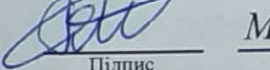


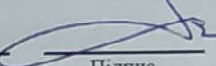
ДИПЛОМНА РОБОТА МАГІСТРА

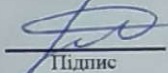
на тему Інформаційна система планування найкращого шляху для доставки вантажу за допомогою задачі комівояжера

Галузь знань 12 – Інформаційні технології
Шифр і назва галузі знань

Спеціальність 122 – Комп'ютерні науки
Шифр і назва спеціальності

Виконав: студент 2 курсу, група КНм-19-1  М.С. Грицюк
Підпис Ініціали, прізвище

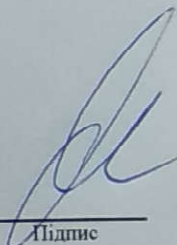
Керівник: к.т.н., доцент кафедри КНІТ  О.А. Пасічник
Підпис Ініціали, прізвище

Нормоконтроль: к.т.н., доцент кафедри КНІТ  Р.О. Багрій
Підпис Ініціали, прізвище

До захисту допускаю:

Зав. кафедри КНІТ, д.т.н., професор

7 12 2020 р.

 О.В. Бармак
Підпис Ініціали, прізвище

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет програмування та комп'ютерних і телекомунікаційних систем

Кафедра комп'ютерних наук та інформаційних технологій

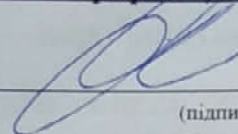
Освітній ступінь магістр

Галузь знань 12 – Інформаційні технології

Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук та інформаційних технологій



(підпис)

д.т.н., професор О.В. Барма

« 7 » 9 2020 року

ЗАВДАННЯ

НА ДИПЛОМНУ РОБОТУ МАГІСТРА

1. Тема дипломної роботи магістра: «Інформаційна система планування найкращого шляху для доставки вантажу за допомогою задачі комівояжера»

2. Завдання видано студенту Грицюку Миколі Станіславовичу

(прізвище, ім'я, по батькові)

3. Керівник роботи к.т.н., доцент Пасічник Олександр Анатолійович

(прізвище, ім'я, по батькові)

4. Затверджені наказом університету від « 3 » 9 2020 р. № 22

5. Зміст пояснювальної записки (перелік питань, які треба розробити):

Мета роботи – полягає у реалізації інформаційної системи для планування найкращого шляху доставки вантажу за допомогою задачі комівояжера. Для досягнення мети необхідно дослідити існуючі підходи оптимізації і пошуку найкращого шляху, а також створити відповідну інформаційну систему й дослідити її ефективність.

Реферат

Дипломна магістерська робота присвячена розробці інформаційної системи планування найкращого шляху для доставки вантажу за допомогою вирішення задачі комівояжера.

Актуальність теми. Розробка нових інформаційних технологій дозволяє розвивати суспільство в тих чи інших напрямках. Вони активно перетворюють інші технології матеріального і нематеріального виробництва, в кінцевому підсумку формуючи новий стиль роботи, спосіб життя в цілому. Суть інформаційних технологій становлять методи і засоби формування та підтримки інформаційних потоків у системах управління об'єктами.

Глобалізація економіки супроводжується небувалими раніше темпами зростання торгівлі. Світовий обсяг експорту за 50 років виріс в 10 разів і продовжує збільшуватися більш високими темпами, ніж ВВП. Щоденний обсяг валютних операцій перевищує \$ 1,5 трлн проти \$ 15 млрд в 1973 році. У цих умовах максимально зростає значення світової транспортної мережі. Транспорт служить матеріальною базою виробничих зв'язків між окремими територіями, виступає як фактор, що організує світовий економічний простір і забезпечує подальше географічне розподіл праці.

Кожен великий бізнес, сфера діяльності якого тісно пов'язана з великими грошима, будь то виробництво, оптова чи широка роздрібна мережа, будує свій бізнес на правильно сформованому вантажоперевезенню. Вчасно доставлений вантаж підтримує в стабільності весь процес бізнесу. Якщо на виробництво не буде вчасно доставлений хоч один компонент все зупиниться і почнуться збитки. У тій же оптовій або роздрібній мережі відсутність певного товару вказує дорогу до конкурентів. Клієнтам не важливо з якої причини є необхідних товарів, їм важливо володіти ними.

Сфера вантажоперевезень активно розвивається, завдяки тому, що через Україну проходять великі транспортні шляхи, дедалі більше людей розпочинають свою кар'єру саме в цій сфері. Завдяки сучасним технологіям,

вантажоперевезення стає максимально ефективним і прибутковим. Подання асортименту в інтернеті є перспективним напрямком розвитку бізнесу. Невеликого стартового капіталу цілком достатньо, щоб відкрити інтернет-магазин, який стане основою для подальшого розширення компанії та досягнення рівня роздрібної мережі.

Автомобільні перевезення широко використовується у всіх галузях економіки, - в торгівлі, виробництві, сільському господарстві.

Сьогодні розвиток технології дозволяє автоматизувати ті процеси, які ще декілька років тому не могли б існувати без підтримки людини. Доволі часто ми чуємо про нові сервіси, які вдосконалюють ті процеси, які людина не здатна вирішувати на високому рівні. Наприклад всім знайомий сервіс для виклику авто (таксі) Uklon, це досить хороший приклад сервісу, який вирішує проблеми, будування найкращого маршруту для водіїв таксі, за допомогою сучасних технологій, що в свою чергу дозволяє скоротити витрати на паливо, задовольнити потреби клієнта, а також дає змогу виконати більше замовлень за один і той же час.

Сфера вантажоперевезень не виключення, завдяки вседоступності до мережі Інтернет, з'являється можливість до реалізації нових ідей, які змінюють всі процеси, які були до цього запроваджені в тій чи іншій галузі.

Вантажоперевезення в Україні займає найбільшу частку серед усіх видів транспортних послуг. Мінімальна частина задіяних в даному процесі перевізників працює на контрактній основі з великими підприємствами-замовниками. Решта учасників ринку взаємодіють з клієнтами без підписання угод про довгострокове співробітництво. Такий варіант двостороннього партнерства є вкрай ризикованим, тому що пошук замовника або перевізника на дошках оголошень може завершитися зустріччю з шахраями. В результаті транспортна організація залишається без заробітку, а замовник позбавляється вантажу.

Чимало водіїв, які працюють в компаніях, що надають послуги вантажоперевезень задумуються про відкриття своєї справи в даній сфері, але

для того, щоб бути незалежним від транспортної компанії недостатньо володіти вантажним автомобілем, так як однією з основних послуг, які надають транспортні компанії - це пошук замовлень. Враховуючи те, що замовлення, які надаються компаніями, не завжди відповідають очікуванням водіїв, а саме місце завантаження може знаходитися в десятках, а то і в сотнях кілометрів від місця розташування водія. Не всі компанії компенсують кошти на паливе, яке використовується для того, щоб дістатися місце завантаження. В результаті водії зобов'язані сплачувати послуги оператора, а також компенсувати з власної кишені добирання до місця завантаження, що суттєво знижує їхній дохід.

Світова глобалізація формує значні і складні виклики щодо забезпечення логістичної підтримки сучасного виробництва. Зв'язку з одночасним розвитком інформаційних технологій, їх високою ефективністю, пристосованістю до вирішення формалізованих задач широкого спектру застосування, найбільш доцільним є вирішення задач організації перевезень на основі інформаційних технологій та систем.

Завдяки сучасним технологіям можна змінити дану ситуацію, а саме надати водієві можливість шукати замовлення напряму, без посередників, в ролі яких виступають різного типу транспортні компанії. Це дозволить ефективно планувати маршрут, а також стати незалежним від транспортних компаній.

Мета і задачі роботи. Мета роботи полягає у реалізація інформаційної системи для пошуку найкращого маршруту для вантажного перевезення за допомогою задачі комівояжера.

Для досягнення поставленої мети визначенні наступні задачі-дослідження:

- провести аналіз існуючих методів, технологій та рішень пошуку найкращого маршруту доставки вантажу, що ґрунтується на задачі комівояжера;
- вдосконалення існуючих методик використання задачі комівояжера для пошуку найкращого маршруту;

- розробити інформаційну систему побудови найкращого маршруту вантажних перевезень, яка поєднує розв’язки задачі комівояжера з сучасними картографічними сервісами;
- виконати експериментальну перевірку інформаційної системи планування найкращого шляху для доставки вантажу.

Об’єкт дослідження – процес побудови шляху доставки вантажу з використанням інформаційних технологій.

Предмет дослідження – моделі, методи, підходи та засоби інформаційної технології побудови маршруту доставки вантажів.

Методи дослідження. Для розв’язання поставлених задач використовуються основні положення методів аналізу даних, дискретної математики, логістики; для вдосконалення існуючих методик пошуку найкращого маршруту методи теорії графів, дистанційного зондування Землі; для реалізації інформаційної технології методології проектування інформаційних систем та об’єктно орієнтований підхід.

Наукова новизна одержаних результатів. В результаті проведеної роботи були отримані наступні результати:

- вдосконалення існуючих методик використання задачі комівояжера для пошуку найкращого маршруту у напрямку їх поєднання з сучасними картографічними сервісами.

Практичне значення одержаних результатів. Завдяки розробленій інформаційній системі було розроблено відповідне програмне забезпечення, яке дозволило автоматизувати весь процес логістичної підтримки процесів доставки вантажу. Застосування інформаційної системи дає можливість отримати комплексний логістичний веб-інструментарій.

Апробація результатів дипломної роботи магістра та публікації. За темою дипломної роботи магістра автором опубліковано наукову публікацію у фаховому виданні, включеному в перелік МОН України, зокрема стаття М.С. Грицюк, О.А. Пасічник, Т.К. Скрипник Інформаційна система планування

найкращого шляху для доставки вантажу за допомогою задачі комівояжера // “Вісник” Хмельницького національного університету 2020 (подано до редакції)

Структура та обсяг роботи. Дипломна робота магістра складається з завдання, реферату, змісту, переліку скорочень, вступу, 4 розділів, висновків, переліку посилань із 35 найменувань та 5 додатків. Загальний обсяг дипломної роботи магістра становить 95 сторінок, з них 73 сторінка основного тексту та 19 сторінок додатків. У роботі наведено 51 рисуноків та 1 таблиця.

Ключові слова: інформаційна система, пошуку маршруту, картографічний сервіс

Зміст

Вступ	5
Розділ 1	
Характеристика предметної області і постановка задачі	10
1.1 Теорія логістики	10
1.2 Алгоритми вирішення задачі комівояжера	11
1.3 Порівняльний аналіз алгоритмів вирішення логістичних завдань	17
1.4 Інформаційне забезпечення систем пошуку маршруту	19
1.5 Висновки до розділу та постановка задачі	24
Розділ 2	
Розробка технологій пошуку маршруту вантажних перевезень	26
2.1 Мурашині алгоритми	26
2.2 Математична модель пошуку оптимального маршруту	33
2.3 Метод розрахунку вартості вантажного перевезення	36
2.4 Висновки до розділу	41
Розділ 3	
Програмна реалізація	42
3.1 Опис інформаційної системи	42
3.2 Структура і функціональне призначення модулів системи, їх взаємозв'язок	46
3.3 Розробка програмних модулів	55
3.4 Висновки до розділу	60
Розділ 4	
Апробація програмної інформаційної системи	61

4.1	Порядок використання інформаційної системи	61
4.2	Порівняльний аналіз додати декілька маршрутів	64
4.3	Висновки до розділу	67
	Загальні висновки	68
	Перелік посилань	70
	ДОДАТКИ	

Перелік скорочень

Скорочення, термін, позначення	Пояснення
ІС	Інформаційна система
БД	База даних
ДРМ	Дипломна робота магістра
MVC	Model View Controller

Вступ

Актуальність теми. Розробка нових інформаційних технологій дозволяє розвивати суспільство в тих чи інших напрямках. Вони активно перетворюють інші технології матеріального і нематеріального виробництва, в кінцевому підсумку формуючи новий стиль роботи, спосіб життя в цілому. Суть інформаційних технологій становлять методи і засоби формування та підтримки інформаційних потоків у системах управління об'єктами.

Глобалізація економіки супроводжується небувалими раніше темпами зростання торгівлі. Світовий обсяг експорту за 50 років виріс в 10 разів і продовжує збільшуватися більш високими темпами, ніж ВВП. Щоденний обсяг валютних операцій перевищує \$ 1,5 трлн проти \$ 15 млрд в 1973 році. У цих умовах максимально зростає значення світової транспортної мережі. Транспорт служить матеріальною базою виробничих зв'язків між окремими територіями, виступає як фактор, що організує світовий економічний простір і забезпечує подальше географічне розподіл праці.

Кожен великий бізнес, сфера діяльності якого тісно пов'язана з великими грошима, будь то виробництво, оптова чи широка роздрібна мережа, будує свій бізнес на правильно сформованому вантажоперевезенню. Вчасно доставлений вантаж підтримує в стабільності весь процес бізнесу. Якщо на виробництво не буде вчасно доставлений хоч один компонент все зупиниться і почнуться збитки. У тій же оптовій або роздрібній мережі відсутність певного товару вказує дорогу до конкурентів. Клієнтам не важливо з якої причини є необхідних товарів, їм важливо володіти ними.

Сфера вантажоперевезень активно розвивається, завдяки тому, що через Україну проходять великі транспортні шляхи, дедалі більше людей розпочинають свою кар'єру саме в цій сфері. Завдяки сучасним технологіям, вантажоперевезення стає максимально ефективним і прибутковим. Подання асортименту в інтернеті є перспективним напрямком розвитку бізнесу. Невеликого стартового капіталу цілком достатньо, щоб відкрити

інтернет-магазин, який стане основою для подальшого розширення компанії та досягнення рівня роздрібною мережі.

Автомобільні перевезення широко використовується у всіх галузях економіки, - в торгівлі, виробництві, сільському господарстві.

Сьогодні розвиток технології дозволяє автоматизувати ті процеси, які ще декілька років тому не могли б існувати без підтримки людини. Доволі часто ми чуємо про нові сервіси, які вдосконалюють ті процеси, які людина не здатна вирішувати на високому рівні. Наприклад всім знайомий сервіс для виклику авто (таксі) Uklon, це досить хороший приклад сервісу, який вирішує проблеми, будування найкращого маршруту для водіїв таксі, за допомогою сучасних технологій, що в свою чергу дозволяє скоротити витрати на паливе, задовольнити потреби клієнта, а також дає змогу виконати більше замовлень за один і той же час.

Сфера вантажоперевезень не виключення, завдяки вседоступності до мережі Інтернет, з'являється можливість до реалізації нових ідей, які змінюють всі процеси, які були до цього запроваджені в тій чи іншій галузі.

Вантажоперевезення в Україні займає найбільшу частку серед усіх видів транспортних послуг. Мінімальна частина задіяних в даному процесі перевізників працює на контрактній основі з великими підприємствами-замовниками. Решта учасників ринку взаємодіють з клієнтами без підписання угод про довгострокове співробітництво. Такий варіант двостороннього партнерства є вкрай ризикованим, тому що пошук замовника або перевізника на дошках оголошень може завершитися зустріччю з шахраями. В результаті транспортна організація залишається без заробітку, а замовник позбавляється вантажу.

Чимало водіїв, які працюють в компаніях, що надають послуги вантажоперевезень задумуються про відкриття своєї справи в даній сфері, але для того, щоб бути незалежним від транспортної компанії недостатньо володіти вантажним автомобілем, так як однією з основних послуг, які надають

транспортні компанії - це пошук замовлень. Враховуючи те, що замовлення, які надаються компаніями, не завжди відповідають очікуванням водіїв, а саме місце завантаження може знаходитися в десятках, а то і в сотнях кілометрів від місця розташування водія. Не всі компанії компенсують кошти на паливе, яке використовується для того, щоб дістатися місце завантаження. В результаті водії зобов'язані сплачувати послуги оператора, а також компенсувати з власної кишені добирання до місця завантаження, що суттєво знижує їхній дохід.

Світова глобалізація формує значні і складні виклики щодо забезпечення логістичної підтримки сучасного виробництва. Зв'язку з одночасним розвитком інформаційних технологій, їх високою ефективністю, пристосованістю до вирішення формалізованих задач широкого спектру застосування, найбільш доцільним є вирішення задач організації перевезень на основі інформаційних технологій та систем.

Завдяки сучасним технологіям можна змінити дану ситуацію, а саме надати водієві можливість шукати замовлення напряму, без посередників, в ролі яких виступають різного типу транспортні компанії. Це дозволить ефективно планувати маршрут, а також стати незалежним від транспортних компаній.

Мета і задачі роботи. Мета роботи полягає у реалізація інформаційної системи для пошуку найкращого маршруту для вантажного перевезення за допомогою задачі комівояжера.

Для досягнення поставленої мети визначенні наступні задачі-дослідження:

- провести аналіз існуючих методів, технологій та рішень пошуку найкращого маршруту доставки вантажу, що ґрунтується на задачі комівояжера;
- вдосконалення існуючих методик використання задачі комівояжера для пошуку найкращого маршруту;

- розробити інформаційну систему побудови найкращого маршруту вантажних перевезень, яка поєднує розв’язки задачі комівояжера з сучасними картографічними сервісами;
- виконати експериментальну перевірку інформаційної системи планування найкращого шляху для доставки вантажу.

Об’єкт дослідження – процес побудови шляху доставки вантажу з використанням інформаційних технологій.

Предмет дослідження – моделі, методи, підходи та засоби інформаційної технології побудови маршруту доставки вантажів.

Методи дослідження. Для розв’язання поставлених задач використовуються основні положення методів аналізу даних, дискретної математики, логістики; для вдосконалення існуючих методик пошуку найкращого маршруту методи теорії графів, дистанційного зондування Землі; для реалізації інформаційної технології методології проектування інформаційних систем та об’єктно орієнтований підхід.

Наукова новизна одержаних результатів. В результаті проведеної роботи були отримані наступні результати:

- вдосконалення існуючих методик використання задачі комівояжера для пошуку найкращого маршруту у напрямку їх поєднання з сучасними картографічними сервісами.

Практичне значення одержаних результатів. Завдяки розробленій інформаційній системі було розроблено відповідне програмне забезпечення, яке дозволило автоматизувати весь процес логістичної підтримки процесів доставки вантажу. Застосування інформаційної системи дає можливість отримати комплексний логістичний веб-інструментарій.

Апробація результатів дипломної роботи магістра та публікації. За темою дипломної роботи магістра автором опубліковано наукову публікацію у фаховому виданні, включеному в перелік МОН України, зокрема стаття М.С. Грицюк, О.А. Пасічник, Т.К. Скрипник Інформаційна система планування

найкращого шляху для доставки вантажу за допомогою задачі комівояжера // "Вісник" Хмельницького національного університету 2020 (подано до редакції)

Структура та обсяг роботи. Дипломна робота магістра складається з завдання, реферату, змісту, переліку скорочень, вступу, 4 розділів, висновків, переліку посилань із 35 найменувань та 5 додатків. Загальний обсяг дипломної роботи магістра становить 95 сторінок, з них 73 сторінка основного тексту та 19 сторінок додатків. У роботі наведено 51 рисуноків та 1 таблиця.

Розділ 1

Характеристика предметної області і постановка задачі

1.1 Теорія логістики

Термін «логістика» походить від грецького слова «logistike», що означає «мислення, розрахунок, доцільність». У Римській імперії розуміли цей термін як «розподіл продуктів харчування». У Візантії логістику вважали способом організації військового постачання і управління армією. Історично склалося, що логістика як практична діяльність розвивалася завдяки військовій справі. У першому тисячолітті нашої ери у військовому лексиконі ряду країн з логістикою пов'язували діяльність з управління перевезеннями, озброєння армії, планування і постачання військ матеріальними ресурсами, утримання запасів. На початку ХХ століття логістика вважалася військовою наукою. Транспортні принципи і моделі широко використовувалися в ході Першої та Другої світових воєн. У період Другої світової війни логістика активно застосовувалася в матеріально-технічному постачанні армії США, що дозволило забезпечити чітку взаємодію військової промисловості, тилових і фронтових постачальницьких баз і транспорту. Подібно дослідженню операцій, математичної оптимізації, мережевим моделям і іншим методам прикладної математики, який демонстрував невідому свою ефективність у військовій області, логістика поступово перейшла в сферу господарської практики і стала широко використовуватися в економіці [1].

Транспортна логістика - це оптимізація управління транспортуванням вантажів, тобто виконання операцій переміщення та зберігання сировини, напівфабрикатів, об'єктів незавершеного виробництва, готової продукції з місць походження до місць споживання з використанням транспортних засобів.

Транспорт є галуззю матеріального виробництва. Особливість транспорту полягає в тому, що він не переробляє сировину і не створює матеріальних продуктів, а результатом його діяльності є послуги. Транспортні послуги

виконуються спеціалізованими транспортними підприємствами або ж підприємствами-виробниками, оптовими і роздрібними торговими підприємствами при здійсненні централізованої доставки вантажів споживачам матеріального потоку.

1.2 Алгоритми вирішення задачі комівояжера

Існують різні алгоритми для вирішення завдання побудови оптимального маршруту при різних умовах і конфігураціях. Проведемо огляд відомих алгоритмів:

Метод гілок і меж

Метод гілок і меж - загальний алгоритмічний метод необхідний для знаходження оптимальних рішень різних задач оптимізації, в особливості дискретної і комбінаторної. По суті, метод гілок і меж є комбінаторним (алгоритм перебору) з відсівом підмножин допустимих рішень, що не містять оптимальних рішень [2].

В основі методу лежить наступна ідея (для завдання мінімізації): якщо нижня межа для підобласті А дерева пошуку більше, ніж верхня межа якої-небудь раніше переглянутої підобласті В, то А може бути виключена з подальшого розгляду (правило відсіву) [2].

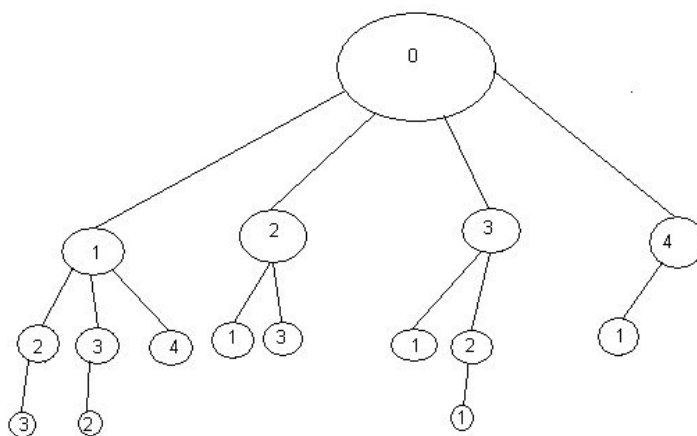


Рисунок 1.1 – Графічне зображення виконання методу гілок і меж

Метод найближчого сусіда

Основа алгоритму найближчого сусіда полягає в тому, що пункти обходу плану послідовно включаються в маршрут. Головною особливістю його служить, що кожен черговий включається пункт повинен бути найближчим до останнього вибраного пункту серед всіх інших, ще не включених до складу маршруту [2].

Одним з евристичних критеріїв оцінки рішення є правило: якщо шлях, пройдений на останніх кроках алгоритму, порівняти з шляхом, який був пройдений на початкових етапах, то можна умовно вважати знайдений маршрут прийнятним або гідною кандидатурою, інакше, ймовірно, існують більш оптимальні рішення [3].

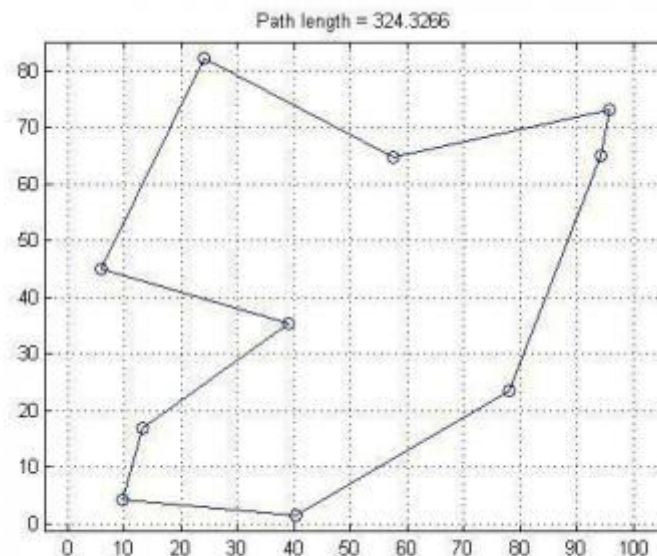
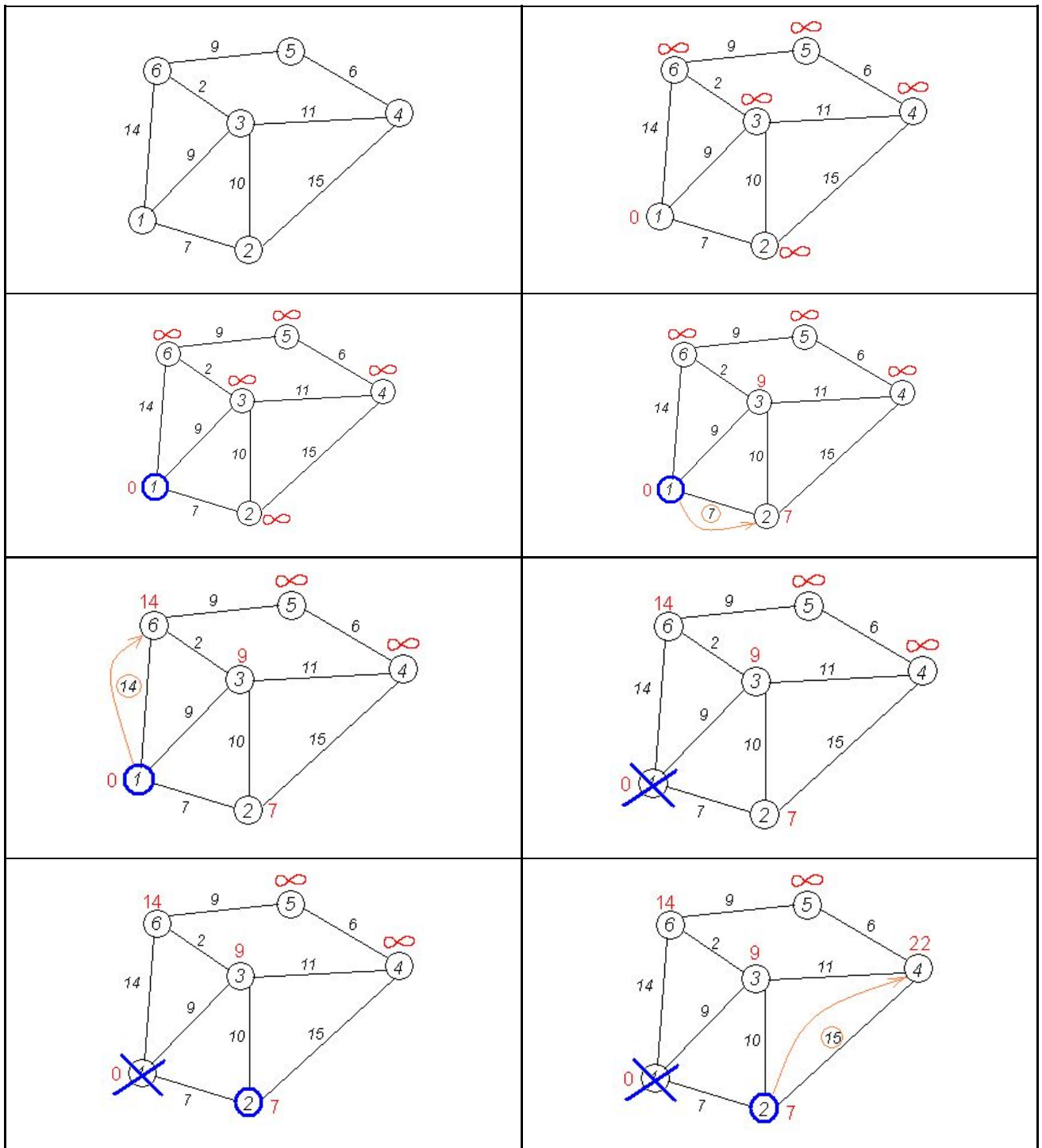


Рисунок 1.2 – Графічне зображення виконання методу найближчого сусіда

Алгоритм Дейкстри

Алгоритм Дейкстри - це алгоритм на графах, винайдений Е. Дейкстрой. Даний алгоритм знаходить найкоротша відстань від однієї з вершин графа до всіх інших вершин. Алгоритм застосовується тільки для графів без ребер негативного ваги. Алгоритм Дейкстри широко поширений в програмуванні і технологіях [2].



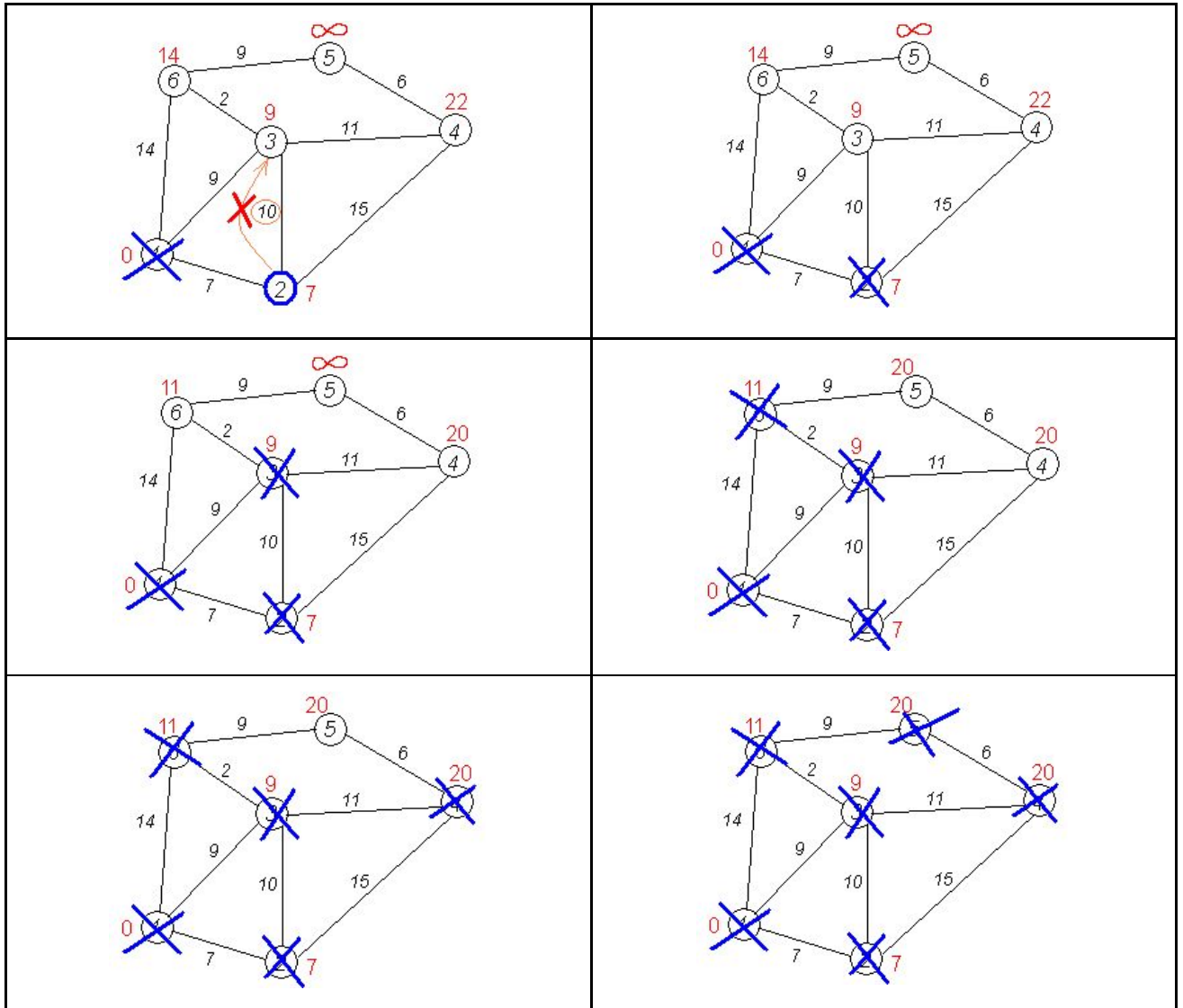


Рисунок 1.3 – Графічне зображення виконання алгоритму Дейкстри

Алгоритм розумних крапель

Алгоритм розумних крапель створює імітацію законів зв'язку окремих потоків, або множин крапель, які взаємодіють як один з одним, так і з навколишнім середовищем, в процесі формування маршруту річки. Для кожного окремого безлічі крапель річка є навколишнім середовищем, яка в значній мірі впливає на його поведінку, але подібний вплив двунправленне.

Обираючи певний маршрут, вода слідує принципом найменшого опору. Наприклад: якщо на певному шляху зустрічається менше кількості ґрунту, ніж на інших, то цей шлях буде більш оптимальним [2].

Генетичний алгоритм

В основі генетичного алгоритму лежить ідея запозичення з природи деяких принципів еволюції. Рішення в генетичному алгоритмі представлено хромосоною індивіда в популяції (безліч рішень). Генетичний алгоритм складається з декількох операторів, що слідують один за одним: селекція, рекомбінація і мутація [3].

Генетичний алгоритм має безліч параметрів, що настроюються: ймовірність мутації, тип селекції (пропорційна, ранговая з лінійним і ранговая з експоненціальним ранжированием) [3].

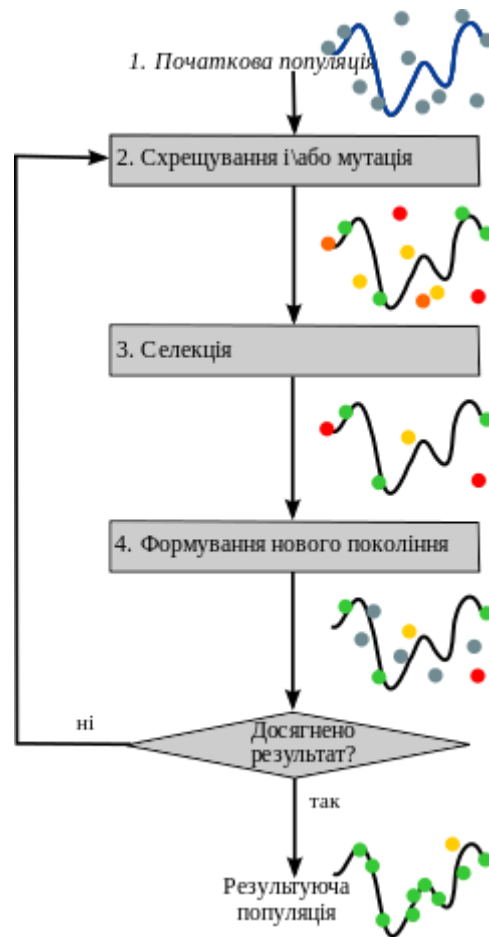


Рисунок 1.4 – Графічне зображення виконання генетичного алгоритму

Мурашині алгоритми

Головна ідея мурашиних алгоритмів (методу мурашиних колоній) полягає в моделюванні поведінки колонії мурах, які за своєю природою здатні знаходити найкоротший шлях від мурашника до джерела їжі і пристосовуватися до мінливих умов зовнішнього середовища, знаходячи новий найкоротший шлях, якщо старий в силу будь-яких обставин виявляється недоступним. При русі мураха залишає феромон, позначаючи свій путь.Благодаря цієї інформації інші мурахи йдуть із зазначеного шляху. Таким чином, взаємодія, яке відбувається між агентами (мурахами) є непрямий комунікацією [4, 5].

