

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

«До захисту допущено»

Завідувач кафедри

_____ **В. О. РОМАНКЕВИЧ**
(підпис) (ініціали, прізвище)

“ ___ ” червня 2025 р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою

«Системне програмування та спеціалізовані комп'ютерні системи»

по спеціальності

123 «Комп'ютерна інженерія»

на тему: Комп'ютерна система організації сумісних перевезень вантажів різних габаритів

Виконав:

студент IV курсу, групи КВ-12

Петрушин Віктор Борисович

(підпис)

Керівник: ас. каф. СПСКС ФПМ Кучмій О.О.

(підпис)

Консультант з нормоконтролю:

доц. каф. СПСКС, к.т.н., доц. Клятченко Я.М.

(підпис)

Рецензент: доц. каф. ПЗКС, к.т.н, доц. Заболотня Т. М.

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному проєкті немає запозичень з праць інших авторів без відповідних посилань.

Студент _____

(підпис)

Київ – 2025 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 123 «Комп'ютерна інженерія»

Освітньо-професійна програма

«Системне програмування та спеціалізовані комп'ютерні системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Віталій РОМАНКЕВИЧ

(підпис)

(ініціали, прізвище)

« 23 » грудня 2024 р.

ЗАВДАННЯ

**на дипломний проєкт студента
Петрушина Віктора Борисовича**

1. Тема проєкту «Комп'ютерна система організації сумісних перевезень вантажів різних габаритів», керівник проєкту ас. каф. СПСКС ФПМ Кучмій О.О., затверджені наказом по університету від «___» травня 2025 р. № _____
2. Термін подання студентом проєкту _____
3. Вихідні дані до проєкту: див. Технічне завдання
4. Зміст пояснювальної записки
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо)

6. Консультанти розділів проєкту*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	доц.каф.СПСКС, к.т.н. Клятченко Я.М.		

7. Дата видачі завдання _____ року

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Терміни виконання етапів проєкту	Примітка
1.	Вивчення літератури за тематикою проєкту	24.12.2024	
2.	Аналіз існуючих рішень	11.01.2025	
3.	Розробка технічного завдання	18.01.2025	
4.	Підготовка матеріалів першого розділу дипломного проєкту	15.02.2025	
5.	Підготовка матеріалів другого розділу дипломного проєкту	21.03.2025	
6.	Підготовка матеріалів третього розділу дипломного проєкту	16.04.2025	
7.	Підготовка матеріалів четвертого розділу дипломного проєкту	30.04.2025	
8.	Підготовка графічної частини дипломного проєкту	05.05.2025	
9.	Оформлення документації дипломного проєкту	29.05.2025	
10.	Попередній огляд матеріалів диплому на кафедрі	05.06.2025	

Студент

(підпис)

Віктор ПЕТРУШИН

Керівник проєкту

(підпис)

Оксана КУЧМІЙ

* Консультантом не може бути зазначено керівника дипломного проєкту.

АНОТАЦІЯ

Об'єкт розробки – комп'ютерна система організації сумісних перевезень вантажів різних габаритів. Мета розробки – створення програмного забезпечення, що дозволяє користувачам публікувати запити на перевезення вантажів, а перевізникам – знаходити відповідні замовлення відповідно до власного маршруту. Система реалізована у вигляді вебзастосунку та забезпечує реєстрацію і авторизацію користувачів, створення маршрутів, фільтрацію замовлень, перегляд профілів та зворотний зв'язок у вигляді відгуків. У процесі розробки були використані технології Flutter, Firebase Firestore, Firebase Authentication.

У процесі розробки було виконано наступне:

- проведено аналіз існуючих систем організації перевезень;
- визначено основні вимоги до функціональності та архітектури застосунку;
- реалізовано клієнтську та серверну частини;
- впроваджено механізм пошуку перевезень за заданим маршрутом;
- забезпечено інтуїтивно зрозумілий інтерфейс користувача, зручний для перевізників та замовників.

Ключові слова:

КОМП'ЮТЕРНА СИСТЕМА, ЛОГІСТИКА, WEB, FLUTTER, ПЕРЕВЕЗЕННЯ
ВАНТАЖІВ, HTTPS, FIREBASE, ВЕБЗАСТОСУНОК, UX

ABSTRACT

The subject of the development is a computer system for organizing joint transportation of cargo of various sizes. The aim of the project is to create software that enables users to publish cargo transportation requests, and allows carriers to find suitable orders according to their route. The system is implemented as a web application and provides user registration and authentication, route creation, order filtering, profile viewing, and feedback in the form of reviews. During the development process, technologies such as Flutter, Firebase Firestore, and Firebase Authentication were used.

The following tasks were completed during development:

- analysis of existing freight transportation systems was conducted;
- key functional and architectural requirements for the application were defined;
- both client-side and server-side components were implemented;
- a search mechanism for transportation based on the specified route was developed;
- a user-friendly and intuitive interface suitable for both carriers and customers was designed.

Keywords:

COMPUTER SYSTEM, LOGISTICS, WEB, FLUTTER, FREIGHT
TRANSPORTATION, HTTPS, FIREBASE, WEB APPLICATION, UX

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	A4	ІАЛЦ.045440.002 ТЗ	Комп'ютерна система організації сумісних перевезень вантажів різних габаритів	4		
			Технічне завдання			
	A4	ІАЛЦ.045440.003 ТП	Комп'ютерна система організації сумісних перевезень вантажів різних габаритів	2		
			Відомість технічного проєкту			
	A4	ІАЛЦ.045440.004 ПЗ	Комп'ютерна система організації сумісних перевезень вантажів різних габаритів	63		
			Пояснювальна записка			
	A4	ІАЛЦ.045440.005 Д1	Структура системи	1		
			Схема структурна			

					ІАЛЦ.045440.001 ОА			
Змін	Арк.	№ докум.	Підпис	Дата				
Розробив		Петрушин В. Б.			Комп'ютерна система організації сумісних перевезень вантажів різних габаритів. Опис альбому	Лп.	Аркуш	Аркушів
Перевірив		Кучмій О. О.					1	2
Консульт.						КПІ ім. Ігоря Сікорського, ФПМ, КВ-12		
Н. контроль		Клятченко Я. М.						
Зав. каф.		Романкевич В. О.						

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ	2
2. ПІДСТАВА ДЛЯ РОЗРОБКИ.....	2
3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ.....	2
4. ДЖЕРЕЛА РОЗРОБКИ	2
5. ТЕХНІЧНІ ВИМОГИ	3
5.1. Вимоги до програмного забезпечення системи, що розробляється	3
5.2. Вимоги до апаратного забезпечення.....	3
5.3. Вимоги до програмного та апаратного забезпечення користувача	3
6. ЕТАПИ РОЗРОБКИ.....	4

					ІАЛЦ. 045440.002 ТЗ		
Зм	Лист	№ докум.	Підп.	Дата			
Розробив		Петрушин В.Б.			Літ.	Аркуш	Аркушів
Перевірив		Кучмій О. О.			1	4	
Н. контр.		Клятченко Я.М.			КПІ ім. Ігоря Сікорського ФПМ, КВ-12		
Затвердив		Романкевич В.О.					
Комп'ютерна система для організації сумісних перевезень вантажів різних габаритів. Технічне завдання							

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки – «Комп'ютерна система організації сумісних перевезень вантажів різних габаритів».

Область застосування: Організація сумісних перевезень вантажів різних габаритів із метою оптимізації використання транспортних засобів і зниження витрат.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на дипломне проектування на здобуття першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський Політехнічний Інститут імені Ігоря Сікорського».

3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проекту є створення програмного продукту, яке дозволяє користувачам планувати та організовувати сумісні перевезення вантажів різних габаритів, зменшуючи витрати на транспортування.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації у періодичних виданнях та електронні статті у мережі Інтернет.

Зм	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.045440.002 ТЗ

Арк.

2

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до програмного забезпечення системи, що розробляється

Програмне забезпечення, що розробляється, повинно забезпечувати наступні функціональні можливості та задовольняти такі вимоги:

- авторизація користувачів;
- можливість створення замовлень на перевезення;
- можливість пошуку замовлень за суміжним маршрутом;
- перегляд створених замовлень та відгуків на них;
- інтуїтивно зрозумілий інтерфейс користувача.

5.2. Вимоги до апаратного забезпечення

- операційна система: Windows 10, macOS 10.15 або будь-який сучасний дистрибутив Linux;
- процесор: не менше, ніж 4 ядра, наприклад Intel Core i5;
- оперативна пам'ять: від 4 Гб;
- наявність доступу до мережі Wi-Fi.

5.3. Вимоги до програмного та апаратного забезпечення користувача

- пристрій з доступом до Інтернету (персональний комп'ютер, ноутбук, планшет або смартфон);
- підтримка сучасного веббраузера: Google Chrome, Safari, Microsoft Edge (останні версії);
- операційна система: Android, iOS, Windows, macOS, Linux (версії, яка підтримує останню версію веббраузера).

Зм	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.045440.002 ТЗ

Арк.

3

6. ЕТАПИ РОЗРОБКИ

Видача завдання на дипломне проєктування – 10.12.2024

Вивчення літератури за тематикою роботи – 11.12.2025 – 11.01.2025

Аналіз існуючих рішень – 11.01.2025 – 15.02.2025

Розроблення та узгодження технічного завдання – 15.02.2025 - 05.03.2025

Розроблення структури додатку – 05.03.2025 - 16.04.2025

Розроблення дизайну та графічних елементів – 16.04.2025 - 25.02.2025

Програмна реалізація додатку – 26.02.2025 - 07.04.2025

Тестування додатку – 08.04.2025 - 14.04.2025

Підготовка матеріалів текстової частини проєкту – 15.04.2025 - 05.05.2025

Підготовка матеріалів графічної частини проєкту – 05.05.2025 - 29.05.2025

Оформлення технічної документації проєкту – 29.05.2025 - 05.06.2025

					ІАЛЦ.045440.002 ТЗ	Арк.
Зм	Арк.	№ докум.	Підпис	Дата		4

ПОЯСНЮВАЛЬНА ЗАПИСКА ДО ДИПЛОМНОГО ПРОЄКТУ

на тему: Комп'ютерна система організації сумісних перевезень
вантажів різних габаритів

Київ – 2025 року

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ.....	3
ВСТУП.....	4
1. ДОСЛІДЖЕННЯ АСПЕКТІВ СУМІСНИХ ПЕРЕВЕЗЕНЬ ВАНТАЖІВ ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ.....	6
1.1. Концепція сумісних перевезень вантажів	6
1.2. Сучасний стан і основні проблеми напрямку сумісних перевезень	8
1.3. Роль цифрових технологій у логістиці	11
1.4. Аналіз існуючих сервісів та обґрунтування необхідності нової системи	13
2. ВИБІР ТЕХНОЛОГІЙ І ЗАСОБІВ РОЗРОБЛЕННЯ СИСТЕМИ ДЛЯ ОРГАНІЗАЦІЇ СУМІСНИХ ПЕРЕВЕЗЕНЬ ВАНТАЖІВ РІЗНИХ ГАБАРИТІВ	19
2.1. Формат програмної реалізації	19
2.2. Аналіз обраних інструментів та платформи для створення веб-системи	21
2.3. Вибір системи управління базами даних.....	24
2.4. Цільова спрямованість дизайну та архітектури системи.....	28
3. РЕАЛІЗАЦІЯ КОМП'ЮТЕРНОЇ СИСТЕМИ ОРГАНІЗАЦІЇ СУМІСНИХ ПЕРЕВЕЗЕНЬ ВАНТАЖІВ РІЗНИХ ГАБАРИТІВ	30
3.1. Характеристика структури та вимоги до системи.....	30
3.2. Розробка клієнтської частини.....	33
3.3. Розробка серверної частини та взаємодія з базою даних.....	41
3.4. Реалізація пошуку перевезень за маршрутом	44
4. ОГЛЯД ТА ТЕСТУВАННЯ РЕАЛІЗОВАНОЇ СИСТЕМИ	48
4.1. Огляд інтерфейсу та тестування системи.....	48
4.2. Перспективи розвитку системи.....	60
ВИСНОВКИ	62
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	63

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

MVP – мінімально життєздатний продукт;

СУБД – система управління базами даних;

SQL – реляційна база даних;

NoSQL – нереляційна база даних;

HCD – Human-Centered Design;

UX – User Experience;

SDK – Software Development Kit;

HTTPS – HyperText Transfer Protocol Secure;

API – Application Programming Interface;

ТЗ – транспортний засіб.

					ІАЛІЦ.045440.004 ПЗ	Арк.
Зм	Арк.	№ докум.	Підп.	Дата		3

ВСТУП

У сучасному світі питання логістики та перевезень набуває особливої актуальності. Щодня по дорогах курсують тисячі вантажівок і мікроавтобусів, значна частина яких пересувається порожніми або з частково заповненим вантажним простором. Така неефективність призводить до зайвих витрат пального, перевантаження доріг та екологічних проблем. Одним із перспективних шляхів вирішення цієї проблеми є оптимізація транспортування за допомогою сумісних перевезень.

Суть підходу, запропонованого в ході виконання дипломного проєкту, полягає в тому, що перевізники, які мають вільне місце у транспорті або планують поїздку без вантажу, наприклад до місця завантаження, можуть взяти додаткові вантажі або посилки по маршруту, який вони й так мають намір подолати. Це вигідно як перевізникам, так і відправникам – перші отримують додаткову оплату, а другі – доступну і порівняно дешеву послугу доставки. Така модель дозволяє ефективніше використовувати наявні транспортні ресурси, зменшує кількість порожніх рейсів та сприяє зниженню загального навантаження на дорожню інфраструктуру й навколишнє середовище.

Для розвитку даної ідеї необхідна зручна цифрова платформа, яка дозволить відправникам публікувати запити на перевезення, а перевізникам – швидко знаходити відповідні пропозиції за маршрутом. Розробка такого програмного забезпечення дозволяє не лише зменшити витрати на перевезення, а й сприяє екологічній та економічній ефективності транспортної галузі.

Зважаючи на широке розповсюдження доступу до Інтернету та активне використання веб-платформ у повсякденному житті, найбільш доцільним рішенням є реалізація даної системи у вигляді веб-застосунку. Це дозволяє забезпечити доступність сервісу з будь-якого пристрою – як для водіїв, так і для замовників – незалежно від операційної системи чи місця знаходження. Веб-

					ІАЛЦ.045440.004 ПЗ	Арк.
Зм	Арк.	№ докум.	Підп.	Дата		4

застосунок забезпечить швидкий обмін інформацією, зручний інтерфейс та доступність послуг у режимі реального часу.

Метою даного дипломного проєкту є розробка комп'ютерної системи організації сумісних перевезень вантажів різних габаритів із можливістю публікації запитів на перевезення з подальшою комунікацією замовника та перевізника. Реалізація даної системи сприятиме розвитку напрямку сумісного транспортування в Україні та дозволить покращувати доступну й екологічну альтернативу традиційним логістичним рішенням.

					ІАЛЦ.045440.004 ПЗ	Арк.
Зм	Арк.	№ докум.	Підп.	Дата		5

1. ДОСЛІДЖЕННЯ АСПЕКТІВ СУМІСНИХ ПЕРЕВЕЗЕНЬ ВАНТАЖІВ ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

1.1. Концепція сумісних перевезень вантажів

У сучасних умовах глобалізації, активного розвитку торгівлі та зростання обсягів логістичних операцій питання ефективної організації вантажоперевезень набуває особливої актуальності. Одним із перспективних напрямів оптимізації транспортних процесів є концепція сумісних перевезень вантажів, яка базується на ідеї раціонального використання вільного простору у транспортному засобі шляхом об'єднання кількох відправлень в одній поїзді.

Сумісні перевезення передбачають доставку вантажів від різних замовників одним транспортним засобом, що прямує заданим маршрутом. Така модель дозволяє зменшити кількість порожніх пробігів, знизити витрати на логістику, підвищити ефективність транспортного використання та водночас зменшити негативний вплив на довкілля. Концепція сумісних перевезень виникла як відповідь на потребу зменшити кількість неефективних рейсів, коли транспортний засіб повертається з пункту розвантаження до місця базування або наступного завантаження без вантажу. В багатьох країнах суттєвий відсоток вантажівок у будь-який момент часу рухаються порожніми, що призводить до значних економічних втрат та нераціонального використання пального.

Переважає більшість компаній не мають можливості утримувати власні автопарки для транспортування та доставки, що значно ускладнює організацію логістики, особливо для малих та середніх підприємств. У таких випадках необхідність у ефективних та економічних рішеннях стає ще більш актуальною. Також доволі часто продукція компаній може мати специфічну форму і габарити, або ж взагалі не підпадає під критерії вантажу, дозволеного для транспортування. У таких ситуаціях традиційні служби доставки, такі як пошта або кур'єрські компанії, часто не підходять або стають занадто дорогими для бізнесу, що не дозволяє забезпечити ефективну та доступну доставку. Зі

					ІАЛЦ.045440.004 ПЗ	Арк.
Зм	Арк.	№ докум.	Підп.	Дата		6

звичайними, на перший погляд, товарами також часто можуть виникати труднощі в питаннях транспортування, до прикладу: заборонено відправляти вживані акумулятори та накопичувальні елементи, а автомобільні салонні стелі не підлягають вимогам через надмірну крихкість. Власне, ці проблеми і є причинами, чому попутна логістика та альтернативні моделі вантажоперевезень набувають все більшої популярності серед підприємців та споживачів, що шукають економічно вигідні способи доставки товарів.

Основна ідея сумісних перевезень – узгодження інтересів відправників і перевізників. Перші прагнуть швидко та недорого доставити свій вантаж, особливо якщо він невеликого обсягу або нестандартного габариту, що унеможливорює економічну доцільність найму окремого транспорту. Другі – маючи вільне місце в кузові або плануючи поїздку без вантажу (наприклад, до місця основного завантаження) – можуть взяти додаткове навантаження, отримавши за це додаткову оплату.

