

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.

_____ Ю. П. Кондратенко

«___» _____ 2022 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

СИСТЕМА ПЕРЕВІРКИ ЕЛЕКТРОННИХ
ПРОЇЗНИХ ДОКУМЕНТІВ У МІЖМІСЬКИХ
ТРАНСПОРТНИХ ЗАСОБАХ

Спеціальність 122 «Комп'ютерні науки»

122 – БКР – 401.2181023

Виконав студент 4-го курсу, групи 401

_____ *Стратонов А.О.*

«___» червня 2022 р.

Керівник: ст. викладач

_____ *Кошовий В.В.*

«___» червня 2022 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет ім. Петра Могили

Факультет комп'ютерних наук

Кафедра інтелектуальних інформаційних систем

Рівень вищої освіти **бакалавр**
Спеціальність **122 «Комп'ютерні науки»**
(шифр і назва)
Галузь знань **12 «Інформаційні технології»**
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.

_____ Ю. П. Кондратенко

«___» _____ 20__ р.

З А В Д А Н Н Я

на виконання кваліфікаційної роботи

Видано студенту групи 401 факультету комп'ютерних наук Артему
Олександровичу Стратонову.

1. Тема кваліфікаційної роботи «Система перевірки електронних проїзних документів у міжміських транспортних засобах».

Керівник роботи Кошовий В. В., ст. викладач.

Затв. наказом Ректора ЧНУ ім. Петра Могили від «___» _____ 20__ р. № _____

2. Строк представлення кваліфікаційної роботи студентом «___» _____ 20__ р.

3. Вхідні (початкові) дані до роботи: методи застосування принципів роботи з базами даних; бібліотеки та технології для створення та зчитування QR-коду.

Очікуваний результат роботи: функціонуюча система перевірки електронних квитків на основі технології QR-код.

4. Перелік питань, що підлягають розробці (зміст пояснювальної записки):

- огляд на сучасні технології та методи, які використовуються для розробки систем перевірки електронних квитків;
- пошук аналогів розроблюваної системи;

- огляд методів і технологій для розробки системи перевірки квитків;
- створення плану та розробка системи перевірки електронних квитків.

5. Перелік графічного матеріалу: презентація.

6. Завдання до спеціальної частини:

- проаналізувати наявні умови праці в робочому приміщенні;
- визначити достатній рівень показників для робочого приміщення, де проводяться роботи з розробки системи;
- визначити основні правила техніки безпеки при роботі з комп'ютерною технікою.

7. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис
Спеціальна частина	О. В. Макарова	

Керівник роботи ст. викладач Кошовий В.В.

(наук. ступінь, вчене звання, прізвище та ініціали)

(підпис)

Завдання прийнято до виконання Стратонова А. О.

(прізвище та ініціали)

(підпис)

Дата видачі завдання « 18 » листопада 2021 р.

АНОТАЦІЯ

Об'єктом розробки є процеси перевірки електронних проїзних документів у міжміських транспортних засобах.

Предметом розробки є використання системи QR-code для створення системи перевірки проїзних документів, створення бази даних MySQL для зберігання квитків.

Мета комплексної дипломної роботи полягає у розробці системи перевірки електронних квитків для міжміських транспортних засобів.

У першому розділі представлені аналіз причин появи електронних квитків, з метою пояснення актуальності теми БКР. Мета дослідження другого підрозділу виявити аналоги та основні принципи роботи систем з перевірки електронних квитків. У другому розділі було представлено та проаналізовано методи розробки систем з перевірки електронних квитків, які технології для цього використовуються. Результатом аналізу було обрано основні методи та технології для реалізації плану роботи системи, а саме: мову програмування PHP, технологію QR-код, MySQL для роботи з базою даних. У третьому розділі було створено план розробки системи перевірки електронних квитків для міжміських транспортних засобів. В результаті виконання створеного плану було розроблено систему, яка складається з бази даних квитків, візуальної оболонки системи, реалізовано створення QR-коду з квитка та зчитування створеного QR-коду.

Ключові слова: Електронні квитки, бази даних, QR-code, MySQL, PHP, транспортні засоби.

Бакалаврська кваліфікаційна робота викладена на 81 сторінці, вона містить 4 розділи, 45 рисунків, 3 таблиці та 25 джерел у переліку посилань.

ANNOTATION

The object of development is the process of checking electronic travel documents in long-distance vehicles.

The subject of development is the use of QR-code system to create a system for checking travel documents, creating a database mySQL for storing tickets.

The purpose of the complex thesis is to develop a system for checking electronic tickets for long-distance vehicles.

The first section presents an analysis of the reasons for the emergence of electronic tickets, in order to explain the relevance of the topic of BCR. The purpose of the study of the second section is to identify analogues and basic principles of electronic ticketing systems. The second section presented and analyzed the methods of developing electronic ticket verification systems, what technologies are used for this purpose. The analysis selected the main methods and technologies for the implementation of the system work plan: PHP programming language, QR-code technology, mySQL for working with the database. In the third section, a plan was developed to create an electronic ticket verification system for long-distance vehicles. As a result of the created plan the system consisting of a database of tickets, a visual cover of system was developed, creation of QR-code from the ticket and reading of the created QR-code was realized.

Keywords: Electronic tickets, databases, QR-code, MySQL, PHP, vehicles.

The bachelor's thesis is set out on 81 pages, it contains 4 sections, 45 figures, 3 tables and 25 sources in the list of references.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА ТЕРМІНІВ	4
ВСТУП	6
1 АНАЛІЗ ОБРНАОЇ ПРЕДМЕТНОЇ ОБЛАСТІ	8
1.1 Електронні квитки. Причини появи та переваги над паперовими квитками.....	8
1.2 Огляд та аналіз наявних аналогів та публікацій.....	10
1.3 Постановка задачі	15
Висновки до розділу 1	15
2 МОДЕЛІ, МЕТОДИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ	16
2.1 Варіанти мов програмування для розробки системи	16
2.1.1 Python	16
2.1.2 Java.....	17
2.1.3 PHP	19
2.2 MySQL.....	20
2.3 Хмарні бази даних.....	23
2.4 Технологія QR-код та зчитування QR-коду з растрового зображення ..	26
2.4.1 Структура QR-коду.....	26
2.4.2 Розпізнавання QR-коду	36
2.5 Принципи роботи NFC	39
2.6 PhpMyAdmin та Openserver.....	44
Висновки до розділу 2	47
3 ПРОЦЕС РОЗРОБКИ СИСТЕМИ.....	48
3.1 План функціонування системи перевірки.....	48
3.2 Розробка системи	53
Висновки до розділу 3	63
4 ОХОРОНА ПРАЦІ.....	65
4.1 Комп'ютер та його апаратне забезпечення.....	66
4.2 Виробниче середовище	72
4.3 Причини виробничого травматизму та захворювань.....	76

Висновки до розділу 4	78
ВИСНОВКИ.....	80
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	82
ДОДАТОК А.....	85

ПЕРЕЛІК СКОРОЧЕНЬ ТА ТЕРМІНІВ

ДССУ – Державна служба статистики України

DBaaS (База даних як послуга) – керована служба хмарних обчислень, забезпечує доступ до бази даних без необхідності встановлення фізичного обладнання.

NFC (Near-field communication) – набір протоколів зв'язку, які забезпечують зв'язок між двома електронними пристроями з відривом 1 1/2 дюйма чи менше.

QR-код (Quick Response) – тип матричного штрих-коду (або двовимірного штрих-коду), винайдений у 1994 році японською автомобільною компанією Denso Wave.

Ідентифікатор – ім'я, яке позначає ідентичність унікального об'єкта, або унікального класу об'єктів, де «об'єкт» або клас може бути ідеєю, фізичним обчислюваним об'єктом (або його класом) або фізичною речовиною, що не обчислюється (або його клас).

Скрипт – послідовність дій, заданих за допомогою скриптового мови програмування для автономного виконання завдання(сервісу, інструменту, функції, тощо.).

ПЗ – програмне забезпечення.

HTML (HyperText Markup Language) – мова розмітки документів веб-сторінок, задана міжнародними стандартами, використовується для перегляду веб-сторінок у мережі Інтернет.

Cookie-файл – у комп'ютерній термінології поняття, яке поняття, яке вказує на невеличкі текстові, або бінарні файли, котрі передаються з веб-серверу на комп'ютер користувача та зберігаються деякий час.

Патерн – об'єкт, який береться за основу створення копії, або набору повторюваних об'єктів.

Хеш (Hash) – функція згортки, що здійснює перетворення масиву вхідних даних довільної довжини у вихідний бітовий рядок встановленої довжини, що виконується певним алгоритмом.

**Пояснювальна записка
до кваліфікаційної роботи
на тему:**

**«Система перевірки електронних проїзних документів у міжміських
транспортних засобах»**

Спеціальність 122 «Комп'ютерні науки»

122 – БКР – 401.21810123

Виконав студент 4-го курсу, групи 401

Стратонов А.О.

(підпис, ініціали та прізвище)

«__» _____ 202_ р.

Керівник: Кошовий В.В., ст. викладач

(наук. ступінь, вчене звання)

(підпис, ініціали та прізвище)

«__» _____ 202_ р.

Миколаїв – 2022

ВСТУП

Міжміські перевезення – це перевезення пасажирів на маршрути, довжина яких більше 50 кілометрів. Вони можуть проходити в межах області(внутрішньообласні), а можуть виходити за її межі (міжобласні). Міжміські перевезення можуть проходити за допомогою різних транспортних засобів. Основними з них в Україні є:

Залізничний транспорт — рейковий транспорт, що складається з пасажирських вагонів, що рухаються залізничною колією. В багатьох країнах займає значний обсяг усіх міжміських перевезень.

Автобусний транспорт — безрейковий транспортний засіб, який пересувається дорогою, призначений для перевезення великих груп людей.

Початком суспільного розвитку перевезень пасажирів можна вважати середину 19 ст., коли європейські країни масово почали будувати залізничні колії та станції для швидшого перевезення мешканців між містами.

Актуальність обраної теми полягає у тому, що у часи тотальної діджиталізації та за умов хвороби COVID-19 потреба у перевірці проїзних документів без фізичного контакту зростає все більше. В Україні залізничним транспортом за даними ДССУ 2018 було перевезено 181 млн. пасажирів, а автомобільним транспортом (автобусами) було перевезено 122 млн. Це свідчить про те, що громадський транспорт користується величезним попитом, що тягне за собою великий ризик отримання хвороби.

Електронний квиток це документ, який дає можливість пасажирові на проїзд аналогічно традиційному квитку, який продається в касах. В Україні такими квитками можуть користуватися пасажирові Укрзалізниці, квиток відображає цифровий запис в базі даних, тому прирівнюється до повноцінного проїзного документу. Також в електронних документах є загальна перевага – його неможливо втратити, адже завжди є можливість роздрукувати його знову.

Об'єктом розробки є процеси перевірки електронних проїзних документів у міжміських транспортних засобах.

Предметом розробки є використання системи QR-code для створення системи перевірки проїзних документів, створення бази даних MySQL для зберігання квитків.

Мета комплексної дипломної роботи полягає у розробці системи перевірки електронних квитків для міжміських транспортних засобів.

Для досягнення вказаної мети роботи треба виконати основні завдання:

1. Провести аналіз способів перевірки електронних проїзних документів;
2. Обрати реляційну систему бази даних для зберігання даних у системі;
3. Обрати спосіб перевірки електронних проїзних документів;
4. Розробка системи перевірки електронних проїзних документів.

1 АНАЛІЗ ОБРНАОЇ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Електронні квитки. Причини появи та переваги над паперовими квитками

Проїзний квиток - документ, що підтверджує право особи на проїзд у будь-якому транспорті. Існують проїзні квитки як для одноразового використання, так і для використання певну кількість разів або на певний термін (абонементні квитки, в розмовній мові – «проїзні квитки»). При використанні одноразового квитка останній спеціальним чином позначається або вилучається, щоб його не можна було використати повторно.

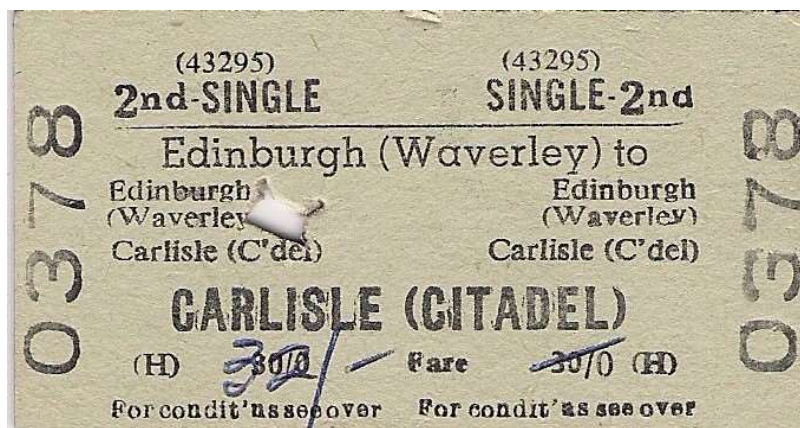


Рисунок 1.1 – Паперовий квиток на потяг

Ранні квитки були схожі на форму валюти, що випускається окремими залізницями, продається агентами та збирається кондукторами, які перевірялися залізницею, щоб переконатися, що запаси квитків відповідають заявленим доходам пасажирів. Оскільки безперервний рух кількома сполученими залізницями став звичайним явищем, купонні квитки із зазубреними частинами для кожної залізничної компанії можуть бути видані в пункті відправлення і послідовно збираються кондукторами залізниць, що забезпечують проїзд, щоб уникнути необхідності купувати додаткові квитки в кожному пункті пересадки.

Наразі в усіх країнах світу проходить процес відмови від паперових квитків та впровадження електронних квитків, спричинене Digital Revolution (Цифровою революцією) та активним процесом діджиталізації.

Digital Revolution (також відома як Третя промислова революція) — це перехід від механічних та аналогових електронних технологій до цифрової електроніки, який розпочався у другій половині 20-го століття з впровадженням та розповсюдженням цифрових комп'ютерів та цифрового ведення записів, що триває й досі. Неявно цей термін також стосується радикальних змін, викликаних цифровими обчислювальними та комунікаційними технологіями у цей період.

Діджиталізація - це процес перетворення інформації в цифровий формат. Результатом є представлення об'єкта, зображення, звуку, документа або сигналу (зазвичай аналогового сигналу), отримане шляхом створення ряду чисел, що описують дискретний набір точок або вибірок. Результат називається цифровим поданням або, точніше, цифровим зображенням для об'єкта та цифровою формою для сигналу. У сучасній практиці оцифровані дані мають форму двійкових чисел, що полегшує обробку цифровими комп'ютерами та інші операції, але оцифрування просто означає перетворення аналогового вихідного матеріалу на числовий формат; натомість можна використовувати десяткову чи будь-яку іншу систему числення.

Діджиталізація має вирішальне значення для обробки, зберігання та передачі даних, тому що вона «дозволяє передавати інформацію всіх видів у всіх форматах з однаковою ефективністю, а також змішувати». Хоча аналогові дані, як правило, більш стабільні, цифрові дані мають потенціал для легшого обміну та доступу, і, теоретично, можуть поширюватися безкінечно без втрати генерації, за умови, що при необхідності вони переносяться в нові, стабільні формати. Цей потенціал привів до інституційних проєктів оцифрування, призначених для покращення доступу та швидкого зростання області цифрового зберігання.

Електронний квиток – діджиталізований нащадок паперового квитку, котрий створений задля спрощення та пришвидшення процесу продажу саме

квитка на потрібний маршрут. Також для зменшення кількості бюрократії у момент посадки пасажирів на маршрут.

Вперше було запроваджено електронні квитки великими авіакомпаніями у 1994 році, серед котрих компанія Southwest Airlines лідирувала. Електронний квиток надає можливість економії часу на бронювання та оплату маршруту поїздки при покупці квитка в Інтернеті через сайти транспортних компаній і агентств. Квиток зберігається в електронному вигляді в базі даних компанії, тому його втрата стає неможливою. E-ticket надає можливість для замовлення поїздки для інших осіб (родичів, друзів, колег), які з якихось причин не можуть зробити це самостійно, і їм доведеться прийти лише для реєстрації на сам маршрут.

1.2 Огляд та аналіз наявних аналогів та публікацій

Укрзалізниця

На відміну від паперового квитка, електронний квиток є цифровим записом в базі даних Укрзалізниці. Купивши електронний квиток, пасажирів не потрібно обмінювати його в касі на паперовий, достатньо пред'явити бланк електронного квитка провіднику разом з посвідченням особи під час посадки в поїзд.

Такий квиток є повноцінним проїзним документом, а також фіскальним документом і приймається бухгалтерією для компенсації проїзду у відрядженні. Електронні квитки не потрібно обмінювати в касі і діють на всі поїзди внутрішнього сполучення, крім приміських, регіональних та додаткових поїздів. Також електронний квиток, оформлений через інтернет, неможливо втратити – його можна роздрукувати необмежену кількість разів із аркуша, який надходить на електронний ящик клієнта. Електронний квиток може бути представлений як у вигляді роздрукованого бланка, що містить QR-код, так і просто у вигляді QR-коду на екрані мобільного телефону, планшета або ноутбука.



Посадочний документ

ТЕРМ. №2301	ПОСАДОЧНИЙ ДОКУМЕНТ		000ВЗВЕА-FC5C-DFCE-0001		ПН:400758126555 ФН:3000182017 ЗН:3700000002 ФК:14746584	
	ПАТ «УКРЗАЛІЗНИЦЯ» м. Київ, вул. Тверська, буд.5		#7БП-Н1-1876292-2709			
МПС	ЦЕЙ ПОСАДОЧНИЙ ДОКУМЕНТ Є ПІДСТАВОЮ ДЛЯ ПРОЇЗДУ					27.09.2016 16:01
Прізвище, Ім'я	Петренко Микола			Поїзд	013 ОБ ФІРМ НШ	
Відправлення	2200001	КИЇВ-ПАСАЖИРСЬКИЙ		Вагон	16 П П-3	
Призначення	2218095	УЖГОРОД		Місце	015 Повний	
Дата/час відпр.	29.09.2016 18:35			Сервіс	БІЛ	
Дата/час приб.	014*	30.09.2016 10:35				
<p>ВАРТ=160,48ГРН(КВ.51,41+ПЛ.47,60+СП.20,83+ПДВ.26,72+СТР.0,17+КЗБ.13,75) В Т.Ч. БІЛ-25,00 (3 ПДВ) #СТР.ВІД Н/В 6000НЕОП.МІН. СК"АРСЕНАЛ СТРАХУВАННЯ", КИЇВ БОРШАГІВСЬКА154, Т0800604453 ЧАС ВІДПРАВЛЕННЯ КИЇВСЬКИЙ</p>						
			<p>Цей Посадочний документ є підставою для проїзду без звернення у касу. Посадочний документ являється розрахунковим документом. Повернення данного Посадочного документа можливе до відправлення поїзда.</p>			

Рисунок 1.2 – Приклад електронного квитка «Укрзалізниці»

QR-код — це тип матричного штрих-коду (або двовимірного штрих-коду), винайдений у 1994 році японською автомобільною компанією Denso Wave. Штрих-код — це оптична етикетка, що є машиночитаємою, яка може містити інформацію про об'єкт, до якого вона прикріплена. На практиці QR-код часто містить дані для локатора, ідентифікатора або трекера, які вказують на веб-сайт або програму. QR-код використовує чотири стандартизовані режими кодування (числовий, буквено-цифровий, байтовий/двійковий та кандзі) для ефективного зберігання даних; також можна використовувати додатки.

Система Quick Response стала популярною за межами автомобільної промисловості через її швидку зчитуваність і більшу ємність у порівнянні зі стандартними штрих-кодами UPC. Додатки включають відстеження товару, ідентифікацію товару, відстеження часу, управління документами та загальний маркетинг.

QR-код складається з чорних квадратів, розташованих у вигляді квадратної сітки на білому тлі, які можуть бути прочитані пристроєм обробки

зображень, таким як камера, і оброблені за допомогою виправлення помилок Ріда-Соломона, доки зображення не буде правильно інтерпретовано. Потім необхідні дані вилучаються із шаблонів, які присутні як у горизонтальному, так і вертикальному компонентах зображення.

