

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра інформаційних систем та технологій**

«На правах рукопису»  
УДК 004.XXX

До захисту допущено:  
Завідувач кафедри  
\_\_\_\_\_ Олександр РОЛІК  
«\_\_» \_\_\_\_\_ 2024 р.

**Магістерська дисертація**

**на здобуття ступеня магістра**

**за освітньо-професійною програмою  
«Інтегровані інформаційні системи»**

**зі спеціальності 126 «Інформаційні системи та технології»**

**на тему: «Еквайрингова платформа з використанням штучного  
інтелекту»**

Виконав:

студент 2 курсу, групи ІА-31мп  
Гулящий Ігор Сергійович \_\_\_\_\_

Керівник:

доцент каф. ІСТ, к.т.н., доц.

Писаренко Андрій Володимирович \_\_\_\_\_

Рецензент:

професор каф. ІПІ, д.т.н., проф.  
Мельник Микола Михайлович \_\_\_\_\_

Засвідчую, що у цій магістерській  
дисертації немає запозичень з праць  
інших авторів без відповідних  
посилань.

Студент \_\_\_\_\_

Київ – 2024 року

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**  
**Факультет інформатики та обчислювальної техніки**  
**Кафедра інформаційних систем та технологій**

Рівень вищої освіти – другий (магістерський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інтегровані інформаційні системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Олександр РОЛІК

«\_\_» \_\_\_\_\_ 2024 р.

**ЗАВДАННЯ**  
**на магістерську дисертацію студенту**  
**Гулящому Ігорю Сергійовичу**

1. Тема дисертації «Еквайрингова платформа з використанням штучного інтелекту», науковий керівник дисертації Писаренко Андрій Володимирович, доц., затверджені наказом по університету від «08» 11 2024 р. № 5016-с
2. Термін подання студентом дисертації «09» 12 2024 р.
3. Об'єкт дослідження: **певна система, обладнання, пристрій, процес, технологія, програмний продукт, інформаційна технологія, інтелектуальний твір, явище, економічна діяльність тощо, що породжує проблемну ситуацію і обране для дослідження.**
4. Вихідні дані: **технічні вимоги, яким має відповідати система (наприклад, швидкість відповіді до 1 секунди, підтримка до 80 запитів у хвилину, відсутність передачі персональних даних у інформаційну систему).**
5. Перелік завдань, які потрібно розробити: **провести аналіз існуючих рішень, спроектувати архітектуру системи, реалізувати бек-енд та фронт-енд, провести тестування, підготувати текстову та графічну частину пояснювальної записки**
6. Орієнтовний перелік графічного (ілюстративного) матеріалу: **структурна схема, діаграма варіантів використання, діаграма діяльності, ER-діаграма і т.п.**

7. Орієнтовний перелік публікацій: **назва статті або тези конференції (за наявності)**

8. Дата видачі завдання 02.09.2024 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.			
2.			
3.			
4.			
5.			
6.			
7.			
8.			

Студент

Ігор ГУЛЯЦЬКИЙ

Науковий керівник

Андрій ПИСАРЕНКО

## РЕФЕРАТ

Інформаційна система підтримки прийняття рішень в агротехнічній галузі:  
111 с., 18 табл., 16 рис., 9 дод., 26 джерел.

КЛЮЧОВЕ СЛОВО 1, КЛЮЧОВЕ СЛОВО 2, КЛЮЧОВЕ СЛОВО 3, ..., КЛЮЧОВЕ СЛОВО 10.

Обсяг реферату не менше половини та не більше аркуша А4, а його текст має містити наступні компоненти:

- актуальність теми (розкриття сутності та стану розв'язування наукової проблеми (задачі) та її актуальності й значущості для розвитку відповідної галузі науки чи виробництва, обґрунтування доцільності проведення дослідження);

- зв'язок дослідження з науковими програмами, планами, темами (висвітлення зв'язку вибраного напрямку досліджень з планами науково-дослідних робіт кафедри, а також з галузевими та (або) державними планами та програмами. Обов'язково зазначають номери державної реєстрації науково-дослідних робіт, а також і роль автора у виконанні цих науково-дослідних робіт);

- мета і задачі дослідження (формулювання мети роботи і задачі, які необхідно вирішити для досягнення поставленої мети (не слід формулювати мету як «дослідження...», «вивчення...» тощо, тому що ці слова вказують на засіб досягнення мети, а не на саму мету). Мета – це запланований результат дослідження. Виконуючи наукову роботу слід пам'ятати, що метою будь-якої наукової праці є виявлення нових фактів, висновків, рекомендацій, закономірностей або ж уточнення відомих раніше, але недостатньо досліджених. Отримати заплановані результати, поступово досягти поставленої мети можна шляхом її деталізації у вигляді певної програми цілеспрямованих дій – завдань дослідження. Завдання дослідження формулюються в двох варіантах: перший – у вигляді самостійно закінчених етапів дослідження; другий – як послідовне вирішення окремих проблем наукового дослідження по відношенню до загальної проблеми всієї магістерської дисертації. Формулювати і конкретизувати завдання

слід дуже ретельно, оскільки опис їх вирішення становить зміст підрозділів кожного з розділу дисертації);

– об’єкт дослідження (визначення об’єкта та предмета дослідження як категорій наукового процесу. Об’єкт дослідження – це певна система, обладнання, пристрій, процес, технологія, програмний продукт, інформаційна технологія, інтелектуальний твір, явище, економічна діяльність тощо, що породжує проблемну ситуацію і обране для дослідження);

– предмет дослідження (предметом дослідження є певні властивості, характеристики об’єкта на які безпосередньо спрямовано само дослідження, оскільки предмет дослідження визначає тему дисертації, яка визначається на титульному аркуші);

– методи дослідження (подання переліку використаних методів дослідження для досягнення поставленої в роботі мети. Перераховувати їх треба не відірвано від змісту роботи, а коротко та змістовно визначаючи, що саме досліджувалось тим чи іншим методом. Це дасть змогу пересвідчитися в логічності та прийнятності вибору саме цих методів);

– наукова новизна одержаних результатів (\*обов’язково для освітньо-наукової програми\* подають коротку анотацію нових здобутків (рішень, висновків), одержаних магістрантом особисто. Необхідно показати відмінність отриманих результатів від відомих раніше, підкреслити ступінь новизни);

– практичне значення одержаних результатів (подання відомостей про застосування результатів досліджень або рекомендацій щодо їхнього впровадження);

– апробація результатів дисертації (вказується, на яких наукових конференціях, семінарах оприлюднені результати досліджень, що включені до дисертації);

– публікації (\*за наявності\* зазначається, в яких статтях у наукових журналах, збірниках наукових праць, матеріалах і тезах конференцій, патентах опубліковані результати дисертації).

## ABSTRACT

Acquiring platform using artificial intelligence: 111 p., 18 tab., 16 draw., 9 app., 26 sources.

Keywords: ACQUIRING PLATFORM, ARTIFICIAL INTELLIGENCE, PCI DSS, E-COMMERCE, INVOICES, PAYMENTS, .

The object of development is web-server for management e-commerce payments and front-end web-interfaces for managing merchant's payments or administrate clients by own web-panel.

The purpose of development is to make safer e-commerce businesses by protecting them from fraud with best user experience while paying.

As a result the goal was achieved, acquiring platform was designed and developed. The server-side has its own open API, but in the same time has its own payment page, for businesses that do not have their own sites.

## ЗМІСТ

ПЕРЕЛІК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ .....	9
ВСТУП.....	10
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	12
1.1 Змістовний опис і аналіз предметної області.....	12
1.2 Аналіз існуючих рішень.....	12
1.2.1 Платформа LiqPay.....	12
1.2.2 Платформа WayForPay .....	14
1.2.3 Платформа UAPAY .....	15
1.3 Аналіз різновидів платежів.....	16
1.3.1 Різновиди e-commerce платежів .....	17
1.4 Аналіз підтвердження платежу .....	17
1.5 Аналіз PCI DSS сертифікації .....	19
Висновки до розділу 1 .....	21
2 ВИБІР ТЕХНОЛОГІЙ ТА ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ.....	22
2.1 Архітектура серверної частини .....	22
2.1.1 Шаблони проєктування.....	23
2.1.2 Вибір шини повідомлень.....	24
2.3 Бази даних.....	25
2.3.1 Вибір бази даних.....	25
2.3.2 Проєктування структури бази даних .....	26
2.4 Вимоги згідно PCI DSS сертифікації.....	35
2.3.1 Вимоги до способу зберігання чутливої інформації .....	35
2.3.2 Вимоги до адміністрування .....	36
2.3.3 Вимоги до інтеграції з еквайринговою платформою .....	36
2.4 Вибір технологій для роботи зі штучним інтелектом .....	37
2.5 Вибір мови програмування для серверної частини .....	38
2.6 Бібліотеки для платформи Node.js .....	38
2.6.1 Фреймворк веб-серверу.....	39

	8
2.6.2 Бібліотека ORM для роботи з базою даних.....	41
2.6.3 Авторизація та аутентифікація.....	43
2.6.4 Другорядні бібліотеки.....	44
2.7 Вибір технологій для інтерфейсів.....	45
Висновки до розділу 2.....	45
3 ОПИС СТРУКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	47
3.1 Структура мікросервісів серверної частини.....	47
3.1.1 Опис мікросервісу API-Gateway.....	47
3.1.2 Опис мікросервісу InvoiceManager.....	49
3.1.3 Опис мікросервісу AIModel.....	51
3.1.4 Опис мікросервісу Reporter.....	51
3.1.5 Опис мікросервісу AdminManager.....	51
3.2 Функціональні вимоги до серверної частини.....	51
3.3. Функціональні вимоги до інтерфейсів.....	51
3.3.1 Функціональні вимоги до платіжної сторінки.....	51
3.3.2 Функціональні вимоги до адміністративної панелі.....	51
3.3.3 Функціональні вимоги до особистого кабінету клієнта.....	51
4 ТЕСТУВАННЯ.....	52
5 РОЗРОБЛЕННЯ СТАРТАП ПРОЄКТУ.....	53
5.1 Опис ідеї проєкту.....	53
5.3 Технологійний аудит ідеї проєкту.....	56
5.4 Розроблення ринкової стратегії проєкту.....	66
Висновки до розділу 5.....	76
ВИСНОВКИ.....	77
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	79



## ПЕРЕЛІК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

PCI DSS – Payment Card Industry Data Security Standard;

БД – База даних;

P2P – Person to Person;

3DS – 3-D Secure;

API – Application Programming Interface;

CVV – Card Verification Value.

RPC – Remote Procedure Call

HTTP – HyperText Transfer Protocol

ORM – Object-relation Mapping

JWT – JSON Web Token

DI – Dependency Injection

## ВСТУП

У сучасному бізнес-середовищі електронна комерція стає невід'ємною складовою діяльності компаній різного масштабу. З розвитком інформаційних систем і поширенням інтернету все більше підприємств переходять до онлайн-продажів, задовольняючи попит споживачів на зручні й швидкі платежі. У цьому контексті особливо важливими стають технології, що дозволяють забезпечити безпечний і ефективний процес обробки фінансових транзакцій. Онлайн-платформам потрібна сертифікація та проходження аудиту для проведення платежів, а також інтеграція з банками-еквайрами для виконання фінансових операцій.

Еквайрингові платформи забезпечують обробку транзакцій і інтеграцію з банками. Вони значно спрощують процес роботи з платежами, допомагаючи бізнесу різного масштабу отримувати оплату від клієнтів у зручний та безпечний спосіб. Окрім технічної інтеграції, еквайрингові платформи сприяють оптимізації бухгалтерської звітності, оскільки беруть на себе ведення фінансової документації, що полегшує процес оподаткування та контроль доходів.

Малий бізнес також отримує значні переваги від використання таких платформ. Підприємства, де працює лише одна або кілька осіб, можуть легко інтегрувати еквайрингові рішення в свою діяльність, відкривши особистий рахунок у банку та підписавши договір з еквайринговою платформою. Це забезпечує легалізацію доходів і спрощує фінансову звітність. Крім того, еквайрингові платформи дозволяють вибирати найвигідніші умови серед банків-еквайрів, залежно від обсягу транзакцій, кількості платежів або підтримки платіжних систем, таких як Google Pay і Apple Pay.

Одним із основних завдань сучасних еквайрингових платформ є забезпечення безпеки фінансових операцій. Технології штучного інтелекту відіграють усе більшу роль у цьому процесі, дозволяючи відслідковувати підозрілу активність, запобігати шахрайству та контролювати коректність транзакцій.

Штучний інтелект здатен працювати в режимі реального часу, забезпечуючи вищий рівень надійності й ефективності еквайрингових рішень.

Еквайрингова платформа допоможе підприємствам будь-якого масштабу безпечно й ефективно працювати в електронній комерції, що стане важливим кроком до підвищення їхньої конкурентоспроможності та зручності для клієнтів.

Тож в ході розроблення будуть вирішуватись наступні задачі:

- а) створення єдиної централізованої платформи для різних варіантів використання;
- б) створення платіжної сторінки, яку клієнти зможуть використовувати в якості отримання оплати за послуги, які останні надають;
- в) створення фрейму, який можна інтегрувати в існуючі веб-рішення клієнтів;
- г) створення веб-інтерфейсу для надання послуг еквайринговою платформою клієнтам, що допоможе вести бухгалтерію, а також контролювати доходи, витрати на комісію, а також в ручному форматі використовувати систему, для створення посилань на оплату послуг.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Змістовний опис і аналіз предметної області

Еквайрингова платформа є важливим компонентом сучасної фінансової інфраструктури, забезпечуючи обробку платежів для торговців і підприємств шляхом інтеграції з платіжними системами та банками. Еквайринг стосується процесу прийняття та обробки безготівкових платежів, що здійснюються за допомогою платіжних карт. Еквайрингова платформа виступає посередником між торговцями, які приймають оплату картами, та банками-еквайрами, які здійснюють обробку фінансових операцій.