Для успішної та ефективної організації сумісних перевезень необхідна платформа для швидкого та зручного пошуку відповідного транспорту або вантажу. Така платформа повинна мати функціональність, що дозволяє публікувати запити на перевезення з вказаними параметрами та допомагає здійснювати пошук перевізників, що планують подорож відповідним маршрутом.

Одним з головних факторів, що впливає на привабливість сумісних перевезень, є вартість логістичних послуг. Традиційні логістичні оператори, особливо у сегменті термінової або адресної доставки, встановлюють досить високі тарифи. Для прикладу розглянемо транспортування вантажу з об'ємною вагою 100 кг із Києва до Львова. У випадку користування послугами традиційного логістичного оператора, такого як «Нова пошта», вартість доставки може становити від 900 до 1200 грн за базову палетну доставку, залежно від габаритів, страхування, типу пакування тощо. Якщо ж додати

					ІАЛЦ.045440.004 ПЗ	Арк.
Зм	Арк.	№ докум.	Підп.	Дата		7

додаткові послуги – для прикладу забір вантажу за адресою (+ 200–400 грн), адресна доставка (+ 200–400 грн) і комісія за переказ коштів (безпосередньо залежить від суми переказу) — загальна сума легко може сягнути 1500–1800 грн [1]. У разі використання послуги сумісного перевезення орієнтовна вартість такої доставки складатиме 500-600 грн. Ціна може бути навіть нижчою, якщо маршрут повністю збігається з уже запланованою поїздкою перевізника (наприклад, він їде на завантаження порожнім).

Що стосується термінів доставки, ситуація тут більш складна і неоднозначна. У традиційних кур'єрських службах терміни доставки зазвичай чітко регламентовані (1–3 дні між великими містами), тоді як у попутній логістиці вони залежать від реального маршруту перевізника, графіку його руху та погодженості з відправником. У більшості випадків попутна доставка триває менше однієї доби, але зважаючи на велику кількість факторів, які можуть впливати на термін доставки, цей процес може тривати й набагато більше часу.

1.2. Сучасний стан і основні проблеми напрямку сумісних перевезень

Сумісні вантажоперевезення є відносно новим, але швидко зростаючим напрямом у сфері логістики, що виник як відповідь на актуальні виклики: неефективне використання транспортних ресурсів, зростання цін на паливо, підвищення екологічних вимог та потребу у швидких, дешевих способах транспортування вантажів. Основна ідея полягає в раціональному використанні вже запланованих рейсів: якщо транспортний засіб рухається в певному напрямку з частково завантаженим кузовом або взагалі без вантажу, логічно й вигідно надати можливість доповнити його вантажем від інших клієнтів. Такий підхід сприяє оптимізації логістичних потоків, економії коштів і зменшенню викидів шкідливих речовин в атмосферу.

					ІАЛЦ.045440.004 ПЗ	Арк.
Зм	Арк.	№ докум.	Підп.	Дата		8

На практиці модель сумісних перевезень виявляється особливо актуальною для малих та середніх підприємств, приватних перевізників, а також для фізичних осіб, яким потрібно доставити вантаж або посилку з одного міста в інше без значних витрат. З іншого боку, для перевізників – це можливість монетизувати "порожні пробіги", які є однією з найболючіших економічних проблем автотранспортної галузі. Статистичні дані вказують на те, що в деяких країнах приблизно 20–30% вантажного транспорту щодня курсують без корисного навантаження [2], тобто не приносять прибутку, але споживають паливо, зношують транспорт та збільшують навантаження на дорожню інфраструктуру [3].

Незважаючи на очевидні переваги, повномасштабне впровадження сумісних перевезень стикається з низкою системних проблем. Перш за все – це відсутність централізованої цифрової інфраструктури, яка б дозволяла оперативно знаходити попутний транспорт або відповідні вантажі. У більшості випадків інформація про вантажі й транспорт зберігається в ізольованих базах даних або передається через особисті контакти, що істотно обмежує швидкість і масштаб співпраці між замовниками та перевізниками.

Ще однією суттєвою проблемою є низький рівень довіри між учасниками перевезень. Відсутність гарантій доставки, непрозорість умов співпраці, можливість шахрайських дій – усе це стримує потенційних користувачів від участі у подібних сервісах. Додатково варто враховувати й такі фактори, як відсутність стандартів упаковки та транспортування, непередбачуваність дорожньої ситуації, недостатній рівень цифрової грамотності окремих користувачів.

У деяких регіонах та країнах світу вже реалізуються ініціативи з підтримки попутних перевезень. В Україні попутні вантажоперевезення залишаються переважно на рівні неформального ринку або реалізуються через приватні оголошення в месенджерах і соціальних мережах, що істотно знижує

					ІАЛЦ.045440.004 ПЗ	Арк.
						9
Зм	Арк.	№ докум.	Підп.	Дата		

ефективність і надійність подібних рішень. Також варто зазначити, що існуючі сервіси в переважній більшості працюють згідно моделі Замовник – Компанія-посередник – Перевізник (Рисунок 1.1).

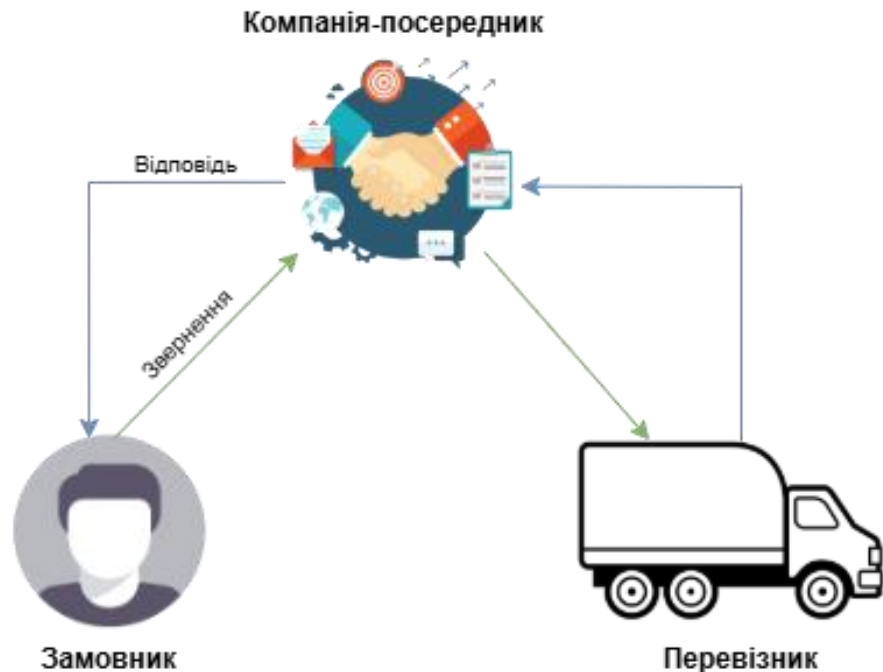


Рисунок 1.1 – Модель співпраці Замовник – Компанія-посередник – Перевізник

Загалом можна відзначити, що концепція сумісних вантажоперевезень перебуває на етапі становлення, поступово інтегруючись у більш широкі логістичні моделі. Її розвиток потребує вирішення як технічних, так і організаційних завдань: впровадження цифрових платформ, правове регулювання, стимулювання екологічної логістики, розвиток партнерських мереж між перевізниками й клієнтами. Ефективна реалізація таких рішень здатна не лише знизити витрати на транспортування, але й стати важливим кроком до сталої та раціональної логістики майбутнього.

1.3. Роль цифрових технологій у логістиці

У сучасному світі логістика перестає бути процесом транспортування товарів – вона перетворюється на складну, технологічно підтриману систему взаємодії численних учасників: замовників, перевізників, постачальників, операторів складів і кінцевих споживачів. Цифрові технології відіграють ключову роль у цій трансформації. Їх впровадження дозволяє забезпечити гнучкість, оперативність, прозорість та економічність усіх логістичних процесів. Особливо це стосується інноваційних напрямів, таких як попутні вантажоперевезення, де без цифрової підтримки неможливо забезпечити ефективну взаємодію між учасниками процесу.

Одним із найбільш визначальних факторів у цифровій логістиці стали веб-платформи. Саме вони виступають головним інструментом для організації сумісних перевезень вантажів. Такі платформи дозволяють у режимі реального часу об'єднувати попит і пропозицію: замовники можуть розміщувати запити на доставку вантажів, а перевізники – переглядати доступні оголошення відповідно до свого маршруту, вагових і об'ємних обмежень, графіку тощо. Це дає змогу максимально ефективно використовувати вільний простір у транспорті, зменшувати кількість порожніх поїздок і відповідно – витрати.

Цифрова веб-платформа – це не лише зручний інтерфейс, а передусім централізована система керування процесами. Вона забезпечує можливість створення профілів користувачів, перевірки їхньої надійності, ведення історії замовлень, побудови маршрутів, обміну повідомленнями, проведення безпечних онлайн оплат, формування електронних документів тощо. Застосування таких технологій суттєво підвищує прозорість та довіру між перевізниками і клієнтами. Особливо важливим це є у моделі попутних перевезень, де логістичні ланцюги будуються не завдяки централізованим складським рішенням, а на основі взаємної кооперації учасників.

					ІАЛЦ.045440.004 ПЗ	Арк.
Зм	Арк.	№ докум.	Підп.	Дата		11

Крім того, сучасні вебплатформи інтегрують алгоритми фільтрації та машинного навчання, які автоматично підбирають найбільш актуальні запити до маршрутів перевізників або навпаки – пропонують користувачу відповідний транспорт під його вантаж. Це значно зменшує час на пошук, підвищує ймовірність успішного співпадіння запиту й пропозиції та робить сервіс зручним навіть для новачків. За допомогою таких рішень платформи стають не просто місцем пошуку перевезення, а повноцінною екосистемою для оптимізації логістики в реальному часі.

У рамках сумісних перевезень особливо важливим є GPS-трекінг і побудова маршрутів, які також доволі часто забезпечуються веб-застосунками. Вони дають змогу перевізнику бачити можливі варіанти вантажів по маршруту, а замовнику – відстежувати переміщення свого вантажу. Така прозорість підвищує рівень довіри, знижує ризики шахрайства, дозволяє оперативно реагувати на непередбачувані обставини в дорозі.

Вебплатформи також сприяють демократизації ринку логістики. Завдяки їм малі перевізники, що не мають доступу до великих логістичних контрактів або інфраструктури, можуть ефективно працювати безпосередньо з клієнтами. Також замовники – фізичні особи чи невеликі бізнеси – отримують можливість знайти доступний спосіб перевезення, який не передбачає співпраці з великою транспортною компанією або використання послуг традиційної пошти. Особливо це актуально у випадках перевезення великогабаритного або нестандартного вантажу, де звичайна пошта або кур'єрські служби пропонують надто дорогі або складні рішення.

Ще одним важливим напрямом є впровадження електронного документообігу. На веб-платформах можливо формувати й обмінюватися цифровими версіями договорів, товарно-транспортних накладних, квитанцій, звітів про доставку. Це суттєво пришвидшує взаємодію між сторонами, знижує

					ІАЛЦ.045440.004 ПЗ	Арк.
Зм	Арк.	№ докум.	Підп.	Дата		12

адміністративні витрати, зменшує кількість паперової роботи та помилок, а також спрощує юридичне оформлення перевезень.

Загалом цифрові технології – а особливо веб-платформи – стали ключовим фактором, що забезпечує життєздатність моделі попутної логістики. Без таких інструментів було б неможливо швидко, зручно й безпечно координувати десятки тисяч учасників, що переміщуються в різних напрямках. Завдяки їм логістика з централізованого, малорухливого й дорогого процесу перетворюється на динамічну, гнучку систему, яка будується в режимі реального часу відповідно до актуальних потреб.

Таким чином, впровадження цифрових технологій, зокрема через розробку та використання веб-застосунків, є не просто додатковою перевагою, а необхідною умовою ефективного функціонування сучасних моделей сумісних вантажоперевезень. Саме завдяки цифровій інфраструктурі з'являється можливість реалізовувати концепцію спільного використання транспортних ресурсів – з вигодою як для перевізників, так і для відправників.

1.4. Аналіз існуючих сервісів та обґрунтування необхідності нової системи

В останні роки цифровізація логістики та вантажних перевезень в Україні набула стрімкого розвитку. З'явилися численні сервіси, що пропонують послуги перевезення вантажів через онлайн-інтерфейси. Проте, незважаючи на кількість платформ, що вже функціонують на ринку, більшість із них мають низку обмежень або не повністю враховують специфіку сумісних перевезень, що є ключовим елементом нової логістичної моделі.

Розглянемо декілька актуальних прикладів таких подібних платформ: Autotransgarant (<https://www.autotransgarant.com>), Avrora Trans (<https://avrora-trans.com>) та CargoDelivery (<https://www.cargodelivery.com.ua>). Кожен із даних сервісів пропонує власний підхід до організації перевезень, однак не всі вони

					ІАЛЦ.045440.004 ПЗ	Арк.
Зм	Арк.	№ докум.	Підп.	Дата		13

повною мірою відповідають основним вимогам ефективної та економної організації попутної логістики.

Платформа Autotransgarant надає послуги з організації вантажоперевезень різного масштабу — від локальних до міжнародних. Основна модель роботи сервісу передбачає посередництво між замовником та перевізником, де Autotransgarant виконує роль логістичного оператора, який самостійно підбирає транспорт під потреби клієнта.

З переваг даного сервісу можна виокремити можливість міжнародних перевезень, у тому числі з оформленням митних документів. Служба підтримки операторів працює цілодобово. Також наявна партнерська мережа перевізників у багатьох містах України та за кордоном. Інтерфейс платформи зображений на Рисунку 1.2.

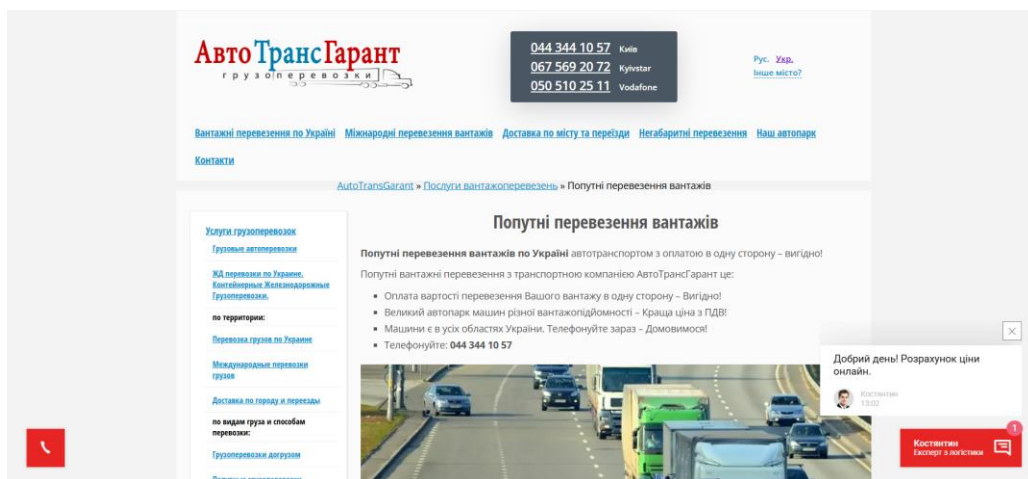


Рисунок 1.2 – Інтерфейс платформи Autotransgarant

До основних недоліків платформи можна віднести:

- Відсутність прямої взаємодії між клієнтом і перевізником: користувач не може самостійно обрати виконавця — це робить логіст менеджер компанії.

					ІАЛЦ.045440.004 ПЗ	Арк.
Зм	Арк.	№ докум.	Підп.	Дата		14

- Закритість платформи — відсутній публічний доступ до бази транспорту або замовлень, через що користувач не має змоги самостійно аналізувати або порівнювати пропозиції.
- Наявність комісій за послуги логістичного супроводу, що робить перевезення дорожчим, особливо для дрібних вантажів.
- Фокус на повноцінні вантажі: модель роботи не орієнтована на сумісні (попутні) перевезення, коли можливо ефективно використати лише частину вільного простору в автомобілі.

Таким чином, Autotransgarant забезпечує зручний сервіс для клієнтів, які прагнуть перекласти всю логістику на оператора, але цей підхід є менш ефективним у випадках, коли потрібно перевезти вантаж невеликих габаритів або частково заповнити вже наявний маршрут перевізника. Сервіс не враховує можливість гнучкої співпраці на умовах прямого контакту, швидкого підбору за маршрутом або економії на спільному транспортуванні.

Avrora Trans є однією з провідних логістичних компаній, що надає послуги з організації вантажоперевезень в Україні та за її межами. Вона виступає в ролі посередника між замовником і перевізником, сама підбирає виконавця, керуючись запитом клієнта.

До переваг даного сервісу належать:

- Повна відповідальність компанії за організацію перевезення.
- Можливість отримати консультацію оператора.
- Підтримка міжміських і міжнародних перевезень.
- Зручний сучасний інтерфейс користувача, завдяки якому нові клієнти можуть з легкістю розібратися у користуванні сервісом.

Звертаючи увагу на недоліки, варто виокремити все ті ж:

- Замовник не може самостійно оцінити пропозиції або обрати перевізника за своїми критеріями.

					ІАЛЦ.045440.004 ПЗ	Арк.
Зм	Арк.	№ докум.	Підп.	Дата		15

- Наявна додаткова комісія сервісу, що збільшує вартість перевезення.

Інтерфейс сервісу зображений на Рисунку 1.3.

