



Приватний заклад вищої освіти
Одеський технологічний університет «ШАГ»
Кафедра інформаційних технологій та фундаментальної підготовки

Випускна кваліфікаційна робота бакалавра

**Інформаційна система управління рухом громадського транспорту
(Індивідуальний проект: управління ресурсами)**

на здобуття бакалаврського рівня вищої освіти
зі спеціальності «122 Комп'ютерні науки»

Виконавці проекту

№	П.І.Б.	Група
1	Солтановский Борис Борисович	КН-П-213
2	Сова Станіслав Сергійович	КН-П-213

Автор звіту

_____ Сова Станіслав Сергійович

Науковий керівник	Доцент кафедри ІТ та ФП, к.т.н., доцент Новак С.М.
-------------------	---

АНОТАЦІЯ

УДК 004.9

Д-30

Новак І. С. Інформаційна система управління рухом громадського транспорту (Індивідуальний проект: управління ресурсами): випускна кваліфікаційна робота на здобуття бакалаврського рівня вищої освіти зі спеціальності «122 Комп'ютерні науки». - Одеса: ОТУШ, 2022 - 27 с.

Робота присвячена дослідженню, розробці та впровадженню інформаційної системи управління гнучкими маршрутами громадського транспорту через адміністративний веб-портал. Основна мета системи FlexTrans – підвищення ефективності транспортної мережі в умовах низького пасажиропотоку, непікових періодів або в районах без значних комерційних чи туристичних об'єктів. Система забезпечує гнучке планування маршрутів, управління зупинками, транспортними засобами та користувачами.

Актуальність роботи зумовлена потребою в оптимізації громадського транспорту в сучасних містах, де традиційні підходи часто виявляються неефективними. FlexTrans використовує сучасні технології, такі як Angular для інтерфейсу та .NET для серверної частини, що гарантує зручність, продуктивність і безпеку системи.

Об'єкт дослідження: алгоритми та технології управління гнучкими маршрутами громадського транспорту, зокрема через адміністративний веб-портал.

Предмет дослідження: система управління гнучкими маршрутами на прикладі проекту FlexTrans.

Завдання дослідження:

- проаналізувати існуючі системи управління транспортними мережами, визначивши їхні особливості та напрями вдосконалення;
- визначити засоби розробки програмного забезпечення для створення веб-порталу з управління маршрутами та сутностями;

- спроектувати систему з необхідним функціоналом для адміністраторів;
- розробити та протестувати програмне забезпечення в різних умовах;
- впровадити засоби захисту даних для підвищення безпеки системи.

Методи дослідження: аналіз систем управління транспортом, вивчення стандартів GTFS, системний аналіз, моделювання та сучасні методології розробки програмного забезпечення.

Зміст

Зміст	3
ВСТУП	4
ТЕХНІЧНЕ ЗАВДАННЯ ДО ПРОЕКТУ	5
1. Загальний опис проекту	5
2. Призначення адміністративного веб-порталу	5
3. Технологічний стек	5
4. Основні модулі та їх функціональність	8
4.1. Управління агенціями	8
4.2. Управління поїздками	8
4.3. Управління періодами	8
4.4. Управління державними святами	9
4.5. Управління реальними сервісами	9
4.6. Управління маршрутами	9
4.7. Управління зупинками	10
4.8. Управління теоретичними сервісами	10
4.9. Управління зонами	11
4.10. Управління транспортними засобами	11
4.11. Управління користувачами	11
4.12. Інтерфейс планування	12
4.13. Інтерфейс статистики	12
РОЗДІЛ 1. ЗАГАЛЬНА ХАРАКТЕРИСТИКА СИСТЕМИ	14
РОЗДІЛ 2. РЕАЛІЗАЦІЯ ФРОНТЕНД ЧАСТИНИ ТА ОСНОВНИХ МЕХАНІЗМІВ АДМІНІСТРАТИВНОГО ВЕБ-ПОРТАЛУ	16
2.1. Аналіз предметної області та проектування доменних сутностей	16
2.2. Реалізація серверної логіки та CRUD-операцій	17
2.3. Реалізація фронтенд частини	17
2.4. Реалізація модуля управління маршрутами	18
2.4.1. Створення та конфігурація маршрутів	18
2.4.2. Валідація та відображення маршрутів	19
2.4.3. Інтерфейс управління маршрутами	20
2.5. Тестування та оптимізація	21
2.6. Висновки щодо реалізації	21
РОЗДІЛ 3	23
РЕЗУЛЬТАТИ ВИПРОБУВАНЬ	23
ВИСНОВКИ	25
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	26

ВСТУП

Сучасний розвиток інформаційних технологій сприяє цифровізації багатьох сфер життя, зокрема громадського транспорту. Стратегія розвитку інформаційного суспільства в Україні зазначає, що «комунікаційні технології є необхідним інструментом соціально-економічного прогресу, одним з основних чинників інноваційного розвитку економіки». У цьому контексті особливої актуальності набувають системи, які дозволяють гнучко управляти транспортними мережами, адаптуючи їх до реальних потреб пасажирів і умов міста.

Предметом цієї роботи є система управління гнучкими маршрутами громадського транспорту, реалізована в рамках проекту FlexTrans. Система вирішує проблему неефективності традиційного транспорту в районах з низьким пасажиропотоком, у непікові години або в зонах без значних комерційних об'єктів. FlexTrans пропонує адміністративний веб-портал для планування маршрутів, управління зупинками, транспортними засобами та користувачами, забезпечуючи гнучкість і адаптивність транспортної інфраструктури.

