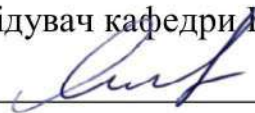


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій

"Допущений до захисту"

Завідувач кафедри ІУСТ



к.т.н, доц. Михелєв І.Л.

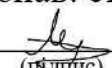
" ____ " ____ 2026 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "бакалавр"

на тему: Розробка інформаційної системи для продажу залізничних
квитків для пасажирських перевезень

Виконав: студент групи


(підпис)

Рускуліс А.В.

Керівник роботи:

(посада, науковий ступінь вчене звання)

(підпис)

Книрик Н.Р.

м. Миколаїв – 2026 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій
Спеціальність 122 "Комп'ютерні науки"
Освітня програма "Комп'ютерні науки"

"ЗАТВЕРДЖУЮ"

Гарант освітньої програми


к.т.н, доц. Гайда А.Ю.

" ____ " _____ 2026 р.

*ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти "бакалавр"*

Студенту _____ Рускулісу Андрісу Вадимовичу
(прізвище, ім'я, по батькові)

1. Тема роботи: _____ Розробка інформаційної системи для продажу залізничних
квитків для пасажирських перевезень

Керівник роботи _____ Книрік Наталя Ромуальдівна

Затверджені наказом ректора № 270-уч від "31" березня 2026 року.

2. Термін подання роботи: _____ 22 червня 2026 р.

3. Вихідні дані до проєкту (роботи) ДСТУ щодо обробки інформації, літературні
джерела, технічна документація на існуючі аналоги систем,

4. Перелік питань, що належать до розробки (найменування розділів)
_____ розробка концепції інформаційної системи для продажу залізничних
квитків; розробка проєкту інформаційної системи для продажу квитків;
реалізація проєкту інформаційної системи для продажу квитків; охорона праці та
техніка безпеки

5. Перелік презентаційних матеріалів _____ презентація Power Point

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1		02.02.2026	23.02.2026
2		23.02.2026	20.03.2026
3		20.03.2026	24.04.2026
4		24.04.2026	01.06.2026

7. Дата видачі завдання 02 лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів роботи	Примітка
1	Аналіз предметної галузі	02.02.2026	
2	Постановка задачі	16.02.2026	
3	Розробка концепції	23.02.2026	
4	Розробка проєкту ІС	23.03.2026	
5	Реалізація проєкту	27.04.2026	
6	Розробка питань з охорони праці	01.06.2026	
7	Розробка графічних матеріалів	05.06.2026	
8	Подання роботи на перевірку на плагіат	08.06.2026	
9	Подання роботи рецензенту	15.06.2026	
10	Подання роботи на захист	22.06.2026	

Студент


(підпис)

Рускуліс А.В.
(ПІБ)

Керівник роботи


(підпис)

Книрик Н.Р.
(ПІБ)

АНОТАЦІЯ

У кваліфікаційній роботі розроблено інформаційну систему для продажу залізничних квитків у сфері пасажирських перевезень, що забезпечує автоматизацію процесів пошуку рейсів, бронювання місць, оплати та створення електронних квитків для пасажирів, а також адміністративна складова вокзалу. Робота включає чотири розділи, в яких послідовно висвітлено аналіз предметної області, проектування та реалізацію системи, а також розкрито питання охорони праці та техніки безпеки.

Ключові слова: інформаційна система; залізниця; пасажирські перевезення; продаж квитків; функціонування вокзалу.

ABSTRACT

This thesis presents the development of an information system for the sale of railway tickets in the passenger transport sector, which automates the processes of searching for services, booking seats, payment and the generation of e-tickets for passengers, as well as managing the administrative functions of the station. The thesis comprises four chapters, which sequentially cover the analysis of the subject area, the design and implementation of the system, and address issues of occupational health and safety.

Keywords: information system; railway; passenger transport; ticket sales; railway station operations.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	7
ВСТУП	8
РОЗДІЛ 1 РОЗРОБКА КОНЦЕПЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ПРОДАЖУ ЗАЛІЗНИЧНИХ КВИТКІВ	10
1.1 Опис предметної області.....	10
1.2 Аналіз предметної області	12
1.3 Формування вимог до інформаційної системи	13
1.4 Розробка концепції інформаційної системи	14
РОЗДІЛ 2 РОЗРОБКА ПРОЄКТУ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ПРОДАЖУ КВИТКІВ	18
2. 1 Загальносистемні рішення	18
2. 2 Рішення з інформаційного забезпечення	20
2. 2. 1 Діаграми потоків даних.....	21
2. 2. 2 Концептуальна модель БД	25
2. 2. 3. Логічна модель БД	28
2. 2. 4. Фізична модель БД.....	29
2. 3 Рішення з математичного забезпечення.....	32
2. 4 Рішення з програмного забезпечення	33
2. 4. 1 Розробка інтерфейсу користувача	35
2.4.2 Розробка клієнтської частини	39
2.4.3 Взаємодія з БД.....	39
2. 5 Рішення з технічного забезпечення.....	40
2.5.1 Вимоги до пристроїв користувачів	40
2.5.2 Вимоги до серверної частини	41

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОЄКТУ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ПРОДАЖУ КВИТКІВ	43
3.1 Керівництво користувача системи.....	43
3.2 Результат використання системи	43
РОЗДІЛ 4 ОХОРОНА ПРАЦІ ТА ТЕХНІКА БЕЗПЕКИ	46
ВИСНОВКИ.....	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	51
ДОДАТОК А	55
ДОДАТОК Б.....	58

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

База даних – БД

Інформаційна система – ІС

Системи управління базами даних – СУБД

Application Programming Interface – API

Діаграма потоку даних – ДПД

Virtual Central Processing Unit - vCPU

Засіб індивідуального захисту – ЗІЗ

ВСТУП

Залізничний транспорт традиційно посідає одне з основних місць у системі пасажирських перевезень. Щодня тисячі людей користуються цим видом транспорту, а отже будь-яка взаємодія із системою має виконуватись швидко та точно. Проте реальна картина на багатьох вокзалах виглядає абсолютно інакше: довгі черги до каси, паперові розклади руху потягів та відсутність будь-якого цифрового сервісу. Це є великою проблемою для пасажирів, котрий звик замовляти усе онлайн, та безпосередньо впливає на потік пасажирів і рентабельність залізниці.

Актуальність теми роботи обґрунтована тим, що процеси обслуговування пасажирів на залізничному вокзалі вимагають нових підходів у нинішньому процесі цифровізації. Повна залежність від касового персоналу, відсутність онлайн-доступу до розкладу та неможливість придбати квиток дистанційно — усе це системний недолік, що знижує конкурентоспроможність залізничного транспорту порівняно з іншими видами пересування. Вирішення цих проблем потребує розробки повноцінної ІС, що покращить недоліки існуючої залізниці.

Питання автоматизації транспортного обслуговування достатньо висвітлене в науковій літературі, особливо окремими аспектами: проєктування БД, побудови застосунків, моделювання бізнес-процесів. Проте, комплексні рішення, що охоплюють увесь цикл створення ІС як концептуально, так і фізично, у відкритому доступі представлені значно рідше. Це підтверджує доцільність виконання даної роботи як цілісного проєктного рішення.

Метою кваліфікаційної роботи є розробка ІС для продажу залізничних квитків, що охоплює повний цикл проєктування — від аналізу предметної області, створення концепції до її повної реалізації.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- описати та проаналізувати предметну область залізничного вокзалу, виявити наявні проблеми та обґрунтувати необхідність автоматизації;
- сформулювати вимоги до ІС та розробити її концепцію;

- проаналізувати можливі рішення для реалізації системи й визначити її архітектуру;
- проаналізувати дані, що необхідні для розробки БД;
- розробити програмне забезпечення системи, включно з інтерфейсом, клієнтською та серверною частинами;
- визначити рішення з математичного та технічного забезпечення;
- реалізувати систему, підготувати керівництво користувача та проаналізувати результати її використання.

Об'єктом проєктування є процес організації та автоматизації продажу квитків для пасажирських залізничних перевезень. Предметом проєктування виступає ІС, що реалізує цей процес, — її архітектура, модулі, моделі даних та програмні компоненти.

Засобами проєктування і створення системи слугували методи структурного аналізу, ДПД для моделювання руху інформації в системі, діаграми для побудови структури даних, а також сучасні інструменти розробки програмного забезпечення. Технічні засоби та програмне забезпечення, що використовувалися в процесі розробки проєкту, детально описані у відповідних розділах роботи.

У результаті виконання роботи розроблено функціональну ІС, що вміщає увесь необхідний функціонал: пошук та перегляд актуальних маршрутів, бронювання та купівлю місць, оформлення електронних квитків, обробку повернень, а також розподіл користувачів на категорії: пасажери, касири та адміністратори вокзалу; із різними рівнями доступу.

РОЗДІЛ 1 РОЗРОБКА КОНЦЕПЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ПРОДАЖУ ЗАЛІЗНИЧНИХ КВИТКІВ

1.1 Опис предметної області

Залізничний вокзал є регіональним транспортним вузлом, що забезпечує пасажирські перевезення. Вокзал обслуговує декілька напрямків руху потягів і щодня приймає велику кількість пасажирів, однак його структура, процеси обслуговування та організації роботи суттєво поступається сучасним стандартам.

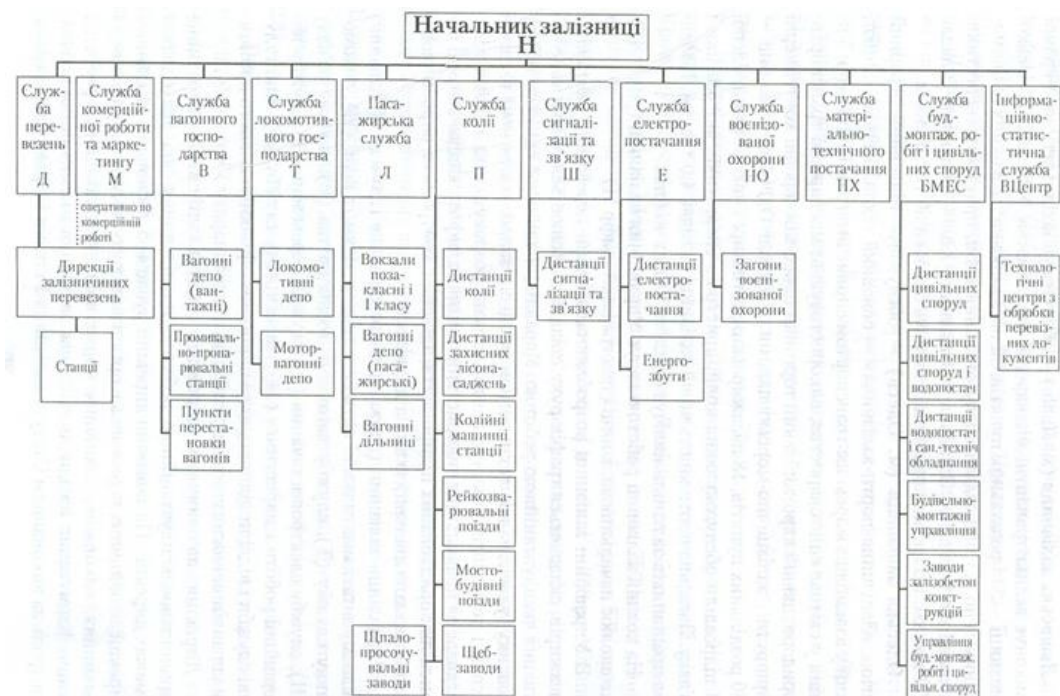


Рисунок 1.1. - Організаційна структура залізничного вокзалу

(відповідно до [1])

Організаційно залізничний вокзал є складною структурою, підпорядкованою начальнику вокзалу. Загальне керівництво здійснює адміністративний апарат, який координує роботу всіх служб, забезпечує виконання графіків руху поїздів, контроль комерційної діяльності та безпеку пасажирів [2].

