

Міністерство освіти і науки України
Криворізький національний університет
Факультет інформаційних технологій
Кафедра моделювання та програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття ступеню вищої освіти-магістра
за спеціальністю 121 – Інженерія програмного забезпечення

На тему: *«Розробка програмного забезпечення для моніторингу та управління рухомим складом залізничного транспорту промислового підприємства»*

Виконав ст. гр.ІПЗ-23-2м	_____	<u>Федік А.В.</u>
Керівник кваліфікаційної роботи	_____	<u>Стрюк А.М.</u>
Економіко- організаційна-частина роботи	_____	/ _____ /
Нормоконтроль	_____	/ _____ /
В.о. завідувача кафедри	_____	<u>Стрюк А.М.</u>

КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**Факультет:** інформаційних технологій**Кафедра:** Моделювання та програмного забезпечення**Ступінь вищої освіти:** магістр**Спеціальність:** 121 – Інженерія програмного забезпечення**ЗАТВЕРДЖУЮ**В.о. зав. кафедри: А.М.Стрюк.« » _____ 20 р.**ЗАВДАННЯ****на кваліфікаційну роботу магістра**студенту групи ІІЗ-23-2м Федіку Антону Вікторовичу

1. Тема роботи: «Розробка програмного забезпечення для моніторингу та управління рухомим складом залізничного транспорту промислового підприємства»

затверджено наказом по університету № _____ від _____.

2. Термін здачі завершеної роботи: _____

3. Вихідні данні по роботі: розроблювана система повинна працювати на базі ASP.NET Core MVC та баз даних SQL.

4. Зміст пояснювальної записки: (перелік питань, що їх треба розробити): виконати аналіз існуючих методів розв'язання задачі, дослідити проблеми та стратегії впровадження програмного забезпечення у залізничний транспорт, розробити програмне забезпечення на основі вхідних даних.

5. Перелік ілюстративного матеріалу: блок-схеми розроблених алгоритмів, знімки екранних форм.

6. Календарний план:

№	Етапи роботи	Термін виконання
1	<i>Вступ</i>	<i>28.07.24</i>
2	<i>Розділ 1</i>	<i>15.08.24</i>
3	<i>Розділ 2</i>	<i>15.09.24</i>
4	<i>Розділ 3</i>	<i>10.10.24</i>
5	<i>Висновки</i>	<i>10.11.24</i>
6	<i>Оформлення кваліфікаційної роботи</i>	<i>18.11.24</i>
7	<i>Підготовка презентації та графічного матеріалу</i>	<i>29.11.24</i>
8	<i>Підготовка доповіді до захисту</i>	<i>01.12.24</i>

6. Дата видачі завдання: _____.

Студент _____ /Федік А.В. /

Керівник _____ /Стрюк А.М. /

АННОТАЦІЯ

Федік А.В. Розробка програмного забезпечення для моніторингу та управління рухомим складом залізничного транспорту промислового підприємства.

Кваліфікаційна робота на здобуття ступеню вищої освіти магістр зі спеціальності 121 – Інженерія програмного забезпечення. – Криворізький національний університет, Кривий Ріг, 2024.

Дане дослідження присвячене вивченню та впровадженню програмного забезпечення для моніторингу та керування транспортними засобами залізничного транспорту промислових підприємств.

Залізничний транспорт завжди відігравав важливу роль у перевезенні вантажів і пасажирів в Україні, і такі компанії, як ПАТ «АрселорМіттал Кривий Рі», є одними з найбільших і найважливіших учасників цієї галузі.

Мета дослідження – проаналізувати сучасні технологічні рішення та підходи до впровадження програмного забезпечення моніторингу та керування транспортними засобами залізничного транспорту на промислових підприємствах, а також їх впровадження в розрізі промислових підприємств. Також визначення переваг та можливостей впровадження програмного забезпечення, а також можливих викликів і перешкод для вдосконалення залізничного транспорту в Україні.

Впровадження ПЗ для моніторингу та управління парком транспортних засобів для компаній залізничного транспорту може не тільки допомогти підвищити продуктивність і безпеку транспортування, але й зменшити негативний вплив на навколишнє середовище та оптимізувати витрати.

У дослідженні висвітлено потенціал ПЗ та надано рекомендації щодо впровадження сучасних програм управління парком для подальшого розвитку промислових підприємств та вдосконалення залізничної галузі всієї України.

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, КОНТРОЛЛЕР, ПАТ АМКР, РУХОМИЙ СКЛАД, РУХ, ВАГОН, БАЗА ДАНИХ, ЦЕХ, ТАБЛИЦЯ.

ANNOTATION

Fedik A.V. Development of software for monitoring and management of rolling stock of railway transport of an industrial enterprise.

Qualification work for obtaining the degree of higher education master in specialty 121 - Software Engineering. - Kryvyi Rih National University, Kryvyi Rih, 2024.

This study is devoted to the study and implementation of software for monitoring and management of railway transport vehicles of industrial enterprises.

Railway transport has always played an important role in the transportation of goods and passengers in Ukraine, and companies such as PJSC "ArcelorMittal Kryvyi Rih" are among the largest and most important participants in this industry.

The purpose of the study is to analyze modern technological solutions and approaches to the implementation of software for monitoring and management of railway transport vehicles at industrial enterprises, as well as their implementation in the context of industrial enterprises. Also, the identification of the advantages and opportunities of implementing software, as well as possible challenges and obstacles to improving rail transport in Ukraine.

Implementing software for monitoring and managing a fleet of vehicles for rail transport companies can not only help increase productivity and safety of transportation, but also reduce the negative impact on the environment and optimize costs.

The study highlights the potential of software and provides recommendations for the implementation of modern fleet management programs for the further development of industrial enterprises and the improvement of the railway industry throughout Ukraine..

SOFTWARE, CONTROLLER, PJSC AMKR, ROLLING STOCK, TRAFFIC, WAGON, DATABASE, WORKSHOP, TABLE.

ЗМІСТ

ВСТУП	9
1 ТЕОРЕТИЧНИЙ АНАЛІЗ І ОЦІНКА ЗАЛІЗНИЧНОГО ТРАНСПОРТУ НА ПРИКЛАДІ ПАТ “АМКР”	10
1.1 Загальний аналіз залізничного транспорту України	10
1.1.1 Значення залізничного транспорту для України	10
1.1.2 Основні залізничні сполучення	10
1.1.3 Роль залізничного транспорту в економічному розвитку країни ...	11
1.1.4 Сучасний стан залізничної галузі України.....	12
1.2 Характеристика підприємства	13
1.2.1 Історичний розвиток компанії	13
1.2.2 Вплив ПАТ “АМКР” на економіку країни та регіональний розвиток.	14
1.2.3 Організаційна структура підприємства	15
1.3. Актуальність ПЗ в транспортному секторі.....	16
1.3.1 Важливість і роль автоматизації в сучасних транспортних системах	16
1.3.2 Виклики та переваги впровадження ПЗ управління рухомим складом	17
1.4 Оцінка поточного стану транспортних процесів	18
1.5 Аналіз існуючих програмних рішень.....	20
Висновки до розділу	28
2 АНАЛІЗ ПРОБЛЕМ ТА СТРАТЕГІЇ ВПРОВАДЖЕННЯ ПЗ В УПРАВЛІННІ РУХОМИМ СКЛАДОМ	29
2.1 Основні проблеми управління рухомим складом.....	29
2.1.1 Роль системи зв’язку в управлінні рухомим складом.....	29
2.1.2 Пропускна спроможність зв’язкових систем	31
2.1.3 Захист систем зв’язку від зовнішніх впливів	32
2.2 Технічні аспекти впровадження ПЗ рухомого складу	33
2.2.1 Інтеграція ПЗ із сучасними залізничними технологіями.....	33
2.2.2 Технічні стандарти та вимоги для ПЗ	40
2.3 Програмні особливості забезпечення	55
2.3.1 Огляд ПЗ для контролю рухомого складу.....	55
2.3.2 Інструменти для збору та аналізу інформації у ПЗ залізничного транспорту	56

	7
2.4 Перспективи розвитку ПЗ для рухомих складів	57
2.4.1 Сучасні тенденції ПЗ транспортних технологій.....	57
2.4.2 Новітні технології для вдосконалення ПЗ.....	59
Висновки до розділу	60
3 РЕАЛІЗАЦІЯ ПЗ УПРАВЛІННЯ РУХОМИМ СКЛАДОМ	62
3.1 Обґрунтування вибору технологій для розробки програмного забезпечення	62
3.2 Проектування архітектури програмного забезпечення керування рухомих складом	65
3.3 Створення бази даних системи.....	62
3.4 Реалізація програмного забезпечення.....	67
Висновки до розділу	80
ВИСНОКИ ТА РЕКОМЕНДАЦІЇ.....	81
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	83
<i>ДОДАТОК А</i>	86
<i>ДОДАТОК Б</i>	88

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

Скорочення:

АМКР - АрселорМіттал Кривий Ріг

ПЗ – програмне забезпечення.

БД – база даних.

ГІС - географічні інформаційні системи

СЦБ - системи сигналізації, централізації, блокування.

АБС - автоматизовані блокувальні системи

VMS - video management software

ETCS - European Train Control System

GSM-R - Global System for Mobile Communications – Railway

IP - Internet Protocol

TCP - Transmission Control Protocol

SNMP - Simple Network Management Protocol

MQTT - Message Queuing Telemetry Transport

MVC - Model-View-Controller

M2M - Machine-to-Machine

Умовні позначення:

ELD - електронні журнали водіїв

ААСІС - Азія, Африка, країни СНД

TMS - системи керування перевезеннями

HIS - безпечний інтерфейс користувача

РАВ - система контролю габариту

RFID - технологія автоматичної ідентифікації та збору даних.

LTE - стандарт бездротового зв'язку четвертого покоління (4G)

GSM - стандарт для мобільних телефонних мереж

ID – унікальна ознака об'єкта.

WDM - технологія передачі інформації в оптичних волоконних мережах

ВСТУП

Залізничний транспорт відіграє ключову роль у світовій інфраструктурі, забезпечуючи перевезення вантажів і пасажирів. Його значення для економіки країни важко переоцінити, оскільки залізниці дозволяють організовувати масштабні та ефективні перевезення на далекі відстані. Одним із провідних підприємств України у сфері гірничо-металургійного виробництва є ПАТ "АрселорМіттал Кривий Ріг" (АМКР). З часу заснування воно відіграє важливу роль у зміцненні економіки та розвитку металургійної галузі.

Тема "Розробка програмного забезпечення для моніторингу та управління рухомим складом залізничного транспорту промислового підприємства" є вкрай актуальною, оскільки вона сприяє подальшому розвитку цього великого виробничого об'єкта. Впровадження відповідного програмного забезпечення не лише підвищує продуктивність і безпеку, але й дозволяє зменшити екологічні ризики та оптимізувати використання ресурсів. Використовуючи сучасні технології та аналітичні інструменти, можливо ефективніше планувати маршрути, скорочувати витрати на паливе та знижувати викиди.

Дослідження в межах дипломного проекту має на меті виявити потенціал використання програмного забезпечення для управління рухомим складом на ПАТ "АрселорМіттал Кривий Ріг" і розробити рекомендації для впровадження новітніх систем управління. Розвиток залізничного транспорту в Україні та забезпечення конкурентоспроможності підприємства потребують інноваційних рішень і сучасних технологій.

Таким чином, створення програмного забезпечення для моніторингу та управління залізничним транспортом є не лише актуальним, але й стратегічно важливим завданням. Ефективна організація цього процесу може позитивно вплинути як на діяльність підприємства, так і на економіку країни в цілому.

1 ТЕОРЕТИЧНИЙ АНАЛІЗ І ОЦІНКА ЗАЛІЗНИЧНОГО ТРАНСПОРТУ НА ПРИКЛАДІ ПАТ “АМКР”

1.1 Загальний аналіз залізничного транспорту України

1.1.1 Значення залізничного транспорту для України

Залізничний транспорт в Україні [1] почав активно розвиватися у середині ХІХ століття. У 1865 році було відкрито першу залізничну лінію, яка з'єднала Харків із Балтійським морем. Ця подія стала значним кроком для покращення торгівлі та перевезення вантажів, а також дала поштовх розвитку міст і промисловості регіону. Завдяки залізничному транспорту стало можливим ефективно перевозити вугілля, залізо, сільськогосподарську продукцію та промислові товари, що сприяло зростанню економіки, розширенню підприємств і урбанізації.

Після утворення Радянського Союзу українська залізнична інфраструктура була інтегрована в загальносоюзну транспортну систему. Продовжуючи свій розвиток, залізниці стали важливим інструментом перевезення товарів і військової техніки під час Другої світової війни.

З набуттям незалежності у 1991 році Україна зберегла залізничний транспорт як одну з ключових галузей. Почалася модернізація та реформа залізничної інфраструктури, орієнтована на підвищення її ефективності та конкурентоспроможності на сучасному ринку.

1.1.2 Основні залізничні сполучення

"Укрзалізниця" [1] — це найбільший залізничний оператор України, який керує майже всією залізничною інфраструктурою країни. Компанія перебуває у державній власності й відповідає за організацію роботи та утримання залізничної мережі. Інфраструктура "Укрзалізниці" охоплює територію всієї України, забезпечуючи сполучення між більшістю областей та ключовими містами. Крім того, залізниця інтегрована в міжнародні транспортні коридори, що сприяє зручному транзиту товарів між Європою та

Азією.

Численні промислові підприємства в Україні мають власну залізничну інфраструктуру для перевезення сировини та продукції. Наприклад, ПАТ "АрселорМіттал Кривий Ріг" володіє розгалуженою внутрішньою мережею, яка використовується для транспортування залізної руди та виробів зі сталі.

Україна є важливою складовою Південного транспортного коридору, що сполучає Європу з Туреччиною та країнами Азії через території Грузії та Туреччини. Цей маршрут має значну роль у перевезенні вантажів. Також Україна виступає ключовим транзитним пунктом на шляху між Північною Європою та Середземномор'ям. Цей напрямок, який проходить через Україну та Білорусь, забезпечує важливий зв'язок між західною та східною частинами Європи.

1.1.3 Роль залізничного транспорту в економічному розвитку країни

Залізничний транспорт в Україні відіграє ключову роль у перевезенні вантажів [2], виступаючи одним із основних видів транспорту для цієї мети. Завдяки залізницям можна ефективно та в значних обсягах транспортувати різноманітні вантажі: від сировини та вугілля до зерна, металопродукції, палива, хімічних речовин і багатьох інших товарів. Це сприяє стабільній роботі промислових підприємств і забезпеченню їх необхідними ресурсами.

Окрім вантажних перевезень, залізничний транспорт активно використовується для перевезення пасажирів, пропонуючи комфортний спосіб пересування на великі відстані. Це сприяє інтеграції регіонів України, розвитку економічних і культурних зв'язків між ними.

Залізнична галузь є вагомим чинником для національної економіки. Вона забезпечує надходження значних доходів від перевезень, створює тисячі робочих місць і робить суттєвий внесок у ВВП країни. Крім того, залізничний транспорт тісно пов'язаний з іншими секторами економіки, такими як сільське господарство, промисловість і логістика. Завдяки залізницям відбувається

перевезення аграрної продукції, сировини для виробництва та готових товарів.

Україна має стратегічне значення як транзитна держава для вантажоперевезень між Європою та Азією. Залізничний транспорт забезпечує ефективний транзит товарів, що сприяє розширенню міжнародної торгівлі та зміцненню внутрішнього ринку.

Залізнична інфраструктура сприяє розвитку регіонів, підтримуючи інвестиційні проекти та стимулюючи економічне зростання. До того ж, залізничний транспорт є екологічно чистішим порівняно з автомобільним або авіаційним транспортом. Його використання знижує екологічне навантаження, що позитивно впливає на стан довкілля.

1.1.4 Сучасний стан залізничної галузі України.

Залізнична мережа України [3] належить до найбільших у Європі, охоплюючи тисячі кілометрів колій, які забезпечують сполучення всередині країни та інтеграцію з міжнародними транспортними коридорами. Проте значна частина інфраструктури потребує модернізації та ремонту.

Зокрема, парк локомотивів і вагонів зазнає старіння, і багато одиниць потребують заміни або капітального ремонту, що негативно впливає на надійність та ефективність перевезень. Тарифна політика, яка регулюється державою, визначає вартість перевезень і значно впливає на конкурентоспроможність залізничного транспорту порівняно з автомобільним та морським видами транспорту.

Залізнична галузь поступово впроваджує сучасні інформаційні технології, які допомагають оптимізувати управління рухомим складом, моніторинг вагонів і маршрутів, а також покращувати логістику й безпеку. Завдяки своєму стратегічному розташуванню, Україна залишається важливим транзитним пунктом у міжнародних торговельних потоках між Європою та Азією, де залізничний транспорт виконує провідну роль.

Для підвищення ефективності та конкурентоспроможності Україна проводить реформи у сфері залізничного транспорту. Це передбачає

модернізацію інфраструктури, удосконалення логістичних процесів і впровадження нових підходів до управління.

Попри значні виклики, такі як обмежені фінансові ресурси, конкуренція з іншими видами транспорту, необхідність оновлення інфраструктури та забезпечення безпеки руху, залізничний транспорт залишається критично важливою галуззю. Впровадження сучасних технологій і системних реформ є ключем до його сталого розвитку.

1.2 Характеристика підприємства

1.2.1 Історичний розвиток компанії

Історія компанії ПАТ "АрселорМіттал Кривий Ріг" [4] починається з 1931 року, коли було засновано Криворізький гірничий комбінат. У перші роки свого існування підприємство зосереджувалось на видобутку та переробці залізної руди. У період Другої світової війни комбінат активно забезпечував фронтові потреби, а після її завершення розпочалася масштабна реконструкція, яка включала будівництво нових виробничих об'єктів і збільшення обсягів видобутку.

Після розпаду Радянського Союзу комбінат був приватизований і зазнав структурних змін, що відкрило шлях до впровадження іноземних інвестицій та сучасних технологій. У 2005 році підприємство увійшло до складу ArcelorMittal, найбільшого світового гравця у галузі виробництва сталі. Це дало можливість модернізувати обладнання, інтегрувати інноваційні рішення та розширити доступ до міжнародних ринків.

Сьогодні "АрселорМіттал Кривий Ріг" залишається лідером у виробництві чорних металів в Україні, спеціалізуючись на виготовленні арматурної сталі, катанки та інших металопродуктів. Виробничий процес охоплює повний цикл — від видобутку залізної руди до отримання готової продукції. У структурі підприємства функціонують коксохімічне, гірничо-збагачувальне та металургійне виробництва, що включає аглодоменне, сталеплавильне і прокатне підрозділи.

Компанія "АрселорМіттал" має глобальний масштаб діяльності, працюючи у 18 країнах на чотирьох континентах. Її продукція експортується у 160 держав світу, забезпечуючи потреби таких галузей, як автомобілебудування, будівництво, виробництво побутової техніки та упаковки. Завдяки великим запасам сировини та ефективній логістиці "АрселорМіттал" залишається світовим лідером у виробництві сталі.

Група організаційно поділена на кілька основних сегментів, серед яких плоский і довгий вуглецевий прокат для Америки, Європи та інших регіонів, а також гірничодобувний сегмент. Додатково функціонують допоміжні відділи, які займаються постачанням, енергозабезпеченням, інформаційними технологіями та іншими напрямками.

ПАТ "АрселорМіттал Кривий Ріг" не лише залишається важливим промисловим центром, але й активно долучається до екологічних і соціальних ініціатив, сприяючи розвитку регіону та забезпечуючи високі стандарти безпеки і технологічності у виробництві.

1.2.2 Вплив ПАТ "АМКР" на економіку країни та регіональний розвиток.

"АрселорМіттал Кривий Ріг" є одним із найбільших роботодавців в Україні [5, 6], забезпечуючи тисячі робочих місць для мешканців регіону. Підприємство залучає до роботи фахівців різних рівнів, включаючи інженерів, технічний персонал та робітників виробничого сектору. Компанія регулярно інвестує значні кошти в оновлення та вдосконалення своїх виробничих потужностей, що позитивно впливає на продуктивність і конкурентоспроможність гірничо-металургійної галузі України.

Продукція компанії експортується на міжнародні ринки, що забезпечує валютні надходження до країни та підтримує її експортний потенціал. Крім того, "АМКР" є важливим платником податків, спрямовуючи значні суми до державного бюджету. Ці кошти використовуються для фінансування соціальних, інфраструктурних і регіональних проєктів. Водночас працівники компанії сприяють розвитку місцевої економіки через споживання товарів та

послуг.

