

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет інформаційних технологій

**ПОГОДЖЕНО
Декан факультету**

(підпис) **Ігор БОЛБОТ**

« ____ » _____ 2026 р.

**ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ
Завідувач кафедри комп'ютерних наук**

(підпис) **Белла ГОЛУБ**

« ____ » _____ 2026 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Програмне забезпечення системи транзакцій та розрахунків на основі блокчейну»

Спеціальність «121 Інженерія програмного забезпечення»

Освітньо-професійна програма «Інженерія програмного забезпечення»

Гарант освітньої програми
к.т.н., доцент

(підпис) **Ганна ВАЙГАНГ**

**Керівник бакалаврської
кваліфікаційної роботи**

(підпис) **Олексій СТЕПАНОВ**

Виконав

(підпис) **Микола НЕКРАШЕВСЬКИЙ**

Київ-2026

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ

І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук

К.Т.Н., доцент _____

(підпис)

Белла ГОЛУБ

«___» _____ 2025 р.

ЗАВДАННЯ

**ДО ВИКОНАННЯ БАКАЛАВРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
ЗДОБУВАЧУ**

Некрашевському Миколі Андрійовичу

(прізвище, ім'я, по батькові)

Спеціальність «121 Інженерія програмного забезпечення»

Освітньо-професійна програма «Інженерія програмного забезпечення»

Тема бакалаврської кваліфікаційної роботи: «Програмне забезпечення системи транзакцій та розрахунків на основі блокчейну»

затверджена наказом від «19» грудня 2025 р. № 3204 «С»

Термін подання завершеної роботи на кафедру «22» травня 2026 р.

Вихідні дані до бакалаврської кваліфікаційної роботи (проєкту): опис організації та принципів функціонування платіжної системи на основі блокчейну, процесів обробки транзакцій, розрахункових операцій, реєстру подій та цифрових фінансових сервісів.

Перелік питань, що підлягають дослідженню:

1. Платіжна система на основі блокчейну як об'єкт дослідження;
2. Проєктування інформаційного забезпечення для платіжної системи;
3. Розробка програмного забезпечення для транзакцій та розрахунків;
4. Рекомендації щодо впровадження, безпеки та експлуатації платіжної системи на основі блокчейну.

Дата видачі завдання «19» грудня 2025 р.

**Керівник бакалаврської
кваліфікаційної роботи**

_____ (підпис)

Олексій СТЕПАНОВ

**Завдання взяв до
виконання**

_____ (підпис)

Микола НЕКРАШЕВСЬКИЙ

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	5
ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	10
1.1 Умова завдання	10
1.2 Опис предметної області	10
1.3 Аналіз сучасних технологій блокчейну	12
1.4 Аналіз існуючих платіжних систем	14
1.5 Аналіз вимог системи	15
2 ПРОЄКТУВАННЯ БАЗИ ДАНИХ	18
2.1 Моделювання бази даних	18
2.2 Концептуальне проєктування	19
2.3 Логічне проєктування	20
2.4 Фізичне проєктування	20
2.5 ER-діаграма бази даних	21
2.6 Опис таблиць бази даних	22
2.7 Проєктування модулів блокчейну	23
3 РОЗРОБКА ПРОГРАМНОЇ ЧАСТИНИ	25
3.1 Вибір середовища розробки	25
3.2 Розробка архітектури WEB-системи	26
3.3 Розробка REST API	27
3.4 Реалізація механізмів блокчейну	28

	4
3.5 Реалізація Settlement Engine	28
3.6 Реалізація Clearing Queue	29
3.7 Реалізація Audit System	32
3.8 Реалізація JWT та 2FA	35
4 РЕКОМЕНДАЦІЇ ЩОДО ВПРОВАДЖЕННЯ ТА ЕКСПЛУАТАЦІЇ СИСТЕМИ	38
4.1 Тестування системи	38
4.2 Вимоги до апаратного та програмного забезпечення	44
4.3 Склад інсталяційного пакету	48
ВИСНОВКИ	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	56
Додаток А	58
Додаток Б	59
Додаток В	60
Додаток Г	61
Додаток Д	62
Додаток Е	63
Додаток Є	64

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – Application Programming Interface

ASP.NET – Active Server Pages .NET

Audit.NET – бібліотека журналювання подій для .NET

AWS – Amazon Web Services

BCrypt – алгоритм хешування паролів

CI/CD – Continuous Integration / Continuous Delivery

CRUD – Create, Read, Update, Delete

Docker – платформа контейнеризації застосунків

ER – Entity-Relationship

HTTPS – HyperText Transfer Protocol Secure

JWT – JSON Web Token

MVC – Model-View-Controller

MySQL – система управління реляційними базами даних

OTP – One-Time Password

ORM – Object-Relational Mapping

REST – Representational State Transfer

SHA256 – Secure Hash Algorithm 256-bit

SQL – Structured Query Language

UI – User Interface

UML – Unified Modeling Language

2FA – Two-Factor Authentication

НФ – нормальна форма

ПЗ – програмне забезпечення

СУБД – система управління базами даних

ВСТУП

У сучасних умовах цифровізації фінансового сектору блокчейн набуває все більшого поширення у сфері електронних платежів та фінансових сервісів. Традиційні платіжні системи часто мають низку недоліків, серед яких централізоване зберігання даних, залежність від банківської інфраструктури, високі комісії за проведення міжнародних транзакцій, ризики витоку інформації та недостатній рівень прозорості фінансових операцій [5]. Крім того, централізовані системи можуть бути вразливими до кібератак, технічних збоїв та несанкціонованого доступу до даних користувачів.

У зв'язку зі стрімким розвитком блокчейну та децентралізованих фінансових сервісів виникає необхідність створення сучасних програмних рішень, які забезпечують безпечне виконання транзакцій, автоматизацію процесів розрахунків, аудит фінансових операцій та високий рівень захисту інформації. Особливо актуальним є впровадження систем, які дозволяють здійснювати обробку транзакцій із використанням механізмів блокчейну, цифрових підписів та сучасних методів автентифікації.

Розробка платіжної системи для транзакцій і розрахунків на основі блокчейну дозволяє автоматизувати фінансові процеси, централізувати управління транзакціями, забезпечити прозорість фінансових операцій та значно підвищити рівень безпеки електронних платежів [4]. Система також дозволяє реалізувати механізми журналювання операцій, моніторингу транзакцій, двофакторної автентифікації та обробки черги фінансових операцій через «clearing queue».

Розроблений програмний продукт сприятиме підвищенню ефективності роботи фінансових сервісів, скороченню часу обробки транзакцій, автоматизації процесів розрахунків та зменшенню ризиків виникнення помилок під час виконання фінансових операцій.

Мета розробки. Метою бакалаврської кваліфікаційної роботи є розробка платіжної системи для транзакцій і розрахунків на основі блокчейну, яка забезпечить безпечне виконання цифрових фінансових операцій, автоматизацію процесів обробки транзакцій та управління розрахунками, аудит фінансових подій і захист даних користувачів.

Для досягнення поставленої мети необхідно вирішити наступні задачі:

- провести аналіз сучасних технологій блокчейну та платіжних систем;
- дослідити принципи функціонування системи транзакцій та розрахунків;
- спроектувати архітектуру платіжної системи на базі блокчейну;
- розробити структуру бази даних та ER-діаграму системи;
- реалізувати REST API для роботи з транзакціями та користувачами;
- реалізувати blockchain transaction processing;
- створити Settlement Engine та Clearing Queue;
- реалізувати JWT authentication та Two-Factor Authentication;
- впровадити систему аудиту фінансових операцій;
- реалізувати адміністративну панель та систему сповіщень;
- провести тестування програмного забезпечення;
- сформулювати рекомендації щодо впровадження та експлуатації системи.

Основна ідея проекту полягає у створенні сучасної платіжної системи на основі блокчейну, яка поєднує механізми блокчейн-транзакцій, управління розрахунками, ведення журналу аудиту та сучасні засоби захисту інформації. Система дозволяє централізовано управляти фінансовими транзакціями, забезпечує прозорість операцій та підтримує масштабування у хмарному середовищі.

Методи та технології. Під час розробки платіжної системи використовувалась мова програмування C# у середовищі Visual Studio 2022. Для створення WEB-застосунку використано платформу ASP.NET Core, яка дозволяє створювати масштабовані та високопродуктивні серверні застосунки [9].

Для реалізації архітектури системи використовувалися принципи MVC та Clean Architecture. Для роботи з базою даних застосовано Entity Framework Core як ORM-систему та MySQL як систему управління базами даних [10].

Для реалізації механізмів безпеки використовувалися:

- JWT Authentication;
- BCrypt password hashing;
- Two-Factor Authentication;
- HTTPS encryption;
- Role-Based Authorization.

Blockchain-механізми реалізовані із використанням алгоритму SHA256 для генерації hash-значень блоків та транзакцій.

Для журналювання системних подій та фінансових операцій використано бібліотеку Audit.NET. Для реалізації системи сповіщень та асинхронної взаємодії компонентів використовувалися SignalR та notification services.

Для розгортання системи використовувалися Docker, AWS EC2, AWS RDS та Nginx. Docker використовується для контейнеризації компонентів системи [16], а AWS забезпечує хмарну інфраструктуру для масштабування сервісів [17].

Апробація програмного продукту. Результати розробки платіжної системи були представлені на студентській науково-практичній конференції у вигляді тез доповіді на тему: «Програмне забезпечення для системи транзакцій і розрахунків на основі блокчейну».

Структура записки. Пояснювальна записка складається зі вступу, чотирьох основних розділів, висновків, списку використаних джерел та додатків.

У першому розділі «Аналіз предметної області» розглянуто сучасні - технології блокчейну, особливості функціонування системи транзакцій і розрахунків, проведено аналіз існуючих платіжних систем, визначено функціональні вимоги до системи та змодельовано основні процеси за допомогою UML-діаграм.

Другий розділ «Проектування бази даних та системи» присвячено побудові логічної та фізичної моделей бази даних, розробці ER-діаграми, проектуванню модулів блокчейну, settlement engine, clearing queue та системи безпеки.

У третьому розділі «Розробка програмної частини» описано процес створення платіжної системи на основі блокчейну, реалізацію REST API, обробку транзакцій, Settlement Engine, Audit System, JWT authentication, Two-Factor Authentication, адміністративної панелі та користувацького інтерфейсу.

У четвертому розділі «Рекомендації щодо впровадження та експлуатації системи» наведено результати тестування програмного забезпечення, визначено вимоги до апаратного та програмного забезпечення, описано склад інсталяційного пакету та рекомендації щодо впровадження платіжної системи у хмарному середовищі AWS.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Умова завдання

Основною задачею є розробка платіжної системи на основі блокчейну, яка дозволяє:

- виконувати цифрові транзакції між користувачами;
- створювати та керувати криптовалютними гаманцями;
- забезпечувати безпечну авторизацію та автентифікацію;
- вести журнал аудиту всіх фінансових операцій;
- реалізувати систему статусів транзакцій;
- підтримувати адміністративну панель управління;
- забезпечувати двофакторну автентифікацію;
- реалізувати систему сповіщень користувачів.

Система повинна бути масштабованою, безпечною та мати WEB-інтерфейс для взаємодії користувачів.

На рисунку 1.1 наведено загальну структуру платіжної системи на основі блокчейну та взаємодію основних компонентів системи.

1.2 Опис предметної області

Платіжна система на основі блокчейну — це сучасна інформаційна система для виконання цифрових фінансових операцій із використанням блокчейну та криптографічного захисту даних.

У сучасному світі цифрові фінансові сервіси стають все більш популярними. Користувачі активно використовують онлайн-платежі, мобільні гаманці, криптовалюти та електронні фінансові сервіси. У зв'язку з цим виникає

необхідність створення безпечних систем, які здатні забезпечити швидкі транзакції, захист персональних даних та прозорість фінансових операцій.

Технологія блокчейн дозволяє реалізувати децентралізований підхід до зберігання інформації [1]. Дані про транзакції зберігаються у вигляді послідовності блоків, пов'язаних між собою криптографічними механізмами.

