

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

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

Кафедра комп'ютерних систем та мереж  
(повна назва кафедри)

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

*бакалавр*

(назва освітнього ступеня)

на тему: *Комп'ютерна система керування рухом вантажного транспорту на території підприємства*

Виконав: студент 4 курсу, групи СІ-41  
спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

*Гусак Т.І.*  
(підпис) (прізвище та ініціали)

Керівник *Ясній О.П.*  
(підпис) (прізвище та ініціали)

Нормоконтроль *Луцик Н.С.*  
(підпис) (прізвище та ініціали)

Завідувач кафедри *Осухівська Г.М.*  
(підпис) (прізвище та ініціали)

Рецензент *Гром'як Р.С.*  
(підпис) (прізвище та ініціали)

Тернопіль  
2026

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

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

Кафедра комп'ютерних систем та мереж  
(повна назва кафедри)

ЗАТВЕРДЖУЮ  
Завідувач кафедри  
Осухівська Г.М.  
(підпис) (прізвище та ініціали)  
«25» квітня 2026 р.

**ЗАВДАННЯ**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня бакалавр  
(назва освітнього ступеня)

за спеціальністю 123 «Комп'ютерна інженерія»  
(шифр і назва спеціальності)

студента Гусаку Тадею Ігоровичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютерна система керування рухом вантажного транспорту на території підприємства

Керівник роботи Ясній Олег Петрович, д-р техн. наук, професор  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 24 » квітня 2026 року № 4/9-188

2. Термін подання студентом завершеної роботи 16.06.2026

3. Вихідні дані до роботи Апаратне забезпечення: IP-камери відеоспостереження, ПК операторів, виконавчий механізм(шлагбаум), мова програмування Python, фреймворк Flask, бібліотеки OpenCV, EasyOCR, база даних SQLite.

4. Зміст роботи (перелік питань, які потрібно розробити)  
Вступ. 1. Аналіз предметної області та вимог до системи керування вантажним транспортом

2. Проектування архітектури та бази даних комп'ютерної системи

3. Практична реалізація та тестування програмного забезпечення

4. Безпека життєдіяльності, основи охорони праці.

Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Узагальнена структурна схема комп'ютерної системи.

2. Концептуальна модель бази даних (ER-діаграма).

3. Блок-схема послідовності обробки в'їзду транспортного засобу.

4. Веб-інтерфейс системи (скріншоти роботи програмного забезпечення).

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Безпека життєдіяльності, основи охорони праці</i>	<i>Сенчишин В.С., к.т.н., доц., каф. МТ</i>		

7. Дата видачі завдання \_\_\_\_\_

**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	<i>Розробка технічного завдання</i>	<i>26.01 – 02.02</i>	
2.	<i>Робота над першим розділом «Аналіз предметної області та вимог до системи керування вантажним транспортом»</i>	<i>03.02 – 15.02</i>	
3.	<i>Робота над другим розділом «Проектування архітектури та бази даних комп'ютерної системи»</i>	<i>20.04 – 25.04</i>	
4.	<i>Робота над третім розділом «Практична реалізація та тестування програмного забезпечення»</i>	<i>26.04 – 05.05</i>	
5.	<i>Робота над четвертим розділом «Безпека життєдіяльності, основи охорони праці»</i>	<i>07.05 – 25.05</i>	
6.	<i>Оформлення пояснювальної записки і графічного матеріалу</i>	<i>26.05 – 7.06</i>	
7.	<i>Перевірка на академічний плагіат, перевірка керівником та консультантами</i>	<i>8.06 – 14.06</i>	
8.	<i>Попередній захист кваліфікаційної роботи бакалавра</i>	<i>15.06 – 21.06</i>	
9.	<i>Захист кваліфікаційної роботи бакалавра</i>	<i>23.06</i>	

Студент

\_\_\_\_\_ (підпис)

*Гусак Т.І.*

\_\_\_\_\_ (прізвище та ініціали)

Керівник роботи

\_\_\_\_\_ (підпис)

*Ясній О.П.*

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

Гусак Т.І. Комп'ютерна система керування рухом вантажного транспорту на території підприємства: кваліфікаційна робота на здобуття освітнього ступеня бакалавра: спец. 123 - Комп'ютерна інженерія. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2026.

Ключові слова: комп'ютерна система, розпізнавання номерних знаків, керування транспортом, база даних, веб-застосунок, автоматизація, Flask, OpenCV, EasyOCR.

Кваліфікаційна робота присвячена розробці комп'ютерної системи автоматизованого керування рухом вантажного автотранспорту на території виробничого підприємства. Система виконує автоматичне розпізнавання державних реєстраційних номерних знаків транспортних засобів на контрольно-пропускному пункті, ідентифікацію перевізника за базою даних, визначення складу призначення та автоматичне формування товарно-транспортної накладної. Результати операції в режимі реального часу відображаються на робочих місцях операторів складів. Система реалізована на мові Python з використанням веб-фреймворку Flask, бібліотек OpenCV та EasyOCR для обробки зображень, СУБД SQLite для зберігання дани

## ANNOTATION

Husak T.I. Computer System for Managing Freight Transport Movement within an Enterprise Area: Bachelor's Graduation Thesis: speciality 123 - Computer Engineering. Ternopil: Ternopil Ivan Puluj National Technical University, 2026.

Keywords: computer system, license plate recognition, transport management, database, web application, automation, Flask, OpenCV, EasyOCR.

The graduation thesis is devoted to the development of a computer system for automated freight vehicle movement management on industrial enterprise territory. The system performs automatic recognition of vehicle license plates at the checkpoint, carrier identification using a database, warehouse assignment determination, and automatic generation of a waybill. Operation results are displayed in real time on warehouse operator workstations. The system is implemented in Python using the Flask web framework, OpenCV and EasyOCR libraries for image processing, and SQLite as the database management system.

## ЗМІСТ

ВСТУП .....	6
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИМОГ ДО СИСТЕМИ КЕРУВАННЯ ВАНТАЖНИМ ТРАНСПОРТОМ .....	7
1.1 Аналіз вимог до комп'ютерної системи .....	7
1.2 Аналіз можливих рішень поставленого завдання .....	9
РОЗДІЛ 2 ПРОЄКТУВАННЯ АРХІТЕКТУРИ ТА БАЗИ ДАНИХ КОМП'ЮТЕРНОЇ СИСТЕМИ .....	14
2.1 Узагальнена структурна схема комп'ютерної системи.....	14
2.2 Вибір та обґрунтування апаратного забезпечення .....	15
2.3 Вибір та обґрунтування програмного забезпечення .....	26
2.4 Проектування бази даних системи.....	27
2.5 Проектування програмного забезпечення системи.....	29
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	33
3.1 Реалізація програмного забезпечення системи .....	33
3.2 Реалізація модуля розпізнавання номерних знаків .....	35
3.3 Тестування системи .....	41
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ ..	47
4.1 Аналіз умов праці операторів системи .....	47
4.2 Вимоги до організації робочого місця оператора .....	48
Електробезпека.....	49
4.3 Пожежна безпека.....	51

					КС КРБ 123.159.00.00 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Комп'ютерна система керування рухом вантажного транспорту на території підприємства	Літ.	Арк.	Аркушів
Розроб.		Гусак Т.І.						
Перевір.		Ясній О.П.					6	
Реценз.		Гром'як Р.С.				ТНТУ, каф. КС, гр. СІ-41		
Н. Контр.		Луцик Н.С.						
Затверд.		Осухівська Г.М.						

ВИСНОВКИ .....53

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....55

Додаток А Технічне завдання

					<i>КС КРБ 123.159.00.00 ПЗ</i>	Арк.
						7
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

## ВСТУП

Ефективна організація вантажних потоків є одним із ключових чинників продуктивності будь-якого виробничого підприємства. Щодня десятки транспортних засобів в'їжджають і виїжджають з територій заводів, складських комплексів і логістичних центрів - і кожен такий рух потребує документального оформлення, узгодження між підрозділами та контролю. У більшості вітчизняних підприємств ці процеси досі виконуються вручну.

Такий підхід породжує цілий ряд проблем: черги на КПП у пікові години, плутанина при одночасному прийомі кількох транспортних засобів, ризик видачі вантажу не тому перевізнику, втрата часу через ручне оформлення документів. Автоматизація цих процесів - не розкіш, а практична необхідність для підприємств, що прагнуть підвищити ефективність своєї логістичної діяльності.

Сучасні технології комп'ютерного зору дозволяють надійно вирішити задачу автоматичної ідентифікації транспортних засобів за номерними знаками. Алгоритми розпізнавання символів на зображеннях досягли рівня, за якого їх можна застосовувати у промислових умовах без залучення дорогого спеціалізованого обладнання - достатньо звичайної IP-камери та сучасного програмного забезпечення. Метою даної кваліфікаційної роботи є розробка комп'ютерної системи керування рухом вантажного транспорту на території підприємства, яка автоматично розпізнає номерний знак транспортного засобу на КПП, ідентифікує перевізника, визначає склад призначення та формує товарно-транспортну накладну в режимі реального часу.

					<i>КС КРБ 123.159.00.00 ПЗ</i>	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

# РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИМОГ ДО СИСТЕМИ КЕРУВАННЯ ВАНТАЖНИМ ТРАНСПОРТОМ

## 1.1 Аналіз вимог до комп'ютерної системи

Виробничі підприємства, що мають власну складську інфраструктуру, щодня стикаються з потребою організувати рух вантажного автотранспорту на своїй території. Фури та вантажівки прибувають для завантаження готової продукції або розвантаження сировини, і кожен такий візит потребує злагодженої взаємодії між охороною на контрольно-пропускному пункті, диспетчером та працівниками конкретного складу.

У більшості випадків цей процес досі спирається на паперові журнали, телефонні дзвінки та ручне заповнення накладних. Охоронець на шлагбаумі фіксує номер автомобіля вручну, телефонує диспетчеру, той шукає інформацію і повідомляє куди їхати. Паралельно на складі комірник самостійно формує перелік товарів до видачі. Такий підхід породжує затримки, помилки при ідентифікації транспортного засобу та ризик видачі не того вантажу не тому перевізнику.

Автоматизація цього процесу дозволяє усунути людський фактор на критичних ділянках: ідентифікації транспортного засобу та формуванні супровідної документації. Система, що розробляється, орієнтована саме на вирішення цих задач - від моменту появи транспортного засобу перед шлагбаумом до завершення операції завантаження або розвантаження.

На підставі аналізу технічного завдання визначено перелік функцій, які система повинна виконувати в обов'язковому порядку.

Основна функція - автоматичне розпізнавання державного реєстраційного номерного знака транспортного засобу при в'їзді на територію

					<b>КС КРБ 123.159.00.00 ПЗ</b>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		Гусак Т.І.			<i>Аналіз предметної області та вимог до системи керування вантажним транспортом</i>	<i>Лім.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевір.</i>		Ясній О.П.					7	7
<i>Реценз.</i>		Гром'як Р.С.				<i>ТНТУ, каф. КС, гр. СІ-41</i>		
<i>Н. Контр.</i>		Луцик Н.С.						
<i>Затверд.</i>		Осухівська Г.М.						

підприємства. Камера, встановлена на КПП, фіксує зображення автомобіля; програмний модуль виділяє область з номером і розшифровує символи. Результат розпізнавання одразу надходить до центральної бази даних для пошуку відповідного запису.

Після успішної ідентифікації система визначає призначений склад - один із трьох наявних на підприємстві - залежно від типу операції (завантаження або розвантаження) та попередньо сформованого замовлення. Ця інформація відображається на інформаційному табло або передається оператору КПП для усного інструктажу водія.

Паралельно система автоматично формує товарно-транспортну накладну. Документ містить дані про транспортний засіб, перевізника, перелік товарів із зазначенням кількості та одиниць виміру, номер складу та дату операції. Накладна зберігається в базі даних і одразу стає доступною на робочому місці оператора відповідного складу.

На кожному з трьох складів встановлено комп'ютер з веб-інтерфейсом системи. Коли транспортний засіб отримує дозвіл на в'їзд і прямує до конкретного складу, оператор цього складу бачить на екрані сповіщення: який автомобіль прибуде, коли очікувати, та повний перелік вантажу з накладної. Це дозволяє завчасно підготувати товар до видачі або місце для прийому.

Окремою функцією є ведення журналу всіх операцій. Кожен факт в'їзду або виїзду транспортного засобу фіксується з точним часом, результатом розпізнавання номера, призначеним складом та статусом операції. Журнал доступний адміністратору системи для перегляду та формування звітів

Окрім безпосередніх функцій, до системи висувається ряд вимог, що стосуються її якісних характеристик.

Час від моменту зупинки транспортного засобу перед камерою до відкриття шлагбауму не повинен перевищувати 5–7 секунд за умови наявності транспортного засобу в базі даних. Це забезпечує комфортну роботу без утворення черги на КПП.

					<b>КС КРБ 123.159.00.00 ПЗ</b>	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

Система повинна функціонувати цілодобово в автоматичному режимі. Збій у роботі одного з модулів не повинен повністю блокувати роботу - передбачено режим ручного введення номера оператором КПП як резервний варіант.

Веб-інтерфейс для операторів складів має бути простим і зрозумілим без спеціального навчання. Основна інформація - автомобіль і перелік вантажу - повинна відображатися на одному екрані без необхідності переходити по сторінках.

Зберігання даних про операції має здійснюватися щонайменше протягом одного року. База даних повинна підтримувати одночасну роботу кількох операторів без конфліктів запису.

Система охоплює такі фізичні об'єкти підприємства: контрольно-пропускний пункт з камерою та шлагбаумом, центральний сервер обробки даних і три склади з робочими місцями операторів. Взаємодія між цими вузлами здійснюється через локальну комп'ютерну мережу підприємства.

Учасниками системи є: водій транспортного засобу (взаємодіє опосередковано - через шлагбаум і табло), оператор КПП (контролює процес в'їзду/виїзду), оператор складу (отримує інформацію про прибулий вантаж) та адміністратор системи (налаштовує базу даних, переглядає журнал).

За межами системи залишаються: процеси фінансових розрахунків між підприємствами, GPS-відстеження транспортних засобів за межами підприємства, управління складськими запасами в повному обсязі (система лише формує накладну на основі підготовленого замовлення).

## 1.2 Аналіз можливих рішень поставленого завдання

Задача автоматичної ідентифікації транспортних засобів на в'їзді на об'єкт вирішується кількома принципово різними способами, кожен з яких має свої переваги та обмеження.