Компанія активно інвестує у вдосконалення регіональної інфраструктури, зокрема доріг, логістичних об'єктів і комунікацій, що підвищує якість життя населення та ефективність транспортування. Також "АМКР" реалізує соціальні ініціативи, спрямовані на розвиток освіти, медицини, культури та підтримку місцевих громад.

На внутрішньому ринку України компанія забезпечує високу якість сталі та чорних металів, що сприяє зниженню залежності країни від імпортової продукції та стимулює розвиток національної промисловості. Розташування виробничих об'єктів у Кривому Розі та Дніпропетровській області позитивно впливає на економічний і соціальний розвиток регіону, створюючи нові можливості для місцевого населення та підприємств.

Таким чином, ПАТ "АрселорМіттал Кривий Ріг" відіграє значну роль у розвитку економіки України, підтримці регіонального прогресу та покращенні добробуту місцевих громад. Це підкреслює важливість гірничо-металургійного сектору як стратегічної складової національної економіки.

1.2.3 Організаційна структура підприємства

Підприємство складається з багатьох взаємопов'язаних департаментів, які забезпечують безперервність виробничих процесів. [5, 6] Однак, повного доступу до структурної схеми нам надано не було, тому можна лише узагальнити інформацію про ті підрозділи, дані про які є у відкритому доступі:

Виробничий блок:

- Гірничо-збагачувальний сегмент включає видобуток залізної руди, шахтне управління та гірничо-збагачувальні підприємства.
- Металургійний сегмент поділяється на:
 - Аглодоменний департамент, що складається з агломераційного цеху металургійного виробництва та агломераційних цехів №1, 2 і 3, а також доменних цехів №1 і 2.
 - Сталеплавильний департамент, до якого входять конвертерний цех, мартенівський цех, цех підготовки складів, вогнетривко-

вапняний цех, копровий цех і цех ремонту металургійних печей.

- Прокатний департамент, що об'єднує цех блюмінгу, сортопрокатний цех №1, прокатний цех №3, вальцетокарний цех та підрозділи, що займаються переробкою металопродукції.
- Коксохімічний сегмент охоплює вуглепідготовчий цех, коксовий цех №1, цех уловлювання, цех очищення від сірки та інші підрозділи, спеціалізовані на виробництві коксу та продуктів його переробки.

Допоміжні підрозділи:

Компанія також має низку допоміжних структур, які забезпечують її стабільну роботу. Серед них:

- Підрозділи з постачання матеріалів та енергії,
- Закупівлі,
- Інформаційні технології,
- Юридичний відділ,
- Управління нерухомістю,
- Відділи з реалізації другорядної продукції.

Ці допоміжні підрозділи забезпечують координацію виробничих процесів та підтримують функціонування підприємства, сприяючи підвищенню ефективності його діяльності.

1.3. Актуальність ПЗ в транспортному секторі

1.3.1 Важливість і роль автоматизації в сучасних транспортних системах

ПЗ сприяє оптимізації використання рухомого складу [7], дозволяє більш ефективно планувати маршрути та розподіляти ресурси, що скорочує час у дорозі та витрати на пальне. Воно допомагає ідентифікувати потенційні загрози на шляху, керувати транспортними потоками та мінімізувати ризик дорожньо-транспортних пригод. Система забезпечує ефективний контроль витрат пального, зменшує рівень викидів шкідливих речовин і знижує негативний вплив на навколишнє середовище.

Завдяки використанню таких рішень ресурси, зокрема пального, час і робочу силу, можна використовувати більш раціонально, що сприяє значній економії. Програмне забезпечення демонструє високий рівень точності та надійності, мінімізуючи людський фактор і допомагаючи уникати помилок та аварій.

Транспортні компанії, які впроваджують програмне забезпечення для управління рухомим складом, мають змогу надавати послуги швидше, надійніше й ефективніше, що підвищує їхню конкурентоспроможність. Це дозволяє зменшити витрати на обслуговування та експлуатацію транспорту, а також оптимізувати логістичні процеси й планування маршрутів. Системи підвищують рівень обслуговування клієнтів, забезпечуючи своєчасність та точність доставки.

Крім того, програмне забезпечення допомагає відповідати вимогам чинного законодавства та стандартів безпеки, знижуючи ризики та ймовірність отримання штрафів за порушення. Воно також виступає каталізатором розвитку нових технологій та інновацій у транспортній сфері, зокрема у сфері електричних і автономних транспортних засобів.

1.3.2 Виклики та переваги впровадження ПЗ управління рухомим складом

Впровадження програмного забезпечення для управління рухомим складом приносить значні переваги. Однією з ключових є підвищення ефективності транспортних операцій. [7]

Таке програмне забезпечення дозволяє точно визначати найкращі маршрути, уникати заторів і скорочувати час у дорозі. Це особливо актуально для перевезення вантажів, де кожна година має важливе значення для своєчасної доставки й оптимізації витрат.

Зменшення витрат також є однією з основних переваг. Завдяки оптимальному плануванню маршрутів і впровадженню принципів економного

водіння, можна значно скоротити витрати на паливе. До того ж системи прогнозування й попередження несправностей допомагають знизити потребу в технічному обслуговуванні й ремонті транспортних засобів.

Поліпшення рівня безпеки на дорозі є ще однією важливою перевагою. Програмне забезпечення здатне виявляти небезпечні ситуації та реагувати на них швидше, ніж це можливо для водія, що знижує ймовірність аварій і сприяє збереженню життя.

Важливим екологічним аспектом є зменшення негативного впливу на довкілля. Завдяки ефективнішому використанню транспортних засобів можна знизити рівень викидів шкідливих речовин і скоротити споживання пального.

Таким чином, використання програмного забезпечення для управління рухомим складом сприяє підвищенню продуктивності, скороченню витрат, поліпшенню безпеки та мінімізації впливу на довкілля, що робить його важливим інструментом для сучасної транспортної галузі.

1.4 Оцінка поточного стану транспортних процесів

Аналіз стану системи контролю рухомого складу на ПАТ АМКР виявив кілька важливих проблем:

- 1. Відсутність системи відстеження в реальному часі.** Неможливість оперативного моніторингу переміщення локомотивів та вагонів призводить до низької ефективності управління та потенційних затримок у роботі.
- 2. Значна кількість ручних процесів та паперової документації.** Це ускладнює управління ресурсами та суттєво знижує продуктивність.
- 3. Недостатня координація між підрозділами.** Відсутність злагодженого зв'язку між різними відділами збільшує ризик помилок та неузгодженості в роботі.

Ці проблеми негативно впливають на загальну продуктивність підприємства.

Запропоновані рішення

На основі проведеного аналізу було розроблено низку рекомендацій щодо

впровадження системи контролю рухомого складу:

1. **Встановлення GPS і супутникового моніторингу.** Впровадження сучасних систем GPS для точного відстеження руху транспорту, його місцезнаходження та оперативного реагування на події.
2. **Використання RFID-технологій.** Інтеграція RFID-систем для автоматичної ідентифікації вагонів, обліку їхнього статусу та відстеження в реальному часі.
3. **Модернізація інформаційних систем.** Оновлення існуючих платформ для автоматизації обліку рухомого складу та покращення комунікації між структурними підрозділами.
4. **Впровадження телеметрії.** Використання систем для дистанційного моніторингу технічного стану транспорту, що дозволить попереджати поломки.
5. **Навчання персоналу.** Організація тренінгів для підвищення кваліфікації працівників у користуванні новими технологіями.

Етапи впровадження

Для реалізації запропонованих заходів слід дотримуватися чіткого плану дій:

- **Закупівля обладнання.** Обрати та придбати необхідні системи (GPS, RFID тощо).
- **Інсталяція та налаштування.** Виконати монтаж і налаштування обладнання на локомотивах та вагонах.
- **Розробка програмного забезпечення.** Створити платформу для обліку та моніторингу транспорту з можливістю віддаленого доступу до даних.
- **Навчання працівників.** Провести освітні заходи для персоналу, щоб забезпечити ефективне використання нових систем.
- **Моніторинг та вдосконалення.** Регулярно аналізувати роботу системи, вносити корективи та підвищувати її ефективність.
- **Аналіз результатів.** Після запуску системи оцінити її вплив на роботу підприємства та підготувати звіти для подальшого вдосконалення.

Оцінка ефективності та ризиків

Для успішного впровадження необхідно провести економічний аналіз, розрахувати потенційні вигоди та врахувати можливі ризики. Також потрібно розробити детальний план із зазначенням графіку, бюджету та розподілом відповідальності між учасниками проекту.

Важливо забезпечити готовність усіх структурних підрозділів компанії до змін, що гарантує успішне впровадження нових технологій.

1.5 Аналіз існуючих програмних рішень

У сфері залізничного транспорту використовуються численні готові рішення для контролю та управління рухомим складом. Розглянемо деякі з них.

Комплексна система від компанії Thales

Французька компанія Thales [8] здійснює постачання комплексних систем для управління рухом поїздів і забезпечення безпечної експлуатації рухомого складу. Ці рішення охоплюють не лише системи сигналізації, централізації та блокування (СЦБ) і управління інтервалами між поїздами, але й системи зв'язку. З їх допомогою забезпечується керування технічним обладнанням, включаючи засоби моніторингу технічного стану рухомого складу в русі.

Для ефективного функціонування залізничної інфраструктури використовується інтегрований диспетчерський центр, який поєднує безпечний користувацький інтерфейс HIS, системи керування перевезеннями TMS, автоматизації та інформування Aramis, а також технічного обслуговування MSe.

Контроль вагонів, що рухаються, здійснюється через пункти моніторингу CheckPoint, які розроблені Thales. Вони оснащені чотирма типами датчиків, що збирають інформацію про стан кожного вагона. Дані передаються до концентратора, де відбувається їхній аналіз, а результати через мережу спрямовуються до центру управління. У разі виникнення тривожної ситуації сигнал автоматично передається диспетчеру. У критичних випадках

поїзд зупиняється автоматично.

На основі навантаження коліс на колію обчислюється завантаження вагонів. Тип вагона ідентифікується автоматичним датчиком, що дозволяє визначити масу тари. Нерівномірний розподіл вантажу виявляється шляхом порівняння навантаження на колеса з обох боків вагону.

Детектор сходу з рейок і юзового руху (DDED) відстежує вагони чи колісні пари, які зійшли з рейок, а також здійснює перевірку стану зчіпних пристроїв і кузова на наявність перекосів. Контроль температури буксових вузлів виконується за допомогою інфрачервоних датчиків.

Всі результати вимірювань із лінійних пунктів CheckPoint передаються в центральний пункт системи, звідки, за необхідності, інформація спрямовується до диспетчерського центру залізниці.

Конфігурація системи CheckPoint для магістралі NSR зображена на рисунку 1.1.

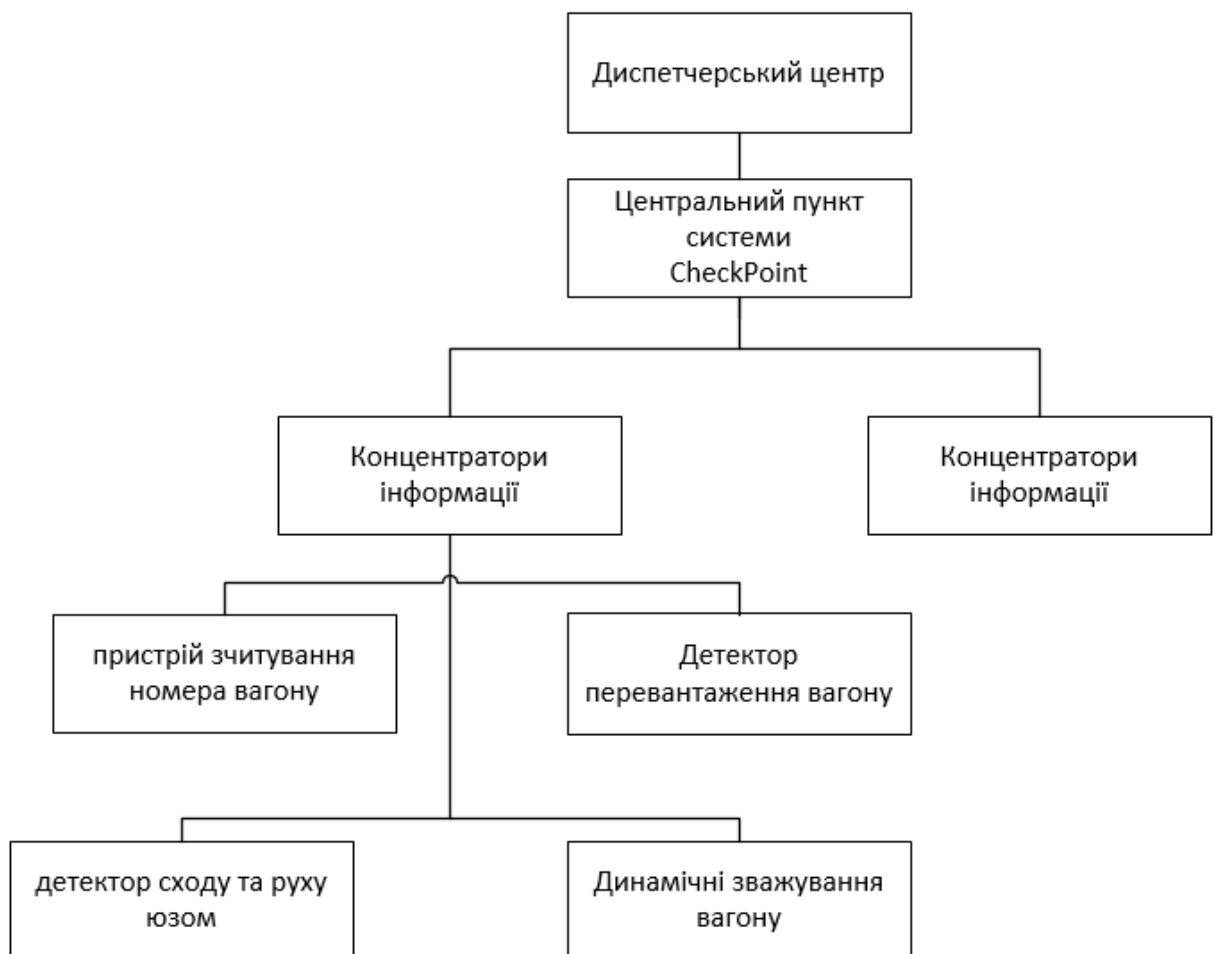


Рисунок 1.1 – Налаштування CheckPoint на основній магістральній лінії NSR

Система Sensorline



Рисунок 1.2 – Оптоволоконний датчик Sensorline

У таких країнах, як Франція, Литва, Іспанія, Південна Корея, Німеччина, Індія та Австрія, для оперативного виявлення дефектів на поверхні катання коліс рухомого складу впроваджено систему Sensorline [9]. Її застосування на залізничних мережах показало значне зниження витрат, пов'язаних із зносом вагонів і залізничної колії, а також зменшення шуму та вібрації. Це, у свою чергу, сприяє підвищенню комфорту під час пасажирських перевезень.

Система Sensorline використовує вбудовані датчики, створені на основі волоконно-оптичних технологій, для вимірювання сили, яку колесо передає на рейки. Тонкий дизайн датчиків дозволяє встановлювати їх замість стандартних гумових підкладок. Коли колесо проходить над датчиком, виникає підвищення сили реакції в системі "рейка-шпала", що призводить до деформації датчика. Зібрані дані аналізуються програмним забезпеченням протягом кількох секунд, визначаючи дефектні колеса. Отримані результати передаються до бази даних.

Інформація про несправні колеса та їхнє місцезнаходження у складі передається до диспетчерського центру через систему передачі даних. Довжина вимірювальної ділянки системи Sensorline варіюється від 4,2 до 7,8 метра. Для підвищення точності роботи передбачено можливість розширення

довжини цієї ділянки, що дозволяє здійснювати більше обертів колеса під час контролю.

Система здатна працювати на швидкостях від 35 до 350 км/год, забезпечуючи мінімальну похибку у 3%, яка досягається за умови руху зі швидкістю до 60 км/год.

TRIMBLE

Trimble Inc. [10] — велика корпорація зі США, яка займається розробкою та постачанням продуктів і послуг для геопросторового інжинірингу, конструювання, геоінформаційних систем (ГІС), агробізнесу, логістики та будівельної галузі. Заснована у 1978 році, компанія має свою штаб-квартиру в Санта-Кларіті, штат Каліфорнія.

Основні напрями діяльності Trimble:

Геодезія та геопросторові технології: компанія створює інноваційні GPS-рішення, обладнання для польових вимірювань (як на суші, так і на воді), а також програмне забезпечення для аналізу геодезичних даних.

Будівництво та інженерія: Trimble пропонує інструменти та софт для управління будівельними процесами, включаючи GPS-технології для точного вирівнювання, вимірювань і навігації на майданчиках.

Логістика та транспорт: компанія розробляє системи для моніторингу автопарків, GPS-відстеження транспорту та планування маршрутів для підприємств, що займаються перевезеннями.

Агробізнес: Trimble пропонує рішення для точного землеробства, автоматизації управління сільськогосподарською технікою та моніторингу врожайності.

Геоінформаційні системи (ГІС): компанія забезпечує своїх клієнтів програмними продуктами та обладнанням для створення, аналізу та управління геопросторовими даними.

Trimble відома своїми інноваційними підходами та постійним вдосконаленням технологій. Компанія активно працює над розвитком сучасних рішень для потреб геопросторової галузі, промисловості та

будівництва.

Одним із ключових продуктів у сфері контролю транспортного руху є система GPS-моніторингу та навігації. Ця технологія дає змогу відстежувати місцезнаходження транспорту, такого як вантажівки, вагони та інші види, оптимізувати маршрути та підвищувати ефективність перевезень.

Stp	Typ	Cust	City	Scheduled	ETA	Arrived	Departure
01	P	ANNIE3	ARCAOK	5/15 5:00	6/17 17:23	6/17 15:55	6/17 15:55
00	D	ANNIE1	YUKOOK	5/15 12:01	6/17 18:33	6/17 15:55	6/17 15:55

Рисунок 1.3 – програмний засіб компанії Trimble

OMNITRACS

Omnitracs [11] — це американська компанія, яка є лідером у сфері технологічних рішень для управління транспортними парками та логістикою. Заснована у 1988 році, вона здобула репутацію однієї з провідних організацій у галузі контролю та оптимізації роботи флотів. Штаб-квартира компанії знаходиться у місті Даллас, штат Техас, США.

До основних продуктів і послуг Omnitracs належать:

- **Системи моніторингу та навігації:** Рішення для відстеження транспорту, які включають GPS-пристрої, програмне забезпечення для маршрутного планування та отримання сповіщень у реальному часі.
- **Електронні журнали водіїв (ELD):** Інструменти для дотримання вимог регулюючих органів щодо ведення обліку робочого часу водіїв.

- **Системи безпеки для водіїв:** Технології для аналізу поведінки за кермом, таких як контроль швидкості, різких маневрів та гальмувань.
- **Управління споживанням палива:** Інструменти для оптимізації витрат на паливо шляхом аналізу маршрутів і ефективності водіння.
- **Аналітика даних:** Рішення для обробки й аналізу інформації про роботу флоту з метою зниження витрат і покращення ефективності.
- **Інтеграція систем:** Можливість поєднання рішень Omnitracs з іншими логістичними платформами та системами управління транспортом.

Omnitracs обслуговує клієнтів з різних галузей, серед яких вантажний транспорт, пасажирські перевезення та легкові автомобілі. Їхні технології полегшують керування транспортними засобами, підвищують безпеку водіїв і сприяють оптимізації логістичних операцій. Основна мета компанії — допомагати клієнтам скорочувати витрати та підвищувати ефективність завдяки інноваціям в управлінні транспортними ресурсами.

