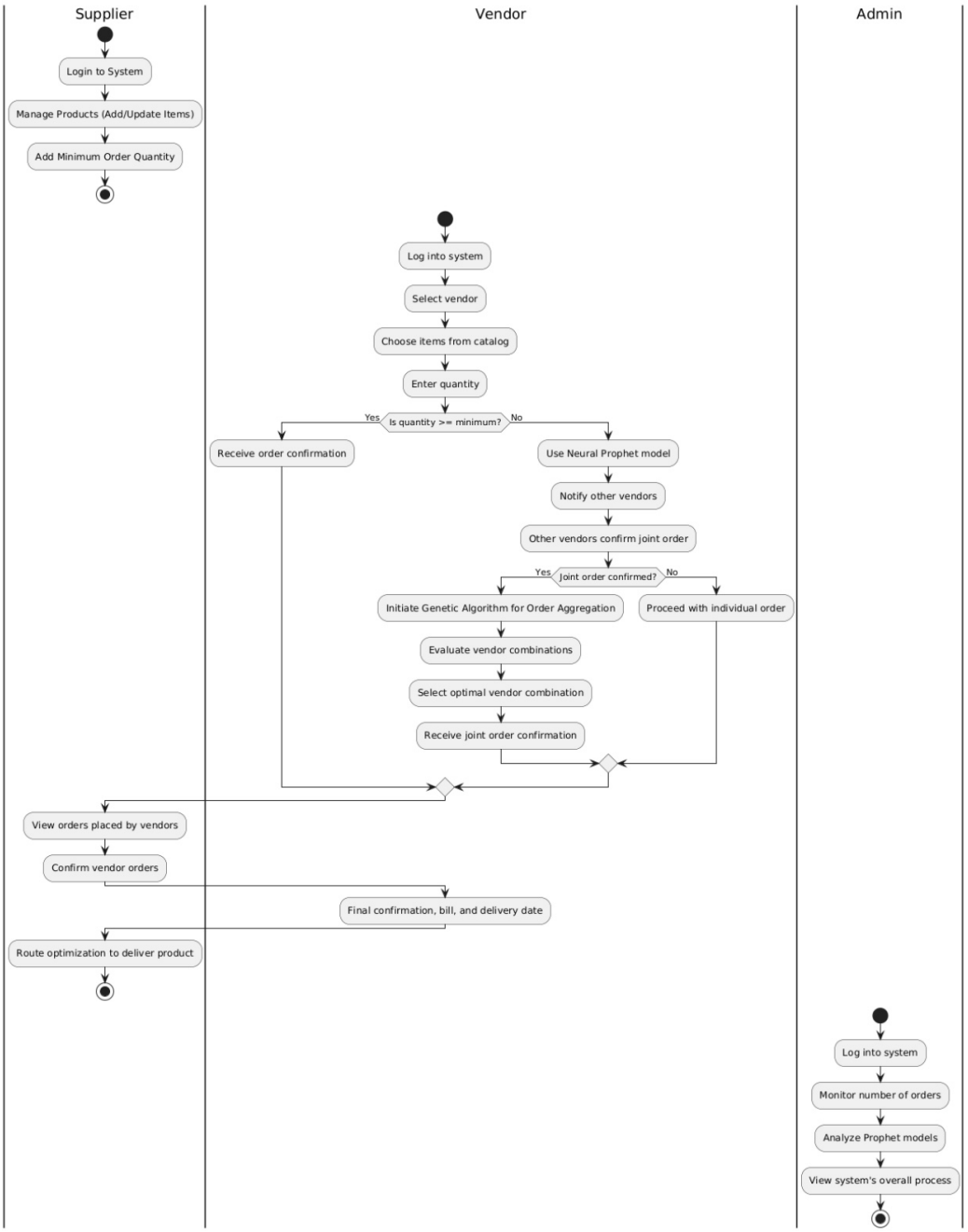


Figure 4.1: Activity Diagram

Figure 4.1 illustrates the activity diagram for the ASTRO Platform, showcasing the sequence of activities performed by suppliers, vendors, and admins. This diagram provides

a high-level overview of the key entities and components involved in the system, detailing their interactions and responsibilities throughout the order processing and delivery process.

4.1.2 Key Entities and Components

1. **Supplier Activities**

   (a) **Login to System:** The supplier logs into the system to access functionalities for managing products and tracking orders.

   (b) **Manage Products (Add/Update Items):** Suppliers add or update product listings to ensure the catalog is up-to-date and accurate.

   (c) **Add Minimum Order Quantity:** Suppliers set a minimum order quantity for each product, preventing inefficient processing of very small orders.

   (d) **View Orders Placed by Vendors:** Suppliers can view pending orders from vendors, preparing them for order fulfillment.

   (e) **Confirm Vendor Orders:** Suppliers review and confirm vendor orders, moving them to the fulfillment phase.

   (f) **Route Optimisation for Delivery:** The supplier optimizes delivery routes to reduce logistics costs and ensure timely product delivery.

2. **Vendor Activities**

   (a) **Log into System:** Vendors start by logging into the system to access the product catalog and ordering functionalities.

   (b) **Select Vendor and Choose Items from Catalog:** Vendors select suppliers to order from, browsing the catalog to choose items.

   (c) **Enter Quantity:** Vendors input desired quantities for each item, triggering a check against the minimum order quantity.

   (d) **Quantity Check (Yes/No):**

   - **If Quantity Meets Minimum Requirement:** The order is confirmed directly.

- **If Quantity Is Below Minimum Requirement:**
  i. **Use Neural Prophet Model:** The system uses a Neural Prophet model to forecast demand and determine if combining orders with other vendors can reach the minimum order threshold.
  ii. **Notify Other Vendors:** Other vendors are notified to consider joining a joint order.
  iii. **Joint Order Confirmation Check (Yes/No):**
    – **If Other Vendors Agree:** The joint order is confirmed.
    – **If Not:** The vendor proceeds with an individual order.
  iv. **Initiate Genetic Algorithm for Order Aggregation:** The system uses a genetic algorithm to identify the best vendor combinations for the joint order.
  v. **Evaluate Vendor Combinations:** Different combinations are assessed to find an optimal one that meets requirements.
  vi. **Select Optimal Vendor Combination:** The system selects the optimal vendor combination, and a joint order confirmation is received.
  vii. **Receive Final Confirmation:** Vendors receive the final confirmation, including billing details and delivery date.

3. **Admin Activities**

   (a) **Log into System:** The admin logs into the system to monitor and analyze overall processes.

   (b) **Monitor Number of Orders:** Admin tracks order volumes, gaining insights into demand trends and operational load.

   (c) **Analyze Prophet Models:** Admin reviews the performance of demand forecasting models, such as Neural Prophet, to ensure accurate predictions and make adjustments if necessary.

   (d) **View System's Overall Process:** Admin has a top-down view of system activities to ensure smooth operations and identify areas for improvement.

Here's how activity diagrams can be specifically helpful in this context:

- Enhanced Clarity of Process Flow: The diagram visually represents the process, making it easier for stakeholders to understand each role's responsibilities and interactions. By defining the steps each role follows, the diagram ensures that all parties (Suppliers, Vendors, and Admins) understand their tasks and when each task needs to be completed.

- Identification of Key Decision Points: The activity diagram highlights critical decision points, such as the quantity check for minimum orders. This helps in identifying points where conditional logic and alternative flows come into play, ensuring that the system can handle various scenarios efficiently. For example, if the order quantity is insufficient, the system automatically checks for joint orders, thus streamlining the process without manual intervention.