Для реалізації використано сучасний технологічний стек: Angular для клієнтської частини, .NET для серверної, а також Azure для розгортання, що забезпечує високу продуктивність і масштабованість.

Детальне технічне завдання наведено далі. У рамках звіту розглядається адміністративний веб-портал, який включає серверну частину для обробки та зберігання даних і клієнтську частину з інтуїтивним інтерфейсом для адміністраторів. Програмне забезпечення розгорнуто на хмарному сервісі Azure.

Висновки відображають досягнуті результати, а загальний висновок узагальнює звіт всіх учасників проекту. Актуальність розробки FlexTrans полягає в її потенціалі оптимізувати транспортну інфраструктуру, покращити якість життя городян і сприяти сталому розвитку міст.

ТЕХНІЧНЕ ЗАВДАННЯ ДО ПРОЕКТУ

1. Загальний опис проекту

Проект **FlexTrans** розроблений для створення гнучкої системи громадського транспорту, яка дозволяє адаптувати маршрути до потреб конкретних міст чи транспортних агенцій. Основна мета — вирішити проблему неефективності традиційного громадського транспорту в районах із низьким пасажиропотоком, у непікові години або в зонах без значних комерційних чи туристичних об'єктів. Система спрямована на забезпечення зручного, доступного та надійного способу пересування для користувачів, зокрема у вечірній час, коли людям потрібно дістатися додому після роботи.

Адміністративний веб-портал є єдиним компонентом системи, призначеним для управління всіма аспектами транспортної мережі, такими як планування маршрутів, налаштування зон, зупинок і транспортних засобів, а також аналіз статистики. Портал побудований із використанням сучасних технологій Material для забезпечення зручного та інтуїтивно зрозумілого інтерфейсу, а також підтримує локалізацію для французької та англійської мов, щоб відповідати потребам різних користувачів.

2. Призначення адміністративного веб-порталу

Адміністративний веб-портал слугує центральним інструментом для управління транспортною системою "FlexTrans". Він дозволяє адміністраторам створювати, переглядати, редагувати та видаляти різні сутності, пов'язані з організацією громадського транспорту, а також планувати поїздки та аналізувати ефективність системи.

3. Технологічний стек

Проект "FlexTrans" побудований із використанням сучасних технологій, які забезпечують надійність, масштабованість та зручність розробки, розгортання та

підтримки системи. Нижче наведено детальний опис технологічного стеку, що охоплює як фронтенд, так і бекенд, а також інструменти для розробки, розгортання та управління даними.

- **Фронтенд:**

- **Angular:** Основний фреймворк для створення адаптивного та інтерактивного інтерфейсу користувача адміністративного веб-порталу. Angular забезпечує модульну структуру, що спрощує розробку складних інтерфейсів та їх підтримку.
- **Material Design:** Використовується для створення стильного, інтуїтивно зрозумілого та сучасного інтерфейсу користувача, що відповідає стандартам дизайну Google. Material Design забезпечує єдиний вигляд і поведінку компонентів, таких як кнопки, таблиці та форми.
- **Локалізація:** Повна підтримка французької та англійської мов для всіх елементів інтерфейсу, включаючи текст, підказки та повідомлення, що забезпечує зручність використання для міжнародних користувачів.

- **Бекенд:**

- **.NET:** Основна платформа для розробки серверної частини системи. Використання .NET забезпечує високу продуктивність, безпеку та масштабованість бекенду.
- **AutoMapper:** Використовується для автоматичного відображення об'єктів між різними моделями даних, що зменшує кількість рутинного коду та підвищує ефективність розробки.
- **MediatR:** Реалізує шаблон "Медіатор" для обробки запитів і команд, що спрощує управління бізнес-логікою та забезпечує чітке розділення обов'язків у коді.
- **FastEndpoints:** Використовується для створення швидких і ефективних API-ендпоінтів, що оптимізують продуктивність серверної частини.

- **База даних:**
 - **Azure SQL Database:** Хмарна реляційна база даних, що забезпечує високу доступність, масштабованість і безпеку для зберігання всіх даних системи, включаючи інформацію про маршрути, зупинки, транспортні засоби та користувачів.
- **Інструменти розробки та розгортання:**
 - **Docker:** Використовується для локальної розробки та тестування, дозволяючи створювати контейнеризовані середовища, що забезпечують однаковість між локальними та продакшн-оточеннями.
 - **Azure DevOps:** Застосовується для автоматизації процесів CI/CD (безперервної інтеграції та доставки). Azure DevOps підтримує пайплайни для автоматичного тестування, складання та розгортання як фронтенд, так і бекенд частин системи.
 - **Azure Container Apps:** Бекенд розгортається в Azure Container Apps, що забезпечує масштабованість, гнучкість і легке управління контейнерами в хмарному середовищі.
 - **Azure Static Web Apps:** Фронтенд розгортається як статичний веб-додаток в Azure Static Web Apps, що забезпечує швидке завантаження сторінок, високу доступність і спрощене масштабування.
- **Додаткові технології:**
 - **TypeScript:** Використовується у фронтенді для написання типізованого JavaScript-коду, що підвищує надійність і читабельність коду.
 - **RxJS:** Бібліотека для реактивного програмування, що інтегрується з Angular для асинхронної обробки подій і запитів.
 - **Entity Framework Core:** ORM (Object-Relational Mapping) для .NET, що спрощує взаємодію з базою даних Azure SQL, забезпечуючи зручний доступ до даних і підтримку міграцій.