Якщо змоделювати переміщення мурашок на деякому графі, ребра якого являють собою шляху можливого переміщення мурашок, то крім позитивного зворотного зв'язку (відкладання феромону) необхідно моделювати негативний

зворотний зв'язок, тобто випаровування феромону. Наявність негативного зворотного зв'язку гарантує, що мурахи будуть шукати інші шляхи від колонії до джерела їжі, тобто знайдене локально оптимальне рішення не буде єдиним. Рішенням завдання послужить найбільш збагачений феромонами шлях на ребрах графа, який і буде оптимальним [4, 5].

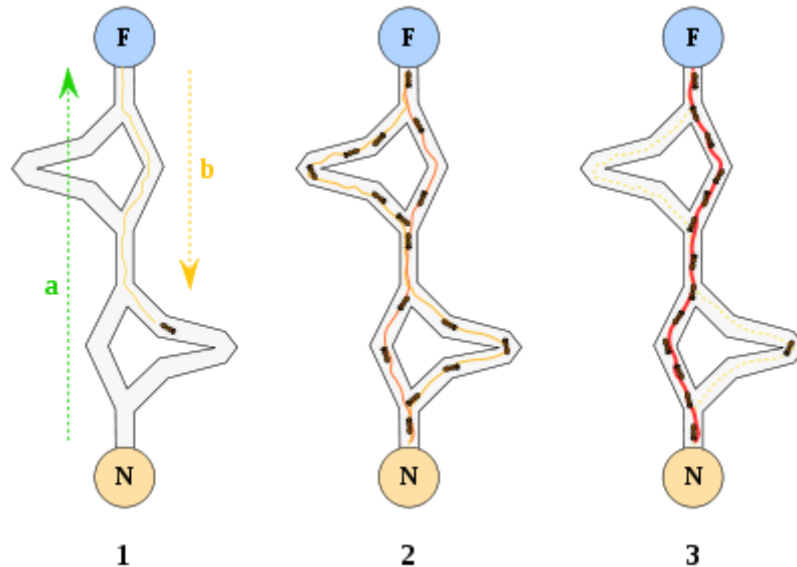


Рисунок 1.5 – Графічне зображення виконання мурашиного алгоритму

1.3 Порівняльний аналіз алгоритмів вирішення логістичних завдань

Метод гілок і меж використовує покращений перебір, в результаті чого відбувається відкидання неоптимальних рішень при кожному наступному кроці алгоритму. Ключовою відмінністю є той факт, що при використанні даного алгоритму для невеликого числа вершин точність виходить хороша, однак при кожному збільшенні числа вершин, похибка буде збільшуватися. Метод найближчого сусіда простий у використанні, швидко виконується, але, як і інші «жадібні» алгоритми, здатний видавати оптимальні рішення. Основним недоліком даного алгоритму є можливість підбору параметрів, в рамках якого алгоритм найближчого сусіда видаватиме найгірше рішення. Ключова

відмінність алгоритму Дейкстри полягає в здійсненні пошуку шляху, що проходить через найбільшу кількість вершин, в результаті з наступним поверненням в початкову. У порівнянні з іншими розглянутими алгоритмами, серйозним недоліком як алгоритму розумних крапель, так і інших біонічних алгоритмів, є велика кількість параметрів, що настроюються, вибір яких застосовується до предмету дослідження, що само по собі є важко здійснюваним завданням.

Генетичний алгоритм, що застосовується для вирішення завдань оптимізації та моделювання випадковим підбором, полягає в пошуку шляхом комбінування різних варіацій параметрів, що нагадують біологічну еволюцію. Особливістю алгоритму є використання 10 операторів схрещування і мутації, які виробляють рекомбінацію рішень-кандидатів, схожою з роллю схрещування в живій природі. Переваги мурашиних алгоритмів сформулюємо наступним переліком: - є порівняно ефективним методом вирішення оптимізаційних завдань; - алгоритм враховує пам'ять про всю колонію (всіх агентів), а не тільки пам'ять про попередньому поколінні; - алгоритм скорочує ймовірність неоптимальні початкових рішень (через випадкового вибору шляху і пам'яті колонії). Недоліки використання мурашиних алгоритмів наведені нижче: - використовується послідовність випадкових (не незалежні) рішень; - розподіл ймовірностей змінюється при ітераціях; - дослідження більше експериментальне, ніж теоретичне; - результати сильно залежать від настроюваних параметрів, які підбираються тільки виходячи з проведених експериментів. Таким чином в основі мурашиних алгоритмів лежать імітації самоорганізації соціальних комах за рахунок використання динамічних механізмів, за допомогою яких система досягає оптимального результату внаслідок низкоуровневого взаємодії агентів. Важлива властивість мурашиних алгоритмів- не конвергентність: навіть після великої кількості ітерацій одночасно досліджується безліч варіантів рішення, внаслідок чого не відбувається тривалих тимчасових затримок при пошуку оптимального рішення.

Все це дозволяє апробувати мурашині алгоритми для вирішення завдання побудови логістичного маршруту. З урахуванням перерахованих особливостей застосування мурашиних алгоритмів необхідно відзначити, що в рамках даної роботи завдання побудови логістичного маршруту буде проводитися на полі обмеженого розміру, а запропоноване рішення не гарантуватиме оптимальність для поля більшої розмірності.

1.4 Інформаційне забезпечення систем пошуку маршруту

Сучасний ринок програмного забезпечення у галузі логістики достатньо великий і включає програмні продукти різної спрямованості та характеристик. Проте системи матеріально-технічного постачання трансформуються відповідно до світових тенденцій на засадах принципів системного і логістичного підходів, раціональності, ідентифікації витрат, транспарентності, довірчих відносин, рівності підходу і забезпечення чесної конкуренції; відбувається зміщення ролі постачання в управлінні підприємством від вирішення тактичних завдань, обслуговування його поточних потреб до вирішення стратегічних [6], що підвищує актуальність проблеми ефективного управління запасами матеріальних ресурсів.

Завдяки продуманному використанню сучасних цифрових технологій у бізнесі підприємства можуть значно підвищити ефективність своєї діяльності. Проте такі технології можуть зумовити як зміни в існуючих бізнес-процесах, так і появу нових. Цифрові технології – це, насамперед, інновації, впровадження та використання яких потребує наявності відповідних фахівців на підприємстві, здатні постійно навчатися та набувати нових компетентностей. Згідно дослідженням [7] майже всі керівники підприємств у сфері логістики (ланцюгів постачання) згодні, що цифрові технології кардинально змінять логістичні процеси в найближчому майбутньому, проте більше половини з них говорить, що не мають наразі

сформованої стратегії розвитку своїх підприємств з урахуванням існуючих та нових цифрових технологій.

Сьогодні розвиток технології дозволяє спростити та автоматизувати ті процеси, які ще декілька років тому не могли б існувати без підтримки людини. Доволі часто ми чуємо про нові сервіси, які вдосконалюють ті процеси, які людина не здатна вирішувати на високому рівні. Наприклад всім знайомий сервіс для виклику авто (таксі) Uklon, це досить хороший приклад сервісу, який вирішує проблеми, будування найкращого маршруту для водіїв таксі, за допомогою сучасних технологій, що в свою чергу дозволяє скоротити витрати на паливо, задовольнити потреби клієнта, а також дає змогу виконати більше замовлень за один і той же час.

Сфера вантажоперевезень не виключення, завдяки вседоступності до мережі Інтернет, з'являється можливість до реалізації нових ідей, які змінюють всі процеси, які були до цього запроваджені в тій чи іншій галузі.

Ефективне вирішення логістичних задач на теперішньому етапі розвитку техніки та технологій можливе з використанням інформаційних систем і з відповідним функціоналом.

Сучасному перевізнику пропонується цілий спектр програмних рішень для забезпечення його діяльності.

Одним з прикладів сервісів, які дозволяють перенести сферу вантажоперевезень в онлайн є сервіс “Transportica” (рисунок 1.6) [8].

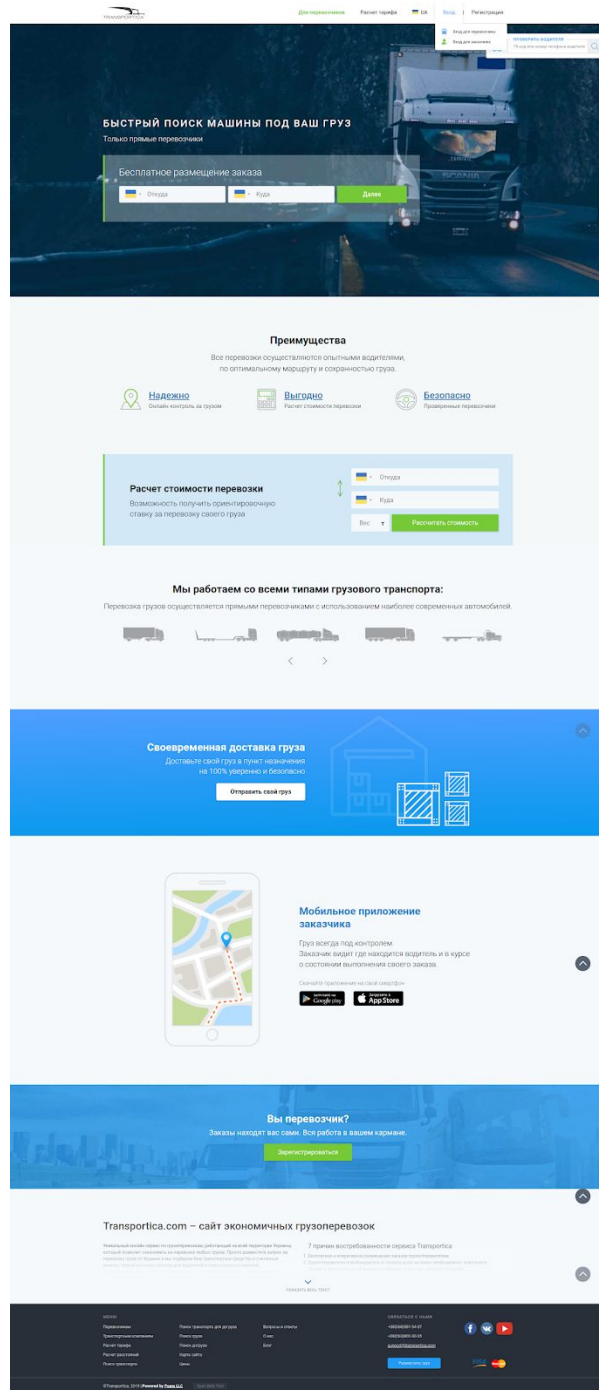


Рисунок 1.6 – Transportica - сайт економічних вантажоперевезень

Головна сторінка сайту організована в стилі landing-page, що дозволяє зосередити основний контент на одній сторінці. Даний веб-сервіс складається з двох частин, а саме з особистого кабінету відправника і особистого кабінету водія (рис 1.7) [8]. Також на сайті є досить зручна і інтерактивна форма для створення замовлень. Зручним є те, що форма вже має запрограмовані варіанти

значень, все що залишається зробити, так це вибрати варіанти, які максимально точно описують потреби замовника (рис 1.8) [8]. Одним з недоліків даного сервісу, є те, що він позиціонує себе, як транспортна компанія, яка бере на себе функцію відбору водіїв за певними критеріями, а також те, що сервіс є платним.

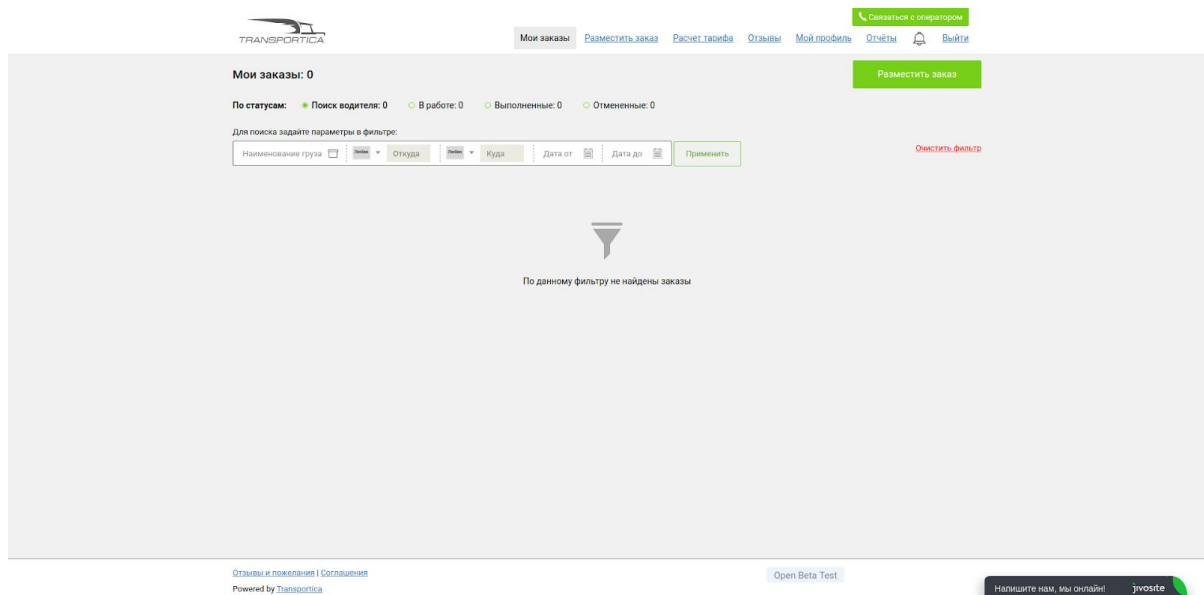


Рисунок 1.7 – Особистий кабінет відправника

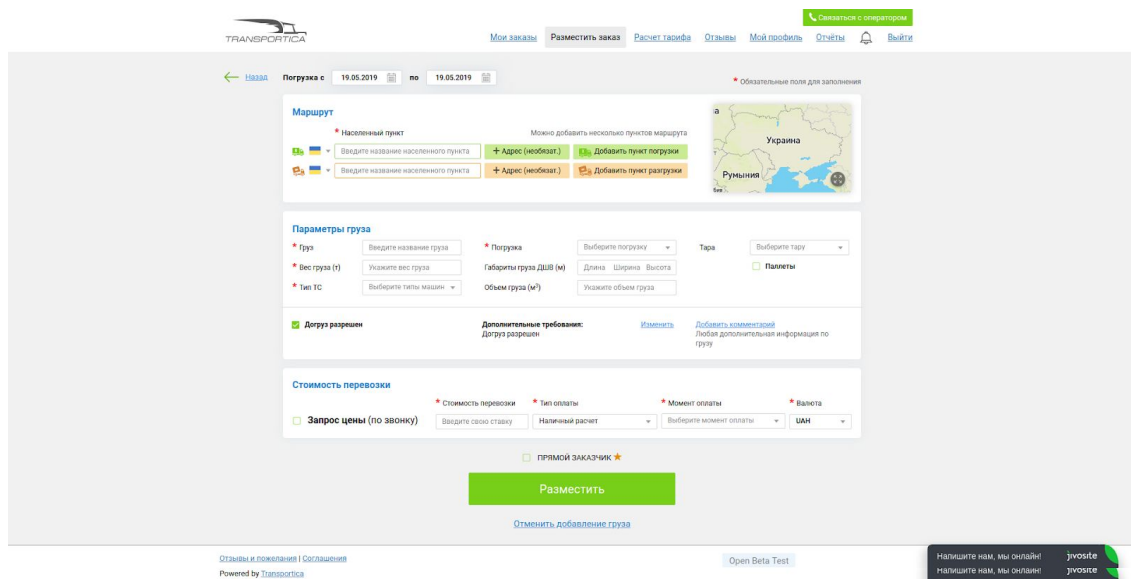


Рисунок 1.8 – Форма створення замовлення

Ще одним прикладом сервісу в сфері вантажоперевезень є “Lardi Trans”, це сервіс, який спеціалізується на міжнародних вантажоперевиженнях (рисунок 1.9) [9].

The screenshot displays the Lardi Trans website interface. At the top, there is a navigation menu with categories like 'Вантажі та транспорт', 'ЛардіМаркет', 'Logist.Today', 'SmartGPS', 'TrucksNearMe', and 'Enki'. The main header features the 'lardi-trans.com' logo and a search bar. Below the header, there are several tabs: 'Вантажі', 'Транспорт', 'Послуги та ціни', 'Зона надійності', 'Каталог', 'Форум', 'Автомаркет', and 'Корисне'. The central part of the page is a search form titled 'Вантажоперевиження онлайн - "Ларді-Транс"'. It includes fields for 'Звідки' (From) and 'Куди' (To), both set to 'Будь-яка обл.' (Any region). There are also fields for 'Авто' (Vehicle), 'Дата' (Date), 'Маса' (Weight), and 'Об'єм' (Volume). A 'Пошук' (Search) button is present. To the right of the search form is a 'Розрахунок відстаней' (Distance calculator) section with input fields for 'З:' and 'У:' and a 'Розрахувати' (Calculate) button. On the left side, there is a 'Чат підтримки' (Support chat) window with a 'Зачекайте оператора в режимі онлайн чату' (Wait for the operator in the online chat mode) message. Below the chat, there is a 'Що робити?' (What to do?) section with a list of links: 'Додати вантаж або транспорт', 'Знайти вантаж або транспорт', 'Додати вантаж на фірму', 'Зайти на фірму', 'Зайти в автомаркет', 'Розрахувати відстань', 'Звернутися до служби підтримки', and 'Зайти в каталог фірм'. Below this is a 'Реклама на сайті' (Advertisement on the site) section. The main content area features a 'VIP-клієнти Ларді-Транс' (VIP clients Lardi-Trans) section with logos for various companies: 'ПАРЖИЦЬКИЙ', 'STARS', 'T.P.G. Логістична компанія', 'СІРІСІВНИЙ', 'Girteko LOGISTICS', 'RAVTRANS', 'Транзит-Траст', 'TRANS LOGISTIC', 'NEOLIT LOGISTICS', and 'ТОВ Бенітола'. Below the logos is a 'Дата Фірма Новини' (Date Company News) section with a list of news items:

- 22.04.2019 ООО "Ларді" Графік роботи контакт-центру Ларді на період празників
- 18.04.2019 ООО "Ларді" 5 прямих польотів TrucksNearMe в нашім ім'я починаються прямо сейчас
- 05.04.2019 ООО "Ларді" Изменение графика работы контакт-центра Беларусь 08.04.2019
- 28.03.2019 ООО "Ларді" Обновления на количество сотрудников в учебных записках
- 15.03.2019 ООО "Ларді" Изменение графика работы контакт-центра Украина 17.03.2019

Below the news list is a 'Транс-блог' (Trans-blog) section with a 'Новини' (News) section containing several news items:

- 23.04.2019 «Сезон с трюмами» 23 апреля 2019 года по состоянию на 09:00 (по Киеву)
- 23.04.2019 В четырех регионах заканчивается весеннее ограничение движения большегрузов
- 23.04.2019 На сельских трассах летние ограничения вступят в силу 20 мая
- 23.04.2019 На калужских трассах в июле ограничат движение большегрузов
- 23.04.2019 На Бельгии с 20 мая ограничат движение большегрузов
- 23.04.2019 Летние ограничения на волгоградских трассах вернут 20 мая
- 23.04.2019 С 1 мая в Камчатском крае ограничат движение большегрузов
- 22.04.2019 «Угроза безопасности» в Ненецкой области за сутки выписали штрафов за перегруз на общую сумму более 26 в тыс. евро
- 22.04.2019 Мининфраструктуры: Урагане требуют минимум 100 систем взвешивания грузовиков в движении
- 22.04.2019 В Польше на Пасху введут временные ограничения на движение большегрузного транспорта

Below the news list is an 'Архів новин' (News archive) section. At the bottom of the page, there is a footer with social media links (Twitter, Facebook), a 'Карта сайту' (Site map) link, 'Контакти' (Contacts), 'Партнери' (Partners), 'Copyright Lardi-Trans © 1999-2019', and a 'Завантажте наш додаток' (Download our app) section with links to 'Мобільна версія' (Mobile version), 'Get it on Google play', and 'Download on the App Store'.

Рисунок 1.9 – Веб-сервіс міжнародних перевезень Lardi Trans

Даний сервіс виступає одразу в декількох категоріях, а саме, він виконує функцію доски для розміщення заявок на перевезення, а також платформи для керування транспортною компанією. Ідея досить хороша, але дизайн та змістовність не чітко продумані, помітно, що даний сервіс не є

клієнтоорієнтованим, оскільки даний дизайн був актуальним в минулому, зараз він не зручний, в якійсь мірі відлякує своєю складністю.

1.5 Висновки до розділу та постановка задачі

Проведений аналіз літературних джерел засвідчує нагальну необхідність у логістичній підтримці виробництва і сфери послуг. Це обумовлено розробкою нових інформаційних технологій, які дозволяють розвивати суспільство в тих чи інших напрямках. Кожен великий бізнес, сфера діяльності якого тісно пов'язана з великими грошима, будь то виробництво, оптова чи широка роздрібна мережа, будує свій бізнес на правильно сформованому вантажоперевезенню. Вчасно доставлений вантаж підтримує в стабільності весь процес бізнесу. Якщо на виробництво не буде вчасно доставлений хоч один компонент все зупиниться і почнуться збитки. У тій же оптовій або роздрібній мережі відсутність певного товару вказує дорогу до конкурентів. Клієнтам не важливо з якої причини є необхідних товарів, їм важливо володіти ними.

Світова глобалізація формує значні і складні виклики щодо забезпечення логістичної підтримки сучасного виробництва. Зв'язку з одночасним розвитком інформаційних технологій, їх високою ефективністю, пристосованістю до вирішення формалізованих задач широкого спектру застосування, найбільш доцільним є вирішення задач організації перевезень на основі інформаційних технологій та систем.

В результаті проведеного аналізу існуючих підходів сформульовані такі завдання дослідження метою якого є реалізація інформаційної системи для пошуку найкращого маршруту для вантажного перевезення за допомогою задачі комівояжера:

1. провести аналіз існуючих методів, технологій та рішень пошуку найкращого маршруту доставки вантажу, що ґрунтується на задачі комівояжера;

2. вдосконалення існуючих методик використання задачі комівояжера для пошуку найкращого маршруту;
3. розробити інформаційну систему побудови найкращого маршруту вантажних перевезень, яка поєднує розв'язки задачі комівояжера з сучасними картографічними сервісами;
4. виконати експериментальну перевірку інформаційної системи планування найкращого шляху для доставки вантажу.

Розділ 2

Розробка технологій пошуку маршруту вантажних перевезень

2.1 Мурашині алгоритми

Мурахи - це соціальні комахи, що живуть у колективі (сім'ї чи колонії). Близько двох відсотків комах соціальні, і мурахи становлять половину з них. Кількість мурах в одній колонії може коливатися від 30 до десятків мільйонів. Мурахи є домінуючими в басейні Амазонки, складаючи понад 30% біомаси місцевих лісів. Поведінка мурашок при транспортуванні їжі, подоланні перешкод, будівництві мурашників та інших операціях майже оптимальна.

Принципи поведінки мурах витримали сто мільйонів років після того, як вони "колонізували" Землю. Колонії мурах дивно живучі: скорочення до 40% комах практично не робить ніякого впливу на функціонування всього суспільства. Масове знищення мурах (наприклад, в результаті хімічної обробки їх місця існування). призводить до об'єднання комах з сусідніх мурах в одну сім'ю, щоб врятувати суспільство.

Соціальна поведінка мурах засноване на самоорганізації, наборі динамічних механізмів, що забезпечують це, система може досягти своєї глобальної мети за рахунок низькорівневої взаємодії між її елементами. Ключовою особливістю цього є взаємодія, яка полягає в тому, що системні елементи використовують тільки локальну інформацію. У цьому випадку будь-який централізований контроль і посилення на глобальну модель, яка відображатиме систему в зовнішньому світі виключена. Самоорганізація - як результат взаємодії між наступними чотирма компонентами:

- багаторазове поновлення;
- випадковість;
- позитивний відгук;
- негативні відгуки.

Існує два способи передачі інформації між мурахами: пряме спілкування (яке включає обмін харчовими продуктами і контакти з нижніми щелепами, візуальні і хімічні контакти) і непряме спілкування, яке називається стигмергією. Стигмергія - це форма спілкування, відокремлена в часі, коли один учасник спілкування модифікує середовище, а інші використовують цю інформацію пізніше, коли вона трапляється в районі модифікованого середовища. Біологічно стигмергія реалізується за допомогою феромонів, спеціальної секреторної хімічної речовини, яка відкладається мурашками, як слід, коли вони рухаються. Чим вища концентрація феромонів на шляху, тим більше число мурашок, що рухаються по ньому. З часом феромони випаровуються, що дозволяє мурахам адаптувати свою поведінку при зміні середовища. Розподіл феромонів - це свого роду динамічно мінлива глобальна пам'ять мурашника. У будь-який момент мураха може відчуті і змінити лише одну локальну клітинку цієї глобальної пам'яті.

На прикладі експериментів з мурахами на асиметричному мосту демонструється, як кооперативна поведінка мурах дозволяє знайти найкоротший шлях до їжі. Асиметричний місток (рис. 2.1) з'єднує мурашине гніздо з джерелом їжі двома гілками різної довжини. Експерименти проводились з лабораторною колонією аргентинських мурах (*Iridomyrmex humilis*), які відкладають феромони на шляхах як від гнізда, так і до нього [10]. Схема експериментів була наступною:

- побудований міст А-В-С-Д;
- ворота в точці А були відкриті;
- кількість мурашок, що вибирають довшу (А-С-Д) і підраховано коротші гілки мосту.

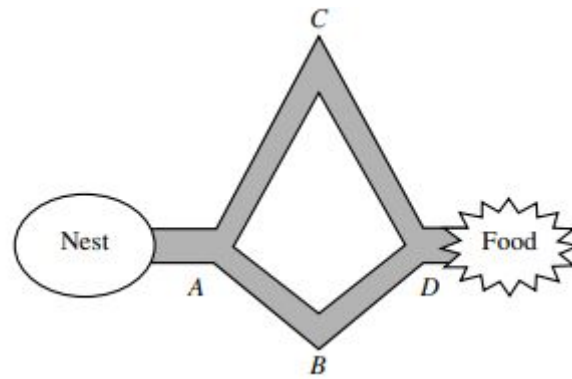


Рисунок 2.1 Асиметричний міст [9].

На ранній стадії експериментів мурахи обирали обидві гілки приблизно з однаковою швидкістю. Через деякий час майже всі мурахи вирішили пересуватися найкоротшим маршрутом A-B-D, що пояснюється наступним чином. По-перше, гілки не мали феромонів; тому гілки A-C-D та A-B-D були обрані з однаковою швидкістю. Мурахи, які обрали коротший шлях A-B-D-B-A, швидше повернулися з їжею до гнізда і проклали феромонні стежки на цій коротшій гілці. Коли їм доводилося вибирати наступного разу, мурахи воліли рухатися уздовж коротшої гілки мосту, оскільки концентрація феромонів на ньому вища. Тому феромони швидше накопичуються на гілці A-B-D, залучаючи мурах, щоб обрати найкоротший шлях. Отже даний алгоритм поведінки мурах, можна застосувати для вирішення задачі комівояжера.

Проблема задачі комівояжера полягає у пошуку найкоротшого закритого маршруту, який проходить один раз через кожне місто. Вибір цієї проблеми для демонстрації ідей мурашиних алгоритмів пояснюється наступним чином:

- проблему можна зручно інтерпретувати з точки зору поведінки мурашок: інтуїтивно переміщення мандрівного продавця схоже на переміщення мурах;
- це NP-важка проблема;
- це традиційна контрольна проблема для комбінаторних методів оптимізації. Існує велика бібліотека тестових проблем продавців-мандрівників та методів їх вирішення, що дає змогу

порівняти ефективність алгоритмів мурашок з іншими підходами оптимізації;

- це дидактична проблема, для якої процес пошуку оптимуму можна пояснити без обговорення технічних деталей алгоритму;
- це перша комбінаторна задача, яка була вирішена мурашиними алгоритмами.

Давайте розглянемо, як чотири компоненти самоорганізації можуть бути реалізовані в оптимізації маршруту комівояжера.

Багаторазове спілкування забезпечується шляхом ітеративного пошуку маршруту продавця, що подорожує, кількома мурахами одночасно. Кожен мураха розглядається як окремий і незалежний мандрівний продавець, який вирішує власну проблему. За одну ітерацію проходить цілий маршрут продавця, що подорожує.

Позитивні відгуки забезпечуються через моделювання поведінки мурашок «прокладання слідів та слідування». Чим більше доріжок прокладено на шляху (на краю графіка в задачі продавця), тим більша кількість мурах вибирає цей шлях. Це призводить до нових слідів на шляху, які на наступних ітераціях алгоритму залучатимуть додаткових мурах. Щодо проблеми продавця, позитивний зворотний зв'язок забезпечується за допомогою наступного стохастичного правила: ймовірність того, що край графіка включений у маршрут мурахи, пропорційний його значенню феромону. Таке ймовірнісне правило реалізує також наступну складову самоорганізації, випадковість. Чим коротший шлях від джерела їжі до мурашника, тим частіше біологічний мураха може пройти його протягом фіксованого інтервалу часу, відкладаючи певну кількість феромонів під час кожного проходження. Для імітації такої поведінки мурах об'єм віртуальних феромонів, покладених на ребро графіка, приймається обернено пропорційним довжині шляху. Чим коротший шлях, тим більше феромонів відкладено на відповідних краях графіка, і тим більше мурахи використовуватимуть феромони для синтезу нових шляхів.

Позитивні відгуки самі по собі ведуть до застою; у цьому випадку всі мурахи обирають один неоптимальний шлях. Щоб уникнути цього, вводиться негативний зворотний зв'язок через випаровування феромонів. Інтенсивність випаровування не повинна бути занадто високою; інакше область пошуку звужиться. Випаровування не повинно бути занадто швидким, щоб уникнути ситуації, коли колір передчасно «забуває» свій досвід, накопичений у минулому (втрата пам'яті), що порушує кооперативну поведінку мурах.

Для кожного мураха перехід від міста i до міста j залежить від наступних трьох компонентів: списку табу, видимості та слідів віртуальних феромонів.

Список табу (пам'ять мурашок) - це структура даних, яка зберігає список вже відвіданих міст, які не слід відвідувати знову. Цей список збільшується в розмірі під час туру і встановлюється нульовим на початку кожної ітерації алгоритму. Позначимо через J_{ik} список міст, які ще мусить відвідати мураха k , розташований у місті i . Зрозуміло, що J_{ik} є доповненням до списку табу.

Видимість - це величина, яка відповідає відстані: $\eta_{ij} = 1/D_{ij}$ де D_{ij} є відстань між містами i та j . Видимість - це місцева статична величина, що відображає евристичне бажання переїхати до міста j з міста i : чим ближче місто, тим сильніше бажання його відвідати.

Слід віртуальних феромонів на краю $(i - j)$ - це бажання, засноване на досвіді колонії, переїхати до міста j з i . На відміну від видимості, розподіл феромонів змінюється після кожної ітерації, що відображає досвід, отриманий мурахами. Кількість віртуальних феромонів на краю $(i - j)$ на ітерації t позначається $\tau_{ij}(t)$.

Імовірність того, що мураха k рухається на ітерації t з міста i в місто j , обчислюється за таким імовірнісно-пропорційним правилом:

$$P_{ij,k}(t) = \begin{cases} \frac{(\tau_{ij}(t))^\alpha (n_{ij})^\beta}{\sum (\tau_{ij}(t))^\alpha (n_{ij})^\beta}, & \text{якщо } j \in J_{ik}, \\ 0, & \text{якщо } j \notin J_{ik} \end{cases} \quad (2.1)$$

де $\alpha \geq 0$ та $\beta \geq 0$ - регульовані параметри, що описують вагу феромонової доріжки та видимість при виборі маршруту. Коли $\alpha = 0$, вибирається найближче місто, що відповідає жадібному алгоритму в класичній теорії оптимізації. Коли $\beta = 0$, враховується лише шлях феромонів, що означає, що всі мурахи обирають один неоптимальний шлях. Щоб забезпечити хорошу динаміку оптимізації, рекомендується встановити $\beta \geq \alpha$.

Зазначимо, що правило (2.1) визначає ймовірність обрання конкретного міста. Сам вибір здійснюється за принципом “колесо рулетки”: кожне місто в ньому має свій сектор із площею, пропорційною ймовірності (формула 2.2). Місто вибирається як метання м'яча рулетки, тобто шляхом генерації випадкового числа для визначення сектору, де він зупиняється.

Коли тур закінчений, k -та мурашка відкладає по краю (i, j) значення феромону (2.2)

$$\Delta\tau_{ij,k}(t) = \begin{cases} \frac{Q}{L_k(t)}, & \text{якщо } (i, j) \in T_k(t), \\ 0, & \text{якщо } (i, j) \notin T_k(t) \end{cases} \quad (2.2)$$

де $T_k(t)$ - маршрут мурашки k на ітерації t , $L_k(t)$ - довжина маршруту $T_k(t)$, а $Q > 0$ - регульований параметр.

Для вивчення всього простору розчинів феромони повинні випаровуватися.

Якщо коефіцієнт випаровування позначити через $p \in [0, 1]$, правило оновлення феромонів набуває вигляду

$$\tau_{ij}(t + 1) = (1 - p)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij,k}(t) \quad (2.3)$$

де m - кількість мурах у колонії. На ранній стадії процесу оптимізації значення феромону по краях приймається рівним малому додатному числу τ_0 .

