

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем та програмної інженерії
(повна назва факультету)

Кафедра програмної інженерії
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: Розробка системи відео-спостереження
для транспортних засобів з використанням штучного інтелекту
та технологій Java, Spring та AWS

Виконав: студент 6 курсу, групи СПм-62
спеціальності 121 «Інженерія програмного
забезпечення»

(шифр і назва спеціальності)

(підпис)

(прізвище та ініціали)

Керівник

(підпис)

Мудрик І.Я.

(прізвище та ініціали)

Нормоконтроль

(підпис)

.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Петрик М.Р.

(прізвище та ініціали)

Рецензент

(підпис)

.

(прізвище та ініціали)

РЕФЕРАТ

Кваліфікаційна робота магістра, виконана Май Артуром Романовичем, студентом Тернопільського національного технічного університету імені Івана Пулюя, присвячена дослідженню використання штучного інтелекту для розробки системи відеоспостереження для транспортних засобів з використанням технологій Java Spring та AWS. Робота має обсяг 60 сторінок, включає 21 рисунок, 4 додатки та бібліографію з 5 джерел.

Метою даної роботи є створення безпечної та ефективної системи відеоспостереження з використанням штучного інтелекту для громадського транспорту. У роботі використовуються такі технології розробки, як MySQL, Java, Spring, AWS, React, HTML, CSS, а також технології для тестування - Junit5.

Розроблювальний сервіс надає можливість користувачам отримувати віддалений доступ до камер в транспортних засобах, переглядати пряму трансляцію, записану історію, отримувати статистику подій, опрацьованих штучним інтелектом, а також для налаштування. Зазначені функціональні можливості сприяють покращенню безпеки та оптимізації роботи громадського транспорту.

Робота вносить важливий внесок у сферу розробки систем відеоспостереження, підкреслюючи значення використання штучного інтелекту для підвищення рівня безпеки та оптимізації громадського транспорту.

Ключові слова роботи: інтернет-сервіс, потокове відео, Java, СУБД, MySQL, AWS, AI, React, IntelliJ IDEA, RUP (Rational Unified Process).

SUMMARY

The master's attestation work, performed by Mai Artur Romanovych from Ternopil National Technical University named after Ivan Puluy, is dedicated to the study of the use of artificial intelligence for the development of a video surveillance system for vehicles using Java Spring and AWS technologies. The work has a volume of 60 pages, includes 21 figures, 4 appendices and a bibliography from 5 sources.

The purpose of the research is to create a safe and effective video surveillance system using artificial intelligence for public transport. The work uses such development technologies as MySQL, Java, Spring, AWS, React, HTML, CSS, as well as testing technologies - Junit5.

The developed service enables users to remotely access cameras in vehicles, view live broadcasts, recorded history, receive statistics of events processed by artificial intelligence, and also for customization. The specified functionality contributes to improving safety and optimizing the operation of public transport.

Keywords of work: Internet service, streaming video, Java, DBMS, MySQL, AWS, AI, React, IntelliJ IDEA, RUP (Rational Unified Process).

The work makes an important contribution to the development of video surveillance systems, emphasizing the importance of using artificial intelligence to improve the level of safety and optimize public transport.

ЗМІСТ

РЕФЕРАТ	2
SUMMARY	3
ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	7
1.1 Аналіз області	7
1.2 Обґрунтування вибору напрямку дослідження	10
1.3 Вибір технології.....	11
2 ПРОЄКТУВАННЯ МОДЕЛІ ТА РЕАЛІЗАЦІЇ ПРОГРАМНОГО КОМПЛЕКСУ	14
2.1 Сутності предметної області	15
2.2 Вибір інструментів розробки.....	17
2.3 Опис процесу проекту з методологію RUP.....	18
2.3.1 Аналіз і планування вимог за методологію RUP	19
2.3.2 Проєктування за методологію RUP	24
2.3.3 Реалізація за методологію RUP.....	25
2.3.4 Тестування за методологію RUP	25
2.4 Огляд та встановлення фреймворків тестування	26
2.4.1 Переваги та недоліки фреймворку	27
2.5 Архітектура програмного забезпечення.....	28
2.6 Взаємодія з СУБД.....	32
2.7 Впровадження сервісу розпізнавання Amazon Rekognition	33
2.7.1 Можливості і переваги Amazon Rekognition	33
2.7.2 Вибір моделі для Amazon Rekognition	34
2.7.3 Опис параметрів Amazon Rekognition.....	36
3 ВИКОРИСТАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	40
3.1 Системні вимоги	40
3.2 Використання системи в реальних умовах	40

4	ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	48
4.1	Охорона праці	48
4.2	Вимоги безпеки під час виконання роботи та вимоги до організації робочих місць користувачів комп'ютерів	48
4.3	Вимоги безпеки в аварійних ситуаціях	50
5	ВИСНОВКИ.....	52
6	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	53

ВСТУП

У цій роботі ми глибоко досліджуємо використання систем відеоспостереження, спеціально зосереджуючись на їх застосуванні в галузі громадського транспорту.

Аналіз широкого обсягу відеоданих, отриманих з камер, розташованих на транспортних засобах, дозволяє нам ефективно виявляти потенційно небезпечні ситуації, контролювати пасажирський потік та сприяти вирішенню завдань щодо поліпшення сервісу та збільшення прибутку громадського транспорту.

Інтеграція штучного інтелекту в системи відеоспостереження громадського транспорту є стратегічною ініціативою, спрямованою на створення більш безпечних та ефективних систем масового перевезення. Ця технологія відповідає вимогам сучасної міської інфраструктури, забезпечуючи зручність та безпеку для пасажирів.

Цей напрямок дослідження є надзвичайно актуальним, оскільки вирішення ключових завдань перед громадським транспортом стає дедалі важливішим. Підрахунок пасажирів та статистика користування допомагають оптимізувати маршрути, уникати переповнення та зменшувати час очікування. Штучний інтелект дозволяє ефективно виявляти небезпечні ситуації, сприяючи покращенню безпеки транспорту.

Ця технологія корисна не лише для операторів громадського транспорту, але й для міських влад, які можуть використовувати отримані дані для оптимізації інфраструктури та планування міського руху. Безпечні та ефективні системи транспорту сприятимуть відмінному громадському обслуговуванню та розвитку міських територій.

Майбутнє використання систем відеоспостереження в громадському транспорті передбачає постійний розвиток та вдосконалення технологій. Завдяки збільшенню обчислювальної потужності та розвитку алгоритмів штучного інтелекту, можна очікувати подальше покращення функціональності систем, забезпечуючи ще більшу точність відстеження та аналізу подій.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз області

В сучасному світі використання штучного інтелекту (ШІ) широко розгортається в різних сферах життя, включаючи транспортні технології. Однією з актуальних областей застосування ШІ є розробка систем відеоспостереження для транспортних засобів. Ця тема є особливо важливою в контексті підвищення безпеки, ефективності та управління громадським транспортом.

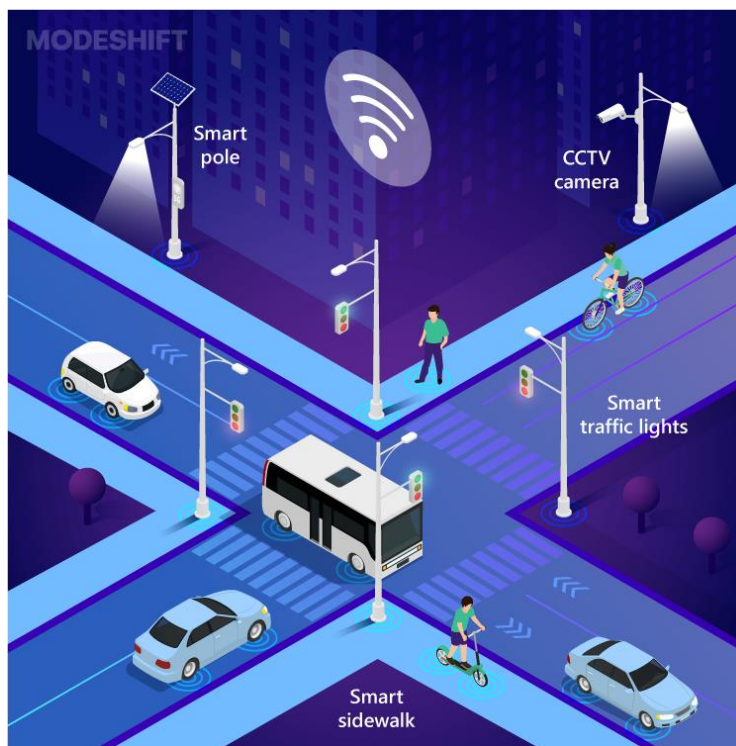


Рисунок 1.1 – Аналіз транспортних потоків

Завдяки поєднанню передових технологій комп'ютерного зору та нейронних мереж, системи відеоспостереження стають більш інтелектуальними та здатними ефективно аналізувати великі об'єми даних. Це відкриває широкі можливості для виявлення та вирішення проблем на дорозі, таких як аварії, затори та інші негативні ситуації.

Розумне міське планування стало пріоритетом для муніципалітетів і органів влади в усьому світі, а транспортні компанії також шукають ШІ як інструмент для вирішення типових проблем, пов'язаних з транспортом, а також для скорочення витрат. Моніторинг часу перебування транспортних засобів, порушень правил паркування та тенденцій щільності руху також допоможе в управлінні розумними містами.

Очікується, що до 2026 року світовий ринок штучного інтелекту на транспорті досягне 3,5 мільярдів доларів США[1]. Інвестиції в штучний інтелект можуть допомогти агентствам використовувати передові технології, такі як комп'ютерне бачення та машинне навчання, щоб сформувати майбутнє транспорту, де безпека пасажирів є пріоритетом, менше аварій і менше заторів на дорогах.

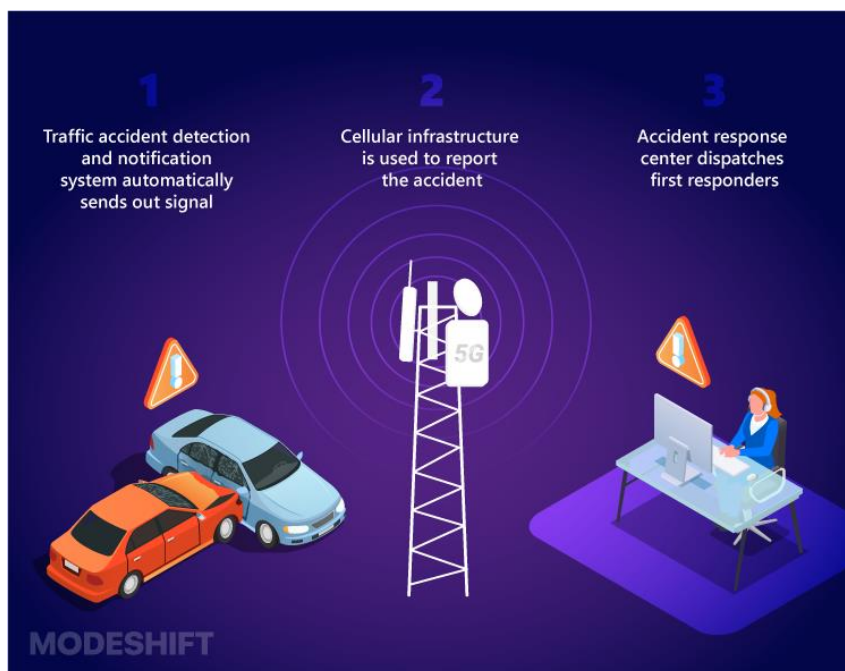


Рисунок 1.2 – Автоматичне виявлення аварій

Однією з ключових переваг використання ШІ у відеоспостереженні транспорту є здатність системи реагувати в реальному часі на зміни в оточенні. Нейронні мережі можуть виявляти аномалії та негайно сповіщати операторів або автоматично вживати заходів для попередження аварій.

Окрім безпеки, використання ІІІ в системах відеоспостереження сприяє вдосконаленню транспортного управління. Аналіз даних, зібраних системою, дозволяє оптимізувати рух транспорту, зменшуючи затори та поліпшуючи загальну ефективність транспортної інфраструктури.