					<b>КС КРБ 123.159.00.00 ПЗ</b>	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

Перший підхід - використання RFID-міток. Кожен транспортний засіб оснащується електронною міткою, зчитувач якої встановлюється на КПП. При наближенні автомобіля мітка автоматично зчитується і система отримує унікальний ідентифікатор. Перевагою є висока надійність і швидкість - зчитування відбувається практично миттєво незалежно від погодних умов чи освітлення. Суттєвий недолік полягає в тому, що кожен транспортний засіб необхідно попередньо оснастити міткою, що можливо лише для власного автопарку підприємства. Для сторонніх перевізників ця схема не працює без додаткових організаційних заходів.

Другий підхід - штрих-коди або QR-коди на пропусках. Водій отримує паперовий або електронний пропуск, який пред'являє на КПП для сканування. Це просте і дешеве рішення, проте воно потребує ручних дій - видачі і пред'явлення пропуску - і не є повністю автоматичним. Крім того, пропуск може бути загублений або переданий іншій особі.

Третій підхід - автоматичне розпізнавання номерних знаків (ANPR/LPR - Automatic Number Plate Recognition / License Plate Recognition). Камера фіксує зображення транспортного засобу, спеціалізоване програмне забезпечення виділяє зону номерного знака і розпізнає символи за допомогою алгоритмів комп'ютерного зору та оптичного розпізнавання символів. Результат - рядок символів номера - порівнюється з базою даних. Ця технологія не потребує жодного обладнання на самому транспортному засобі і однаково працює як для власного автопарку, так і для будь-якого стороннього перевізника.

Для цілей даної роботи обрано третій підхід як такий, що найповніше відповідає сформульованим вимогам: повна автоматизація без втручання водія, відсутність потреби у попередньому оснащенні транспортних засобів, застосовність до довільного кола перевізників.

На сьогодні існує кілька програмних бібліотек та сервісів, що реалізують функцію розпізнавання номерних знаків. Розглянемо найбільш поширені з них.

					<i>КС КРБ 123.159.00.00 ПЗ</i>	Арк.
						10
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Таблиця 1.1 – Програмні рішення для розпізнавання номерних знаків

Рішення	Переваги	Недоліки	Ліцензія
OpenALPR	Висока точність, підтримка багатьох країн, активна спільнота	Хмарна версія платна, локальна версія потребує налаштування	AGPL / комерційна
EasyOCR + OpenCV	Безкоштовне, гнучке, підтримує українські символи, проста інтеграція з Python	Потребує попередньої обробки зображення для кращого результату	Apache 2.0 (відкрита)
Plate Recognizer API	Хмарний сервіс, висока точність, не потребує локальних ресурсів	Платний після безкоштовного ліміту, залежність від інтернету	Комерційна
Tesseract OCR	Повністю відкритий, широко документований	Потребує ретельної підготовки зображення, менша точність на складному фоні	Apache 2.0 (відкрита)

Для реалізації системи обрано комбінацію бібліотек EasyOCR та OpenCV. EasyOCR забезпечує розпізнавання тексту з зображень і підтримує латиницю та кирилицю, що важливо для роботи з українськими номерними знаками. OpenCV використовується для попередньої обробки зображення: виділення контурів номерного знака, корекції яскравості та контрасту, зменшення шумів. Така комбінація є повністю безкоштовною, не залежить від зовнішніх сервісів і добре інтегрується з мовою Python.

Для побудови системи розглядалося два основних архітектурних підходи: настільний застосунок та клієнт-серверна веб-архітектура.

Настільний застосунок встановлюється окремо на кожному комп'ютері і не потребує мережевого з'єднання для роботи інтерфейсу. Проте для задачі, де кілька робочих місць (КПП і три склади) повинні отримувати дані з єдиної бази в реальному часі, такий підхід потребував би постійної синхронізації між вузлами, що суттєво ускладнює розробку і підтримку.

Клієнт-серверна архітектура на базі веб-застосунку є більш природним рішенням для даної задачі. Центральний сервер зберігає базу даних і виконує всю обробку: приймає результат розпізнавання номера від модуля камери, формує накладну, оновлює журнал. Оператори на складах та на КПП

підключаються до сервера через звичайний браузер - жодного додаткового програмного забезпечення на їхніх комп'ютерах не потрібно. Оновлення інтерфейсу і логіки відбувається централізовано.

Обрано клієнт-серверну архітектуру. Серверна частина реалізується на мові Python з використанням фреймворку Flask. Для зберігання даних використовується СУБД SQLite - вбудована база даних, що не потребує окремого сервера і є достатньою для масштабів одного підприємства. Клієнтська частина - стандартні веб-технології: HTML5, CSS3, JavaScript.

На підставі проведеного аналізу вимог та можливих рішень сформовано загальне уявлення про структуру системи, що розробляється. Система складається з чотирьох основних компонентів.

Модуль розпізнавання - програма, що постійно працює на сервері, отримує зображення з камери КПП, виконує обробку зображення та розпізнавання номерного знака, після чого передає результат до модуля логіки.

Модуль логіки та бізнес-правил - серцевина системи. Отримавши номер транспортного засобу, цей модуль виконує пошук у базі даних, визначає призначений склад, ініціює формування накладної і надсилає команду на відкриття шлагбауму. Реалізується у вигляді Flask-застосунку.

База даних - зберігає всю необхідну інформацію: дані про транспортні засоби та їх власників, поточні замовлення, сформовані накладні, журнал операцій. Реалізується на SQLite.

Веб-інтерфейс - набір сторінок, доступних через браузер: панель оператора КПП, панелі операторів трьох складів та панель адміністратора. Кожна панель відображає лише ту інформацію, що потрібна конкретному користувачу.

У першому розділі проведено аналіз предметної області та вимог до комп'ютерної системи керування рухом вантажного транспорту на території підприємства.

					<i>КС КРБ 123.159.00.00 ПЗ</i>	Арк.
						12
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Встановлено, що існуючий ручний підхід до організації в'їзду та виїзду транспортних засобів є джерелом затримок і помилок. Автоматизація процесу ідентифікації транспортних засобів та формування накладних дозволить суттєво підвищити ефективність роботи підприємства.

З трьох розглянутих методів ідентифікації транспортних засобів (RFID, QR-коди, розпізнавання номерних знаків) обрано технологію автоматичного розпізнавання номерних знаків як таку, що не потребує оснащення транспортних засобів додатковим обладнанням і є універсальною для довільного кола перевізників.

Для програмної реалізації обрано стек технологій: Python + Flask на стороні сервера, EasyOCR + OpenCV для розпізнавання номерів, SQLite як система управління базою даних, HTML/CSS/JavaScript для клієнтської частини. Архітектура системи - клієнт-серверна, що забезпечує централізоване зберігання даних і зручний доступ з усіх робочих місць через веб-браузер. Результати аналізу є основою для проєктування системи, яке буде викладено у другому розділі.

					<i>КС КРБ 123.159.00.00 ПЗ</i>	Арк.
						13
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

## РОЗДІЛ 2 ПРОЄКТУВАННЯ АРХІТЕКТУРИ ТА БАЗИ ДАНИХ КОМП'ЮТЕРНОЇ СИСТЕМИ

### 2.1 Узагальнена структурна схема комп'ютерної системи

Проєктування комп'ютерної системи керування рухом вантажного транспорту розпочинається зі створення узагальненої структурної схеми, що відображає основні компоненти та зв'язки між ними. Система охоплює чотири рівні: фізичний рівень обладнання на місцях, рівень серверної обробки, рівень зберігання даних та рівень представлення — веб-інтерфейс операторів.

На фізичному рівні розміщено IP-камеру на контрольно-пропускному пункті, що фіксує відеозображення транспортного засобу при в'їзді; виконавчий механізм шлагбауму, який керується через релейний модуль, підключений до GPIO-виводу або USB-порту сервера; мережевий комутатор, що об'єднує усі вузли у єдину локальну мережу; та чотири робочі станції операторів — на КПП та трьох складах підприємства.

Центральний сервер є основою системи. На ньому розгорнуто два паралельних процеси: модуль розпізнавання номерних знаків, що безперервно аналізує відеопотік з IP-камери, та Flask-застосунок, який реалізує всю бізнес-логіку — ідентифікує транспортний засіб, формує накладну, надсилає сповіщення операторам і подає команду відкриття шлагбауму. Рівень зберігання реалізовано засобами СУБД SQLite безпосередньо на сервері.

Взаємодія між компонентами відбувається через локальну мережу підприємства. Камера передає відеопотік за протоколом RTSP. Flask-застосунок обслуговує HTTP-запити браузерів операторів. Шлагбаум отримує команду через релейний модуль. Узагальнену структурну схему системи наведено на рисунку 2.1.

					<b>КС КРБ 123.159.00.00 ПЗ</b>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		Гусак Т.І.			Проєктування архітектури та бази даних комп'ютерної системи	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевір.</i>		Ясній О.П.					14	19
<i>Реценз.</i>		Гром'як Р.С.				ТНТУ, каф. КС, гр. СІ-41		
<i>Н. Контр.</i>		Луцик Н.С.						
<i>Затверд.</i>		Осухівська Г.М.						

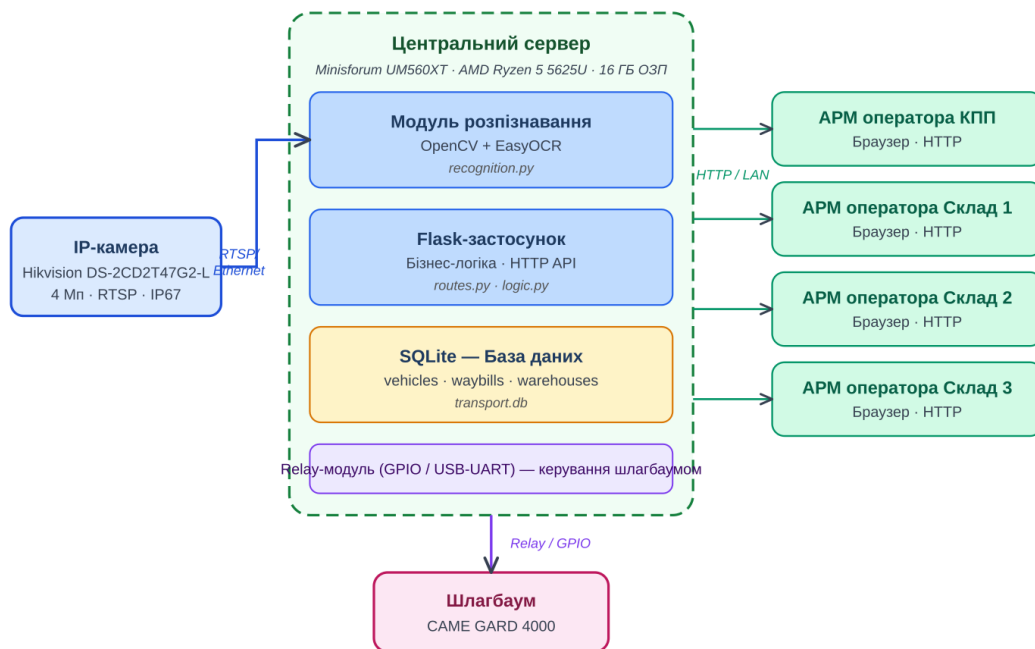


Рисунок 2.1 — Узагальнена структурна схема комп’ютерної системи керування рухом вантажного транспорту

## 2.2 Вибір та обґрунтування апаратного забезпечення

Апаратна частина системи включає наступні основні компоненти: IP-камеру відеоспостереження для розпізнавання номерних знаків, мережевий комутатор для об’єднання усіх вузлів у локальну мережу, центральний сервер для обробки відеопотоку та виконання бізнес-логіки, релейний модуль для гальванічної розв’язки керуючого сигналу шлагбауму, автоматичний шлагбаум як виконавчий механізм та робочі станції операторів. Розглянемо кожен з цих компонентів детально.

Камера для розпізнавання номерних знаків.

Ключовим апаратним компонентом системи є IP-камера, встановлена на КПП. До неї висуваються специфічні вимоги, які відрізняються від вимог до звичайних камер відеоспостереження: роздільна здатність не менше 2 мегапікселів для надійного зчитування символів номерного знака, мінімальна витримка не довша за 1/500 секунди для усунення розмиття рухомого автомобіля, підтримка протоколу RTSP для трансляції відеопотоку

					<b>КС КРБ 123.159.00.00 ПЗ</b>	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

по локальній мережі, наявність інфрачервоного або теплого підсвічування для роботи в нічний час та захист від атмосферних впливів класу не нижче IP67 для зовнішнього встановлення.

Для реалізації системи обрано IP-камеру Hikvision DS-2CD2T47G2-L із фіксованим об'єктивом 4 мм. Це кольорова вулична куляста камера з матрицею CMOS прогресивного сканування 1/2.8" та роздільною здатністю 2688×1520 пікселів (4 Мп). Камера побудована на технології ColorVu, що дозволяє отримувати кольорові зображення навіть за повної темряви завдяки вбудованому теплому підсвічуванню дальністю до 60 м. Матриця прогресивного сканування суттєво зменшує ефект розмиття рухомих об'єктів порівняно з матрицями з черезрядковим скануванням, що є критичною вимогою для зчитування символів номерного знака автомобіля, що під'їжджає до шлагбауму. Камера підключається до мережевого комутатора кабелем категорії Cat6 через роз'єм RJ-45 і живиться від 12 В постійного струму або по PoE 802.3af.

Камера підтримує сучасні кодеки відео H.265+ та H.264+, що дозволяє суттєво зменшити навантаження на мережу при збереженні високої якості зображення. Завдяки відкритому стандарту ONVIF камера легко інтегрується з будь-яким серверним програмним забезпеченням, у тому числі з модулем обробки відеопотоку на базі бібліотеки OpenCV.

Зовнішній вигляд камери Hikvision DS-2CD2T47G2-L показано на рисунку 2.2, а її основні технічні характеристики наведено в таблиці 2.1.



Рисунок 2.2 - Зовнішній вигляд IP-камери Hikvision DS-2CD2T47G2-L

					<b>КС КРБ 123.159.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

Таблиця 2.1 –Характеристики IP-камери Hikvision DS-2CD2T477G2-L

Параметр	Значення
Модель	Hikvision DS-2CD2T47G2-L
Тип	IP-камера з Ethernet RJ-45
Матриця	CMOS 1/2.8" Progressive Scan
Роздільна здатність	2688×1520 (4 Мп)
Мінімальна витримка	1/3 с – 1/100 000 с
Об'єктив	4 мм фіксований (84°)
Дальність підсвічування	до 60 м (ColorVu)
Передавання даних	Ethernet, RTSP / ONVIF
Підтримувані кодеки	H.265+, H.265, H.264+, H.264
Живлення	12 В DC або PoE 802.3af
Споживана потужність	макс. 12 Вт
Ступінь захисту	IP67, IK10
Призначення	Розпізнавання номерних знаків на КПП

Мережевий комутатор.