Загальна кількість мурах у колонії залишається незмінною. Дуже велика колонія призводить до швидкого зростання неоптимальних шляхів, тоді як невелика кількість мурашок може призвести до руйнування їх поведінки в кооперативі через зменшення зв'язку та швидке випаровування феромонів. Зазвичай кількість мурах приймається рівною кількості міст. У цьому випадку кожна мураха починає рухатися зі свого міста.

На відміну від біологічних мурах, віртуальні агенти пам'ятають список відвіданих міст і живуть у дискретний час. Крім того, вони не є повністю "сліпими" і обирають свої шляхи не лише за концентрацією феромонів, але й за допомогою евристики. Ці відмінності регулюються тим, що віртуальних мурах використовують для вирішення проблем оптимізації, а не для моделювання колоній комах. На рисунку 2.1.3 представлена мурашина система оптимізації маршруту продавця, яка реалізує вищезазначені принципи самоорганізації.

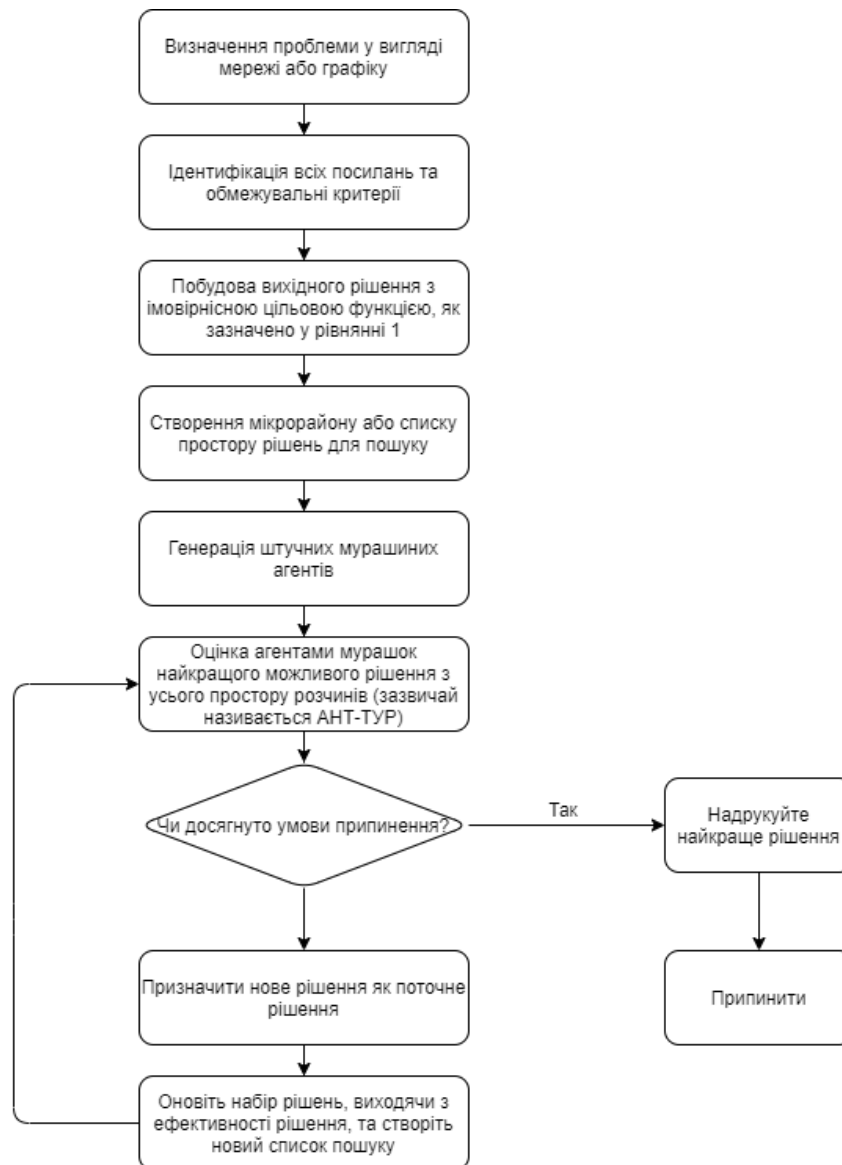


Рисунок 2.4 Графічне зображення використання мурашиного алгоритму

2.2 Математична модель пошуку оптимального маршруту

У класичній постановці завдання комівояжера заключається у пошуку самого вигідного маршруту на зваженому графі G , що проходить через всі його вершини хоч по одному разу з подальшим поверненням у вихідну вершину. У загальному випадку для точного рішення завдання комівояжера необхідні перебір всі можливі варіантів. Це робить таку задачу практично нерозв'язною для сучасних комп'ютерів вже при невеликому кількості вершин графа. Однак

залишається надія на побудову алгоритму, який буде послідовно знаходити маршрути, оптимальні з більшою і більшою ймовірністю. Тут нам і допомагає мураха, який випадково переміщується по графіку, слідуючи деяким специфічними імовірнісним закономірностям.

Для простоти будемо вважати, що граф Γ повний. Мурашиний алгоритм вимагає, щоб ребра графа Γ були додатково позначені дійсними числами, так званими «інтенсивністю феромонів». Перед початком роботи алгоритму інтенсивність феромонів на кожному ребрі вважають рівним нулю. Мурашиний алгоритм моделює роботу системи з багатьма агентами, яких ми будемо називати мурахами. Кожна мураха має невелику кількість пам'яті, а на кожному кроці роботи виконують прості обчислення.

Кожен крок алгоритму - це кінцева послідовність окремих кроків (переходів між вершинами) мурахи, що становлять певний повний шлях на графі. Додатково, після закінчення роботи мурашки на кожному кроці алгоритму обнуляється його пам'ять, а інтенсивності феромонів на ребрах графа перераховуються. «Пам'ять» мурашки - це «список заборон» (tabu list), тобто список вершин, в яких він уже побував. У міру проходження свого шляху, мураха поповнює пам'ять відвідує вершинами. Вибираючи вершину для наступного кроку, мураха не використовує «запам'ятовані» вершини. Крім списку заборон, при виборі чергової вершини мураха керується «привабливістю» ребер, які він може пройти. Вона залежить, по-перше, від відстані між вершинами (тобто від ваги ребра), а по-друге, від інтенсивності феромонів, залишених на ребрі мурахами, які пройшли по ньому раніше.

Вибір наступної вершини визначається з певною ймовірністю. У найпростішому випадку це робиться в такий спосіб. Безліч вершин графа Γ , доступних для переходу з вершини i (що не потрапили раніше в список заборон) ми позначимо символом S_i . Нехай мураха знаходиться в вершині i , а j - це одна з вершин, доступних для переходу з вершини i . Позначимо вага ребра, що з'єднує вершини i і j символом W_{ij} , а інтенсивність феромонів на ньому - символом t_{ij} .

Тоді мураха переходить з вершини i в вершину j з ймовірністю (формула 2.4)

$$p_{ij} = \frac{t_{ij}^{\alpha} + \frac{1}{w_{ij}^{\beta}}}{\sum_{l \in S_i} \left(t_{il}^{\alpha} + \frac{1}{w_{il}^{\beta}} \right)} \quad (2.4)$$

де α і β - це регульовані невід'ємні параметри, що визначають важливість складових (ваги ребра і інтенсивності феромонів) при виборі шляху. Після того, як мурашка успішно проходить маршрут, він залишає на всіх пройдених ребрах слід феромонів, інтенсивність якого обернено пропорційна довжині пройденого шляху:

$$\Delta_{ij} = \begin{cases} \frac{k}{L}, & \text{якщо } (i, j) \in \pi \\ 0, & \text{тоді} \end{cases} \quad (2.5)$$

де π - пройдений мурахою шлях, L - довжина шляху π , а k - ще один регульований параметр.

Крім цього, запах феромонів вивітрюється, тобто інтенсивність феромонів на всіх ребрах зменшується на кожній ітерації алгоритму. Таким чином, в кінці кожної ітерації необхідно оновити значення інтенсивностей:

$$t'_{ij} = e t_{ij} + \Delta_{ij} \quad (2.6)$$

де e - це ще один регульований параметр, змістовно означає швидкість вивітрювання феромонів.

Для успішної роботи алгоритму (тобто для того, щоб його робота приводила до відшукування найкоротшого шляху з високим ступенем ймовірності)

надзвичайно важливим є правильний підбір регульованих параметрів. Неправильний підбір, швидше за все, призведе до невдачі.

Дійсно, повернемося до формули для визначення p_{ij} . Знаменник дробу в правій частині рівності є нормує множителем. Чисельник являє собою функцію, яка зростає зі збільшенням інтенсивності феромонів на шляху від вершини i до вершини j і зменшується зі збільшенням відстані між цими вершинами. В принципі, можна використовувати будь-яку функцію $f(t_{ij}, w_{ij})$ з цією властивістю, відповідно змінивши нормуючий множник. Неформально кажучи, роль впливу інтенсивності феромонів полягає в тому, щоб «відволікти» мурашки від «локальної оптимізації», що складається в знаходженні на кожному кроці найближчих вершин. При цьому увагу мурашки не повинно переключитися на феромони, інакше результатом роботи алгоритму виявиться досить довільний шлях, «складений» з декількох початкових спроб. Функція $f(t_{ij}, w_{ij})$ (а в нашому випадку параметри α і β) повинна вирішувати завдання «балансу» між факторами, що визначають поведінку мурашки.

Вибір правильного співвідношення параметрів є предметом досліджень. Вичерпних готових рекомендацій в даний час не існує. У загальному випадку цей вибір проводиться на підставі статистичного аналізу.

2.3 Метод розрахунку вартості вантажного перевезення

Вантажоперевезення - одна з найважливіших ланок економіки держави. На відміну від інших галузей, транспортування вантажу не створює продукту, але є невід'ємною складовою процесу виробництва у різних сферах послуг, так як без нього неможливе доведення продукції до кінцевого споживача. Транспорт забезпечує зв'язок між різними галузями та підприємствами державної економіки, регіонами держави, виробниками та споживачами.

При розрахунку транспортних тарифів за основу береться собівартість вантажного перевезення, до складу якого входить ціна засобів виробництва, які беруть безпосередню участь у перевезенні, зарплата працівників та інші витрати. Собівартість вантажного перевезення залежить від величини вантажного потоку, його складу, відстані перевезення, виду транспорту та інші супутні процеси. Процес вантажного перевезення складається з трьох етапів.

Ціноутворення на перевезення вантажу є складним процесом, що пов'язано з різновидом продукції, яка транспортується. Для транспортування вантажів і розрахунку транспортних послуг підприємства, встановлюють вантажні тарифи.

Для розрахунку тарифу перевезення використовуються наступні вхідні дані: місце звідки відправляється вантаж, місце куди потрібно доставити вантаж, вага вантажу. Процес транспортування вантажів складається з трьох етапів:

- початкового;
- рухомого;
- кінцевого.

З урахуванням різновидності початкового та кінцевого етапів відокремлюють витрати на здійснення початково-кінцевих та рухомих операцій. До початково-кінцевих відносяться витрати, які пов'язані з утримання рухомого складу на стоянці, на його підготовку та завантаження, на різні роду маневрові роботи, тобто на всі роботи, які не є пов'язані з рухом або ж відстанню перевезень. До рухомих операції відносяться витрати на утримання рухомого складу на шляху транспортування вантажу.

Отже собівартість транспортного тарифу вираховується за формулою:

$$C_{\Pi} = V_{\text{п.к}} + V_{\text{р}}V \quad (2.7)$$

де $V_{\text{п.к}}$ - витрати на початково-кінцеві операції, гр. од.; V - рухомі (шляхові) витрати на 1 км відстані, гр. од.; V - відстань перевезення вантажів, км.

Користуючись цією формулою, можна розрахувати собівартість вантажного перевезення на 1 км вантажу масою 1 т. При збільшенні відстані транспортування вантажів собівартість перевезення 1 т/км буде зменшуватися, тому що зменшуються відносні витрати на початково-кінцеві операції з перевезенням вантажів на 1 км.

Поділення операцій на початково-кінцеві та рухомі має важливе економічне значення. Це дає можливість впроваджувати двоставковий тариф, тобто ставки на нерухомі та рухомі операції, точніше враховувати витрати транспортних компаній, здійснювати техніко-економічні підрахунки і використовувати їх у загальних підрахунках як транспортних, так і інших підприємств, а отже, ефективніше розрахувати ціну виробу.

Будь-яке підприємство, у тому числі й транспортна компанія, для розширення сфери своєї діяльності, оновлення обладнання, покращення технологічних процесів, стимулювання праці, повинно отримувати прибуток. Тому до транспортних тарифів також враховується і прибуток. Рівень прибутку залежить від виду транспорту, типу транспортування вантажів, виду вантажів тощо. Так як транспорт є фондомісткою галуззю, прибутковість додається до собівартості в розмірах, що дають змогу збільшувати обсяги діяльності й створювати заохочувальні фонди.

Найбільша частина перевезень завдяки мобільності припадає на автомобільний транспорт. Для нього характерні висока маневреність і швидкість транспортування вантажів.

Якщо порівняти перевезення вантажів автомобільним транспортом із залізничним та іншими видами транспорту, то собівартість перевезення вантажів автомобільним транспортом найвища. Це пояснюється тим, що він має незначну вантажопідйомність і великі експлуатаційні витрати, особливо пов'язані із заробітною платою працівникам.

Принципи визначення тарифів перевезення вантажів автомобільним транспортом такі самі, що й для інших видів. Тарифи встановлюються на рівні,

що покриває відшкодування собівартості вантажного перевезення і отримання прибутку, який можна буде використати для розвитку транспортного підприємства і стимулювання праці. На собівартість транспортування вантажів автомобільним транспортом впливають ось такі чинники:

- кліматичні умови;
- вид вантажу;
- характер вантажопотоку;
- тип рухомого складу;
- стан шляхів;
- ціни на пальне та мастила;
- норми витрат пального різними видами автомобілів;
- витрати на ремонт автомобілів та запасних частин;
- заробітна плата водіїв і обслуговуючого персоналу;
- розвантажувальні та експлуатаційні витрати.

Діючі тарифи автомобільного транспорту умовно діляться на два типи відрядні та погодинні.

Відрядні тарифи встановлюються на послуги транспортування залежно від ваги вантажу на відповідну відстань. Це найпопулярніший вид тарифів, який використовується тоді, коли можна точно визначити масу вантажу і відстань на яку потрібно перевезти даний вантаж.

Вантажі, які підлягають перевезенню, поділяють на чотири класи в залежності від завантаженості транспортного засобу. До 1-го класу належать вантажі, якими можна завантажити автомобіль на 100 % (зернові, баштанні, сіль, цукор, картопля тощо), до 2-го - на 71-99 % (м'ясо в тушах, молоко та молочні вироби тощо), до 3-го - на 51-70 % (іграшки, тютюн, одяг, кондитерські вироби тощо), до 4-го класу - на 50 %. За основу беруться вантажі 1-го класу, а для інших застосовуються надбавки. Окрім класу вантажів також враховуються рівень завантаженості транспортного засобу і відстань перевезення.

Тарифи автомобільного вантажного транспорту розроблені так, що зі збільшенням відстані транспортування(при цьому збільшується вантажопідйомність автомобіля і коефіцієнт використання пробігу) тарифна ставка в розрахунку на 1 км буде зменшуватися. Ці тарифи спрямовані на стимулювання транспортування вантажів на значні відстані.

Тарифи на транспортування вантажів масою до 5 т і на відстань до 50 км розраховуються залежно від завантаженості автомобіля, класу вантажу, розміру відправлення, відстані перевезення. Ці тарифи також застосовуються при перевезенні дрібних партій вантажів.

Надбавки до тарифів розраховуються залежно від типу перевезення (міжнародні перевезення, потяговий склад тощо). Тариф на міжнародні перевезення включає в себе вартість дозволу на пересування територією інших країн, мита, зеленої картки.

В деяких випадках застосовуються виняткові тарифи, тобто знижені. Здебільшого, вони встановлюються для масових перевезень (наприклад, товари будівельні, промислові, а також ті, які легко відокремлюються від кузова: пісок, щебінь, каміння, гравій, руда). Підставою для застосування таких тарифів є зниження витрат часу для розвантажування.

Погодинні тарифи застосовуються тоді, коли неможливо чітко визначити відстань транспортування вантажу або вантаж потрібно перевезти на відстань менше 5 км; вантаж потрібно доставити щонайменше у п'ять пунктів і його вага невелика; вантажопідйомність автомобіля не перевищує 1 т. Найчастіше такі тарифи використовуються на підприємствах побутового обслуговування, роздрібної торгівлі, масового харчування, де вантажі перевозяться до великої кількості пунктів реалізації товарів невеликими партіями.

Погодинний тариф встановлюється за кожну годину роботи вантажного автомобіля і залежно від його вантажопідйомності. Однак такі тарифи мають недолік: замовник не мотивований збільшувати відстань транспортування вантажів, автомобіль використовується не завжди ефективно, багато простоїв;

водії транспортного підприємства також не є зацікавлені в роботі автомобіля, що також призводить до простоїв.

Відрядні і погодинні тарифи є основними для розрахунку витрат на транспортування вантажів. Зважаючи на різні умови вантажного перевезення встановлюються різного роду надбавки або знижки. Надбавки до тарифів додаються під час транспортування вантажів у спеціалізованому рухомому складі (рефрижератори, автофургони, лісовози, самоскиди тощо). У цьому разі збільшуються витрати на вантажне перевезення. Знижки застосовуються тоді, коли завантажуються вантажівка, що рухається у зворотному напрямку порожнім, або коли зменшення тарифів стимулюється з боку держави чи регулюється державними органами.

У розрахунку загальних витрат на транспортування вантажів необхідно також враховувати додаткову оплату експедиційного обслуговування і немеханізованих вантажних робіт. Як правило, ці роботи виконують самі водії, які водночас і є експедиторами.

Отже даний функціонал, який дозволить розраховувати вартість вантажу буде корисним не лише для користувачів, які відправляють вантаж, а й для самих водіїв початківців, які в свою чергу можуть цей калькулятор використовувати для формування вартості 1 км.

2.4 Висновки до розділу

Реалізовано інформаційну модель системи знаходження найкращого маршруту для доставки вантажу, яка ґрунтується на мурашиному алгоритмі, що дозволяє віднайти оптимальний маршрут з інтегрованим розрахунком вартості перевезення.

Розділ 3

Програмна реалізація

3.1 Опис інформаційної системи

Розроблений веб-сервіс буде корисним у використанні для клієнтів які, часто використовують послуги транспортних компаній, а також для водій, які мають власний вантажний автомобіль, і хочуть бути незалежні від транспортної компанії.

У сервісі є два типи користувачів: «Клієнт» та «Водій». Для кожного з користувачів будуть доступні різні функції.

Користувач «Клієнт» після авторизації, буде мати змогу створити вантаж, який буде містити в собі всю необхідну інформацію, для того, щоб знайти водія з вантажівкою, котра буде відповідати вимогам транспортування вантажу. «Клієнту» буде доступний перегляд усіх пропозицій від водіїв, які він може відхилити або прийняти. Схема доступних функцій для користувача «Клієнт» зображено на рисунку 3.1.

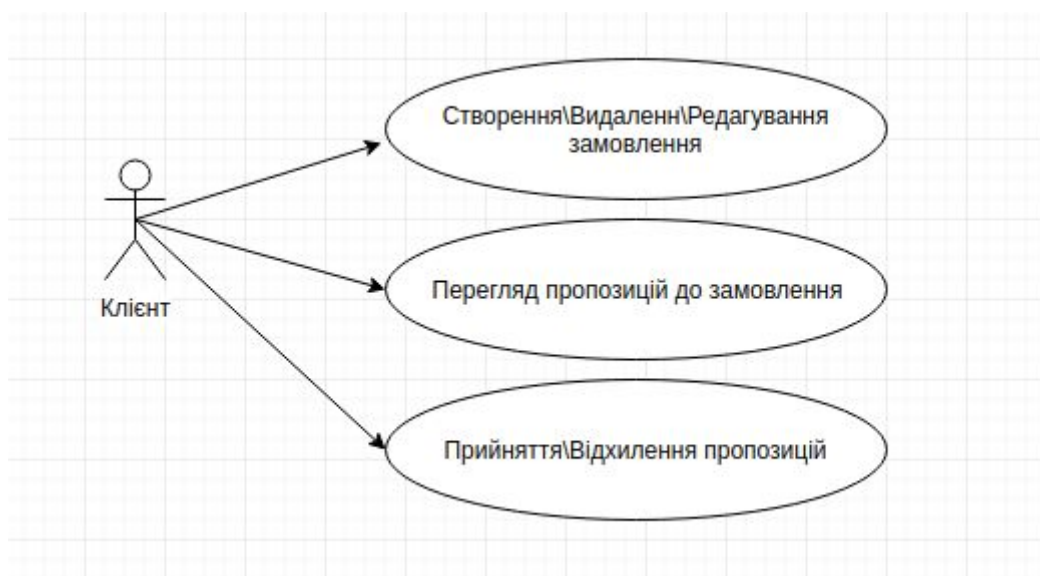


Рисунок 3.1 – Роль користувача «Клієнт»

Користувач «Водій» може отримати доступ до особистого кабінету після авторизації. Даному користувачу буде доступна можливість керування своїми вантажівками, пошук замовлень, і відправлення запиту на замовлення рисунку 3.2.

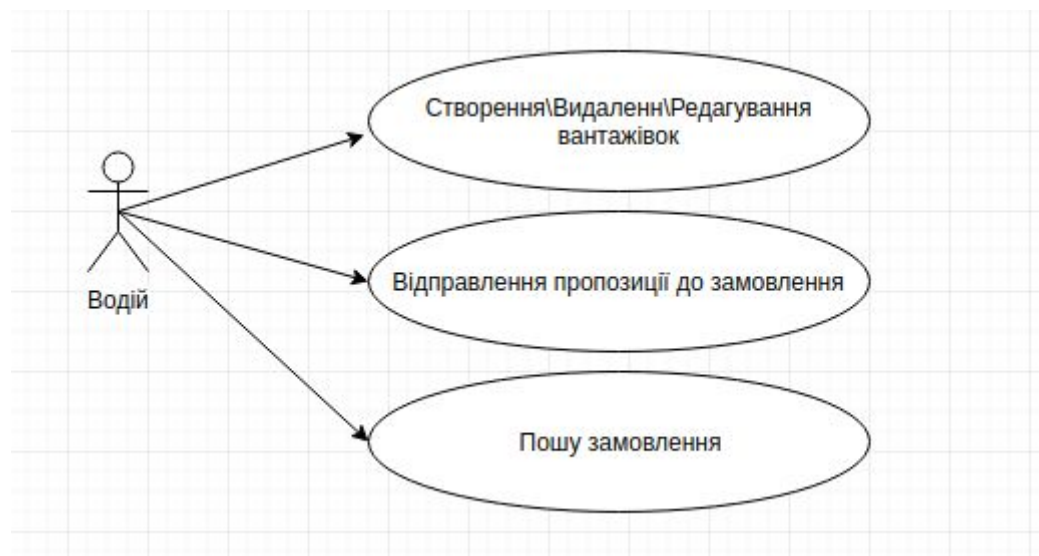


Рисунок 3.2 – Роль користувача «Водій»

Загальну схему інформаційної структури (для користувача) веб-сервісу представлено на рисунку 3.3.

Коли користувач, потенційний клієнт, вперше відвідує його, для нього є цікавим у першу чергу ознайомитися з інформацією на головній сторінці.

Після Авторизацій/Реєстраці клієнт отримує доступ до особистого кабінету, в якому він може змінювати інформацію про себе, створити/редагувати/видалити замовлення, переглянути пропозиції від водіїв, прийняти одну з пропозиції або ж відхилити, після виконання замовлення водієм, клієнт може поставити оцінку водію, яка в подальшому буде впливати на рейтинг водія.

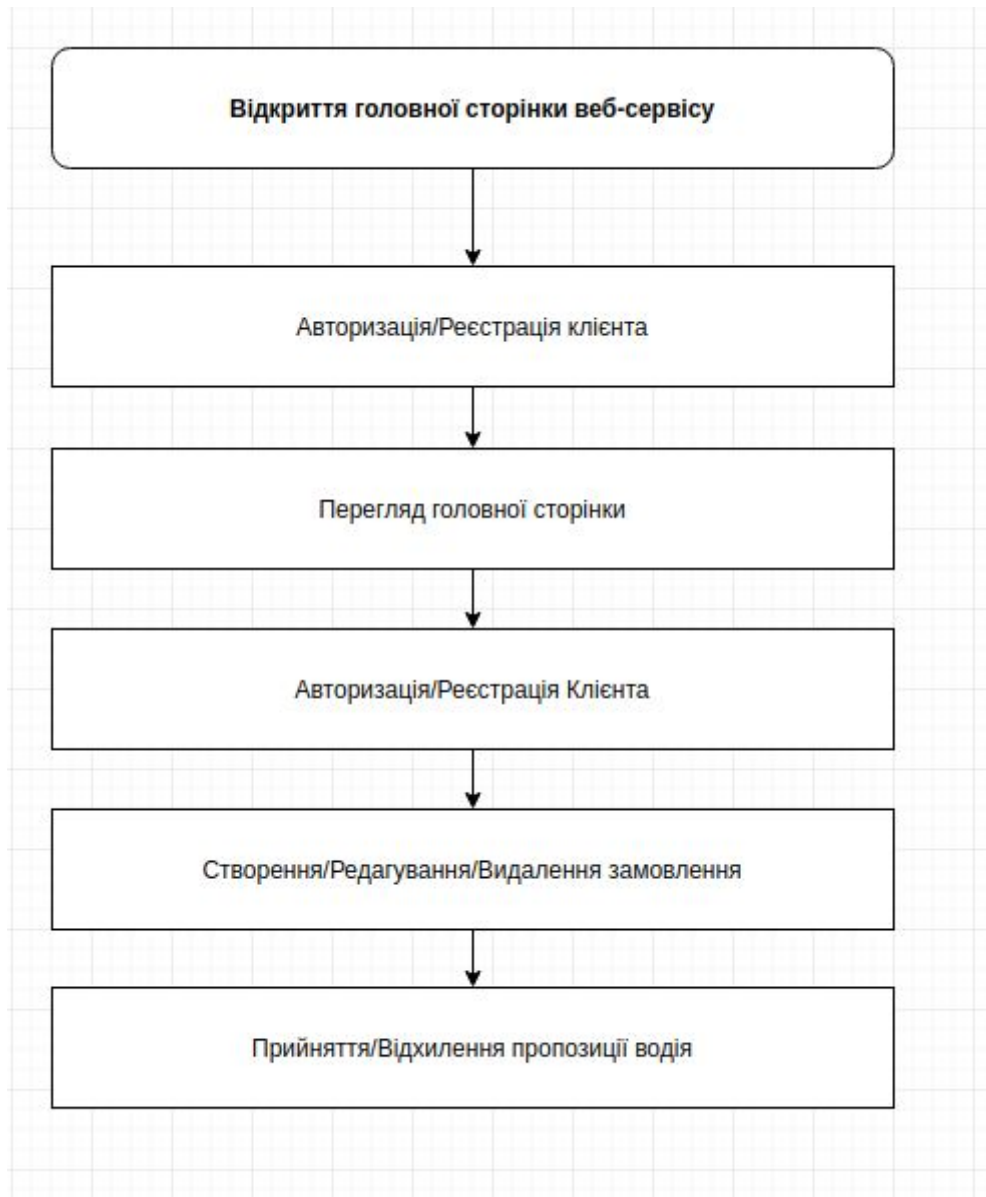


Рисунок 3.3 – Схема користування веб-сервісом (користувач «Клієнт»)

Якщо ж веб-сервіс розглядати зі сторони водія, то в особистому кабінеті будуть зовсім інший функціонал (рисунок 3.4). Після авторизації/реєстрації водій отримує доступ до особистого кабінета, в якому він може створити/редагувати/видалити інформацію про свою вантажівку. Здійснити пошук замовлення за певними критеріями. Відправити запит на замовлення.

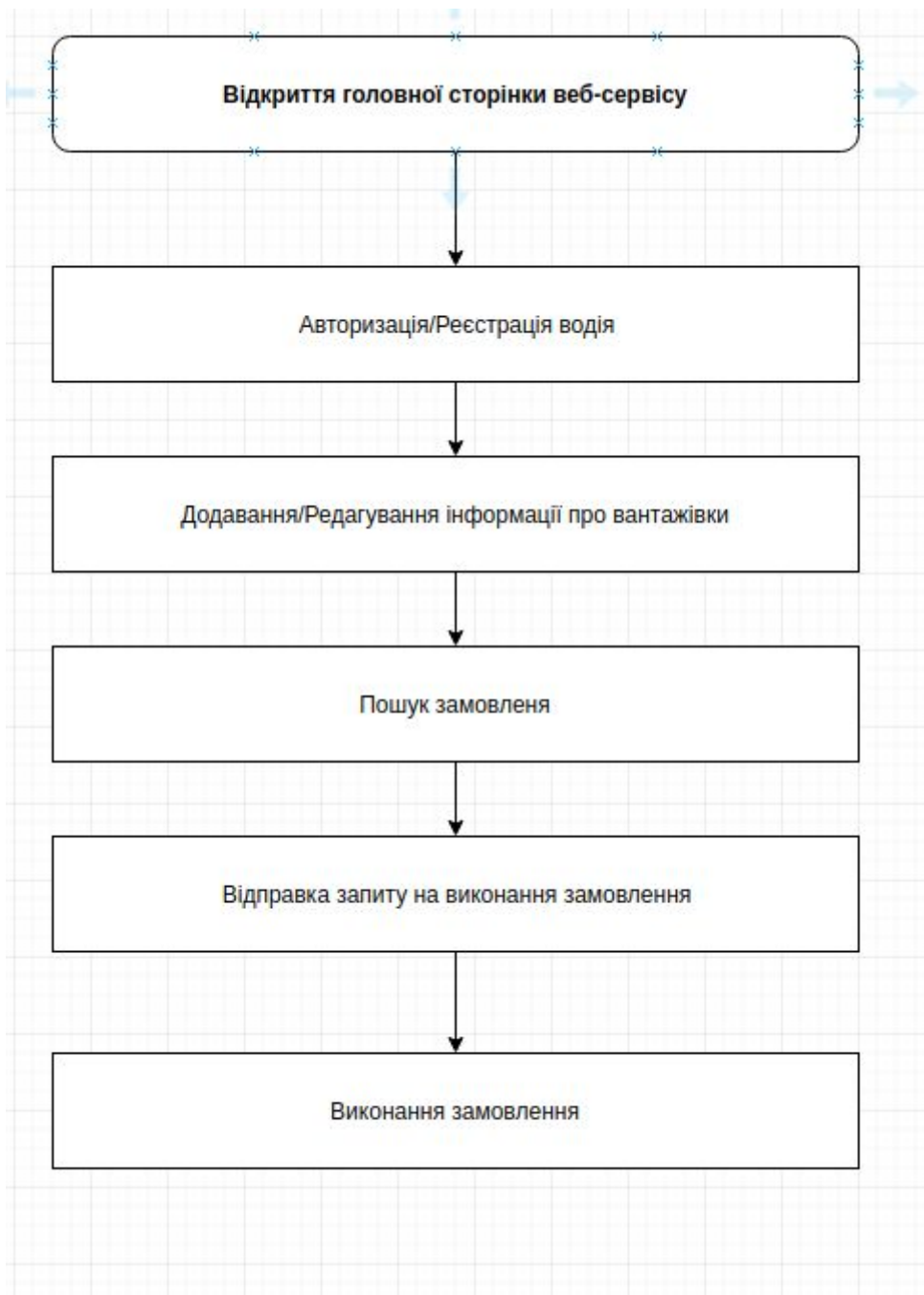


Рисунок 3.4 – Схема користування веб-сервісом (користувач «Водій»)

3.2 Структура і функціональне призначення модулів системи, їх взаємозв'язок

Розробка нових інформаційних технологій дозволяє розвивати суспільство в тих чи інших напрямках. Вони активно перетворюють інші технології матеріального і нематеріального виробництва, в кінцевому підсумку формуючи новий стиль роботи, спосіб життя в цілому. Суть інформаційних технологій становлять методи і засоби формування та підтримки інформаційних потоків у системах управління об'єктами.

Якщо розглядати роботу звичайної транспортної компанії, то можна побачити, як неефективно розподіляються ресурси, дана галузь створює досталь непотрібних, суміжних процесів, які сьогодні можна автоматизувати:

- обробка замовлень;
- планування маршруту;
- виконання замовлень.

Бізнес-процес «Обробка замовлень»

Даний бізнес-процес передбачає опрацювання інформації, що стосується замовлень, й включає в себе наступні функції:

- збір інформації по вантажу;
- розрахунок актуальної вартості доставки;
- оформлення замовлення;
- прийняття пропозиції від водіїв.

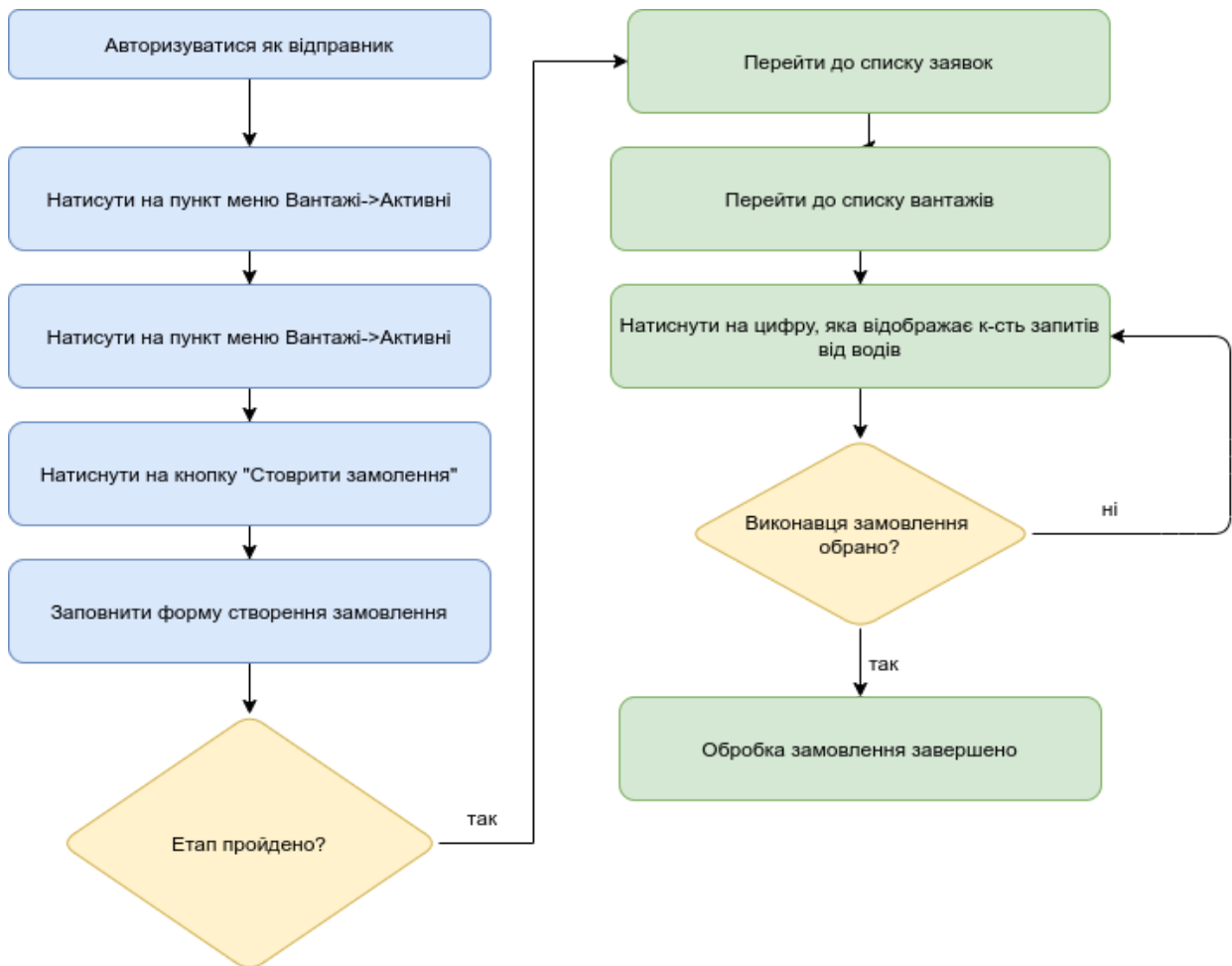


Рисунок 3.5 Бізнес процес “Обробка замовлення”

Бізнес-процес «Планування маршруту»

Даний процес потребує досталь зусиль, для того, щоб спланувати максимально ефективний маршрут, який дозволить збільшити прибутки компанії, а в частості задовольнити відправника вчасним доставленим вантажем.

