Vaasan yliopisto
UNIVERSITY OF VAASA

Riaz Mahmud

Towards Sustainable Food Logistics: An Open-Source Web Application for Optimizing Transportation Scheduling and Vehicle Routing

School of Technology and Innovations
Master’s thesis in Industrial Management
Master's Programme in Industrial Systems Analytics (ISA)

Vaasa 2024

UNIVERSITY OF VAASA			
School of Technology and Innovations			
Author:	Riaz Mahmud		
Title of the thesis:	Towards Sustainable Food Logistics: An Open-Source Web Application for Optimizing Transportation Scheduling and Vehicle Routing		
Degree:	Master of Science in Technology		
Discipline:	Industrial Systems Analytics (ISA)		
Supervisor:	Petri Helo		
Year:	2024	**Pages:**	75

ABSTRACT :

The logistics industry faces significant challenges in optimizing transportation scheduling and vehicle routing, particularly in reducing operational costs, maintaining efficiency, and minimizing environmental impact. This thesis presents RouteShaper, a web-based tool designed to address these challenges, specifically within the context of Honkajoki Oy, a leading animal by-product processor in Finland. This application uses the VRP CLI Python package as the optimization engine, which is built on top of heuristic algorithms. Other key technologies employed in this research include the Python-based Django framework for backend development and the JavaScript-based React framework for frontend development.

The transportation operation at Honkajoki Oy is defined using a Multi-Depot Vehicle Routing Problem with Time Window model, and a novel parameter, Freshness Penalty, is introduced to measure the impact of delay on product quality during transit. Using data analytics and an intuitive reporting dashboard, the RouteShaper tool empowers logistics managers and stakeholders with tangible insights to enhance their operational efficiency. The practical implementation of this tool at Honkajoki Oy shows substantial improvement in reducing costs and traveling distance by 16.27% compared to manual scheduling. The result further shows a 26.91% reduction in overall traveling time. These enhancements directly contribute to efficient use of resources and reduced emissions for a positive environmental impact, particularly when there are a large number of delivery or pickup tasks involved.

This research contributes to the academic and practical fields by demonstrating the effectiveness of integrating open-source technologies, advanced optimization algorithms, and data analytics in logistics management. It underscores the importance of technological innovation in achieving operational efficiency and sustainability. Future work will focus on developing a new variant of the vehicle routing problem incorporating the Freshness Penalty constraint and exploring broader applicability across different industries.

KEYWORDS: Vehicle Routing Problem (VRP), Logistics Optimization, Transportation Scheduling, Sustainable Logistics, Web Application.

Contents

Figures

Tables

Abbreviations

VRP: Vehicle Routing Problem
CVRP: Capacitated Vehicle Routing Problem
VRPTW: Vehicle Routing Problem with Time Windows
VRPPD: Vehicle Routing Problem with Pickup and Delivery
MDVRP: Multi-Depot Vehicle Routing Problem
FP: Freshness Penalty
ERD: Entity Relationship Diagram
API: Application Programming Interface

1 Introduction

In the current globalized and competitive market, the logistics industry has always been seeking new ways to enhance efficiency, reduce costs, and environmental impacts. The transportation of goods, a core component of logistics operations, is particularly challenged by rising fuel prices, regulatory demands for lower emissions, and the need for timely deliveries. There is an increasing need for innovative solutions that utilize advanced technologies to optimize transportation scheduling and vehicle routing to address these issues. This chapter provides a comprehensive background to the study, clearly defining the problem statement, outlining the objectives, and formulating the research questions. Furthermore, it highlights the significance of the study both in academic knowledge and practical advancements in logistics management. Through this introductory framework, this chapter aims to establish a clear context for the research and underscore the importance of developing a robust, efficient, and sustainable logistics optimization tool.

1.1 Background

Logistics operations play a crucial role in the global economy, ensuring the efficient movement of goods from producers to consumers. However, this sector faces significant challenges, including rising fuel costs, increasing environmental concerns, and the need for enhanced operational efficiency. As businesses strive to meet these challenges, the adoption of advanced technologies for optimizing transportation scheduling and vehicle routing has become increasingly important.

1.1.1 Traditional Logistics Operations and Optimization

Traditional logistics operations often depend on manual planning and decision-making processes, which can be time-consuming and prone to errors. Moreover, incorporating different factors such as traffic congestion, variable fuel prices, and delivery time windows in manual scheduling remains a challenge. In addition to that, there is an increasing pressure on emissions reduction from the policy makers. The European Commission has set the goal of reducing 90% emissions related to transportation by 2050 (European Commission, 2021). Manual planning and scheduling are unable to address such critical issue.

Logistics optimization involves the use of sophisticated algorithms and software tools to enhance the efficiency of supply chain operations. This includes optimizing delivery routes, scheduling transportation activities, and managing vehicle fleets. Effective logistics optimization can lead to substantial cost savings, improved customer satisfaction, and reduced environmental impact.

According to Li et al. (2022), the global food system requires approximately 22 trillion tonne-kilometres[1] of transportation annually. It accounts for around 20% of CO2 emission related to the food ecosystem. Studies show that, utilizing computer-based transportation scheduling and planning, companies can save up to 20 percent of their transportation cost, CO2 and greenhouse gas emissions can be reduced from 5 percent to 37 percent (Moghdani et al., 2020).

1.1.2 Advances in Technology for Logistics Optimization

Recent advancements in technology have provided new opportunities for addressing the challenges faced by the logistics industry. Technologies such as metaheuristic algorithms, artificial intelligence (AI), machine learning (ML), and geographic information systems (GIS) are increasingly being integrated into logistics optimization tools. These technologies enable more accurate demand forecasting, dynamic route planning, and real-time tracking of vehicles and shipments.

---

[1] tonne-kilometres is defined as the transportation of one tonne over one kilometre (Wikipedia, 2024).

Wikimedia Foundation. (2024, July 28). *Units of measurement in Transportation*. Wikipedia. https://en.wikipedia.org/wiki/Units_of_measurement_in_transportation

For instance, the use of AI and ML algorithms in vehicle routing problems (VRP) allows for the consideration of multiple constraints, such as delivery time windows, vehicle capacities, and traffic conditions. Bogyrbayeva et al. (2024) summarized a list of research in their survey that incorporated AI and ML algorithms to solve VRP for more efficient and flexible route planning.

1.1.3 Emergence of Web-Based Logistics Applications

Web-based applications have further empowered the logistics industry by providing accessible, scalable, and user-friendly solutions for logistics management. These applications use cloud computing, real-time data analytics, and mobile technologies to offer comprehensive logistics optimization services.

Web-based logistics applications enable businesses to manage their transportation activities from any location, using any device with internet connectivity. This accessibility improves operational flexibility and enables more responsive decision-making. According to a report by Grand View Research (2023), the global market for route optimization software is expected to grow at a compound annual growth rate (CAGR) of 14.4% from 2024 to 2030, driven by the increasing adoption of cloud-based logistics solutions.

1.1.4 Development of the Web Application for Optimizing Transportation Scheduling and Vehicle Routing

In response to the need for more efficient and sustainable logistics operations, this thesis focuses on the development of RouteShaper[2], an open-source web application designed to optimize transportation scheduling and vehicle routing. The application uses the VRP CLI Python package for solving complex vehicle routing problems, incorporating various constraints such as delivery time windows, vehicle capacities, and customer demand (Builuk, 2023). The backend of the application is developed using Django, a high-level

[2] https://routeshaper.live/

Python web framework, while the frontend is built with React, a JavaScript library for building user interfaces. More details are provided in chapters 03 and 04.

This web application aims to provide logistics managers with a powerful tool for enhancing the efficiency of their operations, reducing costs, and minimizing environmental impact. By leveraging advanced heuristic algorithms and data analytics, the application addresses the key challenges faced by the logistics industry and supports sustainable logistics practices.

1.2 Problem Statement and Research Questions

There are a number of open-source applications and tools available in the literature for solving the vehicle routing problem (Erdoğan, 2017; Silva et al., 2023; Stančić et al., 2023; Zhou et al., 2018). However, very few addresses industry-specific challenges, particularly in the food industry. Additionally, most of these open-source tools require substantial technical knowledge to implement and use, as they are often provided as packages or libraries without an intuitive user interface. While some commercial applications offer user-friendly interfaces, they tend to charge a hefty amount to use their services. Considering both the technical and financial constraints, it is very challenging for small and medium sized enterprises (SMEs) to adopt these tools. Therefore, there is a need for an open-source route optimization and scheduling tool with an intuitive user interface that requires minimal technical knowledge to implement and can be used by general practitioners. **The purpose of this work is to develop an open source based web application for efficient vehicle route planning and scheduling.**

This problem statement leads to a few sub-questions:

- How can advanced algorithms be effectively integrated into a web-based application to optimize vehicle routing and scheduling?
- What impact does the optimized routing and scheduling have on operational costs and efficiency?

- How to quantify the impact of delays in transportation on the quality of perishable products?

By addressing the above stated research problem along with subsequent research questions, this study aims to develop a robust solution to improve the efficiency and sustainability of logistics operations in the food industry.

1.3 Objectives of the Study

The study aims to develop and evaluate a web-based application, RouteShaper, designed to optimize transportation scheduling and vehicle routing in logistics operations. By leveraging advanced heuristic algorithms, data analytics, and user-friendly interfaces, the study intends to address key challenges in logistics management, including inefficiency, high operational costs, and environmental impact. The objectives of the study are articulated as follows:

Objective 1: Development of a Web-Based Application

The primary objective is to design and develop a web-based application that integrates advanced vehicle routing algorithms and data analytics. The application will utilize open source technologies such as Django for backend development, React for frontend development, and the VRP CLI Python package for solving vehicle routing problems.

Objective 2: Case Study and Efficiency Improvement Evaluation

Cooperating with Honkajoki Oy, a leading processor of animal by-products in Finland, a case study will be carried out. The aim of the case study is to evaluate the impact of the web application on logistics efficiency by measuring key performance indicators (KPIs) such as travel distance, delivery time, and cost. This involves conducting comparative analysis with traditional logistics operations to quantify the improvements achieved through optimization.

Objective 3: Contribution to Academic and Practical Knowledge

To contribute to the academic and practical knowledge in the field of logistics optimization by publishing the findings through peer-reviewed article and presentations. This includes sharing insights on the application of advanced algorithms and data analytics in logistics.

By addressing these objectives, the study aims to provide a comprehensive solution to the challenges faced by the food industry for transportation of their goods. The development of a web-based application for optimizing transportation scheduling and vehicle routing is expected to improve operational efficiency, reduce costs, and promote sustainability. Through rigorous evaluation, the study will ensure that the application meets the needs of its users and contributes to the advancement of logistics management practices in the food industry.

1.4 Significance and Limitations of the Study

Despite having many VRP solver in the literature, this study's significance lies on focusing on usability and user satisfaction, the study ensures that the application is practical and beneficial for its intended users, including logistics managers and other stakeholders. A user-friendly interface reduces the learning curve and facilitates higher adoption rates, ensuring that the technology delivers its full potential in real-world settings. By optimizing routing and scheduling, this study also aims to enhance operational efficiency, reduce cost, and empower sustainable practices in food transportation.

As this study relies on open-source tools and techniques, it may not address all major issues in food transportation due to the limitations of existing technologies. Further studies may be needed to solve more industry specific issues.

1.5 Structure of the Thesis

This thesis is structured to systematically present the research conducted on the development and evaluation of a web-based application for optimizing transportation scheduling and vehicle routing. The following sections provide a comprehensive overview of each chapter.

Chapter 1: Introduction

The introductory chapter sets the foundation for the research by providing a background to the study, outlining the problem statement, defining the research objectives, formulating the research questions, and highlighting the significance of the study. This chapter establishes the context and rationale for the research, ensuring a clear understanding of its purpose and importance.

Chapter 2: Literature Review

This chapter reviews existing literature related to transportation optimization, advanced routing algorithms, and the use of web-based applications in logistics management. It identifies gaps in current research and establishes the theoretical framework for the study.

Chapter 3: Design and Development of the RouteShaper Web Application

This chapter outlines the research design and methodology used to develop and evaluate the RouteShaper web application. It describes the system architecture, optimization model, backend and frontend development processes, version management, and testing strategies.

Chapter 4: System Implementation and Evaluation

This chapter provides a detailed description of the RouteShaper implementation within the case company Honkajoki Oy, including an overview of the application, its features and functionalities. The case study analyzes the results obtained from the RouteShaper implementation, evaluates its performance, and comparison with existing solutions. It also addresses the sustainability impact of the solution.

Chapter 5: Conclusion and Future Work

The concluding chapter summarizes the key findings of the research, discusses its contributions to the field of logistics optimization, and suggests areas for future research.

By following this structured approach, the thesis aims to provide a comprehensive, clear, and detailed information of the research conducted, ensuring that the findings and contributions are accessible and useful to both academic and professional audiences.

2 Literature Review

This chapter provides a comprehensive review of the literature relevant to the development and optimization of transportation scheduling and vehicle routing systems. By examining the principles of sustainable logistics, the complexities and solutions associated with vehicle routing problems (VRP), the role of open-source software in logistics, and the advanced technologies employed in this field, this chapter sets the stage for understanding the foundational elements that support this study. This literature review aims to identify current challenges, explore innovative solutions, and highlight the potential benefits of integrating advanced algorithms and data analytics in logistics operations. Through this detailed exploration, the chapter aims to establish a theoretical framework that underpins the subsequent development and implementation of a web-based application for optimizing logistics processes.

2.1 Transportation Scheduling and Vehicle Routing Problem (VRP)

The Vehicle Routing Problem (VRP) represents a fundamental issue in logistics and transportation management. It involves determining the optimal set of routes for a fleet of vehicles to service a set of customers while minimizing the total cost of transportation. This problem, first introduced by Dantzig and Ramser in 1959 as the "Truck Dispatching Problem," has evolved significantly over the years to incorporate a variety of real-world constraints and objectives (Dantzig & Ramser, 1959).

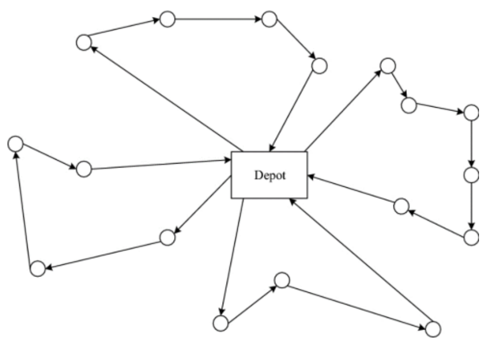


Figure 1. Basic Vehicle Routing Problem (Zhang et al., 2021).

Mathematically, the basic vehicle routing problem can be formulated as follows (Alves et al., 2021).

Let us define $N = \{0,1,2, \ldots \ldots, n\}$ is the set of all locations (customer and depot), where 0 is the location of depot. $K = \{1,2, \ldots \ldots, m\}$ is the set of all vehicles.

Now we define,

$c_{ij} = Cost\ (distance\ or\ time)\ from\ i\ to\ j$

$d_i = Demand\ of\ customer\ i$

$Q = Capacity\ of\ each\ vehicle$

Decision variable x_{ij}^k, u_i are defined as:

$$x_{ij}^k = \begin{cases} 1, if\ vehicle\ k\ visits\ j\ after\ i \\ 0, Otherwise \end{cases}$$

$u_i = Load\ of\ the\ vehicle\ after\ serving\ customer\ i$

Now, the objective function is to minimize the total cost (distance or time) while travelling all the locations.

$$i.e., \min C = \sum_{k=1}^{m} \sum_{i=0}^{n} \sum_{j=0}^{n} c_{ij}\, x_{ij}^k \tag{1}$$

Subject to the following constraints.

$$\sum_{k=1}^{m} \sum_{j=0}^{n} x_{ij}^k = 1, \forall i \in N/\{0\} \tag{2}$$

$$\sum_{j=0}^{n} x_{ij}^k = \sum_{j=0}^{n} x_{ji}^k, \forall i \in N, \forall k \in K \tag{3}$$

$$\sum_{j=0}^{n} x_{0j}^k = 1, \forall k \in K \tag{4}$$

$$\sum_{i=0}^{n} x_{i0}^k = 1, \forall k \in K \tag{5}$$

$$u_i - u_j + Qx_{ij}^k \leq Q - d_j, \forall i, j \in \frac{N}{\{0\}}, \forall k \in K, i \neq j \quad (6)$$

$$d_i \leq u_j \leq Q, \forall i, j \in N/\{0\} \quad (7)$$

$$\sum_{i=0}^{n} d_i \sum_{j=0}^{n} x_{ij}^k \leq Q, \forall k \in K \quad (8)$$

$$x_{ij}^k \in \{1, 0\}, \forall k \in K \quad (9)$$

The constraints, (2) ensures each customer is visited exactly once, (3) ensures that if a vehicle arrives at a customer, it must leave from there. In addition to that, constraints (4) and (5) ensure that each vehicle starts and ends its route at the depot, (6) and (7) eliminate subtours, (8) ensures the vehicle's capacity is not exceeded. Finally, constraint (9) enforces the binary nature of the routing decisions.

VRP Variants and Complexity

The basic VRP can be extended to various complex forms to better reflect real-world scenarios. Some of the most studied variants include the Capacitated VRP (CVRP), VRP with Time Windows (VRPTW), VRP with Pickup and Delivery (VRPPD), and the VRP with Multiple Depots (MDVRP). Each variant introduces additional constraints that make the problem more representative of practical logistics challenges.

Abbreviations	Definition	Abbreviations	Definition
VRP	Vehicle routing problem	GVRP	Green VRP
VRPTW	VRP with time windows	HFVRP	VRP with heterogeneous fleets
CVRP	Capacitated VRP	MDVRP	Multi-depot VRP
EV	Electric vehicle	TDVRP	Time-dependent VRP
ECV	Electric commercial vehicle	TDVRPTW	Time-dependent VRP with time widows
EVRP	Electric VRP	TWAVRP	Time window assignment VRP
EVRPTW	Electric VRP with time widows	VRPSD-PDC	VRP with stochastic demands and probabilistic duration constraints
EVRPTW-SP	EVRPTW at most a single (S) recharge per route, and partial (P) battery recharges are possible	VRP-REP	VRP repository

Figure 2. List of VRP Variants (Tan & Yeh, 2021).

A brief summary of some of the well-studied variants of VRP is described in the following section.