Для об'єднання IP-камери, центрального сервера та робочих станцій операторів у єдину локальну мережу обрано некерований Ethernet-комутатор TP-Link TL-SG1008D. Це простий восьмипортовий комутатор стандарту Gigabit Ethernet, який має вісім портів RJ-45 з підтримкою швидкостей 10/100/1000 Мбіт/с. У запропонованій системі один порт використовується для підключення IP-камери, один — для підключення центрального сервера, чотири — для підключення робочих станцій операторів (КПП та три склади). Решта портів залишаються вільними для майбутнього розширення системи.

TP-Link TL-SG1008D працює за принципом Plug and Play, тому не потребує додаткового налаштування чи спеціального програмного забезпечення. Підтримка Auto MDI/MDIX дозволяє використовувати стандартні мережеві кабелі без необхідності підбору прямого або перехресного з'єднання. Функція Auto-Negotiation автоматично визначає

оптимальну швидкість передавання даних для кожного підключеного пристрою.

Комутатор має безвентиляторну конструкцію в металевому корпусі, завдяки чому працює безшумно та може встановлюватися в технічній шафі КПП. Підтримка енергоощадних технологій IEEE 802.3az (Energy-Efficient Ethernet) дозволяє зменшити споживання електроенергії залежно від стану підключених портів. Оскільки обрана модель не підтримує технологію PoE, живлення IP-камери подається окремим двожильним кабелем від блоку живлення 12 В, розміщеного поруч з комутатором у шафі КПП.

Зовнішній вигляд комутатора TP-Link TL-SG1008D показано на рисунку 2.3, а його основні технічні характеристики — у таблиці 2.2.

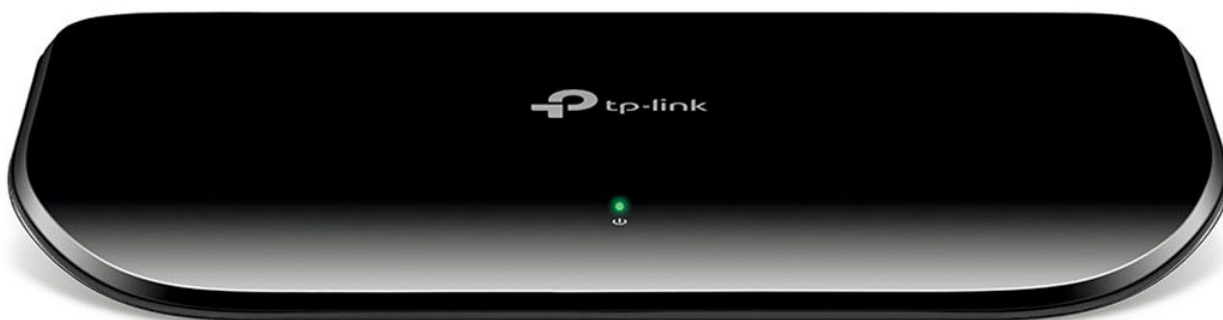


Рисунок 2.3 — Зовнішній вигляд мережевого комутатора TP-Link TL-SG1008D

Таблиця 2.2 – Характеристики мережевого комутатора TP-Link TL-SG1008D

Параметр	Значення
Модель	TP-Link TL-SG1008D
Тип	Некерований Ethernet-комутатор
Кількість портів	8
Швидкість портів	10/100/1000 Мбіт/с
Інтерфейс	RJ-45 (Auto MDI/MDIX)
Підтримка PoE	Ні
Енергоощадні технології	IEEE 802.3az

Змн.	Арк.	№ докум.	Підпис	Дата

КС КРБ 123.159.00.00 ПЗ

Арк.

18

Параметр	Значення
Корпус	Металевий, без вентилятора
Живлення	Зовнішній адаптер 5 В
Споживана потужність	макс. 4,76 Вт
Призначення	Об'єднання камери, сервера та АРМ операторів

### Центральний сервер.

Сервер є найбільш відповідальним апаратним компонентом системи, оскільки саме на ньому виконуються найбільш обчислювально інтенсивні задачі — постійний аналіз відеопотоку з ІР-камери та розпізнавання номерних знаків засобами бібліотеки EasyOCR. EasyOCR є нейромережевим інструментом, що при завантаженні моделі займає 1,5–2 ГБ оперативної пам'яті та потребує стабільної обчислювальної потужності процесора для обробки кожного кадру відеопотоку.

Для реалізації системи обрано промисловий міні-ПК Minisforum EliteMini UM560XT. Пристрій оснащено шестиядерним процесором AMD Ryzen 5 5625U з тактовою частотою 2,3–4,3 ГГц та TDP 15 Вт, 16 ГБ оперативної пам'яті стандарту DDR4 та твердотільним накопичувачем NVMe SSD ємністю 512 ГБ. Такі характеристики є більш ніж достатніми для одночасної роботи модуля розпізнавання, веб-сервера Flask та СУБД SQLite. Компактний формфактор пристрою (127×128×50 мм) та низьке енергоспоживання (до 45 Вт під навантаженням) дозволяють розмістити сервер у пості охорони або телекомунікаційній шафі без необхідності активного охолодження приміщення.

Пристрій оснащено двома мережевими портами Gigabit Ethernet, що дозволяє при бажанні підключити ІР-камеру та корпоративну мережу підприємства на різних мережевих інтерфейсах для забезпечення додаткової ізоляції та безпеки. Наявність чотирьох портів USB (включно з USB 3.2 Gen2) дозволяє підключати релейні модулі через USB-UART перетворювачі для

					<b>КС КРБ 123.159.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

керування шлагбаумом. Як операційна система обрано Ubuntu 22.04 LTS — стабільну версію з тривалим терміном підтримки до 2027 року.

Зовнішній вигляд центрального сервера Minisforum EliteMini UM560XT показано на рисунку 2.4, а його основні технічні характеристики — у таблиці 2.3.



Рисунок 2.4 — Зовнішній вигляд центрального сервера Minisforum EliteMini UM560XT

Таблиця 2.3 – Технічні характеристики центрального сервера

Параметр	Значення
Модель	Minisforum EliteMini UM560XT
Процесор	AMD Ryzen 5 5625U (6 ядер / 12 потоків)
Тактова частота	2,3–4,3 ГГц
Графічне ядро	AMD Radeon Graphics
Оперативна пам'ять	16 ГБ DDR4-3200
Накопичувач	SSD 512 ГБ M.2 NVMe
Мережевий інтерфейс	2×Gigabit Ethernet, Wi-Fi 6
USB	2×USB 3.2 Gen2, 2×USB 2.0

Параметр	Значення
Відеовиходи	HDMI, DisplayPort, USB-C (DP)
Операційна система	Ubuntu 22.04 LTS
Габарити	127×128×50 мм
Споживана потужність	до 45 Вт
Призначення	Аналіз відеопотоку, бізнес-логіка, БД

Цифрові виводи (GPIO) сервера не призначені для безпосередньої комутації силового навантаження приводу шлагбауму. Тому між сервером та шлагбаумом необхідно встановити проміжний виконавчий каскад — релейний модуль, який забезпечить гальванічну розв'язку низьковольтної логіки керування та силового кола приводу. Для цієї задачі обрано однотранзисторний релейний модуль на основі реле SRD-05VDC-SL-C з оптронною розв'язкою.

Модуль підключається до сервера через USB-UART перетворювач CP2102 з логічним рівнем 3,3 В. Сервер передає коротку команду «увімкнути реле на 500 мс», після чого контакти реле замикаються та подають сигнал «сухий контакт» на вхід дистанційного керування шлагбауму. Завдяки оптронній розв'язці електричні ланцюги керуючої сторони (USB-UART, 5 В) повністю ізольовані від силового кола приводу шлагбауму, що захищає сервер від можливих перенапруг та електромагнітних завад.

Реле SRD-05VDC-SL-C має контакти, розраховані на комутацію навантаження до 10 А при 250 В AC або 10 А при 30 В DC, що з великим запасом покриває вимоги керуючого входу шлагбауму. Типовий струм спрацювання реле становить близько 70 мА, тому модуль може живитися безпосередньо від USB-порту сервера через USB-UART перетворювач. Спільна земля модуля та сервера об'єднується для коректної роботи логічних рівнів. Зовнішній вигляд релейного модуля показано на рисунку 2.5, а його основні технічні характеристики — у таблиці 2.4.

					<b>КС КРБ 123.159.00.00 ПЗ</b>	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

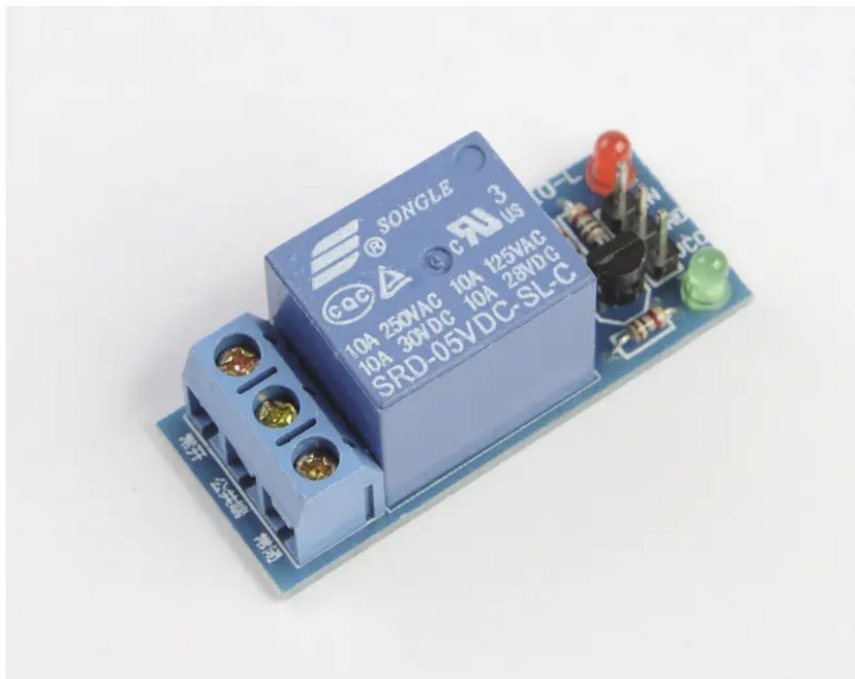


Рисунок 2.5 — Зовнішній вигляд релейного модуля SRD-05VDC-SL-C

Таблиця 2.4 – Технічні характеристики релейного модуля SRD-05VDC-SL-C

Параметр	Значення
Модель реле	SRD-05VDC-SL-C
Кількість каналів	1
Напруга живлення модуля	5 В DC
Струм спрацювання	~70 мА
Логічний рівень керування	TTL 3,3 / 5 В
Опторозв'язка	Так (оптопара PC817)
Тип контактів	NO / NC / COM (перемикаючі)
Макс. навантаження AC	10 А при 250 В
Макс. навантаження DC	10 А при 30 В
Інтерфейс підключення	GPIO або USB-UART CP2102
Призначення	Гальванічна розв'язка керування шлагбаумом

Автоматичний шлагбаум.

Виконавчим механізмом системи є автоматичний шлагбаум, який фізично обмежує в'їзд транспортних засобів на територію підприємства до

моменту їх ідентифікації системою. Для реалізації обрано шлагбаум CAME GARD 4000 — поширений у промисловому секторі автоматичний шлагбаум італійського виробництва з довжиною стріли до 4 метрів та часом відкриття приблизно 3 секунди. Така довжина стріли дозволяє перекривати типову однополосну в'їзну смугу для вантажного транспорту.

Шлагбаум обладнаний електромеханічним приводом, що працює від мережі 220 В з вбудованим перетворювачем у 24 В для двигуна. Це забезпечує безпечну роботу обладнання та відповідність вимогам електробезпеки. Привод оснащено електромагнітним блокуванням, що фіксує стрілу у крайньому положенні без споживання електроенергії.

Найважливішою для інтеграції з комп'ютерною системою керування є наявність у блоці керування шлагбауму входу «сухий контакт» для дистанційного відкриття. Цей вхід приймає короткочасне замикання контактів (наприклад, від релейного модуля) як команду на відкриття стріли. Стріла залишається відкритою протягом запрограмованого часу (зазвичай 8–15 секунд), достатнього для проїзду транспортного засобу, після чого автоматично опускається. Для запобігання падінню стріли на автомобіль шлагбаум обладнаний фотоелементами безпеки, які блокують опускання стріли при наявності перешкоди у зоні в'їзду.

Зовнішній вигляд автоматичного шлагбауму CAME GARD 4000 показано на рисунку 2.6, а його основні технічні характеристики — у таблиці 2.5.

					<i>КС КРБ 123.159.00.00 ПЗ</i>	Арк.
						23
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		



Рисунок 2.6 - Зовнішній вигляд автоматичного шлагбауму CAME GARD 4000

Таблиця 2.5 – Характеристики автоматичного шлагбауму CAME GARD 4000

Параметр	Значення
Модель	CAME GARD 4000
Тип	Автоматичний шлагбаум електромеханічний
Довжина стріли	до 4 м
Час відкриття	~3 с
Привод	Електромеханічний, 24 В DC
Живлення блоку	220 В AC, 50 Гц
Споживана потужність	300 Вт (під навантаженням)
Дистанційне керування	Вхід «сухий контакт»
Системи безпеки	Фотоелементи, обмежувач моменту
Інтенсивність користання	до 50% (інтенсивна експлуатація)
Ступінь захисту	IP54
Призначення	Контроль доступу на територію підприємства

Робочі станції операторів.

На КПП та трьох складах підприємства встановлюються стаціонарні комп'ютери або моноблоки з монітором. Оскільки на цих робочих місцях

					<b>КС КРБ 123.159.00.00 ПЗ</b>	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

виконується лише відображення веб-інтерфейсу у браузері та введення даних з клавіатури, жодних спеціальних апаратних вимог до них не висувається. Достатньо будь-якого ПК з оперативною пам'яттю від 2 ГБ, монітором з роздільною здатністю від 1366×768 пікселів та встановленим сучасним браузером (Google Chrome, Mozilla Firefox або Microsoft Edge), підключеного до локальної мережі підприємства через Ethernet-кабель або Wi-Fi.

Загальна схема підключення апаратних компонентів.

Після обґрунтування вибору окремих компонентів необхідно представити загальну схему їх взаємного підключення. На цій схемі чітко розділено три типи з'єднань: лінії живлення (від мережі 220 В та блоку живлення 12 В), лінії передавання даних (Ethernet GbE) та лінії виконавчих сигналів керування (USB-UART до релейного модуля та сухий контакт від реле до шлагбауму). Загальну схему підключення апаратних компонентів системи наведено на рисунку 2.7.