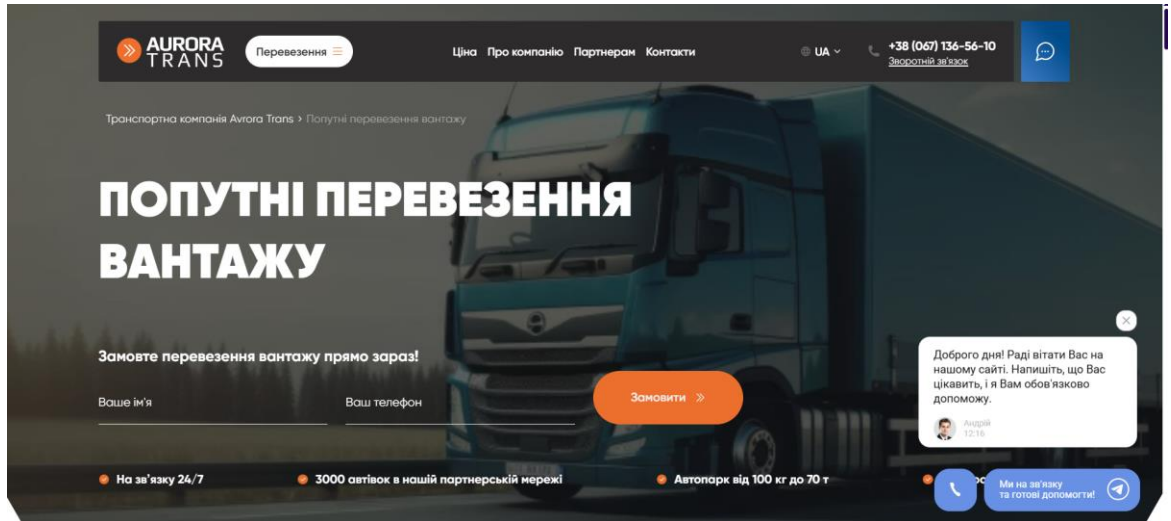


Рисунок 1.3 – Інтерфейс сервісу Aurora Trans

Сервіс CargoDelivery є прикладом традиційної логістичної компанії з елементами цифрової інфраструктури. Користувачі можуть замовити вантажоперевезення через сайт або мобільний додаток, вибрати тип послуги, розрахувати орієнтовну вартість доставки.

Переваги сервісу:

- Наявність власного автопарку та розгалуженої логістичної інфраструктури.
- Зрозуміла тарифікація, онлайн калькулятор.
- Можливість перевезення вантажів у стислі терміни.

До недоліків можна віднести:

- Має доволі примітивний інтерфейс.
- Висока вартість доставки для нестандартних або великогабаритних вантажів.

					ІАЛЦ.045440.004 ПЗ	Арк.
Зм	Арк.	№ докум.	Підп.	Дата		16

- Відсутність можливості індивідуального контакту з перевізником.
- Обмежена гнучкість маршруту: доставка виконується виключно за графіком компанії.

Інтерфейс сервісу зображений нижче, на Рисунку 1.4.

ГОЛОВНА МІЖНАРОДНІ ВАНТАЖОПЕРЕВЕЗЕННЯ ВАНТАЖОПЕРЕВЕЗЕННЯ ПО УКРАЇНІ ПЕРЕЇЗДИ НЕГАБАРИТНІ ПЕРЕВЕЗЕННЯ О КОМПАНІЇ НАШ АВТОПАРК КОНТАКТИ

CargoDelivery » Вантажні перевезення по Україні. Оплата в одну сторону. Найкраща ціна на перевезення вантажів. » Попутні перевезення вантажів — з оплатою в одну сторону. Попутні вантажні перевезення по Україні

Попутні перевезення вантажів – з оплатою в одну сторону. Попутні вантажні перевезення по Україні

🚚 Попутні вантажні перевезення по Україні

Попутні перевезення вантажів – з оплатою в одну сторону. Попутні вантажні перевезення по Україні. Краща ціна на попутні перевезення по Україні – з транспортною компанією CargoDelivery Машини зараз є во всіх містах! Телефонуйте! **+38 (067) 577-50-39**
 Kyivstar **+38 (050) 452-05-05** Vodafone

Транспортна компанія Cargo Delivery – це надійні перевезення вантажів автотранспортом по Україні (всі перевезення попутно з оплатою в одному напрямку) або довантаженням. Ціна за 1 км – недорого. **Телефонуйте нам сьогодні – замовляйте вантажну машину зараз!**

Рисунок 1.4 – Інтерфейс сервісу CargoDelivery

Аналіз вищеописаних сервісів дозволяє зробити висновок: на ринку України відсутній зручний, доступний та орієнтований саме на сумісні вантажоперевезення веб-застосунок, який би надавав користувачам можливість самостійно обирати перевізника, узгоджувати маршрут і ціну без участі посередників. Більшість існуючих рішень або зосереджені на класичних логістичних моделях, або мають посередницький характер, що знижує ефективність та економічність.

Пропонована система змінює саму модель взаємодії. Її ключова перевага – це прямий контакт між замовником і перевізником, без додаткових комісій

або операторів, які підбирають виконавця вручну. Користувач може самостійно бачити перевізників, які рухаються потрібним маршрутом та мають вільне місце в транспорті та самостійно пропонувати ціну виконавцям.

Таким чином, створена веб-платформа, буде орієнтована на організацію попутних вантажоперевезень, заповнюватиме актуальну нішу на ринку, оптимізує логістичні ресурси та запропонує реальну альтернативу дорогим або неефективним сервісам. Вона врахує потреби малого бізнесу, приватних осіб та перевізників, які прагнуть працювати напряму, без комісій і з максимальною ефективністю.

					ІАЛЦ.045440.004 ПЗ	Арк.
Зм	Арк.	№ докум.	Підп.	Дата		18

2. ВИБІР ТЕХНОЛОГІЙ І ЗАСОБІВ РОЗРОБЛЕННЯ СИСТЕМИ ДЛЯ ОРГАНІЗАЦІЇ СУМІСНИХ ПЕРЕВЕЗЕНЬ ВАНТАЖІВ РІЗНИХ ГАБАРИТІВ

2.1. Формат програмної реалізації

У процесі розроблення комп'ютерної системи організації сумісних перевезень вантажів різних габаритів одним із головних етапів було визначення формату реалізації програмного продукту. Вибір залежав насамперед від функціональних потреб системи, а також від формату її використання з боку основних груп користувачів. Система повинна забезпечувати простий доступ до функціоналу, мінімальні вимоги до апаратного забезпечення, зрозумілу логіку роботи, стабільність та гнучкість у подальшій підтримці і розвитку.

Основними користувачами системи є водії, замовники перевезень та логістичні компанії. Кожна з цих категорій має власні цілі та завдання, які вона виконує в межах платформи.

Для водіїв важливо мати швидкий доступ до інформації про доступні запити на перевезення, можливість оперативно відгукнутися на відповідний запит та знайти оптимальне перевезення. Система повинна бути простою у використанні, інтуїтивно зрозумілою, з чіткою навігацією та логічно організованими розділами.

Замовники користуються системою з метою швидкого розміщення запитів на перевезення вантажів, відстеження статусу заявок та перегляду відгуків водіїв. Для них також важливими є зручність інтерфейсу, швидкість обміну інформацією, доступ до історії перевезень та впорядкована подача інформації.

Логістичні компанії або оператори можуть потребувати більшої кількості функцій, зокрема: перегляд великих обсягів даних, фільтрація маршрутів, доступ до статистики тощо. У межах реалізації даної системи враховано можливість розширення функціоналу в майбутньому, а також підтримку користувачів із різним рівнем доступу.

					ІАЛЦ.045440.004 ПЗ	Арк.
Зм	Арк.	№ докум.	Підп.	Дата		19

З огляду на описані потреби користувачів, а також загальні вимоги до стабільної роботи системи, було обрано формат реалізації у вигляді веб-застосунку. Такий підхід дає змогу забезпечити зручну роботу з системою без необхідності встановлення окремого програмного забезпечення, а також забезпечує швидкий обмін даними та мінімальні вимоги до апаратного забезпечення користувачів. Оновлення системи відбувається автоматично (усі користувачі отримують доступ до однієї й тієї ж версії вебзастосунку, що дозволяє уникнути непорозумінь). Сервіс не буде потребувати інсталяції у пам'ять пристрою користувача [4].

Однією з ключових переваг веб-систем є централізована структура – основна логіка та обробка даних відбувається на стороні сервера, а користувач взаємодіє з інтерфейсом через браузер. Це дозволяє забезпечити єдиний актуальний інтерфейс для всіх користувачів, а також зменшити навантаження на клієнтські пристрої. Крім того, веб-системи забезпечують простоту розгортання, що є важливою перевагою під час запуску перших версій програмного продукту. Завдяки цьому можливо швидко перевірити працездатність концепції, зібрати зворотний зв'язок від користувачів та внести необхідні зміни без складного процесу оновлення окремих інсталяцій. Також веб-реалізація дозволяє легше масштабувати систему в майбутньому. Наприклад, при зростанні кількості користувачів або появі нових функцій можна розширити серверну інфраструктуру, не змінюючи принципів взаємодії на стороні клієнта. Це важливо для подальшого розвитку системи без необхідності повного перепроєктування архітектури.

У межах реалізації обрано саме багатосторінкову структуру веб-застосунку, що дає змогу логічно розділити інтерфейс на окремі функціональні блоки. Такий підхід значно покращує навігацію системою, а також дозволяє впровадити різні рівні доступу. Наприклад, водії та замовники бачать лише ті розділи, які відповідають їхній ролі, тоді як логістичний оператор може мати доступ до ширшого кола функцій та розділів. Це забезпечує гнучкість і безпеку

					ІАЛЦ.045440.004 ПЗ	Арк.
Зм	Арк.	№ докум.	Підп.	Дата		20

роботи з платформою.

2.2 Аналіз обраних інструментів та платформи для створення веб-системи

На етапі планування було розглянуто декілька можливих варіантів мов програмування та платформ для реалізації, які активно використовуються для створення сучасних веб-систем. Серед основних критеріїв вибору були: швидкість розробки, наявність інструментів для побудови інтерфейсу користувача, підтримка баз даних та API, а також можливість зручного масштабування.

Для порівняння розглянемо популярні сучасні мови, які використовуються у веб-програмуванні. Кожна з них має свої переваги та обмеження залежно від типу проєкту. Було розглянуто наступні варіанти:

- JavaScript (та його розширення TypeScript)

JavaScript є найбільш розповсюдженою мовою для клієнтської частини веб-застосунків. У поєднанні з фреймворками на зразок React, Vue або Angular вона дозволяє створювати складні інтерактивні інтерфейси. Однак розробка з використанням JavaScript вимагає досить глибоких знань архітектури застосунків, особливо у випадках реалізації багатосторінкових систем із розмежуванням ролей користувачів. Крім того, повноцінна серверна логіка зазвичай реалізується окремими інструментами (наприклад, Node.js), що ускладнює структуру проєкту.

- Python (з фреймворками Django або Flask)

Python добре підходить для створення серверної частини веб-застосунків. Django, зокрема, надає великий набір вбудованих функцій для обробки даних, автентифікації, роботи з базами даних тощо. Водночас, інтерфейс користувача у таких застосунках обмежений, якщо не використовувати

					ІАЛЦ.045440.004 ПЗ	Арк.
Зм	Арк.	№ докум.	Підп.	Дата		21

JavaScript на фронтенді, що спричиняє додаткові труднощі інтеграції двох частин (backend і frontend).

- PHP (Laravel)

PHP залишається популярним завдяки своїй простоті та великій кількості готових рішень. Використання Laravel забезпечує зручну маршрутизацію, шаблонізатори та інші корисні інструменти. Однак, в умовах деяких сучасних тенденцій, PHP поступається JavaScript-орієнтованим технологіям.

- Java (Spring Boot)

Платформа Java є надзвичайно потужною, але вимагає значного досвіду та ресурсів. Її перевага – масштабованість та висока продуктивність у корпоративних системах. Проте для невеликого MVP чи експериментального стартапу цей підхід є доволі складним.

- Dart (з фреймворком Flutter)

Dart – це сучасна мова програмування, розроблена компанією Google, яка активно використовується для створення кросплатформених застосунків. Її ключова особливість полягає в тому, що вона була спроектована зокрема для роботи з фреймворком Flutter, забезпечуючи максимально ефективну інтеграцію між мовою та засобами побудови інтерфейсу.

З огляду на завдання проєкту – розробка багатосторінкової веб-системи з можливістю масштабування та гнучкого оновлення – було прийнято рішення використовувати мову програмування Dart у поєднанні з фреймворком Flutter. Дана мова програмування має синтаксис, подібний до таких популярних мов, як Java та JavaScript, що спрощує її вивчення та використання для розробки. Водночас Dart підтримує об'єктно-орієнтований підхід, має чітку структуру коду і забезпечує високу швидкість виконання програм.

					ІАЛЦ.045440.004 ПЗ	Арк.
Зм	Арк.	№ докум.	Підп.	Дата		22

Особливу популярність Dart здобула завдяки тому, що вона є основною мовою для фреймворку Flutter – одного з найпопулярніших сучасних інструментів для веб-розробки. Найчастіше її використовують для кросплатформеної розробки – створення застосунків, які працюють на різних операційних системах (Android, IOS, web) з єдиною кодовою базою.

Завдяки поєднанню зручності, гнучкості та високої продуктивності мова Dart стала однією з найпопулярніших для створення динамічних, інтуїтивно зрозумілих і масштабованих застосунків. Її найширше використання пов’язане з фреймворком Flutter, який реалізовано повністю на Dart і який, у свою чергу, лежить в основі платформи Flutter Flow, що була обрана для реалізації поставленої задачі.

Flutter Flow – це хмарна low-code платформа, що дозволяє створювати повноцінні веб і мобільні застосунки з використанням графічного інтерфейсу, з можливістю генерації коду на мові Dart та експорту у форматі Flutter-проекту [5]. Вона поєднує в собі зручність візуального конструювання інтерфейсу з гнучкістю коду, що генерується автоматично, дозволяючи у подальшому розширювати функціонал безпосередньо через Flutter SDK.

Flutter Flow надає розробнику можливість створювати складну бізнес-логіку, налаштовувати взаємодію з базами даних, обробку подій, навігацію між сторінками, автентифікацію користувачів та багато іншого. Таким чином, платформа дозволяє суттєво скоротити час розробки та зосередитись на якісній реалізації продукту.

Однією з ключових причин вибору саме Flutter Flow стала можливість швидкої розробки MVP. У контексті стартапів або інноваційних розробок, де швидке тестування гіпотез має вирішальне значення, платформи типу Flutter Flow дають змогу отримати перші робочі прототипи за лічені дні. За рахунок візуального конструювання інтерфейсів та інтегрованих модулів логіки зменшується потреба в написанні базового коду вручну, що скорочує цикл розробки і виводу продукту на ринок.

Flutter Flow є доступною для широкого кола розробників, а наявність зручного графічного інтерфейсу, шаблонів, інтегрованої документації та прикладів значно спрощує освоєння та розробку продуктів на базі даної платформи. Інтерфейс Flutter Flow зображений на Рисунку 2.1.

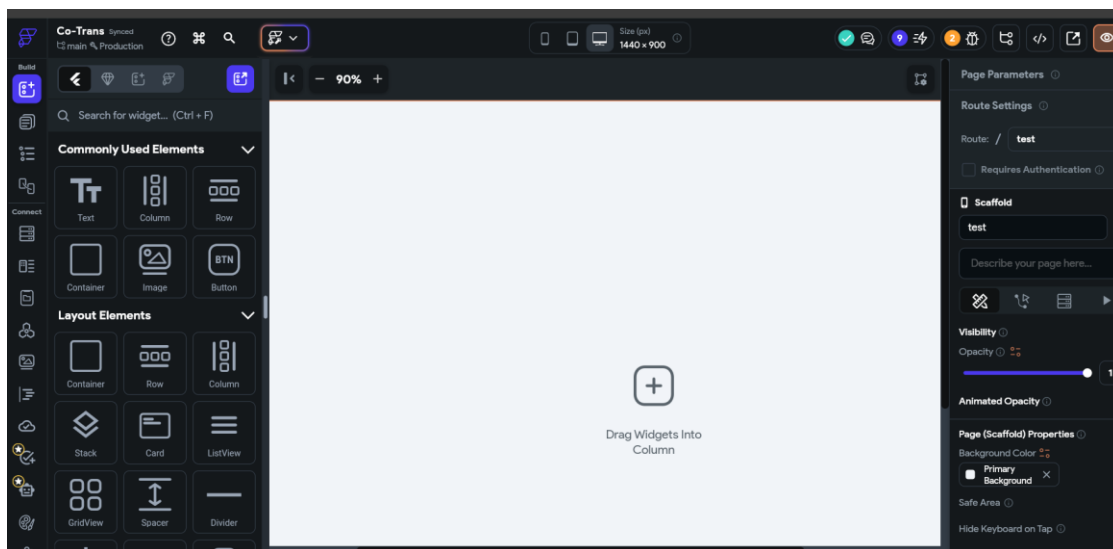


Рисунок 2.1 – Інтерфейс платформи Flutter Flow

Flutter Flow активно використовується для створення широкого спектру застосунків – від простих MVP до повноцінних комерційних продуктів із десятками тисяч користувачів. Платформа підтримується командою, яка має прямий доступ до інфраструктури Flutter та Google Firebase, що забезпечує високу стабільність та інтегрованість. Її популярність серед стартапів, малих команд і незалежних розробників підтверджує ефективність та надійність даного рішення.

2.3 Вибір системи управління базами даних

Не менш важливим компонентом комп'ютерної системи організації сумісних перевезень вантажів є СУБД. Вона виконує функцію центрального сховища інформації, забезпечуючи обмін даними між клієнтською та серверною частинами, зберігання інформації про користувачів, вантажі, маршрути, запити тощо.