Основними перевагами блокчейн-систем є:

- децентралізація;
- неможливість зміни історії транзакцій;
- високий рівень захисту;
- прозорість фінансових операцій;
- незалежність від центральних органів управління.

У платіжній системі користувачі можуть:

- створювати криптовалютні гаманці;
- виконувати транзакції;
- переглядати історію операцій;
- використовувати двофакторну автентифікацію;
- отримувати повідомлення про фінансові операції.

До складу системи входять наступні модулі:

- система управління користувачами;
- transaction engine;
- settlement engine;
- clearing queue;
- audit system;
- notification service;
- blockchain ledger;
- administrative panel.

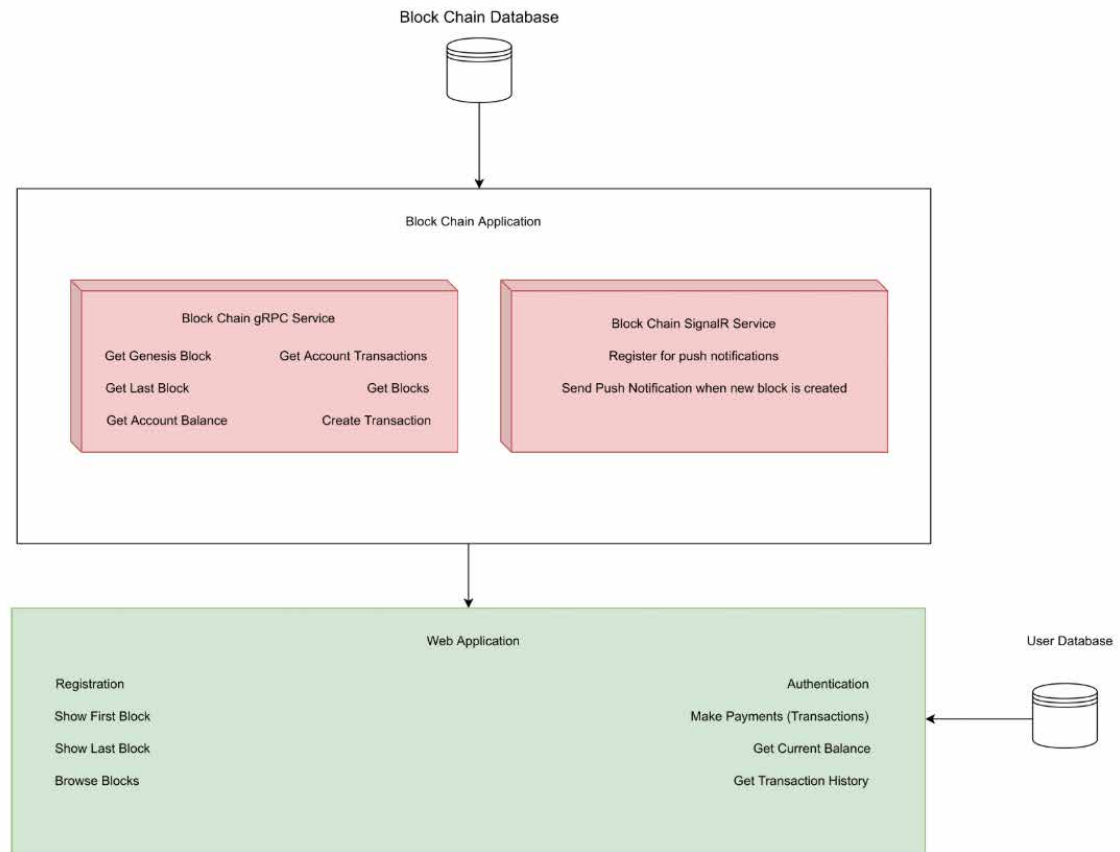


Рис. 1.1 Загальна структура платіжної системи.

1.3 Аналіз сучасних технологій блокчейну

Blockchain є технологією розподіленого реєстру, у якому інформація зберігається у вигляді ланцюга блоків [2].

Кожен блок містить:

- список транзакцій;
- часову позначку;
- hash попереднього блоку;
- власний hash.

1.4 Аналіз існуючих платіжних систем

Сучасні цифрові платіжні системи поділяються на:

- централізовані;
- децентралізовані.
- До централізованих систем належать:
 - PayPal;
 - Stripe;
 - Visa;
 - MasterCard.

Основними недоліками централізованих систем є:

- залежність від банківської інфраструктури;
- комісії;
- обмеження міжнародних переказів;
- ризики централізованого зберігання даних.

Децентралізовані blockchain-системи дозволяють усунути ці недоліки.

Використання blockchain-технологій забезпечує прозорість та незмінність фінансових операцій [4].

Централізовані платіжні системи мають залежність від центрального сервера та банківської інфраструктури [5].

На рисунку 1.3 представлено порівняння централізованої та децентралізованої blockchain архітектури.

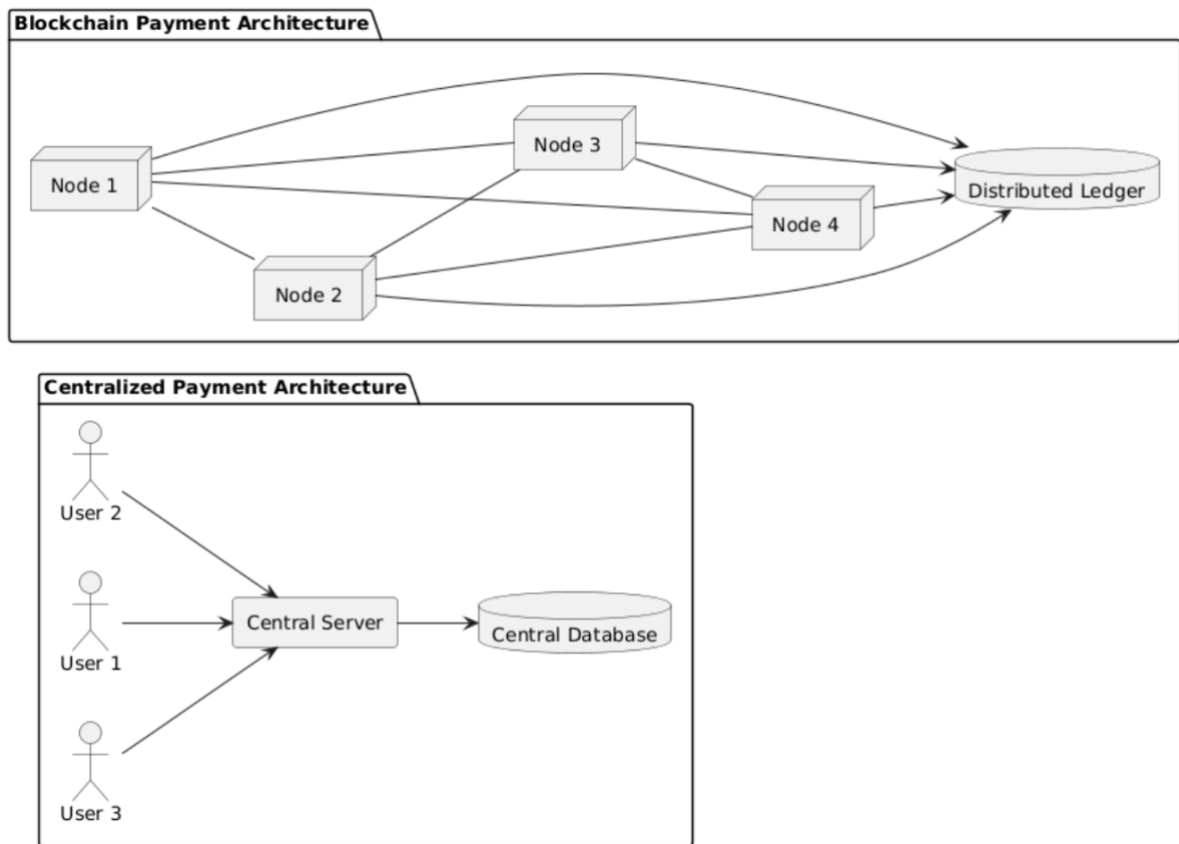


Рис. 1.3 Порівняння централізованої та блокчейн архітектури.

1.5 Аналіз вимог системи

У результаті аналізу предметної області були визначені такі ролі користувачів, як адміністратор і користувач.

Адміністратор має повний доступ до системи та може:

- переглядати всі транзакції;
- керувати користувачами;
- здійснювати аудит операцій;
- переглядати системні журнали;
- керувати статусами транзакцій;
- здійснювати моніторинг системи.

Користувач системи може:

- реєструватися та авторизовуватися;
- створювати криптовалютні гаманці;
- виконувати транзакції;
- переглядати історію операцій;
- використовувати двофакторну автентифікацію;
- отримувати системні сповіщення.

Основні функціональні вимоги

Система повинна підтримувати:

- реєстрацію та авторизацію користувачів;
- JWT-автентифікацію;
- двофакторну автентифікацію;
- створення транзакцій;
- генерацію blockchain-блоків;
- аудит фінансових операцій;
- систему сповіщень;
- адміністративну панель;
- журналювання дій користувачів.

Архітектура платіжної системи базується на багаторівневій моделі:

Presentation Layer – рівень користувацького інтерфейсу, який забезпечує взаємодію користувача із системою.

Business Logic Layer – містить бізнес-логіку системи:

- обробку транзакцій;
- перевірку цифрових підписів;
- валідацію даних;
- аудит операцій;
- управління blockchain.

Data Access Layer – забезпечує доступ до бази даних та збереження інформації про:

- користувачів;
- транзакції;
- гаманці;
- блоки;
- аудити;
- сповіщення.

Вхідними даними системи є:

- дані користувачів;
- параметри транзакцій;
- цифрові ключі;
- інформація про гаманці;
- коди двофакторної автентифікації.

Вихідними даними системи є:

- статуси транзакцій;
- історія операцій;
- системні сповіщення;
- аудиторські журнали;
- фінансова аналітика.

Поділ системи на окремі логічні рівні відповідає принципам Clean Architecture [12].

2 ПРОЄКТУВАННЯ БАЗИ ДАНИХ

2.1 Моделювання бази даних

Проектування бази даних є одним із найважливіших етапів розробки платіжної системи для розрахунків і транзакцій на основі блокчейну. ER-моделювання використовується для проектування структури реляційних баз даних [20].

Під час моделювання були визначені: Дл

- основні сутності системи;
- зв'язки між сутностями;
- типи даних;
- механізми забезпечення цілісності даних.

ER-діаграма blockchain payment system, наведена на рисунку 2.1, відображає основні сутності системи та зв'язки між ними.

Основними сутностями системи є:

- User;
- Wallet;
- Transaction;
- Block;
- AuditLog;
- Notification;
- Role;
- TwoFactorToken.

Для системи були визначені наступні типи зв'язків:

- один-до-одного;
- один-до-багатьох;
- багато-до-багатьох.

2.3 Логічне проєктування

На етапі логічного проєктування були визначені таблиці бази даних, первинні та зовнішні ключі.

Було реалізовано:

- нормалізацію даних;
- систему зовнішніх ключів;
- індексацію таблиць;
- механізми оптимізації запитів.

2.4 Фізичне проєктування

Фізичне проєктування включає:

- створення таблиць;
- визначення типів даних;
- створення індексів;
- налаштування продуктивності MySQL.

2.5 ER-діаграма бази даних

ER-діаграма зображена на рисунку 2.5 відображає структуру бази даних платіжної системи.

На ER-діаграмі присутні:

- таблиці;
- зв'язки;
- первинні ключі;
- зовнішні ключі.

