

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-
КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
АВТОМАТИЗОВАНИХ СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Дослідження системи моніторингу статусів вантажного
транспорту в процесі збору врожаю»

на здобуття освітнього ступеня бакалавра
зі спеціальності 126 Інформаційні системи та технології
(код, найменування спеціальності)
освітньо-професійної програми Інформаційні системи та технології
(назва)

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело*

(підпис)

Олександр ДЕМБРОВСЬКИЙ
Ім'я, ПРІЗВИЩЕ здобувача

Виконав: здобувач вищої освіти гр. ІСД- 42
Олександр ДЕМБРОВСЬКИЙ
Ім'я, ПРІЗВИЩЕ

Керівник: PhD, Віра МИКОЛАЙЧУК
Ім'я, ПРІЗВИЩЕ
науковий ступінь,
вчене звання

Рецензент: _____
Ім'я, ПРІЗВИЩЕ
науковий ступінь,
вчене звання

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

Навчально-науковий інститут Інформаційних технологій

Кафедра Інженерії програмного забезпечення автоматизованих систем

Ступінь вищої освіти бакалавр

Спеціальність Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедрою ІПЗАС

_____ Каміла СТОРЧАК

« ____ » _____ 2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**
Дембровському Олександрю Сергійовичу

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Дослідження системи моніторингу статусів вантажного транспорту в процесі збору врожаю

керівник кваліфікаційної роботи _____ Віра МИКОЛАЙЧУК, PhD
(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «27» лютого 2024 р. № 36

2. Строк подання кваліфікаційної роботи «31» травня 2024 р.

3. Вихідні дані до кваліфікаційної роботи:

1. Науково-технічна література з теми бакалаврської роботи.

2. Технічні вимоги та проектування архітектури системи моніторингу

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Теоритичні основи моніторингу вантажного транспорту в агропромисловому комплексі

2. Аналіз та проектування системи моніторингу статусів вантажного транспорту в

процесі збору врожаю

3. Розробка та реалізація системи моніторингу статусів вантажного транспорту в процесі збору врожаю
Гейміфікація. Визначення та прийоми

5. Ілюстративний матеріал: *презентація*

6. Дата видачі завдання: «27» лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	27.02-05.03.2024	
2	Обґрунтування актуальності роботи	06.03-11.03.2024	
3	Теоритичні основи моніторингу вантажного транспорту в агропромисловому комплексі	12.03-27.03.2024	
4	Аналіз та проектування системи моніторингу статусів вантажного транспорту в процесі збору врожаю	28.03-10.04.2024	
5	Розробка та реалізація системи моніторингу статусів вантажного транспорту в процесі збору врожаю	11.04-15.05.2024	
7	Оформлення роботи: вступ, висновки, реферат	16.05-22.05.2024	
8	Розробка демонстраційних матеріалів	23.05-24.05.2024	

Здобувач(ка) вищої освіти

(підпис)

Олександр ДЕМБРОСЬКИЙ

(Ім'я, ПРІЗВИЩЕ)

Керівник

кваліфікаційної роботи

(підпис)

Віра МИКОЛАЙЧУК

(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 80 стор., 45 рис., 10 табл., 24 джерела.

Мета роботи полягає у розробці системи моніторингу статусів вантажного транспорту, оптимізованої для потреб агропромислового комплексу під час збору врожаю, що сприятиме підвищенню ефективності та зниженню витрат у логістичних процесах.

Об'єктом дослідження є логістичні процеси в агропромисловому комплексі.

Предметом дослідження є система моніторингу статусів вантажного транспорту, яка використовується в межах зазначених логістичних процесів.

У даній кваліфікаційній роботі розроблено та реалізовано систему моніторингу статусів вантажного транспорту для агропромислового комплексу. Робота включає теоретичний огляд сучасних методів. Описано особливості використання GPS та інших навігаційних систем у логістиці, а також проаналізовано системи вантажних перевезень під час збору врожаю. В другому розділі викладено аналіз та проектування системи моніторингу, включаючи вибір технологічного стеку, проектування бізнес-процесів та бази даних. Третій розділ присвячено розробці та реалізації системи, з акцентом на розробку архітектури системи, інтеграцію з базою даних та модульне тестування. Результати тестування підтвердили працездатність та ефективність системи. Робота показує, як інтеграція передових технологій може оптимізувати логістичні процеси і підвищити ефективність управління вантажним транспортом в агропромисловому секторі.

МОНІТОРИНГ ВАНТАЖНОГО ТРАНСПОРТУ, АГРОПРОМИСЛОВИЙ КОМПЛЕКС, GPS-МОНІТОРИНГ, ТЕЛЕМАТИКА, RFID-ТЕХНОЛОГІЇ, СИСТЕМИ УПРАВЛІННЯ ЛОГІСТИКОЮ, ІНТЕГРАЦІЯ СИСТЕМ, БАЗА ДАНИХ, МАШИННЕ НАВЧАННЯ, МОДУЛЬНЕ ТЕСТУВАННЯ.

ABSTRACT

The text part of the qualifying work for obtaining a bachelor's degree: 80 pages, 45 figures, 10 tables, 24 sources.

The goal of the work is to develop a system for monitoring the status of cargo transport, optimized for the needs of the agro-industrial complex during harvest, which will contribute to increasing efficiency and reducing costs in logistics processes.

The object of the study is logistics processes in the agro-industrial complex.

The subject of the study is the monitoring system of freight transport statuses, which is used within the specified logistics processes.

In this qualification work, a system for monitoring cargo transport statuses for the agro-industrial complex was developed and implemented. The work includes a theoretical overview of modern methods. Features of the use of GPS and other navigation systems in logistics are described, and freight transportation systems during harvest are analyzed. The second section describes the analysis and design of the monitoring system, including the selection of the technology stack, the design of business processes and databases. The third chapter is devoted to system design and implementation, with an emphasis on system architecture development, database integration, and unit testing. The test results confirmed the functionality and efficiency of the system. The work shows how the integration of advanced technologies can optimize logistics processes and increase the efficiency of freight transport management in the agro-industrial sector.

TRUCK MONITORING, AGRICULTURAL COMPLEX, GPS MONITORING, TELEMATICS, RFID TECHNOLOGIES, LOGISTICS MANAGEMENT SYSTEMS, SYSTEMS INTEGRATION, DATA BASE, MACHINE LEARNING, MODULAR TESTING.

ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ МОНІТОРИНГУ ВАНТАЖНОГО ТРАНСПОРТУ В АГРОПРОМИСЛОВОМУ КОМПЛЕКСІ.....	12
1.1 Огляд сучасних методів моніторингу вантажного транспорту	12
1.2 Особливості використання GPS та інших навігаційних систем у логістиці вантажних перевезень	14
1.3 Аналіз систем вантажних перевезень в процесі збору врожаю	16
1.4 Постановка задачі.....	24
РОЗДІЛ 2 АНАЛІЗ ТА ПРОЕКТУВАННЯ СИСТЕМИ МОНІТОРИНГУ СТАТУСІВ ВАНТАЖНОГО ТРАНСПОРТУ В ПРОЦЕСІ ЗБОРУ ВРОЖАЮ	27
2.1 Формулювання вимог до системи моніторингу вантажного транспорту	27
2.2 Вибір технологічного стеку для розробки системи моніторингу	31
2.3 Проектування основних бізнес-процесів системи.....	35
2.4 Проектування бази даних для системи моніторингу.....	38
2.5 Проектування архітектури системи моніторингу.....	42
РОЗДІЛ 3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ СИСТЕМИ МОНІТОРИНГУ СТАТУСІВ ВАНТАЖНОГО ТРАНСПОРТУ В ПРОЦЕСІ ЗБОРУ ВРОЖАЮ.....	50
3.1 Розроблення загальної архітектури системи	50
3.2 Розробка коду для керування контролером	62
3.3 Інтеграція бази даних з програмним забезпеченням.....	68
3.4 Розроблення модулів системи	70
3.5 Модульне тестування системи	78
3.6 Тестові сценарії та оцінка результатів розробки системи	79
ВИСНОВКИ	87
ПЕРЕЛІК ПОСИЛАНЬ	89
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)	92

ВСТУП

Актуальність теми роботи впливає з гострої потреби аграрного сектору України у підвищенні ефективності та оптимізації логістичних процесів. У контексті агропромислового виробництва, точний і своєчасний моніторинг транспортних потоків є критичним для забезпечення мінімізації витрат і максимізації прибутковості, особливо під час збору врожаю, коли часові рамки обмежені, а вартість затримок може бути величезною.

У дослідженні [1], автори аналізують вплив використання географічних інформаційних систем (ГІС) на оптимізацію логістичних процесів у аграрному секторі. Центральна тема роботи полягає у вивченні можливостей ГІС для підвищення ефективності та зниження витрат під час збору врожаю шляхом більш точного планування маршрутів транспортних засобів та управління ресурсами.

Автори провели емпіричний аналіз, використовуючи дані з декількох аграрних господарств, які інтегрували ГІС у свої логістичні системи. Вони зосередились на оцінці параметрів, таких як час транзиту, використання палива, загальні логістичні витрати та ефективність використання обладнання. Дослідження включало збір даних через GPS-трекери, встановлені на транспортних засобах, що дозволило авторам отримати детальну картину рухів в реальному часі та аналізувати їх в контексті планування маршрутів.

Значна увага в дослідженні приділялась аналізу того, як зміни в маршрутизації, здійснені на основі даних ГІС, впливають на загальну продуктивність процесів збору врожаю. Автори виявили, що застосування ГІС не тільки сприяє більш ефективному розподілу ресурсів, але й забезпечує значне зниження витрат на паливо та часу, потрібного для збору врожаю. Висновки дослідження підкреслюють потенціал ГІС як інструменту для підвищення прибутковості та конкурентоспроможності в аграрному секторі, зокрема в умовах, де потрібно оптимізувати використання обмежених ресурсів.

Ця кваліфікаційна робота спрямована на вирішення конкретних невирішених проблем: розробка системи моніторингу, оптимізованої для контексту збору

врожаю, яка б враховувала фактори, такі як геолокаційні дані транспортних засобів, швидкість, технічний стан транспортних, тощо. Особливо важливим є впровадження системи, що інтегрує дані в реальному часі з сенсорів і GPS-трекінгу, що дозволить оперативно приймати рішення та оптимізувати маршрути. Саме такий підхід дозволить аграріям України максимально ефективно використовувати транспортні ресурси, знижуючи витрати на логістику і збільшуючи загальну рентабельність виробництва. В умовах постійно зростаючої конкуренції та необхідності адаптації до змін клімату, такі інновації є не тільки доцільними, а й необхідними для забезпечення стійкості і розвитку аграрного сектора в Україні.

Мета роботи полягає у розробці системи моніторингу статусів вантажного транспорту, оптимізованої для потреб агропромислового комплексу під час збору врожаю, що сприятиме підвищенню ефективності та зниженню витрат у логістичних процесах.

Для досягнення поставленої мети були визначені наступні основні задачі:

- вивчення теоретичних основ моніторингу вантажного транспорту в агропромисловому комплексі, включаючи огляд сучасних методів моніторингу, особливостей використання GPS та інших навігаційних систем, та аналіз існуючих систем вантажних перевезень в контексті збору врожаю;

- розробка технічних вимог та проектування архітектури системи моніторингу, зосереджуючись на формулюванні специфікацій, виборі технологічного стеку, проектуванні бізнес-процесів та бази даних, що відповідають унікальним потребам агропромислового збору врожаю;

- реалізація розробленої системи, включаючи програмування контролерів, інтеграцію з базою даних, створення необхідних модулів, а також проведення комплексного тестування для забезпечення стабільності та надійності функціонування системи;

- аналіз отриманих результатів, оцінка ефективності системи моніторингу на практиці та розробка рекомендацій для подальшого удосконалення та можливої адаптації системи в інших секторах логістики.

Об'єктом дослідження є логістичні процеси в агропромисловому комплексі, зокрема процеси транспортування вантажів у період збору врожаю. Ці процеси включають організацію, контроль та управління перевезеннями вантажів, що вирізняються високими вимогами до часу доставки та збереження якості сільськогосподарської продукції.

Предметом дослідження є система моніторингу статусів вантажного транспорту, яка використовується в межах зазначених логістичних процесів.

Для досягнення мети дослідження системи моніторингу статусів вантажного транспорту в процесі збору врожаю використовуються наступні методи дослідження: аналіз літератури, моделювання процесів, експертні оцінки, статистичний аналіз даних та експериментальне тестування.

Практичне значення отриманих результатів полягає у здатності системи ефективно відслідковувати реальний статус транспортних засобів та їх параметри в процесі збору врожаю, забезпечуючи оптимальне розподілення та використання ресурсів. Це дозволяє знижувати час простою транспорту, покращує координацію логістичних операцій і сприяє зменшенню витрат на паливо та ремонт, водночас забезпечуючи своєчасне реагування на зміни в оперативній обстановці.

1 ТЕОРЕТИЧНІ ОСНОВИ МОНІТОРИНГУ ВАНТАЖНОГО ТРАНСПОРТУ В АГРОПРОМИСЛОВОМУ КОМПЛЕКСІ

1.1 Огляд сучасних методів моніторингу вантажного транспорту

Здійснюючи аналіз і синтез інформації, яка надходить з різних джерел і пристроїв моніторингу, можна отримати комплексну картину стану транспортних засобів і зв'язаних з ними процесів. Цей підхід базується на зборі, обробці та інтеграції даних, що стосуються місцеположення транспортних засобів, їхнього стану, рівня навантаження, виконання технічного обслуговування та інших важливих параметрів функціонування.

Аналіз даних дозволяє виявити тенденції, зробити прогнози і виявити можливі проблеми або ризики, пов'язані з експлуатацією транспортних засобів. Синтез же інформації дозволяє побудувати єдину системну картину, яка враховує різноманітні аспекти функціонування транспортних засобів і їхнього взаємодії з оточуючим середовищем. В результаті, здійснення аналізу та синтезу дозволяє забезпечити ефективне управління транспортними потоками, знизити витрати на обслуговування та ремонт, підвищити рівень безпеки та зручності для користувачів транспорту і покращити загальну продуктивність системи транспортного обслуговування [2].

Застосування цих методів дозволяє також адаптувати існуючі технологічні рішення до специфічних умов та вимог агропромислового комплексу, що є особливо важливим в умовах зростаючих вимог до ефективності та екологічності сучасних аграрних виробництв. На рис. 1.1 представлено діаграму розподілу методів моніторингу вантажного транспорту, що демонструє різноманітність підходів та їх застосування в різних сценаріях.



Рис. 1.1 Діаграма розподілу методів моніторингу вантажного транспорту

В останні роки відзначається значне розповсюдження деяких методик, що виявилися ефективними для моніторингу статусів вантажного транспорту. Особливу увагу привертають наступні методики, зокрема:

- GPS-моніторинг [3]. Це один з найпоширеніших методів моніторингу транспорту, який дозволяє в реальному часі відстежувати місцезнаходження транспортних засобів. GPS-моніторинг не тільки спрощує логістику, але й допомагає в аналізі маршрутів, оптимізації швидкості руху та плануванні завдань;
- телематика. Цей метод включає в себе використання телекомунікацій та інформатики для передачі інформації про стан транспортного засобу. Сучасні телематичні системи можуть збирати детальні дані про швидкість руху, використання палива, технічний стан автомобіля та поведінку водія;
- RFID технології. Використання радіочастотної ідентифікації для контролю за вантажами та транспортними засобами стало ефективним рішенням для автоматизації складських операцій і зменшення втрат часу на пунктах перевірки та видачі вантажів; [4]

– інтегровані системи управління логістикою (ERP). Сучасні ERP-системи інтегрують моніторинг транспорту з іншими бізнес-процесами, такими як запаси, замовлення, фінанси, що дозволяє комплексно управляти логістичними ланцюгами, покращуючи їхню прозорість та ефективність; [5]

– використання дронів. Останнім часом розвиток технологій дозволив використовувати безпілотні літальні апарати для моніторингу великих аграрних та логістичних об'єктів. Дрони здатні швидко збирати дані про стан вантажів та вантажного транспорту, особливо на великих відкритих просторах.

За допомогою цих методик значно полегшується ведення логістики та управління транспортними потоками. Вони дозволяють здійснювати точну і оперативну інформаційну підтримку при плануванні та моніторингу руху вантажних транспортних засобів. Застосування таких передових технологій, як GPS-моніторинг, телематика, RFID технології, інтегровані системи управління логістикою та використання дронів, сприяє підвищенню ефективності та зниженню витрат у логістичних операціях.

1.2 Особливості використання GPS та інших навігаційних систем у логістиці вантажних перевезень

Використання GPS та інших навігаційних систем в логістиці вантажних перевезень є фундаментальним елементом для забезпечення ефективного управління логістичними потоками. Ці системи дозволяють здійснювати постійний зв'язок між вантажними транспортними засобами та диспетчерськими центрами, значно підвищуючи ефективність перевезень. Вони забезпечують точне та оперативне визначення географічного положення транспорту, що є вирішальним для ефективного планування маршрутів, своєчасного реагування на аварійні ситуації, дотримання графіків доставки та оптимізації логістичних процесів. Навігаційні системи включають різні технології, які дозволяють покращити моніторинг і управління вантажними потоками, враховуючи актуальні дорожні умови та забезпечуючи водіям інформацію, необхідну для безпечного та

ефективного пересування [6]. На рис. 1.2 представлено діаграму розподілу навігаційних систем у логістиці вантажних перевезень, що відображає широкий спектр доступних технологічних рішень та їх застосування в різних аспектах транспортного процесу.



Рис. 1.2 Діаграма розподілу навігаційних систем у логістиці перевезень

Основні аспекти використання навігаційних систем у логістиці перевезень включають:

- трекінг у реальному часі. Це одна з основних переваг, яка дозволяє операторам відстежувати місцезнаходження кожного транспортного засобу в будь-який момент часу, що поліпшує координацію та планування логістики;
- маршрутизація. Навігаційні системи дозволяють автоматично планувати оптимальні маршрути, враховуючи багато факторів, включаючи відстань, час у дорозі, поточні дорожні умови і навіть погодні умови. Це допомагає мінімізувати витрати на паливо та час доставки;
- геозони та оповіщення. Визначення геозон дозволяє налаштовувати автоматичні сповіщення, коли транспортні засоби входять чи виходять з певних зон. Це корисно для контролю за доставкою в специфічні зони та для забезпечення безпеки вантажу.

– звітність та аналітика [7]. Сучасні системи забезпечують детальні звіти про кожну поїздку, включаючи використання палива, швидкість руху, простої та інші ключові показники. Ці дані є незамінними для аналізу ефективності логістичних операцій та для ухвалення рішень щодо подальших покращень.

Використання навігаційних систем у логістиці вантажних перевезень також вимагає звернення уваги на питання конфіденційності та безпеки даних. Важливо забезпечувати належний захист інформації, що передається та зберігається системами, для запобігання її несанкціонованому доступу чи втраті.

1.3 Аналіз систем вантажних перевезень в процесі збору врожаю

Ефективність систем вантажних перевезень під час збору врожаю є критичним фактором для агропромислового сектору, оскільки затримки у транспортуванні можуть призвести до значних фінансових втрат і погіршення якості сільськогосподарської продукції.

Samsara є інноваційним рішенням у сфері IoT (Інтернет речей), яке надає комплексні послуги для моніторингу та аналізу даних у вантажних перевезеннях (рис. 1.3). Застосунок розроблений для поліпшення ефективності та безпеки транспортних операцій через використання даних з транспортних засобів і датчиків у реальному часі.

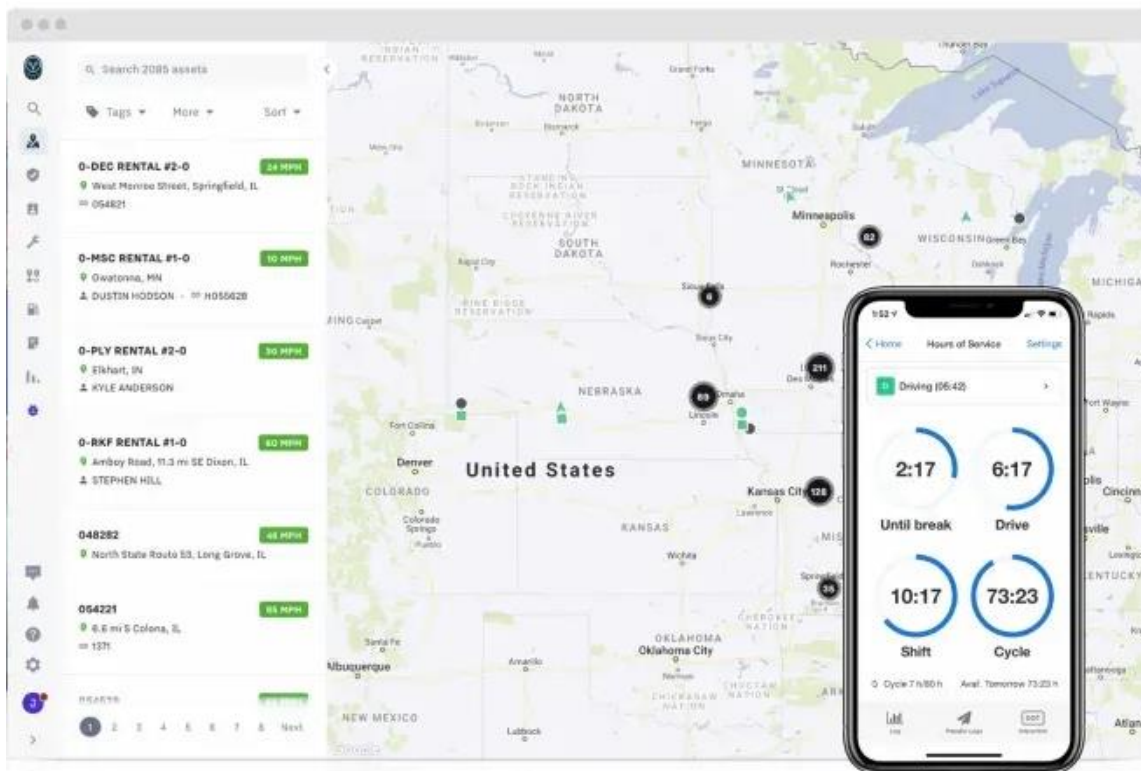


Рис. 1.3 Приклад інтерфейсу системи «Samsara»

Призначення Samsara полягає у наданні водіям, диспетчерам і керівникам логістичних підрозділів інструментів для оптимізації маршрутів, моніторингу стану транспорту, контролю за виконанням норм безпеки, а також збору і аналізу даних для прийняття обґрунтованих управлінських рішень [8].

Архітектура Samsara включає кілька ключових компонентів: обладнання для збору даних, яке встановлюється на транспортні засоби (наприклад, GPS-трекери, сенсори температури, відеокамери); хмарна платформа, що обробляє і зберігає зібрані дані; мобільні та веб-додатки, які забезпечують користувачам доступ до аналітики і управлінських інструментів. Ця інтегрована система дозволяє здійснювати моніторинг у реальному часі і автоматично реагувати на критичні ситуації, що значно підвищує безпеку і знижує операційні ризики.

Переваги застосунку Samsara:

- покращення безпеки. Samsara використовує відеонагляд та сенсори для моніторингу поведінки водіїв та стану техніки, що допомагає запобігти аварійним ситуаціям та покращує загальні стандарти безпеки;

- оптимізація логістики. З допомогою GPS-трекінгу та аналітики, Samsara дозволяє планувати маршрути ефективніше, враховуючи різні фактори, такі як трафік та погодні умови, зменшуючи тим самим час доставки та витрати на паливо;

- збір та аналіз даних. Платформа забезпечує збір великих обсягів даних з різних джерел, що дозволяє проводити глибокий аналіз ефективності перевезень, стану обладнання та інших важливих параметрів;

- інтеграція з іншими системами. Samsara може інтегруватися з іншими бізнес-системами, що забезпечує цілісний підхід до управління транспортом і логістикою в одному інтерфейсі.

Недоліки застосунку Samsara:

- висока вартість. Впровадження та обслуговування системи може бути коштовним, оскільки воно включає витрати на апаратне забезпечення, ліцензії на програмне забезпечення та потенційно, витрати на навчання персоналу;

- залежність від інтернет-з'єднання. Ефективність системи безпосередньо залежить від стабільності та швидкості інтернет-з'єднання, оскільки всі дані обробляються в хмарі;

- комплексність системи. Для деяких користувачів, особливо тих, хто не має технічного досвіду, система може здатися складною в налаштуванні та експлуатації;

- проблеми з приватністю даних. Збір та аналіз великої кількості даних може викликати питання щодо приватності та безпеки інформації, особливо в контексті відеомоніторингу та слідкування за поведінкою водіїв [9].

Отже, Samsara є високотехнологічним рішенням для моніторингу вантажного транспорту, що надає значні переваги для підвищення ефективності, безпеки та оптимізації логістичних процесів. Використання сучасних GPS-технологій та інших сенсорів дозволяє здійснювати точний моніторинг та аналіз даних в реальному часі, що сприяє зниженню витрат і покращенню робочих процесів. Водночас, Samsara має деякі недоліки, як-от високу вартість і потребу в стабільному інтернет-з'єднанні, що можуть становити виклики для деяких

організацій. Незважаючи на це, загальна інтеграція і можливість адаптації Samsara роблять її важливим інструментом для сучасного управління вантажними перевезеннями, особливо в умовах зростаючих вимог до точності і ефективності логістичних операцій.

Teletrac Navman є одним з провідних рішень для управління автопарком та відстеження вантажного транспорту, яке використовує передові технології GPS та аналітичні інструменти для моніторингу та оптимізації транспортних операцій (рис. 1.4). Цей застосунок призначений для забезпечення більшої видимості, керування та підвищення ефективності роботи вантажних автопарків через комплексний аналіз даних, що допомагає компаніям підвищувати продуктивність та знижувати витрати [10].