На чолі структури стоїть начальник вокзалу, який несе відповідальність за загальну організацію роботи, виконання планів перевезень, дотримання техніки безпеки та фінансово-господарську діяльність. Йому безпосередньо підпорядковані керівники основних служб та відділів. На цьому рівні формується звітність про обіг квитків, контролюється виконання розкладу руху потягів та відбувається актуалізація стану ресурсів ІС [3].

Пасажирська служба є одним з ключових підрозділів вокзалу. Вона відповідає за безпосереднє обслуговування пасажирів та надання послуг, що мають відношення до процесу продажу та повернення квитків, а також займаються інформуванням пасажирів про зміни у розкладі потягів. Відповідно до встановлених вимог, вокзал зобов'язаний забезпечувати своєчасне інформування пасажирів про прибуття та відправлення поїздів, оформлення проїзних документів та іншу загальну допомогу пасажирам [4].

Уся операційна інформація з рівня кас передається до централізованої БД. На рівні адміністрації вокзалу здійснюється контроль за виконанням графіків руху, формуванням звітності та підтримкою актуальності наданої інформації [2].

До складу структури також входять технічні та експлуатаційні служби: колії, електрифікації, сигналізації та блокування, матеріально-технічного постачання, які забезпечують технічну складову інфраструктури вокзалу. Окрім того, функціонують допоміжні служби: безпеки, інформаційно-статистичний центр та утримання території [5].

Така ієрархічна організаційна структура є типовою для більшості залізничних вокзалів і характеризується централізованим управлінням, високим навантаженням на пасажирську та касову служби, а також високою вразливістю до помилок через людський фактор. Проте її основним недоліком є недостатня цифровізація процесів обслуговування пасажирів [2; 6].

Описана структура залізничного вокзалу має очевидні обмеження, що впливають на загальну роботу системи, а також на зниження її рентабельності. Для виявлення причин, що негативно впливають на функціонал, потрібно провести аналіз предметної області.

1.2 Аналіз предметної області

Існуюча організація обслуговування пасажирів на залізничному вокзалі має низку системних недоліків, що суттєво впливають на якість надання послуг та ефективність роботи вокзалу.

Основною проблемою є повна залежність процесу продажу та повернення квитків від касового персоналу. Касові служби залишаються єдиним каналом взаємодії з пасажирами. У пікові утворюються значні черги, в яких пасажир змушені простоювати по 20–40 хвилин [7]. Проте, коли навантаження на каси не є значним, вони часто простоюють, що свідчить про нераціональне використання людських ресурсів.

Другою суттєвою проблемою є відсутність сучасних методів обслуговування пасажирів. Актуальний розклад руху поїздів існує лише на паперових таблицях, розміщених безпосередньо на вокзалі. Пасажир вимушений потрапити на вокзал, щоб дізнатися інформацію про рейси, наявність місць та їх вартість. Жодного офіційного цифрового ресурсу для перегляду розкладу онлайн. Цей підхід є особливо незручним для мешканців віддалених районів, людей похилого віку та осіб, котрі мають проблеми із мобільністю.

Придбання та повернення квитків також можливе виключно через касу. Пасажир особисто звертається до касира, вказує маршрут і дату, обирає місце, оплачує квиток і отримує паперовий квиток. Таким чином, будь-яка взаємодія з вокзалом вимагає обов'язкової фізичної присутності пасажирів.

Такий процес оформлення документів підвищує ймовірність помилок, що негативно впливає на надійність та продуктивність системи. Нерівномірне навантаження на касирів та високий вплив людського фактора суттєво знижують загальний рівень сервісу.

Всі ці фактори призводять до зниження якості обслуговування, зростання витрат на утримання великої кількості касового персоналу та втрати частини потоку пасажирів. Для пасажирів, котрі звикли до швидкого онлайн-обслуговування в інших сферах, часто обирають альтернативні види транспорту.

Таким чином, проведений аналіз роботи залізничного вокзалу свідчить про нагальну потребу в цифровізації процесів обслуговування пасажирів. Для вирішення існуючих проблем необхідно створити сучасну ІС, яка буде відповідати цифровим стандартам та дозволить пасажирам самостійно взаємодіяти із системою без допомоги касира. Впровадження такої системи дасть змогу суттєво скоротити черги, оптимізувати роботу персоналу, зменшити кількість помилок і підвищити загальну якість обслуговування пасажирів.

1. 3 Формування вимог до інформаційної системи

Під час розробки ІС для залізничного вокзалу необхідно враховувати потреби двох основних груп користувачів: пасажирів та адміністративно-касового персоналу вокзалу.

Пасажири складають основну групу користувачів та поділяються на дві категорії залежно від способу взаємодії із системою.

Перша категорія пасажирів взаємодіє із системою через вебсайт. Вони самостійно переглядають розклад, обирають рейс та бронюють місце, а в результаті отримують квиток в електронному вигляді. За потреби вони також можуть подати запит на повернення квитка.

Друга категорія пасажирів не має доступу до інтернету або смартфона, зокрема це характерно для людей похилого віку. Вони приходять на вокзал особисто і звертаються до касира, який виконує всі необхідні дії в системі від їхнього імені. Таким чином, обидві категорії пасажирів отримують однаковий результат, але різними шляхами взаємодії із системою. Незалежно від категорії, ключові потреби пасажира залишаються спільними: отримання актуальної інформації про розклад руху, пошук рейсів за маршрутом і датою, вибір вагона та місця, оплата проїзду та можливість повернення квитка.

Адміністративно-касовий персонал вокзалу потребує ефективних засобів для обслуговування пасажирів, які не користуються мобільним додатком, а також для управління діяльністю вокзалу. До їхніх потреб належать оформлення

квитків через касу, їх друк, обробка повернень, контроль наявності вільних місць і формування звітності про продажі.

Для задоволення потреб попередньо описаних груп користувачів, ІС повинна включати два основні компоненти.

Веб-додаток для взаємодії пасажирів з системою. Для взаємодії пасажира із залізницею розробляється сайт, з яким можна взаємодіяти як із мобільного пристрою, так і з комп'ютера. Сайт забезпечуватиме повний набір функцій необхідний для пасажира, включно із переглядом раніше придбаних квитків в особистому кабінеті.

БД, що призначену для зберігання інформації про рейси, розклад, вагони, місця, пасажирів, квитки та транзакції. БД забезпечуватиме цілісність та доступність усіх даних системи, необхідних як для роботи веб-додатка, так і для обслуговування пасажирів касовим персоналом вокзалу.

Впровадження такої системи дозволить суттєво зменшити навантаження на касовий персонал, оптимізувати роботу вокзалу та підвищити якість обслуговування пасажирів.

На основі сформульованих вимог буде розроблена концепція ІС, яка враховуватиме потреби обох груп користувачів та забезпечуватиме надійне функціонування системи.

1.4 Розробка концепції інформаційної системи

Враховуючи попередньо вказані вимоги до функціональності системи, можна окреслити загальну концепцію ІС.

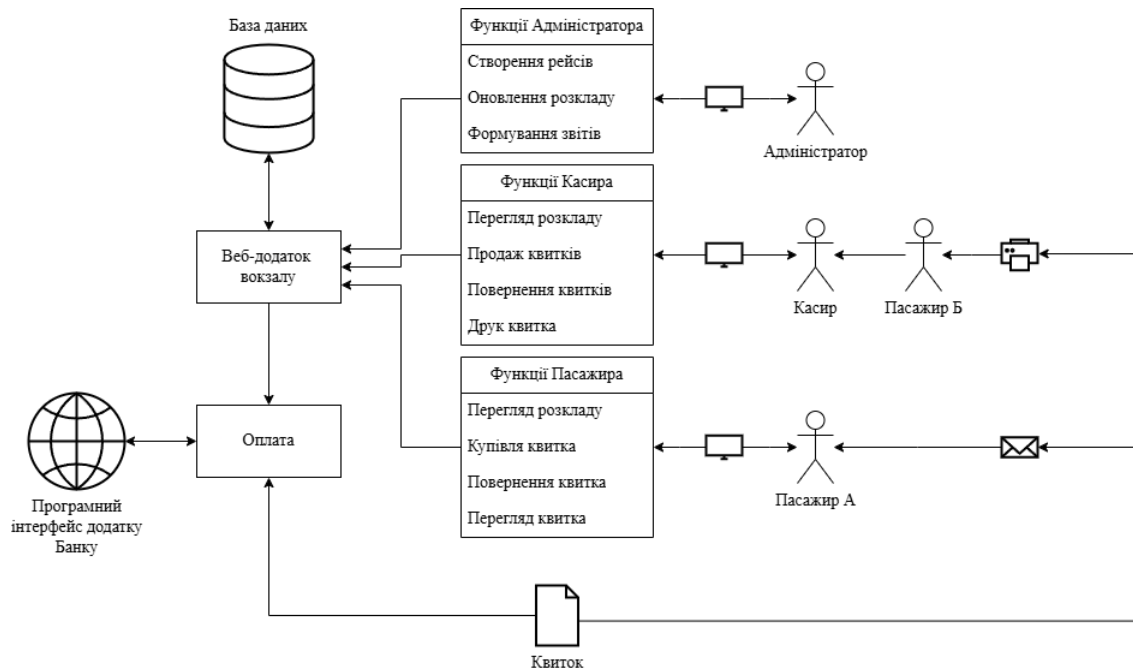


Рисунок 1.2. – Модель концепції ІС продажу залізничних квитків

Адміністратор вокзалу є відповідальним за забезпечення коректної роботи системи. До його функцій належать: управління розкладом рейсів і складом вагонів, контроль за актуальністю даних про наявність вільних місць та формування звітності за вказані періоди.

Пасажир є основним користувачем системи. За допомогою веб-додатку він отримує можливість самостійно здійснювати пошук рейсів за заданими параметрами, обирати вагон і конкретні місця, вносити особисті дані, скористатися додатковими послугами, здійснювати оплату та отримувати електронний квиток. За потреби пасажир також може подати запит на повернення квитка.

Касир забезпечує традиційний метод обслуговування для тих пасажирів, які не мають технологічного забезпечення, мають обмежений доступ до інтернету або недостатній рівень цифрової грамотності. Він виконує ті ж функції, що і пасажир у веб-додатку, проте замість формування електронного квитка, він видає паперовий. Варто зазначити, що роль касира це також надання пасажирам необхідної консультаційної підтримки. Збереження цього каналу обслуговування є принциповим для збереження повної доступності системи для різних верств населення.

Окремий розподіл ролей касира та пасажирів в системі є обґрунтованим, незважаючи на схожість їхніх функцій. Такий підхід забезпечує гнучкість у майбутньому розвитку системи: кожен роль можна вдосконалювати незалежно, додавати функції для одних без обмеження впливу інших. Крім того, розділення ролей спрощує контроль доступу, підвищує безпеку даних та дозволяє швидше адаптувати систему до нових вимог.

Для реалізації проєкту необхідно визначити програмне забезпечення у кількох ключових категоріях: СУБД, мова програмування, фреймворк для розробки веб-додатку та інструменти управління станом застосунку. Детальне обґрунтування вибору кожного з компонентів наведено у розділі 2.1, тут зазначені лише обрані рішення: як СУБД використовується MySQL, мовою розробки обрано TypeScript, основним фреймворком – Next.js, а для управління глобальним станом – Redux.

Для ефективної роботи та повноцінного функціонування системи необхідна інтеграція з низкою зовнішніх сервісів.

Платіжний шлюз забезпечує безпечну обробку онлайн-платежів: прийом карткових транзакцій, підтвердження оплати та ініціювання повернення коштів при скасуванні квитка. Взаємодія з банківською системою відбувається через API [8].

Поштовий сервіс використовується для відправлення листів, а саме для електронного квитка у вигляді вкладення або повідомлення про успішне повернення. Автоматична розсилка таких листів суттєво знижує навантаження на персонал [9].

Після оплати система генерує унікальний QR-код, що містить зашифровані дані про маршрут, вагон, місце та пасажирів. Контролер у поїзді зчитує код за допомогою сканера або мобільного пристрою й миттєво отримує всі необхідні дані про клієнта та його броньоване місце [10].