4. Основні модулі та їх функціональність

Система поділена на логічні модулі, кожен із яких відповідає за управління певним аспектом транспортної мережі. Нижче наведено детальний опис кожного модуля, його призначення та підтримуваних операцій.

4.1. Управління агенціями

Призначення:

Цей модуль дозволяє управляти транспортними агенціями, які надають послуги громадського транспорту в рамках системи "FlexTrans". Агенції є ключовими організаціями, що відповідають за маршрути, транспортні засоби та поїздки в межах певного міста чи регіону.

4.2. Управління поїздками

Призначення:

Цей модуль відповідає за організацію гнучких поїздок, які дозволяють адаптувати маршрути до потреб пасажирів у реальному часі. Поїздки є основою гнучкої транспортної системи "FlexTrans".

Функціональність:

- **Перегляд:** Отримання списку всіх поїздок Flexo або детальної інформації про конкретну поїздку.
- **Планування:** Збереження та перегляд стану планування поїздок для координації роботи.
- **Візуалізація:** Відображення даних поїздки на карті для зручного аналізу.

4.3. Управління періодами

Призначення:

Модуль дозволяє визначати часові періоди, протягом яких діють певні розклади чи правила роботи транспорту. Це корисно для сезонного планування чи врахування особливих умов.

Функціональність:

- **Створення:** Визначення нового періоду з датами початку та закінчення.
- **Перегляд:** Перегляд списку всіх періодів або деталей конкретного періоду.
- **Оновлення:** Зміна дат чи інших параметрів існуючого періоду.
- **Видалення:** Видалення періоду, який більше не актуальний.
- **Пошук:** Швидкий пошук періодів за назвою чи датами.

4.4. Управління державними святами

Призначення:

Цей модуль потрібен для врахування державних свят, які впливають на розклад роботи транспорту, наприклад, зменшення кількості поїздок у святкові дні.

Функціональність:

- **Створення:** Додавання нового свята з датою та назвою.
- **Перегляд:** Перегляд списку всіх свят або деталей конкретного свята.
- **Оновлення:** Редагування дати чи назви свята.
- **Видалення:** Видалення свята, яке більше не актуальне.
- **Пошук:** Швидкий пошук свят за назвою чи датою.

4.5. Управління реальними сервісами

Призначення:

Модуль дозволяє створювати та управляти реальними сервісами — конкретними поїздками з чітко визначеними початком і кінцем, які виконуються транспортними засобами.

Функціональність:

- **Створення:** Додавання нового реального сервісу на основі розкладу чи вручну.
- **Перегляд:** Перегляд списку всіх реальних сервісів або деталей конкретного сервісу.
- **Оновлення:** Зміна параметрів сервісу, наприклад, часу чи маршруту.
- **Видалення:** Видалення сервісу, який більше не потрібен.
- **Пошук:** Швидкий пошук сервісів за назвою чи іншими параметрами.

4.6. Управління маршрутами

Призначення:

Модуль призначений для створення та управління маршрутами, які складаються із зупинок і зон. Маршрути можуть бути (Flexo) або віртуальними, залежно від потреб.

Функціональність:

- **Створення:** Формування нового маршруту з набором зупинок чи зон.
- **Перегляд:** Перегляд списку всіх маршрутів або деталей конкретного маршруту.

- **Оновлення:** Редагування маршруту, наприклад, додавання чи видалення зупинок.
- **Видалення:** Видалення маршруту, який більше не використовується.
- **Пошук:** Швидкий пошук маршрутів за назвою чи іншими характеристиками.

4.7. Управління зупинками

Призначення:

Модуль дозволяє управляти зупинками — як фізичними (реальними точками на карті), так і комерційними (групами фізичних зупинок), які є основою маршрутів.

Функціональність:

- **Створення:** Додавання нової зупинки з географічними координатами чи назвою.
- **Перегляд:** Перегляд списку всіх зупинок або деталей конкретної зупинки.
- **Оновлення:** Зміна даних зупинки, наприклад, її розташування чи назви.
- **Видалення:** Видалення зупинки, яка більше не використовується.
- **Пошук:** Швидкий пошук зупинок за назвою, містом чи іншими параметрами.

4.8. Управління теоретичними сервісами

Призначення:

Цей модуль використовується для планування абстрактних сервісів, які слугують основою для створення реальних поїздок. Вони враховують робочі дні, паузи чи зміни.

Функціональність:

- **Створення:** Додавання нового теоретичного сервісу з розкладом чи правилами.
- **Перегляд:** Перегляд списку всіх теоретичних сервісів або деталей конкретного сервісу.
- **Оновлення:** Редагування параметрів сервісу.
- **Видалення:** Видалення сервісу, який більше не потрібен.
- **Пошук:** Швидкий пошук сервісів за назвою чи іншими характеристиками.

4.9. Управління зонами

Призначення:

Модуль дозволяє створювати та управляти зонами — групами зупинок, які використовуються в гнучких маршрутах типу Flexo.

Функціональність:

- **Створення:** Додавання нової зони з набором зупинок.
- **Перегляд:** Перегляд списку всіх зон або деталей конкретної зони.
- **Оновлення:** Зміна складу зони, наприклад, додавання чи видалення зупинок.
- **Видалення:** Видалення зони, яка більше не актуальна.
- **Пошук:** Швидкий пошук зон за назвою чи іншими параметрами.