- Improved Coordination Between Roles: By detailing the interactions between suppliers, vendors, and admins, the diagram promotes better coordination. Vendors can initiate joint orders when quantities are low, and suppliers can confirm orders and optimize delivery routes accordingly. This collaboration ensures that orders are fulfilled more efficiently and that each role's activities are synchronized.

- Guidance for System Design and Development: For developers, the activity diagram serves as a blueprint for creating system modules. Each activity can be mapped to specific features in the system, like order confirmation, joint order aggregation, route optimisation, and demand forecasting using the Neural Prophet model. This structured approach helps in modular and organized system development, making the codebase easier to manage and maintain.

4.2 Sequence Diagram

A sequence diagram is a type of UML (Unified Modeling Language) diagram that illustrates the interactions between different components or actors in a system over time. It presents a chronological sequence of messages or actions between the various entities, providing a visual representation of the dynamic behavior of the system.

4.2.1 Key elements of Sequence Diagrams

- **Actors and Objects:** Represent the entities (users, systems, or objects) involved in the interactions.
- **Lifelines:** Vertical lines extending from actors or objects, showing their presence over time in the sequence.
- **Messages:** Arrows that represent the communication between lifelines, including method calls, data exchanges, and responses.
- **Activation Bars:** Narrow rectangles on lifelines indicating when an object is actively performing a task.
- **Control Structures:** Elements like loops, conditions, and alternative frames (alt/opt frames) that model conditional, repeated, or optional interactions.
- **Destruction:** Marked by an "X" at the end of a lifeline, indicating the termination of an object in the sequence.

4.2.2 Purpose of Sequence Diagrams

- Understanding System Behavior: Sequence diagrams help in understanding how different components or actors in a system interact with each other over time, depicting the flow of control and communication.
- Communication: They serve as a communication tool among stakeholders, including developers, designers, and project managers, by offering a clear and visual representation of system interactions.
- Design and Analysis: Sequence diagrams aid in the design and analysis phases of software development, helping to identify potential issues in the system's logic or communication flow.
- Visualizing the Workflow: It provides a clear and concise overview of the different steps involved in the Alzheimer's disease detection process.
- Identifying Potential Bottlenecks: It helps to identify any potential bottlenecks in the system, such as slow preprocessing steps or inaccurate model predictions.

- Facilitating Communication: It serves as a common language for developers, researchers, and other stakeholders involved in the project.

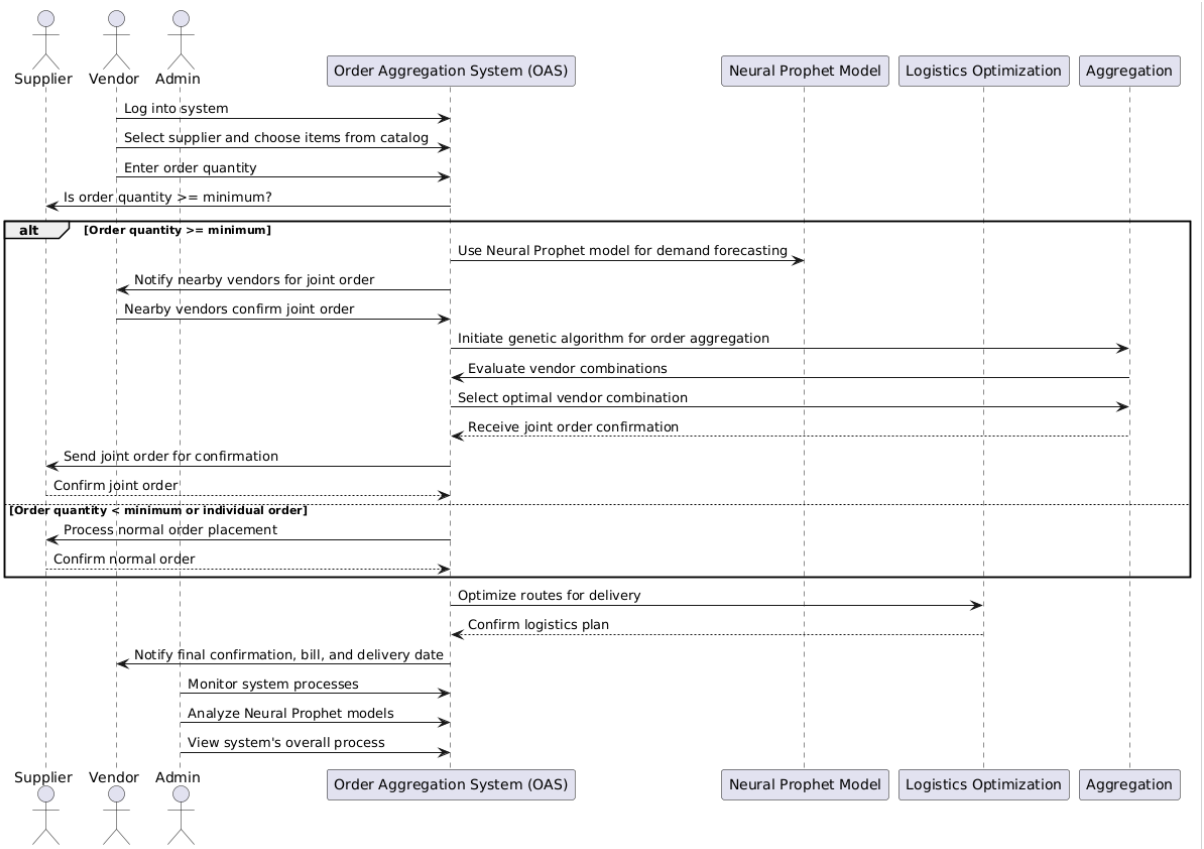


Figure 4.2: Sequence Diagram

Figure 4.2 illustrates the sequence of interactions between the key entities and components of the ASTRO Platform. This diagram provides a step-by-step view of how the system processes an order from initiation to delivery. It shows the flow of messages and the order in which they are exchanged, highlighting the interactions between the Supplier, Vendor, Admin, Order Aggregation System (OAS), Neural Prophet Model, and Logistics Optimisation.

4.2.3 Key Entities and Components

- **Supplier:** Manages product inventory and fulfills confirmed orders.
- **Vendor:** Places orders for products based on the demand forecast and minimum order requirements.

- **Admin:** Monitors the overall process, analyzes model performance, and reviews logistics planning.
- **Order Aggregation System (OAS):** Core system component that manages the flow of order requests, checks conditions, and coordinates with other components.
- **Neural Prophet Model:** A demand forecasting model that helps in predicting demand and evaluating whether joint orders would be beneficial.
- **Logistics Optimisation:** Component responsible for optimizing delivery routes and managing the final logistics.

4.2.4 Sequence of Interactions

1. **Login to System:** The Vendor initiates the sequence by logging into the system to access the product catalog, choose items, and place orders.
2. **Select Supplier and Choose Items from Catalog:** The Vendor selects a supplier and items from the catalog, specifying the order quantity.
3. **Order Quantity Check:** The Order Aggregation System (OAS) checks whether the order quantity meets the minimum order requirement.
    - **If Order Quantity Meets Minimum Requirement:** The order is processed as a standard, individual order. The vendor confirms the order, and it proceeds to the delivery logistics phase.
    - **If Order Quantity Is Below Minimum Requirement:** The OAS initiates steps to evaluate the feasibility of a joint order.
        (a) **Use Neural Prophet Model for Demand Forecasting:** The Neural Prophet Model is used to forecast demand and determine if combining orders from nearby vendors might help reach the minimum order quantity. This forecast provides data-driven insights into the order's potential demand over time.
        (b) **Notify Nearby Vendors for Joint Order:** If the demand forecast suggests that a joint order is viable, the OAS sends notifications to nearby vendors, inviting them to participate in a joint order.