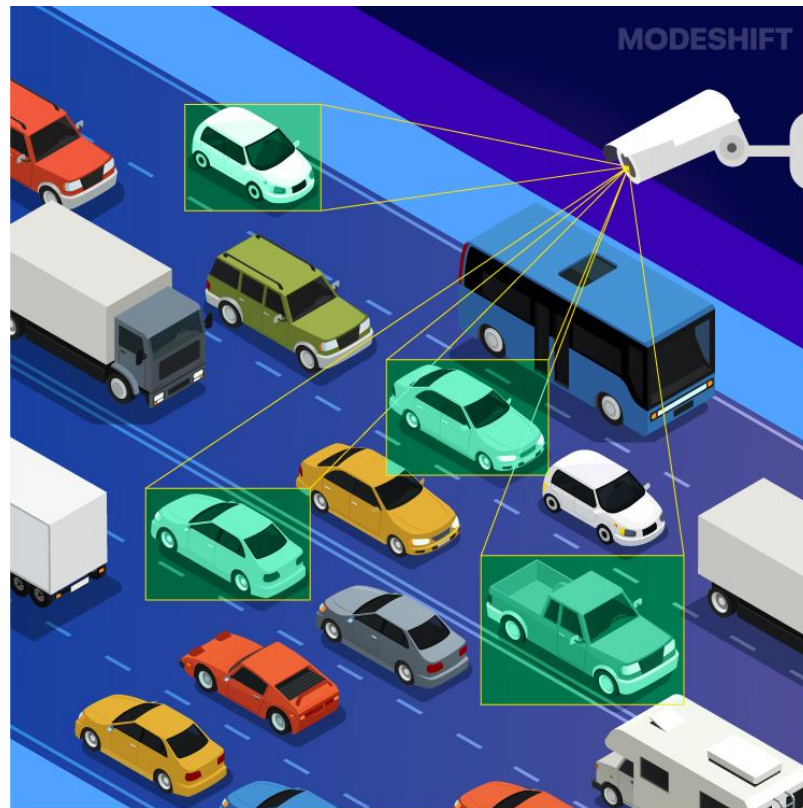


Рисунок 1.3 – Контроль за правилами паркування та дорожнього руху

Важливим аспектом є також можливість використання систем відеоспостереження для впровадження елементів автономного управління транспортними засобами. Інтелектуальні алгоритми можуть допомагати в автоматичному управлінні рухом автономних транспортних засобів на основі зібраних даних та прогнозування дорожньої обстановки.

В даній роботі ми зосередимось на використанні такої системи в сфері громадського транспорту, де вона відкриває нові можливості для підвищення безпеки та оптимізації функціонування громадського транспорту. Завдяки аналізу великої кількості відеоданих, отриманих з камер на транспортних

засобах та на зупинках, системи можуть ефективно виявляти потенційно небезпечні ситуації, контролювати пасажирський потік, а також сприяти вирішенню завдань вдосконалення сервісу та збільшення прибутку для громадського транспорту. Інтеграція штучного інтелекту в системи відеоспостереження громадського транспорту є стратегічною ініціативою, яка сприяє створенню більш безпечних та ефективних систем масового перевезення, що відповідають сучасним вимогам міської інфраструктури та забезпечують зручність для пасажирів.

1.2 Обґрунтування вибору напрямку дослідження

Вибір напрямку дослідження використання штучного інтелекту в системах відеоспостереження для громадського транспорту обґрунтовується важливістю вирішення ключових завдань, що стоять перед громадським транспортом. По-перше, підрахунок пасажирів і статистика користування надає можливість підвищення ефективності маршрутів та планування руху транспорту, що допомагає уникати переповнення та зменшує час очікування для пасажирів.

Додатково, використання ШІ дозволяє системам відеоспостереження ефективно виявляти небезпечні ситуації, такі як агресія чи надто велика щільність пасажирів, сприяючи покращенню безпеки в транспорті. Це може включати в себе виявлення втрати рівноваги пасажирів, попередження про потенційні конфліктні ситуації та оперативну реакцію на екстрені випадки.

Додатковий аспект полягає в аналізі якості обслуговування, де системи відеоспостереження можуть фіксувати поганий сервіс, такий як невідповідність графіку, відсутність зручностей для пасажирів або несправності у транспортних засобах. Це надає операторам транспорту можливість швидко реагувати на проблеми та покращувати якість обслуговування.

Таким чином, обґрунтування вибору цього напрямку дослідження базується на прагматичних потребах громадського транспорту, враховуючи підрахунок пасажирів, статистику користування, виявлення небезпечних ситуацій, удосконалення якості обслуговування та забезпечення більшої безпеки для всіх учасників транспортного процесу.

1.3 Вибір технології

Нижче перелічено ряд технологій які я вибрав для створення даної системи:

Spring Boot: Вибір Spring Boot обумовлено його здатністю до швидкого та простого налаштування проєктів, що базуються на Spring Framework. Його конвенції та автоматична конфігурація спрощують розгортання та підтримку, дозволяючи зосередитися на бізнес-логіці.

Spring Security: Використання Spring Security обумовлено необхідністю забезпечення високого рівня безпеки в системі відеоспостереження для транспорту. Spring Security гарантує ефективну аутентифікацію та авторизацію, що є важливим для збереження конфіденційності та захисту даних.

AWS (Amazon Web Services): Цей сервіс я вибрав через його здатність забезпечити надійний та масштабований хмарний інфраструктурний фундамент. AWS дозволяє легко інтегрувати систему в хмарне середовище, забезпечуючи високу доступність та ефективне зберігання даних.

Amazon Rekognition: Використання Amazon Rekognition обумовлено його потужними можливостями у сфері розпізнавання обличчя, об'єктів та аналізу відео. Цей сервіс надає системі відеоспостереження здатність автоматично визначати та класифікувати об'єкти на відеозаписах, що сприяє підвищенню ефективності виявлення потенційно небезпечних ситуацій та поліпшенню безпеки в громадському транспорті.

Amazon S3 (Simple Storage Service): вибрав цей сервіс для зберігання кадрів з камер, через його ефективність, надійність та можливість забезпечення низької затримки доступу до даних. S3 є відмінним варіантом для зберігання великих обсягів даних, забезпечуючи одночасно високу швидкість передачі та довговічність зберігання, що є ключовим для ефективної роботи системи відеоспостереження.

REST (Representational State Transfer): Використання архітектурного стилю REST обрано для забезпечення простого та ефективного обміну даними між компонентами системи. REST дозволяє створювати гнучкі та легко розширювані веб-сервіси.

Java 17: Використання Java 17 обумовлено її високою продуктивністю та широкою популярністю в галузі розробки. Java надає стабільність та надійність виконання програмного коду, роблячи її оптимальним вибором для розробки великих та складних систем.

Lombok: Використання Lombok пов'язане з його здатністю автоматизувати генерацію boilerplate-коду, що спрощує розробку та підтримку проекту. Анотації Lombok дозволяють уникнути писання рутинного коду, підвищуючи читабельність та підтримку кодової бази.

Hibernate: найкращий варіант для об'єктно-реляційного відображення, що спрощує роботу з базою даних. Hibernate дозволяє зручно взаємодіяти з реляційними базами даних, забезпечуючи оптимальний доступ до даних.

Jedis: Використання Jedis в системі відеоспостереження пов'язане з його використанням для взаємодії з Redis. Це дозволяє швидко та ефективно працювати з кешуванням та збереженням даних в пам'яті, що важливо для оптимізації продуктивності.

JUnit 5: Використання JUnit 5 пов'язане з необхідністю написання та виконання автоматизованих тестів для перевірки правильності роботи окремих компонентів системи. JUnit 5 забезпечує зручний інструментарій для тестування, покращуючи стабільність розробки.

MySQL: Використання MySQL визначено як системи управління базами даних з метою ефективного зберігання та організації даних. MySQL забезпечує надійність та швидкодію, що робить його оптимальним рішенням для обраних завдань системи відеоспостереження для громадського транспорту.

2 ПРОЄКТУВАННЯ МОДЕЛІ ТА РЕАЛІЗАЦІЇ ПРОГРАМНОГО КОМПЛЕКСУ

2.1 Розробка та Архітектурна Модель

Перед тим як перейти до архітектури класів, необхідно ретельно розглянути процес розробки та визначити архітектурну модель системи.

Провівши аналіз архітектурної моделі, обрано архітектурну модель Model-View-Controller (MVC) для оптимальної організації та взаємодії компонентів системи. MVC розподіляє архітектурну структуру на три основні частини: Model (Модель), яка відповідає за управління та обробку даних, View (Вид), яка відповідає за відображення інтерфейсу користувача, та Controller (Контролер), який відповідає за обробку користувацьких введень і взаємодію з Model та View.

Модель MVC відмінно підходить для системи відеоспостереження, оскільки вона забезпечує чітку структуру та розділення відповідальностей. Model в даному випадку відповідає за управління та аналіз потоків відеоданих, забезпечуючи їхню оптимальну обробку. View відповідає за відображення важливої інформації для операторів та віддалений доступ до потоків камер. Controller взаємодіє з даними та видачею команд, що дозволяє реагувати на події в реальному часі.

Вибір MVC моделі в даному випадку обґрунтовується її здатністю до ефективного розподілу відповідальностей між компонентами системи, що сприяє легкій модифікації та розширенню. Ця модель також дозволяє відокремлювати логіку обробки даних від інтерфейсу користувача, що робить систему більш гнучкою та піддається підтримці. В результаті, MVC сприяє швидкому реагуванню на зміни вимог та підтримує практики високої якості в розробці програмного забезпечення.

2.2 Сутності предметної області

Перед розпочатком проектування бази даних визначення інформаційних об'єктів та їх взаємодій є надзвичайно важливим етапом. В цьому контексті використання Unified Modeling Language (UML) стає ключовим інструментом для створення діаграм взаємодії сутностей.

UML є мовою моделювання, яка надає стандартні засоби для візуалізації та аналізу систем. Діаграми взаємодії в UML використовуються для зображення взаємодії між об'єктами чи компонентами системи, включаючи взаємодію сутностей у базі даних.

На відміну від ER-діаграм, яка фокусується на моделюванні структури даних та їх взаємозв'язків, UML дозволяє моделювати більший спектр аспектів системи, охоплюючи як структурні, так і поведінкові характеристики. Використання UML-діаграм взаємодії для подання способу взаємодії сутностей сприяє полегшенню розуміння та аналізу системи, зокрема, у випадках проектування баз даних.

Цей підхід дозволяє зосередитися не лише на статичних відносинах між об'єктами, але й на їхній динаміці та взаємодії в часі. Такий підхід особливо важливий в проектуванні баз даних, де розуміння взаємодії сутностей є ключовим для створення ефективної та відповідної потребам системи бази даних.

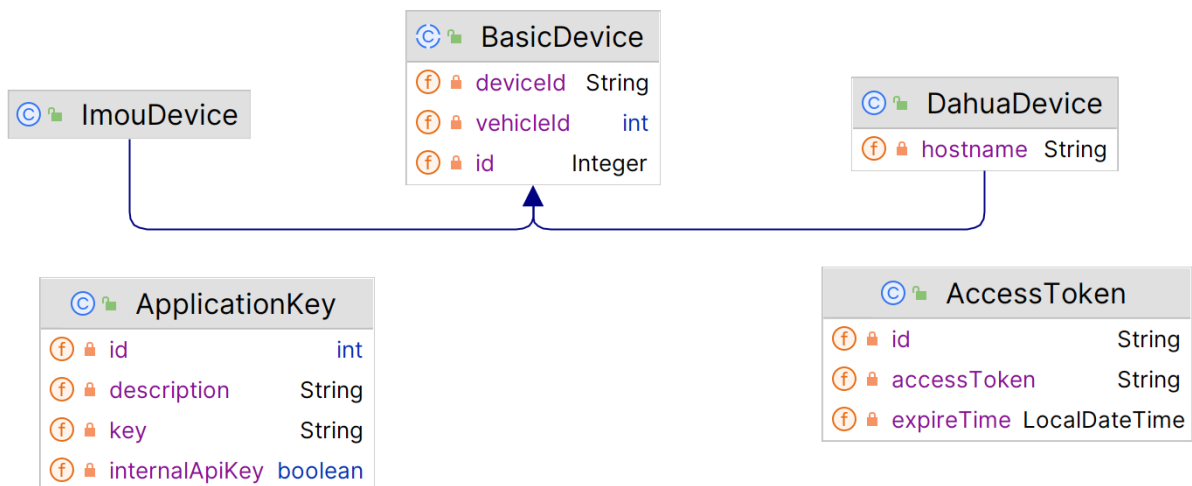


Рисунок 2.1 – UML діаграма зв'язк

- **BasicDevice**: Цей клас є базовим абстрактним класом для репрезентації пристрою. Він містить поля id, deviceId та vehicleId. Він використовується як суперклас для створення інших типів пристроїв в системі (Різні типи камер). Це сутність бази даних, адже вона анотована як @Entity і вживана стратегія наслідування SINGLE_TABLE, тому всі наслідувані класи будуть зберігатися в одній таблиці. Від цього класу мають наслідуватися всі камери, тому кожен тип камери буде мати свій підклас.
- **DahuaDevice**: Цей клас наслідується від BasicDevice і додає додаткове поле hostname. Оскільки він наслідується від BasicDevice, він також анотований як @Entity і може бути представлений як сутність бази даних.
- **ImouDevice**: Цей клас також наслідується від BasicDevice і перевизначає toString() метод, щоб подати інформацію лише про vehicleId. Це також сутність бази даних.
- **AccessToken**: є невід'ємною частиною аутентифікації і сесійного керування в додатках, що використовують Redis як сховище даних.
- **ApplicationKey**: репрезентує сутність ключа в базі даних який використовується для захищеного спілкування між клієнтом і сервером.

2.3 Вибір інструментів розробки

У розробці проекту використовувались такі ключові інструменти, як IntelliJ IDEA для написання та управління кодом, AWS для збереження файлів, Docker та Kubernetes для контейнеризації та оркестрації додатків, Lens для спрощеного моніторингу та управління Kubernetes-кластерами, а також Git (використовуючи платформу Bitbucket) для контролю версій та спільної роботи над кодом.