4.10. Управління транспортними засобами

Призначення:

Модуль дозволяє управляти транспортними засобами, які використовуються для виконання поїздок у системі "FlexTrans".

Функціональність:

- **Створення:** Додавання нового транспортного засобу з характеристиками (наприклад, тип, номер).
- **Перегляд:** Перегляд списку всіх транспортних засобів або деталей конкретного засобу.
- **Оновлення:** Зміна даних транспортного засобу.
- **Видалення:** Видалення транспортного засобу, який більше не використовується.
- **Пошук:** Швидкий пошук транспортних засобів за назвою чи іншими параметрами.

4.11. Управління користувачами

Призначення:

Модуль призначений для управління користувачами системи "FlexTrans", які мають різні ролі, такі як адміністратори, водії чи глобальні адміністратори. Він дозволяє створювати нових користувачів, переглядати їхні дані, редагувати профілі та видаляти облікові записи, забезпечуючи належне адміністрування доступу до системи.

Функціональність:

- **Створення:** Додавання нового користувача з визначенням його ролі (наприклад, Admin, Driver), контактних даних та інших параметрів, таких як ім'я, електронна пошта.
- **Перегляд:** Отримання списку всіх користувачів системи або детальної інформації про конкретного користувача, включаючи його роль та дані.
- **Оновлення:** Редагування даних користувача, наприклад, зміна ролі, контактної інформації чи налаштувань доступу.
- **Видалення:** Видалення облікового запису користувача, якщо він більше не потрібен у системі.
- **Пошук:** Швидкий пошук користувачів за іменем, електронною поштою чи роллю для зручності адміністрування.

4.12. Інтерфейс планування

Призначення:

Інтерфейс планування дозволяє адміністраторам конфігурувати реальні сервіси та планувати поїздки на основі теоретичних даних.

Функціональність:

- **Створення:** Формування реальних сервісів на основі теоретичних розкладів або вручну.
- **Оновлення:** Редагування деталей поїздок, таких як час чи маршрут.
- **Планування:** Перетворення теоретичних поїздок у реальні для їх виконання.

4.13. Інтерфейс статистики

Призначення:

Інтерфейс статистики призначений для моніторингу виконання рейсів, аналізу планових і фактичних показників, а також візуального порівняння теоретичних, запланованих і завершених поїздок.

Функціональність:

- **Візуалізація:**
Виведення статистики у вигляді інтерактивного графіку (ECharts)
- **Аналітика:**
Табличне представлення тих самих даних з розбивкою по днях, що дозволяє детально аналізувати кожну дату.

- **Фільтрація та масштаб:**

Можливість фільтрувати дані за маршрутом, вибрати часовий інтервал (день, тиждень, місяць, рік) та обмежувати період за допомогою календаря.

- **Синхронізоване представлення:**

Графік і таблиця використовують спільне джерело даних, що гарантує узгодженість інформації між візуальним та табличним поданням.

РОЗДІЛ 1. ЗАГАЛЬНА ХАРАКТЕРИСТИКА СИСТЕМИ

Відповідно до технічного завдання, система "FlexTrans" поділяється на наступні архітектурні компоненти:

- **Адміністративний веб-портал,**
- **Серверна частина (Веб API сервіс),**
- **База даних.**

Адміністративний веб-портал призначений для забезпечення адміністраторів транспортної системи зручним інтерфейсом для управління всіма аспектами транспортної мережі. Він надає можливості для створення, редагування, перегляду та видалення сутностей, таких як маршрути, зупинки, транспортні засоби, поїздки, користувачі тощо. Інтерфейс реалізовано з використанням сучасних технологій Angular та Material Design, що забезпечує адаптивний, інтуїтивно зрозумілий та стильний вигляд. Локалізація для англійської та французької мов дозволяє використовувати портал у різних регіонах. Портал взаємодіє з серверною частиною через REST API, забезпечуючи швидкий доступ до даних та їх обробку. Технології, використані для реалізації: Angular, TypeScript, RxJS, Material Design, HTML, CSS.

Серверна частина (Веб API сервіс) відповідає за обробку запитів від адміністративного порталу, управління бізнес-логікою та взаємодію з базою даних. Вона забезпечує стабільну та масштабовану роботу системи, обробляючи операції створення, оновлення, видалення та пошуку даних. Серверна частина побудована на платформі .NET з використанням FastEndpoints для створення ефективних API-ендпоінтів, AutoMapper для спрощення мапінгу даних та MediatR для організації бізнес-логіки за шаблоном "Медіатор". Розгорнута в Azure Container Apps, серверна частина забезпечує високу доступність та масштабованість. Технології: .NET, C#, FastEndpoints, AutoMapper, MediatR, Entity Framework Core.

База даних реалізована на Azure SQL Database і слугує для зберігання всіх даних системи, включаючи інформацію про маршрути, зупинки, транспортні засоби, поїздки, користувачів та агенції. База даних нормалізована до третьої нормальної форми, що забезпечує ефективність та цілісність даних. Використання Entity Framework Core дозволяє зручно взаємодіяти з базою даних, забезпечуючи підтримку міграцій та спрощення доступу до даних. Цілісність забезпечується обмеженнями на значення полів, зовнішніми та внутрішніми ключами, а також механізмами каскадного оновлення та видалення. Технології: Azure SQL Database, Entity Framework Core.