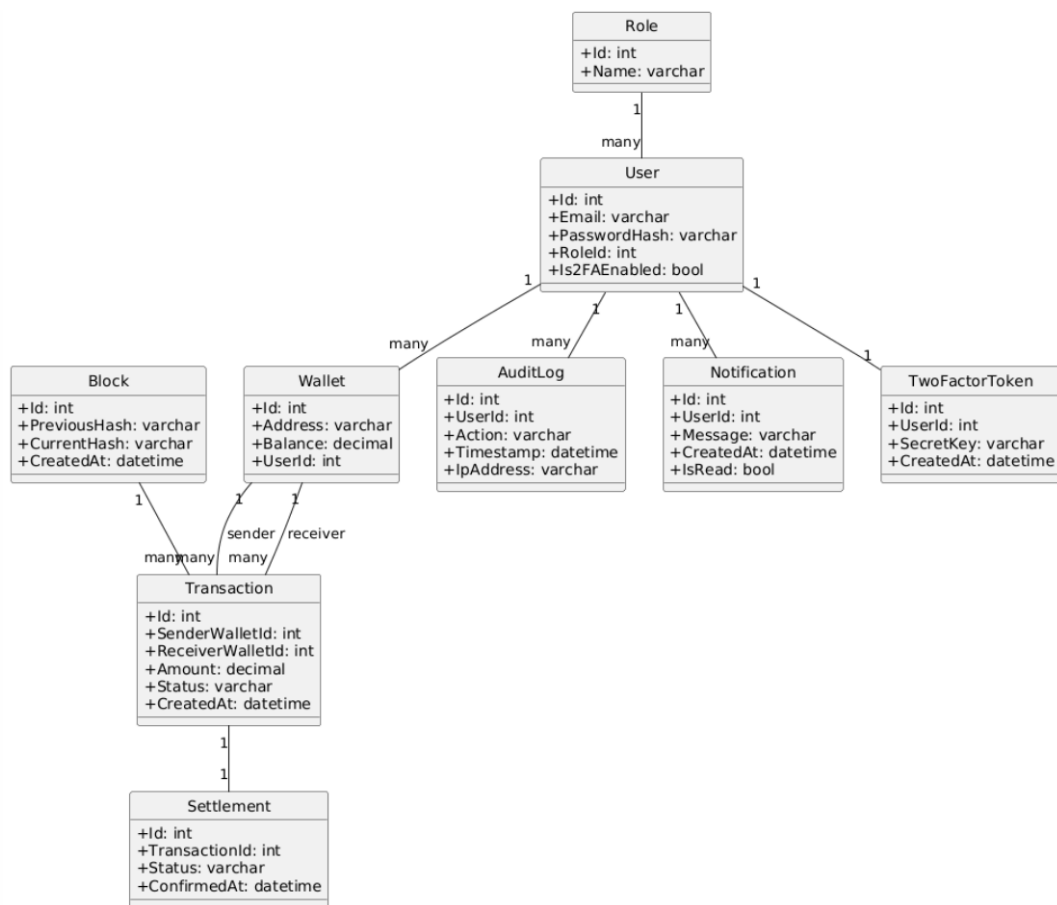


Рис. 2.5 ER-діаграма бази даних.

2.6 Опис таблиць бази даних

Таблиця 2.1

User - інформація про користувачів системи

Поле	Тип	Опис
Id	integer	Унікальний ідентифікатор
Email	varchar	Email користувача
PasswordHash	varchar	Хеш пароля
RoleId	integer	Роль користувача
Is2FAEnabled	boolean	Статус 2FA

Таблиця 2.2

Wallet - інформацію про криптовалютні гаманці

Поле	Тип	Опис
Id	integer	Ідентифікатор гаманця
Address	varchar	Blockchain-адреса
Balance	decimal	Баланс
UserId	integer	Власник гаманця

Таблиця 2.3

Transaction - інформацію про транзакції

Поле	Тип	Опис
Id	integer	Ідентифікатор транзакції
SenderWalletId	integer	Відправник
ReceiverWalletId	integer	Отримувач

Поле	Тип	Опис
Amount	decimal	Сума
Status	varchar	Статус транзакції
CreatedAt	datetime	Дата створення

Таблиця 2.4

AuditLog - призначена для збереження інформації про системні події та дії користувачів у системі

Поле	Тип	Опис
Id	int	Ідентифікатор запису
UserId	int	Ідентифікатор користувача
Action	varchar	Тип дії
EntityName	varchar	Назва сутності
OldValues	text	Попередні значення
NewValues	text	Нові значення
IpAddress	varchar	IP адреса
Timestamp	datetime	Дата та час події

2.7 Проєктування модулів блокчейну

Система містить наступні модулі:

- Transaction Engine;
- Settlement Engine;
- Clearing Queue;
- Audit System;
- Notification Service;
- Authentication Service;
- Blockchain Ledger.

Settlement Engine відповідає за фіналізацію транзакцій.

Clearing Queue забезпечує обробку черги фінансових операцій.

Audit System виконує журналювання всіх критичних дій.

На рисунку 2.7 зображено компонентну структуру платіжної системи на основі блокчейну та взаємодію між основними модулями системи.

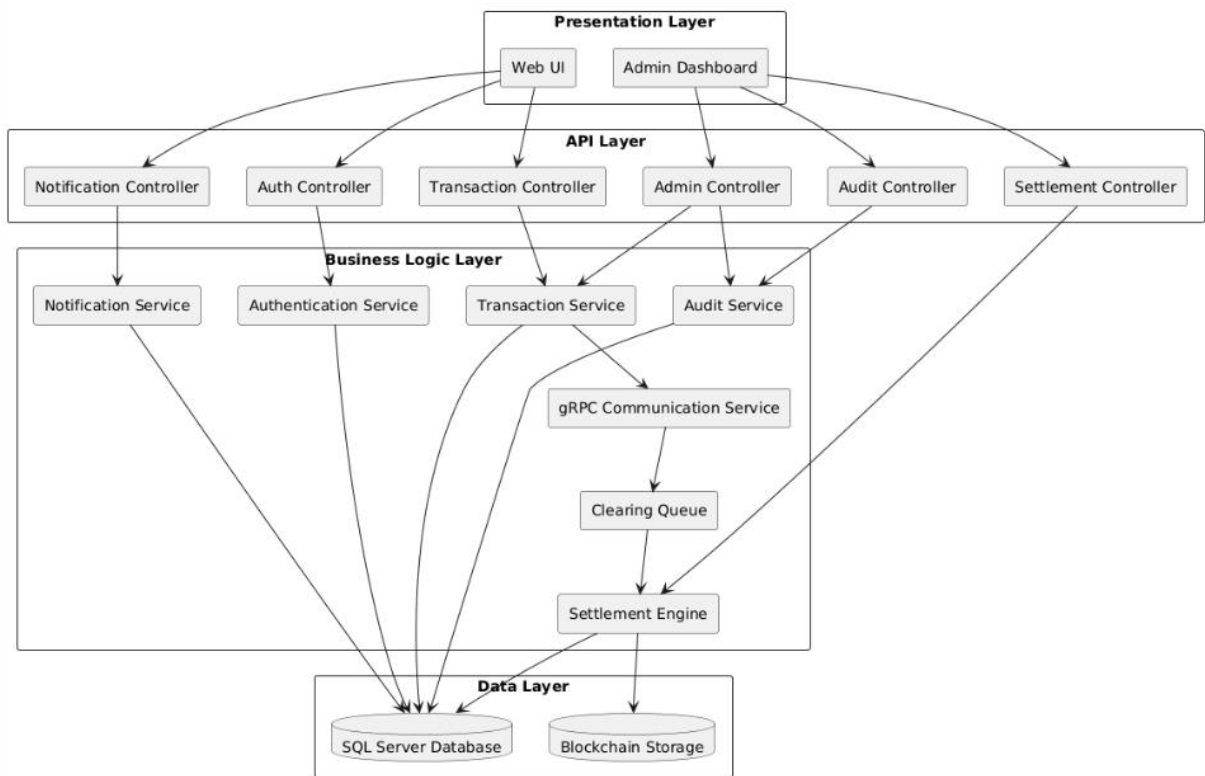


Рис. 2.7 Компонентна структура платіжної системи.

3 РОЗРОБКА ПРОГРАМНОЇ ЧАСТИНИ

3.1 Вибір середовища розробки

Для реалізації платіжної системи для транзакцій і розрахунків було використано сучасний стек технологій.

Основною мовою програмування було обрано C#, оскільки дана мова має:

- високу продуктивність;
- підтримку багатопоточності;
- сучасні механізми безпеки;
- інтеграцію з ASP.NET Core.

ASP.NET Core було використано для створення REST API та WEB-інтерфейсу.

Entity Framework Core використовувався як ORM-система для роботи з базою даних. Використання ORM дозволяє спростити взаємодію застосунку з базою даних [10].

ASP.NET Core забезпечує високу продуктивність та підтримку REST-архітектури [9].

SQLServer застосовувався як система управління базами даних.

Для реалізації blockchain-механізмів використовувалися:

- SHA256;
- JWT;
- BCrypt;
- SignalR;
- Audit.NET.

На рисунку 3.1 зображено структуру проєкту в середовищі Visual Studio.

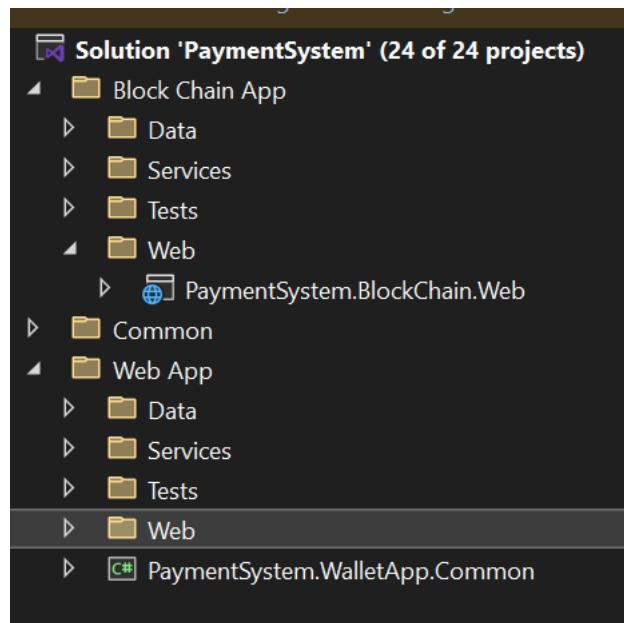


Рис. 3.1 Структура solution у Visual Studio.

3.2 Розробка архітектури WEB-системи

Система реалізована за принципами MVC та Clean Architecture. Використання шаблонів архітектурного проєктування підвищує масштабованість та підтримуваність системи [11]. REST

Основні компоненти:

- API Controllers;
- Services;
- Repositories;
- Database Layer;
- Authentication Layer;
- Blockchain Services.

На рисунку 3.2 представлено загальну архітектуру платіжної системи. Система побудована за багаторівневим принципом і включає клієнтський рівень, REST API, бізнес-логіку, blockchain-рівень та рівень доступу до даних. Такий підхід

дозволяє розділити відповідальність між компонентами, спростити підтримку системи та забезпечити її подальше масштабування.