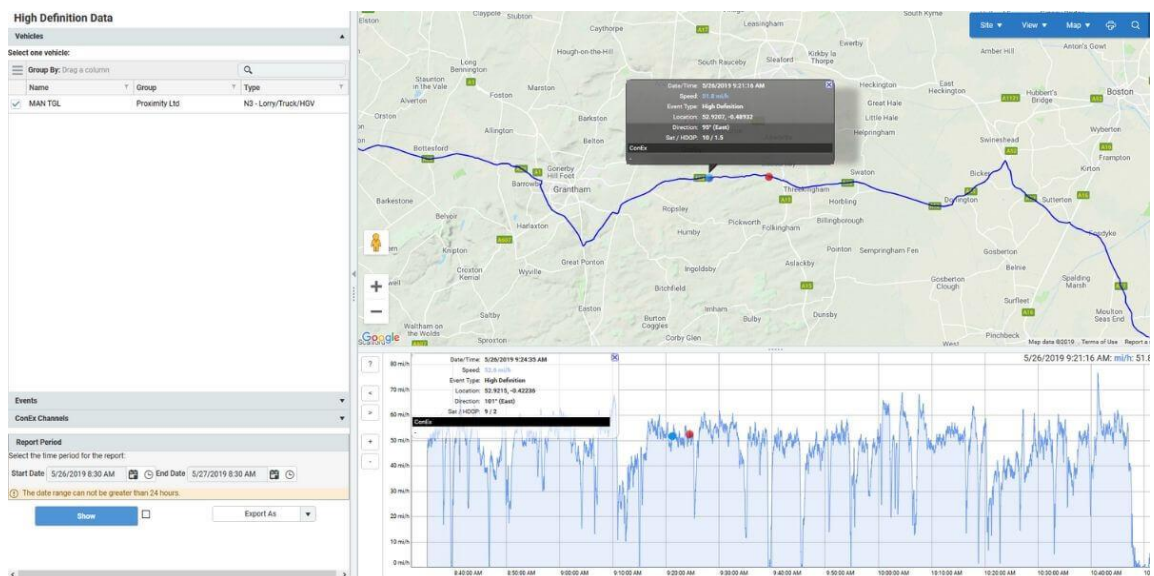


Рис. 1.4 Приклад інтерфейсу системи «Teletrac Navman»

Архітектура Teletrac Navman включає в себе наступні ключові елементи:

- апаратне обладнання. Встановлюється на транспортні засоби для збору даних про їхнє розташування, швидкість, стиль водіння та інші важливі параметри;
- програмне забезпечення. Потужна облачна платформа збирає та аналізує дані з апаратного обладнання, що дозволяє користувачам відстежувати та аналізувати різні аспекти роботи транспортних засобів в режимі реального часу;

– інтерфейс користувача. Дружній до користувача веб- та мобільний інтерфейси надають доступ до інструментів звітності, моніторингу та планування, що допомагає ухвалювати обґрунтовані управлінські рішення.

Teletrac Navman дозволяє компаніям не тільки підвищувати контроль над своїм автопарком, але й оптимізувати робочі процеси та забезпечувати високий рівень обслуговування клієнтів за рахунок точного відстеження та аналізу виконання транспортних завдань [11].

Переваги застосунку Teletrac Navman:

– підвищена точність моніторингу. Teletrac Navman використовує передові GPS-технології, які забезпечують високу точність у відстеженні розташування та поведінки транспортних засобів, що дозволяє компаніям точно контролювати свій автопарк;

– збільшена продуктивність. Застосунок допомагає оптимізувати маршрути та графіки, знижуючи простой та час в дорозі, що призводить до збільшення загальної продуктивності перевезень;

– покращення безпеки. Функції моніторингу стилю водіння та відеонагляд допомагають підвищити рівень безпеки, виявляючи ризиковану поведінку водіїв та дозволяючи своєчасно реагувати на потенційні небезпеки;

– обширні можливості звітності. Teletrac Navman пропонує розширені аналітичні можливості та звітність, що дозволяє керівництву компанії отримувати глибокі інсайти для планування та прийняття рішень.

Недоліки застосунку Teletrac Navman:

– складність інтеграції. Необхідність інтеграції з іншими корпоративними системами може бути складною та часом дорогою, особливо для великих компаній з уже наявними складними ІТ-інфраструктурами;

– висока вартість. Вартість впровадження та підтримки системи може бути високою, з огляду на необхідність придбання обладнання та оплати ліцензійних внесків;

– залежність від мобільного покриття. Ефективність системи залежить від наявності стабільного мобільного зв'язку, що може стати проблемою в віддалених або погано покритих місцях;

– приватність даних [12]. Збір великих обсягів даних про розташування та поведінку водіїв може викликати питання, пов'язані з приватністю та захистом персональних даних, що вимагає ретельного дотримання правил конфіденційності.

Отже, Teletrac Navman є ефективним рішенням для управління автопарком, що використовує передові GPS-технології та аналітичні інструменти для моніторингу та оптимізації логістичних процесів. Завдяки точному відстеженню розташування, аналізу поведінки водіїв та покращенню безпеки, цей застосунок допомагає підвищувати продуктивність та знижувати витрати. Водночас, висока вартість, потреба у стабільному інтернеті та питання приватності є важливими факторами для розгляду. Загалом, Teletrac Navman є цінним інструментом для компаній, які прагнуть модернізувати свої логістичні операції і забезпечити високий рівень управління транспортними ресурсами.

Mix Telematics є глобальним провайдером рішень для управління автопарком, забезпечення безпеки транспортних засобів та оптимізації робочих процесів через телематичні технології. Застосунок призначений для моніторингу різноманітних аспектів діяльності автопарком, включаючи поведінку водіїв, витрати на паливо, стан транспортних засобів та відповідність нормативним вимогам. Це рішення допомагає забезпечити дотримання законодавчих норм, покращити безпеку та знизити експлуатаційні витрати [13].

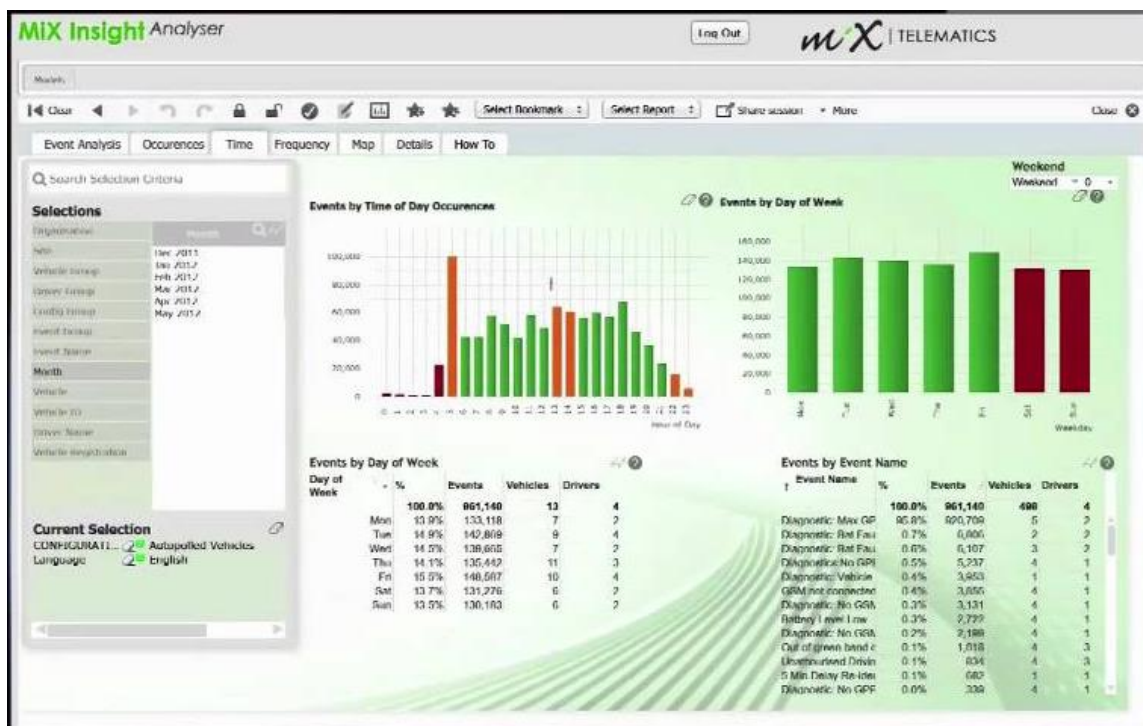


Рис. 1.5 Приклад інтерфейсу системи «Mix Telematics»

Архітектура Mix Telematics орієнтована на збір та аналіз великих обсягів даних з транспортних засобів у реальному часі. Система включає в себе наступні ключові компоненти:

- апаратне обладнання. Встановлюється на транспортні засоби для збору даних, таких як GPS-позиціонування, датчики споживання палива, температури та інші оперативні параметри;
- програмне забезпечення. Хмарна платформа Mix Telematics обробляє зібрані дані, надаючи користувачам інструменти для аналізу, звітності та управління автопарком через веб-інтерфейс та мобільні додатки;
- інтерфейс користувача. Забезпечує інтуїтивно зрозумілі інструменти для відстеження, моніторингу та управління даними, що дозволяють користувачам легко доступати до інформації та управляти своїм автопарком з будь-якої точки світу.

Ця інтегрована система допомагає покращити ефективність управління транспортом, знизити ризики та оптимізувати загальну продуктивність [14].

Переваги застосунку Mix Telematics:

- збільшення безпеки. Mix Telematics включає функції моніторингу поведінки водіїв, що дозволяє ідентифікувати і коригувати небезпечні звички водіння, знижуючи ризик дорожніх інцидентів;
- економія палива. Завдяки детальному моніторингу споживання палива та оптимізації маршрутів, застосунок допомагає значно знизити витрати на паливо;
- дотримання нормативних вимог. Система забезпечує дотримання законодавчих вимог, таких як час водіння, перерви та інші регуляторні обмеження, що є критично важливим для логістичних компаній;
- покращене управління парком транспортних засобів. Автоматизовані звіти та аналітика дозволяють керівництву компанії ефективно управляти автопарком, планувати технічне обслуговування та приймати обґрунтовані рішення на основі даних [15].

Недоліки застосунку Mix Telematics:

- складність впровадження. Необхідність встановлення спеціалізованого обладнання та інтеграції з існуючими системами може виявитися складною та часом затратною для деяких компаній;
- залежність від мобільного покриття. Ефективність системи залежить від стабільного інтернет-з'єднання, що може бути проблематичним у віддалених або слабо покритих місцях;
- висока вартість. Витрати на ліцензування програмного забезпечення, обслуговування та оновлення системи можуть бути значними, особливо для малого та середнього бізнесу;
- проблеми з приватністю та безпекою даних. Збір та аналіз великої кількості даних про місцезнаходження та поведінку водіїв вимагає високого рівня захисту даних та може викликати питання, пов'язані з приватністю персональної інформації [16].

Отже, Mix Telematics є потужним інструментом для управління автопарком, який допомагає підвищити безпеку, знизити витрати на паливо та забезпечити дотримання нормативних вимог через передові технології телематики та GPS-моніторингу. Система пропонує значні переваги у плануванні маршрутів та аналізі

даних, дозволяючи компаніям ефективно управляти своїми транспортними ресурсами. Однак, високі витрати на впровадження, залежність від мобільного покриття та питання, пов'язані з приватністю даних, можуть стати бар'єром для деяких організацій. Загалом, Mix Telematics залишається важливим рішенням для тих, хто шукає інтегроване рішення для управління автопарком з широким спектром аналітичних можливостей.

Ефективне планування та оптимізація логістичних маршрутів з використанням сучасних технологій, таких як GPS-навігація, сприяють зниженню витрат на паливо та часу перевезень, що безпосередньо впливає на загальну рентабельність виробництва. Розгляд погодних умов як чинника, що впливає на логістику, дозволяє адаптувати транспортні операції до складних екологічних умов, забезпечуючи безперебійність і надійність поставок.

1.4 Постановка задачі

Основною задачею дослідження є розробка та імплементація застосунку, який буде використовувати технологію ML.NET для моніторингу статусів вантажного транспорту в реальному часі під час збору врожаю. Застосунок має на меті інтегрувати та аналізувати дані, отримані з GPS трекерів та датчиків стану транспортних засобів, а також інших відповідних джерел. Це забезпечить високу точність та актуальність інформації про переміщення та стан вантажів.

Задля забезпечення цієї мети, дослідження передбачає наступні кроки:

– інтеграція джерел даних. Для забезпечення ефективного моніторингу статусів вантажного транспорту під час збору врожаю, важливим аспектом є інтеграція та аналіз даних з різних джерел. У рамках нашої системи використовуються дані, отримані з GPS-трекерів, встановлених на транспортних засобах, а також з датчиків, які надають інформацію про місцезнаходження, швидкість, стан та навантаження кожного засобу. Ці дані збираються в реальному часі, що дозволяє забезпечити актуальність і точність інформації, необхідної для оптимального управління логістичними процесами.

- розробка програмного забезпечення. Створення додатку на мові програмування C# з використанням бази даних MS SQL Server для зберігання, обробки та аналізу зібраних даних. Додаток буде розроблений таким чином, щоб користувачі могли легко взаємодіяти з системою та отримувати потрібну інформацію.

- застосування алгоритмів машинного навчання: Використання алгоритму Бroyдена-Флетчера-Гольдфарба-Шанно (BFGS), вбудованого у ML.NET, для аналізу зібраних даних з метою прогнозування потенційних збоїв у логістичних операціях. Це дозволить попереджати можливі затримки чи інші проблеми, пов'язані з транспортуванням врожаю;

- тестування та валідація. Проведення тестування системи для забезпечення її надійності, точності та ефективності. Оцінка функціональності застосунку на різних етапах збору врожаю та транспортування.

Завдяки реалізації цього дослідження, кінцевою метою є не тільки підвищення ефективності логістичних процесів, але й мінімізація затримок і витрат, пов'язаних з транспортуванням врожаю. Такий підхід сприятиме зростанню продуктивності та зниженню оперативних ризиків.

У рамках даного розділу було проведено глибоке вивчення теоретичних основ моніторингу вантажного транспорту в агропромисловому комплексі, що охоплює широкий спектр сучасних методів та технологій. Було розглянуто ключові методи моніторингу, такі як GPS-моніторинг, телематика, RFID технології, інтегровані системи управління логістики та використання дронів, кожен з яких відіграє важливу роль у забезпеченні ефективності логістичних процесів.

Також було детально проаналізовано особливості застосування GPS та інших навігаційних систем, що дозволяють покращувати трекінг у реальному часі, оптимізувати маршрутизацію, налаштовувати геозони та оповіщення та формувати детальні звітність та аналітику.

За допомогою аналізу наявних систем моніторингу, таких як Samsara, Teletrac Navman та Mix Telematics, були виявлені їх переваги та недоліки, що забезпечило цінні відомості для подальшої розробки нашого застосунку. Цей аналіз вказав на необхідність інтеграції ефективних інструментів моніторингу та аналізу, які могли б оптимізувати процеси і зменшити потенційні ризики.

Закінченням розділу стала постановка задачі на розробку застосунку на базі C#, MS SQL Server та ML.NET, який інтегрує передові методи машинного навчання для прогнозування та оптимізації логістичних операцій. Отримані результати та аналіз є фундаментом для наступних розділів, де буде здійснено проектування і розробку, спрямованих на підвищення ефективності вантажних перевезень в агропромисловому секторі.

2 АНАЛІЗ ТА ПРОЕКТУВАННЯ СИСТЕМИ МОНІТОРИНГУ СТАТУСІВ ВАНТАЖНОГО ТРАНСПОРТУ В ПРОЦЕСІ ЗБОРУ ВРОЖАЮ

2.1 Формулювання вимог до системи моніторингу вантажного транспорту

Визначення чітких і детальних характеристик та функцій продукту є критичним для забезпечення його відповідності очікуванням клієнтів та користувачів. Розглянемо основні категорії вимог, на які варто звернути увагу при аналізі.

Основа програмної системи складають функціональні вимоги. Вони задають рамки для задач, які система має виконувати. Ці вимоги уточнюють, які дії система має здійснювати в різних ситуаціях, як вона повинна реагувати на дії користувачів та які результати повинна надавати [17]. Враховуючи важливість цих вимог, у табл. 2.1 представлено кожен із категорій вимог, що дає змогу краще зрозуміти їх роль у процесі розробки системи.

Таблиця 2.1

Функціональні вимоги до застосунку

Вимоги	Опис
REQ-1	Вимога до впровадження процесу реєстрації користувачів для доступу до функціоналу програми
REQ-2	Реалізація можливості ведення логу подій у системі
REQ-3	Забезпечення функцій для введення та оновлення даних про користувачів, транспортні засоби та інші об'єкти
REQ-4	Надання можливостей для обробки та навчання прогнозних моделей стану вантажного транспорту під час збирання урожаю
REQ-5	Запровадження функції моніторингу стану транспортних засобів в режимі реального часу та їх прогнозування

Нефункціональні вимоги критично впливають на якість програмного продукту. Вони стосуються аспектів системи, які не прямо пов'язані з її функціональністю. Ці вимоги важливі для забезпечення надійної роботи системи та відповідності очікуванням користувачів, як описано у табл. 2.2.

Список нефункціональних вимог до застосунку

Вимоги	Опис
REQ-6	Вимагається висока продуктивність та ефективність у використанні ресурсів системи в додатку
REQ-7	Запевнення стабільності та надійності програми, включно з можливістю відновлення після збоїв, захистом даних від втрати та обмеженням несанкціонованого доступу
REQ-8	Гарантія конфіденційності та захист персональних даних користувачів з протидією неавторизованому доступу
REQ-9	Компатибельність додатку з різноманітними операційними системами
REQ-10	Адаптивність програми до майбутнього розширення та модифікацій
REQ-11	Інтерфейс, що вирізняється інтуїтивністю, зручністю використання та ергономічністю

Варіант використання є визначенням типу взаємодії між актором та системою, де основною метою є досягнення певної цілі, яку прагне реалізувати актор у рамках цієї системи. Термін «варіант використання» описує специфічний сценарій або набір обставин, в яких актор залучається до взаємодії з системою для виконання визначених завдань чи досягнення бажаних результатів. Це може включати широкий спектр дій, таких як введення даних, обробка запитів, чи навіть управління налаштуваннями.

Табл. 2.3 представляє деталізований перелік різноманітних варіантів використання у системі, де кожен запис включає опис конкретних сценаріїв взаємодії. Ці сценарії можуть виконуватися різними акторами системи, наприклад, адміністраторами, звичайними користувачами або автоматизованими процесами. Кожний варіант використання забезпечує унікальний шлях для актора до взаємодії з системою, під час якої він може виконувати завдання на кшталт оновлення бази даних транспортних засобів, конфігурації параметрів тренування моделей прогнозування, а також інші спеціалізовані дії, що сприяють оптимізації процесів у межах організації.

Опис варіантів використання застосунку

Варіант	Ім'я	Опис
1	2	3
UC1	Реєстрація користувача	Дозволяє зареєструвати нового користувача в системі
UC2	Авторизація користувача	Уможлиблює користувачу виконати вхід у систему для верифікації особистості
UC3	Перегляд списку користувачів	Надає можливість системному адміністратору ознайомитися зі списком всіх зареєстрованих користувачів
UC4	Створення користувача	Дозволяє системному адміністратору створювати профілі нових користувачів
UC5	Модифікація даних користувача	Дозволяє редагувати інформацію про користувачів у системі
UC6	Вилучення користувача	Уможлиблює видалення профілів користувачів з системи
UC7	Перегляд списку транспортних засобів	Дозволяє переглядати інформацію про всі транспортні засоби в системі
UC8	Додавання ТЗ	Забезпечує можливість реєстрації нових транспортних засобів у системі
UC9	Модифікація ТЗ	Уможлиблює оновлення даних про транспортні засоби
UC10	Вилучення ТЗ	Дозволяє системному адміністратору видаляти транспортні засоби з бази даних системи
UC11	Перегляд списку категорій моделей	Уможлиблює огляд усіх доступних категорій моделей в системі
UC12	Додавання категорії	Забезпечує можливість реєстрації нових категорій
UC13	Модифікація категорії	Дозволяє оновлювати інформацію про специфічну категорію моделі машинного навчання

UC14	Вилучення категорії	Дає можливість елімінувати існуючі категорії з системи
UC15	Перегляд списку моделей	Дозволяє доступ до перегляду всіх доступних моделей для прогнозування статусів вантажного транспорту під час збору врожаю
UC16	Додавання моделі	Уможлиблює створення та тренування нових моделей прогнозування
UC17	Вилучення моделі	Дозволяє видалити треновані моделі з системи
UC18	Моніторинг ТЗ	Дозволяє стежити за роботою транспортних засобів та аналізувати їх статуси у процесі врожаю
UC19	Вивід подій	Забезпечує доступ до інформації про всі події, які мали місце в системі, для подальшого аналізу та контролю

На основі вимог було розроблено use-case діаграму прецедентів, яка представлена на рис. 2.1.



Рис. 2.1 Діаграма use-case

На діаграмі варіантів використання системи «Моніторинг статусів вантажного транспорту» представлено зв'язки між акторами та ключовими функціями системи. Кожен варіант використання, від UC1 до UC19, демонструє специфічний спосіб взаємодії користувачів з системою, охоплюючи такі аспекти як реєстрація, авторизація, керування профілями користувачів та транспортними засобами, а також моніторинг. Стрілки на діаграмі вказують на залежності та порядок дій, які сприяють логічному та інтуїтивно зрозумілому взаємодію користувачів з системою, забезпечуючи ефективність робочих процесів.

2.2 Вибір технологічного стеку для розробки системи моніторингу

Вибір мови програмування є ключовим етапом у розробці програмного продукту і впливає на його продуктивність, надійність, масштабованість та інші аспекти. Кожна мова має свої переваги та обмеження, і вибір варіанту повинен бути обґрунтованим з урахуванням специфіки проекту, вимог до швидкодії, доступних ресурсів, відомостей та навичок команди розробників, а також інших факторів.

Python - це високорівнева, інтерпретована мова програмування з простим синтаксисом, що дозволяє розробникам писати код швидше та ефективніше [18]. Вона широко використовується для веб-розробки, аналізу даних, штучного інтелекту та інших областей завдяки великій кількості наявних бібліотек.

Java - це об'єктно-орієнтована мова програмування, яка відома своєю переносимістю та надійністю завдяки використанню віртуальної машини Java (JVM) [19]. Вона широко використовується для створення веб-додатків, мобільних додатків (Android), корпоративних систем та багатьох інших застосувань.

C# (C-Sharp) - це об'єктно-орієнтована мова програмування, розроблена компанією Microsoft, спрямована на створення програм для платформи .NET [20]. Вона популярна для розробки Windows-додатків, ігор, веб-сайтів та різноманітних додатків для платформи Microsoft.

У табл. 2.4 проведено порівняльний аналіз функціональних можливостей розглянутих мов програмування.

Порівняльний аналіз мов програмування

Характеристика	C++	Java	C#
Платформа	Багато-платформенність через компіляцію кожної платформи	Кросплатформенність завдяки JVM	Кросплатформенність через .NET Framework і .NET Core
Управління пам'яттю	Ручне управління	Автоматичний збір сміття	Автоматичний збір сміття
Стандартні бібліотеки	Стандартна бібліотека шаблонів (STL)	Обширна стандартна бібліотека	Велика кількість високорівневих API і широкий доступ до .NET бібліотек
Підтримка паралельного виконання	Ручне управління нитками	Вбудована підтримка ниток	Вбудовані засоби для асинхронного програмування (async/await)
Безпека типів	Частково (не безпечний доступ до пам'яті)	Строга типізація без прямого доступу до пам'яті	Строга типізація з безпечним доступом до пам'яті
Інтегрованість з Windows	Обмежена	Обмежена	Глибока інтеграція з Windows, використання .NET і підтримка COM

Вибір мови програмування C# для розробки системи можна обґрунтувати кількома ключовими аспектами, відображеними у таблиці порівняння:

- кросплатформенність через .NET. C# підтримується .NET Framework для Windows та .NET Core, який є кросплатформенним і дозволяє запускати додатки на Linux та macOS. Це забезпечує гнучкість у розгортанні додатків на різних платформах;

- автоматичний збір сміття. Управління пам'яттю в C# відбувається автоматично, що знижує ризики пов'язані з витокami пам'яті та іншими помилками управління ресурсами, що є типовими для мов з ручним управлінням пам'яттю, як C++;

- вбудовані засоби для асинхронного програмування. C# має потужні засоби для асинхронного програмування через ключові слова `async` та `await`, що дозволяють розробникам легко писати неблокуючий код, критично важливий для сучасних веб-додатків та додатків реального часу;
- інтегрованість з Windows та .NET бібліотеками. Для систем, які цілеспрямовано розробляються під Windows, C# пропонує виняткову інтеграцію з операційною системою та корпоративними середовищами Microsoft, включно з доступом до широкого спектру API і сервісів Microsoft;
- Visual Studio та інструменти розробки. C# тісно інтегрований з Visual Studio, одним з найпотужніших IDE, який забезпечує розширені можливості для дизайну інтерфейсу, дебагінгу, тестування та розгортання програм.

Ці переваги роблять C# ідеальним вибором для розробки різноманітних систем, особливо коли необхідна висока продуктивність, швидка розробка та легкість підтримки.

Вибір підходящої системи управління базами даних (СУБД) є критичним аспектом у процесі розробки програмного забезпечення. Різноманітність доступних на ринку СУБД, кожна з яких пропонує унікальний набір функцій.