Система розгорнута в хмарному середовищі Azure, де фронтенд розміщено на Azure Static Web Apps, а бекенд — на Azure Container Apps. Для автоматизації процесів розробки, тестування та розгортання використано Azure DevOps, що забезпечує безперервну інтеграцію та доставку (CI/CD). Локальна розробка проводилася з використанням Docker для створення контейнеризованих середовищ, що забезпечують однаковість між локальними та продакшн-оточеннями. Система підтримує авторизований доступ для користувачів із різними ролями (адміністратори, водії, глобальні адміністратори) через внутрішню систему аутентифікації з підтримкою Azure B2C. Код системи зберігається в репозиторії Azure DevOps.

РОЗДІЛ 2. РЕАЛІЗАЦІЯ ФРОНТЕНД ЧАСТИНИ ТА ОСНОВНИХ МЕХАНІЗМІВ АДМІНІСТРАТИВНОГО ВЕБ-ПОРТАЛУ

2.1. Аналіз предметної області та проектування доменних сутностей

На початковому етапі розробки адміністративного веб-порталу системи "FlexTrans" було проведено аналіз предметної області за стандартом GTFS (General Transit Feed Specification). Цей стандарт визначає структуру даних для громадського транспорту, включаючи маршрути, зупинки, транспортні засоби та розклади. На основі аналізу виділено основні доменні сутності: зупинки (з типами та географічними координатами), транспортні засоби (з характеристиками, такими як тип і номер) та маршрути (з варіаціями, сегментами та розкладами). Для кожної сутності визначено основні поля, їх типи та обмеження, що забезпечують відповідність стандарту GTFS.

На серверній частині за допомогою Entity Framework Core (EF Core) було налаштовано збереження даних із валідацією значень. Наприклад, для зупинок забезпечено збереження координат у форматі широта, довгота, а для транспортних засобів — унікальних ідентифікаторів та типів. EF Core використано для створення моделей даних із підтримкою міграцій, що спрощує оновлення структури бази даних Azure SQL Database. Для ефективного мапінгу між доменними сутностями та DTO (Data Transfer Objects) використано бібліотеку AutoMapper, яка забезпечує автоматичне відображення об'єктів для операцій створення, отримання, оновлення та видалення (CRUD). Конфігурації AutoMapper налаштовано для підтримки всіх CRUD-операцій: `GetList`, `Autocomplete` (спрощені моделі `ListItem` для автодоповнення), `GetById`, `Create/Update`.

Для забезпечення коректності даних реалізовано валідацію DTO-моделей за допомогою бібліотеки FluentValidation. Наприклад, для транспортних засобів валідація перевіряє унікальність номерів та відповідність типів. Додатково

створено кастомні методи валідації для перевірки бізнес-логіки, такі як перевірка прав користувача на створення транспортного засобу. Ці методи інтегровано в серверну логіку для забезпечення безпеки та відповідності бізнес-вимогам.

2.2. Реалізація серверної логіки та CRUD-операцій

На серверній частині використано паттерн Репозиторій та Unit of Work для чіткого розділення обов'язків і управління транзакціями. Репозиторії відповідають за доступ до даних конкретних сутностей (наприклад, зупинок чи транспортних засобів), а Unit of Work забезпечує координацію транзакцій між кількома репозиторіями. Серверна логіка реалізована на платформі .NET із використанням FastEndpoints для створення швидких API-ендпоінтів та MediatR для обробки запитів і команд. Наприклад, ендпоінт для створення транспортного засобу включає валідацію через FluentValidation, мапінг через AutoMapper та збереження даних через EF Core.

API підтримує повний набір CRUD-операцій для всіх сутностей, включаючи:

- **GetList:** Отримання списку сутностей із підтримкою фільтрації та пагінації.
- **Autocomplete:** Надання спрощених даних для автодоповнення в інтерфейсі.
- **GetById:** Отримання детальної інформації про сутність за ідентифікатором.
- **Create/Update:** Створення або оновлення сутностей із валідацією.

Серверна частина розгорнута в Azure Container Apps, що забезпечує масштабованість та стабільність роботи.

2.3. Реалізація фронтенд частини

Фронтенд адміністративного веб-порталу розроблено на базі Angular із використанням TypeScript, RxJS та Material Design. Для управління взаємодією з API створено базовий сервіс, який інкапсулює всі CRUD-операції. Цей сервіс

відповідає за відправлення запитів до серверної частини, обробку відповідей та їх мапінг на моделі для фронтенду. Для спрощення роботи з формами використано FormControl та FormGroup з Angular Reactive Forms, а також реалізовано мапери для перетворення DTO-моделей у формати, придатні для відображення в інтерфейсі, і навпаки.

Інтерфейс порталу включає:

- **Модальні вікна** для створення та редагування сутностей (наприклад, транспортних засобів чи маршрутів). Модальні вікна використовують компоненти Material Design для забезпечення зручного та сучасного вигляду.
- **Таблиці** для відображення списків даних із підтримкою сортування, фільтрації та дій (редагування, видалення). Таблиці створено з використанням компонентів Material Design, таких як mat-table.

Після перевірки коректності взаємодії фронтенду з сервером (отримання, оновлення та збереження даних) було написано юніт-тести для перевірки маперів, валідаторів та бізнес-логіки.