QR-коди тепер використовуються у значно ширшому контексті, включаючи як комерційні програми для відстеження, так і програми, орієнтовані на зручність, призначені для користувачів мобільних телефонів (звані мобільними тегами). QR-коди можуть використовуватися для відображення тексту користувачеві, для відкриття веб-сторінки на пристрої користувача, для додавання контакту vCard на пристрій користувача, для відкриття універсального ідентифікатора ресурсу (URI), для підключення до бездротової мережі або написати електронний лист або текстове повідомлення. Існує безліч генераторів QR-кодів, доступних у вигляді програмного забезпечення або онлайн-інструментів, які або безкоштовні або вимагають платної підписки. QR-код став одним із найбільш часто використовуваних типів двовимірного коду.

Технологія NFC та її використання для перевірки проїзних документів.

Near-field communication (NFC) — це набір протоколів зв'язку, які забезпечують зв'язок між двома електронними пристроями з відривом 4 див (11/2 дюйма) чи менше. NFC пропонує низькошвидкісне з'єднання за допомогою простого налаштування, яке можна використовувати для запуску більш ефективних бездротових з'єднань. Як і інші технології «безконтактних карт», NFC заснована на індуктивному зв'язку між двома так званими антенами, присутніми на пристроях з підтримкою NFC, наприклад смартфоні та принтері, які обмінюються даними в одному або обох напрямках, використовуючи частоту 13,56 МГц. доступний у всьому світі діапазон радіочастот ISM, що не ліцензується, з використанням стандарту радіоінтерфейсу ISO/IEC 18000-3 зі швидкістю передачі даних в діапазоні від 106 до 424 кбіт/с.

Стандарти були надані NFC Forum. Форум відповідав за просування технології, встановлення стандартів та сертифікацію пристроїв. Безпечний зв'язок доступний завдяки застосуванню алгоритмів шифрування, як це робиться для кредитних карт, і якщо вони відповідають критеріям, що дозволяють вважати їх особистою мережею.

Стандарти NFC охоплюють протоколи зв'язку та формати обміну даними та засновані на існуючих стандартах радіочастотної ідентифікації (RFID), включаючи ISO/IEC 14443 та FeliCa. Стандарти включають ISO/IEC 18092 та стандарти, визначені NFC Forum. На додачу до форуму NFC група GSMA визначила платформу для розгортання стандартів GSMA NFC у мобільних телефонах. Зусилля GSMA включають Trusted Services Manager, Single Wire Protocol, тестування/сертифікацію та безпеку елементу. Портативні пристрої з підтримкою NFC можуть бути забезпечені прикладним програмним забезпеченням, наприклад, для зчитування електронних міток або здійснення платежів при підключенні до системи, сумісної з NFC. Вони стандартизовані для протоколів NFC, замінюючи пропріетарні технології, що використовувалися в ранніх системах.

Програма ліцензування патентів для NFC знаходиться на стадії реалізації патентним фондом France Brevets, створеним у 2011 році. Ця програма розроблялася Via Licensing Corporation, незалежною дочірньою компанією Dolby Laboratories, і була припинена у травні 2012 року. Незалежна від платформи безкоштовна бібліотека NFC з відкритим вихідним кодом, libnfc, доступна за ліцензією GNU Lesser General Public License.

Існуючі та очікувані програми включають безконтактні транзакції, обмін даними та спрощене налаштування складніших комунікацій, таких як Wi-Fi. Крім того, коли один із підключених пристроїв має вихід в Інтернет, інший може обмінюватися даними з онлайн-сервісами.

Факт впровадження технології NFC в громадському транспорті спричинить за собою поява нових елементів і зацікавлених сторін, які

необхідно пояснити і зрозумів. Ці нові елементи та зацікавлені сторони не з'являються даремно причини, але мають конкретну функцію та приносять додаткову цінність у ланцюжок процесів в результаті використання цієї технології. У цьому розділі ми збираємося порівняти поточний випуск та використання безконтактної картки сценарій зі сценарієм, згенерованим із реалізацією NFC.

Нинішній сценарій випуску безконтактних карток і керування ним добре встановлений та стандартизований. Роль різних зацікавлених сторін і елементів чітко визначена, а їхні завдання, ролі та бізнес-модель чітко поставлені. У поточному сценарії користувач громадського транспорту звертається до відповідної компанії за безконтактною картою, яка персоналізована з даними квитка, запитаними у користувача і доставлена останньому. Після того, як користувач має картку, він використовує її для проїзду громадським транспортом. Виділяють два основних моменти ено щодо безконтактної картки:

- 1) Персоналізація картки здійснюється фізично на ній і до її доставки користувачеві, який його запитав.
- 2) Компанія, яка видає безконтактну картку, має канал постачання безконтактної картки. Зацікавлені сторони, які беруть участь у випуску безконтактних карток, є частиною постачання каналу, і є наступними:
 - виробник мікросхеми: виробляє мікросхему (чіп), яка знаходиться в безконтактному картку та зберігає відповідну транспортну програму;
 - виробник безконтактної картки: отримує мікросхему з мікросхеми виробник додає антену для забезпечення зовнішнього зв'язку та зв'язує його в пластик або папір, щоб зробити листівку. Ця зацікавлена сторона зазвичай попередньо персоналізує картку, тобто генерує структуру програми в чіпі без даних користувача;

– транспортна компанія/компанія-емітент картки: отримує попередньо персоналізовані картки від виробника картки, і як тільки користувач подає заявку на послугу, персоналізує картку з необхідними даними;

– кінцевий користувач: отримує персональну безконтактну картку разом із потрібним транспортом квиток, готовий до використання. На малюнку нижче показано короткий опис того, що описано вище щодо проблеми безконтактних карток.

1.3 Постановка задачі

Для досягнення поставленої мети, на основі аналізу способів перевірки електронних проїзних документів, потрібно розробити систему, яка буде складатися з бази даних квитків, візуальної оболонки додатку, скрипта, за допомогою якого буде створюватися QR-код, скрипта для зчитування QR-коду, та запиту до бази даних на основі ідентифікатора, зашифрованого в QR-коді. База даних буде містити у собі таблицю з даними квитка, у яку будуть входити: ідентифікатор квитка, ім'я пасажирів, фамілія, дата народження, час і дата відправлення та прибуття рейсу, місце відправлення та прибуття, вартість квитка. Візуальна оболонка буде представлена на базі сторінок HTML, скрипти буде написано на скриптовій мові програмування PHP.

Висновки до розділу 1

У першому розділі було проведено аналіз історії розвитку транспортних перевезень з метою розглянути чому і коли з'явилась потреба у впровадженні електронних квитків. Було проаналізовано переваги електронних квитків над звичайними з метою визначення мети для створення системи для перевірки проїзних документів. Розглянуто найбільш поширені аналоги в межах України та за кордоном для створення плану розробки системи. Поставлена задача роботи, яка передбачає розробку застосунку з таблицею квитків в базі даних MySQL, до котрої буде надано доступ через сканування заздалегідь створеного QR-коду.

2 МОДЕЛІ, МЕТОДИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

2.1 Варіанти мов програмування для розробки системи

Перш за все треба розглянути мови програмування, які мають найбільшу популярність у наш час. За даними rurl.github.io найпопулярнішими є такі мови програмування (див. табл. 3.1).

Таблиця 2.1 – Популярність мов програмування станом на 22.06.2022

Популярність	Мова програмування	Охват	Тренд
1	Python	27.61 %	-2.8 %
2	Java	17.64 %	-0.7 %
3	JavaScript	9.21 %	+0.4 %
4	C#	7.79 %	+0.8 %
5	C/C++	7.01 %	+0.4 %
6	PHP	5.27 %	-1.0 %
7	R	4.26 %	+0.5 %

Надалі буде розглянуто декілька з цих мов програмування.

2.1.1 Python

Python — це високорівнева, інтерпретована, інтерактивна та об'єктно-орієнтована скриптована мова програмування. Мова програмування розроблена для легкого розуміння коду. Він часто використовує англійські ключові слова, тоді як інші мови використовують розділові знаки, і в ньому менше синтаксичних конструкцій, ніж в інших мовах. Python було розроблено

Гвідо ван Россумом наприкінці 80-х і на початку 90-х років у Національному дослідному інституті математики та комп'ютерних наук у Нідерландах. Вперше випущений у 1991 році як Python 0.9.0.

Опис можливостей Python

Python надає змогу безпосередньо взаємодіяти з інтерпретатором, щоб розробляти власні додатки та підтримує об'єктно-орієнтований метод програмування, який інкапсулює код у об'єкти. Є відмінною мовою для розробників початкового рівня, адже підтримує розробку широкого спектру програм, від простої обробки тексту до веб-браузерів та ігор. Має велику стандартну бібліотеку, більша частина якої є портативною та крос-платформеною, сумісна з UNIX, Windows і Macintosh. Python обробляється інтерпретатором під час виконання, прибираючи потребу компілювати програму перед її виконанням. Підтримує інтерактивний режим, який дозволяє інтерактивно тестувати та налагоджувати фрагменти коду. Може працювати на різних апаратних платформах і має однаковий інтерфейс на всіх платформах. Має функцію додавання низькорівневих модулів інтерпретаторів Python, які дозволяють додавати або налаштовувати свої інструменти для підвищення ефективності. Має змогу працювати з усіма основними базами даних. Python підтримує програми з графічним інтерфейсом, які можна створювати та портувати в багато системних викликів, бібліотек та систем Windows, таких як Windows MFC, Macintosh і систему X Window в Unix. Є можливість використовувати як скриптову мову або компілювати в байт-код для створення великих програм. Він надає динамічні типи даних дуже високого рівня та підтримує динамічну перевірку типів. Python можна інтегрувати з C, C++, COM, ActiveX, CORBA та Java.

2.1.2 Java

Мова програмування Java була спочатку розроблена Sun Microsystems з ініціативи Джеймса Гослінга і випущена в 1995 як основний компонент

платформи Java Sun Microsystems (Java 1.0 [J2SE]).Останнім випуском Java Standard Edition є Java SE 8.

Опис можливостей Java

Об'єктно-орієнтована - у Java все є об'єктом. Java має можливості розширення, оскільки вона базується на об'єктній моделі. Незалежна від платформи на відміну багатьох інших мов програмування, включаючи C і C++, коли Java компілюється, вона компілюється в а незалежний від платформи байт-код. Цей байт-код розповсюджується через Інтернет та інтерпретується віртуальною машиною (JVM) на будь-якій платформі, на якій він використовується. Мова програмування Java розроблена так, щоб її було легко освоїти маючи знання у базових принципах ООП. Є безпечною – завдяки функції безпеки за замовченням дозволяє розробляти системи без вірусів та несанкціонованого доступу. Методи аутентифікації базуються на шифруванні з відкритим ключем. Незалежна від архітектури - компілятор Java генерує незалежний від архітектури формат об'єктного файлу, що робить скомпільований код, що виконується на багатьох процесорах з наявністю Java runtime system. Нейтральність до архітектури і відсутність аспектів специфікації, що залежать від реалізації, робить Java портативною. Компілятор Java написаний на ANSI C з чіткою межею переносимості, яка є підмножиною POSIX. Включає в себе перевірку помилок під час компіляції та перевірки під час виконання. Багатопоточність — за допомогою багатопоточної функції Java можна писати програми, які можуть виконувати багато завдань одночасно. Ця конструктивна особливість дозволяє розробникам створювати інтерактивні програми, які можуть працювати безперебійно. Інтерпретований – байт-код Java на льоту транслюється у власні машинні інструкції та ніде не зберігається. Процес розробки більш швидкий та аналітичний, оскільки зв'язування є поетапним та легким процесом завдяки використанню компіляторів Just-In-Time, що забезпечує високу продуктивність. Java вважається більш динамічним, ніж C або C ++, оскільки

він призначений для адаптації до середовища, що розвивається. Програми, написані на мові Java можуть містити об'єм інформації під час виконання, яку можна використовувати для перевірки та дозволу доступу до об'єктів під час виконання. Він використовується для розробки програм для Android.

Сфера використання Java:

- розробка корпоративного ПЗ;
- розробка мобільних Java-додатків;
- наукові обчислювальні програми;
- аналітика великих даних;
- java-програмування апаратних пристроїв;
- використовується для серверних технологій, таких як Apache, JBoss, GlassFish і т.д.

2.1.3 PHP

Препроцесор гіпертексту (PHP) - це мова програмування, яка дозволяє веб-розробникам створювати динамічний контент, що взаємодіє з базами даних. PHP переважно використовується для розробки веб-додатків. PHP починався як невеликий проект з відкритим вихідним кодом, який розвивався в міру того, як усе більше людей дізнавалися, наскільки він корисний. Расмус Лердорф випустив першу версію PHP ще 1994 року. PHP використовується в 80% веб-сайтів, щоб стати відмінним інженером-програмістом, особливо коли вони працюють у галузі веб-розробки, знання цієї скриптової мови програмування є обов'язковими.

Переваги PHP:

- мова сценаріїв на стороні сервера, вбудована в HTML. Використовується для управління динамічним контентом, базами даних, відстеження сеансів і навіть створення цілих сайтів електронної комерції;
- інтегрований в більшість популярних баз даних, включаючи MySQL, PostgreSQL, Oracle, Sybase, Informix та Microsoft SQL Server;

– PHP швидкий у своєму виконанні, особливо коли він скомпільований як модуль Apache на стороні Unix. Сервер MySQL після запуску виконує дуже складні запити з величезним набором результатів в рекордно короткі терміни;

– PHP підтримує велику кількість основних протоколів, таких як POP3, IMAP та LDAP. У PHP4 додана підтримка Java та розподілених об'єктних архітектур (COM та CORBA), що вперше уможливило багаторівневу розробку.

PHP виконує системні функції, тобто із файлів у системі він може створювати, відкривати, читати, записувати та закривати їх. PHP може обробляти форми, тобто збирати дані з файлів, зберігати дані у файл, електронною поштою можна надсилати дані, повертати дані користувачеві. Надає можливість додавати, видаляти, змінювати елементи у базі даних через PHP, отримання доступу до змінних файлів cookie та встановити файли cookie. Має функцію регулювання доступу користувачів до деяких сторінок сайту та шифрування даних.

Для розробки системи перевірки квитків було обрано скриптову мову програмування PHP через великі можливості до роботи з базами даних, великий потенціал щодо візуального оформлення, та дуже малі вимоги для апаратного забезпечення.

2.2 MySQL

База даних — це структурований набір даних. Це може бути що завгодно: від простого списку покупок до картинної галереї або величезної кількості інформації в корпоративній мережі. Для додавання, доступу та обробки даних, що зберігаються в комп'ютерній базі даних, вам потрібна система керування базами даних, наприклад MySQL Server. Оскільки комп'ютери дуже добре обробляють великі обсяги даних, системи керування базами даних відіграють центральну роль у обчислювальній роботі, як самостійні утиліти або як частина інших програм. Бази даних MySQL є реляційними.



Рисунок 2.1 – Логотип MySQL

Реляційна база даних зберігає дані в окремих таблицях, а не розміщує всі дані в одному великому сховищі. Структури бази даних організовані у фізичні файли, оптимізовані для швидкості. Логічна модель з такими об'єктами, як бази даних, таблиці, уявлення, рядки та стовпці, пропонує гнучке середовище програмування. Ви встановлюєте правила, що регулюють відносини між різними полями даних, наприклад, один до одного, один до багатьох, унікальний, обов'язковий або необов'язковий, а також «вказівники» між різними таблицями. База даних дотримується цих правил, так що за допомогою добре розробленої бази даних ваша програма ніколи не бачить неузгоджених, повторюваних, сиріт, застарілих або відсутніх даних.

Частина SQL «MySQL» означає «Мова структурованих запитів». SQL є найпоширенішою стандартизованою мовою, яка використовується для доступу до баз даних. Залежно від середовища програмування ви можете вводити SQL безпосередньо (наприклад, для створення звітів), вбудовувати оператори SQL у код, написаний іншою мовою, або використовувати спеціальну мову API, яка приховує синтаксис SQL.

SQL визначається стандартом SQL ANSI/ISO. Стандарт SQL розвивається з 1986 року і існує кілька версій. У цьому посібнику «SQL-92» відноситься до стандарту, випущеному в 1992 році, «SQL:1999» відноситься до стандарту, випущеному в 1999 році, а «SQL:2003» відноситься до поточної

версії стандарту. Ми використовуємо фразу «стандарт SQL» для позначення поточної версії стандарту SQL у будь-який час.

Програмне забезпечення MySQL є відкритим вихідним кодом, відкритий код означає, що будь-хто може використовувати та змінювати програмне забезпечення. Будь-хто може завантажити програмне забезпечення MySQL з Інтернету та використовувати його, нічого не сплачуючи. Якщо бажаєте, ви можете вивчити вихідний код і змінити його відповідно до ваших потреб. Програмне забезпечення MySQL використовує GPL (GNU General Public License), <http://www.fsf.org/licenses/>, щоб визначити, що ви можете, а що не можете робити з програмним забезпеченням у різних ситуаціях. Якщо ви відчуваєте незручність із GPL або вам потрібно вставити код MySQL у комерційну програму, ви можете придбати у нас комерційну ліцензійну версію. Дивіться Огляд ліцензування MySQL для отримання додаткової інформації (<http://www.mysql.com/company/legal/licensing/>).

Сервер баз даних MySQL дуже швидкий, надійний, масштабований і простий у використанні.

Якщо це те, що ви шукаєте, ви повинні спробувати. MySQL Server може зручно працювати на настільному комп'ютері або ноутбуку, поряд з іншими додатками, веб-серверами тощо, не вимагаючи практичної уваги або зовсім не вимагаючи. Якщо ви присвячуєте MySQL всю машину, ви можете налаштувати параметри, щоб використовувати всю доступну пам'ять, потужність ЦП і ємність вводу/виводу. MySQL також може масштабуватися до кластерів машин, об'єднаних разом.

MySQL Server спочатку був розроблений для роботи з великими базами даних набагато швидше, ніж існуючі рішення, і вже кілька років успішно використовується у дуже вимогливих виробничих середовищах. Хоча MySQL Server постійно розвивається, сьогодні пропонує багатий і корисний набір функцій. Завдяки можливості підключення, швидкості та безпеки MySQL Server дуже підходить для доступу до баз даних в Інтернеті.

MySQL Server працює в клієнт/сервер або вбудованих системах. Програмне забезпечення баз даних MySQL — це система клієнт/сервер, яка складається з багатопотокового SQL-сервера, який підтримує різні серверні системи, кілька різних клієнтських програм і бібліотек, інструментів адміністрування та широкого спектру інтерфейсів прикладного програмування (API). В MySQL Server є вбудована багатопотокова бібліотека, яку можна підключити до своєї програми, щоб отримати менший, швидший і простий у керуванні автономний продукт.

Доступна велика кількість програмного забезпечення MySQL.

MySQL Server має практичний набір функцій, розроблених у співпраці з користувачами. Майже усі додатки та мови програмування підтримують сервер баз даних MySQL.

2.3 Хмарні бази даних

Хмарна база даних — це база даних, яка зазвичай працює на платформі хмарних обчислень, і доступ до бази даних надається як послуга. Існує дві найпоширеніші моделі розгортання: користувачі можуть запускати бази даних у хмарі незалежно, використовуючи образ віртуальної машини, або вони можуть придбати доступ до служби бази даних, яку підтримує постачальник хмарної бази даних. Частина баз даних, доступних в хмарі, базуються на SQL, а інша частина використовує модель даних NoSQL. Служби бази даних відповідають за масштабованість та високу доступність бази даних та роблять базовий програмний стек прозорим для користувача.

Існує два основних методи запуску бази даних у хмарі:

Образ віртуальної машини

Хмарні платформи дозволяють користувачам придбати екземпляри віртуальних машин на обмежений час, і на таких віртуальних машинах можна запускати базу даних. Користувачі можуть або завантажити власний образ машини з встановленою на ній базою даних, або використовувати готові образи машин, які вже включають оптимізовану установку бази даних.