Сервіс автентифікації забезпечує верифікацію облікових записів пасажирів і адміністраторів, розмежування прав доступу та захист персональних даних.

Редактор шаблонів використовується адміністратором для оновлення розкладу та внесення змін до сторінок сайту без необхідності залучення розробника.

Запропонована концепція створює реальні передумови для підвищення операційної ефективності вокзалу. Конкретно це виражається у скороченні кількості касових працівників, адже більшість пасажирів самостійно купуватимуть квитки онлайн; усунення черг у пікові години завдяки цілодобовому доступу до системи з будь-якого пристрою; зменшення кількості помилок при оформленні квитків через автоматизовану верифікацію введених даних; прискорення процедури бронювання та повернення квитків через онлайн-запит замість особистого візиту до каси; а також отримання адміністрацією детальної аналітики продажів у реальному часі. Результатом впровадження стане вокзал, який обслуговує пасажирів ефективніше, з меншими операційними витратами та значно вищим рівнем задоволеності клієнтів.

РОЗДІЛ 2 РОЗРОБКА ПРОЄКТУ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ПРОДАЖУ КВИТКІВ

2. 1 Загальносистемні рішення

Розробка ІС для продажу квитків вимагає чіткого визначення архітектурних і технологічних рішень, які забезпечать надійність, продуктивність, масштабованість та зручність підтримки системи. Правильний підбір компонентів є основним для створення ефективної системи.

Для зберігання та управління даними ІС обрано реляційну СУБД MySQL. Вона є однією з найбільш популярних відкритих СУБД, поступаючись лише Oracle Database [11]. Ця система широко відома як надійна та безпечна та широко використовується у популярних веб застосунках [12].

Серед ключових технічних переваг MySQL варто відзначити підтримку принципів ACID [13], що гарантує цілісність даних при транзакційних операціях. MySQL забезпечує цілодобову доступність завдяки різноманітним конфігураціям, а також механізмам резервного копіювання і відновлення даних у разі збоїв [14].

Вибір серверної інфраструктури є одним із фундаментальних рішень при проєктуванні ІС, оскільки він безпосередньо впливає на продуктивність, надійність та масштабованість застосунку. Конкретні характеристики серверів визначаються насамперед масштабом системи.

Для регіональних систем із помірним навантаженням достатньо серверів середнього класу з багатоядерними процесорами та оперативною пам'яттю від 32 до 128 ГБ. Такі конфігурації здатні забезпечити стабільну роботу веб-застосунку. При цьому рекомендованою практикою є розподіл системи на два окремі компоненти: сервера застосунків і сервера БД. Такий підхід, дозволить зменшити загальне навантаження на кожен модуль системи, а також масштабувати кожен із них незалежно [15].

Для великих залізничних операторів із значними обсягами пасажирських і вантажних перевезень застосовуються значно потужніші рішення. Зокрема,

сервера класу IBM System z9 з обсягом оперативної пам'яті від 512 ГБ до 4 ТБ, що використовуються для обробки великих обсягів даних у режимі реального часу [16].

Оскільки розробка ІС фокусується на регіональному вокзалі, масштабованість має бути закладена ще на етапі проєктування, адже це суттєво спрощує подальше зростання системи та полегшує її оновлення чи збільшення потоку користувачів [17]. Це означає, що на початковому етапі достатньо конфігурації середнього класу, однак архітектурні рішення закладають можливість нарощування потужності у майбутньому без необхідності зміни системи.

Для реалізації клієнтської та серверної частин застосунку обрано сучасну мову програмування – TypeScript. Вона дозволяє виявляти помилки типізації ще на етапі написання коду, що суттєво прискорює розробку та забезпечує надійність оброблюваних даних [18].

Для взаємодії з БД використовується мова SQL – декларативна мова програмування для опису запитів, що дозволяє повну взаємодію із даними у створеній БД.

Серед популярних веб фреймворків для розробки повноцінних застосунків можна виділити два основних: Next.js та Nuxt.js.

Обидва фреймворки підтримують серверну та статичну побудову сторінки додатку, що та зменшує час початкового завантаження, а також мають вбудовані рішення для оптимізації роботи веб-додатку та використаними зображеннями [19]. Ключова відмінність між фреймворками полягає в екосистемі та спільноті. З точки зору продуктивності, Next.js загалом обробляє більше запитів за секунду в типових сценаріях.

Враховуючи ширшу екосистему, вбудовану підтримку TypeScript, вищу продуктивність у сценаріях, що використовують серверний рендеринг та значно більшу спільноту розробників порівняно, для реалізації проєкту обрано Next.js. Цей фреймворк забезпечує повну розробку додатку в межах єдиного проєкту,

автоматично налаштовує інструменти нижчого рівня та дозволяє створювати інтерактивні, динамічні та швидкі React-застосунки [20].

У застосунках, що створені на основі React управління глобальним станом є одним із ключових архітектурних рішень, що безпосередньо впливає на продуктивність роботи застосунку. Двома найпоширенішими підходами є бібліотека Redux та вбудований механізм Context API.

Принципово важливо розуміти, що Context API не є повноцінним методом управління станом. Його головним мінусом є повторний рендер при кожному оновленні стану усіх, навіть тих, що не були оновлені, компонентів без винятку [21]. У виробничих застосунках із великим деревом компонентів час оновлення може сягати більше 150 мс, що негативно позначається на досвіді користувача [22].

Натомість Redux, зберігає весь стан застосунку в єдиному централізованому сховищі. Він покращує роботу додатку, забезпечуючи оптимізацію продуктивності та кращу масштабованість [23]. За даними із різних джерел, ця бібліотека використовується у 59,6% усіх React-застосунків і забезпечує роботу 72% проєктів великого масштабу [22].

Оскільки розроблювана ІС передбачає одночасну роботу кількох ролей користувачів з різними даних для обробки, складний функціонал системи, а також необхідність відстеження певних даних в реальному часі, для управління глобальним станом застосунку обрано Redux. Ця бібліотека забезпечить високу продуктивність у роботі застосунку та зручне налагодження в процесі розробки.

Сформований технологічний стек є сучасним і збалансованим рішенням для розробки веб-застосунку. Усі обрані технології активно розвиваються, мають широку спільноту розробників і регулярно отримують оновлення, що гарантує актуальність в довгостроковій перспективі. Такий набір інструментів відповідає сучасним стандартам веб-розробки і дозволяє створити надійний, масштабований та зручний у підтримці продукт.

2. 2 Рішення з інформаційного забезпечення

Для належного та безпечного функціонування системи необхідно створити та реалізувати реляційну модель БД. З метою досягнення поставленої цілі буде проведений аналіз вимог користувачів системи, а також даних, котрі будуть зберігатися.

2. 2. 1 Діаграми потоків даних

ДПД уможлиблює візуалізацію потоку даних з урахуванням сховища даних і зовнішніми сутностями, які взаємодіють із системою. Діаграма дає змогу зрозуміти як функціонує додаток, а саме: рух об'єктів у системі та зберігання їх [24].

Основним принципом ДПД є забезпечення збалансованості, а саме: діаграма потоків даних одного рівня точно відображається на діаграмі наступного рівня. Ідеальним рівнем декомпозиції можна вважати рівень, на якому користувачі не можуть надати більш детальний опис процесу, що описується [25].

Для створення ДПД буде використано метод Гейна – Сарсона, що пропонується на рисунку 2.1.

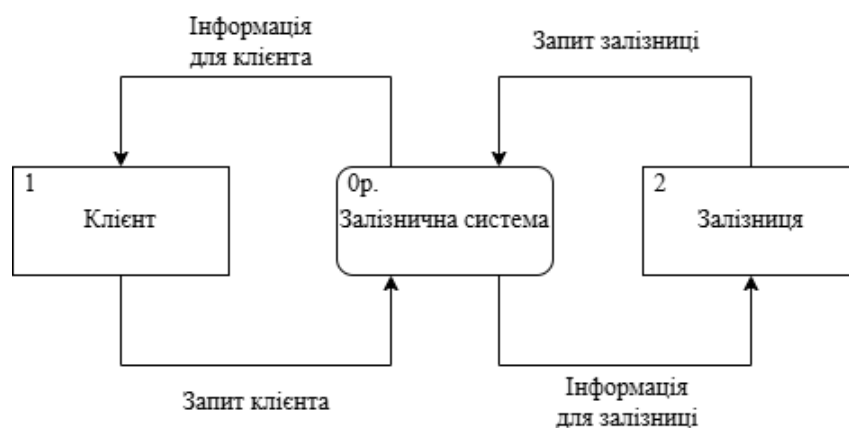


Рисунок 2.1. - ДПД 0-го рівня

Основними користувачами системи є клієнт і залізниця. Вони одночасно надають дані системі та отримують інформацію із неї як це продемонстровано на рис. 2.2.

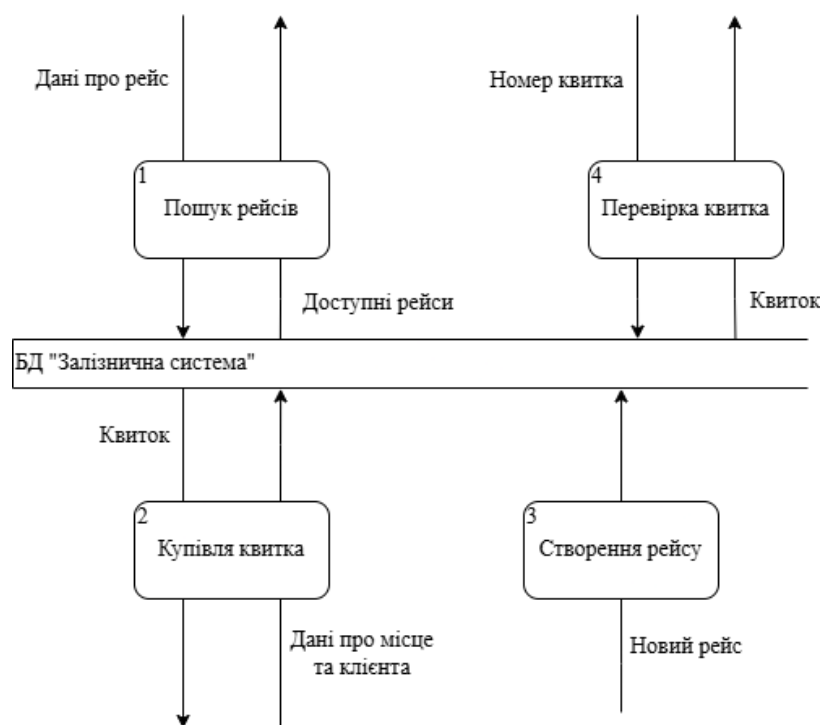


Рисунок 2.2. - ДПД 1-го рівня

Діаграма відображає основні процеси, застосовані в системі: пошук і бронювання місця, додавання й оновлення списку активних рейсів. Кожен із цих процесів є більш складним і потребує подальшої декомпозиції.

Процес купівлі квитка можна розділити на додаткові процеси отримання ціни та оплати (рис. 2.3).