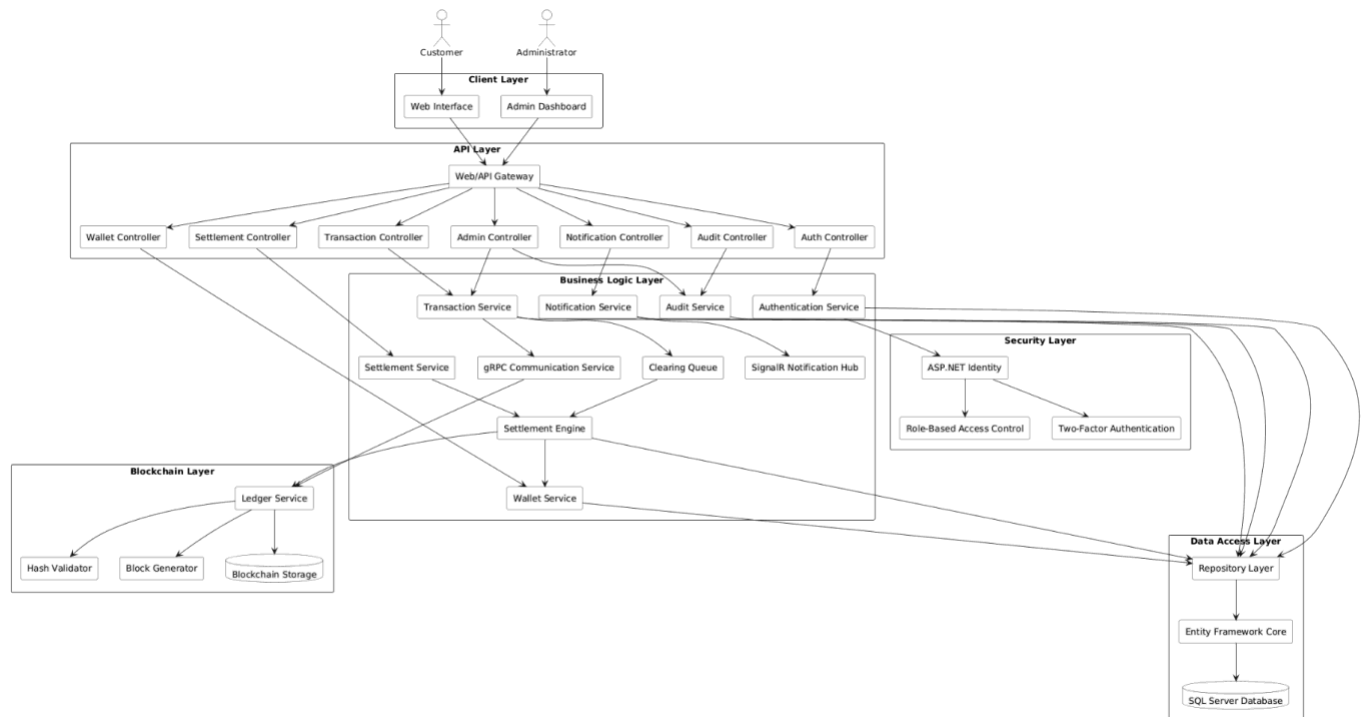


Рис. 3.2 Архітектура WEB- системи

3.3 Розробка REST API

REST API забезпечує взаємодію між frontend та backend частиною системи.

Основні API-модулі:

- Authentication API;
- Transaction API;
- Wallet API;
- Audit API;
- Notification API.

На рисунку 3.3 представлено компонентну архітектуру системи обробки транзакцій та розрахунків. REST-архітектура базується на принципах взаємодії через HTTP-протокол [18].

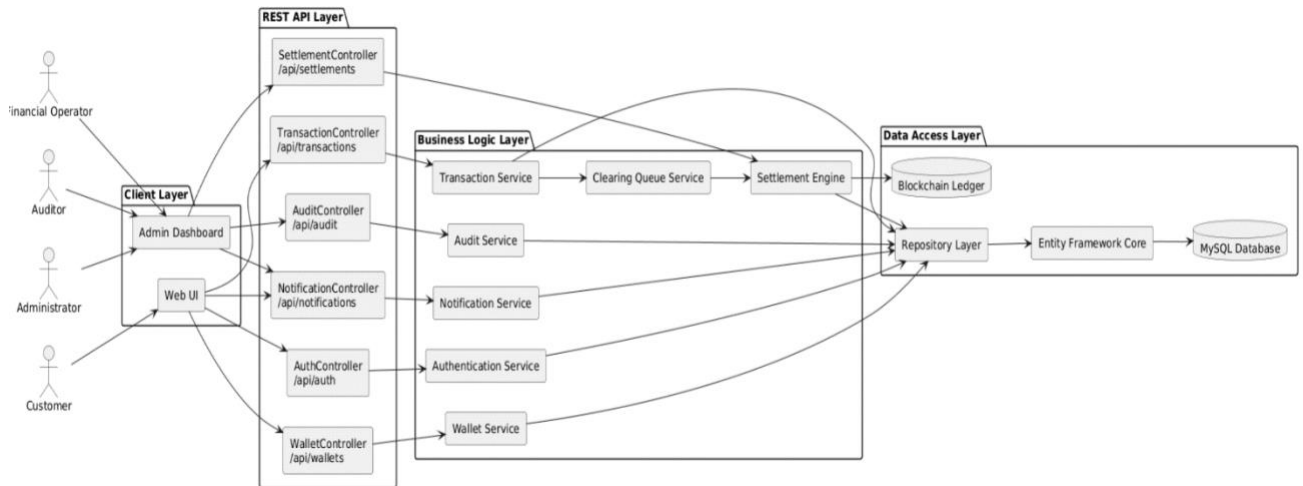


Рис.3.3 Архітектура REST API.

3.4 Реалізація механізмів блокчейну

У системі реалізовано механізми:

- генерації blockchain-блоків;
- hash-обчислень;
- валідації транзакцій;
- цифрових підписів.

Кожна транзакція проходить:

- Валідацію.
- Додавання у clearing queue.
- Settlement processing.
- Запис у blockchain ledger.

3.5 Реалізація Settlement Engine

Settlement Engine відповідає за фіналізацію транзакцій.

Основні функції:

- перевірка балансу;

- зміна статусів транзакцій;
- підтвердження платежів;
- оновлення blockchain ledger.

На рисунку 3.4 зображено архітектуру Settlement Engine, який відповідає за фіналізацію транзакцій та оновлення blockchain ledger.

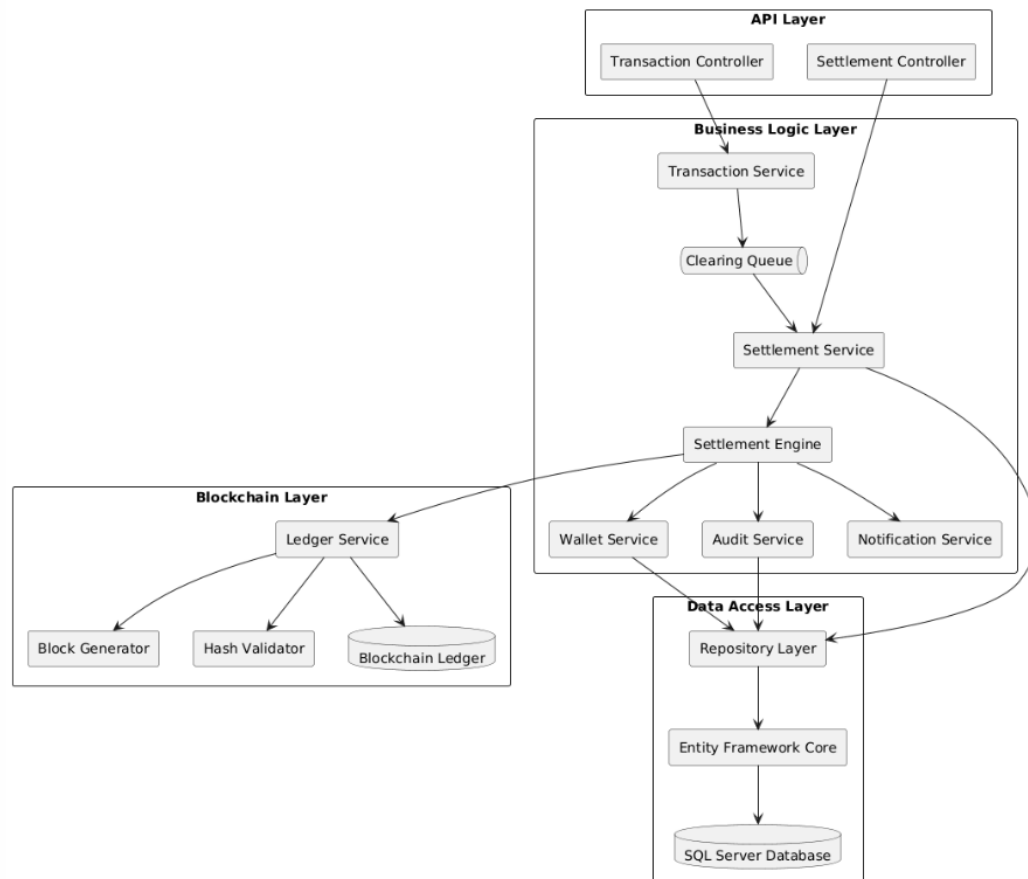


Рис. 3.4 Архітектура Settlement Engine.

3.6 Реалізація Clearing Queue

Clearing Queue використовується для обробки черги транзакцій у blockchain payment system та забезпечує асинхронне виконання settlement operations.

Основною задачею *Clearing Queue* є керування потоком фінансових транзакцій перед їх фіналізацією в *Settlement Engine*. Використання черги дозволяє

уникнути перевантаження системи та забезпечує стабільну обробку великої кількості транзакцій.

Clearing Queue дозволяє:

- балансувати навантаження;
- виконувати асинхронну обробку;
- уникати конфліктів транзакцій;
- контролювати порядок виконання settlement operations;
- повторно обробляти помилкові транзакції;
- забезпечувати стабільність transaction processing.

Після створення транзакції вона отримує статус Pending та додається у Clearing Queue. Далі Settlement Engine поетапно обробляє транзакції з черги, виконує перевірку даних, оновлення балансів та запис інформації у blockchain ledger.

На рисунку 3.5 представлено процес роботи Clearing Queue у blockchain payment system.

Під час обробки транзакцій система перевіряє:

- коректність даних транзакції;
- достатність балансу користувача;
- валідність blockchain hash;
- статус settlement operation.

У випадку виникнення помилки транзакція може бути повторно додана у чергу або переведена у статус Failed.

Для реалізації черги використовується background service, який періодично перевіряє наявність нових транзакцій та передає їх до Settlement Engine.

На рисунка 3.6 та 3.7 це проілюстровано на програмному рівні.

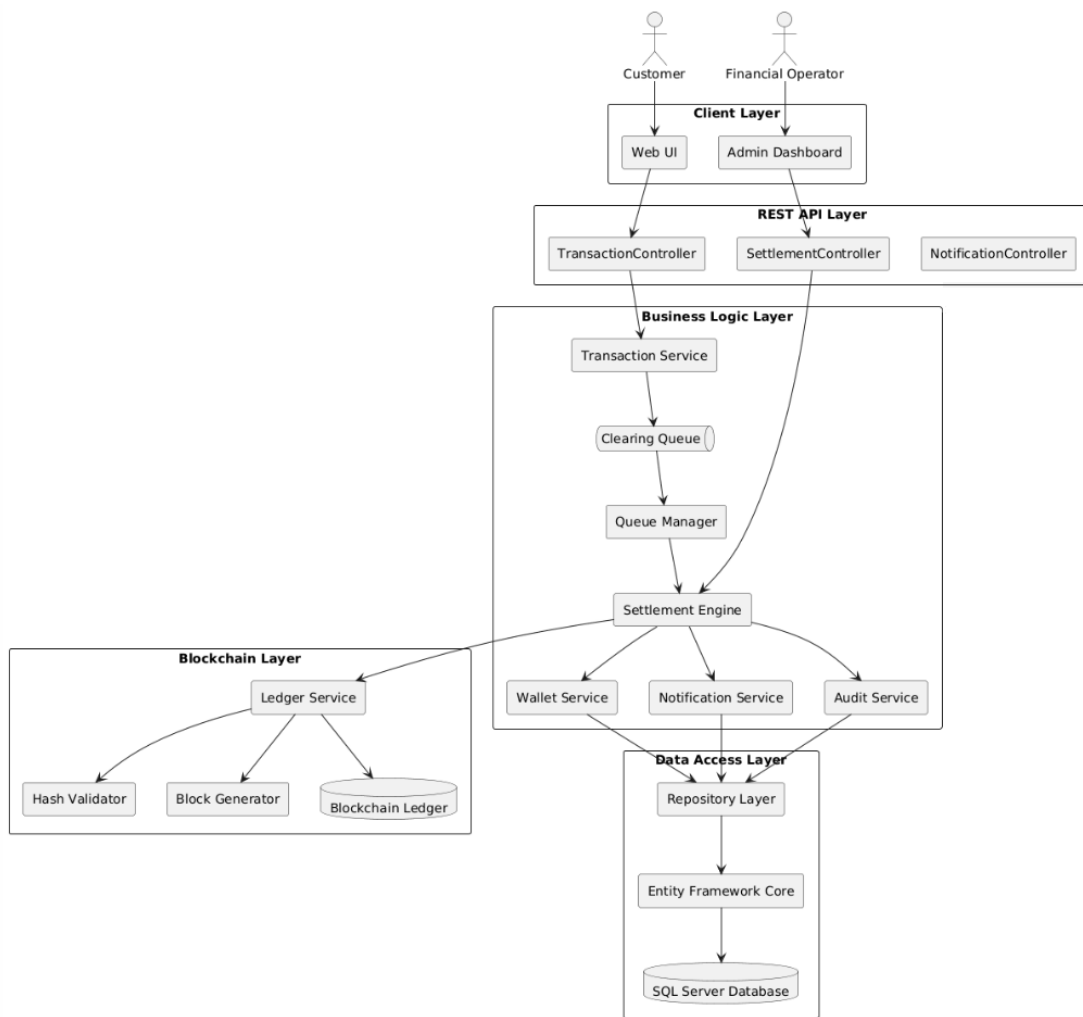


Рис. 3.5 – Робота Clearing Queue

```
await _clearingQueue.EnqueueAsync(transaction);
```

Рис. 3.6 Додавання транзакцій у чергу

```
while (await _clearingQueue.HasItemsAsync())
{
    var transaction = await _clearingQueue.DequeueAsync();
    await _settlementService.ProcessAsync(transaction);
}
```

Рис. 3.7 Обробка черги

На рисунку 3.8 наведено взаємодію Clearing Queue та Settlement Engine.