База даних як послуга (DBaaS)

При використанні бази даних як сервісної моделі власникам програм не потрібно самостійно встановлювати та підтримувати базу даних. Натомість постачальник служби бази даних бере на себе відповідальність за встановлення та обслуговування бази даних, а з власників додатків стягується плата відповідно до використання ними служби. Це тип програмного забезпечення як послуги (SaaS).

Архітектура та загальні характеристики

Більшість служб баз даних пропонують веб-консолі, які кінцевий користувач може використовувати для підготовки та налаштування екземплярів бази даних.

Служби бази даних складаються з компонента диспетчера баз даних, який керує базовими екземплярами бази даних за допомогою служби API. API служби доступний кінцевому користувачеві і дозволяє користувачам виконувати операції з обслуговування та масштабування своїх екземплярів бази даних.

Базовий стек програмного забезпечення зазвичай включає операційну систему, базу даних та інше програмне забезпечення, що використовується для управління базою даних. Постачальник послуг відповідає за встановлення, виправлення та оновлення базового стека програмного забезпечення, а також за забезпечення загального стану та продуктивності бази даних.

Функції масштабування у різних постачальників відрізняються: деякі пропонують автоматичне масштабування, інші дозволяють користувачеві масштабуватися за допомогою API, але не масштабуються автоматично.

Зазвичай існує зобов'язання щодо певного рівня високої доступності (наприклад, 99,9% або 99,99%). Це досягається за рахунок реплікації даних та передачі відмовостійких екземплярів на інші екземпляри бази даних.

При проектуванні та розробці типових систем як ключові будівельні блоки використовуються управління даними та реляційні бази даних. Розширені запити, виражені в SQL, добре працюють із суворими відносинами, накладеними інформацію реляційними базами даних. Однак технологія реляційних баз даних спочатку не проектувалась і не розроблялася для використання у розподілених системах. Цю проблему було вирішено шляхом додавання покращень кластеризації реляційних баз даних, хоча деякі основні завдання вимагають складних і дорогих протоколів, таких як синхронізація даних.

Сучасні реляційні бази даних показали низьку продуктивність у системах з інтенсивним використанням даних, тому ідея NoSQL була використана в системах управління базами даних для хмарних систем. У сховищі, реалізованому NoSQL, немає вимог до фіксованих схем таблиць, і уникає використання операцій з'єднання. Доведено, що бази даних NoSQL забезпечують ефективну горизонтальну масштабованість, нормальну продуктивність та простоту складання в хмарних програмах. Моделі даних, засновані на спрощених алгоритмах ретрансляції, також використовувалися в додатках картографування хмарних обчислень з інтенсивним використанням даних, унікальних для віртуальних середовищ.

Хмарні бази даних, які є реляційними:

Бази даних SQL — це один тип бази даних, який може працювати у хмарі або на віртуальній машині, або як служба залежно від постачальника. У той час як бази даних SQL легко масштабуються по вертикалі, горизонтальна масштабованість є проблемою, яку почали вирішувати служби хмарних баз даних на основі SQL.

Бази даних NoSQL — це ще один тип бази даних, який може працювати у хмарі. Бази даних NoSQL розраховані на великі навантаження читання/запису і можуть легко збільшуватися і зменшуватися, тому вони більше підходять для роботи в хмарі. Однак більшість сучасних програм

побудовані на основі моделі даних SQL, тому для роботи з базами даних NoSQL часто потрібно повністю переписати код програми.

У деяких базах даних SQL було розроблено можливості NoSQL, включаючи JSON, двійковий JSON (наприклад, BSON або аналогічні варіанти) та типи даних сховища "ключ-значення".

Мультимодельні бази даних з реляційними та нереляційними можливостями забезпечують стандартний інтерфейс SQL для користувачів та додатків та, таким чином, полегшує використання таких баз даних для сучасних програм, побудованих на основі моделі даних SQL. Власні бази даних з кількома моделями підтримують кілька моделей даних з одним ядром та уніфікованою мовою запитів для доступу до всіх моделей даних.

2.4 Технологія QR-код та зчитування QR-коду з растрового зображення

2.4.1 Структура QR-коду

У 1992 році в Японії виникла потреба в розробці компактного коду, який міг би зберігати більше інформації, такої як буквено-цифрові символи, символи кандзі (логографічні китайські символи) та символи кана (японський складовий лист). Японська компанія Denso Wave Incorporated займалася розробкою зчитувачів штрих-кодів. Щоб задовольнити цей попит, команда Denso Wave з двох людей під керівництвом Масахіро Хара приступила до розробки нового двовимірного коду.

У традиційних штрих-кодах інформація може бути закодована тільки в одному вимірі (поперечному напрямку), у той час як у двовірних (2D) кодах інформація може бути закодована як у поперечному, так і поздовжньому напрямках. Створення 2D-кодів, які можна сканувати якнайшвидше, було серйозним завданням для команди розробників, оскільки сканерам було важко розпізнати розташування 2D-коду.

Якось Харе спала на думку ідея додати в код інформацію про місцезнаходження, що вказує сканерам його місцезнаходження. На основі цієї

ідеї в трьох кутах кожного коду був доданий шаблон визначення положення, щоб сканери могли швидше точно розпізнавати 2D-коди.

Однак розробка форми шаблону виявлення положення була надзвичайно складним завданням, тому що коли поруч із кодом додавалася фігура аналогічної форми, шаблон не міг бути точно розпізнаний. Щоб запобігти помилковому розпізнаванню, шаблон виявлення положення повинен був мати унікальну форму.

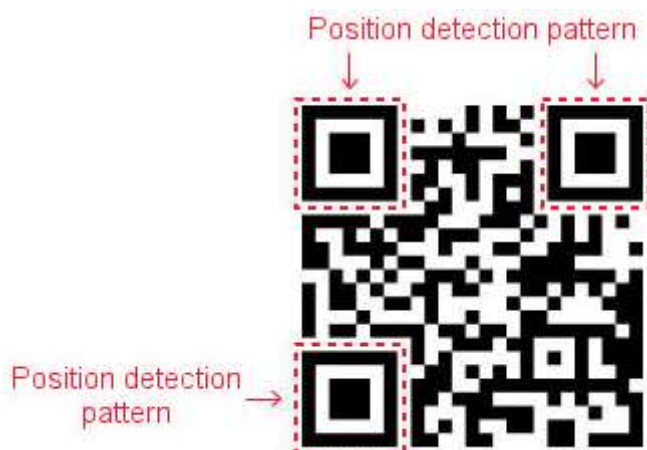


Рисунок 2.1 – Патерни визначення напрямку QR-коду

Вивчивши співвідношення білих та чорних областей на зображеннях та символах, надрукованих на листівках, журналах, картонних коробках та інших документах, команда розробників скоротила шаблон позиційного виявлення до шаблонів із чорними та білими областями. Команда вивчила безліч друкованих документів і, нарешті, визначила співвідношення, яке найменше з'являлося у друкованих матеріалах. Співвідношення було 1:1:3:1:1.

Таким чином, була визначена ширина чорних та білих областей у шаблоні виявлення положення, і сканери змогли виявити код незалежно від кута сканування, знайшовши це унікальне співвідношення.

Через півтора роки після початку розробки, в результаті численних спроб і помилок, була успішно розроблена система QR-коду, яка може зберігати до 7089 буквено-цифрових символів з додатковою можливістю кодування символів Кандзі/Кана.

QR-коди були вперше представлені 1994 року компанією Denso Wave для автомобільної промисловості Японії. QR-коди дозволяли швидко сканувати компоненти та використовувалися для відстеження транспортних засобів під час виробництва. QR-коди швидко завоювали популярність за межами автомобільної промисловості завдяки своїм компактним розмірам, швидкому зчитуванню та більшій ємності пам'яті.

Штрих-коди

Ми всі бачимо штрих-коди у повсякденних продуктах. І QR-коди, і штрих-коди створені на основі абетки Морзе. Штрих-коди працюють, висвітлюючи яскравим світлодіодом чи лазером серію подовжених ліній. Ми знаємо, що чорний колір поглинає світло, а білий його відбиває. Сканер штрих-коду зчитує зразок світла, відображеного назад у світлочутливий електронний компонент, званий фотоелементом.

Коли сканер проходить повз штрих-код, фотоелемент генерує послідовність імпульсів включення-вимкнення, які відповідають чорним і білим смугам. Схема електронного декодера, підключена до сканера, швидко перетворює ці імпульси включення-вимкнення на двійкові цифри (нулі та одиниці). Ці двійкові цифри надсилаються на комп'ютер, підключений до сканера штрих-коду, який виявляє двійковий код.

Усі штрих-коди можуть виглядати однаково неозброєним оком і мати спільні риси, такі як початковий та кінцевий коди, що дозволяє правильно зчитувати їх незалежно від орієнтації. Штрих-коди обмежені приблизно 20 буквено-цифровими символами. Ось тут і з'являються QR або коди швидкої відповіді.

QR-коди

QR-коди зовні відрізняються від штрих-кодів, але працюють однаково. Штрих-коди мають вертикальні смуги різної товщини, а QR-коди мають квадратну форму та містять чорні квадрати, розташовані у вигляді квадратної сітки на білому тлі, які можна сканувати пристроєм обробки зображень,

наприклад камерою смартфона. Для деяких телефонів може бути потрібна програма для читання QR-кодів, яку можна безкоштовно завантажити з магазинів Android або iPhone.

У QR-кодах використовуються стандартні режими кодування (числовий, буквено-цифровий, байтовий/двійковий та кандзі/кана) для ефективного зберігання даних. Обсяг даних, які можна зберегти у зображенні QR-коду, залежить від типу даних, версії та рівня виправлення помилок. Коли QR-код зчитується сканером QR-коду, код спочатку виявляється двовимірним цифровим датчиком зображення, а потім аналізується у цифровому вигляді запрограмованим процесором у пристрої.

Версії QR-кодів

QR-коди бувають різних версій. Вони можуть змінюватись від версії 1, яка має 21 рядок і 21 стовпець модулів, версія 2 має 25 рядків і 25 стовпців модулів і так далі. Версія 40, має 177 рядків і 177 стовпців модулів, має максимальну ємність пам'яті і низький рівень виправлення помилок, що позначається 40-L. Нижче показані приклади версій QR-коду версії 1 та 40.

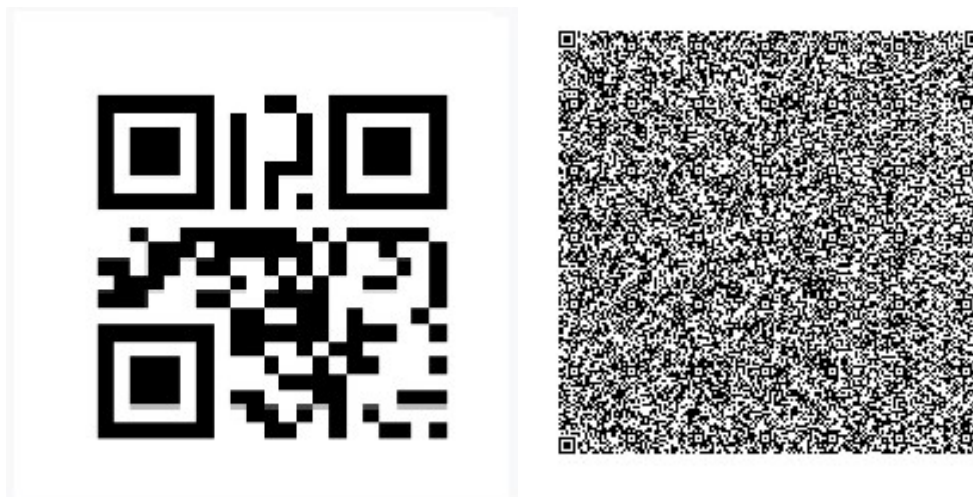


Рисунок 2.2 – QR-код версії 1 та 40

Модулі

Модуль – це основна одиниця QR-коду. Це чорні та білі блоки, з яких складаються дані, закодовані у QR-коді. Модулі зберігають дані та становлять рядки та стовпці QR-коду, званого матрицею даних. Позиціонування бітів

даних починається з нижнього правого кута матриці і переміщується вгору в стовпці шириною в два модулі і змінює напрямок при досягненні вершини, як показано нижче.

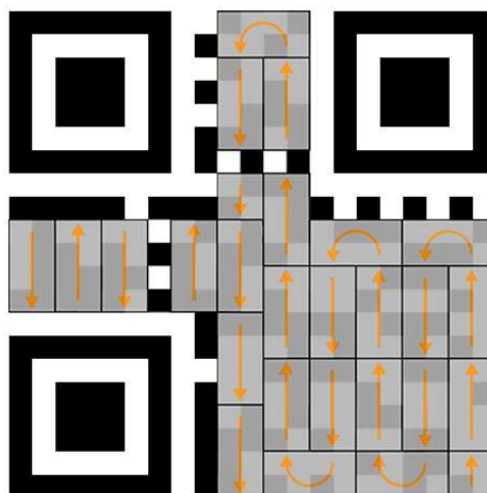


Рисунок 2.3 – Позиціонування бітів даних

Шаблони пошуку

Процесор у зчитувачі QR-коду спочатку знаходить три характерні квадрати (шаблони пошуку) у трьох кутах зображення QR-коду, щоб визначити положення та поворот коду. Більші версії QR-кодів включають менші структури вирівнювання близько четвертого кута. Однак, показаний тут QR-код версії 1 не має маркерів вирівнювання.

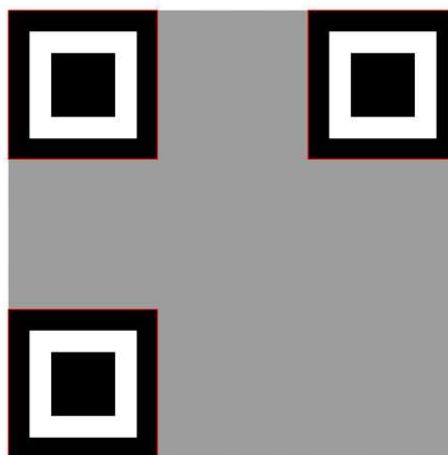


Рисунок 2.4 – Позиціонування шаблонів пошуку

QR-код повинен мати порожній простір, званий тихою зоною навколо кордону QR-коду. Тиха зона допомагає сканеру QR-коду шукати шаблони пошуку.

Сепаратори

Розділювачі допомагають сканеру розрізняти шаблони пошуку та інші фактичні дані у QR-коді.

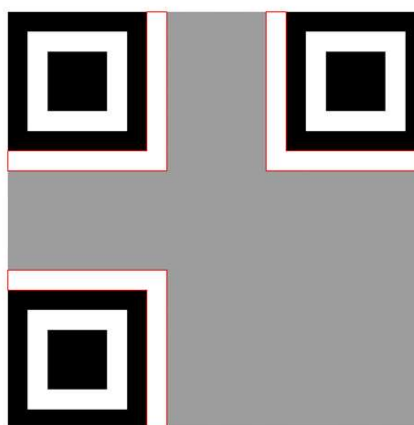


Рисунок 2.5 – Позиціонування сепараторів

Інформація про формат

Наступна структура QR-коду використовується для позначення формату даних, що зберігаються у коді. Інформація про формат містить дві частини даних – рівень виправлення помилок та шаблон маски.

Схема синхронізації

Наступна структура, яка зустрічається у всіх QR-кодах, це шаблон синхронізації. Цей шаблон вирівнюється по краю зовнішніх квадратів позиціонування і чергується між двома кольорами коду розміром 1 модуль. Це дозволяє зчитувачу QR-коду підтвердити версію цього коду, а також синхронізацію бітів по обох осях.

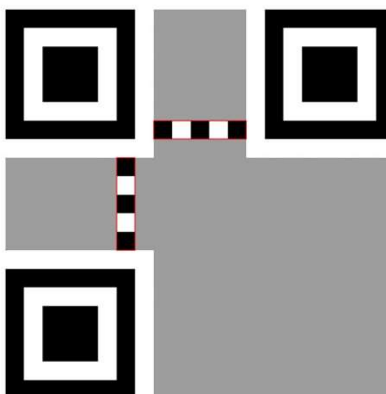


Рисунок 2.6 – Позиціонування схеми синхронізації

Інформація про формат

Наступна структура QR-коду використовується для позначення формату даних, що зберігаються у коді. Інформація про формат містить дві частини даних – рівень виправлення помилок та шаблон маски.

Рівень виправлення помилок

QR-коди можуть мати один із чотирьох типів рівнів виправлення помилок. Чим вищий рівень виправлення помилок, тим менший обсяг пам'яті. У наступній таблиці наведено приблизні можливості виправлення помилок на кожному з чотирьох рівнів:

Таблиця 2.1 – Рівні можливості виправлення помилок

Рівень L (низький)	Можна відновити 7% байтів даних
Рівень M (середній)	Можна відновити 15% байтів даних
Рівень Q (квартиль)	Можна відновити 25% байтів даних
Рівень H (Високий)	Можна відновити 30% байтів даних

Шаблон маски

Друга частина шаблону формату означає маску, яка застосовується до даних. Маскування - інвертування кольорів модулів. При маскуванні темні модулі перетворюються на світлі, а світлі на темні. Маска використовується для розділення великих частин однакових бітів, щоб зробити шаблон більш читаним для сканерів при декодуванні QR-коду. Версії більшого розміру включають шаблон формату, який визначає код версії. Існує вісім різних

шаблонів масок, які використовуються в QR-кодах, і кожен може бути представлений формулою, яка враховує координати модулів, щоб визначити, які модулі замасковані.

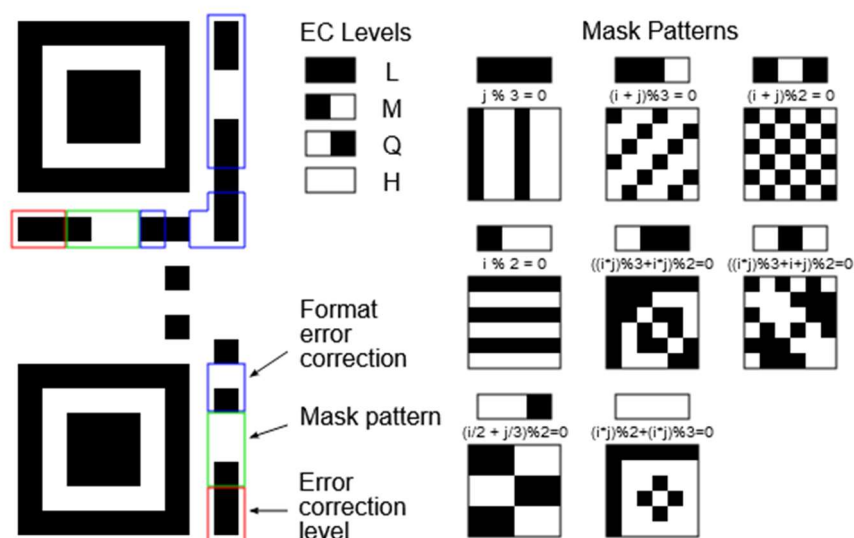


Рисунок 2.7 – Шаблони масок у QR-кодах

Виправлення помилок формату

Розділ шаблону формату, що залишився, — це виправлення помилок для п'яти бітів, які ми розглянули досі. Необхідні дані вилучаються з патернів, присутніх як у горизонтальній, так і вертикальній складових зображення.

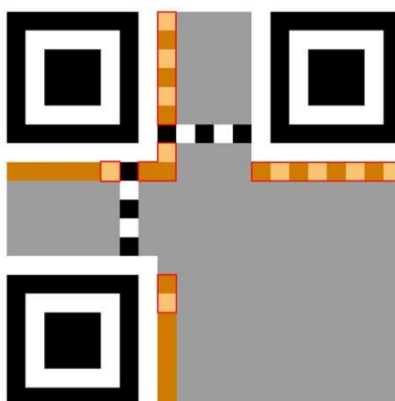


Рисунок 2.8 – Шаблони виправлення помилок формату

Невикористований біт даних

Наступна частина QR-коду — один біт, який завжди представлений як 1. Цей біт не потрібен для даних та інформації про формат. Це відносно не використовується частина коду.