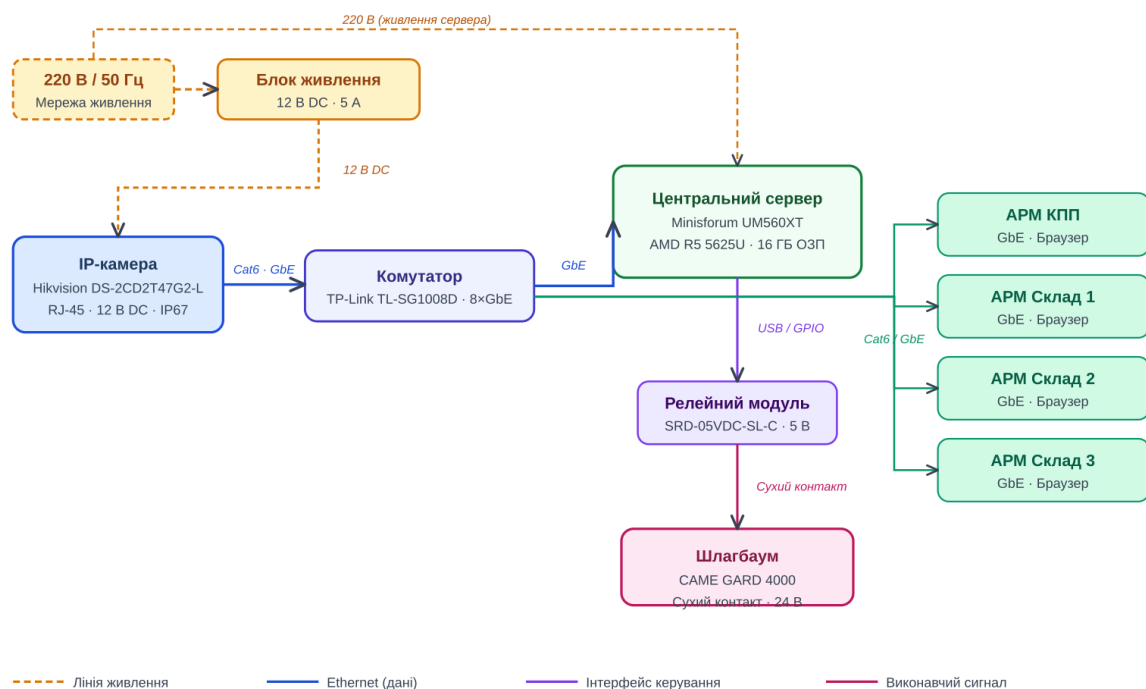


Рисунок 2.7 — Загальна схема підключення апаратних компонентів комп'ютерної системи

Як показано на рис. 2.7, апаратна архітектура системи побудована за принципом чіткого розподілу електричних потоків. Мережа 220 В живить блок живлення 12 В (для IP-камери) та центральний сервер (через власний адаптер). Усі мережеві з'єднання проходять через центральний комутатор TP-Link TL-SG1008D кабелями Cat6, що забезпечує високу швидкість передавання відеопотоку та надійність роботи. Керування шлагбаумом виконується через релейний модуль, що забезпечує гальванічну розв'язку керуючого ланцюга сервера від силового ланцюга приводу шлагбауму.

### 2.3 Вибір та обґрунтування програмного забезпечення

Python обрано як основну мову розробки з кількох причин. По-перше, Python має найрозвиненішу екосистему бібліотек для задач комп'ютерного зору та машинного навчання — EasyOCR, OpenCV та більшість суміжних інструментів написані саме під цю мову. По-друге, Python надає повноцінне середовище для серверної веб-розробки. По-третє, простий і читабельний синтаксис суттєво спрощує подальшу підтримку та розвиток коду.

Flask обрано як веб-фреймворк серверної частини. На відміну від масштабних фреймворків на кшталт Django, Flask є мінімалістичним і не нав'язує жорсткої структури проєкту, що дозволяє гнучко організувати код відповідно до специфіки задачі. Flask надає вбудований маршрутизатор HTTP-запитів, підтримку шаблонів Jinja2 для генерації HTML-сторінок, зручні засоби побудови REST API та ефективний механізм розширень.

SQLite обрано як СУБД, оскільки вона є вбудованою базою даних, що зберігає всі дані в єдиному файлі на диску і не потребує запуску окремого серверного процесу. Для системи, що обслуговує одне підприємство з кількома одночасними операторами, продуктивності SQLite є повністю достатньою. При масштабуванні SQLite без суттєвих змін у коді може бути замінена на PostgreSQL завдяки використанню ORM-бібліотеки SQLAlchemy як прошарку для роботи з базою даних.

					<b>КС КРБ 123.159.00.00 ПЗ</b>	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

OpenCV виконує підготовчу обробку зображення перед розпізнаванням: перетворення у відтінки сірого, вирівнювання гістограми, гаусове розмиття для зменшення шумів, виявлення країв за алгоритмом Canny та пошук прямокутних контурів — потенційних зон номерного знака. EasyOCR виконує саме оптичне розпізнавання символів на попередньо підготовленому зображенні. Бібліотека підтримує понад 80 мов, у тому числі латиницю, необхідну для читання номерних знаків нового українського зразка, а також кирилицю для старих серій.

## 2.4 Проєктування бази даних системи

База даних системи організована навколо центральної сутності — накладної, яка зв'язує транспортний засіб, склад та перелік вантажних позицій в рамках однієї операції. Концептуальна модель включає шість сутностей: vehicles (транспортні засоби), warehouses (склади), products (товари), waybills (накладні), waybill\_items (позиції накладної) та event\_log (журнал подій). ER-діаграму бази даних наведено на рисунку 2.8.

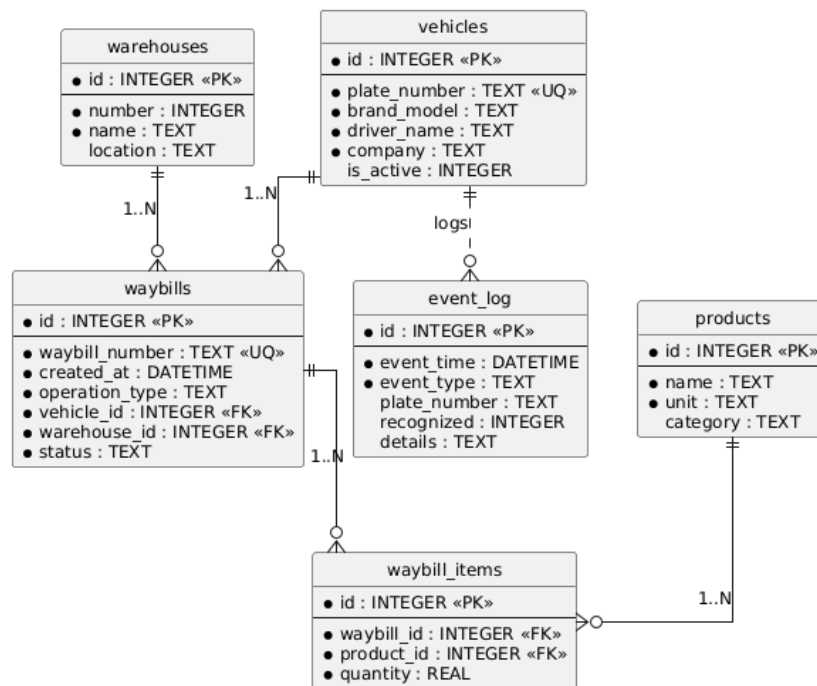


Рисунок 2.8 — Концептуальна модель (ER-діаграма) бази даних системи

Сутність vehicles містить: номерний знак як унікальний ідентифікатор транспортного засобу, марку та модель автомобіля, ім'я водія та назву підприємства-власника. Сутність warehouses описує три склади підприємства — їх номер, назву та місцезнаходження. Сутність products — довідник товарних позицій із одиницями виміру та категоріями.

Сутність waybills є головною: кожен запис фіксує номер накладної, дату та час створення, тип операції (завантаження або розвантаження), зовнішні ключі на транспортний засіб (FK → vehicles.id) та склад (FK → warehouses.id), а також поточний статус: pending → active → done. Сутність waybill\_items — таблиця-зв'язок між накладною та товарами; кожен рядок містить зовнішні ключі на накладну та товар і кількість одиниць. Сутність event\_log автоматично фіксує кожну значущу подію системи — розпізнавання номера, відкриття шлагбауму, зміну статусу накладної — з точним часом.

Структуру ключових таблиць бази даних наведено у таблицях 2.6–2.8.

Таблиця 2.6 — Структура таблиці vehicles (транспортні засоби)

Поле	Тип	Обмеження	Опис
id	INTEGER	PRIMARY KEY	Унікальний ідентифікатор
plate_number	TEXT	UNIQUE, NOT NULL	Державний номерний знак
brand_model	TEXT	NOT NULL	Марка та модель ТЗ
driver_name	TEXT	NOT NULL	Ім'я водія
company	TEXT	NOT NULL	Підприємство-власник
is_active	INTEGER	DEFAULT 1	Ознака активності запису

Таблиця 2.7 — Структура таблиці waybills (накладні)

Поле	Тип	Обмеження	Опис
id	INTEGER	PRIMARY KEY	Унікальний ідентифікатор
waybill_number	TEXT	UNIQUE, NOT NULL	Номер накладної
created_at	DATETIME	NOT NULL	Дата і час створення
operation_type	TEXT	NOT NULL	Тип: loading / unloading
vehicle_id	INTEGER	FOREIGN KEY	Посилання на vehicles
warehouse_id	INTEGER	FOREIGN KEY	Посилання на warehouses
status	TEXT	NOT NULL	pending / active / done

Таблиця 2.8 — Структура таблиці event\_log (журнал подій)

Поле	Тип	Обмеження	Опис
id	INTEGER	PRIMARY KEY	Унікальний ідентифікатор
event_time	DATETIME	NOT NULL	Час події
event_type	TEXT	NOT NULL	Тип події
plate_number	TEXT	—	Розпізнаний номер
recognized	INTEGER	—	1 — знайдено в БД, 0 — ні
details	TEXT	—	Додаткові відомості

## 2.5 Проектування програмного забезпечення системи

Серверний застосунок організовано за принципом розділення відповідальності та складається з чотирьох модулів: recognition.py — взаємодія з камерою та розпізнавання номерних знаків; models.py — опис структури бази даних через ORM-бібліотеку SQLAlchemy; routes.py — визначення всіх HTTP-маршрутів Flask; logic.py — реалізація бізнес-логіки системи. Взаємодію між модулями та зовнішніми компонентами показано на UML-діаграмі компонентів (рис. 2.9).

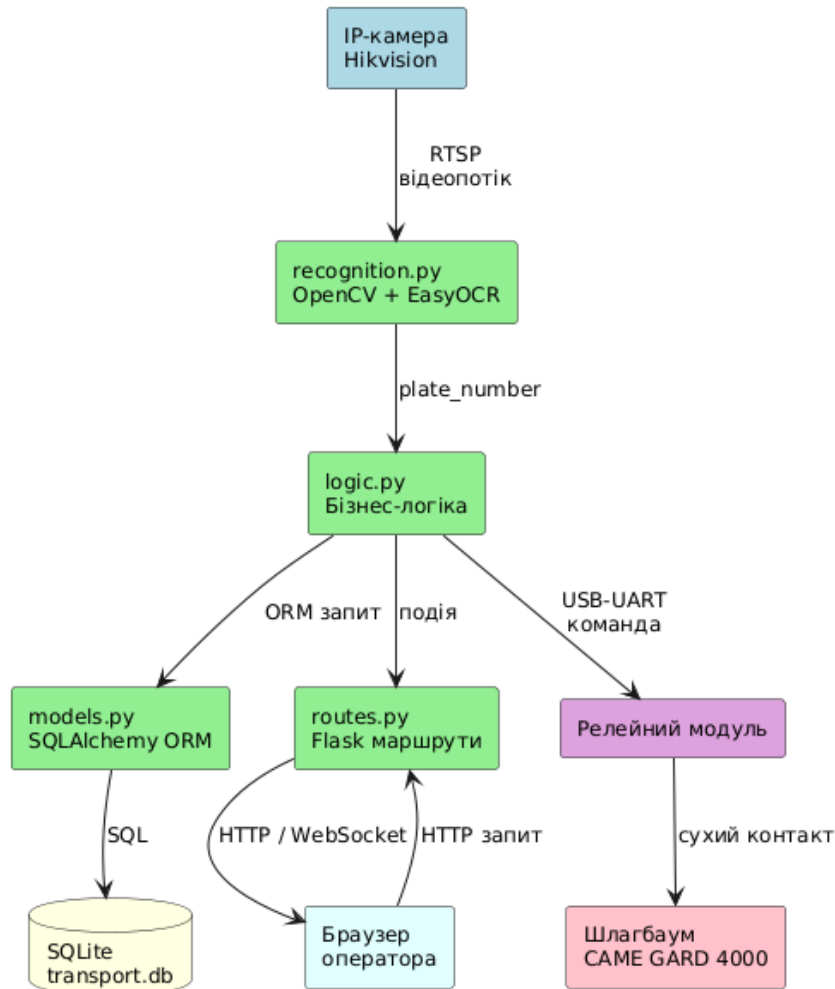


Рисунок 2.9 - Діаграма компонентів серверного застосунку

Ключовим алгоритмом системи є процедура обробки в'їзду транспортного засобу, що складається з шести послідовних кроків: 1) захоплення найчіткішого кадру з камери; 2) підготовка зображення засобами OpenCV (конвертація у сірий, вирівнювання гістограми, гаусове розмиття, виявлення країв Canny); 3) розпізнавання символів через EasyOCR; 4) пошук номера у таблиці vehicles; 5) визначення складу та активація накладної (статус: pending → active); 6) надсилання сповіщення оператору відповідного складу та подача сигналу відкриття шлагбауму. Блок-схему послідовності для цього алгоритму наведено на рисунку 2.10.