Microsoft SQL Server - це реляційна система управління базами даних, розроблена компанією Microsoft, яка надає широкий спектр функцій для керування даними, забезпечуючи високу швидкодію та надійність [21]. Вона часто використовується у корпоративних середовищах для зберігання, обробки та аналізу великих обсягів даних.

MySQL - це відкрита реляційна система управління базами даних, розроблена та підтримувана компанією Oracle Corporation [22]. Вона відома своєю швидкодією, простотою використання та безкоштовними версіями, які часто використовуються для веб-розробки та маленьких проектів.

Oracle Database - це потужна реляційна система управління базами даних, розроблена компанією Oracle Corporation, яка відома своєю масштабованістю, надійністю та багатofункціональністю [23]. Вона використовується у великих корпоративних середовищах для зберігання та обробки великих обсягів даних.

У табл. 2.5 представлено порівняльний аналіз цих СУБД, що допоможе вирішити проблему вибору оптимальної системи для зберігання та обробки даних.

Таблиця 2.5

Порівняльний аналіз СУБД

Характеристика	MS SQL Server	MySQL	Oracle
Підтримка транзакцій	Повна підтримка транзакцій	Повна підтримка транзакцій	Повна підтримка транзакцій
Інтегрованість з інструментами	Висока інтегрованість з Microsoft продуктами	Обмежена інтегрованість з іншими інструментами	Хороша інтеграція з Oracle продуктами
Підтримка бізнес-аналітики	Вбудовані BI інструменти (SSAS, SSRS)	Обмежена підтримка BI інструментів	Вбудовані BI інструменти (Oracle BI)
Вартість	Залежить від версії та ліцензування	Здебільшого безкоштовний, платні додаткові опції	Висока, особливо для великих компаній
Підтримка ОС	Windows, Linux (з SQL Server 2017)	Багатоплатформеність	Багатоплатформеність
Адміністрування	Зручне адміністрування через SQL Server Management Studio	Зазвичай вимагає більше налаштувань	Складне адміністрування

Обравши MS SQL Server для розробки системи, можна скористатися низкою значущих переваг:

- інтегрованість з Microsoft продуктами. MS SQL Server ідеально підходить для систем, які вже використовують інші продукти Microsoft, такі як Windows Server, Microsoft Azure, або Visual Studio. Ця сумісність сприяє легшій інтеграції та управлінню;
- підтримка бізнес-аналітики. Завдяки вбудованим інструментам аналізу даних, таким як SQL Server Analysis Services (SSAS) та Reporting Services (SSRS), система може виконувати складний аналіз даних і звітування без потреби інтеграції сторонніх інструментів;

- ефективне адміністрування. SQL Server Management Studio (SSMS) надає потужні засоби для управління, моніторингу та оптимізації баз даних, забезпечуючи адміністраторам легкий доступ до інструментів для ефективного керування базою даних;

- масштабованість та гнучкість. MS SQL Server підтримує як вертикальне, так і горизонтальне масштабування, дозволяючи системі ефективно зростати та адаптуватися до збільшення обсягів даних та користувачів;

- корпоративна підтримка та спільнота. Широка спільнота та професійна підтримка від Microsoft забезпечують високий рівень допомоги та ресурсів для розробників, що сприяє швидкому вирішенню проблем та оптимізації системи.

Всі ці аспекти роблять MS SQL Server відмінним вибором для комплексних систем, які вимагають надійності, інтегрованості з продуктами Microsoft, та високого рівня бізнес-аналітики.

2.3 Проєктування основних бізнес-процесів системи

Задача проєктування основних бізнес-процесів системи є важливим етапом у розробці інформаційних систем, що спрямовані на підтримку діяльності організації. Процес проєктування бізнес-процесів передбачає визначення потреб користувачів, ідентифікацію ключових етапів та взаємозв'язків між ними, а також встановлення оптимальних стратегій виконання завдань.

На рис. 2.2 представлена діаграма активності ілюструє процес авторизації користувача у системі. Вона починається з ініціалізації компонентів, після чого завантажуються дані користувача. Коли користувач натискає кнопку відправки, система перевіряє, чи дані введені коректно. Якщо дані правильні, то перевіряється, чи існує користувач. У разі успішної перевірки, система ініціює сесію користувача, реєструє вхід у систему, показує головне вікно, а потім реєструє вихід з системи. Якщо користувач не знайдений, система показує повідомлення про помилку входу. У разі, якщо дані введені неправильно, показуються помилки валідації. Наприкінці, форма авторизації закривається.

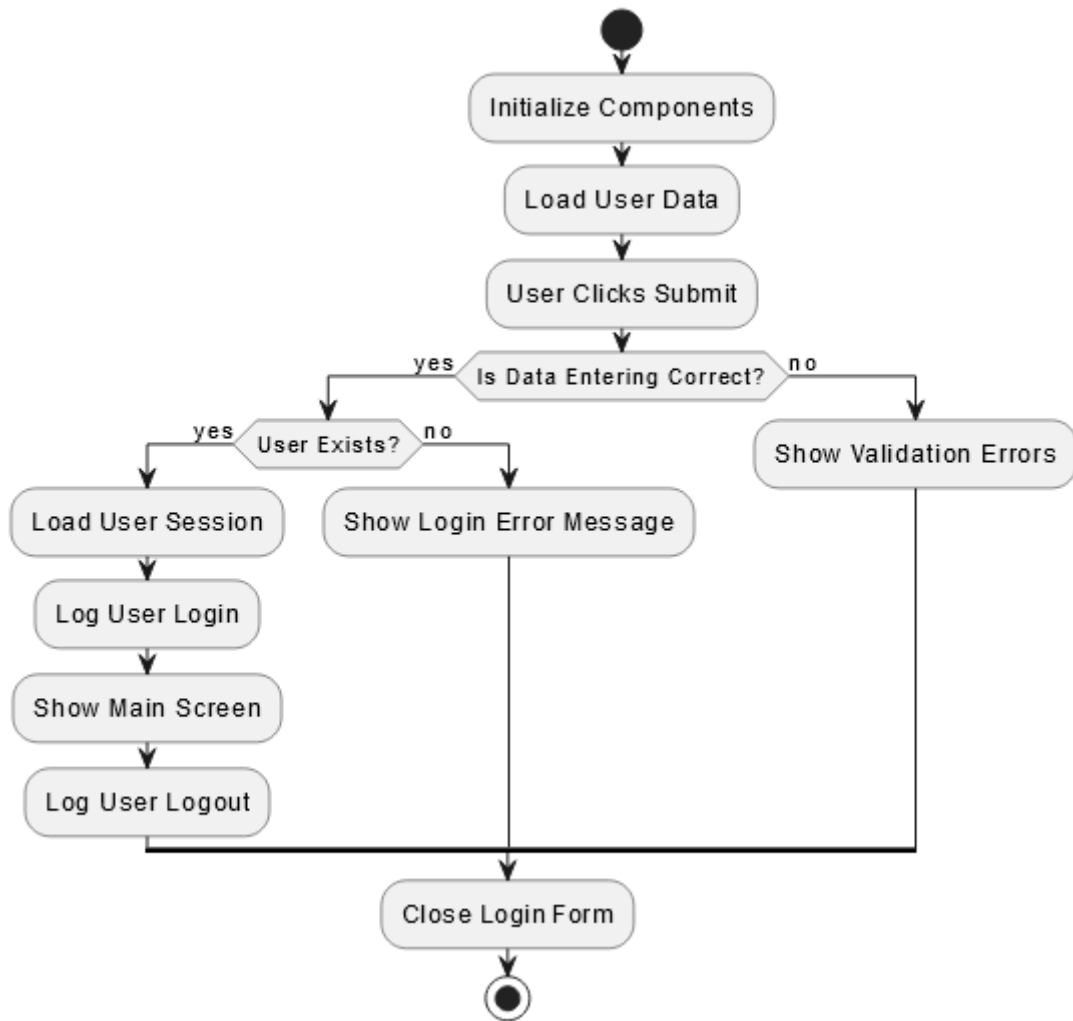


Рис. 2.2 Процес авторизації користувача у системі

Діаграма активності на рис. 2.3 ілюструє процес навчання моделі для моніторингу та прогнозування статусів вантажного транспорту під час збору врожаю. Вона починається з ініціалізації компонентів та вибору сценарію. Далі користувач відкриває файловий діалог для вибору файлу даних. Якщо файл обрано, дані завантажуються з файлу, після чого відбувається їх попередня обробка. Модель тренується за допомогою обраного алгоритму та оцінюється. Результати оцінки моделі відображаються на екрані. У разі успішного навчання моделі, вона зберігається, а подія навчання реєструється в логах. У разі невдачі виводиться повідомлення про помилку. Завершується процес закриттям форми.

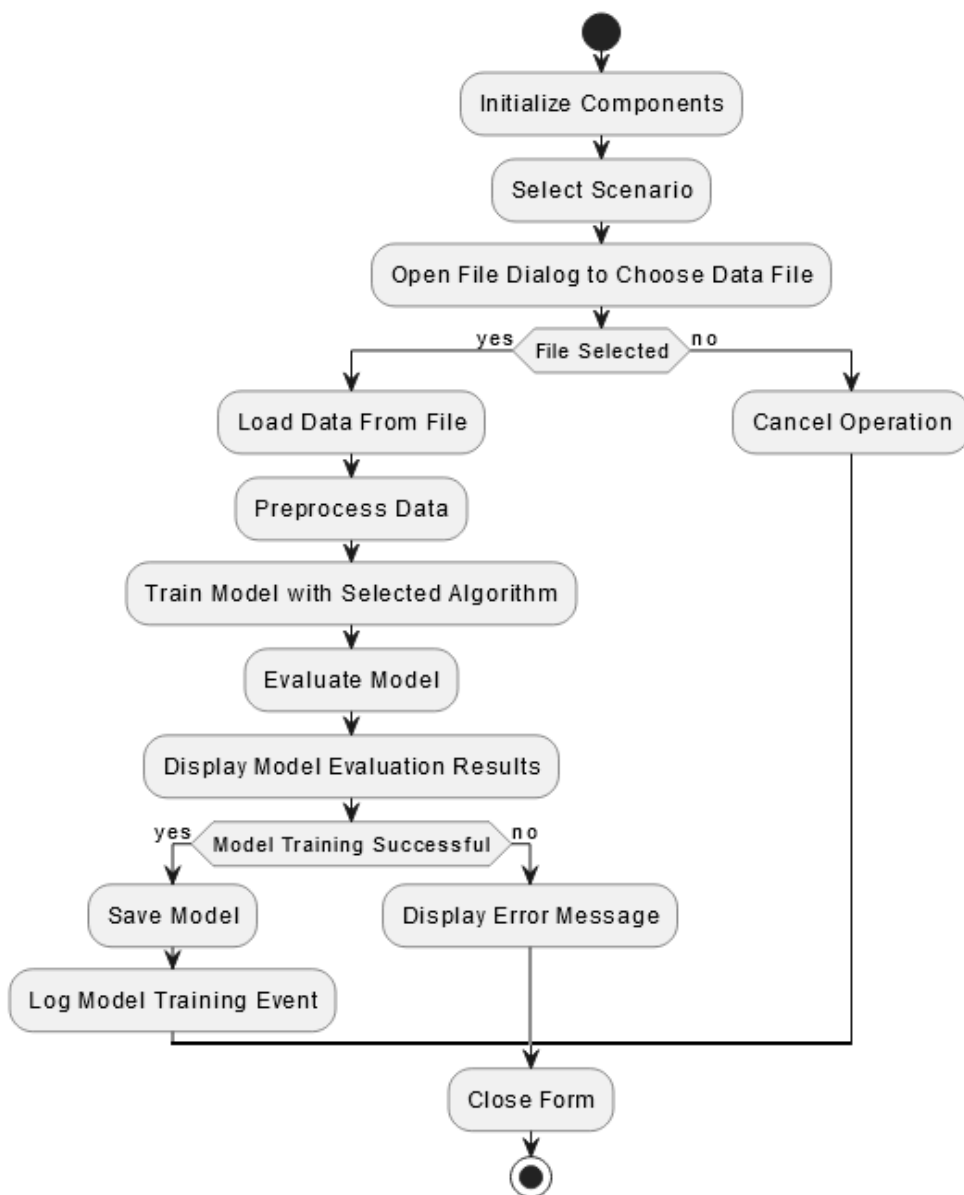


Рис. 2.3 Процес навчання моделі для моніторингу та прогнозування статусів вантажного транспорту

Рис. 2.4 відображає діаграму послідовності, яка ілюструє процес моніторингу та прогнозування статусів вантажного транспорту під час збору врожаю. Вона починається з того, що форма передбачень звертається до постачальника транспорту для отримання списку всіх транспортних засобів, а потім отримує відповідь з транспортними засобами. Також форма запитує у постачальника сценаріїв усі доступні сценарії, отримуючи їх список. Після вибору сценарію, форма запитує модель за цим сценарієм у постачальника моделей, який повертає вибрану модель. Далі створюється двигун прогнозування з контексту машинного

навчання, і за допомогою цього двигуна виконується прогноз статусу транспорту. Результати прогнозування відображаються на формі.

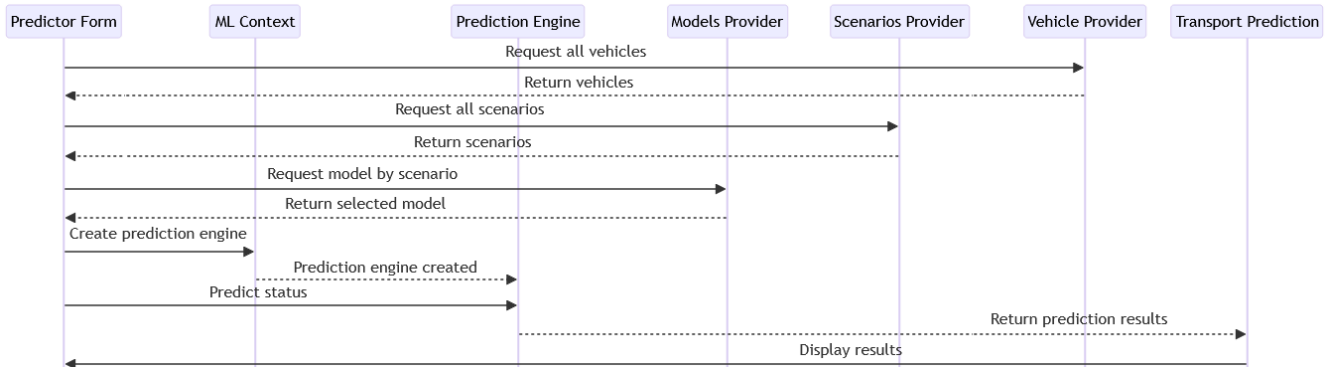


Рис. 2.4 Процес моніторингу та прогнозування статусів вантажного транспорту

2.4 Проектування бази даних для системи моніторингу

Створення структури бази даних є критично важливим аспектом розробки будь-якої сучасної інформаційної системи. У рамках даного проекту, що займається організацією даних для складської системи, було обрано сутнісно-зв'язковий підхід (Entity-Relationship model, ER-модель). Ця модель передбачає створення як логічного, так і фізичного рівнів представлення даних, що дозволяє ефективно моделювати структуру і взаємозв'язки між даними.

Використовувана система базується на реляційній базі даних, яка організована через комплекс таблиць. Кожна таблиця призначена для зберігання інформації, специфічної для певних аспектів діяльності складу. Це структурування допомагає чітко організувати дані таким чином, щоб вони були легкодоступні та зрозумілі для керування, виконання запитів та звітності.

Кожна таблиця відповідає за окремий сегмент інформації, від даних про товарні запаси до інформації про замовлення та логістику, що сприяє не тільки підвищенню ефективності операцій, але і значно спрощує процеси управління складом і аналізу даних. Відповідне використання ER-моделі забезпечує глибоке розуміння структури даних та їх взаємозв'язків, що є фундаментом для розвитку та адаптації системи в мінливих умовах бізнесу.

База даних системи моніторингу статусів вантажного транспорту складається із наступних сутностей та їхніх атрибутів:

Таблиця «Users» призначена для зберігання інформації про користувачів системи (табл. 2.6). Вона дозволяє зберігати та організувати інформацію про користувачів системи, включаючи їх особисті дані та права доступу, що може бути важливим для забезпечення безпеки та управління доступом до системи.

Таблиця 2.6

Атрибути сутності «Users»

№	Найменування	Тип даних	Призначення
1	UsersId	INT	Ідентифікатор користувача
2	FirstName	NVARCHAR (50)	Поле для вказівки імені користувача
3	LastName	NVARCHAR (50)	Поле для вказівки прізвища користувача
4	UserName	NVARCHAR (50)	Поле для вказівки логіну користувача
5	UsersPassword	NVARCHAR (200)	Поле для зберігання хешованого пароля користувача
6	RoleId	INT	Ідентифікатор ролі, що визначає права доступу користувача до системи
7	Description	NVARCHAR (1000)	Поле для додаткового опису або характеристик користувача
8	Email	NVARCHAR (150)	поле для зберігання електронної пошти користувача

Таблиця «Vehicle» призначена для зберігання інформації про транспортні засоби у системі (табл. 2.7).

Таблиця 2.7

Атрибути сутності «Vehicle»

№	Найменування	Тип даних	Призначення
1	VehicleId	INT	Ідентифікатор транспортного засобу
2	RegisterNumb	NVARCHAR (150)	Реєстраційний номер транспортного засобу
3	Brand	NVARCHAR (150)	Назва виробника транспортного засобу
4	Model	NVARCHAR (150)	Модель транспортного засобу
5	LoadCapacity	FLOAT (53)	Максимальна вантажопідйомність ТЗ
6	FuelType	NVARCHAR (50)	Тип палива, яке використовує ТЗ
7	Description	NVARCHAR (MAX)	Додатковий опис або характеристики ТЗ

Таблиця «Logs» призначена для зберігання записів про події, що відбуваються в системі (табл. 2.8). Вона забезпечує важливий механізм реєстрації та відстеження подій, що дозволяє забезпечити аудит та контроль за діяльністю системи.

Таблиця 2.8

Атрибути сутності «Logs»

№	Найменування	Тип даних	Призначення
1	LogsId	INT	Ідентифікатор запису події
2	UsersId	INT	Ідентифікатор користувача, який ініціював подію
3	EventNameShow	NVARCHAR (MAX)	Поле, що містить назву події, яка відображає суть події для зручності моніторингу та аналізу
4	EventDate	DATETIME	Поле дати та часу, що фіксує дату та час виникнення події

Таблиця «Models» призначена для зберігання інформації про моделі у системі. Вона дозволяє зберігати та організовувати інформацію про моделі, які використовуються у системі, що є важливим для їх управління та використання.

Атрибути сутності «Models»

№	Найменування	Тип даних	Призначення
1	ModelsId	INT	Ідентифікатор моделі
2	ModelsNames	NVARCHAR(250)	Поле для вказівки назви моделі
3	ScenariosId	INT	Ідентифікатор сценарію, пов'язаний з моделлю, що дозволяє встановити зв'язок між моделлю та сценарієм, який вона використовує
4	ModelsFileModel	NVARCHAR(MAX)	Поле для зберігання файлу моделі або посилання на нього

Таблиця «Scenarios» призначена для зберігання інформації про сценарії у системі. Вона дозволяє зберігати та організувати інформацію про різні сценарії, які можуть використовуватися у системі для виконання різних завдань.

Таблиця 2.10

Атрибути сутності «Scenarios»

№	Найменування	Тип даних	Призначення
1	ScenariosId	INT	Ідентифікатор сценарію
2	ScenariosName	NVARCHAR(250)	Поле для вказівки назви сценарію
3	Description	NVARCHAR(MAX)	Поле для додаткового опису або характеристик сценарію

Фізична модель бази даних забезпечує докладний опис структур даних, які мають бути реалізовані у базі, включаючи всі атрибути, необхідні для підтримки функціональності та операцій системи. Модель охоплює опис таблиць та встановлює відносини між ними, надаючи цілісне уявлення про організацію даних на фізичному рівні. Така модель є критичною для точного визначення структури бази даних та служить міцною основою для її подальшого розроблення та оптимізації.

На рис. 2.5 представлено сутнісно-зв'язкову діаграму (ER-діаграма), яка була спеціально розроблена для інформаційно-довідкової системи у контексті об'єктно-орієнтованого програмування. Діаграма ілюструє взаємодії та зв'язки між

ключовими сутностями системи, такими як концепції об'єктно-орієнтованого програмування, мови програмування, патерни проектування, завдання та їх рішення, навчальні ресурси та пошукові механізми в довідниках, підкреслюючи їхні взаємозв'язки та залежності.

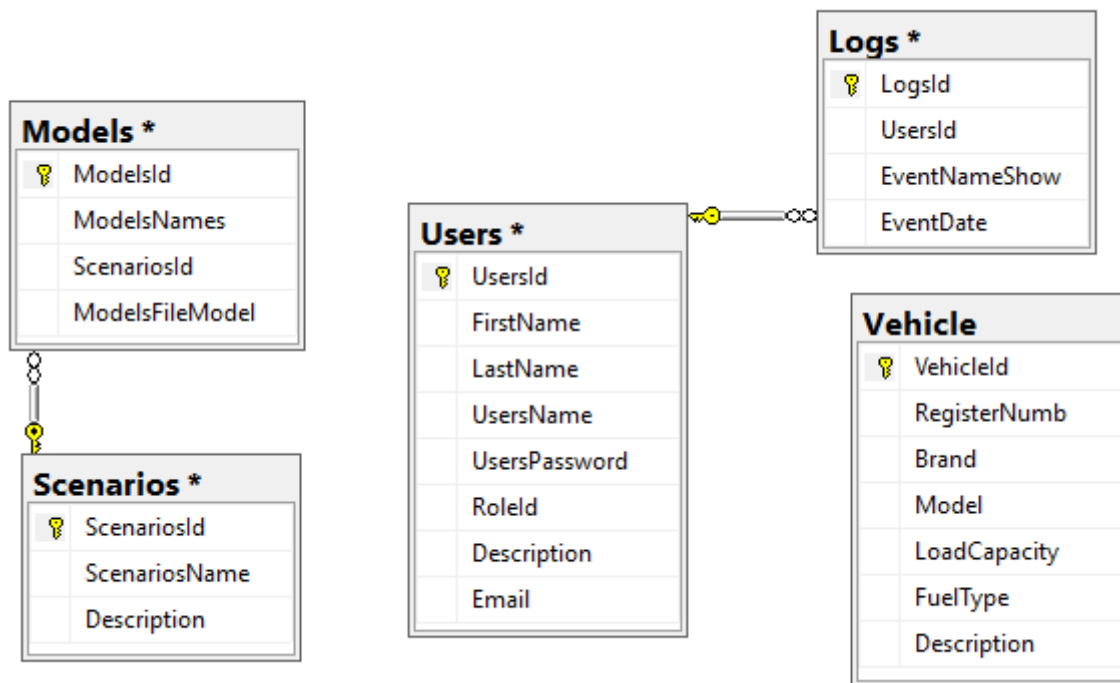


Рис. 2.5 ERD діаграма

Побудована модель бази даних значно спрощує процес розуміння архітектури бази даних і відіграє важливу роль у запобіганні помилок під час розробки, надаючи візуальне представлення структур даних та їхніх відносин. Ця діаграма служить надійною основою для дизайну баз даних, що сприяє точному виконанню та втіленню складних систем управління інформацією.

2.5 Проектування архітектури системи моніторингу

У процесі проектування системи було вирішено впровадити трьохрівневу архітектуру, що є ключовою для ефективного розподілу функцій системи. Використання такої структури обумовлене потребою досягнення гармонійного балансу між масштабованістю, зручністю управління і високою продуктивністю.

Трьохрівнева архітектура передбачає розподіл функціональних елементів на три рівні: презентаційний, бізнес-логічний, і шар доступу до даних [24]. Кожен рівень відіграє свою визначену роль і спеціалізується на певних задачах, сприяючи оптимальному функціонуванню системи.

На презентаційному шарі створюється користувацький інтерфейс, який забезпечує інтуїтивне взаємодіяння користувачів з системою, дозволяючи їм ефективно моніторити і керувати роботою обладнання. Цей шар також відіграє важливу роль у забезпеченні безпеки, ізолюючи користувацький доступ від безпосереднього звернення до бази даних, що знижує ризик несанкціонованих дій.

Бізнес-логічний шар концентрується на обробці та аналізі даних, використовуючи сучасні алгоритми машинного навчання та аналітичні інструменти для обробки інформації, що сприяє прийняттю обґрунтованих рішень. Його модульна конструкція забезпечує легку адаптацію до змін та оновлення без порушення роботи інших компонентів системи.

Шар доступу до даних забезпечує взаємодію системи з базами даних та іншими зовнішніми джерелами, відповідаючи за збір, зберігання, обробку і передачу даних. Ця частина архітектури дозволяє ефективно керувати великими обсягами інформації та гнучко адаптуватися до нових вимог.

Загалом, трьохрівнева архітектура забезпечує чітке розмежування між ключовими областями системи, що сприяє її масштабованості, легкому управлінню та здатності ефективно адаптуватися до змінюваних обставин та вимог.

На початковій стадії проекту було створено важливий компонент – шар доступу до даних, який став основою для взаємодії програми з базою даних. В рамках цього шару розроблено низку класів, що взаємодіють з певними таблицями бази даних. Назви цих класів містять суфікс «Provider», що вказує на їх функцію в доступі та управлінні даними. Щоб ясно відобразити взаємозв'язки між цими класами та методами, які застосовуються для маніпуляції даними, було складено діаграму класів. Представлена на рис. 2.6, ця діаграма забезпечує глибоке розуміння структури шару доступу до даних і візуалізує основні механізми обробки та керування інформацією в системі.