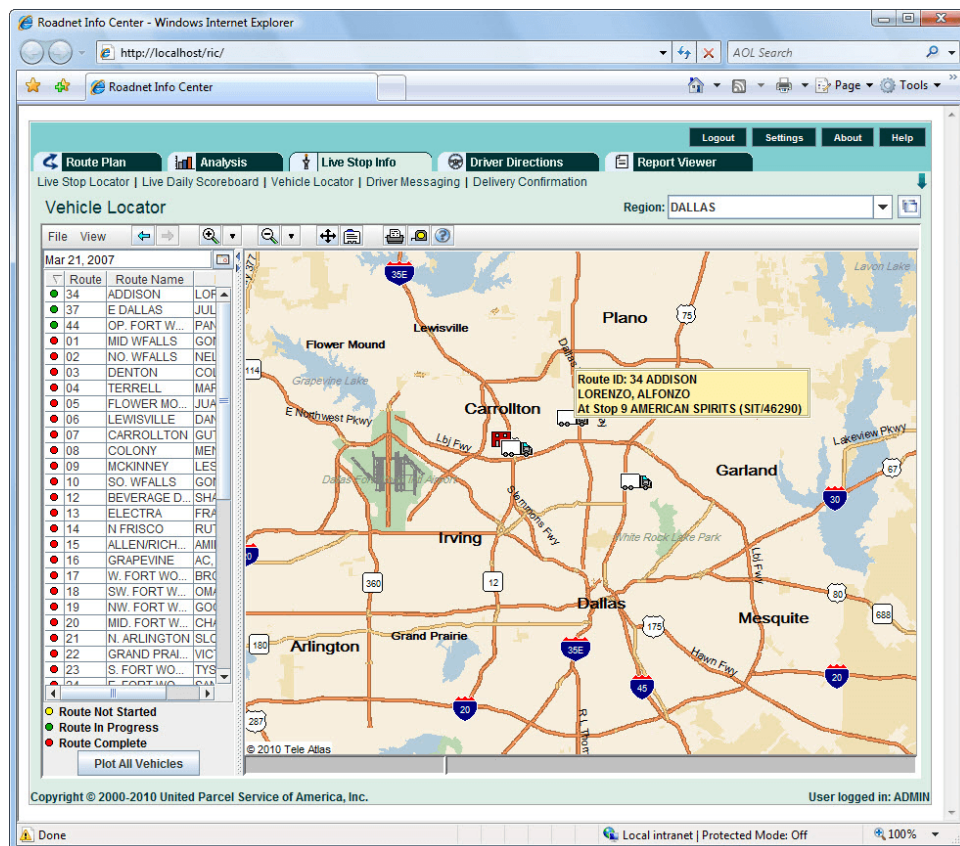


Рисунок 1.4 – програмний засіб компанії Omnitracs

MILESTONE

Milestone Systems — провідна компанія з Данії, яка займає лідируючі позиції на світовому ринку у сфері відеоспостереження та управління відеоданими. Заснована у 1998 році, вона розташувала свою штаб-квартиру в Копенгагені. Milestone Systems відома своїм передовим програмним забезпеченням, яке забезпечує ефективне управління потоками відео та системами безпеки.

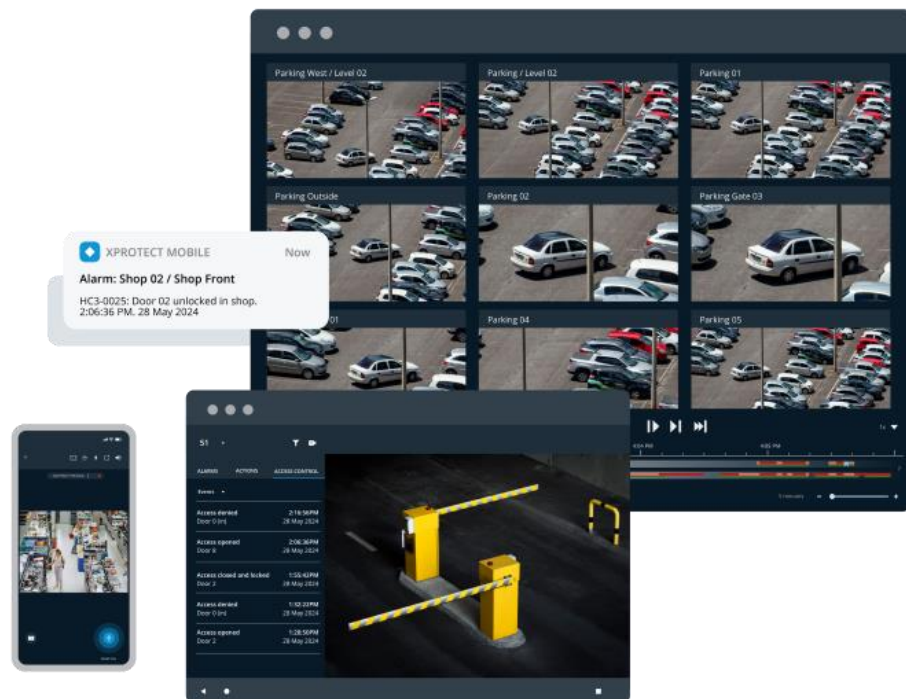


Рисунок 1.5 – програмний засіб компанії Milestone

Основні продукти та послуги Milestone Systems:

- **Програмне забезпечення для управління відео (VMS):** Це інструмент, який дозволяє налаштовувати, контролювати та оптимізувати роботу систем відеоспостереження. Воно підтримує функції запису, відтворення, аналізу відеоданих, а також легко інтегрується з іншими системами безпеки.
- **Аналітичні рішення для відео:** Компанія пропонує інструменти для розпізнавання номерів автомобілів, облич, аналізу поведінки об'єктів і

відстеження їхніх рухів.

- **Інтеграція з іншими технологіями:** Програмне забезпечення Milestone підтримує підключення до систем доступу, протипожежного захисту та інших платформ, забезпечуючи багаторівневу безпеку.

Система Argus [13] — це передова вимірювальна система, яка дозволяє контролювати параметри вагонів, зокрема технічний стан колісних пар. Принцип роботи базується на використанні лазерів для сканування поверхні катання колеса. Завдяки цьому система забезпечує:

- **Безперервний моніторинг стану вагонів:** Контроль технічних параметрів у режимі реального часу без необхідності зупинки поїздів.
- **Точність і надійність:** Argus забезпечує високий рівень вимірювань без впливу на рухомий склад.
- **Три рівні комплектації:** Включає детектори сходу, засоби автоматичного визначення дефектів коліс і комплексні вимірювальні установки для аналізу стійкості вагонів.

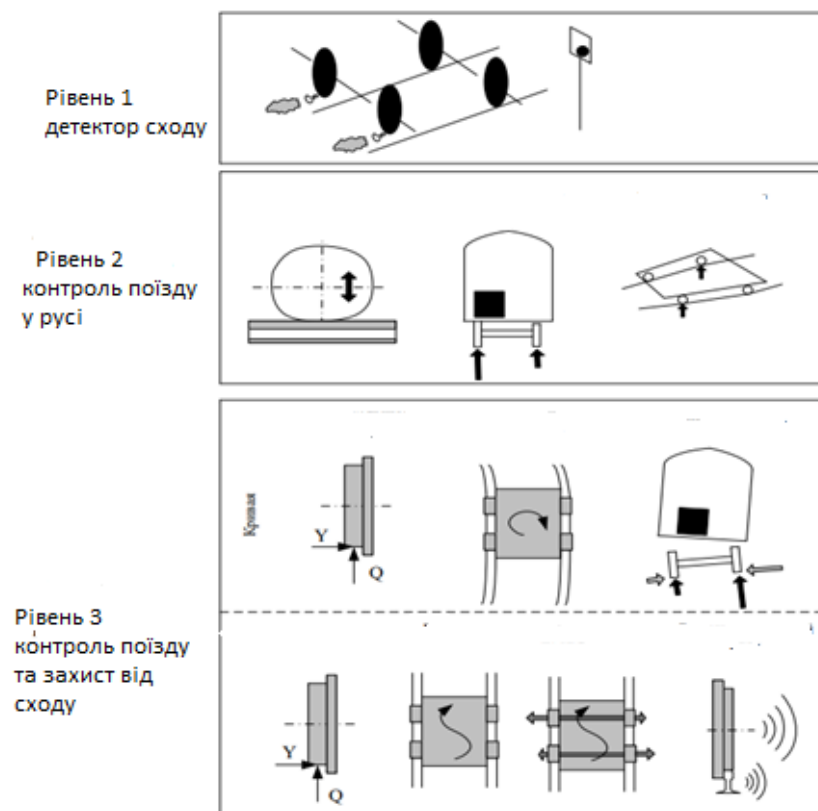


Рисунок 1.6 – Функції та можливості комплектації системи Argus

Система контролю габариту РАВ:

Призначена для виявлення виступаючих елементів вантажу, які можуть порушувати безпеку перевезень. Система також використовується для ідентифікації дефектів рухомого складу, що можуть бути небезпечними, і ухвалення рішень щодо їхнього виключення з експлуатації в потрібних місцях.

Milestone Systems і системи, такі як Argus та РАВ, націлені на підвищення ефективності, точності та безпеки в сфері відеоспостереження, логістики та контролю технічного стану транспортних засобів.

Висновки до розділу

Залізничний транспорт України відіграє важливу роль у національній економіці, забезпечуючи надійне перевезення вантажів і пасажирів, що підкреслює його стратегічне значення для країни.

Дослідження діяльності компанії "ПАТ АМКР" демонструє її значний історичний внесок у розвиток економіки та регіональної інфраструктури. Аналіз організаційної структури підприємства свідчить про його ефективність у досягненні стратегічних цілей.

Впровадження програмного забезпечення (ПЗ) у транспортній галузі визначається як важливий крок для оптимізації операцій і підвищення ефективності. Розгляд переваг і викликів, пов'язаних із цим процесом, дає змогу окреслити перспективи успішної реалізації таких рішень.

Оцінка сучасного стану транспортної інфраструктури та існуючих технологічних розробок виявляє основні напрями для вдосконалення і подальшого розвитку галузі. Узагальнюючи, цей аналіз надає цілісну картину ролі залізничного транспорту в Україні, впливу компанії "ПАТ АМКР" та значущості впровадження інноваційних рішень для формування ефективної та конкурентоспроможної транспортної системи.

2 АНАЛІЗ ПРОБЛЕМ ТА СТРАТЕГІЇ ВПРОВАДЖЕННЯ ПЗ В УПРАВЛІННІ РУХОМИМ СКЛАДОМ

2.1 Основні проблеми управління рухомим складом

2.1.1 Роль системи зв'язку в управлінні рухомим складом

Система зв'язку, яка використовується для управління рухомим складом, є ключовим елементом забезпечення стабільної та ефективної передачі інформації. Розглянемо основні технічні характеристики різних типів зв'язку, застосовуваних у цьому контексті.

Першим проаналізуємо радіозв'язок. Його ключовими параметрами є частотний діапазон, потужність передавача та стандарти передачі даних.

Частотний діапазон визначає, в яких межах частот передаються сигнали. Цей параметр впливає на функціональні можливості систем зв'язку. Високочастотні діапазони забезпечують високу швидкість передачі даних, що особливо актуально для мобільного зв'язку та широкосмугового Інтернету. Низькочастотні діапазони, у свою чергу, здатні проникати через перешкоди, такі як стіни, і мають ширший радіус дії, що важливо для забезпечення зв'язку на великих відстанях.

Потужність передавача визначає, наскільки потужним буде сигнал, що передається. Від цього залежить, як далеко сигнал може дістатися і наскільки ефективно він подолає перешкоди. Вища потужність дозволяє забезпечити стабільний зв'язок на великих відстанях, проте збільшує енергоспоживання та ризик електромагнітних завад. Менша потужність знижує витрати енергії, однак обмежує дальність передачі.

Стандарти передачі даних регламентують правила та протоколи для обміну інформацією між пристроями. Популярними стандартами є GSM (2G) [14] і LTE (4G), кожен із яких має свої переваги. GSM забезпечує базовий зв'язок і передачу текстових повідомлень, тоді як LTE (4G) дозволяє

здійснювати високошвидкісний доступ до Інтернету та передачу великих обсягів даних.

Наступним розглянемо супутниковий зв'язок, основними параметрами якого є висота супутника, швидкість передачі даних та зона покриття.

Висота супутника над Землею визначає зону покриття та якість зв'язку. Супутники, розташовані на великій висоті, забезпечують широке покриття території, охоплюючи навіть віддалені регіони. Це дозволяє сигналу проходити через перешкоди, такі як гори чи густі ліси. Однак значна висота може спричинити затримки у передачі сигналу, що впливає на швидкість реагування, особливо в реальному часі.

Швидкість передачі даних є показником ефективності обміну інформацією через супутниковий зв'язок. Цей параметр має вирішальне значення для забезпечення продуктивності системи. Висока швидкість передачі скорочує час відгуку, що є критичним для таких сфер, як медицина, фінансові операції чи системи управління. Вона також впливає на здатність передавати значний обсяг інформації, що важливо для потокової передачі відео, обміну великими файлами чи одночасного підключення багатьох пристроїв.

Покриття визначає територію, охоплену сигналом супутника. Для забезпечення стабільного зв'язку необхідно оптимізувати розташування супутників, щоб створити широку зону охоплення та уникнути зон, де сигнал відсутній через перешкоди чи рефракцію. Належне покриття забезпечує безперервний зв'язок навіть за несприятливих умов, таких як погодні аномалії чи електромагнітні впливи. Воно також адаптується до специфіки застосування системи. Наприклад, для мобільних глобальних мереж покриття має дозволяти підтримувати зв'язок без втрат при переміщенні між регіонами.

Далі розглянемо оптичні волокна, які характеризуються такими параметрами, як швидкість передачі даних, пропускна здатність і стійкість до перешкод.

Оптичні волокна є сучасною технологією, що використовує світлові

імпульси для передачі інформації. Вони забезпечують надзвичайно високу швидкість передачі завдяки використанню світлових сигналів, що поширюються за принципом внутрішнього відбиття та рефракції. Ця технологія дозволяє передавати дані на швидкостях у гігабітах і навіть терабітах за секунду, що робить її ідеальною для високопродуктивних систем.

Пропускна здатність визначає обсяг даних, який може передаватися за один цикл. Завдяки великій пропускній здатності, оптичні волокна здатні підтримувати одночасну передачу великих обсягів даних, що є особливо важливим для потокового відео або аудіо високої якості. Технологія мультиплексування хвиль (WDM) [15] дозволяє передавати кілька потоків даних через одне волокно, використовуючи різні довжини хвиль світла, що значно збільшує загальну пропускну здатність.

Стійкість до перешкод є ще однією перевагою оптичних волокон. Вони не чутливі до впливу електромагнітних полів, що виключає ризик електромагнітних завад. Відсутність радіочастотних сигналів також робить їх надійними для використання в умовах, де інші типи передачі можуть зазнавати інтерференції. Завдяки цим характеристикам оптичні волокна залишаються ефективним рішенням у різних умовах експлуатації.

2.1.2 Пропускна спроможність зв'язкових систем

Пропускна здатність системи зв'язку визначає, скільки даних може бути передано між двома точками за визначений час. У сфері управління рухомим складом, де необхідний швидкий та об'ємний обмін інформацією, цей параметр має вирішальне значення. Для ефективної роботи системи важливо забезпечити високу швидкість передачі даних. Технології, такі як 4G та 5G, дозволяють досягти потрібної пропускної здатності, що критично для передачі даних у режимі реального часу. Окрім того, важливу роль відіграють системи обробки інформації, які здатні швидко й ефективно працювати з великими масивами даних, зокрема щодо параметрів руху транспорту.

Управління трафіком передбачає налаштування алгоритмів і механізмів для оптимального розподілу потоків даних, попередження перевантаження

мережі та забезпечення стабільної роботи навіть у пікові періоди. Динамічне планування ресурсів та аналіз навантаження на мережу дозволяють оперативно реагувати на зміни у потребах, мінімізуючи ризик утворення "заторів" у передачі даних.

Пріоритетність даних також є важливим аспектом. Наприклад, інформація, що стосується безпеки пасажирів або вантажів, має передаватися першочергово. Системи зв'язку повинні вміти визначати критично важливі дані та забезпечувати їх негайну передачу. Гнучке управління пріоритетами дає можливість адаптуватися до різних ситуацій. Інтеграція штучного інтелекту та аналітики дозволяє оптимізувати розподіл ресурсів і підвищити ефективність мережі. Впровадження прогнозувальних механізмів допомагає запобігати перевантаженням і підтримувати стабільність роботи системи навіть за високого навантаження.

2.1.3 Захист систем зв'язку від зовнішніх впливів

Стійкість до перешкод є комплексним завданням, яке потребує впровадження сучасних технологій, розробки ефективних стратегій і системного підходу для адаптації до різних умов управління рухомим складом. Цей аспект включає детальне вивчення та подолання різноманітних викликів. Розглянемо основні фактори:

Вплив погодних умов. Необхідно враховувати такі параметри, як інтенсивність дощу, кількість снігу та рівень хмарності, які можуть суттєво впливати на якість сигналу. Розробка стратегії зменшення впливу цих факторів є важливою складовою.

Електромагнітні перешкоди. Аналіз електромагнітного спектра допомагає ідентифікувати потенційні джерела завад і обрати оптимальний частотний діапазон. Використання захисного обладнання, такого як фільтри й екрани, мінімізує вплив перешкод на зв'язок.

Фізичні бар'єри. Аналіз розсіяння сигналів та оцінка топографічних особливостей місцевості дозволяють визначити зони слабого сигналу й оптимізувати маршрути передачі. Використання математичних моделей

сприяє ефективному вирішенню проблем, пов'язаних із фізичними перешкодами.

Інноваційні технології та автономні системи. Впровадження бістатичних і мультістатичних систем дозволяє покращити надійність і стійкість зв'язку. Автономні системи здатні автоматично виявляти перешкоди та адаптувати параметри зв'язку для забезпечення стабільної роботи.

Системи резервування. Використання резервних каналів зв'язку й автоматичних систем перемикання забезпечує швидке відновлення роботи системи у разі збоїв або втрати сигналу.

2.2 Технічні аспекти впровадження ПЗ рухомого складу

2.2.1 Інтеграція ПЗ із сучасними залізничними технологіями

У сучасний період стрімкого технологічного розвитку залізничний транспорт також активно модернізується, впроваджуючи інновації. Застосування передових технологій стає вирішальним для покращення безпеки, ефективності та стабільності управління рухомим складом.

Сучасні системи сигналізації вдосконалюються за рахунок використання новітніх технологій та централізованих рішень, що сприяє підвищенню ефективності та координації руху поїздів. Електрифікація залізниць, включаючи використання контактних мереж і концепцій електромобільності, відкриває перспективи для підвищення екологічної стійкості та зменшення впливу на довкілля.

Автоматизовані блокувальні системи, оснащені електронними і програмними рішеннями, виступають важливим елементом для попередження інцидентів і забезпечення високого рівня безпеки руху. Інформаційно-керуючі технології, включаючи системи GPS, забезпечують точне відстеження рухомого складу і доступ до актуальної інформації для пасажирів і персоналу.

Високошвидкісні та маглев-технології, що використовують магнітну левітацію, дозволяють досягти рекордних швидкостей і створюють відповідну

інфраструктуру, яка підтримує ці інновації.

Ці технічні нововведення є складовою масштабного процесу модернізації залізничного транспорту відповідно до сучасних вимог. Інтеграція програмного забезпечення (ПЗ) стає ключовим фактором для оптимізації роботи та гармонізації процесів у цій галузі.

Визначення точок інтеграції ПЗ у залізничному транспорті передбачає ретельний аналіз існуючих систем, таких як автоматичні блокувальні системи та електрифікація, для виявлення можливостей покращення ефективності. Цей процес вимагає не лише технічної експертизи, а й детального розуміння всієї системи.

Особливу увагу приділяють адаптації нових технологій до існуючої інфраструктури, що мінімізує ризики та забезпечує ефективне використання ресурсів. План інтеграції включає конкретні етапи впровадження, строки виконання і підготовку персоналу до роботи з новими системами.

Оцінка впливу на бізнес-процеси враховує можливість скорочення витрат, підвищення продуктивності та покращення якості обслуговування. Завдяки цьому впровадження ПЗ відповідає не тільки технічним стандартам, а й стратегічним цілям галузі.

Розглянемо основні стандарти, які відіграють важливу роль в управлінні рухом залізничного транспорту.

GSM-R (Global System for Mobile Communications – Railway)

Цей стандарт мобільного зв'язку спеціально створений для залізничної галузі. Він функціонує на виділених частотах, забезпечуючи надійний і безпечний зв'язок між усіма компонентами системи. Особливістю GSM-R є здатність передавати як голосові повідомлення, так і дані, що робить його універсальним рішенням для комунікаційних потреб. Система стійка до зовнішніх впливів, таких як висока швидкість руху чи несприятливі погодні умови, гарантуючи стабільність роботи. Крім того, GSM-R має функції для підвищення безпеки: екстрене вимкнення сигналізації або відправлення аварійних повідомлень. Відповідність глобальним

стандартам дозволяє використовувати цей протокол у різних країнах, забезпечуючи інтеграцію зв'язку на міжнародному рівні. Загалом, GSM-R є ключовим інструментом для підвищення ефективності, забезпечуючи надійний зв'язок для внутрішньої та зовнішньої взаємодії залізничних систем.