- **Capacitated VRP (CVRP):** In CVRP, each vehicle has a limited carrying capacity, and the objective is to ensure that the total load on each vehicle does not exceed its capacity. This variant is crucial for industries where the volume or weight of goods plays a significant role in logistics planning (Toth & Vigo, 2014).
- **VRP with Time Windows (VRPTW):** This variant adds the constraint that each customer must be serviced within a specific time window. VRPTW is particularly important in urban logistics and perishable goods delivery, where timing is critical (Solomon, 1987).
- **VRP with Pickup and Delivery (VRPPD):** In VRPPD, goods need to be picked up from certain locations and delivered to others. This variant is common in courier services and reverse logistics (Savelsbergh & Sol, 1995).
- **Multi-Depot VRP (MDVRP):** This variant involves multiple depots, and the objective is to determine the optimal routing strategy that considers different starting and ending points for vehicles. MDVRP is applicable in large-scale logistics operations where goods are distributed from multiple warehouses (Cordeau et al., 1997).

Computational Complexity and Algorithms for Solving VRP

VRP is classified as an NP-hard problem, meaning that the time required to solve it increases exponentially with the size of the problem. This computational complexity has led to the development of various heuristic and metaheuristic algorithms to find near-optimal solutions in a reasonable timeframe. Different types of algorithms for solving vehicle routing problem are presented in the following figure.

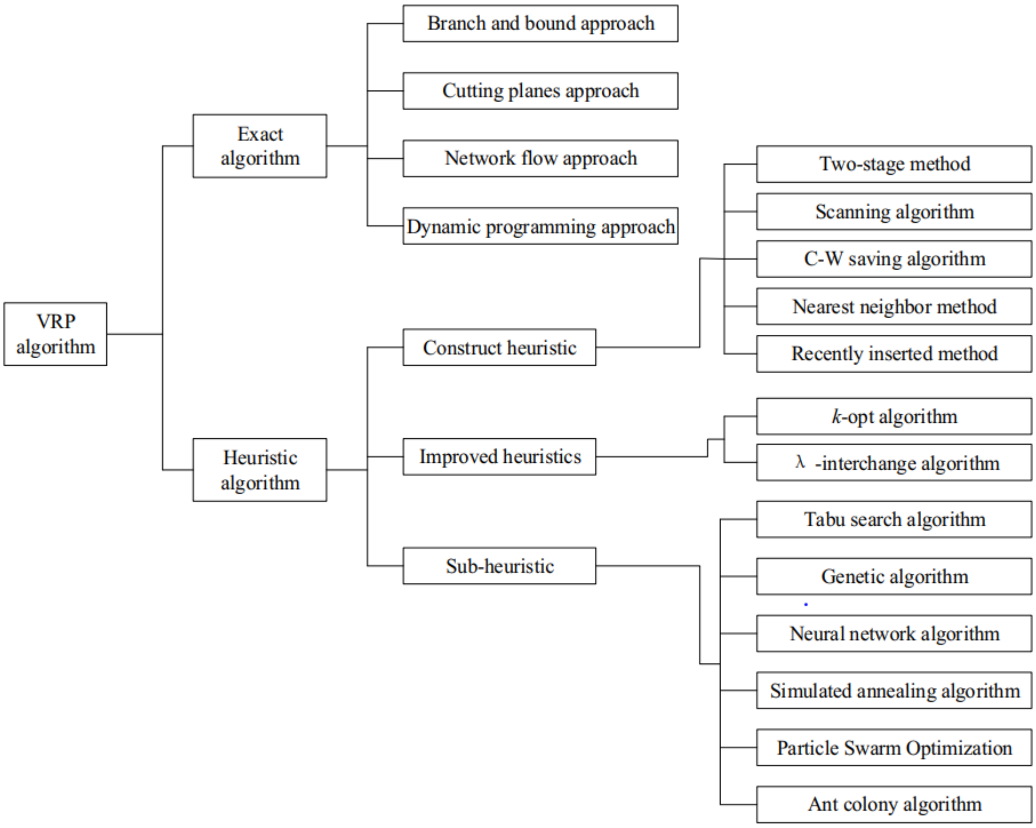


Figure 3. Classifications of VRP Algorithms (Zhang et al., 2021).

- **Exact Algorithms:** These include methods like branch-and-bound, branch-and-cut, and dynamic programming, which guarantee an optimal solution but are computationally expensive and impractical for large-scale problems (Laporte, 1992).
- **Heuristic Algorithms:** These algorithms provide good solutions within a reasonable timeframe by using rules of thumb and intuitive strategies. Common heuristics include the Nearest Neighbor, Savings Algorithm, and Clarke-Wright Algorithm (Clarke & Wright, 1964).
- **Metaheuristic Algorithms:** These algorithms improve upon heuristic methods by employing advanced strategies inspired by natural processes. Examples include Genetic Algorithms (GA), Simulated Annealing (SA), Ant Colony Optimization (ACO), and Tabu Search (TS). Metaheuristics are particularly useful for large and complex VRP instances (Dorigo & Stützle, 2004).

Applications in Food Logistics

In the context of food logistics, VRP is critical due to the perishable nature of food products and the need for timely deliveries. Efficient vehicle routing can reduce wastage, ensure food safety, and minimize transportation costs, contributing to overall sustainability. Studies have shown that optimizing vehicle routes in food distribution networks can significantly maintain quality and reduce greenhouse gas emissions (Awad et al.,2020).

2.2 Optimization Software in Logistics Innovation and Efficiency

Web based optimization software and tools are empowering businesses with fast and efficient solutions for complex vehicle routing problems. Along with many open-source software and tools, there exist a number of commercial solutions for route planning and optimization. Though commercial software offers intuitive and user-friendly interfaces and reporting, often those are associated with high prices. In contrast, open-source tools offer highly customizable solutions at no price. But open-source tools sometimes lack user friendly interfaces, and it may require to have solid technical background to implement open-source solutions.

Overview of Open-Source Tools for VRP

The emergence of open-source software has revolutionized the logistics industry by providing affordable, flexible, and scalable solutions for complex problems like VRP. Open-source tools enable companies of all sizes to leverage advanced optimization techniques without the high costs associated with proprietary software. Several open-source tools have been developed to address VRP and its variants, each with unique features and capabilities. A brief description of some of these tools are provided below.

- **Open Door Logistics Studio (ODL Studio):** ODL Studio is a desktop application designed for solving logistics problems. It uses the GraphHopper routing library and the jsprit optimization toolkit, both of which are open-source. ODL Studio allows users to import data in Excel format, making it accessible for users with varying levels of technical expertise. It supports a wide range of VRP variants, including CVRP, VRPTW, and VRPPD (ODL Studio, 2014).

- **VRP Spreadsheet Solver:** This tool integrates with Microsoft Excel to provide a user-friendly interface for solving VRP. It supports multiple VRP variants and uses advanced algorithms to find near-optimal solutions. The use of Excel as a platform makes it highly accessible and easy to integrate with other business processes (Erdoğan, 2017).
- **GraphHopper and jsprit:** GraphHopper is a Java-based routing engine that provides fast and efficient route calculations. It integrates seamlessly with jsprit, an optimization library for solving VRP. These tools are highly customizable and have been used in various applications, from small-scale logistics to large enterprise solutions (GraphHopper, 2022).
- **Google OR-Tools:** Google OR-Tools is a powerful open-source optimization software by Google. It is customizable, highly flexible, and can be integrated with different programming languages including C++/C#, Java, and Python. This tool can be used to solve complex vehicle routing problems both from academic and commercial context (Google, 2010).
- **VRP-CLI:** VRP CLI is a Python package designed and developed by Builuk (2023). It is one of the most recent Python based VRP solvers which is capable of addressing various types of VRP including CVRP, VRPTW, VRPPD, and MDVRP, etc. This package utilizes a hybrid approach of combing multiple heuristic algorithms that enables VRP CLI to address different complex scenarios. In this study, we utilized VRP CLI package to develop the RouteShaper web application.

Benefits of Open-Source Solutions

Unlike proprietary and commercial solutions, open-source solutions are freely available for everyone. This cost-effective nature helps in reducing the financial barrier for small and medium-sized enterprises (SMEs) to adopt advanced technological solutions. Open-source projects often have more modular designs that allow for greater flexibility and targeted improvements.

Additionally, open-source projects often benefit from knowledge sharing and voluntary contribution from different contributors. This collaborative effort promotes innovation for the software's improvement and ensures that the solution remains up-to-date with the latest advancements (von Krogh & von Hippel, 2006).

Contemporary Transportation and Logistics Solutions

Recent advances in vehicle routing, transportation, and logistics have significantly focused on integrating innovative technologies to enhance efficiency, reduce costs, and improve customer satisfaction. In a recent study, Alnaggar et al. (2021) presented a systematic review of modern crowdsourced delivery systems like UberEats, Amazon Flex, DoorDash, and Postmates. They highlighted three major categories of scheduling processes adopted by these platforms, including self-scheduling, hybrid and centralized scheduling, and en-route matching. Parallel to crowdsourced delivery systems, there is an emerging trend of utilizing unmanned aerial vehicles (UAVs) for last-mile deliveries (X. Li et al., 2024; Y. Li et al., 2022).

To solve contemporary VRP problems, Bozanta et al. (2022) proposed a reinforcement learning (RL) based framework for dynamic pickup and delivery problems in food delivery services, utilizing approaches like Double Deep Q-Networks (DDQN). Further study shows that reinforcement learning-based solutions can contribute to up to 50% higher fleet utilization in modern-day last-mile food delivery (Mehra et al., 2024). These advancements demonstrate the significant impact of leveraging technology and data-driven approaches to address complex routing and delivery challenges in the transportation and logistics sector.

Commercial Route Optimization Software

In addition to the above-mentioned open-source tools, there are some commercial solutions also available in the market for routing and scheduling. Though most of these tools offer user friendly interfaces and reporting but they also charge a good amount for their services. Some of these tools are listed below including the amount they charge for their basic plan.

Table 1. Commercial VRP Software.

Software	**Website**	**Basic Plan**
Routific	https://www.routific.com/	49 USD/Month *For one vehicle*
Route4Me	https://www.route4me.com/	36 EURO/Month *For single user*
Badger	https://www.badgermapping.com/	58 USD/Month *For single user*
OptimoRoute	https://optimoroute.com/	35.10 USD/Month *For single driver*
Onfleet	https://onfleet.com/	550 USD/Month *For 2000 delivery*
Upper	https://www.upperinc.com/	40 USD/Month *For single user*

The list shows that OptimoRoute offers the most cost-effective commercial solution at $35.10/month/user whereas Badger charge the highest $58/month/user for their basic plan. Only Onefleet has a different pricing strategy. Unlike other solutions, they charge based on the number of delivery or pickups completed through their platform instead of the number of users.

2.3 Sustainable Logistics and Transportation

Sustainable logistics aims to reduce the environmental impact of transportation and logistics activities while maintaining efficiency and service quality. This section explores the principles of sustainable logistics.

Principles of Sustainable Logistics

Sustainable logistics involves a holistic approach that considers the environmental, social, and economic impacts of logistics activities. Key principles include:

- **Reduction of Emissions:** Minimizing greenhouse gas emissions from transportation through optimized routing, the use of alternative fuels, and the adoption of low-emission vehicles (McKinnon et al., 2010).
- **Resource Efficiency:** Efficient use of resources, including fuel, labor, and infrastructure, to reduce waste and lower operational costs (Rossi et al., 2013).

In summary, vehicle routing problem (VRP) is a well-studied topic in the field of transportation, logistics management, and optimization. From its emergence, there has been a constant effort from researchers to define different variants of VRP considering different constraints to make it more relatable to the real-world scenario. VRP is a NP-hard problem which makes it difficult to solve and find an optimal solution. The algorithms for solving this problem can be divided into three major categories; including exact, heuristic, and metaheuristic methods. Where heuristic and metaheuristic approaches are more suitable for the large scale VRP. In this era of artificial intelligence and machine learning, reinforcement-based learning algorithms are widely utilized to solve modern day last mile food delivery. Parallel to that, the feasibility of employing unmanned aerial vehicles (UAVs) in the last mile deliveries is being studied in the literature. The sustainability aspect of food transportation lies in reduced carbon emissions and efficient use of resources.

There are a number of open-source tools available to solve vehicle routing problem and generate optimized vehicle routing and scheduling. Most of these tools require substantial technical knowledge to implement and use as they come in the form of packages or libraries. Commercial optimization software have user friendly interfaces but they charge a high amount for their services. This necessitates the development of an open-source route planning and scheduling tool with intuitive user interface. This study intends to build the open-source RouteShaper web application for route optimization. More details on the development and evaluation of the tool are described in the next chapters.

3 Design and Development of the RouteShaper Web Application

In this chapter, we outline the systematic approach and methodologies adopted to develop and evaluate the RouteShaper web application. The primary goal is to ensure that the application not only meets the functional requirements but also aligns with the principles of sustainability and efficiency in food logistics. This chapter is divided into several sections, each containing specific aspects of the development process, including research design, system architecture, backend and frontend development, version management, and testing and validation strategies. By providing a comprehensive overview of the methodologies employed, this chapter aims to demonstrate the structured approach taken to achieve the objectives of this study.

3.1 Research Design

This study adopts a design science approach, focusing on the development and evaluation of RouteShaper web application. Design science research involves creating and evaluating artifacts designed to solve identified problems, emphasizing both the practical utility and theoretical contribution of the research (Wieringa, 2014). The primary goal is to design an innovative solution that meets specific requirements, implement the technical solution, and rigorously evaluate its effectiveness.

Wieringa (2014) described three steps of artifact development: Design and Specification, Implementation, and Validation and Evaluation. In this study, we followed these principles for the development of the RouteShaper web application. The design and specification part is described in this chapter. Implementation and evaluation are provided in the next chapter.

3.2 Design and Specifications

A set of requirements was identified through consultations with stakeholders at Honkajoki Oy and a review of best practices in logistics management. These requirements guide the design and development of the application.

Table 2. List of requirements for the RouteShaper web application.

Requirement ID	Description
R1	Optimize routes to minimize travel distance, time, and cost
R2	Provide a user-friendly interface for logistics managers
R3	Support multiple depots and specific warehouse assignments
R4	Incorporate a Freshness Penalty parameter to maintain product quality
R5	Provide a shareable scheduling report

3.2.1 System Architecture

The system architecture of the RouteShaper is structured to ensure efficiency, scalability, and maintainability. It is a crucial part of the development phase. The architecture is divided into several key components: the optimization model, backend, frontend, data layer, third-party integrations, and deployment. This section provides a detailed overview of each component, their interactions, and the technologies used.

3.2.2 Overview

The web application follows a client-server architecture, where the frontend (client) communicates with the backend (server) via RESTful APIs. The backend handles business logic, data processing, and integration with the vehicle routing problem (VRP) solver, VRP-CLI (Builuk, 2023). The data layer is responsible for persistent storage and management of logistics data. The system also integrates with external services for additional functionalities such as map rendering and geocoding.

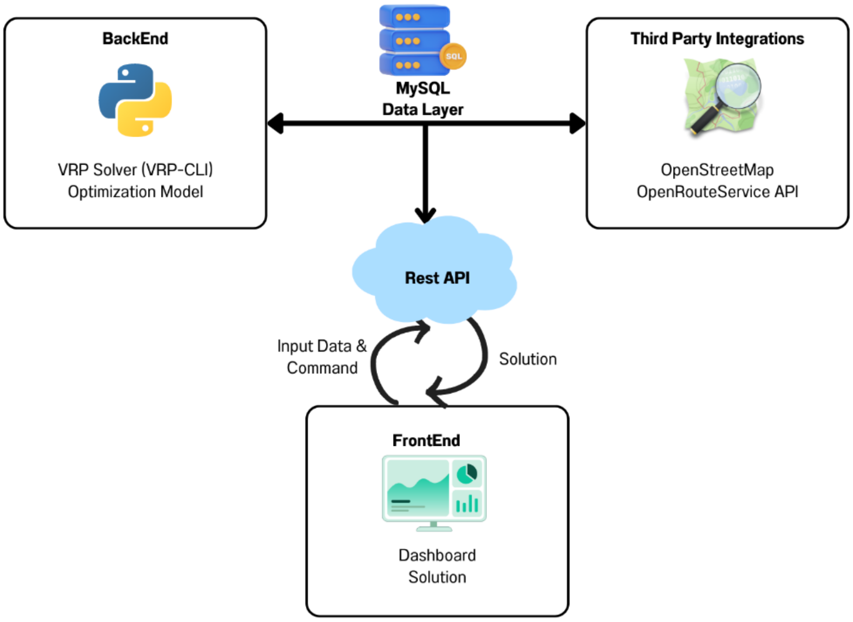


Figure 4. RouteShaper System Architecture.

3.2.3 Optimization Model

In the context of our case company Honkajoki Oy, a leading animal by-product processor in Finland, they have 2 processing plants in two different locations. They collect their raw materials from their partner factories from different locations in Finland. The raw materials need to be picked up within a specified timeframe and has to be transported to a specific processing plant. This nature of the problem makes it a Multi-Depot Vehicle Routing Problem with Time Windows (MDVRPTW). Below is the detailed optimization model, considering the constraints and objectives of Honkajoki Oy's operations.