Враховуючи, що кількість бізнесів зростає, а Україна все більше і більше цифровізується, з'являються нові схеми обходу системи оподаткування, незаконного збагачення, відмивання грошей, цифрових крадіжок. Еквайрингова платформа є невід'ємною частиною кожного бізнесу, який пов'язаний з продажами, тож ці схеми частково стосуються і цієї ніші. Розглянемо найпопулярніші еквайрингові платформи, задля аналізу їх системи безпеки. Також виконаний огляд функціоналу і можливостей цих систем, щоб зрозуміти переваги та недоліки в конкурентному середовищі.

### 1.2 Аналіз існуючих рішень

Оскільки темою дослідження є еквайрингові платформи, то доцільно буде оцінювати якість систем, які безпосередньо є конкурентами. Було розглянуто три подібних сервіси, оскільки переважна більшість мають однаковий функціонал.

#### 1.2.1 Платформа LiqPay

LiqPay – це українська платіжна система, заснована ПриватБанком, яка дозволяє підприємствам і приватним особам приймати онлайн-платежі. Платформа підтримує широкий спектр платіжних інструментів, таких як банківські картки,

мобільні платежі, перекази через соціальні мережі та інші методи, що робить її зручною для різних типів користувачів. LiqPay є популярною серед малого та середнього бізнесу, особливо в Україні, завдяки своїй простоті інтеграції та доступності.

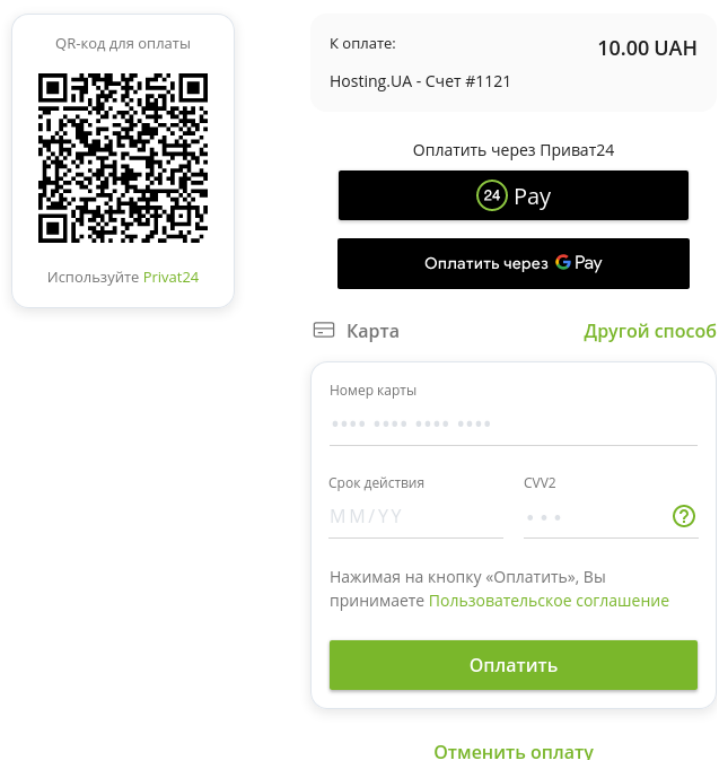


Рисунок 1.1 – Платіжна сторінка LiqPay [1]

Її можливості:

- а) перекази з карти на рахунок;
- б) перекази з рахунку на карту;
- в) фрейм для токенизації карти. Це допоможе для кастомізації платіжної сторінки і повної інтеграції з LiqPay;
- г) оплата за допомогою мобільного додатку Privat24. Але ця можливість є лише у клієнтів цього банку.

Недоліки:

- а) неможливість вибирати кращі тарифи для користування, оскільки інтеграція відбувається виключно з банком-екваєром – «ПриватБанк»;

б) особливість оплати за допомогою додатку Privat24, є перевагою та недоліком одночасно, оскільки не всі платники є клієнтами цього банку;

в) немає можливості P2P переказу.

Також не було знайдено жодної згадки використання штучного інтелекту, або інших механізмів, для покращення конверсії чи то збільшення безпеки клієнтів.

### 1.2.2 Платформа WayForPay

WayForPay – українська еквайрингова платформа, яка не має жодного банку, як основного екваєру. Послуги, які надаються:

- а) P2P перекази;
- б) створення інвойсу, рахунку на оплату;
- в) онлайн кредитування;
- г) антифрод API;
- д) переказ коштів з рахунку на рахунок.

В цій платформі є антифрод API, який надається клієнтам, але це означає, що він опціональний. Тобто кожен, хто не хоче витратити час та ресурси на повну інтеграцію – залишиться в потенційній небезпеці.

Недоліки:

а) немає власної кастомізації платіжної сторінки. Тобто клієнти зобов'язані користуватись тією сторінкою, яку надає платформа;

б) немає можливості оплати GooglePay чи ApplePay (рисунок 1.2).

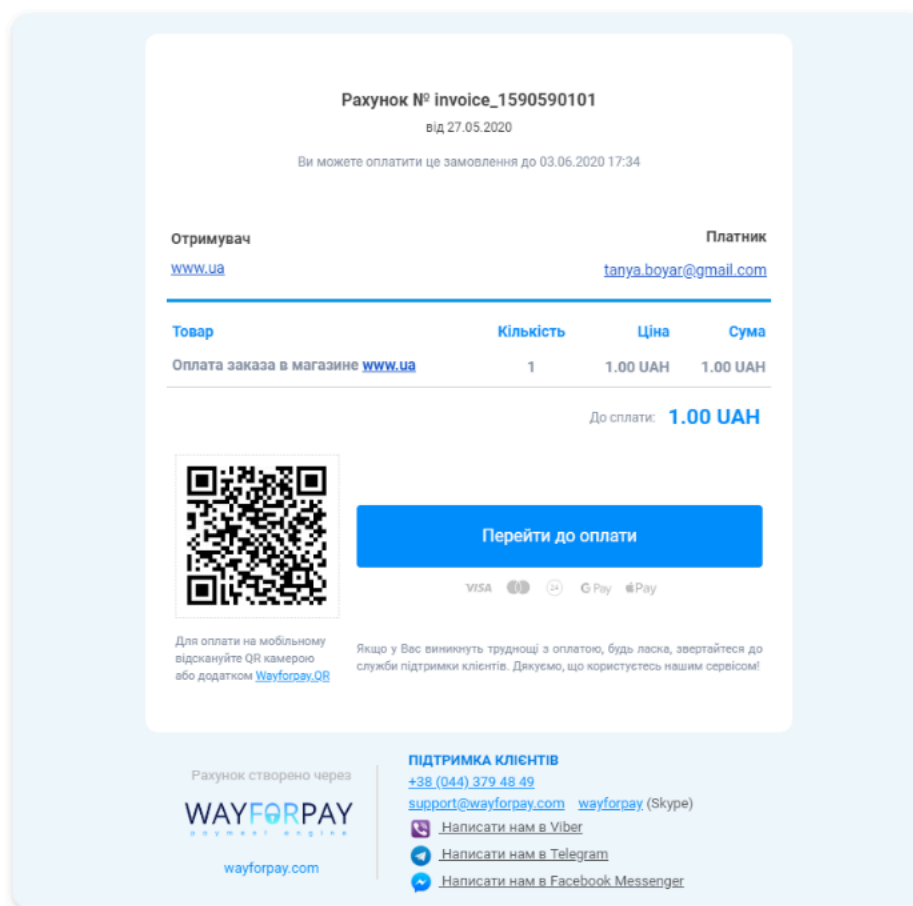


Рисунок 1.2 – Платіжна сторінка WayForPay [2]

В відкритій документації не було згадки використання штучного інтелекту.

Загалом WayForPay надає більше різних способів використання аніж LiqPay, але також має і недоліки.

### 1.2.3 Платформа UAPAY

UAPAY – українська еквайрингова платформа, яка є одним із партнерів OLX, але також має відкритий API для проведення транзакцій.

Функціонал, який надає дана платформа:

- оплата за інвойсом, на платіжній сторінці (рисунок 1.3);
- картковий фрейм, для повної інтеграції;
- панель адміністрування мерчантом, де можна створити посилання на оплату на платіжну сторінку.

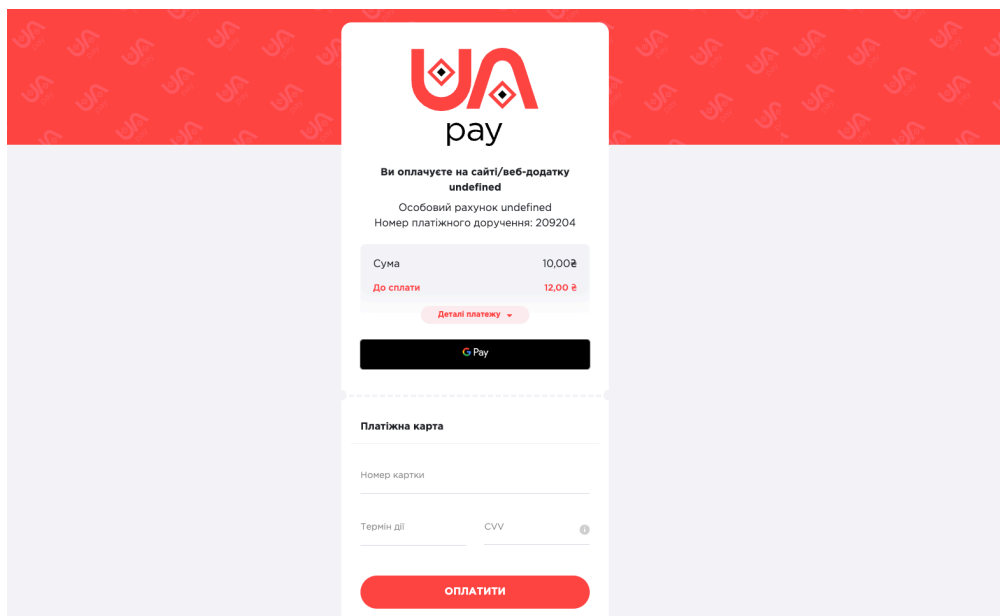


Рисунок 1.3 – Платіжна сторінка UAPAY [3]

Недоліки:

- а) для повної інтеграції недоступна оплата GooglePay та ApplePay;
- б) відсутність будь-яких інструментів для забезпечення безпеки, чи протидії фроду.

### 1.3 Аналіз різновидів платежів

Платежі, які проводяться в еквайринговій платформі зазвичай поділяються на два види:

- а) P2P – переказ з карти на карту. Тобто перекидання грошей від фізичної особи до фізичної особи;
- б) E-commerce – перекази з картки на рахунок. Тобто оплата грошей з карти на юридичний рахунок, ФОП рахунок, будь-який рахунок IBAN.

Враховуючи, що система має слідкувати за шахрайством, а заробіток в обхід оподаткуванню – є шахрайством, платформа буде надавати можливість проводити лише e-commerce платежі.



### 1.3.1 Різновиди e-commerce платежів

E-commerce платежі також поділяються на декілька типів[4]:

- а) авторизаційний платіж. Платіж який знімає одразу вказану суму з карти платника на користь отримувача;
- б) преавторизаційний платіж. Це платіж, який відбувається в два етапи. Перший, коли кошти бронюються на рахунку платника, очікуючи другий етап. Та другий етап – коли підтверджується списання коштів, але із вказанням суми. В свою чергу, сума може бути менша або більша згідно договору;
- в) рекурентний платіж. Платіж-підписка, який відбувається з підтвердженням зі сторони платника лише одноразово, і в залежності від умов повторно списує гроші коли підходить термін списання.

Для гнучкості в наданні послуг розроблена еквайрингова платформа підтримуватиме усі види e-commerce платежів.

### 1.4 Аналіз підтвердження платежу

Важливу роль відіграє безпека зі сторони платника, щоб унеможливити списання коштів з фізичної карти, або у випадку зламу, платіжних даних.

Для цього існують різні типи підтвердження платежу платником. Розглянемо основні типи підтвердження платежу: 3DS та LOOKUP.

LOOKUP більш простий в реалізації. Його суть полягає в тимчасовому блокуванні певної суми коштів, які повертаються після підтвердження за допомогою коду з СМС повідомлення. Повна послідовність зображена на рисунку 1.4.

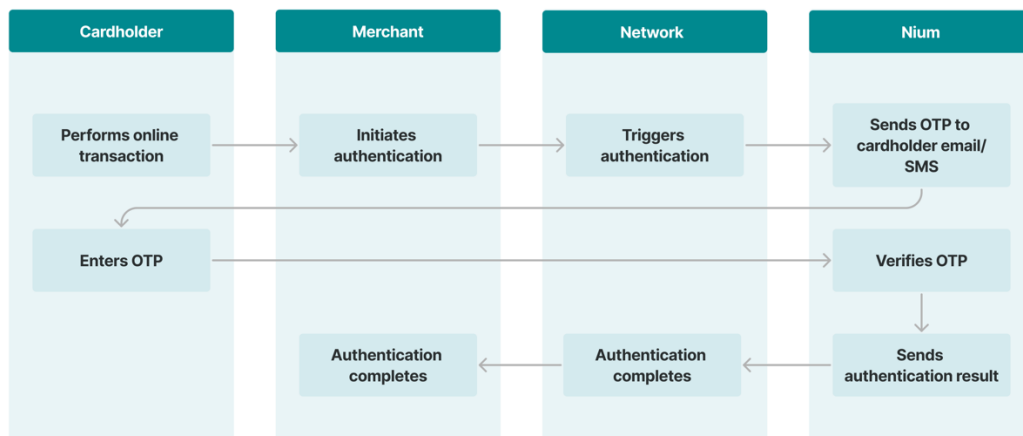


Рисунок 1.4 – Послідовність підтвердження платежу за допомогою LOOKUP [5]

Перехопити СМС повідомлення в час розвитку технологій не так складно, а отже запроваджений новий тип підтвердження платежу – 3DS. Послідовність зображено на рисунку 1.5.