ETCS (European Train Control System)

Ця система є важливим проєктом Європейського Союзу [16], спрямованим на стандартизацію контролю руху поїздів у країнах ЄС. Вона створена для уніфікації різноманітних національних систем контролю та забезпечення єдиного підходу до безпеки. ETCS базується на сучасних технологіях програмного забезпечення та зв'язку, дозволяючи автоматизувати управління рухом поїздів із високою точністю. Однією з головних переваг є її здатність працювати на будь-якій залізниці Європи, забезпечуючи сумісність між інфраструктурами різних країн. Система також легко інтегрується з іншими рішеннями для управління рухом, що сприяє гармонізації роботи залізничної інфраструктури. Впровадження ETCS є кроком до модернізації залізниць, підвищення ефективності перевезень та покращення безпеки.

IEEE 1473:2007

Цей стандарт, розроблений Інститутом інженерів з електротехніки та електроніки (IEEE), встановлює вимоги до систем зв'язку, моніторингу та управління у залізничному транспорті. На фізичному рівні стандарт регламентує способи з'єднання пристроїв, використовуючи як дротові, так і бездротові технології, що забезпечують надійність зв'язку. Програмний рівень охоплює протоколи обміну даними між системами контролю, забезпечуючи їх сумісність. Також стандарт включає вимоги до систем моніторингу, спрямовані на підтримання безпеки та оптимізацію роботи залізничного транспорту. Він враховує специфіку таких елементів, як вагони, локомотиви та електропоїзди, регулюючи їх інтеграцію в єдину систему.

Завдяки IEEE 1473:2007 створюється уніфікована інфраструктура, яка забезпечує ефективну взаємодію між різними компонентами залізничного транспорту.

Серед протоколів можна віділити: TCP/IP, MQTT, SNMP. Розглянемо у деталях їх.

TCP/IP (Transmission Control Protocol/Internet Protocol) – це набір протоколів, який буде визначати, яким чином наші дані будуть надані і опрацьовані у комп'ютерних мережах, наприклад у Internet. розглянемо передові аспекти заданих протоколів у деталях:

Internet Protocol (IP). IP означає основи комунікацій у мережах. Його основні аспекти включають: Адресацію. Кожен пристрій у мережі має унікальну IP-адресу, яка використовується для ідентифікації та маршрутизації. Маршрутизацію. IP визначає, як дані мають бути направлені через мережу від відправника до отримувача. Фрагментацію та збірку: У випадку великих об'ємів даних IP може розділити їх на менші фрагменти для передачі та потім відновити їх на приймачі.

TCP (Transmission Control Protocol): Цей протокол працює поверх IP і забезпечує надійність передачі даних та управління з'єднанням. TCP гарантує правильний порядок доставки пакетів і уникнення втрат даних. Він також регулює швидкість обміну інформацією між пристроями для запобігання перевантаження мережі, встановлюючи та завершуючи з'єднання перед передачею даних.

TCP/IP є основним набором протоколів, що забезпечує зв'язок в Інтернеті та багатьох локальних мережах. Завдяки своїй універсальності та ефективності, він дозволяє надійно передавати дані по всьому світу.

MQTT (Message Queuing Telemetry Transport): Легкий, відкритий і продуктивний протокол, оптимізований для передачі даних в системах Інтернету речей (IoT). Завдяки своїй архітектурі він має низькі вимоги до ресурсів, що дозволяє ефективно здійснювати обмін даними між пристроями.

MQTT базується на моделі клієнт-сервер, де пристрої можуть

виконувати ролі клієнтів, серверів або обох одночасно, що спрощує взаємодію між ними. Його ключовий механізм — модель "публікація-підписки", яка дозволяє клієнтам підписуватися на певні теми та отримувати повідомлення, пов'язані з ними. Тема виступає ієрархічною адресою, яка структурує обмін інформацією. Протокол підтримує три рівні QoS (якість обслуговування), що забезпечує різні рівні надійності доставки даних.

MQTT відзначається своєю ефективністю навіть на пристроях з обмеженими можливостями, що робить його популярним у сферах домашньої автоматизації, телеметрії, медицини, сільського господарства тощо. Легкість розгортання, масштабованість і мінімальне споживання ресурсів виділяють його як ідеальне рішення для великих IoT-мереж.

SNMP (Simple Network Management Protocol): Протокол, призначений для управління та моніторингу пристроїв у комп'ютерних мережах. Його архітектура складається з двох основних компонентів: менеджера, який відповідає за контроль, та агента, який реалізує функції на стороні пристрою.

SNMP використовує клієнт-серверну модель, де менеджер надсилає запити до агента для отримання інформації чи внесення змін у конфігурацію. Серед основних операцій — GET (отримання даних), SET (зміна параметрів) та TRAP (автоматичне сповіщення про події).

Протокол застосовується для роботи з такими пристроями, як маршрутизатори, комутатори, сервери тощо. Версії SNMP (v1, v2c, v3) включають різні функції безпеки, зокрема аутентифікацію та шифрування (у SNMPv3).

Завдяки цьому протоколу забезпечується ефективний моніторинг мереж, швидке реагування на події, контроль за станом пристроїв і безпечна передача даних. Використання таких стандартів підвищує надійність і сумісність систем управління, що є важливим у залізничній галузі для забезпечення стабільності та безпеки управління рухомим складом.

Розробка інтерфейсів для систем залізничного транспорту є складним і стратегічним завданням, спрямованим на створення ефективних, надійних і

простих у використанні інструментів для взаємодії між різними елементами системи. Цей процес має на меті забезпечити стабільну та безпечну комунікацію між компонентами.

На початковому етапі виконують глибокий аналіз вимог користувачів і технічних параметрів системи. Визначаються основні характеристики інтерфейсів, зокрема види передаваних даних, технічні можливості систем та необхідність дотримання чинних стандартів і протоколів.

Особлива увага приділяється забезпеченню сумісності інтерфейсів із вже існуючою інфраструктурою та технологічними процесами. Для цього враховуються актуальні стандарти, які діють у залізничній сфері.

Тестування інтерфейсів передбачає перевірку їх надійності, безпеки та ефективності. Важливим аспектом є також створення інтуїтивно зрозумілого дизайну, який забезпечує зручність використання для операторів і персоналу.

Процес розробки вимагає ґрунтовного розуміння системи управління рухомим складом, а також врахування технічних, ергономічних та безпекових аспектів, що дозволяє досягти максимальної ефективності їх впровадження.

Системи моніторингу та діагностики в залізничному транспорті відіграють важливу роль у забезпеченні безпеки, ефективності та контролю над рухомим складом і інфраструктурою. Ці системи ґрунтуються на сучасних технологіях і впроваджують інноваційні технічні рішення для оптимальної роботи.

Вони використовують різноманітні сенсори, зокрема відеокамери, акустичні та температурні датчики, для збору інформації в режимі реального часу. Ці дані аналізуються з метою моніторингу стану рухомого складу та інфраструктури, а також виявлення можливих аномалій, аварійних ситуацій чи інших несправностей.

Системи діагностики орієнтовані на автоматизоване виявлення технічних несправностей і слабких місць у русловому складі та інфраструктурі. Вони застосовують різноманітні сенсори, такі як акселерометри, гіроскопи, температурні датчики та інші, для постійного

контролю стану компонентів і виявлення відхилень у їх роботі.

Використання технологій штучного інтелекту забезпечує обробку великих обсягів даних і їх аналіз у реальному часі, що дозволяє прогнозувати можливі збої або пошкодження. Це сприяє створенню передбачуваних графіків технічного обслуговування, що оптимізує використання ресурсів, підвищуючи рівень безпеки та ефективності.

Зібрані дані не лише полегшують оперативне управління, але й використовуються для стратегічного планування, аналізу тенденцій і вдосконалення роботи транспортної системи загалом. Такі системи є ключовими елементами сучасної залізничної інфраструктури, спрямованими на забезпечення високої якості та надійності перевезень.

Системи безпеки у залізничному транспорті враховують широкий спектр факторів для забезпечення захисту пасажирів і безпечного управління вантажами. Одним із головних компонентів є технології виявлення аномалій та автоматизованого гальмування, які допомагають уникнути аварій і зменшити їх наслідки. Реалізація відеоспостереження та моніторингу в режимі реального часу дозволяє контролювати рухомий склад і швидко реагувати на несправності.

Системи доступу та ідентифікації персоналу забезпечують додатковий рівень безпеки для працівників і пасажирів. Надійність транспортної мережі підтримується завдяки регулярному технічному обслуговуванню, діагностиці та використанню запасних компонентів, що дозволяють оперативно усувати несправності.

Окрім цього, критично важливим є навчання і сертифікація персоналу для ефективного реагування на надзвичайні ситуації. Постійний контроль стану інфраструктури, включно з коліями, мостами й тунелями, сприяє вчасному виявленню можливих проблем і запобіганню збоям у роботі.

Досягнення високого рівня безпеки та стабільності в залізничному секторі потребує комплексного підходу, що об'єднує технічні, організаційні та експлуатаційні аспекти, щоб запобігти аваріям і гарантувати безперервне

функціонування системи.

2.2.2 Технічні стандарти та вимоги для ПЗ

Технічні вимоги та стандарти для програмного забезпечення управління рухомим складом охоплюють широкий спектр характеристик і умов, спрямованих на забезпечення його стабільної та ефективної роботи. Одним із основних компонентів є вимоги до електроживлення, які визначають не лише стандарти напруги й частоти, але й специфікації інтерфейсів для сумісності з іншими системами.

Комунікаційні вимоги включають стандартизацію протоколів передачі даних між різними елементами системи. Це забезпечує надійну взаємодію та оперативний обмін інформацією між компонентами управління.

Безпека також є ключовим фактором: встановлюються заходи для захисту від можливих загроз, таких як несанкціонований доступ або шкідливе програмне забезпечення. Крім того, визначаються показники надійності та доступності, що гарантують безперебійну роботу системи протягом тривалого часу без серйозних збоїв.

З огляду на сучасні технології, стандарти бездротового зв'язку враховують не лише ефективність, а й безпечність передачі даних. Паралельно створюються протоколи, які забезпечують інтеграцію системи управління з іншими елементами інфраструктури, роблячи її більш адаптивною.

Енергоефективність також посідає важливе місце: визначаються параметри раціонального використання енергії та зменшення негативного впливу на навколишнє середовище.

Узагальнюючи, ці стандарти й вимоги формують основу для розробки та впровадження інноваційних систем управління рухомим складом. Вони сприяють дотриманню високих показників ефективності, надійності та безпеки в транспортній галузі.

2.3 Програмні особливості забезпечення

2.3.1 Огляд ПЗ для контролю рухомого складу

Огляд програмного забезпечення для управління рухомим складом охоплює аналіз функціональних аспектів, які спрямовані на покращення моніторингу й оптимізацію роботи транспортної системи. Це програмне забезпечення об'єднує низку компонентів, які працюють разом для підвищення ефективності та надійності рухомого складу.

Серед ключових функцій таких систем є модулі відстеження та контролю руху поїздів. Вони застосовують алгоритми для прогнозування та відображення місцезнаходження поїздів, що сприяє вирішенню завдань диспетчеризації та взаємодії з іншими елементами транспортної мережі.

Інтелектуальні алгоритми прогнозування руху дозволяють оптимізувати маршрути, знижувати затримки й уникати перевантаження на ключових ділянках. Це забезпечує скорочення часу подорожей і підвищення ефективності роботи системи загалом.

У сфері безпеки програмне забезпечення включає функції для моніторингу можливих загроз і швидкого реагування на них. Воно забезпечує контроль за дотриманням стандартів безпеки, своєчасне виявлення відхилень і запуск автоматизованих механізмів реагування.

Для диспетчерських центрів розробляються спеціалізовані інтерфейси, що забезпечують операторам доступ до необхідних даних у реальному часі. Це дозволяє швидко аналізувати ситуацію й приймати обґрунтовані рішення.

Важливим є відповідність програмного забезпечення технічним стандартам, які передбачають надійність роботи, захист даних і сумісність із обладнанням рухомого складу.

Таким чином, програмне забезпечення для управління рухомим складом є важливим інструментом, що сприяє вдосконаленню транспортної системи, забезпечуючи високий рівень безпеки, ефективності та надійності в її функціонуванні..

2.3.2 Інструменти для збору та аналізу інформації у ПЗ залізничного транспорту

Системи збору даних у залізничному транспорті відіграють важливу роль у забезпеченні ефективності й безпеки цієї галузі. Завдяки широкому спектру сенсорів та датчиків, розташованих на різних елементах інфраструктури, вдається вимірювати критично важливі параметри, які використовуються для управління та контролю.

Інтегровані системи, вбудовані в рухомий склад, здійснюють автоматичний збір і фіксацію даних під час експлуатації. Використовуючи сучасні технології, такі як GPS-трекінг, системи контролю руху та діагностики обладнання, вони забезпечують точний моніторинг параметрів у реальному часі. Це дозволяє операторам отримувати актуальну інформацію про стан техніки та інфраструктури.

Зібрані дані передаються до центральних обчислювальних систем, де вони аналізуються за допомогою спеціалізованого програмного забезпечення й алгоритмів. Цей аналіз включає виявлення відхилень, прогнозування можливих несправностей і оптимізацію роботи рухомого складу для підвищення ефективності та зниження ризиків.

Особливе значення мають системи збору й обробки даних у реальному часі. Вони забезпечують оперативну реакцію диспетчерів на змінні умови та нестандартні ситуації, такі як аварії чи технічні збої. Це сприяє мінімізації негативних наслідків і підвищенню безпеки перевезень.

Використання технологій Інтернету речей (IoT) і міжмашинної взаємодії (M2M) додає ефективності передачі інформації між елементами системи. Завдяки цьому вдосконалюється координація між різними складовими залізничної мережі, що позитивно впливає на управління та моніторинг.

У підсумку, системи збору й аналізу даних у залізничному транспорті є критично важливими для забезпечення надійності, ефективності та безпеки галузі. Вони не лише дозволяють відстежувати ситуацію в реальному часі, а й

виступають стратегічним інструментом для планування, прогнозування та оптимізації роботи залізничної інфраструктури.

2.4 Перспективи розвитку ПЗ для рухомих складів

2.4.1 Сучасні тенденції ПЗ транспортних технологій

Розвиток технологій програмного забезпечення для залізничного транспорту визначає перспективи галузі, впливаючи на її продуктивність, безпеку та екологічність. Проаналізуємо основні напрями цих змін.

Використання машинного навчання та аналітики даних у залізничній сфері сприяє оптимізації управління рухомих складом. Такі технології застосовують для передбачення потреб у ремонті, оптимізації розкладів руху, виявлення аномалій та ефективного управління процесами.

Прогнозування ремонтів здійснюється шляхом аналізу великих масивів даних про технічний стан обладнання, що дозволяє виявляти потенційні несправності на основі історичних записів. Оптимізація розкладів руху базується на аналізі пасажиропотоків, динаміки руху та використанні алгоритмів для створення максимально ефективних графіків. Системи виявлення аномалій забезпечують моніторинг роботи рухомого складу та автоматичну реакцію на виявлені відхилення.

Технології такого рівня створюють адаптивні інтелектуальні системи, які сприяють покращенню ефективності, безпеки та надійності залізничного транспорту.

Програмне забезпечення для безпеки транспорту призначене для виявлення та нейтралізації потенційних загроз, що сприяє забезпеченню високого рівня безпеки перевезень. Сучасні системи використовують сенсори, камери спостереження, алгоритми машинного навчання та технології розпізнавання образів.

Виявлення відхилень у русі відбувається за допомогою моніторингових систем, які реагують на аномалії в реальному часі. Відеоконтроль та

розпізнавання об'єктів на коліях дозволяють уникати зіткнень, попереджаючи загрози. У разі аварійних ситуацій автоматичні системи безпеки виконують екстрене гальмування та передають сигнали до диспетчерських центрів.

Окрім цього, системи безпеки враховують погодні умови та застосовують аналітику даних для розробки оптимальних стратегій реагування. Інтеграція таких рішень створює інтелектуальні системи, що забезпечують надійний захист рухомого складу та мінімізують ризики в залізничному транспорті.

Використання технологій 5G у залізничному транспорті знаменує новий етап у керуванні рухомим складом. Ця передова технологія надає швидкий і стабільний зв'язок, що критично важливо для ефективного управління та забезпечення безпеки в галузі.

5G забезпечує передачу великих обсягів даних у реальному часі, що дозволяє значно покращити контроль і оптимізацію процесів. Завдяки мінімальній затримці та високій надійності, ця технологія забезпечує оперативну взаємодію між елементами системи, що є ключовим для програмного забезпечення безпеки й управління рухом.

Можливість обслуговування численних пристроїв одночасно дозволяє ефективно використовувати 5G на обмежених територіях, що важливо для моніторингових систем, сенсорів та комунікації між рухомим складом і станціями.

Розвиток Інтернету речей (IoT) та технологій M2M завдяки 5G сприяє покращенню взаємодії елементів транспортної системи, що дозволяє ефективніше управляти трафіком, прогнозувати ситуації та планувати маршрути. У цілому, застосування 5G відкриває нові можливості для створення інтелектуальних транспортних систем, які забезпечують високу швидкість, надійність та безпеку.

Електричний і гібридний рухомий склад є важливим кроком до сталого розвитку залізничного транспорту. Електропоїзди, що працюють на енергії з відновлюваних джерел, зменшують рівень викидів та підвищують

екологічність перевезень.

Ці технології сприяють енергоефективності завдяки впровадженню інновацій, таких як великоємні батареї та системи регенерації енергії. Гібридні поїзди комбінують електричний і дизельний приводи, забезпечуючи гнучкість на електрифікованих та неелектрифікованих ділянках.

Використання електричного та гібридного транспорту сприяє економії енергоресурсів, знижує залежність від традиційного пального та мінімізує вплив на довкілля. Це важливий елемент стратегії сталого розвитку, спрямованої на екологічно чистий і ефективний транспорт.

Інтелектуальна інфраструктура в залізничному транспорті впроваджує передові рішення для оптимізації руху поїздів і управління станціями. Вона застосовує сучасні алгоритми для синхронізації руху, уникнення заторів і коригування інтервалів між потягами.

Системи керування сигналізацією дозволяють зменшити затримки, тоді як програмне забезпечення на станціях, включаючи автоматизацію багажу та електронні платежі, забезпечує зручність для пасажирів.

Ключова особливість інтелектуальної інфраструктури — здатність діагностувати можливі несправності та оперативно реагувати на них. Вона також забезпечує безперебійну взаємодію з рухомим складом, підвищуючи рівень безпеки та оперативності.

Таким чином, розумна інфраструктура інтегрує інноваційні технології, створюючи сучасну транспортну систему, що є ефективною, екологічною та орієнтованою на безпеку.

2.4.2 Новітні технології для вдосконалення ПЗ

Подальший розвиток програмного забезпечення у сфері залізничного транспорту орієнтований на впровадження передових технологій, таких як штучний інтелект і машинне навчання. Ці інструменти допомагають прогнозувати рух поїздів і вдосконалювати розклади. Інтернет речей (IoT) та технології зв'язку між пристроями відіграють важливу роль у моніторингу технічного стану рухомого складу та інфраструктури, забезпечуючи швидку

реакцію на будь-які зміни чи несправності. Технології розширеної та віртуальної реальності використовуються для навчання персоналу та підвищення якості діагностики несправностей. Водночас кіберфізичні системи, які інтегрують датчики та виконавчі пристрої, сприяють створенню інтелектуальних рішень для управління й оптимізації логістичних процесів. Ці інновації відкривають нові горизонти для підвищення ефективності, безпеки та стійкості залізничного транспорту, формуючи перспективні підходи до модернізації галузі.

Висновки до розділу

Аналізуючи сучасні тенденції у програмному забезпеченні для залізничного транспорту, можна виділити ключові напрямки розвитку. Інноваційні технології, такі як штучний інтелект, Інтернет речей, розширена реальність та інші, формують майбутнє систем управління рухомим складом.

Дотримання технічних стандартів є основою для створення безпечного й ефективного програмного забезпечення. Воно забезпечує реалізацію алгоритмів для прогнозування, впровадження систем безпеки та інтеграцію з іншими інформаційними платформами.

Реальний моніторинг і аналіз стану рухомого складу стають можливими завдяки використанню технологій IoT і M2M, які підтримують системи збору й обробки даних у режимі реального часу. Технологія 5G полегшує швидку передачу великих обсягів інформації, що є критично важливим для управління процесами та гарантування безпеки.

Аналітика даних і машинне навчання дозволяють прогнозувати ремонтні роботи, удосконалювати розклади руху та виявляти аномальні ситуації. Програмне забезпечення для систем безпеки допомагає ідентифікувати й усувати потенційні ризики, забезпечуючи високий рівень захисту.