У межах клієнт-серверної архітектури саме СУБД відіграє роль бекенд-компонента, що відповідає за збереження, обробку, пошук, фільтрацію та передачу даних. Тобто будь-яка дія користувача, пов'язана з оновленням або переглядом інформації (наприклад, створення нового запиту на перевезення, перегляд відгуків водіїв, історії замовлень), супроводжується взаємодією з базою даних через API або інші інтерфейси.

При виборі СУБД для розроблення вебсистеми для перевезень основними вимогами були:

- Доступність – база даних повинна бути доступною в режимі 24/7, забезпечуючи надійне з'єднання з клієнтськими додатками.
- Масштабованість – можливість обробки зростаючого обсягу даних без суттєвої втрати продуктивності при збільшенні кількості користувачів.
- Швидкодія – обробка запитів у реальному часі, що особливо важливо при роботі із запитами на перевезення та повідомленнями між користувачами.
- Інтеграційна сумісність – легка інтеграція з середовищем розроблення, зокрема з платформою Flutter Flow.
- Безпека – забезпечення захищеного зберігання персональних і службових даних.

Бази поділяються на SQL та NoSQL. У реляційних базах даних, таких як MySQL, PostgreSQL або Oracle, дані зберігаються у форматі таблиць, вони суворо структуровані та пов'язані одна з одною. Вони добре підходять для систем зі стабільною структурою даних і складними транзакціями, але можуть бути менш гнучкими при динамічній зміні моделі даних або великій кількості паралельних запитів із мобільних клієнтів. У свою чергу, нереляційні бази даних (MongoDB, Firebase Firestore, Couchbase тощо) зберігають дані у вигляді документів, які легше масштабувати, змінювати, реплікувати. Їх перевагою є гнучкість структури даних, простота інтеграції з веб та мобільними

застосунками, а також швидке реагування на запити [6]. Детальне порівняння видів баз даних наведено у Таблиці 2.1.

Таблиця 2.1 – Порівняння реляційних та нереляційних баз даних

	SQL	NoSQL
Структура даних	Таблична структура (рядки й стовпці)	Гнучка структура: документи, ключ-значення, граfi тощо
Схема	Жорстко визначена, потребує попереднього опису	Динамічна, дозволяє змінювати структуру в процесі
Підтримка транзакцій	Повна	Обмежена
Продуктивність при великих обсягах	Може знижуватись із ростом обсягу даних	Висока продуктивність на великих обсягах і при частому читанні
Типові сценарії використання	Традиційні бізнес-додатки, бухгалтерія, CRM	Веб-сервіси, мобільні застосунки, IoT
Підтримка складних запитів	Висока, мова SQL підтримує складну аналітику	Обмежена, залежно від типу NoSQL
Гнучкість у зберіганні даних	Низька – структура фіксована	Висока – підтримка вкладених структур, масивів тощо
Приклади СУБД	MySQL, PostgreSQL, Microsoft SQL Server	Firebase Firestore, MongoDB, Couchbase

З урахуванням архітектури застосунку, використання Flutter Flow та специфіки даних, було вирішено обрати нереляційну СУБД Firebase Firestore.

Firebase Firestore – це сучасна документо-орієнтована СУБД також від компанії Google, яка є частиною хмарної платформи Firebase [7]. Однією з ключових переваг Firestore є її швидкість: усі зміни в базі можуть миттєво передаватися на клієнтську частину (наприклад, зміна статусу запиту на

перевезення або нові відгуки водіїв оновлюються та відображаються в режимі реального часу).

Однією з додаткових переваг вибору Firebase Firestore стала зручна інтеграція з платформою Flutter Flow, що суттєво спрощує підключення та взаємодію з базою даних на етапі розробки. Завдяки готовим механізмам зв'язку між інтерфейсом застосунку та структурою Firestore, можливо швидко реалізовувати основні функції роботи з даними – збереження, оновлення, читання та фільтрацію. Така сумісність дозволяє значно зменшити витрати часу на технічну інтеграцію, забезпечуючи при цьому стабільну роботу системи. А зручний інтерфейс даної СУБД (Рисунок 2.2) та додаткові функції на кшталт перегляду статистики та створення кастомних правил забезпечують зручність її використання.

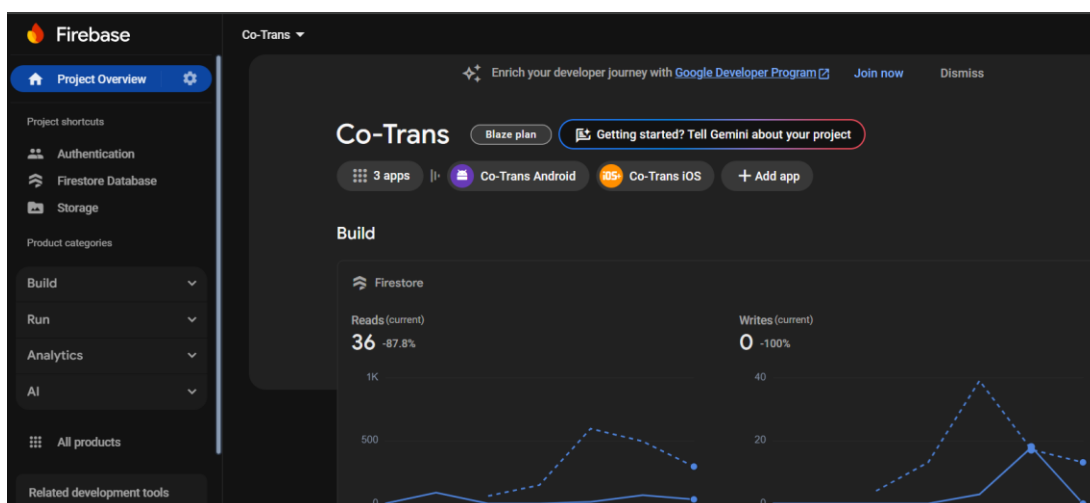


Рисунок 2.2 – Інтерфейс Firebase

Firestore використовує ієрархічну модель даних, що складається з колекцій та документів. Колекція – це набір документів, кожен із яких містить дані у форматі ключ-значення. Документи можуть містити вкладені колекції, що дозволяє створювати складні структури без втрати продуктивності.

Firestore має гнучку систему керування доступом на основі ролей та автентифікації. Для захисту даних у системі реалізовано:

- Firebase Authentication – сервіс для реєстрації, входу в систему та автентифікації користувачів (через email, Google, інші платформи).
- Правила безпеки Firestore – спеціальні умови, що визначають, хто і які дії може виконувати з конкретними колекціями та документами.
- Захист на стороні сервера – контроль доступу до чутливих даних виконується централізовано, що виключає можливість прямого втручання з боку клієнта.

Усі ці заходи дозволяють гарантувати безпечне зберігання даних та надійний захист інформації від несанкціонованого доступу.

2.4 Цільова спрямованість дизайну та архітектури системи

Під час розроблення комп'ютерної системи організації сумісних перевезень вантажів важливу роль також відіграв вибір концепції дизайну користувацького інтерфейсу та загальної архітектури застосунку. У контексті даного проєкту ключовим завданням було створення не універсального, а спеціалізованого цифрового рішення, орієнтованого на вирішення конкретної логістичної проблеми. Саме тому перевага була надана мінімалістичному, інтуїтивно зрозумілому дизайну, а також чітко визначеній функціональній структурі, яка відповідає виключно потребам основних категорій користувачів – водіїв та замовників.

Основним аргументом на користь мінімалізму при виборі стилю дизайну стала необхідність забезпечення простого, зрозумілого та передбачуваного середовища взаємодії для кінцевого користувача. У випадку з мобільними та веб-сервісами, орієнтованими на виконання прикладних дій у польових умовах (наприклад, під час водіння або на етапі завантаження вантажу), важливу роль відіграє максимальне скорочення зайвих візуальних елементів, переходів і взаємодій.

Мінімалізм є однією з провідних тенденцій у сучасному веб-дизайні. Світові лідери в галузі цифрових сервісів (Google, Apple, Microsoft) активно впроваджують принципи «flat design», «material design» та «functional simplicity», що орієнтовані на досягнення максимальної ефективності за допомогою лаконічних візуальних рішень [8].

З огляду на специфіку взаємодії користувачів із системою в контексті прикладного використання, важливим аспектом проектування дизайну стала доступність інтерфейсу для широкого кола користувачів з різним рівнем цифрової грамотності. Це передбачає застосування принципів UX-дизайну, що базуються на моделі HCD. Використання даної моделі означає проектування з урахуванням потреб, обмежень і поведінкових характеристик кінцевого користувача на всіх етапах створення продукту [9]. Застосування цієї моделі дозволило досягти вищого рівня інтуїтивності та зручності, що є критично важливим у системах, де користувачам необхідно швидко орієнтуватися та діяти без тривалого навчання чи інструктажу.

Застосування цих принципів у даному проєкті проявляється у використанні обмеженої палітри кольорів, контрастного шрифту, простої ієрархії інтерфейсних елементів, а також чіткої структури екранів. Такий підхід сприяє не лише зручності у використанні, але й загальному відчуттю професійності та надійності цифрового продукту.

Дана система орієнтована на вирішення однієї чітко окресленої задачі – ефективну організацію спільних перевезень вантажів різних габаритів. Всі елементи інтерфейсу та навігаційна логіка побудовані з урахуванням конкретних потреб замовників та водіїв. Такий цільовий функціональний мінімалізм дозволяє зменшити час ознайомлення нових користувачів, прискорює процес прийняття рішень і загалом підвищує ефективність використання системи.

					ІАЛЦ.045440.004 ПЗ	Арк.
Зм	Арк.	№ докум.	Підп.	Дата		29

3. РЕАЛІЗАЦІЯ КОМП'ЮТЕРНОЇ СИСТЕМИ ОРГАНІЗАЦІЇ СУМІСНИХ ПЕРЕВЕЗЕНЬ ВАНТАЖІВ РІЗНИХ ГАБАРИТІВ

3.1. Характеристика структури та вимоги до системи

Реалізована комп'ютерна система організації сумісних перевезень вантажів різних габаритів побудована за принципом клієнт-серверної взаємодії з використанням хмарної платформи Firebase. Така архітектурна модель дозволяє забезпечити надійний розподіл обов'язків між компонентами, масштабованість, централізоване зберігання даних та ефективну синхронізацію в реальному часі.

Загальна структура системи складається з двох основних частин:

- Клієнтська частина – реалізована як вебзастосунок за допомогою фреймворку Flutter, що надає доступ до інтерфейсу користувача через веббраузер. Цей застосунок виконує функції введення та виведення даних, обробки дій користувачів (водіїв та замовників) і формування запитів до серверної частини.
- Серверна частина – базується на інфраструктурі Firebase, яка об'єднує три основні сервіси:
 1. Firestore Database – основне середовище зберігання структурованих даних, таких як облікові записи користувачів, замовлення на перевезення, маршрути, відгуки водіїв.
 2. Storage – використовується для зберігання додаткових файлів, зокрема зображень профілів користувачів та зображень для наповнення окремих сторінок застосунку.
 3. Authentication – сервіс, відповідальний за автентифікацію користувачів, тобто за процеси реєстрації та входу користувачів у систему.

Взаємодія між клієнтом і сервером відбувається через захищені HTTPS-запити, що гарантує конфіденційність та цілісність переданих даних. Клієнтський застосунок надсилає запити до бази даних чи інших сервісів Firebase, отримує відповіді та відображає їх у відповідному вигляді користувачеві. Всі дії – створення, пошук, оновлення та видалення об'єктів — відбуваються в режимі реального часу.

Схематичне зображення структури системи наведено на Рисунку 3.1. Воно ілюструє взаємодію між вебзастосунком на стороні клієнта та мікросервісами Firebase, які забезпечують повноцінну роботу застосунку без потреби в окремому сервері.

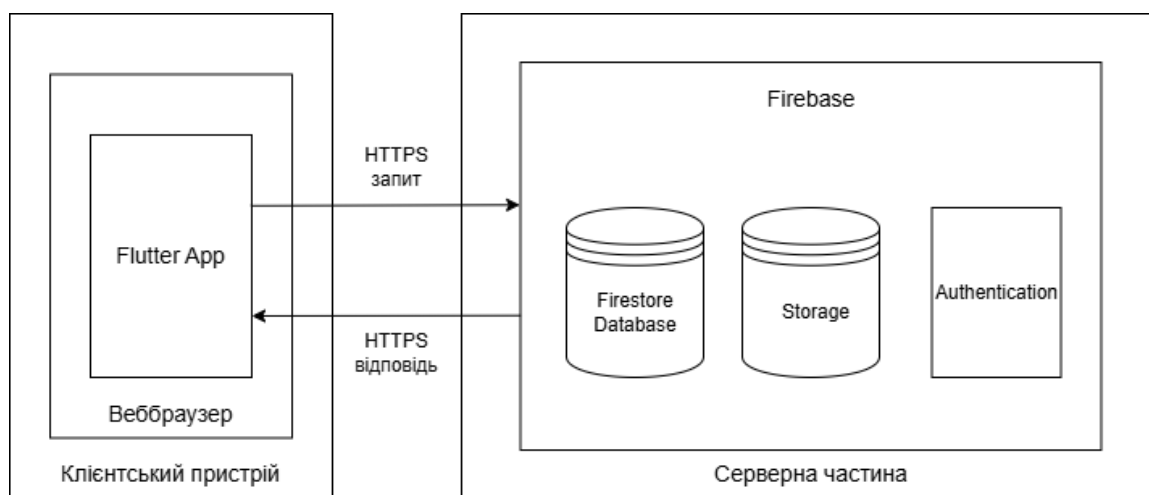


Рисунок 3.1 – Структура системи

Дана система повинна відповідати таким функціональним та нефункціональним вимогам:

- Реєстрація та авторизація користувачів.
- Заповнення даних про користувача, додавання транспорту.
- Можливість перегляду та редагування даних в особистому кабінеті.
- Кожен користувач може виступати в ролі замовника та перевізника (окрім своїх замовлень).
- Створення запитів на перевезення вантажів.

- Можливість пропозиції вартості перевезення.
- Додаткові пункти-уточнення: крихкий вантаж та термінове замовлення.
- Можливість прокладати маршрут, переглядати доступні за ним перевезення та подавати відгуки про готовність виконати доставку.
- Можливість переглядати опубліковані замовлення, змінювати їхній статус, переглядати відгуки водіїв.
- Доступність актуального інтерфейсу для всіх користувачів через різні браузери.
- Відповідність принципам UX/UI дизайну – інтуїтивно зрозумілий інтерфейс, адаптивна верстка.
- Безпечна обробка даних користувачів та відповідність загальним вимогам захисту інформації.
- Швидке реагування інтерфейсу та підтримка роботи в режимі реального часу.
- Можливість масштабування системи у випадку зростання кількості користувачів або запитів.

Функціональні можливості та інтерфейс застосунку повинні бути чітко орієнтовані на основний сценарій роботи користувача. Це означає, що всі елементи інтерфейсу мають сприяти швидкому й інтуїтивно зрозумілому виконанню ключових дій. Схема типового сценарію роботи користувача у застосунку проілюстрована на Рисунку 3.2.

					ІАЛЦ.045440.004 ПЗ	Арк.
Зм	Арк.	№ докум.	Підп.	Дата		32

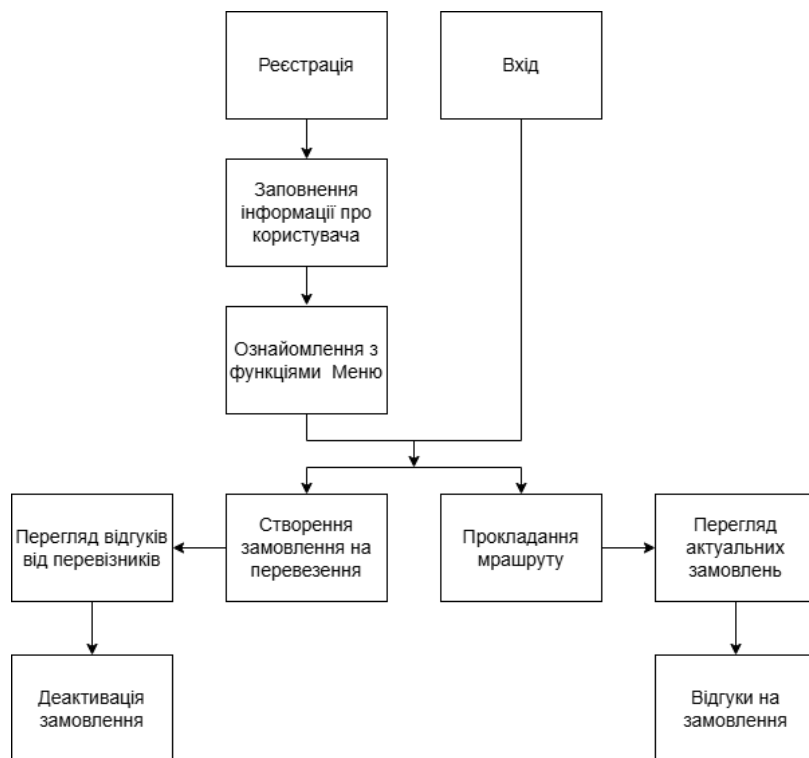


Рисунок 3.2 – Схема типового сценарію роботи користувача

3.2. Розробка клієнтської частини

Клієнтська частина комп'ютерної системи організації сумісних перевезень вантажів різних габаритів виконує роль основного інтерфейсу взаємодії кінцевого користувача із сервісом. Її реалізація була здійснена з використанням фреймворку Flutter.

Вона забезпечує введення даних через інтерактивні форми, зокрема під час реєстрації, створення запитів на перевезення, заповнення інформації про користувача, редагування профілю тощо. Дані, що вводяться, проходять базову валідацію ще на стороні клієнта, що дозволяє зменшити кількість некоректно введених даних та не пропустити обов'язкові для заповнення поля.