(c) **Nearby Vendors Confirm Joint Order:** Other vendors respond to the joint order request. If they agree, the joint order process moves forward; otherwise, the initial vendor proceeds with an individual order.

(d) **Initiate Genetic Algorithm for Order Aggregation:** The OAS employs a Genetic Algorithm to evaluate different combinations of vendors for the joint order, aiming to find an optimal mix that maximizes efficiency and meets the minimum order requirements.

(e) **Evaluate Vendor Combinations:** The Genetic Algorithm component iterates through various vendor combinations to find the best possible aggregation. This ensures that the order is cost-effective and meets quantity requirements.

(f) **Select Optimal Vendor Combination:** The system selects the optimal vendor combination for the joint order and proceeds to confirmation.

(g) **Receive Joint Order Confirmation:** Once an optimal combination is selected, the OAS sends a joint order confirmation to all participating vendors. The joint order is now officially confirmed.

4. **Order Processing and Final Confirmation:** For both joint and individual orders, the system notifies vendors of the final confirmation, bill, and delivery date.

5. **Optimize Routes for Delivery:** The Logistics Optimisation component optimizes delivery routes based on the confirmed orders, minimizing delivery time and cost by consolidating deliveries where possible.

6. **Admin Activities:** The Admin monitors system processes and performance. Admin also reviews and analyzes the accuracy of Neural Prophet Model forecasts and evaluates system functionality, using insights to make improvements to the process.

4.3 Class Diagram

A class diagram is a visual blueprint of a system in object-oriented programming, showcasing the classes, their attributes (properties), operations (methods), and the relationships between them. It's like a map that reveals the building blocks and their connections, guiding the development and understanding of the system.

4.3.1 Key Elements of Class Diagrams

- **Classes:** These are the fundamental building blocks, represented as rectangles with the class name inside. Each class encapsulates a specific concept or entity within the system, like "Customer" or "Order."
- **Attributes:** These define the data associated with a class, like "name" and "address" for the "Customer" class. They're listed within the class rectangle, often with data types specified.
- **Operations:** These represent the actions a class can perform, like "placeOrder" or "calculateTotal" for the "Order" class. They're shown as functions within the class rectangle, with their parameters and return values if applicable.
- **Relationships:** These depict the connections between classes, indicating how they interact with each other.

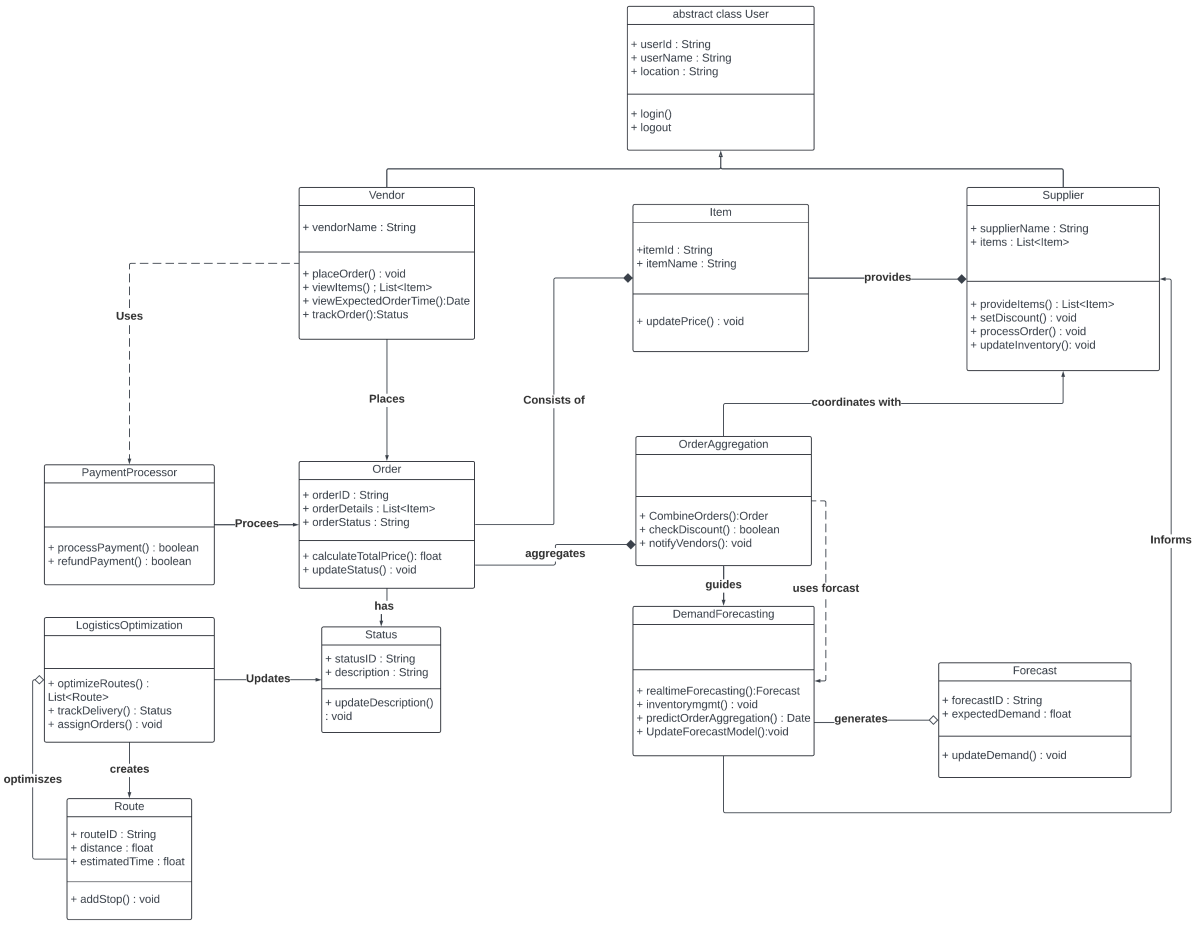


Figure 4.3: Class Diagram

Figure 4.3 illustrates the class diagram for the ASTRO Platform, showcasing the key classes and their relationships. This diagram provides a structural overview of the system, detailing the classes like User, Vendor, Supplier, Item, Order, Status, OrderAggregation, PaymentProcessor, LogisticsOptimization, DemandForecasting, and Forecast, along with their attributes and methods.

4.3.2 Class Descriptions

- **User:** An abstract class representing a generic user with attributes like userId, userName, and location, along with methods login() and logout(). Inherited by Vendor and Supplier classes.

- **Vendor:** Represents a vendor who manages orders, with attributes such as vendorName and methods like placeOrder(), viewItems(), viewExpectedOrderTime(), and trackOrder(). Associated with Order and interacts with PaymentProcessor for payment handling.

- **Supplier:** Represents a supplier managing inventory with attributes like supplierName and items, and methods like provideItems(), setDiscount(), processOrder(), and updateInventory(). Coordinates with OrderAggregation and supplies items to the Item class.

- **Item:** Represents individual items with attributes like itemId and itemName, and a method updatePrice() to modify the item's price. Associated with Order as orders consist of multiple items.

- **Order:** Represents a customer order, including attributes orderID, orderDetails (list of items), and orderStatus, with methods calculateTotalPrice() and updateStatus(). Consists of multiple Item objects and is associated with Status.

- **Status:** Represents the order's current status with attributes statusID and description, and a method updateDescription() to modify the status description. Linked to Order to provide status details.

- **OrderAggregation:** Manages combined orders with methods CombineOrders(),

checkDiscount(), and notifyVendors(). Coordinates with Supplier for order processing and informs DemandForecasting about aggregate demand.