Планування маршруту передбачає опрацювання інформації, що стосується, починаючи від стану доріг, закінчуючи відстанню між пунктом відправки й пунктом призначення, даний процес включає в себе наступні дії:

- розрахунок відстані між точкою А та точкою Б;
- дослідження можливих маршрутів;

- врахування різних факторів, які впливають на тривалість доставки вантажу;
- вибір найкращого маршруту;

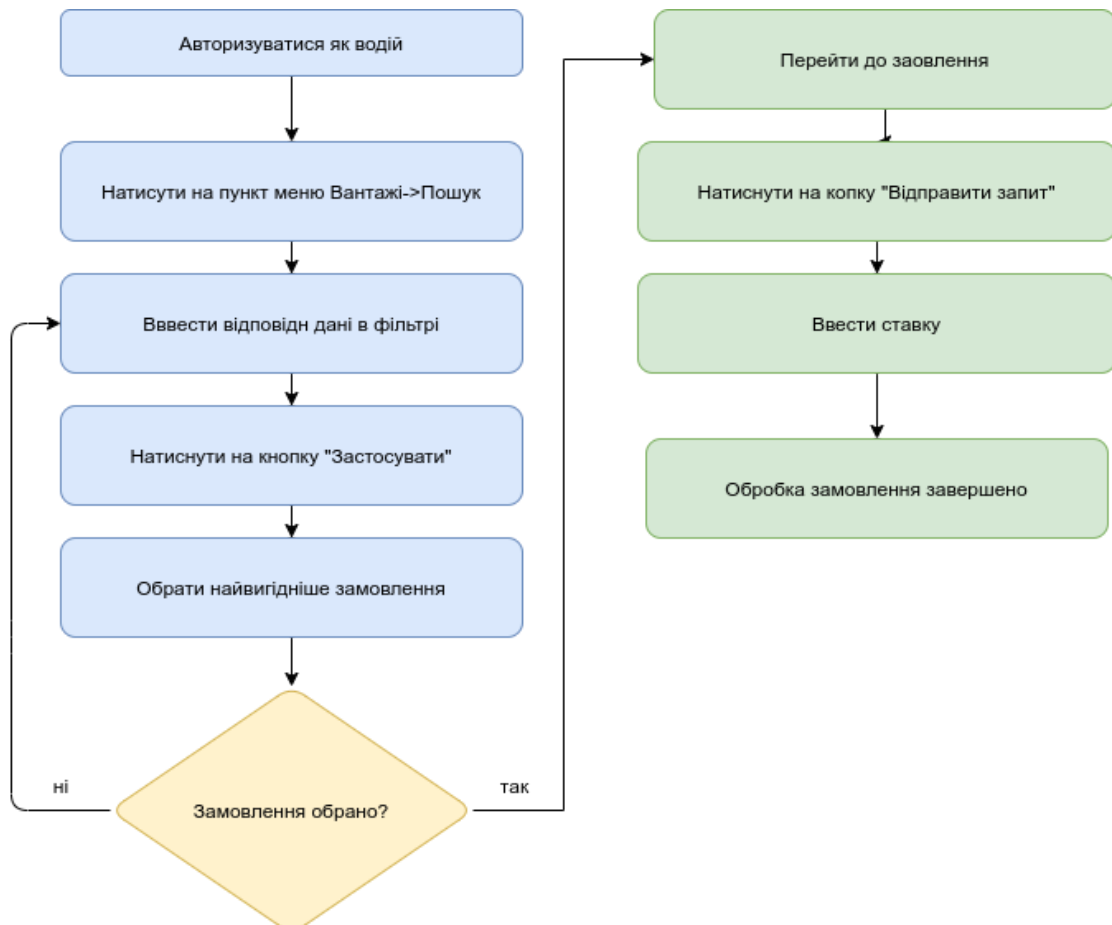


Рисунок 3.6 Бізнес процес “Планування маршруту”

Бізнес-процес «Виконання замовлень»

При роботі з замовленням водій, водій отримує можливість напряду спілкуватися з клієнтом, через влаштований месенджер, що дозволяє отримати максимально точну інформацію щодо вантажу, в свою чергу це дозволяє водію спланувати свій маршрут максимально ефективно, а саме взяти попутні замовлення, якщо це дозволяє параметри його автомобіля. Виконання замовлення включає в себе наступні функції:

- перегляд даних замовлення;

- будування найкращого маршруту;
- виконання попутних замовлень;
- відмова від замовлення;
- керування рейтингом водія.

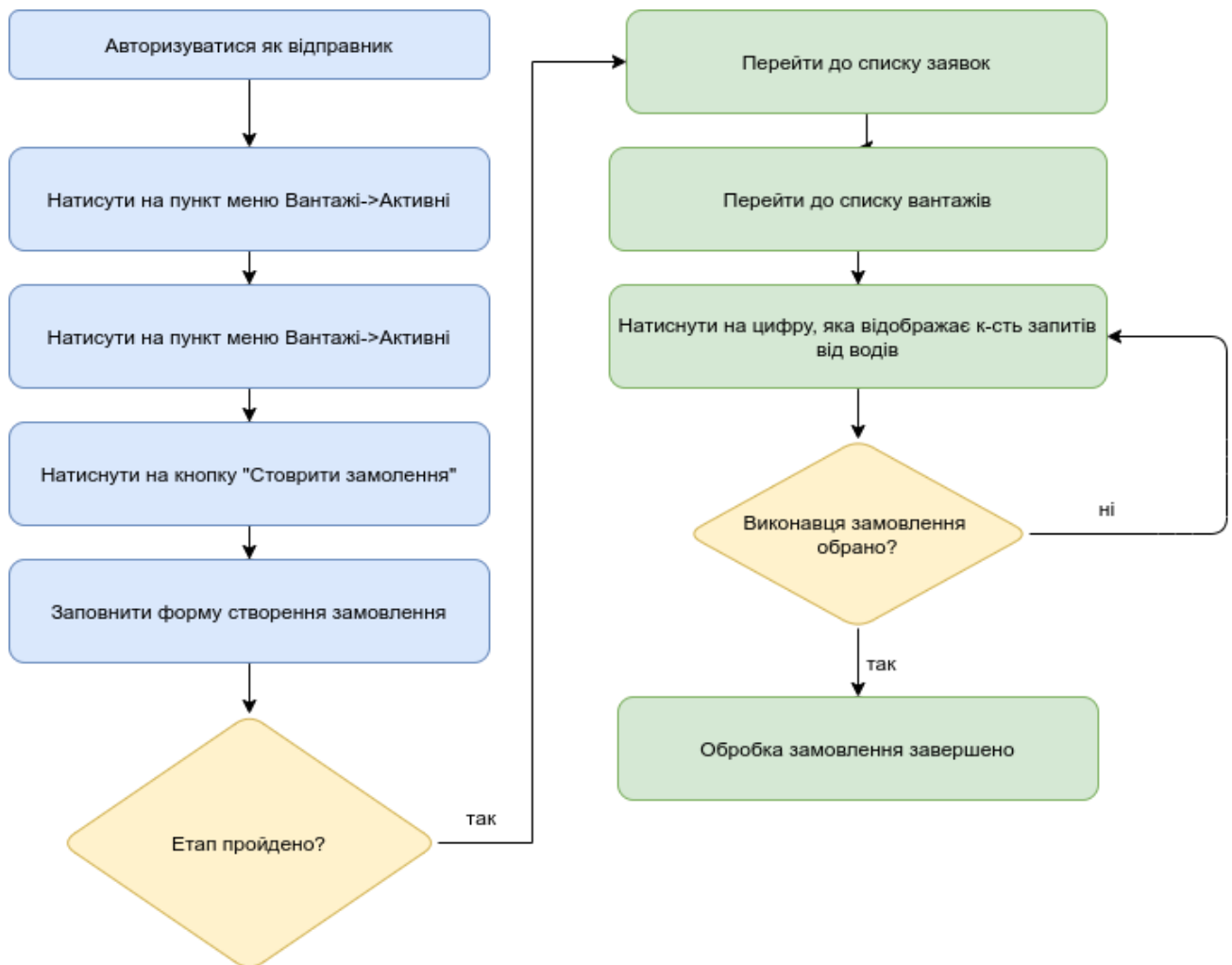


Рисунок 3.7 Бізнес процес “Планування маршруту”

Для того, щоб автоматизувати роботу веб-сервісу, дані потрібно організувати у таблиці, створити базу даних і провести нормалізацію БД.

Даталогічна модель БД створюється відповідно до бізнес-процесів та їх функцій (додаток А). Тому потрібно створити таблиці «Users», «Trucks»

«TruckTypes», «TruckSubtypes», «TruckCategory», «TrucksLoadingTypes», «Orders», «OrderRequests», «Notification». (рисунок 3.8).

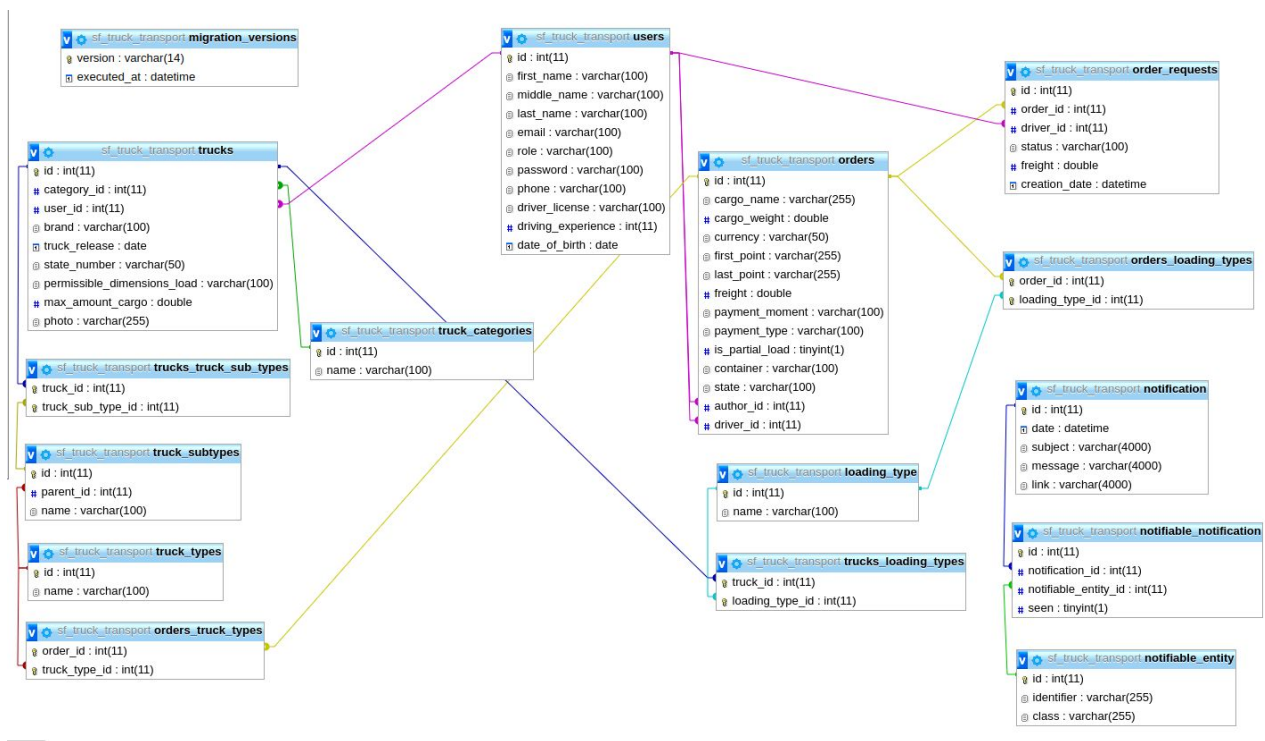


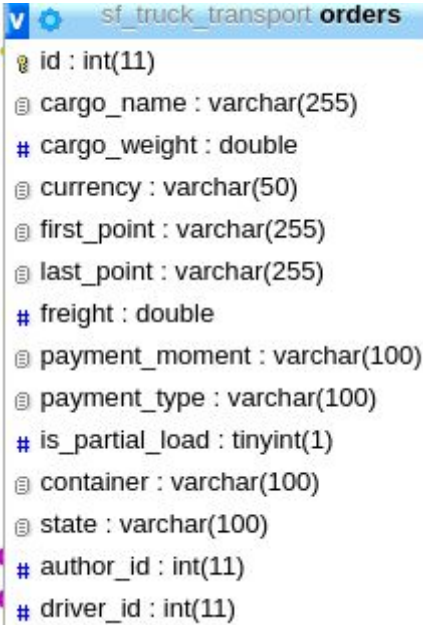
Рисунок 3.8 – Схема даних БД

Таблиця «Users» (рисунок 3.9). Вміщує в собі текстові поля «first_name», «last_name», «middle_name», «email_name», «role», «phone», «driver_license», а також числові поля: «id», «driving_experience» і одне поле з датою «date_of_brith».

sf_truck_transport users	
PK	id : int(11)
	first_name : varchar(100)
	middle_name : varchar(100)
	last_name : varchar(100)
	email : varchar(100)
	role : varchar(100)
	password : varchar(100)
	phone : varchar(100)
	driver_license : varchar(100)
	# driving_experience : int(11)
	date_of_birth : date

Рисунок 3.9 – Таблиця «Users»

Таблиця «Orders» (рисунок 3.10). Поле «id» є первинним ключем і має цілочисельний тип даних. Поля «cargo_name», «currency», «first_point», «last_point», «payment_moment», «payment_type», «state» мають стрічковий тип даних, «cargo_weight» та «freight» – тип double, а поля «author_id», «driver_id» мають цілочисельний тип даних, адже ці поля є вторинним ключем, що пов’язує його із таблицею «Users».



```

sf_truck_transport orders
id : int(11)
cargo_name : varchar(255)
# cargo_weight : double
currency : varchar(50)
first_point : varchar(255)
last_point : varchar(255)
# freight : double
payment_moment : varchar(100)
payment_type : varchar(100)
# is_partial_load : tinyint(1)
container : varchar(100)
state : varchar(100)
# author_id : int(11)
# driver_id : int(11)

```

Рисунок 3.10 – Таблиця «Orders»

Таблиця «OrderRequests» (рисунок 3.11) має поля «id», «order_id», «driver_id». Перше з них є первинним ключем і матиме цілочисельний тип, друге є вторинним ключем, що пов’язує з таблицею «Orders», а третє також є вторинним ключем, що пов’язує з таблицею «Users». Також дана таблиця має стрічкове поле «status», числове поле «freight» з типом double і поле з типом datetime «creation_date».

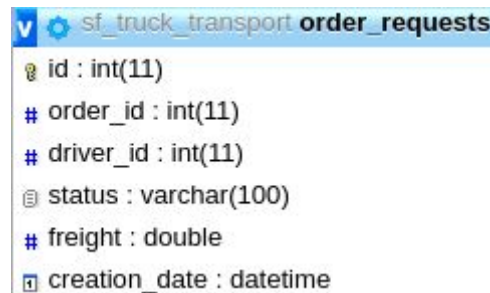


Рисунок 3.11 – Таблиця «OrderRequests»

Таблиця «Trucks» (рисунок 3.12) вміщує в собі такі числові поля з типом int: «id», «category_id», вторинний ключ, що з'єднює з таблицею «TruckCateogyr», «user_id», вторинний ключ, що з'єднює з таблицею «Users». Також дана таблиця вміщує стрічкові поля: «brand», «state_number», «photo», і одне поле типу date «truck_release».



Рисунок 3.12 – Таблиця «Trucks»

Таблиця «TruckCategories» (рисунок 3.13) матиме два поля «id» та «name». Перше з них буде первинним ключем і матиме цілочисельний тип, а інше – стрічковий тип.



Рисунок 3.13 – Таблиця «TruckCategories»

Таблиця «TruckTypes» (рисунок 3.14) матиме два поля «id» та «name». Перше з них буде первинним ключем і матиме цілочисельний тип, а інше – стрічковий тип.

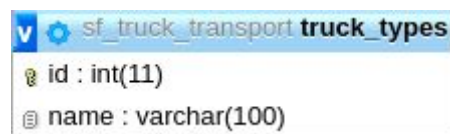


Рисунок 3.14 – Таблиця «TruckTypes»

Таблиця «TruckSubTypes» (рисунок 3.11). Поле «id» є первинним ключем і має цілочисельний тип даних. Поля «parent_id» є вторинним ключем, що з'єднює з таблицею «TruckTypes», «name» – стрічковий тип.



Рисунок 3.15 – Таблиця «TruckSubTypes»

Таблиця «LoadingTypes» (рисунок 3.16) матиме два поля «id» та «name». Перше з них буде первинним ключем і матиме цілочисельний тип, а інше – стрічковий тип.



Рисунок 3.16 – Таблиця «LoadingTypes»

Таблиця «Notification» (рисунок 3.17) складається з трьох стрічкових полів: «subject», «message», «link» і з одного поля «date» типу DateTime.

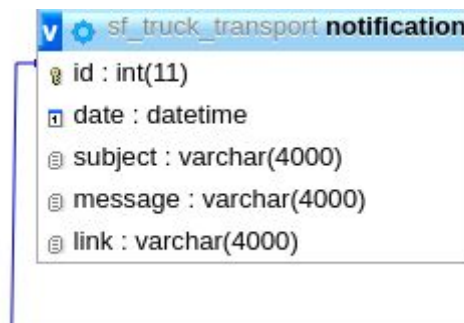


Рисунок 3.17 – Таблиця «Notification»

Серед сутностей виділимо три модулі: «Users», «Orders» та «OrderRequests» відповідно до розглянутих бізнес-процесів.

До даталогічної моделі бізнес-процесу «Обробка замовлень» (рисунок 3.18) віднесено таблиці «Trucks», «Orders», «OrderRequest», також дана даталогічна модель є актуальна до двох інших бізнес-процесів, а саме «Планування маршруту» і «Виконання замовлення».

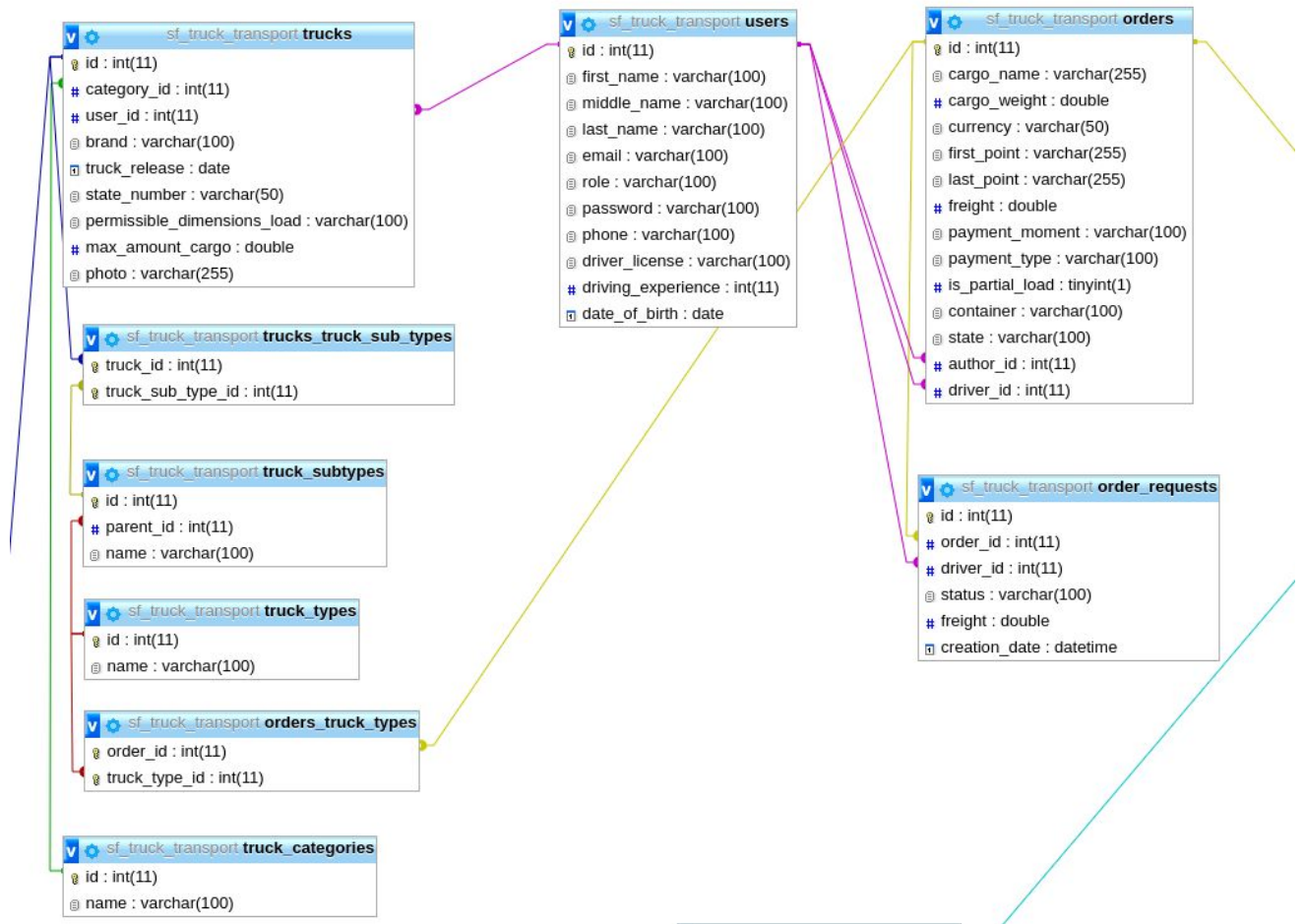


Рисунок 3.18 – Даталогічна модель бізнес-процесу «Обробка замовлень»

3.3 Розробка програмних модулів

Для реалізації веб-сервісу для розміщення заявок щодо перевезення було обрано мову програмування PHP фреймворк Symfony 4. БД створювалась за допомогою MySQL.

Реалізований веб-сервіс має свою власну структуру, яка відповідає функціям, які має виконувати сервіс для пошуку збірних вантажів. Діаграма класів веб-сервісу представлена на рисунках 3.19-3.20 та у додатку В.

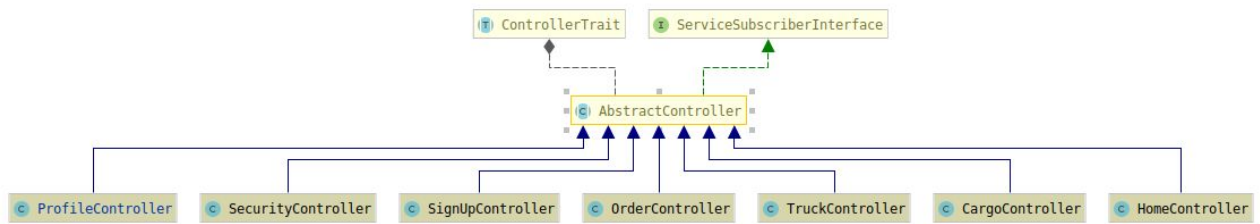


Рисунок 3.19 – Діаграма класів контролерів

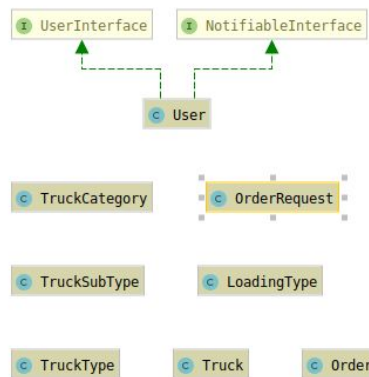


Рисунок 3.20 – Діаграма класів для доменної моделі сутностей
 «LoadingType», «Order», «OrderRequest», «Truck», «TruckCategory»,
 «TruckSubType», «TruckType», «User»

Для реалізації ІС пошуку найкращого маршруту потрібно розробити компоненти наступних видів: таблиці БД, доменну модель веб-додатку, MVC

контролери та представлення сторінок. Створені таблиці (рисунок 3.21): «LoadingType», «Order», «OrderRequest», «Truck», «TruckCategory», «TruckSubType», «TruckType», «User» заповнюємо даними, що відповідають даній предметній області та вимогам сутностей.

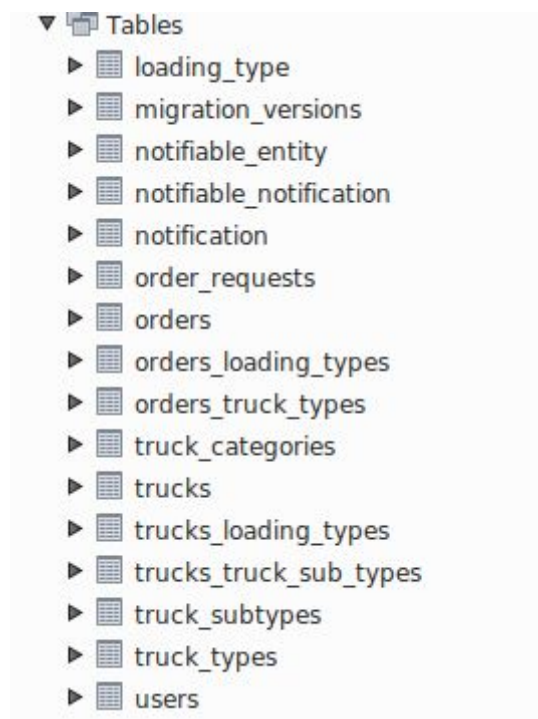


Рисунок 3.21 – Перелік таблиць створеної БД

Відповідно до створених таблиць створюємо моделі даних (рисунок 3.22) у доменній моделі веб-додатку.

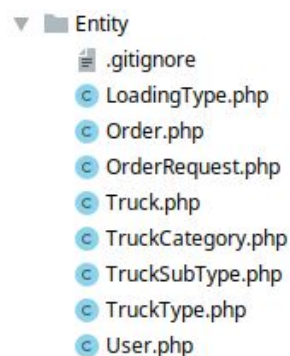


Рисунок 3.22 – Доменна модель веб-застосунку

На головній сторінці сайту «TruckTransport» відображається інформація про даний сервіс (рисунок 3.23). Головна сторінка формується програмним кодом, що розміщений у додатку В.

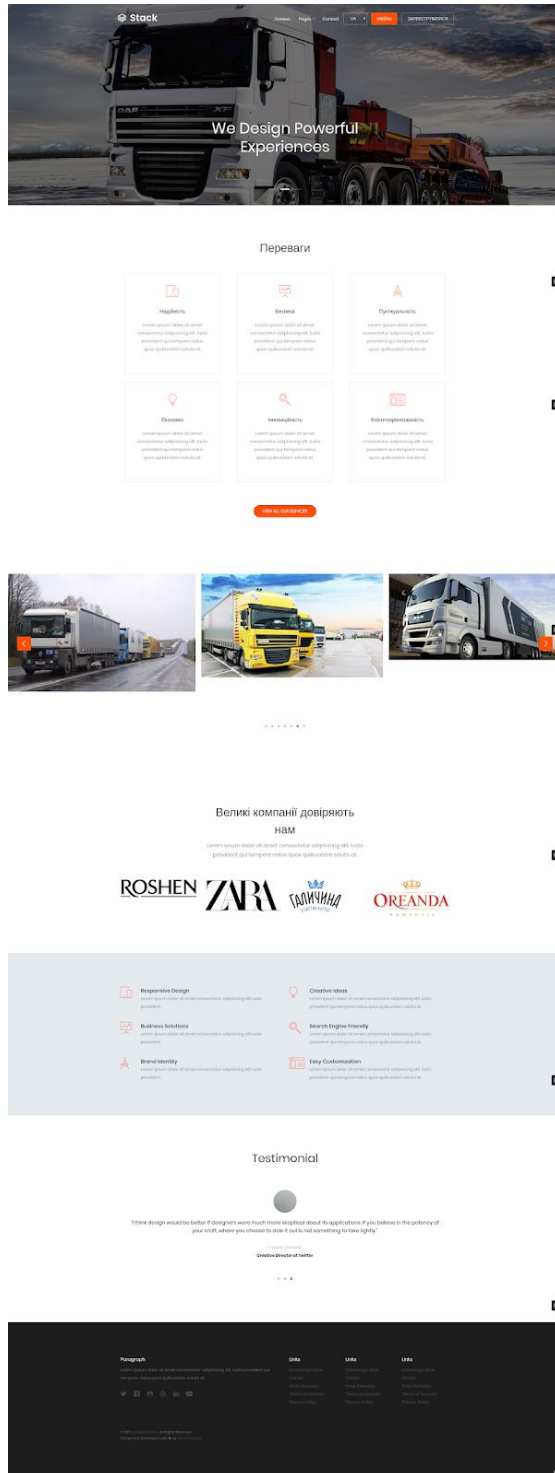


Рисунок 3.23 – Головна сторінка веб-сервісу

Додаткова інформація з контактами представлена на вкладці «Contact» (рисунок 3.24), програмна реалізація відповідної сторінки знаходиться у додатку В.

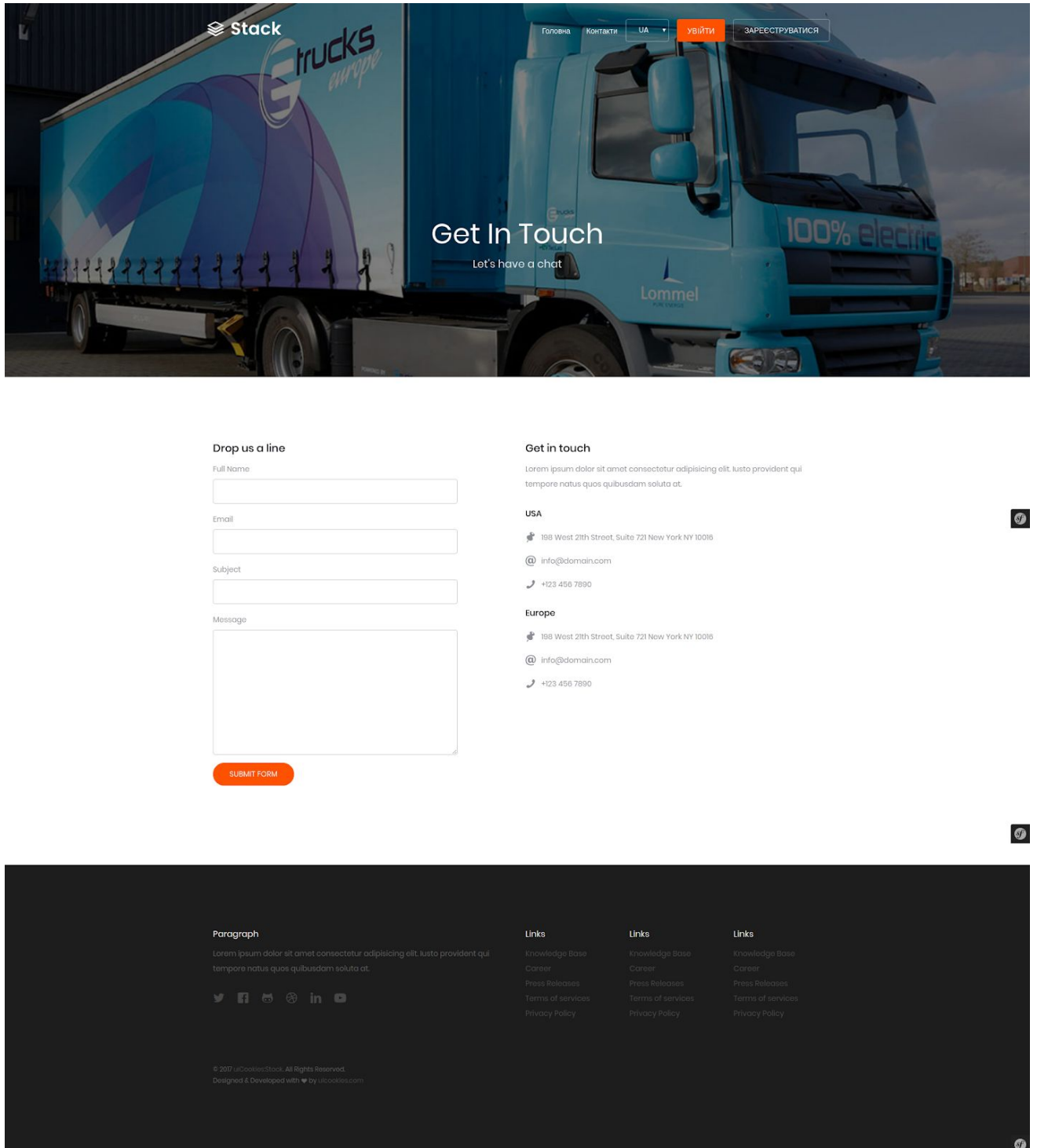


Рисунок 3.24 – Сторінка «Contacts»

Після авторизації клієнт отримує доступ до особистого кабінету (рисунок 3.25). Лістинг знаходиться у додатку В.