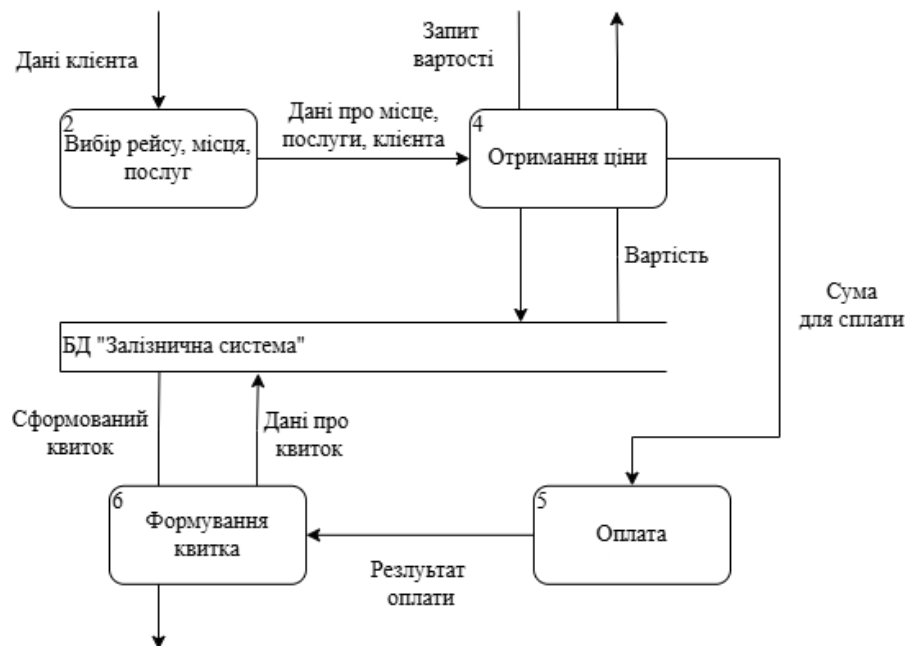


Рисунок 2.3. - ДПД 2-го рівня для процесу «Купівля квитка»

Узагальнені дані щодо відношень потоків даних та атрибутів репрезентуємо в таблиці 2.1.

Таблиця 2.1.

ВІДНОШЕННЯ ПОТОКІВ ДАНИХ ТА АТРИБУТІВ

ДПД 0 рівня	ДПД 1 рівня	ДПД 2 рівня	Атрибути
Запит клієнта	Дані про рейс		Місце відправлення Місце прибуття Дата відправлення
	Дані клієнта про місце та клієнта	Дані про місце, послуги, клієнта	Ім'я Прізвище По батькові Дата народження Номер телефона Електронна пошта Номер рейсу Номер вагону

			Тип вагону Номер місця Додаткові послуги
		Запит вартості	Номер рейсу Тип вагону Додаткові послуги
Інформація для клієнта	Доступні рейси		Список доступних рейсів
	Квиток	Вартість	Вартість квитка з урахуванням додаткових послуг додаткових послуг
		Дані про квиток	Номер квитка Номер рейсу Номер вагону Номер місця Статус квитка
		Результат оплати	Код оплати Метод оплати Повна вартість квитка Статус оплати Дата оплати
Запит залізниці	Новий рейс		Номер рейсу Місце відправлення Місце прибуття Кількість вагонів, інформація про кожен із них (номер, тип, кількість місць, вартість)
	Номер квитка		Номер існуючого квитка

Інформація для залізниці	Інформація про квиток		Номер квитка Номер вагону Номер місця Статус квитка Номер клієнта Номер рейсу Додаткові послуги
-----------------------------	--------------------------	--	---

Проведений аналіз за методологією ДПД, дозволяє відокремити основні функції користувачів та виділити необхідні дані для збереження. Описані атрибути надають змогу створити модель БД для розроблюваної ІС.

2. 2. 2 Концептуальна модель БД

Концептуальний рівень архітектури визначає логічну схему БД. На цьому рівні виконується концептуальне проєктування, яке охоплює аналіз інформаційних потреб користувачів. Він створює єдиний концепт БД, що є унікальним для будь-якої частини додатку. Представлення БД на концептуальному рівні системи описується концептуальною схемою [26] .

Ураховуючи необхідні дані для розробки додатку, в концептуальну модель включено основні сутності й необхідні атрибути (табл. 2.2).

Таблиця 2.2.

АТРИБУТИ СУТНОСТЕЙ

Сутність	Атрибут	Опис
Клієнт	Ім'я	Ім'я клієнта
	Прізвище	Прізвище клієнта
	По батькові	По батькові клієнта
	Дата народження	Дата народження клієнта
	Номер телефону	Номер телефона клієнта

	Електронна пошта	Електронна пошта клієнта для квитанцій та підтвердження
Рейс	Номер рейсу	Унікальний номер рейсу
	Місце відправлення	Місце станції відправлення
	Місце прибуття	Місце станції прибуття
	Дата відправлення	День та час відправлення
	Дата прибуття	День та час прибуття
Вагон	Унікальний номер вагону	Унікальний ідентифікатор
	Номер вагону	Номер вагону для рейсу
	Тип вагону	Тип вагону (плацкарт, купе, люкс)
	Кількість місць	Кількість місць у вагоні
	Вартість	Ціна за квиток у вагоні
Квиток	Номер квитка	Унікальний номер квитка
	Номер місця	Номер місця у вагоні
	Статус квитка	Статус квитка (активний, повернутий)
Транзакція	Код транзакції	Унікальний код оплати
	Метод оплати	Метод оплати (готівка, термінал)
	Вартість	Вартість квитка з урахуванням додаткових послуг
	Дата оплати	Дата проведення оплати
	Статус оплати	Статус оплати (успішно, в обробці, помилка)

Додаткова послуга	Тип послуги	Тип додаткової послуги
	Назва	Назва додаткової послуги
	Вартість	Вартість додаткової послуги

Проведений аналіз дозволив нам установити зв'язки між сутностями, які ми продемонструємо в таблиці 2.3.

Таблиця 2.3.

ЗВ'ЯЗКИ МІЖ СУТНОСТЯМИ

Сутність від	Сутність до	Тип зв'язку
Квиток	Транзакція	1:1
Квиток	Клієнт	M:1
Квиток	Рейс	M:1
Квиток	Додаткові послуги	M:N
Рейс	Вагон	1:N
Вагон	Квиток	1:M

Моделювання «сутність-зв'язок» є основним інструментом проєктування БД та має важливе значення для моделювання інформаційних систем. Загальновизнано, що ці моделі добре відображають основну семантику багатьох ситуацій [27].

Стандартна процедура створення реляційної моделі на основі моделі «сутність-зв'язок» передбачає створення одного відношення для кожної сутності й ще одного для тернарного зв'язку; до цього проєкту додаються обмеження цілісності, що базуються на відношеннях первинний ключ – зовнішній ключ [27].

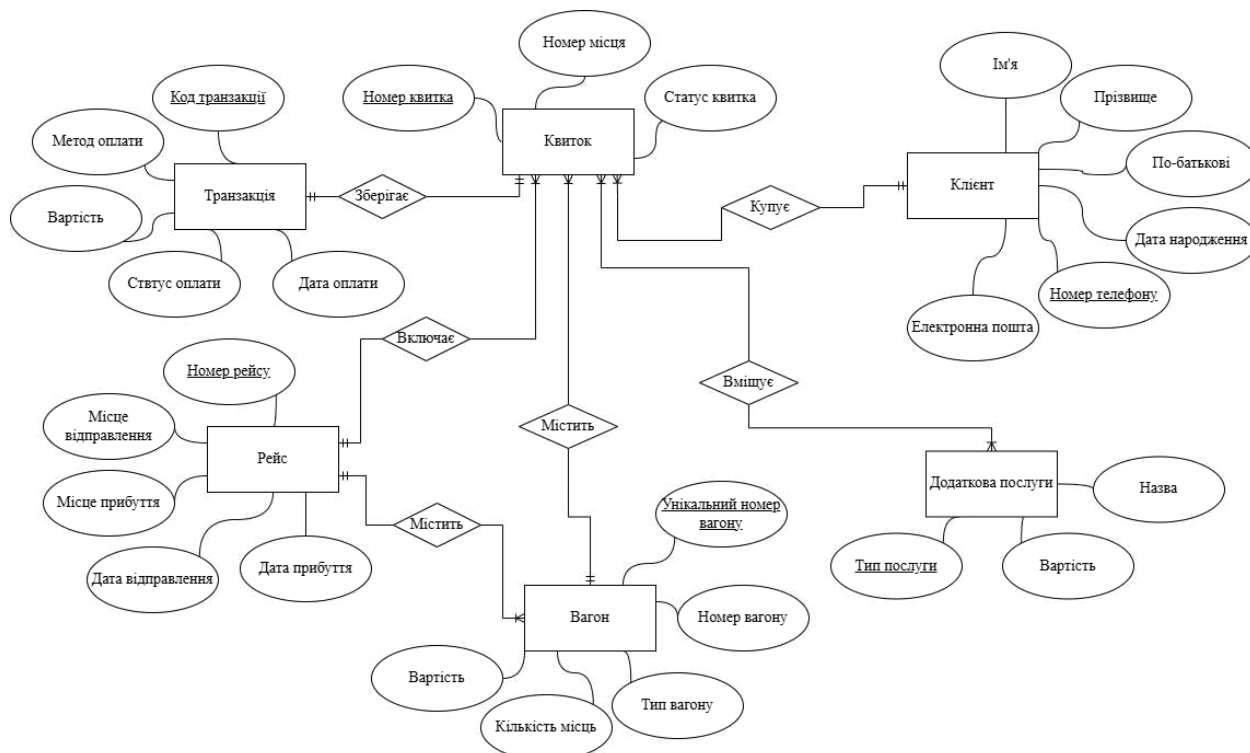


Рисунок 2.4. - Концептуальна модель БД

2. 2. 3. Логічна модель БД

Створення логічної моделі полягає у створенні схеми, визначенні сутностей та кількості атрибутів у кожному, задля побудови структури таблиць. Концептуальні моделі за певними правилами перетворюються в логічні моделі даних [28]. Іншим необхідним аспектом реалізації є визначення первинних ключів (РК) та зовнішніх ключів (FK) для кожної сутності.

У моделі відповідної БД наявні шість сутностей. «Клієнт» має шість атрибутів, із яких «Номер телефону» є первинним ключем.

Сутність «Транзакція» має шість атрибутів, «Код транзакції» та «Номер квитка» є первинним і зовнішнім ключами відповідно.

«Додаткова послуга» має три атрибути, з яких первинним ключем є «Тип послуги».

Усі попередньо зазначені таблиці мають зв'язок із сутністю «Квиток», що містить сім атрибутів, має первинний ключ «Номер квитка»; інші атрибути охоплюють чотири зовнішні ключі:

- «Номер рейсу»,

- «Номер клієнта»,
- «Номер рейсу»,
- «Додаткові послуги».

Кожен «Квиток» також має зв'язок із сутністю «Рейс», яка має п'ять атрибутів, із них «Номер рейсу» є первинним ключем.

Кожна сутність «Рейс» має зв'язок із таблицею «Вагон», що має шість атрибутів, «Унікальний номер вагону» – первинний ключ, а «Номер рейсу» – зовнішній ключ.

Кожна таблиця «Вагон» має зв'язок із таблицею «Квиток».

Зв'язок сутностей продемонструємо на рисунку 2. 5.