- **PaymentProcessor:** Handles payments and refunds with methods processPayment() and refundPayment(). Interacts with Vendor for transaction management.
- **LogisticsOptimization:** Optimizes delivery routes with methods optimizeRoutes(), trackDelivery(), and assignOrders(), creating Route objects as part of logistics planning.
- **Route:** Represents a delivery route with attributes routeID, distance, and estimatedTime, and a method addStop() to add stops. Created by LogisticsOptimization.
- **DemandForecasting:** Predicts demand and manages inventory with methods like realtimeForecasting(), inventoryMgmt(), predictOrderAggregation(), and updateForecastModel(). Uses data from Forecast to guide order aggregation and inventory planning.
- **Forecast:** Provides demand forecasts with attributes forecastID and expectedDemand, and a method updateDemand(). Generated by DemandForecasting and used for planning inventory and order aggregation.

4.3.3 Summary of Associations

1. Vendor places orders using Order and interacts with PaymentProcessor.
2. Supplier provides Items to the system and coordinates with OrderAggregation for bulk orders.
3. OrderAggregation combines orders and coordinates with Supplier, also informing DemandForecasting.
4. DemandForecasting uses forecasts from Forecast to plan inventory and order aggregation.
5. LogisticsOptimization creates Routes to optimize delivery.

4.3.4 Common Relationships

- **Association:** Shows a simple connection between two classes, like "Customer" has an "Order."
- **Aggregation:** A "has-a" relationship where one class (the whole) is made up of parts (the other class), like "Order" has "OrderItems."
- **Composition:** A stronger "has-a" relationship where the parts (the other class) cannot exist independently of the whole (one class), like "Car" has "Wheels."
- **Inheritance:** Shows a hierarchical relationship where one class (subclass) inherits properties and methods from another class (superclass), like "Employee" inherits from "Person."

4.3.5 Benefits of Using Class Diagrams

- **Clarity:** They provide a visual representation of complex systems, making them easier to understand and communicate.
- **Communication:** They serve as a common language for developers, designers, and other stakeholders to agree on the system's structure.
- **Analysis:** They help identify potential problems or inconsistencies in the system's design before implementation.
- **Documentation:** They serve as a valuable reference for understanding and maintaining the system over time.

Where You Might Encounter Them

- **Software Development:** Class diagrams are used throughout the software development lifecycle, from initial design to implementation and maintenance.
- **System Design:** They are crucial for modeling the architecture of complex systems, including web applications and enterprise software.
- **Documentation:** They are often included in technical documentation to provide a clear overview of the system's structure.

4.4 Use Case Diagram

In the Unified Modeling Language (UML), a Use Case Diagram is a sort of behavioral diagram that depicts the interactions between various actors (users or external systems) and a system that is being studied. It offers a high-level perspective on how different entities use the features or functionalities of a system. A use case diagram's main objective is to represent and illustrate the various ways users engage with a system and the results they receive.

4.4.1 Key Elements of Use Case Diagrams

- **Use Case:** A use case is an example of a particular functionality or group of related functions that the system offers to its users, also known as actors. Use cases are designated to indicate a particular action or objective and are usually represented as ovals.

- **Actor:** An outside party interacting with the system is called an actor. Actors might be physical devices, other systems, or human actors. Stick figures are used to represent actors, who engage with the system by taking part in one or more use cases.

- **System Boundary:** Depicted as a box, the system boundary establishes the parameters of the system and demarcates its external participants. Actors reside outside the system boundaries, whereas use cases exist inside it.

- **Association:** Associations are shown as lines joining actors to use cases. These lines show that an actor participates in or communicates with a certain use case. Associations serve as a channel of communication between the use case and the actor.

- **Include Relationship:** An include relationship shows that one use case incorporates the functionality of another use case. It is represented by a dashed arrow. It suggests that the behavior of the base use case includes the added use case.

- **Extend Relationship:** An extended connection shows that, under certain circumstances, one use case can extend the behavior of another use case. It is represented

by a dashed arrow with an open arrowhead. It suggests that the expanding use case is called upon under particular circumstances and is optional.

Some use cases might not include interactions with a particular actor; instead, they might represent system-wide operations or processes. We call these use cases for the system.

4.4.2 Value of Use Case Diagrams

Use Case Diagrams are valuable tools for:

- **Communication:** Use case diagrams offer a visual representation that helps stakeholders—developers, designers, and non-technical stakeholders—communicate with one other.
- **Requirements Analysis:** By recognizing and recording user-system interactions, they aid in the elicitation and clarification of system requirements.
- **System Design:** The architecture and user interfaces of the system are designed using use case diagrams as a guide.
- **Testing:** By figuring out scenarios in which users engage with the system, they can be used to generate test cases.
- **Project Planning:** By highlighting important features and their relationships, use case diagrams can help with project planning.