Перехід на електричні та гібридні технології живлення сприяє підвищенню енергоефективності й мінімізації екологічного впливу. Інтелектуальна інфраструктура застосовує алгоритми для оптимізації руху

потягів і управління станціями, що покращує продуктивність транспортної системи.

Подальший розвиток включає вдосконалення штучного інтелекту, розширення IoT-рішень, використання VR-технологій і впровадження інновацій для підвищення автоматизації та ефективності функціонування залізничного транспорту. Усі ці напрямки сприяють створенню високотехнологічної системи, яка відповідає вимогам безпеки, продуктивності й сталого розвитку.

3 РЕАЛІЗАЦІЯ ПЗ УПРАВЛІННЯ РУХОМИМ СКЛАДОМ

3.1 Обґрунтування вибору технологій для розробки програмного забезпечення

Спершу зазначу, що розробка програми виконувалась за допомогою платформи ASP.NET Core MVC. Цей фреймворк дозволяє створювати високоефективні та модульні веб-додатки і сервіси, реалізуючи концепцію Model-View-Controller (MVC). Її основний принцип полягає в поділі додатка на три ключові компоненти.

Модель у контексті ASP.NET Core MVC відповідає за управління даними додатка. Вона виконує важливі функції, зокрема: визначає структуру та формат даних, які використовуються в програмі, включаючи класи для опису об'єктів даних, їх атрибути та зв'язки. Крім того, модель забезпечує взаємодію з базами даних чи іншими джерелами для збереження, оновлення, видалення або отримання інформації.

Окремо слід зазначити, що модель включає бізнес-логіку, яка визначає правила обробки даних і взаємодії між ними в рамках додатка. Це можуть бути алгоритми, обчислення чи правила валідації, які регламентують функціонування системи.

У певних ситуаціях модель має можливість повідомляти інші компоненти (наприклад, контролери чи подання) про зміни в даних. Це забезпечує автоматичне оновлення інтерфейсу користувача або виконання інших необхідних дій у відповідь на зміни.

Також модель організовує обмін даними з іншими компонентами системи, такими як служби чи контролери, забезпечуючи єдиний інтерфейс для інтеграції.

Таким чином, модель є центральною частиною, яка керує даними та бізнес-логікою в додатку, створеному на ASP.NET Core MVC. Вона спрощує розробку і підтримку коду завдяки чіткому поділу відповідальностей.

View (Вид) у шаблоні MVC, реалізованому в ASP.NET Core, виконує важливу функцію забезпечення відображення інформації для користувача. Основні його аспекти включають:

1) **Візуалізація даних.** Головна мета виду полягає у поданні даних, отриманих із моделі, у зрозумілому для користувача форматі. Це може бути реалізовано через HTML-сторінки, шаблони чи часткові види, які визначають структуру та зовнішній вигляд відображуваної інформації.

2) **Елементи інтерфейсу.** Вид містить складові користувацького інтерфейсу, такі як форми, кнопки, таблиці й інші інтерактивні елементи, що спрощують роботу користувачів із додатком.

3) **Шаблони та макети.** Для забезпечення стилістичної цілісності застосунку види використовують шаблони та макети, які дозволяють створити єдиний дизайн і структуру сторінок.

4) **Динамічність.** У контексті ASP.NET Core MVC види можуть містити інтерактивні елементи, які оновлюються без повного перезавантаження сторінки. Це значно підвищує гнучкість та зручність користування.

5) **Обмін даними з контролером.** Види отримують необхідну інформацію від контролерів, після чого відповідають за її подання. Така взаємодія забезпечує передачу параметрів між компонентами системи.

6) **Робота з подіями.** Вид може включати обробники подій та скрипти, які дозволяють реагувати на дії користувача, наприклад, натискання кнопок чи введення тексту.

7) **Зручність для користувача.** Важливим завданням виду є забезпечення інтуїтивно зрозумілого інтерфейсу, що сприяє легкій навігації та ефективній взаємодії.

Загалом View у ASP.NET Core MVC відповідає за створення зручного та ефективного інтерфейсу, який перетворює дані додатка на зрозумілий і візуально привабливий формат, забезпечуючи комфорт користувача під час роботи з програмою.

Контролер (Controller) у структурі ASP.NET Core MVC є важливим

компонентом, який виконує функції обробки запитів користувача, управління логікою взаємодії з моделлю та передачі результатів до виду. Розглянемо основні аспекти його роботи:

Обробка запитів. Контролер приймає вхідні HTTP-запити, що визначають дії, які потрібно виконати. Наприклад, це може бути запит на завантаження сторінки або виконання певної операції.

Робота з моделлю. Після отримання запиту контролер звертається до моделі для отримання необхідних даних або виконання певних операцій. Він використовує методи моделі для взаємодії з базою даних чи реалізації бізнес-логіки.

Підготовка даних для відображення. Контролер обробляє дані, отримані від моделі, готує змінні або створює об'єкти, які будуть використані у виді для відображення інформації.

Виклик виду. Після завершення обробки запиту контролер обирає відповідний вигляд для відображення результатів і передає підготовлені дані у вид.

Обробка відповідей. Контролер визначає, як реагувати на подальші дії користувача. Це може включати перенаправлення на іншу сторінку, додаткові запити до моделі або повторне оновлення інтерфейсу.

Посередник між моделлю та видом. Основна функція контролера — координація роботи між моделлю та видом, забезпечуючи безперервний потік даних від запиту користувача до відображення результатів.

Контролери відіграють ключову роль у відокремленні логіки обробки даних від їхнього подання, що робить код більш структурованим і легким для підтримки.

ASP.NET Core — це універсальна платформа з відкритим вихідним кодом, яка підтримує розробку різних типів додатків, зокрема веб-сайтів, API та мікросервісів. Шаблон MVC дозволяє розділити програмну логіку на окремі компоненти, що спрощує процес розробки та обслуговування.



Рисунок 3.1 – Структура функціонування ASP.NET Core

3.2 Проектування архітектури програмного забезпечення керування рухомим складом

Перед початком роботи над програмним забезпеченням для керування рухомим складом необхідно правильно сформулювати модель цього процесу. Для цього спочатку варто розглянути структуру системи.



Рисунок 3.2 – Організація роботи зазначеної системи

Для зображення структурно-функціональної моделі було створено контекстну діаграму, яка демонструє основні процеси, що відбуваються під час роботи програми. У процесі побудови діаграми було визначено такі елементи: вхідні, вихідні, керуючі дані та механізми.

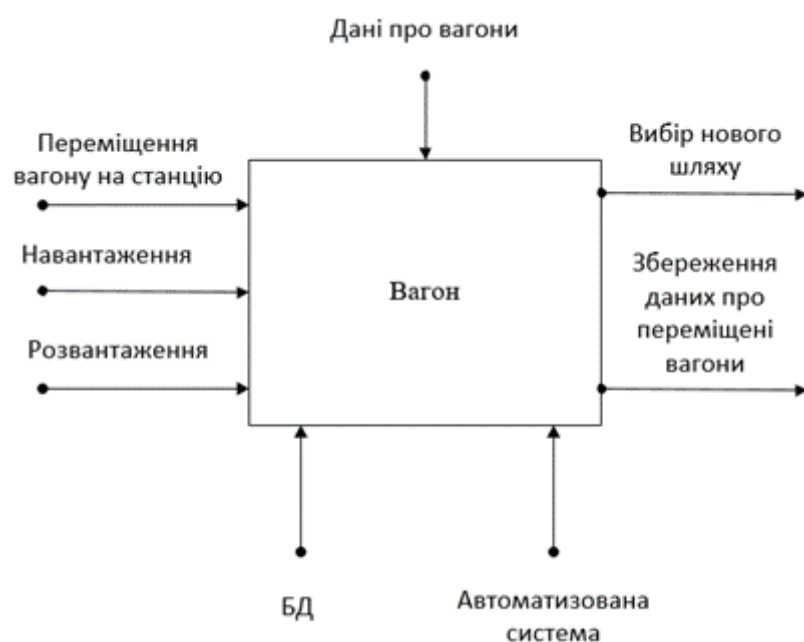


Рисунок 3.3 – Діаграма контексту для вагону

Вхідні дані для вагона включають його переміщення на станцію, процеси навантаження та розвантаження.

Вихідними даними є вибір нового маршруту для вагона та збереження інформації про попередні переміщення.

Керуючі дані представляють собою інформацію про рухомий склад.

Механізмами виступають база даних та сама система.

Далі розглядається авторизація користувача в програмі, яка зображена на рисунку 3.4.

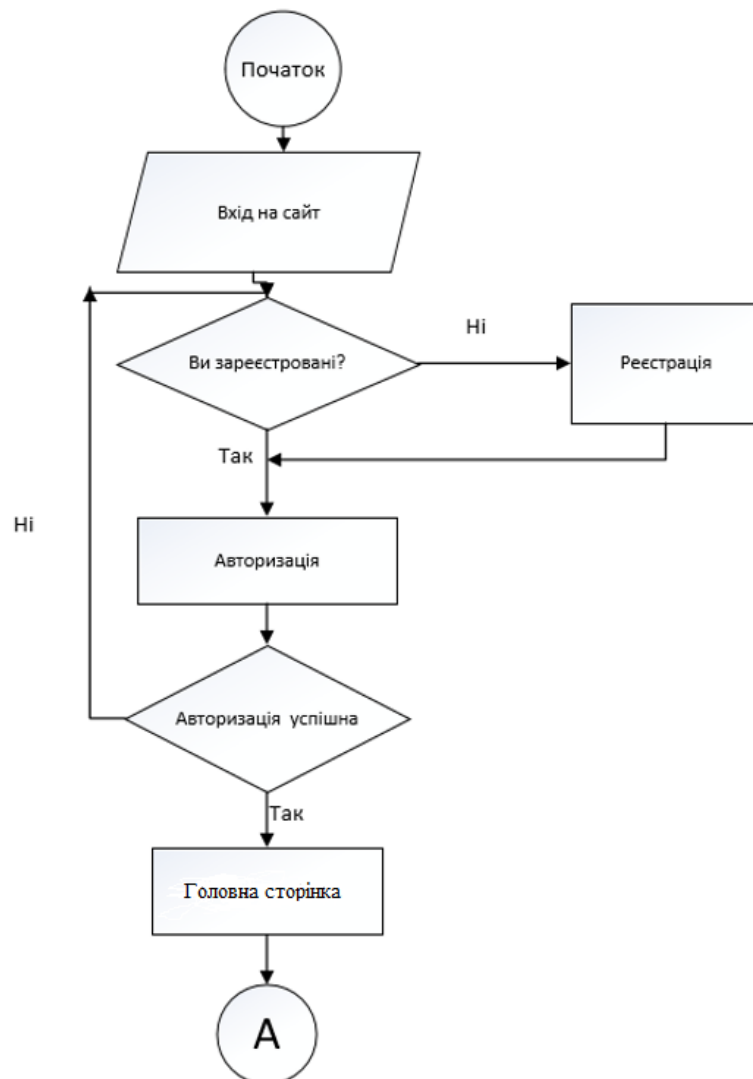


Рисунок 3.4 – Блок схема для процесу авторизації.

На рисунку 3.5 наведено діаграму, що ілюструє послідовність дій користувача в системі.

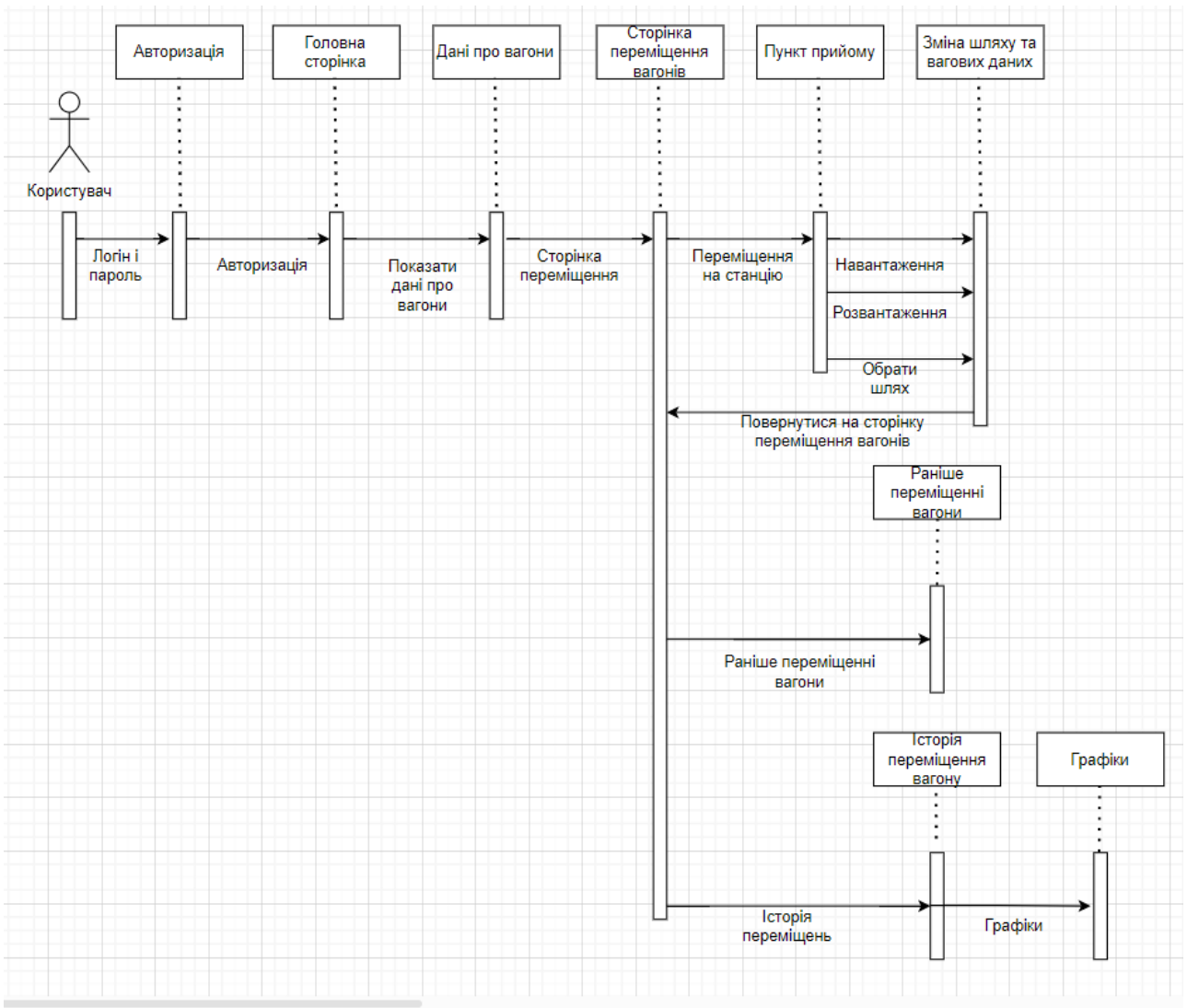


Рисунок 3.5 – Діаграма послідовності виконання нашої програми

Процес взаємодії користувача із системою розпочинається з реєстрації або авторизації. Після входу до системи користувач потрапляє на головну сторінку, де може перейти до розділу з інформацією про вагони. Наступним етапом є перехід до сторінки переміщення вагонів, де користувач має можливість перемістити вагон на певну станцію.

На приймальному пункті, куди прибуває вагон, користувач може обрати маршрут, виконати навантаження чи розвантаження.

Після завершення цих операцій користувач повертається на сторінку переміщень вагонів, де можна переглянути дані про переміщення інших вагонів або переглянути історію руху конкретного вагона у вигляді графіків.

3.3 Створення бази даних системи

Одним із ключових елементів у будь-якій інформаційній системі є база даних. У своїй роботі я використовував Microsoft SQL Server Management Studio. База даних має назву Universe і складається з трьох таблиць: Universe_1, Universe_2 та Users. Перш ніж перейти до деталей, варто коротко ознайомитися з особливостями MS SQL Server.

Microsoft SQL Server (MS SQL) — це система управління базами даних, створена компанією Microsoft. Вона призначена для організації, зберігання та обробки даних, що використовуються в різних додатках, включаючи програми та веб-сайти. Основні можливості MS SQL включають збереження даних, підтримку мови SQL (Structured Query Language), а також обробку й аналіз інформації.

Основні характеристики та функціонал MS SQL Server:

База даних. У MS SQL Server база даних є структурованим контейнером для збереження інформації. Вона дозволяє організувати дані у вигляді таблиць, зв'язків, функцій, процедур та інших об'єктів, що входять до складу бази.

Таблиці — це основна структура для зберігання даних у MS SQL Server. Кожна таблиця має свою назву та складається з колонок і рядків. Колонки представляють характеристики даних, а рядки містять конкретні значення, які їм відповідають. Усі колонки мають унікальні імена та певний тип даних (наприклад, INTEGER — для чисел, VARCHAR — для тексту, DATE — для дат).

Ключі. Первинний ключ (PRIMARY KEY): використовується для однозначної ідентифікації записів у таблиці. Зовнішній ключ (FOREIGN KEY): встановлює зв'язки між таблицями, вказуючи на ключ іншої таблиці.

Індекси. Індекси створюються для підвищення швидкості доступу до даних, дозволяючи знаходити потрібні записи без перегляду всієї таблиці.

Обмеження. Вони визначають правила для таблиць, наприклад:

Обмеження унікальності гарантує, що значення в певній колонці не повторюються. Обмеження за замовчуванням дозволяє автоматично встановлювати значення, якщо воно не було вказане під час додавання нового запису.

Тригери. Це спеціальні процедури, які виконуються автоматично при певних діях у базі даних, наприклад, під час додавання, оновлення або видалення записів.

Процедури — це набір команд або операцій, які можна зберігати в базі даних і викликати за необхідності. Вони дозволяють об'єднати логіку виконання певних дій в одному місці, замінюючи повторювані інструкції єдиним викликом. Процедури підтримують передачу параметрів, використання умов, циклів та інших елементів програмування, що дає змогу реалізовувати складну логіку.

Функції також є набором інструкцій, однак їх ключова особливість полягає в тому, що після виконання вони повертають значення. Функції можна інтегрувати в SQL-запити та використовувати їх результати в інших частинах коду.

Функції поділяються на:

Скалярні функції: повертають одне значення.

Табличні функції: результати представлені у вигляді таблиці.

Відмінності між процедурами та функціями:

Функції завжди повертають результат через оператор RETURN, тоді як процедури цього не роблять.

Процедури можуть виконувати зміни у базі даних і передавати результати через параметри, у той час як функції лише повертають значення.

Функції використовуються у SQL-виразах, а процедури викликаються окремо як самостійні об'єкти.

Обидві ці конструкції є важливими інструментами в управлінні базами даних, що сприяють спрощенню обслуговування, оптимізації коду та підвищенню продуктивності роботи з даними. Завдяки такій структурі бази

даних можуть ефективно обробляти великі обсяги інформації, забезпечуючи швидкий доступ до даних для користувачів і програм.

SQL (Structured Query Language) — це мова, створена для роботи з реляційними базами даних. У середовищі MS SQL Server SQL використовується для виконання різноманітних операцій із даними. Основні дії, які можна виконати за допомогою SQL у цій СУБД, включають:

Команда **SELECT** дозволяє отримати дані з таблиць. Можна обирати всі записи або лише ті, що відповідають заданим критеріям.

За допомогою команди **INSERT** можна вносити нові рядки в таблиці.

UPDATE дозволяє змінювати наявні дані, вказуючи конкретні умови для їх оновлення.

Команда **DELETE** застосовується для вилучення рядків із таблиці відповідно до заданих умов.

Завдяки функціям групування SQL дає змогу об'єднувати записи за певними критеріями та виконувати агрегатні обчислення, як-от **SUM, AVG, COUNT**.

Операції **JOIN** використовуються для отримання даних із кількох таблиць, встановлюючи зв'язки між ними.

SQL підтримує створення **запитів усередині інших запитів**, що допомагає отримати додаткові дані або уточнити результати.

Команди **CREATE, ALTER, DROP** застосовуються для створення, модифікації або видалення таких об'єктів, як таблиці, процедури, функції та індекси.

SQL є універсальним інструментом для ефективного управління даними, виконання запитів і забезпечення роботи структури бази даних. Його застосовують для **створення, оновлення, видалення даних**, а також для оптимізації процесів взаємодії з реляційними базами, такими як MS SQL Server.

Окрім стандартної мови SQL, MS SQL Server підтримує сучасні інструменти й мови програмування, що розширюють її функціональність і

можливості інтеграції з іншими платформами.

T-SQL (Transact-SQL) — це розширення стандартного SQL, яке використовується саме в MS SQL Server. Воно надає додаткові інструменти для створення складної логіки безпосередньо у запитах. T-SQL підтримує використання змінних, функцій, процедур, обробку помилок, створення тригерів та інші програмні конструкції, що робить її надзвичайно потужною для роботи з базами даних..