Sets and Indices

$D = \{1, 2, \dots \dots, n\};\ Set\ of\ processing\ plants\ (depots), indexed\ by\ d$

$F = \{1, 2, \dots \dots, m\};\ Set\ of\ partner\ factories, indexed\ by\ i\ and\ j$

$V = Set\ of\ all\ vehicles, indexed\ by\ k$

$V_d = Set\ of\ vehicles\ at\ d\ processing\ plant\ (depot)$

$N = D \cup F; Set\ of\ all\ locations\ (plants\ and\ factories), indexed\ by\ i\ and\ j$

Parameters

$t_{ij} = Travel\ time\ from\ location\ i\ to\ location\ j$

$dis_{ij} = Distance\ from\ location\ i\ to\ location\ j$

$c_{ij} = Total\ cost\ (time\ \&\ distance)\ from\ location\ i\ to\ location\ j$

$[e_i, l_i] = Time\ window\ during\ which\ pickup\ at\ factory\ i\ must\ start$

$s_i = Service\ time\ required\ at\ factory\ i\ (e.g., loading\ time)$

$Q_k = Capacity\ of\ Vehicle\ k$

$d_i = Demand\ of\ Factory\ i\ (amount\ of\ raw\ materials\ to\ be\ picked\ up)$

$M = A\ large\ positive\ number\ (Typically\ N + 1, used\ for\ constraints)$

$D_i \subseteq D = Set\ of\ process\ plant\ that\ can\ receive\ raw\ materials\ from\ factory\ i$

Decision Variables

$$x_{ij}^k = \begin{cases} 1, if\ vehicle\ k\ travels\ from\ location\ i\ to\ j \\ 0, Otherwise \end{cases}$$

$a_i = Arrival\ time\ at\ location\ i$

$$y_i^k = \begin{cases} 1, if\ factory\ i\ serviced\ by\ vehicle\ k \\ 0, Otherwise \end{cases}$$

Objective Function

The objective is to minimize the total cost (both travel time and distance):

$$Minimize \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} c_{ij}\, x_{ij}^k \qquad (10)$$

Constraints

Flow Conservation Constraints:

Each factory is visited exactly once by one vehicle:

$$\sum_{k \in V} \sum_{j \in N} x_{ij}^k = 1, \forall i \in F \qquad (11)$$

If a vehicle arrives at a factory, it must leave from there:

$$\sum_{j \in N} x_{ij}^{k} = \sum_{j \in N} x_{ji}^{k}, \forall i \in N, \forall k \in V \tag{12}$$

Vehicles must leave from their respective processing plants:

$$\sum_{j \in N} x_{dj}^{k} = 1, \forall d \in D, \forall k \in V_d \tag{13}$$

Time Window Constraints

Vehicles must arrive within the time window at each factory:

$$e_i \leq a_i \leq l_i, \forall i \in F \tag{14}$$

Ensure the correct sequencing of visits with time consideration:

$$a_j \geq a_i + s_i + t_{ij} - M\left(1 - x_{ij}^{k}\right), \forall i, j \in F, i \neq j, \forall k \in V \tag{15}$$

Vehicle Capacity Constraints

The total demand serviced by a vehicle must not exceed its capacity:

$$\sum_{i \in V} d_i\, y_i^{k} \leq Q_k, \forall k \in V \tag{16}$$

Assignment Constraints

Raw materials from each factory i must be transported to an allowable processing plant:

$$\sum_{d \in D_i} \sum_{k \in V_d} \sum_{j \in N} x_{ij}^{k} = 1, \forall i \in F \tag{17}$$

Binary Constraints

$$x_{ij}^{k} \in \{1, 0\}, \forall k \in K, \forall i, j \in N \tag{18}$$

$$y_i^{k} \in \{1, 0\}, \forall k \in K, \forall i \in F \tag{19}$$

Freshness Penalty (FP)

Maintaining the freshness of raw materials is crucial in the context of Honkajoki Oy, to ensuring the quality of the end products. To address this issue, we have introduced a

Freshness Penalty parameter into our vehicle routing problem (VRP) model. This parameter is designed to quantify the impact of the time delay on the freshness of the raw materials transported from various factories to different plants for further processing. We define the freshness penalty parameter as follows.

$T_i^{ready} = Time\ at\ which\ raw\ materials\ are\ ready\ for\ pickup\ at\ factory\ i$

$T_{ij}^{arrive} = Time\ at\ which\ raw\ materials\ from\ factory\ i\ arrive\ at\ plant\ j$

$F_{ij} = Freshness\ penalty\ for\ transporting\ raw\ materials\ from\ i\ to\ j$

$P = Penalty\ per\ hour\ of\ delay$

We calculate freshness penalty as:

$$F_{ij} = \left(T_{ij}^{arrive} - T_i^{ready}\right) \times P, \forall j \in N, \forall i \in F\ , i \neq j \qquad (20)$$

Here, P is a critical parameter determined in consultation with stakeholders, including logistics managers and quality assurance teams at Honkajoki Oy. This collaborative approach ensures that the parameter accurately reflects the cost implications of delays in product freshness.

3.2.4 Backend Architecture

The backend of the RouteShaper web application is a critical component that manages data processing, business logic, and integration with the vehicle routing problem (VRP) solver VRP-CLI. The backend is implemented using Django, a high-level Python web framework, and incorporates the Django REST Framework (DRF) for building robust APIs. This section covers the overall architecture of backend, key modules, and integration with the VRP-CLI package.

Technologies Used:

- **Django**: A high-level Python web framework for rapid web development.
- **Django REST Framework (DRF)**: A powerful toolkit for building Web APIs.
- **VRP-CLI**: A Python package for solving vehicle routing problems (Builuk, 2023).

Components:

- **API Endpoints**: Exposed using DRF to handle CRUD operations for logistics data.
- **Business Logic**: Implemented in Django views and serializers to process data and apply business rules.
- **VRP Integration**: Utilizes VRP-CLI for optimizing routes based on given constraints and requirements.
- **Authentication**: Managed using Django's built-in authentication system along with JWT (JSON Web Token) for secure API access.

Structure:

- **project_root/**
    - **vrp/**: Contains Django settings and configuration files.
    - **base/**: Main application directory containing Django models, views, serializers, and APIs.
        - **models/**: Handles the fields and behaviors of each data entity.
        - **rest_views/**: Manages business logic and integration with VRP-CLI.
        - **serializers/**: Manages validation and rendering of data.
        - **urls/**: Defines all API endpoints.
    - **manage.py**: Django command-line utility for administrative tasks.

Key Modules and Their Functions

The backend architecture is designed to be modular, scalable, and maintainable. It consists of several key modules, each responsible for different aspects of the application's functionality. The core modules include:

- **User Management**: Handles user authentication, authorization, and profile management.
- **Work Management**: Manages work details, containing multiple jobs and vehicles.
- **Job Management**: Manages delivery and pick-up job details, including geolocation, demand, and time window.

- **Fleet Management**: Manages vehicle details, including type, capacity, and shift.
- **Optimization Engine**: Integrates with the VRP-CLI package to generate optimized routes and scheduling based on input data.

Integration with VRP-CLI

The vrp_cli package is the main optimization engine of RouteShaper web application that solves the Vehicle Routing Problem (VRP). The backend integrates with vrp_cli to optimize routing and generate scheduling based on various constraints such as demand, vehicle capacity, delivery time windows, and route distances. Below is a pseudocode representation of how vrp_cli is implemented within the backend:

Pseudo code for vrp_cli solver in RouteShape

```
Function solve(request, pk):
    work = get Work by pk
    jobs = get Jobs related to work
    vehicles = get Vehicles related to work
    fleet, profiles = transform vehicles to internal format
    fleet = create Fleet with vehicles and profiles
    problem = create Problem with jobs and fleet

    matrices = []
    if custom_matrix:
        v_locations = get vehicle locations from fleet
        j_locations = get job locations from jobs
        durations, distances = get durations and distance be-
tween each location using OpenRouteService
        matrix = create RoutingMatrix using OpenRouteService
with vehicle profile, durations, distances
        matrices.append(matrix as JSON)

    # Solve the problem using vrp_cli
    problem_json = convert problem to JSON
    matrices_json = convert matrices to JSON
    config_json = convert config to JSON
    vrp_solution = call vrp_cli.solve_pragmatic with prob-
lem_json, matrices_json, config_json
    solution = parse vrp_solution into Solution object


    work_serializer = serialize work
    job_serializer = serialize jobs
    vehicle_serializer = serialize vehicles
    solution_data = solution as JSON
```

```
solution_model = create Solution for work with solution_data

context = {
    "id": solution_model.id,
    "solution": solution as dictionary,
    "work": work_serializer data,
    "jobs": job_serializer data,
    "vehicles": vehicle_serializer data,
}
return Response with context
```

3.2.5 Frontend Architecture

The frontend of the RouteShaper web application focuses on creating an intuitive and responsive user interface that allows users to interact with the system efficiently. The frontend is implemented using React, a popular JavaScript library for building user interfaces, complemented by context and custom hooks for state management and Axios for handling API requests. This section describes the frontend development process, covering the overall architecture, key components, and user interface design.

Technologies Used:

- **React**: A JavaScript library for building user interfaces.
- **React Context**: For state management across the application.
- **React Router**: For handling navigation and routing within the application.
- **Leaflet**: A JavaScript library to render maps and geolocations.
- **Axios**: A promise-based HTTP client for making API requests.
- **Tailwind CSS**: A utility-first CSS framework for building responsive and visually appealing designs.

Components:

- **User Interface (UI) Components**: Flowbite UI Component library has been utilized for modular and reusable components for different parts of the application such as forms, tables, maps, and dashboards.

- **Routing**: Implemented using React Router to handle navigation between different views and components.
- **API Integration**: Axios is used for communication with the backend APIs to fetch and update data.

Structure:

- **src/**
  - **components/**: Contains all UI components.
  - **hooks/**: Custom hooks contain actions, reducers, and store configuration for state management.
  - **managers/**: Contains API service configurations.
  - **pages/**: Contains different page views like Dashboard, Work, Job Category, Jobs, Vehicle Profiles and Fleet.
  - **App.js**: Main application file.
  - **index.js**: Entry point for the React application.

Key Modules and Their Functionalities

The frontend architecture is designed to be modular, scalable, and maintainable. It follows a component-based structure, where each UI element is encapsulated within a React component. The application state is managed using React Context and custom hooks to ensure a consistent state across the entire application. Axios is used to handle HTTP requests to the backend API. The frontend is composed of several key components, each responsible for a specific part of the user interface. These components are organized into a directory structure for better maintainability and scalability. Key modules and their functionalities are described below.

- **Dashboard**: The main dashboard that provides an overview of key metrics and quick access to different statistics of the optimized route and scheduling for associated works.

- **Work**: Components to display a list of works, detailed view, and forms for creating/updating works.
- **Job**: Components to display a list of jobs, detailed view, and forms for creating/updating jobs.
- **Fleet**: Components to display a list of vehicles, detailed view, and forms for creating/updating vehicles.

3.2.6 Data Layer

Technologies Used:

- **MySQL**: A powerful, open-source object-relational database system.
- **Django Object-Relational Mapping (ORM)**: Used for database operations and migrations.

Components:

- **Database Schema**: Designed to store information about works, job categories, jobs, vehicle profiles, vehicles, routes, and schedules.
- **Models**: Django models representing database tables.
- **Migrations**: Handled by Django's migration framework to manage schema changes.

Entity Relationship Diagram (ERD): To provide a clear understanding of the data layer, an Entity Relationship Diagram (ERD) is used. The ERD illustrates the entities, their attributes, and the relationships between them.

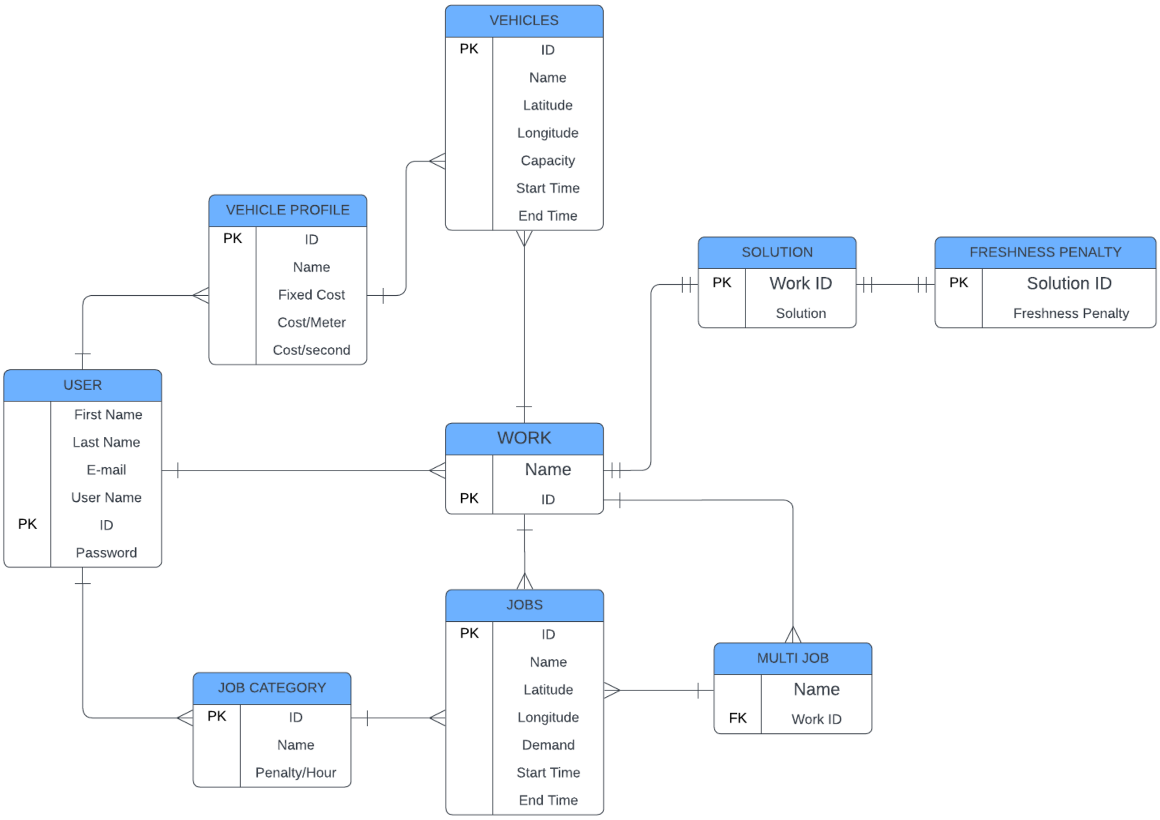


Figure 5. Entity Relationship Diagram (EDR) of RouteShaper WebApp.

The ERD includes the following entities:

- **User**

  Attributes: user_id (PK), username, password, email, first_name, last_name

- **Work**

  Attributes: work_id (PK), name

- **Job Category**

  Attributes: category_id (PK), name, penalty/hour

- **Jobs**

  Attributes: job_id (PK), name, latitude, longitude, demand, start_time, end_time

- **Vehicle Profile**

  Attributes: profile_id (PK), name, fixed_cost, cost/meter, cost/second

- **Vehicle**

  Attributes: vehicle_id (PK), name, latitude, longitude, capacity, start_time, end_time
- **Multi Job**

  Attributes: work_id (FK), name
- **Solution**

  Attributes: work_id (PK), solution
- **Freshness Penalty**

  Attributes: solution_id (PK), freshness_penalty

Relationships:

- **User** to **Work**: One-to-many (A user can manage works)
- **User** to **Job Category**: One-to-many (A user can manage multiple job categories)
- **User** to **Vehicle Profile**: One-to-many (A user can manage multiple vehicles profiles)
- **Multi Job** to **Job**: One-to-many (One multi job can have multiple jobs)
- **Work** to **Vehicle**: One-to-many (One work can have multiple vehicles)
- **Work** to **Job**: One-to-many (One work can have multiple jobs)
- **Work** to **Multi Job**: One-to-many (One work can have multiple multi jobs)
- **Work** to **Solution**: One-to-one (One work can have only one solution)
- **Solution** to **Freshness Penalty**: One-to-one (There will be only one freshness penalty calculation for a given solution)

Structure:

- **models.py**: Contains Django models for each data entity.

Migrations:

- **Migrations Directory**: Contains migration files generated by Django to handle changes in the database schema.

This detailed representation, including the ERD, helps in understanding how different entities interact within the system, ensuring that the data layer is well-structured and efficient for managing logistics data.

3.2.7 Third-Party Integrations

Technologies Used:

- **OpenStreetMap**: For interactive map rendering and visualization.
- **OpenRouteService API**: For generating routing matrix using geographic coordinates.

Components:

- **Map Rendering**: Integrated in the frontend to display routes and vehicle locations.
- **Geocoding**: Used in the backend to convert addresses into coordinates and generate routing matrix.

Structure:

- **frontend/src/components/ResultMap.js**: Contains logic for interacting with OpenStreetMap for visualizing optimized route for a given task.
- **backend/base/vrp_extra/utils.py**: Contains utility functions for interacting with OpenRouteService API for generating routing matrix utilizing coordinates of the geolocations.

3.2.8 Interactions and Data Flow

1. **User Interaction**: Users interact with the web application through the frontend, performing actions such as scheduling deliveries and optimizing routes.
2. **API Requests**: The frontend makes API requests to the backend to fetch or update data.
3. **Optimization Model Execution**: The backend processes these requests, applying optimization model and business logics.
4. **VRP Solver**: For routing optimization, the backend uses VRP-CLI to compute the best routes based on input data.
5. **Data Persistence**: Results from the VRP solver and other operations are stored in the MySQL database.

6. **Response to Frontend**: The backend sends the processed data back to the frontend, which updates the user interface accordingly.

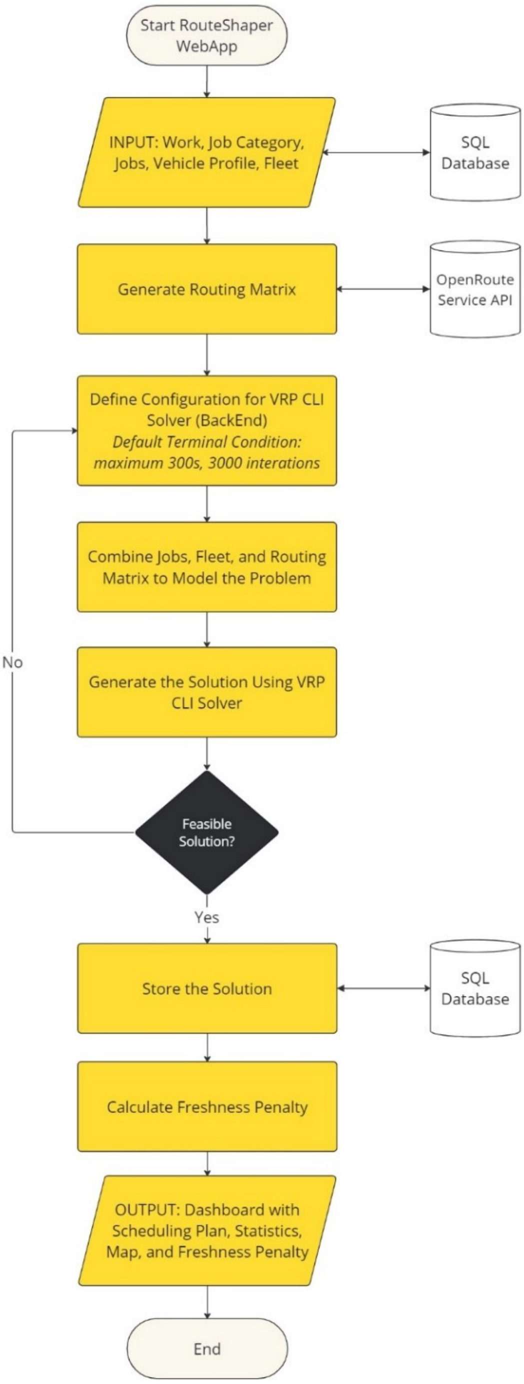


Figure 6. Sequential procedure for generating solution using RouteShaper.

By utilizing this architecture, the system ensures efficient handling of transportation scheduling and vehicle routing tasks, promoting sustainable logistics practices.