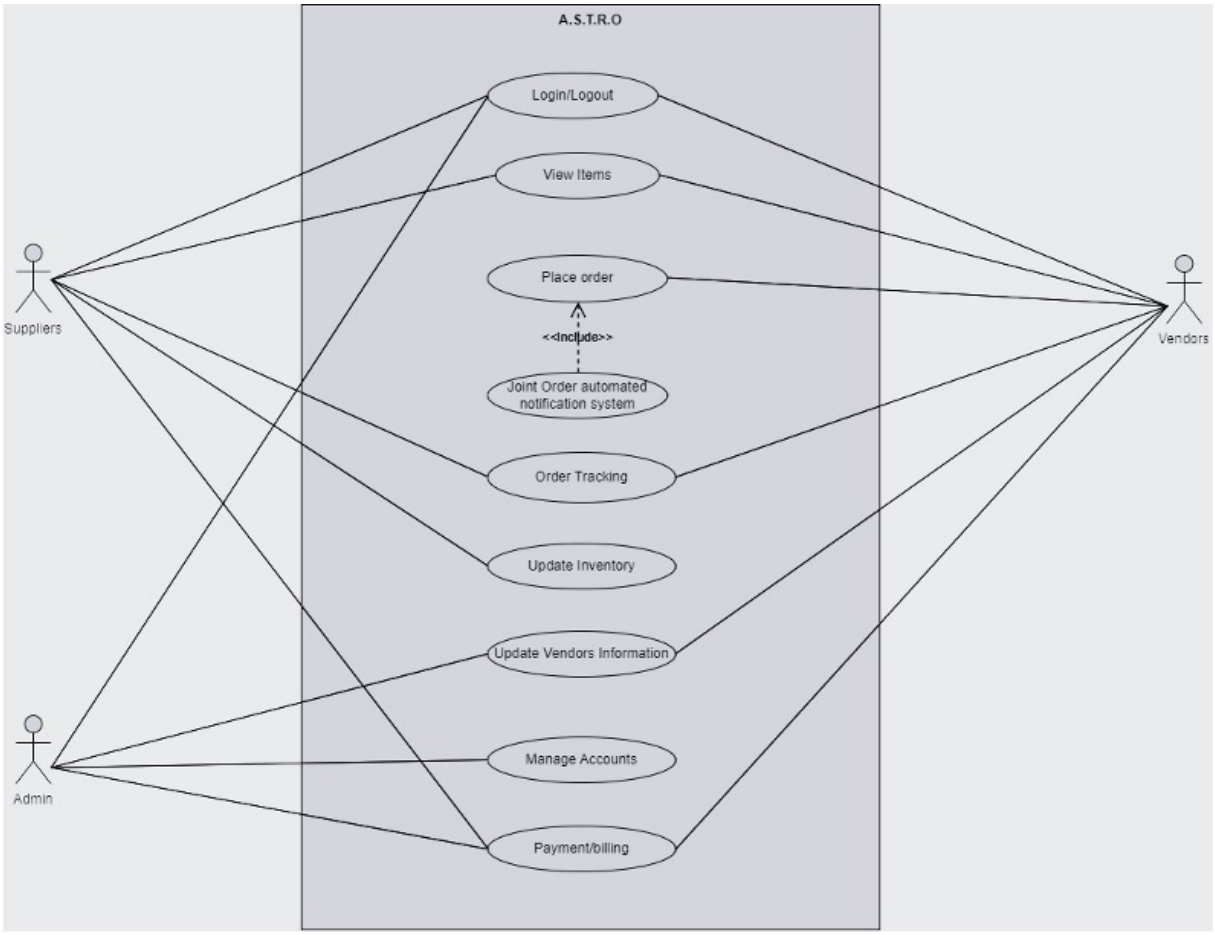


Figure 4.4: Use Case Diagram

Figure 4.4 illustrates the use case diagram for the ASTRO Platform, showcasing the interactions between the actors (Suppliers, Vendors, and Admin) and the system. This diagram provides a high-level view of the functionalities available to each actor and how they interact with the system to place orders, manage inventory, track orders, and perform administrative tasks.

4.4.3 Actors

1. **Suppliers:**
    - Responsible for providing inventory or items for the system.
    - Has access to most functions related to orders, inventory, and tracking.
2. **Vendors:**
    - Likely responsible for purchasing or receiving items in the system.

- Similar to Suppliers, has access to most functions, especially those related to placing orders, notifications, and tracking.

3. **Admin:**

   - Responsible for overall management, including accounts and billing.
   - Has the ability to manage vendor information and accounts in addition to other functionalities.

4.4.4 Use Cases

1. **Login/Logout:**

   - All users (Suppliers, Vendors, and Admin) can log in and log out of the system, indicating that authentication is a standard feature.

2. **View Items:**

   - Suppliers and Vendors have access to this use case, allowing them to view items available in the system, possibly for ordering or tracking purposes.

3. **Place Order:**

   - Both Suppliers and Vendors can place orders in the system, which might include ordering goods or services. This use case includes an include relationship to the Joint Order Automated Notification System use case, meaning that when an order is placed, an automated notification system is triggered.

4. **Joint Order Automated Notification System:**

   - An included use case for the Place Order use case.
   - This automated system sends notifications for joint orders, possibly to inform both Suppliers and Vendors of order status or details.

5. **Order Tracking:**

   - Both Suppliers and Vendors can track the status of their orders through the system, giving them visibility into order fulfillment stages.

6. **Update Inventory:**

    - This use case allows both Suppliers and Vendors to update inventory information, which could involve adding, editing, or removing items based on stock levels or requirements.

7. **Update Vendors Information:**

    - Accessible by all actors, including Admin.
    - This use case allows updating information related to Vendors, which could involve updating contact details, address, or other essential information.

8. **Manage Accounts:**

    - Specific to the Admin, who has control over managing accounts in the system.
    - This might involve user account creation, permissions, and other administrative tasks.

9. **Payment/Billing:**

    - Only the Admin has access to this functionality, likely responsible for handling the financial aspects of transactions in the system, including processing payments and issuing bills.

4.4.5 Relationships

- **Include Relationship:** The Place Order use case includes the Joint Order Automated Notification System, meaning that this automated notification is a required part of placing an order. This might help keep all parties informed about order updates.
- **System Boundary:** The shaded box labeled ASTRO represents the system boundary, meaning all the use cases within this box are functionalities provided by the ASTRO system.

Chapter 5

Conclusions & Future Scope

The ASTRO platform represents a comprehensive solution designed to empower small-scale vendors by addressing their core operational challenges. By facilitating collaborative purchasing, optimizing logistics, and offering data-driven financial services, ASTRO significantly enhances the competitive edge of small vendors, enabling them to compete more effectively with large retail chains. The platform's multifaceted approach not only reduces operational costs and improves efficiency but also provides vendors with the financial resources and data insights necessary for informed decision-making and strategic growth.

Through collaborative purchasing, ASTRO enables vendors to achieve bulk pricing discounts, thereby reducing the cost of goods sold and increasing profitability. The logistics optimisation feature streamlines delivery routes, minimizing transportation costs and ensuring timely deliveries, which enhances customer satisfaction and operational reliability. Additionally, the integration of data-driven financial services addresses the critical issue of limited access to capital, providing vendors with tailored credit assessments and loan options based on their transaction data. This financial support is instrumental in facilitating business expansion, inventory management, and investment in technology.

ASTRO's real-time demand forecasting tool equips vendors with the ability to anticipate market demand accurately, optimizing inventory levels and reducing the risk of overstocking or stockouts. This predictive capability not only enhances operational efficiency but also ensures that vendors can meet customer needs promptly and effectively. The platform's user-friendly interface ensures accessibility and ease of use, allowing vendors with varying levels of technical expertise to leverage its features effectively.

Overall, ASTRO fosters economic resilience and sustainability within local communities by empowering small-scale vendors to thrive in a competitive marketplace. By addressing the challenges of limited purchasing power, inefficient logistics, restricted fi-

nancial access, and lack of data-driven decision-making, ASTRO contributes to the long-term success and stability of small vendors, promoting a more diverse and robust local economy.

While ASTRO has made significant strides in empowering small-scale vendors, there are numerous opportunities for future enhancements and expansions to further augment its impact and effectiveness. The following areas outline potential future developments:

1. **Advanced Forecasting and Predictive Analytics**: Integrate more sophisticated machine learning algorithms to enhance the accuracy and granularity of demand forecasting, allowing vendors to anticipate market trends more accurately and optimize inventory management.

2. **AI-Enhanced Credit Assessment**: Develop AI-driven credit scoring models that utilize a broader range of data points, including transaction history and purchasing patterns, to provide personalized and accurate financial services.

3. **Integration with Broader Supply Chains**: Establish partnerships with larger suppliers and logistics providers to expand ASTRO's logistics optimisation features, enabling vendors to access a wider range of products at competitive prices.

4. **Sustainability Initiatives**: Incorporate environmentally sustainable practices, such as carbon footprint tracking and eco-friendly route planning, to align vendors with growing consumer demand for eco-friendly products.

5. **Customizable Vendor Portals and Data Insights**: Develop customizable dashboards for tailored analytics based on vendor-specific needs, enabling strategic business decisions.

6. **Expansion of Financial Services**: Introduce more financial products, such as insurance and investment options, to provide vendors with comprehensive financial support.

7. **Enhanced Collaboration Features**: Implement tools for communication and resource sharing among vendors, fostering a sense of community and mutual support.

8. **Mobile Application Development**: Develop a dedicated mobile application for on-the-go access to ASTRO's features, enhancing accessibility and user engagement.

9. **Localization and Customization**: Adapt ASTRO to different regions' specific needs and regulatory requirements, ensuring relevance across diverse contexts.

10. **Integration with E-Commerce Platforms**: Integrate ASTRO with popular e-commerce platforms to enable seamless online ordering and sales management, expanding vendors' market reach.

11. **User Training and Support Programs**: Provide training programs to help vendors maximize ASTRO's benefits, enhancing adoption and success.

12. **Data Privacy and Security Enhancements**: Implement advanced measures for data protection, building user trust and ensuring data confidentiality.

13. **Feedback and Continuous Improvement Mechanism**: Establish a feedback system to gather user input and improve ASTRO's features and functionalities over time.

14. **Scalability and Performance Optimisation**: Ensure the platform's scalability and performance can handle increased users and data as ASTRO grows.

15. **Exploration of New Markets and Industries**: Expand ASTRO's framework to apply to other markets and industries beyond small-scale vendors, opening new avenues for growth and impact.