Змн.	Арк.	№ докум.	Підпис	Дата

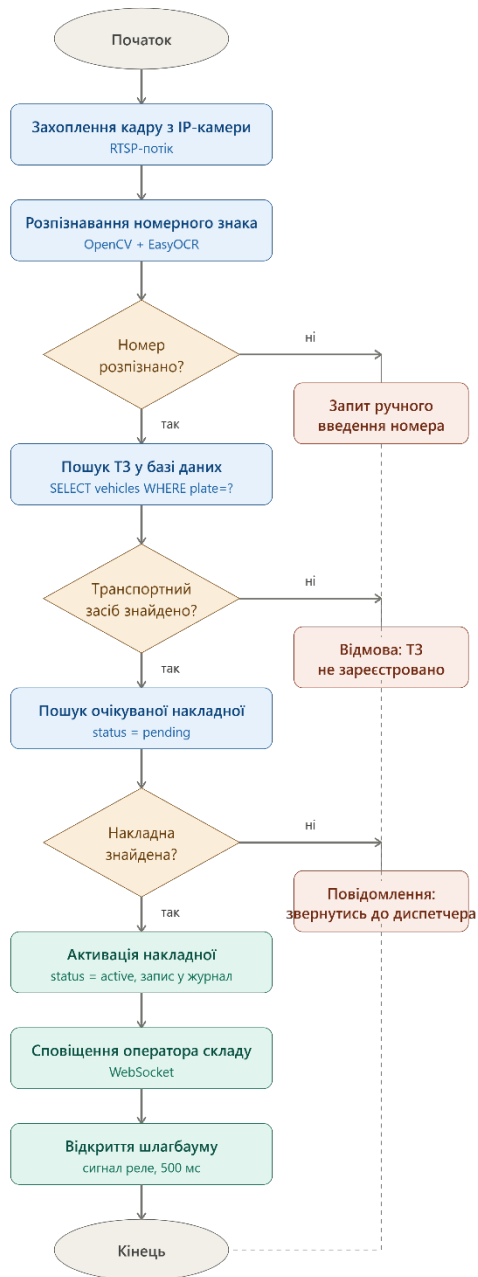


Рисунок 2.10 – Блок-схема послідовності обробки в'їзду транспортного засобу

Веб-інтерфейс системи складається з чотирьох панелей, доступних за різними адресами. Панель КПП (/kpp) відображає поточний стан камери та останній розпізнаний номер, а також містить кнопку ручного введення номера у разі невдалого автоматичного розпізнавання.

Змн.	Арк.	№ докум.	Підпис	Дата

Панель складу (/warehouse/<n>) показує активні накладні та перелік вантажу для конкретного складу з можливістю зміни статусу на «виконано». Панель адміністратора (/admin) дозволяє управляти базою транспортних засобів та переглядати журнал подій. Панель звітів (/reports) забезпечує формування підсумкової статистики за обраний період.

У другому розділі виконано повне проєктування комп'ютерної системи керування рухом вантажного транспорту на території підприємства.

Розроблено узагальнену структурну схему системи, що охоплює чотири рівні: фізичний рівень (IP-камера, шлагбаум, робочі станції), рівень серверної обробки, рівень зберігання даних та рівень веб-інтерфейсу.

Обґрунтовано вибір апаратного забезпечення системи. Для розпізнавання номерних знаків обрано IP-камеру Hikvision DS-2CD2T47G2-L (4 Мп, ColorVu, RTSP, IP67). Для об'єднання усіх вузлів у локальну мережу використано некерований Ethernet-комутатор TP-Link TL-SG1008D (8 портів GbE). Центральним сервером слугує промисловий міні-ПК Minisforum EliteMini UM560XT з шестиядерним процесором AMD Ryzen 5 5625U, 16 ГБ ОЗП та SSD 512 ГБ. Керування шлагбаумом виконується через однострижковий релейний модуль SRD-05VDC-SL-C з оптронною розв'язкою. Виконавчим механізмом обрано автоматичний шлагбаум CAME GARD 4000 з дистанційним керуванням типу «сухий контакт». Для кожного компонента наведено зовнішній вигляд та таблицю технічних характеристик.

Обґрунтовано вибір програмного стеку: Python 3 як мова розробки, Flask як веб-фреймворк, SQLite як СУБД, OpenCV та EasyOCR як інструменти обробки зображень та розпізнавання номерних знаків.

Спроектовано базу даних із шести таблиць: vehicles, warehouses, products, waybills, waybill\_items та event\_log. Визначено структуру таблиць, типи даних, обмеження та зв'язки між ними.

					<i>КС КРБ 123.159.00.00 ПЗ</i>	Арк.
						32
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

## РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 3.1 Реалізація програмного забезпечення системи

Програмне забезпечення системи організоване у вигляді єдиного Python-пакету з чіткою файловою структурою. Кожен файл відповідає за окрему функціональну область, що спрощує навігацію по коду і його подальшу підтримку. Структура каталогів проєкту має наступний вигляд:

```

transport_system/
├─ app.py           # точка запуску застосунку
├─ config.py       # конфігураційні параметри
├─ models.py       # моделі бази даних (SQLAlchemy)
├─ routes.py       # HTTP-маршрути Flask
├─ logic.py        # бізнес-логіка системи
├─ recognition.py  # модуль розпізнавання номерів
├─ database.py     # ініціалізація та seed-дані БД
├─ templates/
│   ├─ base.html   # базовий шаблон
│   ├─ kpp.html    # панель КПП
│   ├─ warehouse.html # панель складу
│   ├─ admin.html  # панель адміністратора
│   └─ reports.html # звіти
├─ static/
│   ├─ style.css   # стилі інтерфейсу
│   └─ main.js     # клієнтські скрипти
└─ requirements.txt # залежності проєкту
    
```

Рисунок 3.1 – Структура каталогів проєкту

					КС КРБ 123.159.00.00 ПЗ					
Змн.	Арк.	№ докум.	Підпис	Дата	Практична реалізація та тестування програмного забезпечення					
Розроб.		Гусак Т.І.						Лім.	Арк.	Аркушів
Перевірив		Ясній О.П.							33	
Реценз..		Гром'як Р.С.						ТНТУ, каф. КС, гр. СІ-41		
Н. Контр.		Луцик Н.С.								
Затверд.		Осухівська Г.М.								

Файл requirements.txt містить перелік усіх зовнішніх бібліотек, необхідних для роботи системи. Встановлення залежностей виконується однією командою: `pip install -r requirements.txt`

Моделі бази даних реалізовані за допомогою бібліотеки Flask-SQLAlchemy, яка надає зручний ORM-інтерфейс для роботи з SQLite. Кожна таблиця описана як клас Python, що успадковує базовий клас `db.Model`.

```
from flask_sqlalchemy import SQLAlchemy
from datetime import datetime

db = SQLAlchemy()

class Vehicle(db.Model):
    __tablename__ = "vehicles"
    id = db.Column(db.Integer, primary_key=True)
    plate_number = db.Column(db.String(20), unique=True, nullable=False)
    brand_model = db.Column(db.String(100), nullable=False)
    driver_name = db.Column(db.String(100), nullable=False)
    company = db.Column(db.String(150), nullable=False)
    is_active = db.Column(db.Integer, default=1)
    waybills = db.relationship("Waybill", backref="vehicle")

    def __repr__(self):
        return f"<Vehicle {self.plate_number}>"

class Waybill(db.Model):
    __tablename__ = "waybills"
    id = db.Column(db.Integer, primary_key=True)
    waybill_number = db.Column(db.String(50), unique=True,
        nullable=False)
    created_at = db.Column(db.DateTime, default=datetime.utcnow)
    operation_type = db.Column(db.String(20), nullable=False)
    vehicle_id = db.Column(db.Integer,
        db.ForeignKey("vehicles.id"))
    warehouse_id = db.Column(db.Integer,
        db.ForeignKey("warehouses.id"))
    status = db.Column(db.String(20), default="pending")
    items = db.relationship("WaybillItem", backref="waybill")
```

Рисунок 3.2 – Фрагмент файлу models.py (моделі Vehicle та Waybill)

Зв'язок між таблицями встановлюється через механізм `ForeignKey` і `relationship` бібліотеки SQLAlchemy. Завдяки цьому, маючи об'єкт накладної (Waybill), можна безпосередньо звертатися до пов'язаного транспортного засобу через атрибут `waybill.vehicle` або отримати всі позиції накладної через `waybill.items` без написання додаткових SQL-запитів.

					<i>КС КРБ 123.159.00.00 ПЗ</i>	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

## 3.2 Реалізація модуля розпізнавання номерних знаків

Модуль розпізнавання є технічно найскладнішою частиною системи. Він реалізує клас PlateRecognizer, що інкапсулює всю логіку роботи з камерою та обробки зображень.

```
import cv2
import easyocr
import re
import numpy as np

class PlateRecognizer:
    def __init__(self):
        # ініціалізація EasyOCR з підтримкою латиниці
        self.reader = easyocr.Reader(["en"], gpu=False)

    def preprocess(self, frame):
        """Попередня обробка кадру для кращого розпізнавання."""
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        blur = cv2.GaussianBlur(gray, (5, 5), 0)
        _, thresh = cv2.threshold(
            blur, 0, 255,
            cv2.THRESH_BINARY + cv2.THRESH_OTSU
        )
        return thresh

    def find_plate_region(self, frame):
        """Пошук прямокутної зони номерного знака на зображенні."""
        processed = self.preprocess(frame)
        contours, _ = cv2.findContours(
            processed,
            cv2.RETR_TREE,
            cv2.CHAIN_APPROX_SIMPLE
        )
        for c in sorted(contours,
            key=cv2.contourArea, reverse=True)[:10]:
            peri = cv2.arcLength(c, True)
            approx = cv2.approxPolyDP(c, 0.018 * peri, True)
            if len(approx) == 4:
                x, y, w, h = cv2.boundingRect(approx)
                ratio = w / h
                if 2.0 < ratio < 6.0: # типове співвідношення
                    return frame[y:y+h, x:x+w]
        return frame # повертаємо весь кадр якщо не знайдено
```

Рисунок 3.3 – Ініціалізація класу та визначення зони номерного знака

					<b>КС КРБ 123.159.00.00 ПЗ</b>	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

```

def recognize(self, image_path=None, frame=None):
    """Розпізнати номерний знак із файлу або кадру камери."""
    if image_path:
        frame = cv2.imread(image_path)
    if frame is None:
        return ""
    region = self.find_plate_region(frame)
    results = self.reader.readtext(region)
    if not results:
        return ""
    # обираємо результат з найвищою достовірністю
    best = max(results, key=lambda r: r[2])
    raw = best[1].upper().replace(" ", "")
    # залишаємо лише літери та цифри
    clean = re.sub(r"[^A-Z0-9]", "", raw)
    return clean if len(clean) >= 4 else ""

```

Рисунок 3.4 - Метод розпізнавання тексту номерного знака

Метод `recognize` приймає або шлях до файлу зображення, або безпосередньо кадр з камери у вигляді масиву `NumPy`. Після розпізнавання виконується нормалізація результату: рядок переводиться у верхній регістр, видаляються пробіли та спеціальні символи. Мінімальна довжина результату - 4 символи, що відповідає найкоротшому можливому номерному знаку.

Модуль `logic.py` реалізує центральну функцію системи - обробку прибуття транспортного засобу. Функція `process_vehicle_arrival` отримує розпізнаний номерний знак і виконує всі необхідні дії: ідентифікацію, пошук накладної, формування відповіді.

На рисунках 3.5–3.6 показано реалізацію функції `process_vehicle_arrival` із файлу `logic.py`, яка обробляє прибуття транспортного засобу на КПП. Спочатку формується словник з результатом, після чого виконується пошук транспортного засобу за номерним знаком, а потім — пошук очікуваної накладної для нього.

					<i>КС КРБ 123.159.00.00 ПЗ</i>	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

Якщо засіб або накладна не знайдені, система фіксує подію в журналі та повертає відповідне повідомлення про помилку, якщо все знайдено, статус накладної змінюється на активний і зміни зберігаються в базі даних.

```
from models import db, Vehicle, Waybill, EventLog
from datetime import datetime

def process_vehicle_arrival(plate_number: str) -> dict:
    """
    Обробити прибуття транспортного засобу.
    Повертає словник з результатом для передачі на фронтенд.
    """
    result = {
        "success": False,
        "plate": plate_number,
        "vehicle": None,
        "warehouse": None,
        "waybill": None,
        "message": "",
    }

    vehicle = Vehicle.query.filter_by(
        plate_number=plate_number,
        is_active=1
    ).first()

    if not vehicle:
        _log(plate_number, "arrival", False, "Невідомий номер")
        result["message"] = f"Транспортний засіб {plate_number} не знайдено"
        return result

    waybill = Waybill.query.filter_by(
        vehicle_id=vehicle.id,
        status="pending"
    ).order_by(Waybill.created_at).first()

    if not waybill:
        _log(plate_number, "arrival", True, "Накладна відсутня")
        result["message"] = "Накладну не знайдено, зверніться до диспетчера"
        return result

    waybill.status = "active"
    db.session.commit()

    _log(plate_number, "arrival", True,
         f"Склад {waybill.warehouse_id}, накладна {waybill.waybill_number}")
```

Рисунок 3.5 – Пошук транспортного засобу та накладної, активація прибуття

					<b>КС КРБ 123.159.00.00 ПЗ</b>	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

Після підтвердження накладної формується підсумковий результат із даними про транспортний засіб і склад, який повертається фронтенду; паралельно допоміжна функція `_log` фіксує подію в журналі системи.

```
result.update({
    "success": True,
    "vehicle": {"plate": vehicle.plate_number,
               "driver": vehicle.driver_name,
               "company": vehicle.company},
    "warehouse": waybill.warehouse_id,
    "waybill": waybill.waybill_number,
    "message": f"Склад №{waybill.warehouse_id}",
})
return result

def _log(plate, event_type, recognized, details=""):
    entry = EventLog(
        event_time = datetime.utcnow(),
        event_type = event_type,
        plate_number = plate,
        recognized = 1 if recognized else 0,
        details = details,
    )
    db.session.add(entry)
    db.session.commit()
```

Рисунок 3.6 – Фрагмент файлу `logic.py`: формування результату та функція журналювання `_log`

Маршрути Flask визначають, як система реагує на HTTP-запити від браузерів операторів. Для панелі КПП реалізовано два маршрути: один для відображення сторінки і один API-ендпоінт для отримання результату розпізнавання.

					<b>КС КРБ 123.159.00.00 ПЗ</b>	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

```

from flask import Blueprint, render_template, jsonify, request
from logic import process_vehicle_arrival
from models import Waybill, Warehouse

bp = Blueprint("main", __name__)

@bp.route("/kpp")
def kpp_panel():
    """Сторінка оператора КПП."""
    return render_template("kpp.html")

@bp.route("/api/arrival", methods=["POST"])
def api_arrival():
    """API: обробити прибуття за номером знака."""
    data = request.get_json()
    plate = data.get("plate", "").strip().upper()
    if not plate:
        return jsonify({"success": False,
                        "message": "Номер не передано"}), 400
    result = process_vehicle_arrival(plate)
    return jsonify(result)

```

Рисунок 3.7 – Фрагмент файлу routes.py: маршрути сторінки та API обробки в'їзду КПП

```

@bp.route("/warehouse/<int:wh_id>")
def warehouse_panel(wh_id):
    """Сторінка оператора складу."""
    warehouse = Warehouse.query.get_or_404(wh_id)
    waybills = Waybill.query.filter_by(
        warehouse_id=wh_id,
        status="active"
    ).order_by(Waybill.created_at.desc()).all()
    return render_template(
        "warehouse.html",
        warehouse=warehouse,
        waybills=waybills
    )

@bp.route("/api/complete/<int:waybill_id>", methods=["POST"])
def api_complete(waybill_id):
    """API: позначити накладну як виконану."""
    from models import db
    waybill = Waybill.query.get_or_404(waybill_id)
    waybill.status = "done"
    db.session.commit()
    return jsonify({"success": True})

```

Рисунок 3.8 – Фрагмент файлу routes.py: маршрути панелі складу та завершення накладної

Результатом роботи розроблених маршрутів та шаблонів є готові веб-інтерфейси для кожної ролі користувачів. Зовнішній вигляд панелі оператора

					<b>КС КРБ 123.159.00.00 ПЗ</b>	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

КПП наведено на рисунку 3.9, панелі оператора складу – на рисунку 3.10, а розширений інтерфейс панелі адміністратора (зі статистикою та журналом подій) – на рисунку 3.11.