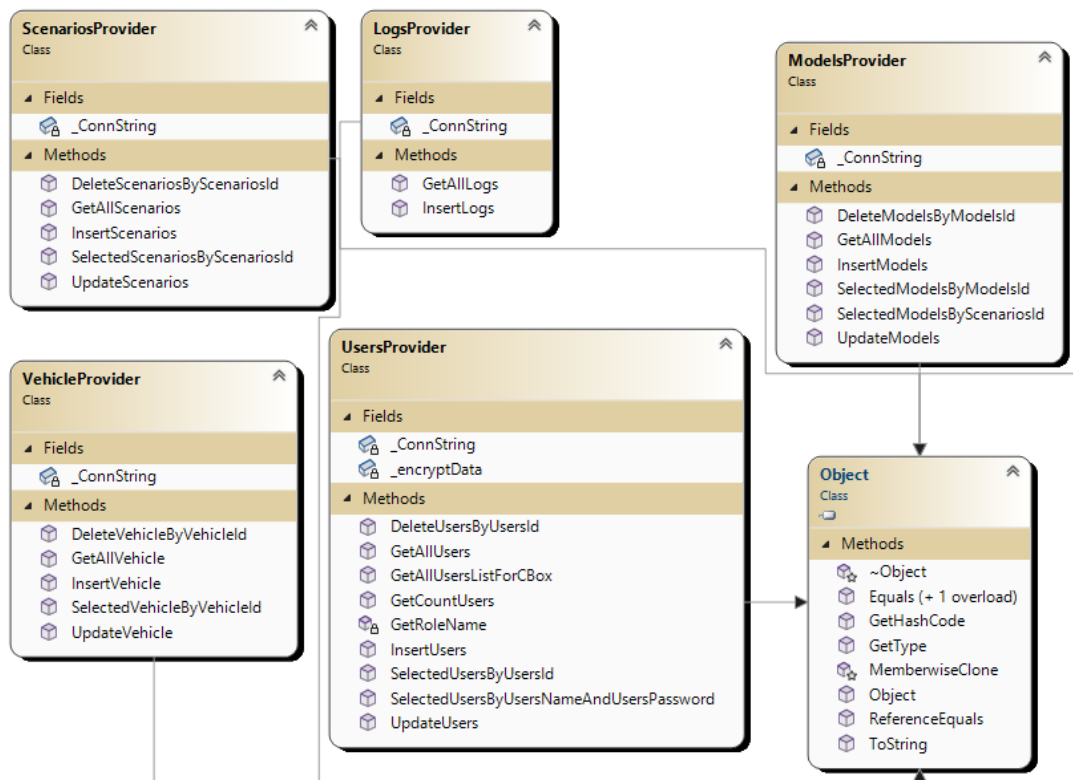


Рис. 2.6 Діаграма класів рівня даних

Діаграма даного рівня складається із таких класів:

- клас `LogsProvider` відповідає за збереження та відстеження подій, що відбуваються в системі. Він дозволяє створювати нові записи подій, витягувати існуючі події для перегляду, оновлювати деталі подій, та видаляти застарілі або непотрібні події;
- клас `ModelsProvider` управляє моделями машинного навчання, які використовуються для аналізу даних і прогнозування. Він надає інтерфейс для додавання нових моделей, отримання інформації про існуючі моделі, оновлення параметрів моделей і видалення застарілих моделей;
- клас `ScenariosProvider` керує сценаріями, які визначають, як використовувати моделі машинного навчання для конкретних аналітичних задач. Він дозволяє додавати сценарії, переглядати деталі існуючих сценаріїв, модифікувати їх за потребою, та видаляти ті, що більше не використовуються;
- клас `UsersProvider` відповідає за управління інформацією про користувачів системи. Це включає можливості для додавання нових користувачів,

витягу інформації про існуючих користувачів , оновлення профілів користувачів , а також видалення користувачів, які вже не мають доступу до системи;

– VehicleProvider керує інформацією про вантажні транспортні засоби в системі. Клас дозволяє реєструвати нові транспортні засоби , отримувати деталі про існуючі транспортні засоби , оновлювати їх дані , та видаляти інформацію про ті транспортні засоби, які більше не використовуються.

Ці класи рівня даних забезпечують структурований та ефективний спосіб управління ключовими даними у системі, відповідаючи за збереження цілісності, безпеку та доступність необхідної інформації.

Також, важливу увагу приділено розробці інтерактивного користувацького інтерфейсу, який представлено на рис. 2.7.

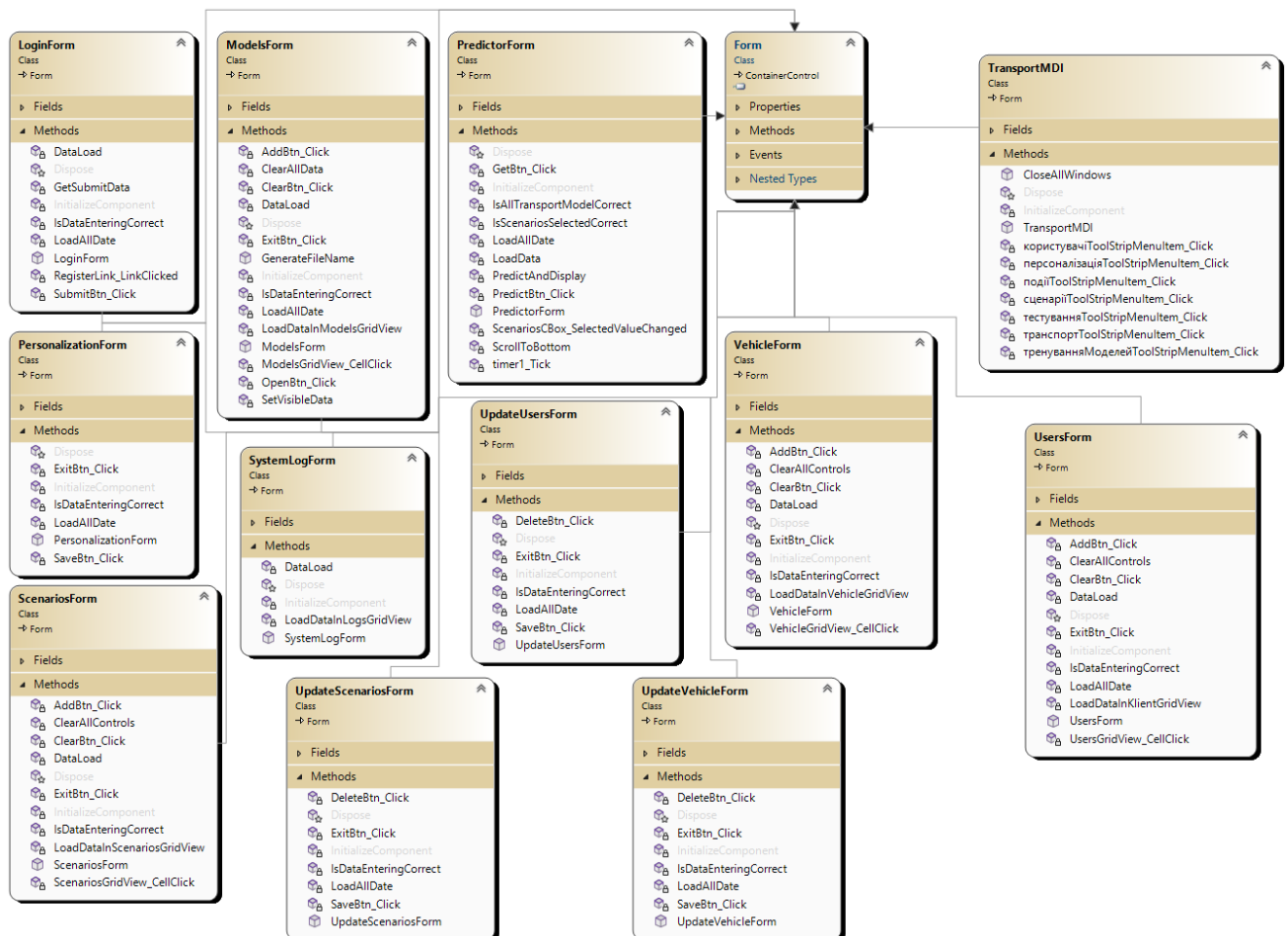


Рис.. 2.7 Діаграма класів інтерфейсу користувача

Діаграма даного рівня складається з дванадцяти класів рівня UI та є похідними від класу Form, тобто мають графічний інтерфейс:

- клас `LoginForm` використовується для авторизації користувачів у системі. Він дозволяє користувачам вводити свої логіни та паролі для доступу до функціоналу системи, забезпечуючи важливий рівень безпеки;
- клас `PersonalizationForm` дозволяє користувачам налаштувати інтерфейс системи відповідно до своїх особистих переваг, таких як тема інтерфейсу, мова, та інші налаштування;
- клас `SystemLogForm` надає користувачам доступ до перегляду логів системи. Він дозволяє відслідковувати всі важливі події та зміни, що відбуваються в системі, сприяючи кращому розумінню та управлінню;
- клас `UpdateUsersForm` дозволяє адміністраторам редагувати або видаляти існуючих користувачів системи, забезпечуючи управління доступом та ролі користувачів;
- клас `UsersForm` призначений для додавання нових користувачів та перегляду списку всіх користувачів системи. Це центральне місце для управління користувацькими акаунтами;
- клас `TransportMDI` призначений для управління модулями, пов'язаними з транспортними засобами, включаючи перегляд, додавання, оновлення стану та інші операції;
- клас `PredictorForm` використовується для моніторингу та прогнозування стану вантажних транспортних засобів. Він включає інтерфейси для перегляду даних, введення параметрів прогнозування та візуалізації результатів;
- клас `ModelsForm` призначений для створення та тренування моделей машинного навчання, який надає інструменти для конфігурації параметрів тренування та запуску процесів навчання;
- клас `ScenariosForm` призначений для створення нових та перегляду існуючих сценаріїв прогнозування, які використовуються в системі для аналітики та вирішення специфічних завдань;
- клас `UpdateScenariosForm` використовується для оновлення та видалення сценаріїв, що дозволяє модифікувати або видаляти застарілі або непотрібні сценарії в системі;

- клас `VehicleForm` дозволяє додавати нові транспортні засоби до системи та переглядати детальну інформацію про існуючі засоби;
- клас `UpdateVehicleForm` забезпечує функціональність оновлення або видалення інформації про транспортні засоби в системі, допомагаючи тримати дані актуальними і точними.

Розробка бізнес-логіки зайняла ключову роль у структурі системи, служачи фундаментом для всіх бізнес-процесів. Цей елемент системи об'єднує важливі бізнес-правила та алгоритми, які потрібні для аналізу та прогнозування параметрів емісій двигунів. Він відповідає за логіку обробки вхідних даних, їхній аналіз та виведення висновків, становлячи основу діяльності системи.

На рис. 2.8 представлено діаграму архітектури бізнес-логіки, яка чітко ілюструє її взаємодію з іншими компонентами системи та демонструє зони відповідальності. Це допомагає краще зрозуміти структуру та логіку програми, показуючи, як різні елементи взаємодіють між собою та впливають на загальну продуктивність і ефективність системи.

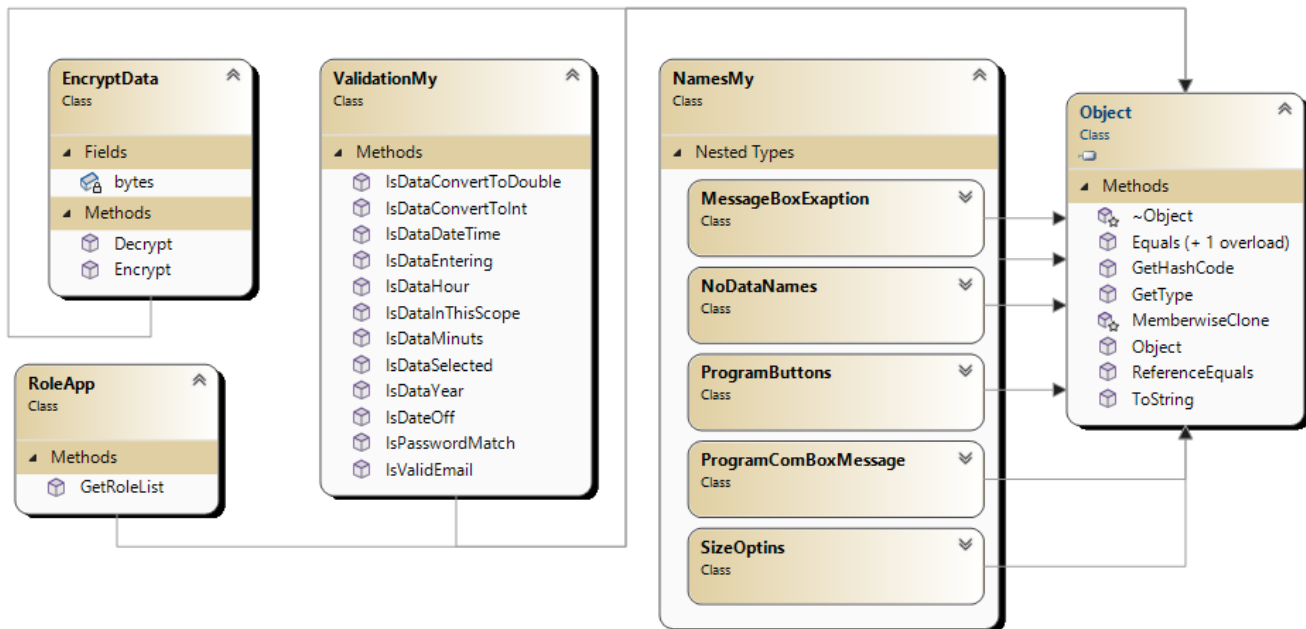


Рис. 2.8 Діаграма класів бізнес логіки

Класи рівня бізнес-логіки в системі відповідають за обробку даних, застосування бізнес-правил та алгоритмів, необхідних для реалізації основних функціональних вимог системи:

- клас `ValidationMy` відповідає за перевірку валідності вхідних даних у програмі. Його основна функція полягає у забезпеченні правильності та безпеки даних, що користувачі вводять або передають через систему. Він може включати методи для перевірки форматів даних, таких як електронні адреси, телефонні номери, ідентифікаційні номери та інші користувацькі входи, гарантуючи, що вони відповідають визначеним критеріям і стандартам;

- клас `NamesMy` використовується для управління і форматування назв в системі. Це включає стандартизацію назв осіб, місць, предметів тощо згідно з внутрішніми конвенціями;

- клас `EncryptData` відіграє ключову роль у захисті чутливих даних, шифруючи інформацію перед її зберіганням або передачею. `EncryptData` може включати методи для шифрування і дешифрування даних, використовуючи сучасні криптографічні алгоритми. Це допомагає захистити персональні дані користувачів, фінансову інформацію, та інші чутливі дані від несанкціонованого доступу;

- клас `RoleApp` керує ролями та дозволами у системі. Він визначає, які дії можуть виконувати користувачі в залежності від їхньої ролі в системі. Методи класу можуть включати додавання, видалення або зміну ролей, а також налаштування правил доступу для різних сегментів системи. Це забезпечує гнучке та безпечне управління доступом до функцій і ресурсів системи, гарантуючи, що кожен користувач має доступ лише до тих даних та дій, що відповідають його або її ролі.

Розроблені класи бізнес-логіки допомагають створити структуровану та безпечну обробку інформації, що є необхідною для забезпечення безпеки та ефективності системи.

У рамках даного розділу було виконано комплексне проектування системи моніторингу статусів вантажного транспорту, що охоплює всі аспекти від формулювання вимог до реалізації архітектури системи. Були розроблені та детально описані функціональні та нефункціональні вимоги системи, що дозволило визначити основні завдання та очікувані характеристики застосунку. Використання

use-case діаграми забезпечило ясне уявлення про варіанти використання системи та її інтеракції з користувачами.

Вибір технологічного стеку здійснено на основі аналізу переваг програмування на C# та використання баз даних MS SQL Server, що забезпечить оптимальну продуктивність і безпеку даних. Це вибори допоможуть у подальшому легше інтегрувати систему з іншими продуктами Microsoft та забезпечити високу масштабованість і адаптивність системи.

Було створено діаграми активності та послідовності для основних бізнес-процесів, які включають авторизацію користувачів і моніторинг статусів транспорту. Це дозволило відобразити і оптимізувати процеси обробки та аналізу даних, зробивши їх більш ефективними та зрозумілими.

Проектування бази даних з використанням фізичної моделі дало змогу структурувати і зберігати необхідні дані у логічно організованій формі, що сприятиме швидкому доступу та обробці інформації в реальному часі.

Впровадження трьохрівневої архітектури системи забезпечило чітке відділення логіки доступу до даних, бізнес-логіки та презентаційного шару, що сприяє кращій організації коду та спрощенню подальшої підтримки та розвитку системи. Кожен рівень було детально розроблено та описано, що підвищує загальну ефективність роботи системи.

Результати цього розділу дозволять у наступному розділі зосередитися на реалізації і тестуванні системи, а також на аналізі її ефективності та відповідності встановленим вимогам.

3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ СИСТЕМИ МОНІТОРИНГУ СТАТУСІВ ВАНТАЖНОГО ТРАНСПОРТУ В ПРОЦЕСІ ЗБОРУ ВРОЖАЮ

3.1 Розроблення загальної архітектури системи

Створення ефективної архітектури починається з аналізу ключових завдань, які система повинна виконувати, а саме збір, передачу, зберігання, обробку та аналіз даних про стан і місцеперебування транспортних засобів. Необхідно забезпечити, щоб система була масштабованою, надійною і безпечною, а також щоб вона інтегрувалася з існуючими технологічними рішеннями і підтримувала потенційне розширення функціоналу.

На першому етапі визначаються основні компоненти системи, такі як датчики та контролери на транспортних засобах, мобільні зв'язки для передачі даних, сервери баз даних та аналітичні інструменти. При цьому особлива увага приділяється вибору технологій, що найкраще відповідають вимогам по швидкості обробки, обсягу зберігання даних та їх захисту. Застосування сучасних протоколів передачі даних та методів шифрування забезпечує захист інформації від несанкціонованого доступу. Далі важливою складовою є розробка диспетчерського застосунку, що надає інтерфейс для візуалізації даних, управління сповіщеннями та плануванням технічного обслуговування. Застосунок повинен бути інтуїтивно зрозумілим, швидким і мобільно сумісним, щоб забезпечити легкий доступ до інформації в будь-який час і з будь-якого місця.

Також необхідно передбачити систему моніторингу і сповіщення, яка дозволяє оперативно реагувати на зміни стану вантажів або технічні несправності. Система має автоматизувати процеси колекціонування даних, їх аналізу та представлення результатів у зручній формі, щоб оптимізувати робочий процес і підвищити загальну ефективність логістики.

Arduino Leonardo – це мікроконтролерна плата, яка базується на мікроконтролері ATmega32u4. Це одна з численних розробок в лінійці Arduino, відомої своєю доступністю, гнучкістю та великою спільнотою користувачів, що

робить її популярним вибором для освітян і інженерів для реалізації різноманітних проектів.

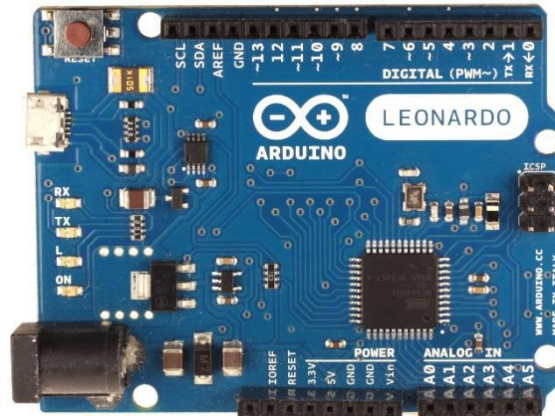


Рис. 3.1 Зовнішній вигляд мікроконтролера Arduino Leonardo

Основні характеристики Arduino Leonardo:

- мікроконтролер: ATmega32u4;
- робоча напруга: 5V;
- вхідна напруга (рекомендована): 7-12V;
- вхідна напруга (межі): 6-20V;
- цифрові I/O піни: 20, з яких 7 можуть використовуватися як PWM виходи і 12 як аналогові входи;
- аналогові входи: 12;
- серійні порти: 1 (на базі UART);
- пам'ять: 32 KB Flash (з яких 4 KB використовується для загрузчика), 2.5 KB SRAM, 1 KB EEPROM;
- швидкість годинника: 16 MHz.

Призначення Arduino Leonardo в системі моніторингу вантажного транспорту:

- збір даних з датчиків. Arduino може підключатися до різноманітних датчиків через свої аналогові та цифрові порти. У системі це будуть датчики швидкості, тиску, GPS-модулі для визначення локації тощо. Ці дані зчитуються та первинно обробляються мікроконтролером для подальшої передачі;

– комунікація з центральною базою даних. За допомогою модуля SIM800L, Arduino Leonardo може відправляти зібрані дані на сервер через мобільний інтернет. Він також може приймати команди з сервера для управління налаштуваннями або параметрами датчиків.

Переваги використання Arduino Leonardo:

- низька вартість. Arduino Leonardo є бюджетним варіантом, що робить його доступним для широкого спектру проектів;
- гнучкість програмування. Завдяки підтримці мови програмування C++, Arduino дозволяє розробляти складні програмні логіки з легкістю;
- велика кількість готових бібліотек та прикладів. Спільнота Arduino надає велику кількість готових до використання бібліотек та прикладів коду, що значно спрощує інтеграцію різноманітних модулів та датчиків.

Таким чином, Arduino Leonardo є оптимальним рішенням для розробки систем моніторингу вантажного транспорту, де потрібна надійність, гнучкість та здатність до масштабування.

Модуль SIM800L – це компактний GSM/GPRS модуль, який використовується для додавання мобільного зв'язку до мікроконтролерних проектів, таких як Arduino (рис. 3.2). Він підтримує чотири діапазони частот GSM (850, 900, 1800, 1900 MHz), що дозволяє йому працювати з мобільними мережами у більшості країн світу.



Рис. 3.2 Зовнішній вигляд модуля SIM800L

Основні характеристики модуля SIM800L:

- підтримка мереж: Quad-band 850/900/1800/1900 MHz, що забезпечує сумісність у всесвітньому масштабі;
- GPRS multi-slot class: 12/10. GPRS mobile station class B;
- підтримка протоколів: Підтримує GPRS даних до 85.6 kbps (download/upload);
- трансмісія даних: підтримка SMS текстових повідомлень, даних, голосу;
- інтерфейси: підтримує стандартні AT команди через серійний порт;
- напруга живлення: 3.4 - 4.4V, рекомендовано 4V.
- Струм споживання: У режимі передачі може досягати до 2A, тому важливо забезпечити адекватне живлення.

У системі моніторингу вантажного транспорту модуль SIM800L використовується для забезпечення двостороннього зв'язку між вантажним транспортом та центральною базою даних. Ось як він функціонує у системі:

- передача даних. Модуль може відправляти дані зібрані датчиками і контролерами (наприклад, GPS-координати, швидкість, стан вантажу) до сервера через мережу GSM. Це включає в себе відправлення реального часу та історичних даних для аналізу та моніторингу;
- отримання команд. Модуль також може отримувати команди від сервера, які можуть бути використані для управління параметрами транспортного засобу або для відправки спеціальних інструкцій водію;
- сповіщення: SIM800L може відправляти SMS сповіщення або виконувати голосові виклики в разі екстрених ситуацій або для передачі важливих попереджень.

Переваги використання SIM800L:

- універсальність. Підтримка кількох GSM частот гарантує роботу в більшості географічних регіонів;

- невеликі розміри та простота інтеграції. Модуль легко інтегрується з різними мікроконтролерами, включаючи Arduino, завдяки малому розміру та простоті використання серійного зв'язку;

- економічність. SIM800L є бюджетним рішенням, яке забезпечує функціональність мобільного зв'язку без значних витрат.

Завдяки цим характеристикам, SIM800L є відмінним вибором для систем моніторингу, які потребують надійного та ефективного зв'язку для передачі даних і управління транспортними засобами.

Модуль NEO-6M GPS від u-blox є одним із найпопулярніших GPS-модулів у середовищі DIY (зроби сам) і вбудованих систем через свою високу точність, надійність та доступність (рис. 3.3). Цей модуль широко використовується в проєктах з Arduino для визначення геопозиціонування транспортних засобів, дронів та інших мобільних пристроїв.



Рис. 3.3 Зовнішній вигляд модуля NEO-6M GPS від u-blox

Основні характеристики модуля NEO-6M GPS:

- чіпсет. U-blox NEO-6M з високою чутливістю та низьким енергоспоживанням;
- прийомник. 50 каналів, GPS L1 (1575.42 MHz) C/A код;
- час старту. Холодний старт близько 27 секунд, гарячий старт близько 1 секунди, автономний старт при повторному вмиканні близько 5 секунд;

- чутливість. До -161 dBm при відстеженні, забезпечуючи відмінну продуктивність навіть у умовах поганого сигналу;
- точність. Горизонтальна позиційна точність близько 2.5 метрів;
- швидкість оновлення. Стандартно 1 Гц (1 оновлення в секунду), але можливе збільшення до 5 Гц;
- енергоспоживання. Дуже низьке енергоспоживання (приблизно 50 мА під час роботи).

У системі моніторингу вантажного транспорту, модуль NEO-6M GPS використовується для забезпечення точних та надійних даних про місцезнаходження транспортних засобів. Ці дані критично важливі для таких аспектів системи, як:

- відстеження руху. Модуль дозволяє системі точно визначати і відслідковувати розташування кожного транспортного засобу в реальному часі;
- логістика та маршрутизація. Використання даних GPS дозволяє оптимізувати маршрути пересування та планування логістики, підвищуючи тим самим ефективність роботи транспорту;
- безпека та сповіщення. У разі відхилення від заданих маршрутів або інших непередбачених ситуацій, система може використовувати дані GPS для відправлення аварійних сповіщень.