3.3 Version Management

Version management is a critical aspect of the development process, ensuring that the project remains organized, and changes are tracked. This section outlines the version management strategies and tools used in the development of the RouteShaper web application, focusing on the use of Git and GitHub.

3.3.1 Version Control System

Git: Git is a distributed version control system that tracks changes in source code during software development. It allows multiple developers to work on the same project simultaneously without interfering with each other's work. Git was chosen for its widespread adoption, and powerful branching and merging capabilities.

GitHub: GitHub is a web-based platform that provides hosting for Git repositories. It also has additional features such as issue tracking, project management, and collaboration tools. It serves as the central repository for the project, enabling team members to collaborate effectively.

3.3.2 Documentation

README.md: The README file in the repository provides an overview of the project, setup instructions, and usage guidelines. It is updated regularly to reflect the latest changes and instructions.

By following this detailed version management process, the project maintains a high level of organization, collaboration, and quality control, ensuring that the web application is developed efficiently and effectively.

3.4 Testing and Validation

Testing and validation are critical components of the software development lifecycle, ensuring that the web application functions as intended, is free of bugs, and meets the requirements specified by the stakeholders. This section outlines the testing strategies and validation processes implemented for the web application.

3.4.1 User Acceptance Testing (UAT)

The testing strategy for the RouteShaper web application includes User Acceptance Testing (UAT) to ensure comprehensive coverage and reliability. The strategy employs manual testing by end-users to validate user experience and edge cases against their requirements and expectations.

Process:

- **Planning**: Define the scope and objectives of UAT, involving key stakeholders.
- **Test Case Development**: Develop test cases based on user stories and requirements.
- **Execution**: Users execute the test cases and document any issues or feedback.
- **Feedback and Iteration**: Address any issues raised during UAT and iterate as necessary.

Example Test Case:

Table 3. Example Test Case.

Test Case ID	**TC-001**
Description	Verify that a user can create a new work entry
Steps	1. Login as a valid user
	2. Navigate to the "Work" page
	3. Fill out the form and submit to create a work
Expected Result	A new work entry is created and visible in the list

By implementing this comprehensive testing and validation strategies, the project ensures that the web application is robust, reliable, and meets the user requirements.

4 System Implementation and Evaluation

Chapter 4 delves into the practical aspects of the system implementation for the RouteShaper web application. This chapter provides a description of how the theoretical concepts and methodologies discussed in previous chapters were translated into a functional software solution. This chapter begins with an overview of the RouteShaper Application, then the core features and functionality of the tool are described. Finally, a case study of Honkajoki Oy is presented to measure how the RouteShaper web application contributes to the efficient and sustainable transportation operations.

4.1 Overview of the Application

This section provides a comprehensive overview of the application's functionality, user interface, and core features.

4.1.1 Purpose and Objectives

The primary purpose of the application is to provide logistics managers and stakeholders with a tool for efficient planning and optimizing delivery routes and schedules. The application addresses several key objectives:

- **Efficiency**: Reduce travel time and distance, thereby lowering fuel consumption and operational costs.
- **Sustainability**: Minimize the carbon footprint of logistics operations by optimizing routes and schedules.
- **User-Friendliness**: Offer an intuitive and easy-to-use interface that simplifies complex logistics tasks.

4.2 Features and Functionality

This section contains details about the key features and functionalities of the application, highlighting how each contributes to the overall efficiency and sustainability of logistics operations. Before delving further into the details of features and functionality, a sequential workflow of a user in the RouteShaper web application is presented below.

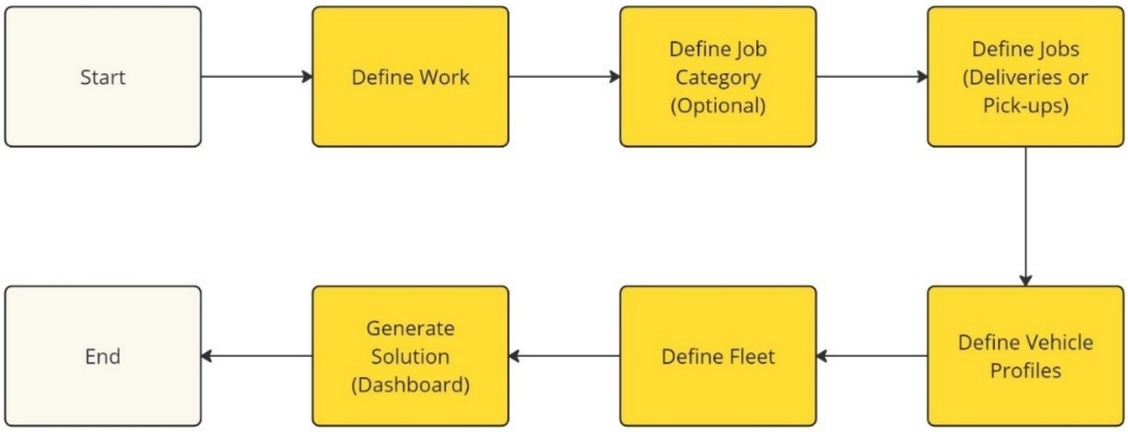


Figure 7. User Workflow in the RouteShaper WebApp.

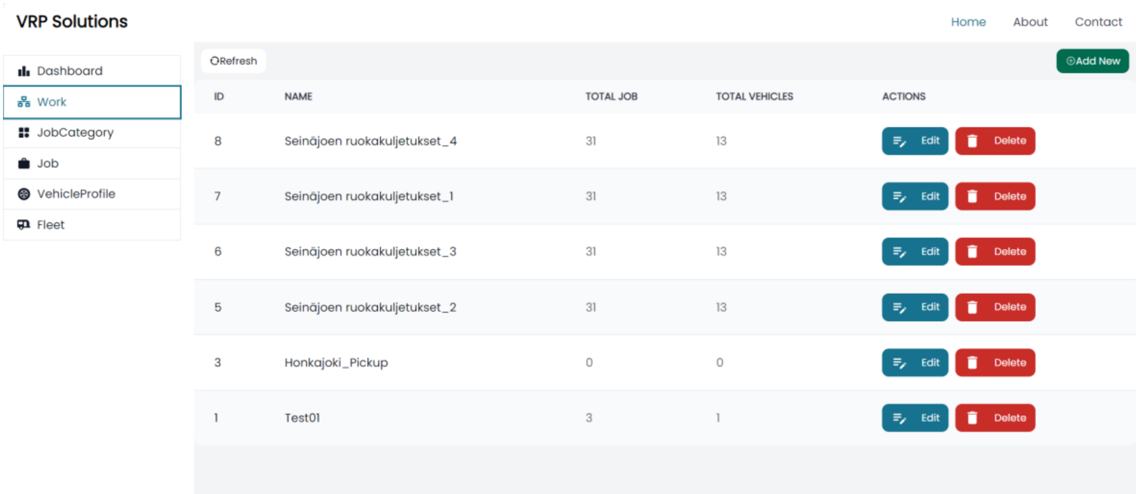
4.2.1 Work Management

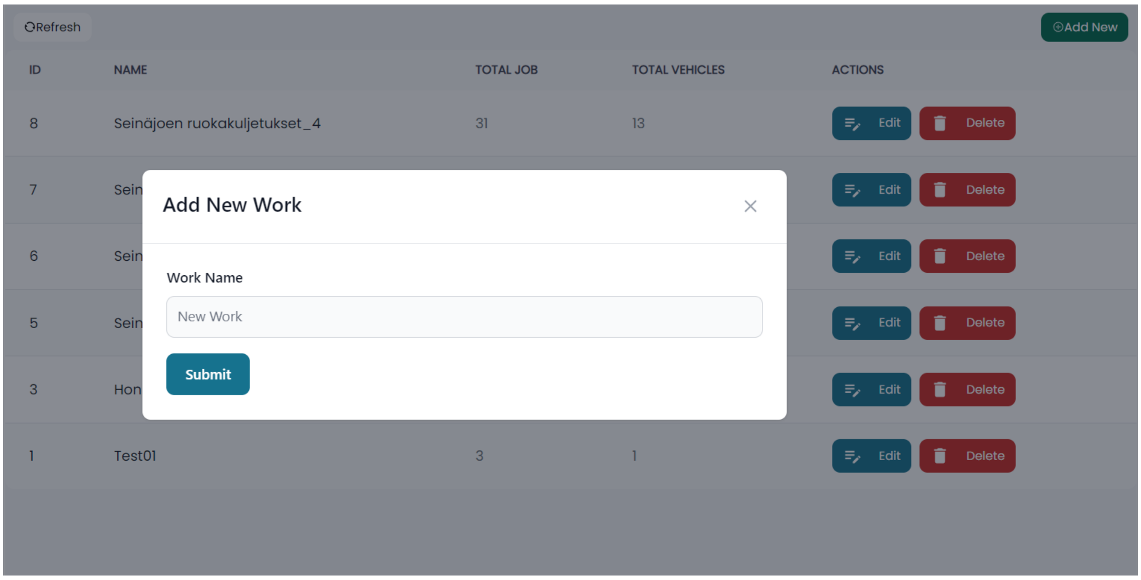
Work is a set of jobs i.e. delivery or pick-up tasks associated with a fleet of vehicles. A work can be constructed combining all the delivery tasks that need to be completed in a day along with the available vehicles for the same day. However, logistics managers have the liberty of designing and defining their own 'Work' based on certain given criteria. The Work page enables users to add new work, edit, update, and delete existing works.

Key Features and Functionality:

- **Work Creation**: Users can add new work, edit, update, and delete existing ones. Each work entries needs to be unique; no two different works can have the same name. The name of the work should be less than 255 characters.

- **One Click Delete**: This feature allows users to delete any existing work along with all related jobs and vehicles in a single click. It provides users with clear control over their data and reduces the effort required to individually delete associated entries.





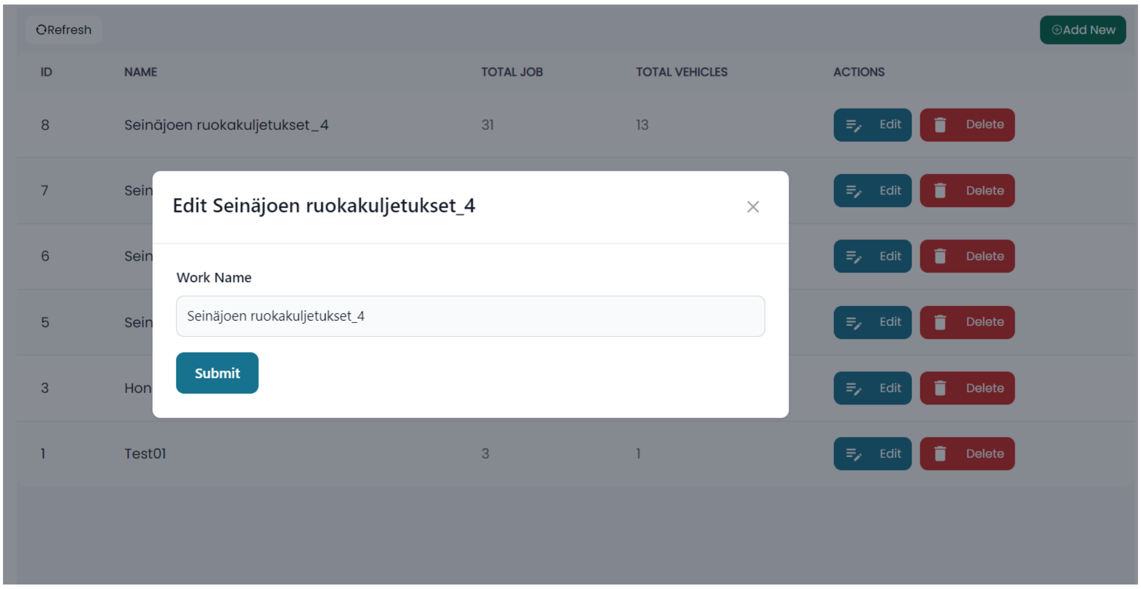


Figure 8. Work Management in the RouteShaper WebApp.

4.2.2 Job Category Management

It is an optional feature. If any company has their pick-up or delivery tasks divided into different categories with a defined hourly freshness penalty cost, then they can utilize this unique feature. Otherwise, it can be left blank.

Key Features and Functionality:

- **Job Category Creation**: Users can input job categories with defined hourly freshness penalty cost. Each category should be unique, and the name of the category should not be more than 255 characters. The penalty per hour is a positive real number defined by the concerned stakeholders.
- **Edit and Delete**: Users can delete job category entries or edit to make adjustments as needed.

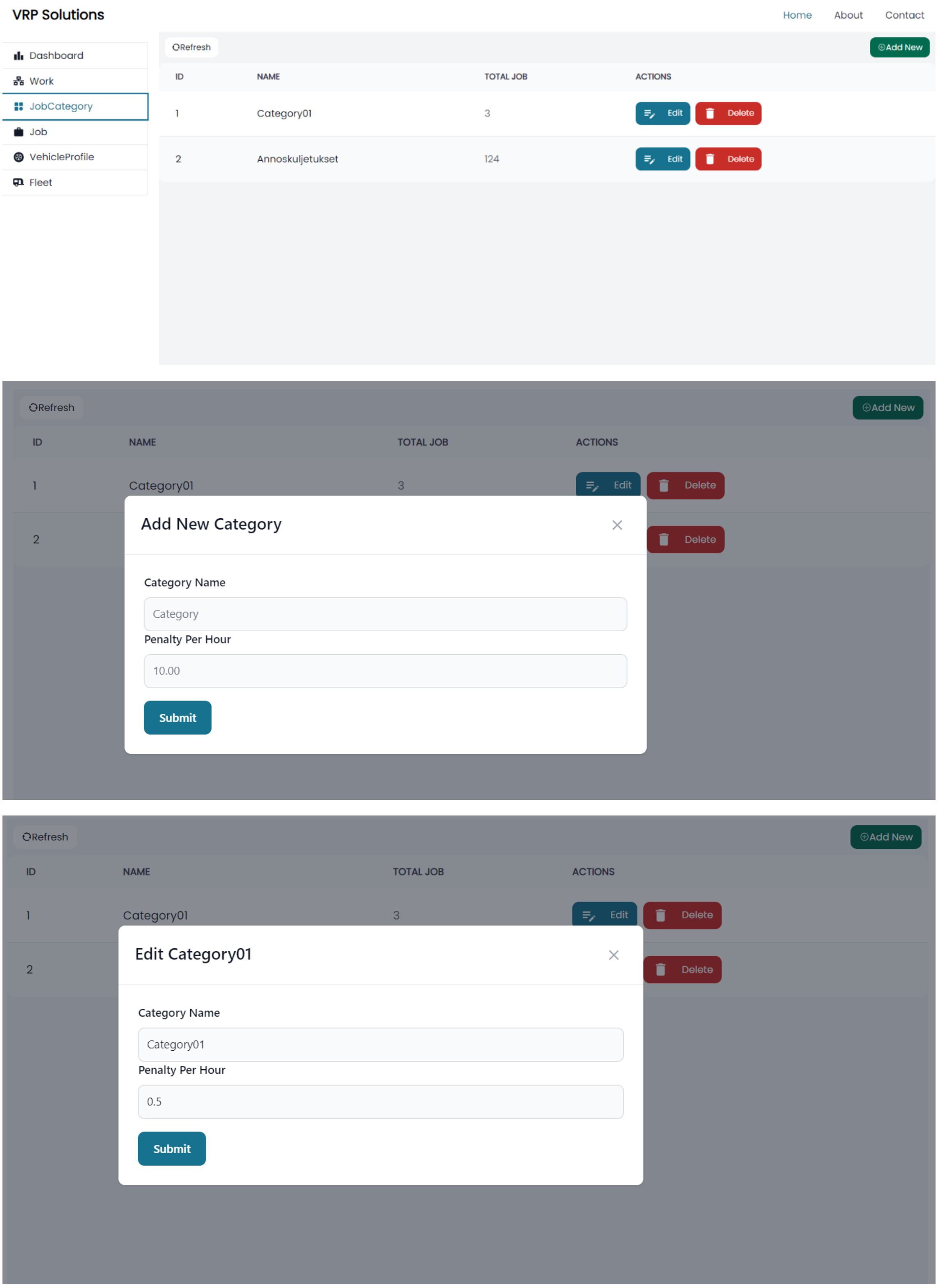


Figure 9. Job Catgory Management in the RouteShaper WebApp.

4.2.3 Job Management

The delivery and pick-up tasks associated with a route planning is defined as jobs. With intuitive interface and interactive maps, RouteShaper web application enables users to define tasks very easily.

Key Features and Functionality:

- **Individual Job Creation**: Users can input pickup and delivery locations, define time windows, and customer demand. The location can be picked up from an interactive map. If a task contains multiple delivery and pick-up jobs, users can define that by selecting 'Multi' from the Job Type drop down. Each Job entry must be unique, and the name of the Job must not exceed 255 characters.

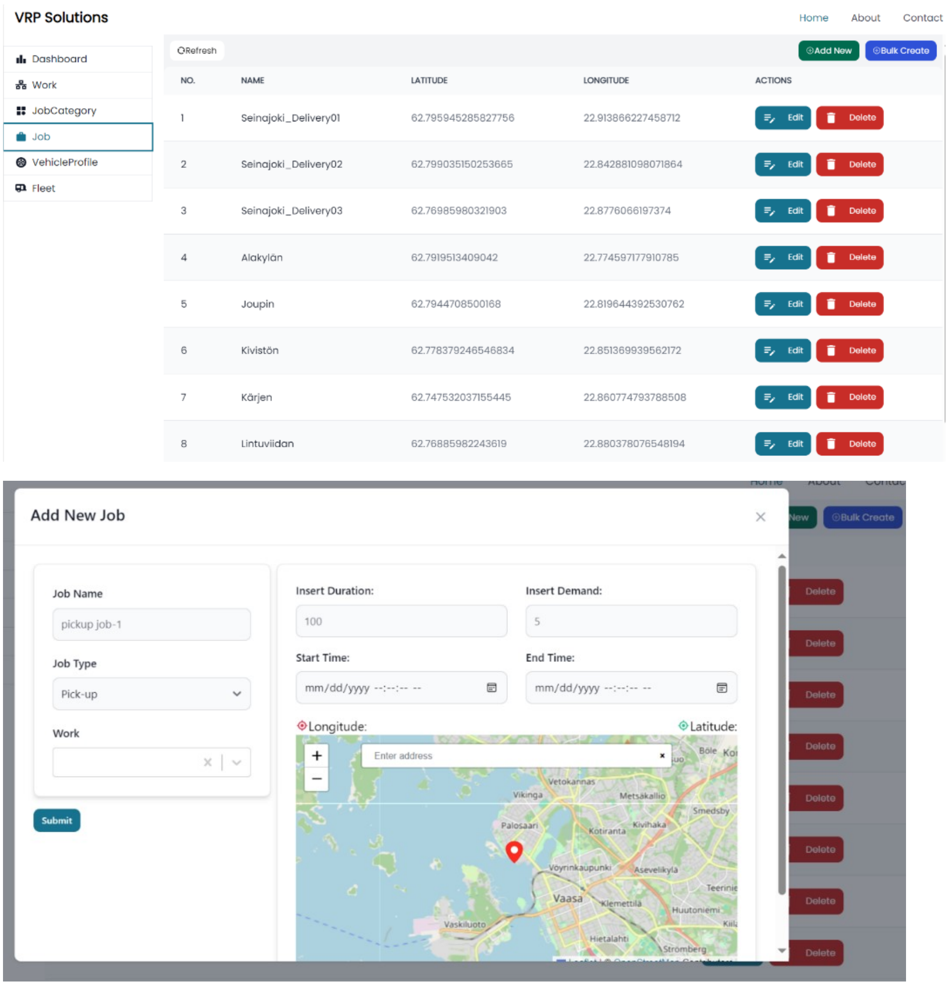


Figure 10. Individual Job Entry in The RouteShaper WebApp.