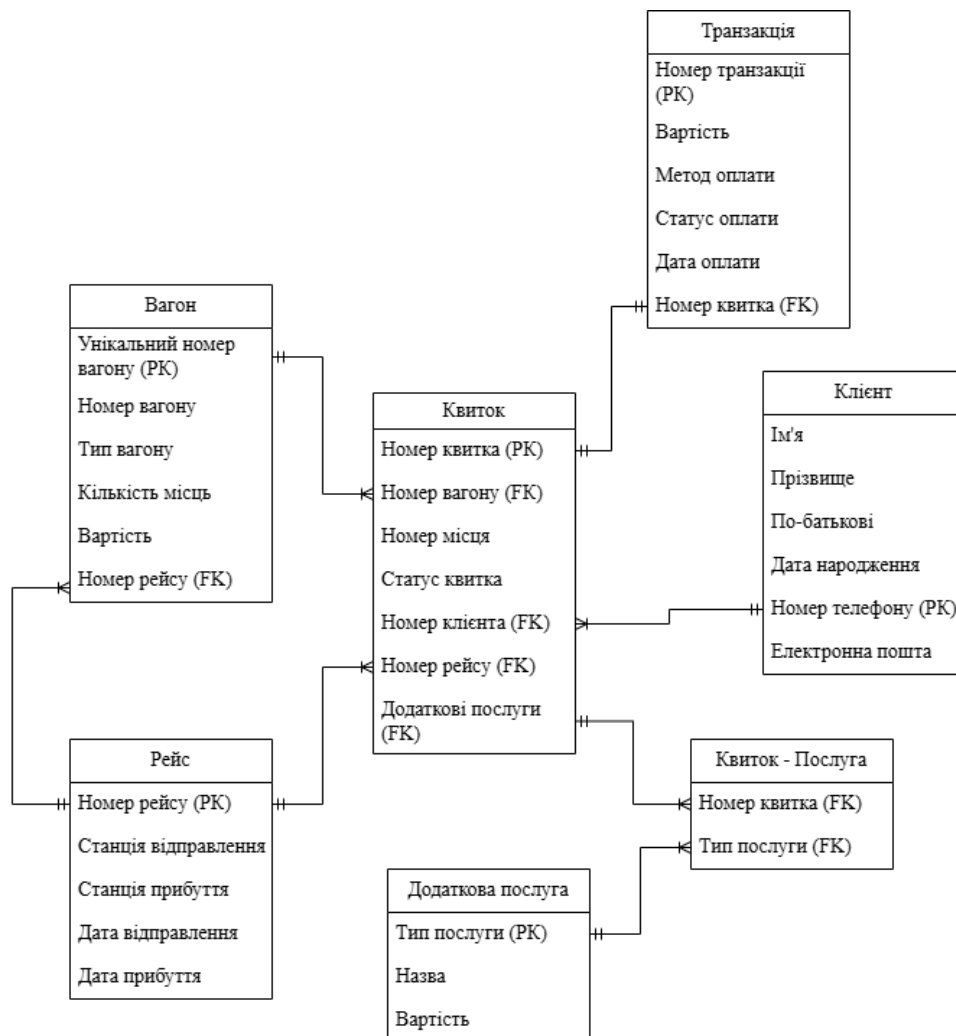


Рисунок 2.5. - Логічна схема БД

2. 2. 4. Фізична модель БД

Для правильного формування БД модель тестується за допомогою правил нормалізації. Вони дозволяють перевірити структури бази на узгодженість і логічну цілісність. [28]

Нормалізація БД – це процес представлення даних у вигляді відношень у стандартних нормальних формах, де перша нормальна форма є мінімальною вимогою. Системи БД, які не вимагають першої нормальної форми, часто називають NoSQL-системами. Концепція нормалізації та нормальних форм започаткована після створення реляційної моделі БД. Процес нормалізації є ключовим для уникнення неузгодженості в системі реляційних БД [29;30].

Перша нормальна форма є найпростішим рівнем нормалізації. Відношення знаходиться в першій нормальній формі тільки тоді, коли жоден стовпець таблиці не має таблиці в якості значення. Це передбачає, що кожна таблиця в БД має первинний ключ і кожен стовпець таблиці містить атомарні значення. Іншими словами, кожен рядок у таблиці повинен мати унікальний ідентифікатор, а кожне значення в таблиці повинно бути неподільним [29;30].

Після аналізу фізичної форми можна стверджувати, що структура задовольняє першу нормальну форму.

Друга нормальна форма ґрунтується на фундаменті попередньої і передбачає забезпечення залежності кожного неключового стовпця таблиці від первинного ключа. Отже, в таблиці не повинно бути часткових залежностей [29].

Оскільки всі таблиці відповідають першій нормальній формі та атрибути в кожній із них залежать від первинного ключа, можна стверджувати, що модель задовольняє другу нормальну форму. Таблиця ticketService не впливає на результат оцінки, адже вона є асоціативною (зв'язковою) таблицею з комбінованим первинним ключем.

Третя нормальна форма вміщає попередню і передбачає, що кожен неключовий стовпець таблиці не є транзитивно залежним від первинного ключа. Значення в записі, які не є частиною ключа цього запису, не повинні міститися в таблиці. Загалом, якщо вміст групи полів може застосовуватися до більш ніж одного запису в таблиці, можна розмістити ці поля в окремій таблиці [29;31].

Проведена перевірка моделі дозволяє стверджувати, що запропонована структура порушує вимоги третьої нормальної форми. У таблиці «Рейс» атрибути «Станція Відправлення» і «Станція Прибуття» є неключовими, функційно залежать від первинного ключа й містять дані, що мають власну семантичну структуру тобто проявляють певну сутність.

Отже, відбувається прихована транзитивна залежність: «Номер Рейсу» → «Станція Відправлення» → «Назва Станції». Така структура може створити аномалії у БД, адже назви станцій «Київ» та «київ» будуть сприйматися як два різні значення.

Для вирішення цієї залежності варто ввести нову таблицю «Станція», що матиме унікальний номер станції та її назву. У таблиці «Рейс» назви станцій будуть замінені на зовнішні ключі, які посилаються на новостворену таблицю.

Четверта нормальна форма, яку також називають нормальною формою Бойса-Кодда, і п'ята нормальна форма досить рідко використовуються на практиці. Зазвичай їх застосовують для вирішення специфічних проблем, пов'язаних із багатозначними залежностями [31].

З урахуванням розробленої логічної моделі та попереднього аналізу за правилами нормалізації створено таку фізичну модель БД.

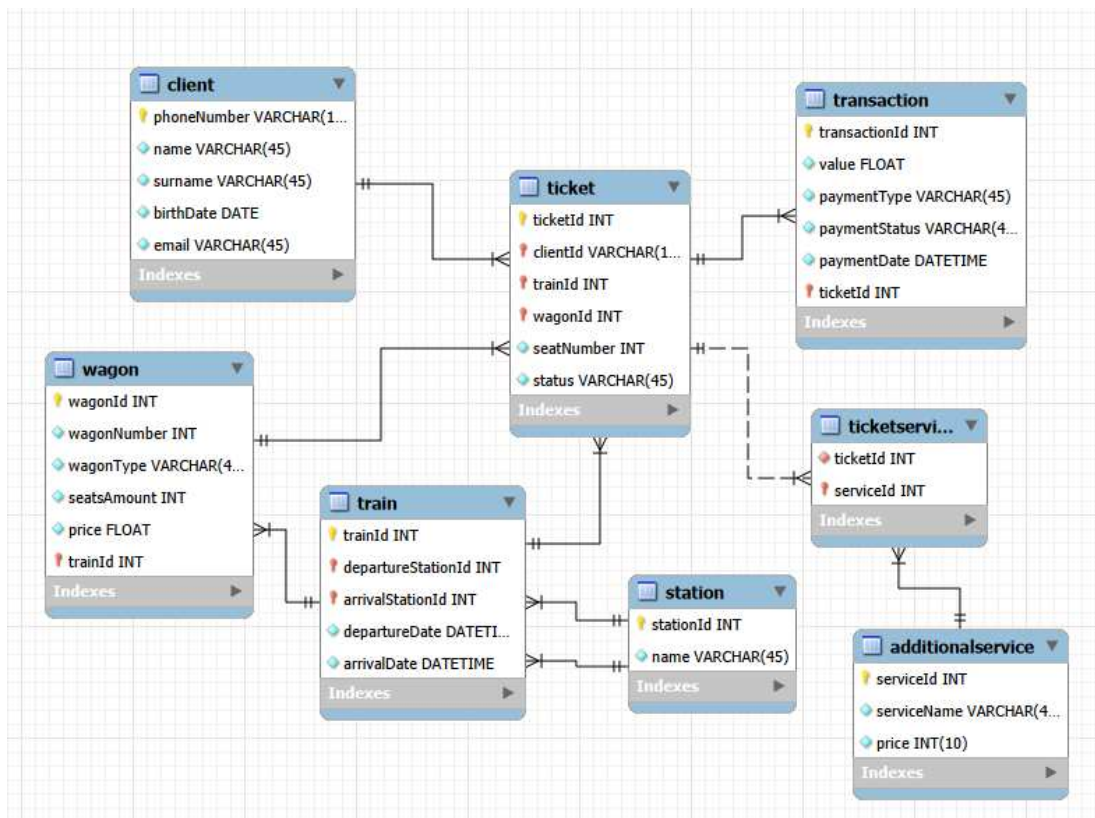


Рисунок 2.6. - Фізична модель БД

У підрозділі було розроблено інформаційне забезпечення системи бронювання залізничних квитків. Створено діаграми потоків даних, концептуальну, логічну, фізичну моделі БД та проведено нормалізацію структури. Після завершення проєктування БД необхідно перейти до розробки математичного забезпечення системи, зокрема розглянути алгоритми вирішення основних задач.

2. 3 Рішення з математичного забезпечення

Для розрахунку загальної вартості квитка в застосунку реалізовано функцію обчислення суми до сплати з урахуванням обраних додаткових послуг.

Функція перевіряє, чи користувач вибрав вагон — якщо ні, то загальна вартість квитка дорівнює 0. В іншому випадку за основу береться вартість відповідного типу вагона для обраного маршруту. Далі функція використовує цикл для перевірки наявності кожної із можливих додаткових послуг у квитку. Якщо ідентифікатор послуги міститься у списку обраних, її ціна додається до

загальної суми. Хук `useMemo` гарантує, що перерахунок відбувається лише при зміні об'єкта квитка або списку послуг [32].

```
const totalSum = useMemo(() => {  
  if (!ticket.wagon) return 0;  
  
  let sum = ticket.wagon.price;  
  services.map((service) => {  
    if (ticket.services.includes(service.serviceId))  
      sum += service.price;  
  });  
  
  return sum;  
}, [ticket, services]);
```

Такий підхід забезпечує коректне відображення вартості в реальному часі та робить неможливим розбіжності між обраними опціями та підсумковою сумою до сплати.

2. 4 Рішення з програмного забезпечення

Програмне забезпечення застосунку побудоване за допомогою фреймворку `Next.js` із використанням мови програмування `TypeScript`. Усі модулі системи розділені на логічні групи відповідно до їх призначення. Така архітектура додатку сприяє легкому та зручному масштабуванню у майбутньому.

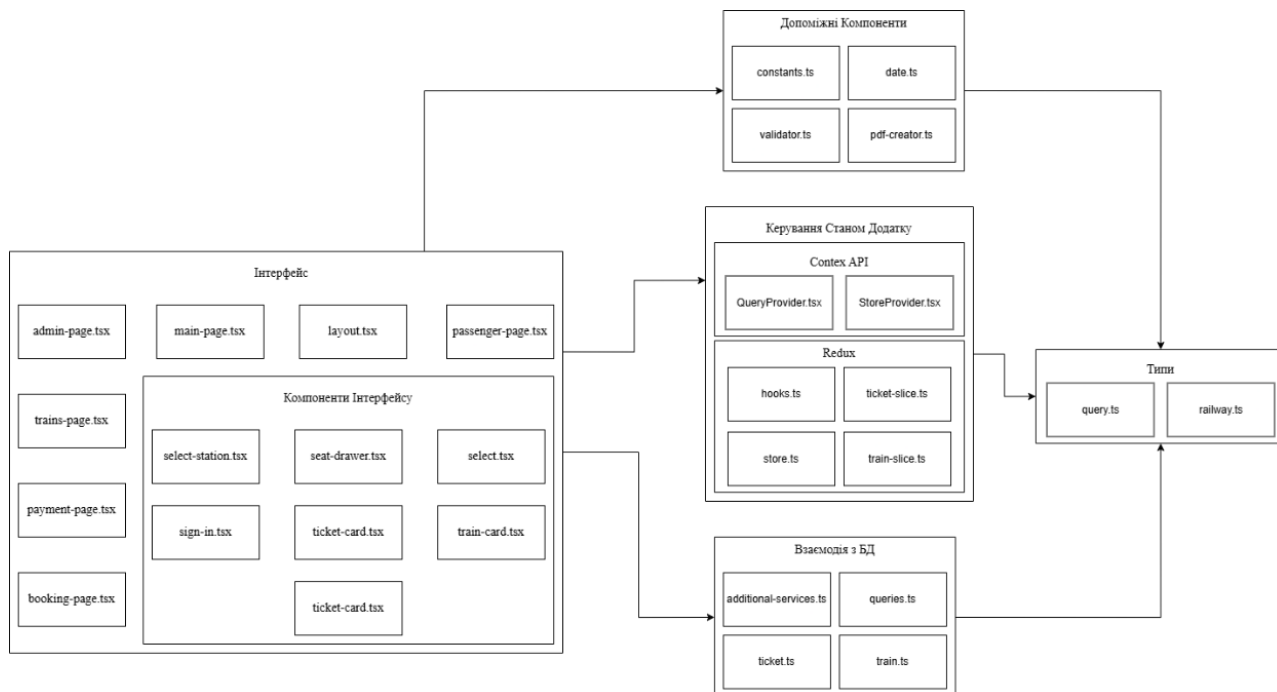


Рисунок 2.7 – Діаграма компонентів додатку

Як видно з вище наведеної діаграми, застосунок складається з п'яти основних компонентних груп.