Переваги використання NEO-6M:

- висока точність та надійність. Забезпечує точне позиціонування, що є важливим для систем моніторингу;
- легкість інтеграції. Серійний інтерфейс дозволяє легко підключити модуль до більшості мікроконтролерів;
- економічність. NEO-6M є доступним, знижуючи вартість розробки та імплементації систем моніторингу.

Використання модуля NEO-6M у системі моніторингу значно підвищує її ефективність, забезпечуючи важливі дані для контролю та управління вантажним транспортом.

Датчик DS3231 є високоточним, реального часу годинником (RTC - Real-Time Clock), який широко використовується в різних проектах на базі мікроконтролерів, включаючи системи моніторингу та контролю (рис. 3.4). Цей модуль забезпечує точне відстеження часу та дати, незалежно від зовнішніх умов, і здатний підтримувати ці дані завдяки наявності акумуляторної батареї навіть при відключенні живлення основної системи.

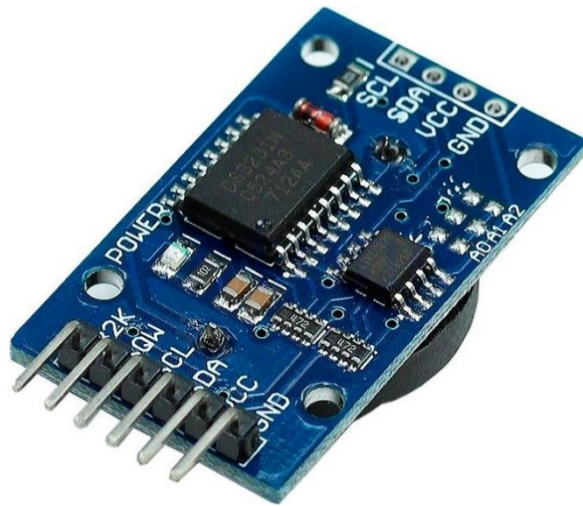


Рис. 3.4 Зовнішній вигляд датчика DS3231

Основні характеристики датчика DS3231:

- точність. DS3231 відрізняється високою точністю вимірювання часу, забезпечуючи відхилення від реального часу менше ніж ± 2 хвилини на рік при температурі від -40°C до $+85^{\circ}\text{C}$;
- енергоспоживання. Дуже низьке енергоспоживання, що робить його ідеальним для використання в мобільних та автономних системах;
- живлення від батареї. Має вбудований механізм для підтримки годинника від батареї при відсутності основного живлення;
- інтерфейс. Забезпечує I2C інтерфейс для зв'язку з мікроконтролерами, такими як Arduino.
- температурна компенсація. Включає вбудований температурний сенсор для автоматичної температурної компенсації.
- формат часу. Підтримка 24-годинного або 12-годинного формату з AM/PM індикацією.

У системі моніторингу вантажного транспорту, де важливо точно фіксувати час подій, таких як місцеперебування, швидкість руху, статуси завантаження і вивантаження та інші критичні дані, DS3231 забезпечує наступні можливості:

- точний часовий штамп. Кожна подія, що реєструється системою, може бути точно відмічена часовим штампом, що є критично важливим для логістичного планування та аналізу ефективності;
- синхронізація часу. Забезпечує однорідність часу між всіма компонентами системи, що дозволяє ефективно координувати дії та процеси;
- надійність та автономність. Датчик підтримує роботу системи в точному часовому режимі навіть при відключенні живлення, завдяки підтримці від батареї.

Переваги використання DS3231:

- висока точність та надійність. Забезпечує стабільну і точну роботу в широкому діапазоні температур;
- легкість інтеграції. I2C інтерфейс дозволяє легко підключити датчик до більшості мікроконтролерів;
- економія енергії. Низьке споживання енергії робить його ідеальним для використання в автономних системах, де джерело живлення обмежене.

Таким чином, використання DS3231 в системах моніторингу дозволяє підвищити точність та ефективність управління та контролю транспортних засобів.

Датчик HX711 є спеціалізованим 24-бітним аналого-цифровим перетворювачем (ADC), який оптимізований для точного та надійного зчитування даних з датчиків навантаження (рис. 3.5). Датчик використовується в електронних вагах та інших засобах вимірювання сили, де потрібна висока точність.

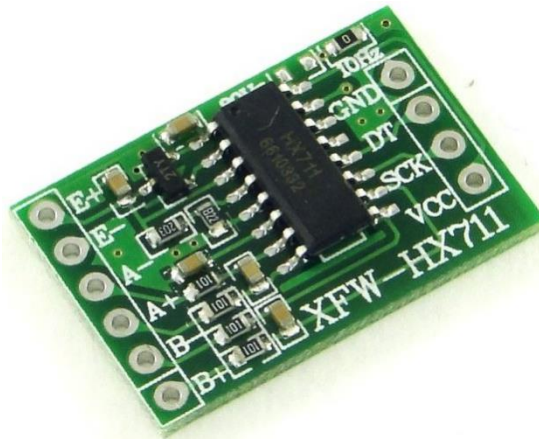


Рис. 3.5 Зовнішній вигляд датчика HX711

Основні характеристики датчика HX711:

- точність. Високоточний 24-бітний ADC, який забезпечує високу роздільну здатність та точність зчитувань;
- швидкість оновлення. Здатний надавати швидкість оновлення до 10 або 80 Гц, що дозволяє користувачам вибрати оптимальну баланс між швидкістю та точністю;
- інтерфейс. Простий двопровідний інтерфейс для зв'язку з мікроконтролерами, наприклад Arduino;
- підтримка каналів. Підтримує два канали А і В для зчитування даних з датчиків. Канал А має програмоване підсилення 128 або 64, тоді як канал В має фіксоване підсилення 32;
- живлення. Робоча напруга від 2.7 до 5V.

У системі моніторингу вантажного транспорту, HX711 використовується для точного вимірювання ваги або навантаження, яке перевозиться транспортом. Це може бути особливо корисним для:

- моніторингу вантажопідйомності. Датчик дозволяє точно визначати масу вантажу на борту, що важливо для дотримання правил безпеки та оптимізації логістичних процесів;
- запобігання перевантаженню. Точне визначення ваги допомагає запобігти перевантаженню транспорту, що може призвести до штрафів або пошкодження транспортного засобу;

– вимірювання зміни навантаження. Відстеження змін у вазі може вказувати на завантаження або розвантаження товарів, що дає змогу автоматично реєструвати ці події в системі.

Переваги використання HX711:

– висока точність та надійність. Забезпечує стабільне та точне вимірювання в широкому діапазоні умов;

– простота інтеграції. Легко інтегрується з більшістю мікроконтролерів завдяки простому серійному інтерфейсу;

– низьке енергоспоживання. Підходить для використання в мобільних системах, де джерело живлення може бути обмеженим.

HX711 є важливим компонентом для систем, які вимагають надійного моніторингу механічних навантажень і ваги, забезпечуючи точність і ефективність для логістичних і транспортних операцій.

Датчик швидкості LM393, як правило, використовується у вигляді компонента в різних електронних схемах, але важливо уточнити, що LM393 сам по собі не є датчиком швидкості (рис. 3.6). Це насправді подвійний компаратор напруги, який часто використовується у різних датчиках, включаючи ті, що займаються вимірюванням швидкості. LM393 може бути частиною сенсорної системи, яка вимірює швидкість, наприклад, за допомогою оптичних або магнітних датчиків, але сам він не здатний вимірювати швидкість без інших сенсорних компонентів.

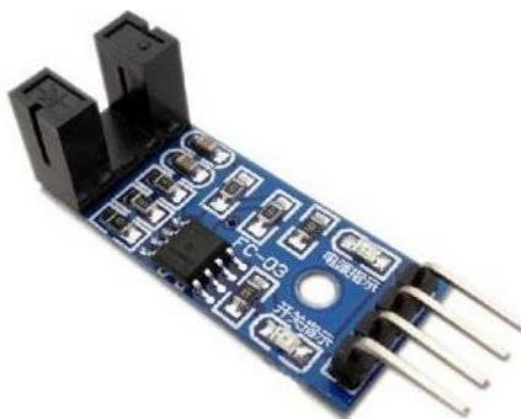


Рис. 3.6 Зовнішній вигляд датчика LM393

Основні характеристики LM393:

- тип. Двоканальний диференціальний компаратор напруги;
- вхідне напруга. Широкий діапазон вхідної напруги від 2 V до 36 V;
- вихід. Відкритий колектор, що означає, що вихідний стан може бути використаний для підключення до логічних схем або реле;
- споживана потужність. Низька споживана потужність;
- робоча температура. Від -40°C до 85°C ;
- упаковка. Доступний у різних типах корпусів, включаючи SOIC та DIP.

LM393 у системі використовується як частина сенсора швидкості, який моніторить оберти коліс або інших рухомих частин транспортного засобу. Наприклад, якщо використовувати LM393 з оптичними датчиками, які реагують на кожне проходження мітки на колесі, можна точно виміряти кількість обертів, а отже і швидкість руху транспортного засобу.

Переваги використання LM393:

- точність і надійність. LM393 дозволяє точно порівнювати напруги, що забезпечує високу точність у системах вимірювання;
- універсальність. Може бути використаний у широкому спектрі додатків і легко інтегрується з іншими компонентами;
- економічність. LM393 є досить доступним, що робить його популярним вибором для багатьох електронних проектів.

Використання LM393 у вантажних моніторингових системах може значно покращити точність вимірювань і забезпечити надійне та стабільне функціонування системи вимірювання швидкості.

На рис. 3.7 представлено архітектуру розробленої системи для моніторингу статусів вантажного транспорту в процесі збору врожаю.

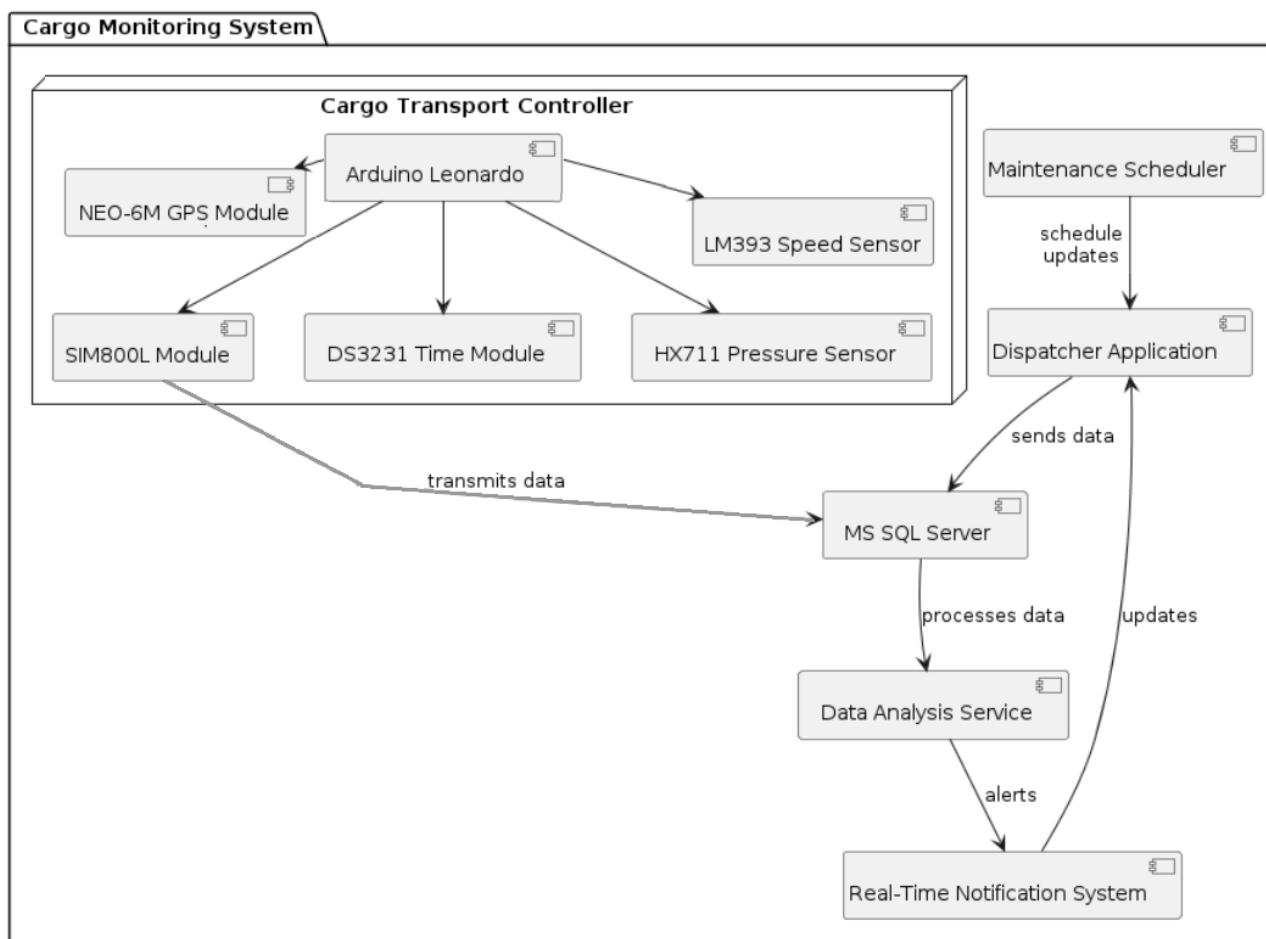


Рис. 3.7 Загальна архітектура системи

Система моніторингу вантажного транспорту розроблена для забезпечення ефективного збору, передачі, обробки та аналізу даних у реальному часі. Вантажні транспортні засоби обладнані контролерами на базі Arduino Leonardo, які підключені до різноманітних датчиків. GPS-модуль NEO-6M забезпечує точне відстеження місцезнаходження, модуль часу DS3231 відповідає за фіксацію часу подій, датчик тиску HX711 вимірює вагу вантажу, а датчик швидкості LM393 фіксує швидкість руху. Вся зібрана інформація передається через модуль SIM800L до центральної бази даних MS SQL Server.

Дані, що надходять до бази, обробляються сервісом аналізу даних, та відображаються у системі. Система сповіщень в реальному часі активно інформує диспетчерів про будь-які незвичайні події чи відхилення від норми, що дозволяє швидко реагувати на потенційні проблеми.

Диспетчерський застосунок відіграє ключову роль у взаємодії з усією системою, надаючи диспетчерам доступ до усієї необхідної інформації, а також засоби для управління логістикою вантажного транспорту.

Загалом, розроблена система може використовуватись для логістичних операцій під час збору врожаю та дозволить мінімізувати час простою, оптимізувати витрати на паливо та підвищити загальну ефективність роботи вантажного транспорту в процесі збору врожаю.

3.2 Розробка коду для керування контролером

Завдяки своїй гнучкості та широкому спектру використання, Arduino Leonardo відмінно підходить для збору даних від датчиків, управління комунікаціями та виконання інших задач контролю. Скетч для Arduino написаний на мові програмування C++, що дозволяє використовувати структуровані та ефективні програмні патерни.

Під час програмування особлива увага приділяється надійності та точності збору даних. Код повинен включати функції моніторингу стану обладнання, а також забезпечувати ефективне управління енергоспоживанням для автономної роботи в транспортних засобах.

На початковому етапі розробки необхідно підключити додаткові бібліотеки, які використовуються для реалізації різних функціональних можливостей в проєкті, зокрема для зчитування даних з GPS, роботи з реальним часом та вимірювання ваги (ліст. 3.1).

Лістинг 3.1 – Підключення бібліотек

```
#include <SoftwareSerial.h>
#include <TinyGPS++.h>
#include <Wire.h>
#include <DS3231.h>
#include «HX711.h»
```

У проєкті використовуються наступні бібліотеки:

– `SoftwareSerial.h` – ця бібліотека дозволяє створити програмний серійний порт на будь-яких цифрових пінах мікроконтролера Arduino. Використання програмного серійного порту корисне в тих випадках, коли фізичні серійні порти на платі вже зайняті або їх недостатньо. Це особливо актуально для з'єднання зовнішніх модулів, як-от модулів GSM або GPS, що комунікують за допомогою серійного зв'язку;

– `TinyGPS++.h` – ця бібліотека призначена для спрощення роботи з GPS модулями. Вона витягує та парсить дані, отримані від GPS модуля у форматі NMEA, який є стандартним форматом для GPS комунікацій. `TinyGPS++` надає зручні функції для отримання такої інформації, як широта, довгота, висота, час, дата, курс, швидкість руху та інші. Це робить її ідеальною для використання в проектах, де необхідно точне відстеження геопозиції;

– `Wire.h` – стандартна бібліотека для Arduino, яка використовується для здійснення комунікацій через інтерфейс I2C. Цей інтерфейс широко використовується для підключення різних сенсорів та модулів, таких як модулі часу реального часу (RTC), датчики температури, LCD дисплеї тощо. У вашому проекті ця бібліотека може використовуватися для зв'язку з модулем DS3231 для відстеження часу;

– `DS3231.h` – бібліотека для роботи з RTC модулем DS3231. DS3231 – це точний годинник реального часу з вбудованим термокомпенсованим кристалом та EEPROM. Бібліотека надає функції для читання та налаштування часу та дати на модулі, що є критично важливим для додатків, де потрібно точно маркувати дані часовими мітками;

– `HX711.h` – бібліотека для роботи з модулем HX711, який є аналого-цифровим перетворювачем для вимірювання механічного тиску.

Код у ліст. 3.2 організує налаштування модулів для системи моніторингу вантажного транспорту, яка використовує Arduino для збору та передачі даних.

Лістинг 3.2 – Налаштування модулів

```
// Налаштування пінів для HX711
const int LOADCELL_DOUT_PIN = 3;
```

```

const int LOADCELL_SCK_PIN = 2;
HX711 scale;
// Налаштування SIM800L
SoftwareSerial sim800l(10, 11); // RX, TX
String APN = «MyAPN_qwsdsd»;
String server = «freight_transport»; // Адреса сервера MS SQL
String sqlCommand = «INSERT INTO TransportData (gps, time, pressure, speed) VALUES (?,
?, ?, ?);»;
// Налаштування GPS
TinyGPSPlus gps;
SoftwareSerial ss(4, 5); // RX, TX
// Налаштування модуля часу
DS3231 clock;
bool Century=false;
bool h12Flag;
bool PM;

```

У даному коді проведено налаштування модулів, зокрема:

- налаштування пінів для HX711: Код визначає піни, до яких підключений HX711 – модуль для вимірювання ваги. LOADCELL_DOUT_PIN і LOADCELL_SCK_PIN встановлені на піни 3 та 2 відповідно. Об'єкт scale створюється для подальшого взаємодії з цим модулем. HX711 використовується для зчитування даних від датчиків навантаження, які можуть вимірювати вагу вантажу на транспорті;
- налаштування SIM800L. Використовується об'єкт SoftwareSerial для створення додаткового серійного зв'язку з модулем SIM800L, який забезпечує мобільний зв'язок для передачі даних. Піни 10 та 11 використовуються для RX (прийом) і TX (передача) відповідно. Змінні APN, server, і sqlCommand використовуються для зберігання налаштувань мережі, адреси сервера та SQL команди для передачі даних;
- налаштування GPS. Для роботи з GPS-модулем використовується бібліотека TinyGPSPlus, яка дозволяє легко парсити дані, отримані з GPS. Ще один

об'єкт `SoftwareSerial` створений для GPS модуля з пінами 4 та 5 для RX і TX. Це дозволяє визначити місцеположення транспорту;

- налаштування модуля часу. Об'єкт `DS3231` використовується для роботи з точним модулем часу (RTC). Цей модуль дозволяє точно трекати час і дату, що є критично важливим для реєстрації часових міток подій та даних. Змінні `Century`, `h12Flag`, і `PM` можуть використовуватися для отримання додаткових параметрів часу, таких як формат 12/24 годин або відстеження століття.

У наведеному ліст. 3.3 відбувається ініціалізація основних компонентів системи вантажного моніторингу на базі `Arduino`.

Лістинг 3.3 – Ініціалізація компонентів системи

```
void setup() {
  Serial.begin(9600);
  sim800l.begin(9600);
  ss.begin(9600);
  Wire.begin();
  scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);
  // Калібрування HX711
  scale.set_scale();
  scale.tare();
  // Ініціалізація SIM800L
  initializeGSM();
}
```

Функція `setup()` використовується для налаштування серійних зв'язків, ініціалізації модулів та калібрування датчиків, що забезпечує правильну підготовку обладнання до збору даних.

Ліст. 3.4 приводить код функції «`loop`», яка виконується безперервно після старту програми на `Arduino`, відбувається зчитування та обробка даних з підключених датчиків, а також відправка цих даних на сервер.

Лістинг 3.3 Код функції «`loop`»

```
void loop() {
  while (ss.available() > 0) {
    gps.encode(ss.read());
```

```

}
if (gps.location.isValid()) {
    float latitude = gps.location.lat();
    float longitude = gps.location.lng();
    String gpsData = String(latitude, 6) + «,» + String(longitude, 6);
    // Читання часу
    DateTime now = clock.getDateTime();
    String timeStr = String(now.hour()) + «:» + String(now.minute()) + «:» +
String(now.second());
    // Читання тиску
    long pressure = scale.get_units(10);
    // Читання швидкості int speed = analogRead(A0); // Відправка даних на сервер
    sendDataToServer(gpsData, timeStr, pressure, speed);
}
delay(1000);
}

```

Спочатку перевіряється, чи є доступні дані від GPS модуля за допомогою функції «available» на об'єкті ss. Якщо дані присутні, вони зчитуються і обробляються функцією «encode», яка розбирає GPS дані для подальшого використання.

Якщо дані про місцезнаходження валідні, визначаються широта та довгота, які потім перетворюються у рядковий формат із шістьма знаками після коми. Ці дані об'єднуються у рядок gpsData для подальшої передачі.

Потім з модуля реального часу отримується поточний час, і він форматується у рядок timeStr, який включає години, хвилини та секунди.

Також зчитується тиск з датчика ваги HX711 і перетворюється у значення тиску. Припускається, що швидкість може бути зчитана за допомогою аналогового входу, де підключено датчик швидкості.

На закінчення, усі зібрані дані – геопозиція, час, тиск і швидкість – відправляються на сервер через виклик функції «sendDataToServer». Функція «loop» закінчується паузою на одну секунду, після чого цикл повторюється.

Також важливо привести та описати код функції «sendDataToServer», що відповідає за відправлення зібраних даних на веб-сервер за допомогою GSM модуля SIM800L (ліст 3.5). Процес передачі даних організовано через HTTP запит.

Лістинг 3.5 Код функції «sendDataToServer»

```
void sendDataToServer(String gps, String time, long pressure, int speed) {
    String httpData = «GET « + server + «sqlCommand=« + sqlCommand + «&gps=« + gps +
«&time=« + time + «&pressure=« + pressure + «&speed=« + speed + « HTTP/1.1\r\nHost: « + server +
«\r\n\r\n»;
    sim800l.println(«AT+HTTPINIT»);
    delay(2000);
    sim800l.println(«AT+HTTPPARA=»URL»,\»« + httpData + «\»»);
    delay(2000);
    sim800l.println(«AT+HTTPACTION=0»);
    delay(5000);
    sim800l.println(«AT+HTTPTERM»);
    delay(300);
}
```

У середині функції спочатку формується HTTP запит у вигляді рядка, що містить метод GET із параметрами, які передаються на сервер. Ці параметри включають зібрані дані про GPS координати, час, тиск та швидкість, а також SQL команду для їх вставки у базу даних.

Для ініціації з'єднання з сервером через SIM800L, спочатку використовується команда AT для ініціалізації HTTP сесії. Після короткої паузи, необхідної для стабілізації з'єднання, модулю відправляється URL для HTTP запиту. Ще одна затримка перед виконанням HTTP запиту дозволяє GSM модулю коректно обробити усі параметри. Після запуску HTTP запиту система чекає на відповідь від сервера, після чого завершує HTTP сесію. Знову ж таки, використовується коротка затримка для того, щоб забезпечити належне закриття з'єднання.

Загальний код скетчу для управління контролером приведено у додатку А.

3.3 Інтеграція бази даних з програмним забезпеченням

Інтегрування бази даних з ПЗ вимагає детального розуміння того, як взаємодіють ключові компоненти, відповідальні за обмін і обробку інформації. Ефективність використання баз даних значною мірою залежить від правильного застосування з'єднань, команд та читачів даних чи адаптерів, які разом формують структуру для взаємодії між базою даних і програмою. На рисунку 3.8 представлена діаграма компонентів, яка демонструє ці взаємозв'язки в розроблюваній системі.

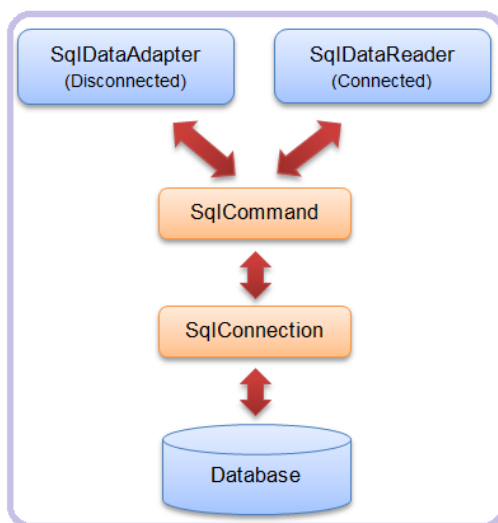


Рис. 3.8 Схема підключення застосунку до бази даних

На діаграмі визначені ключові компоненти, які відіграють важливу роль у забезпеченні взаємодії між програмним забезпеченням та базою даних, що зберігає всю необхідну інформацію для додатку. Ці компоненти включають:

- Database – фізичне сховище даних, де зберігається уся інформація, до якої потрібен доступ;
- SqlConnection – компонент, що відповідає за створення з'єднання між додатком та базою даних;
- SqlCommand – інструмент, який дозволяє виконувати SQL-операції, такі як вибірка, вставка, оновлення, та видалення даних через встановлене з'єднання;

– `SqlDataAdapter (Disconnected)` – використовується для роботи з даними без постійного з'єднання з базою, що дозволяє керувати даними офлайн та синхронізувати їх із базою пізніше;

– `SqlDataReader (Connected)` – застосовується для читання даних безпосередньо з бази даних, забезпечуючи швидкий доступ до даних відразу після їх запити.