Обране середовище розробки — IntelliJ IDEA, яке забезпечує потужні можливості для роботи з мовами програмування, зокрема з Java, яка є основною мовою програмування для проекту.

Щодо зберігання даних, використано MySQL — реляційну базу даних, яка взаємодіє з Java-додатками. Цей підхід дозволяє ефективно управляти та організувати дані, використовуючи реляційну структуру таблиць.

Git використовувався через платформу Bitbucket для ефективного контролю версій та спільної роботи над кодом. Bitbucket надає зручний інтерфейс для спільної роботи команди та збереження історії змін в коді.

У розробці використовуються дві основні мови програмування: Java та React. Java використовується для написання backend-частини додатку, де відбувається обробка даних та виконання бізнес-логіки. У цьому випадку, MySQL, реляційна база даних, взаємодіє з Java-додатками. З іншого боку, React використовується для написання frontend-частини додатку, тобто користувацького інтерфейсу, що відображається на стороні клієнта.

Ця двошівнева архітектура забезпечує відокремлення бекенду та фронтенду, дозволяючи ефективно розробляти та підтримувати обидві частини додатку. Java виступає як основний інструмент для обробки та управління даними на сервері, тоді як React дозволяє створювати інтерактивний та адаптивний інтерфейс для користувачів. Основний програмний код з роботою та підключенням БД наведено в додатку В.

2.4 Опис процесу проекту з методологію RUP

Для цього проекту було обрано методологію проектування RUP. Ця методологія має довгий список переваг, серед яких чіткість і стандартизація. Вимоги інтегровані протягом усього процесу розробки. Вона може впоратися зі змінами у вимогах, незалежно від того, чи вони надходять від замовника, чи виникають під час проекту.

RUP є ітеративним, тобто повторюваним і гнучким. Ітеративним він є тому, що всі ключові дії процесу повторюються протягом усього проекту. Процес є гнучким, оскільки різні компоненти можуть бути скориговані, а фази циклу можуть повторюватися до тих пір, поки програмне забезпечення не буде відповідати вимогам і цілям.

Кожен цикл відповідає періоду часу, протягом якого, наприклад, розробляється нове покоління продукту. Як і в інших процесах, RUP визначає, хто, коли, де і що робить. "Хто" в цьому процесі - це працівники, які беруть активну участь у створенні системи. Під "що" мається на увазі щось конкретне, частина інформації. Ці "результати" можуть мати різні форми, наприклад, кейси користувачів або прототипи. Різні фази вже вказують на різні види діяльності, пов'язані з розробкою системи. Основні види діяльності описані більш детально нижче.

Рациональний уніфікований процес (Rational Unified Process, RUP) поділяє розробку на чотири послідовні фази:

- аналіз та планування вимог
- проектування
- застосування
- тестування.

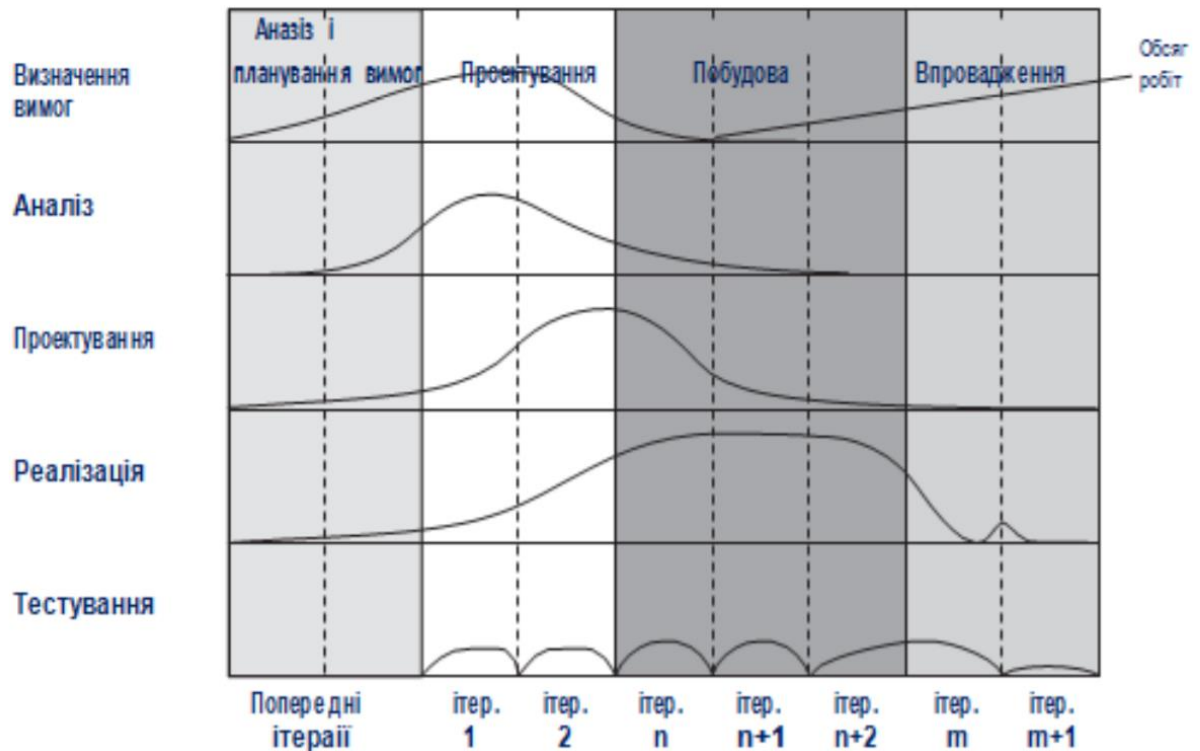


Рисунок 2.2 – Фази проектування RUP

2.4.1 Аналіз і планування вимог за методологією RUP

Було здійснено ряд вимог:

- техніко-економічне обґрунтування;
- економічне обґрунтування;
- визначення основних вимог;
- ідентифікація ризиків

Техніко-економічне обґрунтування

Система відеоспостереження є рішенням для поліпшення безпеки та комфорту пасажирів громадського транспорту. Застосування штучного інтелекту та передових алгоритмів машинного навчання в сфері відеоспостереження дозволить створити адаптивну та ефективну систему, яка реагує на потреби пасажирів та забезпечує найвищий рівень безпеки.

Економічне обґрунтування

Впровадження системи відеоспостереження в громадському транспорті може бути визначено через ряд факторів, які впливають на безпеку пасажирів,

ефективність обслуговування та загальний рентабельний результат транспортної компанії. Ось деякі аспекти економічного обґрунтування:

а) Зменшення втрат від злочинності:

- 1) Система відеоспостереження може значно знизити ризик крадіжок, вандалізму та інших злочинних дій в громадському транспорті.
- 2) Зменшення інцидентів може сприяти підвищенню безпеки пасажирів та майна, що у свою чергу може позитивно позначитися на зменшенні витрат на відшкодування збитків.

б) Підвищення довіри пасажирів:

- 1) Пасажири відчують себе більш захищеними та безпечними завдяки наявності відеоспостереження.
- 2) Підвищення рівня довіри може призвести до збільшення пасажиропотоку та доходів для транспортної компанії.

в) Оптимізація маршрутів та обслуговування:

- 1) За допомогою аналізу зібраних даних про маршрути та обслуговування, транспортна компанія може вдосконалити свою роботу та зменшити зайві витрати.
- 2) Ефективне використання інформації може покращити роботу громадського транспорту, забезпечуючи оптимальні маршрути та зменшуючи час простою.

г) Збільшення ефективності служб безпеки:

- 1) Автоматичні повідомлення про підозрілу діяльність дозволяють швидко реагувати на можливі загрози.
- 2) Збільшена ефективність служб безпеки може зменшити витрати на взаємодію з правоохоронними органами та врегулювання конфліктів.

д) Збільшення продуктивності роботи транспортної компанії:

- 1) Аналітика та оптимізація можуть призвести до зменшення непланованих витрат та покращення загального функціонування компанії.

- 2) Система зможе порівняти кількість активних квитків на даний момент в транспорті з реальною кількістю пасажирів для контролю пасажирів, які не купили квиток – що також призведе до зменшенню збитків



Рисунок 2.3 – Діаграма прецедентів

Загальне економічне обґрунтування впровадження системи відеоспостереження полягає в забезпеченні безпеки, підвищенні довіри та оптимізації функціонування громадського транспорту, що може призвести до позитивних економічних результатів для транспортної компанії.

Визначення основних вимог

Для системи відеоспостереження в громадському транспорті має включати технічні, функціональні та безпекові аспекти. Ось деякі ключові вимоги:

- а) Відображення високоякісного відео:

- 1) Опис: Система повинна забезпечувати високоякісне відеоспостереження, щоб ефективно реагувати на інциденти та забезпечити чітку ідентифікацію подій.
 - 2) Вимоги: Роздільна здатність, кадрова частота, стандарти стиснення відео.
- b) Віддалений доступ та моніторинг:
- 1) Опис: Система повинна надавати можливість віддаленого доступу до відеопотоків для централізованого моніторингу.
 - 2) Вимоги: Забезпечення безпеки з'єднання, швидкість передачі даних, зручний інтерфейс для операторів.
- c) Автоматичні повідомлення:
- 1) Опис: Система повинна виявляти підозрілу діяльність та автоматично надсилати повідомлення для швидкого реагування.
 - 2) Вимоги: Ефективність алгоритмів виявлення, налаштування чутливості, можливість налаштування типів повідомлень.
- d) Аналіз та збір даних:
- 1) Опис: Система повинна забезпечувати збір та аналіз даних для оптимізації маршрутів та поліпшення обслуговування.
 - 2) Вимоги: Збір та зберігання структурованих даних, аналітичні можливості, інтеграція з іншими системами.
- e) Ідентифікація людей за допомогою штучного інтелекту:
- 1) Опис: Система повинна використовувати AI для розпізнавання людей та підрахунку пасажирів.
 - 2) Вимоги: Точність розпізнавання, швидкість обробки даних, забезпечення конфіденційності даних.
- f) Стабільність та надійність:
- 1) Опис: Система повинна працювати стабільно та надійно навіть в умовах високих навантажень або можливих технічних неполадок.
 - 2) Вимоги: Резервування систем, автоматичне відновлення, тестування на витривалість.

g) Забезпечення конфіденційності та захисту даних:

- 1) Опис: Система повинна забезпечувати високий рівень конфіденційності та захисту особистих даних.
- 2) Вимоги: Шифрування даних, контроль доступу, відповідність стандартам безпеки.

Виявлення ризиків

Існує низка потенційних викликів, які можуть вплинути на успішність проекту. Ось деякі можливі ризики:

a) Технічні проблеми:

- 1) Опис ризику: Непередбачені технічні проблеми з обладнанням, програмним забезпеченням або мережами можуть призвести до перебоїв у роботі системи.
- 2) Можливі наслідки: Затримки в моніторингу, втрата відомостей про інциденти.

b) Проблеми з безпекою даних:

- 1) Опис ризику: Злам системи або несанкціонований доступ може призвести до витоку конфіденційної інформації або порушення правил збереження даних.
- 2) Можливі наслідки: Порушення конфіденційності пасажирів, втрата довіри до системи.

c) Низька ефективність системи виявлення:

- 1) Опис ризику: Алгоритми виявлення підозрілої діяльності можуть бути недостатньо ефективними, що приведе до невиявлення інцидентів або великої кількості помилок.
- 2) Можливі наслідки: Зниження довіри до системи, можливість втрати подій.

d) Відсутність інтеграції з іншими системами:

- 1) Опис ризику: Неуспішна інтеграція з іншими системами (наприклад, системами керування транспортом) може обмежити можливості аналізу та оптимізації.

2) Можливі наслідки: Ускладнене використання зібраних даних для прийняття стратегічних рішень.

е) Невірне визначення вимог:

1) Опис ризику: Неправильне або неповне визначення потреб користувачів та вимог до системи може призвести до розбіжностей між очікуваннями та реальною функціональністю.

2) Можливі наслідки: Затримки у виконанні проекту, неспівпадіння очікувань користувачів.

ф) Зміни законодавства та правил збереження даних:

1) Опис ризику: Зміни в законодавстві щодо конфіденційності даних можуть вимагати перегляду або змін в системі.

2) Можливі наслідки: Витрати на адаптацію системи до нових вимог.

г) Неадекватна підготовка персоналу:

1) Опис ризику: Недостатній рівень навчання персоналу, що відповідає за моніторинг та реакцію на інциденти, може призвести до неефективного використання системи.

2) Можливі наслідки: Затримки у реагуванні на інциденти, можливість втрати важливих деталей.

Врахування цих ризиків у процесі планування та впровадження системи дозволить приймати запобіжні заходи та ефективно управляти можливими труднощами на різних етапах проекту.

2.4.2 Проектування за методологію RUP

На етапі проектування оцінюються та аналізуються системні вимоги та необхідна архітектура. Саме тут проект починає набувати форми. Мета етапу проектування - проаналізувати продукт і закласти основи майбутньої архітектури.