Виведення даних реалізовано у вигляді списків або окремих блоків з інформацією, які динамічно формуються на основі даних із бази. Сюди належать, зокрема, актуальні замовлення на перевезення, інформація про профіль користувача, відгуки водіїв тощо.

Обробка дій користувачів полягає у відслідковуванні кліків та взаємодії з кнопками. Це означає, що коли користувач натискаючи кнопку бажає виконати певну дію – застосунок фіксує цю дію, перевіряє поля для введення (за наявності) за допомогою форм валідації. перетворює дані у відповідний формат і запускає процедуру відправлення їх до серверної частини.

Щодо формування запитів до серверної частини, то вебзастосунок взаємодіє з базою даних Firestore через вбудований Firebase SDK який формує HTTPS запити до Firestore API [10]. Ці запити охоплюють:

- Зчитування наявних записів
- Створення нових записів
- Оновлення або видалення записів

Усі ці операції відбуваються асинхронно, що дозволяє не переривати взаємодію користувача з інтерфейсом у процесі обробки даних. В результаті, клієнтська частина ефективно виконує функції введення, обробки та виведення інформації, забезпечуючи безперервну й логічно зрозумілу роботу всієї системи.

У рамках реалізації вебзастосунку було створено 16 екранів, кожен із яких виконує окрему роль у процесі користування системою. Серед основних – екран авторизації, сторінка меню, екран створення нового замовлення на перевезення, сторінка прокладання маршруту, профіль користувача, сторінка для перегляду створених замовлень та відгуків на них а також допоміжні інтерфейси для редагування, фільтрації чи перегляду деталей конкретних оголошень. Кожен з них реалізовано у Flutter як окремий екран – структурна одиниця інтерфейсу, що відповідає певній логіці взаємодії з користувачем. На рисунку 3.3 зображено схему взаємозв'язків екранів застосунку.

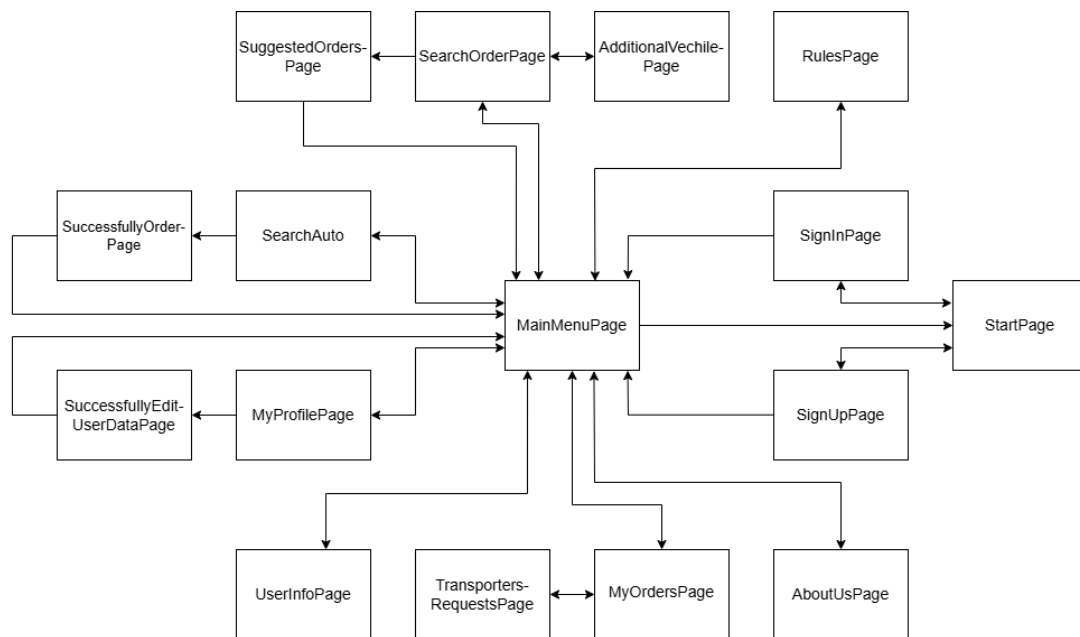


Рисунок 3.3 – Схематичне зображення взаємозв'язків екранів системи

Кожен екран, у свою чергу, формується з віджетів – базових будівельних блоків інтерфейсу у Flutter, які відповідають за відображення тексту, кнопок, списків, форм введення, зображень, інтерактивних елементів тощо. FlutterFlow, як візуальний конструктор, дозволяє компоувати ці віджети у вигляді деревоподібної структури, що визначає порядок, вкладеність і логіку елементів на сторінці. Наприклад, список перевезень реалізовано як список контейнерів з динамічним підвантаженням записів з бази даних Firestore Database, а форми створення запиту – як набір TextField та Dropdown-віджетів, з прив'язкою до відповідних полів у базі.

Кожна сторінка має власну логіку переходів на наступні або попередні екрани, обробку натискань кнопок, валідацію введених даних, відображення повідомлень про помилки чи підтвердження. Вся логіка подій задається у вигляді Actions – подій, які виконуються у відповідь на дії користувача, наприклад, надсилання форми, зміна елемента або навігація між екранами. Ці події також можуть включати виклики функцій Firebase (запис в базу даних, вхід в обліковий запис, завантаження файлу тощо).

Завдяки подібній структурі застосунок є модульним і легко редагованим. Кожен екран, набір віджетів та логіка дій є незалежними й можуть бути змінені або розширені без впливу на інші частини системи.

Для прикладу розглянемо блок коду, який формує кнопку «Зберегти» на екрані `UserInfoPage` та містить фрагмент коду, який виконується при натисканні кнопки. Він відповідає за відображення кнопки згідно заданих параметрів, а при натисканні на неї – виконує перевірку та зберігання інформації про користувача:

```
Align(  
  alignment: AlignmentDirectional(0, 0),  
  child: Padding(  
    padding: EdgeInsetsDirectional.fromSTEB(0, 20, 0, 0),  
    child: FFButtonWidget(  
      onPressed: () async {  
        if (_model.formKey.currentState == null ||  
            !_model.formKey.currentState!.validate()) {  
          return;  
        }  
        await currentUserReference!.update(createUsersRecordData(  
          displayName: _model.textController1.text,  
          sex: _model.dropDownValue,  
          phoneNumber: _model.textController2.text,  
          vechileName: _model.textController3.text,  
          photoUrl: _model.uploadedFileUrl,  
        ));  
        context.pushNamed(MainMenuPageWidget.routeName);  
      }  
    )  
  )  
);
```

```

    },
    text: 'Зберегти',
    options: FFButtonOptions(
      width: 450,
      height: 60,
      padding: EdgeInsetsDirectional.fromSTEB(16, 0, 16, 0),
      iconPadding: EdgeInsetsDirectional.fromSTEB(0, 0, 0, 0),
      color: Color(0xFF25DA6C),
      textStyle: FlutterFlowTheme.of(context).displayLarge.override(
        font: GoogleFonts.interTight(
          fontWeight:
FlutterFlowTheme.of(context).displayLarge.fontWeight,
          fontStyle:
FlutterFlowTheme.of(context).displayLarge.fontStyle,
        ),
        letterSpacing: 0.0,
        fontWeight:
          FlutterFlowTheme.of(context).displayLarge.fontWeight,
        fontStyle:
FlutterFlowTheme.of(context).displayLarge.fontStyle,
      ),
      elevation: 0,
      borderRadius: BorderRadius.circular(8),
    ),
  ),
),
),

```

Зм	Арк.	№ докум.	Підп.	Дата

ІАЛІЦ.045440.004 ПЗ

Арк.
37

)

З наведеного вище блоку коду ми можемо бачити, що кнопка розміщена на екрані за допомогою віджету `Align` із центруванням елемента по горизонталі. До кнопки застосовано вертикальний відступ за допомогою віджета `Padding`, що забезпечує візуальну відокремленість елемента від інших компонентів інтерфейсу.

Кнопка оформлена відповідно до загального стилю застосунку: має фіксовані розміри (450×60 пікселів), зелений колір (#25DA6C), скруглені кути та текстове наповнення «Зберегти». Текст стилізується відповідно до активної теми інтерфейсу через `FlutterFlowTheme`, що забезпечує візуальну узгодженість елементів у межах усього застосунку.

Основний функціональний елемент – це `FFButtonWidget`, що є стилізованим віджетом кнопки. При натисканні на кнопку спрацьовує асинхронна функція, яка перевіряє валідність заповненої користувачем форми. У випадку, якщо форма не створена або деякі поля не відповідають вимогам валідації, подальше виконання функції переривається. Якщо ж перевірка пройдена успішно, відбувається оновлення відповідного документа поточного користувача у базі даних `Firestore`. Оновлення здійснюється шляхом виклику методу `update()`, який приймає параметри, сформовані на основі введених у формі значень: ім'я користувача, стать, номер телефону, назва транспортного засобу та URL завантаженого зображення. Значення для кожного з полів отримуються безпосередньо з відповідних контролерів введення.

Після успішного збереження змін у базі даних, виконується навігація до головної сторінки застосунку – `MainMenuPage`, що означає повернення користувача до основного інтерфейсу взаємодії після оновлення даних свого профілю.

Тепер проаналізуємо фрагмент коду, який виконується при натисканні кнопки «Опублікувати замовлення» на сторінці `SearchAuto`:

```

onPressed: () async {
  await UsersOrdersRecord.collection
    .doc()
    .set(createUsersOrdersRecordData(
      orderName: _model.textController1.text,
      length: double.tryParse(_model.textController2.text),
      width: double.tryParse(_model.textController4.text),
      height: double.tryParse(_model.textController3.text),
      weight: double.tryParse(_model.textController5.text),
      orderFrom: _model.dropDownValue1,
      orderWhere: _model.dropDownValue2,
      orderPrice: double.tryParse(_model.textController6.text),
      fragile: _model.checkboxValue1,
      urgent: _model.checkboxValue2,
      uid: currentUserReference,
      status: true,
      creationTime: getCurrentTimestamp,
    ));
  context.pushNamed(SuccessfullyOrderPageWidget.routeName);
}

```

При натисканні на кнопку виконується дана асинхронна функція, що створює новий запис у колекції `users_orders` у базі даних `Firestore Database`. Спочатку викликається метод `doc()` без аргументів, що автоматично генерує унікальний ідентифікатор для нового документа. Далі використовується метод `set()`, який записує дані, сформовані функцією `createUsersOrdersRecordData()`. До структури запису входять усі параметри, які користувач заповнив у формі: текстове поле з назвою замовлення, числові значення довжини, ширини,

					ІАЛЦ.045440.004 ПЗ	Арк.
Зм	Арк.	№ докум.	Підп.	Дата		39

висоти, ваги та вартості, обрані з випадючих списків міста відправлення та призначення, логічні значення щодо крихкості та терміновості вантажу, ідентифікатор поточного користувача, статус замовлення та час його створення.

Після успішного створення нового документа застосунок автоматично виконує навігацію на сторінку `SuccessfullyOrderPage`, яка інформує користувача про те, що замовлення було успішно опубліковане. Таким чином, цей код забезпечує цикл обробки та збереження даних замовлення у `Firestore Database`, а також завершує дію на інтерфейсному рівні переходом до екрана підтвердження успішної публікації замовлення.

Також проаналізуємо фрагмент коду, який реалізує фотографію у профілі користувача на сторінці меню:

```
if (!functions.emptyPhoto(currentUserPhoto))  
  
  Padding(  
  
    padding: EdgeInsetsDirectional.fromSTEB(40, 40, 0, 0),  
  
    child: AuthUserStreamWidget(  
  
      builder: (context) => Container(  
  
        width: 100,  
  
        height: 100,  
  
        clipBehavior: Clip.antiAlias,  
  
        decoration: BoxDecoration(  
  
          shape: BoxShape.circle,  
  
        ),  
  
        child: Image.network(  
  
          currentUserPhoto,  
  
          fit: BoxFit.cover,  
  
        ),  
  
      ),  
  
    ),  
  
  ),  
  
),
```

					ІАЛЦ.045440.004 ПЗ	Арк.
Зм	Арк.	№ докум.	Підп.	Дата		40

) ,
) ,
)

У цьому фрагменті коду реалізовано відображення фото профілю користувача у вигляді кола. Виконується перевірка: чи змінна `currentUserPhoto` не є порожньою – для цього використовується користувацька функція `functions.emptyPhoto()`. Якщо користувач завантажив фото, тоді запускається побудова відповідного інтерфейсного елемента.

Фото обгорнуто у віджет `Padding`, який додає відступи зліва і зверху, завдяки чому зображення знаходиться на відстані до меж екрану. Основна візуальна частина реалізована за допомогою `Container`, що має фіксовані розміри (100×100 пікселів), округлення у формі кола та властивість `clipBehavior: Clip.antiAlias`, яка обрізає зображення по межах кола з плавними краями.

Вміст контейнера – це мережеве зображення, яке завантажується за посиланням, що зберігається у змінній `currentUserPhoto`. Зображення масштабоване відповідно до розмірів контейнера через параметр `fit: BoxFit.cover`, що дозволяє заповнити весь доступний простір без спотворення пропорцій.

3.3. Розробка серверної частини та взаємодія з базою даних

Розробка серверної частини та взаємодія з базою даних у межах комп'ютерної системи організації сумісних перевезень вантажів реалізована за допомогою хмарної платформи `Firebase`, яка забезпечує централізоване зберігання даних, авторизацію користувачів і обробку запитів. У даному проєкті не використовується окрема серверна логіка в традиційному розумінні (наприклад, через `Node.js` або `Django`), натомість функціональність досягається через прямий доступ до бази даних за допомогою вбудованих інструментів `Firebase SDK`, що інтегрується через `Flutter`.

					ІАЛЦ.045440.004 ПЗ	Арк.
Зм	Арк.	№ докум.	Підп.	Дата		41

Ключовим компонентом бази даних є Firestore – NoSQL база даних, яка дозволяє зберігати дані у вигляді колекцій та документів. Всі користувацькі дані зберігаються у відповідній структурі, а операції зчитування, додавання, оновлення чи видалення записів реалізуються через вбудований Firebase SDK який формує HTTPS запити до Firestore API. Для прикладу, проаналізуємо фрагмент блоку коду, який реалізує отримання з бази даних документів, що представляють собою відгуки водіїв на замовлення:

```
await UsersTripsRecord.collection.doc().set({  
  ...  
});
```

В наведеному вище фрагменті `.doc()` це метод класу `CollectionReference` з Firebase SDK, який створює новий документ. `set({})` це також метод класу `DocumentReference` з Firebase SDK. Він записує дані у документ в Firestore, саме цей метод запускає запис даних до хмари. За допомогою такого типу вбудованих методів ми можемо формувати запити до бази даних та виконувати над нею необхідні дії.

Авторизація в системі реалізована за допомогою сервісу `Firebase Authentication`, який дозволяє користувачам безпечно входити в обліковий запис за допомогою електронної пошти та пароля. Після успішної авторизації користувач отримує унікальний ідентифікатор, який використовується для персоналізації даних у базі. Завдяки централізованій системі авторизації забезпечується контроль доступу до даних і можливість відображення лише релевантної інформації для кожного користувача.

Уся взаємодія з базою даних здійснюється за захищеним з'єднанням через протокол HTTPS. Firebase SDK забезпечує авторизований доступ, керуючи безпекою через правила доступу до колекцій. Завдяки такому підходу серверна частина системи стає максимально простою для розгортання та підтримки, адже обчислювальна логіка розподілена між клієнтом і хмарними сервісами, а

					ІАЛЦ.045440.004 ПЗ	Арк.
Зм	Арк.	№ докум.	Підп.	Дата		42

інтеграція здійснюється без проміжних API або окремого серверного програмного забезпечення. Такий підхід особливо ефективний для швидкої розробки MVP або застосунків.

Для роботи застосунку загалом було створено 4 колекції даних: users, users_orders, users_trips та orders_transporters. Колекція users зберігає особисті дані користувачів і кожен документ містить унікальний ідентифікатор користувача, який допомагає створювати взаємозв'язки між полями документів інших колекцій з документами користувачів. Колекція users_orders зберігає документи з замовленнями: кожен документ представляє конкретне замовлення на перевезення. users_trips містить документи, які представляють подорожі перевізників: унікальний ідентифікатор користувача та маршрут (список міст), за яким буде пролягати поїздка. Колекція orders_transporters зберігає документи, які представляють відгуки перевізників на замовлення. Схема взаємозв'язків між колекціями бази даних зображена на Рисунку 3.4.

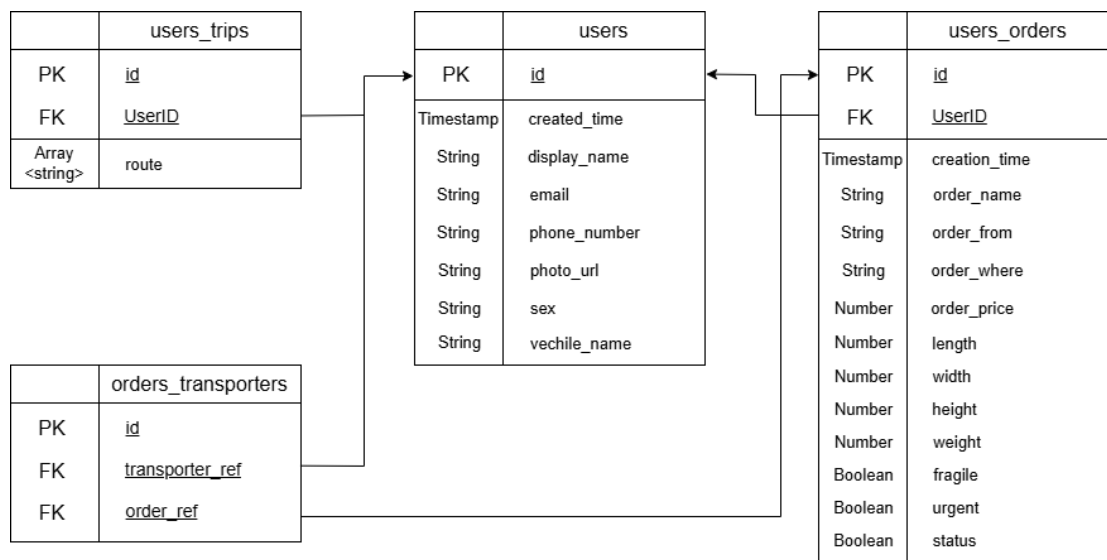


Рисунок 3.4 - Схема взаємозв'язків між колекціями бази даних

3.4. Реалізація пошуку перевезень за маршрутом

Механізм пошуку замовлень на перевезення за прокладеним маршрутом реалізовано на стороні клієнтського застосунку та тісно інтегровано з базою даних Firestore, що надає можливість оперативного отримання відповідної інформації.