Початкова взаємодія та послідовність дій між цими компонентами розпочинається з `SqlConnection`, який ініціює з'єднання з базою даних. Після ініціації з'єднання, контроль передається до `SqlCommand`, який генерує SQL-запити. `SqlDataReader` та `SqlDataAdapter` працюють з `SqlCommand` для виконання операцій над даними, де `SqlDataReader` вимагає постійного з'єднання для читання даних, а `SqlDataAdapter` дозволяє керувати даними локально та синхронізувати їх із базою пізніше. Використання цих компонентів залежить від конкретних потреб та сценаріїв застосування.

В конфігураційному файлі «App.config» створено змінну «CONNECT», яка включає рядок підключення до бази даних із всіма необхідними параметрами, деталі яких роз'яснені у документації проекту. На рис. 3.9 демонструється структура цього рядка підключення.

```

| <!-- Підключення до бази даних -->
| <appSettings>
| <add key="CONNECT" value="Data Source=(LocalDB)\MSSQLLocalDB;
| AttachDbFilename=|DataDirectory|\DB.mdf;Integrated Security=True" />
| </appSettings>

```

Рис. 3.9 Параметри конфігурації змінної бази даних

Цей рядок підключення дозволяє інтегрувати таблиці бази даних з формами програми в середовищі Visual Studio 2022, забезпечуючи зручний доступ до управління даними. Завдяки поліпшенням у Visual Studio 2022, процес інтеграції бази даних стає простішим, що сприяє швидшій розробці та спрощує модифікацію даних.

3.4 Розроблення модулів системи

Розробка користувацького інтерфейсу була ініційована із конструюванням основної форми та інтеграції елемента MenuStrip у центральний графічний інтерфейс. MenuStrip, який є ефективним засобом управління в рамках Windows Forms, надає можливість формувати графічні меню із численними опціями. Це значно полегшує процес навігації для користувачів і забезпечує їм прямий доступ до важливих функцій системи (рис. 3.10).

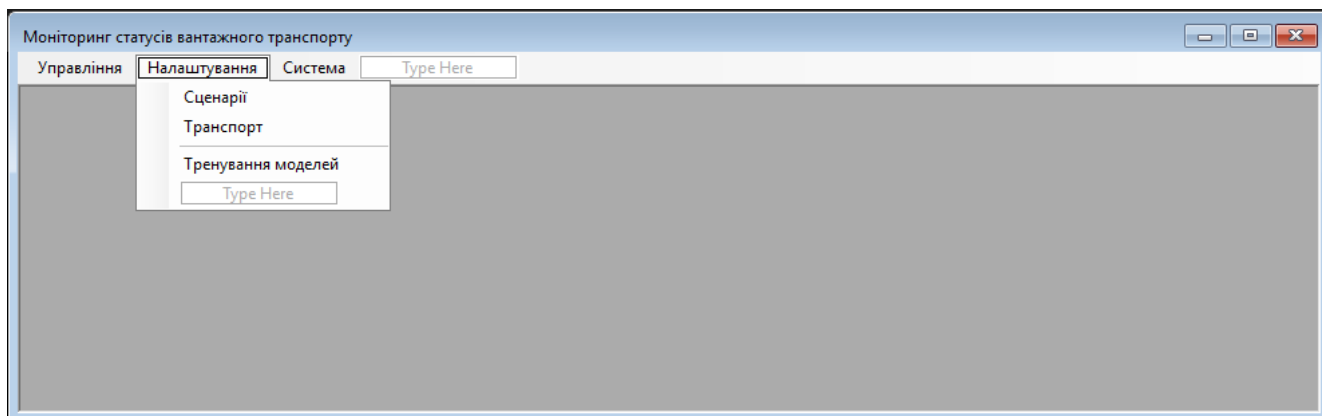


Рисунок 3.10 Основна форма

Для кожного вибраного пункту меню реалізовано специфічну реакцію системи, що активізується при його виборі. Як ілюструє рис. 3.11, при виборі опції «Моніторинг», система ініціює відкриття комплекту інструментів, необхідних для проведення аналізу та тестування моделей.

```
private void тестуванняToolStripMenuItem_Click(object sender, EventArgs e) {
    CloseAllWindows();
    PredictorForm predictorForm = new PredictorForm();
    predictorForm.MdiParent = this;
    predictorForm.WindowState = FormWindowState.Maximized;
    predictorForm.Show();
}
```

Рисунок 3.11 Активація форми «Моніторингу»

Ключовим аспектом є розробка інтерфейсу для тренування моделей машинного навчання, що демонструється на рис. 3.12.

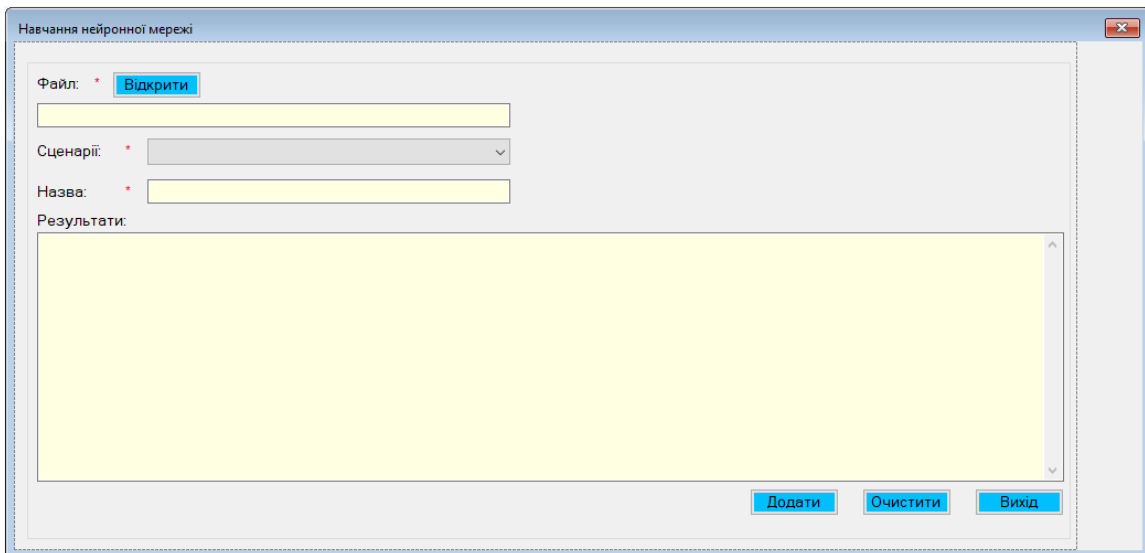


Рисунок 3.12 Форма тренування моделей

На рис. 3.13 представлено частину коду, що дозволяє користувачу вибирати файл із локальної файлової системи, забезпечуючи програмі можливість отримати шлях до цього файлу для подальшого використання.

```
// Відображення діалогового вікна та обробка результату
if (openFileDialog.ShowDialog() == DialogResult.OK) {
    _Path = openFileDialog.FileName;
    FileNameTextBox.Text = openFileDialog.FileName;
}
```

Рисунок 3.13 Код відкриття файлу із тренувальними наборами

Після відкриття файлу проходить ініціалізація контексту машинного навчання в програмі (рис. 3.14).

```
||| // Створення контексту ML
mlContext = new MLContext(seed: 0);
```

Рисунок 3.14 Створення контексту для машинного навчання

У фрагменті коду відбувається створення нового екземпляра `MLContext` із зазначенням начального значення для генератора випадкових чисел, щоб забезпечити відтворюваність результатів при кожному запуску моделі.

Після цього проходить завантаження даних із файлу для подальшого тренування моделі (рис. 3.15).

```
// Завантаження даних
dataView = mlContext.Data.LoadFromTextFile<PhishingData>(_Path,
    hasHeader: true,
    separatorChar: ',');
```

Рисунок 3.15 Завантаження даних

Після цього потрібно підготувати дані для машинного навчання (рис. 3.16).

```
// Виведення повідомлення про тренування моделі
ReportTBBox.Text += $"Training the model... \r\n";
// Підготовка даних
var dataProcessPipeline = mlContext.Transforms.Conversion.MapValueToKey(outputColumnName: "Label",
    inputColumnName: "Status")
    .Append(mlContext.Transforms.Concatenate("Features",
        nameof(TransportModel.Latitude),
        nameof(TransportModel.Longitude), nameof(TransportModel.Speed),
        nameof(TransportModel.LoadCapacity)))
    .Append(mlContext.Transforms.NormalizeMinMax("Features"));
```

Рисунок 3.16 Підготовка конвеєру даних для проведення тренування

Підготовка даних включає кілька етапів обробки, здійснених за допомогою інструментів машинного навчання. Спочатку проводиться конвертація значень статусу транспорту в унікальні ключі, що використовуються як мітки для навчання моделі. Потім ці мітки об'єднуються із координатами широти, довготи, швидкістю руху транспорту та його вантажопідйомністю для формування єдиного набору характеристик. Останнім кроком є нормалізація отриманих характеристик за методом мінімуму та максимуму для оптимізації процесу навчання моделі. Ці операції забезпечують коректну підготовку даних, що є критичною для ефективного прогнозування та аналізу.

Далі обирається алгоритм машинного навчання (рис. 3.17).

```
// Вибір алгоритму для багатокласової класифікації
var trainer =
    mlContext.MulticlassClassification.Trainers.LbfgsMaximumEntropy("Label", "Features");
```

Рисунок 3.17 Вибір алгоритму для багатокласової класифікації

Вибір алгоритму для багатокласової класифікації здійснюється за допомогою методу, який використовує техніку максимальної ентропії з оптимізацією через алгоритм обмеженої меморії Бройдена – Флетчера – Гольдфарба – Шанно (L-BFGS). Цей метод добре підходить для розв'язання задач, де необхідно

класифікувати вхідні дані, представлені набором числових характеристик, у кілька можливих категорій. Тренер алгоритму сприймає характеристики і мітки, заздалегідь підготовлені і нормалізовані, і використовує їх для вивчення моделі, яка ефективно може розрізняти різні статуси транспорту, такі як швидкість, розташування чи вантажопідйомність.

Після цього необхідно створити тренувальний конвеєра (рис. 3.18).

```
var trainingPipeline = dataProcessPipeline.Append(trainer)
    .Append(mlContext.Transforms.Conversion.MapKeyToValue("PredictedLabel"));
```

Рисунок 3.18 Створення тренувального конвеєра

Створення тренувального конвеєра включає кілька послідовних кроків обробки даних та їх класифікації. Спочатку до вже створеної послідовності обробки даних, яка включає перетворення статусів і агрегацію характеристик, додається алгоритм класифікації. Цей алгоритм тренує модель на основі підготовлених характеристик і вихідних міток для розпізнавання категорій статусів.

Після тренування моделі, конвеєр завершується застосуванням функції перетворення, яка перекладає ключі, отримані з алгоритму класифікації, назад у зрозумілі значення міток. Це дозволяє представити прогнози моделі у формі, яка безпосередньо відображає категорії статусів, роблячи результати зрозумілими для користувачів і придатними для подальшого аналізу та звітності.

На наступному кроці відбувається тренування моделі при якому застосовується попередньо визначений потік обробки даних та алгоритм навчання до завантаженого із файлу набору (рис. 3.19).

```
// Тренування моделі
trainedModel = trainingPipeline.Fit(trainingDataView);
```

Рисунок 3.19 Тренування моделі

Процес тренування моделі включає аналіз цих даних для вивчення залежностей між вхідними характеристиками та вихідними категоріями статусів.

Результатом тренування є налагоджена модель, здатна точно прогнозувати статуси на основі вхідних даних, що підвищує ефективність моніторингу в реальному часі.

Після цього відбувається процес оцінки моделі в результаті навчання та виведення її метрик у відповідне поле (рис. 3.20).

```
// Оцінка моделі
var predictions = trainedModel.Transform(trainingDataView);
var metrics = mlContext.MulticlassClassification.Evaluate(predictions);
ReportTBox.Text += ($"MacroAccuracy: {metrics.MacroAccuracy}\r\n");
ReportTBox.Text += ($" MicroAccuracy: {metrics.MicroAccuracy}\r\n");
ReportTBox.Text += ($"LogLoss: {metrics.LogLoss}\r\n");
```

Рисунок 3.20 Оцінка моделі та визначення метрик

За допомогою вбудованих функцій оцінки, проводиться аналіз якості класифікації, що включає вимірювання таких параметрів як макро- і мікроточність, а також LogLoss, що вказує на загальну надійність моделі. Макроточність враховує середню точність по всіх класах, в той час як мікроточність розглядає точність всіх класифікацій як єдиний набір даних, що забезпечує більш універсальний показник точності. LogLoss вимірює точність прогнозів моделі в термінах їх імовірності.

Для збереження моделі реалізовано подію «AddBtn_Click», код якої представлено на рис. 3.21.

```
private void AddBtn_Click(object sender, EventArgs e) {
    if (IsDataEnteringCorrect()) {
        //Save model
        string pathName = @"teach\" + GenerateFileName() + ".zip";
        string localProj =
            System.IO.Path.GetDirectoryName(System.Reflection.Assembly.GetExecutingAssembly().Location);
        _ModelsProvider.InsertModels(ModelsNamesTBox.Text,
            Convert.ToInt32(ScenariosCBox.SelectedValue), pathName);
        mlContext.Model.Save(trainedModel, trainingDataView.Schema, localProj + pathName);
        ClearAllData();
        _LogsProvider.InsertLogs(LoginForm.CurrentUser.UsersId,
            "Було навчено модель " +
            ModelsNamesTBox.Text, DateTime.Now);
        MessageBox.Show("Дані успішно збережено!");
    }
}
```

Рисунок 3.21 Метод збереження даних моделі

У розробленій системі, код кнопки додавання виконує кілька ключових функцій, зокрема перевірку коректності введення даних, збереження навченої

моделі та логування дій користувача. Якщо дані, що вводяться, є коректними, процес починається зі створення шляху для збереження файлу моделі, який включає генерацію назви файлу та додавання розширення .zip.

Далі, визначається директорія локального проекту, взята з місцезнаходження виконуваного файлу програми. Модель разом із її схемою даних зберігається у зазначеному місці. Після цього, проводиться очищення усіх введених даних через інтерфейс.

Важливим аспектом є ведення журналу дій, де реєструються зміни стану моделі та ідентифікатор користувача, який виконав навчання. Це логування включає час і деталі дії, що дає змогу відстежувати важливі події в системі. Врешті-решт, користувач отримує сповіщення через спливаюче вікно, що повідомляє про успішне збереження даних, завершуючи процес додавання нової моделі.

Для моніторингу статусів вантажного транспорту в процесі збору врожаю було реалізовано форму «Моніторинг», зовнішній вигляд якої представлено на рис. 3.22.

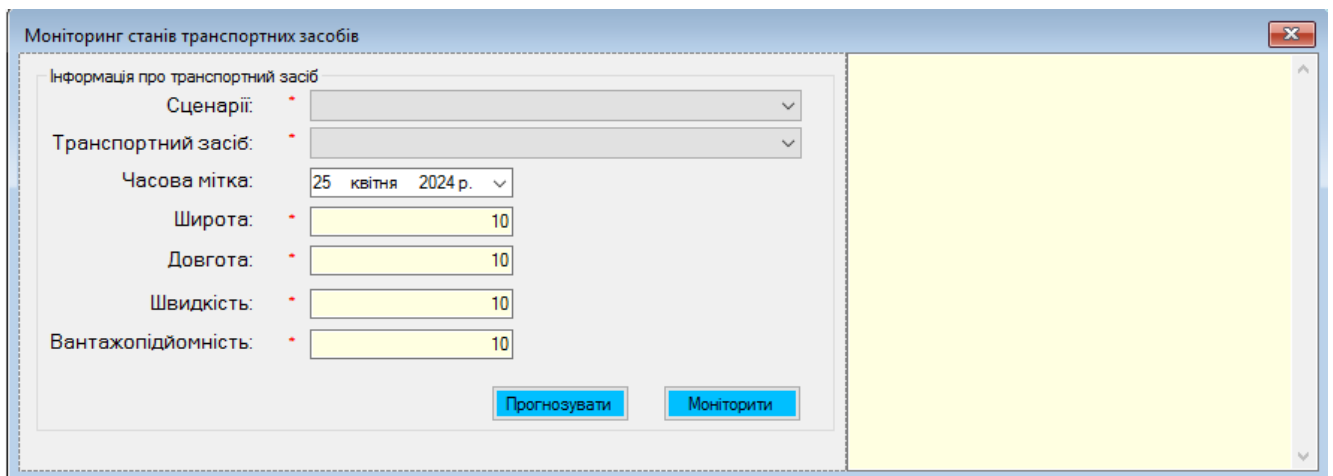


Рисунок 3.22 Вигляд форми «Моніторинг»

Для зміни моделі яка прогнозує статуси моніторингу транспорту в процесі збору врожаю у випадяючому списку, реалізовано подію, код якої показано на рис. 3.23.

```
private void ScenariosCBox_SelectedValueChanged(object sender, EventArgs e) {
    if (!_IsThemesLoad && _ScenariosList[0].Message != NamesMy.NoDataNames.NoDataInScenarios) {
        _SelectedNeural = _NeuralProvider.SelectedNeuralByScenariosId(
            Convert.ToInt32(ScenariosCBox.SelectedValue));
        LoadData(_SelectedNeural.NeuralFileModel);
    }
}
```

Рисунок 3.23 Код події зміни сценарію прогнозування статусів

Код для обробника події зміни вибраного значення у випадяючому списку сценаріїв виконує перевірку на завантаження сценаріїв. Якщо сценарії вже були завантажені, код продовжує з вибором моделей, що відповідають обраному ідентифікатору сценарію. Цей ідентифікатор перетворюється з вибраного значення у випадяючому списку до цілого числа, яке використовується для отримання відповідних моделей. Далі, отримані моделі передаються у метод завантаження даних. Цей метод відповідає за завантаження файлу, асоційованого з моделями, що були вибрані. Завантаження файлу дозволяє подальше використання даних моделі в системі для відображення деталей чи інших операцій, залежно від потреб користувача. Таким чином, код дозволяє динамічно взаємодіяти з користувачем та адаптувати інтерфейс і доступні опції згідно з вибраними сценаріями.

Для реалізації процесу моніторингу статусів обраною моделлю або зупинки в залежності від її поточного стану розроблено подію «RunBtn_Click» (рис. 3.24).

```
private void GetBtn_Click(object sender, EventArgs e) {
    if (IsScenariosSelectedCorrect()) {
        if (timer1.Enabled) {
            timer1.Enabled = false;
            GenBtn.Text = "Генерувати";
        } else {
            timer1.Enabled = true;
            GenBtn.Text = "Зупинити";
        }
    }
}
```

Рисунок 3.24 Код події «RunBtn_Click»

У системі моніторингу, код кнопки «Моніторити» забезпечує управління таймером на основі вибраних сценаріїв. Починаючи з перевірки правильності вибору сценаріїв, цей код дозволяє користувачу запускати або зупиняти процес

генерації даних через таймер, змінюючи текст на кнопці для відображення поточного стану таймера.

Коли користувач натискає кнопку, система перевіряє, чи правильно вибрані сценарії. Якщо вибір є коректним, далі йде перевірка стану таймера. Якщо таймер уже активовано, він буде деактивований, і текст на кнопці зміниться на «Моніторити», що сигналізує про можливість знову запустити генерацію даних. Якщо таймер не активовано, він запускається, і текст на кнопці змінюється на «Зупинити», що інформує користувача про те, що процес генерації в дії і його можна зупинити. Це динамічно змінює поведінку системи залежно від потреб користувача, дозволяючи легко керувати процесом моніторингу статусів.

Також реалізовано метод «PredictBtn_Click», що дозволяє тестувати моделі за допомогою введення даних диспетчером (рис. 3.25).

```

private void PredictBtn_Click(object sender, EventArgs e) {
    if (IsAllTransportModelCorrect()) {
        var prediction = predictionEngine.Predict(new TransportModel {
            DateTime = (float)DateTimeDTP.Value.ToOADate(),
            VehicleId = Convert.ToInt32(VehicleCBox.SelectedValue),
            Latitude = (float)(Convert.ToDouble(LatitudeTBox.Text)),
            Longitude = (float)(Convert.ToDouble(LongitudeTBox.Text)),
            Speed = (float)(Convert.ToDouble(SpeedTBox.Text)),
            LoadCapacity = (float)(Convert.ToDouble(LoadCapacityTBox.Text))
        });
        var vehicleInfo = new StringBuilder();
        // Прогноз до виводу
        vehicleInfo.AppendLine("\r\n--- Прогноз ---");
        vehicleInfo.AppendLine($"Статус вантажного засобу: {prediction.PredictedStatus}");
        RaportTBox.Text = vehicleInfo.ToString();
    }
}

```

Рисунок 3.25 Метод тестування модель за допомогою введених даних

У системі моніторингу, код кнопки «Прогнозувати» виконує функцію прогнозування статусу транспортного засобу на основі введених даних. Перш за все, система перевіряє, чи всі поля моделі транспорту заповнені коректно. Якщо дані є валідними, код ініціює процес прогнозування.

Для цього створюється новий екземпляр моделі транспорту, де вказуються необхідні характеристики, такі як дата і час, ідентифікатор транспортного засобу, географічні координати широти та довготи, швидкість руху та вантажопідйомність.

Ці дані конвертуються у відповідні формати та передаються в механізм прогнозування.

Після обробки введених даних, механізм генерує прогноз, який відображає передбачуваний статус вантажного засобу. Цей прогноз потім виводиться у текстове поле інтерфейсу. Для цього використовується конструктор текстових рядків, що дозволяє зібрати інформацію про прогноз у зрозумілу форму. Результатом є детальний опис статусу транспортного засобу, що відображається користувачу, дозволяючи зрозуміти поточний стан та прийняти відповідні рішення з управління або взаємодії з транспортом.

3.5 Модульне тестування системи

В рамках цього проекту особлива увага була приділена юніт-тестуванню, яке дозволяє ефективно перевіряти функціональність окремих компонентів програми. Використання бібліотеки MSTest допомогло розробити та реалізувати тести, які оцінювали ключові аспекти системи. Тестування бізнес-логіки включало розробку сценаріїв для перевірки алгоритмів, таких як визначення оптимальних маршрутів, оцінка часу паркування та калькуляція вартості послуг, гарантуючи, що важливі функції відповідають стандартам та вимогам. Було також здійснено тестування механізмів доступу до даних для перевірки їх надійності у збереженні та обробці інформації.

На рис. 3.26 представлено результати цих модульних тестів, що підтверджують ефективність тестування у забезпеченні якості та безпеки програмного продукту.

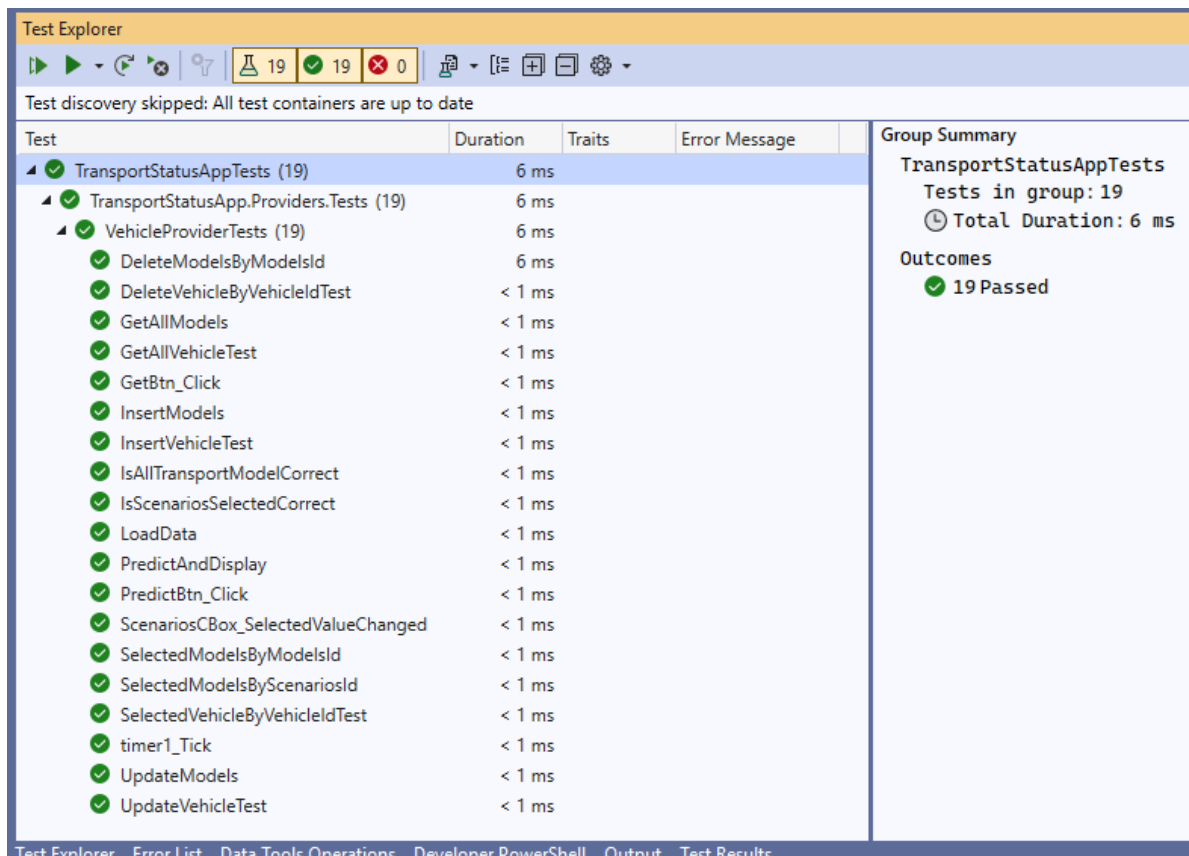


Рисунок 3.26 Результати модульного тестування

Звіти проведеного тестування, які відображено на відповідному рисунку, ілюструють рівень покриття тестами критичних сегментів системи та ефективність виявлення потенційних вразливостей та помилок.