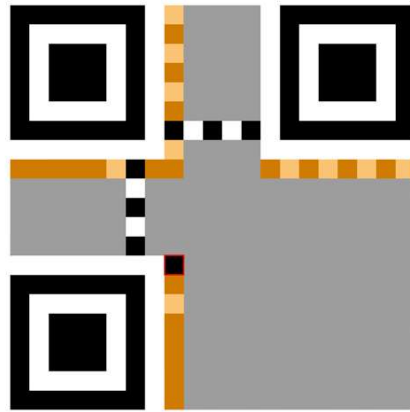


Рисунок 2.9 – Не використаний біт даних

Дані

Решта QR-коду використовується для представлення збережених даних. Дані розташовані таким чином, що вони починаються в нижньому правому куті, починаючи з найстаршого біта даних, а потім слідує шаблону, як показано нижче.

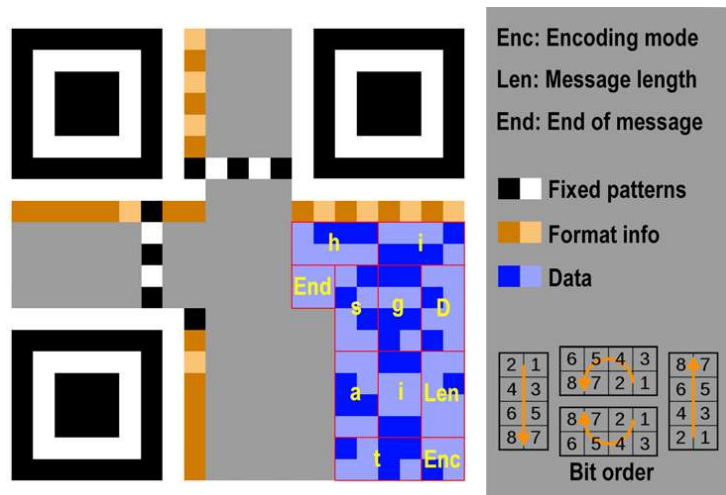


Рисунок 2.10 – дані у QR-кодах

Кодування (Enc)

Перші чотири біти визначають тип кодування, у якому зберігаються дані. Сюди входять числові, буквено-цифрові символи, кандзі/кана, цілі байти, а також інші складніші структури. Наші дані зберігатимуться цілими байтами, тому відповідний формат кодування зберігається у першому байті даних.

Примітка. QR-код може зберігати кілька різних типів даних в тому самому коді, де кожен розділ починається зі свого власного індикатора кодування.

Довжина повідомлення (Len)

Другий байт даних зберігає довжину повідомлення. Текстове повідомлення, яке зберігається в цьому QR-коді - Digitash. Наше повідомлення має довжину 8 байт, яке представлене у двійковому вигляді як 00001000 та перенесено у другий байт.

Повідомлення

Наступний байт зберігається в великій літері ASCII «D», яка представлена двоичними бітами 01000100. Ці двійкові біти переносяться в наступний байт у нашому просторі даних. Так само всі інші літери нашого повідомлення зберігаються в наступних 7 байтах даних.

Кінець повідомлення (End)

Наслідуючи наші дані, наступні чотири біти в нашому коді - це індикатор кінця повідомлення. Цей індикатор містить чотири нулі, які представлені чотирма світловими модулями в QR-коді.

Виправлення помилки

Тепер, коли повідомлення в нашому QR-коді завершено, у нас все ще є 16 байтів пам'яті, що залишилися, які використовуються для зберігання даних виправлення помилок нашого повідомлення. QR-код використовує виправлення помилок Ріда-Соломона доти, доки зображення не буде правильно інтерпретовано.

Після застосування даних виправлення помилок Ріда-Соломона та шаблону маски наш QR-код готовий. Завдяки вбудованій корекції помилок QR-коди менш схильні до пошкодження водою або брудом. Навіть за суворих виробничих умов QR-коди можна правильно сканувати. Наведений нижче QR-код, як і раніше, може правильно скануватися, навіть якщо всі ці кольори та текст додані поверх коду.

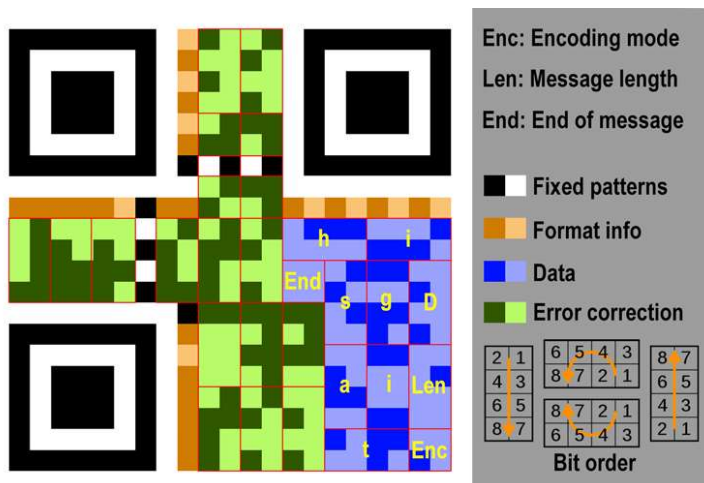


Рисунок 2.11 – шаблон виправлення помилки

2.4.2 Розпізнавання QR-коду

Розпізнавання та аналіз QR-коду можуть бути засновані на пакеті ZXing. Складність полягає в тому, як швидко і точно знайти область двовимірного коду на екрані і знайти три збігові точки шаблону двовимірного коду.

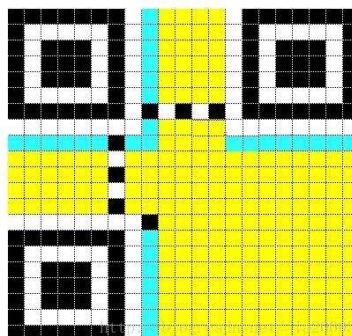


Рисунок 2.12 – Вигляд структури QR-коду

Особливу увагу слід приділити трьом чорним квадратним областям на малюнку, які використовуються для визначення розташування двовимірного коду. Перше, що потрібно зробити для сканування та виявлення двовимірного коду, – це знайти ці три області. Вони потрібні для того, щоб зчитувач міг зрозуміти кут нахилу та відносно них зчитував рисунок з QR-кодом. Знайшовши ці три області зчитувач може зрозуміти, де знаходиться QR-код, і може знайти та ідентифікувати його. Інші частини QR-коду:

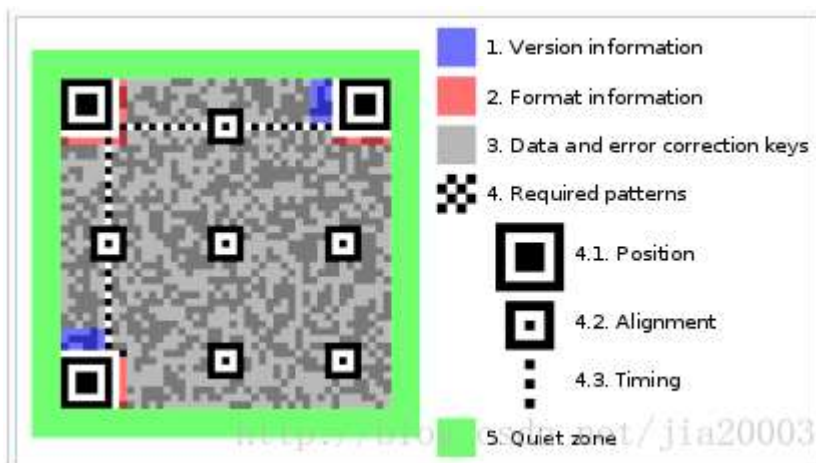


Рисунок 2.13 – Інша частина QR-коду

Квадратні області в трьох кутах розташовані ліворуч, а відношення чорного до білого становить 1: 1: 3: 1: 1 зверху вниз. Спочатку перетворюється вхідне зображення на зображення в градаціях сірого:



Рисунок 2.14 – Зображення в градаціях сірого

Перетворюється на двійкове зображення по ОТСУ



Рисунок 2.15 – Двійкове зображення по ОТСУ

Використовується виявлення контурів, щоб отримати контури для бінарних зображень.

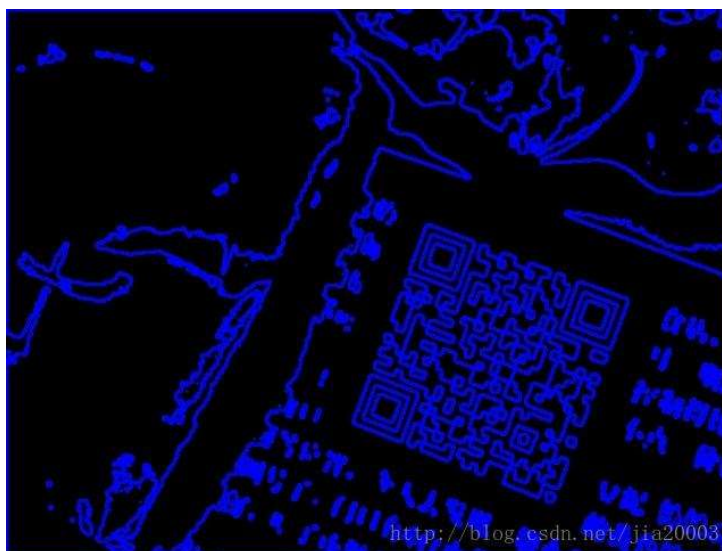


Рисунок 2.16 – двійкове зображення виявлення контурів

На наступному малюнку - вихідне зображення зліва та результат позиціонування QR-коду праворуч.



Рисунок 2.17 – Результат позиціонування QR-коду

2.5 Принципи роботи NFC

NFC-чіп

Система зв'язку NFC складається з двох окремих частин: чіпа-зчитувача NFC і мітки NFC. Чип зчитувача NFC є активною частиною системи, він «зчитує» (або обробляє) інформацію, перш ніж ініціювати певну відповідь. Він забезпечує живлення та відправляє команди NFC на пасивну частину системи, мітку NFC.

Технологія NFC часто використовується у громадському транспорті, де користувачі можуть платити за свій квиток за допомогою NFC або смартфона. У цьому прикладі чип зчитувача NFC буде вбудований в термінал оплати автобуса, а пасивна мітка NFC перебуватиме в квитку (або смартфоні), який отримує та відповідає на команди NFC, надіслані терміналом.

Існує три типи мікросхем NFC: зчитувачі NFC, мітки NFC та мікросхеми контролера NFC.

Чип зчитувача NFC

Чип зчитувача NFC можна як основний контролер системи зв'язку, оскільки він ініціює зв'язок, включає мітку NFC і відправляє команди через магнітне полі на пасивну мітку. Пристрій читання NFC також можна назвати пристроєм запису NFC через його здатність записувати дані в тег NFC.

Зазвичай у поєднанні з мікроконтролером чіп зчитувача NFC включається та обмінюється інформацією з однією або декількома мітками NFC. Підтримуючи кілька радіочастотних протоколів та функцій, чіп зчитувача NFC можна використовувати у трьох різних режимах: читання/запис, одноранговий зв'язок (P2P) та емуляція карти. Чипи зчитувачів NFC часто є вбудованими в смартфони, платіжні термінали, квиткові автомати, дверні ручки автомобілів, центральні консолі автомобілів або є частиною ширшої платіжної системи, як автомобільний або транспортний додаток або ігровий пристрій.

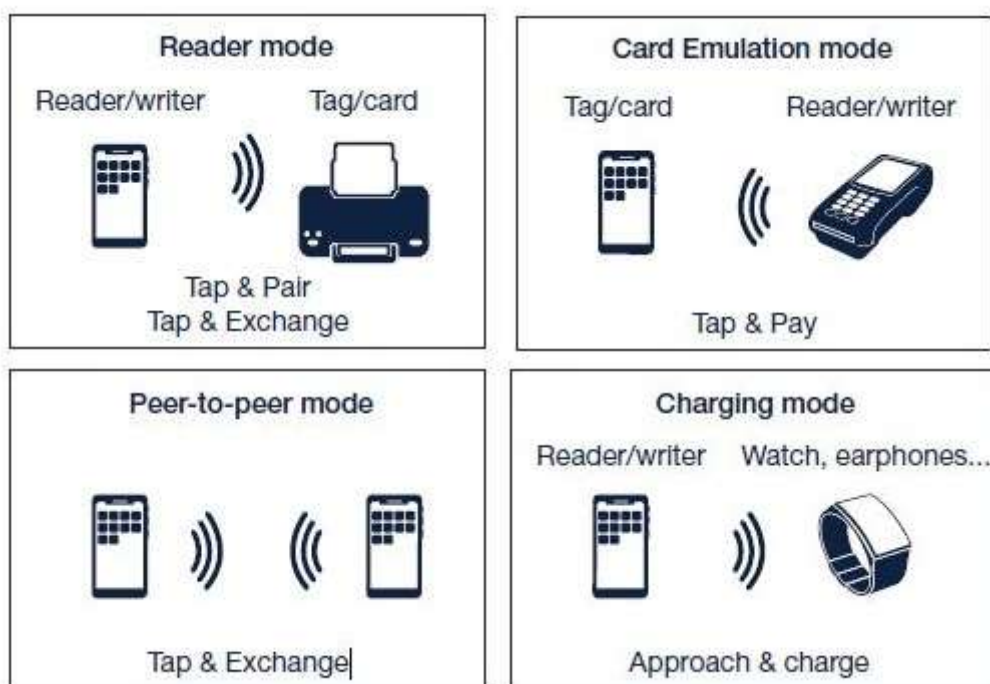


Рисунок 2.18 – Варіанти використання чіпу зчитування NFC
Чіп мітки NFC

Чіп мітки NFC є пасивним пристроєм: вбудований в антену, він живиться від магнітного поля, що створюється зчитувачем NFC (наприклад, смартфоном).

Мітка NFC відповідає на певні інструкції NFC. Наприклад, він може відповісти на таку команду, як "Прочитати команду", відобразивши вміст своєї пам'яті. Він також може включати спеціальні функції, що забезпечують захист паролем, автентифікацію або виявлення несанкціонованого доступу за умови, що ці функції попередньо інтегровані в мітку.

Чіпи тегів NFC часто використовуються в таких додатках як точки дотику NFC для:

- взаємодії зі споживачами;
- аутентифікації;
- контролю доступу;
- відстеження активів;
- домашньої автоматизація;
- вимірювання;
- освітлення.

Чіпи NFC можна монтувати безпосередньо на пластикову плівку або тканину для формування мітки NFC. Вони можуть бути поміщені в пластиковий корпус для монтажу на друкованій платі (PCB).

Існує два типи міток NFC: чіпи NFC з одним інтерфейсом та з двома інтерфейсами.

Чіп NFC з одним інтерфейсом – це пасивний пристрій, вбудований у мітку NFC, який взаємодіє лише з іншою частиною системи NFC, зчитувачем NFC. Після активації зчитувач NFC чіп NFC з єдиним інтерфейсом забезпечує обмін даними між міткою і зчитувачем, коли вони знаходяться в безпосередній близькості.

NFC — єдина бездротова технологія, яку можна використовувати для взаємодії з мітками NFC. Їх невеликий розмір дозволяє легко інтегрувати їх в

об'єкти з обмеженим простором, такі як наклейки, етикетки, брелоки та багато іншого.

Крім можливості читання та запису даних, деякі мікросхеми NFC також підтримують певні функції, такі як механізми виявлення несанкціонованого доступу та цифровий підпис, які підходять для програм ідентифікації та аутентифікації. Наприклад, коли смартфон з підтримкою NFC знаходиться на відстані зчитування від винної пляшки, в пробку якої вбудована мітка NFC, зчитувач NFC може перевірити походження мітки та визначити, чи була пляшка відкрита, тим самим гарантуючи безпеку продукту.



Рисунок 2.19 – Приклад інтегрування чіпу NFC у кришку пляшки.

Чіп NFC з подвійним інтерфейсом

Мітка з подвійним інтерфейсом, також звана динамічною міткою NFC, пов'язана з мікроконтролером через провідний інтерфейс (наприклад, I2C), тим самим пропонуючи другий інтерфейс зв'язку на додаток до бездротового зв'язку NFC.

Мікросхема динамічної мітки NFC забезпечує двосторонній бездротовий зв'язок між двома електронними системами, навіть якщо на об'єкт, що несе динамічну мітку NFC, не подається живлення.

Динамічна мітка NFC діє як міст між двома електронними системами та здатна до обміну даними та збору енергії з радіочастотного поля NFC для живлення мікроконтролера.

Наприклад, якщо смартфон з підтримкою NFC знаходиться на відстані читання електронного пристрою (наприклад, гарнітури Bluetooth LE), BLE виконується автоматично без будь-яких додаткових дій з боку користувача. Зчитувач NFC (смартфон) виявляє динамічну мітку NFC, вбудовану в гарнітуру, та відновлює інформацію, яка забезпечує пару BLE.

Динамічні позначки NFC також можна використовувати для контролю стану товарів. Наприклад, смартфон з підтримкою RFID або NFC може зчитувати інформацію, вбудовану в мітку NFC пральної машини, наприклад, відомості про стан машини, включаючи діагностику неполадок. Оскільки багато смартфонів зараз пропонують можливість підключення NFC, динамічні мітки NFC можуть використовуватися споживчими брендами для відправки клієнтам інформації про придбані ними продукти через мобільний додаток, тим самим підвищуючи зручність використання NFC.

Для конкретних програм, таких як управління освітленням або двигуном, динамічні мітки NFC можуть бути розроблені без використання мікроконтролера. У цьому випадку інтерфейс широтно-імпульсної модуляції, вбудований у динамічний тег, дозволяє користувачам налаштовувати драйвери світлодіодів на етапі виробництва або під час обслуговування.

Чіп контролера NFC

Мікросхема контролера NFC поєднує як зчитувач NFC, і мітку NFC для інтегрованого рішення. Він часто вбудований у пристрої з підтримкою NFC, такі як:

- смартфони;
- мобільна точка продажу (POS);
- програми для електронних гаманців (банківські картки, автобусні квитки, медичні картки, картки лояльності тощо);

- цифрові ключі від машини;
- бездротові зарядні пристрої.

Чіп контролера NFC забезпечує безпеку транзакцій завдяки наявності захисного елемента або SIM-карти та працює в режимі емуляції картки, тобто діє як безконтактна смарт-картка.

2.6 PhpMyAdmin та Openserver

PhpMyAdmin — це інструмент, який використовується для управління та адміністрування баз даних, таких як MariaDB та MySQL. Цей інструмент є безкоштовним, з відкритим кодом, запрограмованого з використанням скриптової мови програмування PHP, на основі Microsoft MySQL – WebAdmin. Цей інструмент призначений для використання більш ніж однієї системи баз даних або систем управління базами даних, а також підключених до них серверів та вузлових одиниць.

Інструмент адміністрування phpMyAdmin містить нескладний графічний інтерфейс користувача, який може надати користувачеві зручний засіб редагування запитів для створення, оновлення, видалення та маніпулювання різними операціями над будь-яким SQL-запитом. Це також дозволяє користувачам копіювати та вставляти запити з інших систем, відкривати або імпортувати файл, що містить запит, виконання поточного або всіх запитів, відстежувати час виконання кожного запиту, запущеного в інструменті, відображати для перегляду виводу, що витягуються з виконуваних запитів і т.д.

PhpMyAdmin можна визначити як сервісний інструмент, де послуги грають роль встановлення зв'язку між системами баз даних та відповідними об'єктами сервера. Цей інструмент створено за допомогою програми PHP для роботи з добре обладнаним веб-інтерфейсом як зовнішній додаток для підключення та роботи з системами баз даних. Дозволяє вилучати результати з цієї програми після виконання необхідних операцій з базою даних, зберігати

їх окремо, або експортувати в бажаному формат файлу, такому як CSV, CSQ, PDF, Excel, Word, XML.