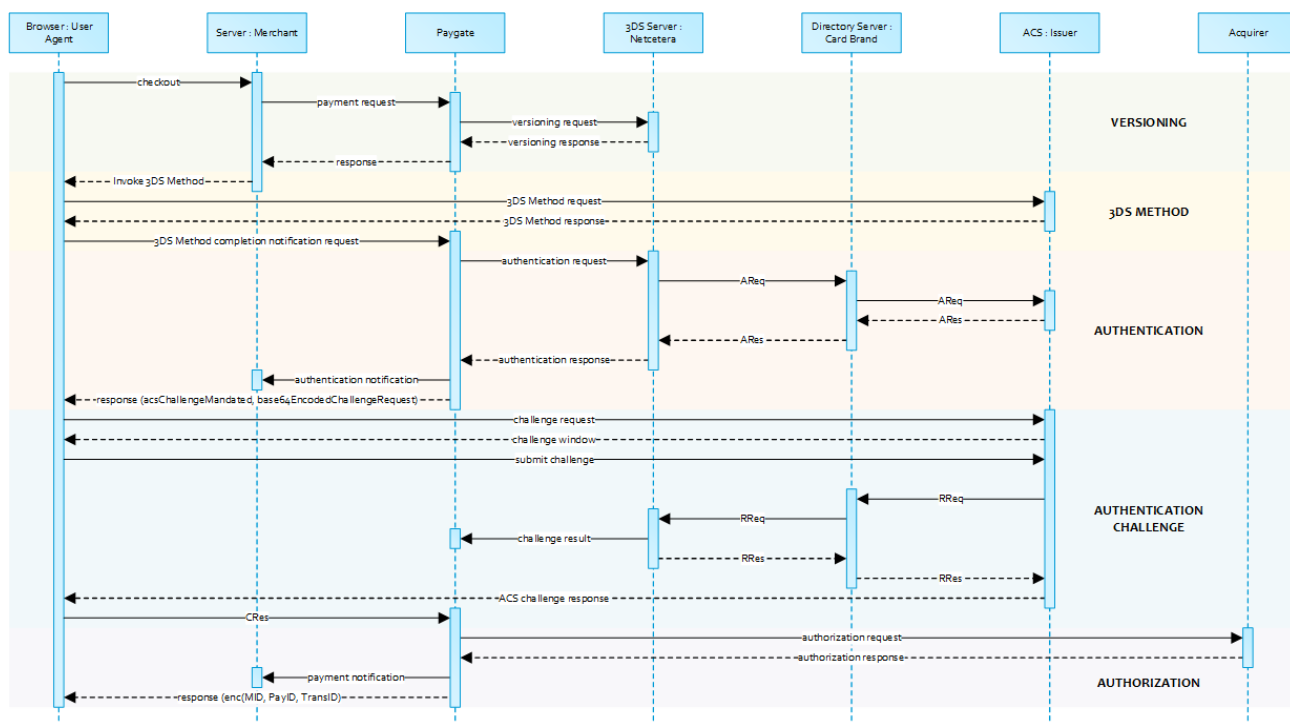


Рисунок 1.5 – Послідовність проходження 3DS перевірки [6]

Основна особливість такого підходу – залученість усіх трьох сторін: банк-екваєр, банк-емітент та ініціатор платежу. Старі версії цього протоколу також відправляють СМС, але починаючи з другої версії підтвердження відбувається в додатку банку-емітенту за його наявності в платника [6].

### 1.5 Аналіз PCI DSS сертифікації

PCI DSS – це набір вимог і рекомендацій, розроблених для забезпечення безпеки даних платіжних карт. Стандарт впроваджений Радою зі стандартів безпеки індустрії платіжних карт до якої входять провідні платіжні системи, такі як Visa, MasterCard, American Express, Discover та JCB. Основною метою PCI DSS є захист конфіденційної інформації платіжних карток від несанкціонованого доступу, шахрайства та витоків даних. Ліцензія зображена на рисунку 1.6.



Рисунок 1.6 – PCI DSS ліцензія [7]

Важливою особливістю стандарту є його гнучкість у застосуванні, оскільки він дозволяє адаптувати вимоги до конкретних умов бізнесу. Однак недотримання стандарту може призвести до значних штрафних санкцій, втрати довіри з боку клієнтів і платіжних систем, а також підвищених ризиків витоку даних.

У контексті еквайрингових платформ, відповідність PCI DSS є основною вимогою для забезпечення безпеки фінансових транзакцій та захисту даних користувачів. Використання штучного інтелекту в таких системах може сприяти підвищенню рівня безпеки через автоматичне виявлення аномалій та потенційних загроз у режимі реального часу, що полегшує дотримання стандартів безпеки.

Стандарт містить 12 основних вимог, які охоплюють різні аспекти безпеки інформаційних систем і процесів обробки платіжних карт. Ці вимоги поділяються на шість основних категорій і охоплюють різні аспекти захисту інформації, яка обробляється, зберігається або передається під час використання платіжних карт.

Перша категорія – забезпечення безпечної мережі. Вона включає в себе:

а) підтримку та встановлення безпечної мережеву інфраструктуру, використовуючи брендмауери;

б) використання належної конфігурації безпеки для всіх систем і виключення можливості використовувати стандартні паролі та інші налаштування, надані постачальником.

Друга категорія – захист даних власників платіжних карт. Вона включає:

а) захист збережених даних власників карт;

б) шифрування даних карт при передачі по відкритих мережах.

Третя категорія – підтримка системи управління вразливостями:

а) зобов'язання використовувати й регулярно оновлювати антивірусне програмне забезпечення або інші механізми захисту, встановлення та виправлення знайдених вразливостей;

б) всі програми та системи мають оновлюватись постійно до найновіших версій.

Четверта категорія – контроль доступу:

а) обмеження доступу до даних карт;

б) призначення і використання унікальних ідентифікаційних даних кожному користувачеві, що має доступ до системи;

в) обмеження фізичного доступу до даних карт, захист фізичних серверів.

П'ята категорія – моніторинг і тестування мережі:

а) відстеження та контроль всіх доступів до мережевих ресурсів і даних карт;

б) регулярне тестування системи безпеки та процесів.

Шоста категорія – підтримка політики безпеки інформації. Зобов'язує підтримувати політику безпеки для всього персоналу.

Ці вимоги включають контроль доступу до даних, шифрування інформації, моніторинг мережевої активності, регулярне тестування системи безпеки та забезпечення належного захисту від шкідливого програмного забезпечення. PCI DSS призначений для всіх організацій, які зберігають, передають або обробляють дані платіжних карт, незалежно від їх розміру чи обсягу операцій.

## Висновки до розділу 1

В цьому розділі було розглянуто три різні еквайрингові платформи. Кожна має переваги над іншою, та покриває потреби кінцевого користувача, але жодна з систем не впровадила використання штучного інтелекту для вирішення задач безпеки чи покращення конверсії платформи. Тому поставлена задача є актуальною.

## 2 ВИБІР ТЕХНОЛОГІЙ ТА ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ

### 2.1 Архітектура серверної частини

Еквайрингова платформа матиме велике навантаження. Основною вимогою до такої системи є стабільність та відмовостійкість, оскільки це опрацювання мільйонів платежів, тисячі генерацій бухгалтерських звітів, десятки інтеграцій та багато всього іншого. Тож прийняте рішення використовувати мікросервісну архітектуру.

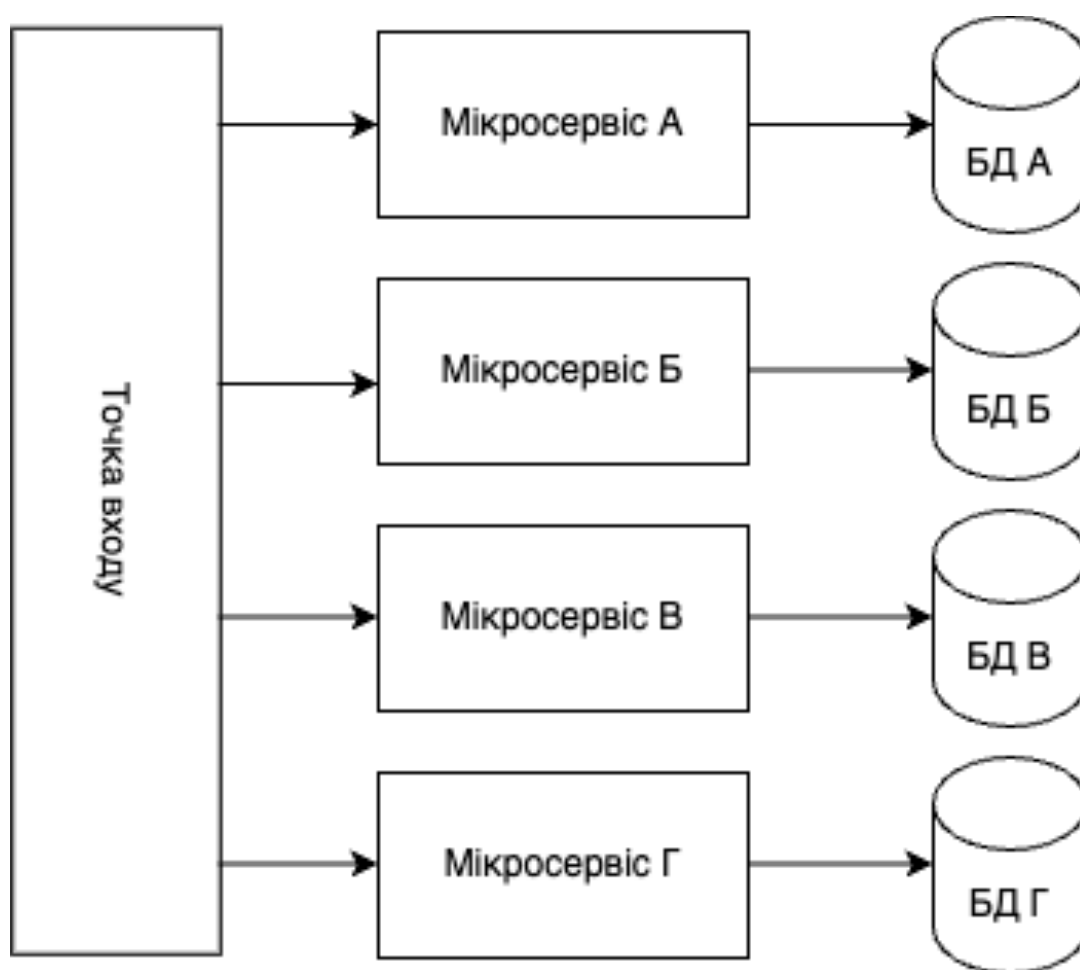


Рисунок 2.1 – Мікросервісна архітектура

Головними перевагами такого підходу є:

а) легкість масштабування. В свою чергу масштабування є горизонтальне та вертикальне. Горизонтальне – це збільшення кількості екземплярів певного

мікросервісу. Вертикальне – це збільшення ресурсів віртуальної чи фізичної машини;

б) відмовостійкість. Кожен мікросервіс має свою базу даних, тож при непередбачуваних обставинах лише частина функціоналу усієї системи не буде працювати, допоки не відбудеться відновлення;

в) розподілене розроблення. В такій великій системі повинна бути присутня підтримка усієї екосистеми, а це і декілька веб-проектів, і підтримка різних інтеграцій, і підтримка бекенду, який в свою чергу має обслуговувати усі ці кінцеві продукти, і підтримка мобільної інтеграції. Тож, розділивши усе це на міні-проекти та бекенд на мікросервіси, буде не складно розділити відповідальність між різними командами, які мають компетенцію в конкретній галузі.

### 2.1.1 Шаблони проєктування

Оскільки було обрано мікросервісну архітектуру, треба обрати потрібні шаблони проєктування системи, щоб переваги, які надає перша, були підсилені реалізацією.

Шаблон проєктування CQRS вирішує проблему розділення запитів на запис і на читання, що дозволяє керувати навантаженням на систему. Ідея в тому, щоб розділити механізми транспорту запитів між мікросервісами на два види:

- а) асинхронні запити для запису;
- б) синхронні запити для читання.

Асинхронні запити дозволяють розвантажити систему, оскільки початковий користувач не очікуватиме довго відповідь від навантаженого запиту. Наприклад, платник оплачує інвойс. В цьому кейсі відбуваються операції запису на диск, щоб зберегти інформацію в БД, також відбувається спілкування з банком-екваєром. І весь цей час користувач очікує відповіді з лоадером на веб-сторінці. Асинхронні запити дозволяють реалізувати «оптимістичне створення». Платіж по інвойсу буде створений в будь-якому випадку, але не завжди буде успішним. Тож нічого не

забороняє оптимістично створити платіж, і не затримувати кінцевого користувача. Такий підхід реалізує патерн «Команда» та шина повідомлень

Реалізація синхронних запитів в мікросервісній архітектурі може бути вирішене наступними рішеннями:

- а) HTTP запити між мікросервісами;
- б) RPC запити за допомогою шини повідомлень.

Переваги HTTP рішення – швидка реалізація. Потрібно лише створювати веб-сервер для кожного мікросервіса.

Недоліки HTTP рішення – відсутність «кешу» запитів.

Головна перевага RPC за допомогою шини повідомлень – є сама шина. Вона дозволяє «заморожувати» запити між мікросервісами, якщо кінцевий мікросервіс не відповідає. Це підсилює цілісність системи, оскільки непрацюючий сервіс обробить запити, які йому надсилались, після усунення проблем в роботі.