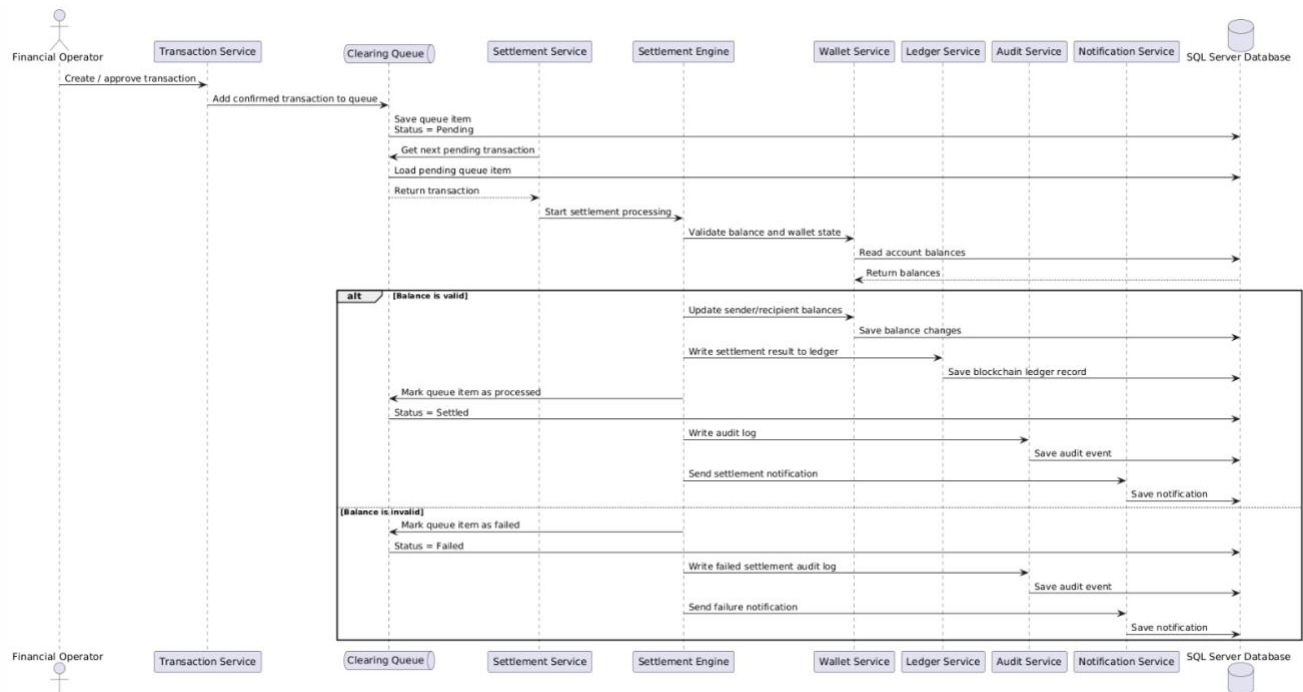


Рис. 3.8 – Взаємодія Clearing Queue та Settlement Engine

Реалізація Clearing Queue дозволила забезпечити стабільну асинхронну обробку фінансових транзакцій та підвищити продуктивність blockchain payment system при високому навантаженні.

3.7 Реалізація Audit System

Audit System забезпечує журналювання всіх системних подій системи та дозволяє контролювати виконання фінансових операцій, дії користувачів і адміністративні зміни в системі.

Основною задачею системи аудиту є:

- фіксація критичних подій;
- моніторинг безпеки;

- збереження історії транзакцій;
- контроль settlement operations;
- виявлення підозрілої активності.

У платіжній системі логуються:

- входи користувачів;
- фінансові операції;
- зміни статусів транзакцій;
- settlement events;
- адміністративні дії;
- помилки системи;
- спроби несанкціонованого доступу.

Для реалізації аудиту використовується бібліотека Audit.NET, яка дозволяє автоматизувати процес створення audit logs та збереження інформації про системні події у базі даних.

На рисунку 3.9 представлено архітектуру Audit System.

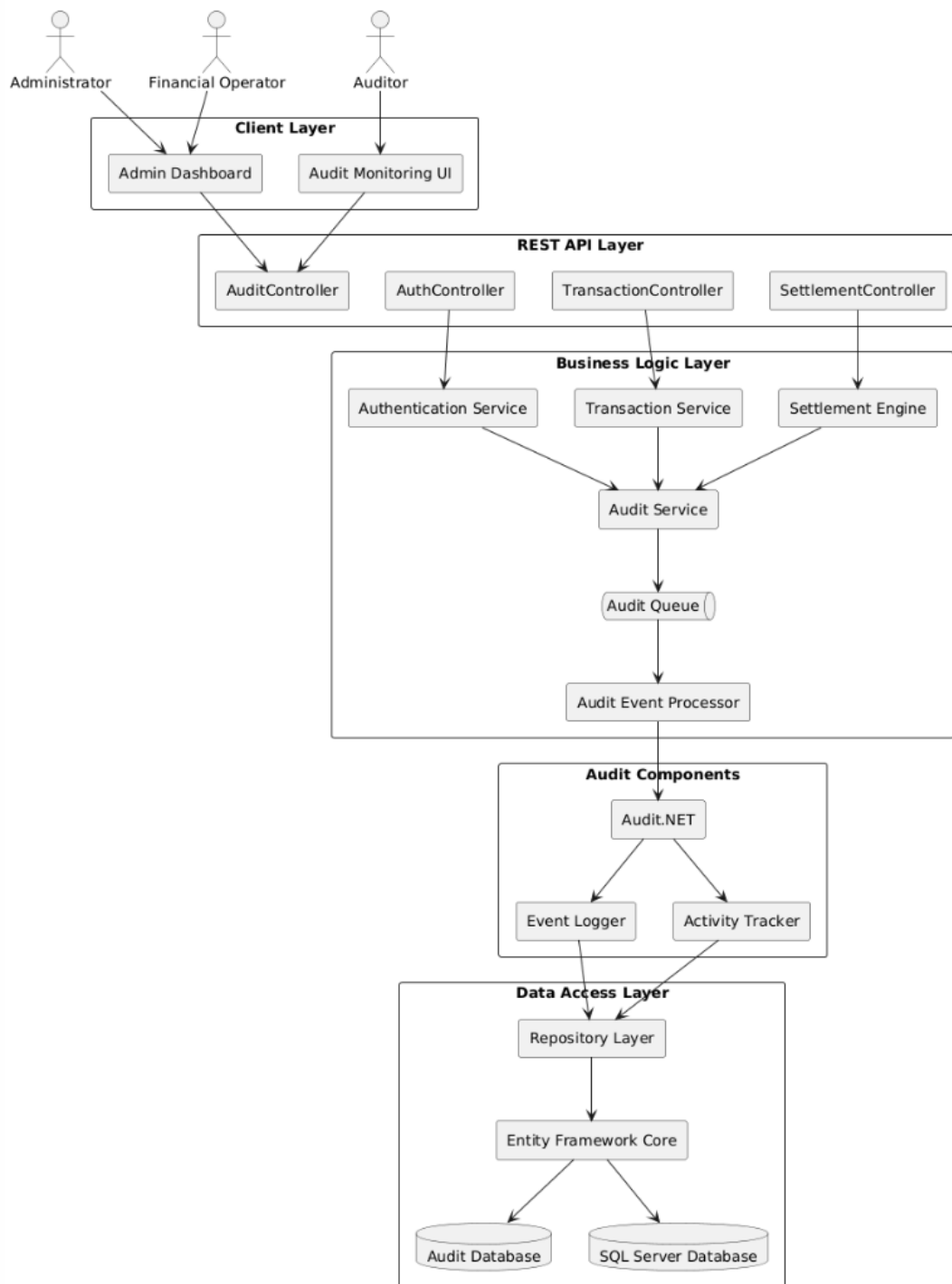


Рис 3.9 Архітектура Audit System

Для збереження інформації про системні події використовується таблиця AuditLog, яка містить інформацію про користувача, тип події, часову позначку та IP-адресу.

На рисунку 3.10 зображено код для конфігурації Audit.NET

```
Audit.Core.Configuration.Setup()
    .UseEntityFramework(ef => ef
        .AuditTypeMapper(t => typeof(AuditLog))
        .AuditEntityAction<AuditLog>((ev, entry, auditEntity) =>
        {
            auditEntity.Action = entry.Action;
            auditEntity.Timestamp = DateTime.UtcNow;
        }));
```

Рис. 3.10 конфігурація Audit.NET

3.8 Реалізація JWT та 2FA

Для забезпечення безпеки в платіжній системі реалізовано сучасні механізми автентифікації та захисту користувацьких акаунтів.

У системі використовується:

- JWT Authentication;
- BCrypt password hashing;
- Two-Factor Authentication;
- QR-code генерація;
- Role-Based Authorization.

JWT Authentication використовується для захисту REST API та авторизації користувачів після успішного входу в систему. Після проходження автентифікації користувач отримує JWT token, який використовується для подальших запитів до API. BCrypt забезпечує безпечне хешування паролів та захист від brute-force атак [20].

Для захисту паролів застосовується алгоритм BCrypt, який забезпечує безпечне хешування та унеможливорює збереження паролів у відкритому вигляді.

Для підвищення рівня безпеки реалізовано Two-Factor Authentication. Користувач може підключити Google Authenticator для підтвердження входу в систему за допомогою одноразових ОТР-кодів.

На рисунку 3.11 представлено процес генерації JWT-токену для двофакторної автентифікації.

```
var token = new JwtSecurityToken(  
    issuer: _configuration["Jwt:Issuer"],  
    audience: _configuration["Jwt:Audience"],  
    expires: DateTime.UtcNow.AddHours(2),  
    signingCredentials: credentials);
```

Рис. 3.11 генерація JWT-токен

```
var isValid = _twoFactorService.VerifyCode(user.SecretKey, code);
```

Рис. 3.12 перевірки ОТР-коду

Після генерації токен, відбувається перевірка ОТР- коду, зображено на рисунку 3.12.

Під час налаштування 2FA система генерує QR-код, який користувач сканує через Google Authenticator.

На рисунку 3.13 представлено процес авторизації користувача із використанням JWT та Two-Factor Authentication.