На відміну від інших інструментів управління базами даних, phpMyAdmin включає в себе механізми зберігання даних та можливість доступу до них. Коли користувач натискає опцію Engines, представлену у верхній частині інтерфейсу програми, система переходить до відображення відомостей про Engine. Він показує список всіх систем зберігання, здатних підключатися до phpMyAdmin, разом із відповідними серверами. Як установлений параметр підключений механізм зберігання буде відображатися як MyISAM з пам'яттю на основі хеша для зберігання тимчасових таблиць.

Також додаток дозволяє користувачам створювати таблиці, додавати або оновлювати стовпці та рядки, додавати або оновлювати вміст таблиці тощо. Необхідно ввести ім'я таблиці, тип таблиці, довжину/значення, атрибути, індекси, коментарі тощо для створення таблиць, а також вставки, переходу, перегляду вкладок і т. д. для додавання даних та елементів до таблиці.

Ще однією важливою функцією інструменту phpMyAdmin є функція «Пошук». Це допомагає користувачеві пройти через всю систему бази даних та підключені вузли, коли користувач запитує операцію пошуку, використовуючи ключ щодо вмісту, який потрібно знайти.

Переваги та недоліки phpMyAdmin:

Переваги

- витончені операції обслуговування без будь-яких додаткових витрат, крім тих, які будуть витрачені на володіння системою управління базами даних.
- підтримує та діє гнучко для більшості широко використовуваних форматів файлів, що зручно для цілей документування під час роботи з системами управління базами даних.
- окремі панелі для роботи з базою даних, редагування SQL-запитів, відстеження статусу тощо

- відображає усі активні плагіни підключені до баз даних.

Недоліки

- його не можна використовувати для всіх без винятку баз даних, оскільки він підтримує лише MySQL та MariaDB.
- хоча додаток підтримує класичні сервери та системи, його масштаб не такий великий через зростання галузевих стандартів.

Openserver — це портативний локальний WAMP/WNMP-сервер з багатофункціональною програмою керування та великим вибором доповнюваних компонентів. Представлений пакет програм є повноцінним професійним інструментом, створеним для веб-розробників з урахуванням їхніх рекомендацій і побажань.

Компоненти та інструменти

Для налагодження скриптів у різних середовищах Open Server пропонує на вибір відразу два типи HTTP-серверів, різні версії модулів PHP і СУБД, а також можливість швидкого перемикання між ними.

- HTTP-модулі: Apache 2.2.21 і Nginx 1.0.11;
- модулі СУБД: MySQL 5.1.61, MySQL 5.5.20 та PostgreSQL 9.1.1;
- модулі PHP: PHP 5.2.17 (IMagick 2.2.1, Zend Optimizer 3.3.3, IonCube Loader 4.0.7, Memcache 2.2.4) і PHP 5.3.9 (IMagick 2.3.0, Xdebug 2.1.3, Loder 4.0.4 10, Memcache 2.2.6).

Можливості Openserver:

- детальний перегляд журналів усіх компонентів в режимі реального часу;
- вибір модулів HTTP, СУБД і PHP в будь-якій комбінації;
- підтримка доменів SSL та кирилиці з коробки;
- підтримка псевдонімів або інших указників домену, а також зручна форма їх налаштування;
- створення локального субдомену без втрати видимості основного домену в Інтернеті;
- доступ до доменів і швидкий доступ до шаблонів конфігурації модулів;

– багатомовний інтерфейс (російська, українська, білоруська, англійська).

Висновки до розділу 2

У другому розділі було проведено аналіз методів та інформаційних технологій для вирішення поставленої задачі. Проаналізовано найпопулярнішу реляційну базу даних MySQL. Варіант розгортання бази даних у хмарі має гарний потенціал для масштабних комерційних проєктів, бажанням передати контроль та управління хостингом в інші руки. Для розробки системи було обрано openserver. Причиною цього є відсутність потреби викладення сайту у відкритий доступ, зручні можливості управління хостингом та портативність. Серед методів управління базою даних було обрано додаток phpMyAdmin через зручний для користувача інтерфейс управління та можливостями експорту бази даних. Серед методів реалізації перевірки квитків було проведено аналіз технології NFC, котра надає великі можливості до перевірки, минаючи етап зчитування QR-коду, та надаючи одразу доступ до бази даних. Цей спосіб є найсучаснішим та найбільш ефективним, але вимагає наявності в розробника чіпів NFC та забезпечення наявності у всіх користувачів технології NFC на пристрої, що сильно обмежує потенційних користувачів додатку. Після вивчення технології QR-коду його було обрано за основу етапу перевірки квитка через її велике поширення у сучасному світі, та зручність у використанні.

3 ПРОЦЕС РОЗРОБКИ СИСТЕМИ

3.1 План функціонування системи перевірки

Розглядаючи принцип роботи системи перевірки, потрібно створити план її роботи.



Рисунок 3.1 – План роботи системи

Ідея полягає в тому, що на головній сторінці буде знаходитися 2 кнопки, за якими можна перейти на 2 різні сторінки. На першій вкладці буде знаходитися форма для створення QR-коду квитка пасажира, після заповнення котрої буде виводитися результуюча картинка з QR-кодом та зберігатися на носії в потрібну папку для наступних дій. Друга вкладка буде містити форму для заповнення назви файлу в заздалегідь зазначеної директорії, котрий буде зчитано скриптом. Результатом виконання скрипта буде таблиця з даними самого квитка, що буде свідчити про те, що в базі даних знаходиться такий квиток, даючи можливість допустити пасажира до проїзду.

Односторінкові та багатосторінкові сайти.

На односторінковому веб-сайті весь контент знаходиться на одній веб-сторінці HTML або документі HTML. Частіше за все використовується в маркетингових цілях.

Існує кілька різновидів сайтів односторінників. До основних їх типів можна віднести такі:

– лендинг продажу – основна мета створення такого сайту впливає з його назви. Вона спрямована на продаж конкретного товару чи послуги прямо зараз;

– передплатний – такий варіант лендингу передбачає збір контактних даних відвідувачів для подальшого розсилання інформації про товар, послугу, пропозицію тощо. Зазвичай пропонується залишити електронну адресу або номер мобільного телефону;

– інформаційний – у разі відвідувачам надається можливість познайомитися з новим продуктом, послугою, подією тощо. Основна мета – підігріти інтересу аудиторії;

– вірусний – такий тип посадкових сторінок найчастіше використовується для просування будь-якого бренду. Основне завдання полягає в тому, щоб схилити відвідувача зробити репост сайту в соціальних мережах або іншим способом поширити інформацію про нього;

– диференціальний – даний тип лендингу поєднує у собі відразу кілька наведених вище різновидів. Найчастіше використовується інтернет-магазинами;

Інший тип — веб-сайт із кількома сторінками (кілька HTML-документів). Кожна веб-сторінка HTML має різноманітний контент. Вони можуть мати або не мати загальні стилі, включаючи елементи заголовка та нижнього колонтитулу.

Переваги багатосторінкового сайту

Веб-сайт з обмеженим вмістом для відображення кінцевим користувачам може використовувати переваги односторінкового веб-сайту. Але є так багато причин, через які ми бачимо багато сторінок веб-сайтів в Інтернеті. Такий спосіб краще розділяє вміст на сайті. Замість того, щоб додавати весь контент для веб-сайту на одну сторінку за допомогою заголовків, багатосторінковий веб-сайт може краще керувати та

демонструвати контент на різних сторінках залежно від контексту. Багатосторінковий веб-сайт пропонує кращий інтерфейс користувача і користувальницький досвід. Користувачі використовують верхню навігацію для навігації між веб-сторінками веб-сайту. Меню навігації дає зрозуміти, що на такій сторінці знаходиться такий контент. Зручний для пошукових систем. Коли у вас є багато контенту, який відрізняється за контекстом між ними, це чудова ідея мати багатосторінковий веб-сайт. Не тільки користувачі зможуть краще зрозуміти вміст веб-сайту, але і роботи пошукових систем, які сканують ваш веб-сайт, зможуть краще розділяти вміст на основі веб-сторінок.

Для більш зручної навігації у системі було обрано багатосторінковий тип, у якому кожна сторінка буде містити свою інформацію та виконувати поставлену за планом задачу.

База даних.

В базі даних буде створено таблицю, яка буде зберігати інформацію з даними квитка. Квиток буде складатися з таких табличних даних:

- ідентифікатор;
- ім'я пасажера;
- прізвище пасажера;
- дата народження пасажера;
- дата та час відправлення;
- дата та час прибуття;
- місто відправлення;
- місто прибуття;
- зазначена ціна квитка.

Ідентифікатор буде відповідати за доступ до окремої строки таблиці, тобто до самого квитка, може бути словом, числом, літерою, символом або будь-якою їх комбінацією. За замовченням в додатку phpMyAdmin є функція «AUTO_INCREMENT», яка відповідає за автоматичну нумерацію нового елемента таблиці, тому ідентифікатор в цьому випадку буде числом.

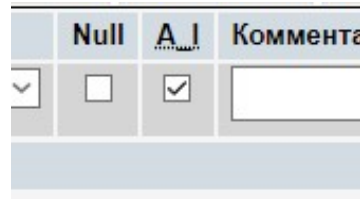


Рисунок 3.2 – Функція «AUTO_INCREMENT» в phpMyAdmin

Дані про Ім'я, прізвище та дату народження потрібні для підтвердження особи, яку буде допущено до поїздки після перевірки білету. Решта даних квитка буде потрібна для перевірки чи на той саме рейс потрапив пасажир.

Реалізація процесу створення та зчитування квитка буде виконана за допомогою заздалегідь установлених бібліотек PHP та власноруч написаних скриптах. Потрібні бібліотеки буде встановлено за допомогою менеджера залежностей Composer.

В сфері програмування бібліотека є набором енергонезалежних ресурсів, що використовуються комп'ютерними програмами для розробки програмного забезпечення. Вони можуть включати дані конфігурації, документацію, довідкові дані, шаблони повідомлень, попередньо написаний код та підпрограми, класи, значення або специфікації типів.

Бібліотека також є набором реалізацій поведінки, наприклад, люди, які хочуть написати програму вищого рівня, можуть використовувати бібліотеку для виконання системних викликів замість того, щоб виконувати ці системні виклики знову і знову. Програма викликає поведінку, яку надає бібліотека, через механізм мови. Наприклад, у простій імперативній мові поведінка в бібліотеці викликається за допомогою звичайного виклику функції. Виклик функції бібліотеки від виклику звичайної функції в такій програмі відрізняється способом організації коду в системі. Відмінною особливістю є те, що бібліотека організована для повторного використання незалежними програмами чи підпрограмами, і користувачеві потрібно знати лише інтерфейс, а не внутрішні деталі бібліотеки.

Цінність бібліотеки полягає у повторному використанні стандартизованих елементів програми. Коли програма викликає бібліотеку, вона отримує поведінку, реалізовану всередині бібліотеки, без необхідності реалізовувати цю поведінку самостійно. Бібліотеки заохочують обмін кодом за модульним принципом та спрощують поширення коду.

Якщо доступ до коду бібліотеки здійснюється під час складання програми, бібліотека називається статичною бібліотекою. Альтернативою є створення файлу програми, що виконується, і його підключення незалежно від реалізації бібліотеки. Поведінка бібліотеки підключається після того, як файл викликається для виконання з початку процесу запуску виконання, або в середині виконання. Така бібліотека називається динамічною бібліотекою (завантажується під час виконання). Динамічна бібліотека може бути завантажена та пов'язана компоувальником під час підготовки програми до виконання. В якості альтернативи, в середині виконання програма може явно запросити завантаження модуля.

Для підключення бібліотек у проект PHP використовується менеджер залежностей Composer.

Composer — це інструмент керування залежностями з відкритим вихідним кодом для PHP, створений насамперед для полегшення розповсюдження та обслуговування пакетів PHP як окремих компонентів програми. Composer дозволяє встановлювати необхідні бібліотеки для кожного проекту окремо. Це дозволяє використовувати різні версії однієї і тієї ж бібліотеки в різних проектах PHP. Є можливість встановити бібліотеку глобально, але це не рекомендується. Щоб встановити та використовувати бібліотеки, якими управляє Composer, вам просто потрібно оголосити їх у своєму проекті у стандартному форматі. Наприклад, якщо ви хочете встановити бібліотеку `phpmailer` за допомогою Composer, вам потрібно запустити команду «require» в кореневій папці вашого проекту. Це встановить бібліотеку `phpmailer` та її залежність у каталог «vendor» проекту. Одразу буде

створено файли `composer.json` та `composer.lock`, які будуть використовуватись для відстеження залежностей вашого проекту.

3.2 Розробка системи

З початку треба встановити додаток Openserver для можливості розгорнути локальний сервер.

Имя	Дата изменения	Тип	Размер
domains	15.06.2022 19:16	Папка с файлами	
modules	18.04.2020 16:32	Папка с файлами	
progs	18.04.2020 15:00	Папка с файлами	
userdata	07.05.2021 22:54	Папка с файлами	
Open Server	19.04.2020 21:12	Приложение	8 534 КБ

Рисунок 3.3 – Директорія з встановленим Openserver

Далі треба створити домен проекту в директорії «domains». В ньому будуть знаходитися файли сторінок зі скриптами PHP та HTML розміткою. В директорії вже створено домен за замовченням «localhost», для розробки системи було створено новий домен «passenger».

Имя	Дата изменения	Тип	Размер
localhost	15.06.2022 19:15	Папка с файлами	
passenger	19.06.2022 17:36	Папка с файлами	

Рисунок 3.4 – Директорія домену

Наступним кроком буде запуск локального серверу. Про працездатність свідчить зелений прапор у панелі управління.

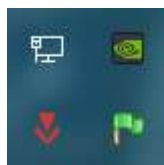


Рисунок 3.5 – Робота локального серверу

Першим кроком розробки буде створення бази даних у phpMyAdmin за розробленим планом у підпункті 3.1. Для цього потрібно відкрити вкладку «додатково» та обрати «PhpMyAdmin»

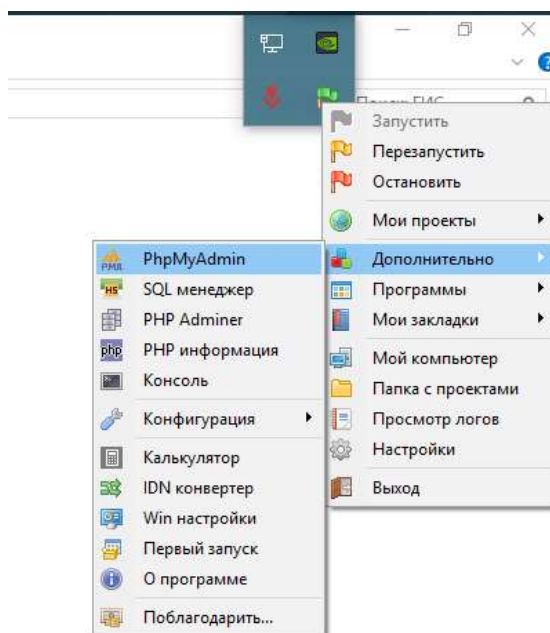


Рисунок 3.6 – Запуск phpMyAdmin

Щоб отримати доступ до управління базами даних треба ввести пароль та логін (стандартний пароль та логін – root, root). Надалі було створено базу даних для нашого домену та таблицю у базі даних.

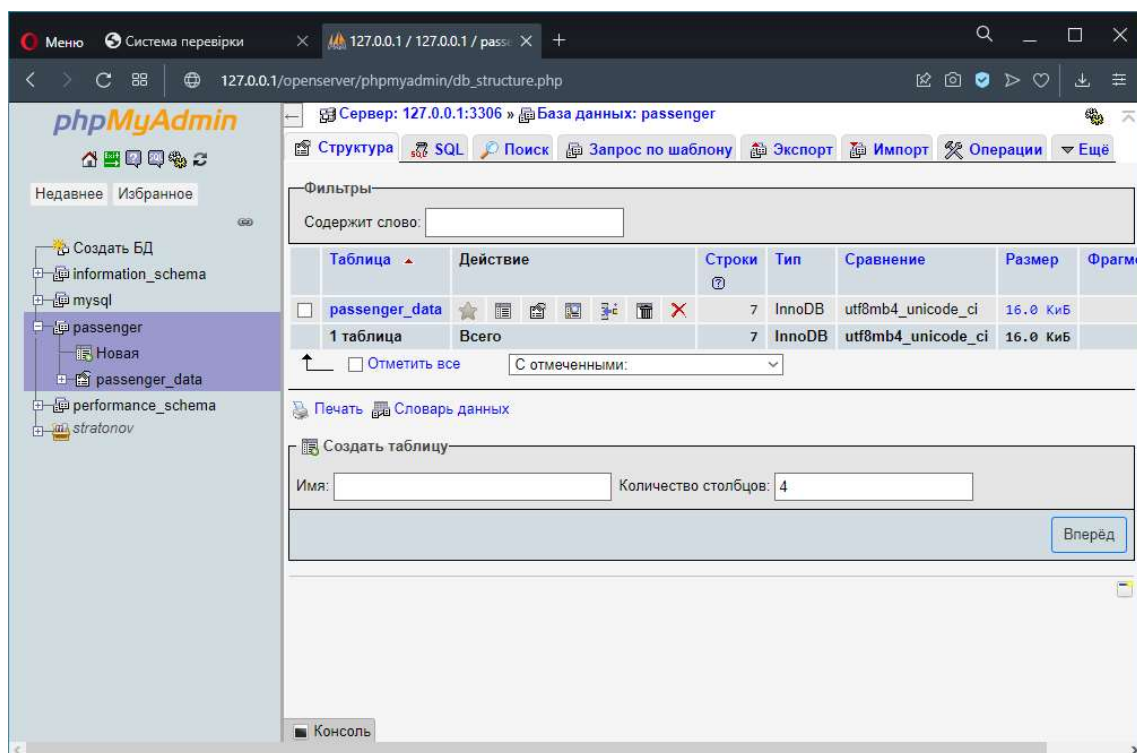


Рисунок 3.7 – база даних у phpMyAdmin

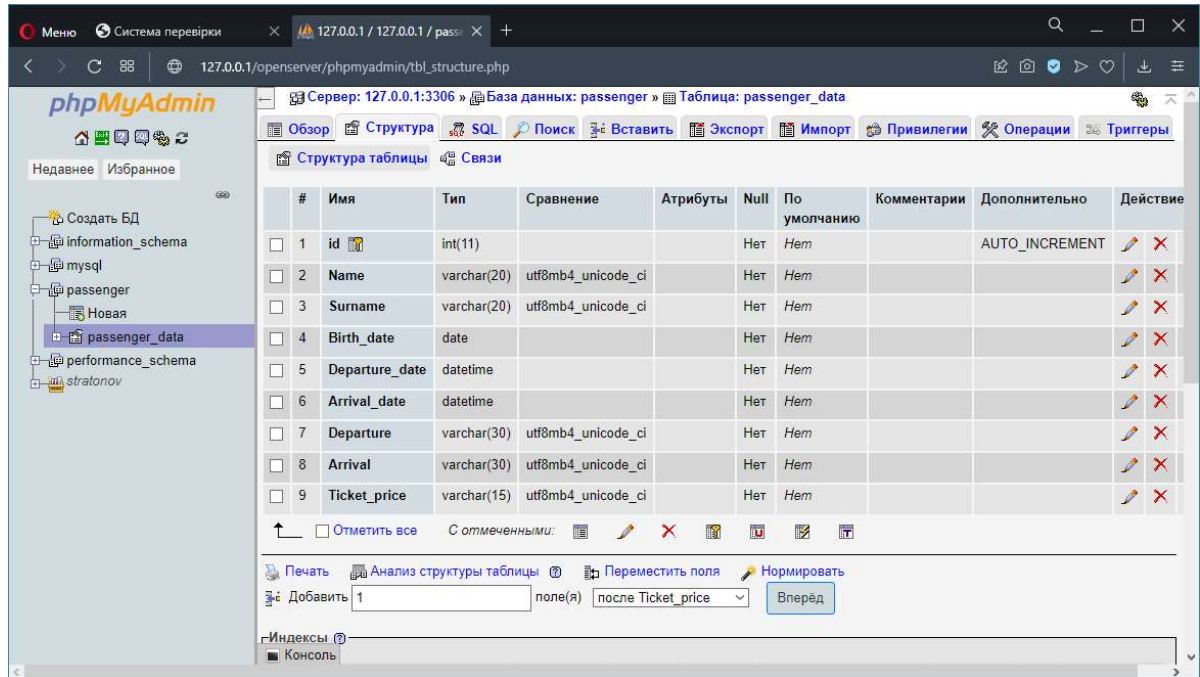


Рисунок 3.8 – структура таблиці бази даних

Щоб додати рядок до бази даних треба обрати вкладку «Вставити». Не можна заповнювати значення стовбця id, бо у ньому працює функція «AUTO_INCREMENT», тобто стовбець буде заповнено автоматично згідно порядковому номеру рядка таблиці.