Доменну організацію структури (далі DDD) було обрано як основний підхід до написання мікросервісу. DDD реалізує конкретну модель обраної предметної області, тобто кожна сутність, відповідає тільки за зміну свого стану і процеси, які пов'язані з нею.

### 2.1.2 Вибір шини повідомлень

При виборі шини повідомлень було розглянуто наступні варіанти:

- а) RabbitMQ;
- б) Kafka;
- в) AWS SQS.

Переваги RabbitMQ:

- а) підтримка протоколів TOMP, MQTT, AMQP
- б) більше можливостей в маршрутизації повідомлень в черги, та їх перенаправлення;

- в) можливість реалізації RPC протоколу.

Переваги Kafka:



- а) доступне API роботи зі стрімами. Але синхронізація баз даних вирішується на інфраструктурному рівні;
- б) паралельна обробка черг.

Переваги AWS SQS:

- а) проста реалізація оформлення черг;
- б) проста документація;
- в) підтримка екосистеми AWS.

Враховуючи вимоги до використання шини повідомлень, RabbitMQ покриє усі потреби в використанні шини повідомлень, а саме: реалізація протоколу RPC, та реалізація патерну «Команда».

## 2.3 Базы даних

### 2.3.1 Вибір бази даних

Під час вибору БД було розглянути два варіанти: PostgreSQL та MySQL. Для порівняння було розглянуто наступні фактори:

- а) продуктивність читання. PostgreSQL зазвичай швидше обробляє складні запити завдяки розширеним можливостям індексування та підтримці Index-Only Scans. MySQL ефективніше працює із простими завданнями читання, але поступається PostgreSQL у випадку складних запитів;

- б) продуктивність запису. PostgreSQL значно перевершує MySQL у навантаженнях, орієнтованих на записи. Його механізм Write-Ahead Logging забезпечує кращу багатозадачність і швидше відновлення після збоїв. У тестах PostgreSQL був приблизно в 3,5 рази швидше за MySQL у завданнях запису [8];

- в) загальна затримка. PostgreSQL демонструє у 2,3 рази кращу продуктивність у середньому по набору поширених операцій, особливо перевершуючи MySQL у завданнях групування та сканування таблиць;

- г) конкурентність та масштабованість. PostgreSQL використовує багатOVERСІЙну контрольну систему конкурентності, що дозволяє виконувати транзакції без блокування читання, забезпечуючи високу пропускну здатність для

великої кількості одночасних запитів. MySQL також підтримує MVCC, але гірше справляється з високим навантаженням. PostgreSQL має паралельне виконання запитів, розділення таблиць на партиції та кращі можливості горизонтального масштабування, що робить його ідеальним для великих і складних систем [9];

д) типи даних. PostgreSQL підтримує більшу кількість типів даних, зокрема JSONB для роботи з напівструктурованими даними, а також геометричні та мережеві типи. Це робить його більш універсальним для сучасних додатків;

е) індексація. PostgreSQL підтримує розширені функції, як-от часткові та функціональні індекси, які суттєво покращують продуктивність у специфічних сценаріях. MySQL таких функцій не має;

ж) надійність та відповідність стандартам. PostgreSQL суворо дотримується принципів ACID і стандартів SQL, забезпечуючи високу цілісність даних. Хоча MySQL також підтримує ACID, його обробка транзакцій менш надійна у складних сценаріях [10].

PostgreSQL поєднує масштабованість, продуктивність і розширені функції, що робить його для проєктів з високими вимогами до обробки даних.

### 2.3.2 Проєктування структури бази даних

Проєктування структури БД є настільки ж важливим, як і вибір технологій чи вибір архітектури. Гарно спроектована БД завжди дається в знаки – легко підтримувати, легко розширювати. Проєктування БД для еквайрингової платформи розділимо на три частини:

- а) проєктування основної частини;
- б) проєктування частини адміністрування;
- в) проєктування зберігання чутливої інформації про карти.

Зберігання чутливих даних карток зображено на рисунку 2.3.

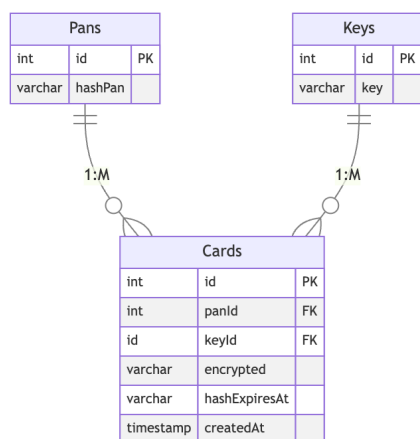


Рисунок 2.2 – Структура БД зберігання чутливої інформації

Детальний опис таблиць з рисунку 2.2 наведено в таблицях 2.1–2.3.

Таблиця 2.1 – Таблиця карт платників

Обмеження	Назва поля	Тип даних	Опис поля
Первинний ключ	id	char(36)	Ідентифікатор
Вторинний ключ	panId	char(36)	Посилання на ідентифікатор пану картки
Вторинний ключ	keyId	char(36)	Посилання на ідентифікатор ключа яким було зашифровано чутливу інформацію
	encrypted	varchar	Зашифрована чутлива інформація
	hashExpiresAt	varchar	Хешована дата життя карти
	createdAt	timestamp	Дата створення карти в системі

Таблиця 2.2 – Таблиця панів карт платників

Обмеження	Назва поля	Тип даних	Опис поля
Первинний ключ	id	char(36)	Ідентифікатор
	hashPan	varchar	Захешований пан карти платника

Таблиця 2.3 – Таблиця ключів шифрування

Обмеження	Назва поля	Тип даних	Опис поля
Первинний ключ	id	char(36)	Ідентифікатор
	key	varchar	Зашифрований ключ шифрування

Проектування основної частини знаходиться на рисунку 2.3.

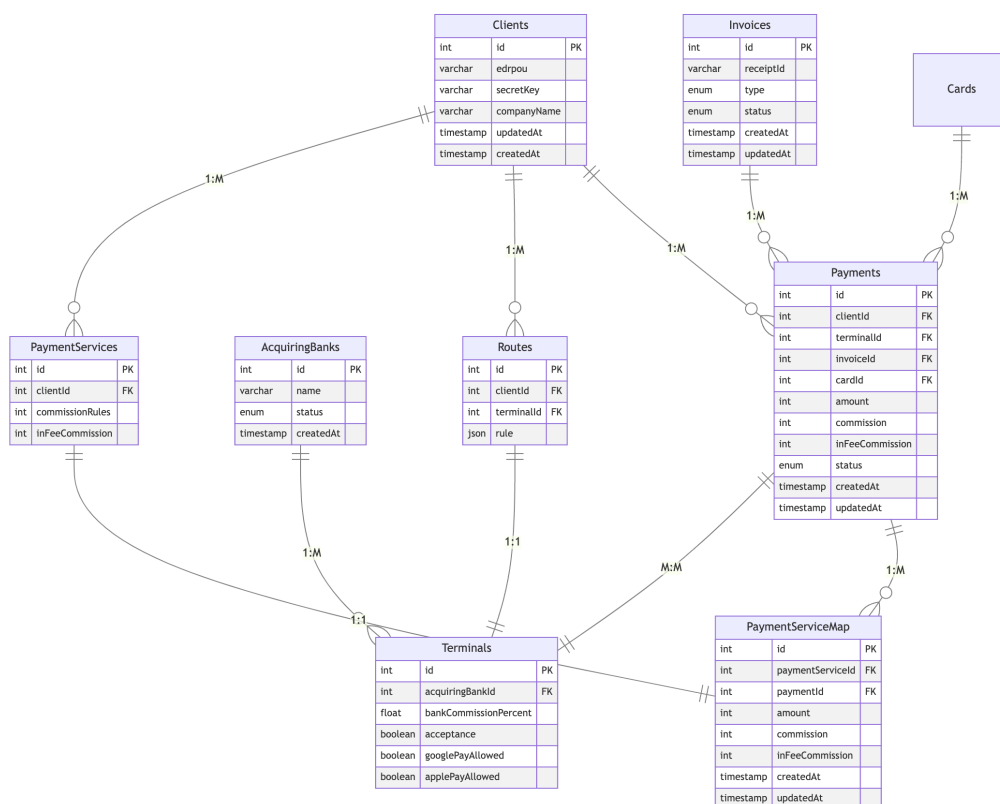


Рисунок 2.3 – Структура БД основної частини

Детальний опис таблиць з рисунку 2.3 наведено в таблицях 2.4–2.11.

Таблиця 2.4 – Таблиця клієнтів

Обмеження	Назва поля	Тип даних	Опис поля
Первинний ключ	id	char(36)	Ідентифікатор
	edrpou	varchar	ЄДПОУ ТОВа, або ПІН ФОПа
	secretKey	varchar	Приватний ключ клієнта
	companyName	varchar	Назва організації
	createdAt	timestamp	Дата створення
	updatedAt	timestamp	Дата оновлення

Таблиця 2.5 – Таблиця інвойсів

Обмеження	Назва поля	Тип даних	Опис поля
Первинний ключ	id	char(36)	Ідентифікатор
	receiptId	char (36)	Ідентифікатор квитанції
	type	enum	Тип інвойсу
	status	enum	Статус інвойсу
	createdAt	timestamp	Дата створення
	updatedAt	timestamp	Дата оновлення

Таблиця 2.6 – Таблиця платежів

Обмеження	Назва поля	Тип даних	Опис поля
Первинний ключ	id	char(36)	Ідентифікатор
Вторинний ключ	clientId	char(36)	Посилання на клієнта
Вторинний ключ	terminalId	char(36)	Посилання на термінал

Обмеження	Назва поля	Тип даних	Опис поля
Вторинний ключ	invoiceId	char(36)	Посилання на інвойс
Вторинний ключ	cardId	char(36)	Посилання на карту платника
	amount	integer	Тіло платежу
	commission	integer	Комісія внутрішня
	inFeeCommission	integer	Комісія зовнішня
	status	enum	Статус платежу
	createdAt	timestamp	Дата створення
	updatedAt	timestamp	Дата оновлення

Таблиця 2.7 – Таблиця банків екваєрів

Обмеження	Назва поля	Тип даних	Опис поля
Первинний ключ	id	char(36)	Ідентифікатор
	name	Varchar	Назва банку
	status	enum	Статус
	createdAt	timestamp	Дата створення

Таблиця 2.8 – Таблиця реквізитів клієнта

Обмеження	Назва поля	Тип даних	Опис поля
Первинний ключ	id	char(36)	Ідентифікатор
Вторинний ключ	clientId	char(36)	Посилання на клієнта
	commissionRules	json	Правила розрахунку комісій

Обмеження	Назва поля	Тип даних	Опис поля
	inFeeCommission	json	Правила розрахунку комісій клієнта

Таблиця 2.9 – Таблиця терміналів

Обмеження	Назва поля	Тип даних	Опис поля
Первинний ключ	id	char(36)	Ідентифікатор
Вторинний ключ	acquiringBankId	char(36)	Посилання на банк-екваєр
	bankCommissionPercent	float	Відсоток комісії банку
	acceptance	boolean	Підтримка безакцептності
	googlePayAllowed	boolean	Підтримка оплати Google Pay
	applePayAllowed	boolean	Підтримка оплати Apple Pay

Таблиця 2.10 – Таблиця маршрутизації транзакцій клієнта

Обмеження	Назва поля	Тип даних	Опис поля
Первинний ключ	id	char(36)	Ідентифікатор
Вторинний ключ	clientId	char(36)	Посилання на клієнта
Вторинний ключ	terminalId	char(36)	Посилання на термінал

Обмеження	Назва поля	Тип даних	Опис поля
	rule	Json	Правило маршрутизації

Таблиця 2.11 – Таблиця зв'язку платежів з реквізитами

Обмеження	Назва поля	Тип даних	Опис поля
Первинний ключ	id	char(36)	Ідентифікатор
Вторинний клбч	paymentServiceId	char(36)	Посилання на реквізити
Вторинний ключ	paymentId	char(36)	Посилання на платіж
	amount	integer	Винагорода
	commission	integer	Комісія внутрішня
	inFeeCommission	integer	Комісія зовнішня
	createdAt	timestamp	Дата створення
	updatedAt	timestamp	Дата оновлення

Проектування частини адміністрування зображено на рисунку 2.4.

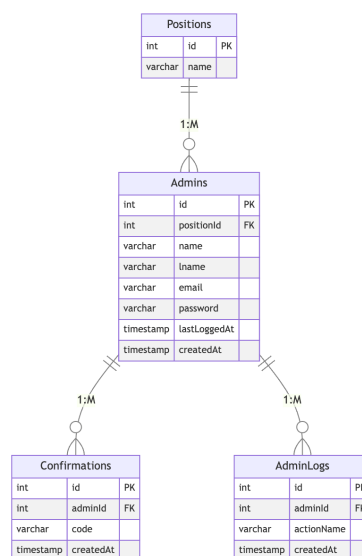


Рисунок 2.4 – Структура БД адміністративної частини



Детальний опис таблиць з рисунку 2.4 наведено в таблицях 2.12–2.15.