- **Bulk Job Creation**: In industrial settings, there are often numerous delivery or pick-up tasks to be completed. Manually entering each task into the system can be impractical. Therefore, we introduced a bulk job creation feature. Users can define their jobs in a Microsoft Excel sheet in the format shown in the following figure and save it as a CSV UTF-8 file. Then, they select the desired work from the drop-down menu and upload the CSV file. Using this feature, users can register multiple jobs in the system with a single click.

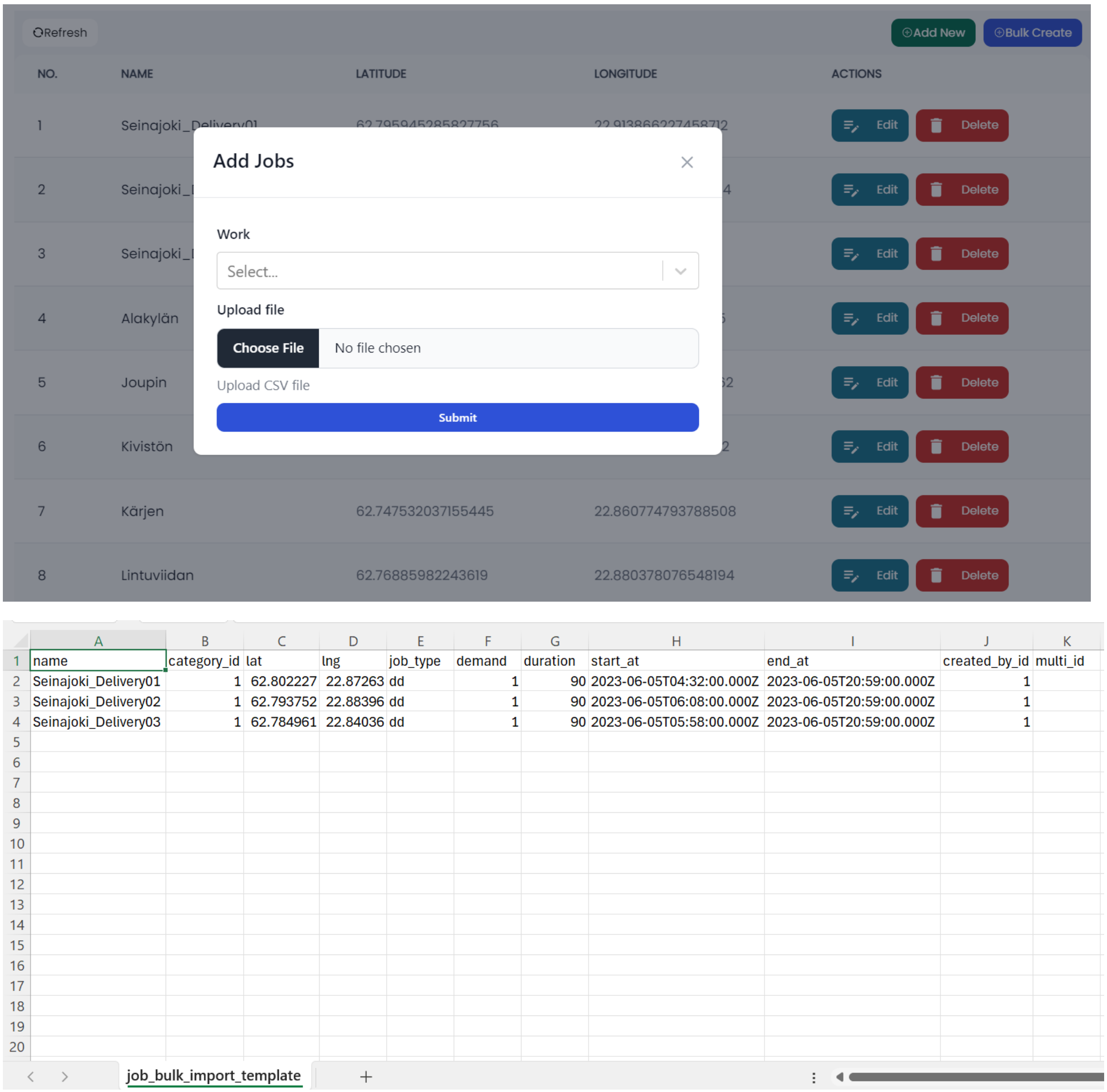


Figure 11. Bulk Job Entry in The RouteShaper WebApp.

- **Edit and Delete**: Users can delete any job entries or edit to make adjustments as needed.

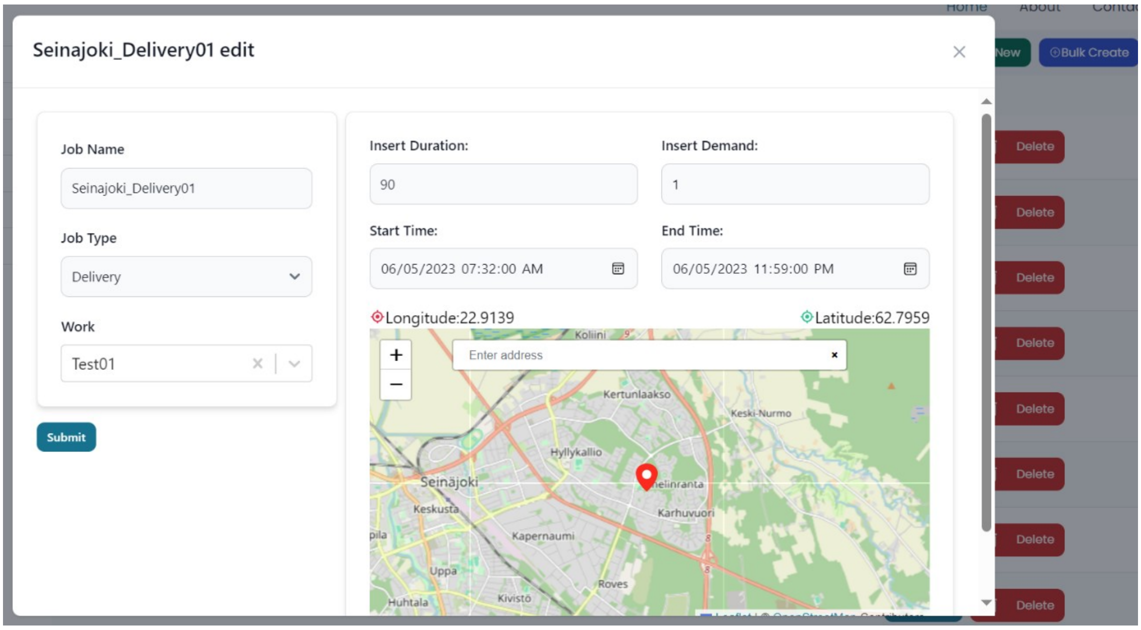


Figure 12. Job Edit in The RouteShaper WebApp.

4.2.4 Vehicle Profile Management

Each type of vehicle, such as cars, trucks, covered vans, etc., has some fixed costs associated with each trip. Additionally, the cost per unit of time and distance varies based on the type of vehicle. The vehicle profile management feature allows users to define the types of vehicles in their fleet along with their associated costs.

Key Features and Functionality:

- **Profile Creation**: Users can define vehicle profile along is fixed cost, cost per meter, and cost per second. The profile entries have to be unique and the name of the profile should be less than 255 characters. Fixed cost, cost per meter, and cost per seconds are non-negative real numbers.
- **Edit and Delete**: Users can delete any vehicle profile entries or edit to make adjustments as needed.

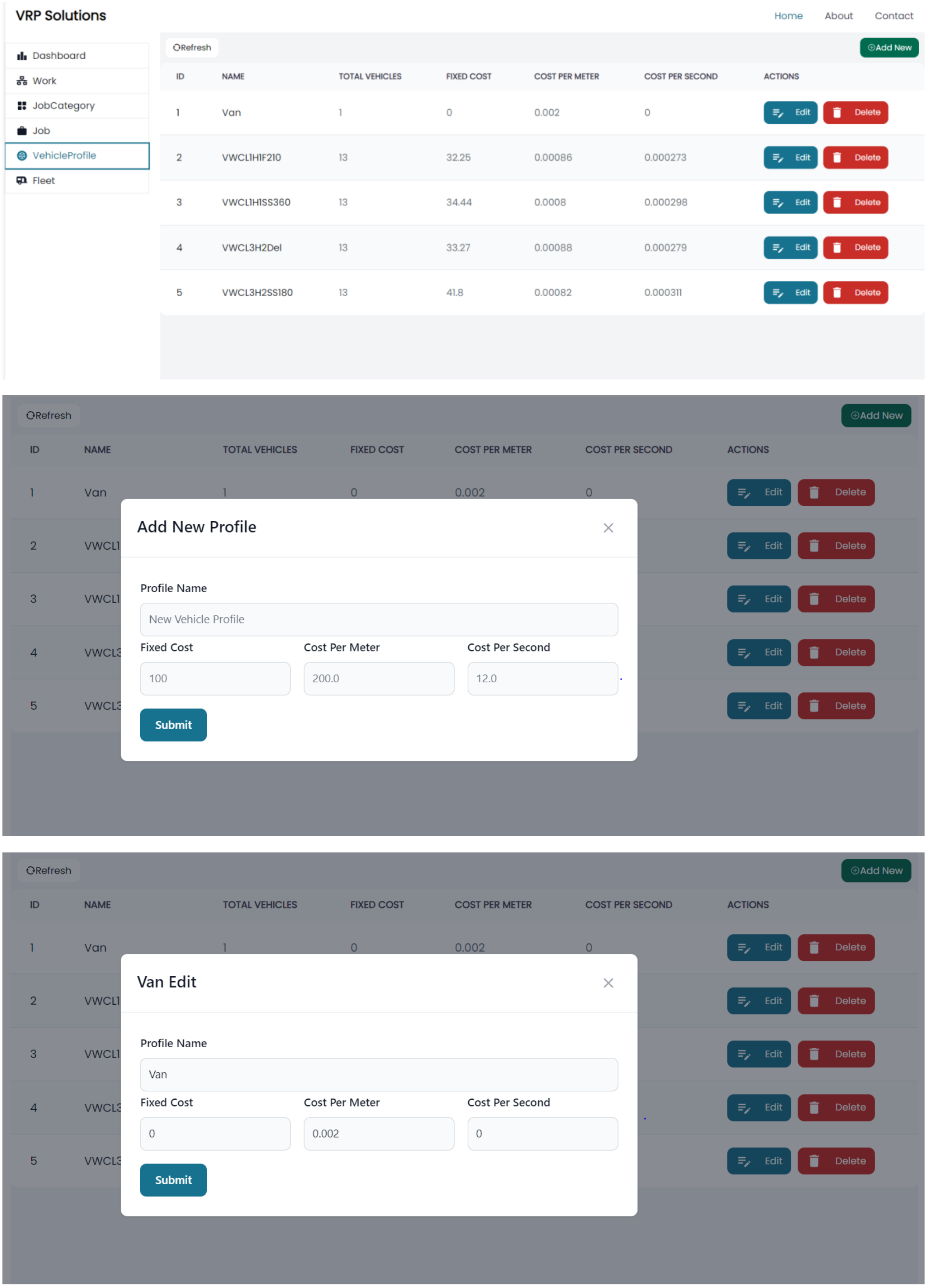


Figure 13. Vehicle Profile Management in The RouteShaper WebApp.

4.2.5 Fleet Management

A fleet is a set of vehicles assigned to specific tasks. The fleet management feature allows users to add, update, and delete vehicles in a fleet as per the requirements. Similar to job entries, users can create vehicle entries both individually and in bulk.

Key Features and Functionality:

- **Individual vehicle Creation**: Users can input depot locations, define time windows, and set vehicle capacities. The location can be picked up from an interactive map. Each Vehicle entry must be unique, and the name of the vehicle must not exceed 255 characters.

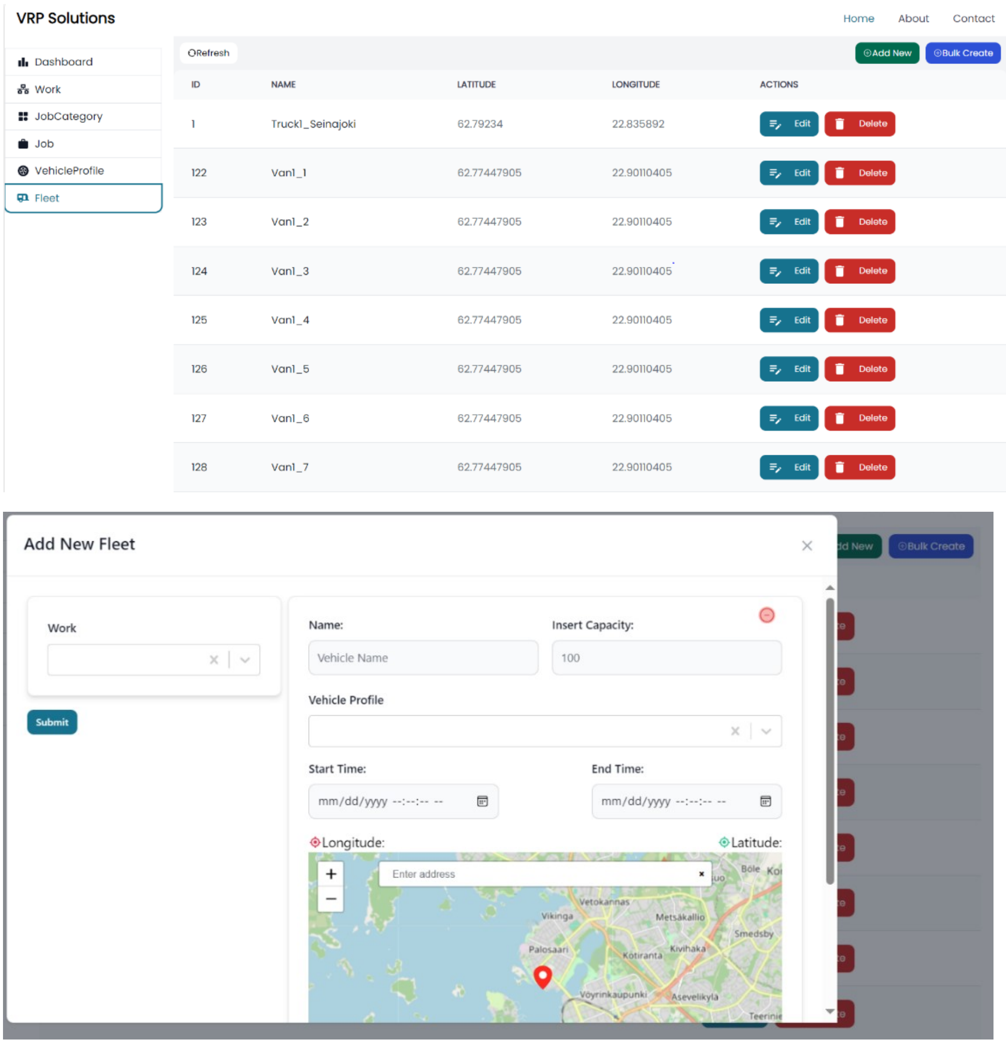


Figure 14. Individual Vehicle Entry in The RouteShaper WebApp.

- **Bulk Fleet Creation**: In industrial settings, a fleet often includes a number of vehicles. Manually entering each vehicle into the system may not be feasible always. Therefore, we introduced a bulk fleet creation feature. Users can define their vehicles in a Microsoft Excel sheet in the format shown in the following figure and save it as a CSV UTF-8 file. Then, they select the desired work from the drop-down menu and upload the CSV file. Using this feature, users can register multiple vehicles in the system with a single click.