References

[1] S. J. Taylor and B. Letham, "Forecasting at scale," *PeerJ Preprints*, vol. 5, p. e3190v2, Sep. 2017. [Online]. Available: https://doi.org/10.7287/peerj.preprints.3190v2

[2] X. Wang and L. Wang, "Green routing optimization for logistics distribution with path flexibility and service time window," in *2018 15th International Conference on Service Systems and Service Management (ICSSSM)*, 2018, pp. 1–5.

[3] E. M. U. S. B. Ekanayake, S. P. C. Perera, W. B. Daundasekara, and Z. A. M. S. Juman, "A modified ant colony optimization algorithm for solving a transportation problem," *Journal of Advances in Mathematics and Computer Science*, vol. 35, no. 5, p. 83–101, Aug. 2020. [Online]. Available: https://journaljamcs.com/index.php/JAMCS/article/view/1473

[4] X. Ai, Y. Yue, H. Xu, and X. Deng, "Optimizing multi-supplier multi-item joint replenishment problem for non-instantaneous deteriorating items with quantity discounts," *PLOS ONE*, vol. 16, no. 2, pp. 1–22, 02 2021. [Online]. Available: https://doi.org/10.1371/journal.pone.0246035

[5] O. Triebe, H. Hewamalage, P. Pilyugina, N. Laptev, C. Bergmeir, and R. Rajagopal, "Neuralprophet: Explainable forecasting at scale," 2021. [Online]. Available: https://arxiv.org/abs/2111.15397

[6] S. B. Jouida and S. Krichen, "A genetic algorithm for supplier selection problem under collaboration opportunities," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 34, no. 1, pp. 53–79, 2022. [Online]. Available: https://doi.org/10.1080/0952813X.2020.1836031

[7] M. Seifbarghy, M. Shoeib, and D. Pishva, "Coordination of a single-supplier multi-retailer supply chain via joint ordering policy considering incentives for

retailers and utilizing economies of scale," *Scientia Iranica*, pp. –, 2022. [Online]. Available: https://scientiairanica.sharif.edu/article_22670.html

[8] J. Zhou, "Research on multi-objective green vehicle path optimization based on whale algorithm," in *2023 International Conference on Networking, Informatics and Computing (ICNETIC)*, 2023, pp. 385–389.

[9] K. A. Mamoun, L. Hammadi, A. E. Ballouti, A. G. Novaes, and E. S. D. Cursi, "Vehicle routing optimization algorithms for pharmaceutical supply chain: A systematic comparison," *Transport and Telecommunication Journal*, vol. 25, no. 2, pp. 161–173, 2024. [Online]. Available: https://doi.org/10.2478/ttj-2024-0012

[10] A. Maroof, B. Ayvaz, and K. Naeem, "Logistics optimization using hybrid genetic algorithm (hga): A solution to the vehicle routing problem with time windows (vrptw)," *IEEE Access*, vol. 12, pp. 36 974–36 989, 2024.

Appendix A: Presentation

DESIGN PRESENTATION

Advanced Supply Trade Resource Optimization
(A.S.T.R.O.)

Group members:
Amel Chandra : U2109009
Bharath S : U2109018
Joepaul Vilsan : U2109033
Shane George Salphie : U2109063

Guide:
Dr Nikhila T Bhuvan

1

Introduction

- The project develops a platform that helps small-scale vendors compete with large retail chains by enabling collaborative purchasing and optimized logistics.
- The platform provides data-driven financial services, empowering vendors to make informed decisions, achieve bulk pricing, and enhance local economic resilience.

2

Problem Definition

- **Problem :**
  Small-scale vendors struggle to compete with large retail chains due to limited purchasing power, inefficient logistics, and restricted access to financial services. These challenges prevent them from managing demand fluctuations effectively, securing competitive pricing, and making data-driven business decisions.
- **Solution :**
  Our platform empowers small-scale vendors by enabling collaborative purchasing to achieve bulk discounts, optimizing logistics to reduce costs, and providing data-driven financial services. This solution enhances vendors' competitiveness, allowing them to make informed decisions and thrive in the market.

3

Scope and Motivation

Scope:

- Development of a platform for small-scale vendors.
- Features include user registration, order aggregation, and real-time demand management.
- Vendors can collaborate on orders to achieve bulk pricing.
- Integrated logistics optimization and data-driven financial services.

Motivation:

- Empower small-scale vendors to compete with large retail chains.
- Enhance economic resilience and sustainable growth in local communities.
- Provide vendors with tools for informed decision-making and operational efficiency.

4

Target Groups

- **Small-Scale Vendors**: Independent businesses aiming to compete with large retail chains through cost-effective purchasing and logistics.
- **Local Suppliers**: Providers seeking to increase sales by collaborating with multiple vendors.
- **Platform Administrators**: Users managing vendor and supplier interactions, order coordination, and logistics optimization.

5

Objectives

1. Enable Collaborative Purchasing :
- Develop a platform for small vendors to combine orders, securing bulk pricing and discounts.

2. Optimize Logistics :
- Implement algorithms to improve delivery routes, reduce costs, and enhance efficiency.

3. Real-Time Demand Forecasting :
- Use forecasting tools to help vendors anticipate demand and optimize inventory.

6

Challenges

- **Complex Models**: Requires tuning and resources for NeuralProphet and Genetic Algorithms.
- **High Computational Needs**: Limits real-time processing capabilities.
- **Scalability**: Managing performance as vendors, products, and locations grow.
- **Data Consistency**: Ensuring seamless data flow across forecasting, supplier selection, and logistics.

7

Resources Needed

- **Software Requirements :**
  - **OS :** Ubuntu / Windows / MacOS
  - **Frontend Library :** Flutter
  - **Backend Library :** Fast API
  - **IDE :** Visual Studio Code
  - **Programming language :** Python, Dart
  - **Containerization :** Docker
- **Dataset :**
  - Store Item Demand Forecasting Challenge – Kaggle Dataset (Kaggle Link)
- **Hardware Requirements :**
  - **CPU :** Ryzen 3 1200 – 3.1Ghz / Core i5-4460 – 3.2Ghz / Apple M1 chip
  - **RAM :** 8GB
  - **Video card :** AMD Raedon R9 380 / NVIDIA GeForce GTX 960 / Apple M1 chip 8-core GPU
  - **HDD :** 256 GB

8

LITERATURE REVIEW

X. Wang and L. Wang, "Green Routing Optimization for Logistics Distribution with Path Flexibility and Service Time Window," *2021 15th International Conference on Service Systems and Service Management (ICSSSM), Hangzhou, China, 2021*

Methods:

- **Path Flexibility**: Optimizes delivery routes by allowing flexibility in path choices, accounting for dynamic factors like traffic or road conditions.
- **Service Time Window**: Integrates delivery time constraints to ensure that products reach vendors or customers within the specified time windows for improved service.
- **Optimization Techniques**: Utilizes advanced optimization algorithms (such as genetic algorithms) to calculate the most efficient delivery routes while meeting all logistical constraints.

9

Continued

- **Environmental Impact Consideration:** Focuses on minimizing carbon emissions through optimized route selection, reducing the environmental footprint of logistics operations.

Modules to Use for Project

- **Path Flexibility Module:** Allows for route adjustments in real-time based on varying factors (traffic, weather, etc.) to ensure timely deliveries.
- **Service Time Window Module:** Ensures that all deliveries are made within specified time frames, enhancing customer satisfaction and vendor reliability.

Advantages

- **Optimized Delivery Routes**: Minimizes time and distance, reducing operational costs.
- **Environmental Sustainability**: Lower emissions by selecting the most eco-friendly paths.
- **Scalability**: Efficiently handles logistics for various vendors and locations, adaptable to growth.