3.4 Висновки до розділу

В даному розділі було описано бізнес-процеси, представлено та описано структуру бази даних. Обрані засоби розробки інформаційної системи, а також представлено її структуру, класи, взаємодію між ними. Було розроблену схему функцій користувача і описано макети користувацького інтерфейсу. Реалізовано програмний веб-застосунок інформаційної система планування найкращого шляху для доставки вантажу за допомогою задачі комівояжера

Розділ 4

Апробація програмної інформаційної системи

4.1 Порядок використання інформаційної системи

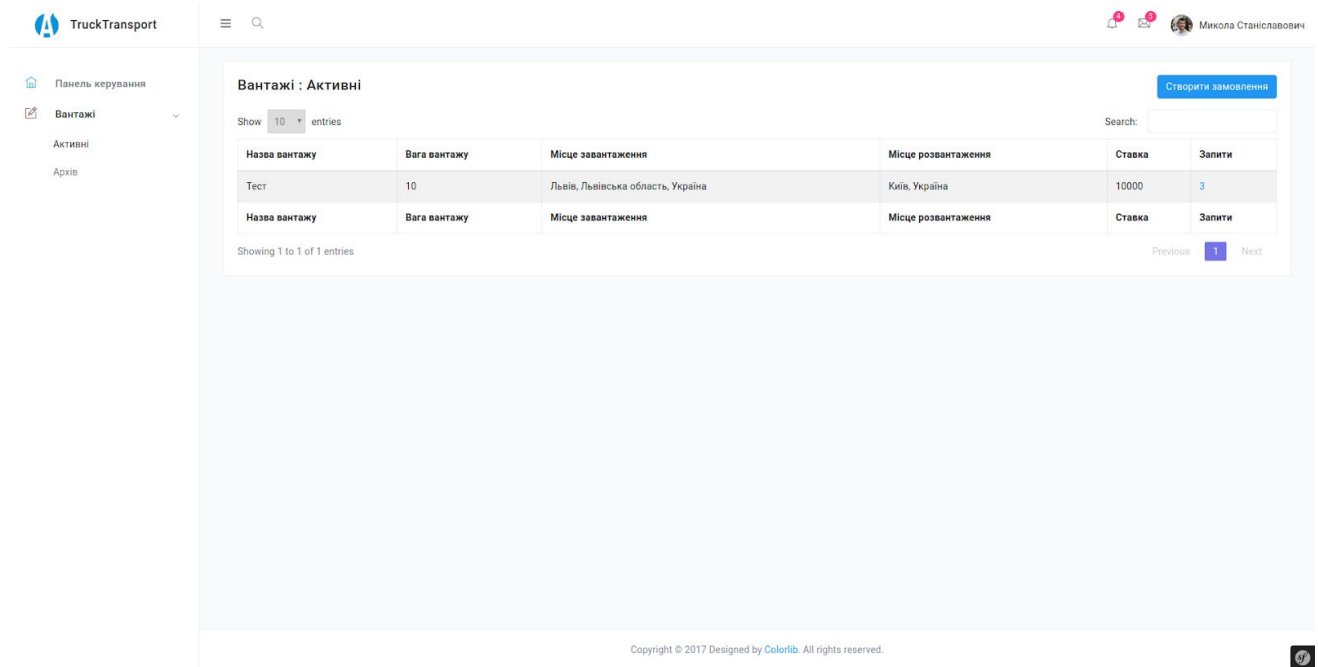


Рисунок 4.5 – Особистий кабінет клієнта

Водії також мають свій особистий кабінет. Код реалізації знаходиться в додатку В. Результат виконання програмного коду можна побачити на рисунку 4.6.

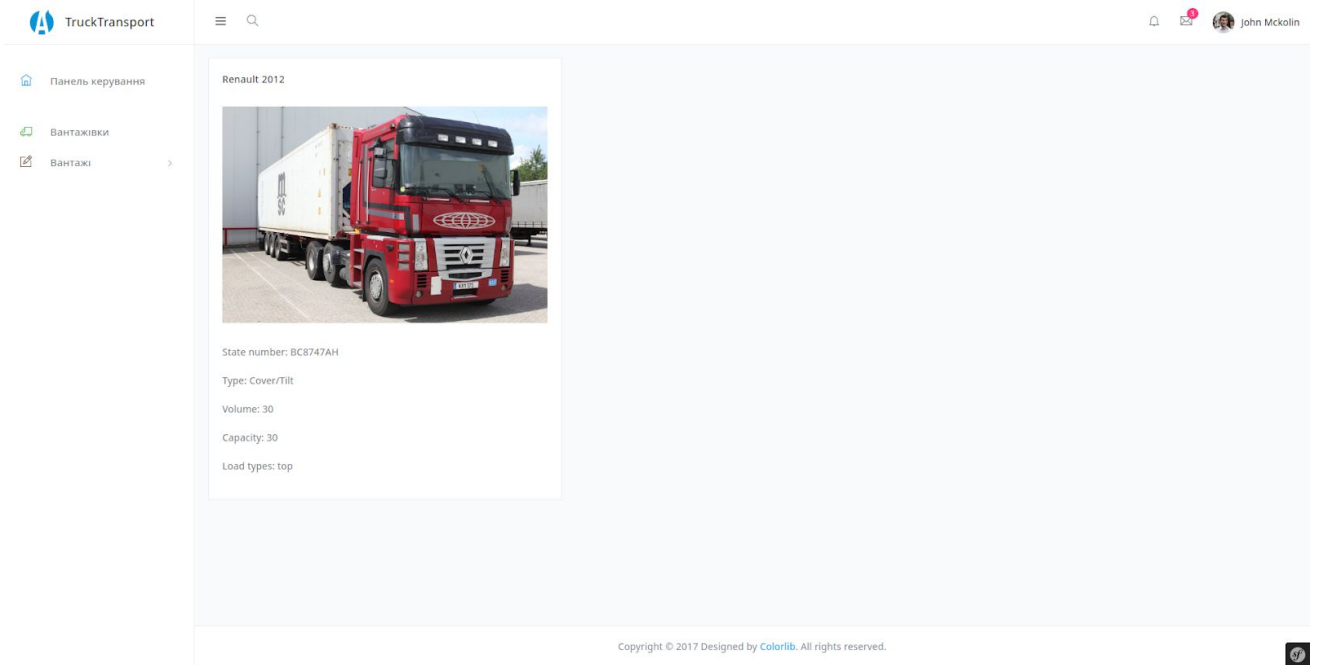


Рисунок 4.6 – Особистий кабінет водія

Пошук замовлень реалізовано програмним кодом, що знаходиться в додатку В. Представлення того, що бачить користувач зображено на рисунку 4.7.

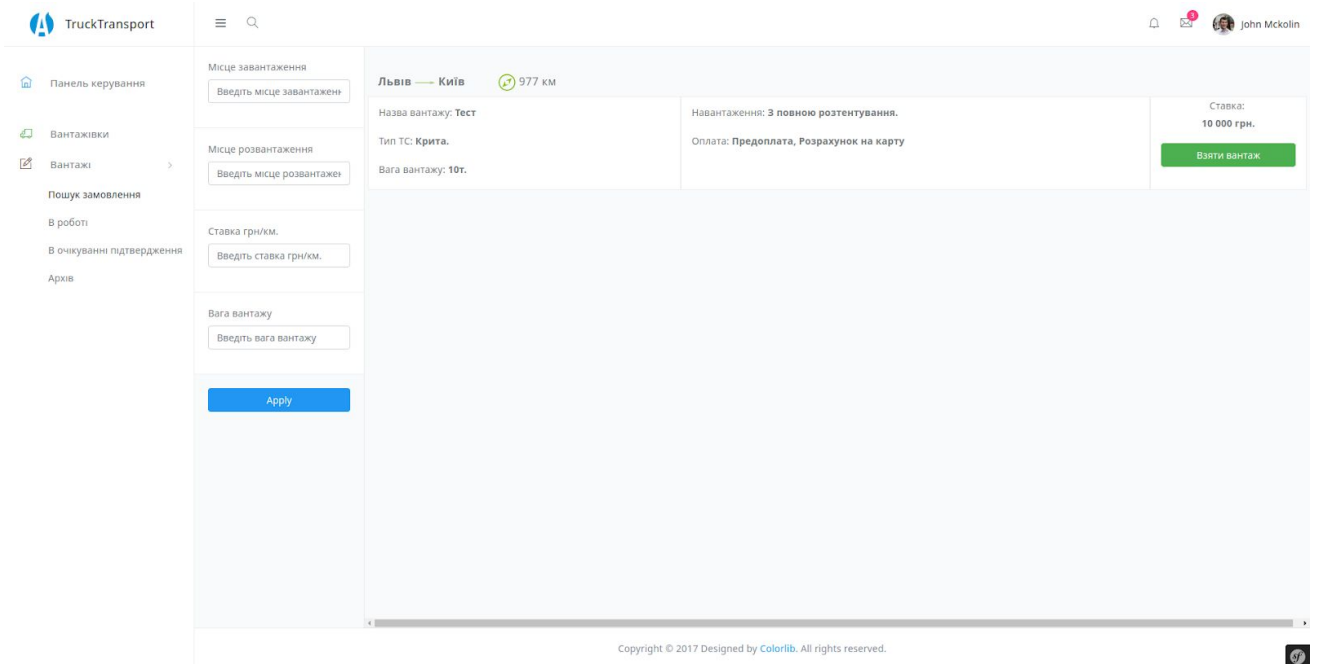


Рисунок 4.7 – Інтерфейс пошуку вантажів

Створення замовлення клієнтом реалізовано у програмному кодї додатку В. На сторінці відображається форма редагування/додавання елемента (рисунок 4.8).

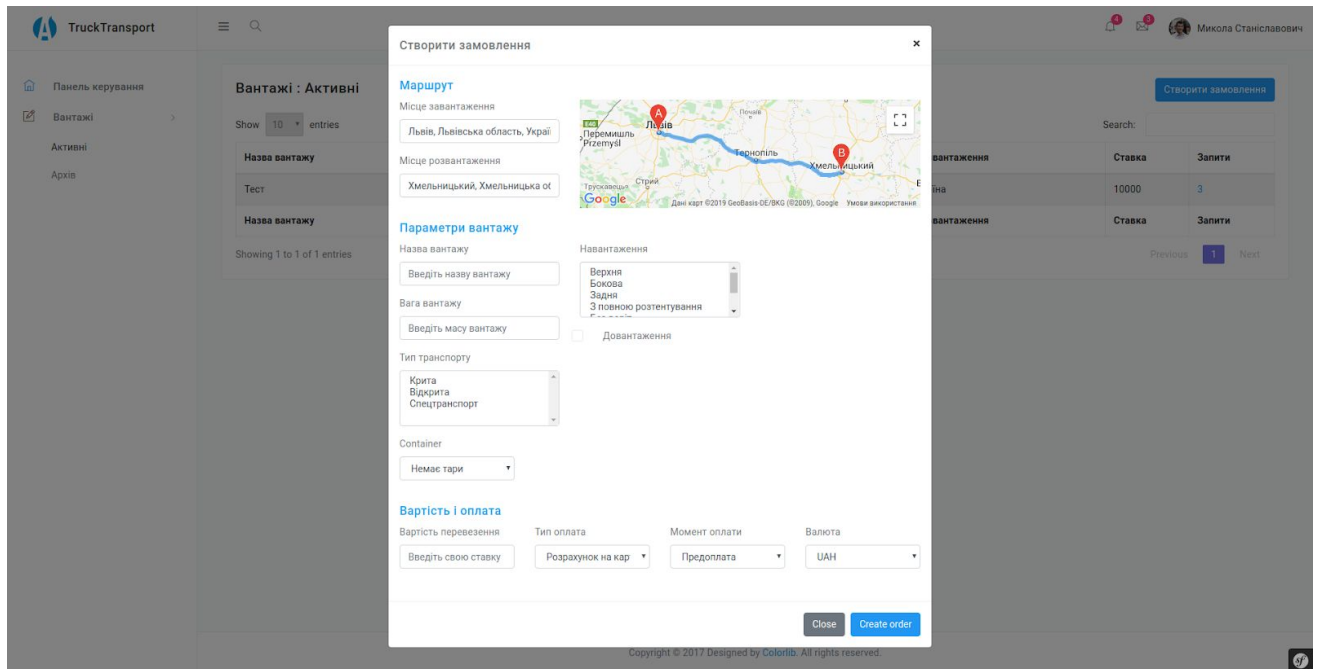


Рисунок 4.8– Інтерфейс створення замовлення

За допомогою розроблених програмних модулів були автоматизовані бізнес-процеси взаємодії клієнта і водія вантажівки.

Для перевірки правильності роботи інформаційної системи пошуку найкращого маршруту вантажного перевезення за допомогою задачі комівояжера, було взяти наступні вхідні дані:

- точка початку вантажного перевезення Львів;
- проміжна точка Чернівці;
- проміжна точка Тернопіль;
- кінцева точка вантажного перевезення Хмельницький.

4.2 Порівняльний аналіз додати декілька маршрутів

Симуляція №1 Львів-Івано-Франківськ-Тернопіль-Рівне:

Розроблена ІС

The screenshot displays the TruckTransport web application interface. On the left is a navigation menu with options like 'Панель керування', 'Вантажівки', 'Вантажі', 'Пошук замовлення', 'В роботі', 'В очікуванні підтвердження', and 'Архів'. The main area shows a route simulation with the following details:

- Route:** Львів — Івано-Франківськ — Тернопіль — Рівне (423 km)
- Назва вантажу:** Картопля
- Навантаження:** Бокова.
- Тип ТС:** тент.
- Оплата:** Предоплата, Розрахунок на карту
- Вага вантажу:** 40т.
- Ставка:** 40 грн.
- Взяти вантаж** (button)

Additional fields include 'Місце завантаження' (Львів, Львівська область), 'Проміжна точка #1' (Івано-Франківськ, Івано-Франківська область), 'Проміжна точка #2' (Тернопіль, Тернопільська область), 'Місце розвантаження' (Рівне, Рівненська область), and 'Ставка грн/км.' (Введіть ставку грн/км.).

Google Maps

The screenshot shows Google Maps with a route from Lviv to Rivne via Ivano-Frankivsk and Ternopil. The route is highlighted in blue and takes 7 hours and 422 km. The interface includes a search bar, a list of destinations, and various map controls. The route starts in Lviv, goes to Ivano-Frankivsk, then to Ternopil, and finally to Rivne. A red arrow points to the route information box in the bottom right corner of the map, which displays '7 hr' and '422 km'.

Transportica

TRANSPORTICA.COM | ІНФОРМАЦІЙНИЙ СЕРВІС

Вантаж | Калькулятор | **Для перевізників** | Вхід | Реєстрація

Головна → Розрахунок відстаней

Відстань Львів — Рівне ~ 403 км

Щоб розрахувати відстань по маршруту заповніть поля звідки/ куди. Для розрахунку вартості поїздки додатково вкажіть середню витрату палива вашого автомобіля і ціну за літр на АЗС.

Відстань між містами **Львів — Рівне: 403 км** Друк

Приблизний час у дорозі: 5 г 45 хв

- ■ Львів, Львівська область
- ■ Івано-Франківськ, Івано-Франківська область
- ■ Тернопіль, Тернопільська область
- ■ Рівне, Рівненська область

Звідки
 ✕
Додати пункт

Через
 ✕
Додати пункт

Через
 ✕
Додати пункт

Куди
 ✕
Додати пункт

Розрахувати відстань

Витрати палива (л/100км) **Ціна за літр (грн)**

Ваш автомобіль
 Легковий Вантажний

Симуляція №2 Тернопіль-Чернівці-Ужгород-Львів:

Розроблена ІС

TruckTransport

Місце завантаження: Тернопіль, Тернопільськ

Проміжна точка #1: Чернівці, Чернівецька о

Проміжна точка #2: Ужгород, Закарпатська с

Місце розвантаження: Львів, Львівська область

Ставка грн/км.

Тернопіль — Чернівці — Ужгород — Львів ● 871 км

Назва вантажу: Пошта

Тип ТС: тент.

Вага вантажу: 1т.

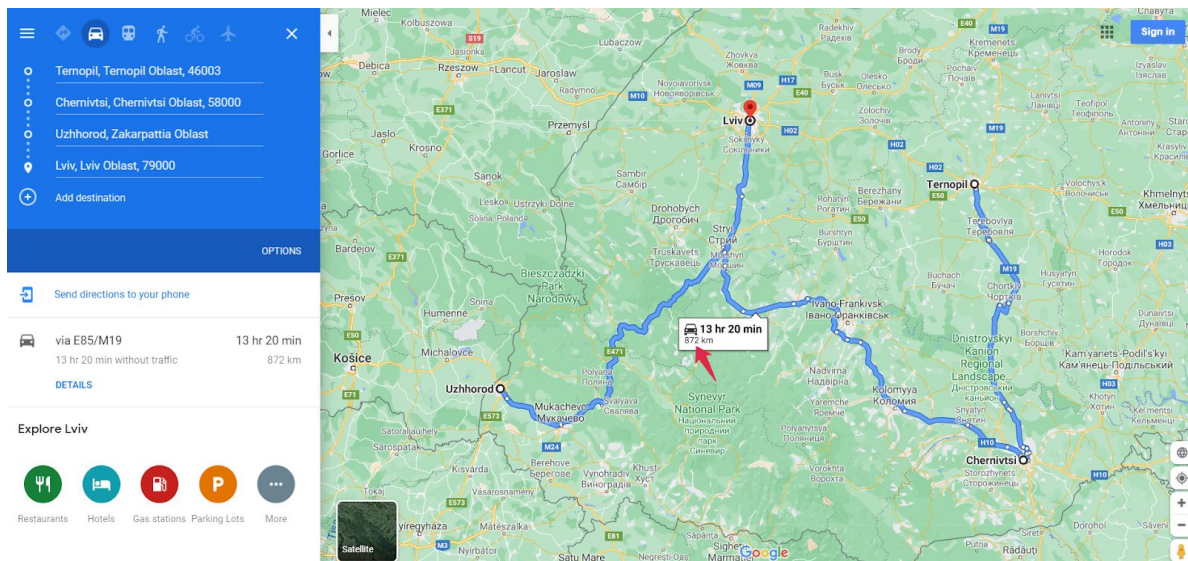
Навантаження: Бокова, Задня.

Оплата: Предоплата, Розрахунок на карту

Ставка: 12 грн./км

Взяти вантаж

Google Maps



Transportica

TRANSPORTICA.COM | ІНФОРМАЦІЙНИЙ СЕРВІС

Вантаж [Калькулятор](#) [Для перевізників](#) Вхід | Реєстрація

Головна → Розрахунок відстаней

Відстань Тернопіль – Львів ~ 867 км

Щоб розрахувати відстань по маршруту заповніть поля звідки/ куди. Для розрахунку вартості поїздки додатково вкажіть середню витрату палива вашого автомобіля і ціну за літр на АЗС.

Відстань між містами **Тернопіль – Львів: 867 км** ↗ Друк

Приблизний час у дорозі: 12 г 23 хв

- Тернопіль, Тернопільська область
- Чернівці, Чернівецька область
- Ужгород, Закарпатська область
- Львів, Львівська область

Звідки

Тернопіль, Тернопільська область ×

Додати пункт

Через

Чернівці, Чернівецька область ×

Додати пункт

Через

Ужгород, Закарпатська область ×

Додати пункт

Куди

Львів, Львівська область ×

Додати пункт

Витрати палива (л/100км)

Ціна за літр (грн)

Ваш автомобіль

Легковий Вантажний

Розрахувати відстань

Симуляція №3 Ужгород-Чернівці-Хмельницький-Львів:

Розроблена ІС:

The screenshot displays the TruckTransport web application interface. On the left is a navigation menu with options: Панель керування, Вантажівки, Вантажі, Пошук замовлення, В роботі, В очікуванні підтвердження, and Архів. The main area contains a form for creating a simulation:

- Місце завантаження:** Ужгород, Закарпатська с
- Місце розвантаження:** Львів, Львівська область
- Ставка грн/км:** Введіть ставку грн/км.
- Вага вантажу:** Введіть вагу вантажу

Below the form are buttons for "Apply" and "Add additional point". The right side of the interface shows a summary of the simulation:

- Route:** Ужгород — Чернівці — Хмельницький — Львів (860 km)
- Назва вантажу:** Овочі
- Тип ТС:** тент.
- Вага вантажу:** 15т.
- Навантаження:** Бокова, Задня.
- Оплата:** Предоплата, Розрахунок на карту
- Ставка:** 16 грн./км
- Action:** Взяти вантаж

Google Maps

The screenshot shows Google Maps with a route from Uzhhorod to Lviv. The route is highlighted in blue and passes through several cities including Mukachevo, Drohobych, Striy, and Ternopil. A red arrow points to a specific location on the route near Drohobych. The map interface includes a search bar, a list of destinations, and a summary of the route:

- Route:** Uzhhorod, Zakarpattia Oblast to Lviv, Lviv Oblast, 79000
- Distance:** 863 km
- Duration:** 13 hr 2 min
- Route:** via E471/E50/M06 and H10

Below the route summary are icons for exploring nearby points of interest: Restaurants, Hotels, Gas stations, and Parking Lots.

Transportica

TRANSPORTICA.COM СЕРВІС

Головна → Розрахунок відстаней

Відстань Ужгород – Львів ~ 858 км

Щоб розрахувати відстань по маршруту заповніть поля звідки/ куди. Для розрахунку вартості поїздки додатково вкажіть середню витрату палива вашого автомобіля і ціну за літр на АЗС.

Відстань між містами **Ужгород – Львів: 858 км** Друк

Приблизна час у дорозі: 12 г 15 хв

- Ужгород, Закарпатська область
- Чернівці, Чернівецька область
- Хмельницький, Хмельницька область
- Львів, Львівська область

Звідки: ×

Через: ×

Через: ×

Куди: ×

Витрати палива (л/100км):

Ціна за літр (грн):

Ваш автомобіль: Легковий Вантажний

Порівняння отриманих результатів представлено в таблиці 4.1

Симуляція	Розроблена ІС	Google Maps	Transportica	Похибка
1	423 км	422 км	403 км	6.7 %
2	871 км	872 км	867 км	0.7 %
3	860 км	863 км	858 км	0.7 %

Таблиця 4.1 - Результати випробування

Проведений порівняльний аналіз засвідчує достатньо точність підрахунку довжини маршруту. Відхилення величини довжини маршруту, від отриманих за допомогою інших картографічних сервісів, не перевищує 7%, що у поєднанні із вбудованим калькулятором вартості визначає наявні переваги реалізованої ІС прокладання маршруту.

4.3 Висновки до розділу

Виконано порівняльний аналіз визначення довжини маршруту за допомогою реалізованої ІС та сучасних картографічних сервісів. Величина відхилення маршруту не перевищує 7 %. Інтегрований калькулятор вартості створює додаткові зручності при використанні реалізованої ІС для вирішення логістичних задач вантажних перевезень.

Загальні висновки

Проведений аналіз літературних джерел засвідчує нагальну необхідність у логістичній підтримці виробництва і сфери послуг. Це обумовлено розробкою нових інформаційних технологій, які дозволяють розвивати суспільство в тих чи інших напрямках.

Світова глобалізація формує значні і складні виклики щодо забезпечення логістичної підтримки сучасного виробництва. Зв'язку з одночасним розвитком інформаційних технологій, їх високою ефективністю, пристосованістю до вирішення формалізованих задач широкого спектру застосування, найбільш доцільним є вирішення задач організації перевезень на основі інформаційних технологій та систем.

В результаті виконання ДРМ виконано такі завдання дослідження метою якого була реалізація інформаційної системи для пошуку найкращого маршруту для вантажного перевезення за допомогою задачі комівояжера:

5. проведено аналіз існуючих методів, технологій та рішень пошуку найкращого маршруту доставки вантажу, що ґрунтується на задачі комівояжера;

6. вдосконалено існуючих методик використання задачі комівояжера для пошуку найкращого маршруту на основі мурашиного алгоритму із реалізацією вбудованого калькулятора вартості перевезення;

7. розроблено інформаційну систему побудови найкращого маршруту вантажних перевезень, яка поєднує розв'язки задачі комівояжера з сучасними картографічними сервісами;

8. виконано експериментальну перевірку інформаційної системи планування найкращого шляху для доставки вантажу.

Реалізовано інформаційну модель системи знаходж..., яка ґрунтується на мурашиному алгоритмі, що дозволяє віднайти оптимальний маршрут з інтегрованим розрахунком вартості перевезення.

Виконано порівняльний аналіз визначення довжини маршруту за допомогою реалізованої ІС та сучасних картографічних сервісів. Величина відхилення маршруту не перевищує %. Інтегрований калькулятор вартості створює додаткові зручності при використанні реалізованої ІС для вирішення логістичних задач вантажних перевезень. В даному розділі було описано бізнес-процеси, представлено та описано структуру бази даних. Обрані засоби розробки інформаційної системи, а також представлено її структуру, класи, взаємодію між ними. Було розроблену схему функцій користувача і описано макети користувацького інтерфейсу. Реалізовано програмний веб-застосунок для пошуку оптимального маршруту і розрахунку вар вантажного перевезення.

Перелік посилань

1. Алесінская Т. В. Основы логістики. Загальні питання логістичного управління / Т. В. Алесінская. - Таганрог: Изд-во ТРТУ, 2005. - 121 с.
2. Кузьменко В.М. Аналіз сучасних методів штучного інтелекту / В.М. Кузьменко, С.В. Таран // № 1Е (6), 2006. - С.16-19. 4.
3. Гладков Л.А. Генетичні алгоритми. / Л.А. Гладков, В.М. Курейчик, В.В. Курейчик // - Ростов-на-Дону: ТОВ «Ростіздат», 2004р. - С.105-109.
4. Чураков М.А. Мурашиний алгоритм. / М.А. Чураков, Якушев А.Є., 2006. - 21с.
5. Чураков М.А. Мурашиний алгоритм. / М.А. Чураков, Якушев А.Є., 2006. - 21с. Штовба С. Д. Мурашині алгоритми. Математика в додатках, 2003 №4, стор. 70-75.
6. A. T. Kearney. Creating value through strategic supply management. 2004 Assessment off excellence in procurement. 2005, p. 1.
7. Mahajan S., Sandip Saha S., Macias A. Analytics: Laying the Foundation for Supply Chain DigitalTransformation. November, 2017
URL:<https://www.thehackettgroup.com/wp-content/uploads/2017>
8. Transportica - Режим доступу: <https://transportica.com/>
9. Lardi Trans - Режим доступу: <https://lardi-trans.com/en/>
10. Goss, S., Aron S., Deneubourg J.L., and Pasteels, J.M., Self-Organized Shortcuts in the Argentine Ant, Naturwissenschaften, 1989, no. 76, pp. 579–581
11. The Traveling Salesman Problem: A Computational Study / D.L. Applegate, R.E. Bixby, V. Chvátal & W.J. Cook (2007). // Princeton University Press 2007
12. Глава 3. Метод грубой силы: Задача коммивояжера. Алгоритмы: введение в разработку и анализ (Introduction to The Design and Analysis of Algorithms). / Ананий В. Левитин (2006). // М.: «Вильямс». с. 159–160. ISBN 0-201-74395-7.

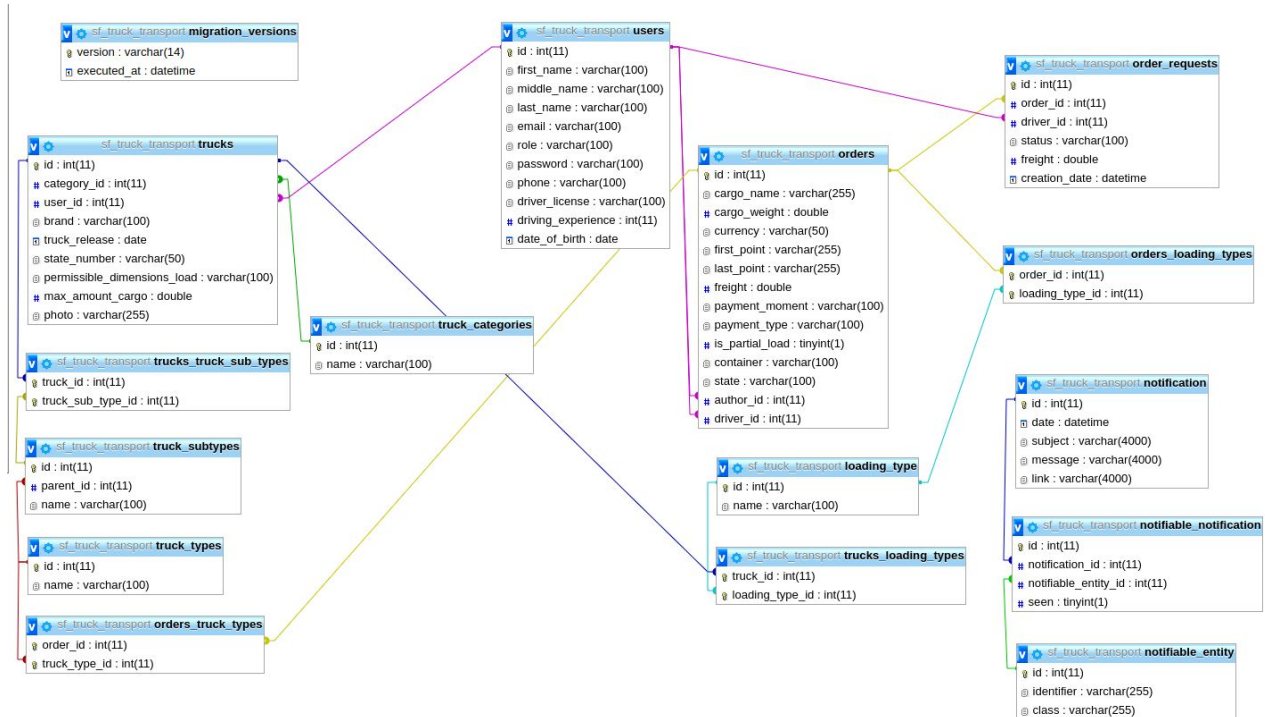
13. Combinatorial Optimization (вид. 3-тє). / Bernhard Korte, Jens Vygen (2006). // Springer. ISBN 3-540-25684-9.
14. Fast heuristics for large geometric traveling salesman problems. / Reinelt, Gerhard (1992) // ORSA Journal on computing, 4:206-217
15. Эффективность грузовых автомобильных перевозок: Состояние, проблемы, перспективы. / Ковалев В.П. // Мн.: Беларусь, 1984. — 112 с.
16. Математические методы в логистике. / Просветов Г.И. // М.: РДЛ, 2006. — 272 с.
17. Задача коммивояжера. Вопросы теории / Меламед И.И., Сергеев С.И., Сигал И.Х // Автоматика и телемеханика. – № 9. – 1989. – С. 3-33.
18. The clustered traveling salesman problem / Chisman J.A. // Computers & Operations Research. – Volume 2. – Issue 2. – September 1975. – P. 115-119.
19. Внукова З.А. Інтелектуальна система вибору маршруту в логістиці з використанням алгоритму мурахи / З.А. Внукова, І.В. Свиридова, В.Є. Шопські, Е.Е. Мандрикіна, А.К. Кичигина // 2014.
20. Гладков Л.А. Генетичні алгоритми. / Л.А. Гладков, В.М. Курейчик, В.В. Курейчик // - Ростов-на-Дону: ТОВ «Ростіздат», 2004р.
21. Курейчик В.В. Мурашиний алгоритм для розв'язання оптимізаційних задач з явно вираженою цільовою функцією / В.В. Курейчик, М.А. Жиленков // Інформатика, обчислювальна техніка та інженерну освіту. - 2015.
22. МакКоннелл Дж. Основи сучасних алгоритмів. - М .: Техносфера, 2004.
23. Семенюта Е.В. Застосування мурашиних алгоритмів для пошуку оптимальних маршрутів вантажоперевезень / Е.В. Семенюта, М.В. Привалов // Донецький національний технічний університет, кафедра автоматизованих систем управління - 1997.
24. Сергеев В.І. Менеджмент в бізнес-логістики. М: 2-е вид. 2003.
25. Чураков М.А. Муравьиный алгоритм. / М.А. Чураков, Якушев А.Е., 2006.

26. Штовба С. Д. Мурашині алгоритми. Математика в додатках, 2003 №4,
27. Щербаков В.С. Оптимізація траси автомобільної дороги на рельєфі з перешкодами методом ймовірнісної дорожньої карти / В.С. Щербаков, М.С. Коритов // Вісник СибАДИ. 2012. №2.
28. Bhattacharya P. Roadmap-based path planning- Using theVoronoi diagram for a clearance-based shortest path / P.Bhattacharya, M. L. GavriloVA // Robotics & Automation Magazine, IEEE. 2008. №15 (2) .
29. Pan P. Improved Ant Colony Algorithm for Path Planning of Soccer Robot // Journal of Convergence Information Technolog. 2013.№8(7).
30. Schrijver A. On the history of combinatorial optimization (till 1960) // Handbook of Discrete Optimization. 2005.
31. Vasek Chvatal, William J. Cook, George B. Dantzig, Delbert Ray Fulkerson, and Selmer M. Johnson. Solution of a large-scale traveling-salesman problem. In 50 Years of Integer Programming 1958-2008 - From the Early Years to the State-of-the-Art. 2010.
32. H. Abeledo, R. Fukasawa, A. Pessoa, and E. Uchoa, The Time Dependent Traveling Salesman Problem: Polyhedra and Algorithm, Mathematical Programming Computation 5 (2013).
33. M.Dorigo, G. Di Caro, and L.M. Gambardella, “Ant algorithms for discrete optimization,” Artificial Life, vol. 5, no. 2, 1999.
34. Hitoshi K., Junichi O. Solving Time-Dependent Traveling Salesman Problems Using Ant Colony Optimization Based on Predicted Traffic // Advances in Intelligent and Soft Computing. 2012.
35. Dorigo, M., Stutzle, T. Ant colony optimization. The MIT press, 2004.