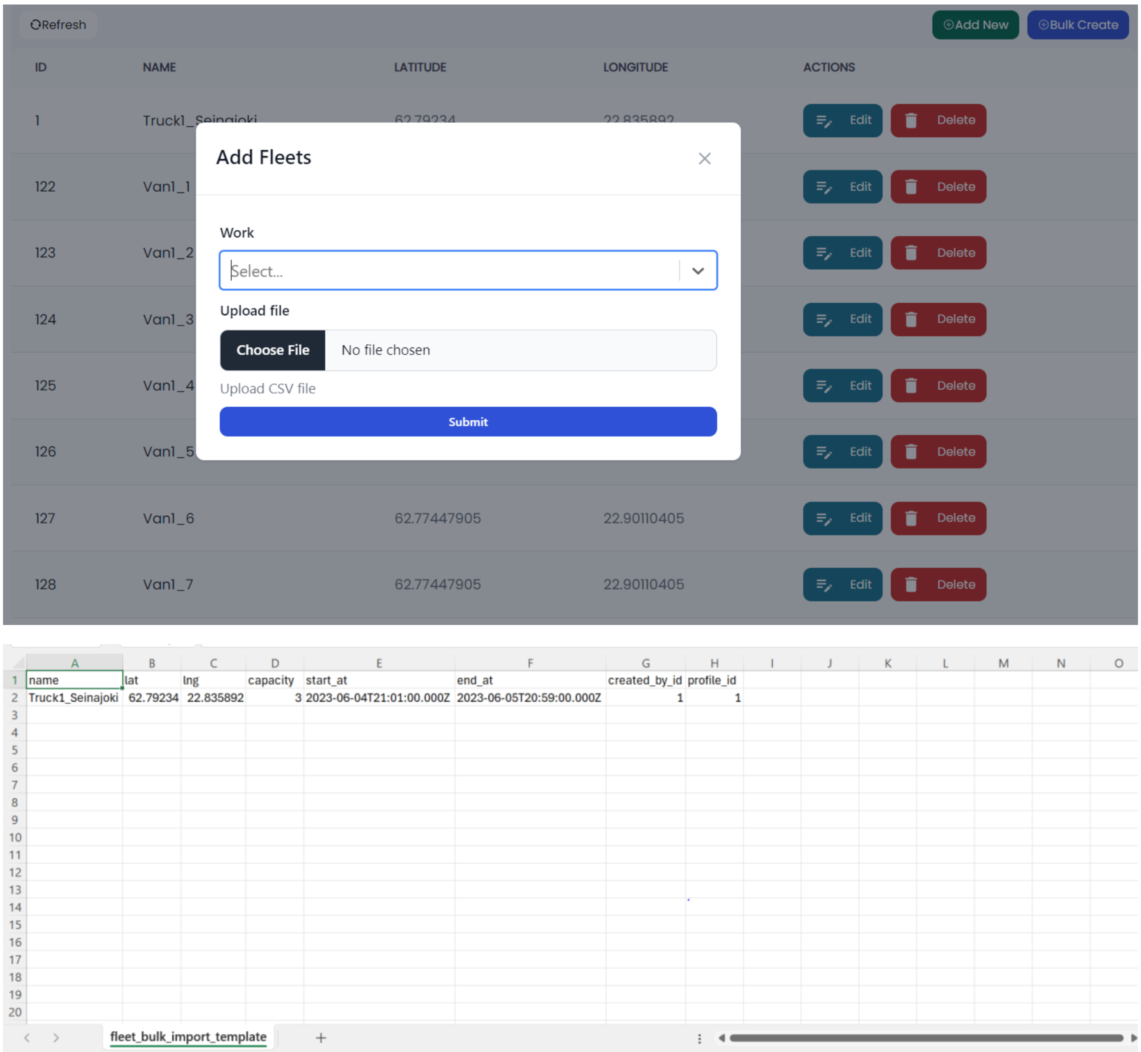


Figure 15. Bulk Fleet Entry in The RouteShaper WebApp.

- **Edit and Delete**: Users can delete any vehicle entries or edit to make adjustments as needed.

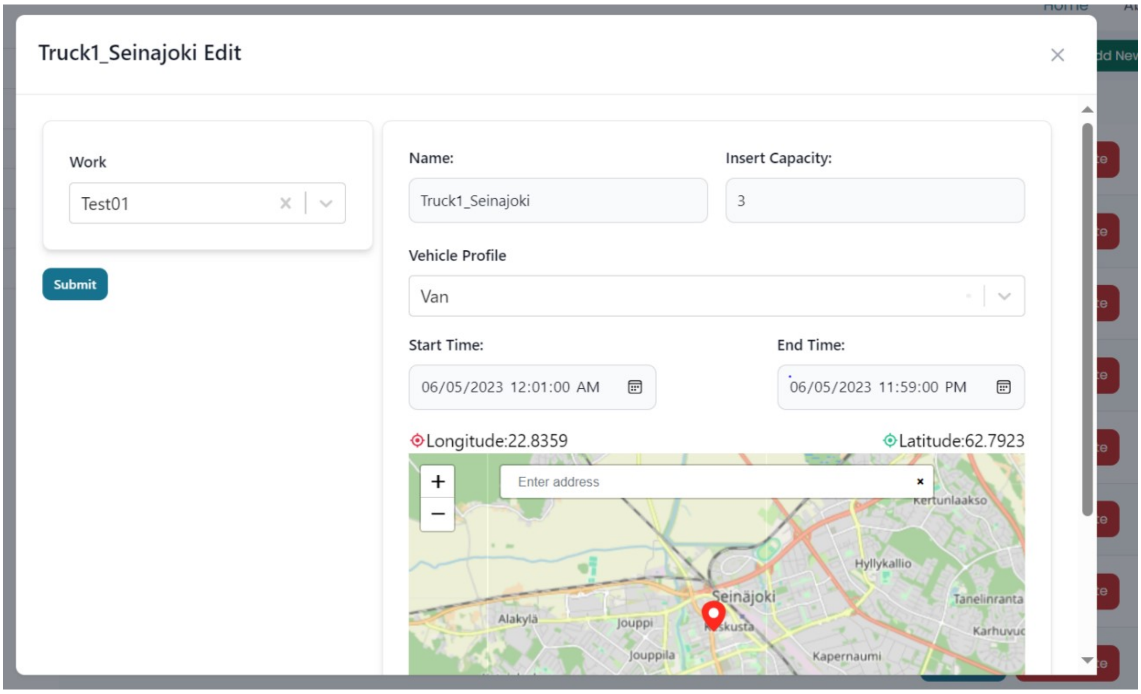


Figure 16. Vehicle Edit in The RouteShaper WebApp.

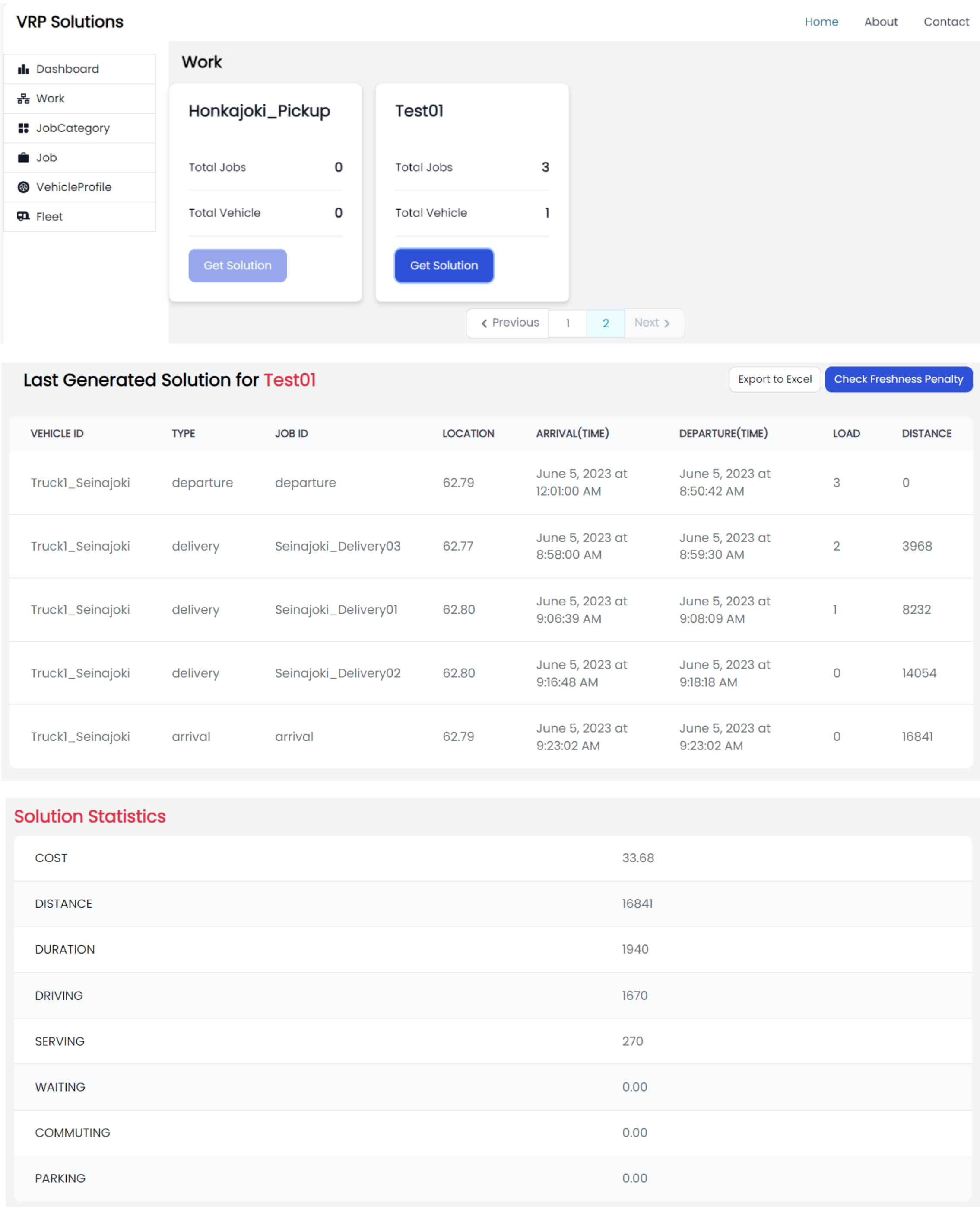
4.2.6 Dashboard

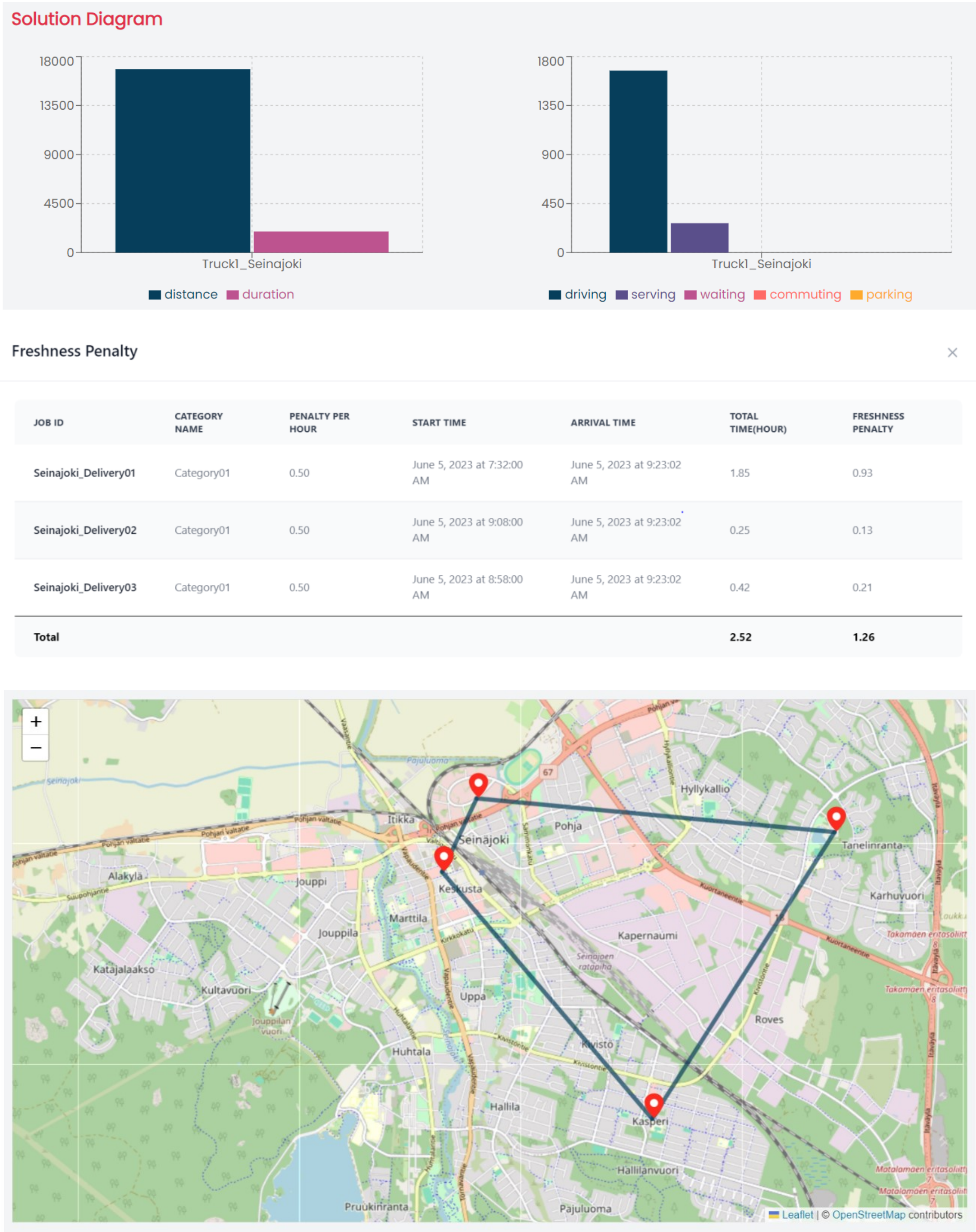
The Dashboard serves as the central hub for users, providing a high-level overview of key metrics and quick access to various functionalities.

Key Features and Functionality:

- **Summary Widgets**: Displays all the work entries with the number of all associated vehicles and jobs. Users can generate solutions for any of the work by clicking on the 'Get Solution' button.
- **Solution Table**: Displays the solution data for each job and vehicle of a given work. Shows arrival time, departure time, load, and distance covered at each location.
- **Statistics Table**: Displays the total cost, total distance covered, total time needed including driving time, serving time, waiting time, and parking time for a given solution.
- **Freshness Penalty Calculation Table**: By clicking on the 'Check freshness Penalty' button, users can check the freshness penalty for a given solution.

- **Performance Charts**: Visual representations of data trends, such as duration and distance covered by each individual vehicles, breakdown of driving, serving, waiting, and parking time for each vehicle.
- **Export and Share**: Allows users to export optimized routes and scheduling in CSV format and share them with drivers and stakeholders.





JOB ID	CATEGORY NAME	PENALTY PER HOUR	START TIME	ARRIVAL TIME	TOTAL TIME(HOUR)	FRESHNESS PENALTY
Seinajoki_Delivery01	Category01	0.50	June 5, 2023 at 7:32:00 AM	June 5, 2023 at 9:23:02 AM	1.85	0.93
Seinajoki_Delivery02	Category01	0.50	June 5, 2023 at 9:08:00 AM	June 5, 2023 at 9:23:02 AM	0.25	0.13
Seinajoki_Delivery03	Category01	0.50	June 5, 2023 at 8:58:00 AM	June 5, 2023 at 9:23:02 AM	0.42	0.21
Total					2.52	1.26

Figure 17. Dashboard in The RouteShaper WebApp.

By providing a comprehensive set of features and functionalities, the RouteShaper web application enables logistics managers to optimize transportation scheduling and vehicle routing, enhancing the efficiency and sustainability of food logistics operations.

4.3 Case Study: Optimizing Transportation for Honkajoki Oy

To demonstrate the effectiveness and practical application of the RouteShaper web application, a case study was conducted with Honkajoki Oy, a Seinäjoki based mid-sized company in Finland specializing in the processing of animal by-products (Honkajoki Oy, 2024).

4.3.1 Background

Honkajoki Oy has two different processing plants located in two different locations. They collect raw materials from their 16 partner factories from different locations in Finland. Honkajoki Oy faced several challenges in its logistics operations, including high fuel costs, inefficient routing leading to delayed deliveries, and the need to maintain the freshness of perishable goods. The company aimed to reduce operational costs, improve delivery efficiency, and minimize its environmental impact by adopting advanced logistics optimization tools.

4.3.2 Objectives

The primary objectives of the case study were to:

- Assess the impact of the web application on reducing operational costs.
- Evaluate the improvements in delivery time and route efficiency.
- Measure the overall enhancement in logistics management.

4.3.3 Implementation

Data Collection: Honkajoki Oy shared details about their delivery routes, including pickup locations, time windows, and demand for a particular day. They also provided their vehicle data including the capacity of each vehicle. In addition to that, they also shared the manual scheduling and route planning of the same day for a comparative analysis. To measure the impact of transportation delays on product freshness, we consulted with the logistics manager and stakeholders of Honkajoki Oy. Together, we defined three product categories, each with respective freshness penalties as listed below.

Table 4. Product Categories and Freshness Penalties.

Category	**Freshness Penalty Per Hour**
Category 01	2.2 eur/h/container
Category 02	2.9 eur/h/container
Category 03	8.8 eur/h/container *1400.00 eur/container if transportation delays more than 24 hours*

Data Processing and Solution: We prepared a Microsoft Excel template to convert existing logistics data retrieved from their ERP system into a format ready to use for the RouteShaper web application. The data was then uploaded to the system using the bulk import feature, and we generated the solution.

4.3.4 Results

Honkajoki Oy had the requirement to pick up 59 containers for their first plant and 10 containers for another plant from 16 different partner factories. To generate the solution, we considered an ideal fleet of a total of 24 trucks for transportation: 20 at the first plant and 4 at the second plant. If enough vehicles are not available, dummy vehicles can be added to generate the solution. Each truck has the capacity to carry 3 containers. We also considered the following constraints:

- **Time Window:** Each container must be picked up within the same day i.e. for our case study each container must be picked up by 23:59 latest.
- **Vehicle Shift:** All vehicles are available 24 hours.
- **Unit Cost:** Cost per kilometer distance is 2 EURO and cost of waiting is 0.10 EURO per minute.

We split the data for both the processing plants as per the requirements. As our optimization engine, VRP CLI, runs on top of heuristic algorithms, we generated solutions 10 times for each processing plant and calculated their means and standard deviations to measure the efficiency of the solver. Results are presented in the following tables and figures.

Table 5. Solutions for the Processing Plant 01.