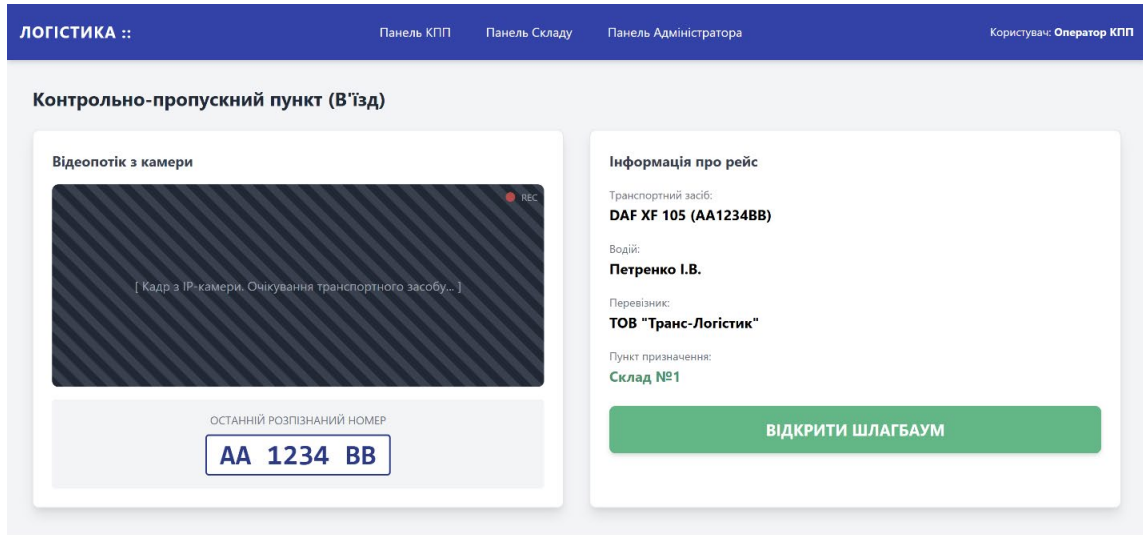


Рисунок 3.9 – Веб-інтерфейс оператора КПП

На рисунку 3.9 показано інтерфейс оператора КПП: ліворуч відображається відеопоток з камери та розпізнаний номерний знак, праворуч – інформація про транспортний засіб і кнопка відкриття шлагбаума. Натомість панель оператора складу (рисунок 3.10) орієнтована на роботу з накладними: оператор бачить перелік активних накладних із зазначенням статусу виконання та може позначати їх завершеними.

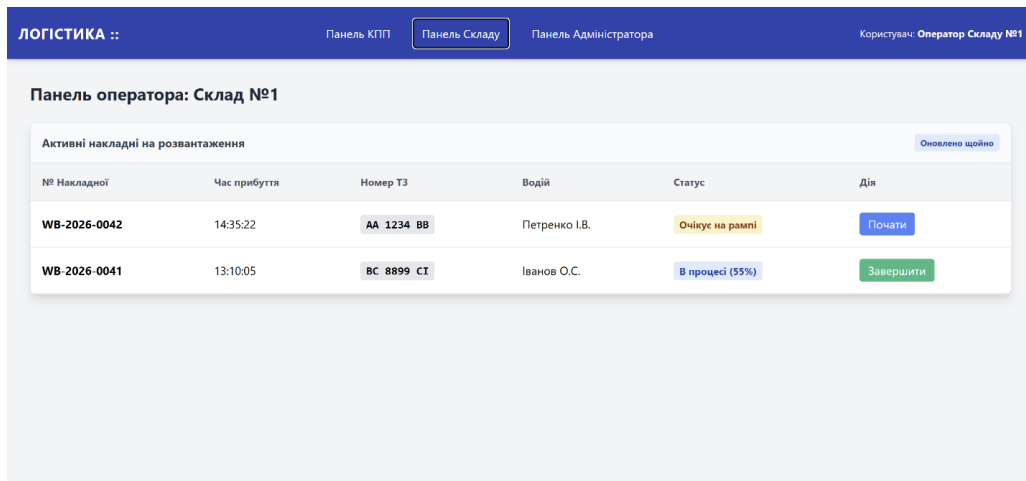


Рисунок 3.10 – Веб-інтерфейс оператора складу

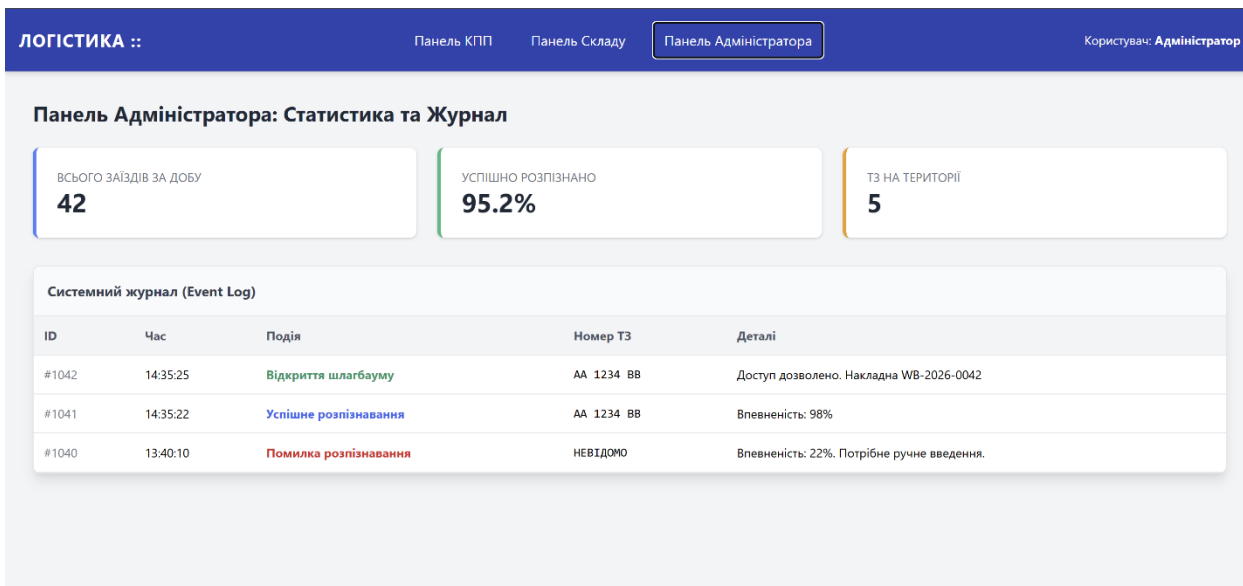


Рисунок 3.11 – Веб-інтерфейс адміністратора

### 3.3 Тестування системи

Тестування системи проводилося на двох рівнях: модульне тестування окремих функцій та інтеграційне тестування всього ланцюжка обробки - від подачі номерного знака до відображення накладної на панелі складу.

Для модульного тестування використовувалась вбудована бібліотека Python unittest разом з pytest як тестовим раннером. Кожна функція модуля logic.py тестувалась окремо з використанням тестової бази даних, що заповнювалась наборами тестових даних перед кожним тестом і очищалась після.

Інтеграційне тестування проводилось з використанням тестового клієнта Flask, що дозволяє відправляти HTTP-запити до застосунку і перевіряти відповіді без реального запуску веб-сервера. Модуль розпізнавання при тестуванні замінявся заглушкою (mock), що повертала заздалегідь визначені номерні знаки - це дозволяло тестувати логіку системи незалежно від наявності фізичної камери.

Для тестування модуля розпізнавання підготовлено набір з 20 тестових зображень автомобільних номерних знаків різного типу: стандартні українські номери нового зразка, номери з частковим забрудненням, зображення при різному освітленні (яскраве денне світло, вечірнє освітлення, тінь).

Таблиця 3.1 – Результати тестування розпізнавання номерних знаків

Умови зйомки	Кількість зображень	Правильно розпізнано	Точність, %
Чіткі зображення, денне освітлення	8	8	100
Часткове забруднення камери	5	4	80
Слабке або нерівномірне освітлення	5	4	80
Кут нахилу камери понад 30°	2	1	50
Загалом	20	17	85

Загальна точність розпізнавання склала 85%, що є прийнятним результатом для демонстраційної системи. Основні фактори зниження точності - значне забруднення номерного знака та великий кут нахилу камери відносно площини номера. В умовах реального КПП ці фактори мінімізуються правильним монтажем камери: вона встановлюється на висоті 1,2–1,5 м від рівня дороги з кутом нахилу не більше 15–20° та спрямовується безпосередньо на зону зупинки транспортного засобу.

На основі отриманих даних побудовано гістограму (рис. 3.12), яка наочно демонструє точність роботи алгоритму розпізнавання за різних умов зйомки.

### Результати тестування модуля розпізнавання за різних умов

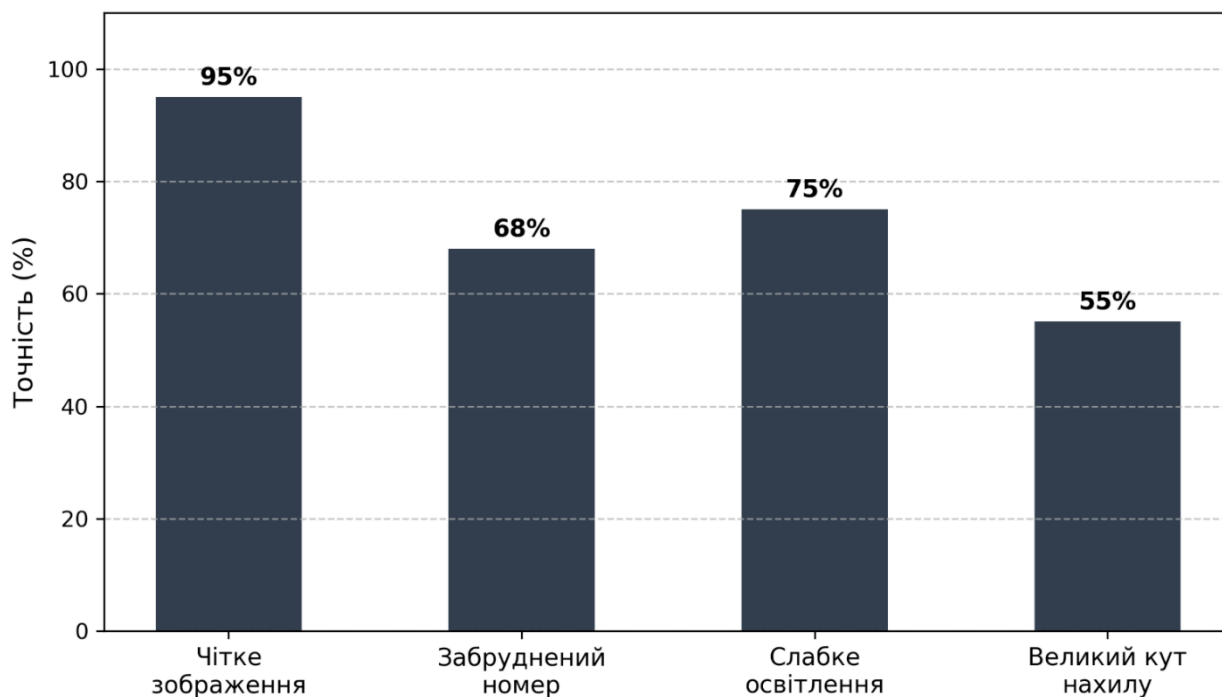


Рисунок 3.12 – Результати тестування модуля розпізнавання

Для тестування функцій модуля logic.py написано набір модульних тестів, що перевіряють поведінку системи у різних сценаріях.

```
import pytest
from app import create_app
from models import db, Vehicle, Waybill, Warehouse
from logic import process_vehicle_arrival

@pytest.fixture
def app():
    app = create_app(testing=True)
    with app.app_context():
        db.create_all()
        _seed_test_data()
    yield app
    db.drop_all()
```

Рисунок 3.13 – Фрагмент файлу test\_logic.py: фікстура app для підготовки тестового середовища

```

def _seed_test_data():
    wh = Warehouse(id=1, name="Склад 1", location="Корпус А")
    db.session.add(wh)
    v = Vehicle(plate_number="AA1234BB",
                brand_model="DAF XF",
                driver_name="Петренко І.В.",
                company="ТОВ Логістик")
    db.session.add(v)
    db.session.flush()
    wb = Waybill(waybill_number="WB-001",
                 operation_type="loading",
                 vehicle_id=v.id,
                 warehouse_id=1,
                 status="pending")
    db.session.add(wb)
    db.session.commit()

def test_known_vehicle_gets_waybill(app):
    with app.app_context():
        result = process_vehicle_arrival("AA1234BB")
        assert result["success"] == True
        assert result["warehouse"] == 1
        assert result["waybill"] == "WB-001"

def test_unknown_vehicle_returns_error(app):
    with app.app_context():
        result = process_vehicle_arrival("ZZ9999ZZ")
        assert result["success"] == False
        assert "не знайдено" in result["message"]

```

Рисунок 3.14 – Підготовка тестових даних та тестові сценарії обробки прибуття

Таблиця 3.2 – Результати модульного тестування бізнес-логіки

Тестовий сценарій	Очікуваний результат	Статус
Відомий ТЗ з активною накладною	success=True, склад визначено	Пройдено
Невідомий номерний знак	success=False, повідомлення про помилку	Пройдено

## Продовження таблиці 3.2

Тестовий сценарій	Очікуваний результат	Статус
Відомий ТЗ без накладної	success=False, прохання до диспетчера	Пройдено
Повторний в'їзд (накладна active)	Повідомлення про вже активну накладну	Пройдено
Порожній рядок номера	success=False, помилка валідації	Пройдено
Зміна статусу накладної на done	Статус оновлено в БД	Пройдено

Усі шість тестових сценаріїв пройдені успішно. Бізнес-логіка системи коректно обробляє як штатні ситуації, так і граничні випадки - відсутність транспортного засобу в базі, відсутність накладної, повторний в'їзд

Тестування веб-інтерфейсу проводилось у браузерях Google Chrome та Mozilla Firefox. Перевірялась коректність відображення сторінок на екранах різного розміру, швидкість відгуку при оновленні даних та коректність роботи форм.