Наступною фазою є фаза побудови, яка вимагає чіткого визначення бізнес-плану.

Опис етапу проектування проекту.

1. Створення бази даних. Мається на увазі база даних з усіма таблицями, полями та зв'язками. Створіть сутності для кожної таблиці та поля;
2. Визначення методів DAO для підключення до бази даних; і
3. Створення класів сервісу. Включаючи валідацію;
4. Написання тестів.
5. Створення представлення системи

2.4.3 Реалізація за методологію RUP

Під час фази реалізації Rational Unified Process (RUP) система програмного забезпечення розробляється у повному обсязі, з основним акцентом на створенні компонентів та інших можливостей системи. На цьому етапі також проводиться більшість програмування. У цьому виробничому процесі наголошується на ефективному управлінні витратами та ресурсами, а також на забезпеченні високої якості. Результатами етапу реалізації є:

- Повністю розроблена система програмного забезпечення.
- Посібник користувача

Для успішної виконання фази Побудова важливо завершити ряд завдань, зокрема:

- Визначення вимог - визначення всіх невизначених вимог;
- Аналіз - завершення моделі аналізу;
- Проектування - доопрацювання моделі спроектованої системи;
- Реалізація - створення базової функціональності;
- Тестування - тестування базової функціональності

2.4.4 Тестування за методологію RUP

Після завершення бета-тестування і остаточного розгортання системи починається етап тестування. Ця фаза передбачає виправлення будь-яких дефектів, виявлених під час бета-тестування, і підготовку до масового розгортання програмного забезпечення на всіх сайтах користувачів.

Відповідно до цього проекту, я поставив цілі для етапу тестування

- Виправлення помилок.

- Створення автоматизованих тестів для графічного інтерфейсу

- Створення модульних тестів
- Перевірка сумісності з усіма браузерами
- Налаштування точності роботи програми на екранах з різним нахилом;
- Створення користувацької документації

2.5 Огляд та встановлення фреймворків тестування

У розвитку мого Java-проекту, відданого створенню надійного та ефективного програмного забезпечення, виникла необхідність в реалізації автоматизованого тестування. Для досягнення цієї мети, я вирішив використати фреймворк JUnit 5, який забезпечує широкі можливості для написання тестів та впровадження надійної системи контролю якості.

Початковим етапом було підключення JUnit 5 до мого Java-проекту. Для цього я використовував систему управління залежностями, Maven, і додав відповідні залежності у файл pom.xml[2].

```
<dependency>  
  <groupId>org.junit.jupiter</groupId>  
  <artifactId>junit-jupiter-api</artifactId>  
  <version>5.10.1</version>  
  <scope>test</scope>  
</dependency>
```

Рисунок 2.4 – Залежність Junit для Maven

Після успішного підключення JUnit 5, я розпочав створювати тестові класи. Кожен тестовий клас був спроектований для перевірки конкретної частини програми. Використовуючи анотації JUnit, я позначав тестові методи, налаштовував передумови для тестування та організовував чисте середовище для кожного тесту.

Написання тестових методів виконувалося за допомогою анотації `@Test`, яка індикатором того, що метод є тестом. Використовуючи ці тестові методи, я здійснював перевірку різних аспектів функціональності мого програмного забезпечення.

За допомогою анотацій JUnit, таких як `@BeforeEach` і `@AfterEach`, я визначав методи, які викликалися перед і після кожного тесту. Це надавало можливість налаштувати початкові та кінцеві умови перед проведенням тестів. Крім того, я використовував інші функціональні можливості JUnit 5, такі як організація тестових груп, параметризовані тести та інші, щоб забезпечити повноту та ефективність моєї системи тестування.

Після написання тестів, я використовував IntelliJ IDEA та інші інструменти для запуску та аналізу результатів. Цей етап дозволяє швидко виявляти та виправляти помилки, забезпечуючи надійність та стабільність програми.

Приклад автоматизованого тесту в додатку В.

2.5.1 Переваги та недоліки фреймворку

У певних випадках, підключенням і використанням фреймворку JUnit 5 можна забезпечити ефективне тестування Java-проектів. Простота використання та широкий функціонал роблять його популярним в розробницькій спільноті. Анотації, можливості конфігурації, інтеграція з іншими інструментами, паралельне виконання тестів і активна спільнота - всі ці фактори сприяють успішному впровадженню JUnit 5 в розробку програмного забезпечення.

З іншого боку, варто враховувати обмеження JUnit 5. У деяких випадках конфігурація тестів може бути обмеженою, порівняно з іншими фреймворками. Відсутність вбудованих можливостей для параметризації тестів може вимагати використання додаткових бібліотек або самостійної реалізації цієї функціональності. Також, JUnit 5 слабше підтримує тести залежностей, що може становити проблему для організації складних тестових сценаріїв. Крім того, тестування асинхронного коду може викликати труднощі та вимагати додаткових зусиль для належної реалізації.

2.6 Архітектура програмного забезпечення

На етапі проектування архітектури системи використовується об'єктно-орієнтований підхід (ООП), який дозволяє структурувати код відповідно до концепцій об'єктів та класів. Кожен клас представляє собою об'єктно-орієнтований елемент, який має свої атрибути та методи. ООП дозволяє розділити функціональність системи на окремі класи, що спрощує розробку, розуміння та розширення коду.

На етапі сутностей, кожен клас визначається як об'єкт з конкретними атрибутами та методами. Використання ООП дозволяє логічно групувати дані та функціональність, пов'язані з кожним класом.

Крім того, класи бінів (Bean Classes), обробка винятків, конфігураційні файли, контролери, сервіси, завдання та репозиторії реалізують принципи ООП для визначення структури та взаємодії компонентів системи. Це дозволяє відокремити логіку, специфічну для кожного класу, що полегшує розвиток, тестування та обслуговування системи.

ООП сприяє високій модульності та розширюваності проекту, а його концепції такі, як інкапсуляція та спадкування, роблять код більш зрозумілим та підтримуваним на великих проектах.

Нижче наведений список об'єктів які формують архітектуру системи:

- Класи:

На етапі сутностей (Entity Classes) у моєму проекті існують класи сутностей, такі як BasicDevice, ApplicationKey та AccessToken. Наприклад, клас BasicDevice містить не лише основні атрибути пристрою, але і методи для забезпечення валідації та ефективного використання. Клас ApplicationKey відповідає за обробку ключів додатків та забезпечує безпеку взаємодії з іншими сервісами. AccessToken, у свою чергу, відповідає за управління та обробку доступу до ресурсів.

- Біни:

Також в моєму проєкті присутні класи бінів (Bean Classes), наприклад, `SnapsByTimeBean`, `ImageDeviceBean` та `DeviceServiceContainer`. `SnapsByTimeBean` визначає структуру для обробки та передачі зображень за часовими інтервалами. `ImageDeviceBean` служить об'єктом передачі даних для оптимізованої обробки інформації про зображення. Клас `DeviceServiceContainer`, у свою чергу, поєднує різні сервіси та забезпечує їхню взаємодію.

- Обробка Винятків:

Робота з винятками здійснюється за допомогою класів, таких як `ApiExceptionHandler` та власні винятки, наприклад, `NoDataException`. `ApiExceptionHandler` визначає логіку обробки виняткових ситуацій, що можуть виникнути в ході виконання додатку, забезпечуючи структурований та інформативний вивід помилок.

- Конфігураційні Файли:

Проєкт має кілька конфігураційних класів, таких як `S3Config`, `RedisConfig`, `AsyncConfig`. Наприклад, `RedisConfig` визначає параметри конфігурації для взаємодії з `Redis`, визначаючи адресу сервера та інші параметри. `AsyncConfig` визначає налаштування для асинхронного виконання завдань у системі.

- Контролери:

Клас `S3Controller` виступає в ролі веб-шару мого додатку та надає API клієнтам для обробки вхідних запитів та надсилання відповідей. Наприклад, `S3Controller` визначає методи для завантаження файлів у хмарне сховище (`Amazon S3`) та отримання їхнього змісту.

- Сервіси:

Шар сервісів мого додатку представлений класами, такими як `S3Service`, `DeviceVehicleService` та `HttpSendService`. Сервіси містять бізнес-логіку програми та взаємодіють з репозиторіями. Наприклад, `DeviceVehicleService` надає методи для обробки даних про транспортні засоби та їхнє збереження в базі даних.

- Завдання:

Клас `ScheduleTask` є типовим виконавцем завдань Spring, який запускає деякі задачі з фіксованими інтервалами. Наприклад, `ScheduleTask` може бути відповідальним за планування регулярної синхронізації даних зовнішніх джерел.

- Репозиторії:

Інтерфейси `DeviceVehicleRepository` використовуються для абстрагування операцій з базою даних для сутності `DeviceVehicle`. Наприклад, `DeviceVehicleRepository` може містити методи для отримання та збереження інформації про транспортні засоби.

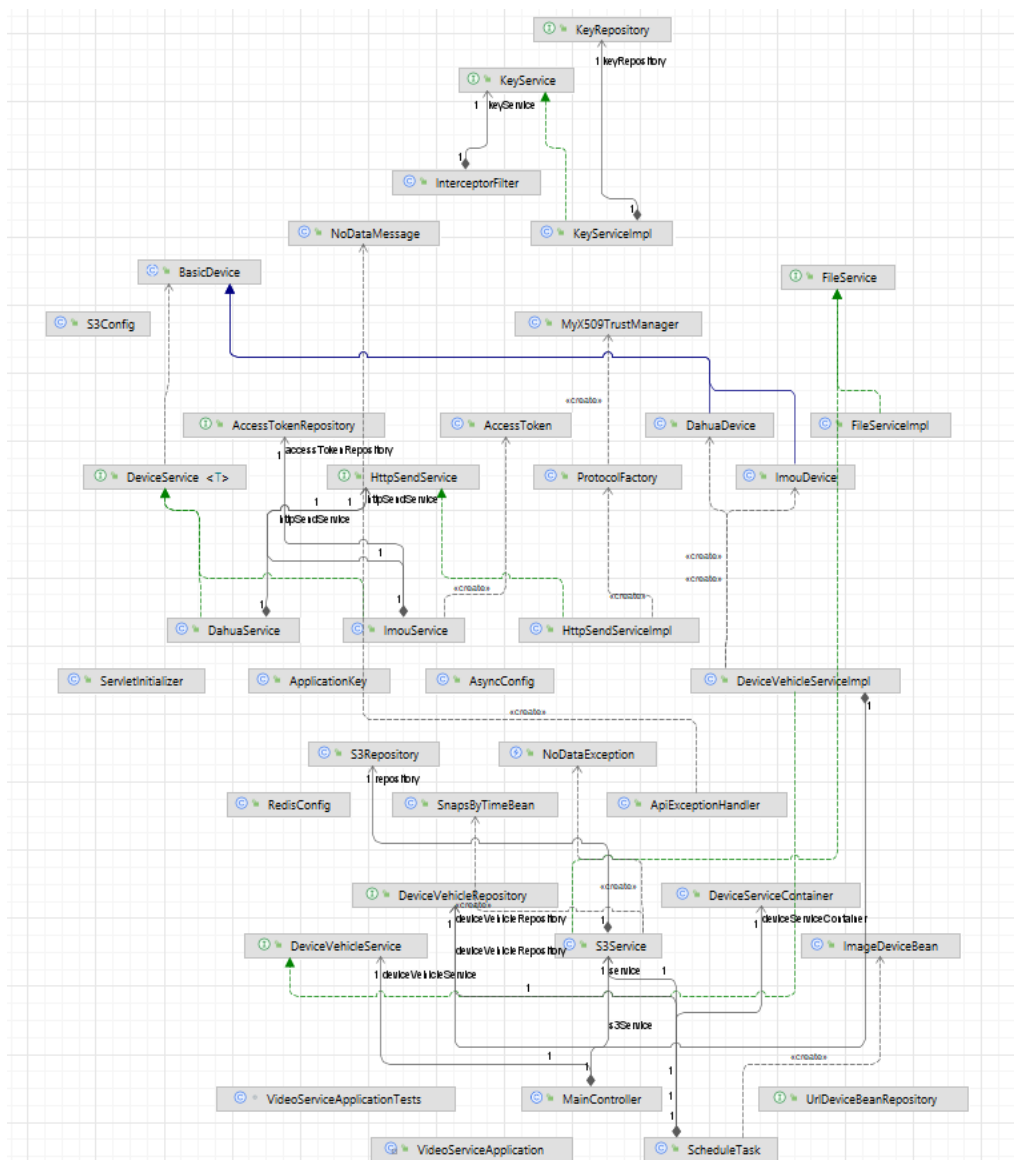


Рисунок 2.5 – Діаграма класів

В даній діаграмі класів не враховувались класи пов'язані з фреймворками тестування.

