RSET
RAJAGIRI SCHOOL OF
ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

Project Report/Seminar report On

Title

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

Bachelor of Technology

in

Name of the Programme

By

Name (UID)

Under the guidance of

Name of the Guide

Name of the Department
Rajagiri School of Engineering & Technology (Autonomous)
(Parent University: APJ Abdul Kalam Technological University)
Rajagiri Valley, Kakkanad, Kochi, 682039
July 2023

CERTIFICATE

This is to certify that the project report/seminar report entitled ***"Title"*** *is a bonafide record of the work done by* ***Student Name (UID)****, 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 "Name of the Programme" during the academic year 20XX-20XX.*

Project Guide	Project Co-guide	Project Co-ordinator
Designation	Designation	Designation
Dept. Name	Dept. Name	Dept. Name
RSET	RSET	RSET

Name of HoD
Designation
Dept. Name
RSET

ACKNOWLEDGMENT

I wish to express my sincere gratitude towards **Name**, Principal of RSET, and "Name of HoD", Head of the Department of "Name of the Department" for providing me with the opportunity to undertake my project, "Project Title".

I am highly indebted to my project coordinators, **Name(s)**, Designation, Department, for their valuable support.

It is indeed my pleasure and a moment of satisfaction for me to express my sincere gratitude to my project guide **Name of Guide** for his/her patience and all the priceless advice and wisdom he/she has shared with me. I also express my sincere thanks to my co-guide(s), **Name of Co-Guide** for his/her support. *(Edit the contents accordingly)*

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

Name of Student

Abstract

Small-scale vendors often face significant challenges when competing with large retail chains due to limited purchasing power, inefficient logistics, and restricted access to financial services. These challenges threaten their ability to manage demand fluctuations and make informed business decisions. Our project addresses these issues by developing a platform that empowers vendors through collaborative purchasing, optimized logistics, and data-driven decision-making, ultimately promoting economic resilience and sustainable growth in local communities.

The core of our project revolves around creating a robust platform that incorporates several key concepts, including collaborative purchasing, real-time demand management, and optimized logistics. The platform allows vendors to combine orders, thereby achieving bulk pricing and competitive discounts. It also includes algorithms for route optimization, reducing transportation costs and improving delivery efficiency. The platform further leverages data analytics to assess vendors' creditworthiness and provide tailored financial services, while offering insights on market trends and customer behavior to support strategic decision-making.

The primary goal of the project is to level the playing field for small vendors, enabling them to compete more effectively with larger retail chains. Our deliverables include a fully functional platform that facilitates vendor collaboration, an optimized logistics system that reduces operational costs, and an integrated financial service module that empowers vendors with the resources they need to grow. By achieving these goals, the project aims to strengthen local economies, enhance vendor competitiveness, and drive sustainable development.

Contents

List of Abbreviations

Acronym - Expansion

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 Optimization) platform addresses these challenges by providing a comprehensive solution that empowers small vendors through collaborative purchasing, logistics optimization, 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 optimization 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 Optimization) 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 optimization 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 often operate in environments characterized by intense competition, limited resources, and fluctuating market demands. These factors create a challenging landscape for maintaining profitability and achieving business growth. The primary problems ASTRO seeks to address are:

1. **Limited Purchasing Power:** Small vendors typically purchase goods in smaller quantities, resulting in higher per-unit costs compared to larger retailers who benefit from bulk purchasing discounts. This limitation affects their ability to offer competitive pricing and maintain healthy profit margins.

2. **Inefficient Logistics:** Managing supply chain logistics is a significant hurdle for small vendors. Without access to advanced logistics systems, vendors face challenges such as higher transportation costs, longer delivery times, and inefficient inventory management. These inefficiencies can lead to delays in restocking, increased operational expenses, and reduced customer satisfaction.

3. **Restricted Access to Financial Services:** Access to capital is a persistent issue for small vendors. Traditional financial institutions often perceive small-scale businesses as high-risk, resulting in limited access to credit and loans. This financial constraint restricts vendors' ability to invest in inventory, technology, and business expansion, further impeding their growth potential.