Основні сторінки додатку формують користувацький інтерфейс і є точками входу для відповідних маршрутів застосунку. Кожна сторінка відповідає за певний етап взаємодії користувача із системою. Для побудови інтерфейсу сторінки використовують компоненти інтерфейсу, які відповідають за окремі візуальні блоки. Такий поділ дозволяє використовувати одні й ті самі компоненти на різних сторінках, зменшуючи дублювання коду.

Управління глобальним станом додатку реалізовано за допомогою бібліотеки Redux, модулі якої відповідають за оновлення та отримання інформації на базовому рівні додатку. Ці компоненти забезпечують доступ до даних на рівні всього застосунку, огортаючи дерево компонентів. Redux модулі відповідають за зберігання інформації, яка використовується для бронювання та купівлі квитків.

Взаємодія з БД здійснюється через окремі файли, що містять усі запити до серверної частини. Такий розподіл функціоналу та перенесення запитів до БД у окремий модуль спрощує їх повторне використання.

Допоміжні компоненти надають спільну функціональність — перевірку на правильність введених даних, форматування дат, генерацію PDF-документів із квитками та збереження постійних змінних додатку.

Збереження окремих типів і структур гарантує цілісність даних та їх перевірку на наявність необхідних властивостей.

2. 4. 1 Розробка інтерфейсу користувача

Інтерфейс для розробки додатку буде складатися із п'яти сторінок, попередній функціонал, необхідний для реалізації додатку, описано в пункті 1. 3. Візуальна частина додатку вміщує всі необхідні модулі для повноцінного функціонування додатку.

Сторінка профілю містить два стани: неавторизований та авторизований користувач. У неавторизованому стані відображається поле для введення номера телефону та кнопка «Увійти». Після введення номера користувач отримує доступ до свого профілю. В авторизованому стані відображається список квитків користувача. Кожен квиток представлений у вигляді картки, що містить маршрут і дату відправлення, час відправлення, місто призначення, номер вагона та місця. Поруч з кожним квитком розміщено згенерований QR-код для швидкого сканування під час контролю на посадці.



Рисунок 2.7. – 2.8 - Сторінка вибору рейсів

Сторінка вибору рейсів містить два списки: вибір місця відправлення й призначення; можливість вибору дати відправлення. Після заповнення необхідної інформації, користувач перейде на наступну сторінку з відображенням доступних рейсів на основі наданих даних.

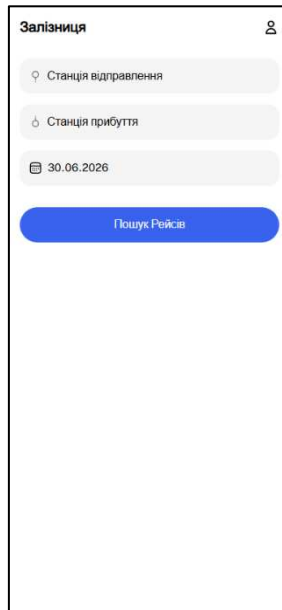


Рисунок 2.9. - Сторінка вибору рейсів

На етапі списку рейсів користувач має змогу побачити всі доступні рейси, що були відображені, враховуючи введену інформацію. Кожен із наданих рейсів має коротку інформацію про себе, а саме: час відправлення та час прибуття, загальний час подорожі та мінімальну ціну квитка.

Після натискання на рейс на сторінці відображаються всі доступні вагони. Серед наданих даних користувач може побачити тип та номер вагону, кількість доступних місць, а також вартість одного квитка.

Після вибору необхідного вагону, клієнт має змогу обрати номер місця у вікні методом прокручення списку доступних місць.

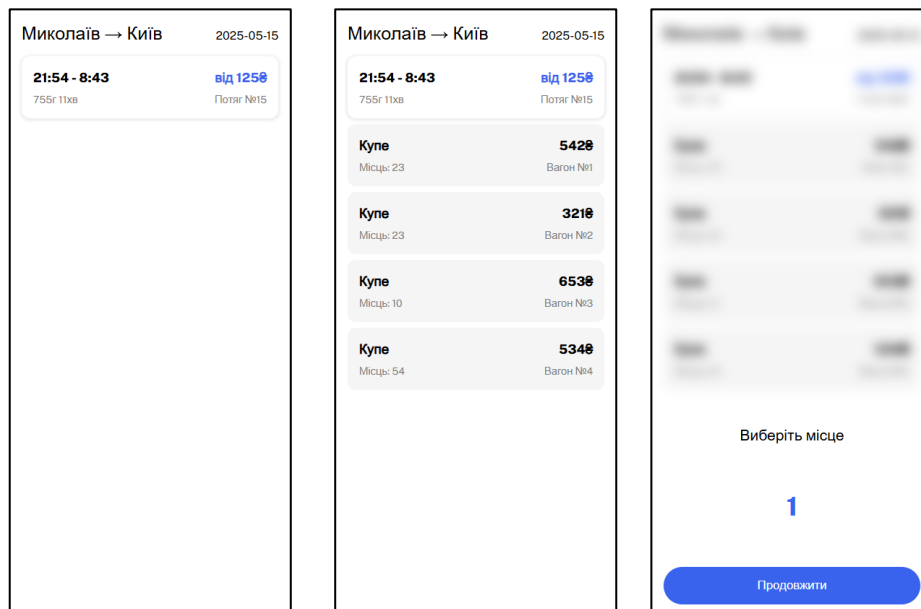


Рисунок 2.10. – 2.11. - Список доступних рейсів і вибір доступного місця

На сторінці заповнення особистих даних і вибір послуг клієнт повинен надати особисті дані для майбутнього формування квитка й збереження інформації. Форма містить поля для: ім'я, прізвища, номера телефона, дати народження й електронної пошти, на яку буде відправлено куплений квиток. Також користувач має можливість обрати додаткові послуги з наданого списку. Їхня сума буде додана до загальної вартості квитка для оплати.

The image shows a form for entering passenger information. It includes fields for name, surname, phone number, email, and date of birth. There are also checkboxes for additional services: 'Чай #30', 'Постільна білизна #80', and 'Обід #250'. A 'Продовжити' (Continue) button is located at the bottom of the form.

Рисунок 2.12. - Форма для особистих даних

Сторінка оплати надає змогу обрати один із можливих методів оплати, що охоплюють Google Pay, Apple Pay або за номером карти. Опісля успішної оплати клієнт отримає квиток на вказану електрону пошту.

The screenshot shows a mobile application interface for a payment screen titled "Оплата". It displays the following information: "Потяг №15" (Train No. 15), "Купе, Вагон 4, Місце 6" (Cabin, Car 4, Seat 6), "Відправлення" (Departure) from "Миколаїв" on "15.05.2025, 21:54:00", and "Прибуття" (Arrival) in "Київ" on "16.06.2025, 08:43:00". It also lists "Додаткові сервіси" (Additional services) as "Постільна білизна 880". Under "Спосіб оплати" (Payment method), "Apple Pay" is selected. At the bottom, there is a large blue button labeled "Сплатити 2614".

Рисунок 2.13. - Сторінка оплати

Користувач з адміністративними правами може використовувати сторінку адміністративної панелі для додавання та видалення рейсів, а також для отримання звітів активних рейсів. Для вибору станцій буде застосовуватися випадний список із переліком усіх доступних станцій.

The screenshot shows a mobile application interface for an administrative panel titled "Адміністративна панель". It contains several sections: "Отримання звіту" (Report generation) with a button "Отримати звіт для усіх активних потягів"; "Створення рейсу" (Train creation) with input fields for "Станція відправлення" (Departure station), "Станція прибуття" (Arrival station), "Дата відправлення" (Departure date), and "Дата прибуття" (Arrival date); and "Вагони" (Carts) with buttons "Додати вагон" (Add cart), "Видалити вагон" (Delete cart), and a large "Створити потяг" (Create train) button at the bottom.

Рисунок 2.14. - Сторінка адміністратора

2.4.2 Розробка клієнтської частини

Інтерфейс застосунку побудований на основі компонентного підходу, де кожен компонент відповідає за окрему частину інтерфейсу. Такий підхід забезпечує повторне використання коду та спрощує супровід застосунку.

У застосунку реалізовано такі сторінки:

- / – домашня сторінка для пошуку доступних рейсів за маршрутом і датою;
- /trains?from=&to=&date= – відображення списку доступних рейсів на основі заданих параметрів пошуку. Для пошуку використовуються параметри from, to, date, які уможливають сортування та пошук необхідних рейсів, відповідно запиту.
- /booking – сторінка бронювання квитка, використовується для збору даних про пасажирів та деталей подорожі;
- /payment – вибір методу оплати та проведення транзакції;
- /passenger – перегляд попередньо куплених квитків та можливість відміни бронювання;
- /admin – адміністративна сторінка для управління рейсами та перегляду звітності.

2.4.3 Взаємодія з БД

Для підключення до бази даних MySQL використовується бібліотека mysql2 [33]. Усі SQL запити, що використовуються додатком описані у Додатку А. З метою кращої організації коду всі запити до БД розподілено між окремими файлами відповідно до предметної області:

- queries.ts — централізований файл із усіма SQL-запитами, що використовуються в додатку;
- ticket.ts — логіка виконання запитів, пов'язаних з оформленням квитка, збереженням даних клієнта та обробкою оплати;

- train.ts — функції для створення нових рейсів і вагонів, а також пошуку доступних рейсів за заданими параметрами;
- additional-service.ts — взаємодія з таблицею додаткових послуг та їх прив'язка до квитка.

Така архітектура забезпечує чіткий розподіл відповідальності між модулями та спрощує подальше розширення функціональності.

Таким чином, у розділі розроблено програмне забезпечення інформаційної системи продажу залізничних квитків. Спроектовано інтерфейс користувача, що охоплює всі необхідні етапи взаємодії з системою. Клієнтська частина реалізована на основі компонентного підходу з чіткою маршрутизацією між сторінками. Взаємодія з базою даних організована через розділені модулі відповідно до предметної області, що забезпечує зручність супроводу та подальшого розширення системи.

2. 5 Рішення з технічного забезпечення

Для забезпечення коректної та стабільної роботи ІС необхідно визначити мінімальні вимоги для пристроїв користувачів та персоналу залізниці, а також для серверної інфраструктури.

2.5.1 Вимоги до пристроїв користувачів

Оскільки система є веб-застосунком, взаємодія з нею здійснюється виключно через браузер, це допомагає уникнути встановлення додаткового програмного забезпечення, а також, підвищує доступність додатку для різних категорій користувачів.

Таблиця 2.4.

ВИМОГИ ДО ПРИСТРОЇВ КОРИСТУВАЧІВ

Параметр	Комп'ютер / Ноутбук	Смартфон / Планшет
Операційна система	Windows 10+, macOS 10.15+, Ubuntu 20.04+ / Debian 11+	iOS 16.4+ або Android 11.0+ [34]

Оперативна пам'ять	Не менше 2 ГБ	Не менше 2 ГБ
Браузер	Chrome 111+, Edge 111+, Firefox 111+, Safari 16.4+ [35]	Safari (iOS), Chrome (Android) або інший сучасний браузер
Підключення	Wi-Fi, не менше 5 Мбіт/с	4G/LTE або Wi-Fi
JavaScript	Має бути увімкнено	Має бути увімкнено

Зазначені вимоги є мінімальними і задовольняються переважною більшістю сучасних пристроїв, що забезпечує доступність системи для широкого спектра пасажирів.