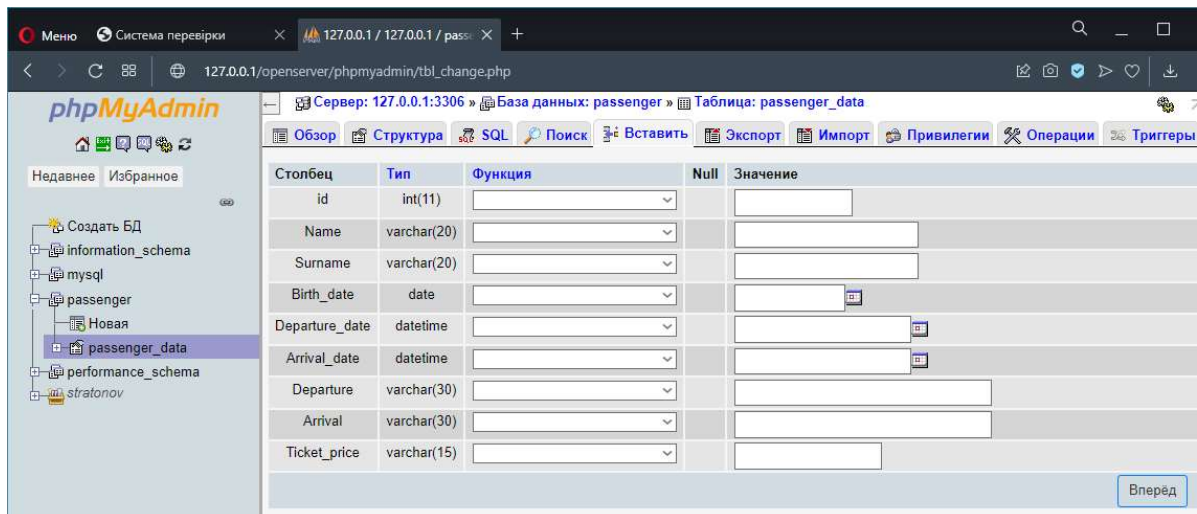


Рисунок 3.9 – форма для додавання рядку до таблиці бази даних

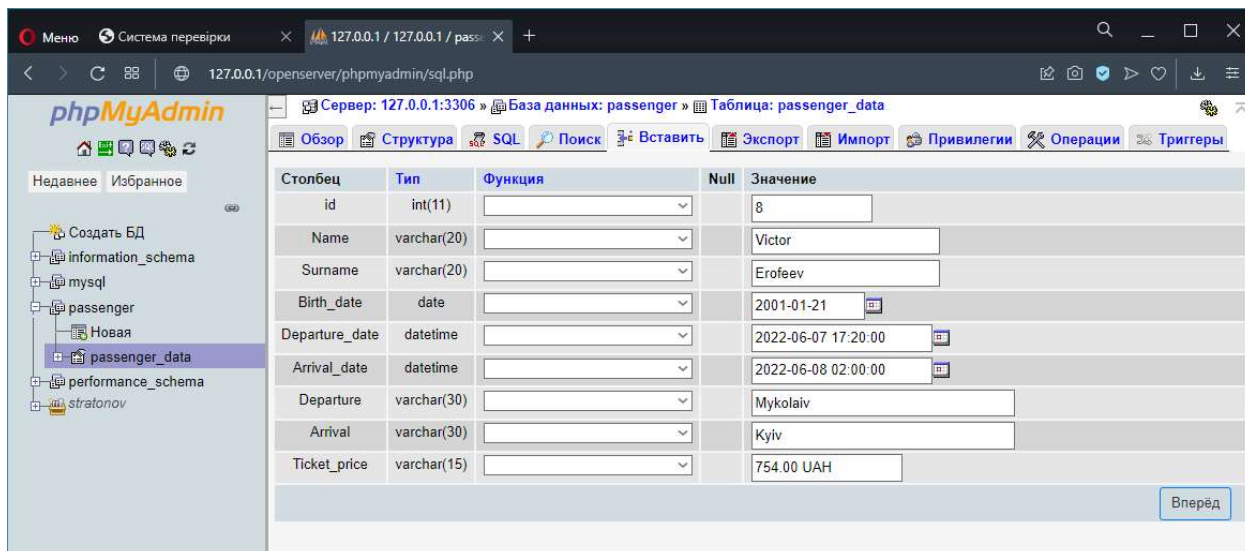


Рисунок 3.10 – заповнена форма для додавання рядку до таблиці бази даних

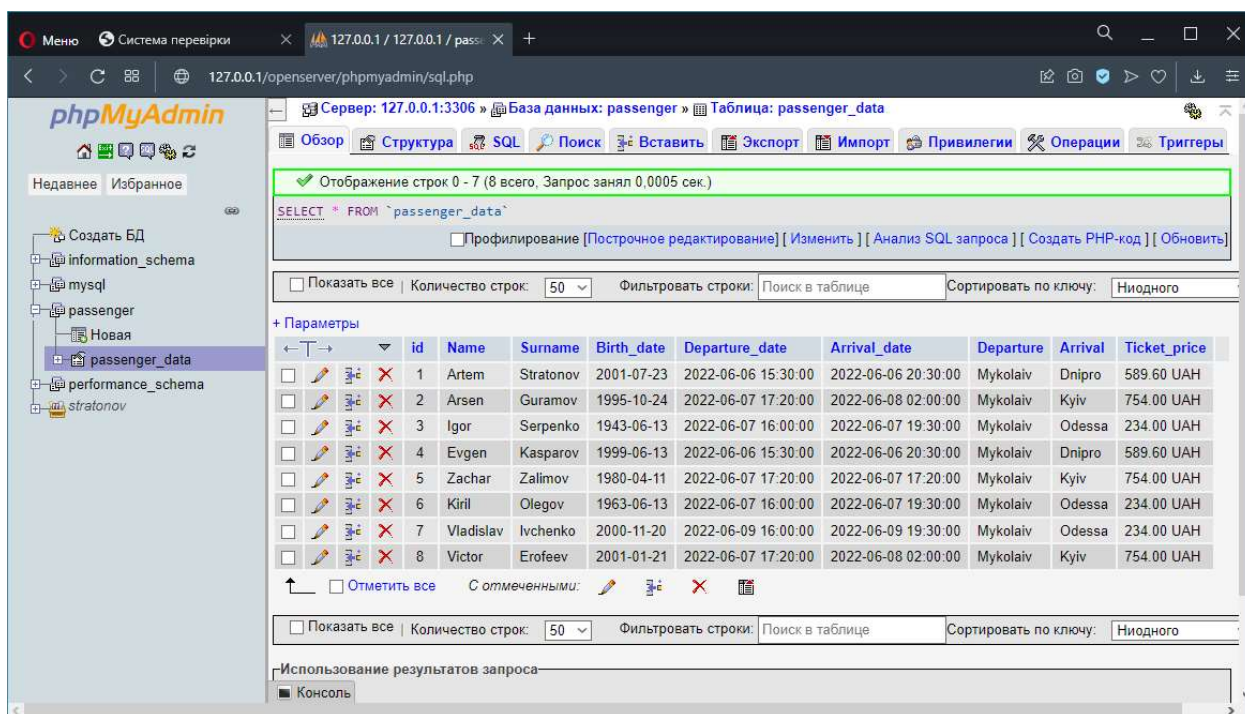


Рисунок 3.11 – заповнена таблиця бази даних

Другим етапом розробки є створення сторінок з кодом PHP та HTML, написання скриптів, підключення бібліотек, виконання запитів до створеної бази даних, та просте візуальне оформлення системи. Система буде складатися з 5 сторінок згідно з планом 3.1 підрозділу, у двох з яких буде реалізовано скрипти (тобто будуть виводити результат обробки).

create_qr_id	17.06.2022 1:24	Исходный файл ...	1 КБ
index	19.06.2022 18:00	Исходный файл ...	1 КБ
qr_creator	19.06.2022 18:00	Исходный файл ...	1 КБ
qr_id_reader	17.06.2022 19:45	Исходный файл ...	3 КБ
ticket_checker	19.06.2022 18:01	Исходный файл ...	1 КБ

Рисунок 3.12 – створені файли сторінок

Файл «index.php» є базовою сторінкою, на котрій створено 2 кнопки для переходу на окремі сторінки з виконанням скриптів створення та зчитування QR-коду.

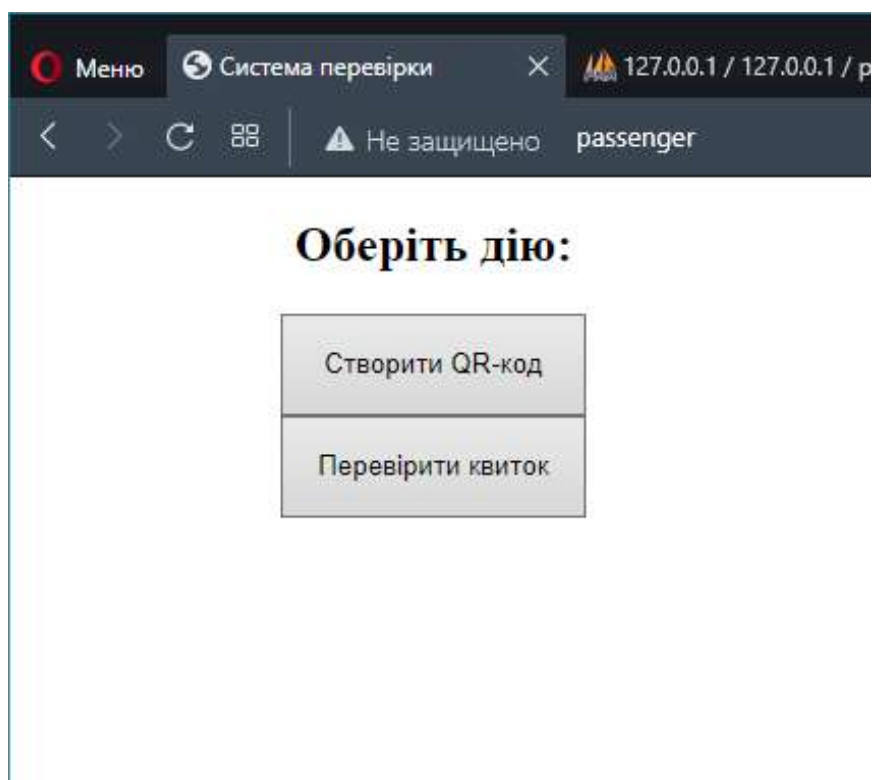
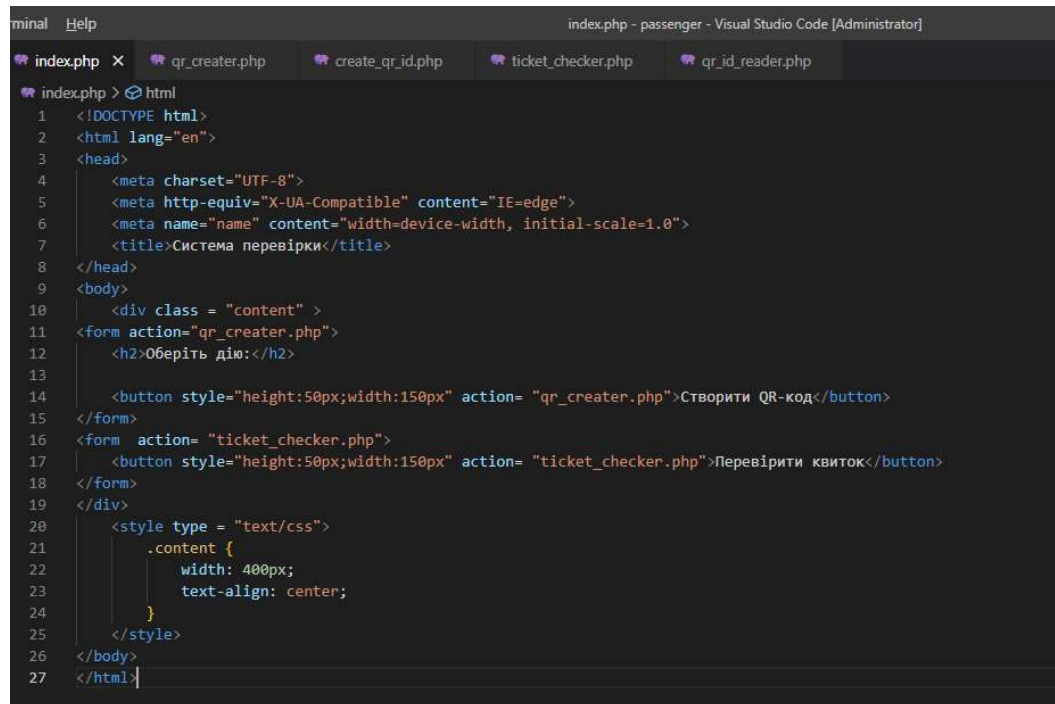


Рисунок 3.13 – Перша сторінка системи перевірки



```

index.php - passenger - Visual Studio Code [Administrator]
index.php X qr_creator.php create_qr_id.php ticket_checker.php qr_id_reader.php
index.php > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="name" content="width=device-width, initial-scale=1.0">
7   <title>Система перевірки</title>
8 </head>
9 <body>
10  <div class = "content" >
11  <form action="qr_creator.php">
12    <h2>Оберіть дію:</h2>
13
14    <button style="height:50px;width:150px" action= "qr_creator.php">Створити QR-код</button>
15  </form>
16  <form action= "ticket_checker.php">
17    <button style="height:50px;width:150px" action= "ticket_checker.php">Перевірити квиток</button>
18  </form>
19 </div>
20  <style type = "text/css">
21    .content {
22      width: 400px;
23      text-align: center;
24    }
25  </style>
26 </body>
27 </html>

```

Рисунок 3.14 – лістинг коду першої сторінки

Перша кнопка переводить до сторінки з формою, яка приймає ID строки таблиці бази даних, яка містить інформацію з квитком пасажира та передає записану інформацію у скрипт, який створює на основі неї QR-код, який можна буде перевірити. Надалі приклади виконання скриптів буде надано за першим рядком у таблиці бази даних з ідентифікатором «1».

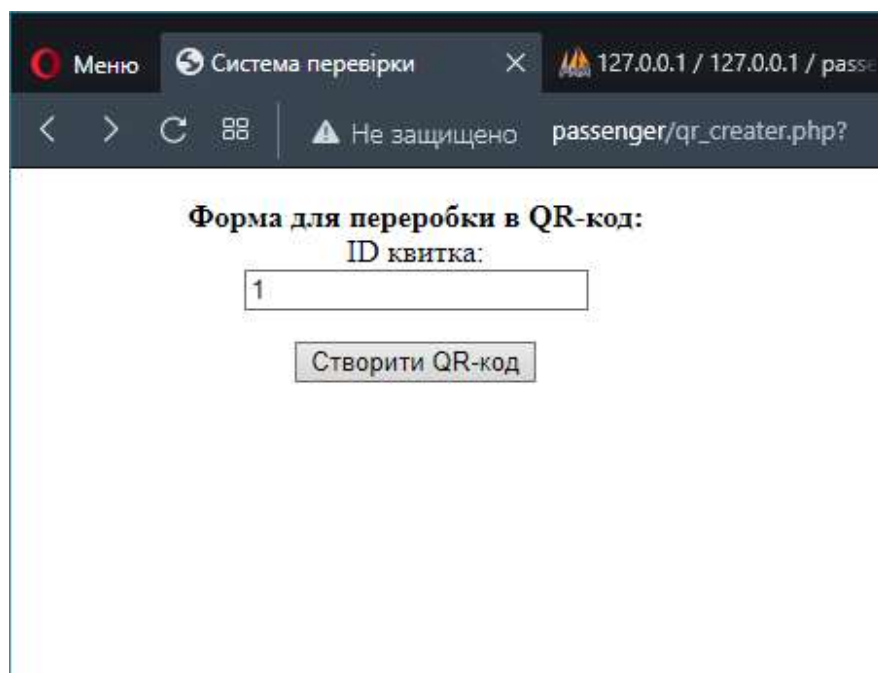


Рисунок 3.15 – сторінка створення QR-коду для квитка

```

terminal Help qr_creator.php - passenge
index.php qr_creator.php X create_qr_id.php ticket_checker.php q
qr_creator.php > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="name" content="width=device-width, initial-scale=1.0">
7   <title>Система перевірки</title>
8 </head>
9 <body>
10 <div class = "content" >
11 <form action="create_qr_id.php" method="post">
12   <p><b>Форма для переробки в QR-код:</b><br></p>
13   ID квитка: <br /> <input type="text" name="ticket_id" /><br />
14 <p></p>
15   <input type="submit" value="Створити QR-код" />
16 </form>
17 </div>
18 <style type = "text/css">
19   .content {
20     width: 400px;
21     text-align: center;
22   }
23 </style>
24 </body>
25 </html>

```

Рисунок 3.16 – лістинг коду сторінки створення QR-коду для квитка

Після натискання на кнопку «Створити QR-код» відкривається сторінка зі скриптом, котрий створює QR-код. Він додатково зберігається у заздалегідь створеній директорії «tickets» з назвою t*ID квитка*.



Рисунок 3.17 – результат виконання скрипта створення QR-коду


```

Help create_qr_id.php -
qr_creator.php create_qr_id.php X ticket_checker.php
create_qr_id.php > create_qr_id
<?php
function create_qr_id()
{
    if(isset($_POST["ticket_id"])){
        $text = $_POST["ticket_id"];
    }

    include 'phpqrcode/qrlib.php';

    $path = "tickets/t";
    $file = "$path$text.png";

    $ecc = 'L';
    $pixel_size = 10;
    $frame_size = 10;

    QRcode::png($text, $file, $ecc, $pixel_size, $frame_size);

    echo "<center><img src='".$file."></center>";
}

create_qr_id();
?>

```

Рисунок 3.18 – лістинг коду скрипта створення QR-коду

Надалі створений QR-код для квитка можна використати для перевірки квитка, котра виконується на сторінці перевірки квитка. У форму записується лише ID квитка, префікс «t» заздалегідь додано в код для додаткової зручності користувача системою.

Рисунок 3.19 – сторінка перевірки QR-коду для квитка

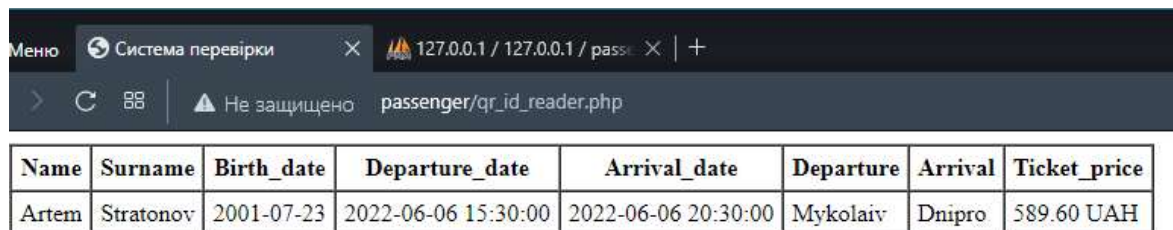
```

ticket_checker.php > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="name" content="width=device-width, initial-scale=1.0">
7   <title>Система перевірки</title>
8 </head>
9 <body>
10 <div class = "content" >
11 <form action="qr_id_reader.php" method="post">
12 <p><b>Форма для перевірки квитка:</b><br>
13 Назва файлу з QR-кодом: <br /> <input type="text" name="ticket_id" /><br />
14 <p></p>
15 <input type="submit" value="Зчитати QR-код" />
16 </form>
17
18 </div>
19 <style type = "text/css">
20   .content {
21     width: 400px;
22     text-align: center;
23   }
24 </style>
25 </body>
26 </html>

```

Рисунок 3.20 – лістинг коду сторінки перевірки QR-коду для квитка

Після натискання на кнопку «Зчитати QR-код» буде виконано скрипт зчитування QR-коду та запиту до таблиці бази даних з подальшим виводом рядку з інформацією квитка на екран.