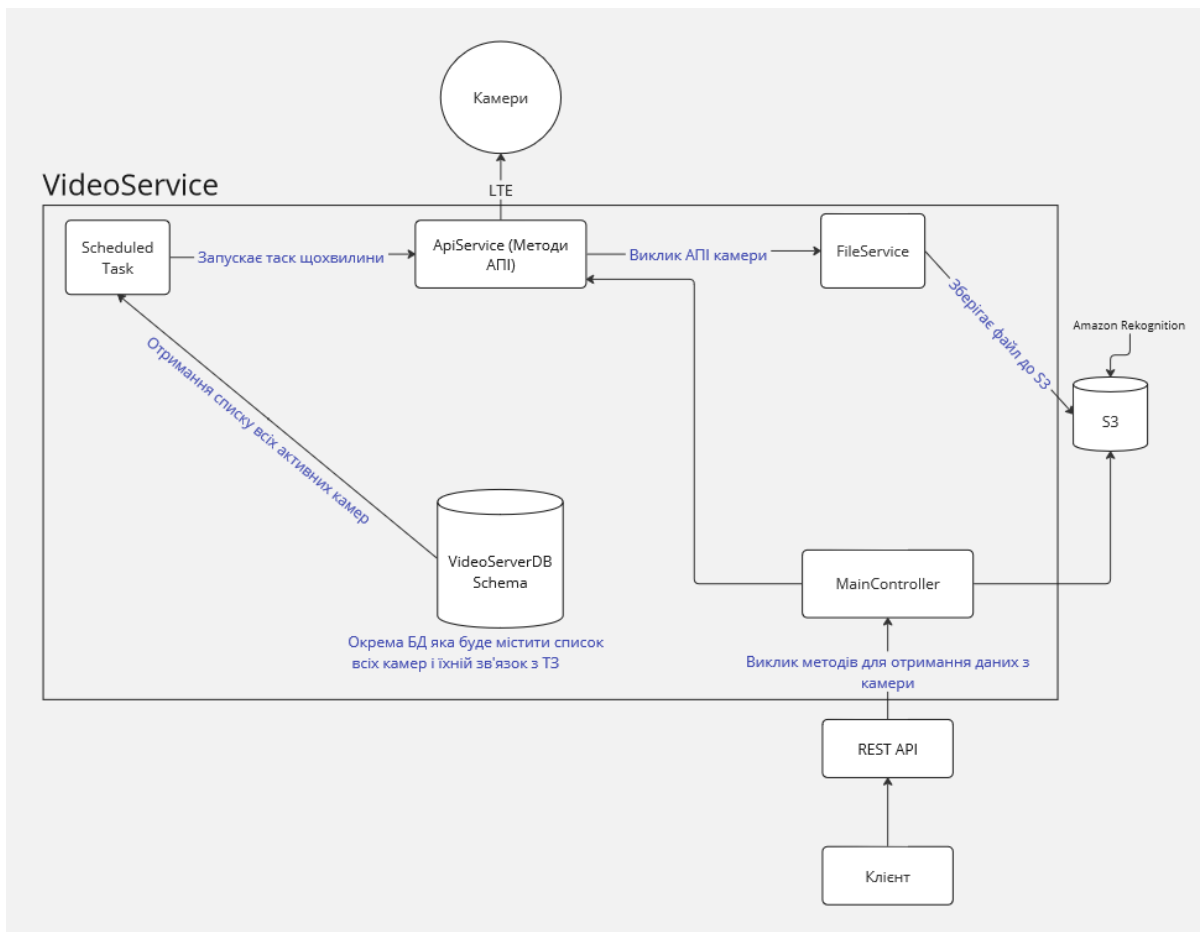


Рисунок 2.6 – Схема роботи програми

Схема роботи програми у випадку обробки кадрів ШІ - щохвилини запускається завдання в `ScheduledTask`, яке витягує список всіх камер з БД, тоді за допомогою `ApiService` класу і його методів, викликається API камери для того щоб витягнути останні кадри, далі ці файли за допомогою `FileService` зберігаються в `S3`, де підключений `Amazon Rekognition` який і опрацьовує ШІ фото і додає дані до метадати. Після цього, клієнт може викликати `S3Controller` для того щоб отримати ці дані. У випадку якщо клієнт хоче отримати історію записів чи інший функціонал не пов'язаний з ШІ, то шляхом виклику `S3Controller` через `REST API` є можливість зв'язатися з `ApiService` і отримати дані з камери напряму.

2.7 Взаємодія з СУБД

Для надійної взаємодії з сервером MySQL, конфігураційні дані зберігаються у файлі `application.properties`. У цьому файлі містяться всі необхідні параметри, такі як URL бази даних, ім'я користувача та пароль. Це забезпечує зручність та безпеку у керуванні доступом до бази даних. Зокрема, завдяки параметрам середовища, можливо динамічно змінювати ці дані, що важливо для ефективного ведення розробки в різних середовищах, таких як робочий, тестовий чи релізний.

Основний інструмент для взаємодії з базою даних - це репозиторії, які використовують підтримку Spring Data JPA. Вони дозволяють визначати інтерфейс репозиторію, що успадковує від `CrudRepository`, забезпечуючи розширену функціональність для взаємодії з базою даних. Це дозволяє розробникам ефективно виконувати CRUD-операції (створення, читання, оновлення, видалення) з використанням простого та зрозумілого коду.

Важливо відзначити, що завдяки Spring Data JPA і його можливостям автоматичного створення запитів, розробники можуть писати більш читабельний код та економити час при роботі з базою даних. Динамічні методи, які можна визначити в інтерфейсі репозиторію, спрощують створення запитів та забезпечують гнучкість у виборі даних з бази.

Такий підхід до взаємодії з базою даних сприяє підтримці чистоти та структурованості коду, що важливо для підтримки та розвитку проекту.

Код підключення наведений в додатку В.

2.8 Впровадження сервісу розпізнавання Amazon Rekognition

2.8.1 Можливості і переваги Amazon Rekognition

Amazon Rekognition - це сервіс, що надає можливість легко інтегрувати аналіз зображень та відео в ваші додатки[3]. Достатньо надати зображення або відео через API Amazon Rekognition сервіс може:

- Визначати мітки (об'єкти, концепції, людей, сцени та дії) і текст.
- Виявляти неприпустимий контент.
- Надавати високоточний аналіз обличчя, порівняння облич та можливості пошуку облич.

Основні випадки використання Amazon Rekognition включають:

- Пошукові бібліотеки зображень та відео: Робить зображення і відео доступними для пошуку за допомогою міток об'єктів, концепцій та сцен.
- Виявлення живого обличчя: Допомагає визначати шахраїв у випадках перевірки особистості на основі обличчя. Виявляє спроби обдурити систему, такі як фотографії, відео, або 3D-маски.
- Підтвердження користувача на основі обличчя: Дозволяє підтверджувати особистості користувачів порівнюючи їхнє живе зображення з посиланням на обличчя.
- Виявлення та аналіз обличчя: Дозволяє виявляти та аналізувати різні компоненти обличчя та атрибути, такі як емоційні вирази, демографічна інформація, закриті частини обличчя тощо.
- Пошук облич: Забезпечує можливість пошуку облич за збереженими зображеннями в спеціальному контейнері.
- Виявлення захисних засобів: Виявляє захисні засоби на зображеннях осіб, що може бути корисно в галузях, таких як будівництво, виробництво, охорона здоров'я тощо.

- Виявлення небезпечного контенту: Виявляє дорослий та насильницький контент на зображеннях та відео для подальшого фільтрування.
- Розпізнавання знаменитостей: Розпізнає тисячі знаменитостей в різних категоріях, таких як політика, спорт, бізнес, розваги та медіа.
- Виявлення тексту: Дозволяє розпізнавати та виділяти текстовий контент на зображеннях.
- Спеціалізовані мітки: За допомогою Amazon Rekognition Custom Labels можна ідентифікувати об'єкти та концепції на зображеннях, специфічних для вашого бізнесу.

Переваги використання Amazon Rekognition включають:

- Інтеграція в додатки: Легко інтегрує потужний аналіз зображень та відео в додатки без потреби у глибоких знаннях у сфері комп'ютерного зору чи глибокого навчання.
- Аналіз на основі глибокого навчання: Використовує технологію глибокого навчання для точного аналізу зображень та порівняння обличчя у них.
- Масштабований аналіз зображень: Дозволяє аналізувати мільйони зображень для кураторства та організації великих обсягів візуальних даних.
- Аналіз та фільтрація зображень за їхніми властивостями: Дозволяє аналізувати властивості зображень, такі як якість чи кольори.
- Інтеграція з іншими службами AWS: Розроблений для безпроблемної роботи з іншими службами AWS, такими як Amazon S3 та AWS Lambda.
- Низькі витрати: Ви платите лише за аналізовані зображення та відео, а також за метадані обличчя, які зберігаються.

2.8.2 Вибір моделі для Amazon Rekognition

Для цього проекту вибрано модель «IndexFaces»[5] яка виявляє обличчя на вхідному зображенні та додає їх до вказаної колекції. Сервіс з використанням цієї моделі не зберігає фактичні виявлені обличчя, натомість основний алгоритм виявлення спочатку визначає обличчя у вхідному зображенні, далі для кожного обличчя алгоритм витягує риси обличчя у вектор ознак і зберігає його у

серверній базі даних. Amazon Rekognition використовує вектори ознак, коли виконує операції зіставлення обличчя та пошуку за допомогою операцій SearchFaces і SearchFacesByImage. Щоб додати обличчя до колекції, можна скористатися операцією IndexFaces. При цьому важливо враховувати версію моделі виявлення облич, оскільки різні версії можуть індексувати різну кількість облич у вхідному зображенні. Наприклад, версія 1.0 індексує 15 найбільших облич, тоді як пізніші версії можуть обробляти 100 найбільших облич.

Моделі обличчя версії 4 або новішої не повертають інформацію про орієнтацію зображення в полі OrientationCorrection. Для визначення версії моделі можна використовувати операцію DescribeCollection або перевіряти значення FaceModelVersion у відповіді від IndexFaces.

```
{
  "CollectionId": "MyCollection",
  "Image": {
    "S3Object": {
      "Bucket": "bucket",
      "Name": "input.jpg"
    }
  },
  "ExternalImageId": "input.jpg",
  "DetectionAttributes": [
    "DEFAULT"
  ],
  "MaxFaces": 1,
  "QualityFilter": "AUTO"
}
```

Рисунок 2.7 – приклад тіла запиту до API моделі

Amazon Rekognition також дозволяє пов'язувати виявлені обличчя з необов'язковим ExternalImageId, який може бути наданий для вхідного зображення. При виклику операції ListFaces зовнішній ідентифікатор повертається у відповіді, що дозволяє створити індекс та пов'язати обличчя з кожним зображенням.

Інші параметри, такі як `MaxFaces` для вказівки максимальної кількості обличчя для індексації та `QualityFilter` для фільтрації за якістю, також можуть бути використані для оптимізації операцій.

Операція `IndexFaces` повертає масив метаданих, включаючи обмежувальну рамку, значення достовірності та ідентифікатор обличчя. Додаткові атрибути обличчя, такі як орієнтація та оклюзія, можуть бути отримані за допомогою параметра `detectionAttributes`.

Важливо враховувати, що обличчя, які не відповідають критеріям індексації, не будуть проіндексовані. Це може статися з обличчями, які занадто малі, розпливчасті, темні або мають недостатньо деталей для пошуку. Операція також підтримує роботу зображень у форматі PNG або JPEG через передачу байтів або посилання на зображення в Amazon S3.

```
{
  "FaceModelVersion": "3.0",
  "FaceRecords": [
    {
      "Face": {
        "BoundingBox": {
          "Height": 0.3247932195663452,
          "Left": 0.5055555701255798,
          "Top": 0.2743072211742401,
          "Width": 0.21444444358348846
        },
        "Confidence": 99.99998474121094,
        "ExternalImageId": "input.jpg",
        "FaceId": "b86e2392-9da1-459b-af68-49118dc16f87",
        "ImageId": "09f43d92-02b6-5cea-8fbd-9f187db2050d"
      },
      "FaceDetail": {
        "BoundingBox": {
          "Height": 0.3247932195663452,
          "Left": 0.5055555701255798,
          "Top": 0.2743072211742401,
          "Width": 0.21444444358348846
        }
      }
    }
  ]
}
```

Рисунок 2.8 – приклад тіла відповіді API моделі

2.8.3 Опис параметрів Amazon Rekognition

CollectionId - Ідентифікатор існуючої колекції, до якої планується додати обличчя, виявлені на вхідних зображеннях.

Тип: Рядок

Обмеження по довжині: Мінімальна довжина 1. Максимальна довжина 255.

Шаблон: [a-zA-Z0-9_.-]+

Обов'язково: Так

DetectionAttributes - Масив атрибутів обличчя, які ви хочете отримати у відповіді. Завжди буде повертатися стандартний набір таких атрибутів - BoundingBox, Confidence, Pose, Quality та Landmarks. Можна запитувати конкретні атрибути (додатково до стандартного списку) за допомогою ["DEFAULT", "FACE_OCCLUDED"] або просто ["FACE_OCCLUDED"]. Також можна запитувати всі атрибути за допомогою ["ALL"]. Запит на більше атрибутів може збільшити час відповіді.

Якщо надати параметри ["ALL", "DEFAULT"], сервіс використовує логічний оператор AND для визначення атрибутів для повернення (у цьому випадку - всі атрибути).