4. **Lack of Data-Driven Decision Making:** Large retail chains utilize data analytics to make informed business decisions, optimize operations, and understand market trends. In contrast, small-scale vendors typically lack the resources and expertise to collect, analyze, and leverage data effectively. This deficiency limits their ability to respond to market changes, forecast demand, and implement strategic business practices.

5. **Market Competition and Visibility:** Competing with large retail chains also means that small vendors often struggle with market visibility and brand recognition. Without extensive marketing resources, small vendors find it challenging to attract and retain customers, limiting their market share and growth prospects.

ASTRO aims to mitigate these problems by providing a comprehensive platform that enhances the operational capabilities of small-scale vendors. Through collaborative purchasing, logistics optimization, and data-driven financial services, ASTRO empowers vendors to overcome their inherent challenges, enabling them to compete more effectively with larger retail chains and sustain their businesses in a competitive marketplace.

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 Optimization:** 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. **Data-Driven Financial Services:** ASTRO offers financial services, including credit assessments and tailored loan options based on vendors' transaction data. This feature addresses the financial barriers faced by small vendors, providing them with the necessary capital to invest in their businesses and facilitate growth.

4. **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.

5. **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 optimization 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. **Offer Data-Driven Financial Services:**

   - **Objective:** Develop and integrate financial service modules that provide credit assessments and tailored loan options based on vendors' transaction history and financial needs.
   - **Outcome:** Enhanced access to capital, enabling vendors to invest in inventory, technology, and business expansion.

5. **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.

6. **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.

7. **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 optimization, 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, optimizing logistics, and providing access to financial services. 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 optimization, predictive analytics for demand forecasting, and data-driven financial services, 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 optimization 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. **Financial Inclusion and Accessibility:**

   - **Impact:** The platform's data-driven financial services are particularly relevant in promoting financial inclusion. By offering tailored credit assessments and loan options based on vendors' transaction data, ASTRO makes financial resources more accessible to small vendors who might otherwise be excluded from traditional financial systems. This inclusion supports business growth and reduces financial disparities.

6. **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.

7. **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.

8. **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 optimization, contributing to the overall strength and cohesion of the local vendor network.

9. **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.

10. **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.

11. **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.

12. **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.

13. **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.

14. **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.

15. **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

X. Wang and L. Wang, in their study titled *"Green Routing Optimization for Logistics Distribution with Path Flexibility and Service Time Window"*, presented at the 2021 15th International Conference on Service Systems and Service Management (ICSSSM) in Hangzhou, China, address optimizing logistics routes by focusing on minimizing environmental impact (especially CO_2 emissions) and allowing path flexibility based on real-time conditions. The study also considers service time windows to ensure timely deliveries.

Information and Methods

The study investigates two main factors: green routing, which minimizes environmental impact, and path flexibility, which allows choosing alternate routes based on real-time conditions. Additionally, service time windows are incorporated to ensure deliveries are made within specified time frames.

Optimization Model

The authors developed a multi-objective optimization model that considers:

- **Green routing objectives**: Minimizing CO_2 emissions.
- **Logistical objectives**: Minimizing delivery time, distance, and transportation costs.
- **Service time windows**: Ensuring deliveries happen within pre-set time intervals.

An improved genetic algorithm (GA) is proposed to solve this optimization problem. The algorithm uses evolutionary principles (selection, crossover, and mutation) to iteratively generate and evolve solutions.

Model Inputs

- **Delivery points**: Locations for each delivery.

- **Time windows**: Constraints on the time within which deliveries must occur.
- **Vehicle capacities**: The size and limitations of vehicles.
- **Road network data**: Information on available routes and traffic conditions.

Data Preparation and Datasets

Data was sourced from real-world logistics companies and road network databases, containing information on delivery points, road conditions, traffic patterns, and environmental factors. Data preprocessing includes handling missing values, outliers, and incorrect time windows.

Results

The genetic algorithm generated optimized routes that minimized both CO_2 emissions and total delivery time, while allowing flexibility in route selection. The model also met service time windows, ensuring customer satisfaction.