ДОДАТКИ

Додаток А

Структура бази даних інформаційної системи для пошуку оптимального маршрута для вантажоперевезень



Додаток В

Програмні коди

Лістинг HomeController.php

```
<?php

namespace App\Controller;

use
Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\Routing\Annotation\Route;

class HomeController extends AbstractController
{
    /**
     * @Route("/", name="home")
     * @Route("/{_locale}",
requirements={"_locale": "uk|en"},
name="home-translations")
     */
    public function index(Request $request)
    {
        return
$this->render('home/index.html.twig', [
            'controller_name' =>
'HomeController',
        ]);
    }

    /**
     * @Route("/contacts", name="contacts")
     * @Route("/{_locale}/contacts",
requirements={"_locale": "uk|en"},
name="contacts-translations")
     */
    public function contacts(Request $request)
    {
        return
$this->render('home/contacts.html.twig');
    }
}
```

Лістинг SecurityController.php

```
<?php

namespace App\Controller;

use
Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;
use
Symfony\Component\Security\Http\Authentication\AuthenticationUtils;

class SecurityController extends
AbstractController
{
    /**
     * @Route("/login", name="app_login")
     */
    public function login(AuthenticationUtils
$authenticationUtils): Response
    {
        // get the login error if there is one
        $error =
$this->getLastAuthenticationError();
        // last username entered by the user
        $lastUsername =
$this->getLastUsername();

        return $this->redirectToRoute('home');
    }
}
```

Лістинг SignUpController.php

```

<?php

namespace App\Controller;

use App\Entity\User;
use App\Form\DriverForm;
use App\Form\ShipperForm;
use
Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\Routing\Annotation\Route;
use
Symfony\Component\Security\Core\Authentication\Token\UsernamePasswordToken;

class SignUpController extends AbstractController
{
    /**
     * @Route("/sign-up", name="sign-up")
     * @Route("/{_locale}/sign-up")
     */
    public function index(Request $request)
    {
        $user = new User();

        $driverForm =
$this->createForm(DriverForm::class, $user);
        $driverForm->handleRequest($request);

        $shipperForm =
$this->createForm(ShipperForm::class, $user);
        $shipperForm->handleRequest($request);

        $response =
$this->render('signup/index.html.twig', ['driverForm'
=> $driverForm->createView(), 'shipperForm' =>
$shipperForm->createView()]);

        if ($driverForm->isSubmitted() &&
$driverForm->isValid()) {
            $em = $this->getDoctrine()->getManager();

            $user->setRole('ROLE_DRIVER');
            $em->persist($user);
            $em->flush();

            $this->auth($user);

            $response = $this->redirectToRoute('home');
        }

        if ($shipperForm->isSubmitted() &&
$shipperForm->isValid()) {
            $em = $this->getDoctrine()->getManager();

            $user->setRole('ROLE_SHIPPER');
            $em->persist($user);
            $em->flush();

            $this->auth($user);

            $response = $this->redirectToRoute('home');
        }

        return $response;
    }

    private function auth(User $user)
    {
        $token = new UsernamePasswordToken($user, null,
'main', $user->getRoles());

$this->container->get('security.token_storage')->setToken($token);

$this->container->get('session')->set('_security_main',
serialize($token));
    }
}

```

Лістинг ProfileController.php

```

<?php

namespace App\Controller;

use App\Entity\Order;
use
Sensio\Bundle\FrameworkExtraBundle\Configuration\
Security;
use
Symfony\Bundle\FrameworkBundle\Controller\
AbstractController;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\Routing\Annotation\Route;

class ProfileController extends
AbstractController
{
    /**
     * @Route("/profile", name="profile")
     * @Route("/{_locale}/profile",
requirements={"_locale": "uk|en"})
     * @Security("is_granted('ROLE_USER')")
     */
    public function index(Request $request)
    {
        $em = $this->getDoctrine()->getManager();
        $user = $this->getUser();
        $params = [
            'totalActiveOrders' => 0,
            'totalInWorkOrders' => 0,
            'totalOrders' => 0
        ];

        if ($user->getRole() === 'ROLE_SHIPPER')
        {
            $ordersRepository =
$em->getRepository(Order::class);

            $params['totalActiveOrders'] =
count($ordersRepository->findBy(['author' =>
$user->getId(), 'state' => 'active']));
            $params['totalInWorkOrders'] =
count($ordersRepository->findBy(['author' =>
$user->getId(), 'state' => 'inWork']));
            $params['totalOrders'] =
count($ordersRepository->findBy(['author' =>
$user->getId()]));
        }

        return
$this->render('profile/index.html.twig',
$params);
    }
}

```

Лістинг CargoController.php

```

<?php
namespace App\Controller;

use App\Entity\Order;
use App\Entity\OrderRequest;
use App\Entity\Truck;
use App\Entity\User;
use App\Form\OrderForm;
use
Sensio\Bundle\FrameworkExtraBundle\Configuration\Param
converter;
use
Sensio\Bundle\FrameworkExtraBundle\Configuration\Securi
ty;
use
Symfony\Bundle\FrameworkBundle\Controller\AbstractContr
oller;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;

class CargoController extends AbstractController
{
    /**
     * @Route("/cargoes", name="cargoes")
     * @Route("/{_locale}/cargoes",
     requirements={"_locale": "uk|en"})
     * @Security("is_granted('ROLE_USER')")
     */
    public function index(Request $request)
    {
        $params = $request->query->all();

        $repository =
$this->getDoctrine()->getRepository(Order::class);
        $params = array_merge($params, ['author' =>
$this->getUser()->getId()]);
        $orders = $repository->findBy($params);

        $order = new Order();
        $order->setAuthor($this->getUser());

        $orderForm = $this->createForm(OrderForm::class,
$order);
        $orderForm->handleRequest($request);
        if ($request->getMethod() === 'POST' &&
$orderForm->isSubmitted() && $orderForm->isValid()) {
            $sem = $this->getDoctrine()->getManager();
            $sem->persist($order);
            $sem->flush();

            $response =
$this->redirectToRoute('cargoes');
        } else {
            $response = $this->render(
                'profile/cargoes.html.twig',
                ['orders' => $orders, 'orderForm' =>
$orderForm->createView()]);
        }

        return $response;
    }

    /**
     * @Route("/cargoes/search", name="cargoes-search")
     * @Route("/{_locale}/cargoes/search",
     requirements={"_locale": "uk|en"})
     * @Security("is_granted('ROLE_USER')")
     */
    public function search(Request $request)
    {
        $repository =
$this->getDoctrine()->getRepository(Order::class);

        $params = array_filter($request->query->all());

        $orders = $repository->findBy($params);

        return
$this->render('profile/search-cargoes.html.twig',
                ['orders' => $orders]);
    }

    /**
     * @Route("/cargoes/take", name="cargoes-take")
     * @Route("/{_locale}/cargoes/take",

```

Лістинг OrderController.php

```

<?php
namespace App\Controller;

use App\Entity\Order;
use App\Entity\OrderRequest;
use App\Entity\Truck;
use App\Entity\User;
use App\Form\OrderForm;
use
Sensio\Bundle\FrameworkExtraBundle\Configuration\Param
converter;
use
Sensio\Bundle\FrameworkExtraBundle\Configuration\Securi
ty;
use
Symfony\Bundle\FrameworkBundle\Controller\AbstractContr
oller;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;
use
Symfony\Component\HttpKernel\Exception\AccessDeniedHttp
Exception;
use Symfony\Component\Routing\Annotation\Route;

class OrderController extends AbstractController
{
    /**
     * @Route("/delete/order/{order}",
     name="delete-order")
     * @Route("/{_locale}/delete/order/{order}",
     requirements={"_locale": "uk|en"})
     * @Security("is_granted('ROLE_USER')")
     *
     * @ParamConverter("order",
     class="App\Entity\Order")
     */
    public function deleteOrder(Order $order)
    {
        $user = $this->getUser();

        if ($order->getAuthor()->getId() ===
$user->getId()) {
            $sem = $this->getDoctrine()->getManager();

            $sem->remove($order);
            $sem->flush();
        } else {
            throw new AccessDeniedHttpException();
        }

        return $this->redirectToRoute('cargoes');
    }

    /**
     * @Route("/edit/order/{order}", name="edit-order")
     * @Route("/{_locale}/edit/order/{order}",
     requirements={"_locale": "uk|en"})
     * @Security("is_granted('ROLE_USER')")
     *
     * @ParamConverter("order",
     class="App\Entity\Order")
     */
    public function editOrder(Order $order, Request
$request)
    {
        $user = $this->getUser();

        if ($order->getAuthor()->getId() ===
$user->getId()) {
            $orderForm =
$this->createForm(OrderForm::class, $order);

            $orderForm->handleRequest($request);
            if ($request->getMethod() === 'POST' &&
$orderForm->isSubmitted() && $orderForm->isValid()) {
                $sem =
$this->getDoctrine()->getManager();
                $sem->persist($order);
                $sem->flush();

                $response =
$this->redirectToRoute('cargoes');
            } else {
                $response =
$this->render('profile/edit-order.html.twig',
                    ['orderForm' => $orderForm->createView()]);
            }
        }
    }

```

```

requirements={"_locale": "uk|en"})
 * @Security("is_granted('ROLE_USER')")
 */
public function take(Request $request)
{
    $orderId = $request->query->get('order');
    $freight = $request->query->get('freight');

    if ($orderId) {
        $em =
$this->getDoctrine()->getManager();
        $orderRepository =
$this->getDoctrine()->getRepository(Order::class);

        $order =
$orderRepository->findOneById($orderId);

        if ($order instanceof Order) {
            $orderRequest = new OrderRequest();

$orderRequest->setDriver($this->getUser());
            $orderRequest->setOrder($order);
            $orderRequest->setFreight($freight);

            $em->persist($orderRequest);
            $em->flush();
        }
    }

    return new Response();
}

/**
 * @Route("/cargoes/requests",
name="cargoes-order-requests")
 * @Route("/{_locale}/cargoes/requests",
requirements={"_locale": "uk|en"})
 * @Security("is_granted('ROLE_USER')")
 */
public function cargoRequests(Request $request)
{
    $user = $this->getUser();
    $em =
$this->getDoctrine()->getManager();
    $ordersRequestsRepository =
$em->getRepository(OrderRequest::class);
    $ordersRequests = [];
    $orderId =
$request->query->get('orderId');

    if ($user instanceof User) {
        if ($user->getRole() === 'ROLE_DRIVER') {
            $ordersRequests =
$ordersRequestsRepository->findBy(['driver' =>
$user->getId()]);
        } elseif ($user->getRole() ===
'ROLE_SHIPPER' && $orderId) {
            $orderRepository =
$em->getRepository(Order::class);

            $order =
$orderRepository->findOneById($orderId);

            if ($order instanceof Order &&
$order->getAuthor()->getId() === $user->getId()) {
                $ordersRequests =
$ordersRequestsRepository->findBy(['order' =>
$order->getId()]);
            }
        }
    }

    return
$this->render('profile/cargoes-requests.html.twig',
['ordersRequests' => $ordersRequests]);
}

/**
 * @Route("/cargoes/requests/{orderRequest}",
name="cargoes-order-request")
 *
 * @Route("/{_locale}/cargoes/requests/{orderRequest}",
requirements={"_locale": "uk|en"})

```

```

}
    } else {
        throw new AccessDeniedHttpException();
    }
}

return $response;
}
}

```

Лістинг TruckController.php

```

<?php
namespace App\Controller;

use App\Entity\Truck;
use App\Form\TruckForm;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Security;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\Routing\Annotation\Route;

class TruckController extends AbstractController
{
    /**
     * @Route("/trucks", name="profile-trucks")
     * @Route("/{_locale}/trucks",
requirements={"_locale": "uk|en"})
     * @Security("is_granted('ROLE_DRIVER')")
     */
    public function index(Request $request)
    {
        $trucks =
$this->getDoctrine()->getRepository('App:Truck')->findBy(['user' => $this->getUser()->getId()]);
        $truck = new Truck();
        $truckForm = $this->createForm(TruckForm::class,
$truck);
        $truckForm->handleRequest($request);
        if ($request->getMethod() === 'POST' &&
$truckForm->isSubmitted() && $truckForm->isValid()) {
            $em = $this->getDoctrine()->getManager();
            $em->persist($truck);
            $em->flush();

            $response =
$this->redirectToRoute('profile-trucks');
        } else {
            $response =
$this->render('profile/trucks.html.twig', ['trucks' =>
$trucks, 'truckForm' => $truckForm->createView()]);
        }

        return $response;
    }
}

```

Лістинг User.php

```

<?php
namespace App\Entity;

use Doctrine\ORM\Mapping as ORM;
use Symfony\Component\Security\Core\User\UserInterface;
use Mgilet\NotificationBundle\Annotation\Notifiable;
use Mgilet\NotificationBundle\NotifiableInterface;

/**
 *
 * @ORM\Entity(repositoryClass="App\Repository\UserRepository")
 * @ORM\Table(name="users")
 * @ORM\HasLifecycleCallbacks()
 *
 * @ORM\EntityListeners({"App\EventListener\DoctrineEntityListener\UserEntityListener"})
 * @Notifiable(name="user ")
 */
class User implements UserInterface,
NotifiableInterface
{
    /**

```

```

    * @Security("is_granted('ROLE_USER')")
    *
    * @ParamConverter("orderRequest",
class="App\Entity\OrderRequest")
    */
    public function cargoRequestsByOrder(OrderRequest
$orderRequest)
    {
        $truckRepository =
$this->getDoctrine()->getRepository(Truck::class);

        $truck = $truckRepository->findBy(['user' =>
$orderRequest->getDriver()->getId()]);

        return
$this->render('profile/cargoes-request.html.twig',
['orderRequest' => $orderRequest, 'truck' =>
current($truck)]);
    }

/**
 * @Route("/cargoes/requests/{orderRequest}/accept",
name="cargoes-order-request-accept")
 *
 * @Route("/{_locale}/cargoes/requests/{orderRequest}/acce
pt", requirements={"_locale": "uk|en"})
 * @Security("is_granted('ROLE_USER')")
 *
 * @ParamConverter("orderRequest",
class="App\Entity\OrderRequest")
    */
    public function cargoRequestsAccept(OrderRequest
$orderRequest)
    {
        if
($orderRequest->getOrder()->getAuthor()->getId() ===
$this->getUser()->getId()) {
            $sem = $this->getDoctrine()->getManager();
            $order = $orderRequest->getOrder();

$order->setDriver($orderRequest->getDriver());
$order->setState('inWork');

            $orderRequest->setStatus('accepted');

            $sem->persist($orderRequest);
            $sem->persist($order);

            $sem->flush();
        }
    }

/**
 * @Route("/cargoes/requests/{orderRequest}/cancel",
name="cargoes-order-request-cancel")
 *
 * @Route("/{_locale}/cargoes/requests/{orderRequest}/canc
el", requirements={"_locale": "uk|en"})
 * @Security("is_granted('ROLE_USER')")
 *
 * @ParamConverter("orderRequest",
class="App\Entity\OrderRequest")
    */
    public function cargoRequestsCancel(OrderRequest
$orderRequest)
    {
        if
($orderRequest->getOrder()->getAuthor()->getId() ===
$this->getUser()->getId()) {
            $sem = $this->getDoctrine()->getManager();

            $orderRequest->setStatus('canceled');

            $sem->persist($orderRequest);
            $sem->flush();
        }
    }
}

```

Лістинг Order.php

```

<?php
namespace App\Entity;

use Doctrine\Common\Collections\ArrayCollection;
use Doctrine\Common\Collections\Collection;
use Doctrine\ORM\Mapping as ORM;

/**

```

```

 * @ORM\Id()
 * @ORM\GeneratedValue()
 * @ORM\Column(type="integer")
 */
private $id;

/**
 * @ORM\Column(type="string", length=100)
 */
private $firstName;

/**
 * @ORM\Column(type="string", length=100)
 */
private $middleName;

/**
 * @ORM\Column(type="string", length=100)
 */
private $lastName;

/**
 * @ORM\Column(type="string", length=100)
 */
private $email;

/**
 * @ORM\Column(type="string", length=100)
 */
private $role;

/**
 * @ORM\Column(type="string", length=100)
 */
private $password;

private $plainPassword;

/**
 * @ORM\Column(type="string", length=100)
 */
private $phone;

/**
 * @ORM\Column(type="string", length=100,
nullable=true)
 */
private $driverLicense;

/**
 * @ORM\Column(type="integer", nullable=true)
 */
private $drivingExperience;

/**
 * @ORM\Column(type="date", nullable=true)
 */
private $dateOfBirth;

public function getId(): ?int
{
    return $this->id;
}

public function getFirstName(): ?string
{
    return $this->firstName;
}

public function setFirstName(string $firstName):
self
{
    $this->firstName = $firstName;

    return $this;
}

public function getMiddleName(): ?string
{
    return $this->middleName;
}

public function setMiddleName(string $middleName):
self
{
    $this->middleName = $middleName;

    return $this;
}

public function getLastName(): ?string

```

```

*
@ORM\Entity(repositoryClass="App\Repository\OrderRepository")
* @ORM\Table(name="orders")
*/
class Order
{
    /**
     * @ORM\Id()
     * @ORM\GeneratedValue()
     * @ORM\Column(type="integer")
     */
    private $id;

    /**
     * @ORM\Column(type="string", length=255)
     */
    private $cargoName;

    /**
     * @ORM\Column(type="float")
     */
    private $cargoWeight;

    /**
     * @ORM\Column(type="string", length=50)
     */
    private $currency;

    /**
     * @ORM\Column(type="string", length=255)
     */
    private $firstPoint;

    /**
     * @ORM\Column(type="string", length=255)
     */
    private $lastPoint;

    /**
     * @ORM\Column(type="float")
     */
    private $freight;

    /**
     * @ORM\ManyToOne(targetEntity="LoadingType")
     * @ORM\JoinTable(name="orders_loading_types",
     *
     joinColumns={@ORM\JoinColumn(name="order_id",
     referencedColumnName="id")},
     *
     inverseJoinColumns={@ORM\JoinColumn(name="loading_type_id",
     referencedColumnName="id")}
     *
     )
     */
    private $loadingTypes;

    /**
     * @ORM\ManyToOne(targetEntity="TruckType")
     * @ORM\JoinTable(name="orders_truck_types",
     *
     joinColumns={@ORM\JoinColumn(name="order_id",
     referencedColumnName="id")},
     *
     inverseJoinColumns={@ORM\JoinColumn(name="truck_type_id",
     referencedColumnName="id")}
     *
     )
     */
    private $transportTypes;

    /**
     * @ORM\Column(type="string", length=100)
     */
    private $paymentMoment;

    /**
     * @ORM\Column(type="string", length=100)
     */
    private $paymentType;

    /**
     * @ORM\Column(type="boolean")
     */
    private $isPartialLoad;

    /**
     * @ORM\Column(type="string", length=100)
     */
    private $container;

    /**
     * @ORM\ManyToOne(targetEntity="User")

```

```

{
    return $this->lastName;
}

public function setLastName(string $lastName): self
{
    $this->lastName = $lastName;

    return $this;
}

public function getEmail(): ?string
{
    return $this->email;
}

public function setEmail(string $email): self
{
    $this->email = $email;

    return $this;
}

public function getRole(): ?string
{
    return $this->role;
}

public function setRole(string $role): self
{
    $this->role = $role;

    return $this;
}

public function getPassword(): ?string
{
    return $this->password;
}

public function setPassword(string $password): self
{
    $this->password = $password;

    return $this;
}

public function getPlainPassword(): ?string
{
    return $this->plainPassword;
}

public function setPlainPassword(string $plainPassword): self
{
    $this->plainPassword = $plainPassword;

    return $this;
}

/**
 * Returns the roles granted to the user.
 *
 *     public function getRoles()
 *     {
 *         return array('ROLE_USER');
 *     }
 *
 * Alternatively, the roles might be stored on a
 * `roles` property,
 * and populated in any number of different ways
 * when the user object
 * is created.
 *
 * @return (Role|string)[] The user roles
 */
public function getRoles()
{
    return [$this->role];
}

/**
 * Returns the salt that was originally used to
 * encode the password.
 *
 * This can return null if the password was not
 * encoded using a salt.
 *
 * @return string|null The salt
 */
public function getSalt()

```

```

    * @ORM\JoinColumn(name="author_id",
referencedColumnName="id", onDelete="SET NULL")
    */
    private $author;

    /**
     * @ORM\ManyToOne(targetEntity="User")
     * @ORM\JoinColumn(name="driver_id",
referencedColumnName="id", onDelete="SET NULL")
     */
    private $driver;
    /**
     * @ORM\Column(type="string", length=100)
     */
    private $state = 'new';

    public function __construct()
    {
        $this->loadingTypes = new ArrayCollection();
        $this->transportTypes = new ArrayCollection();
    }

    public function getId(): ?int
    {
        return $this->id;
    }

    public function getCargoName(): ?string
    {
        return $this->cargoName;
    }

    public function setCargoName(string $cargoName):
self
    {
        $this->cargoName = $cargoName;

        return $this;
    }

    public function getCargoWeight(): ?float
    {
        return $this->cargoWeight;
    }

    public function setCargoWeight(float $cargoWeight):
self
    {
        $this->cargoWeight = $cargoWeight;

        return $this;
    }

    public function getCurrency(): ?string
    {
        return $this->currency;
    }

    public function setCurrency(string $currency): self
    {
        $this->currency = $currency;

        return $this;
    }

    public function getFirstPoint(): ?string
    {
        return $this->firstPoint;
    }

    public function setFirstPoint(string $firstPoint):
self
    {
        $this->firstPoint = $firstPoint;

        return $this;
    }

    public function getLastPoint(): ?string
    {
        return $this->lastPoint;
    }

    public function setLastPoint(string $lastPoint):
self
    {
        $this->lastPoint = $lastPoint;

        return $this;
    }
}

```

```

    {
        // TODO: Implement getSalt() method.
    }

    /**
     * Returns the username used to authenticate the
user.
     *
     * @return string The username
     */
    public function getUsername()
    {
        return $this->getEmail();
    }

    /**
     * Removes sensitive data from the user.
     *
     * This is important if, at any given point,
sensitive information like
     * the plain-text password is stored on this object.
     */
    public function eraseCredentials()
    {
        // TODO: Implement eraseCredentials() method.
    }

    public function getPhone(): ?string
    {
        return $this->phone;
    }

    public function setPhone(string $phone): self
    {
        $this->phone = $phone;

        return $this;
    }

    public function getDriverLicense(): ?string
    {
        return $this->driverLicense;
    }

    public function setDriverLicense(string
$driverLicense): self
    {
        $this->driverLicense = $driverLicense;

        return $this;
    }

    public function getDrivingExperience(): ?int
    {
        return $this->drivingExperience;
    }

    public function setDrivingExperience(?int
$drivingExperience): self
    {
        $this->drivingExperience = $drivingExperience;

        return $this;
    }

    public function getDateOfBirth():
?\DateTimeInterface
    {
        return $this->dateOfBirth;
    }

    public function setDateOfBirth(?\DateTimeInterface
$dateOfBirth): self
    {
        $this->dateOfBirth = $dateOfBirth;

        return $this;
    }
}

```

Лістинг OrderRequest.php

```

<?php
namespace App\Entity;

use Doctrine\Common\Collections\ArrayCollection;
use Doctrine\Common\Collections\Collection;
use Doctrine\ORM\Mapping as ORM;

```

```

public function getFreight(): ?float
{
    return $this->freight;
}

public function seFreight(float $freight): self
{
    $this->freight = $freight;

    return $this;
}

public function getPaymentMoment(): ?string
{
    return $this->paymentMoment;
}

public function setPaymentMoment(string
$paymentMoment): self
{
    $this->paymentMoment = $paymentMoment;

    return $this;
}

public function getPaymentType(): ?string
{
    return $this->paymentType;
}

public function setPaymentType(string $paymentType):
self
{
    $this->paymentType = $paymentType;

    return $this;
}

public function getIsPartialLoad(): ?bool
{
    return $this->isPartialLoad;
}

public function setIsPartialLoad(bool
$isPartialLoad): self
{
    $this->isPartialLoad = $isPartialLoad;

    return $this;
}

public function getContainer(): ?string
{
    return $this->container;
}

public function setContainer(string $container):
self
{
    $this->container = $container;

    return $this;
}

public function getState(): ?string
{
    return $this->state;
}

public function setState(string $state): self
{
    $this->state = $state;

    return $this;
}

/**
 * @return Collection|LoadingType[]
 */
public function getLoadingTypes(): Collection
{
    return $this->loadingTypes;
}

public function addLoadingType(LoadingType
$loadingType): self
{
    if
(!$this->loadingTypes->contains($loadingType)) {
        $this->loadingTypes[] = $loadingType;
    }
}

/**
 * @ORM\Entity()
 * @ORM\Table(name="order_requests")
 * @ORM\HasLifecycleCallbacks()
 */
@ORM\EntityListeners({"App\EventListener\DoctrineEntity
Listener\OrderRequestEntityListener"})
/**
 * Class OrderRequest
 */
class OrderRequest
{
    /**
     * @ORM\Id()
     * @ORM\GeneratedValue()
     * @ORM\Column(type="integer")
     */
    private $id;

    /**
     * @ORM\ManyToOne(targetEntity="Order")
     * @ORM\JoinColumn(name="order_id",
referencedColumnName="id", onDelete="SET NULL")
     */
    private $order;

    /**
     * @ORM\ManyToOne(targetEntity="User")
     * @ORM\JoinColumn(name="driver_id",
referencedColumnName="id", onDelete="SET NULL")
     */
    private $driver;
    /**
     * @ORM\Column(type="string", length=100)
     */
    private $status = 'pending';

    /**
     * @ORM\Column(type="float")
     */
    private $freight;

    /**
     * @ORM\Column(type="datetime")
     */
    private $creationDate;

    public function getId(): ?int
    {
        return $this->id;
    }

    public function getStatus(): ?string
    {
        return $this->status;
    }

    public function setStatus(string $status): self
    {
        $this->status = $status;

        return $this;
    }

    public function getOrder(): ?Order
    {
        return $this->order;
    }

    public function setOrder(?Order $order): self
    {
        $this->order = $order;

        return $this;
    }

    public function getDriver(): ?User
    {
        return $this->driver;
    }

    public function setDriver(?User $driver): self
    {
        $this->driver = $driver;

        return $this;
    }

    public function getFreight(): ?float
    {
        return $this->freight;
    }
}

```

```

        return $this;
    }

    public function removeLoadingType (LoadingType
$loadingType): self
    {
        if ($this->loadingTypes->contains ($loadingType))
        {
            $this->loadingTypes->removeElement ($loadingType);
        }

        return $this;
    }

    /**
     * @return Collection|TruckType[]
     */
    public function getTransportTypes(): Collection
    {
        return $this->transportTypes;
    }

    public function addTransportType (TruckType
$transportType): self
    {
        if
        (! $this->transportTypes->contains ($transportType)) {
            $this->transportTypes[] = $transportType;
        }

        return $this;
    }

    public function removeTransportType (TruckType
$transportType): self
    {
        if
        ($this->transportTypes->contains ($transportType)) {
            $this->transportTypes->removeElement ($transportType);
        }

        return $this;
    }

    public function getAuthor(): ?User
    {
        return $this->author;
    }

    public function setAuthor (?User $author): self
    {
        $this->author = $author;

        return $this;
    }

    public function getDriver(): ?User
    {
        return $this->driver;
    }

    public function setDriver (?User $driver): self
    {
        $this->driver = $driver;

        return $this;
    }

    public function setFreight (float $freight): self
    {
        $this->freight = $freight;

        return $this;
    }
}

```

Лістинг Truck.php

```
<?php
```

```

namespace App\Entity;

use Doctrine\Common\Collections\ArrayCollection;
use Doctrine\Common\Collections\Collection;
use Doctrine\ORM\Mapping as ORM;

```

```

    public function setFreight (float $freight): self
    {
        $this->freight = $freight;

        return $this;
    }

    public function getCreationDate(): ?\DateTimeInterface
    {
        return $this->creationDate;
    }

    /**
     * @param \DateTimeInterface $creationDate
     *
     * @return OrderRequest
     *
     * @ORM\PrePersist()
     */
    public function setCreationDate(): self
    {
        $this->creationDate = new \DateTime();

        return $this;
    }
}

```

Лістинг LoadingType.php

```
<?php
```

```

namespace App\Entity;

use Doctrine\Common\Collections\ArrayCollection;
use Doctrine\Common\Collections\Collection;
use Doctrine\ORM\Mapping as ORM;

/**
 *
 * @ORM\Entity (repositoryClass="App\Repository>LoadingType
Repository")
 */
class LoadingType
{
    /**
     * @ORM\Id()
     * @ORM\GeneratedValue()
     * @ORM\Column (type="integer")
     */
    private $id;

    /**
     * @ORM\Column (type="string", length=100)
     */
    private $name;

    public function getId(): ?int
    {
        return $this->id;
    }

    public function getName(): ?string
    {
        return $this->name;
    }

    public function setName (string $name): self
    {
        $this->name = $name;

        return $this;
    }
}

```

Лістинг TruckType.php

```
<?php
```

```

namespace App\Entity;

use Doctrine\Common\Collections\ArrayCollection;
use Doctrine\Common\Collections\Collection;
use Doctrine\ORM\Mapping as ORM;

/**
 *
 * @ORM\Entity (repositoryClass="App\Repository>TruckTypeRe

```

```

/**
 *
 * @ORM\Entity(repositoryClass="App\Repository\TruckRepository")
 * @ORM\Table(name="trucks")
 */
class Truck
{
    /**
     * @ORM\Id()
     * @ORM\GeneratedValue()
     * @ORM\Column(type="integer")
     */
    private $id;

    /**
     * @ORM\Column(type="string", length=100)
     */
    private $brand;

    /**
     * @ORM\Column(type="string", length=100)
     */
    private $model;

    /**
     * @ORM\Column(type="integer")
     */
    private $truckRelease;

    /**
     * @ORM\Column(type="string", length=50)
     */
    private $plateNumber;

    /**
     * @ORM\ManyToOne(targetEntity="LoadingType")
     * @ORM\JoinTable(name="trucks_loading_types",
     *
     * joinColumns={@ORM\JoinColumn(name="truck_id",
     * referencedColumnName="id")},
     *
     *
     * inverseJoinColumns={@ORM\JoinColumn(name="loading_type_
     * id", referencedColumnName="id")}
     *
     * )
     */
    private $loadingTypes;

    /**
     * @ORM\ManyToOne(targetEntity="TruckSubType")
     * @ORM\JoinTable(name="trucks_truck_sub_types",
     *
     *
     * joinColumns={@ORM\JoinColumn(name="truck_id",
     * referencedColumnName="id")},
     *
     *
     * inverseJoinColumns={@ORM\JoinColumn(name="truck_sub_typ
     * e_id", referencedColumnName="id")}
     *
     * )
     */
    private $transportTypes;

```

```

pository")
 * @ORM\Table(name="truck_types")
 */
class TruckType
{
    /**
     * @ORM\Id()
     * @ORM\GeneratedValue()
     * @ORM\Column(type="integer")
     */
    private $id;

    /**
     * @ORM\Column(type="string", length=100)
     */
    private $name;

    /**
     * @ORM\OneToMany(targetEntity="TruckSubType",
     * mappedBy="parent")
     */
    private $subtypes;

    public function __construct()
    {
        $this->subtypes = new ArrayCollection();
    }

    public function getId(): ?int
    {
        return $this->id;
    }

    public function getName(): ?string
    {
        return $this->name;
    }

    public function setName(string $name): self
    {
        $this->name = $name;

        return $this;
    }

    /**
     * @return Collection|TruckSubType[]
     */
    public function getSubtypes(): Collection
    {
        return $this->subtypes;
    }

    public function addSubtype(TruckSubType $subtype): self
    {
        if (!$this->subtypes->contains($subtype)) {
            $this->subtypes[] = $subtype;
            $subtype->setParent($this);
        }

        return $this;
    }

    public function removeSubtype(TruckSubType
    $subtype): self
    {
        if ($this->subtypes->contains($subtype)) {
            $this->subtypes->removeElement($subtype);
            // set the owning side to null (unless
            already changed)
            if ($subtype->getParent() === $this) {
                $subtype->setParent(null);
            }
        }

        return $this;
    }
}

```

Лістинг TruckSubType.php

```

<?php
namespace App\Entity;

use Doctrine\ORM\Mapping as ORM;

/**
 *

```

```

/**
 * @ORM\Column(type="string", length=100)
 */
private $permissibleDimensionsLoad;

/**
 * @ORM\Column(type="float")
 */
private $maxAmountCargo;

/**
 * @ORM\Column(type="string")
 */
private $photo;

/**
 * @ORM\ManyToOne(targetEntity="User")
 * @ORM\JoinColumn(name="user_id",
referencedColumnName="id", onDelete="CASCADE")
 */
private $user;

public function __construct()
{
    $this->loadingTypes = new ArrayCollection();
    $this->transportTypes = new ArrayCollection();
}

public function getId(): ?int
{
    return $this->id;
}

public function getBrand(): ?string
{
    return $this->brand;
}

public function setBrand(string $brand): self
{
    $this->brand = $brand;

    return $this;
}

public function getModel(): ?string
{
    return $this->model;
}

public function setModel(string $model): self
{
    $this->model = $model;

    return $this;
}

public function getTruckRelease(): ?int
{
    return $this->truckRelease;
}

```

```

@ORM\Entity(repositoryClass="App\Repository\TruckSubType
eRepository")
 * @ORM\Table(name="truck_subtypes")
 */
class TruckSubType
{
    /**
     * @ORM\Id()
     * @ORM\GeneratedValue()
     * @ORM\Column(type="integer")
     */
    private $id;

    /**
     * @ORM\Column(type="string", length=100)
     */
    private $name;

    /**
     * @ORM\ManyToOne(targetEntity="TruckType",
inversedBy="subtypes")
     * @ORM\JoinColumn(name="parent_id",
referencedColumnName="id", onDelete="CASCADE")
     */
    private $parent;

    public function getId(): ?int
    {
        return $this->id;
    }

    public function getName(): ?string
    {
        return $this->name;
    }

    public function setName(string $name): self
    {
        $this->name = $name;

        return $this;
    }

    public function getParent(): ?TruckType
    {
        return $this->parent;
    }

    public function setParent(?TruckType $parent): self
    {
        $this->parent = $parent;

        return $this;
    }
}

```

Лістинг UserEntityListener.php

```

<?php
namespace App\EventListener\DoctrineEntityListener;

use App\Entity\User;
use Doctrine\ORM\Event\PreUpdateEventArgs;
use Doctrine\ORM\Mapping as ORM;
use Doctrine\Common\Persistence\Event\LifecycleEventArgs;
use Symfony\Component\Security\Core\Encoder\UserPasswordEncoderInterface;

class UserEntityListener
{
    /**
     * @var UserPasswordEncoderInterface
     */
    private $encoder;

    /**
     * @param UserPasswordEncoderInterface $encoder
     */
    public function
__construct(UserPasswordEncoderInterface $encoder)
    {
        $this->encoder = $encoder;
    }

    /**
     * Encode password when persisting new user.