Criteria	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	Mean	STD
Cost	13,260	13,261	13,315	13,260	13,260	13,260	13,260	13,260	13,347	13,315	**13,280**	3.1E+01
Distance	6,630	6,631	6,658	6,630	6,630	6,630	6,630	6,630	6,673	6,658	**6,640**	1.6E+01
Time (h)	129	129	130	129	129	129	129	129	130	130	**129**	2.7E-01
Driving	114	114	115	114	114	114	114	114	115	115	**114**	2.7E-01
Loading	15	15	15	15	15	15	15	15	15	15	**15**	0.00
FP	2,770	2,523	2,579	**2,771**	2,544	2,595	2,697	2,726	**2,414**	2,432	**2,605**	124

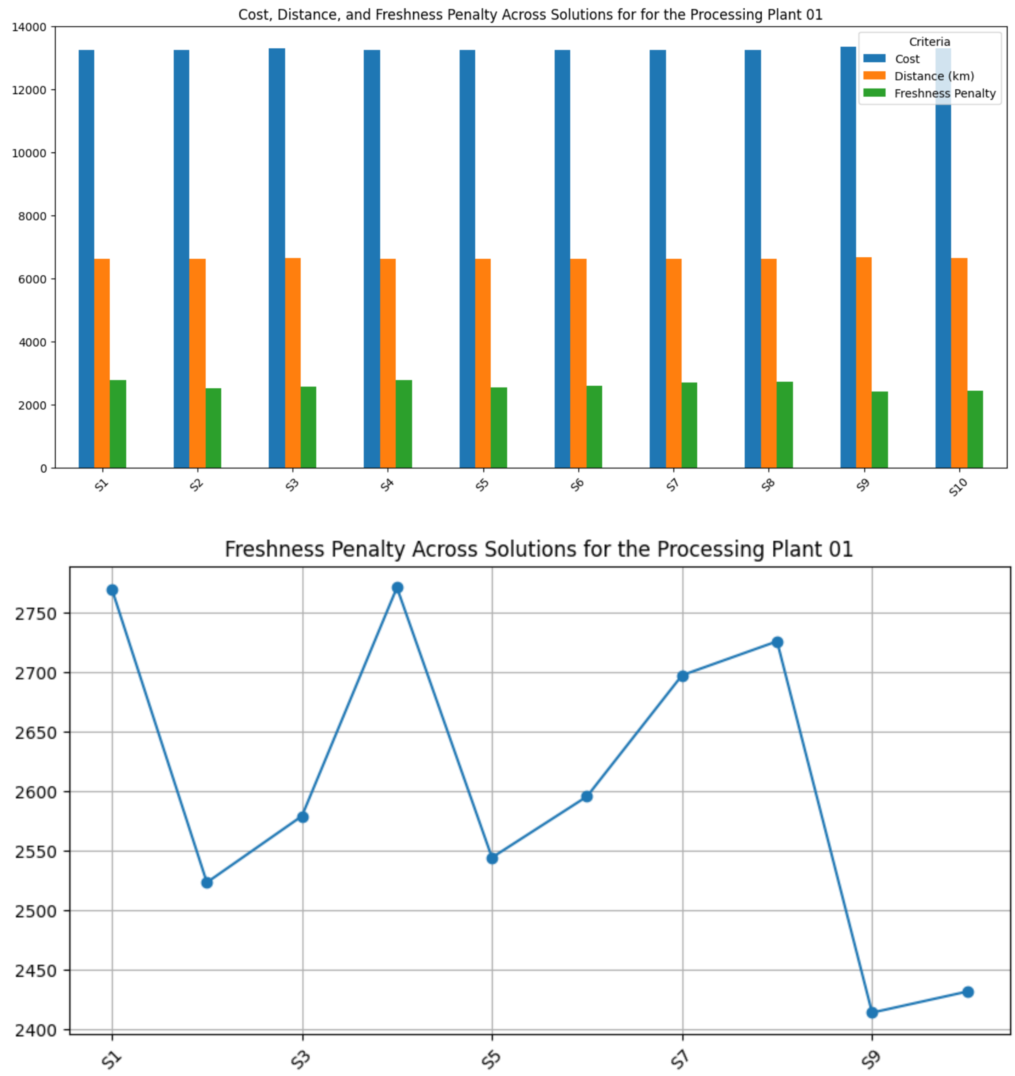
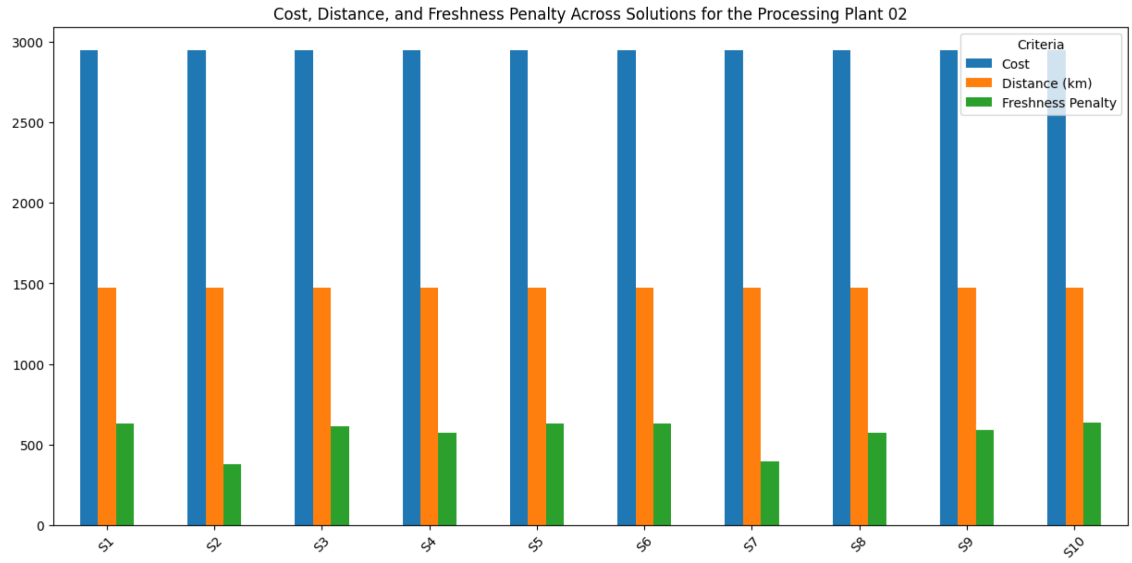


Figure 18. Solutions for the Processing Plant 01.

The results show that RouteShaper generates consistent solutions with negligible standard deviations for all the parameters except the Freshness Penalty (FP). The average cost is 13,280 EURO with a standard deviation of 31, the average distance covered is 6,640 km with a standard deviation of 16, and the average time required is 129 hours with a standard deviation of 0.27. The average FP is 2,605 EURO with a standard deviation of 124, with the FP ranging from 2,414 to 2,771.

Table 6. Solutions for the Processing Plant 02.

Criteria	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**	**Mean**	**STD**
Cost	2,947	2,947	2,947	2,947	2,947	2,947	2,947	2,947	2,947	2,947	**2,947**	4.5E-13
Distance	1,474	1,474	1,474	1,474	1,474	1,474	1,474	1,474	1,474	1,474	**1,474**	2.3E-13
Time (h)	27	27	27	27	27	27	27	27	27	27	**27**	7.1E-15
Driving	24	24	24	24	24	24	24	24	24	24	**24**	3.6E-15
Loading	3	3	3	3	3	3	3	3	3	3	**3**	0.00
FP	628	**375**	615	574	628	628	395	574	587	**635**	**564**	92



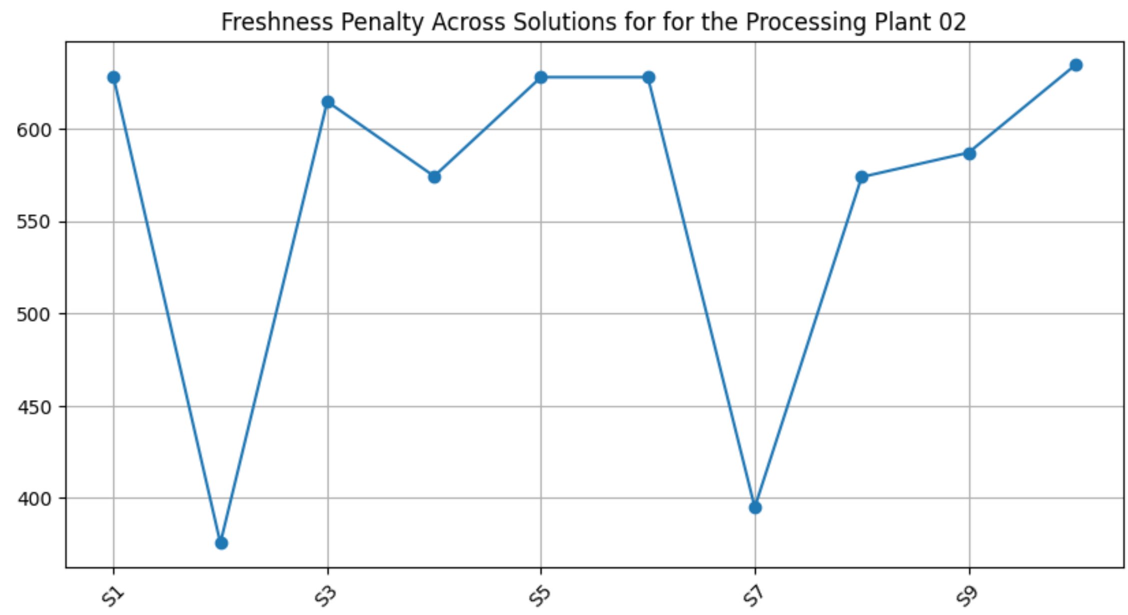


Figure 19. Solutions for the Processing Plant 02.

The results show that RouteShaper generates consistent solutions with negligible standard deviations for all parameters except the Freshness Penalty (FP). The average cost is 2,947 EURO, the average distance covered is 1,474 km, and the average time required is 27 hours, all with nearly zero standard deviations. The average FP is 564 EURO with a standard deviation of 92, with the FP ranging from 375 to 635.

4.3.5 Comparative Analysis and Evaluation

The comparative analysis has been conducted from the quantitative perspective. Honkajoki Oy has provided the manual scheduling followed on that particular day. The key parameters of the manual scheduling have been quantitatively compared to those of the RouteShaper solver to measure the efficiency of the tool.

The manual scheduling of Honkajoki Oy contains key information such as container numbers, partner factory names, processing plants, time when the container is ready for pickup, time of departure, and time of arrival. Using this information, we have calculated key parameters such as total distance traversed, total driving time, and freshness penalty.

The distance from A to B is measured using Google Maps, driving time is calculated from the given data, and freshness penalty is calculated using equation (10). These parameters are then compared with those of the RouteShaper solution.

Table 7. Manual Solution vs RouteShaper Solutions for the Processing Plant 01.

Criteria	Manual Solution	RouteShaper Solution (Mean)
Cost (EURO)	15,860.80	13,279.69
Total Distance (km)	7,930.40	6,639.84
Total Driving Time (h)	156.53	114.41
Freshness Penalty (EURO)	2,127.35	2,605.07

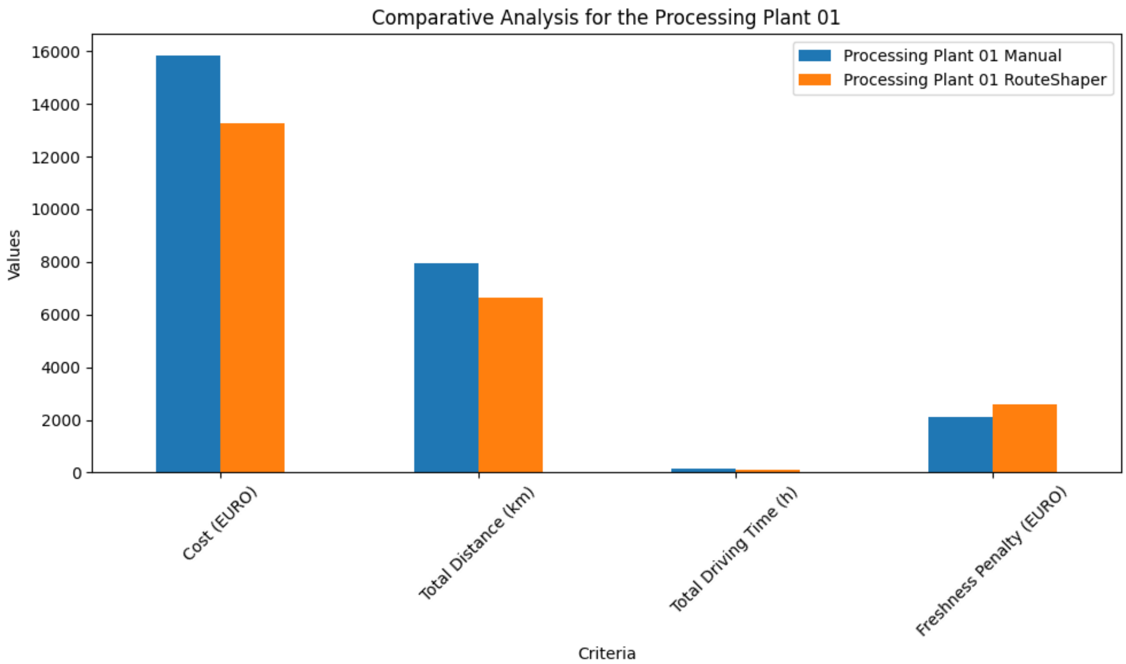


Figure 20. Manual Solution vs RouteShaper Solutions for the Processing Plant 01.

Table 8. Manual Solution vs RouteShaper Solutions for the Processing Plant 02.

Criteria	Manual Solution	RouteShaper Solution (Mean)
Cost (EURO)	2,586.00	2,947.26
Total Distance (km)	1,293.00	1,473.63
Total Driving Time (h)	26.10	24.25
Freshness Penalty (EURO)	359.92	563.82

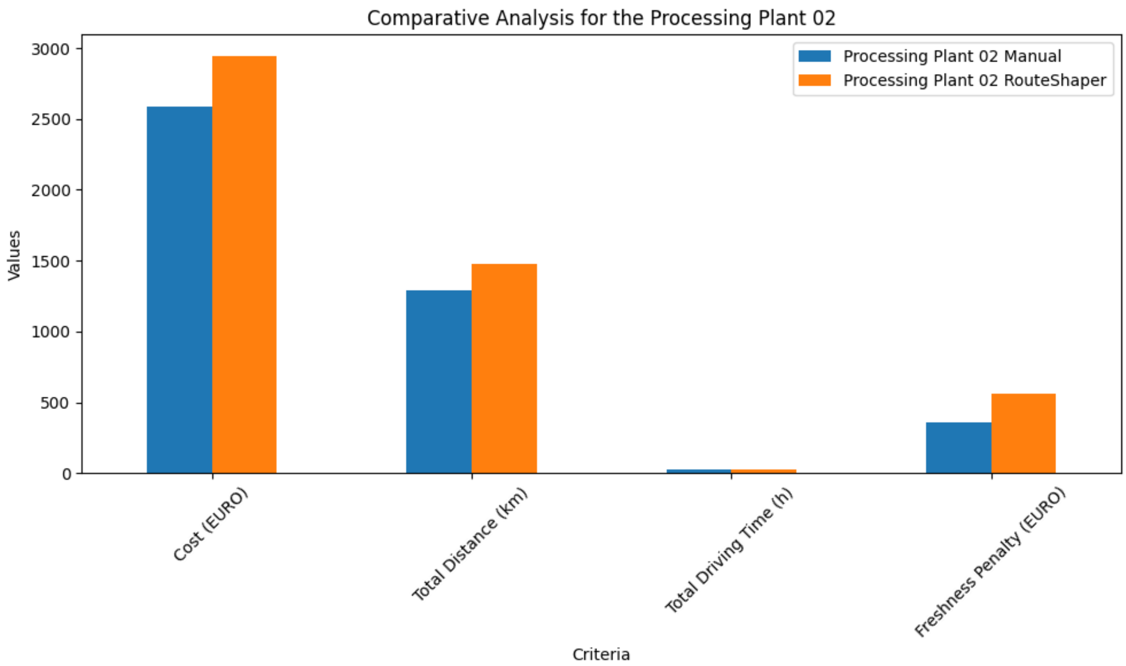


Figure 21. Manual Solution vs RouteShaper Solutions for the Processing Plant 02.

The results show that the RouteShaper solution outperforms the manual solution in every parameter except the freshness penalty for the processing plant 01. On the other hand, for the processing plant 02, the manual scheduling provides a better solution than the RouteShaper solver for every parameter except the total driving time. Since plant 02 has only 10 pickup tasks and plant 01 has 59, it indicates that the RouteShaper solver works better for scheduling when a larger number of tasks are assigned. For a limited number of tasks, manual scheduling works fine.

To summarize, the intuitive and user-friendly interface of RouteShaper, along with features like bulk import, enables logistics managers and practitioners to generate routing and scheduling in minutes with just a few clicks. This not only reduces the labor and time required for manual scheduling but also improves overall operational efficiency. Additionally, the results show that the RouteShaper solution outperforms the manual solution for Processing Plant 01. The RouteShaper solution contributes to approximately a 16.27% reduction in traveling distance and cost, along with 26.91% savings in traveling time. This directly helps in emissions reduction and supports Honkajoki Oy's sustainability goals.

However, manual scheduling provides a better solution for Processing Plant 02, indicating that RouteShaper may not be suitable for scheduling and routing of a small number of tasks. The freshness penalty parameter shows substantial deviations across solutions for both the processing plant, which demands further study.

Overall, the case study with Honkajoki Oy, along with the comparative analysis, demonstrates the tangible benefits of using the RouteShaper web application for optimizing transportation scheduling and vehicle routing for a large number of delivery and pickup tasks.

5 Conclusion and Future Work

This study aimed to develop a robust web-based application, named RouteShaper, to optimize transportation scheduling and vehicle routing within Honkajoki Oy, a leading animal by-product processor company in Finland. The primary purpose was to address the significant challenges in logistics operations, particularly in reducing operational costs, maintaining efficiency, and minimizing environmental impact. RouteShaper successfully integrates advanced algorithms, data analytics, and a novel Freshness Penalty parameter to improve logistics operations.

5.1 Addressing the Problem Statement

The primary objective of this research was to develop an open-source-based web application for vehicle route planning and scheduling. RouteShaper effectively fulfills this objective by providing an efficient solution that optimizes vehicle routes and schedules. The application minimizes travel distances, cost, and time. Which directly implies reduced fuel consumption and emissions, contributing to improved operational efficiency and sustainability at Honkajoki Oy.

To address the sub-questions related to the problem statement:

1. **How can advanced algorithms be effectively integrated into a web-based application to optimize vehicle routing and scheduling?**
   RouteShaper integrates advanced algorithms using the VRP CLI Python package, which solves various forms of the Vehicle Routing Problem (VRP). The VRP CLI employs heuristic algorithms to address different scenarios including multi-depot capabilities, time windows, customer demand, and vehicle capacity, etc. This allows dynamic and responsive optimization of vehicle routes and schedules.

2. **What impact does the optimized routing and scheduling have on operational costs and efficiency?**
   The implementation of RouteShaper led to substantial reductions in travel distances, required time, and operational costs. Empirical testing demonstrated a 16.27% reduction in distance traveled and associated operational costs. Further, 26.91% reduction is observed in total time of traveling. These improvements highlight the significant positive impact on operational efficiency and cost savings.

3. **How to quantify the impact of delays in transportation on the quality of perishable products?**
   The Freshness Penalty parameter is introduced to quantify the impact of delays on raw material quality by calculating the product of the delay in transit and a penalty rate per hour. This parameter ensures that the quality of perishable products is maintained by minimizing delays and reducing the freshness penalty.