10

Continued

- **Disadvantages**
  - **High Computational Demand**: Requires significant processing power for real-time optimization.
  - **Complex Algorithm Integration**: Involves combining various optimization techniques, which may be challenging to implement.
  - **Limited Real-Time Application**: May face challenges in providing immediate route changes due to computational constraints.

11

LITERATURE REVIEW

O. Triebe, H. Hewamalage, P. Pilyugina, N. Laptev, C. Bergmeir, and R. Rajagopal, "NeuralProphet: Explainable Forecasting at Scale," *arXiv preprint, Nov. 2021*

Methods

- **Hybrid Forecasting Model:** Combines Neural Networks and Classical Time-Series Models to capture complex patterns in demand data.
- **Explainable Forecasting**: Decomposes forecasts into trend, seasonality, and holiday effects, providing insights into what drives demand.
- **Scalability**: Designed to handle large-scale datasets efficiently, making it ideal for forecasting across multiple stores and items in the vendor collaboration platform.

12

Continued

Modules Utilized in the Project

- **Demand Forecasting Module**: Uses NeuralProphet to predict future sales at the item level for various stores.
- **Trend and Seasonality Decomposition**: Extracts and interprets the trend and seasonality components to optimize inventory and purchasing decisions.

Advantages

- **Accurate Forecasts**: Captures complex demand patterns for better stock management and purchasing decisions.
- **Scalable for Multiple Locations**: Efficiently handles demand forecasting across different store locations and multiple items.

13

Continued

Disadvantages

- **High Computational Demand**: NeuralProphet requires significant processing power, especially for large datasets with complex patterns.
- **Complex Model Tuning**: The hybrid nature of the model requires expertise to fine-tune and optimize for best results.

Use Case in Vendor Collaboration Platform

- **Inventory Optimization**: Accurate demand forecasts allow vendors to align stock levels with predicted sales, reducing stockouts and excess inventory.

14

LITERATURE REVIEW

S. Ben Jouida and S. Krichen, "A Genetic Algorithm for Supplier Selection Problem Under Collaboration Opportunities," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 34, no. 1, pp. 53-79, Jan. 2022. doi: 10.1080/0952813X.2020.1836031.

Methods

- **Genetic Algorithm (GA)**: Uses evolutionary techniques to optimize supplier selection by considering various factors like price, quality, and collaboration opportunities.
- **Supplier Collaboration**: Evaluates potential supplier collaboration to maximize benefits such as bulk order discounts, cost savings, and optimized logistics.
- **Fitness Function**: A fitness function is defined based on multiple criteria, including cost minimization, supplier reliability, and the potential for joint ordering with other vendors.

15

Continued

Modules Utilized in the Project

- **Supplier Selection Module**: Uses the Genetic Algorithm to identify the best suppliers based on multiple performance metrics and collaboration potential.
- **Collaboration Opportunity Module**: Optimizes joint orders between vendors to meet minimum order quantities (MOQ) for bulk discounts, leveraging genetic algorithms to explore the most cost-effective collaborative orders.

Advantages

- **Cost Optimization**: Selects suppliers that provide the best combination of cost, reliability, and collaborative potential.
- **Improved Supplier Collaboration**: Leverages collaboration opportunities between vendors to achieve bulk discounts and more favorable terms.

16

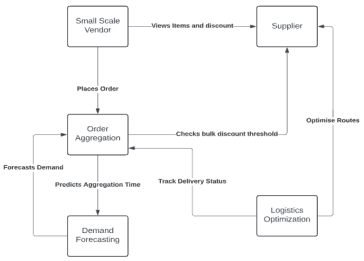
Continued

Disadvantages

- **Computational Intensity**: The genetic algorithm requires significant computational resources to evaluate many potential supplier combinations.
- **Complex Model Tuning**: The GA's parameters (e.g., population size, crossover rate) require fine-tuning to achieve optimal results.

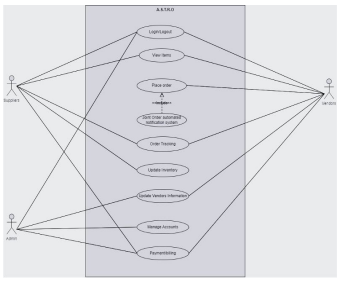
17

SYSTEM ARCHITECTURE



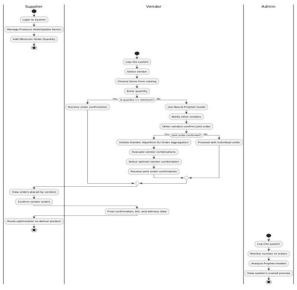
18

USE CASE DIAGRAM



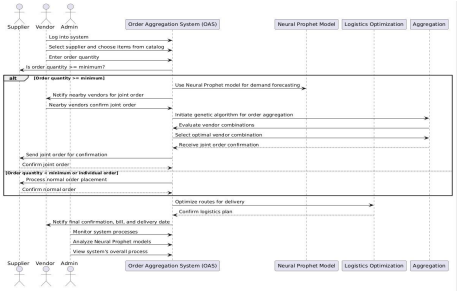
19

ACTIVITY DIAGRAM



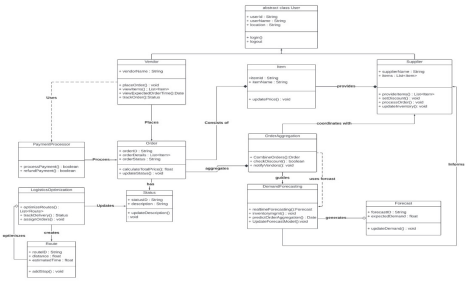
20

SEQUENCE DIAGRAM



21

CLASS DIAGRAM



22

Algorithms in Use-Route Optimization

- **Hybrid Demand Forecasting and Route Optimization Algorithm**

The Hybrid Demand Forecasting and Route Optimization Algorithm integrates demand forecasting with advanced route optimization techniques to enhance logistics efficiency. This algorithm ensures that vendor orders meet the bulk discount thresholds while optimizing delivery routes based on forecasted sales data.

Algorithm Steps

- **Define Bulk Discount Threshold**:
  - Establish the Minimum Order Quantity (MOQ) that must be reached for bulk discounts to apply.
  - Example: Set MOQ at 500 units.

23

Continued

- **Forecast Vendor Demand**:
  - Utilize NeuralProphet to predict the expected daily sales for each vendor.

  Example:
    - Vendor A: 100 units/day
    - Vendor B: 150 units/day
    - Vendor C: 200 units/day
- **Aggregate Vendor Orders**:
  - Collect initial orders from all participating vendors based on the forecast data.
  - Formulate a list of order combinations to achieve the MOQ.
- **Calculate Cumulative Demand**:
  - Sum the sales from all vendors to determine total demand.
  - Monitor cumulative sales daily to ensure MOQ is met:

24

Continued

- **Route Optimization Using Genetic Algorithm**:
  - **Initialize Routes**: Generate initial route combinations for deliveries based on vendor locations and demand.
  - **Fitness Evaluation**:
    - Evaluate routes based on factors like total distance, delivery time, and fuel costs.
    - Define a fitness score for each route based on these criteria
  - **Selection**:
    - Select the top-performing routes based on fitness scores.
- **Crossover and Mutation**:
  - Combine selected routes (crossover) to create new route combinations.
  - Introduce small changes (mutation) to explore potential improvements in route efficiency.