Advantages

1. **Environmental Impact Reduction**: Green routing reduces carbon emissions, contributing to sustainable logistics operations.
2. **Flexibility in Routing**: Path flexibility allows adaptation to real-time traffic conditions, road closures, and weather disruptions.
3. **Timeliness of Delivery**: Meeting service time windows improves operational efficiency and customer satisfaction.
4. **Multi-Objective Optimization**: Optimizes cost, time, and environmental impact, providing a comprehensive solution for logistics.

Disadvantages

1. **Computational Complexity**: The genetic algorithm can be computationally intensive, especially for large networks.
2. **Dependency on Traffic and Road Data**: Optimization accuracy depends on reliable traffic and road data.
3. **Limited Scalability**: May struggle to scale for large logistics systems with many variables.

A. Maroof, B. Ayvaz, and K. Naeem, in their study titled *"Logistics Optimization Using Hybrid Genetic Algorithm (HGA): A Solution to the Vehicle Routing Problem With Time Windows (VRPTW)"* published in IEEE Access, vol. 12, pp. 100245-100255, 2024, address the Vehicle Routing Problem with Time Windows (VRPTW). This research minimizes transportation costs while respecting delivery time windows.

Hybrid Genetic Algorithm (HGA)

The authors propose a Hybrid Genetic Algorithm (HGA) that combines a traditional genetic algorithm with local search methods like 2-opt and swap heuristics to improve solution quality and convergence speed, helping avoid local optima and ensuring a global solution.

Optimization Goal

The goal is to find optimal routes for a fleet of vehicles that minimizes total travel distance while meeting the time constraints of each delivery point.

Algorithm Components

- **Genetic Algorithm (GA)**: Selection, crossover, and mutation to explore and exploit the solution space.
- **Local Search (2-opt and Swap)**: Post-processing of GA solutions to optimize routes and remove inefficiencies.

Data Preparation and Datasets

Similar to Wang and Wang's study, the dataset includes vehicle capacities, service time windows, and road network data. Preprocessing ensures reasonable time windows and correct vehicle capacities.

Results

The HGA minimized total travel distance while meeting time windows and outperformed traditional genetic algorithms in terms of convergence speed and solution quality.

Advantages

1. **Improved Solution Quality**: Hybridization avoids local optima and improves solution quality.
2. **Handling Complex Constraints**: Effectively manages constraints like time windows and vehicle capacities.

3. **Computational Efficiency**: More efficient than basic GAs, suitable for large-scale problems.

4. **Adaptability**: Flexible for various VRPTW problems, including multiple depots and fleet sizes.

Disadvantages

1. **Local Optima Issues**: Despite improvements, careful tuning is required to avoid local optima.

2. **Complexity in Hybridization**: Balancing GA and local search adds complexity.

3. **Static Approach**: Does not account for real-time dynamic changes like traffic disruptions or sudden demand shifts.

Comparison and Integration

- **Green Routing** (Wang & Wang, 2021): Emphasizes sustainability through green routing, reducing environmental impact, a valuable feature for eco-conscious platforms.

- **Hybrid Genetic Algorithm** (Maroof et al., 2024): Ensures time-sensitive, cost-efficient deliveries, ideal for small-scale vendors requiring reliable logistics.

Conclusion

Combining green routing (Wang & Wang, 2021) and optimized delivery scheduling (Maroof et al., 2024) provides environmental sustainability and logistical efficiency, making it highly suitable for optimizing small-scale vendor logistics. Contents [1]

Here's a structured chapter in the form of **System Architecture** with sections and subsections corresponding to each component you've provided:

—

Chapter 3

System Architecture

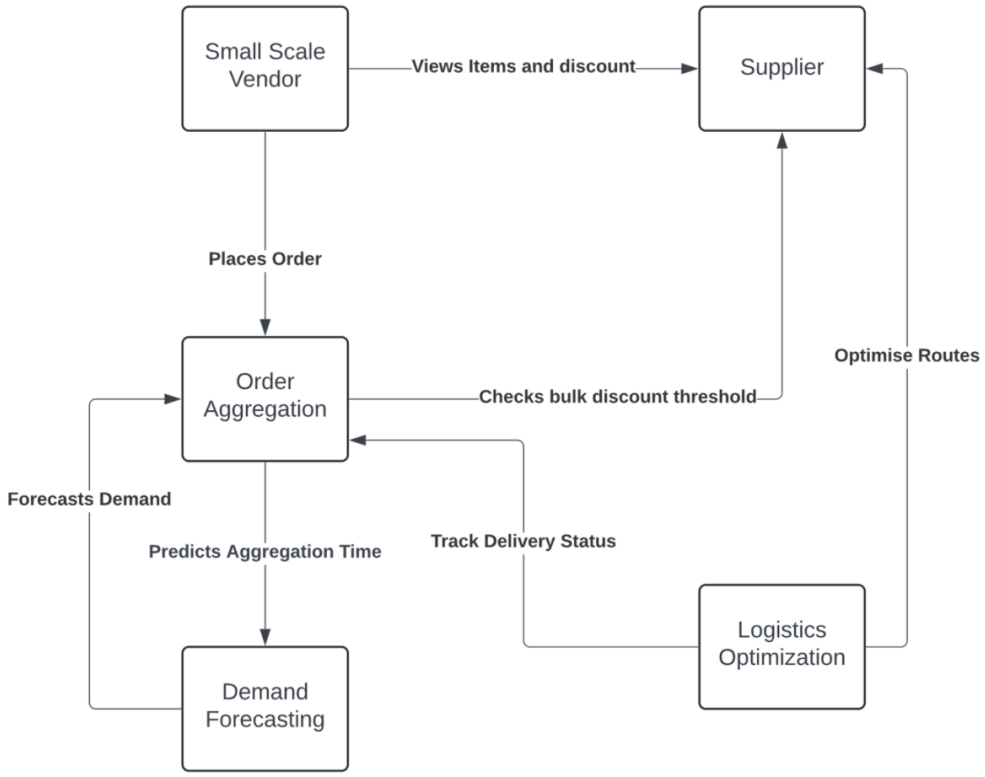

Figure 3.1: System Architecture of Vendor Collaboration Platform

Figure 3.1 illustrates the system architecture for a Vendor Collaboration Platform, designed to support small-scale vendors by facilitating bulk order discounts, demand forecasting, and logistics optimization. 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 optimization, 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 Optimization 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 Optimization through Green Routing:** The architecture emphasizes the logistics optimization 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 optimization 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 improving 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 Vendor Collaboration Platform. It provides a clear visual representation of the platform's workflow, highlighting key features such as order aggregation, demand forecasting, logistics optimization, 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 optimization.

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 our Vendor Collaboration 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 our 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, our platform empowers small-scale vendors with insights that drive collaborative efficiency and competitiveness.

3.4 Project Modules and Their Integration

Our Vendor Collaboration Platform comprises three core modules: Demand Forecasting, Order Aggregation, and Path Optimization. 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 Optimization Module

The Path Optimization 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 Optimization algorithms to design an optimal delivery route.

Since the joint order often involves multiple vendors located in different areas, the path optimization 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 How the Modules Work Together

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 Optimization**: 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 optimization module.

- **Path Optimization for Delivery**: The path optimization 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 optimization 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 optimization 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

In our Vendor Collaboration 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 our 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 Optimization**: Predicted delivery schedules align with logistics planning, enabling route optimization and efficient load distribution.

By integrating Neural Prophet's forecasts into our 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 our 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 How the Process Works

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:

   Chromosome 1010 : Vendor 1 and Vendor 3 are included, while Vendor 2 and Vendor 4 are excl

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 optimization and delivery scheduling, ensuring seamless fulfillment. By leveraging the Genetic Algorithm, our 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 Optimization Algorithm

Logistics optimization is a key component of our platform, ensuring efficient delivery of aggregated orders to vendors while minimizing transportation costs and environmental impact. In our project, we employ the Green Route Optimization 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 How Logistics Optimization Works

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 Optimization 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:

$$\text{Route: Supplier} \rightarrow \text{Vendor A} \rightarrow \text{Vendor B} \rightarrow \text{Vendor C} \rightarrow \text{Supplier}$$

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 optimization 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.