Тип: Масив рядків

Допустимі значення: DEFAULT | ALL | AGE_RANGE | BEARD | EMOTIONS | EYE_DIRECTION | EYEGLASSES | EYES_OPEN | GENDER | MOUTH_OPEN | MUSTACHE | FACE_OCCLUDED | SMILE | SUNGLASSES

Обов'язково: Ні

ExternalImageId - Ідентифікатор, який ви хочете призначити всім виявленим обличчям на зображенні.

Тип: Рядок

Обмеження по довжині: Мінімальна довжина 1. Максимальна довжина 255.

Шаблон: [a-zA-Z0-9_.-:]+

Обов'язково: Ні

Image - Вхідне зображення у вигляді закодованих у форматі base64 байтів або об'єкта S3. Якщо використовувати AWS CLI для виклику операцій Amazon Rekognition, передача байтів зображення у форматі base64 не підтримується.

Якщо ж використовувати AWS SDK для виклику Amazon Rekognition, то може не знадобитися кодування байтів зображення у форматі base64 при використанні поля Bytes.

Тип: Об'єкт Image

Обов'язково: Так

MaxFaces - Максимальна кількість облич, які потрібно проіндексувати. Значення MaxFaces повинно бути більше або дорівнювати 1. IndexFaces повертає не більше 100 виявлених облич у зображенні, навіть якщо вказати більше значення для MaxFaces.

Якщо IndexFaces виявляє більше облич, ніж значення MaxFaces, то спочатку фільтруються обличчя з найнижчою якістю. Якщо облич залишається більше, ніж значення MaxFaces, то фільтруються обличчя з найменшими обмежувальними рамками (до кількості, необхідної для визначення значення MaxFaces). Інформація про непроіндексовані обличчя доступна в масиві UnindexedFaces.

Обличчя, які повертаються IndexFaces, впорядковані за розміром обмежувальної рамки обличчя від найбільшого до найменшого, у порядку зменшення.

MaxFaces може бути використане з колекцією, пов'язаною з будь-якою версією моделі обличчя.

Тип: Ціле число

Допустимий діапазон: Мінімальне значення 1.

Обов'язково: Ні

QualityFilter - Фільтр, який вказує якість обличчя для використання при фільтрації. Відфільтровані обличчя не індексуються. Якщо вказати AUTO, Amazon Rekognition вибирає панель якості. Якщо вказати LOW, MEDIUM або HIGH, фільтрація видаляє всі обличчя, які не відповідають вибраній панелі якості. Значення за замовчуванням - AUTO. Якість базується на різних загальних випадках використання. Детекція низької якості може статися з різних причин, таких як об'єкт, який неправильно ідентифікується як обличчя, розмите обличчя

або обличчя з позою, яка занадто екстремальна для використання. Якщо вказати NONE, фільтрація не виконується.

Для використання фільтрації якості, колекція, яка використовується повинна бути пов'язана з версією 3 моделі обличчя або вище.

Тип: Рядок

Допустимі значення: NONE | AUTO | LOW | MEDIUM | HIGH

Обов'язково: Ні.

Код підключення наведений в додатку В

3 ВЕРИФІКАЦІЯ ВИМОГ ТА ВИКОРИСТАННЯ СИСТЕМИ

3.1 Системні вимоги

Враховуючи те що інтерфейс даної системи реалізований у формі веб-сайту, то для її використання достатньо мінімальних системних вимог, а саме: наявність браузера та підключення до мережі Інтернет.

3.2 Використання системи в реальних умовах

Щоб почати користуватися сайтом, потрібно натиснути на відповідне посилання. Для того щоб отримати доступ до перегляду зображення з відеокамер, потрібно мати акаунт з відповідним доступом .



Увійти

Будь ласка заповніть всі поля

[Забули пароль?](#)

Рисунок 3.1 – Головна сторінка авторизації

Після вводу облікових даних користувача перемістить на головну сторінку де зображена карта та навігаційне меню

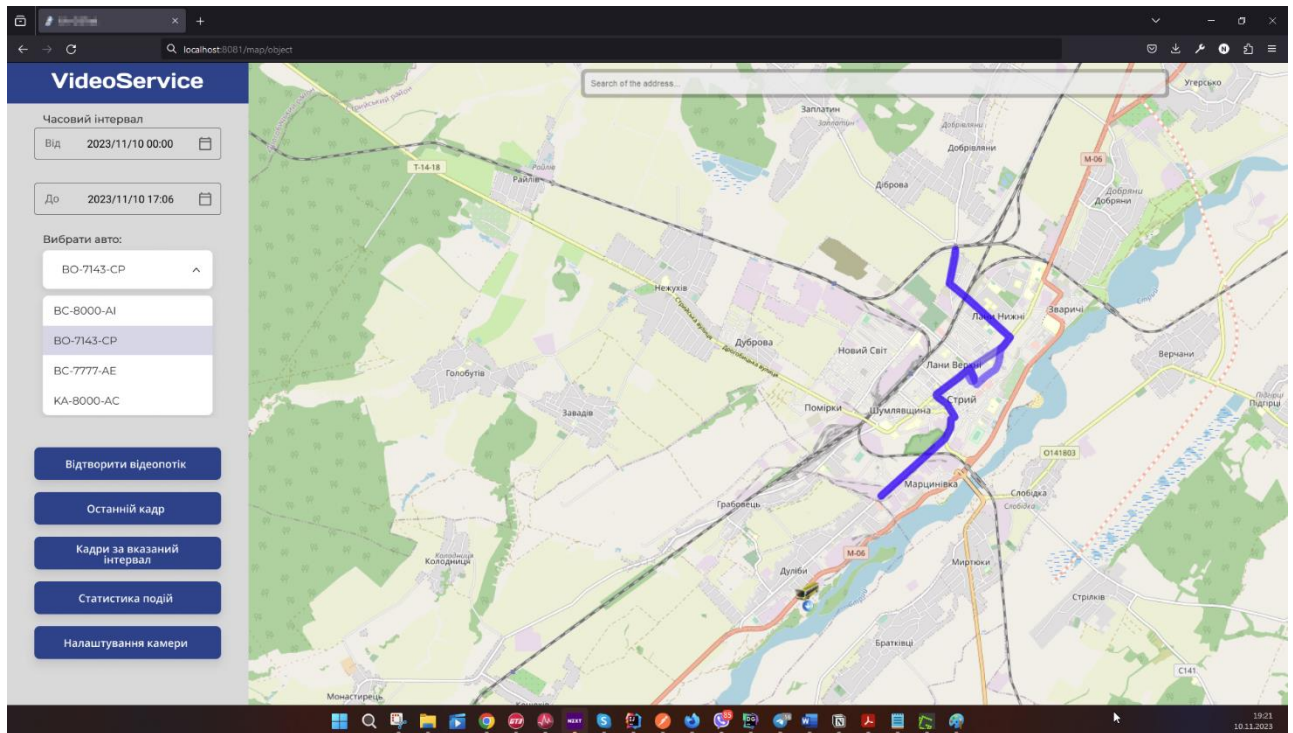


Рисунок 3.2 – Головна сторінка з картою

Зліва знаходиться навігаційне меню, справа – карта, де можна переглянути розташування транспортного засобу.

Якщо в селекторі часового інтервалу вказати потрібний час, вибрати ТЗ в якому є камера і натиснути кадри за вказаний інтервал, то відкриється карусель-галерея в якій можна переглянути всі кадри за вказаний період.

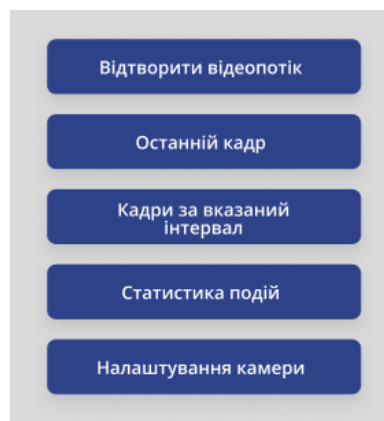


Рисунок 3.3 – Кнопки навігації

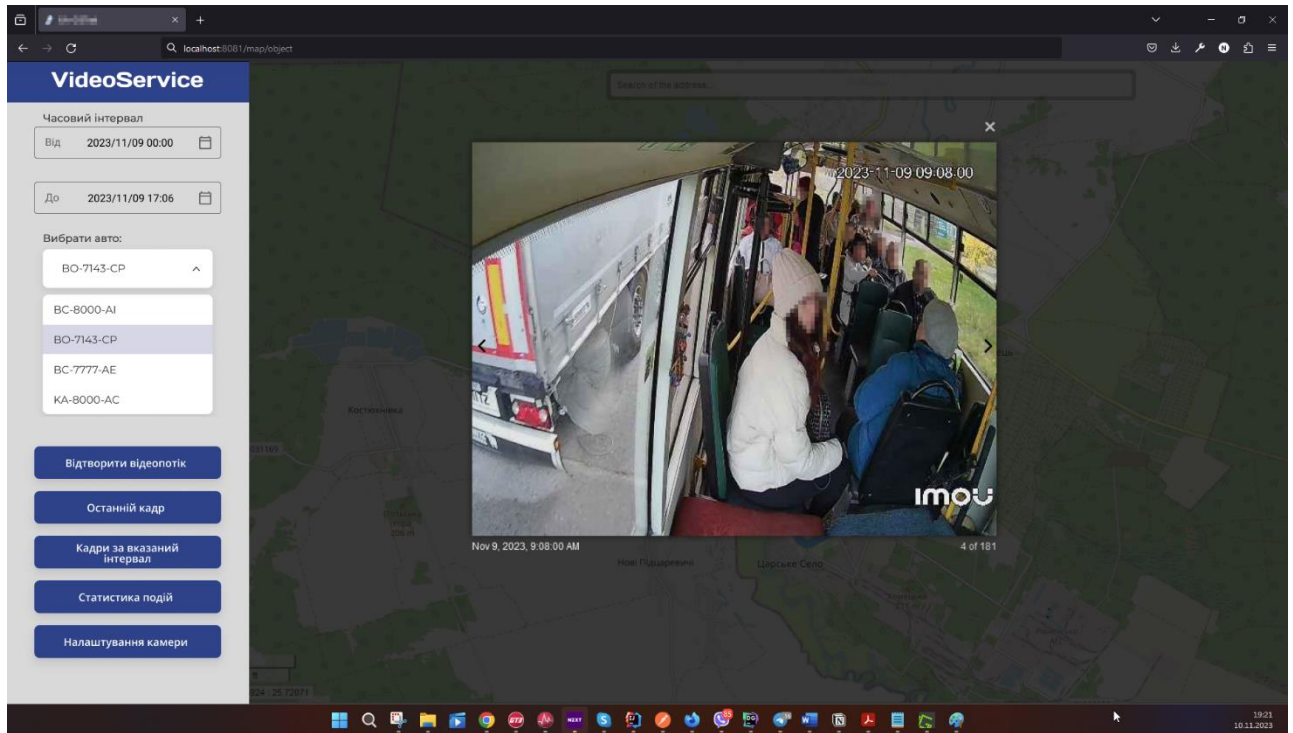


Рисунок 3.4 – Галерея кадрів

Якщо натиснути кнопку останній кадр, то відкриється останній кадр який був збережений для вибраного ТЗ

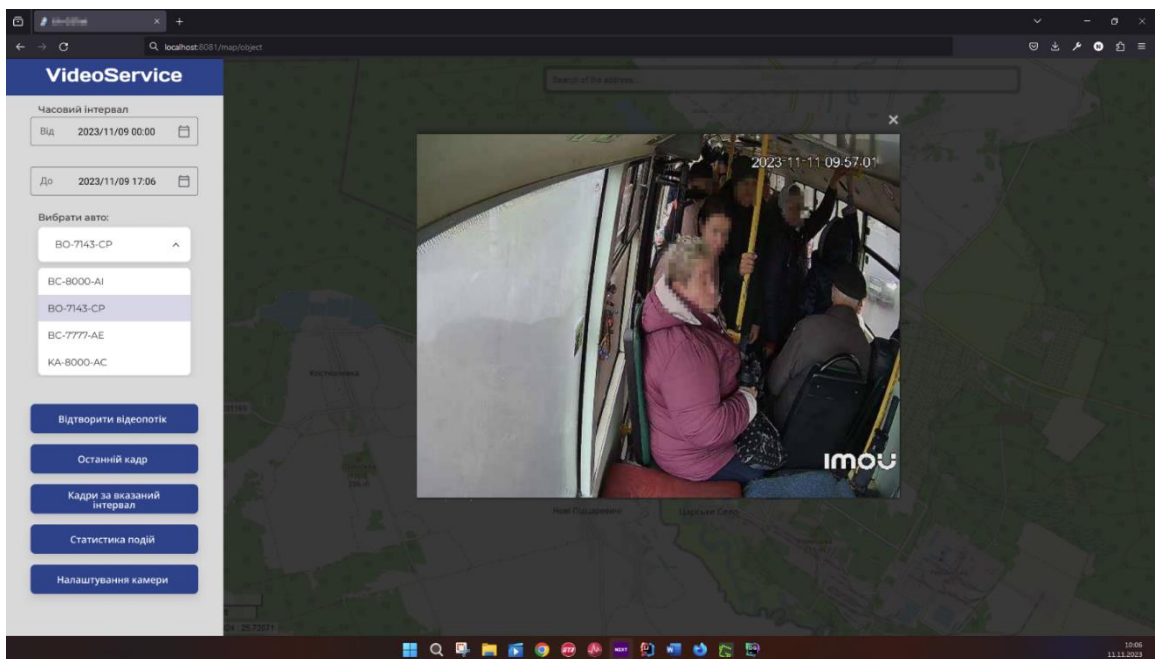


Рисунок 3.5 – Останній кадр

Натиснувши на кнопку «Відтворити відеопотік» ми відкриємо пряму трансляцію з вибраного ТЗ