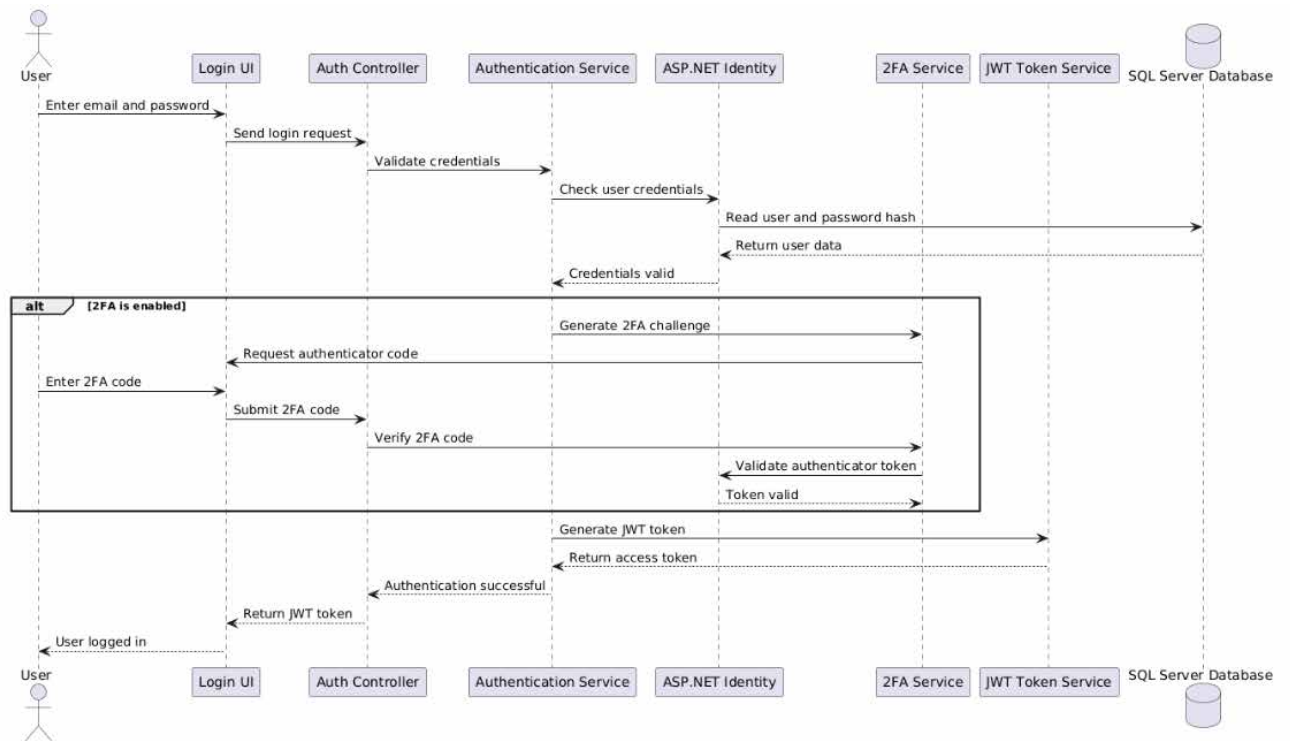


Рис. 3.13 – Авторизація користувача з JWT та 2FA

4 РЕКОМЕНДАЦІЇ ЩОДО ВПРОВАДЖЕННЯ ТА ЕКСПЛУАТАЦІЇ СИСТЕМИ

4.1 Тестування системи

Тестування blockchain payment system проводилось з метою перевірки стабільності, безпеки та коректності функціонування програмного забезпечення.

Під час тестування перевірялися:

- працездатність REST API;
- коректність blockchain transaction processing;
- робота Settlement Engine;
- функціонування Clearing Queue;
- робота JWT authentication;
- коректність Two-Factor Authentication;
- стабільність роботи бази даних;
- журналювання подій через Audit.NET;
- захист від несанкціонованого доступу.

Для тестування використовувалися:

- Swagger UI;
- Postman;
- MySQL Workbench;
- Visual Studio;
- браузери Google Chrome та Microsoft Edge.

Було проведено модульне та інтеграційне тестування системи.

Модульне тестування дозволило перевірити окремі компоненти системи:

- authentication service;
- transaction service;

- settlement service;
- audit service;
- notification service.

На рисунку 4.1 зображено форму автентифікації.

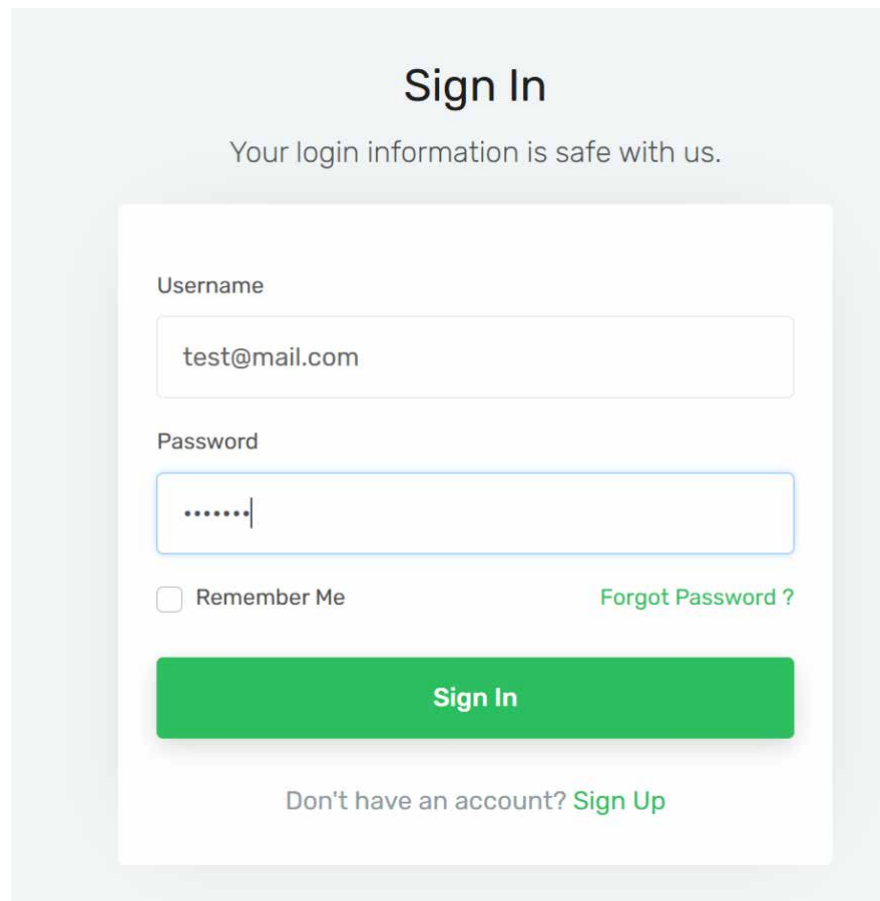
The image shows a 'Sign In' form on a light gray background. At the top, the text 'Sign In' is centered in a large, bold, black font. Below it, a smaller line of text reads 'Your login information is safe with us.' The form itself is a white rounded rectangle. It contains two input fields: 'Username' with the text 'test@mail.com' and 'Password' with masked characters '.....'. Below the password field is a checkbox labeled 'Remember Me' and a green link 'Forgot Password ?'. A large green button with the text 'Sign In' is positioned below these elements. At the bottom of the form, there is a link 'Don't have an account? Sign Up'.

Рис. 4.1 Форма авторизації.

На рисунку 4.2 представлено головну сторінку платіжної системи для розрахунків і транзакцій на основі блокчейну [20]. Головна сторінка містить інформацію про основні можливості системи, включаючи blockchain transaction processing, secure payment operations та settlement services. Користувач може виконати авторизацію або реєстрацію нового акаунта, а також переглянути інформацію про переваги використання blockchain-технологій у фінансовій сфері.

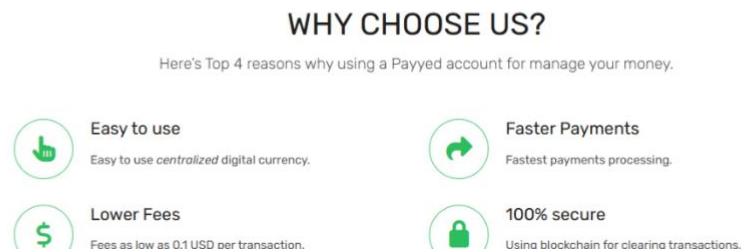
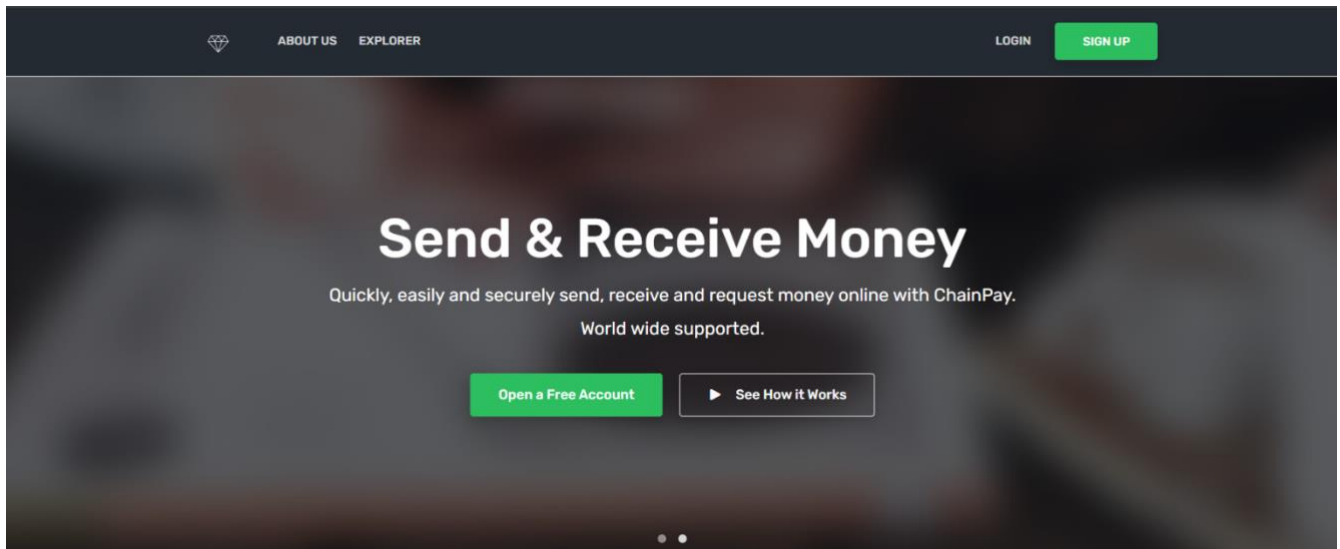


Рис. 4.2 Головна сторінка платіжної системи

Після успішної авторизації користувач отримує доступ до dashboard, який дозволяє контролювати баланс гаманця, переглядати історію фінансових операцій та відстежувати статуси settlement operations. Dashboard містить інформацію про recent activity, deposits, withdrawals та blockchain transactions. Це зображено на рисунку 4.3

Інтеграційне тестування виконувалося для перевірки взаємодії між компонентами системи.

Було перевірено:

- взаємодію API та бази даних;
- роботу blockchain ledger;
- обробку settlement operations;
- роботу clearing queue;

● систему audit logging.