Optimization 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 Why Green Route Optimization Algorithm is Used

The Green Route Optimization Algorithm is specifically chosen for our 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 our 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 Optimization 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 optimization 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.

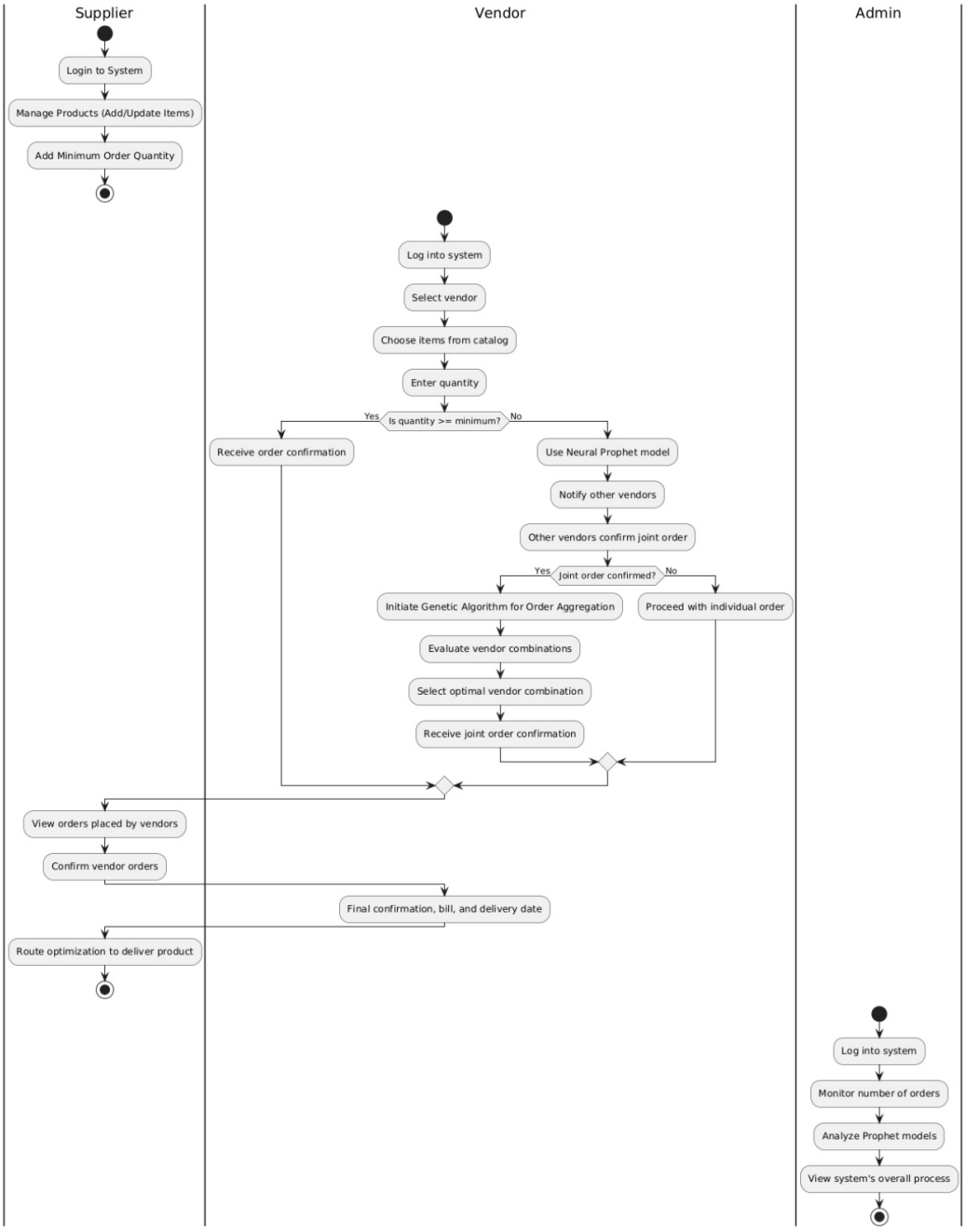

Figure 4.1: Activity Diagram

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 Optimization 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 optimization, 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.