Name	Surname	Birth_date	Departure_date	Arrival_date	Departure	Arrival	Ticket_price
Artem	Stratonov	2001-07-23	2022-06-06 15:30:00	2022-06-06 20:30:00	Mykolaiv	Dnipro	589.60 UAH

Рисунок 3.21 – результат виконання скрипта зчитування QR-коду для квитка та запиту до бази даних

Виведений рядок є в базі даних, тож пасажир можна допустити до перевезення. План реалізації було повністю виконано, система зберігає квитки у базі даних, має прості елементи для навігації, може створювати для квитка QR-код, зчитувати його та виводити на основі нього рядок з бази даних.

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Система перевірки</title>
8 </head>
9 <body>
10 <table cellpadding="5" cellspacing="0" border="1">
11 <?php
12
13 function show_ticket_id()
14 {
15     if(isset($_POST["ticket_id"])){
16         $txt = (string) $_POST["ticket_id"];
17         $path = 'D:\Study\opensever\OpenServer\domains\passenger\tickets\t';
18         $path = $path.$txt.'.png';
19     }
20
21     require __DIR__ . "/vendor/autoload.php";
22     $qrcode = new ZXing\QRReader($path);
23     $text = $qrcode->text();
24
25     return $text;
26 }
27 $decoded_qr = show_ticket_id();
28
29 $HOST = 'localhost';
30 $USER = 'root';
31 $PASSWORD = 'root';
32 $DB_NAME = 'passenger';
33
34 $link = mysqli_connect($HOST, $USER, $PASSWORD, $DB_NAME) or die(mysqli_connect_error($link));
35
36 mysqli_query($link, "SET NAMES 'utf8'");
37
38 $query = sprintf("SELECT * FROM `passenger_data` WHERE `id` = $decoded_qr ");
39
40 $result = mysqli_query($link, $query) or die(mysqli_error($link));
41 $num_rows = mysqli_num_rows($result);
42
43 mysqli_close($link);
44
45
46
47 while ($row = mysqli_fetch_assoc($result) ) {

```

Рисунок 3.22 – перша частина лістингу коду сторінки зі скриптом зчитування QR-коду для квитка та запиту до бази даних

```

terminal Help
index.php qr_creator.php create_qr_id.php ticket_d
qr_id_reader.php > html > body > table > show_ticket_id
47 while ($row = mysqli_fetch_assoc($result) ) {
48     $mass[] = $row;
49 }
50
51 foreach ($mass as $key => $value) {
52     foreach ($value as $skey => $svalue) {
53         $sarr[] = $skey;
54         $sarr = array_unique($sarr);
55     }
56 }
57 echo "<tr>";
58 for ($i=0; $i < count($sarr) ; $i++) {
59     echo "<th>".$sarr[$i]."</th>";
60 }
61 echo "</tr>";
62 foreach ($mass as $key => $value) {
63     echo "<tr>";
64     foreach ($value as $subkey => $subvalue) {
65         echo "<td>".$subvalue."</td>";
66     }
67     echo "</tr>";
68 }
69 ?>
70 </table>
71
72
73 </body>
74 </html>
75

```

Рисунок 3.23 – Друга частина лістингу коду сторінки зі скриптом зчитування QR-коду для квитка та запиту до бази даних

Висновки до розділу 3

У першому підпункті третього розділу було створено конкретний план розробки системи перевірки електронних квитків для міжміських транспортних засобів. За основу плану розробки було взято третій підпункт першого підрозділу БКР – «Постановка задачі». Методи та технології для розробки системи було обрано на основі висновків з другого розділу. В результаті виконання створеного плану було розроблено систему, яка складається з бази даних, в котрій знаходиться таблиця з даними квитків, 5 сторінок візуальної оболонки системи на основі розмітки HTML та скриптової мови програмування PHP. В системі реалізовано створення QR-коду з квитка та зчитування створеного QR-коду з подальшим виводом на сторінку закодованого в QR-коді електронного квитка, що являє собою процес перевірки квитка.

Спеціальний розділ
ОХОРОНА ПРАЦІ
до кваліфікаційної роботи

на тему:

«Система перевірки електронних проїзних документів у міжміських транспортних засобах»

Спеціальність 122 «Комп'ютерні науки»

122 – БКР – 401.21810123

Виконав студент 4-го курсу, групи 401

А.О.Стратонов

(підпис, ініціали та прізвище)

«__» _____ 2022 р.

Консультант ст. викладач

(наук. ступінь, вчене звання)

О. В. Макарова

(підпис, ініціали та прізвище)

«__» _____ 2022 р.

Миколаїв – 2022

4 ОХОРОНА ПРАЦІ

Характерною ознакою сучасного науково-технічного прогресу практично у всіх сферах діяльності людини є широке застосування комп'ютерних технологій, заснованих на використанні електронно обчислювальних машин (ЕОМ). Сьогодні, а тим більше, майбутнє, вже важко уявити без комп'ютерів та іншої електронної техніки. Адже саме завдяки їм стала можливою швидка переробка величезних обсягів інформації, проведення необхідних розрахунків, виконання різних видів робіт, пов'язаних обробкою текстових та ілюстраційних зображень, організація оперативного отримання та передачі інформації, збереження її значних обсягів електронним способом. Стрімке впровадження комп'ютерів не тільки в сфері управління виробництвом, але також на транспорті, в банківській системі бізнесу, системі освіти, галузі теле- та радіокомунікацій, у видавничо-поліграфічному комплексі, сфері обслуговування призвело до того, що десятки мільйонів людей у всьому світі виявились втягнутими у взаємодію людини з комп'ютером. Природно виникає запитання: настільки безпечною є ця взаємодія для людини? Адже відома аксіома про те, що будьяка взаємодія людини та засобів праці двостороння. Людина впливає на удосконалення засобів праці, а останні – на працюючу людину. Отже, навіть сучасні технології та техніка, до яких безперечно, належать комп'ютерні технології та ЕОМ несуть у собі певні потенційні небезпеки. У зв'язку з цим набуває актуальності вивчення, фізіологічних, психологічних, соціальних та виробничих наслідків взаємодії системи «людина – комп'ютер» та розробка й упровадження заходів щодо нормалізації праці та збереження здоров'я працівників на комп'ютеризованих робочих місцях. Метою спеціальної частини є створення безпечних і здорових умов праці на робочому місці або у виробничому приміщенні при розробці системи.

Відповідно до мети спеціальної частини, виділено наступні завдання:

- Проаналізувати наявні умови праці в робочому приміщенні;

- Визначити достатній рівень показників для робочого приміщення, де проводяться роботи з розробки системи;
- Визначити основні правила техніки безпеки при роботі з комп'ютерною технікою.

4.1 Комп'ютер та його апаратне забезпечення

Відповідно до ДСТУ 2938-94 «Системи обробки інформації. Основні поняття. Терміни та визначення» комп'ютер – це функціональний пристрій, що складається з одного чи кількох взаємопов'язаних центральних процесорів і периферійних пристроїв і може виконувати обчислення без участі людини. Основними функціями комп'ютера є введення та виведення інформації, її зберігання та обробка. В якості пристроїв введення часто використовуються клавіатура та сканер, який забезпечує більшу швидкодію. У зв'язку з бурхливим розвитком комп'ютерної техніки щороку зростає спектр їх різновидів. За призначенням комп'ютери можна умовно поділити на:

— побутові комп'ютери – власне ПК, що призначені для індивідуальної роботи в домашніх умовах;

— навчальні комп'ютери – призначені для використання в системі освіти, як вищої так і середньої. Основні вимоги таких комп'ютерів – надійність, достатня потужність та невисока ціна (можливість придбання більшої кількості однотипних комп'ютерів закладами освіти). Користувачі – учні, студенти, викладачі та ін.;

— професійні комп'ютери – робочі станції для роботи на виробництві, в офісах установ, які, як правило, об'єднані в локальну комп'ютерну мережу. Від «побутових» відрізняються більш високими показниками за всіма параметрами – продуктивністю, функціональними можливостями, якістю зображення на дисплеї та ін. Користувачі – службовці, оператори ВДТ;

— сервери – потужні комп'ютери, призначені для локальних та глобальних мереж. Вони виконують функції керування робочими станціями, зберігання значних масивів інформації та ін. Користувачі – менеджери, і

адміністратори локальних комп'ютерних мереж, системні та прикладні програмісти для підтримки програмного забезпечення тощо;

— графічні станції – використовуються для роботи з графічними зображеннями, відео та анімацією. Володіють надзвичайно високими ресурсами за всіма основними параметрами.

На сьогодні найбільш розповсюдженими є персональні комп'ютери. В мінімальний базовий комплект ПК входять наступні блоки чи компоненти:

— системний блок, в якому зосереджені життєво важливі елементи комп'ютера;

— дисплей (монітор), який призначений для виведення (відображення) інформації;

— клавіатура, яка призначена для введення інформації в комп'ютер;

— графічний маніпулятор «миша», який слугує для керування роботою програм шляхом вибору різних пунктів меню, виділення та «перетягування» об'єктів.

Системний блок комп'ютера з точки зору охорони праці не несе особливої небезпеки для життя та здоров'я користувача. Найбільшу небезпеку несе підвищене значення напруги, що подається з електромережі на блок живлення системного блоку. Несприятливий вплив на користувача може здійснювати шум, що створюється при роботі вентиляторів та накопичувачів системного блоку. Більшість електронних компонентів комп'ютера живляться напругою «+5 В», двигуни накопичувачів «+12 В», деякі пристрої – «-5 В» та «-12 В». Такі значення напруги не несуть небезпеки для людини. В той же час на вхід блоку живлення подається напруга електромережі (220 В), яка є небезпечною для людини, з точки зору її можливого ураження. Тому до блоку живлення висувається низка вимог електробезпеки. Зокрема, електропроводи та кабелі повинні мати надійну ізоляцію, а на випадок короткого замикання або інших аварійних режимів в електричній схемі блоку живлення повинні

бути передбачені елементи захисту. Сам же він знаходиться в корпусі, який перекриває доступ до струмоведучих елементів блоку.

Дисплей є одним з основних блоків комп'ютера (і одним із найдорожчих), що слугує для візуалізації інформації. Від його характеристик в значній мірі залежить працездатність та стан здоров'я користувача комп'ютера. Залежно від призначення та сфери застосування дисплеї поділяють на такі групи:

— А – кольорові дисплеї, які використовуються для демонстрації наочних засобів у навчальному процесі, встановлюються на тренажерах, гральних автоматах, тощо;

— Б – кольорові дисплеї для персональної роботи користувачів у навчальному процесі та на виробництві, де немає потреби у постійній напруженій зоровій роботі;

— В – кольорові дисплеї для професійної роботи з текстовими документами і насиченими графічними зображеннями;

— Г – монохромні дисплеї для комплектування шкільних комплексів навчальної обчислювальної техніки, професійна обробка текстів та ін.

Однією із основних характеристик дисплею є частота регенерації екрану, тобто число «картинок», які дисплей змінює за секунду. Коли частота регенерації екрану становить 70 – 75 Гц, більшість людей не помічає подразнюючого мерехтіння, а починаючи з 85 Гц неприємні відчуття не виникають навіть у найчутливіших до цього показника користувачів. Іншою важливою характеристикою дисплею є чіткість зображення, яка визначається роздільною здатністю, тобто кількістю пікселів по горизонталі та вертикалі. Ще однією характеристикою дисплея є розмір екрана. Як правило, наводять розмір його діагоналі в дюймах. Найбільш розповсюдженими є дисплеї з розміром екрану 14 дюймів. Однак для професійного використання графічних пакетів та настільних видавничих систем застосовуються дисплеї з екраном 17 і більше дюймів.

Останньою важливою характеристикою дисплея є частота модуляції інтенсивності електронного променя. По суті, це максимальна частота сигналів, що керують яскравістю пікселів. З одного боку, все більші значення роздільної здатності вимагають збільшення частоти модуляції інтенсивності електронного променя. З іншого, підвищення частоти модуляції призведе до того, що навіть короткі провідники перетворяться на антени, посилюючи шкідливий вплив електромагнітного та електростатичного полів на користувача. Компроміс може бути досягнуто лише тоді, коли будуть розроблені надійні засоби захисту, що гарантують відносну безпеку користувачеві. Дисплеї, сконструйовані на основі ЕПТ, є джерелом кількох видів електромагнітного випромінювання та полів, а саме:

- іонізуючого (рентгенівського) випромінювання;
- оптичного випромінювання;
- електромагнітних випромінювань та полів.

Реальна інтенсивність, напруженість, рівень та інші параметри кожного виду електромагнітного випромінювання залежить від технічної конструкції конкретного дисплея, режимів його роботи, екранування та інших факторів. Найсерйознішої уваги заслуговують рідиннокристалічні (РК) та плазмові дисплеї, з огляду на їх значно менший несприятливий вплив на користувача порівняно з дисплеями на основі ЕПТ.

Основними функціями комп'ютера є введення та виведення інформації, її зберігання та обробка. В якості пристроїв введення часто використовуються клавіатура та сканер, який забезпечує більшу швидкодію. У зв'язку з бурхливим розвитком комп'ютерної техніки щороку зростає спектр їх різновидів. За призначенням комп'ютери можна умовно поділити на:

- побутові комп'ютери – власне ПК, що призначені для індивідуальної роботи в домашніх умовах;
- навчальні комп'ютери – призначені для використання в системі освіти, як вищої так і середньої. Основні вимоги таких комп'ютерів –

надійність, достатня потужність та невисока ціна (можливість придбання більшої кількості однотипних комп'ютерів закладами освіти). Користувачі – учні, студенти, викладачі та ін.;

— професійні комп'ютери – робочі станції для роботи на виробництві, в офісах установ, які, як правило, об'єднані в локальну комп'ютерну мережу. Від «побутових» відрізняються більш високими показниками за всіма параметрами – продуктивністю, функціональними можливостями, якістю зображення на дисплеї та ін. Користувачі – службовці, оператори ВДТ;

— сервери – потужні комп'ютери, призначені для локальних та глобальних мереж. Вони виконують функції керування робочими станціями, зберігання значних масивів інформації та ін. Користувачі – менеджери, і адміністратори локальних комп'ютерних мереж, системні та прикладні програмісти для підтримки програмного забезпечення тощо;

— графічні станції – використовуються для роботи з графічними зображеннями, відео та анімацією. Володіють надзвичайно високими ресурсами за всіма основними параметрами. На сьогодні найбільш розповсюдженими є персональні комп'ютери.

В мінімальний базовий комплект ПК входять наступні блоки чи компоненти:

— системний блок, в якому зосереджені найважливіші елементи комп'ютера;

— дисплей (монітор), який призначений для виведення (відображення) інформації;

— клавіатура, яка призначена для введення інформації в комп'ютер;

— графічний маніпулятор «миша», який слугує для керування роботою програм шляхом вибору різних пунктів меню, виділення та «перетягування» об'єктів.

Системний блок комп'ютера з точки зору охорони праці не несе особливої небезпеки для життя та здоров'я користувача. Найбільшу небезпеку

несе підвищене значення напруги, що подається з електромережі на блок живлення системного блоку. Несприятливий вплив на користувача може здійснювати шум, що створюється при роботі вентиляторів та накопичувачів системного блоку.

Клавіатура є універсальним стандартним пристроєм введення інформації, що дозволяє передавати комп'ютеру певні символи та команди. Час використання клавіатури користувачем особливо при введенні інформації та у діалоговому режимі роботи, досить значний. Тому від її ергономічного виконання та зручності роботи значною мірою залежать навантаження, що припадають, в основному, на кисті руш користувача. В конструкції клавіатури передбачено чимало рішень, які полегшують роботу на ній. Виконання клавіатури у вигляді окремого елемента комп'ютера дає можливість встановлювати її у зручному для користувача місці, а регульовані опори дозволяють змінювати кут її нахилу конструктивною особливістю клавіатури є також і те, що на ній виділено (кольором та місцем розташування) окремі групи чи «функціональної зони» клавіш. Окрім того, кілька клавіш, що найчастіше використовуються продубльовані з обох сторін клавіатури. Посередині кожної клавіші виконано заглиблення, що за формою відтворює поверхню пучки пальця. Використання, так званих, «м'яких» клавіатур забезпечує невеликі зусилля натискання на клавіші (0,25 – 0,5 Н). Практично всі клавіатури введення даних в світі побудовані на тому, що натиснення клавіші визначається шляхом сканування матриці клавіатури. При такому принципі роботи клавіатура випромінює електромагнітні поля низьких рівнів. Проте велика тривалість використання клавіатури протягом робочої зміни та невелика відстань нервових закінчень пальців користувача від джерел електромагнітного випромінювання – контактів клавіш, можуть бути причиною несприятливого впливу на його здоров'я. Пристрій типу «миша» або графічний маніпулятор в зв'язку з масовим впровадженням графічної операційної оболонки Windows фактично набув статусу обов'язкового

елемента стандартного комплексу комп'ютера. Використання різноманітних пристроїв на основі кульового покажчика дає можливість уникнути значних навантажень, які припадають, в основному, на плечевий та ліктьовий суглоби руки та на зап'ясток кисті. Вибір графічного маніпулятора залежить не лише від розмірів руки та стилю роботи, а також і від уподобань користувача.

4.2 Виробниче середовище

Вагомий вплив на працездатність та здоров'я користувачів комп'ютерів здійснює виробниче середовище. Це середовище у виробничих приміщеннях (офісах), в основному, визначається мікрокліматом, освітленням, наявністю шкідливих речовин у повітрі, рівнем шуму, випромінювання. Під виробничим мікрокліматом розуміють стан повітряного середовища виробничого приміщення, який визначається температурою, відносною вологістю, рухом повітря та тепловим випромінюванням нагрітих поверхонь, що в сукупності впливають на тепловий стан організму людини.

В процесі трудової діяльності людина перебуває у постійній тепловій взаємодії з виробничим середовищем. Кількість тепла, що утворюється в організмі, залежить від фізичного навантаження працівника, а рівень тепловіддачі – від мікрокліматичних умов виробничого приміщення. Оскільки робота за комп'ютером характеризується малими фізичними навантаженнями, то цей вид діяльності належить до категорії легких робіт за критерієм енерговитрат організму. Порушення теплового балансу може призвести до перегрівання або ж переохолодження організму людини і, з рештою, до захворювання. Віддача тепла організмом людини здійснюється, в основному, за рахунок випромінювання і випаровування вологи з поверхні шкіри. Чим нижча температура повітря і швидкість його руху, тим більше тепла віддається випромінюванням. При високій температурі значна частина тепла втрачається випаровуванням поту.

Вологість повітря істотно впливає на віддачу тепла випаровуванням. Через високу вологість випаровування погіршується і віддача тепла

зменшується. Зниження вологості покращує процес тепловіддачі випаровуванням. Однак, надто низька вологість викликає висихання слизових оболонок, їх пересихання та розтріскування, забруднений хвороботворними мікробами. Рухомість повітря визначає рівень тепловіддачі з поверхні шкіри конвекцією і випаровуванням. Різкі коливання температури в приміщенні, яке продувається холодним повітрям (перетягом) значно порушуючи терморегуляцію організму і можуть викликати простудні захворювання.

Таким чином, для нормального теплового самопочуття людини важливо забезпечити певне співвідношення температури, відносної вологості та швидкості руху повітря, тобто певні мікрокліматичні умови. Такі умови визначаються, в основному, категорією роботи, що виконується, та періодом року і можуть бути оптимальними та допустимими. Під оптимальними мікрокліматичними умовами розуміють такі співвідношення його параметрів, при якому в умовах тривалої та систематичної дії на людину створюються комфортні теплові відчуття та відбувається збереження нормального теплового стану організму без напруження механізмів терморегуляції. Допустимі мікрокліматичні умови передбачають можливість виникнення дискомфортних відчуттів та зміни теплового стану організму, однак вони швидко минають і нормалізуються за рахунок напруження механізмів терморегуляції в межах фізіологічних пристосувальних можливостей.

Вивчаючи ергономічні вимоги до робочих місць користувачів комп'ютерів, слід мати на увазі, що площа, виділена на одне робоче місце з відеотерміналом або ПК, має бути не менше 6 кв.м, а об'єм – не менше 20 куб.м. Площа, виділена для одного робочого місця без відеотерміналу або ПК, має бути не менше 4,5 кв.м, а об'єм – не менше 15 куб.м.

Приміщення для відпочинку, прийому їжі, психологічного розвантаження й інші побутові приміщення повинні мати площу й об'єм з розрахунку на одну людину: площа – 4,5 кв.м і об'єм – 15 куб.м на одну

людину з урахуванням максимальної кількості осіб, які одночасно працюють у зміні. Розташовуючи робочі місця з відеотерміналами і ПК, необхідно дотримуватися таких вимог:

- робочі місця з відеотерміналами і ПЕОМ розташовуються на відстані не менше 1 м від стін зі світловими отворами;
- відстань між бічними поверхнями відеотерміналів має бути не менше 1,2 м;
- відстань між тильною поверхнею одного відеотерміналу й екраном іншого має бути не менше 2,5 м;
- прохід між рядами робочих місць має бути не менше 1 м.

Доволі важливим фактором впливу на здоров'я та працездатність людини є освітленість приміщення. Для визначення того, чи є достатнім забезпечення природного освітлення в приміщенні порівнюють значення фактичної загальної площі вікон в приміщенні до мінімальної [23][24]. Мінімальна площа вікон в приміщенні визначається за формулою 4.1:

$$S_e = \frac{e_N \cdot k_z \cdot \eta_e \cdot S_{\text{підл}} \cdot k_{\text{бюд}}}{\tau_{\text{заг}} \cdot r_1 \cdot 100} \quad (4.1)$$

де e_N - нормований коефіцієнт природного освітлення для розглянутих умов праці,

k_z - коефіцієнт запасу, що приймається при розрахунках природного освітлення,

η_e - світлова характеристика вікна,

$S_{\text{підл}}$ - площа підлоги виробничого приміщення ,

$\tau_{\text{заг}}$ - загальний коефіцієнт пропускання світла,

r_1 - коефіцієнт, що враховує підвищення коефіцієнта природного освітлення за рахунок світла, яке відбивається від внутрішніх поверхонь приміщення,

$K_{б\text{уд}}$ - коефіцієнт, що враховує вплив протилежної будівлі на освітленість у виробничому приміщенні.

Вхідні дані для розрахунку мінімальної площі вікна для забезпечення природної освітленості наведені в таблиці 4.1.

Таблиця 4.1 – Вхідні дані для розрахунку необхідної площі вікна.

№	Параметр	Позначення	Розмірність	Значення
1	Нормований коефіцієнт природного освітлення	e_N	%	0,9
2	Коефіцієнт запасу	κ_3	1	1,2
3	Світлова характеристика вікна	η_e	1	9.87
4	Площа підлоги	$S_{\text{підл}}$	м ²	8.23
5	Загальний коефіцієнт пропускання світла	$\tau_{\text{заг}}$	1	2,4
6	Коефіцієнт, що враховує підвищення коефіцієнта природного освітлення	r_1	1	0,621
7	Коефіцієнт, що враховує вплив протилежної будівлі на освітленість	$K_{б\text{уд}}$	м	1.01

Визначення площі вікна, яке наявне в приміщенні за формулою 4.1:

$$S_{\text{нв}} = h * l = 1.6 * 1 = 1.6(\text{м}^2) \quad (4.1)$$

Наступним кроком необхідно обчислити значення площі вікна для забезпечення природної освітленості приміщення за формулою 1.1:

$$S_{\epsilon} = \frac{0.9 \cdot 1,2 \cdot 9,87 \cdot 8,32 \cdot 1.01}{2,4 \cdot 0,621 \cdot 100} = 0,6(\text{м}^2)$$

Згідно попередніх обчислень, можна зробити висновки, що того вікна, яке наявне в приміщенні більш ніж достатньо для забезпечення природного освітлення в робочому приміщенні для робіт за комп'ютером.

4.3 Причини виробничого травматизму та захворювань

Постійний і різнобічний аналіз травматизму і профзахворювань розглядається як одна з головних функцій керування безпекою праці та прийняття основних заходів щодо усунення причин травматизму та захворювань.

Технічні причини, котрі можна охарактеризувати як причини, що залежать від рівня організації праці на виробництві, а саме: недосконалий технологічний процес, конструктивні недоліки обладнання, інструментів та пристосувань, недостатня механізація важких робіт; недосконале огороження, відсутність спеціальних захисних засобів, засобів сигналізації та блокувань, недостатня міцність та надійність машин, шкідливі властивості оброблюваного матеріалу тощо. Ці причини інколи називають конструктивними або інженерними.

Організаційні причини, що повністю залежать від рівня організації праці на виробництві. До них можна віднести: незадовільний стан території, проїздів, проходів, порушення правил експлуатації обладнання, транспортних засобів, порушення технологічного регламенту, порушення правил і норм при транспортуванні, складанні і зберіганні матеріалів і деталей; порушення норм і правил при плановому технічному обслуговуванні та ремонті обладнання, транспортних засобів і інструменту; недоліки при навчанні робітників безпечним методам праці; недостатній технічний нагляд за небезпечними роботами; використання машин, механізмів і інструменту не за призначенням; відсутність або незадовільне огороження робочої зони; відсутність або невикористання засобів індивідуального захисту тощо.

Санітарно-гігієнічні причини, до котрих можна віднести: перевищення (відносно) запиленості та загазованості повітря робочої зони; відсутність або недостатнє природне освітлення, підвищену пульсацію світлового потоку; підвищений рівень шуму та вібрації, інфразвукових та ультразвукових коливань на робочому місці; підвищений рівень ультразвукової та інфрачервоної радіації тощо. Психофізіологічні причини, до котрих відносяться фізичні, нервово-психічні перевантаження працюючих.

Психофізіологічні причини – грубі помилки в діях, пов'язані з фізіологічним (втомленість), психічним (підвищена дратливість) або хворобливим станом працівників. Найбільш частими конкретними причинами виробничого травматизму на виробничих підприємствах є: відсутність інструкцій з охорони праці; робота на несправному обладнанні або на обладнанні без засобів захисту; відсутність засобів проти випадкового ураження працівників електричним струмом; відсутність драбин, які б відповідали вимогам правил техніки безпеки; розвантаження і транспортування вантажів без застосування відповідних механізмів і пристосувань; користування несправним реманентом, пристосуванням та інструментом.

Людина може припускатися помилок у своїх діях внаслідок фізичного, статичного або динамічного перевантаження, розумового перенапруження, перенапруження аналізаторів (зорового, слухового, тактильного), монотонності праці, стресових ситуацій, хворобливого стану. Травму може викликати незадовільність анатомо-фізіологічних і психічних особливостей організму людини залежно від характеру виконуваної роботи. У сучасних складних технічних системах управління, в конструкціях машин, приладів і систем управління ще недостатньо враховуються фізіологічні і антропологічні особливості і можливості людини.

Незадовільна організація праці зумовлює надмірні фізичні і нервові перевантаження, що прискорює стомлюваність робітників. У такому стані

знижується чутливість до різних подразників виробничого середовища, притуплюється увага, пильність. Це призводить до того, що ближче до кінця робочої зміни різко підвищується кількість нещасних випадків, причинами яких є помилкові дії потерпілих.

Важливе значення серед факторів, які зумовлюють виробничий травматизм, мають попередні нещасні випадки, психофізіологічний стан потерпілих. При цьому несприятливий психофізіологічний стан може бути пов'язаний як з об'єктивними причинами (погана організація праці), так і суб'єктивними, залежними від особливостей особистого стану потерпілих (необережність, поспіх, втома, роздратування, ризик тощо).

Особисті якості працівників (швидкість реакції, активність, відповідальність, дисциплінованість і т. ін.) також впливають на їх схильність до нещасних випадків. Це вказує, що особисті якості потерпілих значно впливають на той факт, що в схожих екстремальних ситуаціях одні стають жертвами нещасних випадків, а другі – ні.

Висновки до розділу 4

Під час дослідження умов праці було виявлено, як важлива ця сфера для здоров'я та самопочуття людини, встановлено норми приміщень для роботи людини за комп'ютером. Вагомий вплив на працездатність та здоров'я користувачів комп'ютерів здійснює виробниче середовище. Це середовище у виробничих приміщеннях (офісах), в основному, визначається мікрокліматом, освітленням, наявністю шкідливих речовин у повітрі, рівнем шуму, випромінювання. Було зазначено що для безпеки праці також треба приміщення для відпочинку, прийому їжі, психологічного розвантаження та інші побутові приміщення. Адже це складає дуже важливий комфорт для працівників.

Було досліджено комп'ютер та його комплектуючі, а також їх окремий вплив на здоров'я людини. Якщо окремо фактори ризику для кожної

комплектуючої і дуже малі, разом у зібраному комп'ютері загроза стає досить серйозною. У цьому є причина дотримуватися принципів охорони праці.

Було зазначено класифікацію причин захворювань та травматизму на робочому місці. Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці.

ВИСНОВКИ

В ході виконання дослідження теоретичної частини бакалаврської кваліфікаційної роботи було розкрито причини появи та розповсюдження електронних квитків. У першому підпункті першого розділу було зазначено фактори переходу до діджиталізованого аналогу паперових квитків та їх переваги над застарілим варіантом. Надалі було розглянуто найпопулярніші варіанти реалізації електронних квитків в Україні та в світі. Результатом першого розділу було поставлено задачі для створення системи перевірки електронних проїзних документів.

В наступному розділі було представлено наявні моделі, методи та інформаційні технології для вирішення поставленої задачі. Було розглянуто два варіанти розгортання бази даних для системи: базу даних у хмарі та реляційну базу даних MySQL. Варіант з базою даних у хмарі більше підходить для великих комерційних проєктів з великою кількістю користувачів, тому для розробки системи було обрано розгортання бази даних на локальному сервері на базі Openserver з підключеним інструментом для адміністрування баз даних PhpMyAdmin. Для реалізації процесу перевірки електронних квитків було обрано технологію QR-код, яка виконує функцію зберігання в растровому зображенні посилання на елемент таблиці бази даних квитків.

В останньому розділі пояснювальної записки кваліфікаційної роботи було використано здобуті знання для створення плану розробки системи. За цим планом було створено працюючу систему на основі сторінок HTML з включеними скриптами на мові програмування PHP, котрі реалізують можливість створення QR-коду з посиланням на квитки у таблиці бази даних для перевірки та виводу на сторінку елемента таблиці, котрим є електронний квиток.

Таким чином в ході виконання завдання БКР було виконано мету, яка полягла у розробці системи перевірки електронних квитків для міжміських

транспортних засобів. Для досягнення вказаної мети було виконано наступні основні завдання:

- проведено аналіз способів перевірки електронних проїзних документів;
- обрано реляційну систему бази даних для зберігання даних у системі;
- обрано спосіб перевірки електронних проїзних документів, а саме способом зчитування QR-коду;
- розроблено повністю функціонуючу систему перевірки електронних проїзних документів.

. В ході виконання завдання БКР було закріплено навички роботи з базами даних, отримані в ході вивчення дисципліни «Організація баз даних» та навички програмування на скриптовій мові програмування PHP, отримані в ході вивчення дисципліни «Web-програмування та система управління контентом, веб-дизайн». Було засвоєно навик підключення та використання бібліотек PHP.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Factors Influencing the Adoption of E-Ticketing System in the Bus Transport Sector in Ghana Aug. 10, 2020. (с. 163-167)
2. ДССУ, Транспорт і зв'язок України 2018, Статистичний збірник. (с. 57)
3. Wikipedia: веб-сайт. URL: <https://en.wikipedia.org> (дата звернення: 04.05.2022-16.06.2022).
4. The New York Times: веб-сайт. URL: <https://www.nytimes.com/1997/08/10/travel/e-tickets-begin-to-catch-on.html#:~:text=Major%20airlines%20first%20offered%20electronic,Southwest%20Airlines%20leading%20the%20way> (дата звернення: 10.05.2022).
5. THE USE OF ELECTRONIC TICKETING - A CASE STUDY - by Olav C. Unger June 1996 (с. 31-35)
6. PopularitY of Programming Language – URL: <https://pypl.github.io/PYPL.html>
7. How does QR code technology work - URL: <https://digitash.com/technology/how-does-qr-code-technology-work/>(дата звернення: 15.06.2022).
8. Бібліотека розшифрування QR-коду - URL: <https://github.com/khanamiryan/php-qr-code-detector-decoder> (дата звернення: 10.06.2022).
9. The Beginnings of American Railroads and Mapping - URL: <https://www.loc.gov/collections/railroad-maps-1828-to-1900/articles-and-essays/history-of-railroads-and-maps/the-beginnings-of-american-railroads-and-mapping/> (дата звернення: 10.05.2022).
10. В.Я. Яскілка, старший викладач, М.З. Олійник, асистент. Конспект лекцій з курсу «ОХОРОНА ПРАЦІ В ГАЛУЗІ» – с. 31-33
11. Г.В. Пронюк, Т.Є. Стиценко, Н.М. Сердюк. Методичні вказівки до самостійної роботи з дисципліни «БЕЗПЕКА ПРАЦІ В ІНДУСТРІЇ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ» – с. 15
12. «ВДОСКОНАЛЕННЯ ОХОРОНИ ПРАЦІ В ІТ-ІНДУСТРІЇ» – с. 10

13. Левон М.С. Система факторов, влияющих на производительность труда в it-фирмах // Экономика, Статистика и Информатика. – 2015. – №3. С.127-128.
14. Разумний О.В. Тези доповіді «Порівняльний аналіз якості засвоєння лекційного матеріалу студентами в аудиторії і вдома в режимі реального часу» // XII Регіональна студентська науково-технічна конференція «Наука – перші кроки»: тези доповідей: в 4 т. Т. 1. – Маріуполь: ПДТУ, 2016. – С. 291-292.
15. Землянова, Л.М. Сетевое общество, информационализм и виртуальная культура / Л.М. Землянова // Вестн. Моск. ун-та. – 1999. – Сер.10. - №2. – С. 58-56
16. Пивоваров М.Г., Медко Д.А. Перспективы создания и развития информационно-коммуникационной системы Украины // Экономика: проблемы теории та практики: Зб. наук. праць. – Вип. 49. – Дніпропетровськ: Дніпропетр. Нац. Ун-т, 2000. – С.56-61
17. Психология безопасности труда / Укладач Кальянов А.В. // Донецкий областной совет профсоюза, 2008. – 32 с.
18. PHP QR Code open source (LGPL) library - URL: <http://phpqrcode.sourceforge.net/index.php#demo> (дата звернення: 15.06.2022).
19. Open Server Panel / Локальный веб-сервер для Windows - URL: <https://ospanel.io/> (дата звернення: 10.06.2022).
20. Офіційний сайт додатку PhpMyAdmin - URL: <https://www.phpmyadmin.net/>(дата звернення: 11.06.2022).
21. Programming PHP, 4th Edition by Kevin Tatroe, Peter MacIntyre - с. 26-30
22. MySQL :: MySQL 8.0 Reference Manual - URL: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html> (дата звернення: 12.06.2022).
23. Overview of E-Ticketing System Advantage and Disadvantage of E-Ticketing-URL: <https://text-id.123dok.com/document/lq51vl0jy-overview-of-e-ticketing-system-advantage-and-disadvantage-of-e-ticketing.html> (дата звернення: 12.05.2022).

24. What is Python? Executive Summary - URL:
<https://www.python.org/doc/essays/blurb/> (дата звернення: 04.05.2022-
16.06.2022).

25. What is Java technology and why do I need it? - - URL:
https://www.java.com/en/download/help/whatis_java.html#:~:text=Java%20is%20a%20programming%20language,services%20and%20applications%20are%20built
(дата звернення: 1.06.2022).

ДОДАТОК А

Програмний код PHP усіх сторінок візуальної оболонки

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="name" content="width=device-width, initial-scale=1.0">
  <title>Система перевірки</title>
</head>
<body>
  <div class = "content" >
<form action="qr_creator.php">
  <h2>Оберіть дію:</h2>

  <button style="height:50px;width:150px" action= "qr_creator.php">Створити
QR-код</button>
</form>
<form action= "ticket_checker.php">
  <button style="height:50px;width:150px" action=
"ticket_checker.php">Перевірити квиток</button>
</form>
</div>
  <style type = "text/css">
    .content {
      width: 400px;
      text-align: center;
    }
  </style>
</body>
</html>
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="name" content="width=device-width, initial-scale=1.0">
  <title>Система перевірки</title>
</head>
<body>
<div class = "content" >
<form action="create_qr_id.php" method="post">