25

Continued

- **Final Route Selection**:
  - After several iterations, converge on the optimal route(s) that minimize costs while meeting the bulk order requirements.
  - Ensure that the selected route(s) accommodate delivery timing and vendor locations efficiently.
- **Performance Benefits:**
  - **Cost Reduction**: Minimizes transportation costs through optimized routing.
  - **Efficiency**: Ensures timely deliveries by efficiently managing routes based on demand forecasts.
  - **Scalability**: Adapts to changing demand patterns and vendor participation in real-time.

26

Algorithms In Use – Demand Forecasting

- **1. Define the Bulk Discount Threshold**
  - The supplier typically sets a minimum order quantity (MOQ) or a bulk discount threshold that needs to be met before a bulk order is placed.
  - Example: Supplier sets a bulk discount threshold of 500 units for a particular product.
- **2. Use Forecast Data for Sales**
  - Use the forecasted sales data from NeuralProphet to estimate how much each vendor is likely to sell over time.
  - Input: Forecasted sales from multiple vendors.
  - Forecast Data Example:
    - Vendor A is expected to sell 100 units/day.
    - Vendor B is expected to sell 150 units/day.
    - Vendor C is expected to sell 200 units/day.

27

Continued

- **3. Calculate Daily Cumulative Sales**
  - Based on the forecasted sales, you can calculate the cumulative sales of all participating vendors.
  - Cumulative Sales Example:
    - Day 1: 100 (A) + 150 (B) + 200 (C) = 450 units.
    - Day 2: 100 (A) + 150 (B) + 200 (C) = another 450 units.
  - After 2 days, the cumulative sales would be 900 units, crossing the 500-unit bulk threshold.
- **4. Estimate the Aggregation Time**
  - Once you have the daily cumulative sales forecasted, you can calculate when the bulk order threshold will be reached:

28

Continued

- **Expected Aggregation Time:**
  - In the example, the threshold of 500 units is expected to be reached in just over 1 day (since 450 units are sold on Day 1 and an additional 50 units on Day 2 would meet the threshold).

29

Algorithms In Use – Genetic Algorithm

Initialize Vendor Orders

- Start by gathering orders from multiple vendors based on their demand forecasting results. Each order includes quantity and expected sales.

Create Initial Population (Order Combinations)

- Generate an initial set of possible combinations of vendor orders that could meet the bulk order threshold. Each combination (solution) represents an aggregated order scenario.

Evaluate Fitness (Minimize Cost or Maximize Profit)

- For each order combination, evaluate its "fitness" by considering factors such as total cost, potential discounts, and logistics. Higher fitness means a better solution.

30

Continued

Selection and Crossover

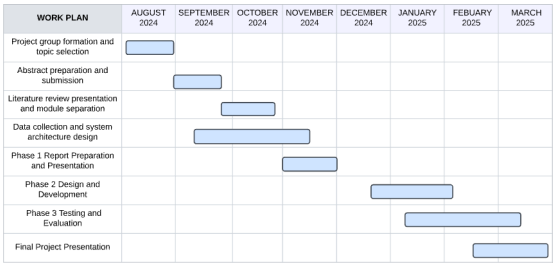
- Select the best-performing order combinations and combine them (crossover) to create new combinations. This step helps to find better solutions by mixing the strengths of different combinations.

Mutation and Final Solution

- Introduce small changes (mutation) to the order combinations to explore new possibilities. After several iterations, the algorithm converges to the most optimal aggregated order that maximizes vendor savings while meeting the bulk threshold.

31

Work Plan



32

Conclusion

- **Empowering Small Vendors**: Supports competitive edge through order aggregation and logistics optimization.
- **Key Achievements**: Accurate demand forecasting, eco-friendly routing, and cost savings via joint orders.
- **collaborative purchasing** :access to bulk discounts, improving supply chain efficiency.

References

- X. Wang and L. Wang, "Green Routing Optimization for Logistics Distribution with Path Flexibility and Service Time Window," *2021 15th International Conference on Service Systems and Service Management (ICSSSM), Hangzhou, China, 2021*
- O. Triebe, H. Hewamalage, P. Pilyugina, N. Laptev, C. Bergmeir, and R. Rajagopal, "NeuralProphet: Explainable Forecasting at Scale," *arXiv preprint, Nov. 2021*
- S. Ben Jouida and S. Krichen, "A Genetic Algorithm for Supplier Selection Problem Under Collaboration Opportunities," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 34, no. 1, pp. 53-79, Jan. 2022.

Appendix B: Vision, Mission, PO, PSO, and CO

RAJAGIRI SCHOOL OF ENGINEERING AND TECHNOLOGY (AUTONOMOUS)

Vision

To evolve into a premier technological and research institution, molding eminent professionals with creative minds, innovative ideas and sound practical skill, and to shape a future where technology works for the enrichment of mankind.

Mission

To impart state-of-the-art knowledge to individuals in various technological disciplines and to inculcate in them a high degree of social consciousness and human values, thereby enabling them to face the challenges of life with courage and conviction.

DEPARTMENT OF COMPUTER SCIENCE AND BUSINESS SYSTEMS

Vision

To evolve into a department of excellence in information technology by the creation and exchange of knowledge through leading-edge research, innovation and services, which will in turn contribute towards solving complex societal problems and thus building a peaceful and prosperous mankind.

Mission

To impart high-quality technical education, research training, professionalism and strong ethical values in the young minds for ensuring their productive careers in industry and academia so as to work with a commitment to the betterment of mankind.

Programme Outcomes (PO)

Engineering Graduates will be able to:

1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem Analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/Development of Solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct Investigations of Complex Problems: Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The Engineer and Society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and Sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and Team Work: Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project Management and Finance: Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments.

12. Life-long Learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Programme Specific Outcomes (PSO)

A graduate of the Computer Science and Business Systems Programme will:

- **PSO 1: Programming and Software Development Skills**
  Demonstrate ability to analyze, design, and implement software solutions incorporating various programming concepts.
- **PSO 2: Engineering Management and Collaboration**
  Comprehend professional, managerial, and financial aspects of business and collaborate on the design, implementation, and integration of engineering solutions.
- **PSO 3: Decision-Making and Analytical Techniques in Engineering and Business**
  Create, select, and apply appropriate techniques and business tools, including prediction and data analytics, for complex engineering activities and business solutions.

Course Outcomes (CO)

After successful completion of the course, the students will be able to:

- CO1: Model and solve real-world problems by applying knowledge across domains (Cognitive knowledge level: Apply).
- CO2: Develop products, processes, or technologies for sustainable and socially relevant applications (Cognitive knowledge level: Apply).
- CO3: Function effectively as an individual and as a leader in diverse teams and to comprehend and execute designated tasks (Cognitive knowledge level: Apply).
- CO4: Plan and execute tasks utilizing available resources within timelines, following ethical and professional norms (Cognitive knowledge level: Apply).
- CO5: Identify technology/research gaps and propose innovative/creative solutions (Cognitive knowledge level: Analyze).
- CO6: Organize and communicate technical and scientific findings effectively in written and oral forms (Cognitive knowledge level: Apply).

Appendix C: CO-PO-PSO Mapping

Mapping of Course Outcomes (CO) with Programme Outcomes (PO)

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO 1	2	2	2	1	2	2	2	1	1	1	1	2
CO 2	2	2	2		1	3	3	1	1		1	1
CO 3									3	2	2	1
CO 4					2			3	2	2	3	2
CO 5	2	3	3	1	2							1
CO 6					2			2	2	3	1	1

Mapping of Course Outcomes (CO) with Programme Specific Outcomes (PSO)

	PSO 1	**PSO 2**	**PSO 3**
CO 1	3	1	2
CO 2	3	3	2
CO 3		3	
CO 4		1	1
CO 5	1	1	1
CO 6		2	