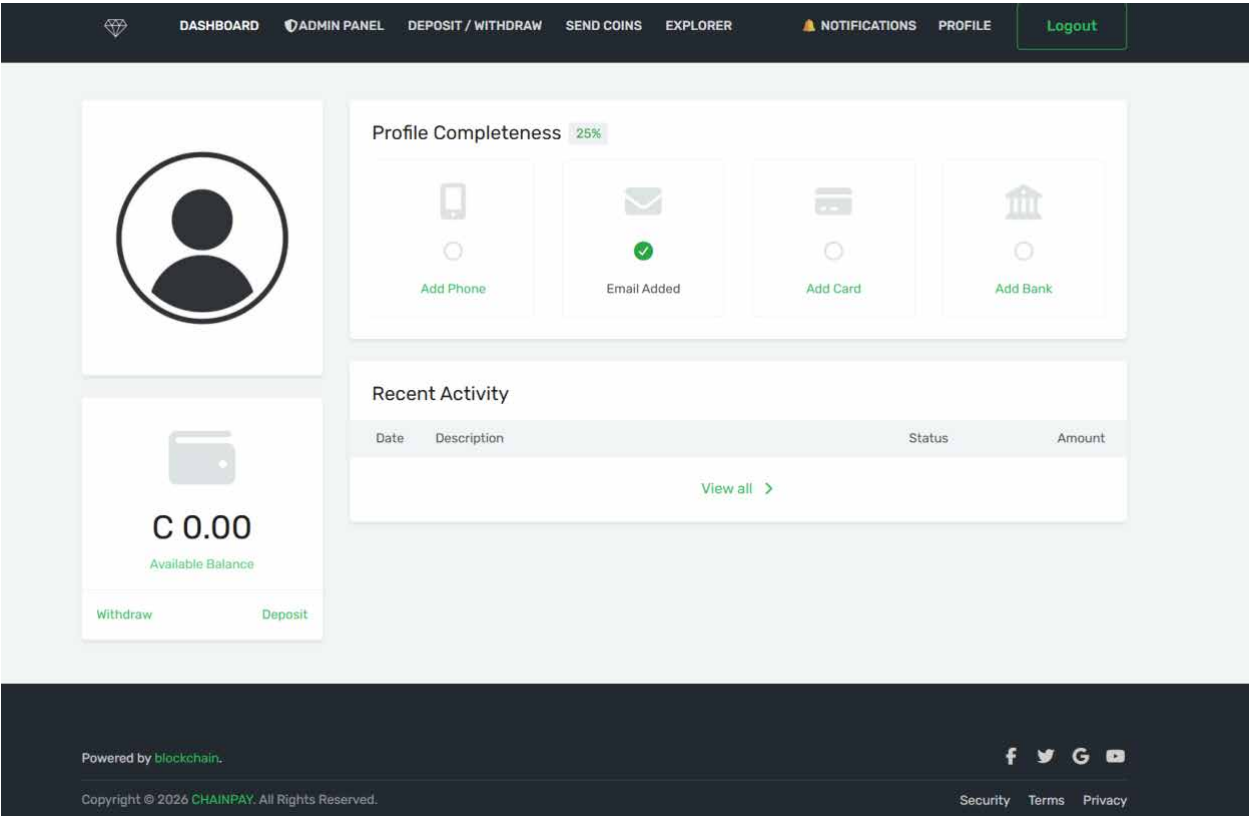


Рис. 4.3 Інтерфейс користувача.

На рисунка 4.4 і 4.5 зображено, як для демонстрації роботи блокчейн транзакцій користувач може переглядати інформацію про транзакції, включаючи sender address, receiver address, transaction amount та transaction status. Система підтримує відображення успішних, pending та failed transactions.

Transactions						
Hash	Sender	Recipient	Amount	Fee	Status	Created
799F8CBB94...	4A01FC19F6...	34BB5DFB9B...	7777	0	Pending	23/05/2026 18:32

Рис. 4.4 Тестування transaction processing.

Hash	Sender	Recipient	Amount	Status	Created	Confirmed	Settled	Failure
102E2AFD92...	3A4F2A0E34...	B79B9C9C18...	600	Settled	23/05/2026 19:42	-	23/05/2026 19:42	-
635327B9A3...	4A01FC19F6...	3A4F2A0E34...	325	Settled	23/05/2026 19:41	-	23/05/2026 19:41	-
6FD76ABD77...	4A01FC19F6...	3A4F2A0E34...	325	Settled	23/05/2026 19:41	-	23/05/2026 19:41	-
C4DABE364A...	3A4F2A0E34...	B79B9C9C18...	200	Pending	23/05/2026 19:15	-	-	-
0DCFD4B6D9...	4A01FC19F6...	3A4F2A0E34...	235	Settled	23/05/2026 19:14	-	23/05/2026 19:14	-
338B65F214...	34BB5DFB9B...	B79B9C9C18...	497	Failed	23/05/2026 19:07	-	-	-
9DCDCFA83F...	34BB5DFB9B...	B79B9C9C18...	4000	Failed	23/05/2026 19:03	-	-	-
D422B3622D...	34BB5DFB9B...	B79B9C9C18...	4000	Failed	23/05/2026 19:02	-	-	-
968378F35B...	34BB5DFB9B...	34BB5DFB9B...	4000	Failed	23/05/2026 18:58	-	-	-
03F722B8E6...	4A01FC19F6...	34BB5DFB9B...	4353	Settled	23/05/2026 18:58	-	23/05/2026 18:58	-
B86BEF8A65...	4A01FC19F6...	34BB5DFB9B...	4636	Pending	23/05/2026 18:45	-	-	-
799F8CB894...	4A01FC19F6...	34BB5DFB9B...	7777	Pending	23/05/2026 18:32	-	-	-

Рис. 4.5 Перевірка settlement operations.

Для перевірки безпеки системи було проведено:

- перевірку JWT token validation;
- тестування password hashing;
- перевірку Two-Factor Authentication;
- перевірку role-based authorization;
- тестування HTTPS-з'єднання.

Two-Factor Authentication (2FA) забезпечує додатковий рівень захисту користувацьких акаунтів шляхом підтвердження входу одноразовим ОТР-кодом.

Після введення логіна та пароля користувач повинен підтвердити автентифікацію за допомогою коду, згенерованого у Microsoft Authenticator. Зображено на рис. 4.6. Це дозволяє значно знизити ризик несанкціонованого доступу навіть у випадку компрометації пароля.

Two-factor authentication

Your login is protected with an authenticator app. Enter your authenticator code below.

Authenticator code

☐ Remember this machine

Log in

Don't have access to your authenticator device? You can [log in with a recovery code](#).

Рис. 4.6 Перевірка Two-Factor Authentication.

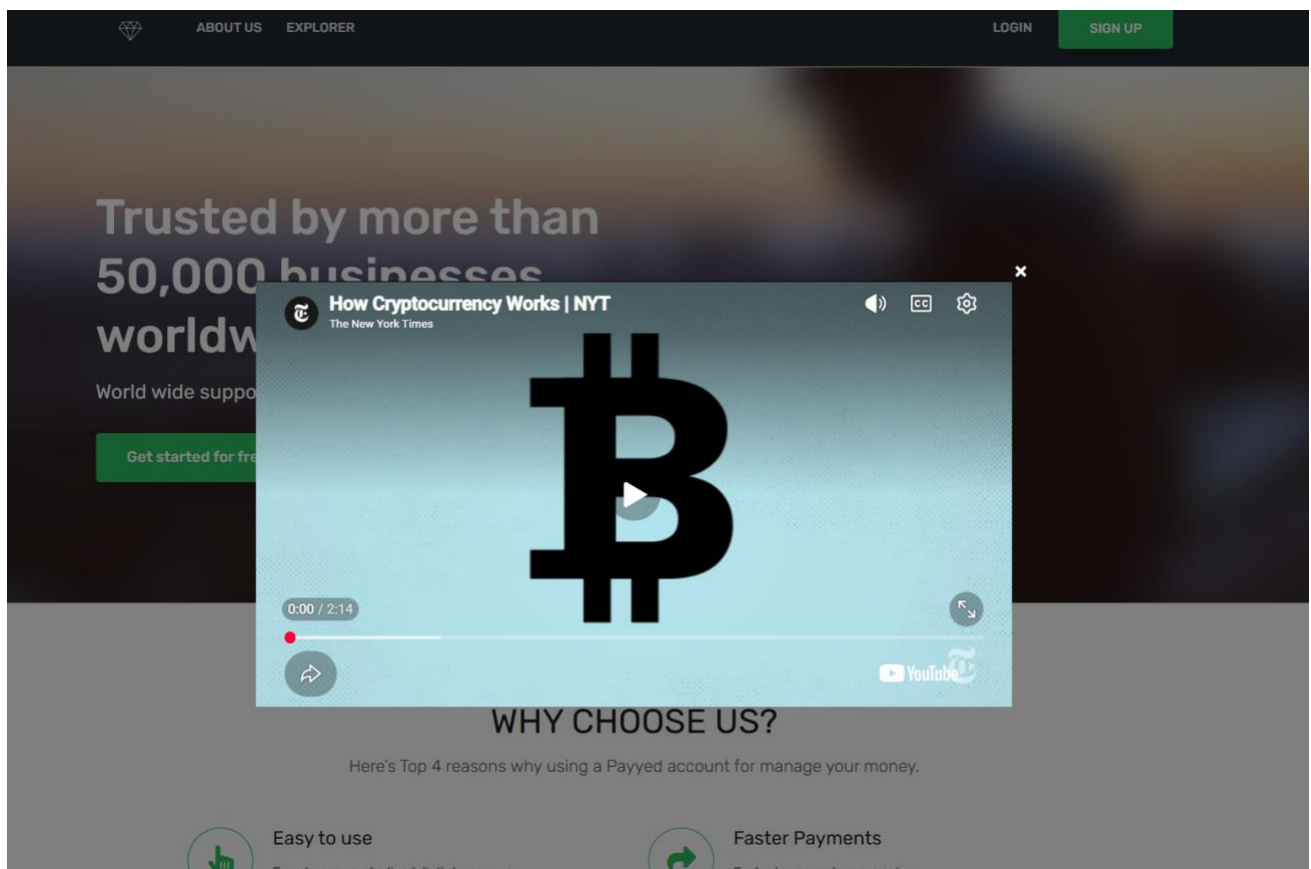


Рис. 4.7 Відео з поясненням принципів функціонування цифрових валют

Для ознайомлення користувачів із принципами роботи платіжної системи реалізовано інтеграцію інформаційного відео про криптовалюти та принципи роботи блокчейну. Це продемонстровано на рисунку 4.7. Відео відкривається у модальному вікні та містить пояснення принципів функціонування цифрових валют і blockchain transaction processing.

На рис. 4.8 можна побачити, як під час перегляду інформації про фінансові операції користувач може отримати деталізовану інформацію про blockchain transactions. Відображаються transaction amount, transaction status, blockchain addresses та settlement information, що забезпечує прозорість роботи системи.

Audit Logs					
User	Action	Entity	Details	Date	
3A4F2A0E34D3DF43852635493B90FC67816508D739B224345881600834553DD	Create Transaction	Transaction	Transaction 102E2AFD92071DB466751B10589D7E884B5ED5735B48F7FEA8EA36B250D0BB43 created with amount 600	23/05/2026 19:42	
4A01FC19F646E2386F8D109AB0CE2BD31741D1BCFB376B94C81123629D44AEB7	Create Transaction	Transaction	Transaction 635327B9A3BE973711576FD8572906EFF9ADF28F17BD0FE446A5ADC9CAE665CF created with amount 325	23/05/2026 19:41	
4A01FC19F646E2386F8D109AB0CE2BD31741D1BCFB376B94C81123629D44AEB7	Create Transaction	Transaction	Transaction 6FD76ABD777ED4FE88E6B19F5844F431E399E0CFF1F0D5D1642BCFA50867F7A1 created with amount 325	23/05/2026 19:41	
3A4F2A0E34D3DF43852635493B90FC67816508D739B224345881600834553DD	Create Transaction	Transaction	Transaction C4DABE364AC55BDD8A7F19D53D8CB015FF82D41032377D9906DA916B956A6D48 created with amount 200	23/05/2026 19:15	
4A01FC19F646E2386F8D109AB0CE2BD31741D1BCFB376B94C81123629D44AEB7	Create Transaction	Transaction	Transaction 0DCFD4B6D9AE8E9A9290BA88B503064CA44DAC2FF99A319DB408F590A24BF6B7 created with amount 235	23/05/2026 19:14	
34BB5DFB9B90AF8B22E703D90CBE9F2081D9242C0D91E22FA649CD465B8F73D6	Create Transaction	Transaction	Transaction 33B865F214343080231D6996DB48DC270405EB573A276021A212DCA378F361F0 created with amount 497	23/05/2026 19:07	
34BB5DFB9B90AF8B22E703D90CBE9F2081D9242C0D91E22FA649CD465B8F73D6	Create Transaction	Transaction	Transaction 9DCDCFA83F8A6B91E5567D3C89499DF1A94D406A8B314B530E0C88D38F315DE0 created with amount 4000	23/05/2026 19:03	
34BB5DFB9B90AF8B22E703D90CBE9F2081D9242C0D91E22FA649CD465B8F73D6	Create Transaction	Transaction	Transaction D42283622D27B95CE9276D2EFD0D09FE2F7B690C9ACBF6C736526D230E354F54 created with amount 4000	23/05/2026 19:02	

Рис. 4.8 Тестування журналу аудиту.