Для пошуку замовлень користувач повинен перейти зі сторінки меню на сторінку SearchOrderPage за допомогою натискання кнопки «Пошук замовлень». При переході на сторінку SearchOrderPage, користувач повинен по черзі в точному порядку вибирати міста, через які буде пролягати його поїздка, з випадаючого списку, де будуть знаходитися запропоновані міста. Після вибору кожного міста необхідно натискати на кнопку «Додати», що знаходиться під випадаючим списком. При натисканні даної кнопки місто додається до списку, а складений список відображається в полі нижче і динамічно оновлюється. Приклад відображення маршруту зображено на Рисунку 3.5. Для зручності, нижче на сторінці, користувачеві запропонована до перегляду карта України, за допомогою якої користувач може швидше та точніше будувати свій маршрут у застосунку.

Рисунок 3.5 – Приклад динамічного відображення маршруту

Коли маршрут буде повністю складений – користувач повинен натиснути на кнопку «Знайти замовлення» для переходу на сторінку SuggestedOrdersPage, де будуть відображатися актуальні для нього замовлення.

Маршрут, який прокладає перевізник, зберігається у вигляді впорядкованого списку міст у відповідному документі колекції users_trips в полі route у вигляді масиву елементів типу string. Після збереження цього маршруту у базу даних, система здійснює порівняння актуальних замовлень з вказаним напрямком руху. Пошук виконується за допомогою користувацької функції getSuggestedOrders, код даної функції наведений нижче:

```
List<UsersOrdersRecord>? getSuggestedOrders(  
    List<UsersOrdersRecord>? entryOrdersCollection,  
    List<String>? usersRoute,  
) {  
    if (entryOrdersCollection == null || usersRoute == null) {  
        return [];  
    }  
  
    List<UsersOrdersRecord> matchingOrders = [];  
  
    for (var order in entryOrdersCollection) {  
        // Перевірка статусу  
        if (order.status == true) {  
            String? from = order.orderFrom;  
            String? to = order.orderWhere;  
  
            // Перевірка, чи є ці міста в маршруті  
            if (from != null && to != null) {  
                int fromIndex = usersRoute.indexOf(from);  
                int toIndex = usersRoute.indexOf(to);  
  
                // Перевірка послідовності міст у маршруті  
                if (fromIndex != -1 && toIndex != -1 && fromIndex < toIndex) {  
                    matchingOrders.add(order);  
                }  
            }  
        }  
    }  
}
```

Зм	Арк.	№ докум.	Підп.	Дата

ІАЛЦ.045440.004 ПЗ

Арк.
45

```

    }
    }
    }
}

return matchingOrders;
}

```

З наведеного фрагменту коду ми бачимо, що вона приймає два параметри: список усіх наявних замовлень (`entryOrdersCollection`) та список міст, через які пролягає маршрут користувача (`usersRoute`). Якщо будь-який із цих параметрів є порожнім або відсутнім, функція одразу повертає порожній список, що запобігає помилкам виконання.

Далі відбувається перебір кожного замовлення зі списку. Спочатку функція перевіряє, чи є замовлення активним, тобто чи має воно значення `true` в полі `status`. Лише активні замовлення потрапляють до подальшого аналізу. Для кожного замовлення зчитуються міста відправлення (`orderFrom`) та призначення (`orderWhere`). Якщо обидва ці значення присутні, функція перевіряє, чи входять вони до маршруту перевізника, та чи розташоване місто відправлення перед містом призначення у межах цього маршруту. Це дозволяє гарантувати, що перевізник дійсно проїжджатиме через обидва міста в правильній послідовності.

Якщо всі ці умови виконуються, замовлення додається до результатного списку. Таким чином, функція формує відфільтрований список замовлень, який відповідає маршруту перевізника.

З технічного боку, на сторінці пошуку замовлень, користувач заповнює список міст маршруту, що представлений змінною `RouteCities`, а при натисканні кнопки «Знайти замовлення», виконується функція `getSuggestedOrders`, в ході виконання якої застосунок формує запит до бази даних `Firestore Database` з використанням фільтрації (`where`) за полями `status`, `orderFrom` та `orderWhere`. Отримані дані порівнюються з маршрутом, збереженим у

FFAppState().RouteCities. Після виконання запиту, відповідні документи передаються на візуалізацію у вигляді списку актуальних для перевезення замовлень, де вказано основну інформацію про кожне замовлення, аби перевізник міг обрати ті, які йому найбільше підходять за іншими параметрами.

					ІАЛЦ.045440.004 ПЗ	Арк.
Зм	Арк.	№ докум.	Підп.	Дата		47

4. ОГЛЯД ТА ТЕСТУВАННЯ РЕАЛІЗОВАНОЇ СИСТЕМИ

4.1. Огляд інтерфейсу та тестування системи

При запуску системи користувач потрапляє на початкову сторінку, де можна пройти реєстрацію нового користувача або здійснити вхід у наявний профіль (Рисунок 4.1).

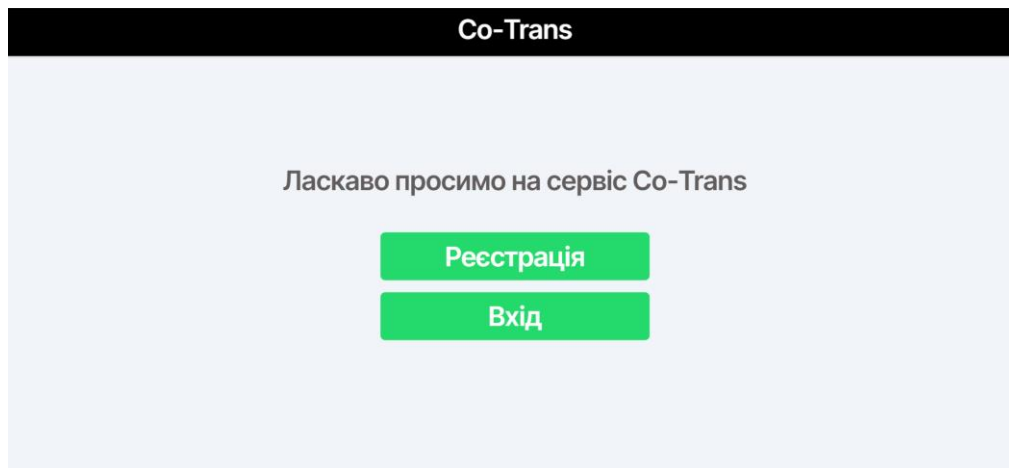


Рисунок 4.1 – Початкова сторінка

При натисканні на кнопку «Реєстрація», користувач потрапляє на сторінку реєстрації, де необхідно ввести email, 8-ми значний пароль та продублювати пароль (Рисунок 4.2).

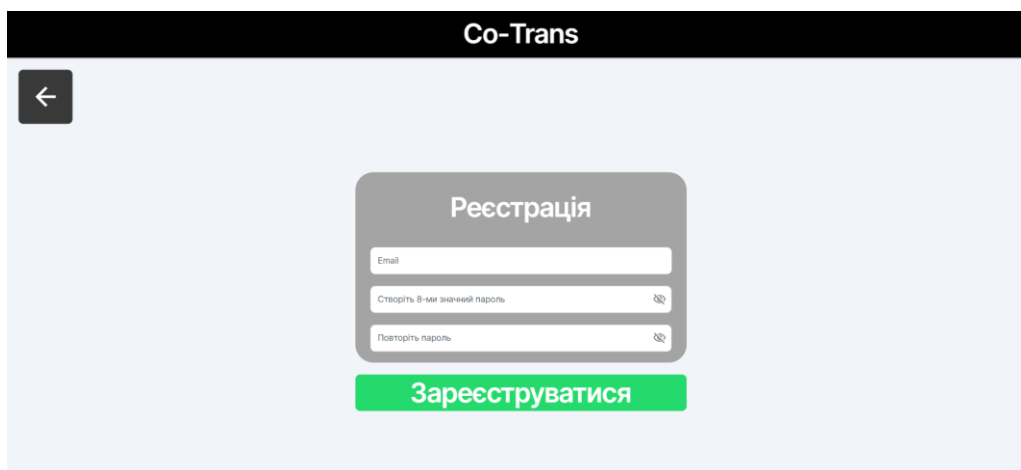


Рисунок 4.2 – Сторінка реєстрації в системі

Зм	Арк.	№ докум.	Підп.	Дата

ІАЛЦ.045440.004 ПЗ

Арк.
48

При натисканні кнопки «Вхід» на початковому екрані, користувач потрапляє на сторінку входу, де потрібно ввести email та пароль для входу в наявний профіль (Рисунок 4.3).

Рисунок 4.3 – Сторінка входу в систему

При автоматичному вході в систему після реєстрації (при першому вході), відкривається форма, в якій необхідно заповнити ряд даних користувача. Деякі поля є необов'язковими для заповнення, але при спробі зберегти дані, не заповнивши хоча б одне з обов'язкових полів – система повідомить про необхідність заповнення (Рисунок 4.4).

Рисунок 4.4 – Сторінка для збору додаткової інформації про користувача

Після коректного заповнення форми та натискання кнопки «Зберегти» (або при звичайному вході в існуючий профіль з раніше заповненою даною формою) – відкривається сторінка меню, де представлений функціонал платформи (Рисунок 4.5).

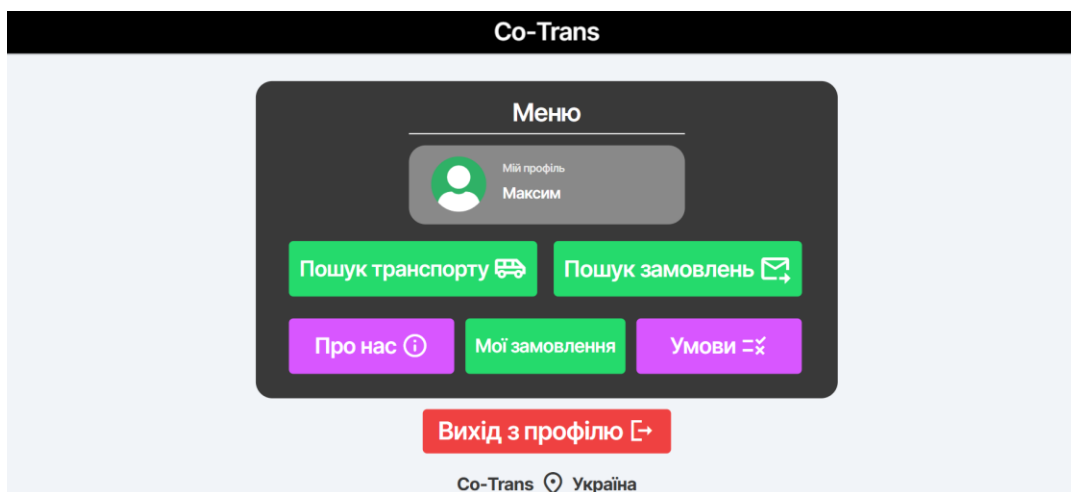


Рисунок 4.5 – Інтерфейс сторінки меню

У верхній частині панелі меню знаходиться розділ «Мій профіль». При переході на нього користувач може відредагувати внесені раніше особисті дані (зміна значень полів, додавання або зміна фото профілю)(Рисунок 4.6).

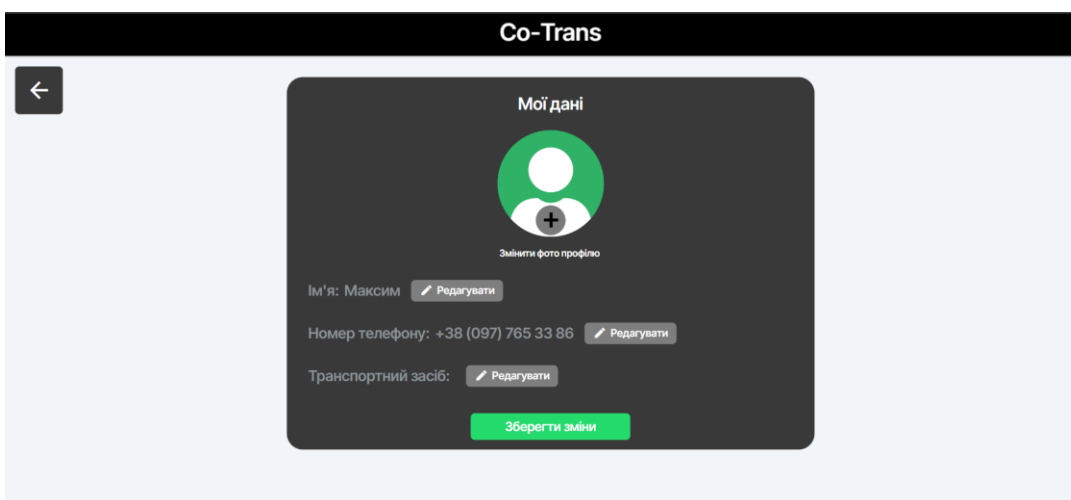


Рисунок 4.6 – Сторінка профілю користувача

Зм	Арк.	№ докум.	Підп.	Дата

ІАЛЦ.045440.004 ПЗ

Арк.
50

При натисканні на кнопки «Редагувати», з'являються активні форми для введення нових значень на місці відповідних полів. При натисканні на емблему «Змінити фото профілю», відкривається вікно файлового провідника, де можна обрати потрібну фотографію (Рисунок 4.7).

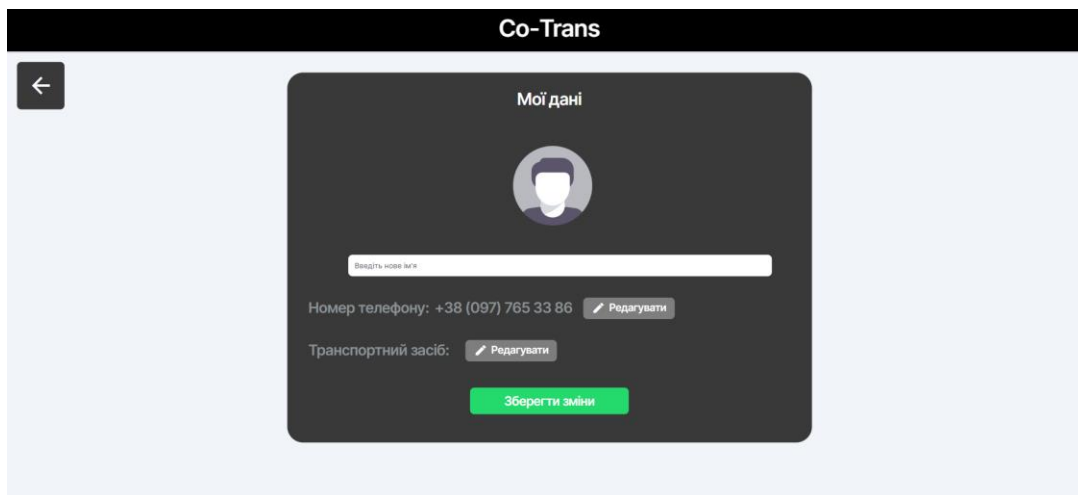


Рисунок 4.7 – Форма для редагування імені

Після натискання кнопки «Зберегти зміни» користувач автоматично потрапляє на сторінку підтвердження успішної зміни даних (Рисунок 4.8).

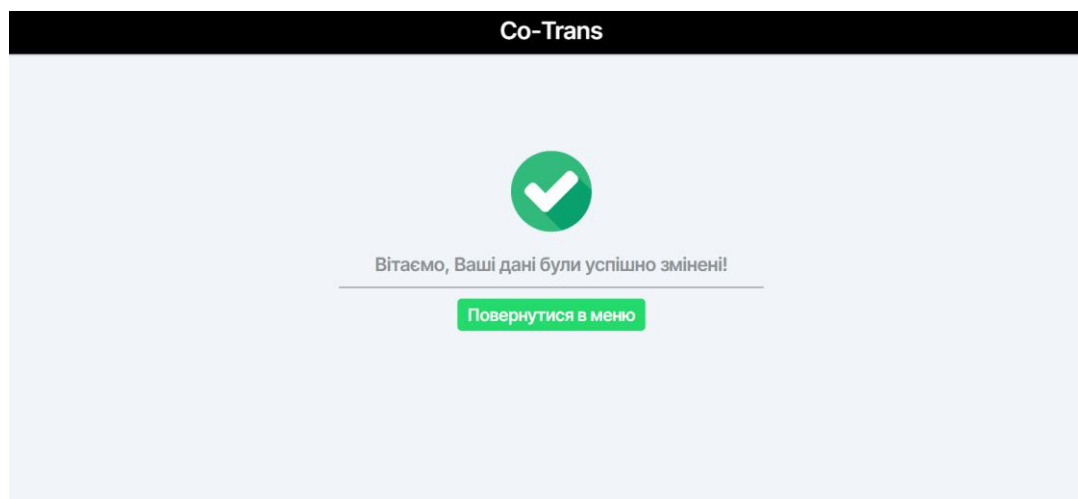


Рисунок 4.8 – Сторінка підтвердження успішної зміни даних

Зм	Арк.	№ докум.	Підп.	Дата

ІАЛЦ.045440.004 ПЗ

Арк.
51

При натисканні кнопки «Повернутися в меню» – користувач повертається на сторінку меню, де одразу можна побачити внесені зміни (Рисунок 4.9).