2.4. Реалізація модуля управління маршрутами

Модуль управління маршрутами є ключовим компонентом системи "FlexTrans". Маршрут визначено як масив зупинок із теоретичними часами прибуття (departure times), що формують розклад. На початковому етапі маршрути відображалися в табличному вигляді, де кожна зупинка була рядком, а стовпці представляли варіації часу прибуття. Для реалізації таблиці використано кастомні компоненти Angular замість стандартних таблиць, із застосуванням стилів CSS для створення візуального табличного представлення. Це рішення забезпечило гнучкість у відображенні даних та їх редагуванні.

2.4.1. Створення та конфігурація маршрутів

Процес створення маршруту включає наступні етапи:

- Вказання назви маршруту, його типу (Regular або Virtual) та кольору лінії для візуального відображення на карті чи в інших компонентах системи.
- Налаштування варіацій маршруту. Кожна варіація визначає розклад, дні тижня, коли маршрут активний, та періоди неактивності (наприклад, святкові дні). Варіації дозволяють створювати схожі маршрути з мінімальними відмінностями, що спрощує планування.

Кожна варіація містить масив сегментів, які поділяються на два типи:

- **Regular:** Водій зобов'язаний відвідати всі зупинки сегмента в заданому порядку.
- **Zone:** Зупинки відвідуються лише за потребою пасажирів, що забезпечує гнучкість маршруту.

Кожен сегмент складається з масиву зупинок, де кожна зупинка має унікальне поле `sequence` для визначення порядку. Для Regular-сегментів обов'язково вказуються часи прибуття (`departure times`), які валідуються за допомогою кастомних валідаторів. Наприклад, час прибуття до наступної зупинки має бути більшим за попередній, що перевіряється як на фронтенді (через Angular Reactive Forms), так і на бекенді (через FluentValidation).

2.4.2. Валідація та відображення маршрутів

Для забезпечення коректності даних реалізовано валідацію:

- На фронтенді: Використано Angular Reactive Forms із кастомними валідаторами для перевірки формату часу, унікальності `sequence` та інших параметрів.
- На бекенді: FluentValidation перевіряє бізнес-логіку, наприклад, відповідність типу сегмента та наявність обов'язкових полів для Regular-сегментів.

Після налаштування маршруту користувач може:

- Переглянути маршрут на карті за допомогою [Google Maps API або іншого інструменту для візуалізації].
- Скопіювати варіацію маршруту для створення нової з мінімальними змінами.
- Зберегти маршрут у режимі Draft (чернетка, не відображається в інших модулях) або Publish (доступний для використання водіями та в інших частинах системи).

2.4.3. Інтерфейс управління маршрутами

Інтерфейс модуля управління маршрутами включає:

- Форму для створення/редагування маршруту з полями для назви, типу, кольору та варіацій.
- Кастомну таблицю для відображення зупинок і часу прибуття у вигляді розкладу. Таблиця підтримує редагування на місці (in-place editing) та додавання/видалення сегментів.
- Інтеграцію з картою для візуалізації маршрутів і зупинок.

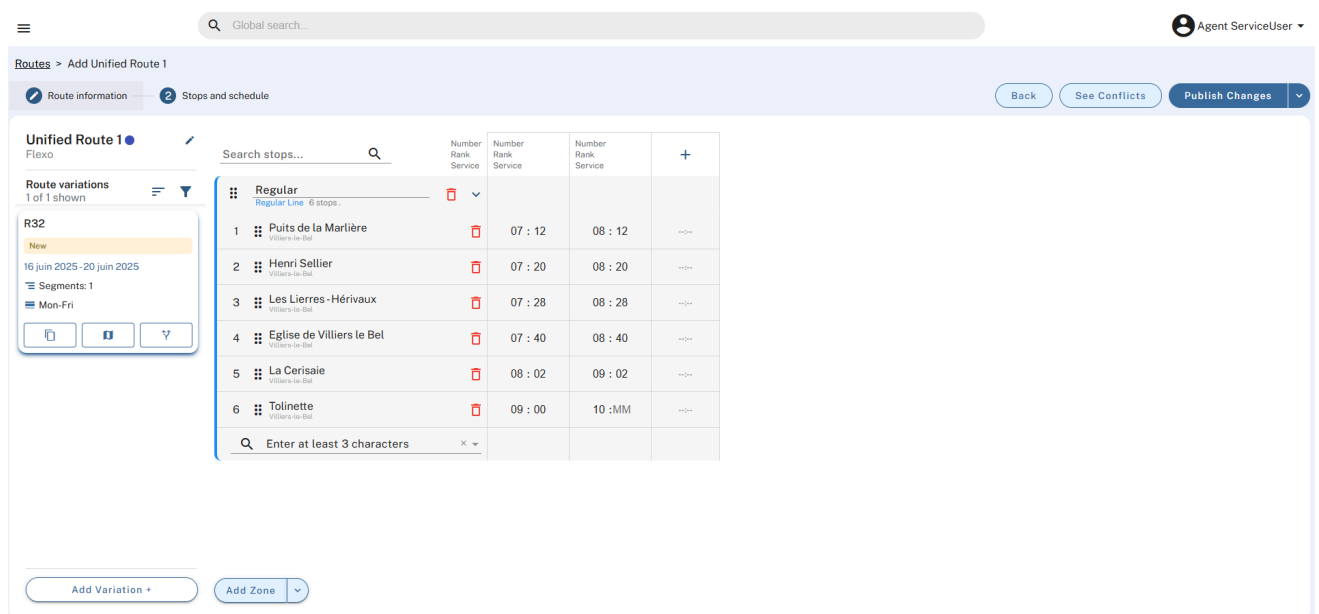


Рисунок 1 – Інтерфейс створення та редагування маршрутів.