```

```

    public function setTruckRelease(int $truckRelease):
self
    {
        $this->truckRelease = $truckRelease;

        return $this;
    }

    public function getPLateNumber(): ?string
    {
        return $this->plateNumber;
    }

    public function setPlateNumber(string $plateNumber):
self
    {
        $this->plateNumber = $plateNumber;

        return $this;
    }

    public function getPermissibleDimensionsLoad():
?string
    {
        return $this->permissibleDimensionsLoad;
    }

    public function setPermissibleDimensionsLoad(string
$permissibleDimensionsLoad): self
    {
        $this->permissibleDimensionsLoad =
$permissibleDimensionsLoad;

        return $this;
    }

    public function getMaxAmountCargo(): ?float
    {
        return $this->maxAmountCargo;
    }

    public function setMaxAmountCargo(float
$maxAmountCargo): self
    {
        $this->maxAmountCargo = $maxAmountCargo;

        return $this;
    }

    public function getPhoto(): ?string
    {
        return $this->photo;
    }

    public function setPhoto(string $photo): self
    {
        $this->photo = $photo;

        return $this;
    }

```

```

    *
    * @param User $user
    * @param LifecycleEventArgs $args
    *
    * @ORM\PrePersist
    */
    public function encodePasswordOnPrePersist(User
$user, LifecycleEventArgs $args)
    {
        $this->encodePassword($user);
    }

    /**
     * Encode new password when updating a user.
     *
     * @param User $user
     * @param PreUpdateEventArgs $args
     *
     * @ORM\PreUpdate
     */
    public function encodePasswordOnPreUpdate(User
$user, LifecycleEventArgs $args)
    {
        $this->encodePassword($user);

        $om = $args->getEntityManager();
        $uow = $om->getUnitOfWork();
        $meta = $om->getClassMetadata(get_class($user));

        $uow->recomputeSingleEntityChangeSet($meta,
$user);
    }

    /**
     * Encode the value of plainPassword property to
password property.
     *
     * @param User $user
     */
    private function encodePassword(User $user)
    {
        if (null === $plainPassword =
$user->getPlainPassword()) {
            return;
        }

        $encoded = $this->encoder->encodePassword(
            $user,
            $plainPassword
        );

        $user->setPassword($encoded);
    }
}

```

Лістинг OrderRequestEntityListener.php

```

<?php

namespace App\EventListener\DoctrineEntityListener;

use App\Entity\OrderRequest;
use Doctrine\ORM\Mapping as ORM;
use Doctrine\Common\Persistence\Event\LifecycleEventArgs;
use Mgiilet\NotificationBundle\Manager\NotificationManager;

class OrderRequestEntityListener
{
    private $notificationManager;

    public function __construct(NotificationManager
$notificationManager)
    {
        $this->notificationManager =
$notificationManager;
    }

    /**
     * @ORM\PostPersist()
     */
    public function sendNotification(OrderRequest
$orderRequest, LifecycleEventArgs $args)
    {
        $notify =
$this->notificationManager->createNotification('New

```

```

/**
 * @return Collection|LoadingType[]
 */
public function getLoadingTypes(): Collection
{
    return $this->loadingTypes;
}

public function addLoadingType(LoadingType
$loadingType): self
{
    if
(!$this->loadingTypes->contains($loadingType)) {
        $this->loadingTypes[] = $loadingType;
    }

    return $this;
}

public function removeLoadingType(LoadingType
$loadingType): self
{
    if ($this->loadingTypes->contains($loadingType))
    {
        $this->loadingTypes->removeElement($loadingType);
    }

    return $this;
}

/**
 * @return Collection|TruckSubType[]
 */
public function getTransportTypes(): Collection
{
    return $this->transportTypes;
}

public function addTransportType(TruckSubType
$transportType): self
{
    if
(!$this->transportTypes->contains($transportType)) {
        $this->transportTypes[] = $transportType;
    }

    return $this;
}

public function removeTransportType(TruckSubType
$transportType): self
{
    if
($this->transportTypes->contains($transportType)) {
        $this->transportTypes->removeElement($transportType);
    }

    return $this;
}

```

```

order request:
'. $orderRequest->getOrder()->getCargoName());

$notify->setMessage('OrderRequest:'. $orderRequest->getI
d());

$this->notificationManager->addNotification([$orderRequ
est->getOrder()->getAuthor()], $notify, true);
}
}

```

Лістинг LoginFormAuthenticator.php

```

<?php
namespace App\Security;

use App\Entity\User;
use Doctrine\ORM\EntityManagerInterface;
use Symfony\Component\HttpFoundation\RedirectResponse;
use Symfony\Component\HttpFoundation\Request;
use
Symfony\Component\Routing\Generator\UrlGeneratorInterfa
ce;
use
Symfony\Component\Security\Core\Authentication\Token\To
kenInterface;
use
Symfony\Component\Security\Core\Encoder\UserPasswordEnc
oderInterface;
use
Symfony\Component\Security\Core\Exception\CustomUserMes
sageAuthenticationException;
use
Symfony\Component\Security\Core\Exception\InvalidCsrfTo
kenException;
use
Symfony\Component\Security\Core\Security;
use
Symfony\Component\Security\Core\User\UserInterface;
use
Symfony\Component\Security\Core\User\UserProviderInterf
ace;
use
Symfony\Component\Security\Csrf\CsrfToken;
use
Symfony\Component\Security\Csrf\CsrfTokenManagerInterface;
use
Symfony\Component\Security\Guard\Authenticator\Abstract
LoginFormAuthenticator;
use
Symfony\Component\Security\Http\Util\TargetPathTrait;

class LoginFormAuthenticator extends
AbstractLoginFormAuthenticator
{
    use TargetPathTrait;

    private $entityManager;
    private $urlGenerator;
    private $csrfTokenManager;
    private $userPasswordEncoder;

    public function __construct(EntityManagerInterface
$entityManager, UrlGeneratorInterface $urlGenerator,
CsrfTokenManagerInterface $csrfTokenManager,
UserPasswordEncoderInterface $userPasswordEncoder)
    {
        $this->entityManager = $entityManager;
        $this->urlGenerator = $urlGenerator;
        $this->csrfTokenManager = $csrfTokenManager;
        $this->userPasswordEncoder =
$userPasswordEncoder;
    }

    public function supports(Request $request)
    {
        return 'app_login' ===
$request->attributes->get('_route')
        && $request->isMethod('POST');
    }

    public function getCredentials(Request $request)
    {
        $credentials = [
            'email' => $request->request->get('email'),
            'password' =>
$request->request->get('password'),
            'csrf_token' =>
$request->request->get('_csrf_token'),

```

```

    }

    public function getUser(): ?User
    {
        return $this->user;
    }

    public function setUser(?User $user): self
    {
        $this->user = $user;

        return $this;
    }
}

```

```

    ];
    $request->getSession()->set(
        Security::LAST_USERNAME,
        $credentials['email']
    );

    return $credentials;
}

public function getUser($credentials,
    UserProviderInterface $userProvider)
{
    $token = new CsrfToken('authenticate',
        $credentials['csrf_token']);
    if
    (!$this->csrfTokenManager->isTokenValid($token)) {
        throw new InvalidCsrfTokenException();
    }

    $user =
    $this->entityManager->getRepository(User::class)->findO
    neBy(['email' => $credentials['email']]);
    if (!$user) {
        throw new
        CustomUserMessageAuthenticationException('Email could
        not be found.');
```

```

    }

    return $user;
}

public function checkCredentials($credentials,
    UserInterface $user)
{
    return
    $this->userPasswordEncoder->isPasswordValid($user,
        $credentials['password']);
}

public function onAuthenticationSuccess(Request
    $request, TokenInterface $token, $providerKey)
{
    return new
    RedirectResponse($this->urlGenerator->generate('home'))
;
}

protected function getLoginUrl()
{
    return
    $this->urlGenerator->generate('app_login');
}
}

```

М.С. ГРИЦЮК, О.А. ПАСІЧНИК, Т.К. СКРИПНИК
Хмельницький національний університет

ІНФОРМАЦІЙНА СИСТЕМА ПЛАНУВАННЯ НАЙКРАЩОГО ШЛЯХУ ДЛЯ ДОСТАВКИ ВАНТАЖУ ЗА ДОПОМОГОЮ ЗАДАЧІ КОМІВОЯЖЕРА

Розроблена інформаційна система планування найкращого шляху для доставки вантажу за допомогою задачі комівояжера. Даний сервіс забезпечує прокладання маршруту у відповідності з початковими даними (початкова та кінцева точка) та обмеженнями (проміжні точки маршруту, часові обмеження, масо-габаритні тощо). Інформаційна система також містить вбудований калькулятор вартості перевезень, який враховує довжину маршруту та вагу вантажу.

Ключові слова: вантажні перевезення, далекобійники, вантаж, задача комівояжера, оптимальний маршрут.

M.S. HRYTSYUK, O.A. PASICHNYK, T.K. SKRYPNYK
Khmelnitskyi National University

PLANNING INFORMATION SYSTEM FOR THE BEST WAY TO DELIVER CARGO WITH THE HELP OF THE TRAVELLING SALESMAN PROBLEM SOLVER

The Internet has made it possible to transfer most areas of activity online, as well as solve various types of problems without leaving home. As a result, many innovative projects have emerged that have simplified and improved routines. Freight transportation in Ukraine occupies the largest share among all types of transport services. The minimum part of the carriers involved in this process works on a contractual basis with large enterprises-customers. Other market participants interact with customers without signing long-term cooperation agreements. This option of bilateral partnership is extremely risky because the search for a customer or carrier on the bulletin boards can end in a meeting with fraudsters. As a result, the transport organization is left without earnings, and the customer loses the cargo. Many drivers working in trucking companies are thinking about starting a business in this field, but in order to be independent of the transport company, it is not enough to own a truck, as one of the main services provided by transport companies in the search. orders. Given that the orders placed by companies do not always meet the expectations of drivers, and the place of loading can be in the tens or even hundreds of kilometres from the location of the driver. Not all companies reimburse the fuel used to get to the place of loading. As a result, drivers are required to pay for the services of the operator, as well as to compensate out of pocket to get to the place of loading, which significantly reduces their income. Therefore, an information system was developed to plan the best way to deliver the goods using the task of a salesman. This service provides routing in accordance with the initial data (start and end point) and restrictions (intermediate points of the route, time restrictions, mass and size, etc.). The information system also contains a built-in cost calculator that takes into account the length of the route and the weight of the cargo.

Keywords: freight transportation, truckers, cargo, the traveller's task, optimal route.

Аналіз предметної області

Кожен великий бізнес, сфера діяльності якого тісно пов'язана з великими грошима, будь-то виробництво, оптова чи широка роздрібна мережа, будує свій бізнес на правильно сформованому вантажоперевезенню. Вчасно доставлений вантаж підтримує в стабільності весь процес бізнесу. Якщо на виробництво не буде вчасно доставлений хоч один компонент все зупиниться і почнуться збитки. У тій же оптовій або роздрібній мережі відсутність певного товару вказує дорогу до конкурентів. Клієнтам не важливо з якої причини є необхідних товарів, їм важливо володіти ними.

Сфера вантажоперевезень активно розвивається, завдяки тому, що через Україну проходять великі транспортні шляхи, дедалі більше людей розпочинають свою кар'єру саме в цій сфері. Завдяки сучасним технологіям, вантажоперевезення стає максимально ефективним і прибутковим. Подання асортименту в Інтернеті є перспективним напрямком розвитку бізнесу. Невеликого стартового капіталу цілком достатньо, щоб відкрити інтернет-магазин, який стане основою для подальшого розширення компанії та досягнення рівня роздрібною мережі.

Веб-додатки, які вирішують проблеми вантажоперевезень, стають дедалі популярними серед перевізників.

Автомобільні перевезення широко використовується у всіх галузях економіки: в торгівлі, виробництві, сільському господарстві.

Аналіз існуючого програмного забезпечення предметної області

Сьогодні розвиток технології дозволяє автоматизувати ті процеси, які ще декілька років тому не могли б існувати без підтримки людини. Доволі часто ми чуємо про нові сервіси, які вдосконалюють ті процеси, які людина не здатна вирішувати на високому рівні. Наприклад, всім знайомий сервіс для виклику авто (таксі) Uklon – це досить хороший приклад сервісу, який вирішує проблеми, будівництва найкращого маршруту для водіїв таксі, за допомогою сучасних технологій, що, в свою чергу, дозволяє скоротити витрати на паливо, задовольнити потреби клієнта, а також дає змогу виконати більше замовлень за один і той же час.

Сфера вантажоперевезень не виключення, завдяки доступності мережі Інтернет, з'являється можливість до реалізації нових ідей, які змінюють всі процеси, які були до цього запроваджені в тій чи

іншій галузі.

Одним з прикладів сервісів в сфері вантажоперевезень є “Lardi Trans”[1] – це сервіс, який спеціалізується на міжнародних вантажоперевезеннях (рис. 1)

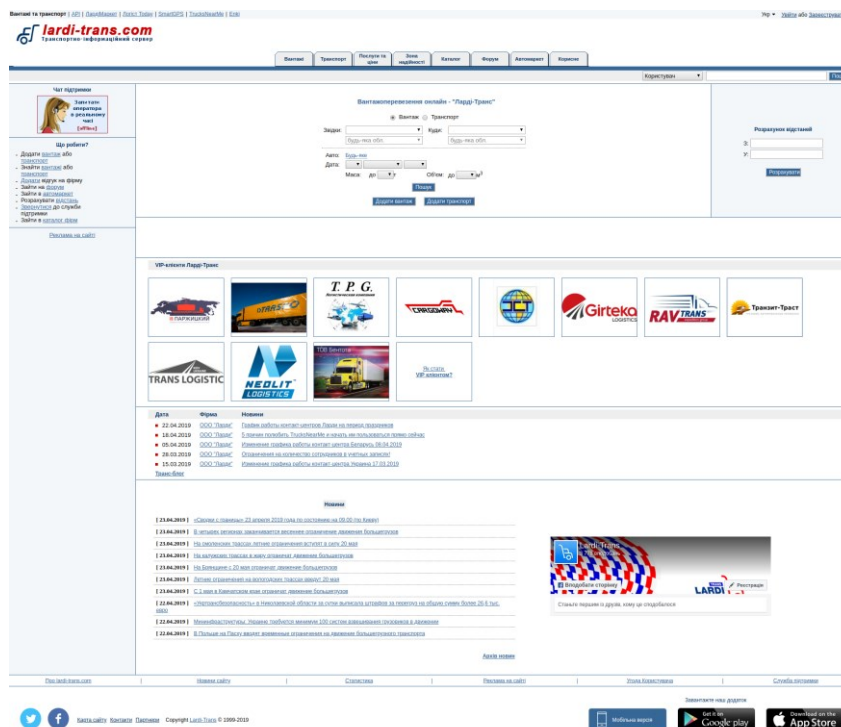


Рис. 1. Даталогічна модель БД

Наявність певних недоліків в існуючому програмному забезпеченні для вирішення задачі планування найкращого шляху доставки вантажу за допомогою задачі комівояжера в межах інформаційної системи потребує реалізації такого функціоналу: впровадити зручний і сучасний інтерфейс користувача, пошук маршруту вказуючи початкову, проміжні і кінцеві точки, що в свою чергу дозволить спланувати рейс для водія.

Аналіз та автоматизація обробки інформаційних потоків

Розробка нових інформаційних технологій дозволяє розвивати суспільство в тих чи інших напрямках. Вони активно перетворюють інші технології матеріального і нематеріального виробництва, в кінцевому підсумку формуючи новий стиль роботи, спосіб життя в цілому. Суть інформаційних технологій становлять методи і засоби формування та підтримки інформаційних потоків у системах управління об'єктами.

Якщо розглядати роботу звичайної транспортної компанії, то можна побачити, як неефективно розподіляються ресурси, дана галузь створює багато непотрібних, суміжних процесів, які сьогодні можна автоматизувати: обробка замовлень, планування маршруту, виконання замовлень.

Бізнес-процес «Обробка замовлень»

Даний бізнес-процес передбачає опрацювання інформації, що стосується замовлень, й включає в себе наступні функції: збір інформації по вантажу, розрахунок актуальної вартості доставки, прийняття пропозиції від водіїв, оформлення замовлення;

Бізнес-процес «Планування маршруту»

Даний процес потребує багато зусиль, для того, щоб спланувати максимально ефективний маршрут, який дозволить збільшити прибутки компанії, й, зокрема, задовольнити відправника вчасним доставленим вантажем.

Планування маршруту передбачає опрацювання інформації, що стосується, починаючи від стану доріг, закінчуючи відстанню між пунктом відправки й пунктом призначення, даний процес включає в себе наступні дії: розрахунок відстані між точкою А та точкою Б, дослідження можливих маршрутів, врахування різних факторів, які впливають на тривалість доставки вантажу, вибір найкращого маршруту.

Бізнес-процес «Виконання замовлень»

При роботі з замовленням водій, водій отримує можливість напряму спілкуватися з клієнтом, через влаштований месенджер, що дозволяє отримати максимально точну інформацію щодо вантажу, в свою чергу це дозволяє водію спланувати свій маршрут максимально ефективно, а саме взяти попутні замовлення, якщо це дозволяє параметри його автомобіля. Виконання замовлення включає в себе наступні функції: перегляд даних замовлення, будівництво найкращого маршруту, виконання попутних замовлень, відмова від

замовлення, керування рейтингом водія.

Для того, щоб автоматизувати роботу веб-сервісу, дані потрібно організувати у таблиці бази даних і провести нормалізацію.

Даталогічна модель БД створюється відповідно до бізнес-процесів та їх функцій (рис. 2)

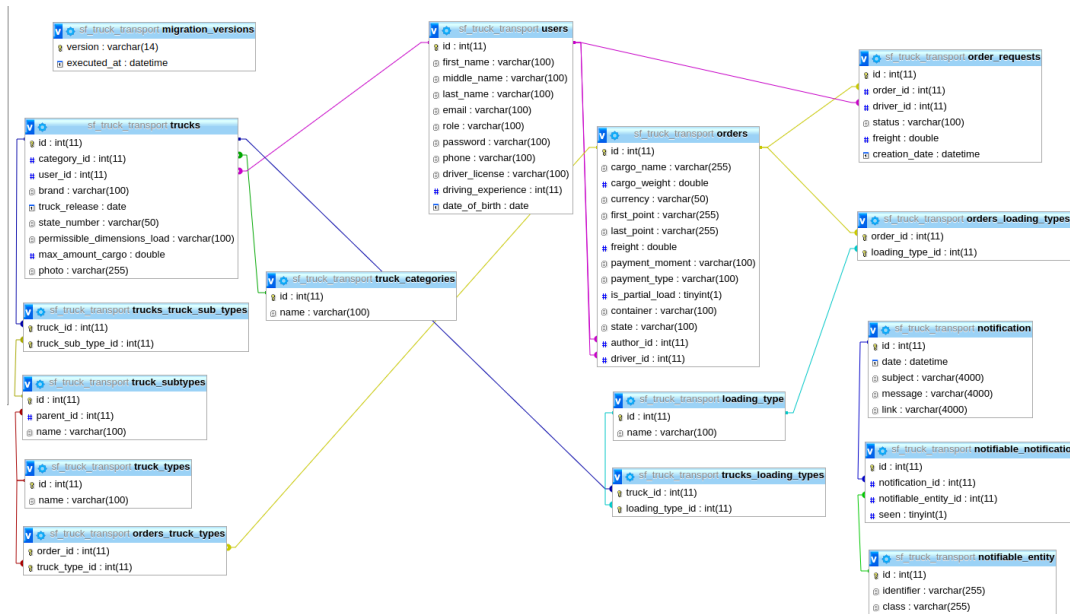


Рис. 2. Даталогічна модель БД

Розробка структури інформаційної системи

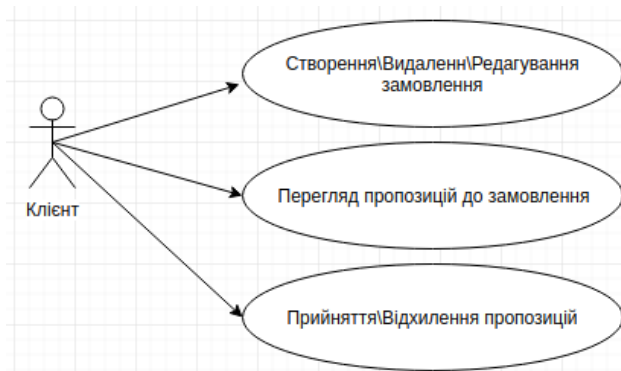


Рис. 3. Роль користувача «Клієнт»

Розроблений веб-сервіс буде корисним у використанні для клієнтів які, часто використовують послуги транспортних компаній, а також для водіїв, які мають власний вантажний автомобіль, і хочуть бути незалежні від транспортної компанії.

У сервісі буде два типи користувачів: «Клієнт» та «Водій». Для кожного з користувачів будуть доступні відповідні функції.

Користувач «Клієнт» після авторизації, буде мати змогу створити вантаж, який буде містити в собі всю необхідну інформацію, для того, щоб знайти водія з вантажівкою, котра буде відповідати вимогам транспортування

вантажу. «Клієнту» буде доступний перегляд усіх пропозицій від водіїв, які він може відхилити або прийняти. Схема доступних функцій для користувача «Клієнт» зображено на рис. 3.

Користувач «Водій» може отримати доступ до особистого кабінету після авторизації. Даному користувачу буде доступна можливість керування своїми вантажівками, пошук замовлень, і відправлення запиту на замовлення (рис. 4).

Інтерпретація алгоритму задачі комівояжера

Задача комівояжера полягає у пошуку найкращого маршруту, що проходить через обрані міста хоча б один раз. В умовах завдання вказуються критерій вигідності маршруту (найкоротший, найдешевший, сукупний критерій тощо) і відповідні матриці відстаней, вартості тощо. Зазвичай задано, що маршрут повинен проходити через кожне місто тільки один раз, в такому випадку розв'язок знаходиться серед гамільтонових циклів. Для реалізації пошуку найкращого шляху було використано метод грубої сили. Є N міст, з'єднаних між собою

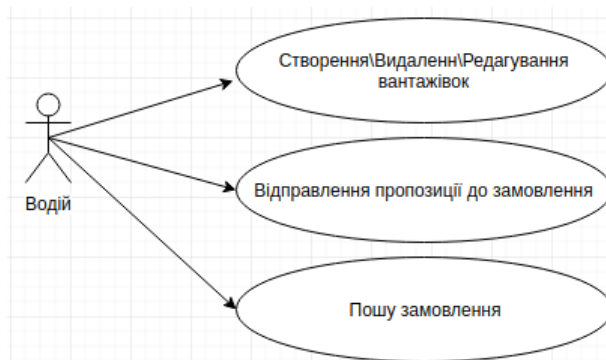


Рис. 4. Роль користувача «Водій»

дорогами. Між ними необхідно прокласти найкоротший замкнутий маршрут, що проходить через кожне місто тільки один раз. Намальований граф (рис. 5), що складається з п'яти вузлів (міст), з'єднаних ребрами (дорогами). Довжини доріг приведені поруч з ребрами. Це зв'язний граф, в якому кожен вузол з'єднаний з будь-яким іншим. Ребра графа ненаправлені і переміщення по ним від вузла до вузла можливо в будь-якому напрямку. Позначимо відстань між містами i та j через D_{ij} . Зазвичай передбачається, що $D_{ij} \geq 0$, $D_{ij} = D_{ji}$

Взагалі кажучи, D_{ij} – не обов'язково "фізична довжина" дороги. Це може бути час переміщення, вартість квитка або довільно заданий невід'ємне число. Проте, у всіх цих випадках, D_{ij} буде називатися відстанню. Запишемо відстані між містами у вигляді матриці (рис. 6). Наприклад, відстань з міста 2 в місто 3 (i з 3 в 2) дорівнює 7. Так як граф ненаправлений, ця матриця симетрична. Прочерками відзначені "заборонені" переходи з міста в нього ж.

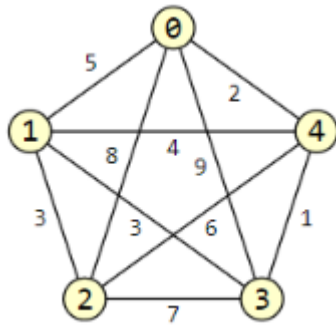


Рис. 5. Граф

	0	1	2	3	4
0	-	5	8	9	2
1	5	-	3	3	4
2	8	3	-	7	6
3	9	3	7	-	1
4	2	4	6	1	-

Рис. 6. Матриця відстані між містами

Шуканий шлях замкнутий, тому будь-яке місто можна вибрати в якості початкового (і кінцевого). Нехай таким буде нульовий місто. Тоді будь-яка перестановка чисел від 1 до 4, оточена нулями, символізує певний шлях, що проходить через кожне місто один раз. Наприклад, 0,1,3,2,4,0, означає, що стартуючи з міста 0, ми подорожуємо у місто 1, потім в місто 3 і т.д. На останньому кроці з міста 4 повертаємося в стартову точку – місто 0. Довжина цього шляху дорівнює $5 + 3 + 7 + 6 + 2 = 23$ і він не найкоротший. В даному випадку є дві перестановки довжиною 17. Це 0,2,1,3,4,0 і 0,4,3,1,2,0. Насправді це один і той же шлях, який проходить в "прямому" або "зворотному" напрямку. Подібні перестановки, що відрізняються зверненням шляху, вважаються тотожними. Надалі фінальне місто буде опускатися і мається на увазі, що він завжди збігається зі стартовим. Таким чином, на наведеному вище графі існує єдине оптимальне рішення 0,2,1,3,4 з довжиною шляху, рівній 17.

Нехай один з N міст фіксований. Тоді решта $N-1$ міст можна переставити $(N-1)!$ способами. Половина з них є зверненням шляху (циклічної перестановки). Тому існує $(N-1)! / 2$ варіантів різних шляхів, серед яких необхідно знайти хоча б один шлях мінімальної довжини.

Калькулятор вартості перевезень

Для розрахунку вартості перевезення використовуються наступні вхідні дані: місце відправки вантажу, місце прибуття вантажу, вага вантажу. Процес перевезення вантажів складається з трьох етапів: початкового, рухомого та кінцевого. З урахуванням типовості початкового та кінцевого етапів виокремлюють витрати на здійснення початково-кінцевих та рухомих операцій. До початково-кінцевих належать витрати на утримання рухомого складу на стоянках, на його підготовку та завантаження, на різні маневрові роботи, тобто на всі роботи, не пов'язані з рухом і відстанню перевезень. Рухомі операції передбачають витрати на утримання рухомого складу на шляху пересування вантажу.

Загалом собівартість транспортного тарифу розраховується за формулою: $Стт = Вп.к + ВрВ$, де $Вп.к$ – витрати на початково-кінцеві операції, гр. од.; $Вр$ – рухомі (шляхові) витрати на 1 км відстані, гр. од.; $В$ – відстань перевезення вантажів, км.

Використовуючи цю формулу, розраховують собівартість перевезення на 1 км вантажу масою 1 т. Зі збільшенням відстані перевезення вантажів собівартість перевезення 1 т/км зменшується, тому що зменшуються відносні витрати на початково-кінцеві операції з переміщення вантажів на 1 км.

Даний функціонал, який дозволить розраховувати вартість вантажу буде корисним не лише для користувачів, які відправляють вантаж, а й для самих водіїв початківців, які, в свою чергу, можуть цей калькулятор використовувати для формування вартості 1 км.

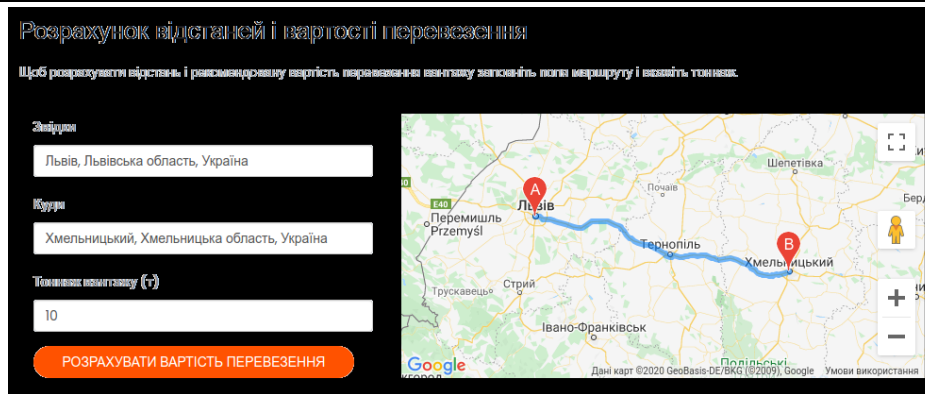


Рис. 7. Інтерфейс калькулятора розрахунку вартості перевезення вантажу

Програмна реалізація інформаційної системи

Реалізована інформаційна система для розв’язання задачі планування найкращого шляху для доставки вантажу за допомогою задачі комівояжера забезпечує прокладання маршруту у відповідності з початковими даними та обмеженнями (рис. 8, рис. 9), а також забезпечує калькуляцію вартості перевезень (рис. 10).

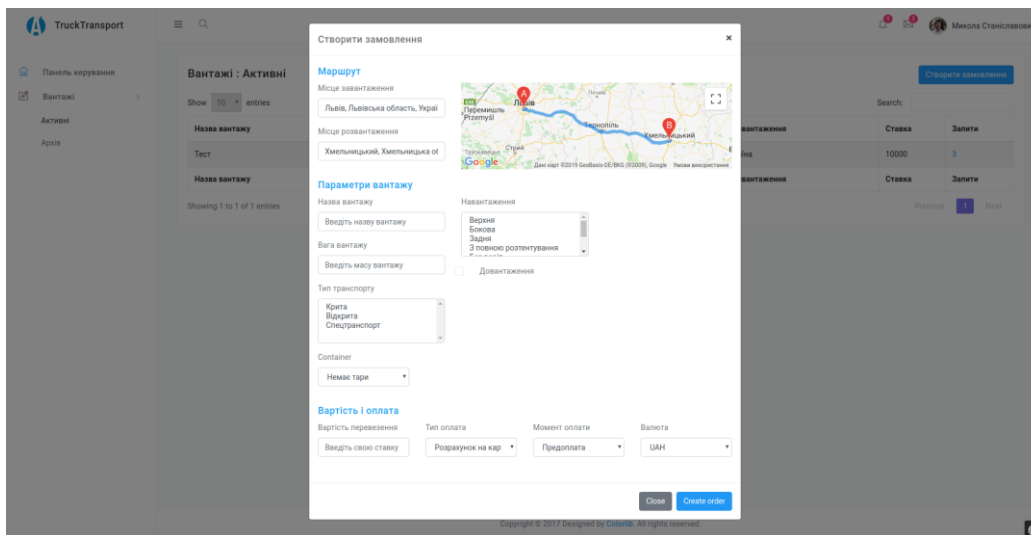


Рис. 8. Інтерфейс створення замовлення

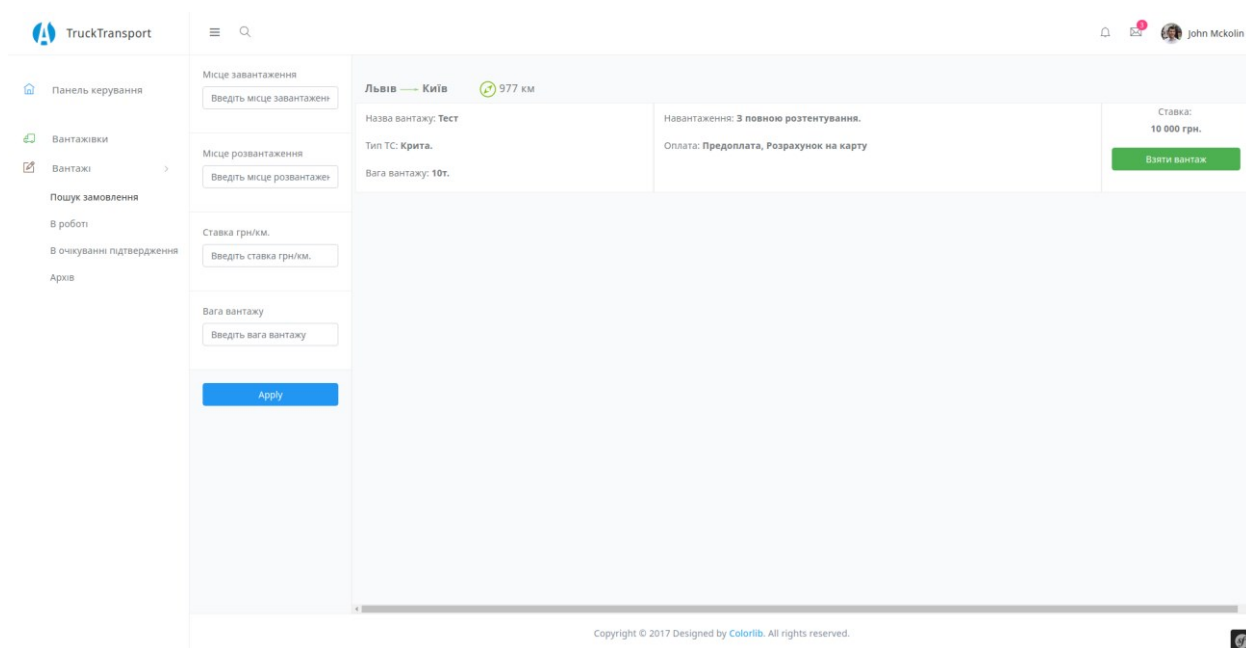


Рис. 9. Інтерфейс пошуку замовлення

Розрахунок відстаней і вартості перевезення

Щоб розрахувати відстань і рекомендовану вартість перевезення вантажу заповніть поля маршруту і вкажіть тоннаж.

Звідки

Куди

Тоннаж вантажу (т)

РОЗРАХУВАТИ ВАРТІСТЬ ПЕРЕВЕЗЕННЯ

Рис. 10. Інтерфейс пошуку замовлення

Для реалізації було вибрано мову програмування PHP. Завдяки своїй популярності, PHP в арсеналі має достатньо хороших фреймворків, які дозволяють ефективно, швидко і надійно розробляти веб-додатки. Для вирішення своїх потреб, ми обрали фреймворк Symfony 4 – один з найпопулярніших фреймворків, який славиться своєю архітектурою. Доволі багато фреймворків, CMS і E-commerce базують своє ядро на Symfony. Для організації бази даних було обрано систему керування базами даних MySQL.

Висновки

Реалізована інформаційна система для вирішення задачі планування найкращого шляху для доставки вантажу за допомогою задачі комівояжера. Програмний застосунок забезпечує прокладання маршруту у відповідності з початковими даними (початкова та кінцева точка) та обмеженнями (проміжні точки маршруту, часові обмеження, масо-габаритні тощо). Інформаційна система містить вбудований калькулятор вартості перевезень, який враховує довжину маршруту та вагу вантажу.