Таблиця 2.12 – Таблиця адміністраторів

Обмеження	Назва поля	Тип даних	Опис поля
Первинний ключ	id	char(36)	Ідентифікатор
Вторинний ключ	positionId	char(36)	Посилання на позицію адміністратора
	name	varchar	Ім'я адміністратора
	lname	varchar	Прізвище адміністратора
	email	varchar	Робоча пошта
	password	varchar	Захешований пароль
	lastLoggedAt	timestamp	Дата останньої авторизації
	createdAt	timestamp	Дата створення

Таблиця 2.13 – Таблиця позицій

Обмеження	Назва поля	Тип даних	Опис поля
Первинний ключ	id	char(36)	Ідентифікатор
	name	varchar	Назва позиції

Таблиця 2.14 – Таблиця кодів підтверджень

Обмеження	Назва поля	Тип даних	Опис поля
Первинний ключ	id	char(36)	Ідентифікатор
Вторинний ключ	adminId	char(36)	Посилання на адміністратора

Обмеження	Назва поля	Тип даних	Опис поля
	code	varchar	Код підтвердження
	createdAt	timestamp	Дата створення

Таблиця 2.15 – Таблиця логів адміністраторів

Обмеження	Назва поля	Тип даних	Опис поля
Первинний ключ	id	char(36)	Ідентифікатор
	adminId	char(36)	Посилання на адміністраторів
	actionName	varchar	Дія
	createdAt	timestamp	Дата дії

Зв'язки між таблицями БД наведено в таблиці 2.16

Таблиця 2.16 – Зв'язки між таблицями БД

Назва таблиці	Зв'язок	Назва таблиці
Таблиця панів карт	1:M	Таблиця карт платника
Таблиця ключ шифрування	1:M	Таблиця карт платника
Таблиця карт платника	1:M	Таблиця платежів
Таблиця інвойсів	1:M	Таблиця платежів
Таблиця клієнтів	1:M	Таблиця платежів
Таблиця платежів	1:M	Таблиця зв'язку платежа та реквізитів
Таблиця платежів	M:N	Таблиця реквізитів
Таблиця платежів	1:1	Таблиця терміналів
Таблиця зв'язку платежа та реквізитів	1:1	Таблиця терміналів
Таблиця терміналів	1:1	Таблиця маршрутизації
Таблиця терміналів	1:M	Таблиця банків-екваєрів

Таблиця реквізитів	M:1	Таблиця клієнтів
Таблиця адміністраторів	1:1	Таблиця позицій
Таблиця адміністраторів	1:1	Таблиця логів

## 2.4 Вимоги згідно PCI DSS сертифікації

Вище розглянуті вимоги до PCI DSS сертифікації, згідно якої аудит відбувається для кожної еквайрингової платформи без виключення. Ці вимоги дуже загальні, тож потрібно висвітити підходи, які використані в ході розроблення програмного забезпечення, щоб відповідати даним вимогам.

### 2.3.1 Вимоги до способу зберігання чутливої інформації

Щоб відповідати вимозі зберігання чутливої інформації, поставлені наступні задачі:

а) в БД ніколи не має зберігатись CVV карти платника. Код підтвердження карти потрібен лише на момент оплати для передачі банкам-екваєрам, а отже, не зберігаючи CVV, знімається відповідальність з системи, у разі інцидентів з витоком даних;

б) в БД ніколи не має зберігатись інформація про номер карти та дата придатності в відкритому вигляді, а отже тільки в зашифрованому. Такий підхід дозволить вберегти чутливу інформацію, навіть у випадку витоку і злому БД;

в) пошук карти платника може відбуватись за допомогою контрольної суми, тобто хешу. З таким підходом, не потрібно кожен раз розшифровувати чутливу інформацію, щоб дізнатися чи існує така карта;

г) побудувати процес зміни ключа шифрування. Проектування БД, яке розглянуто раніше, дозволяє використовувати різні ключі шифрування. Що в свою чергу дозволить безболісно оновлювати і перешифровувати дані, не вимикаючи сервіс, оскільки чутлива інформація, яка ще не була перешифрована, може використати старий ключ шифрування для передачі інформації;

д) передача інформації про карту платника, при спілкуванні з банками-екваєрами має відбуватись тільки зашифрованому вигляді.

Дотримуючись усіх цих підходів, система буде відповідати вимозі зберігання і шифрування чутливої інформації.

### 2.3.2 Вимоги до адміністрування

Четверта, п'ята та шоста вимоги – про безпеку, контроль доступу, моніторинг системи. Щоб дотримуватися цих вимог, поставлені наступні задачі:

а) чітка сітка ролей і їх доступів до можливостей адміністрування. Спроектowana БД для адміністрування включає можливість розділення адміністраторів на ролі, і відповідно на можливості в адміністративній панелі;

б) авторизація має відбуватися кожні 15хв;

в) авторизація має відбуватися за допомогою двофакторного підтвердження;

г) логування і контроль дій адміністраторів. Спроектowana БД записує усі дії адміністраторів;

д) адміністратори не мають прав на перегляд чутливої інформації. Оскільки уся інформація зашифрована, адміністрація може користуватись автоматичними механізмами і ідентифікаторами, для завершення дій пов'язаних з платежами: зарахування, повернення тощо.

### 2.3.3 Вимоги до інтеграції з еквайринговою платформою

Еквайрингова платформа надає гнучкі можливості для клієнтів в способі використання:

а) готова платіжна сторінка;

б) клієнтська панель адміністрування, де можна створити посилання на готову платіжну сторінку;

в) повна інтеграція, задля кастомізації своєї сторінки оплати.

Останній пункт дуже чутливий, оскільки клієнт, який не матиме PCI DSS ліцензії, зможе реалізувати власні рішення. За таких обставин дуже важливо дотримуватися вимоги зберігання чутливої інформації.

Отже, щоб відповідати цим вимогам, треба:

- а) ізолювати інтеграцію клієнта від створення карткових даних в нашій системі. Цей пункт легко вирує реалізація фрейму для створення карти платника в системі;
- б) не мати в відкритому API точок входу, які віддають чутливу інформацію;
- в) для взаємодії з системою використовувати асиметричне шифрування для підписів усіх запитів і відповідей.

## 2.4 Вибір технологій для роботи зі штучним інтелектом

Інтеграція штучного інтелекту в систему керування платежами стає основним інструментом для забезпечення високої точності та ефективності у виявленні підозрілих транзакцій і прийнятті рішень щодо їх подальшої обробки.

Python є ідеальним вибором для задач виявленню фроду завдяки таким факторам:

- а) розвинена екосистема бібліотек. Бібліотеки для машинного та глибинного навчання PyTorch, TensorFlow [9] спрощують процес створення, навчання та тестування моделей;
- б) інтеграція з БД та API. Python забезпечує підключення до БД, що полегшує збір і обробку транзакційних даних.

Зважаючи на вимоги до точності та швидкості, глибинні нейронні мережі на базі TensorFlow є оптимальним рішенням. Цей підхід дозволить:

- а) виявити складні, приховані шаблони в транзакціях;
- б) забезпечити масштабованість у випадку збільшення обсягів даних;
- в) інтегрувати рішення в еквайрингову систему.

Такий набір технологій дозволить вирішувати завдання визначення безпеки транзакцій і прийняття рішень без додаткової перевірки, забезпечуючи високу точність і швидкість обробки.

## 2.5 Вибір мови програмування для серверної частини

JavaScript та TypeScript є популярними мовами програмування, які використовуються для створення серверної частини веб-додатків на платформі Node.js. Обидві мови мають низку переваг, які роблять їх хорошим вибором для еквайрингової платформи. Основні переваги:

- а) швидкість розробки, яка дозволяє швидко створювати прототипи та тестувати ідеї;
- б) асинхронність, найшвидша серед усіх мов програмування. JavaScript має лише один потік, на відміну від Java чи C#, але проблеми навантаженості вирішуються горизонтальним масштабуванням на рівні інфраструктури, що нівелює перевагу раніше згаданих конкурентів.
- в) гнучкість, оскільки має дуже великий спектр бібліотек, що дає свободу у виборі підходу до розроблення.

TypeScript є надбудовою над JavaScript, яка додає статичну типізацію та покращує зручність і надійність розроблення. Явна типізація дозволяє різним командам, швидко орієнтуватись в кодовій базі, задля вирішення питань.

Тож, враховуючи усі ці фактори, було обрано TypeScript як основну мову програмування серверної частини.

## 2.6 Бібліотеки для платформи Node.js

Node.js[10] надає великі можливості як у виборі бібліотек, так і виборі пакетних менеджерів. Надійний та протестований часом є – node package manager, або npm[11]. Але, оскільки кожен може викласти свою бібліотеку в відкритий

доступ, слід вибирати дуже ретельно, щоб це ніяк не повпливало на роботу систему, чи її злом, через застарілість чи відсутність підтримки бібліотеки.

### 2.6.1 Фреймворк веб-серверу

Було обрано три найпопулярніші фреймворки для розроблення веб-серверів: Koa.js, Fastify та Express.js

Бенчмарк-порівняння пропускної спроможності кожного з них зображено на рисунку 2.5

Framework	Version	Router?	Requests/sec
Express	4.17.3	✓	14,200
hapi	20.2.1	✓	42,284
Restify	8.6.1	✓	50,363
Koa	2.13.0	✗	54,272
<b>Fastify</b>	<b>4.0.0</b>	<b>✓</b>	<b>77,193</b>
-			
<code>http.Server</code>	16.14.2	✗	74,513

Рисунок 2.5 – Бенчмарк пропускної спроможності фреймворків [15]

Отже Fastify виглядає найкращим з кандидатів. Було розглянуто і порівняно інші критерії, які наведені в таблиці 2.17.

Таблиця 2.17 – Порівняльна характеристика фреймворків

Назва критерію	Fastify	Koa.js	Express.js
Продуктивність	90,000–100,000 запитів в секунду	70,000 запитів в секунду	50,000–60,000 запитів в секунду
Підтримку модульності та розширюваності	Має вбудовану підтримку плагінів	Доведеться самостійно інтегрувати модулі	Складна інтеграція модулів та плагінів
Типізація TypeScript	Найкраща підтримка TypeScript	Підтримує TypeScript	Підтримується, але через старішу архітектуру не завжди зрозуміла
Вбудована валідація	Існує	Не має	Не має
Спільності та документація	Спільнота молода, проєкт також, але активно зростає. Документація чітка	Має невелику, але активну спільноту. Документація більш технічна, потребує часу на вивчення	Найбільша спільнота та найбільша кількість ресурсів, але документація може бути застарілою для нових вимог

Fastify є найбільш технічно досконалим рішенням, що поєднує високу продуктивність, зручність у розробці та відповідність сучасним стандартам. Це робить його кращим вибором для серверної частини еквайрингової платформи.



## 2.6.2 Бібліотека ORM для роботи з базою даних

Об'єктно-реляційні відображення – це додаткова абстракція, яка дозволяє працювати з таблицями БД, як з об'єктами ООП мови програмування. Було розглянуто наступні бібліотеки: Drizzle, TypeORM, Prisma. Порівняння наведено в таблиці 2.18.

Таблиця 2.18 – Порівняння бібліотек ORM

Критерій	Drizzle	TypeORM	Prisma
Продуктивність	Забезпечує високу продуктивність завдяки генерації SQL-запитів під час компіляції TypeScript. Це дозволяє уникнути зайвого оверхеду під час виконання, який часто є в інших ORM	Відомий як важчий ORM. Генерація SQL-запитів може бути повільнішою, особливо для складних запитів	Продуктивність добра, але через генерацію запитів і специфічну інфраструктуру (Prisma Client і Prisma Engine) може бути трохи повільнішим порівняно з Drizzle у сценаріях високого навантаження.
Типізація та інтеграція з TypeScript	Забезпечує статичну типізацію на основі схеми бази даних. Ви отримуєте автодоповнення і статичну перевірку типів, які	Має підтримку TypeScript, але типізація менш надійна і менш автоматизована. Вимагає ручної роботи для	Має дуже сильну підтримку типів і автоматично генерує типи на основі схеми Prisma

	допомагають уникнути помилок під час розробки	синхронізації моделі і схеми бази	
Гнучкість	Не накладає строгих обмежень на структуру проєкту, дозволяючи створювати запити SQL на основі бізнес-логіки. Це добре підходить для кастомних та високопродуктивних додатків	Більш "монолітний". Підтримує декоратори, які спрощують визначення моделей, але ускладнюють кастомізацію	Підходить для стандартних CRUD-операцій, але може бути менш гнучким для складних кастомних запитів через свій декларативний підхід
Підтримка міграцій	Міграції генеруються на основі змін у схемі. Система проста і дозволяє легко керувати версіями бази даних	Має вбудовану систему міграцій, але її використання може бути складним і часом ненадійним	Має добре продуману систему міграцій, але її складність зростає при роботі зі складними базами даних

Порівнявши готові рішення, було обрано Drizzle, як основну бібліотеку ORM, через її високу продуктивність, надійність типізації, гнучкості для складних запитів та реалізації простих міграцій.

### 2.6.3 Авторизація та аутентифікація

Авторизація та аутентифікація і підхід до реалізації також дуже важливий з точки зору безпеки еквайрингової платформи. Авторизація буде присутня в двох місцях системи, а аутентифікація в одному.

Було розглянуто наступні варіанти авторизації: JWT та сесії.

Принцип роботи сесії зображено на рисунку 2.6.

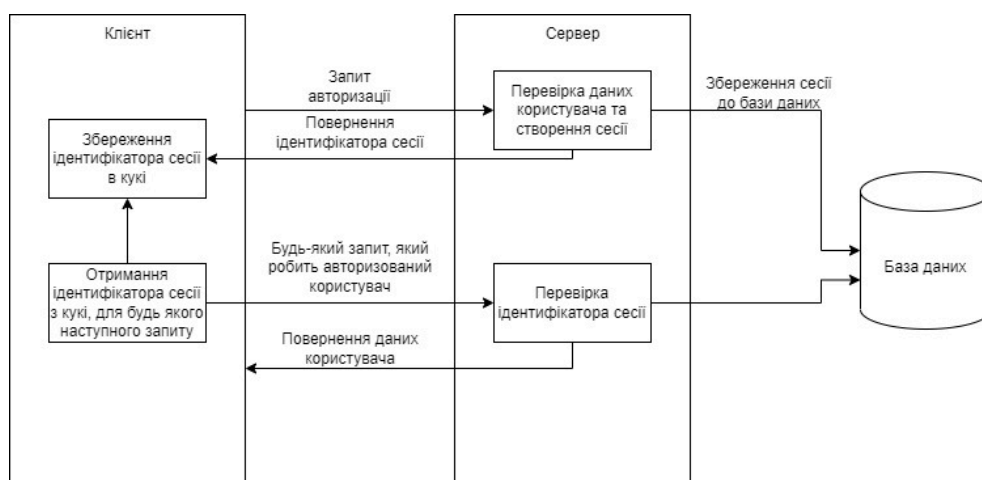


Рисунок 2.5 – Принцип роботи сесії

Основна ідея, що будь-який авторизований запит буде проходити через додатковий запит в БД, що в рази навантажує систему.