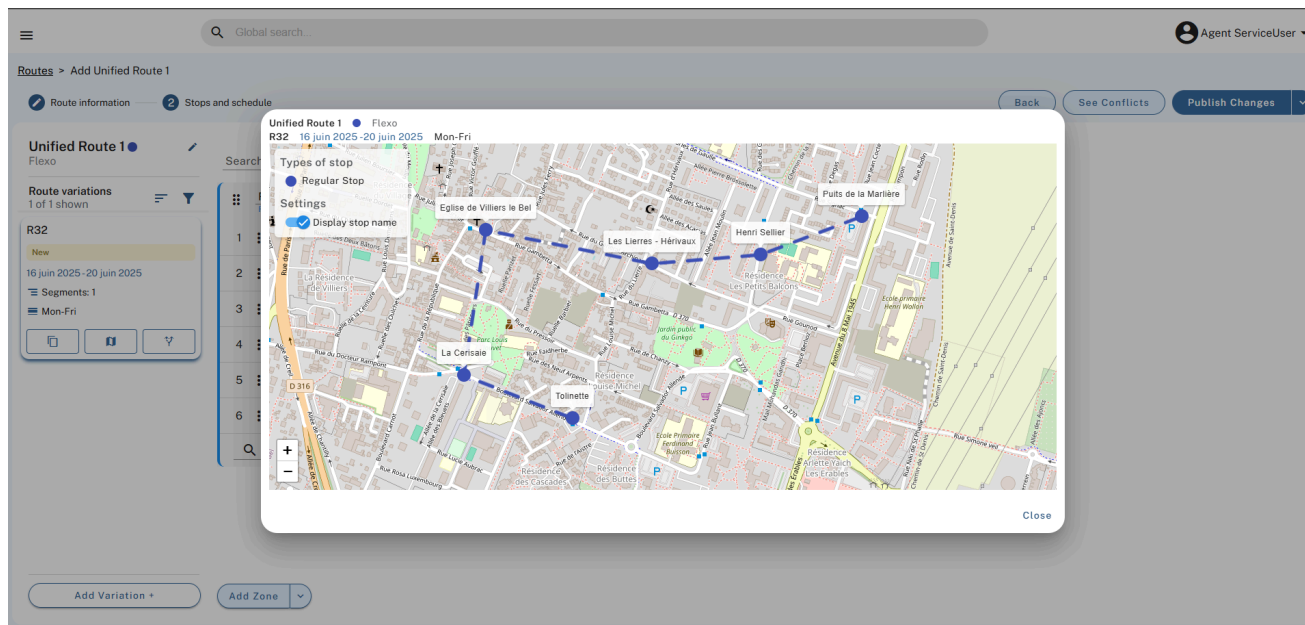


Рисунок 2 – Інтерфейс перегляду маршрутів на мапі .

2.5. Тестування та оптимізація

Для забезпечення якості модуля управління маршрутами проведено юніт-тестування маперів, валідаторів та бізнес-логіки на фронтенді та бекенді. На фронтенді тести перевіряли коректність роботи форм, мапінгу даних та відображення таблиць. На бекенді тести охоплювали валідацію DTO, збереження даних та бізнес-логіку. Інтеграційні тести підтвердили коректність взаємодії між фронтендом, API та базою даних.

Тестування в реальному середовищі проводилося користувачами, які створювали маршрути, додавали зупинки та перевіряли їх відображення на карті. Зворотний зв'язок від користувачів вказав на необхідність оптимізації швидкості завантаження великих маршрутів. Виявлені помилки виправлено, а система оптимізована для забезпечення стабільної роботи.

2.6. Висновки щодо реалізації

Розробка модуля управління маршрутами та інших компонентів адміністративного веб-порталу дозволила створити гнучку та зручну систему для планування транспортних маршрутів. Використання Angular, Material Design та

RxJS забезпечило сучасний і адаптивний інтерфейс, а .NET із FastEndpoints, AutoMapper та FluentValidation — стабільну та масштабовану серверну частину. Реалізація кастомних компонентів для табличного відображення маршрутів дозволила уникнути обмежень стандартних таблиць і забезпечила гнучкість. Валідація на всіх рівнях (фронтенд, бекенд) та тестування підтвердили надійність системи.

РОЗДІЛ 3

РЕЗУЛЬТАТИ ВИПРОБУВАНЬ

Для тестування розробленої системи "FlexTrans" програмне забезпечення було розгорнуто в хмарному середовищі Microsoft Azure, що забезпечує гнучке масштабування, високу доступність та управління сервісами.

Microsoft Azure використано для розгортання системи з наступними сервісами:

- **Azure SQL Database:** Керована реляційна база даних для зберігання даних про маршрути, зупинки, поїздки, користувачів тощо. Використовується план служби Standard S0: 10 DTU, 250 ГБ сховища, що забезпечує автоматичне оновлення, резервне копіювання та моніторинг без участі користувача.
- **Azure Static Web Apps:** Використовується для розгортання фронтенду адміністративного веб-порталу, забезпечуючи швидке завантаження сторінок та високу доступність.
- **Azure Container Apps:** Застосовано для розгортання серверної частини (API), що забезпечує масштабованість та стабільну роботу. Використовується стек часу виконання .NET 9, операційна система Linux, план служби Standard S1: 100 ACU, 1.75 ГБ пам'яті.
- **Azure Key Vault:** Використовується для безпечного зберігання секретів, таких як ключі API, паролі та сертифікати.