```

```

<p><b>Форма для переробки в QR-код:</b><br>
ID квитка: <br /> <input type="text" name="ticket_id" /><br />
<p></p>
<input type="submit" value="Створити QR-код" />
</form>
</div>
<style type = "text/css">
    .content {
        width: 400px;
        text-align: center;
    }
</style>
</body>
</html>
<?php

function create_qr_id()
{
    if(isset($_POST["ticket_id"])){

        $text = $_POST["ticket_id"];
    }

    include 'phpqrcode/qrlib.php';

    $path = "tickets/t";
    $file = "$path$text.png";

    $secc = 'L';
    $pixel_Size = 10;
    $frame_Size = 10;

    QRcode::png($text, $file, $secc, $pixel_Size, $frame_size);

    echo "<center><img src='\".$file.\"></center>";

}

```

```

create_qr_id();
?>
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="name" content="width=device-width, initial-scale=1.0">
  <title>Система перевірки</title>
</head>
<body>
<div class = "content" >
<form action="qr_id_reader.php" method="post">
  <p><b>Форма для перевірки квитка:</b><br>
  Назва файлу з QR-кодом: <br /> <input type="text" name="ticket_id" /><br />
  <p></p>
  <input type="submit" value="Зчитати QR-код" />
</form>

</div>
<style type = "text/css">
  .content {
    width: 400px;
    text-align: center;
  }
</style>
</body>
</html>
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Система перевірки</title>
</head>
<body>
<table cellpadding="5" cellspacing="0" border="1">
<?php

function show_ticket_id()
{
  if(isset($_POST["ticket_id"])){

```

```

$txt = (string) $_POST["ticket_id"];
$path = 'D:\Study\openserver\OpenServer\domains\passenger\tickets\t';
$path = $path.$txt.'.png';
}

require __DIR__ . "/vendor/autoload.php";
$qr = new Zxing\QrReader($path);
$text = $qr->text();

return $text;
}
$decoded_qr = show_ticket_id();

$HOST = 'localhost';
$USER = 'root';
$PASSWORD = 'root';
$DB_NAME = 'passenger';

$link = mysqli_connect($HOST, $USER, $PASSWORD, $DB_NAME) or
die(mysqli_connect_error($link));

mysqli_query($link, "SET NAMES 'utf8'");

$query = sprintf("SELECT * FROM `passenger_data` WHERE `id` =
$decoded_qr" );

$result = mysqli_query($link, $query) or die(mysqli_error($link));
$num_rows = mysqli_num_rows($result);

mysqli_close($link);

while ($row = mysqli_fetch_assoc($result) ) {
    $mass[] = $row;
}

foreach ($mass as $key => $value) {
    foreach ($value as $skey => $svalue) {
        $arr[] = $skey;
        $arr = array_unique($arr);
    }
}
}

```

```
echo "<tr>";
for ($i=0; $i < count($arr) ; $i++) {
    echo "<th>".$arr[$i]."</th>";
}
echo "</tr>";
foreach ($mass as $key => $value) {
    echo "<tr>";
    foreach ($value as $subkey => $subvalue) {
        echo "<td>".$subvalue."</td>";
    }
    echo "</tr>";
}
?>
</table>

</body>
</html>
```