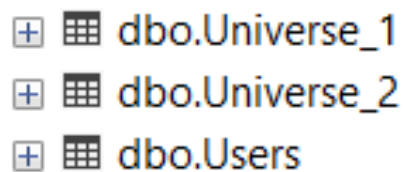


Рисунок 3.6 – Робочі таблиці.

Universe_1 — це таблиця, яка містить дані про вагони. Вона включає такі стовпці: ID, Numbers (номер вагона), Point (станція), Cargo (вантаж), Arrive_Time (час відправлення зі станції), End_Time (час прибуття на станцію), Path (шлях вагона), End_Point (кінцева станція), Weight (вага) та Max_Weight (максимальна вага).

ID	int	<input type="checkbox"/>
Numbers	int	<input type="checkbox"/>
Point	nvarchar(50)	<input type="checkbox"/>
Cargo	nvarchar(50)	<input type="checkbox"/>
Arrive_Time	datetime	<input type="checkbox"/>
Weight	float	<input checked="" type="checkbox"/>
MAX_weight	float	<input checked="" type="checkbox"/>
Path	nvarchar(50)	<input checked="" type="checkbox"/>
End_Time	datetime	<input checked="" type="checkbox"/>
End_Point	nvarchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Рисунок 3.7 – Загальна структура таблиці Universe_1

Пропоную переглянути, як виглядає заповнена таблиця Universe_1.

	ID	Numbers	Point	Cargo	Arrive_Time	Weight	MAX_weight	Path	End_Time	End_Point
1	7265	17700073	Станція: 1	вугілля	2024-11-06 16:11:27.243	70170	72934	Шлях: 2	NULL	NULL
2	7266	17700074	Станція: 1	вугілля	2024-11-06 16:08:35.967	49777	55702	Шлях: 1	NULL	NULL
3	7267	17700075	Станція: 1	вугілля	2024-11-06 16:08:35.987	58670	67289	Шлях: 1	NULL	NULL
4	7268	10658060	Станція: 1	вугілля	2024-11-06 16:08:35.993	93421	102982	Шлях: 2	NULL	NULL
5	7269	10658061	Станція: 1	чавун	2024-11-06 16:08:36.000	41020	48884	Шлях: 2	NULL	NULL
6	7270	10658062	Станція: 1	чавун	2024-11-06 16:08:36.003	81799	90340	Шлях: 2	NULL	NULL
7	7271	11276472	Станція: 1	чавун	2024-11-06 16:08:36.010	78321	85937	Шлях: 3	NULL	NULL
8	7272	11276473	Станція: 1	чавун	2024-11-06 16:08:36.013	88095	93692	Шлях: 3	NULL	NULL
9	7273	11276474	Станція: 1	чавун	2024-11-06 16:08:36.020	52757	57917	Шлях: 3	NULL	NULL
10	7274	17700076	Станція: 2	вугілля	2024-11-06 16:08:26.023	56508	64276	Шлях: 1	NULL	NULL
11	7275	17700077	Станція: 2	вугілля	2024-11-06 16:08:26.030	67865	77298	Шлях: 1	NULL	NULL
12	7276	17700078	Станція: 2	вугілля	2024-11-06 16:08:26.033	61438	68798	Шлях: 1	NULL	NULL
13	7277	10658063	Станція: 2	вугілля	2024-11-06 16:08:26.040	47907	57865	Шлях: 2	NULL	NULL
14	7278	10658064	Станція: 2	чавун	2024-11-06 16:08:26.047	58774	63830	Шлях: 2	NULL	NULL
15	7279	10658065	Станція: 2	чавун	2024-11-06 16:08:26.050	47886	56403	Шлях: 2	NULL	NULL
16	7280	11276475	Станція: 2	чавун	2024-11-06 16:08:26.057	47292	54077	Шлях: 3	NULL	NULL
17	7281	11276476	Станція: 2	чавун	2024-11-06 16:08:26.067	72816	81282	Шлях: 3	NULL	NULL

Рисунок 3.8 – Таблиця Universe_1 з наповненими даними

Наступною розглянемо таблицю Universe_2, яка містить інформацію про вагони, що вже були переміщені на інші станції. Її структура схожа до таблиці Universe_1 та має аналогічні поля.

	ID	Numbers	Point	Cargo	Arrive_Time	Weight	MAX_weight	Path	End_Time	End_Point
1	7265	17700073	Станція: 1	вугілля	2024-11-06 16:11:27.243	70170	72934	Шлях: 2	NULL	NULL
2	7266	17700074	Станція: 1	вугілля	2024-11-06 16:08:35.967	49777	55702	Шлях: 1	NULL	NULL
3	7267	17700075	Станція: 1	вугілля	2024-11-06 16:08:35.987	58670	67289	Шлях: 1	NULL	NULL
4	7268	10658060	Станція: 1	вугілля	2024-11-06 16:08:35.993	93421	102982	Шлях: 2	NULL	NULL
5	7269	10658061	Станція: 1	чавун	2024-11-06 16:08:36.000	41020	48884	Шлях: 2	NULL	NULL
6	7270	10658062	Станція: 1	чавун	2024-11-06 16:08:36.003	81799	90340	Шлях: 2	NULL	NULL
7	7271	11276472	Станція: 1	чавун	2024-11-06 16:08:36.010	78321	85937	Шлях: 3	NULL	NULL
8	7272	11276473	Станція: 1	чавун	2024-11-06 16:08:36.013	88095	93692	Шлях: 3	NULL	NULL
9	7273	11276474	Станція: 1	чавун	2024-11-06 16:08:36.020	52757	57917	Шлях: 3	NULL	NULL
10	7274	17700076	Станція: 2	вугілля	2024-11-06 16:08:26.023	56508	64276	Шлях: 1	NULL	NULL
11	7275	17700077	Станція: 2	вугілля	2024-11-06 16:08:26.030	67865	77298	Шлях: 1	NULL	NULL
12	7276	17700078	Станція: 2	вугілля	2024-11-06 16:08:26.033	61438	68798	Шлях: 1	NULL	NULL
13	7277	10658063	Станція: 2	вугілля	2024-11-06 16:08:26.040	47907	57865	Шлях: 2	NULL	NULL
14	7278	10658064	Станція: 2	чавун	2024-11-06 16:08:26.047	58774	63830	Шлях: 2	NULL	NULL
15	7279	10658065	Станція: 2	чавун	2024-11-06 16:08:26.050	47886	56403	Шлях: 2	NULL	NULL
16	7280	11276475	Станція: 2	чавун	2024-11-06 16:08:26.057	47292	54077	Шлях: 3	NULL	NULL
17	7281	11276476	Станція: 2	чавун	2024-11-06 16:08:26.067	72816	81282	Шлях: 3	NULL	NULL

Рисунок 3.9 – Universe_2 з вагонами які ми перемістили раніше.

Пропоную переглянути вигляд таблиці Universe_2 у коді мого додатку.

```

Ссылк: 8
public class UniverseEntity_Unnormal
{
    Ссылк: 0
    public int ID { get; set; }
    Ссылк: 4
    public int Numbers { get; set; }
    Ссылк: 3
    public string Point { get; set; }
    Ссылк: 3
    public string Cargo { get; set; }
    Ссылк: 3
    public DateTime Arrive_Time { get; set; }
    Ссылк: 3
    public double Weight { get; set; }
    Ссылк: 4
    public double Max_Weight { get; set; }
    Ссылк: 3
    public string Path { get; set; }
    Ссылк: 5
    public DateTime? End_Time { get; set; }
    Ссылк: 3
    public string? End_Point { get; set; }
}

```

Рисунок 3.10 – клас UniverseEntity_Unnormal, що відображає таблицю

Третьою таблицею в нашій базі є Users, яка зберігає дані про

zareestrovanih korystuvachiv systemi. Її стовпці включають: id, Email, Password (пароль), Number (номер телефону) та address (адреса користувача).

id	int	<input type="checkbox"/>
Email	nvarchar(50)	<input type="checkbox"/>
Password	nvarchar(50)	<input type="checkbox"/>
Number	int	<input type="checkbox"/>
Adress	nvarchar(50)	<input type="checkbox"/>
		<input type="checkbox"/>

Рисунок 3.11 – загальна структура таблиці Users.

За взаємодію програми з базою даних відповідає клас ApplicationDbContext.

```

namespace Diplom_Mag.Models
{
    Ссылка: 8
    public class ApplicationDbContext : DbContext
    {
        Ссылка: 0
        public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options) : base(options) { }

        Ссылка: 14
        public DbSet<UniverseEntity> Universe_1 { get; set; }
        Ссылка: 3
        public DbSet<UniverseEntity_Unnormal> Universe_2 { get; set; }
        Ссылка: 3
        public DbSet<UserModel> Users { get; set; }
    }
}

```

Рис. 3.12 – Загальна структура класу ApplicationDbContext

ApplicationDbContext є класом, що використовується в рамках Entity Framework у додатках ASP.NET Core для роботи з базою даних. Цей клас є похідним від DbContext і слугує посередником між додатком та реляційною базою даних.

Його головна задача — забезпечити зручний доступ до даних та виконання операцій із таблицями. Клас містить властивості, які відповідають таблицям у базі, та методи для роботи з цими даними.

Таким чином, ApplicationDbContext є основним інструментом для роботи з базою даних у середовищі ASP.NET Core, що дозволяє значно спростити процес обробки реляційних даних у додатку.

Для налаштування роботи з Entity Framework необхідно встановити NuGet-пакети:

Microsoft.EntityFrameworkCore

Microsoft.EntityFrameworkCore.SqlServer

Microsoft.EntityFrameworkCore.Tools

NuGet є менеджером пакетів, який забезпечує зручний спосіб додавання, оновлення або видалення бібліотек і компонентів у проектах, створених у середовищі Visual Studio.

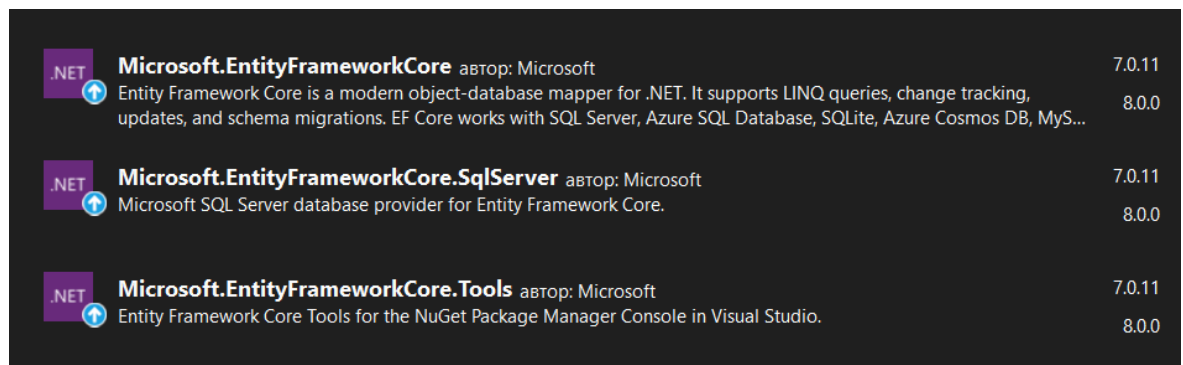


Рисунок 3.12 – Список інстальованих пакетів NuGet

Розглянемо кожний з встановлених пакетів, а саме яка їх функція.

Microsoft.EntityFrameworkCore - це пакет, що надає зручний інтерфейс для взаємодії з базами даних в програмах, написаних на .NET. Він дозволяє розробникам використовувати об'єктно-орієнтовані об'єкти в коді для взаємодії з даними в реляційних базах даних. Він автоматично вирішує відображення між об'єктами в коді і таблицями в базі даних. Також він надає механізм міграцій, що дозволяє автоматично оновлювати структуру бази даних відповідно до змін у моделі даних. EF Core відстежує зміни в об'єктах і автоматично вносить відповідні зміни в базу даних, що спрощує роботу з даними. Загалом він ідеально поєднується з ASP.NET Core для створення повнофункціональних веб-додатків з доступом до бази даних.

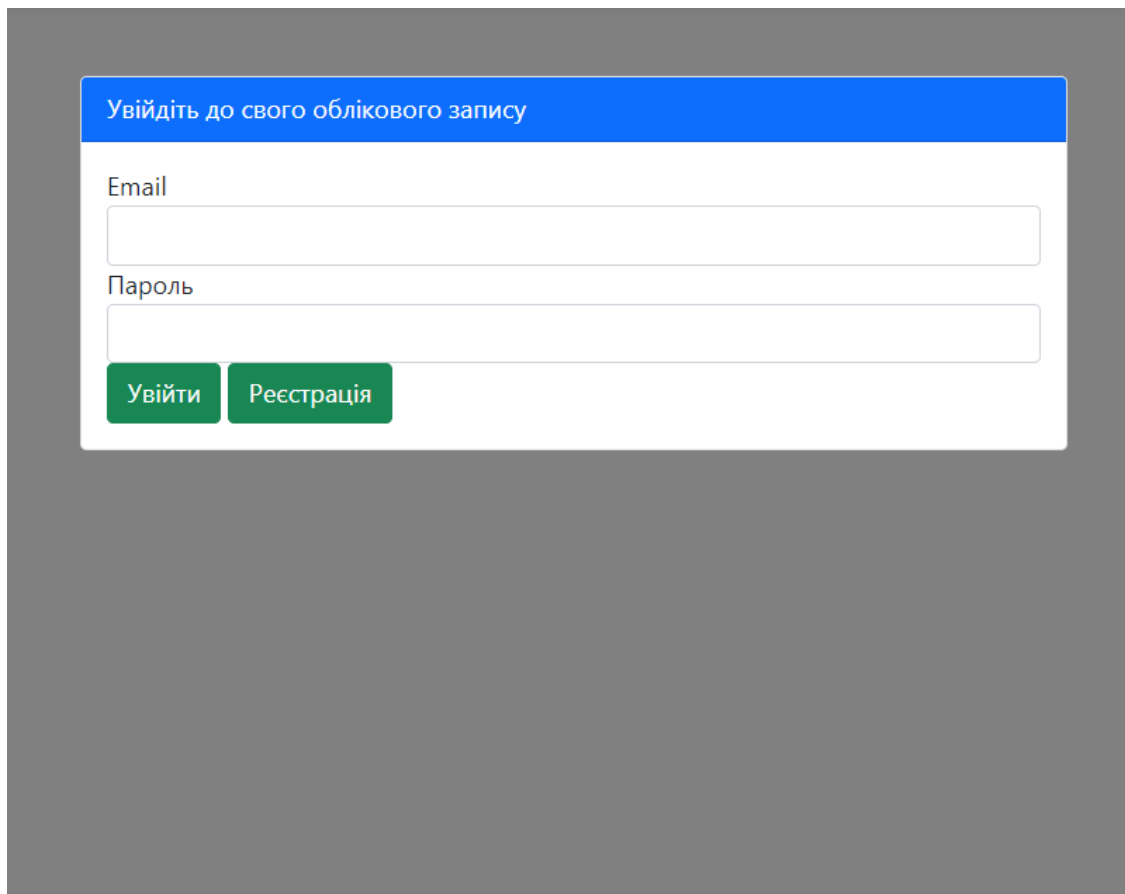
Microsoft.EntityFrameworkCore.SqlServer - це додатковий пакет для Entity Framework Core, який надає підтримку взаємодії з базами даних SQL Server. Його можна використати для забезпечення підтримки для SQL Server.

Пакет включає постачання (provider) для Entity Framework Core, яке дозволяє EF Core взаємодіяти з базою даних SQL Server. Забезпечує підтримку особливостей, які притаманні саме SQL Server, такі як індекси, тригери, процедури, функції та інше.

В свою чергу Microsoft.EntityFrameworkCore.Tools є пакетом інструментів для Entity Framework Core, який надає ряд команд інтерфейсу командного рядка (CLI) для спрощення роботи з Entity Framework Core в процесі розробки. Цей пакет забезпечує різні корисні команди, які можна використовувати для міграцій бази даних, генерації коду та інших завдань.

3.4 Реалізація програмного забезпечення

Під час першого запуску нашої програми користувач потрапляє на головну сторінку, приклад якої представлений на рисунку 3.13.



The image shows a login form with a blue header bar containing the text "Увійдіть до свого облікового запису". Below the header are two input fields: "Email" and "Пароль". At the bottom of the form are two green buttons: "Увійти" and "Реєстрація".

Рисунок 3.13 – Основне вікно програми

Якщо ми обираємо реєстрація то переходимо в меню де потрібно ввести свої дані. Після чого потрібно ввести дані у відповідні поля Email та пароль.

```
public class UserModel
{
    Ссылка: 0
    public int Id { get; set; }

    [Required(ErrorMessage = "Поле Email обов'язкове для заповнення")]
    [EmailAddress(ErrorMessage = "Введіть коректний email")]
    Ссылка: 5
    public string Email { get; set; }

    [Required(ErrorMessage = "Поле Пароль обов'язкове для заповнення")]
    Ссылка: 3
    public string Password { get; set; }

    [Required(ErrorMessage = "Поле Адреса обов'язкове для заповнення")]
    Ссылка: 2
    public string Address { get; set; }

    [Required(ErrorMessage = "Поле Номер обов'язкове для заповнення")]
    [Range(1, int.MaxValue, ErrorMessage = "Номер повинен бути більше 0")]
    Ссылка: 2
    public int Number { get; set; }
}
```

Рисунок 3.14 – Клас UserModel який відповідає за процес реєстрації

При реєстрації всі поля форми є обов'язковими для заповнення, а також реалізовано механізм валідації введених даних.

Реєстрація

Email
test@gmail.com

Пароль
....

Адреса
Віталія Матусевича 11

Номер телефону
0564090606

Зареєструватися

Рисунок 3.15 – Вікно для реєстрації

Після успішного завершення реєстрації введена інформація автоматично

зберігається в таблиці Users, яку ми розглядали раніше..

12	test1@gmail.com	test1	976374760	Віталія Матусевича 11
13	test1@gmail.com	test2	976374657	Віталія Матусевича 11
14	test2@gmail.com	test1	97865263	Віталія Матусевича 11
15	test1@gmail.com	test3	987626237	Віталія Матусевича 11

Рисунок 3.16 – Дані зареєстрованих користувачів

Наступним кроком є авторизація: користувач вводить свій Email та пароль. У разі успішного входу відкривається особистий кабінет, який є першою сторінкою після входу.

Оберіть які вагони вас цікавлять?

Дані про вагони

Рисунок 3.17 – Особистий кабінет

Переходячи на наступну сторінку за допомогою відповідної кнопки, користувач отримує інформацію про вагони.