Принцип роботи JWT зображено на рисунку 2.7.

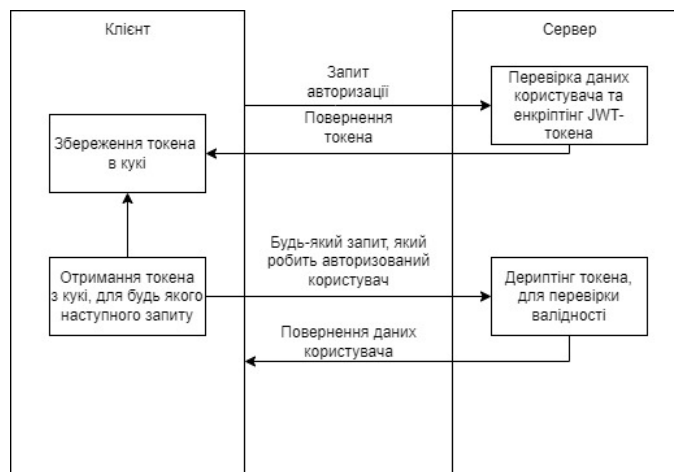


Рисунок 2.7 – Принцип роботи JWT

Уникнення додаткового запиту в БД, який пропонує принцип роботи з JWT кращий.

Аутентифікація потрібна, якщо клієнт платформи бажає інтегрувати «онлайн оплату» безпосередньо в свою систему. В цьому кейсі необхідно бути впевненим, що запити які надсилаються ззовні мають бути завірені. Таку проблему вирішує підпис. Існують різні алгоритми підпису, але підхід асиметричного шифрування вирішує наступу проблему. В разі витоку даних клієнта система не буде нести відповідальність за те що потенційно саме через неї відбувся «витік даних». Симетричне шифрування має лише один ключ, тож в цьому випадку така ймовірність існує.

#### 2.6.4 Другорядні бібліотеки

Раніше було порівняно та обрано бібліотеки широкого спектру використання або їх концептуально важливого змісту. Було також обрано низку другорядних бібліотек, які не мали аналогів, та вирішували вузькі проблеми. Список таких бібліотек наведено в таблиці 2.19.

Таблиця 2.19 – Список другорядних бібліотек

Назва бібліотеки	Опис
Fastify	Фреймворк для веб-серверу [15]
Drizzle	ORM [16]
zod	Бібліотека валідації [17]
amqplib	Бібліотека спілкування з RabbitMQ [18]
InversifyJS	Бібліотека імплементації DI [19]
jsonwebtoken	Бібліотека генерації JWT [20]
dotenv	Бібліотека змінних оточення [21]
pino	Бібліотека логування [22]
iconv	Бібліотека генерування звітів [23]
ioredis	Бібліотека підключення до Redis [24]

Назва бібліотеки	Опис
typescript	Бібліотека TypeScript [25]

## 2.7 Вибір технологій для інтерфейсів

Для розробки було обрано Vue3 фреймворк для мови JavaScript. Vue3 — це прогресивний фреймворк для розроблення веб-додатків, створений для того, щоб бути простим у використанні, але досить потужним для побудови складних додатків. Vue 3 є останньою версією цього фреймворку, яка представила кілька значних покращень у продуктивності, функціональності та гнучкості порівняно з попередніми версіями.

Особливості фреймворку:

- а) новий спосіб організації логіки компонентів, який дає більше гнучкості та покращує повторне використання коду. Це доповнює традиційний Options API;
- б) підтримка реактивності на основі Proху;
- в) покращена продуктивність. Vue 3 оптимізований для швидкості завантаження, рендерингу та оновлення. Це досягається завдяки новій компіляційній стратегії та оптимізації компонентів;
- г) підтримка Fragment, Teleport і Suspense.

## Висновки до розділу 2

Отже, було обрано мікросервісну архітектуру задля вирішення проблеми масштабування та навантаженості. Обрано технології для роботи зі штучним інтелектом. Обрано технології для розроблення веб-інтерфейсів. Спроектовано БД, структура якої описує усі можливі варіанти використання. Також було обрано протоколи взаємодії за допомогою шини повідомлень і саму шину повідомлень



### 3 ОПИС СТРУКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

#### 3.1 Структура мікросервісів серверної частини

Класичні «мікросервіси» виглядають як «одна сутність – один мікросервіс». Тим самим відмовостійкість – збільшується, але швидкість роботи – зменшується. Натомість, було обрано більш сконцентрований на оптимізацію підхід до вибору домену, де домен – є абстракцією з багатьма сутностями, які його описують, а не сутність. Нижче було розглянуто сутності та їх властивості в ході розроблення ПЗ.

##### 3.1.1 Опис мікросервісу API-Gateway

API-Gateway мікросервіс – це точка доступу для реалізації клієнт-серверної архітектури та інтеграції клієнтів платформи. Реалізований за допомогою патерну API Gateway, задача якого полягає в маршрутизації трафіку запитів. Уся логіка рознесена по іншим сервісам.

Цей мікросервіс не має сутностей, але має багато контролерів та інших функцій, відповідаючих за безпеку. Контролери були розглянуті в таблицях 4.1-3.3.

Таблиця 3.1 – Контролер клієнтів

Назва методу	Додатковий захист	Опис методу
signUp	-	Реєстрація клієнта
signIn	-	Авторизація клієнта, результатом є отримання JWT
get	JWT	Отримання загальної інформації про клієнта із системи
getPublicKey	JWT	Отримання публічного ключа для інтеграції з точкою доступу

Таблиця 3.2 – Контролер карт платників

Назва методу	Додатковий захист	Опис методу
createCard	Для зінтегрованих клієнтів – підпис запиту алгоритмом RSA-2048. Для платіжної сторінки відсутній, оскільки він не потрібен	Створення картки платника
getCard	Для зінтегрованих клієнтів – підпис запиту алгоритмом RSA-2048. Для платіжної сторінки відсутній, оскільки він не потрібен	Отримання інформації про карту платника

Таблиця 3.3 – Контролер інвойсів

Назва методу	Додатковий захист	Опис методу
createInvoice	JWT або підпис запиту алгоритмом RSA-2048	Створення інвойсу
getInvoice	JWT або підпис запиту алгоритмом RSA-2048	Отримання інформації про інвойс
invoicePay	JWT або підпис запиту алгоритмом RSA-2048	Оплата інвойсу
checkStatus	JWT або підпис запиту алгоритмом RSA-2048	Отримання статусу проходження аутентифікації
confirmPayment	JWT або підпис запиту алгоритмом RSA-2048	Підтвердження LOOKUP



completePayment	JWT або підпис запиту алгоритмом RSA-2048	Завершення преавторизаційного платежу
-----------------	--	---

Описані контролери є відкриті контролери як для готових рішень платіжної сторінки, так і для повної інтеграції клієнта зі своєю системою. Присутні також закриті контролери, які відповідають за додаткові налаштування клієнтів, системи, та можливості надані в адміністративній панелі.

### 3.1.2 Опис мікросервісу InvoiceManager

Даний мікросервіс включає в собі роботу з інвойсами, платежами, зв'язку платежів з його реквізитами.

Нижче в таблиці 3.4 описано структуру сутності інвойсів.

Таблиця 3.4 – Структура сервісу інвойсів

Назва властивості чи методу	Опис властивості чи методу
paymentService	Композиція класу сервісу платежів, опис якого знаходиться в таблиці 3.5
clientServicesService	Композиція класу реквізитів клієнта, опис якого знаходиться в таблиці 3.6
get	Отримання інформації про інвойс
checkStatus	Отримання статусу аутентифікації платежу
invoicePay	Оплата інвойсу
confirmPayment	Підтвердження LOOKUP
completePayment	Завершення преавторизаційного платежу

Таблиця 3.5 – Структура сервісу платежу

Назва властивості чи методу	Опис властивості чи методу
repository	Властивість, композиція класу підключення до БД
psm	Композиція класу зв'язку платежу з реквізитами
routeService	Композиція класу роутів
terminalService	Композиція класі терміналів
create	Створення платежу
calculateCommission	Розрахунок комісії в залежності від тіла платежу
checkStatus	Отримання статусу проходження 3DS перевірки
confirmPayment	Підтвердження аутентифікації LOOKUP
completePayment	Завершення преавторизаційного платежу

Таблиця 3.6 – Структура сервісу зв'язку платежу з реквізитами

Назва властивості чи методу	Опис властивості чи методу
psmRepository	Композиція класу підключення до БД
create	Створення зв'язку
get	Отримання зв'язку

Сутність зв'язку платежу з реквізитами потрібна, щоб покрити кейс, коли оплата має відбуватись одночасно за різні послуги. Приклад – оплата комунальних послуг. В цьому кейсі усі отримувачі мають різний IBAN. Тож сутність описує розділення абстрактної сутності платежів на менші частинки, для чіткого розподілення винагороди між усіма отримувачами.

3.1.3 Опис мікросервісу AIModel

3.1.4 Опис мікросервісу Reporter

3.1.5 Опис мікросервісу AdminManager

3.2 Функціональні вимоги до серверної частини

3.3. Функціональні вимоги до інтерфейсів

3.3.1 Функціональні вимоги до платіжної сторінки

3.3.2 Функціональні вимоги до адміністративної панелі

3.3.3 Функціональні вимоги до особистого кабінету клієнта

#### 4 ТЕСТУВАННЯ

## 5 РОЗРОБЛЕННЯ СТАРТАП ПРОЄКТУ

### 5.1 Опис ідеї проєкту

Еквайрингова платформа, яка використовує штучний інтелект для виявлення шахрайства, пропонує сучасне рішення для оптимізації процесів обробки платежів. Основною метою проєкту є надання зручного, функціонального та безпечного інструменту для створення, управління та проведення транзакцій через інвойси.

Для оцінки потенціалу впровадження проєкту та визначення ключових ринкових можливостей, розроблено опис базової ідеї стартапу. У ньому деталізуються зміст пропозиції, напрями застосування платформи в різних галузях та вигоди, які отримають користувачі.

В таблиці 5.1 наведено основні характеристики стартапу, включаючи унікальність рішення, можливості адаптації до різних сегментів ринку та користь для цільової аудиторії.

Таблиця 5.1 – Опис ідеї стартап-проєкту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Еквайрингова платформа з особистим кабінетом та можливістю створення посилань на різні типи інвойсів	1. Послуги фінансових технологій (FinTech)	Автоматизація створення та обробки інвойсів, що знижує витрати часу на ручну роботу
Платформа забезпечує створення та проведення оплат по мультиреквізитним, одноразовим і багаторазовим інвойсам	2. Інтернет-торгівля (e-commerce)	Забезпечення зручності для клієнтів через підтримку мультиреквізитності та різних способів оплати

Унікальність рішення в застосуванні штучного інтелекту для детекції фроду	3. Бухгалтерські та платіжні системи підприємств	Підвищення рівня безпеки транзакцій за допомогою алгоритмів ШІ для виявлення шахрайських дій
Підтримка різних типів бізнесу: малий, середній, великий	4. Криптовалютні платформи	Адаптація до потреб компаній різного масштабу, від малого бізнесу до корпорацій
Простий та інтуїтивно зрозумілий інтерфейс особистого кабінету	5. Онлайн-сервіси з підписками (SaaS-платформи)	Спрощення взаємодії для користувачів через доступний і зрозумілий інтерфейс
Підтримка регулярних платежів та налаштування автоматичних інвойсів	6. Роздрібна торгівля з інтеграцією платіжних рішень	Зручність повторних платежів для постійних клієнтів через автоматизацію процесів
Гнучка інтеграція платформи з існуючими системами підприємств	7. Послуги фінансових технологій (FinTech)	Легка інтеграція з іншими сервісами або бухгалтерськими програмами, що знижує витрати на адаптацію

Для оцінки конкурентоспроможності еквайрингової платформи проведено порівняльний аналіз її техніко-економічних характеристик із основними конкурентами: LiqPay, UAPAY та WayForPay. У таблиці 5.2 визначено сильні,

слабкі та нейтральні сторони проекту, що дозволяє сформувати стратегічні пріоритети для вдосконалення платформи.

Таблиця 5.2 – Технологічна здійсненність ідеї проекту

№	Техніко-економічні характеристики ідеї	Товари/концепції конкурентів				Слаб-ка сторо-на	Нейра-льна сто-рона	Силь-на сто-рона
		Мій про-єкт	Конку-рент LiqPay	Конку-рент UAPA Y	Конку-рент WayForPay			
1	Швидкість обробки платежів	Ви-сока	Висок а	Висок а	Висока		+	
2	Зручність інтерфейсу	Ви-сока	Сере-дня	Сере-дня	Висока			+
3	Інтеграція ІІІ для виявлення шахрайства	+	-	-	-			+
4	Гнучність комісій	Ви-сока	Сере-дня	Сере-дня	Середня			+
5	Точність виявлення шахрайства	Ви-сока	Низь-ка	Немає	Немає			+
6	Вартість інтергації	Низь-ка	Сере-дня	Сере-дня	Висока			+
7	Доступність API	+	+	+	+		+	
8	Підтримка e-commerce платежів	Усі	Усі	Усі	Усі		+	
9	Можливість інтеграції	Не-має	Є	Є	Є	+		

	GooglePay/Apple Pay							
10	SDK для мобільної інтеграції	Не-має	€	€	€	+		

### 5.3 Технологійний аудит ідеї проєкту

Таблиця 5.3 – Технологічна здійсненність ідеї проєкту

№	Ідея проєкту	Технології реалізація	Наявність технологій	Доступність технологій
1.	Система управління платежами	Інтеграція з банками-екваєрами	Існують. Потребують інтеграції	Потребують договорів
2.	Відкритий API	Розробка API для роботи з платіжними системами	Існують	Доступні
3.	Алгоритми штучного інтелекту для виявлення шахрайства	TensorFlow, моделі Deep Learning	Існують	Доступні
4.	Особистий кабінет	Фронтенд і бекенд розробка	Потрібно розробити	Доступні
5.	Реєстрація клієнта	Фронтенд і бекенд розробка, а	Потрібно розробити	Потребують договорів



		також інтеграція з відкритими реєстрами		
Основні компоненти MVP включають інтеграцію з банками-екваєрами, базовий функціонал API, а також ключові алгоритми фроду на базі відкритих бібліотек ІІІ.				