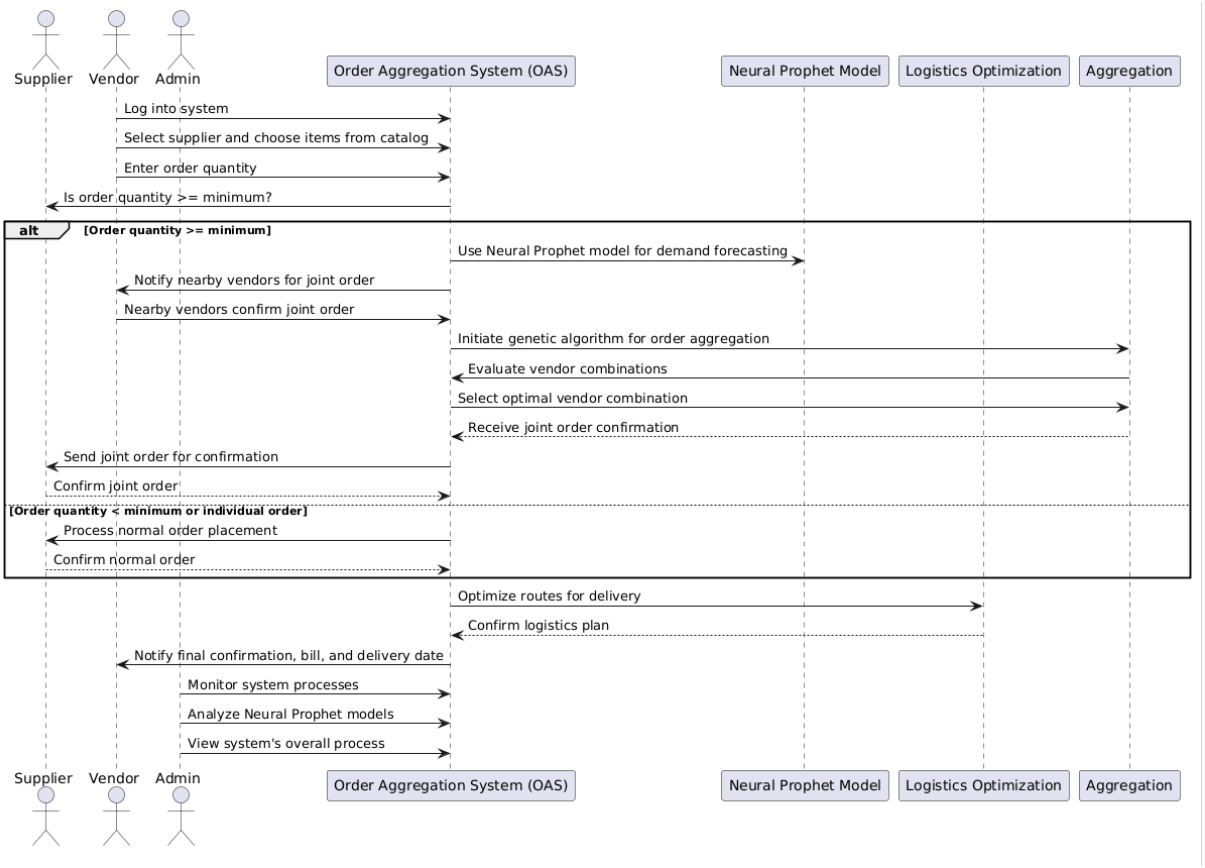

Figure 4.2: Activity Diagram

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 Optimization:** 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 ag-

gregation. 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 Optimization 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.