2.5.2 Вимоги до серверної частини

Серверна інфраструктура системи складається з двох логічних компонентів: сервера застосунку та сервера БД. Залежно від масштабу розгортання їх можна розмішувати як на одному фізичному або віртуальному сервері, так і на окремих вузлах.

Таблиця 2.5.

ВИМОГИ ДО СЕРВЕРНОЇ ІНФРАСТРУКТУРИ

Параметр	Сервер застосунку	Сервер БД
Операційна система	Ubuntu Server 22.04 LTS [35]	Ubuntu Server 22.04 LTS
Середовище виконання	Node.js 18.17+ [36]	—
Процесор	Мінімум 1 ядро	Мінімум 2 ядра [37]
Оперативна пам'ять	Мінімум 1 ГБ, рекомендовано, 2–4 ГБ (рекомендовано) [38]	Мінімум 1 ГБ, рекомендовано 2–4 ГБ (рекомендовано) [39]
Сховище	20 ГБ SSD, рекомендовано NVMe [40]	20 ГБ SSD

МЕРЕЖЕВІ ВИМОГИ

Параметр	Вимога
Пропускна здатність	Мінімум 100 Мбіт/с
Шифрування	SSL/TLS-сертифікат обов'язковий
IP-адреса	Статична, для прив'язки домену та платіжного шлюзу

Для початкового етапу розгортання регіональної системи обидва компоненти можуть функціонувати на єдиному сервері. У процесі масштабування програмного забезпечення та, відповідно, підвищення навантаження архітектура передбачає можливість розділення на окремі вузли без необхідності зміни у застосунку.

В ході проєктування було прийнято основні архітектурні та технологічні рішення, спроектовано та побудовано БД, розроблено інтерфейс користувача та реалізовано програмну частину ІС. Сукупність прийнятих рішень забезпечує надійну, масштабовану та зручну систему. Отримані результати дозволяють перейти до експлуатації застосунку та провести аналіз результатів використання системи.

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОЄКТУ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ПРОДАЖУ КВИТКІВ

3.1 Керівництво користувача системи

Для забезпечення повноцінної роботи ІС передбачено три групи взаємодії, кожна із власним набором інструментів та інструкцій:

- технічний персонал вокзалу, що відповідає за публікацію веб-застосунку, хостинг БД та налаштування середовища виконання;
- адміністративно-касовий персонал безпосередньо керує системою: продаж квитків для пасажирів із обмеженими технічними можливостями, обробка повернення квитків, управління рейсами та створення звітності;
- пасажир, що взаємодіє з веб-застосунком самостійно та проходить через увесь цикл бронювання та оплати квитка, у результаті отримуючи електронний квиток та, за необхідності, може оформити заявку на повернення квитка.

Детальні покрокові інструкції для кожної групи наведено у Додатку Б.

3.2 Результат використання системи

Розроблений додаток успішно досяг первинних цілей, забезпечуючи базовий функціонал, необхідний для виконання основних задач, таких як: пошук і бронювання квитків, отримання інформації про доступні маршрути. На цьому етапі система здатна задовольняти базові потреби користувачів, що свідчить про успішну реалізацію основного завдання. Однак, враховуючи зростаючі очікування користувачів щодо зручності, безпеки й персоналізації, додаток потребує подальшого вдосконалення для забезпечення успіху на ринку та серед конкурентів.

Упровадження кращої системи автентифікації через пошту або за номером телефону. Наразі для входу у персональний кабінет користувач повинен лише увести свій номер телефону, проте така система не є надійною та потребує

покращення. Для підвищення безпеки й зручності користувачів пропонується реалізувати систему автентифікації, яка охоплюватиме підтвердження особи за номером телефону чи за допомогою підтвердження через пошту. За допомогою нововведень, користувачу потрібно буде увести додатковий код, що буде відправлений на вказаний номер телефону, або на електронну адресу.

Для забезпечення більшої цінності для користувачів пропонується додати функціонал загальної статистики та звітів. Наприклад, користувачі могли б отримувати інформацію про кількість здійснених поїздок, витрачені кошти, найпопулярніші маршрути тощо. Ці дані можуть бути представлені у вигляді графіків або таблиць. Крім того, аналітичні звіти можуть бути корисними для бізнес-аналітики, дозволяючи адміністраторам додатку аналізувати поведінку користувачів.

Поточна організація створення й обробки локальних файлів потребує вдосконалення з метою полегшення майбутньої модернізації. Зокрема, можна переглянути архітектуру зберігання даних, оптимізувати структуру файлів у клієнтській частині.

Наразі для збереження даних користувача у БД використовується його номер телефону як первинний ключ. Проте така схема збереження не є повністю надійною. У майбутньому, користувач може змінити власний номер телефону або пошту, а отже прив'язка до таких атрибутів може вплинути на втрату доступу до даних користувача після зміни. Для вирішення цієї проблеми пропонується запровадити додатковий унікальний та незмінний атрибут – айді користувача.

Одним із ключових аспектів підвищення продуктивності додатку є зменшення кількості запитів до БД. Наразі вибір вагонів може генерувати надмірну кількість дрібних запитів, що знижує швидкість роботи системи й створює додаткове навантаження на сервер. Для вирішення цієї проблеми пропонується реалізувати запити, що повертають більший обсяг інформації, необхідної користувачеві, це зменшить затримки, навантаження на БД та підвищить загальну швидкість роботи додатку.

На основі проведеного аналізу констатуємо, що розроблений додаток має міцну основу для виконання базових функцій, але потребує подальшого розвитку. Впровадження кращої системи автентифікації сприятиме підвищенню безпеки та зручності для користувачів, тоді як аналітичні звіти дозволять як користувачам, так і адміністраторам отримувати цінну інформацію. Оптимізація взаємодії з БД шляхом зменшення кількості запитів, є критично важливими для підвищення продуктивності й масштабованості системи.

РОЗДІЛ 4 ОХОРОНА ПРАЦІ ТА ТЕХНІКА БЕЗПЕКИ

Охорона праці на залізничному транспорті регулюється Законом України «Про охорону праці» [41] та галузевими нормами, що встановлює Державна служба з питань транспортної безпеки України [42]. Відповідно до цих вимог, працівники всіх рівнів зобов'язані проходити інструктажі з безпеки, використовувати ЗІЗ та дотримуватися протоколів, що відповідають їхній посаді.

Таблиця 4.1

ТЕХНІКА БЕЗПЕКИ НА ПІДПРИЄМСТВІ

Посада	Обов'язки	Норми безпеки
Машиніст локомотива	Керування поїздом, дотримання сигналів і зв'язку з диспетчерами.	Дотримання графіку роботи для уникнення втоми (не більше 12 годин на зміну). Використання: шоломів, рукавичок. Навчання з питань правил руху потягів і реагування на надзвичайні ситуації.
Провідник потяга	Забезпечення безпеки пасажирів, контроль поведінки на борту.	Навчання з надання першої медичної допомоги. Знання евакуаційних процедур у разі аварій. Використання уніформи, світловідбиваючих жилетів під час роботи на перонах.
Станційний майстер	Координація руху поїздів, контроль сигналізації.	Дотримання правил сигналізації та зв'язку. Регулярне навчання з безпеки руху.
Робітники колії	Ремонт й обслуговування	Використання шоломів, рукавичок, та, світловідбиваючих жилетів.

	колій, сигнальних систем.	Дотримання протоколів безпеки під час наближення поїздів (сигналізація, зупинка робіт). Навчання з електробезпеки у процесі роботи з контактними мережами.
Сигналісти та зв'язківці	Обслуговування сигнальних систем і зв'язку.	Дотримання електробезпеки під час роботи з обладнанням. Регулярне тестування сигнальних систем. Використання захисних окулярів та рукавичок.
Механічні та електричні робітники	Ремонт й обслуговування локомотивів, вагонів, обладнання.	Дотримання процедур блокування/маркування під час роботи з механізмами. Використання захисних окулярів, рукавичок та спец-одягу. Навчання з безпеки у процесі роботи з електрикою й важкими механізмами.
Адміністративний персонал	Управлінська робота в офісах, координація діяльності.	Дотримання правил пожежної безпеки (наявність вогнегасників, евакуаційних планів). Використання ергономічного обладнання (стілці, столи). Навчання з безпеки в офісі.
ІТ-спеціалісти	Підтримка інформаційних систем, захист даних.	Дотримання стандартів кібербезпеки (ISO/IEC 27001). Регулярне створення резервних копій даних.

		Використання антивірусного програмного забезпечення.
Служба безпеки	Патрулювання, контроль доступу, реагування на інциденти.	Дотримання протоколів патрулювання й звітності. Навчання з реагування на надзвичайні ситуації. Використання уніформи та світловідбивачів.

З таблиці 4.1 видно, що вимоги до безпеки суттєво відрізняються залежно від характеру роботи. Персонал, що працює безпосередньо з рухомим складом і колійною інфраструктурою має найширший перелік вимог, включно з обов'язковим використанням ЗІЗ та регулярним навчанням з електробезпеки. Для адміністративного персоналу та ІТ-спеціалістів пріоритетними є організаційні заходи: пожежна безпека, ергономіка робочого місця та дотримання стандартів кібербезпеки.

Таким чином, система охорони праці на залізничному підприємстві є багаторівневою і не може бути описана єдиним універсальним протоколом. Кожна описана посада має власний набір вимог, що відповідає специфіці її діяльності.

ВИСНОВКИ

У кваліфікаційній роботі розроблено ІС для продажу залізничних квитків, спрямовану на автоматизацію процесів обслуговування пасажирів на регіональному залізничному вокзалі в умовах цифрової трансформації транспортних перевезень.

У процесі виконання роботи проведено повний цикл створення системи. На першому етапі детально описано та проаналізовано структуру залізничного вокзалу та існуючі бізнес-процеси продажу квитків. Виявлено основні недоліки: тривалі черги до кас, відсутність можливості дистанційного бронювання та придбання квитків, повна залежність від касового персоналу, використання друкованих чеків та квитків, а також значний вплив людського фактора на роботу системи. Перелічені проблеми суттєво знижують якість обслуговування пасажирів і конкурентоспроможність залізничного транспорту.

На основі проведеного аналізу сформовано вимоги до ІС з урахуванням потреб різних категорій користувачів – пасажирів, котрі розділені на дві групи в залежності від технічних навичок та можливостей та адміністративно-касового персоналу вокзалу. Розроблено концепцію системи, загальносистемні рішення та архітектуру.

Основними науково-практичними результатами роботи є:

- Спроектовані ДПД різних рівнів, концептуальна, логічна та фізична моделі БД за допомогою СУБД MySQL з проведенням нормалізації структури.
- Реалізований сучасний веб-додаток на базі технологічного стеку Next.js 15, TypeScript, Redux та MySQL, що забезпечує високу продуктивність, зручність підтримки та потенціал масштабування.
- Повноцінна реалізація ключових функціональних можливостей: пошук і перегляд рейсів, вибір вагона та місця, бронювання, онлайн-оплата, генерація електронних квитків з QR-кодом, обробка повернень, а також адміністративне управління розкладом, вагонами та звітністю.

Розроблена ІС дозволяє значно скоротити час обслуговування пасажирів, зменшити навантаження на касовий персонал, зменшити кількість помилок під час купівлі залізничних квитків та забезпечити цілодобовий доступ до функціоналу залізниці. Важливою перевагою нової системи є збереження базового обслуговування через касу, що робить систему доступною для всіх категорій населення.

Хоча система на поточному етапі повністю виконує поставлені завдання, визначено перспективні напрямки її розвитку: удосконалення механізмів автентифікації, оптимізація запитів до бази даних, а також розширення шляхів для аналітики.