5.2 Theoretical Contribution

RouteShaper offers several theoretical contributions compared to existing open-source solutions:

1. **Freshness Penalty Parameter**: One of the unique features of RouteShaper is the incorporation of the Freshness Penalty parameter, which quantifies the impact of delays in transportation on raw material quality. This parameter is critical for industries like Honkajoki Oy, where maintaining the freshness of perishable goods is essential. By minimizing the freshness penalty, RouteShaper ensures higher product quality and reduces waste, a feature not commonly found in existing solutions.

2. **Real-Time Data Integration**: RouteShaper utilizes OpenRouteService API to generate real time routing matrix. This enables RouteShaper to generate more accurate routing and scheduling. On the other hand, most of the other open source tool do not have this feature of real time routing matrix calculation.

3. **Comparative Feature Analysis**: Compared to existing open-source solutions like ODL Studio and Spreadsheet Solver, RouteShaper has some unique features (ODL Studio, 2014; Erdoğan, 2017). In addition to that, RouteShaper demonstrates better performance in some other critical parameters compared to the existing tool. A table is provided below showing comparison of different features and parameters of RouteShaper with other tools.

Table 9. Comparison of the features of RouteShaper with other open-source tools.

Criteria	**RouteShaper**	**ODL Studio**	**Spreadsheet Solver**
Technology	Python	Java	VBA
Optimization Engine	VRP CLI	jsprit	Custom
Solution Generation Time	Less than 1 minutes *For 50+ jobs*	No data available	15 minutes for 50 customers
Routing Matrix	Real Time, using ORS API	Not Real Time	Not Real Time
Freshness Penalty Calculation	Available	Not Available	Not Available
Reporting	In CSV format	In PDF format	In Excel

5.3 Practical Contributions

Unlike many other open-source tools, RouteShaper provides a user-friendly interface and dashboard. This helps practitioners to generate route planning and scheduling easily and quickly. Additionally, the optimized routing and scheduling capabilities of RouteShaper contribute to sustainability goals by reducing carbon emissions and improving resource utilization. The application aligns with Honkajoki Oy's commitment to environmental sustainability and demonstrates the potential for similar applications in other industries.

5.4 Future Work

While this research has achieved its objectives, several areas for future work have been identified. Addressing these areas can further enhance the application and broaden its applicability.

The newly introduced parameter, Freshness Penalty (FP), has been applied in the analytics and reporting phase after solving the VRP. Consequently, we observed significant deviations in the value of FP. To address this issue, a new variant of VRP can be developed in the future where the parameter FP is to be included in the objective function. This new variant of VRP can be helpful for optimizing the timely delivery of perishable goods at minimum cost. In addition to that, future work can explore the integration of more advanced optimization algorithms, such as machine learning-based approaches, to further improve the efficiency and accuracy of route planning. Adaptive algorithms that learn from historical data and real-time conditions can provide more dynamic and responsive solutions.

To summarize, this study has demonstrated the potential of integrating advanced algorithms and data analytics in optimizing transportation scheduling and vehicle routing. By addressing key challenges in logistics operations and focusing on both efficiency and sustainability, the developed application RouteShaper provides a valuable tool for Honkajoki Oy and potentially other industries. Continued research and development in this area hold promise for further advancements and broader impacts in the field of logistics management.

References

Alnaggar, A., Gzara, F., & Bookbinder, J. H. (2021). Crowdsourced delivery: A review of platforms and academic literature. *Omega*, *98*, 102139. https://doi.org/10.1016/j.omega.2019.102139

Alves, F., Pacheco, F., Rocha, A. M., Pereira, A. I., & Leitão, P. (2021). Solving a logistics system for vehicle routing problem using an open-source tool. *Lecture Notes in Computer Science*, 397–412. https://doi.org/10.1007/978-3-030-86976-2_27

Awad, M., Ndiaye, M., & Osman, A. (2020). Vehicle routing in Cold Food Supply Chain Logistics: A literature review. *The International Journal of Logistics Management*, *32*(2), 592–617. https://doi.org/10.1108/ijlm-02-2020-0092

Bogyrbayeva, A., Meraliyev, M., Mustakhov, T., & Dauletbayev, B. (2024). Machine learning to solve vehicle routing problems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, *25*(6), 4754–4772. https://doi.org/10.1109/tits.2023.3334976

Bozanta, A., Cevik, M., Kavaklioglu, C., Kavuk, E. M., Tosun, A., Sonuc, S. B., Duranel, A., & Basar, A. (2022). Courier routing and assignment for Food Delivery Service using reinforcement learning. *Computers & Industrial Engineering*, *164*, 107871. https://doi.org/10.1016/j.cie.2021.107871

Builuk, I. (2023). A new solver for rich vehicle routing problem. computer software. Retrieved from https://doi.org/10.5281/zenodo.4624037.

Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, *12*(4), 568–581. https://doi.org/10.1287/opre.12.4.568

Cordeau, J., Gendreau, M., & Laporte, G. (1997). A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, *30*(2), 105–119.

Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, *6*(1), 80–91. https://doi.org/10.1287/mnsc.6.1.80

Dorigo, M., & Stützle, T. (2004). *Ant colony optimization*. MIT Press.

Erdoğan, G. (2017). An open source spreadsheet solver for vehicle routing problems. *Computers & Operations Research*, *84*, 62–72. https://doi.org/10.1016/j.cor.2017.02.022

European Commission. (2021, April 12). Transport and the green deal. https://commission.europa.eu/strategy-and-policy/priorities-2019-2024/european-green-deal/transport-and-green-deal_en

Google. (2010). OR-Tools. *Google Developers*. computer software. Retrieved from https://developers.google.com/optimization/routing/vrp.

Grand View Research. (2023). Route Optimization Software Market Size. https://www.grandviewresearch.com/industry-analysis/route-optimization-software-market-report

GraphHopper. (2022). GraphHopper routing engine. https://www.graphhopper.com/

Honkajoki Oy - About Us. Honkajoki Oy. (2024, May 21). https://honkajokioy.fi/en/about-us/

Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, *59*(3), 345–358. https://doi.org/10.1016/0377-2217(92)90192-c

Li, M., Jia, N., Lenzen, M., Malik, A., Wei, L., Jin, Y., & Raubenheimer, D. (2022). Global food-miles account for nearly 20% of total food-systems emissions. *Nature Food*, *3*(6), 445–453. https://doi.org/10.1038/s43016-022-00531-w

Li, X., Yan, P., Yu, K., Li, P., & Liu, Y. (2024). Parcel consolidation approach and routing algorithm for last-mile delivery by Unmanned Aerial Vehicles. *Expert Systems with Applications*, *238*, 122149. https://doi.org/10.1016/j.eswa.2023.122149

Li, Y., Chen, Y., Guo, G., Wu, H., & Yuan, Z. (2022). Integrated routing for a vehicle-robot pickup and delivery system with time constraints. *International Conference on Industrial Engineering and Applications (ICIEA 2022)*.

McKinnon, A. C., Cullinane, S., Browne, M., & Whiteing, A. (2010). *Green logistics improving the environmental sustainability of Logistics*. Kogan Page Publishers.

Mehra, A., Saha, S., Raychoudhury, V., & Mathur, A. (2024). DeliverAI: A Distributed Path-Sharing Network based solution for the Last Mile Food Delivery Problem. *arXiv*.

Moghdani, R., Salimifard, K., Demir, E., & Benyettou, A. (2021). The Green Vehicle Routing Problem: A Systematic Literature Review. *Journal of Cleaner Production*, *279*, 123691. https://doi.org/10.1016/j.jclepro.2020.123691

ODL Studio. (2014). Open Door Logistics Studio. computer software. Retrieved from https://www.opendoorlogistics.com.

Rossi, S., Colicchia, C., Cozzolino, A., & Christopher, M. (2013). The logistics service providers in eco-efficiency innovation: An empirical study. *Supply Chain Management: An International Journal*, *18*(6), 583–603.

Savelsbergh, M. W., & Sol, M. (1995). The general pickup and delivery problem. *Transportation Science*, *29*(1), 17–29. https://doi.org/10.1287/trsc.29.1.17

Silva, A. S., Alves, F., Diaz de Tuesta, J. L., Rocha, A. M., Pereira, A. I., Silva, A. M., & Gomes, H. T. (2023). Capacitated waste collection problem solution using an open-source tool. *Computers*, *12*(1), 15. https://doi.org/10.3390/computers12010015

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, *35*(2), 254–265. https://doi.org/10.1287/opre.35.2.254

Stančić, N., Kovačević, J., Cvijetinović, Ž., Brodić, N., & Mihajlović, D. (2023). Solving the Vehicle Routing Problem In the Open-Source Software 'ODL Studio.' *IPSI Transactions on Advanced Research*, *19*(1), 5–12.

Tan, S.-Y., & Yeh, W.-C. (2021). The vehicle routing problem: State-of-the-art classification and Review. *Applied Sciences*, *11*(21), 10295. https://doi.org/10.3390/app112110295

Toth, P., & Vigo, D. (2014). *Vehicle routing: Problems, methods, and applications*. Society for Industrial and Applied Mathematics (SIAM).

von Krogh, G., & von Hippel, E. (2006). The promise of research on open source software. *Management Science*, *52*(7), 975–983. https://doi.org/10.1287/mnsc.1060.0560

Wieringa, R. J. (2014). *Design Science Methodology for Information Systems and Software Engineering*. https://doi.org/10.1007/978-3-662-43839-8

Zhang, H., Ge, H., Yang, J., & Tong, Y. (2021). Review of vehicle routing problems: Models, classification and solving algorithms. *Archives of Computational Methods in Engineering*, *29*(1), 195–221. https://doi.org/10.1007/s11831-021-09574-x

Zhou, X., Tong, L., Mahmoudi, M., Zhuge, L., Yao, Y., Zhang, Y., Shang, P., Liu, J., & Shi, T. (2018). Open-source vrplite package for vehicle routing with pickup and delivery: A path finding engine for scheduled transportation systems. *Urban Rail Transit*, *4*(2), 68–85. https://doi.org/10.1007/s40864-018-0083-7

Appendices

Appendix 1. Links for Source Code and User Guide Tutorial

Backend: https://github.com/r1azmh/vrp-backend

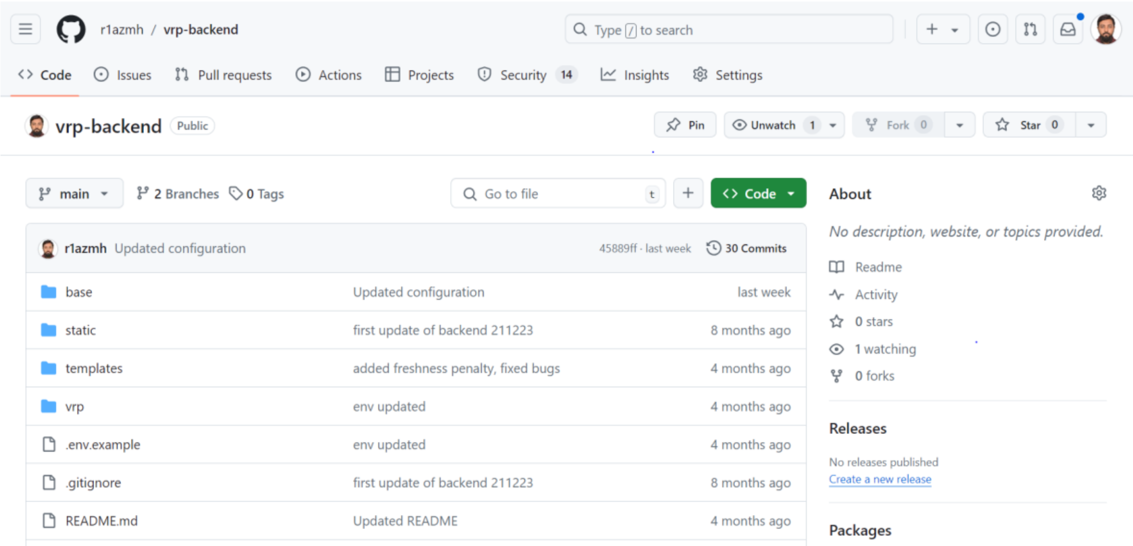


Figure 22. GitHub page for the backend source code.

Frontend: https://github.com/r1azmh/vrp-frontend

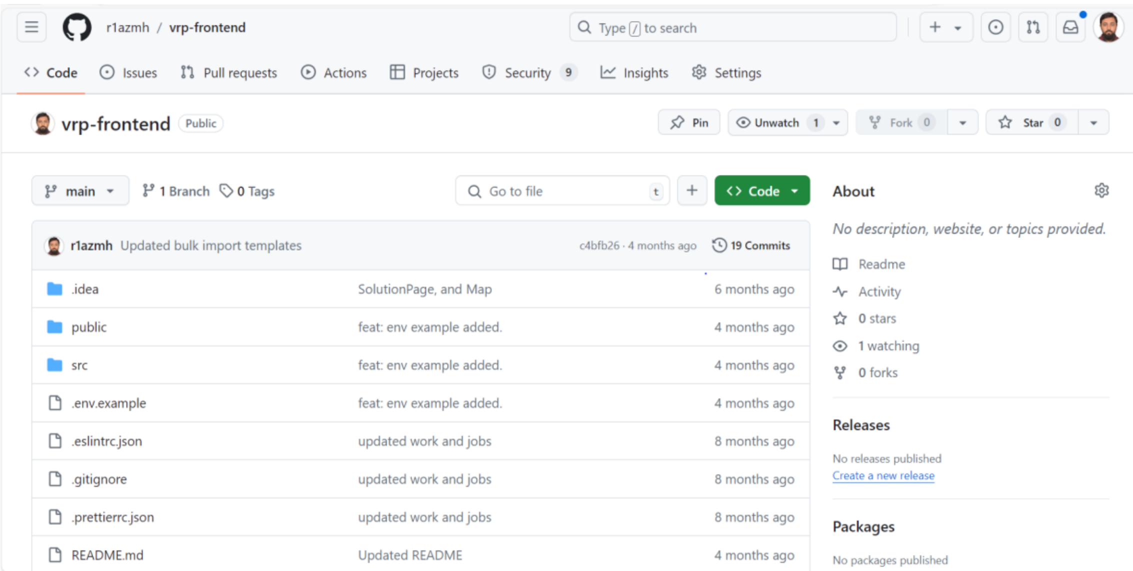


Figure 23. GitHub page for the frontend source code.

Video Tutorial (Individual Entry): https://youtu.be/6cJ2dfyzdoo

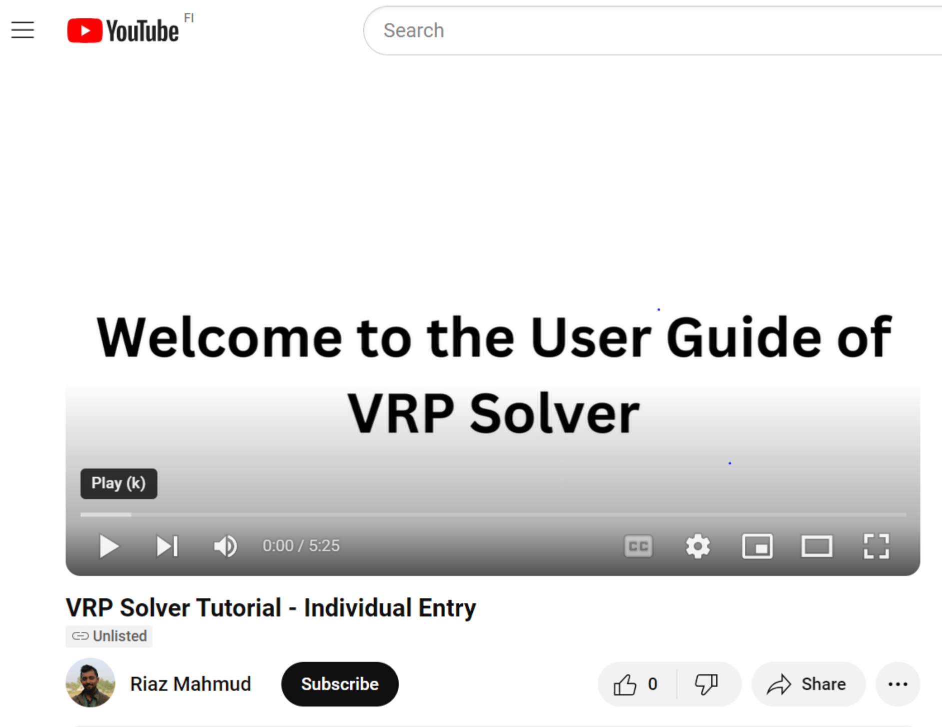


Figure 24. User guide video tutorial for individual entry.

Video Tutorial (Bulk Entry): https://youtu.be/1_MVMrGe0Q8

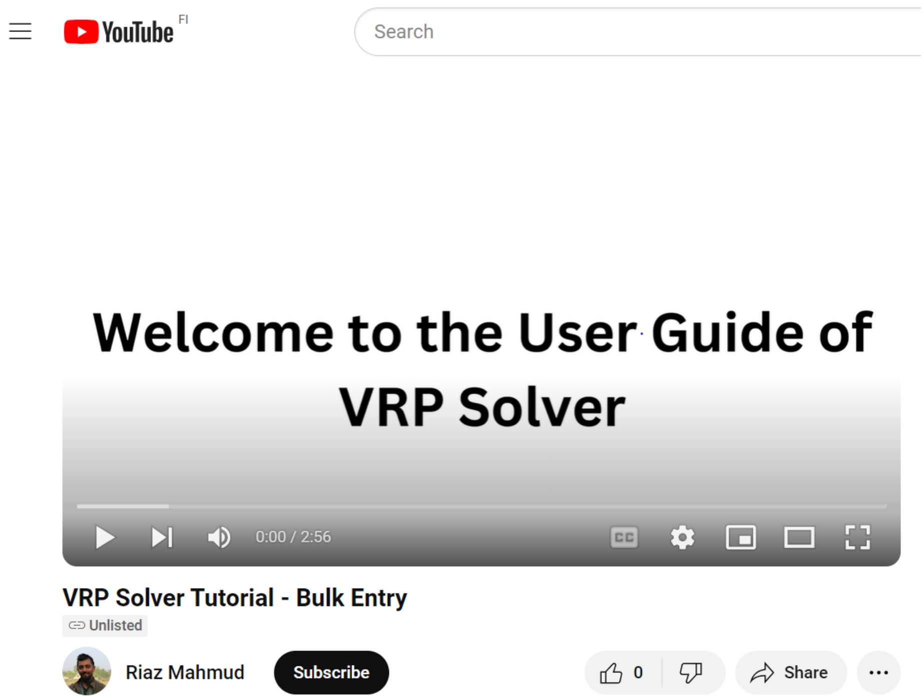


Figure 25. User guide video tutorial for bulk entry.