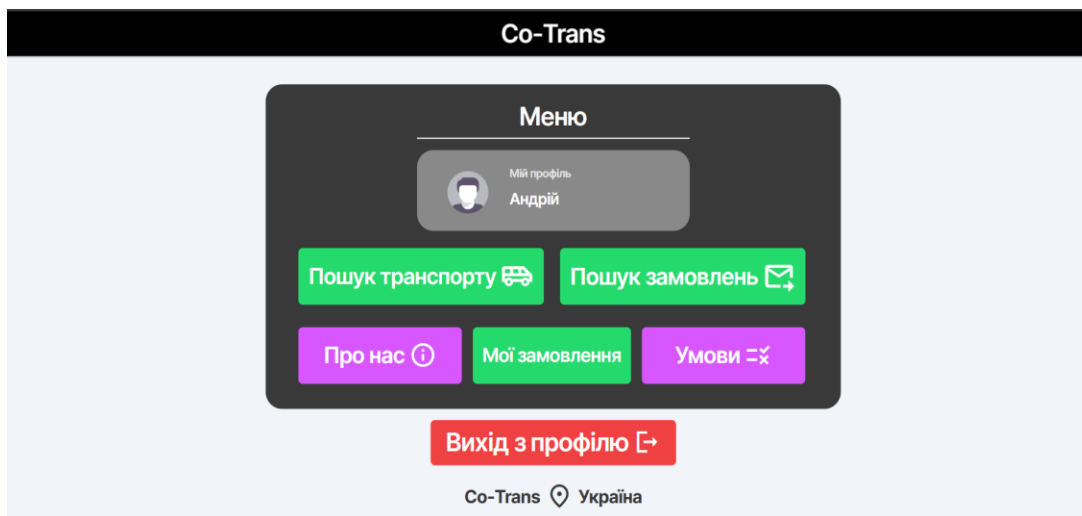


Рисунок 4.9 – Відображення внесених змін на сторінці меню

Фіолетовим кольором виділені кнопки для переходу на інформаційні сторінки системи, де користувач може ознайомитися з потрібною йому інформацією. Наприклад, на Рисунку 4.10 зображено сторінку «Про нас».



Рисунок 4.10 – Сторінка «Про нас»

Зм	Арк.	№ докум.	Підп.	Дата

ІАЛЦ.045440.004 ПЗ

Арк.
52

Практично на кожній сторінці системи (де це необхідно), у верхньому лівому куті є кнопка для зворотної навігації на попередню сторінку.

При натисканні кнопки «Умови» – відкривається сторінка, на якій прописані умови користування системою (Рисунок 4.11).



Рисунок 4.11 – Сторінка «Умови»

При натисканні кнопки «Пошук транспорту» – відкривається сторінка з формою для заповнення даних замовлення на перевезення. У ній необхідно внести основні дані про вантаж, обрати місто, з якого буде їхати вантаж та місто, до якого він прямує. Також можна опційно вибрати пункти «Крихкий вантаж» та «Терміново», вказати запропоновану оплату (Рисунок 4.12).

Рисунок 4.12 – Сторінка для створення заявки на перевезення

Якщо всі обов'язкові поля коректно заповнені – при натисканні кнопки «Опублікувати замовлення» відкривається сторінка, яка повідомляє користувача про успішно створене замовлення (Рисунок 4.13).

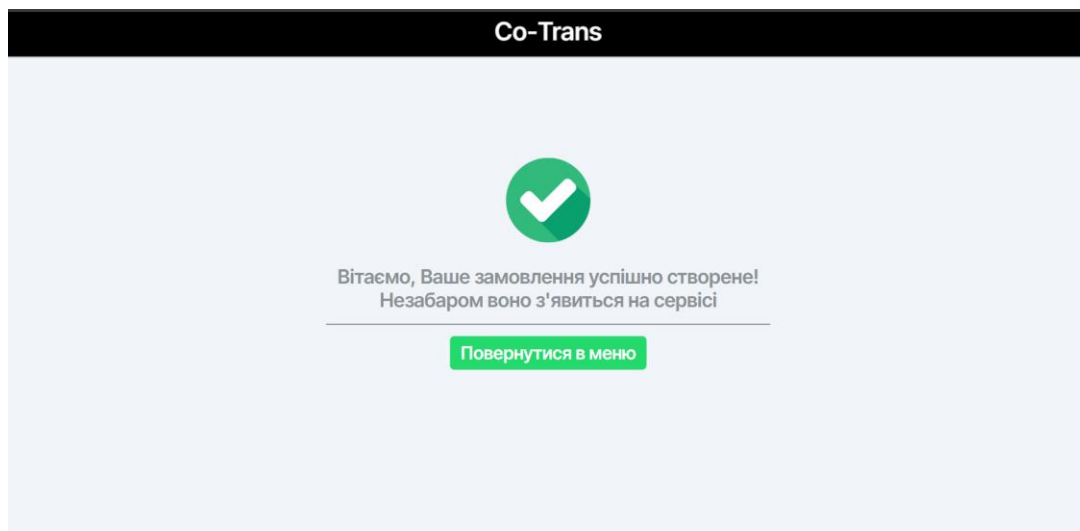


Рисунок 4.13 – Сторінка підтвердження успішного створення замовлення

При натисканні кнопки «Повернутися в меню» – користувач повертається до меню. Натискаючи кнопку «Мої замовлення» в меню – відкривається сторінка, на якій користувач може переглянути створені ним замовлення (Рисунок 4.14).

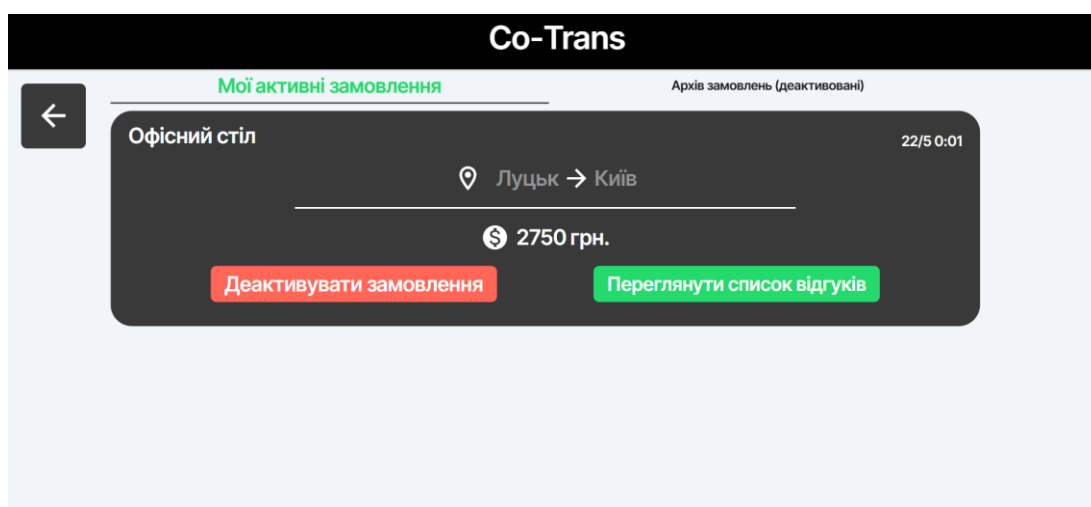


Рисунок 4.14 – Сторінка «Мої замовлення»

На сторінці «Мої замовлення» наразі відображається єдине, щойно створене замовлення на перевезення офісного столу з Луцька до Києва, воно знаходиться в розділі «Мої активні замовлення».

Тепер виконаємо вхід в інший профіль, для того щоб при пошуку замовлення на перевезення знайти саме створене нами замовлення (користувач в ролі перевізника не бачить своїх замовлень на перевезення для зручності).

Виходимо з актуального профілю, натискаючи кнопку «Вихід з профілю». Після входу в інший профіль (Рисунок 4.15) відкриваємо розділ «Пошук замовлень» - відкривається сторінка, на якій буде прокладатися маршрут поїздки (Рисунок 4.16).

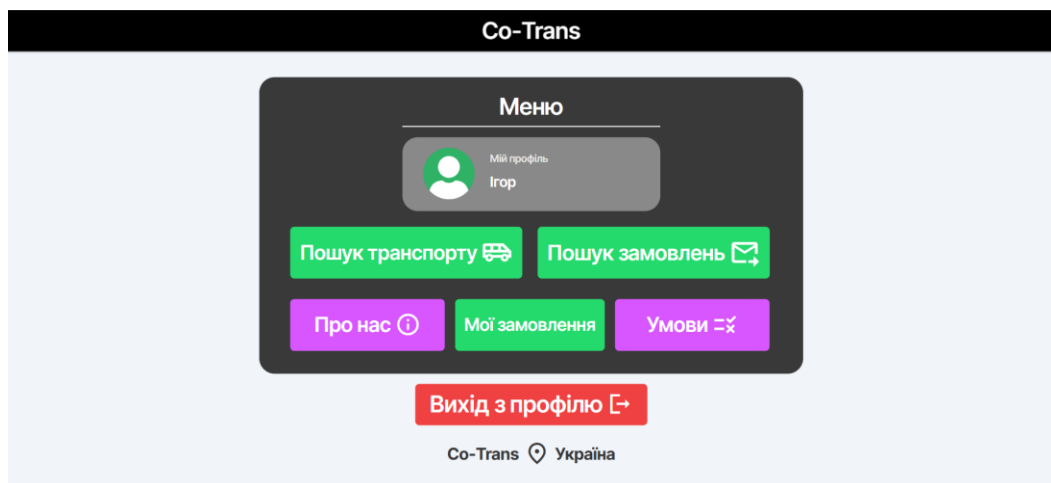


Рисунок 4.15 – Здійснений вхід в інший профіль для тестування

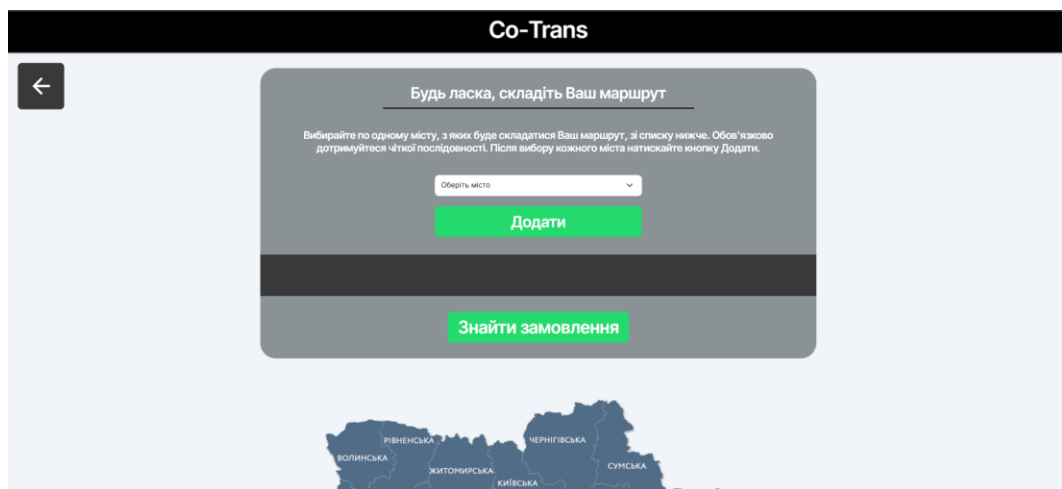


Рисунок 4.16 – Сторінка для прокладення маршруту поїздки

Зм	Арк.	№ докум.	Підп.	Дата

ІАЛЦ.045440.004 ПЗ

Арк.
55

Для зручності, на сторінці прокладення маршруту, під формою знаходиться карта України, аби користувачі могли точно та послідовно обрати список міст (Рисунок 4.17).

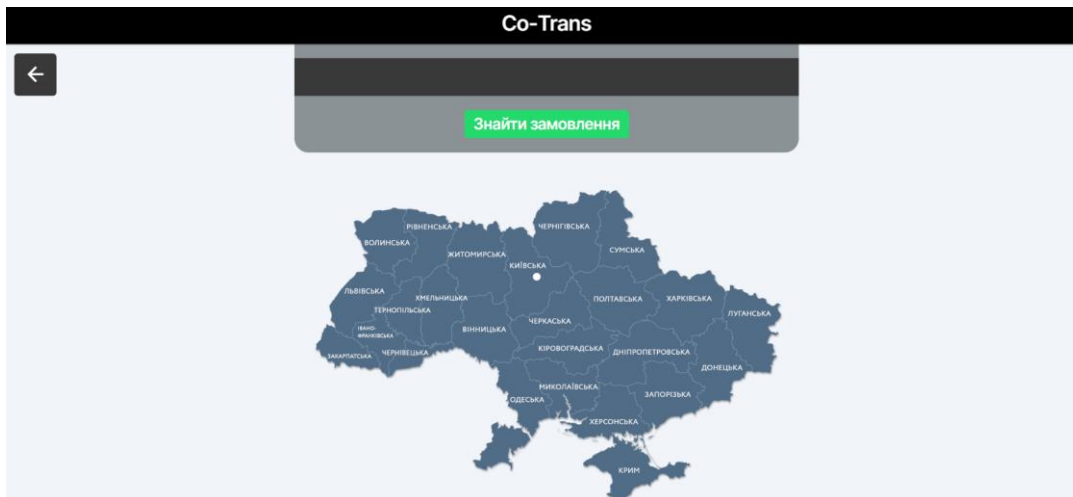


Рисунок 4.17 – Карта України, для зручного прокладання шляху

Для прокладання маршруту необхідно по черзі, в чіткій послідовності, обирати міста з випадаючого списку, натискаючи після вибору кожного міста кнопку додати. Маршрут буде динамічно відображатися на темно-сірому контейнері (Рисунок 4.18).

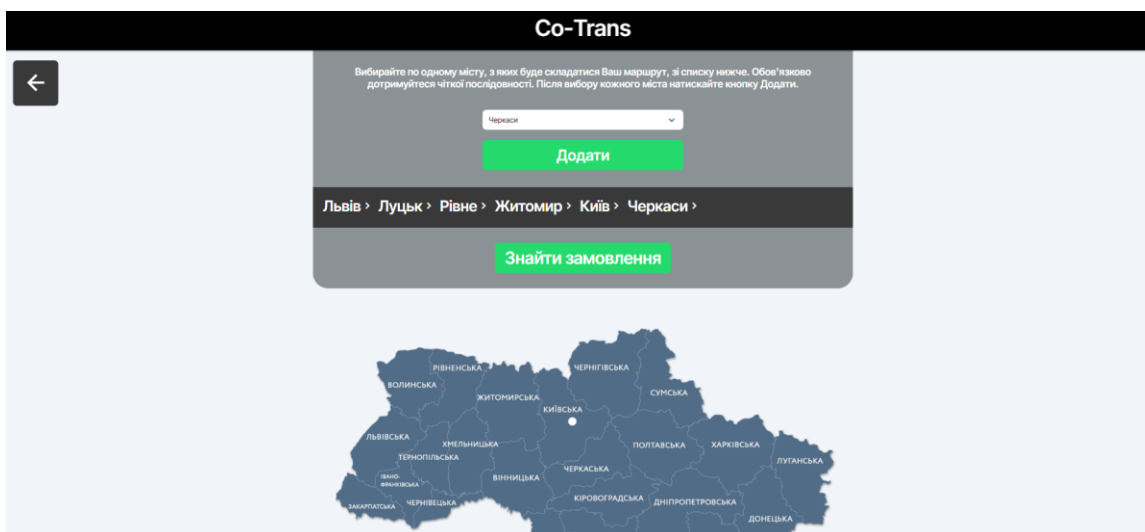


Рисунок 4.18 – Відображення прокладеного маршруту

Зм	Арк.	№ докум.	Підп.	Дата

Даний маршрут було прокладено так, аби продемонструвати коректність роботи алгоритму з підбору замовлень за маршрутом: він містить міста Луцьк та Київ у потрібному порядку. При натисканні кнопки «Знайти замовлення» - відкривається екран з підібраними за маршрутом замовленнями (Рисунок 4.19).