Після розгортання системи в Azure було проведено заповнення бази даних тестовими даними. За допомогою адміністративного веб-порталу створено тестових маршрутів, зупинок, поїздок та користувачів. Кожен елемент системи, як-от маршрути чи зупинки, було доповнено текстовими описами та географічними координатами та іншими даними.

Для перевірки функціональності системи проведено наступні види тестування:

- **Unit-тести:** Перевірка всіх API-ендпоінтів, бізнес-логіку та мапінг даних. Тести виконано з використанням xUnit.
- **Інтеграційні тести:** Перевірка взаємодії між модулями серверної частини та операцій створення, редагування та видалення сутностей (маршрутів, зупинок, поїздок тощо).
- **Тестування в реальному середовищі:** Система була протестована реальними користувачами (адміністраторами транспортних агенцій), які виконували операції планування маршрутів, управління зупинками та аналізу статистики. Користувачі залишали відгуки та повідомляли про знайдені помилки.

Тестування підтвердило стабільну роботу системи, правильність архітектури бази даних, ефективність обраного технологічного стеку. Зокрема, API-ендпоінти обробляли запити швидко, а інтерфейс порталу виявився зручним для користувачів завдяки Material Design та локалізації.

Виявлені недоліки включали необхідність додаткових валідаційних механізмів для введення даних, наприклад, перевірки формату координат зупинок. Зворотний зв'язок від користувачів вказав на потребу оптимізації інтерфейсу для певних сценаріїв, наприклад, швидшого пошуку маршрутів та відображення зупинок на мапі. У перспективі планується вдосконалення системи шляхом додавання резервувань та відстеження водіїв у реальному часі, а також мобільний додаток для пасажиро-клієнта.

ВИСНОВКИ

Проведено аналіз інформаційних систем для управління громадським транспортом, виявлено значне поширення систем, що дозволяють гнучко адаптувати маршрути до потреб пасажирів. У якості прикладу для реалізації обрано систему "FlexTrans", яка забезпечує ефективне управління транспортною інфраструктурою в районах із низьким пасажиропотоком.

Визначено програмні засоби та технології (Angular, .NET, Azure, Docker, Azure DevOps), а також API-інтерфейси, які дозволяють розробити адміністративний веб-портал, серверну частину та базу даних для управління транспортною системою. Ці засоби використано під час створення системи.

Спроектовано, реалізовано та випробувано адміністративний веб-портал, серверну частину (API-сервіс) та базу даних Azure SQL. Система отримала необхідні засоби управління безпекою, розподілу ролей (адміністратори, водії, глобальні адміністратори). Реалізовано інтерфейс для планування поїздок, управління маршрутами, зупинками, транспортними засобами та користувачами.

Систему випробувано шляхом розгортання в хмарному середовищі Azure з використанням Azure Static Web Apps, Azure Container Apps та Azure SQL Database. Проведено unit- та інтеграційні тести, а також тестування в реальному середовищі реальними користувачами, які залишили відгуки та повідомили про помилки. Результати випробувань підтвердили працездатність системи, правильність обраної архітектури та достатню ефективність використаних технологій. Виявлено потребу в додаткових валідаційних механізмах та оптимізації інтерфейсу.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Стратегія розвитку інформаційного суспільства в Україні. Розпорядження Кабінету Міністрів України від 15 травня 2013 р. № 386-р. // Урядовий портал. URL: <https://www.kmu.gov.ua/npas/246420577>
2. Стратегія кібербезпеки України. Указ Президента України від 14 травня 2021 року // Урядовий кур'єр від 28.08.2021 — № 165.
3. Microsoft Azure Documentation. Microsoft Learn. URL: <https://learn.microsoft.com/en-us/azure/>
4. Azure SQL Database Documentation. Microsoft Learn. URL: <https://learn.microsoft.com/en-us/azure/azure-sql/>
5. Azure Static Web Apps Documentation. Microsoft Learn. URL: <https://learn.microsoft.com/en-us/azure/static-web-apps/>
6. Azure Container Apps Documentation. Microsoft Learn. URL: <https://learn.microsoft.com/en-us/azure/container-apps/>
7. Azure DevOps Documentation. Microsoft Learn. URL: <https://learn.microsoft.com/en-us/azure/devops/>
8. .NET Documentation. Microsoft Learn. URL: <https://learn.microsoft.com/en-us/dotnet/>
9. Entity Framework Core Documentation. Microsoft Learn. URL: <https://learn.microsoft.com/en-us/ef/core/>
10. Angular Official Documentation. Angular. URL: <https://angular.dev/>
11. Material Design Documentation. Google. URL: <https://m3.material.io/>
12. TypeScript Documentation. TypeScript. URL: <https://www.typescriptlang.org/docs/>
13. RxJS Documentation. ReactiveX. URL: <https://rxjs.dev/>
14. Docker Documentation. Docker. URL: <https://docs.docker.com/>
15. FastEndpoints Documentation. FastEndpoints. URL: <https://fast-endpoints.com/>
16. General Transit Feed Specification: <https://gtfs.org/documentation/schedule/reference/>
17. Echarts. URL: <https://echarts.apache.org/en/index.html>