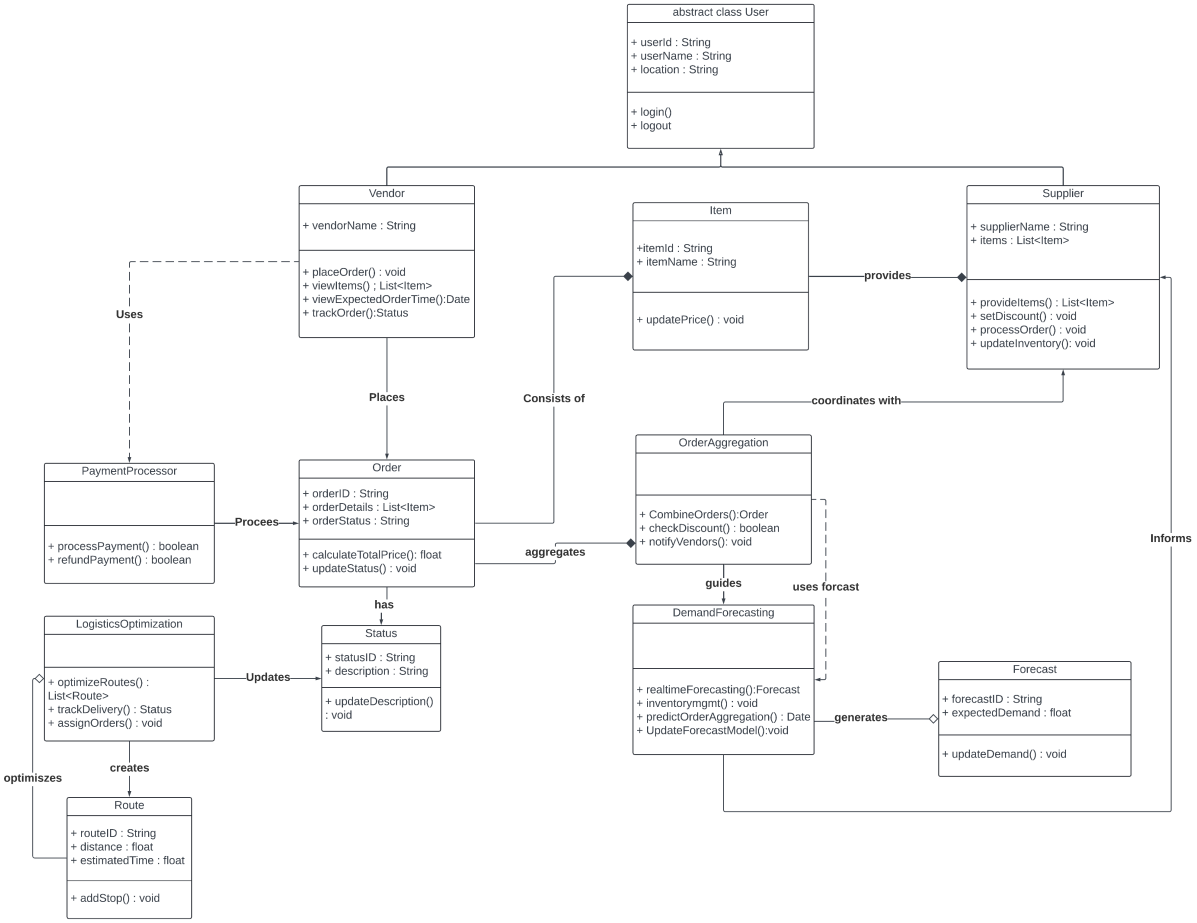

Figure 4.3: Class Diagram

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 update-

ForecastModel(). 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.