За результатами аналізу технологічна реалізація проєкту можлива, але з певними обмеження.

### 5.3 Аналіз ринкових можливостей запуску стартап-проєкту

Для оцінки потенціалу ринкових можливостей стартап-проєкту важливо провести попередній аналіз стану ринку. Це дозволяє зрозуміти поточний стан конкурентного середовища, обсяг ринку, динаміку його розвитку, а також можливі бар'єри для входу та вимоги до сертифікації. Нижче наведено таблицю 3.3 ключові показники, що допомагають оцінити ринок та сформулювати стратегічний план впровадження продукту.

Таблиця 5.4 – Попередня характеристика потенційного ринку стартап-проєкту

№	Показники стану ринку (найменування)	Характеристика
1.	Кількість головних гравців, од	Близько 10 ключових гравців, включаючи LiqPay, UAPAY, WayForPay
2.	Загальний обсяг продаж, грн/ум.од	Понад 200 млрд грн на рік (обсяг електронних платежів в Україні)

3.	Показники стану ринку (найменування)	Характеристика
4.	Динаміка ринку (якісна оцінка)	Зростає через активне впровадження фінтех-рішень та електронної комерції
5.	Наявність обмежень для входу	Юридичні: реєстрація як ТОВ або ФОП. Специфічні ліцензії не потрібні
6.	Специфічні вимоги до стандартизації та сертифікації	Відповідність PCI DSS (сертифікація забезпечує безпеку даних карток)
7.	Середня норма рентабельності в галузі, %	10–30% залежно від обсягу транзакцій та структури витрат

В таблиці 5.5 наведено потенційні групи клієнтів, характеристики, які покриває потреба цих груп системою та орієнтовний перелік вимог до продукту.

Таблиця 5.5 – Характеристика потенційних клієнтів стартап-проєкту

№	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1.	Безпека транзакцій, зниження рівня шахрайства, швидкість обробки платежів	Малий та середній бізнес (e-commerce, фізичні магазини), Фінтех-компанії, Клієнти великих корпорацій	Малий бізнес шукає простоту у впровадженні. Банки потребують інтеграції з іншими системами та високого рівня безпеки	До продукції: інтуїтивно зрозумілий інтерфейс, точність у виявленні фроду, швидкість обробки До

				постачальника: технічна підтримка, наявність ліцензій, довіра до бренду
--	--	--	--	--

Після визначення груп клієнтів, було проведено аналіз ринкового середовища, а саме факторів загроз та можливостей, які наведено в таблицях 5.6-5.7.

Таблиця 5.6. – Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1.	Висока конкуренція	Наявність сильних гравців на ринку, що вже мають лояльну клієнтську базу	Використання сучасних технологій для підвищення точності та швидкості обробки транзакцій
2.	Складність отримання ліцензій PCI DSS	Регуляторні вимоги можуть затягнути процес виходу на ринок	Найм фахівців з юридичного супроводу та партнерство з регуляторами
3.	Технічні проблеми	Можливі збої у роботі платформи, особливо на етапі запуску	Тестування платформи перед запуском, найм висококваліфікованих технічних спеціалістів

4.	Регуляторні бар'єри	Високі витрати на відповідність нормам знижують рентабельність	Залучення інвесторів для покриття початкових витрат
5.	Брак клієнтів	Відсутність користувачів може призвести до збитків через договори і оплату послуг партнерів	Активна маркетингова кампанія для залучення клієнтів і партнерів

Таблиця 5.7 – Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1.	Зростання попиту на безпечні транзакції	Бізнеси все більше шукають рішення для зниження шахрайства та підвищення безпеки фінансових операцій	Розробка та інтеграція інноваційних алгоритмів на основі штучного інтелекту для виявлення фроду
2.	Тенденція діджиталізації бізнесу	Зростання кількості e-commerce і потреба у платіжних рішеннях	Просування продукту через партнерства з інтернет-магазинами та платформами електронної комерції
3.	Інтерес інвесторів до фінтех-проектів	Фінансові компанії та венчурні фонди активно інвестують у новітні технології	Активне залучення інвесторів через презентації, демонстрацію потенційного прибутку і високої рентабельності
4.	Розвиток партнерських мереж	Можливість інтеграції з іншими платформами	Налагодження партнерств із банками, фінтех-

		(банківськими сервісами, платіжними системами)	компаніями та глобальними платіжними системами
5.	Попит на прості та зручні рішення	Малий та середній бізнес шукає платформу з інтуїтивним інтерфейсом	Забезпечення простого дизайну і максимальної зручності для користувачів

В таблиці 5.8 наведено аналіз конкуренції на ринку.

Таблиця – 5.8 Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства
1. Тип конкуренції: олігополія	Ринок онлайн-еквайрингу в Україні домінують кілька великих гравців (LiqPay, UAPAY, WayForPay)	Використання унікальних інновацій (ШІ для детекту фроду), пропозиція конкурентних цін та функцій
2. Рівень конкурентної боротьби: національний	Ринок обмежений межами України, основні гравці — українські фінтех-компанії	Зосередження на потребах українських користувачів, локалізація функцій та високий рівень підтримки клієнтів
3. За галузевою ознакою: внутрішньогалузева	Конкуренція між фінтех-компаніями, які надають послуги онлайн-еквайрингу	Виділення через технологічні інновації, особливо ШІ для фроду, і покращення функціоналу
4. Конкуренція за видами товарів: товарно-видова	Основний акцент на послугах онлайн-платежів із	Баланс між функціональністю, зручністю та конкурентною ціною

	функціональними відмінностями	
5. За характером конкурентних переваг: цінова та нецінова	Основні переваги — доступна ціна, інновації (ШІ для виявлення фроду) та зручність використання	Пропонувати конкурентні ціни та унікальний функціонал для залучення клієнтів
6. За інтенсивністю: немарочна	Ваша платформа не має брендової впізнаваності, в той час як конкуренти мають сформовані бренди	Інвестувати у маркетинг, брендинг і побудову довіри серед клієнтів

Таблиця 5.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	LiqPay, UAPAY, WayForPay	Компанії, які можуть увійти на ринок з подібними послугами	Постачальники серверів, хмарних послуг, API для AI	Українські підприємці, онлайн-магазини, фінансові компанії	Альтернативні методи оплати (банківські перекази, криптовалюти, готівковий розрахунок)
Висновки	Інтенсивність конкуренції висока	Бар'єри входження високі через	Постачальники можуть диктувати умови через	Клієнти диктують вимоги до безпеки,	Замінники можуть становити загрозу через

	через велику кількість реалізованих систем	необхідність інвестицій та технічної складності	обмежену кількість якісних рішень	зручності та швидкості роботи	дешевизну та доступність
--	--	---	-----------------------------------	-------------------------------	--------------------------

Таблиця 5.10 – Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування
1.	Використання штучного інтелекту	Забезпечує високу точність і швидкість виявлення фроду, що відсутнє у конкурентів
2.	Інтеграція з іншими системами	Легке впровадження в існуючу інфраструктуру бізнесу завдяки готовим SDK
3.	Висока швидкість обробки даних	Забезпечує мінімізацію часу транзакцій і кращий клієнтський досвід
4.	Відповідність регуляторним вимогам	Сертифікація НБУ та PCI DSS підвищує довіру споживачів і відповідає законодавству
5.	Зручність використання	Інтуїтивно зрозумілий інтерфейс і адаптивність під потреби різних користувачів
6.	Конкурентна ціна	Оптимальний баланс між ціною і функціональністю порівняно з іншими платформами

За результатами обґрунтування факторів конкурентоспроможності було оцінено і порівняно проєкт із конкурентами в таблиці 5.11.

Таблиця 5.11 – Порівняльний аналіз сильних та слабких сторін «Еквайрингової платформи»

№	Фактор конкуренто-спроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з UAPAY, LiqPay, WayForPay						
			-3	-2	-1	0	+1	+2	+3
	Використання штучного інтелекту	20	UAPAY, LiqPay, WayForPay						
	Інтеграція з іншими системами	10			UAPAY			LiqPay, WayForPay	
	Висока швидкість обробки даних	19			UAPAY, LiqPay, WayForPay				
	Відповідність регуляторним вимогам	20						UAPAY, LiqPay, WayForPay	
	Зручність використання	15						UAPAY, LiqPay, WayForPay	
	Конкурентна ціна	14	LiqPay		UAPAY, WayForPay				



Згідно результатів порівняльного аналізу зроблено SWOT-аналіз, який наведений в таблиці 5.12.

Таблиця 5.12 – SWOT-аналіз стартап-проєкту

Сильні сторони: швидкість обробки платежів, безпека за рахунок використання ШІ для виявлення фроду	Слабкі сторони: відсутність інтеграцій з державними реєстрами, які надають інформацію про клієнта, за його діяльністю. Відсутність SDK
Можливості: співпраця з реєстрами та НБУ, для надання платіжної інформації, можливість розширення функціоналу системи, та залучення мобільного розроблення для цього	Загрози: великі конкуренти, які мають клієнтську базу, відсутність фінансування, нестабільність системи, залежність від партнерів

Результат SWOT-аналізу є розроблено альтернативу ринкової поведінки, яка наведена в таблиці 5.13.

Таблиця 5.13 – Альтернативи ринкового впровадження стартап-проєкту

№	Альтернатива ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Розробка SDK для інтеграції системи з платіжними сервісами клієнтів	Висока (можливе залучення грантів або інвесторів)	6-9 місяців
2.	Інтеграція з державними реєстрами для збору даних про клієнтів (через співпрацю з НБУ)	Середня (залежить від доступу до державних API)	9-12 місяців
3.	Запуск мобільного застосунку для бізнес-клієнтів з аналітичним модулем	Середня (потрібне фінансування для мобільної розробки)	6-8 місяців

4.	Партнерство з банками та фінансовими організаціями для пілотного запуску продукту	Висока (існує зацікавленість банків у антифрод-рішеннях)	3-6 місяців
5.	Запуск маркетингової кампанії з фокусом на безпеку платежів і швидкість обробки	Середня (залежить від наявного бюджету на рекламу)	2-4 місяці
6.	Побудова відносин із потенційними інвесторами та венчурними фондами для отримання фінансування	Середня (інтерес до FinTech рішень є, але потрібні переконливі результати)	3-6 місяців
7.	Вивчення конкурентів та оновлення функціоналу відповідно до їхніх ключових переваг	Висока (можливо внутрішні ресурси достатні)	Постійний процес (періодично)
8.	Проведення технічного аудиту системи для підвищення стабільності та зменшення залежності від партнерів	Висока (можливе залучення власних інженерів)	2-3 місяці

#### 5.4 Розроблення ринкової стратегії проєкту

Таблиця 5.14 – Вибір цільових груп споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
---	--	---	---	--------------------------------------	--------------------------

	Малі та середні підприємства, які використовують онлайн-платежі	Висока (зацікавленість у зниженні ризиків фроду)	Середній, але стабільний	Середня (конкуренція з локальними системами)	Висока (мінімальні бар'єри входу)
	Великі ритейлери з онлайн-продажами	Середня (залежить від ціни та функціональності)	Високий	Висока (домінують міжнародні платформи)	Середня (потрібні партнерства з банками)
	Фінансові компанії (небанківські установи, кредитні спілки)	Середня (залежить від технічної інтеграції)	Високий	Висока (велика кількість спеціалізованих рішень)	Середня (потрібен технічний ресурс)
	Стартапи у FinTech сфері	Висока (відкритість до нових технологій)	Низький, але можливий швидкий ріст	Низька (сегмент вузький)	Висока (спрощена співпраця в індустрії)
	Державні установи та компанії, що працюють за державними контрактами	Низька (потребують сертифікацій та перевірок)	Середній	Середня (конкуренція з великими гравцями)	Низька (бюрократичні бар'єри)

За результатами дослідження було обрано:

- а) малі та середні підприємства, які використовують онлайн-платежі;

б) великі ритейлери з онлайн-продажами.

Визначення базової стратегії розвитку наведено в таблиці 5.15.

Таблиця 5.15 – Визначення базової стратегії розвитку

№	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1.	Орієнтація на малі та середні підприємства (SME), які використовують онлайн-платежі	Концентрований маркетинг	Доступна вартість послуг. Простота інтеграції. AI-виявлення фроду, що мінімізує ризики	Стратегія спеціалізації: зосередження на потребах SME та забезпечення якісного обслуговування
2.	Орієнтація на великі ритейлери з онлайн-продажами	Диференційований маркетинг	Висока продуктивність системи. Гнучкість для адаптації до потреб ритейлерів. Надійна інфраструктура.	Стратегія диференціації: створення унікальних функцій для великих клієнтів, які підвищують їхню лояльність

Стратегія спеціалізації зосереджена на вузькому сегменті клієнтів із пропозицією максимально адаптованого продукту. Акцент зроблено на простоті

інтеграції, швидкості впровадження, доступності послуг та зниженні витрат для клієнта.