Сортувати по номерам	Сортувати по місто відправки	Сортувати по грузу	Сортувати по часу відправки	Сортувати по вазі
Сортувати по максимальній вазі тари	Сортувати по шляху			

Дані про вагони

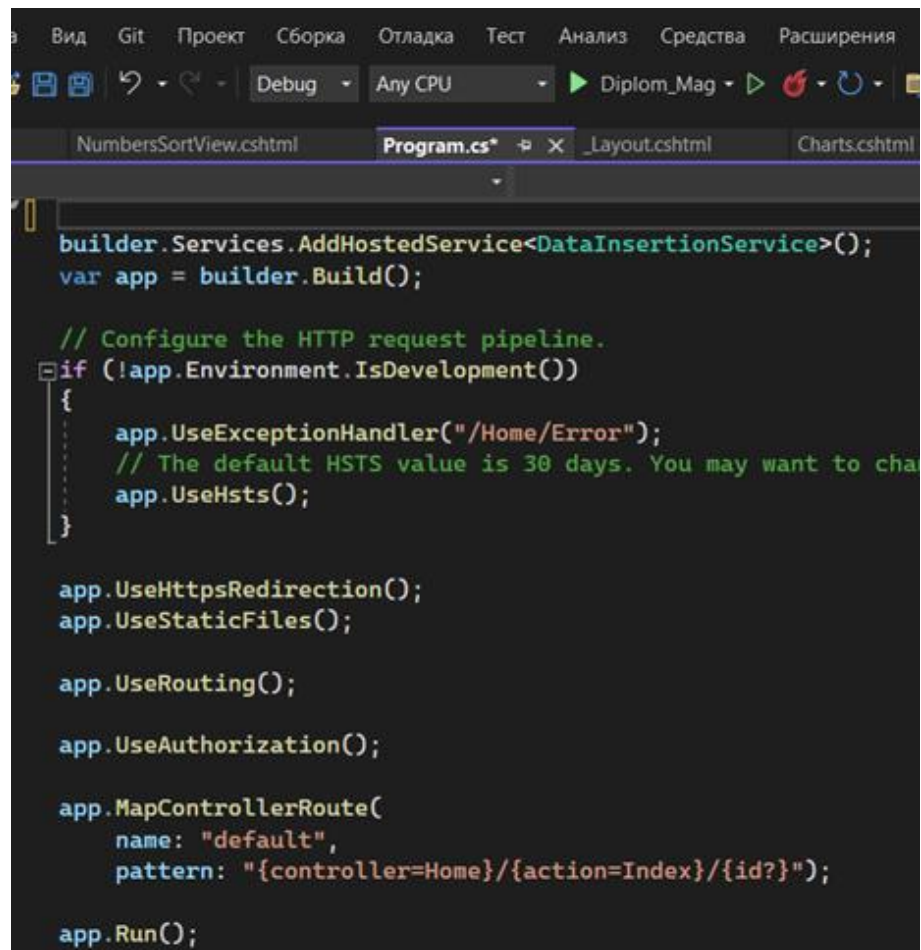
Номер	З станції	Груз	Час відправки	Вага	Максимальна вага тари	Шлях	Кінцевий час	На станцію
15849852	Станція: 1	чавун	2024-11-06 16:32:05	85319	95181	Шлях: 1		
15849853	Станція: 1	чавун	2024-11-06 16:32:07	72798	78930	Шлях: 1		
15849854	Станція: 1	вугілля	2024-11-06 16:32:07	45252	51680	Шлях: 1		
17500095	Станція: 1	вугілля	2024-11-06 16:32:07	95665	101828	Шлях: 2		
17500096	Станція: 1	чавун	2024-11-06 16:32:07	66513	73708	Шлях: 2		
17500097	Станція: 1	вугілля	2024-11-06 16:32:07	58322	65308	Шлях: 2		
16448058	Станція: 1	вугілля	2024-11-06 16:32:07	46885	56076	Шлях: 3		
16448059	Станція: 1	чавун	2024-11-06 16:32:07	97584	105693	Шлях: 3		
16448060	Станція: 1	чавун	2024-11-06 16:32:07	50245	57674	Шлях: 3		
15849855	Станція: 2	чавун	2024-11-06 16:32:07	77390	82830	Шлях: 1		
15849856	Станція: 2	вугілля	2024-11-06 16:32:07	58087	63835	Шлях: 1		
15849857	Станція: 2	чавун	2024-11-06 16:32:07	46304	55773	Шлях: 1		
17500098	Станція: 2	чавун	2024-11-06 16:32:07	51988	57130	Шлях: 2		
17500099	Станція: 2	чавун	2024-11-06 16:32:07	74993	83547	Шлях: 2		
17500100	Станція: 2	вугілля	2024-11-06 16:32:07	61731	68609	Шлях: 2		

Рисунок 3.18 – Вікно з інформацією про вагони

Для кожного вагону номер формується з восьмизначного числа, а значення ваги генерується випадковим чином. Загальна кількість вагонів складає двадцять сім: на кожній із трьох станцій розміщено три шляхи, на кожному з яких знаходиться три вагони.

Генерація даних була реалізована за допомогою спеціального сервісу, інтегрованого в клас Program. У рамках ASP.NET Core клас Program виконує роль конфігуратора веб-додатку. У ньому налаштовуються сервіси та програмується маршрути (middleware), необхідні для роботи додатку.

Термін "сервіси" (services) в ASP.NET Core охоплює компоненти, що забезпечують додаткові функції, такі як робота з базами даних, логування, кешування, конфігурація тощо. Для організації доступу до сервісів ASP.NET Core використовує впровадження залежностей (Dependency Injection, DI).



```
builder.Services.AddHostedService<DataInsertionService>();
var app = builder.Build();

// Configure the HTTP request pipeline.
if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Home/Error");
    // The default HSTS value is 30 days. You may want to change this for production.
    app.UseHsts();
}

app.UseHttpsRedirection();
app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");

app.Run();
```

Рисунок 3.19 – Фрагмент коду для підключення сервісу та налаштування маршрутизації.

Додавання сервісів полягає в реєстрації їх у контейнері залежностей, який відповідає за створення та передачу об'єктів у потрібні місця програми. Після реєстрації ці сервіси можна використовувати в різних частинах додатку, зокрема в контролерах, middleware або фільтрах..

```

var cargo = (random.Next(1, 3) == 1) ? "вугілля" :
var arriveTime = DateTime.Now;
var weight = random.Next(40000, 100000);
var maxWeight = weight + random.Next(5000, 10000);
var numbers = pathNumbers[path]; // Отримуємо число
var numberExists = dbContext.Universe_1.Any(e => e
if (!numberExists)
{
    dbContext.Universe_1.Add(new UniverseEntity
    {
        Numbers = numbers,
        Point = $"Станція: {station}",
        Cargo = cargo,
        Arrive_Time = arriveTime,
        Weight = weight,
        Max_Weight = maxWeight,
        Path = $"Шлях: {path}",
        End_Time = null, // Установлюємо End_Time
        End_Point = null
    });
    pathNumbers[path]++; // Збільшуємо число для
}
}

```

Рисунок 3.20 – метод GenerateAndInsertData.

За допомогою зазначеного методу в нашому коді здійснюється генерація інформації про вагони.

Як можна помітити, окрім створення даних про вагони, ця сторінка також відповідає за сортування інформації.

```
switch (sortField)
{
    case "Numbers":
        universeData = universeData.OrderBy(e => e.Numbers).ToList();
        break;
    case "Point":
        universeData = universeData.OrderBy(e => e.Point).ToList();
        break;
    case "Weight":
        universeData = universeData.OrderBy(e => e.Weight).ToList();
        break;
    case "Max_Weight":
        universeData = universeData.OrderBy(e => e.Max_Weight).ToList();
        break;
    case "Path":
        universeData = universeData.OrderBy(e => e.Path).ToList();
        break;
    default:
        // By default, sort by ID or do nothing
        universeData = universeData.OrderBy(e => e.ID).ToList();
        break;
}

return View(universeData);
```

Рисунок 3.21 – Фрагмент коду, що відповідає за сортування.

Коли ми обираємо конкретну станцію, відкривається сторінка, на якій відображаються всі вагони, прив'язані до цієї станції. Наприклад, якщо ми натиснемо на "Станція 1", то отримаємо список вагонів, які належать до першої станції..

Дані про вагони на станції

Оберіть номер вагону

Оберіть станцію призначення

Перемістити вагон

Перейти на сторінку переміщених вагонів

Раніше переміщенні вагони

Номер	З станції	Груз	Час відправки	Вага	Максимальна вага тари	Шлях	Кінцевий час	На станцію
15849852	Станція: 1	чавун	2024-11-06 16:32:05	85319	95181	Шлях: 1		
15849853	Станція: 1	чавун	2024-11-06 16:32:07	72798	78930	Шлях: 1		
15849854	Станція: 1	вугілля	2024-11-06 16:32:07	45252	51680	Шлях: 1		
17500095	Станція: 1	вугілля	2024-11-06 16:32:07	95665	101828	Шлях: 2		
17500096	Станція: 1	чавун	2024-11-06 16:32:07	66513	73708	Шлях: 2		
17500097	Станція: 1	вугілля	2024-11-06 16:32:07	58322	65308	Шлях: 2		
16448058	Станція: 1	вугілля	2024-11-06 16:32:07	46885	56076	Шлях: 3		

Рисунок 3.22 – Сторінка з інформацією про вагони на першій станції

Перемістимо один із вагонів з першої станції на іншу, наприклад, на третю. Спочатку у випадяючому списку вибираємо номер вагона.

Дані про вагони на станції

Оберіть номер вагону

Оберіть номер вагону

15849852

15849853

15849854

17500095

17500096

17500097

16448058

16448059

16448060

17937623

17937624

17937625

11829685

11829686

11829687

11123079

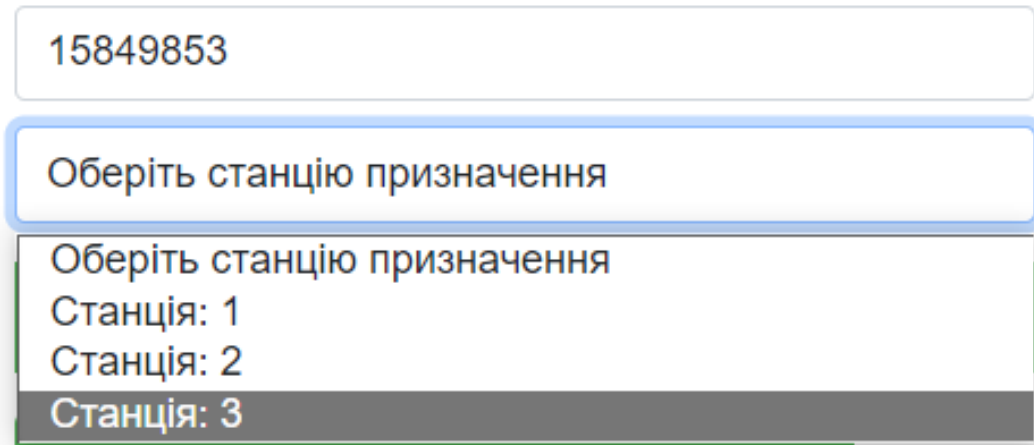
11123080

11123081

Рисунок 3.23 – Вибір номеру вагона зі списку

Потім визначаємо станцію призначення.

Дані про вагони на станції



15849853

Оберіть станцію призначення

Оберіть станцію призначення

Станція: 1

Станція: 2

Станція: 3

Рисунок 3.24 – Вибір станції призначення вагона

Після цього натискаємо кнопку для переміщення вагона. У верхній частині екрану з'являється відповідне повідомлення.

Дані про вагони на станції

Вагон 15849853 переміщено на Станція: 3

Рисунок 3.25 – повідомлення про успішне переміщення вагону.

Далі переходимо на сторінку з інформацією про переміщені вагони, яка виглядає наступним чином.

Інформація про переміщений вагон

Оберіть шлях

Зберегти шлях

Навантаження

Розвантаження

Перейти до списку вагонів

Номер	З станції	Груз	Час відправки	Вага	Максимальна вага тари	Шлях	Кінцевий час	На станцію
15849853	Станція: 3	чавун	2024-11-06 3.26.40	72798	78930		2024-11-06 13.26.40	

Рисунок 3.26 – Вікно з інформацією про переміщений вагон.

Під час переміщення вагона система автоматично заповнює поле з часом завершення операції, що відображає момент переміщення, а також вказує станцію призначення, куди було доставлено вагон.

```
[HttpPost]
Ссылка: 0
public IActionResult MoveWagon(MoveWagonViewModel model)
{
    try
    {
        if (string.IsNullOrEmpty(model.WagonNumber) || string.IsNullOrEmpty(model.DestinationStation))
        {
            ViewBag.ErrorMessage = "Необхідно ввести номер вагона та станцію призначення";
            return RedirectToAction("StationSortView");
        }

        // Знайти вагон у базі даних за номером
        var wagon = _context.Universe_1.FirstOrDefault(e => e.Numbers.ToString() == model.WagonNumber);

        if (wagon == null)
        {
            ViewBag.ErrorMessage = $"Вагон з номером {model.WagonNumber} не знайдено";
            return RedirectToAction("StationSortView");
        }

        // Зберегти старі значення станції та шляхи
        var oldStation = wagon.Point;
```

Рисунок 3.27 – Фрагмент коду, що реалізує логіку переміщення вагонів.

Розгляньмо детальніше функціонал цієї сторінки. Тут доступні дві кнопки — "навантаження" та "розвантаження". Під час їх натискання додається або віднімається вага відповідно. Основна мета цієї сторінки — вибір маршруту для вагона та збереження його. Наприклад, можна обрати другий маршрут і натиснути кнопку для збереження.

Номер	З станції	Груз	Час відправки	Вага	Максимальна вага тари	Шлях	Кінцевий час	На станцію
15849853	Станція: 3	чавун	2024-11-06 13:26:40	72798	78930	Шлях: 2		

Рисунок 3.28 – Інформація про вагон після вибору нового маршруту.

Як результат, вагон отримує нову станцію та новий маршрут, а час відправлення відповідає моменту виконання переміщення. Проведемо аналогічні операції кілька разів і переглянемо таблицю Universe_2, яка містить дані про раніше переміщені вагони. Сторінка з цією інформацією виглядає так.

Дані про минулі переміщення вагонів

Номер	З станції	Груз	Час відправки	Вага	Максимальна вага тари	Шлях	Кінцевий час	На станцію
15849853	Станція: 1	чавун	2024-11-06 16:32:07	72798	78930	Шлях: 1	2024-11-06 13:22:52	Станція: 3
15849853	Станція: 3	чавун	2024-11-06 13:22:52	72798	78930		2024-11-06 13:26:30	Станція: 2
15849853	Станція: 2	чавун	2024-11-06 13:26:30	72798	78930	Шлях: 1	2024-11-06 13:26:40	Станція: 3
17500095	Станція: 1	вугілля	2024-11-06 16:32:07	95665	101828	Шлях: 2	2024-11-06 13:40:11	Станція: 3
15849853	Станція: 3	чавун	2024-11-06 13:26:40	72798	78930	Шлях: 2	2024-11-06 13:40:36	Станція: 2
15849853	Станція: 2	чавун	2024-11-06 13:40:36	72798	78930	Шлях: 2	2024-11-06 13:40:44	Станція: 1
15849853	Станція: 1	чавун	2024-11-06 13:40:44	72798	78930	Шлях: 3	2024-11-06 13:40:51	Станція: 3
15849853	Станція: 3	чавун	2024-11-06 13:40:51	72798	78930	Шлях: 1	2024-11-06 13:41:06	Станція: 2
15849853	Станція: 2	чавун	2024-11-06 13:41:06	72798	78930	Шлях: 1	2024-11-06 13:41:13	Станція: 2
15849853	Станція: 2	чавун	2024-11-06 13:41:13	72798	78930	Шлях: 2	2024-11-06 13:41:21	Станція: 2
17500096	Станція: 1	чавун	2024-11-06 16:32:07	66513	73708	Шлях: 2	2024-11-06 13:41:36	Станція: 1
15849853	Станція: 2	чавун	2024-11-06 13:41:21	72798	78930	Шлях: 3	2024-11-06 13:42:38	Станція: 1
17500095	Станція: 3	вугілля	2024-11-06 13:40:11	95665	101828	Шлях: 1	2024-11-06 13:43:22	Станція: 3
15849853	Станція: 1	чавун	2024-11-06 13:42:38	72798	78930	Шлях: 1	2024-11-06 13:43:49	Станція: 3

Рисунок 3.29 – Інформація про попередні переміщення вагона.

Повертаючись до сторінки переміщення вагонів між станціями, при виборі номера вагона можна переглянути історію всіх попередніх переміщень.

Історія вагона

Номер	З станції	Груз	Час відправки	Вага	Максимальна вага тари	Шлях	Кінцевий час	На станцію
15849853	Станція: 1	чавун	2024-11-06 16:32:07	72798	78930	Шлях: 1	2024-11-06 13:22:52	Станція: 3
15849853	Станція: 3	чавун	2024-11-06 13:22:52	72798	78930		2024-11-06 13:26:30	Станція: 2
15849853	Станція: 2	чавун	2024-11-06 13:26:30	72798	78930	Шлях: 1	2024-11-06 13:26:40	Станція: 3
15849853	Станція: 3	чавун	2024-11-06 13:26:40	72798	78930	Шлях: 2	2024-11-06 13:40:36	Станція: 2
15849853	Станція: 2	чавун	2024-11-06 13:40:36	72798	78930	Шлях: 2	2024-11-06 13:40:44	Станція: 1
15849853	Станція: 1	чавун	2024-11-06 13:40:44	72798	78930	Шлях: 3	2024-11-06 13:40:51	Станція: 3
15849853	Станція: 3	чавун	2024-11-06 13:40:51	72798	78930	Шлях: 1	2024-11-06 13:41:06	Станція: 2
15849853	Станція: 2	чавун	2024-11-06 13:41:06	72798	78930	Шлях: 1	2024-11-06 13:41:13	Станція: 2
15849853	Станція: 2	чавун	2024-11-06 13:41:13	72798	78930	Шлях: 2	2024-11-06 13:41:21	Станція: 2
15849853	Станція: 2	чавун	2024-11-06 13:41:21	72798	78930	Шлях: 3	2024-11-06 13:42:38	Станція: 1
15849853	Станція: 1	чавун	2024-11-06 13:42:38	72798	78930	Шлях: 1	2024-11-06 13:43:49	Станція: 3

Рисунок 3.30 – Записи про попередні переміщення вагона

```

@if (Model != null && Model.Any())
{
    <div class="table-responsive">
        <table class="table table-bordered table-striped">
            <thead>
                <tr>
                    <th>Номер</th>
                    <th>З станції</th>
                    <th>Груз</th>
                    <th>Час відправки</th>
                    <th>Вага</th>
                    <th>Максимальна вага тари</th>
                    <th>Шлях</th>
                    <th>Кінцевий час</th>
                    <th>На станцію</th>
                </tr>
            </thead>
            <tbody>
                @foreach (var item in Model)
                {
                    <tr>
                        <td>@item.Numbers</td>
                        <td>@item.Point</td>
                        <td>@item.Cargo</td>

```

Рисунок 3.31 – Фрагмент коду для сторінки з раніше переміщеними вагонами.

Тепер виконаємо встановлення бібліотеки AmCharts 4 і додамо її в каталог lib в нашій проєкті.

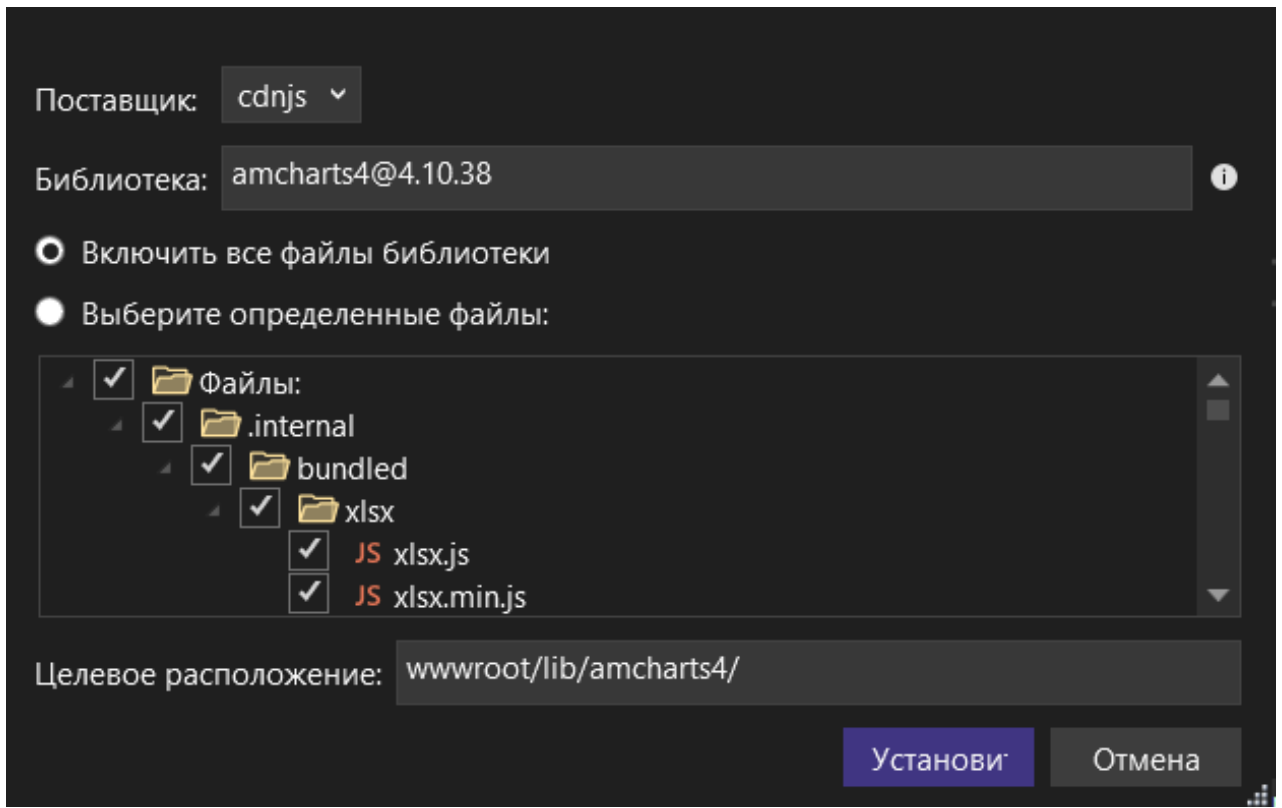


Рисунок 3.32 – інсталяція Amcharts4.