Час від відправлення POST-запиту з номерним знаком до отримання відповіді від сервера у тестовому середовищі склав у середньому 180 мілісекунд. Час повної обробки з урахуванням розпізнавання зображення (без GPU) - від 3 до 6 секунд залежно від якості зображення. Обидва показники відповідають сформульованим у технічному завданні вимогам.

Сторінки операторів складів оновлюються автоматично кожні 10 секунд за допомогою JavaScript-функції, що виконує фоновий запит до API і оновлює список активних накладних без перезавантаження сторінки. Це забезпечує своєчасне сповіщення оператора про прибулий транспортний засіб.

У третьому розділі виконано практичну реалізацію та тестування комп'ютерної системи керування рухом вантажного транспорту на території підприємства.

					<b>КС КРБ 123.159.00.00 ПЗ</b>	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

Реалізовано програмне забезпечення системи у складі п'яти основних модулів: `models.py` (моделі БД через SQLAlchemy), `recognition.py` (розпізнавання номерних знаків через OpenCV та EasyOCR), `logic.py` (бізнес-логіка обробки прибуття транспортного засобу), `routes.py` (HTTP-маршрути Flask) та шаблони веб-інтерфейсу для чотирьох типів користувачів.

За результатами тестування модуля розпізнавання на наборі з 20 тестових зображень досягнуто загальну точність 85%. Усі шість сценаріїв модульного тестування бізнес-логіки пройдено успішно. Час відгуку серверної частини складає в середньому 180 мс, загальний час обробки з розпізнаванням - 3–6 секунд, що відповідає вимогам технічного завдання.

Система коректно обробляє як штатні сценарії (відомий транспортний засіб з підготовленою накладною), так і нештатні ситуації (невідомий номер, відсутність накладної, повторний в'їзд). В усіх випадках оператор КПП отримує зрозуміле повідомлення для прийняття подальшого рішення.

					<i>КС КРБ 123.159.00.00 ПЗ</i>	Арк.
						46
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

## РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

### 4.1 Аналіз умов праці операторів системи

Розроблена система керування рухом вантажного транспорту передбачає роботу персоналу на чотирьох робочих місцях: контрольно-пропускний пункт (оператор КПП) та три склади (оператори складів). Усі робочі місця обладнані персональними комп'ютерами або моноблоками з моніторами і підключені до локальної комп'ютерної мережі підприємства.

Основним знаряддям праці операторів є персональний комп'ютер з монітором. Робота передбачає постійне перебування за монітором у режимі очікування та реагування на сповіщення системи. Характер роботи - переважно спостережний, з короткими інтервалами активної взаємодії з інтерфейсом під час прийому або відправлення транспортного засобу.

З точки зору охорони праці, робота операторів відноситься до категорії робіт з використанням відеодисплейних терміналів (ВДТ). Відповідно до ДСанПіН 3.3.2.007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин», для таких робочих місць встановлюються спеціальні вимоги щодо організації праці, параметрів мікроклімату, освітлення та режиму роботи.

На робочих місцях операторів системи можуть діяти такі шкідливі та небезпечні виробничі фактори:

- електромагнітне випромінювання від монітора та системного блоку комп'ютера;
- підвищена або знижена температура повітря у приміщенні (особливо актуально для оператора КПП, будка якого може недостатньо опалюватись взимку або перегріватись влітку);

					<b>КС КРБ 123.159.00.00 ПЗ</b>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		Гусак Т.І.			<b>Безпека життєдіяльності, основи охорони праці</b>	<i>Лім.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевірив</i>		Ясній О.П.					47	
<i>Консульт.</i>		Сенчишин В.С.				<b>ТНТУ, каф. КС, гр. СІ-41</b>		
<i>Н. Контр.</i>		Луцик Н.С.						
<i>Затверд.</i>		Осухівська Г.М.						

- недостатнє або нерівномірне освітлення робочої зони, що спричиняє підвищене навантаження на органи зору;
- статичне навантаження на м'язи ший, спини та верхніх кінцівок внаслідок тривалого перебування в одній позі;
- психоемоційне навантаження, пов'язане з необхідністю постійної уваги та швидкого реагування на сповіщення;
- небезпека ураження електричним струмом при несправності електроустаткування або порушенні ізоляції.

Додатковим фактором для оператора КПП є перебування у безпосередній близькості до в'їзної зони, де рухається вантажний автотранспорт. Це створює ризики, пов'язані з вихлопними газами та шумом від двигунів транспортних засобів.

#### 4.2 Вимоги до організації робочого місця оператора

Приміщення, в яких розташовані робочі місця операторів системи, повинні відповідати вимогам ДБН В.2.2-28:2010 «Будинки і споруди. Будинки адміністративного та побутового призначення» та ДСанПіН 3.3.2.007-98.

Площа на одне робоче місце з ВДТ повинна становити не менше 6,0 м<sup>2</sup>, об'єм - не менше 20,0 м<sup>3</sup>. Приміщення повинні мати природне та штучне освітлення.

Таблиця 4.1 – Параметри мікроклімату для робочих місць операторів

Параметр	Холодний період	Теплий період	Одиниця виміру
Температура повітря	21–23	22–24	°С
Відносна вологість	40–60	40–60	%
Швидкість руху повітря	не більше 0,1	не більше 0,1	м/с

Для забезпечення нормальних параметрів мікроклімату приміщення операторів обладнуються системами опалення та вентиляції. У приміщенні КПП, де умови експлуатації є жорсткішими, рекомендується встановлення кліматичної системи (обігрівач взимку, кондиціонер влітку).

Освітленість робочої поверхні столу в зоні розташування документів повинна становити 300–500 лк. Освітлення повинно бути рівномірним, без різких тіней і відблисків на екрані монітора.

Монітор слід розташовувати так, щоб вікна знаходились збоку від нього (ліворуч для правші), а не позаду або попереду - це запобігає засліпленню та появі відблисків на екрані. При необхідності вікна обладнуються жалюзі або шторами для регулювання природного освітлення.

Для зменшення навантаження на зір рекомендується дотримуватись відстані від очей до екрана монітора в межах 60–75 см. Кут нахилу монітора відносно вертикальної площини - 10–20° назад. Верхній край екрана повинен знаходитись на рівні очей або дещо нижче.

Відповідно до ДСанПіН 3.3.2.007-98, при роботі з ВДТ встановлюються обов'язкові регламентовані перерви залежно від виду та інтенсивності роботи.

Робота операторів системи відноситься до групи Б (робота з введенням інформації та читанням з екрана) категорії I (до 2 годин роботи з ВДТ на зміну).

Для операторів системи рекомендується робити перерви тривалістю 10–15 хвилин через кожні 2 години роботи. Під час перерв доцільно виконувати комплекс вправ для очей та рук, а також короткі прогулянки для зняття статичного навантаження.

#### Електробезпека

Все електроустаткування системи - сервер, комп'ютери операторів, IP-камера, мережеве обладнання - живиться від мережі змінного струму напругою 220 В. За ступенем небезпеки ураження електричним струмом приміщення операторів відносяться до категорії приміщень без підвищеної

					<b>КС КРБ 123.159.00.00 ПЗ</b>	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

небезпеки (відсутня висока вологість, агресивні середовища, провідна підлога).

Незважаючи на відносно сприятливі умови, при роботі з електроустаткуванням необхідно дотримуватись вимог НПАОП 40.1-1.01-97 «Правила безпечної експлуатації електроустановок». Основні вимоги:

- усе електроустаткування повинно мати надійне захисне заземлення або занулення;
- електричні кабелі і дроти повинні бути прокладені у захисних каналах або кабель-каналах, без механічних пошкоджень ізоляції;
- не допускається прокладення кабелів живлення разом з мережевими кабелями передачі даних без екранування;
- розетки і вилки повинні бути у справному стані, без слідів підгоряння або механічних пошкоджень;
- не допускається самостійне розкриття корпусів обладнання і виконання ремонтних робіт персоналом, який не має відповідного допуску.

Для захисту обладнання від стрибків напруги в мережі живлення рекомендується підключати сервер та комп'ютери операторів через джерела безперебійного живлення (ДБЖ). Це також забезпечить короткочасну роботу системи під час аварійного відключення електроенергії, що особливо важливо для збереження цілісності даних у базі.

Основним технічним заходом захисту від ураження електричним струмом є захисне заземлення корпусів електроустаткування. Опір заземлювального пристрою не повинен перевищувати 4 Ом відповідно до ПУЕ (Правила улаштування електроустановок).

Додатково рекомендується встановлення пристроїв захисного відключення (ПЗВ) на лініях електроживлення серверної кімнати та приміщень операторів. ПЗВ спрацьовує при витoku струму на землю більше 30 мА, що запобігає ураженню людини навіть при непомітному порушенні ізоляції обладнання.

					<i>КС КРБ 123.159.00.00 ПЗ</i>	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

Організаційні заходи включають: інструктаж персоналу з електробезпеки при прийомі на роботу та щорічно, заборону самостійного ремонту обладнання, обов'язкове відключення обладнання від мережі перед проведенням будь-яких робіт з його переміщення або підключення нових пристроїв.

#### 4.3 Пожежна безпека

Приміщення, в яких розміщується обладнання системи, за пожежною небезпекою відносяться до категорії В (горючі тверді матеріали - пластикові корпуси обладнання, паперова документація). Відповідно до ДСТУ 3675-98 та ДБН В.1.1-7:2016, в таких приміщеннях обов'язковою є наявність первинних засобів пожежогасіння.

Для гасіння пожежі в приміщеннях з електроустаткуванням категорично забороняється використовувати воду та піну - це призведе до ураження електричним струмом і пошкодження обладнання. Для таких приміщень призначені вуглекислотні або порошкові вогнегасники.

На кожному поверсі будівлі, де розміщені робочі місця операторів, повинні бути встановлені автоматичні димові сповіщувачі. У серверній кімнаті рекомендується встановлення системи автоматичного газового пожежогасіння, яка не пошкоджує обладнання.

Основні організаційні заходи з пожежної безпеки:

- виходи з приміщень не повинні бути захарашені та повинні бути позначені відповідними знаками евакуації;
- персонал повинен бути ознайомлений з планом евакуації та правилами поведінки при пожежі;
- заборона куріння в приміщеннях і поблизу обладнання;
- своєчасна утилізація паперових відходів і пакувальних матеріалів.

					<b>КС КРБ 123.159.00.00 ПЗ</b>	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

У четвертому розділі проведено аналіз умов праці операторів комп'ютерної системи керування рухом вантажного транспорту та розроблено заходи з охорони праці й безпеки життєдіяльності.

Визначено, що основним шкідливим фактором на робочих місцях є тривала робота з відеодисплейними терміналами. Встановлено вимоги до параметрів мікроклімату (температура 21–24 °С, відносна вологість 40–60%), освітленості (300–500 лк на робочій поверхні) та режиму праці (перерви 10–15 хвилин кожні 2 години).

Розглянуто питання електробезпеки: обладнання системи підключається до мережі 220 В і потребує обов'язкового захисного заземлення, встановлення ПЗВ та дотримання правил НПАОП 40.1-1.01-97. Для забезпечення безперервності роботи та захисту даних рекомендовано використання ДБЖ.

Визначено вимоги до пожежної безпеки: приміщення відносяться до категорії В, для гасіння використовуються вуглекислотні або порошкові вогнегасники, обов'язкова наявність автоматичних димових сповіщувачів. Дотримання наведених вимог забезпечить безпечні умови праці для всього персоналу, що обслуговує систему.

					<i>КС КРБ 123.159.00.00 ПЗ</i>	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

У кваліфікаційній роботі вирішено актуальну задачу автоматизації керування рухом вантажного транспорту на території виробничого підприємства шляхом розробки відповідної комп'ютерної системи.

За результатами виконання роботи отримано такі основні результати.

1) Проведено аналіз предметної області, який показав, що ручний підхід до організації в'їзду і виїзду транспортних засобів є джерелом затримок, помилок при ідентифікації перевізників та додаткового навантаження на персонал. Розглянуто три методи ідентифікації транспортних засобів: RFID-мітки, QR/штрих-коди та автоматичне розпізнавання номерних знаків. Обґрунтовано вибір технології розпізнавання номерних знаків як універсальної - такої, що не потребує оснащення транспортних засобів додатковим обладнанням.

2) Розроблено архітектуру системи, що охоплює чотири рівні: фізичний (камера, шлагбаум, робочі місця), серверний (обробка зображень і бізнес-логіка), рівень зберігання (база даних) та рівень представлення (веб-інтерфейс). Обґрунтовано вибір технологічного стеку: Python 3, Flask, SQLite, OpenCV, EasyOCR.

3) Спроектовано базу даних системи з шести таблиць, що охоплюють усі необхідні сутності: транспортні засоби, склади, товари, накладні, позиції накладних та журнал подій. Розроблено детальний алгоритм обробки прибуття транспортного засобу у шести кроках - від захоплення кадру до відкриття шлагбауму.

4) Реалізовано програмне забезпечення системи у складі п'яти модулів. Модуль розпізнавання використовує OpenCV для попередньої обробки зображення та EasyOCR для розпізнавання символів номерного знака. Серверна частина на Flask реалізує REST API для взаємодії з клієнтськими інтерфейсами. Веб-інтерфейс включає чотири панелі: оператора КПП, оператора складу (три екземпляри), адміністратора та звітів.

					<i>КС КРБ 123.159.00.00 ПЗ</i>	Арк.
						53
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

5) Проведено тестування системи. Точність розпізнавання номерних знаків на тестовому наборі з 20 зображень склала 85%. Час серверної обробки одного запиту - в середньому 180 мс; загальний час від зупинки транспортного засобу до відкриття шлагбауму - 3–6 секунд. Усі шість сценаріїв модульного тестування бізнес-логіки пройдені успішно.

6) Розроблено рекомендації з охорони праці для операторів системи: визначено вимоги до мікроклімату, освітленості, режиму праці, електробезпеки та пожежної безпеки відповідно до чинних нормативних документів України.

Практичне значення отриманих результатів полягає в тому, що розроблена система може бути розгорнута на реальному підприємстві з мінімальними витратами на обладнання - достатньо наявного комп'ютерного парку та встановлення IP-камери на КПП. Програмне забезпечення побудоване виключно на бібліотеках з відкритим вихідним кодом, що виключає витрати на ліцензування.

Напрямами подальшого розвитку системи є: інтеграція з GPS-трекінгом для відстеження транспортних засобів за межами підприємства, розширення функцій адміністративної панелі, реалізація мобільного застосунку для операторів, а також підвищення точності розпізнавання за рахунок донавання нейромережевої моделі на зображеннях реальних українських номерних знаків.