Стратегія диференціації з основним акцентом на надання ритейлерам унікальних можливостей, таких як швидкість обробки великої кількості транзакцій та налаштування продукту під специфічні потреби великих клієнтів.

Далі було вибрано стратегії конкурентної поведінки, які наведено в таблиці 5.16.

Таблиця 5.16 – Визначення базової стратегії конкурентної поведінки

№	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1.	Проект є «першопрохідцем» на ринку для певного сегмента (наприклад, AI-аналіз платежів без 3DS, та виявлення фроду)	Переважно шукати нових споживачів, створюючи попит серед компаній, які ще не використовували такі сервіси	Ні, компанія створює власні унікальні рішення (AI-виявлення, простота інтеграції, адаптивність до ринкових умов)	Стратегія лідера: розширення первинного попиту через впровадження нових функцій та формування попиту на інноваційний продукт
2.	Проект не є «першопрохідцем», але	Фокус на залученні	Так, частково копіюватиме	Стратегія виклику

	має значний потенціал в окремих нішах ринку (наприклад, малий бізнес)	клієнтів конкурентів, пропонуючи вигідніші умови та унікальні функції	характеристики конкурентів, але вдосконалюючи їх	лідера: флангова атака на слабкі сторони конкурентів, включаючи зниження цін та поліпшення сервісу
--	---	---	--	--

У разі першопрохідця проекту доцільно акцентувати на формуванні попиту, пропонуючи революційні рішення, які ще не присутні на ринку. Це дозволить швидко завоювати лояльність клієнтів і закріпити позиції.

Для сегментів, де конкуренція вже існує, ефективною буде флангова атака. Це дозволить використати слабкі сторони лідерів ринку, забезпечуючи конкурентоспроможність.

В таблиці 5.17 наведено результат розроблення стратегії позиціонування.

Таблиця 5.17 – Визначення стратегії позиціонування

№	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)

1.	Надійність і безпека (ефективне виявлення шахрайства), простота інтеграції, підтримка швидких транзакцій	Диверсифікація, створення унікальних продуктів для різних сегментів клієнтів	Унікальна технологія виявлення шахрайства без 3DS. Простота інтеграції API. Висока швидкість обробки транзакцій.	Безпека, інноваційність, зручність
2.	Мінімальні витрати на інтеграцію, підтримка малих бізнесів, адаптація до специфіки локальних ринків	Концентрація на нішевих сегментах	Орієнтація на малий та середній бізнес. Ефективна підтримка локальних ринків. Доступна вартість інтеграції.	Доступність, локалізація, простота

### 5.5 Розроблення маркетингової програми стартап-проєкту

Далі було сформовано маркетингову концепцію товару, яка наведена в таблиці 5.18, який отримає клієнт.

Таблиця 5.18 – Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
---	---------	----------------------------	-------------------------------------

	Захист транзакцій від шахрайства	Підвищення рівня безпеки платежів за допомогою інноваційної AI-технології	Існуючі: передова технологія виявлення шахрайства, відсутність потреби у 3DS. Необхідні: розширення алгоритмів для аналізу специфіки локальних ринків
	Проста інтеграція платіжних рішень у бізнес-процеси клієнтів	Мінімальні витрати часу та ресурсів на інтеграцію API	Існуючі: гнучкість і простота в налаштуванні. Необхідні: додаткові інструкції та інструменти для самостійної інтеграції
	Швидкість обробки транзакцій	Зменшення часу на виконання платежів і зниження ризику втрати клієнтів	Існуючі: високошвидкісна обробка даних. Необхідні: забезпечення масштабованості для великих обсягів даних
	Доступна вартість інтеграції та обслуговування	Економія коштів для малого та середнього бізнесу	Існуючі: конкурентні ціни. Необхідні: акції для нових клієнтів і програма лояльності
	Підтримка локальних особливостей бізнесу та регуляторних вимог	Можливість адаптації під конкретний ринок	Існуючі: модульний підхід до налаштувань. Необхідні: більше можливостей кастомізації та відповідність локальним регуляціям



Таблиця 5.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Задоволення потреба у проведенні платежів, та відсутності необхідності проходження будь-яких сертифікацій, щоб мати змогу працювати з платежами		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	Економічні: низька вартість інтеграції  Призначення: можливість проводити безпечні платежі для e-commerce продажів  Ергономічні: зручний інтерфейс	-/+	+ / + / + / + / +
	Якість: постійне тестування, та зворотній зв’язок		
	Марка: FreePay		
III. Товар із підкріпленням	До продажу: маркетингова кампанія		
	Після продажу: підтримка та розвиток системи		
За рахунок ліцензії і реєстрації бренду			

Наступним кроком є визначення меж встановлення ціни на товар, який наведено в таблиці 5.20, оскільки це є основним прибутком компанії.

Таблиця 5.20 – Визначення меж встановлення ціни

№	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1.	Платежі через стандартні	Інші AI-рішення для запобігання	Малий та середній бізнес з річним	Нижня межа ціни: \$40/місяць

платіжні системи (без AI). Ціна за транзакцію – \$0.30 - \$1.00	шахрайства, які мають схожу функціональність. Ціна послуг - \$50 - \$200/місяць	доходом до \$5 млн. середня заробітна плата працівників \$1,500 - \$4,000/місяць.	(для малого бізнесу) Верхня межа ціни: \$150/місяць (для середнього бізнесу)
--	--	---	---

Далі в таблиці 5.21 є визначення оптимальної системи збуту, в межах якого приймається рішення.

Таблиця 5.21 – Формування системи збуту

№	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1.	Цільова аудиторія: Малі та середні підприємства, які активно використовують онлайн-платформи для бізнесу, шукають прості та ефективні рішення для інтеграції у свої процеси. Зазвичай це довгострокові рішення на основі підписки.	Функції постачальника: розробка та доставка програмного забезпечення, технічна підтримка, надання консультацій та навчання, оновлення продукту	Глибина каналу: Канал прямий, без великих посередників, оскільки продукт є спеціалізованим і потребує прямої взаємодії з кінцевими споживачами	Власна система збуту: Прямий збут через онлайн-платформи, прямі продажі через вебсайт, підтримка через телефон чи чат. Альтернатива — залучення сторонніх консультантів для надання консультацій.

Останньою складовою маркетингової програми є розроблення концепції маркетингу, яку наведено в таблиці 5.22.

Таблиця 5.22 – Концепція маркетингових комунікацій

№	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
	Цільова аудиторія: Малі підприємства, які шукають ефективні та економічні рішення для автоматизації бізнес-процесів, зокрема через онлайн-інструменти. Вони схильні до використання сучасних технологій і прості та	Канали комунікацій: 1. Соціальні мережі (Facebook, Instagram, LinkedIn) 2. Електронна пошта 3. Онлайн-реклама (Google Ads, контекстна реклама) 4. Вебінари та онлайн-курси 5. Рекомендації партнерів і	Ключові позиції: 1. Простота та ефективність товару 2. Надійність та підтримка 3. Своєчасність та доступність	Завдання рекламного повідомлення: 1. Підкреслити основні вигоди використання продукту (збільшення продуктивності, зменшення витрат). 2. Надати чітку інформацію про переваги перед конкурентами (найкраща ціна, простота інтеграції).	Концепція рекламного звернення: 1. Емоційний акцент: "Простота, яка допоможе вам зекономити час і гроші". 2. Інформативний акцент: "Лідери ринку обирають наше рішення для автоматизації бізнесу" — підкреслюючи надійність і

	доступні рішення	клієнтів (сарафанне радіо)			технологіч- ність продукту.
--	---------------------	----------------------------------	--	--	-----------------------------------

### Висновки до розділу 5

У даному розділі розроблена маркетингова програма для стартап-проекту, яка охоплює всі основні елементи стратегії просування продукту на ринок. На основі аналізу конкурентоспроможності, потреб споживачів та характеристик ринку було сформовано стратегію позиціонування та цінову політику, що забезпечують оптимальний баланс між конкурентоспроможністю і прибутковістю. Визначено ефективні канали збуту та комунікації, що сприяють залученню та утриманню цільової аудиторії. Розроблена програма дає змогу стартапу максимально адаптуватися до ринкових умов та забезпечити його успішне впровадження.

## ВИСНОВКИ

У ході виконання магістерської роботи проведено комплексний аналіз існуючих рішень для еквайрингових платформ, зокрема LiqPay, UAPAY та WayForPay. Це дозволило визначити основні підходи й функціональні можливості, які забезпечують ефективну обробку платежів та мінімізацію ризиків для учасників фінансових операцій. Розгляд основних типів платежів, таких як p2p та e-commerce, дозволив вивчити їхні особливості, а також визначити можливості для інтеграції у різні бізнес-моделі.

Дослідження різновидів транзакцій для e-commerce, включаючи авторизаційні, преавторизаційні та рекурсивні платежі, підкреслило потребу в гнучкому та адаптивному підході до обробки транзакцій. Це, своєю чергою, обумовило вибір архітектури серверної частини, яка здатна ефективно масштабуватися й забезпечувати стабільну роботу платформи за різних умов навантаження.

У процесі дослідження також було вивчено вимоги до платіжних систем, зокрема, щодо PCI DSS сертифікації, що є критичним для забезпечення безпеки даних користувачів. Це дозволило побудувати платформу, що відповідає міжнародним стандартам безпеки, а також адаптувати систему до вимог щодо захисту персональних даних.

Розроблена платформа побудована на основі мікросервісної архітектури, що забезпечує гнучкість і масштабованість, а також спрощує її подальший розвиток. Для забезпечення надійної взаємодії між компонентами системи була використана шина повідомлень, що дозволяє ефективно обробляти великий обсяг транзакцій і даних. Платформа використовує сучасні технології, зокрема TypeScript та Python, що дозволяє збудувати потужну систему для обробки та аналізу платежів, а також для реалізації алгоритмів штучного інтелекту для виявлення фроду.

Завдяки використанню мікросервісної архітектури, система може бути легко адаптована для роботи з новими платіжними провайдерами та інтеграцією нових функціональних можливостей. Обрана реляційна база даних спроектована таким

чином, щоб забезпечити надійність і цілісність даних, а також масштабованість для роботи з великими обсягами транзакцій.

Розроблена еквайрингова платформа зі штучним інтелектом має значний потенціал для подальшого розвитку та масштабування. Вона здатна задовольнити потреби як малого, так і великого бізнесу, що працює в сфері електронної комерції, надаючи ефективні інструменти для боротьби з шахрайством і забезпечення безпеки транзакцій. Рішення, яке включає в себе новітні технології для виявлення фроду та високу надійність системи, дозволяє прогнозувати успіх цієї платформи на ринку України.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Про LiqPay – Режим доступу до ресурсу:  
<https://www.liqpay.ua/doc>
2. Про WayForPay – Режим доступу до ресурсу:  
<https://wiki.wayforpay.com/>
3. Про UAPAY – Режим доступу до ресурсу:  
<https://uapayua.atlassian.net/wiki/spaces/AC/pages/753795086>
4. Про E-commerce – Режим доступу до ресурсу:  
<https://interkassa.com/blog/shho-take-elektronna-komerciya-e-commerce-dlya-pochatkivciv>
5. Про LOOKUP – Режим доступу до ресурсу:  
<https://docs.nium.com/apis/docs/otp-based-3ds-authentication-flow>
6. Про 3DS – Режим доступу до ресурсу:  
[https://uk.wikipedia.org/wiki/3-D\\_Secure](https://uk.wikipedia.org/wiki/3-D_Secure)
7. Про PCI DSS сертифікацію – Режим доступу до ресурсу:  
<https://getpci.com/>
8. Режим доступу до ресурсу: <https://www.dolthub.com/blog/2024-07-16-mysql-postgres-sysbench-latency/>
9. Режим доступу до ресурсу: <https://blog.pipeops.io/postgresql-and-mysql-which-is-better-a-2024-comparison/>
10. Режим доступу до ресурсу: <https://www.pingcap.com/article/mysql-vs-postgresql-a-complete-comparison-in-2024/>
11. Про використання mermaid – Режим доступу до ресурсу:  
<https://mermaid.js.org/>
12. Про TensorFlow – Режим доступу до ресурсу:  
<https://www.tensorflow.org/>
13. Про NodeJS – Режим доступу до ресурсу: <https://nodejs.org/en>

14. Про пакетні менеджери – Режим доступу до ресурсу:  
<https://romanglushach.medium.com/comparing-npm-yarn-and-pnpm-package-managers-which-one-is-right-for-your-distributed-project-to-4d7de2f0db8e>
15. Про Fastify фреймворк – Режим доступу до ресурсу:  
<https://www.npmjs.com/package/fastify#benchmarks>
16. Про ORM – Режим доступу до ресурсу:  
<https://www.npmjs.com/package/drizzle-orm>
17. Про роботу з RabbitMQ – Режим доступу до ресурсу:  
<https://www.npmjs.com/package/amqplib>
18. Про бібліотеку валідації – Режим доступу до ресурсу:  
<https://www.npmjs.com/package/zod>
19. Про JWT – Режим доступу до ресурсу:  
<https://www.npmjs.com/package/jsonwebtoken>
20. Про TypeScript – Режим доступу до ресурсу:  
<https://www.npmjs.com/package/typescript>
21. Про роботу з Redis Режим доступу до ресурсу:  
<https://www.npmjs.com/package/ioredis>
22. Про бібліотеку для генерації звітів – Режим доступу до ресурсу:  
<https://www.npmjs.com/package/iconv>
23. Про бібліотеку логування – Режим доступу до ресурсу:  
<https://www.npmjs.com/package/pino>
24. Про змінні оточення – Режим доступу до ресурсу:  
<https://www.npmjs.com/package/dotenv>