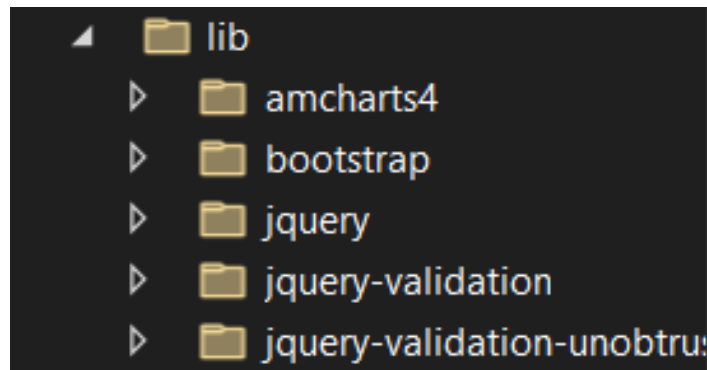


Рисунок 3.33 – розташування бібліотеки Amcharts4.

AmCharts 4 — це потужна JavaScript-бібліотека, призначена для створення інтерактивних графіків та діаграм. Вона дозволяє легко вбудовувати візуальні елементи для відображення даних, таких як лінійні графіки, стовпчасті діаграми, кругові діаграми та інші. Завдяки широкому вибору типів графіків, бібліотека забезпечує гнучкість у виборі способу візуалізації даних, що найкраще відповідає вашим потребам.

AmCharts 4 підтримує анімовані графіки, роблячи візуалізацію динамічною та привабливою. Крім того, вона дозволяє оновлювати графіки в реальному часі, що особливо корисно для демонстрації поточних даних або стрімінгових подій. Бібліотека включає вбудовані теми для швидкого налаштування зовнішнього вигляду графіків, а також підтримує детальну кастомізацію через CSS. Окремо слід зазначити можливість створення інтерактивних географічних карт із маркуванням регіонів та відображенням даних за географічним принципом.

Ця бібліотека є універсальним інструментом для розробників, які прагнуть інтегрувати зручні та ефективні рішення для візуалізації даних у свої проєкти. Хоча AmCharts 4 є комерційною бібліотекою, вона пропонує безкоштовну версію з певними обмеженнями. Перший графік є графіком зміни ваги відносно станцій.

На нашій сторінці історії вагонів при натисканні на кнопку графіків будуються два типи візуалізації:

Перший - графік змін ваги вагонів відносно станцій.

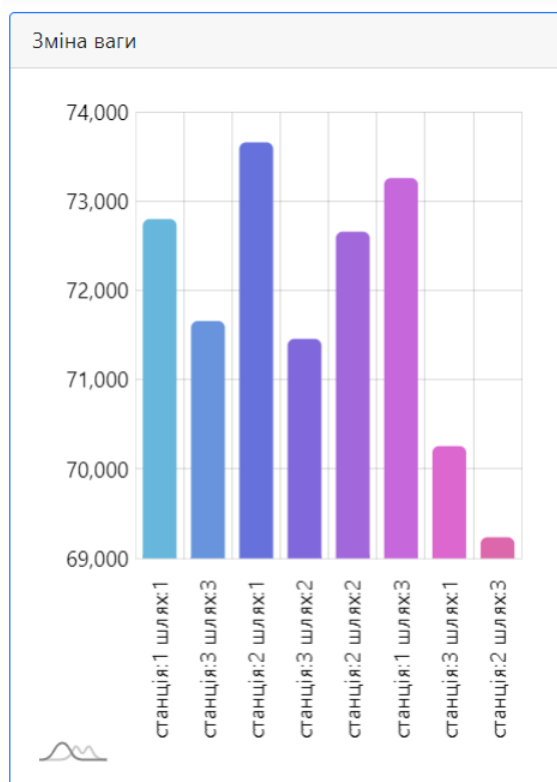


Рисунок 3.34 – Графік, що відображає зміну ваги.

Другий - графік, що відображає відсоткове співвідношення перебувань вагона на різних станціях.

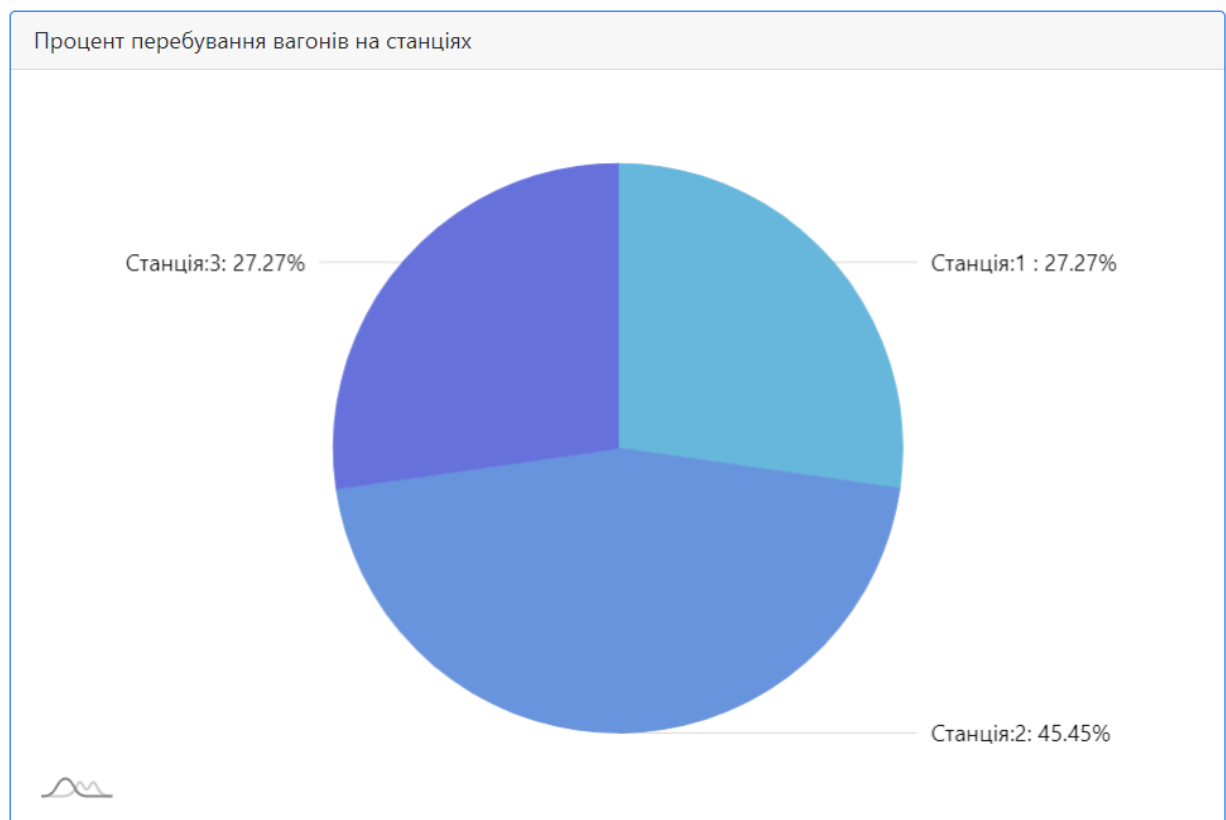


Рисунок 3.35 – Графік, що показує відсоток часу перебування вагона на станціях.

Таким чином, було продемонстровано роботу системи з усіх аспектів: переміщення вагонів, їх навантаження та розвантаження, збереження історії операцій, побудова графіків і взаємодія з програмним кодом. Детально показано процес змін станцій і маршрутів, а також візуалізовано показники ваги на графіках.

Висновки до розділу

У цьому розділі також розглянуто, що таке ASP.NET Core, описано його основні особливості, а також пояснено концепції model, view і controller. Проаналізовано таблиці проєкту з поясненням їхніх полів і додано скріншоти. Також надано огляд MS SQL, пояснено, за що відповідає кожен компонент, і наведено контекстну діаграму для кращого розуміння роботи системи. У

підсумку, показано всі аспекти роботи програмного забезпечення для контролю рухомого складу, зокрема його графічний інтерфейс, програмний код, графіки та інтеграцію всіх елементів системи.

ВИСНОКИ ТА РЕКОМЕНДАЦІЇ

У результаті кваліфікаційної роботи магістра, було побудовано ПЗ керування рухомим складом. Основною ідеєю при побудові даної системи була автоматизація та модернізація роботи підприємства. В межах виконаної магістерської роботи вдалось успішно вирішити низку ключових завдань, починаючи від розробки ефективної системи переміщення вагонів між станціями та завершуючи використання ПЗ для управління процесами завантаження та розвантаження.

Результати роботи вказують на значний прогрес у вдосконаленні логістичних операцій на підприємстві. Перевагою розробленої системи стало впровадження графіків на основі історії переміщень вагонів, й зміни його ваги що відкриває можливості для глибшого аналізу та стратегічного планування. Це робить систему не тільки потужним інструментом оптимізації поточних операцій, але і цінним інструментом для прийняття рішень на основі даних.

Зазначені у висновку практичні досягнення також відкривають перспективи для майбутніх досліджень і розвитку. Проект може слугувати базою для вдосконалення алгоритмів, впровадження новітніх технологій та розширення функціональності системи з метою відповіді на зростаючі вимоги та потреби підприємства.

У ході роботи над першим розділом системи було розглянуто структуру підприємства, проаналізовано роль залізничного транспорту в економічному розвитку країни, розглянуто вплив ПАТ "АМКР" на економіку України. Розглянуті загальні відомості про технологічний процес. Наведено інші існуючі ПЗ, задля кращого розуміння суті нашої роботи.

У ході роботи над другим розділом було розглянуто організаційні проблеми при управлінні рухомим складом, технічні і програмні аспекти та вимоги ПЗ рухомим складом, детально проведено аналіз перспектив розвитку ПЗ рухомого складу на основі сучасних тенденцій.

В свою чергу при роботі з третім розділом описано базу даних й усі її

таблиці, пояснено що повинна в собі містити кожна таблиця й кожне її поле, роз'яснено що таке ASP Net Core MVC та MS SQL. Після чого наведено контекстну діаграму вагону. З описом усіх можливих елементів, а саме: вхідні данні, вихідні данні, керуючі елементи й механізми. Побудовано UML діаграму послідовності роботи програми. Описано хід роботи нашої програми, з поясненням кожної сторінки в ній та наведеними скріншотами для кращого візуального розуміння програми.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Залізничний транспорт в Україні – URL: <https://mtu.gov.ua/content/informaciya-pro-ukrainski-zalznici.html> (дата звернення 14.08.2024)
2. Вплив залізничного транспорту на економіку – URL: <http://www.economy.nauka.com.ua/?op=1&z=2391> (дата звернення 18.08.2024)
3. Аналіз сучасного стану залізничного транспорту – URL: <http://lib.kart.edu.ua/bitstream/123456789/9213/1/Orlova.pdf> (дата звернення 19.08.2024)
4. АрселорМіттал – URL: <https://uk.wikipedia.org/wiki/ArcelorMittal> (дата звернення 29.08.2024)
5. Відомості про АМКР – URL: <http://krt.dp.ua/ua/pratsyuyuchi/21-arselor/48-pat-arselormittal-krivij-rig> (дата звернення 01.09.2024)
6. Історія розвитку АМКР – URL: <https://history.1kr.ua/publication/715> (дата звернення 01.09.2024)
7. АНАЛІЗ ПРОБЛЕМ АВТОМАТИЗАЦІЇ РУХОМИМ СКЛАДОМ – URL: <http://lib.kart.edu.ua/bitstream/123456789/5075/1/Автореферат%20Ковалева.pdf> (дата звернення 07.09.2024)
8. Thales. – URL: <https://www.thalesgroup.com/en> (дата звернення 07.09.2024)
9. Sensorline. – URL: <https://sensorline.de> (дата звернення 07.09.2024)
10. Trimble Inc. – URL: <https://uk.wikipedia.org/wiki/Trimble> (дата звернення 07.09.2024)
11. Omnitrac. – URL: <https://www.omnitrac.com> (дата звернення 07.09.2024)
12. Milestone Systems – URL: <https://www.milestonesys.com> (дата звернення 07.09.2024)
13. Argus –

- URL: <http://lib.kart.edu.ua/bitstream/123456789/15177/1/Chyhyryk.pdf>
(дата звернення 07.09.2024)
14. GSM-R – URL: https://www.rohde-schwarz.com/cac/technologies/cellular/gsm-r/gsm-r-technology/gsm_r_technology_55926.html (дата звернення 10.09.2024)
15. Технологія WDM –
URL: <https://fiberroad.com/uk/resources/glossary/wdm-technology/> (дата звернення 19.09.2024)
16. ETCS – URL: https://www.ukma.edu.ua/ects_eng/index.php/82-2011-04-05-15-19-11/24-about- (дата звернення 21.09.2024)
17. Система контролю рухомого складу –
URL: https://uk.wikipedia.org/wiki/Система_контролю_рухомого_складу
(дата звернення 22.09.2024)
18. ASP.NET Core MVC –
URL: <https://metanit.com/sharp/aspnetmvc/1.1.php> (дата звернення 25.09.2024)
19. Controller – URL: <https://metanit.com/sharp/mvc5/3.1.php> (дата звернення 05.10.2024)
20. View – URL: <https://elbrusboot.camp/blog/что-такое-mvc/> (дата звернення 05.10.2024)
21. Model – URL: <https://metanit.com/sharp/aspnetmvc/5.1.php> (дата звернення 05.10.2024)
22. MS SQL – URL: <https://metanit.com/sql/sqlserver/1.1.php> (дата звернення 10.10.2024)
23. Збережені процедури – URL: <https://metanit.com/sql/sqlserver/11.1.php> (дата звернення 10.10.2024)
24. Entity Framework – URL: <https://metanit.com/sharp/aspnetmvc/11.1.php>
(дата звернення 10.10.2024)
25. Сортування – URL: <https://metanit.com/sharp/aspnetmvc/11.4.php> (дата звернення 20.10.2024)

26. Insert – URL: <https://metanit.com/sql/sqlserver/4.1.php> (дата звернення 20.10.2024)

27. Select– URL: <https://metanit.com/sql/sqlserver/4.2.php> (дата звернення 20.10.2024)

28. Update – URL: <https://metanit.com/sql/sqlserver/4.7.php> (дата звернення 20.10.2024)

29. Delete – URL: <https://metanit.com/sql/sqlserver/4.8.php> (дата звернення 20.10.2024).

Зображення фрагментів Model

```
namespace Diplom_Mag.Models
{
    Ссылка: 1
    public class MoveWagonViewModel
    {
        Ссылка: 6
        public string WagonNumber { get; set; }
        Ссылка: 4
        public string DestinationStation { get; set; }
        Ссылка: 0
        public string Path { get; set; }
    }
}
```

Рисунок А.1 – клас за допомогою якого рухаємо вагон між станціями

```
namespace Diplom_Mag.Models
{
    Ссылка: 3
    public class ErrorViewModel
    {
        Ссылка: 2
        public string? RequestId { get; set; }

        Ссылка: 1
        public bool ShowRequestId => !string.IsNullOrEmpty(RequestId);
    }
}
```

Рисунок А.2 – клас ErrorViewModel

```
public class UniverseEntity
{
    Ссылка: 1
    public int ID { get; set; }
    Ссылка: 23
    public int Numbers { get; set; }
    Ссылка: 16
    public string Point { get; set; }
    Ссылка: 7
    public string Cargo { get; set; }
    Ссылка: 8
    public DateTime Arrive_Time { get; set; }
    Ссылка: 12
    public double Weight { get; set; }
    Ссылка: 6
    public double Max_Weight { get; set; }
    Ссылка: 12
    public string Path { get; set; }
    Ссылка: 9
    public DateTime? End_Time { get; set; }
    Ссылка: 4
    public string? End_Point { get; set; }
}
}
```

Рисунок А.3 – клас таблиці Universe_1

Лістинг програмного коду

```
[HttpPost]
Ссылка: 0
public IActionResult Login(LoginViewModel model)
{
    if (ModelState.IsValid)
    {
        if (IsValidUser(model))
        {
            return RedirectToAction("Road");
        }
        else
        {
            ViewBag.ErrorMessage = "Неправильний логін або пароль";
        }
    }
    return View("MyView", model);
}
```

Рисунок Б.1 – метод авторизації

```
[HttpPost]
Ссылка: 0
public IActionResult Register(UserModel model)
{
    if (ModelState.IsValid)
    {
        var existingUser = _context.Users.FirstOrDefault(u => u.Email == model.Email);
        if (existingUser is not null)
        {
            ModelState.AddModelError("Email", "Користувач з таким email вже існує.");
            return View("Registration", model);
        }

        _context.Users.Add(model);
        _context.SaveChanges();

        // Перенаправлення на сторінку "MyView" після успішної реєстрації
        return RedirectToAction("MyView");
    }

    return View("Registration", model);
}
```

Рисунок Б.2 – метод реєстрації

```

<body style="background-color: #gray;">
  <div class="container mt-5">
    <div class="row justify-content-center">
      <div class="col-md-6">
        <div class="card">
          <div class="card-header bg-primary text-white">Увійдіть до свого облікового запису</div>
          <div class="card-body">
            <form asp-controller="Home" asp-action="Login" method="post">
              <div class="form-group">
                <label asp-for="Email">Email</label>
                <input asp-for="Email" class="form-control" />
                <span asp-validation-for="Email" class="text-danger"></span>
              </div>
              <div class="form-group">
                <label asp-for="Password">Пароль</label>
                <input asp-for="Password" type="password" class="form-control" />
                <span asp-validation-for="Password" class="text-danger"></span>
              </div>
              <div class="form-group">
                <button type="submit" class="btn btn-success btn-block">Увійти</button>
                <a asp-controller="Home" asp-action="Registration" class="btn btn-success btn-block">Реєстрація</a>
              </div>
            </form>
          </div>
        </div>
      </div>
    </div>
  </div>

```

Проблеми не знайдені. Стр: 43 Симв: 2 Пробілы: 0

Струментальних засвідків для обробки даних Консоль диспетчера пакетів Список помибок Вивод

розробників

Для розробників

Visual Studio 2022 Developer PowerShell v17.2.3
 Copyright (c) 2022 Microsoft Corporation
 \lev20\OneDrive\Робочий стол\Diplom_Mag

Рисунок Б.3 – програмний код початкової сторінки

```

<h2>Реєстрація</h2>
<form asp-controller="Home" asp-action="Register" method="post">
  <div class="form-group">
    <label asp-for="Email">Email</label>
    <input asp-for="Email" class="form-control" />
  </div>
  <div class="form-group">
    <label asp-for="Password">Пароль</label>
    <input asp-for="Password" type="password" class="form-control" />
  </div>
  <div class="form-group">
    <label asp-for="Address">Адреса</label>
    <input asp-for="Address" class="form-control" />
  </div>
  <div class="form-group">
    <label asp-for="Number">Номер телефону</label>
    <input asp-for="Number" class="form-control" />
  </div>
  <button type="submit" class="btn btn-success btn-block">Зареєструватися</button>
</form>

```

Рисунок Б.4 – програмний код сторінки реєстрації

```

{
  <p class="text-success">@ViewBag.SuccessMessage</p>
}

@if (!string.IsNullOrEmpty(ViewBag.ErrorMessage))
{
  <p class="text-danger">@ViewBag.ErrorMessage</p>
}

<!-- Форма для вводу номера вагона, станції и шляха -->
<form class="move-form" method="post" asp-action="MoveWagon">
  <div class="form-group">
    <select class="form-control" name="WagonNumber" required>
      <option value="">Оберіть номер вагону</option>
      @foreach (var item in Model)
      {
        <option value="@item.Numbers">@item.Numbers</option>
      }
    </select>
  </div>

  <div class="form-group" style="margin-top: 10px;">
    <select class="form-control" name="DestinationStation" required>
      <option value="">Оберіть станцію призначення</option>
      @if (ViewData["AllStations"] is not null)
      {

```

Рисунок Б.5 – програмний код сторінки переміщення вагонів

```

.table {
  width: 100%;
  border-collapse: collapse;
  margin-bottom: 20px;
  background-color: #fff;
  box-shadow: 0 1px 4px rgba(0, 0, 0, 0.1);
}

.table th,
.table td {
  padding: 12px 15px;
  text-align: left;
  border-bottom: 1px solid #ddd;
}

.table th {
  background-color: #4CAF50;
  color: white;
  font-weight: bold;
  cursor: pointer;
}

.table th a,
.table th a:visited {
  color: white;

```

Рисунок Б.6 – стилізація сторінки.