3.6 Тестові сценарії та оцінка результатів розробки системи

Процес збору та формування тренувального файлу для системи моніторингу стану вантажного транспорту включає кілька етапів збору даних та їх структурування в один консолідований файл. Спочатку дані збираються з різних джерел, таких як GPS-трекери на транспортних засобах, системи управління флотом, а також оперативні системи, які забезпечують інформацію про статус та вантажопідйомність кожного транспортного засобу.

Ці дані містять інформацію про дату та час (DateTime), ідентифікатор вантажного транспортного засобу (VehicleId), географічні координати (Latitude та

Longitude), швидкість (Speed), поточний статус вантажного транспорту (Status) та його вантажопідйомність (LoadCapacity). Ці параметри відіграють важливу роль у тренуванні моделей для прогнозування поточних та майбутніх станів транспорту.

Далі, зібрані дані піддаються процесу очищення та нормалізації. Це включає виправлення помилок у координатах, видалення аномальних значень швидкості та коригування ідентифікаторів транспортних засобів. Також здійснюється перетворення дат та часів до формату, сумісного з аналітичними інструментами.

Після процесу очищення, дані інтегруються у тренувальний файл, що структуровано у формі таблиці з вказаними колонками. Кожен рядок у файлі представляє окремий запис про стан транспортного засобу в конкретний момент часу. Формат файлу зазвичай є CSV, що забезпечує легку інтеграцію з багатьма платформами машинного навчання.

На рис. 3.27 представлено фрагмент вікна із підготовленими даними для проведення тренування моделі машинного навчання.

```

DateTime,VehicleId,Latitude,Longitude,Speed,Status,LoadCapacity
45397,42,66,18.48855,-39.15989,5,В процесі завантаження,100
45400,7,84,75.63693,-115.0996,1,В процесі завантаження,800
45406,96,34,-73.2293,102.8441,7,В очікуванні ремонту,500
45399,95,15,-25.16049,128.6338,0,На стоянці,3100
45406,57,68,36.44734,-39.56709,1,В очікуванні ремонту,1500
45406,5,23,7.883798,-155.268,5,В режимі очікування,100
45407,74,97,-12.83452,-166.4613,9,У ремонті,1800
45405,45,56,-62.92709,144.2385,6,В процесі розвантаження,2300
45400,05,80,-6.812085,11.47862,3,В процесі завантаження,3500
45409,38,55,14.73673,-153.8195,54,В русі,3700
45401,72,96,-10.55457,-160.0123,0,На перевірці,3300
45398,5,23,-75.29176,91.07745,6,В процесі розвантаження,3400
45398,91,84,-47.68406,115.2593,9,На перевірці,900
45404,64,56,1.535314,-3.509701,0,В очікуванні ремонту,3500

```

Рисунок 3.27 Фрагмент вікна із підготовленими даними для навчання моделі

Для тренування моделі прогнозування статусів вантажного транспорту в процесі збору врожаю у застосунку було створено сценарій для навчання із назвою «Сценарій №1 (статусу збору врожаю)» (рис. 3.28).

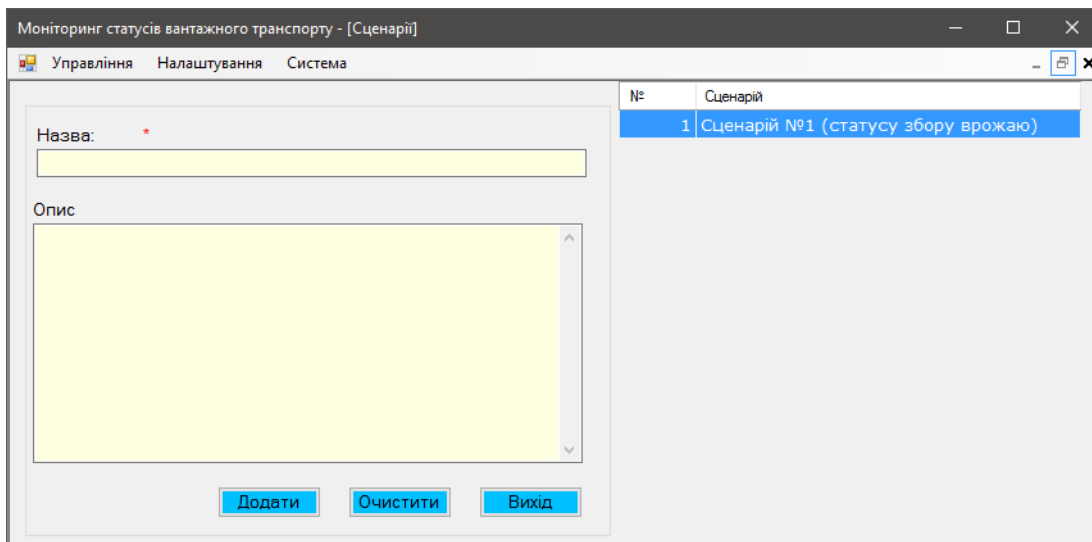


Рисунок 3.28 Створення сценарію прогнозування

Для тренування нової моделі створеного сценарію згідно підготовленого датасету, було здійснено перехід по меню програми «Довідники» → «Тренування моделей» та обрано сценарій «Сценарій №1 (статусу збору врожаю)». Після обрання підготовленого датасету у діалоговому вікні навчання моделі почалось автоматично. Результат навчання представлено на рис. 3.29.

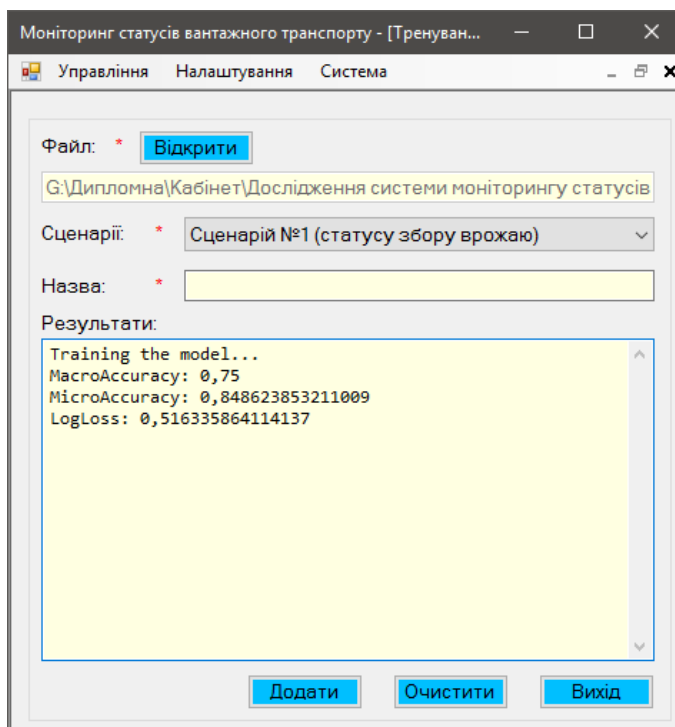


Рисунок 3.29 Результат навчання моделі

Процес навчання моделі супроводжувався повідомленням «Training the model...» та виведенням значень метрик навчання:

– MacroAccuracy (Макроточність) Макроточність є середньою точністю по всім класам, при цьому кожен клас вважається однаково важливим, незалежно від його частоти у навчальному наборі. Значення в 0.75 свідчить, що модель в середньому правильно класифікує 75% випадків для кожного класу. Це доволі високий показник, який вказує на здатність моделі добре працювати з різними категоріями даних;

– MicroAccuracy (Мікроточність). Мікроточність вимірює точність класифікації по всім класам взагалі, враховуючи загальну кількість вірних прогнозів. Значення близько 0.85 є дуже хорошим і вказує на високий загальний рівень точності моделі у класифікації екземплярів, що відповідає високій ефективності моделі у вирішенні поставленої задачі;

– LogLoss (Логарифмічні втрати). LogLoss вимірює надійність прогнозів моделі, оцінюючи, наскільки впевнено модель передбачає кожен клас. Ця метрика пеналізує помилкову впевненість, тому нижчі значення є кращими. Значення 0.516 є порівняно низьким, що свідчить про те, що модель, в більшості випадків, дає достовірні прогнози. Це вказує на ефективність моделі у виробленні прогнозів із хорошою ймовірнісною оцінкою.

Загалом, модель демонструє високу ефективність з міцною макро- та мікроточністю та низьким LogLoss. Це вказує на те, що модель навчена вирішувати поставлену задачу з високим ступенем точності, а її прогнози є надійними. Отже, можна використовувати цю модель для подальших застосувань, пов'язаних із класифікацією у вашій доменій області.

Після цього навчену модель було збережено у системі, для проведення експериментального дослідження (рис. 3.30).

№	Назва моделі	Файл	Видалити
1	Модель	\teach\2024_4_21_13_8_19.zip	Видалити

Рисунок 3.30 Збереження моделі в системі

Для проведення моніторингу статусів вантажного транспорту в процесі збору врожаю за допомогою збереженої моделі необхідно здійснити перехід по меню програми «Управління» – > «Моніторинг». У формі, що відкриється можна проводити моніторинг даних зібраних контролером, що встановлений на транспортному засобі та прогнозування статусів за допомогою натренованої моделі.

Для виявлення точності навченої моделі було згенеровано декілька потенційних сценаріїв за допомогою розробленого генератора сценаріїв, який імітує зібрані із контролерів дані та проаналізовано їх. На рис. 3.31 зображено перший згенерований сценарій.

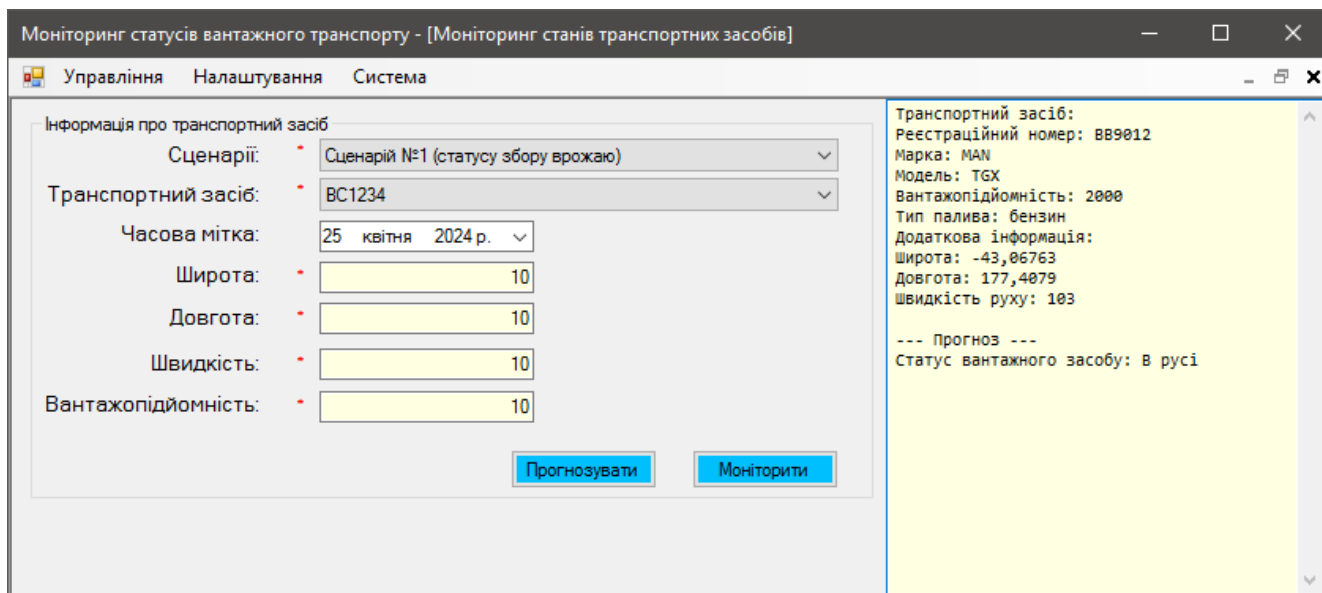


Рисунок 3.31 Результат генерації 1-го сценарію

Аналізуючи наданий сценарій, можна виділити кілька ключових аспектів:

- основні характеристики транспортного засобу. Дані включають марку MAN, модель TGX, і вантажопідйомність 2000 кг, що свідчить про значний потенціал для перевезення вантажів. Транспортний засіб використовує бензин як джерело палива;
- геопозиціонування та швидкість. Вказані координати (широта - 43.06763 і довгота 177.4079) разом із швидкістю руху 103 км/год підтверджують, що транспорт знаходиться у динамічному стані, ймовірно, на трасі або шосе;

– прогноз статусу. Вантажний засіб передбачувано знаходиться «В русі». Це статус відображає активний пересувний режим, який є очікуваним при зазначеній швидкості, і вказує на те, що транспорт виконує заплановані перевезення в рамках збору врожаю;

– оцінка точності прогнозу. Зазначена швидкість та геопозиціонування підтверджують адекватність прогнозу статусу «В русі». Враховуючи дані про марку, модель, та вантажопідйомність, прогноз виявляється логічним та відповідає очікуванням щодо поведінки вантажного транспорту в процесі збору врожаю.

Загалом, сценарій демонструє точність і релевантність моделі прогнозування, що здатна коректно аналізувати та інтерпретувати вхідні дані для визначення актуального стану вантажного транспорту. Це підтверджує ефективність системи в контексті реальних транспортних операцій.

На рис. 3.32 зображено скріншот результат виконання 2-го сценарію.

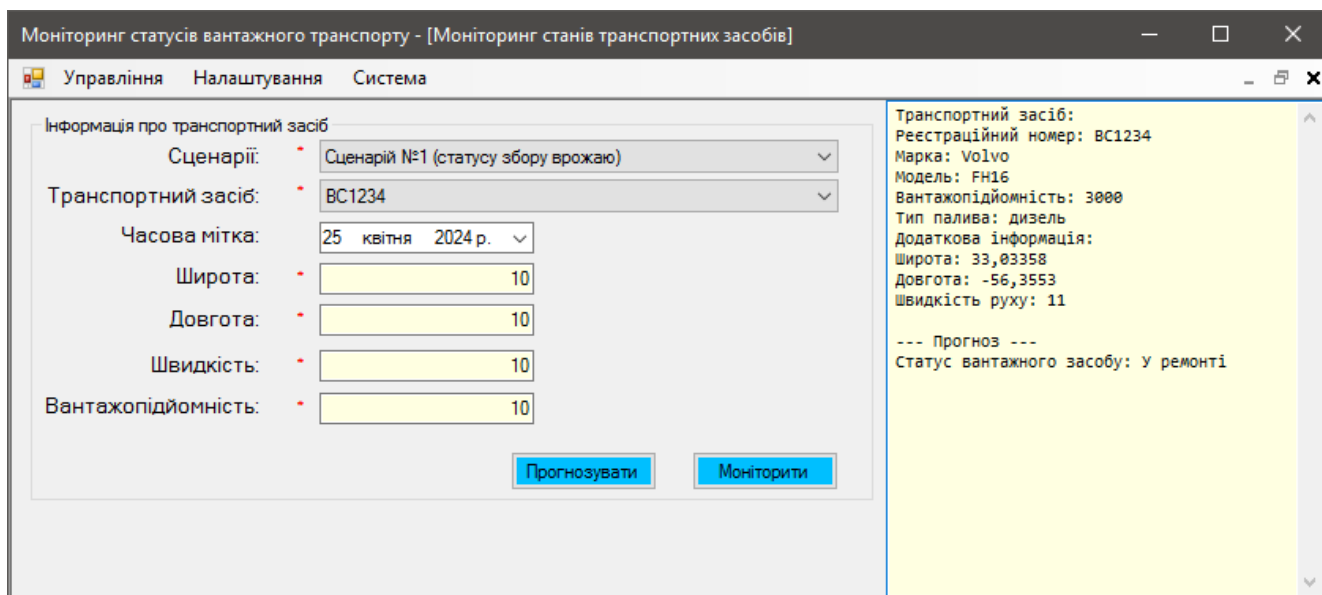


Рисунок 3.32 Результат генерації 2-го сценарію

Аналізуючи наданий сценарій із згенерованими даними, можна зробити наступні висновки:

– основні характеристики транспортного засобу: Дані містять інформацію про вантажівку марки Volvo, модель FH16, з вантажопідйомністю 3000 кг. Це вказує на потужний та великий вантажний транспорт, призначений для

перевезення великих вантажів. Використання дизельного палива є характерним для вантажних автомобілів, що потребують великої тягової сили;

- геопозиціонування та швидкість. Координати широти 33.03358 та довготи -56.3553, разом із швидкістю руху 11 км/год, свідчать про дуже повільний рух або зупинку транспортного засобу. Це може бути пов'язано із заторами, зупинкою на контрольних пунктах або технічними проблемами;

- прогноз статусу. Вантажний засіб передбачувано знаходиться «У ремонті». Цей статус може відображати ситуацію, коли транспорт зупинений для технічного обслуговування або через виникнення несправностей, що вимагають ремонту.

- оцінка точності прогнозу. Низька швидкість руху транспортного засобу підтримує гіпотезу про його поточний технічний стан, який потребує ремонту або обслуговування. Хоча конкретних доказів того, що зупинка пов'язана саме з ремонтом, немає, прогноз здається логічним у контексті повільного руху.

Загалом, прогноз «У ремонті» для вантажного транспорту, що рухається зі швидкістю 11 км/год, є вірогідним і має раціональне обґрунтування на основі наданих даних. Це демонструє ефективність системи у виявленні та реагуванні на потенційні технічні проблеми транспортних засобів в процесі їх експлуатації.

Отже, розроблена система прогнозування статусу вантажного транспорту, що використовує дані про місцезнаходження, швидкість та інші параметри транспортних засобів, може стати цінним інструментом для оптимізації логістики та забезпечення безперебійної роботи транспортних потоків під час збору врожаю

У рамках даного розділу була розроблена система моніторингу статусів вантажного транспорту для процесу збору врожаю, яка забезпечує ефективний збір, передачу, обробку та аналіз даних у реальному часі. Архітектура системи базується на використанні контролерів Arduino Leonardo з підключенням до різноманітних датчиків, включаючи GPS-модуль NEO-6M для точного відстеження місцезнаходження, модуль часу DS3231 для фіксації часу подій, датчик тиску HX711 для вимірювання ваги вантажу, та датчик швидкості LM393 для реєстрації швидкості руху. Вся інформація передається через модуль SIM800L до центральної

бази даних MS SQL Server, де вона обробляється та відображається у системі. Система сповіщень в реальному часі активно інформує диспетчерів про будь-які незвичайні події або відхилення від норми, що дозволяє оперативно реагувати на потенційні проблеми. Код для керування контролером було розроблено для забезпечення точного та надійного збору даних. Інтеграція програмного забезпечення із базою даних здійснена з використанням сучасних, що забезпечує надійність та ефективність обміну даними. Процес модульного тестування виконаний за допомогою MSTest, що підтверджує високу якість та працездатність розробленої системи. Тестові сценарії, проаналізовані в ході оцінки результатів, демонструють надійність і точність системи у реальних умовах експлуатації, підтверджуючи її здатність точно відстежувати та аналізувати стан вантажного транспорту.

ВИСНОВКИ

Кваліфікаційна робота зосереджена на розробці імітаційної моделі системи моніторингу статусів вантажного транспорту, призначеної для застосування в агропромисловому комплексі. Вона передбачає створення віртуальної платформи, яка демонструє можливості збору, обробки, аналізу та представлення даних від транспортних засобів, використовуваних у сільськогосподарській логістиці. Робота надає імітаційний вигляд реальних викликів, з якими стикаються оператори вантажного транспорту під час збору врожаю, і використовує теоретичні дані для моделювання робочих процесів. Ця імітаційна модель допомагає оцінити ефективність потенційних систем управління та моніторингу, забезпечуючи основу для розробки реальних систем, спрямованих на підвищення продуктивності та ефективності логістичних операцій в агросекторі.

У першому розділі була виконана детальна розробка теоретичних основ моніторингу вантажного транспорту, оглянуто сучасні методи, такі як GPS-моніторинг, телематика, RFID-технології, інтегровані системи управління логістикою, а також застосування дронів. Також було проаналізовано використання GPS та інших навігаційних систем у логістиці вантажних перевезень, вивчено особливості та можливості їх використання для збору врожаю.

В другому розділі проведено аналіз та проектування системи моніторингу. Було сформульовано вимоги до системи, вибрано технологічний стек для розробки, який включає C# і MS SQL Server, та розроблено ключові бізнес-процеси з допомогою діаграм активності та послідовності. Проектування бази даних і архітектури системи було виконано з урахуванням всіх визначених вимог і стандартів.

Третій розділ був присвячений розробці та реалізації системи моніторингу. Була розроблена загальна архітектура системи, забезпечуючи ефективний збір, передачу, обробку та аналіз даних. Використання контролерів Arduino Leonardo та різноманітних датчиків дозволило точно відстежувати параметри вантажних

транспортних засобів. Також були розроблені та інтегровані модулі системи, проведено модульне тестування за допомогою MSTest, що підтвердило її працездатність.

Розробка системи для моніторингу статусів вантажного транспорту в агропромисловому комплексі відкриває значні перспективи для підвищення ефективності та продуктивності в сільськогосподарській логістиці. Основна користь, яку може принести такий застосунок, охоплює кілька важливих аспектів:

- оптимізація логістики. Система дозволяє точно відстежувати місцезнаходження та стан транспортних засобів, що сприяє оптимізації маршрутів та зниженню часу доставки. Таке управління може значно знижувати логістичні витрати та підвищувати загальну ефективність транспортних операцій;

- покращення реагування на непередбачені обставини. Швидке виявлення та реагування на проблеми, такі як затримки через дорожні умови або технічні несправності, забезпечують менші простои та більшу надійність служби доставки;

- збільшення продуктивності. Автоматизація збору даних та їх аналіз сприяє кращому розумінню потреб флоту та плануванню обслуговування. Це допомагає підтримувати транспортні засоби в оптимальному стані, зменшує час простою та збільшує їх експлуатаційний період;

- збір та аналіз даних. Накопичення даних про використання транспортних засобів дозволяє вдосконалити аналітику, що може використовуватися для стратегічного планування та вдосконалення процесів;

- екологічні переваги. Ефективніше управління маршрутами та покращене планування можуть знижувати споживання палива та викиди вуглецю, сприяючи екологічній стійкості.

Таким чином, розробка цієї системи має потенціал не тільки підвищити ефективність операційних процесів, але й забезпечити значні стратегічні переваги агропромисловим підприємствам у їхній прагненні до технологічних інновацій і сталого розвитку.

ПЕРЕЛІК ПОСИЛАНЬ

1. Features of GIS application in agriculture and logistics. URL: https://www.researchgate.net/publication/346113967_Features_of_GIS_application_in_agriculture_and_logistics (дата звернення 21.04.2024).
2. Демченко Є. Б., Дорош А. С., Сковрон І. Я. Сучасні інформаційні системи на ринку вантажних перевезень України. Транспортні системи та технології перевезень, 2022, 23: 79с.
3. Фоменко В. Особливості GPS-моніторингу перевезення вантажу. Наука–виробництву, 2023.– 158с.
4. Похлебіна Т. І.; Лимаренко Ю. Л.; Урсуленко В. В. Використання цифрових технологій в логістичних системах. Збірник наукових праць Одеської державної академії технічного регулювання та якості, 2023. – 55с.
5. Зелінська О. В.; Волонтир Л. О. Інформаційно-логістичні системи управління аграрним підприємством. 2018.– 93с.
6. Синявська О. Б. Оптимізація вантажних перевезень за допомогою системи GPS-моніторингу – 2020. – 358 с.
7. Chmel Piotr, et al. Object-oriented Project Management Blok Traffic by Means of GPS Monitoring. Вісник Львівського державного університету безпеки життєдіяльності, 2013. – 102 с.
8. Schelter Sebastian, et al. Samsara: Declarative machine learning on distributed dataflow systems. In: NIPS Workshop MLSystems. 2016.– 6р.
9. Crossley John N. Samsara. In: Mathematical Problems from Applied Logic I: Logics for the XXIst Century. New York, NY: Springer New York, 2005. – 266 p.
10. Teletrac Navman Fleet Director Review URL: <https://www.teletracnavman.com/> (дата звернення 19.04.2024).
11. Teletrac Navman GPS & Telematics Integration URL: <https://www.fleetio.com/app-directory/fleet-telematics/teletrac-navman> (дата звернення 19.04.2024).

12. Fleet Tracking System: 5 Key Business Benefits - Teletrac Navman URL: <https://www.teletracnavman.com.au/fleet-management-software/fleet-tracking/resources/fleet-tracking-system-5-key-business-benefits> (дата звернення 20.04.2024).

13. Mix Telematics URL: <https://www.mixtelematics.com/za/> (дата звернення 20.04.2024).

14. Архітектура Mix Telematics. URL: <https://vehicletech.co.nz/products/mix-telematics> (дата звернення 20.04.2024).