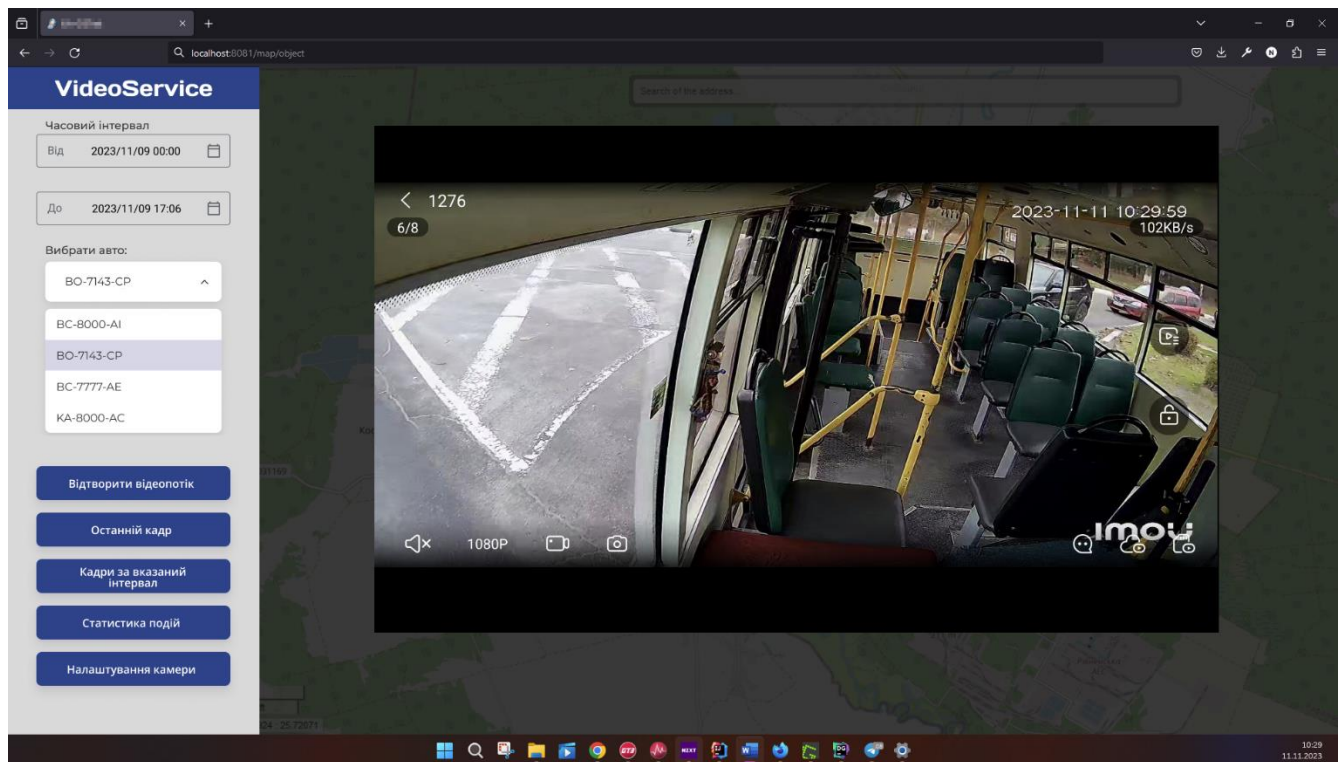


Рисунок 3.5 – Перегляд прямої трансляції

В цьому режимі є такі можливості:

- Перегляд прямої трансляції зі звуком
- Можливість зробити кліп або фото з вибраних моментів відео
- Переглянути історію збережених відео на SD картці
- Переглянути сповіщення подій виявлених ШІ

Якщо ми відкриємо меню сповіщень, то можна переглянути всі події які ШІ зафіксував за вибраний період часу.

Одні з можливих подій:

- Переповнений ТЗ
- Відкриті двері під час руху
- Водій курить
- Кількість пасажирів не співпадає з кількістю активних квитків

Всі ці події можна переглянути зразу після того як вони відбулися, диспетчери отримують сповіщення і можуть миттєво реагувати на ситуацію.

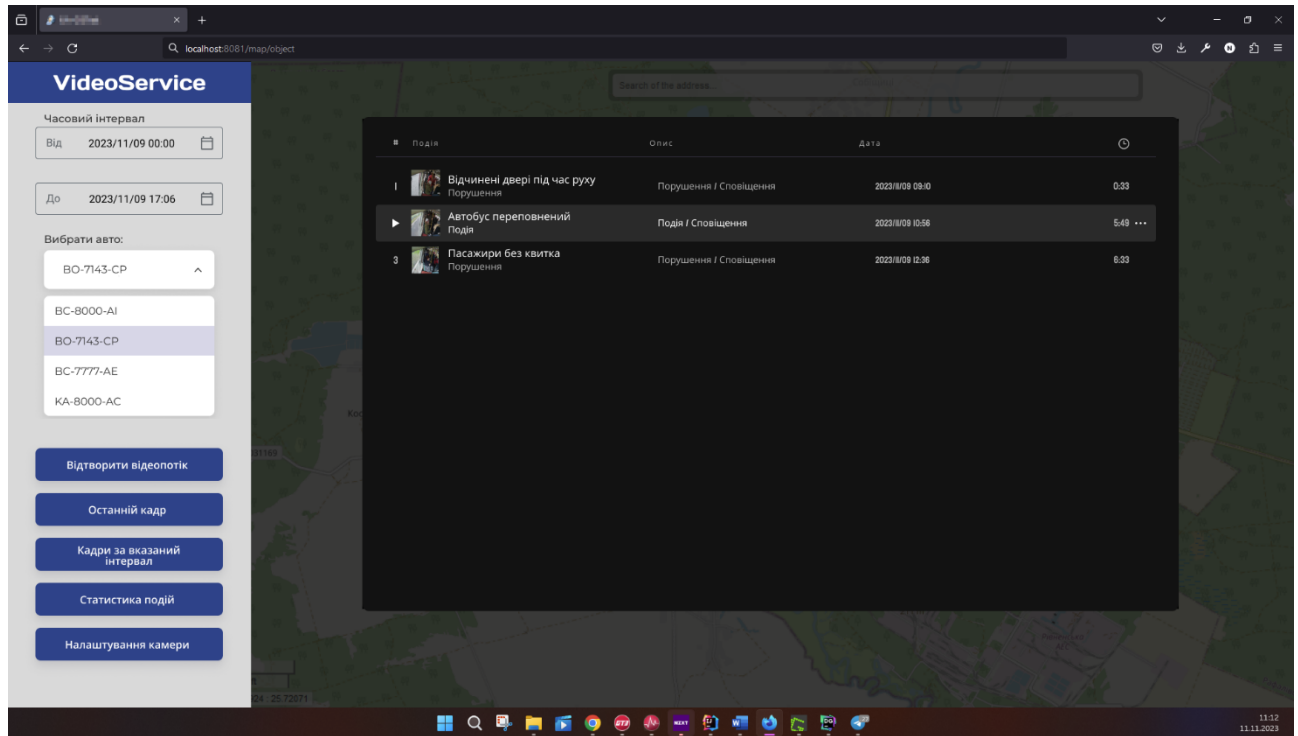


Рисунок 3.7 – Меню сповіщень III

Відкривши сповіщення «Автобус переповнений» ми справді можемо спостерігати що людей було більше ніж доступних місць в ТЗ

#	Подія	Опис	Дата	
1	Відчинені двері під час руху Порушення	Порушення / Сповіщення	2023/11/09 09:10	0:33
▶	Автобус переповнений Подія	Подія / Сповіщення	2023/11/09 10:56	5:49 ...
3	Пасажири без квитка Порушення	Порушення / Сповіщення	2023/11/09 12:36	6:33

Рисунок 3.8 – Список подій

Маючи цю інформацію, диспетчери можуть робити зміни у маршрутах і транспортних засобах, які їх обслуговують, для того щоб рівномірно розподіляти трафік пасажирів і уникати переповнених автобусів.

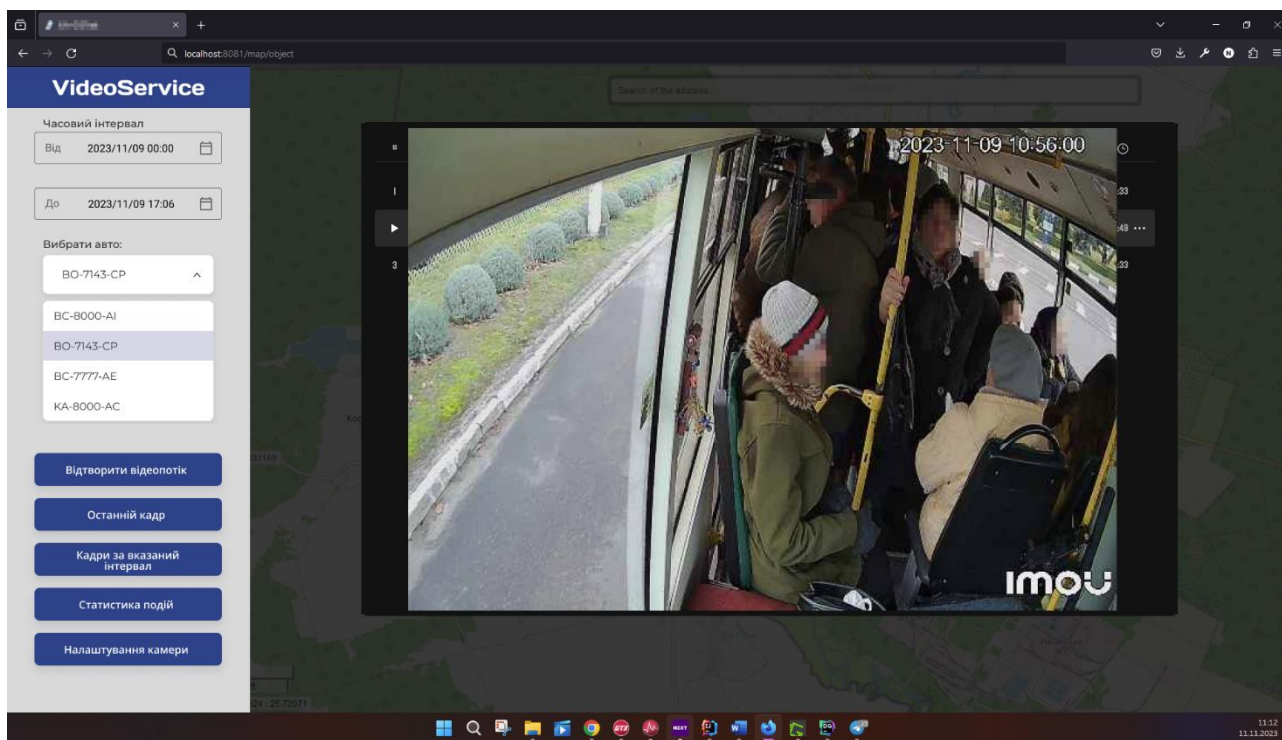


Рисунок 3.9 – Відкрита подія

У випадку якщо у нас є зафіксований випадок невідповідності кількості активних квитків і реальної кількості пасажирів в ТЗ, ми можемо переглянути цю подію.



Рисунок 3.10 – Відкрите порушення

Також є можливість переглянути статистику подій, де буде показана кількість певних подій за вказаний проміжок часу для вибраного ТЗ

Дата: 2023/11/09 09:00 - 2023/11/09 16:30
Об'єкт: ВС-8888-М

#	Подія	Опис	Кількість випадків	Перегляд
1	Відчинені двері під час руху Порушення	Порушення	2	▶
2	Пасажири без квитка Порушення	Порушення	3	▶
3	Водій курить Порушення	Порушення	1	▶
4	Переповнений автобус Подія	Подія	5	▶

Рисунок 3.11 – Статистика

Кожен список подій можна переглянути у вигляді кадрів або відео натиснувши на кнопку перегляд

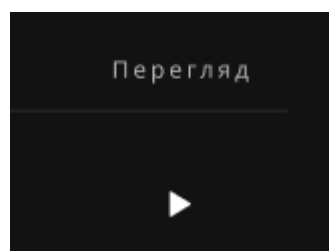


Рисунок 3.12 – Відтворення подій в статистиці

Останній функціонал навігаційного меню це налаштування камери, відкривши його з'являється можливість налаштувати такі параметри камери.