					<i>КС КРБ 123.159.00.00 ПЗ</i>	Арк.
						54
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Жаровський Р.О., Луцик Н.С., Осухівська Г.М., Паламар А.М., Тиш Є.В. Методичні вказівки до виконання кваліфікаційної роботи бакалавра для здобувачів першого (бакалаврського) рівня вищої освіти за спеціальністю 123 «Комп'ютерна інженерія» усіх форм навчання. Тернопіль: ТНТУ, 2024. 39 с.
2. Буров Є.В., Митник М.М. Комп'ютерні мережі. Підручник. Том перший. Львів: «Магнолія 2006», 2024. 333 с.
3. Микитишин А.Г., Митник М.М., Стухляк П.Д., Пасічник В.В. Комп'ютерні мережі. Книга 1. Львів: «Магнолія 2006», 2024. 256 с.
4. Kharchenko A., Vodnarchuk I., Yatsysyn V. The Method for Comparative Evaluation of Software Architecture with Accounting of Trade-offs. American Journal of Information Systems. 2014. Vol. 2, No. 1. P. 20-25.
5. Паламар М.І., Стрембіцький М.О., Паламар А.М. Проектування комп'ютеризованих вимірювальних систем і комплексів. Навчальний посібник. Тернопіль: ТНТУ. 2019. 150 с.
6. Луцків А., Лупенко С., Пасічник В. Паралельні та розподілені обчислення. Навчальний посібник. Львів: Видавництво «Магнолія 2006», 2024. 566 с.
7. Лупенко С.А., Пасічник В.В., Тиш Є.В. Комп'ютерна логіка. Навчальний посібник. Львів: Видавництво «Магнолія 2006», 2024. 354 с.
8. Микитишин А.Г., Митник М.М., Стухляк П.Д., Пасічник В.В. Комп'ютерні мережі. Книга 2. Львів: «Магнолія 2006», 2024. 328 с.
9. Flask Documentation. Pallets Projects, 2024. URL: <https://flask.palletsprojects.com> (дата звернення: 10.03.2025).
10. OpenCV Documentation. OpenCV Team, 2024. URL: <https://docs.opencv.org> (дата звернення: 12.03.2025).
11. EasyOCR GitHub Repository. Jaided AI, 2024. URL: <https://github.com/JaidedAI/EasyOCR> (дата звернення: 14.03.2025).

					<i>КС КРБ 123.159.00.00 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

12. SQLAlchemy Documentation. SQLAlchemy authors, 2024. URL: <https://docs.sqlalchemy.org> (дата звернення: 15.03.2025).
13. SQLite Documentation. SQLite Consortium, 2024. URL: <https://www.sqlite.org/docs.html> (дата звернення: 16.03.2025).
14. pytest Documentation. pytest team, 2024. URL: <https://docs.pytest.org> (дата звернення: 18.03.2025).
15. Jinja2 Documentation. Pallets Projects, 2024. URL: <https://jinja.palletsprojects.com> (дата звернення: 20.03.2025).
16. Bradski G., Kaehler A. Learning OpenCV 4. O'Reilly Media, 2019. 1024 p.
17. Grinberg M. Flask Web Development. O'Reilly Media, 2018. 316 p.
18. Lutz M. Learning Python, 5th Edition. O'Reilly Media, 2013. 1540 p.
19. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016. 800 p.
20. Barroso J., Pinheiro P. Automatic License Plate Recognition: A Review. IEEE Access, 2022. Vol. 10. P. 12345–12360.
21. NumPy Documentation. NumPy community, 2024. URL: <https://numpy.org/doc> (дата звернення: 19.03.2025).
22. ДСанПіН 3.3.2.007-98. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин. Київ, 1998. 24 с.
23. НПАОП 40.1-1.01-97. Правила безпечної експлуатації електроустановок. Київ, 1997. 48 с.
24. ДБН В.1.1-7:2016. Пожежна безпека об'єктів будівництва. Загальні вимоги. Київ, 2017. 40 с.
25. ISO/IEC 25010:2011. Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE). Geneva: ISO, 2011. 34 p.

					<i>КС КРБ 123.159.00.00 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

Додаток А  
Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

**“Затверджую”**

Завідувач кафедри КС

\_\_\_\_\_ Осухівська Г.М.

“ 02 ” лютого 2026 р

КОМП'ЮТЕРНА СИСТЕМА КЕРУВАННЯ РУХОМ ВАНТАЖНОГО  
ТРАНСПОРТУ НА ТЕРИТОРІЇ ПІДПРИЄМСТВА

**ТЕХНІЧНЕ ЗАВДАННЯ**

на 9 листках

**Вид робіт:**

Кваліфікаційна робота

**На здобуття освітнього ступеня «Бакалавр»**

**Спеціальність 123 «Комп'ютерна інженерія»**

«УЗГОДЖЕНО»

«ВИКОНАВЕЦЬ»

Керівник кваліфікаційної роботи

Студент групи СІ-41

\_\_\_\_\_ д.т.н., проф. Ясній О.П.

\_\_\_\_\_ Гусак Т.І.

« \_\_\_\_\_ » \_\_\_\_\_ 2026 р.

« \_\_\_\_\_ » \_\_\_\_\_ 2026 р.

**Тернопіль 2026**

## 1 Загальні відомості

### 1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи: «Комп'ютерна система керування рухом вантажного транспорту на території підприємства».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.159.00.00 ПЗ

### 1.2 Виконавець

Студент групи СІ-41, факультету комп'ютерно-інформаційних систем і програмної інженерії, кафедри комп'ютерних систем та мереж, Тернопільського національного технічного університету імені Івана Пулюя, Гусак Тадей.

### 1.3 Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи є наказ по університету (№4/9-188 від 24.04.2026 р.)

### 1.4 Планові терміни початку та завершення роботи

Плановий термін початку виконання кваліфікаційної роботи – 27.01.2025 р.

Плановий термін завершення виконання кваліфікаційної роботи – 23.06.2025 р.

### 1.5 Порядок оформлення та пред'явлення результатів роботи

Порядок оформлення пояснювальної записки та графічного матеріалу здійснюється у відповідності до чинних норм та правил ISO, ЕСКД, ЕСПД та ДСТУ.

Пред'явлення проміжних результатів роботи здійснюється у відповідності до графіку, затвердженого керівником роботи. Попередній захист кваліфікаційної роботи відбувається при готовності роботи – наявності пояснювальної записки та графічного матеріалу.

Пред'явлення результатів кваліфікаційної роботи відбувається шляхом захисту на відповідному засіданні ЕК, ілюстрацією основних досягнень за допомогою графічного матеріалу.

## 2 Призначення і цілі створення системи

### 2.1 Призначення системи

Комп'ютерна система керування рухом вантажного транспорту призначена для автоматичного розпізнавання державних реєстраційних номерних знаків транспортних засобів на контрольно-пропускному пункті, ідентифікації транспортного засобу та визначення призначеного складу, автоматичного формування товарно-транспортної накладної, відображення інформації на робочих місцях операторів складів у режимі реального часу, ведення журналу всіх операцій в'їзду та виїзду транспортних засобів.

### 2.2 Мета створення системи

Метою кваліфікаційної роботи є розробка комп'ютерної системи автоматизованого керування рухом вантажного автотранспорту на території

виробничого підприємства, яка забезпечує автоматичне розпізнавання номерних знаків, ідентифікацію перевізника, визначення складу призначення та формування товарно-транспортної накладної засобами комп'ютерного зору.

## 2.3 Характеристика об'єкту

Об'єктом розробки є комп'ютерна система, призначена для використання на виробничих підприємствах, складських комплексах та логістичних центрах, що здійснюють приймання та відвантаження вантажів транспортними засобами.

## 3 Вимоги до системи

### 3.1 Вимоги до системи в цілому

#### 3.1.1 Вимоги до структури та функціонування системи

Комп'ютерна система керування рухом вантажного транспорту повинна складатись з:

- модуля розпізнавання номерних знаків (OpenCV + EasyOCR);
- серверної частини на мові Python з використанням веб-фреймворку Flask;
- бази даних SQLite;
- веб-інтерфейсу для операторів КПП, операторів складів та адміністратора;
- підсистеми керування шлагбаумом.

#### 3.1.2 Вимоги до способів та засобів зв'язку між компонентами системи

Взаємодія між компонентами системи здійснюється через локальну комп'ютерну мережу підприємства. Серверна та клієнтська частини

взаємодіють через REST API на базі протоколу HTTP. Всі компоненти повинні бути узгодженими за форматами та протоколами передачі даних.

### 3.1.3 Вимоги до режимів функціонування системи

Для системи визначено два режими функціонування:

- нормальний режим – автоматична обробка в'їзду/виїзду транспортних засобів, розпізнавання номерних знаків, формування накладних;
- аварійний режим – у разі відмови компонентів система продовжує ведення журналу подій та забезпечує ручне управління шлагбаумом оператором.

### 3.1.4 Вимоги по діагностуванню системи

Для діагностування системи використовуються вбудовані засоби операційної системи, журнали подій Flask-сервера, а також інструменти тестування бізнес-логіки на базі pytest. Інструменти повинні забезпечувати зручний інтерфейс для перегляду діагностичних подій та моніторингу роботи програмних модулів.

### 3.1.5 Перспективи розвитку та проектування системи

Система може бути розширена шляхом:

- інтеграції з GPS-трекінгом для відстеження транспортних засобів за межами підприємства;
- розробки мобільного застосунку для операторів;
- донавчання нейромережевої моделі на зображеннях українських номерних знаків;
- підключення до корпоративних ERP-систем підприємства

## 3.2 Показники призначення

Система повинна забезпечувати точність розпізнавання номерних

знаків не нижче 80% на типових умовах освітлення. Час серверної обробки одного запиту не повинен перевищувати 500 мс. Загальний час від зупинки транспортного засобу до відкриття шлагбауму – не більше 10 секунд.

### 3.2.1 Вимоги до надійності

Система повинна забезпечувати працездатність та відновлення функцій при виникненні таких ситуацій:

- збоїв в системі електропостачання апаратної частини;
- помилках в роботі апаратних засобів або мережевого обладнання;
- помилках, пов'язаних з програмним забезпеченням (ОС, драйвери пристроїв).

Для захисту серверного та клієнтського обладнання від стрибків напруги рекомендується застосовувати джерела безперебійного живлення (ДБЖ).

### 3.3 Вимоги до безпеки

Зовнішні елементи технічних засобів системи, що перебувають під напругою, повинні мати захист від випадкового дотику. Все обладнання повинне мати захисне заземлення або занулення.

Система електроживлення повинна забезпечувати захисне вимикання при перевантаженнях і коротких замиканнях. Загальні вимоги пожежної безпеки повинні відповідати нормам на електрообладнання згідно ДБН В.1.1-7:2016.

#### 3.3.1 Вимоги до експлуатації, технічного обслуговування та зберігання компонентів системи

Мікроклімат в приміщеннях операторів системи повинен відповідати вимогам ДСанПіН 3.3.2.007-98:

- температура повітря в межах від +21°C до +24°C;
- відносна вологість повітря в межах від 40% до 60%;
- швидкість руху повітря не більше 0,1 м/с.

Технічне обслуговування обладнання системи проводиться відповідно до вимог технічної документації, але не рідше ніж один раз на рік.

### 3.4 Вимоги до захисту інформації від несанкціонованого доступу

Система повинна забезпечувати захист від несанкціонованого доступу. Підсистема захисту повинна забезпечувати:

- ідентифікацію та автентифікацію користувача (оператора, адміністратора);
- розмежування прав доступу: оператор КПП, оператор складу, адміністратор;
- захист бази даних від несанкціонованого читання та модифікації.

#### 3.4.1 Вимоги по збереженню інформації при аваріях

При виникненні аварійних ситуацій дані бази даних SQLite повинні бути збережені. Рекомендується налаштування резервного копіювання бази даних на окремий носій з регулярністю не рідше одного разу на добу.

### 3.4.2 Вимоги по стандартизації і уніфікації

Система повинна відповідати вимогам ергономіки та зручності використання. Програмне забезпечення побудоване виключно на бібліотеках з відкритим вихідним кодом (Python, Flask, OpenCV, EasyOCR, SQLite), що забезпечує відсутність витрат на ліцензування.

### 3.4.3 Вимоги до функцій (завдань), що виконуються системою

- розпізнавання номерного знака транспортного засобу за допомогою алгоритмів комп'ютерного зору;
- ідентифікація перевізника та визначення складу призначення;
- автоматичне формування товарно-транспортної накладної;
- відображення інформації на робочих місцях операторів у режимі реального часу;
- забезпечення зручного веб-інтерфейсу для всіх категорій користувачів;
- забезпечення високої швидкодії – не більше 3–6 секунд на обробку одного транспортного засобу.

## 4 Вимоги до документації

Документація повинна відповідати вимогам ЄСКД та ДСТУ.

Комплект документації повинен складатись з:

- пояснювальної записки (50–70 сторінок);
- графічного матеріалу:

- а) Структурна схема комп'ютерної системи керування транспортом.
- б) Схема бази даних системи.
- в) Алгоритм обробки в'їзду транспортного засобу (блок-схема).
- г) Веб-інтерфейс системи (скріншоти роботи програмного забезпечення).

\*Примітка: У комплект документації можуть вноситися зміни та доповнення в процесі розробки.

## 5 Стадії та етапи проектування

Таблиця 1 – Стадії та етапи виконання кваліфікаційної роботи бакалавра

№ етапу	Назва етапу виконання кваліфікаційної роботи	Термін виконання
1	Розробка технічного завдання	26.01 – 02.02
2	Аналіз предметної області та вимог до системи керування вантажним транспортом (розділ 1)	03.02 – 15.02
3	Проектування архітектури та бази даних комп'ютерної системи (розділ 2)	20.04 – 25.04
4	Практична реалізація та тестування програмного забезпечення (розділ 3)	26.04 – 05.05
5	Безпека життєдіяльності, основи охорони праці (розділ 4)	07.05 – 25.05
6	Оформлення пояснювальної записки і графічного матеріалу	26.05 – 7.06
7	Перевірка на академічний плагіат, перевірка керівником та консультантами	8.06 – 14.06
8	Попередній захист кваліфікаційної роботи бакалавра	15.06 – 21.06
9	Захист кваліфікаційної роботи бакалавра	23.06

## 6 Додаткові умови виконання кваліфікаційної роботи

Під час виконання кваліфікаційної роботи у дане технічне завдання можуть вноситися зміни та доповнення.

Програмне забезпечення системи побудоване виключно на бібліотеках з відкритим вихідним кодом, що виключає витрати на ліцензування та забезпечує можливість розгортання системи на реальному підприємстві з мінімальними капітальними витратами.