15. MiX Telematics: Employee Benefits and Perks . URL: https://www.glassdoor.co.in/Benefits/MiX-Telematics-India-Benefits-EI_IE505835.0,14_IL.15,20_IN115.htm (дата звернення 20.04.2024).

16. Robinson Deanna. The Telematics Advanced Program: A Noble Idea Bites the Dust. Canadian Journal of Communication, 1999. URL: <https://cjc.utpjournals.press/doi/10.22230/cjc.1999v24n3a1113> (дата звернення 23.04.2024).

17. Функціональна модель системи. URL: <https://core.ac.uk/download/pdf/11320265.pdf> (дата звернення 23.04.2024).

18. Мокін, Б. І. Навчальний посібник для опанування студентами способів розв'язання задач з функціонального аналізу мовою Python. – Вінниця : ВНТУ, 2022. – 124 с.

19. Придатко О. В., Бурак Н. Є., Хлевной О. В. Основи програмування (мовою Java) : курс лекцій. – Львів : ЛДУ БЖД, 2019. – 180 с.

20. Ріхтер Д. CLR via C#. – К.: Видавнича група «Буква», 2018. – 896 с.

21. What is SQL Server?: веб-сайт. URL: https://www.tutorialspoint.com/ms_sql_server/ms_sql_server_overview.htm (дата звернення 20.04.2024).

22. Klimek, B., & Skublewska-Paszowska, M. (2021). Comparison of the performance of relational databases 39. PostgreSQL and MySQL for desktop application. Journal of Computer Sciences Institute. URL: <https://doi.org/10.35784/jcsi.2314> (дата звернення 23.04.2024).

23. Павловський В.І., Петрашенко А.В. Бази даних та засоби управління. Практикум. . – Київ : КПІ ім. Ігоря Сікорського, 2021. – 112 с.

24. Трирівнева архітектура. URL: <https://javarush.com/ua/quests/lectures/ua.questservlets.level14.lecture01> (дата звернення 23.04.2024).

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)

**Державний університет інформаційно-комунікаційних
технологій**

Кафедра Інженерії програмного забезпечення автоматизованих систем

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«Дослідження системи моніторингу статусів
вантажного транспорту в процесі збору врожаю»**

на здобуття освітнього ступеня магістра
зі спеціальності 126 Інформаційні системи та технології
освітньо-професійної програми Інформаційні системи та технології

Виконав: Дембровський О.С, ІСД-42

Науковий керівник роботи:

Миколайчук В.Р

Актуальність теми: потреба аграрного сектору України у підвищенні ефективності та оптимізації логістичних процесів.

Об`єкт дослідження: логістичні процеси в агропромисловому комплексі.

Предмет дослідження: система моніторингу статусів вантажного транспорту.

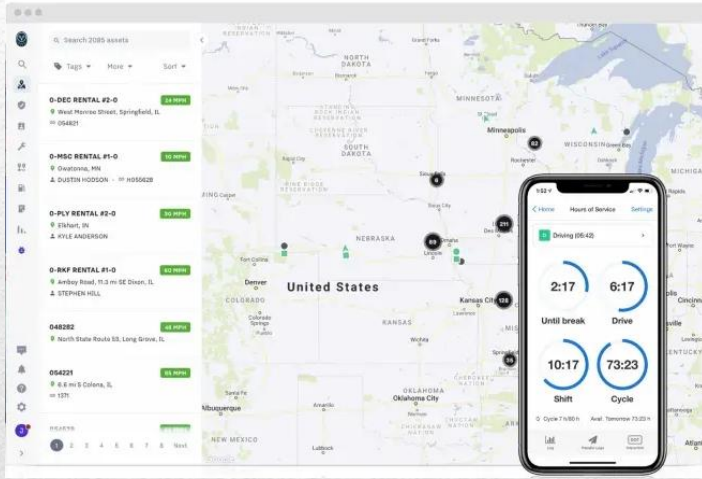
Мета дослідження: розробка системи моніторингу статусів вантажного транспорту, оптимізованої для потреб агропромислового комплексу під час збору врожаю.

Завдання дослідження:

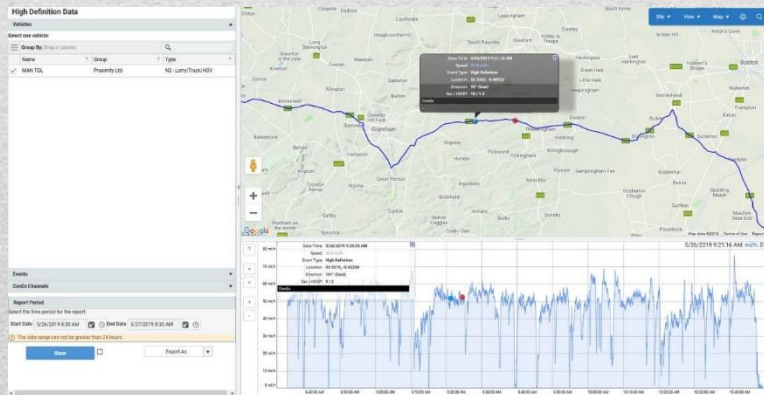
1. Вивчення теоретичних основ моніторингу вантажного транспорту в агропромисловому комплексі;
2. Розробка технічних вимог та проектування архітектури системи моніторингу;
3. Реалізація розробленої системи;
4. Аналіз отриманих результатів, оцінка ефективності системи моніторингу.

Аналіз аналогічних програмних рішень

Samsara



Teletac Navman



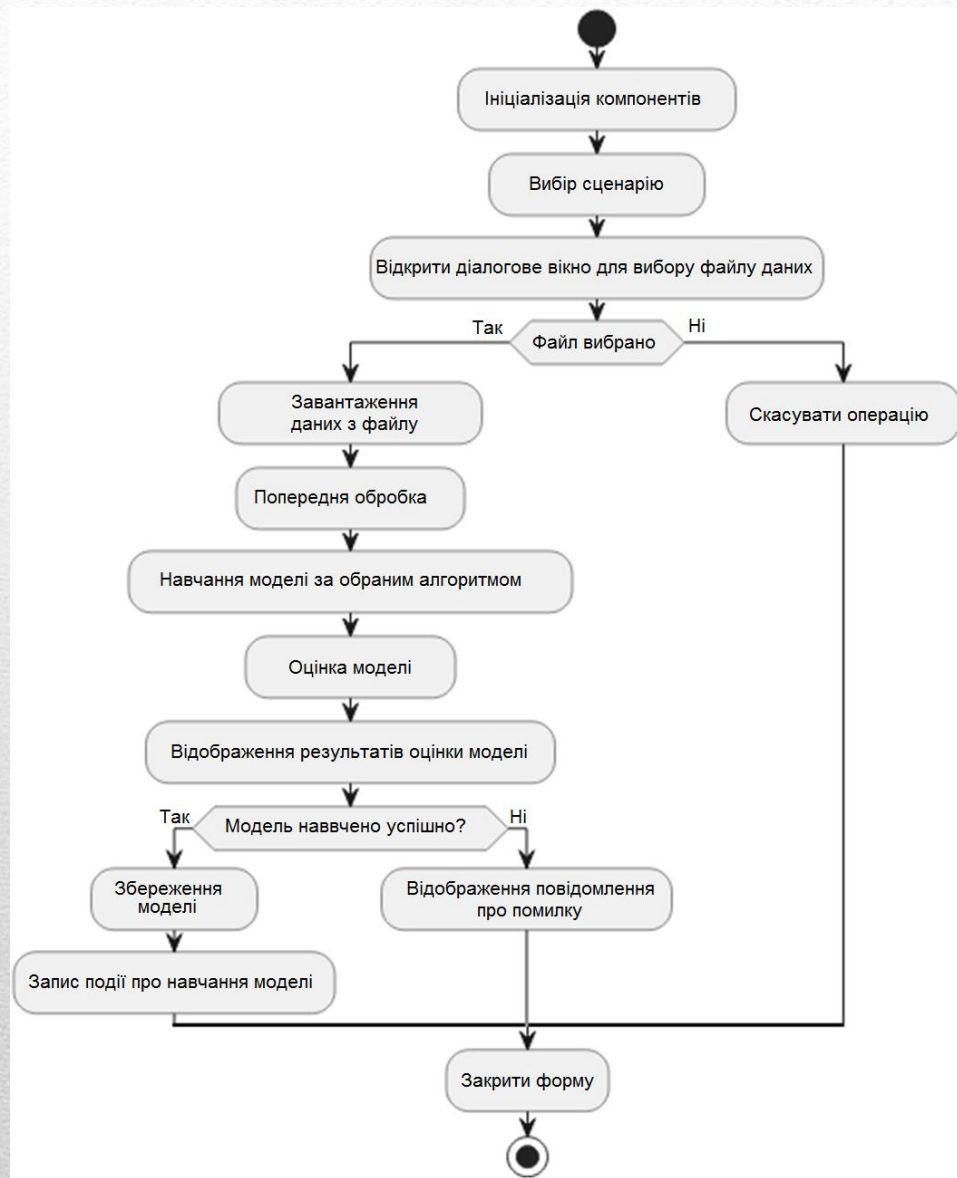
Mix Telematics



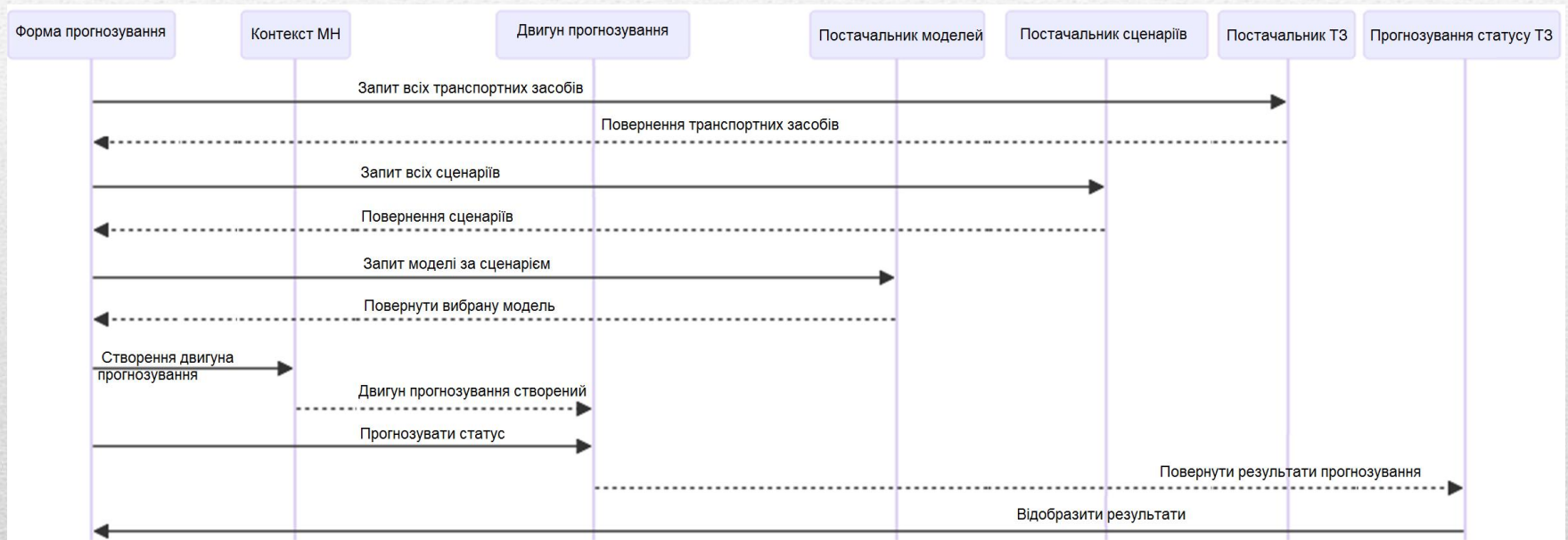
Діаграма прецедентів



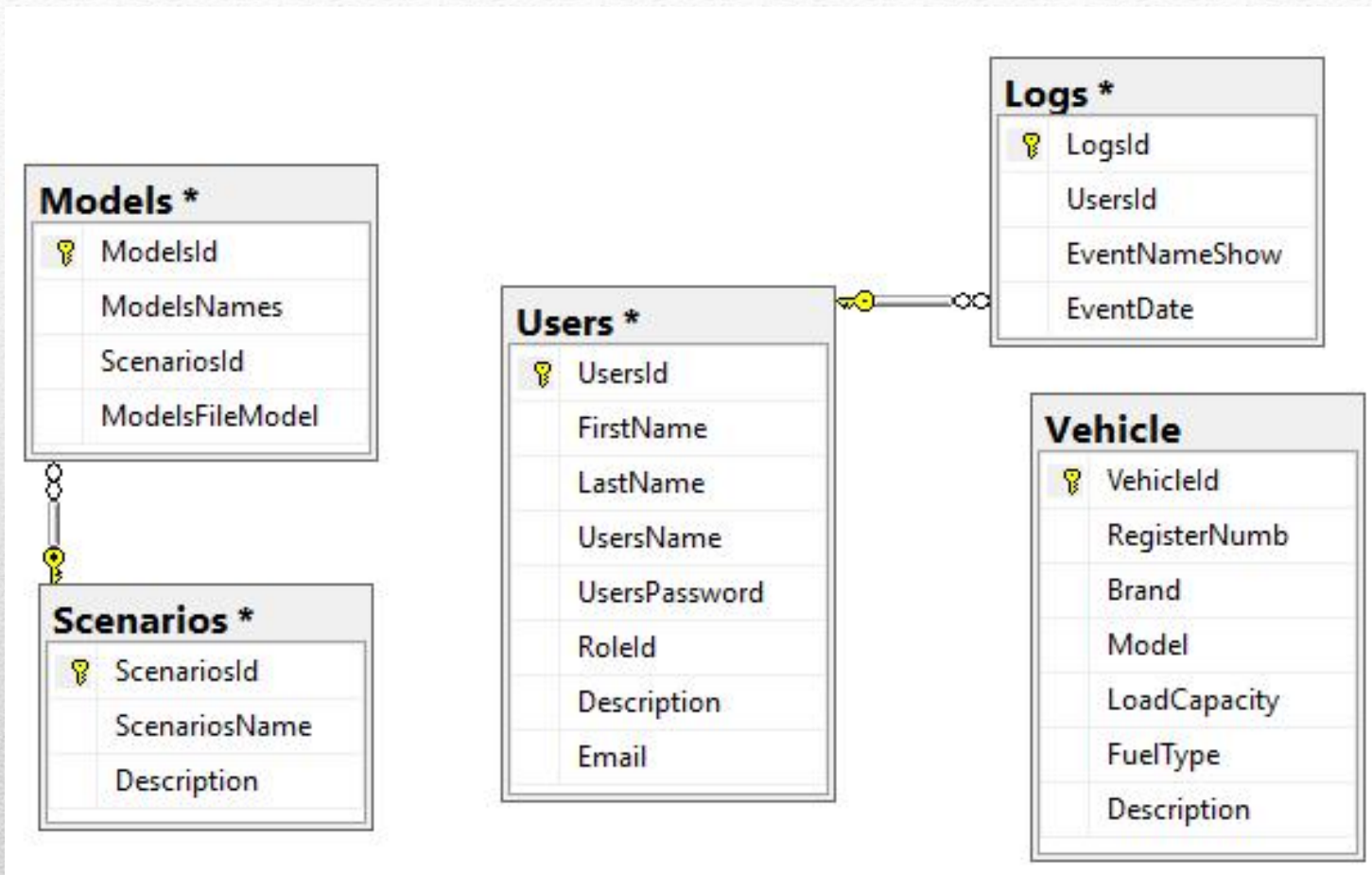
Процес навчання моделі для моніторингу та прогнозування статусів вантажного транспорту



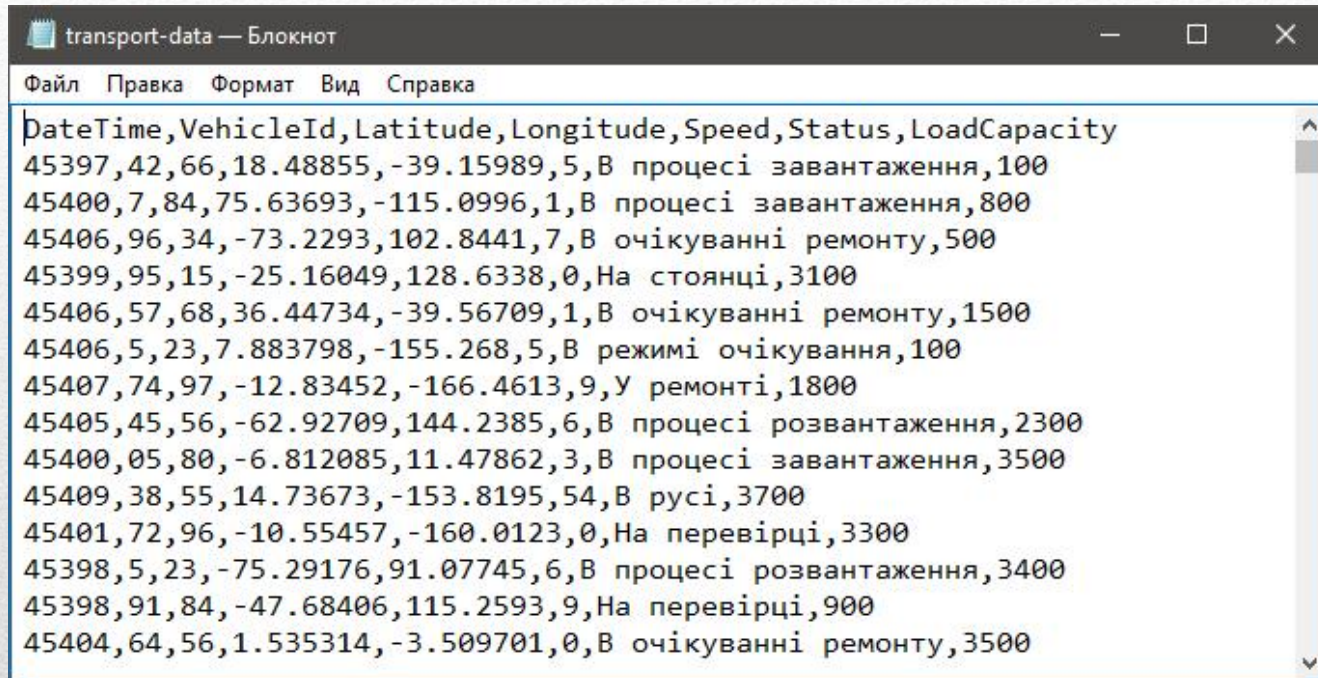
Моніторинг та прогнозування статусів вантажного транспорту



ERD діаграма бази даних

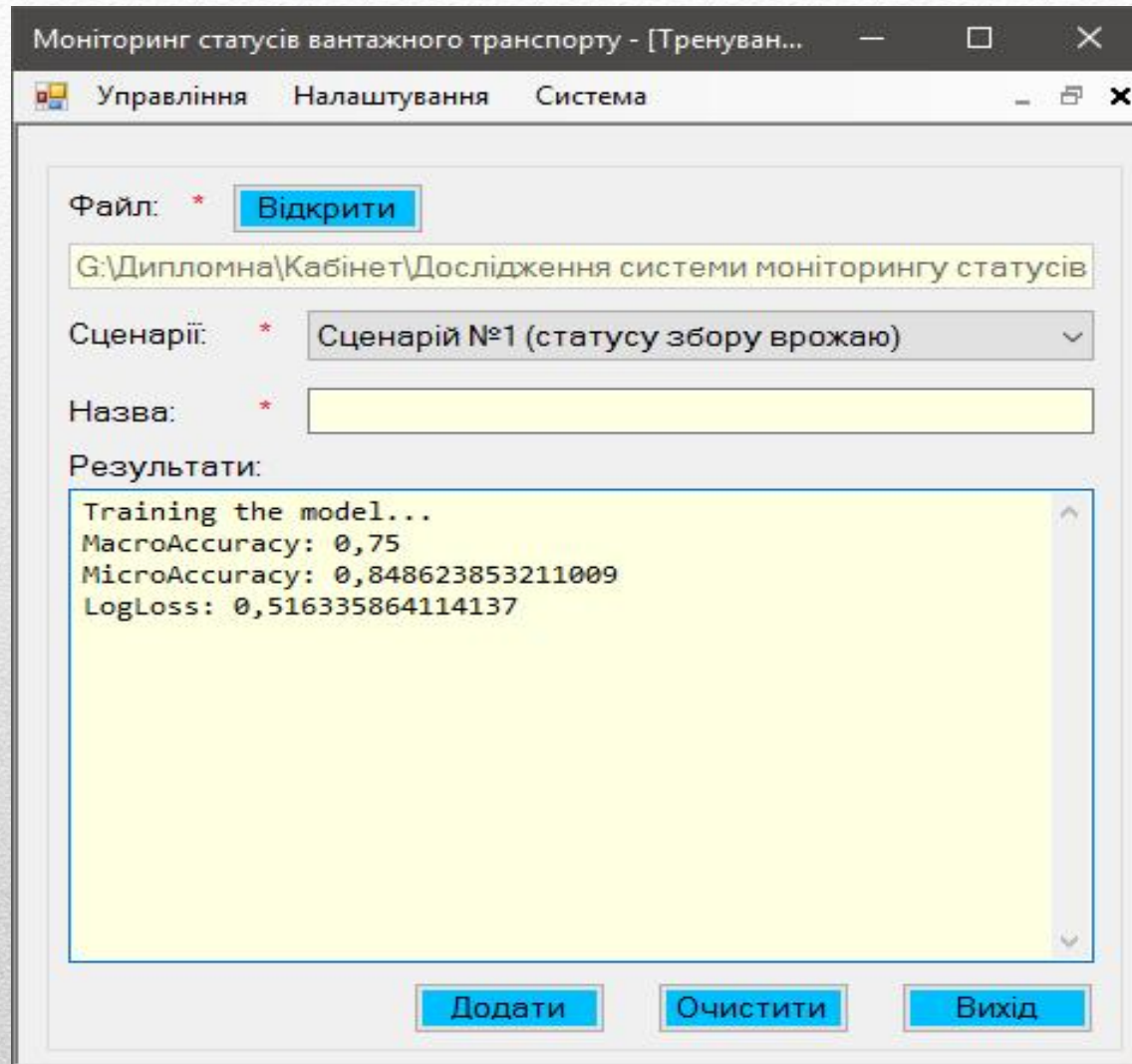


Підготовлені дані для навчання моделі



```
transport-data — Блокнот
Файл  Правка  Формат  Вид  Справка
DateTime,VehicleId,Latitude,Longitude,Speed,Status,LoadCapacity
45397,42,66,18.48855,-39.15989,5,В процесі завантаження,100
45400,7,84,75.63693,-115.0996,1,В процесі завантаження,800
45406,96,34,-73.2293,102.8441,7,В очікуванні ремонту,500
45399,95,15,-25.16049,128.6338,0,На стоянці,3100
45406,57,68,36.44734,-39.56709,1,В очікуванні ремонту,1500
45406,5,23,7.883798,-155.268,5,В режимі очікування,100
45407,74,97,-12.83452,-166.4613,9,У ремонті,1800
45405,45,56,-62.92709,144.2385,6,В процесі розвантаження,2300
45400,05,80,-6.812085,11.47862,3,В процесі завантаження,3500
45409,38,55,14.73673,-153.8195,54,В русі,3700
45401,72,96,-10.55457,-160.0123,0,На перевірці,3300
45398,5,23,-75.29176,91.07745,6,В процесі розвантаження,3400
45398,91,84,-47.68406,115.2593,9,На перевірці,900
45404,64,56,1.535314,-3.509701,0,В очікуванні ремонту,3500
```

Результат навчання моделі



Сценарії прогнозування

Моніторинг статусів вантажного транспорту - [Моніторинг станів транспортних засобів]

Управління Налаштування Система

Інформація про транспортний засіб

Сценарії: * Сценарій №1 (статусу збору врожаю) ▾

Транспортний засіб: * BC1234 ▾

Часова мітка: 18 травня 2024 р. ▾

Широта: *

Довгота: *

Швидкість: *

Вантажопідйомність: *

Транспортний засіб:
Реєстраційний номер: BV9012
Марка: MAN
Модель: TGX
Вантажопідйомність: 2000
Тип палива: бензин
Додаткова інформація:
Широта: -43.06763
Довгота: 177.4079
Швидкість руху: 103

--- Прогноз ---
Статус вантажного засобу: в русі

Моніторинг статусів вантажного транспорту - [Моніторинг станів транспортних засобів]

Управління Налаштування Система

Інформація про транспортний засіб

Сценарії: * Сценарій №1 (статусу збору врожаю) ▾

Транспортний засіб: * BC1234 ▾

Часова мітка: 18 травня 2024 р. ▾

Широта: *

Довгота: *

Швидкість: *

Вантажопідйомність: *

Транспортний засіб:
Реєстраційний номер: BC1234
Марка: Volvo
Модель: FH16
Вантажопідйомність: 3000
Тип палива: дизель
Додаткова інформація:
Широта: 33.03358
Довгота: -56.3553
Швидкість руху: 11

--- Прогноз ---
Статус вантажного засобу: у ремонті

Висновки

У кваліфікаційній роботі розроблено імітаційну модель системи моніторингу статусів вантажного транспорту для агропромислового комплексу. Робота включає створення віртуальної платформи для збору, обробки, аналізу та представлення даних транспортних засобів під час збору врожаю.

Система була розроблена з використанням C# та MS SQL Server, з інтеграцією контролерів Arduino Leonardo та різноманітних датчиків. Реалізовано ефективний збір та передача даних, модулі моніторингу та аналітики, а також проведено тестування для забезпечення стабільної роботи системи.