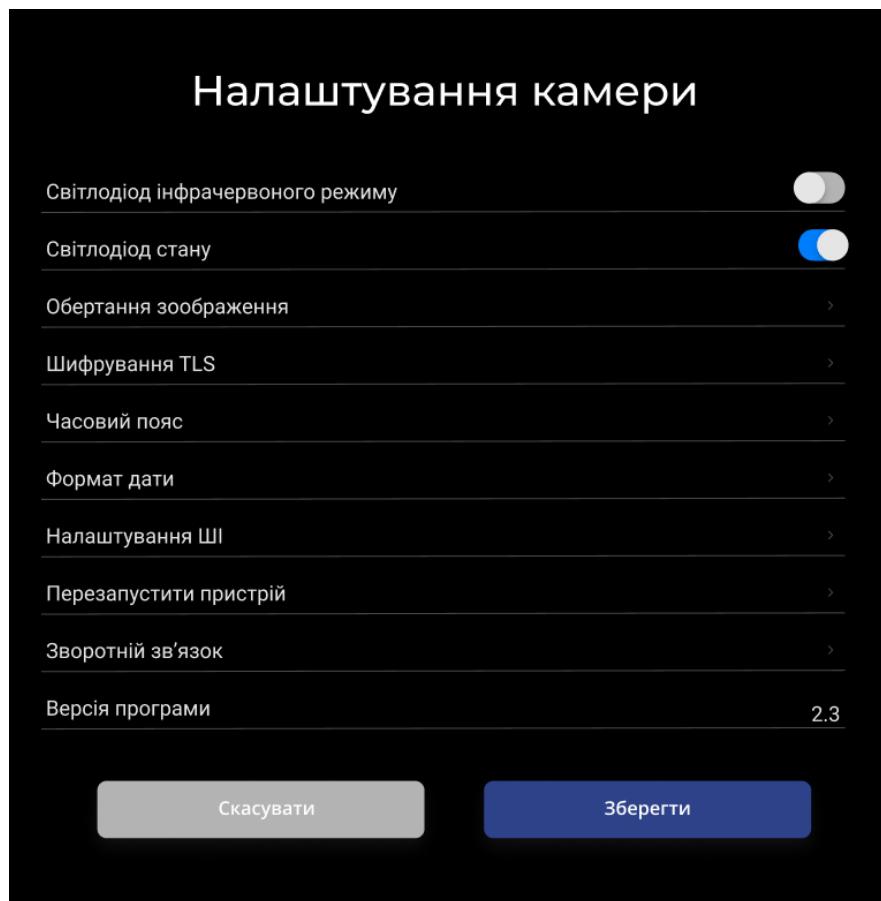


Рисунок 3.13 – Налаштування камери

IP камера має корисні налаштування, такі як включення інфрачервоного підсвічування для нічного бачення, стан камери показується через зелений або червоний світлодіод, можливість повертати зображення на різних кутах, шифрування для захисту даних, встановлення часового поясу та формату дати, а також налаштування штучного інтелекту для розпізнавання облич та руху. Камеру можна перезапустити, і є опція відправки відгуку виробнику. Також можна перевірити версію програмного забезпечення та скасувати або зберегти внесені зміни в налаштуваннях.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Охорона праці

Охорона праці на робочому місці також передбачає систематичний аналіз ризиків та впровадження заходів з їх зменшення або усунення. Робочий процес повинен бути організованим таким чином, щоб уникати можливих небезпек та забезпечувати безпеку працівників. Додатково, працівникам слід надавати відповідну інструкцію з використання обладнання та матеріалів, а також навчати їх правилам та процедурам у випадку надзвичайних ситуацій.

Ці заходи спрямовані на забезпечення не лише фізичної безпеки працівників, але й на створення здорового та безпечного робочого середовища. Регулярні перевірки та оновлення систем управління охороною праці допомагають підтримувати високі стандарти безпеки на роботі та запобігати можливим негативним наслідкам для працівників.

Загальні вимоги:

- Керуватися вимогами правил внутрішнього трудового розпорядку, інструкцій з охорони праці, пожежної безпеки та електробезпеки.
- Не допускати на своє робоче місце сторонніх осіб.
- Знати місця розташування аптечки та первинних засобів пожежогасіння і вміти користуватися ними.

4.2 Вимоги безпеки під час виконання роботи та вимоги до організації робочих місць користувачів комп'ютерів

Перед початком роботи фахівець повинен переконатися візуально у відсутності механічних ушкоджень корпусів екранного пристрою та засобів

оргтехніки, у наявності інженерно-технічних засобів безпеки, аптечки і засобів пожежогасіння.

Утримувати в порядку і чистоті робоче місце.

Стежити за справністю екранного пристрою, засобів оргтехніки та іншого устаткування, дотримуватися правил їх експлуатації.

Організація робочого місця користувача комп'ютера повинна відповідати вимогами НПАОП 0.00-1.31-10. Так, площа, на якій розташовується одне робоче місце з відео дисплейним терміналом (ВДТ), повинна становити не менше $6,0 \text{ м}^2$, а об'єм приміщення – не менше 20 м^3 . Робочі місця з ВДТ розміщуються на відстані не менше 1 м від стіни зі світловими прорізами; відстань між бічними поверхнями ВДТ має бути не менше 1,2 м; відстань між тильною поверхнею одного ВДТ та екраном іншого не повинна бути меншою за 2,5 м; прохід між рядами робочих місць має бути не менше метра. Необхідно також враховувати розміри меблів для комп'ютеризованих робочих місць, тобто висота 725 мм, ширина 600 – 1400 мм, глибина 800 – 1000 мм. Зокрема, розміри столу для ВДТ складають: ширина – 1200 мм, глибина – 800 мм.

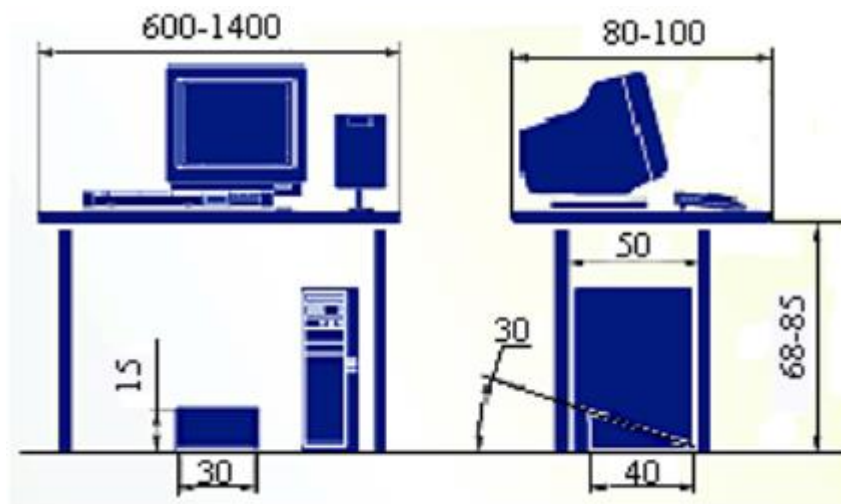


Рисунок 4.1 – Організація робочого місця (розміри наведено в см)

Особливу увагу необхідно звернути на розміщення відео терміналів. Для того щоб уникнути дзеркального відображення на екрані ВДТ джерел природного освітлення, їх необхідно розставити вздовж стіни з вікнами. З метою

зменшення потрапляння шуму з суміжних робочих місць та забезпечення концентрації уваги під час виконання робіт, що вимагають напруженості, необхідно відокремити робочі місця перегородками висотою 1,5 – 2 м. Вагомим фактором у забезпеченні безпеки праці користувачів комп'ютерів є характер розташування на робочому місці відео терміналу, клавіатури та принтера.

Розташування екрана (дисплея) повинно забезпечувати зручність зорового спостереження у вертикальній площині під кутом $\pm 30^\circ$ від лінії зору оператора. Найкращі зорові умови й можливість розпізнавання цифр, символів для клавіатури комп'ютера на поверхні робочого стола повинен бути простір для переміщення та поворотів. Положення клавіатури та кут її нахилу повинні відповідати побажанням користувача комп'ютера. Кут нахилу клавіатури може змінюватись у межах $5 - 10^\circ$. Якщо у конструкції клавіатури не передбачено простору для долонь, то їх потрібно розташовувати на відстані не менше 100 мм від краю стола в оптимальній зоні моторного поля. Допускається розташування клавіатури на спеціальній робочій поверхні, окремо від стола.

Розташування принтера або іншого пристрою введення-виведення інформації на робочому місці повинно забезпечувати добру видимість екрана комп'ютера, зручність ручного керування пристроєм введення-виведення інформації в зоні досяжності моторного поля (висоті 900 - 1300 мм, глибина 400 – 500 мм).

Після закінчення роботи відключити від електромережі екранний пристрій, засоби оргтехніки та інше устаткування, крім обладнання, яке використовується цілодобово.

Привести себе у порядок, вимити руки і обличчя та перевдягнутись

4.3 Вимоги безпеки в аварійних ситуаціях

При виникненні аварії або ситуації, яка може призвести до неї, потрібно вжити всіх можливих заходів з її усунення, попередити тих, хто поруч,

про небезпеку, повідомити про подію керівництво підприємства та діяти згідно з планом ліквідації аварії.

У випадку пожежі необхідно взяти заходів щодо евакуації працівників із приміщення відповідно до плану евакуації та негайно повідомити пожежну службу.

Якщо стався нещасний випадок і є потерпілі, а також при раптовому захворюванні працівника необхідно усунути дію на організм небезпечних та шкідливих факторів, які загрожують здоров'ю і життю потерпілого.

Передусім, безпека на роботі має бути пріоритетом для кожного працівника. Будь-яке порушення вимог безпеки може призвести до серйозних наслідків, включаючи травми ділянки, втручання у процес роботи і, в крайньому випадку, загрози життю.

ВИСНОВКИ

У цьому проєкті реалізовано сучасний підхід до забезпечення безпеки та комфорту пасажирів громадського транспорту. Використання штучного інтелекту та передових алгоритмів машинного навчання дозволяє системі ефективно відповідати на потреби користувачів. Технічна база проєкту, зокрема використання Spring, AWS та Java, підтверджує високий рівень професійності та надійності.

Інтеграція з AWS, зокрема з використанням Amazon Rekognition, гарантує ефективну обробку даних. Акцент на комфорті та безпеці пасажирів, використання засобів машинного навчання для виявлення небезпечних ситуацій та реагування на них, робить цей проєкт новим стандартом безпеки та комфорту для пасажирів громадського транспорту.

Переваги впровадження цього сервісу включають підвищення довіри пасажирів, збір даних для вдосконалення сервісу, запобігання злочинності та збирання доказів для інцидентів. Загальна мета полягає в створенні безпечного та ефективного середовища для користувачів громадського транспорту.

Завершуючи, важливо відзначити, що проєкт амбіційний у своїх планах на майбутнє. Планується активний розвиток системи, включаючи інтеграцію відеоспостереження з іншими системами моніторингу транспорту. Це дозволить ще більш ефективно виявляти потенційно небезпечні ситуації та підвищити загальний рівень безпеки. Проєкт залишається відкритим для нововведень, спрямованих на подальше збільшення комфорту та захищеності пасажирів громадського транспорту, що робить його ключовим удосконаленням для майбутніх викликів у цій сфері.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. What Is The Future Of Artificial Intelligence AI In Transportation? – [Електронний ресурс]: [<https://www.modeshift.com/what-is-the-future-of-artificial-intelligence-ai-in-transportation/>]
2. Junit Jupiter API from MvnRepository – [Електронний ресурс]: [<https://mvnrepository.com/artifact/org.junit.jupiter/junit-jupiter-api>].
3. What is Amazon Rekognition Image?. – [Електронний ресурс] : [<https://aws.amazon.com/rekognition/image-features/?nc=sn&loc=3&dn=3>].
4. Creating AWS video analyzer applications using the AWS SDK for Java – [Електронний ресурс] : [https://github.com/awsdocs/aws-doc-sdk-examples/tree/main/javav2/usecases/video_analyzer_application].
5. IndexFaces – [Електронний ресурс] – [https://docs.aws.amazon.com/rekognition/latest/APIReference/API_IndexFaces.html]
6. Adding faces to a collection – [Електронний ресурс] – [<https://docs.aws.amazon.com/rekognition/latest/dg/add-faces-to-collection-procedure.html>]
7. Detect people and objects in a video with Amazon Rekognition using an AWS SDK – [Електронний ресурс] – [https://docs.aws.amazon.com/rekognition/latest/dg/example_cross_RekognitionVideoDetection_section.html]
8. Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0508-18#Text>.
9. Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99 [Електронний ресурс]. – 1999. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/va042282-99#Text>.

10. Основні причини аварій літаків [Електронний ресурс]. – 2020. –
Режим доступу до ресурсу: <https://suspilne.media/7904-letiti-bez-strahu-vidpovidaemona-najposirenisi-zapitanna-pro-aviakatastrofi/>

11. Комп'ютерний зоровий синдром [Електронний ресурс]. – 2018. –
Режим доступу до ресурсу:
<https://linza.com.ua/uk/articles/blog/kompyuternyyzritelny-sindrom-simptomy-i-lechenie/>.