Література

1. Lardi Trans [Електронний ресурс]. – Режим доступу : <https://lardi-trans.com/en/>
2. Applegate D.L., Bixby R.E., Chvátal V. & Cook W.J. (2007). The Traveling Salesman Problem: A Computational Study. Princeton University Press.
3. Левитин А. В. Алгоритмы: введение в разработку и анализ. Глава 3. Метод грубой силы: Задача коммивояжера / Ананий В. Левитин. – М. : Вильямс, 2006. – С. 159–160. – ISBN 0-201-74395-7.
4. Bernhard Korte, Jens Vygen (2006). Combinatorial Optimization. 3d ed. Springer. ISBN 3-540-25684-9.
5. Reinelt, Gerhard (1992). Fast heuristics for large geometric traveling salesman problems. ORSA Journal on computing, 4:206-217
6. Ковалев В.П. Эффективность грузовых автомобильных перевозок: Состояние, проблемы, перспективы / Ковалев В.П. – Минск : Беларусь, 1984. – 112 с.
7. Просветов Г.И. Математические методы в логистике / Просветов Г.И. – М. : РДЛ, 2006. – 272 с.
8. Меламед И.И. Задача коммивояжера. Вопросы теории / И.И. Меламед, С.И. Сергеев, И.Х. Сигал // Автоматика и телемеханика. – 1989. – № 9. – С. 3–33.
9. Chisman J.A. The clustered traveling salesman problem / J.A. Chisman // Computers & Operations Research. — September 1975. – Volume 2. – Issue 2. – P. 115–119.

References

1. Lardi Trans. URL: <https://lardi-trans.com/en/>
2. Applegate D.L., Bixby R.E., Chvatal V. & Cook W.J. (2007). The Traveling Salesman Problem: A Computational Study. Princeton University Press.
3. Levitin A. V. Algoritmy: vvedenie v razrabotku i analiz. Glava 3. Metod gruboj sily: Zadacha kommivoyazhera / Ananij V. Levitin. – M. : Vilyams, 2006. – S. 159–160. – ISBN 0-201-74395-7.
4. Bernhard Korte, Jens Vygen (2006). Combinatorial Optimization. 3d ed. Springer. ISBN 3-540-25684-9.
5. Reinelt, Gerhard (1992). Fast heuristics for large geometric traveling salesman problems. ORSA Journal on computing, 4:206-217
6. Kovalev V.P. Effektivnost' gruzovyh avtomobilnyh perezovok: Sostoyanie, problemy, perspektivy / Kovalev V.P. – Minsk : Belarus, 1984. – 112 s.
7. Prosvetov G.I. Matematicheskie metody v logistike / Prosvetov G.I. – M. : RDL, 2006. – 272 s.
8. Melamed I.I. Zadacha kommivoyazhera. Voprosy teorii / I.I. Melamed, S.I. Sergeev, I.H. Sigal // Avtomatika i telemehaniка. – 1989. – № 9. – S. 3–33.
9. Chisman J.A. The clustered traveling salesman problem / J.A. Chisman // Computers & Operations Research. — September 1975. – Volume 2. – Issue 2. – P. 115–119.

Рецензія/Peer review : 19.10.2020 р.

Надрукована/Printed : 06.11.2020 р.

Інформаційна система планування найкращого шляху для доставки вантажу за допомогою задачі комівояжера

Студент: Грицюк М.С.
Керівник: Пасічник О.А.



Актуальність

Глобалізація економіки супроводжується небувалими раніше темпами зростання торгівлі. Світовий обсяг експорту за 50 років виріс в 10 разів і продовжує збільшуватися більш високими темпами, ніж ВВП. Щоденний обсяг валютних операцій перевищує \$ 1,5 трлн проти \$ 15 млрд в 1973 році. У цих умовах максимально зростає значення світової транспортної мережі. Транспорт служить матеріальною базою виробничих зв'язків між окремими територіями, виступає як фактор, що організує світовий економічний простір і забезпечує подальше географічне розподіл праці.

Мета і завдання

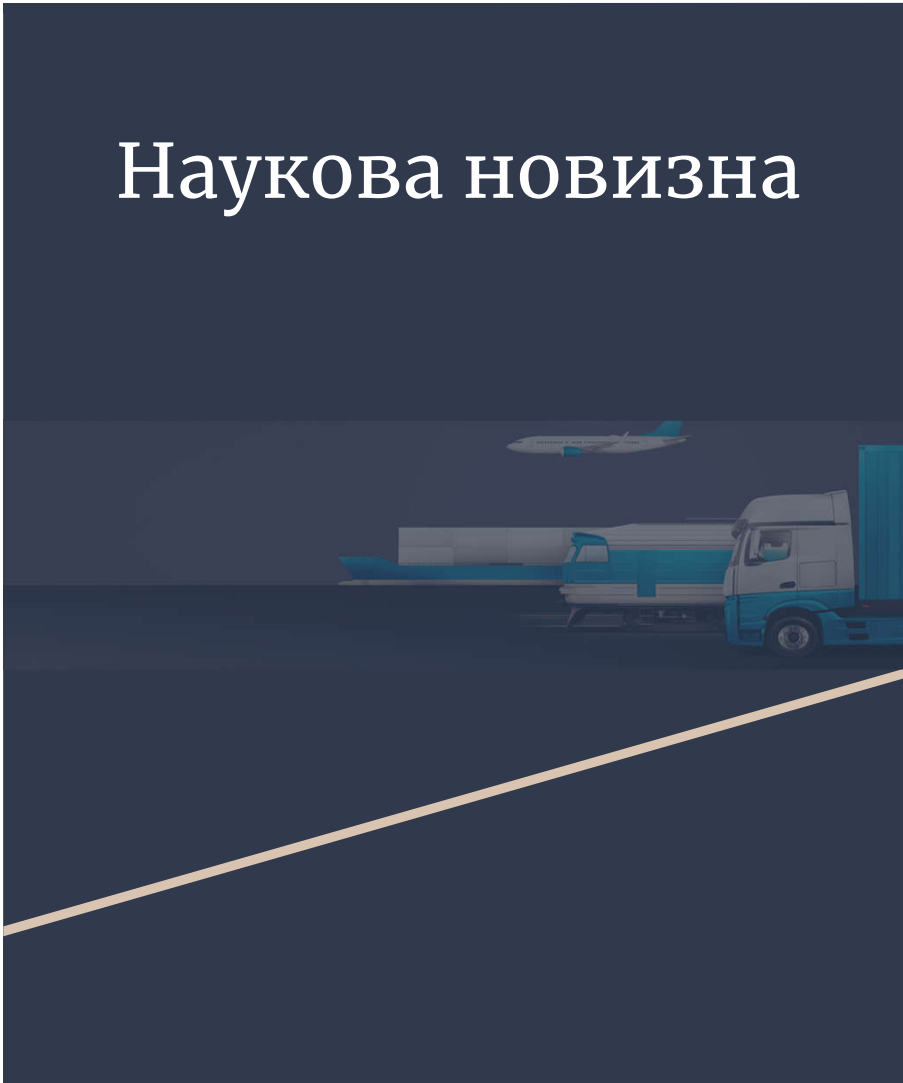
Мета роботи – полягає у реалізації інформаційної системи для планування найкращого шляху доставки вантажу за допомогою задачі комівояжера. Для досягнення мети необхідно дослідити існуючі підходи оптимізації і пошуку найкращого шляху, а також створити відповідну інформаційну систему й дослідити її ефективність.

Основним завданням було розробити інформаційну систему побудови найкращого маршруту вантажних перевезень, яка поєднує розв'язки задачі комівояжера з сучасними картографічними сервісами.



Наукова новизна

В результаті проведеної роботи було вдосконалено методики використання задачі комівояжера для пошуку найкращого маршруту у напрямку їх поєднання з сучасними картографічними сервісами.



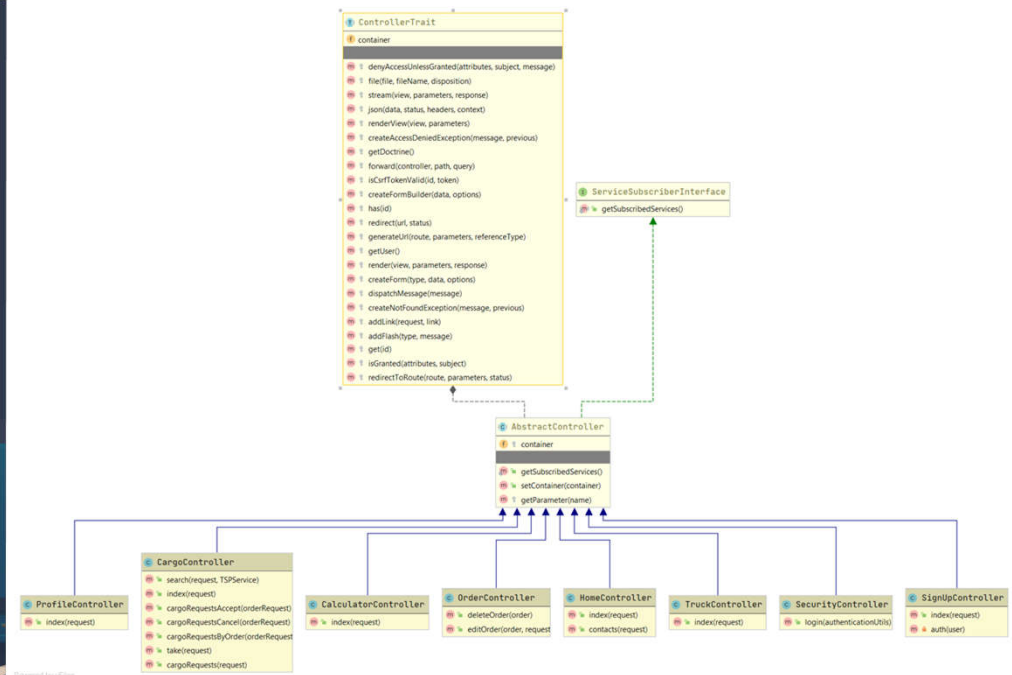
Практична цінність



Сьогодні розвиток технології дозволяє спростити та автоматизувати ті процеси, які ще декілька років тому не могли б існувати без підтримки людини. Доволі часто ми чуємо про нові сервіси, які вдосконалюють ті процеси, які людина не здатна вирішувати на високому рівні. Наприклад всім знайомий сервіс для виклику авто (таксі) Uklon, це досить хороший приклад сервісу, який вирішує проблеми, будування найкращого маршруту для водіїв таксі, за допомогою сучасних технологій, що в свою чергу дозволяє скоротити витрати на пальноє, задовольнити потреби клієнта, а також дає змогу виконати більше замовлень за один і той же час.

Сфера вантажоперевезень не виключення, завдяки вседоступності до мережі Інтернет, з'являється можливість до реалізації нових ідей, які змінюють всі процеси, які були до цього запроваджені в тій чи іншій галузі.

Діаграма класів (контролерів)

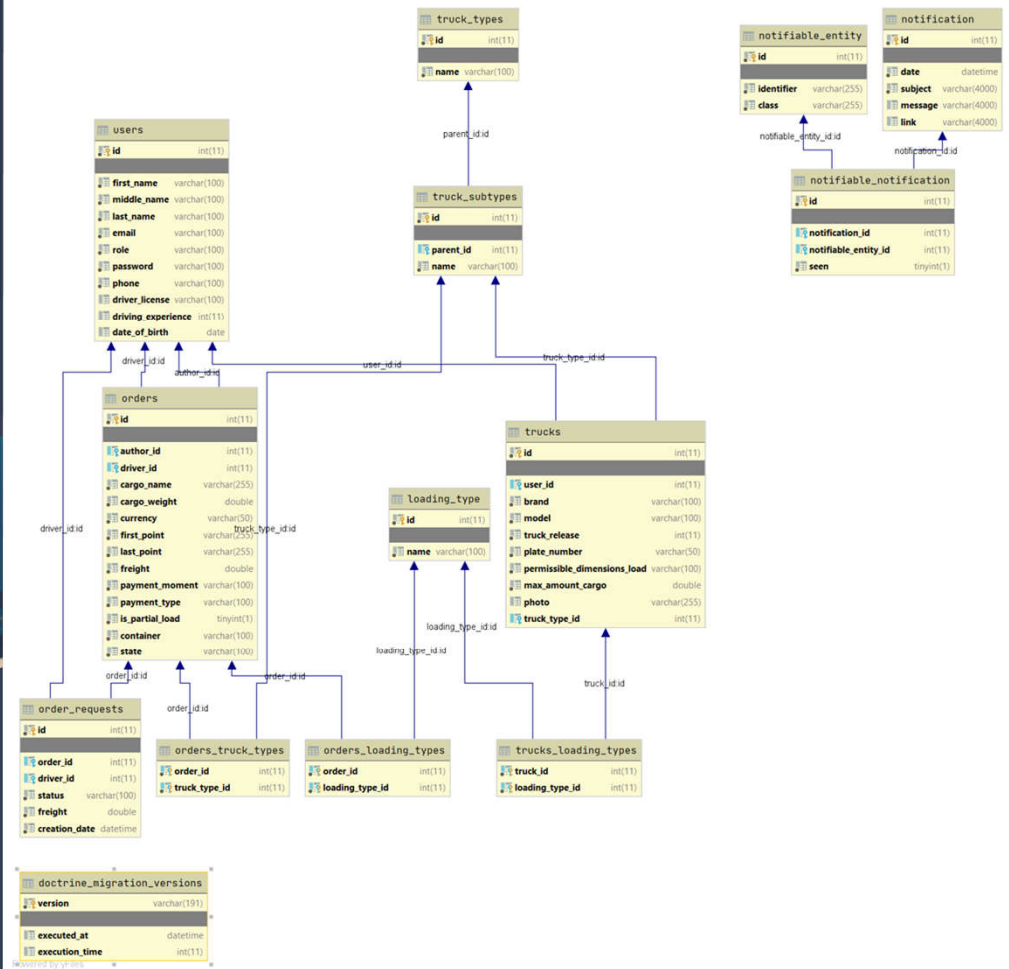


Діаграма класів (сутностей)

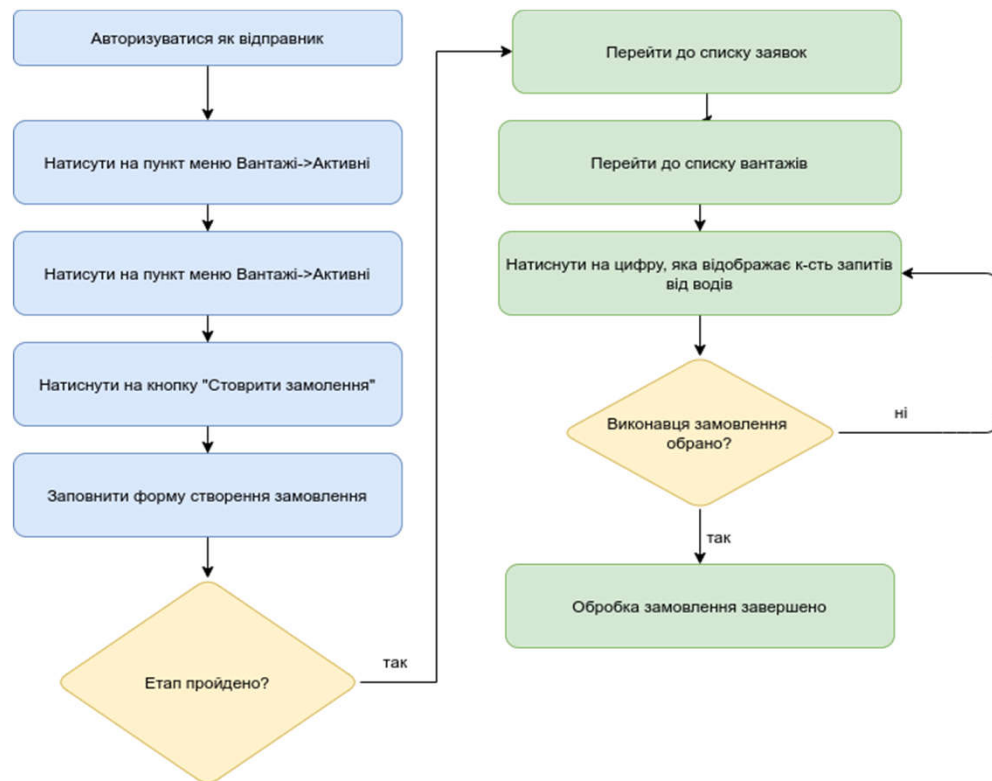


Powered by ynote

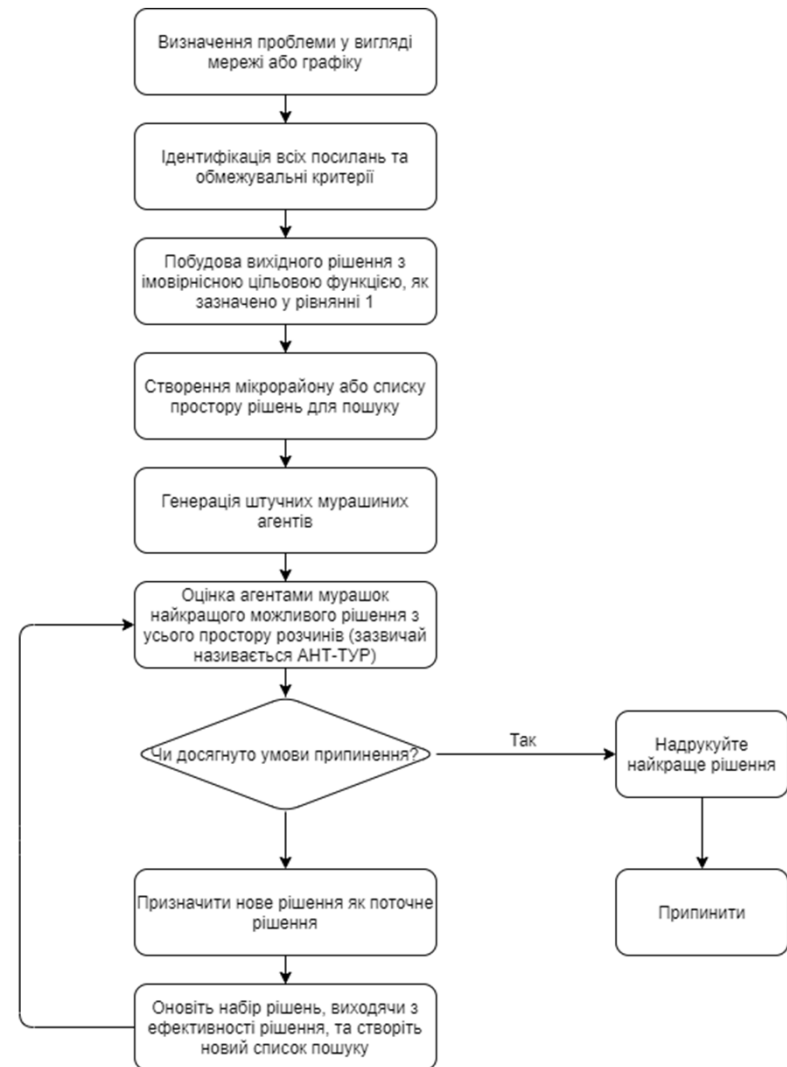
Структура бази даних



Бізнес-процес «Обробка замовлень»

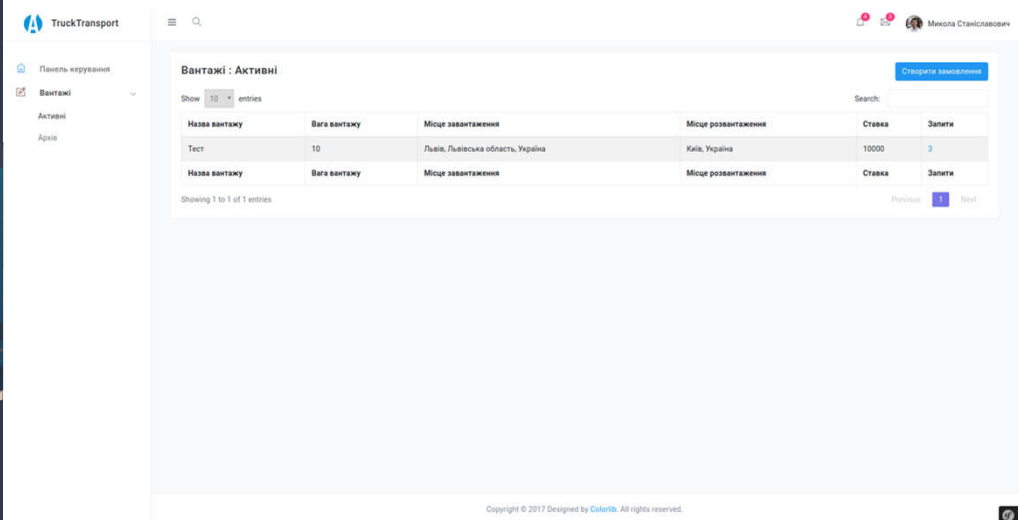


Алгоритм пошуку оптимального маршруту



Результати роботи

Особистий кабінет водія



The screenshot displays the 'TruckTransport' web application interface for a driver's personal cabinet. The page title is 'Вантажі : Активні' (Cargo : Active). A table lists one active cargo entry with the following details:

Назва вантажу	Вага вантажу	Місце завантаження	Місце розвантаження	Ставка	Запити
Тест	10	Львів, Львівська область, Україна	Київ, Україна	10000	3

The interface includes a sidebar with navigation options: 'Панель керування', 'Вантажі', 'Активні', and 'Архів'. A search bar is located at the top right of the main content area. The footer contains the text 'Copyright © 2017 Designed by Colorlib. All rights reserved.' and a small HP logo.

Результати роботи

Керування вантажівками



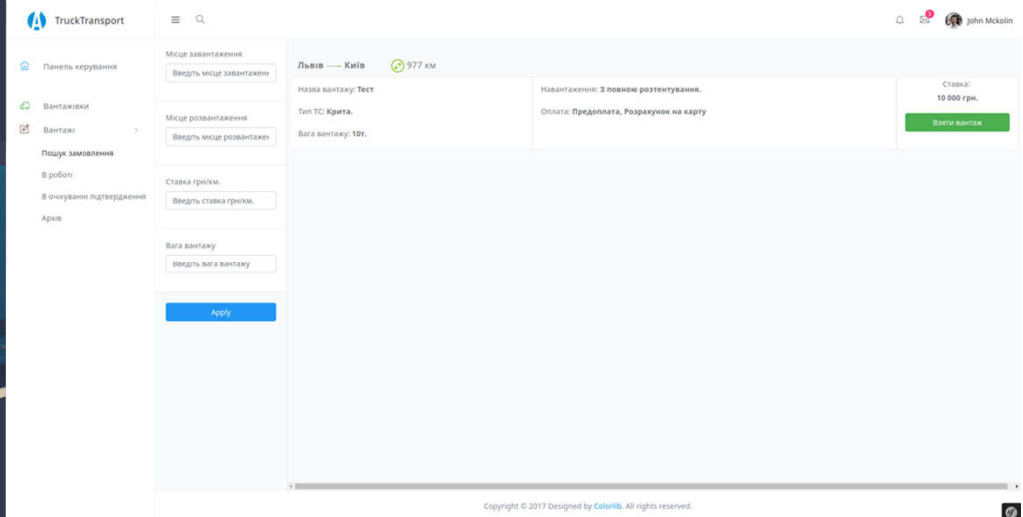
The screenshot displays the TruckTransport web application interface. The header includes the logo and name "TruckTransport", a search icon, and a user profile for "John McKinn". The left sidebar contains navigation options: "Панель керування" (Control Panel), "Вантажівки" (Trucks), and "Вантажі" (Loads). The main content area shows a detailed view for a "Renault 2012" truck, featuring a photograph of the vehicle and the following specifications:

- State number: BC8747AH
- Type: Cover/Tilt
- Volume: 30
- Capacity: 30
- Load types: top

At the bottom of the page, there is a copyright notice: "Copyright © 2017 Designed by Colorlib. All rights reserved." and a small logo in the bottom right corner.

Результати роботи

Інтерфейс пошуку вантажів



The screenshot shows the 'TruckTransport' web application interface. On the left is a navigation sidebar with options: 'Панель керування', 'Вантажівки', 'Вантажі', 'Пошук замовлення', 'В роботі', 'В очікуванні підтвердження', and 'Архів'. The main content area is titled 'Пошук вантажів' and contains several input fields: 'Місце завантаження' (with a 'Введіть місце завантаження' placeholder), 'Місце розвантаження' (with a 'Введіть місце розвантаження' placeholder), 'Ставка грошом.' (with a 'Введіть ставку грошом.' placeholder), and 'Вага вантажу' (with a 'Введіть вагу вантажу' placeholder). A blue 'Apply' button is located below these fields. The search results section shows a route from 'Львів' to 'Київ' (377 km). The cargo details include: 'Назва вантажу: Тест', 'Тип ТС: Критя.', 'Навантаження: З повною розетнування.', 'Оплата: Преплата, Розрахунок на карту', and 'Ставка: 10 000 грн.'. A green 'Вантаж' button is visible next to the rate. The footer contains the text 'Copyright © 2017 Designed by Colorlib. All rights reserved.' and a small logo.

Результати роботи

Інтерфейс створення замовлення

TruckTransport

Панель керування
Вантажі
Активні
Архів

Вантажі : Активні
Show 10 entries

Назва вантажу
Тест

Назва вантажу
Showing 1 to 1 of 1 entries

Створити замовлення

Маршрут

Місце завантаження
Львів, Львівська область, Україна

Місце розвантаження
Хмельницький, Хмельницька об.

Параметри вантажу

Назва вантажу
Введіть назву вантажу

Вага вантажу
Введіть масу вантажу

Тип транспорту
Критя
Відкрита
Спецтранспорту

Соплати
Немає тари

Вартість і оплата

Вартість перевезення
Введіть свою ставку

Тип оплати
Розрахунок на карт

Момент оплати
Предоплата

Валюта
UAH

Навантаження
Версія
Бюкова
Забір
З повною розвантаженою

Довантаження

Close Create order

Copyright © 2017. Designed by Doklady. All rights reserved.

Результати роботи

Пошук оптимального маршруту

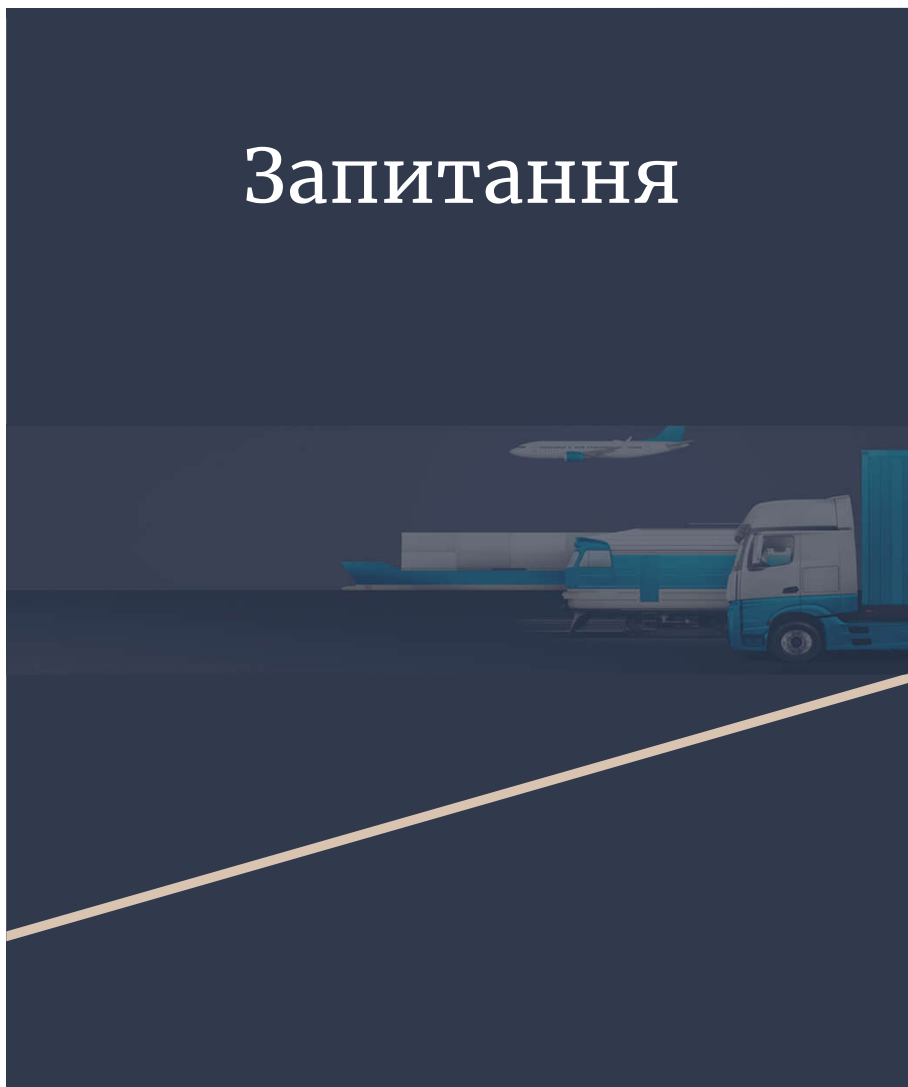
The screenshot displays the TruckTransport web application interface. On the left is a navigation menu with the following items: "Панель керування", "Вантажівки", "Вантажі", "Пошук замовлення", "В роботі", "В очікуванні підтвердження", and "Архів". The main content area shows a route search result for the path: Львів — Івано-Франківськ — Тернопіль — Рівне, with a distance of 423 km. The route is visualized with a red arrow pointing to the distance. The search parameters are: "Місце завантаження: Львів, Львівська область", "Проміжна точка #1: Івано-Франківськ, Івано-Франківська область", "Проміжна точка #2: Тернопіль, Тернопільська область", and "Місце розвантаження: Рівне, Рівненська область". The cargo details include: "Назва вантажу: Картопля", "Тип ТС: темп.", "Вага вантажу: 40т.", "Навантаження: Бокова.", and "Оплата: Предоплата, Розрахунок на карту". The rate is listed as "Ставка: 40 грн." with a green button labeled "Взяти вантаж". The user's name "Микола Грицюк" is visible in the top right corner.

Висновки

Світова глобалізація формує значні і складні виклики щодо забезпечення логістичної підтримки сучасного виробництва. Зв'язку з одночасним розвитком інформаційних технологій, їх високою ефективністю, пристосованістю до вирішення формалізованих задач широкого спектру застосування, найбільш доцільним є вирішення задач організації перевезень на основі інформаційних технологій та систем.

Тому в рамках дипломної роботи магістра було реалізовано інформаційну систему знаходження оптимального маршруту, яка ґрунтується на мурашиному алгоритмі, що дозволяє віднайти оптимальний маршрут з інтегрованим розрахунком вартості перевезення.

Запитання



Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 2.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилоч в документах: 8%**

ID: 81737 Назва: Інформаційна система планування найкращого шляху для доставки вантажу за допомогою задачі комівояжера Додано в БД: 2020-11-30 Автора: Грицюк Микола Станіславович Керівники: Пасічник О.А. Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	57655	429	2355 (4%)	24 (6%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

**РІШЕННЯ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Інформаційна система планування найкращого шляху для доставки вантажу за допомогою задачі комівояжера

Автор: Грицюк М.С.

Спеціальність: 122 Комп'ютерні науки

Науковий керівник: к.т.н., доцент Пасічник О.А.

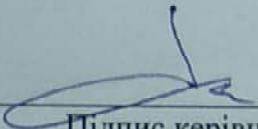
Після аналізу звіту подібності зроблено такий висновок:


№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних). Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	-
3	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	-
4	Інше:	-

Підтвердження: Виявлені запозичення не є плагіатом т.я. розміщені в розділах, які не описують безпосередньо авторське дослідження, складають 6,8% та мають посилання на приведений список літературних джерел

02.12.2020

Дата


Підпис керівника


Підпис завідувача кафедри

ВІДГУК ОПОНЕНТА
на дипломну роботу магістра

Магістра гр. КНМ-19-1 Грицюк Микола Станіславович

На тему: Інформаційна система планування найкращого шляху для доставки вантажу за допомогою задачі комівояжера.

1. Актуальність і значення теми

Вантажоперевезення в Україні займає найбільшу частку серед усіх видів транспортних послуг. Мінімальна частина задіяних в даному процесі перевізників працює на контрактній основі з великими підприємствами-замовниками. Решта учасників ринку взаємодіють з клієнтами без підписання угод про довгострокове співробітництво. Такий варіант двостороннього партнерства є вкрай ризикованим, тому що пошук замовника або перевізника на дошках оголошень може завершитися зустріччю з шахраями. В результаті транспортна організація залишається без заробітку, а замовник позбавляється вантажу.

Чимало водіїв, які працюють в компаніях, що надають послуги вантажоперевезень задумуються про відкриття своєї справи в даній сфері, але для того, щоб бути незалежним від транспортної компанії недостатньо володіти вантажним автомобілем, так як однією з основних послуг, які надають транспортні компанії - це пошук замовлень. Враховуючи те, що замовлення, які надаються компаніями, не завжди відповідають очікуванням водіїв, а саме місце завантаження може знаходитися в десятках, а то і в сотнях кілометрів від місця розташування водія. Не всі компанії компенсують кошти на паливе, яке використовується для того, щоб дістатися місце завантаження. В результаті водії зобов'язані сплачувати послуги оператора, а також компенсувати з власної кишені добування до місця завантаження, що суттєво знижує їхній дохід.

Світова глобалізація формує значні і складні виклики щодо забезпечення логістичної підтримки сучасного виробництва. Зв'язку з одночасним розвитком інформаційних технологій, їх високою ефективністю, пристосованістю до вирішення формалізованих задач широкого спектру застосування, найбільш доцільним є вирішення задач організації перевезень на основі інформаційних технологій та систем.

Завдяки сучасним технологіям можна змінити дану ситуацію, а саме надати водієві можливість шукати замовлення напряму, без посередників, в ролі яких виступають різного типу

транспортні компанії. Це дозволить ефективно планувати маршрут, а також стати незалежним від транспортних компаній.

2. Оцінка якості та достовірності проведених досліджень.

Отримані результати з невеликою похибкою співвідносяться з результатами, наведеними в наукових роботах і довідниках.

3. Оцінка запропонованих заходів та пропозицій, практичної цінності та ефективності.

Проведені дослідження представляють науково-технічну цінність, є ефективним дослідженням в галузі вантажних перевезень, їх можна використати з метою оптимізації маршрутів для здійснення вантажних перевезень.

4. Загальний висновок та оцінка

Робота виконана в повному обсязі. Досліджені та проаналізовані дані в рамках допустимих відхилень. Пояснювальна записка оформлена в відповідності з нормами. Є невеликі недоліки, які не знижують цінності дипломної роботи. За своєю структурою, практичними цінностями, поставленій меті та вирішеними задачами робота відповідає вимогам вищої школи і вимогам, що пред'являються до освітньо-кваліфікаційного рівня «магістр», а її автор Грицюк М.С. заслуговує присвоєння кваліфікації магістра з комп'ютерних наук та інформаційних технологій.

Робота заслуговує на оцінку « добре ».

Опонент

Мергелия В.К., д.т.н., проф.