Таким чином, мета роботи була досягнута в повному обсязі. Створений програмний продукт може бути використаний як готове рішення або як основа для подальшої модернізації системи продажу квитків на регіональних залізничних вокзалах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Корнійчук М., Липовець Н., Шамрай Д. Технологія галузі і технічні засоби залізничного транспорту. К. 2006. 500 с. URL: <https://studfile.net/preview/13144966/>
2. Hlavatý J., Ližbetin J. Comparison of Organisational Structures of Selected Passenger Railway Transport Companies and their Impact on Strategic Goals of Companies. *LOGI – Scientific Journal on Transport and Logistics*, September 2023. 238 p. URL: https://www.researchgate.net/publication/374090193_Comparison_of_Organisational_Structures_of_Selected_Passenger_Railway_Transport_Companies_and_their_Impact_on_Strategic_Goals_of_Companies
3. Jean Rey. Adapting to Future Society. International Union of Railways, Paris, October 2013. 72 p. URL: https://uic.org/IMG/pdf/2013_nextstation_railway_stations_adapting_to_future_society.pdf
4. Huisman D., Kroon L., Lentink R., Vromans M. Operations Research in Passenger Railway Transportation. Econometric Institute Report, 2005. 33 p. URL: <https://repub.eur.nl/pub/1941/ei200516.pdf>
5. Lunardona A., Vladimirova D., Boucseina B..How railway stations can transform urban mobility and the public realm: The stakeholders' perspective. *Journal of Urban Mobility*, December 2023. 10 p. URL: <https://www.sciencedirect.com/science/article/pii/S2667091723000031>
6. Xie M., Chena F., Heb H., Yuanc M., Yangd L. Research on Innovative Design of railway passenger service. E3S Web of Conferences 2023. 5 p. URL: https://www.e3s-conferences.org/articles/e3sconf/pdf/2023/09/e3sconf_icuems_2023_01010.pdf
7. Тестов П. Укрзалізниця: пунктуальна, але незручна. Основні проблеми і що робити, щоб врятувати пасажирські перевезення. *TEXTY.ORG.UA*, Січень 2019. URL: <https://texty.org.ua/articles/90697>
8. Kaustubh S., Siddhesh A., Aditya A., Vishal A., Nachiket B.. Next-Gen Railway Ticketing: ARTS with Real-Time QR Code Validation. *International Scientific*

- Journal of Engineering and Management*, May 2025. 10 p. URL: <https://isjem.com/download/next-gen-railway-ticketing-arts-with-real-time-qr-code-validation/>
9. Kavishka R. Undergraduate Build Diary #1 - A Ticket Booking System. August 2024. URL: <https://medium.com/@ravindukd/undergraduate-build-diary-1-a-ticket-booking-system-daad712549d6>
 10. Waghmare A., Pavate A. QR code-based Railway e-Ticket. *IOSR Journal of Engineering*, May 2019. 8 p. URL: https://www.researchgate.net/publication/333344979_QR_code_based_Railway_e-Ticket
 11. Erickson J. MySQL: Understanding What It Is and How It's Used. August 2024. URL: <https://www.oracle.com/mysql/what-is-mysql/>
 12. Advantages of Using MySQL. July 2017. URL: <https://devops.com/8-advantages-using-mysql/>
 13. Liberatore A. The Magic of ACID: How Databases Keep Your Information Safe. July 2024. URL: <https://medium.com/@liberatoreanita/the-magic-of-acid-how-databases-keep-your-information-safe-4c2c06e89a8b>
 14. Castro S. 5 Reasons Why MySQL Is Still The Go-To Database Management System. April 2022. URL: <https://www.jobsity.com/blog/5-reasons-why-mysql-is-still-the-go-to-database-management-system>
 15. Ashila A. How to Ensure Server Scalability for High Performance and Growth. URL: <https://www.supportpro.com/blog/how-to-ensure-server-scalability/>
 16. Rizzolo R., Foote T., Crafts J., Grosch D., Leung T., Lund D. IBM System z9 eFUSE applications and methodology. *IBM Journal of Research and Development*, January 2007. 75 p. URL: <https://ieeexplore.ieee.org/document/5388714>
 17. Building Scalable Web Applications: Best Practices for 2025. January 2025. URL: <https://anadea.info/blog/building-scalable-web-applications-2025/>
 18. Piotr. JavaScript vs TypeScript in 2024. 2024. URL: <https://houseofangular.io/javascript-vs-typescript-in-2024/>

- 19.Fateh D. Nuxt vs. Next.js: Two popular JavaScript frameworks compared. November 2024. URL: <https://www.contentful.com/blog/nuxt-next-js-compared/>
- 20.Khambholja M. Nuxt vs Next: A Comparative Analysis for Modern Web Development. April 2024. URL: <https://www.openxcell.com/blog/nuxt-vs-next/>
- 21.Shubham Singh R. React State Management: Redux vs Context API. April 2021. URL: <https://www.oodlestechnologies.com/dev-blog/react-state-management:-redux-vs-context-api/>
- 22.Sandu B. React Context vs Redux: What to Use When. February 2026. URL: <https://tms-outsource.com/blog/posts/react-context-vs-redux/>
- 23.Abelhamid M. Redux vs. Context.Provider: Choosing State Management in React Applications. August 2024. URL: <https://dev.to/dosht/redux-vs-contextprovider-choosing-state-management-in-react-applications-6bm>
- 24.Структурний аналіз потоків даних. URL: https://mte.nure.ua/pdf/studying/zov_lk8_tipproces.pdf.
- 25.Rosziati I., Siow Yen Yen. Formalization of the Data Flow Diagram Rules for Consistency Check. *International Journal of Software Engineering & Applications*, October 2010. 17 p. URL: https://www.researchgate.net/publication/47619337_Formalization_of_the_Data_Flow_Diagram_Rules_for_Consistency_Check
- 26.Костенко Б., Гавриленко І. Організація баз даних та знань: конспект лекцій (для студентів денної і заочної форм навчання першого (бакалаврського) рівня вищої освіти за спеціальністю 126 – Інформаційні системи та технології). Х.: ХНУМГ ім. О. М. Бекетова, 2021. 92 с. URL: <https://eprints.kname.edu.ua/60505/>
- 27.Badia A. Entity-Relationship modeling revisited. March 2004. 82 p. URL: https://www.researchgate.net/publication/220415410_Entity-Relationship_modeling_revisited
- 28.Полотай О. Логічне Проектування Баз Даних. URL: <https://virt.ldubgd.edu.ua/mod/resource/view.php?id=13330>

29. Nguyen L. A Brief Guide to Database Normalization. July 2023.
URL: <https://medium.com/@ndleah/a-brief-guide-to-database-normalization-5ac59f093161>.
30. URL: https://en.wikipedia.org/wiki/First_normal_form.
31. Description of the database normalization basics. June 2024.
URL: <https://learn.microsoft.com/en-us/office/troubleshoot/access/database-normalization-description>.
32. URL: <https://react.dev/reference/react/useMemo>
33. URL: <https://sidorares.github.io/node-mysql2/docs>
34. URL: https://myip.ms/view/comp_browsers/11132/Google_Chrome_111.html
35. URL: <https://nextjs.org/docs/architecture/supported-browsers>
36. URL: <https://nextjs.org/docs/app/getting-started/installation>
37. On-Premise MySQL Server Hardware Requirements. February 2020. URL:
<https://docs.webchartnow.com/resources/system-specifications/on-premise-deployment/on-premise-mysql-server-hardware-requirements/>
38. Node.js VPS Hosting: What You Need to Know Before Deploying Your App. December 2025. URL: <https://vps.us/blog/nodejs-vps-hosting/>
39. How MySQL Uses Memory. URL:
<https://dev.mysql.com/doc/refman/8.0/en/memory-use.html>
40. Thameem A. VPS Hosting for Node.js: Best Options & Setup Guide. February 2026. URL: <https://skynethosting.net/blog/vps-hosting-for-node-js/>
41. Закон України про охорону праці. Грудень 2025. URL:
<https://zakon.rada.gov.ua/laws/show/2694-12#Text>
42. URL: <https://dsbt.gov.ua>

Інформація про усі квитки клієнта:

```
SELECT
    ticket.*,
    wagon.wagonNumber,
    train.departureDate,
    train.arrivalDate,
    dep.name AS departureStation,
    arr.name AS arrivalStation
FROM ticket
JOIN train USING(trainId)
JOIN station AS dep ON train.departureStationId = dep.stationId
JOIN station AS arr ON train.arrivalStationId = arr.stationId
JOIN wagon ON ticket.wagonId = wagon.wagonId
WHERE ticket.clientId = ?
ORDER BY train.departureDate;
```

Повна інформація про потяг:

```
SELECT
    t.*,
    s_dep.name AS departureStation,
    s_arr.name AS arrivalStation
FROM train t
LEFT JOIN station s_dep ON t.departureStationId = s_dep.stationId
LEFT JOIN station s_arr ON t.arrivalStationId = s_arr.stationId
WHERE t.trainId = ?;
```

Вибір усіх вільних вагонів та кількості вільних місць для потяга:

```
SELECT
    w.*,
    (w.seatsAmount - COUNT(tk.ticketId)) AS freeSeats
FROM wagon w
LEFT JOIN ticket tk ON w.wagonId = tk.wagonId
WHERE w.trainId = ?
GROUP BY w.wagonId
HAVING freeSeats > 0;
```

Пошук усіх активних потягів із відображенням найнижчої ціни у рейсі:

```

SELECT
    t.trainId,
    t.departureDate,
    t.arrivalDate,
    MIN(w.price) AS startingPrice,
    s_dep.name AS departureStation,
    s_arr.name AS arrivalStation
FROM train t
JOIN station s_dep ON t.departureStationId = s_dep.stationId
JOIN station s_arr ON t.arrivalStationId = s_arr.stationId
JOIN wagon w ON t.trainId = w.trainId
WHERE s_dep.name = ?
    AND s_arr.name = ?
    AND DATE(t.departureDate) = ?
    AND t.departureDate > NOW()
    AND w.seatsAmount > (
        SELECT COUNT(*)
        FROM ticket tk
        WHERE tk.wagonId = w.wagonId
    )
GROUP BY
    t.trainId,
    t.departureDate,
    t.arrivalDate,
    s_dep.name,
    s_arr.name;

```

Відображення усіх вільних місць за номерами:

```

WITH RECURSIVE allSeats AS (
    SELECT 1 AS seatNumber
    UNION ALL
    SELECT seatNumber + 1
    FROM allSeats
    WHERE seatNumber < (
        SELECT seatsAmount FROM wagon WHERE wagonId = ?
    )
)
SELECT seatNumber
FROM allSeats
WHERE seatNumber NOT IN (
    SELECT seatNumber
    FROM ticket
    WHERE wagonId = ? AND status != 'Скасовано'
)
ORDER BY seatNumber;

```

Отримання загального звіту з кількістю проданих квитків для рейсу:

```
SELECT
    w.wagonNumber,
    w.wagonType,
    w.seatsAmount,
    COUNT(tk.ticketId) AS seatsBought
FROM wagon w
LEFT JOIN ticket tk ON w.wagonId = tk.wagonId AND tk.status != 'Скасовано'
WHERE w.trainId = ?
GROUP BY w.wagonId, w.wagonNumber, w.wagonType, w.seatsAmount
ORDER BY w.wagonNumber ASC;
```

Для розробника:

- Клонувати репозиторій – <https://github.com/AndRockSYS/railway-app>
- Встановити СУБД та супроводжуючі додатки для повноцінного функціонування
- Створити БД за допомогою наявного скрипту із репозиторію
- Розгорнути сайт та підключити домен за необхідності
- Змінити змінні середовища для коректного підключення до сайту

Для адміністратора:

- Відкрити сторінку адміністратора за посиланням – /admin
- Створити новий рейс за допомогою наявної форми на сторінці
- Отримати звіт із продажом квитків з усіх актуальних рейсів

Для пасажирів:

- Відкрити головну сторінку веб-додатку
- За необхідності виконати вхід у власний кабінет за поштою
- Вибрати станцію відправлення, дату відправлення та прибуття виконати пошук
- Серед наявного переліку рейсів обрати необхідний
- У меню вибору місця прогорнути до потрібного номеру
- Після вибору вагону та номеру місця, потрібно ввести персональні дані пасажирів та вибрати додаткові послуги за необхідності
- Підтвердження перенесе користувача на сторінку із перевіркою даних про квиток після якої користувач повинен буде вибрати спосіб оплати та сплатити повну вартість квитка
- Новостворений квиток буде наявний у кабінеті користувача