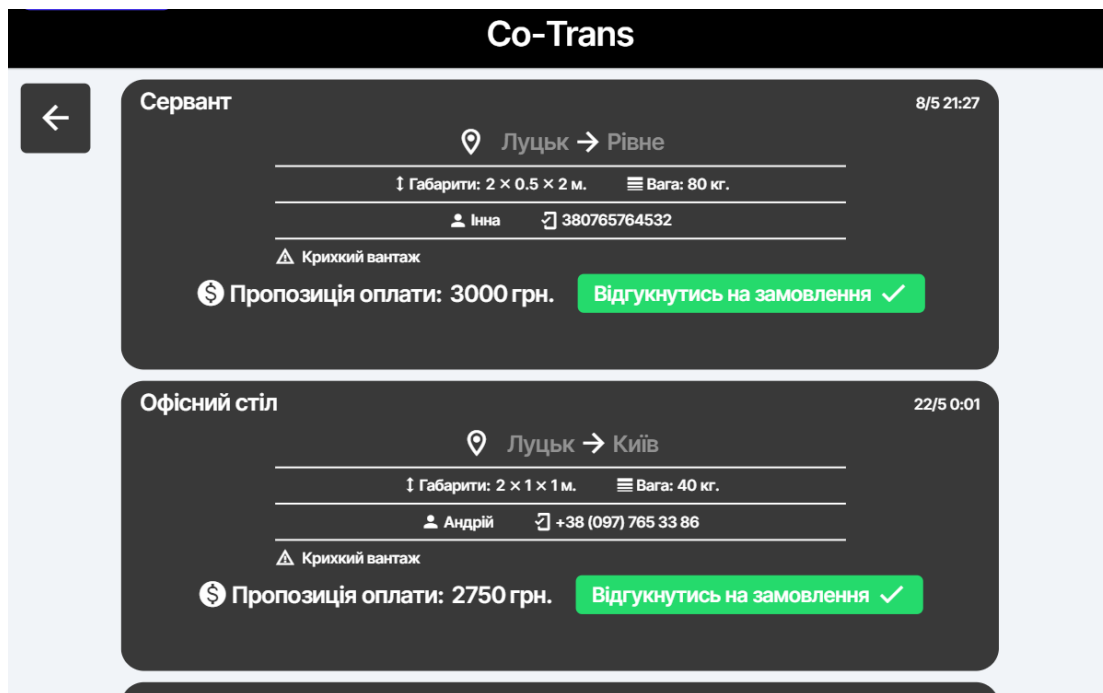


Рисунок 4.19 – Запропоновані за маршрутом замовлення

На рисунку, наведеному вище ми можемо бачити, що створене нами замовлення з іншого профілю успішно відображається у списку запропонованих. Воно знаходиться на другій позиції через те, що перше в списку замовлення необхідно перевезти за коротшим суміжним маршрутом (Луцьк - Рівне), що дасть змогу перевізнику, наприклад, виконати його та одразу взяти новий вантаж у м.Рівному.

При натисканні кнопки «Відгукнутись на замовлення» на певній заявці – дана кнопка на вибраній заявці змінює колір та стає неактивною, для того, щоб перевізник бачив на які заявки він вже відгукнувся та для забезпечення відсутності дублікатів відгуків (Рисунок 4.20).

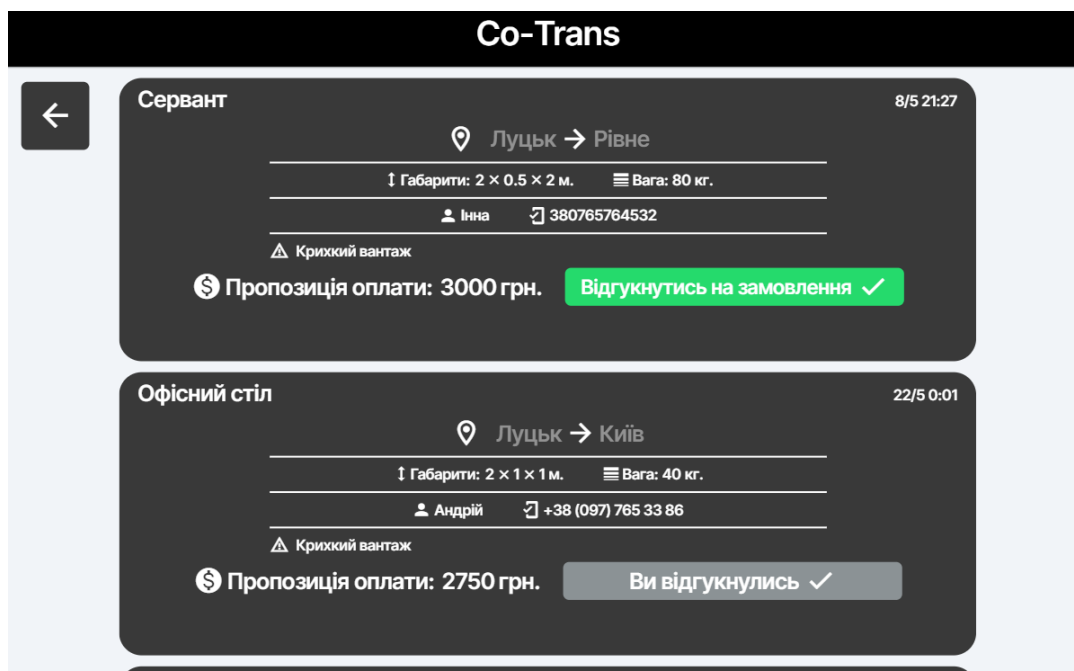


Рисунок 4.20 – Приклад відображення замовлення, на яке вже відгукнувся даний водій

Тепер знову авторизуємось у першому профілі, для того, щоб побачити відгук, залишений з другого профілю. Заходимо в розділ «Мої замовлення» і на створеному нами замовленні натискаємо кнопку «Переглянути список відгуків». У вікні, що відкрилось, бачимо залишений нами відгук з іншого профілю (Рисунок 4.21).

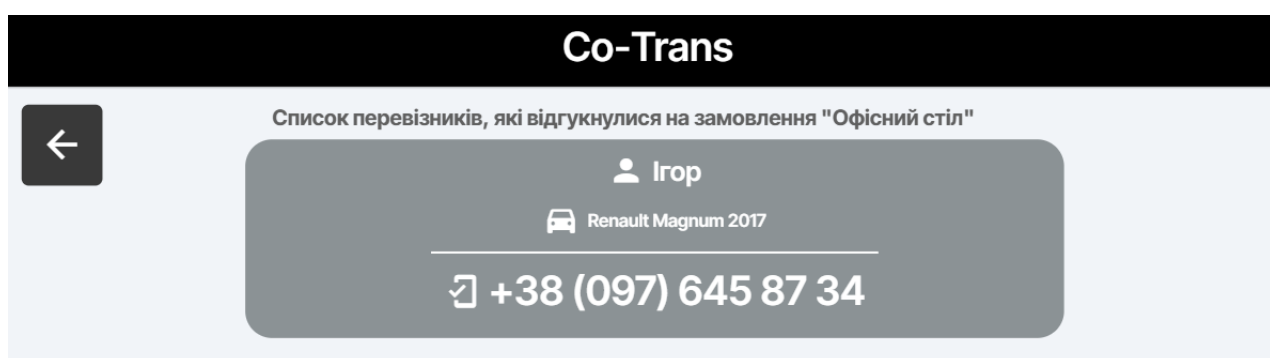


Рисунок 4.21 – Сторінка відгуків перевізників на замовлення

Замовник отримує список контактів водіїв, які готові виконати їхнє замовлення. А так як поле запропонованої вартості перевезення, при створенні замовлення, є опційним і автор може запропонувати свою ціну – перевізники

при перегляді заявок за суміжним маршрутом також можуть звернутися до автора за вказаним у замовленні номером телефону та запропонувати свою вартість конкретного перевезення.

При натисканні кнопки «Деактивувати замовлення» на конкретному замовленні у розділі «Мої замовлення» (Рисунок 4.22) - оголошення для зручності переноситься зі списку «Мої активні замовлення» у список «Архів замовлень (деактивовані)» (Рисунок 4.23).

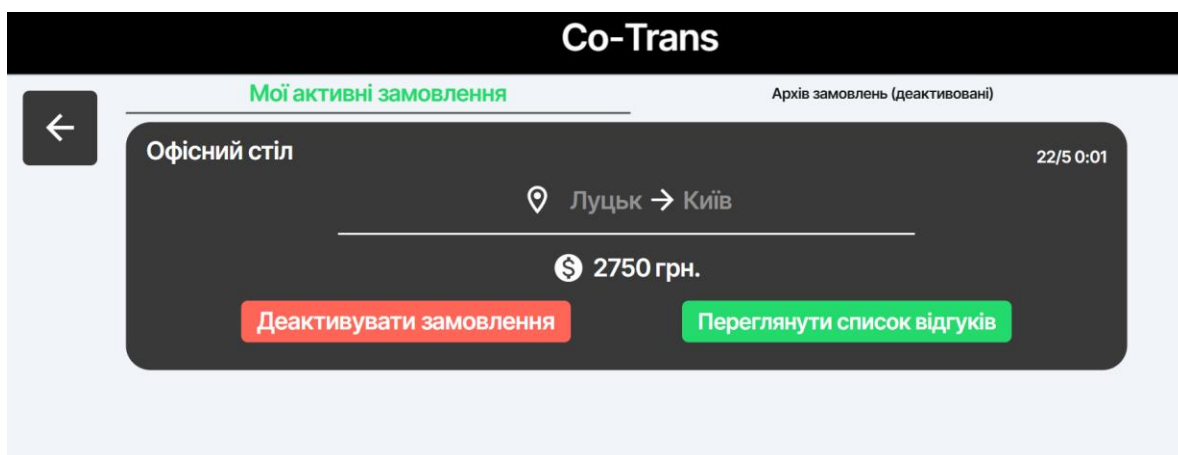


Рисунок 4.22 – Список «Мої активні замовлення»

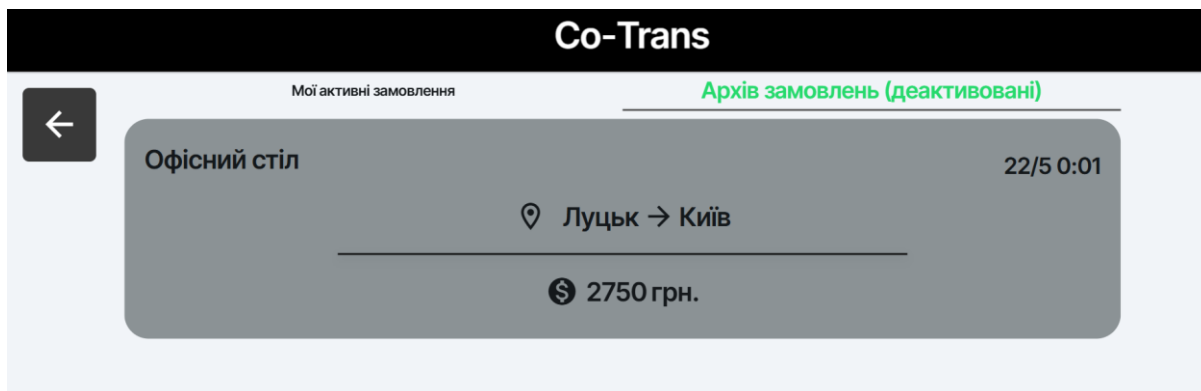


Рисунок 4.23 – Список «Архів замовлень (деактивовані)»

Якщо користувач під час заповнення особистих даних, після реєстрації, пропустив поле назви свого транспорту і потім спробує виконати пошук

замовлень на перевезення – автоматично відкриється сторінка, на якій необхідно буде дозаповнити поле назви транспортного засобу (Рисунок 4.24).

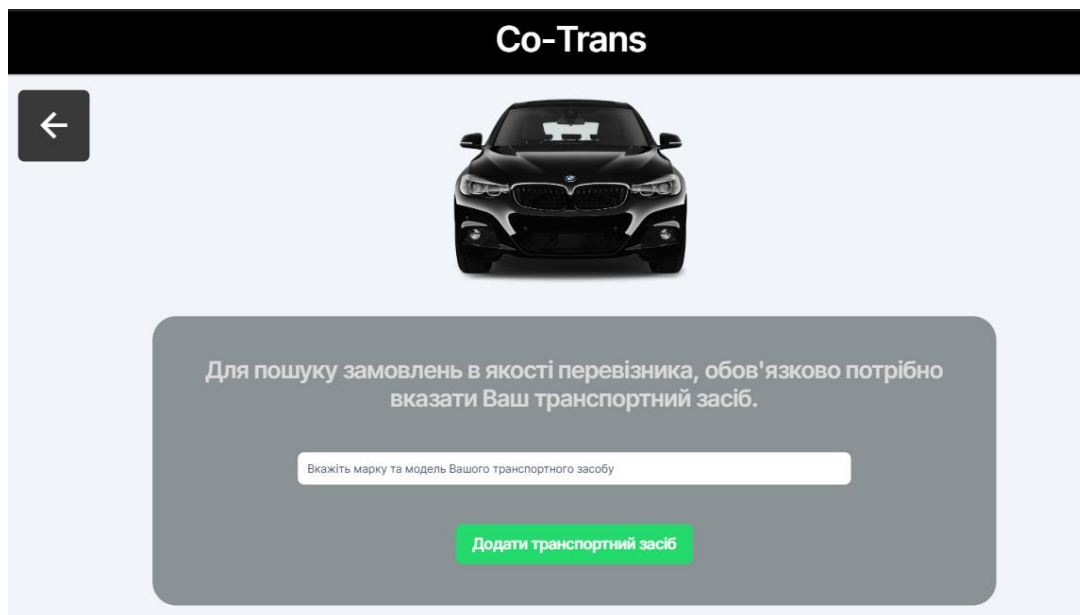


Рисунок 4.24 – Сторінка додавання назви ТЗ

Після введення назви ТЗ та натискання кнопки «Додати транспортний засіб» - оновлені дані зберігаються і користувач може виконувати пошук замовлень в ролі перевізника. Початково дане поле є опційним (не обов'язковим для заповнення), через те, що частина користувачів може використовувати дану систему виключно для пошуку перевізників, не маючи власного ТЗ.

Тестування системи, проведене вище, методом виконання типового сценарію роботи користувача, демонструє коректну роботу системи та відповідність до вимог.

4.2. Перспективи розвитку системи

У процесі експлуатації створеної системи, будуть виникати нові вимоги та потреби користувачів, що створює передумови для її подальшого вдосконалення. Зокрема, перспективними напрямками розвитку є впровадження

системи рейтингів водіїв та замовників. Наявність такої функціональності дозволить підвищити рівень довіри між учасниками перевезень, стимулюватиме відповідальну поведінку користувачів та сприятиме формуванню репутації на основі фактичного досвіду взаємодії.

У середньостроковій перспективі можливий вихід системи на іноземний ринок. Це дозволить залучити нову аудиторію та реалізовувати перевезення на міжнародному рівні. Для цього необхідно передбачити підтримку декількох мов, врахування нормативних вимог різних країн та додавання нових міст і країн до списку запропонованих. Крім того, актуальним буде розширення географічного охоплення платформи. Станом на даний момент, реалізація орієнтована на великі міста (обласні центри) України, проте доцільним також є залучення користувачів із невеликих населених пунктів, що дасть змогу більш ефективно покривати логістичні потреби у віддалених регіонах.

Також доцільною буде реалізація можливості редагування маршруту водієм після створення поїздки. Можливість оновлювати список населених пунктів, через які пролягатиме маршрут, або змінювати послідовність уже вказаних точок надасть більше свободи в плануванні та дозволить оперативно реагувати на зміни в дорожній ситуації чи особисті обставини перевізника.

Для підвищення рівню безпеки в системі важливим кроком є реалізація функціоналу контролю оплат. Запровадження механізму гарантування платежів через платформу дозволить уникнути шахрайських випадків та забезпечити надійність розрахунків між замовниками та перевізниками. Це також відкриває можливість для інтеграції платіжних систем.

Усі зазначені напрями є реалістичними з огляду на обрану архітектуру системи, що дозволяє масштабувати функціональність та адаптувати її до нових викликів.

					ІАЛЦ.045440.004 ПЗ	Арк.
Зм	Арк.	№ докум.	Підп.	Дата		61

ВИСНОВКИ

У ході виконання дипломної роботи було розроблено комп'ютерну систему, призначену для організації сумісних перевезень вантажів різних габаритів, що дозволяє ефективно поєднувати логістичні потреби замовників і актуальні маршрути водіїв. Такий підхід сприяє підвищенню ефективності перевезень, зменшенню витрат на логістику та оптимізації використання транспортних ресурсів.

Проведений аналіз довів доцільність створення даної системи, особливо з огляду на зростаючі потреби у гнучких і доступних логістичних рішеннях, зокрема для середніх і великих за розміром відправлень.

У процесі реалізації системи було обґрунтовано вибір формату вебзастосунку, інструментів для розробки, було розроблено архітектуру клієнтської та серверної частин, реалізовано механізми взаємодії з базою даних, авторизації користувачів, створення та пошуку замовлень на перевезення. Окрему увагу приділено реалізації пошуку замовлень відповідно до маршруту перевізника, що стало ключовим функціональним елементом системи.

Результуюче рішення надає простий та інтуїтивно зрозумілий інтерфейс, що дозволяє як водіям, так і замовникам швидко знаходити відповідні перевезення. Реалізовані функції забезпечують базову безпеку, та чіткість виконання ключових завдань. Запропонована система має потенціал до подальшого розвитку та масштабування.

Таким чином, поставлені у роботі завдання були успішно виконані, а результати реалізації підтверджують доцільність та актуальність розробки.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Нова пошта – Доставка по Україні. URL: <https://novaposhta.ua/delivery>
2. Supply Chain Dive: Shefali Kapadia. “Analysis: One-third of Class 8 truck miles are driven empty”, 2019. URL: <https://www.supplychaindive.com/news/convoy-empty-miles-trucking/559972>
3. SCCEU: Delivery trucks are driving billions of miles completely empty, 2022 URL: <https://sccEU.org/delivery-trucks-are-driving-billions-of-miles-completely-empty>
4. Wezom. “Як створити вебзастосунок”, 2023. URL: <https://bit.ly/4mxxbcJ>
5. Єлизавета Данілова. “Огляд FlutterFlow”, 2023 URL: <https://dou.ua/forums/topic/43122>
6. Alternative Science: SQL чи NoSQL — ось в чому питання. URL: <https://bit.ly/4kDWyIf>
7. Документація Flutter – інтеграція з Firebase. URL: <https://docs.flutter.dev/data-and-backend/firebase>
8. Applify: Порівняння Material і Flat дизайну. URL: <https://www.applify.com.sg/blog/material-vs-flat-design>
9. ISO 9241-210:2010, Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems
10. Документація Firebase Firestore Client SDK. URL: <https://firebase.google.com/docs/firestore/client/libraries>