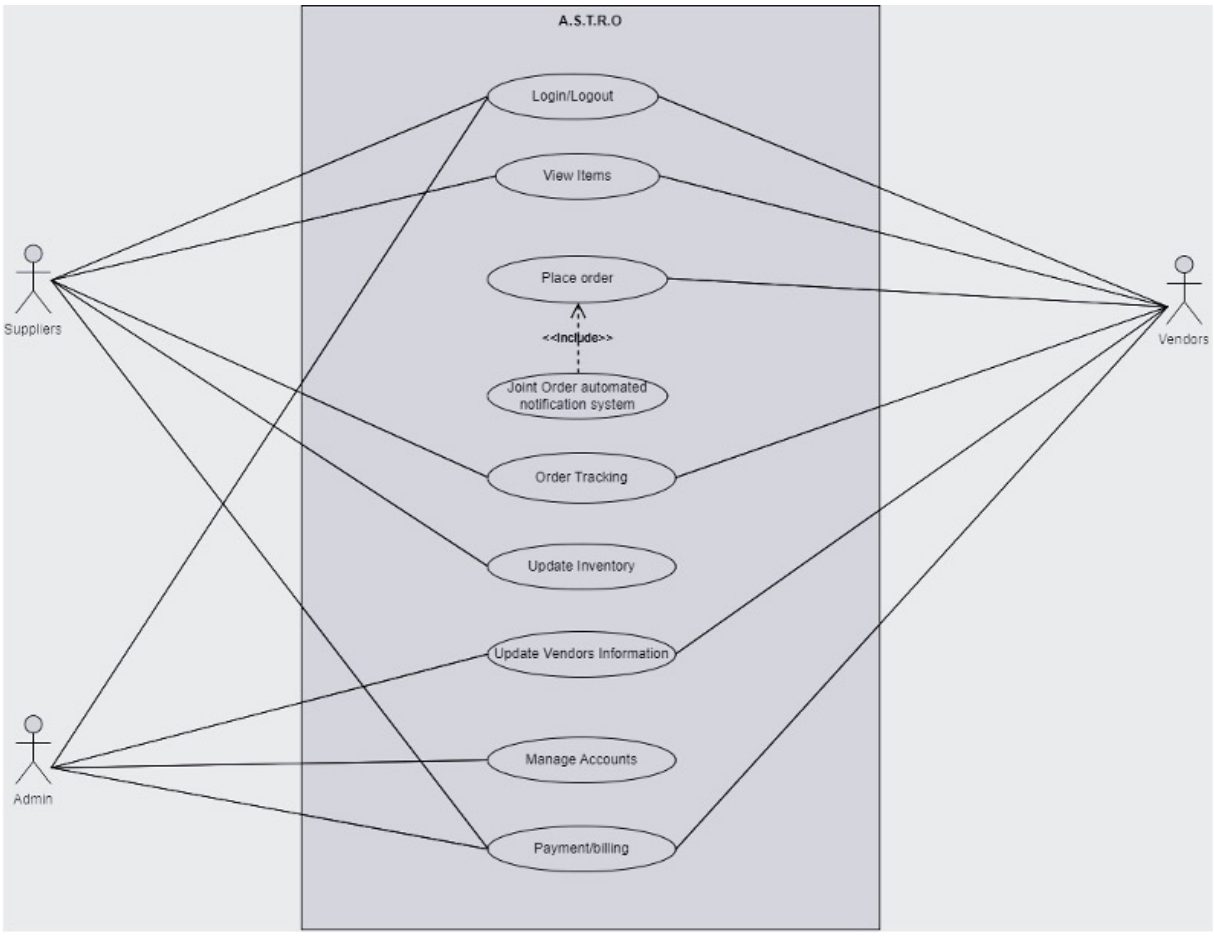

Figure 4.4: Use Case Diagram

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 A.S.T.R.O represents the system boundary, meaning all the use cases within this box are functionalities provided by the A.S.T.R.O 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 optimization 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 optimization 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 Optimization**: 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] Y. Xu, J. Zhang, Q. Zhang, and D. Tao, "Vitpose++: Vision transformer for generic body pose estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

List of Publications

1. All the list of publications should be in IEEE Journal format as given in the references.

2. Publication 1

3. Publication 2

Appendix A: Presentation

Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes

Vision, Mission, Programme Outcomes and Course Outcomes

Institute Vision

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

Institute 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 Vision

Department Mission

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 the 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 with the engineering community and with society at large. Be able to comprehend and write effective reports 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)

Course Outcomes (CO)

Appendix C: CO-PO-PSO Mapping

CO - PO Mapping

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

CO - PSO Mapping

CO	PSO 1	PSO 2	PSO 3
1			
2			
3			
4			
5			

Justification

Mapping	Justification
CO1 - PO1	Reason
CO2 - PO2	Reason