У результаті тестування було встановлено, що платіжна система стабільно працює та забезпечує коректне виконання транзакцій і settlement-операцій.

4.2 Вимоги до апаратного та програмного забезпечення

Для стабільної роботи платіжної системи на основі блокчейну необхідне відповідне апаратне та програмне забезпечення.

Таблиця 4.1

Мінімальні апаратні вимоги

Компонент	Мінімальні вимоги
Процесор	Intel Core i3 або аналог
Оперативна пам'ять	8 GB RAM
Накопичувач	SSD 20 GB
Мережа	Підключення до Інтернету

Таблиця 4.2

Рекомендовані апаратні вимоги

Компонент	Рекомендовані вимоги
Процесор	Intel Core i5/i7
Оперативна пам'ять	16 GB RAM
Накопичувач	SSD 50 GB
Мережа	Gigabit Ethernet

Програмні вимоги

Для роботи системи необхідно:

- Windows Server 2019 / Windows 11;
- .NET 8 SDK;
- ASP.NET Core Runtime;
- MySQL Server 8.0;
- Docker Desktop;
- IIS або Nginx;
- Visual Studio 2022.

Для cloud deployment можуть використовуватись:

- AWS EC2;
- AWS RDS;
- Docker containers;
- Ubuntu Server.

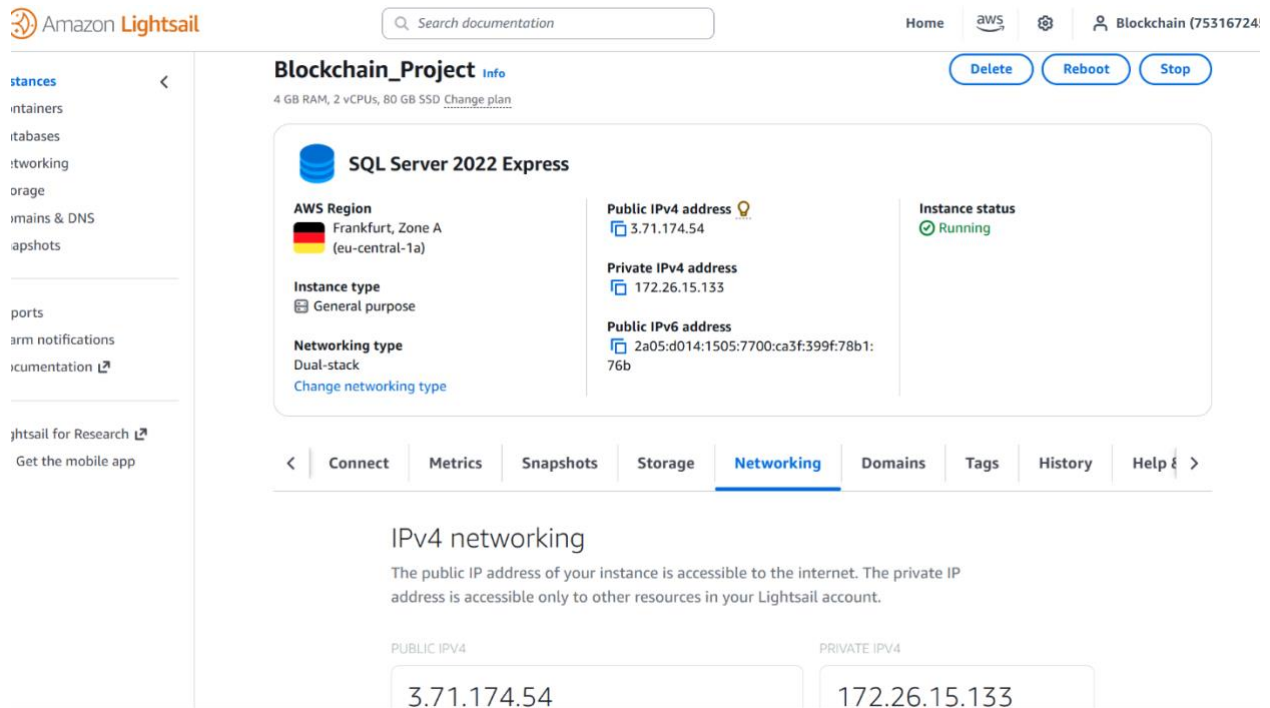


Рис. 4.9 Розгортання платіжної системи на AWS ES2.

Для забезпечення стабільної роботи blockchain payment system та можливості cloud deployment було використано хмарну інфраструктуру AWS. Розгортання системи виконувалось у середовищі Amazon Lightsail із використанням SQL Server 2022 Express та окремого віртуального серверного інстансу. На рисунку 4.7 представлено процес розгортання blockchain payment system у середовищі AWS.

Для зберігання інформації про користувачів, транзакції, blockchain blocks, settlement operations та audit logs використовується SQL Server. База даних містить таблиці для реалізації blockchain transaction processing, фінансових операцій та системи журналювання подій. На рисунку 4.8 представлено конфігурацію SQL Server та структуру таблиць blockchain payment system.

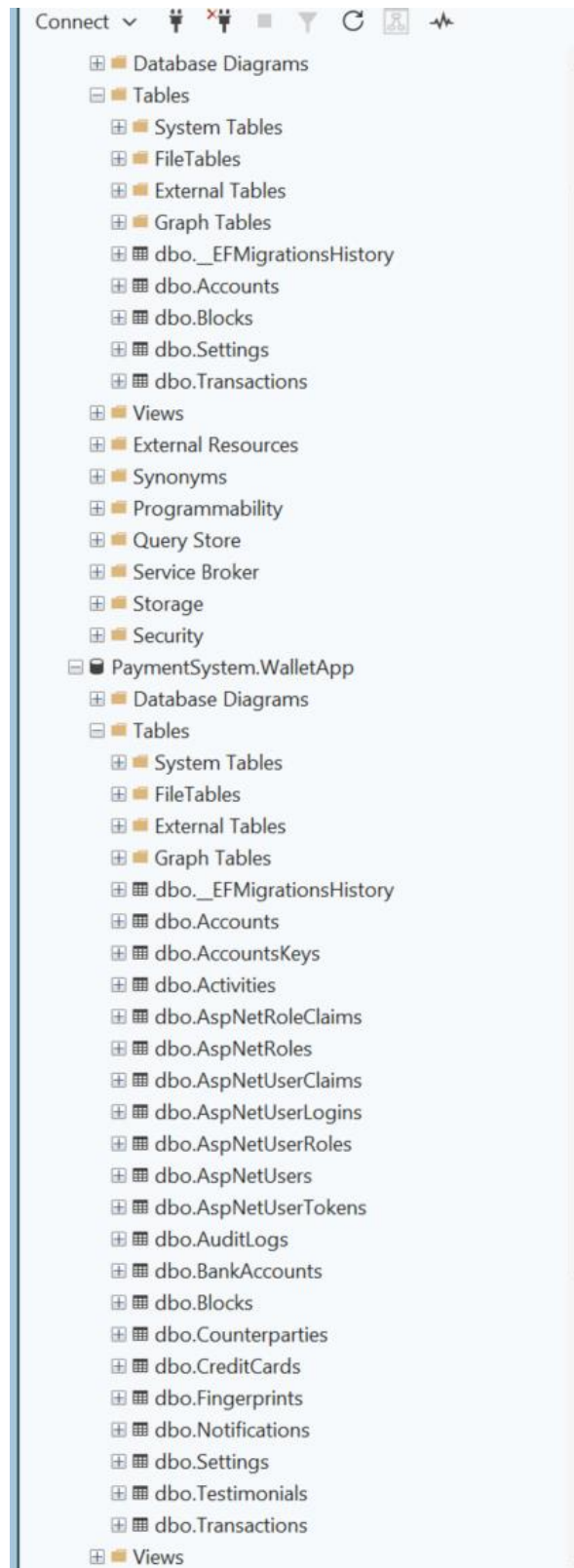


Рис. 4.10 Конфігурація SQL Server.

Вимоги до безпеки

Для безпечної експлуатації системи необхідно:

- використовувати HTTPS;
- застосовувати JWT authentication;
- активувати Two-Factor Authentication;
- регулярно оновлювати серверне ПЗ;
- виконувати резервне копіювання бази даних;
- контролювати audit logs.

4.3 Склад інсталяційного пакету

Інсталяційний пакет blockchain payment system містить усі необхідні компоненти для встановлення та запуску системи.

До складу інсталяційного пакету входять:

- backend application;
- frontend WEB application;
- database scripts;
- Docker configuration;
- API configuration files;
- SSL certificates;
- installation manual.

Структура інсталяційного пакету

- BlockchainPaymentSystem;
- Backend;
- Frontend;
- Database;
- Docker;
- Scripts;

- Logs;
- appsettings.json;
- README.md.

Встановлення blockchain payment system включає:

1. Встановлення .NET Runtime.
2. Встановлення MySQL Server.
3. Імпорт SQL-скриптів.
4. Налаштування appsettings.json.
5. Налаштування JWT secrets.
6. Запуск Docker containers.
7. Налаштування IIS або Nginx.
8. Запуск WEB-застосунку.

На рисунку 4.11 добре платіжної системи. Встановлення системи включає налаштування середовища виконання, бази даних SQL Server, конфігурацію REST API, запуск backend та frontend компонентів, а також перевірку працездатності системи.

Рекомендації щодо експлуатації

Для стабільної роботи системи рекомендується:

- регулярно виконувати резервне копіювання;
- контролювати навантаження серверів;
- перевіряти audit logs;
- оновлювати SSL certificates;
- використовувати моніторинг API та бази даних;
- виконувати тестування після оновлення системи.

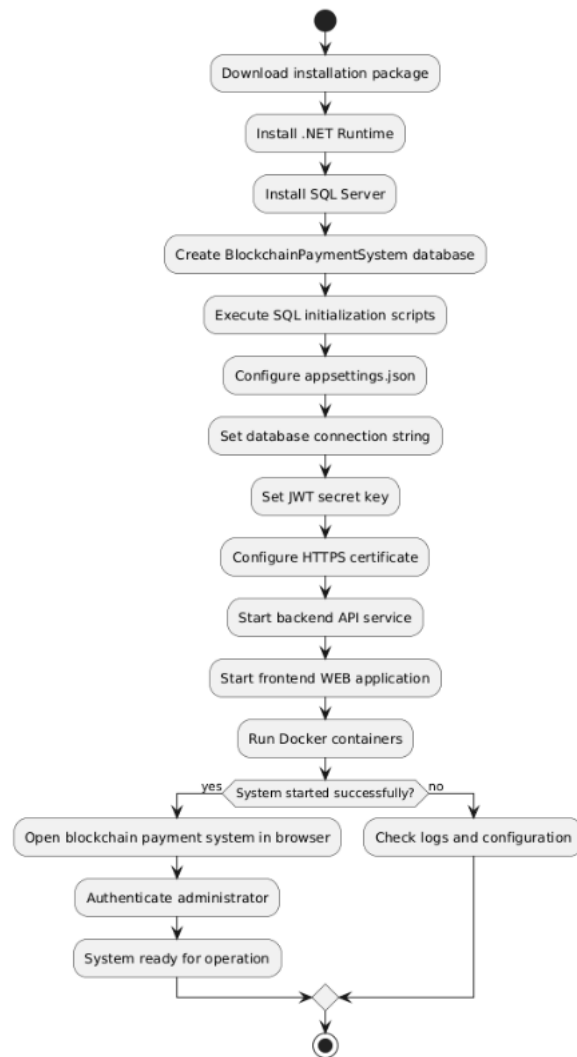


Рис. 4.11 Процес встановлення платіжної системи.

Для адміністрування blockchain payment system реалізовано administrative dashboard, який забезпечує моніторинг роботи системи та контроль фінансових операцій у режимі реального часу. На рисунку 4.12 представлено панель моніторингу платіжної системи.

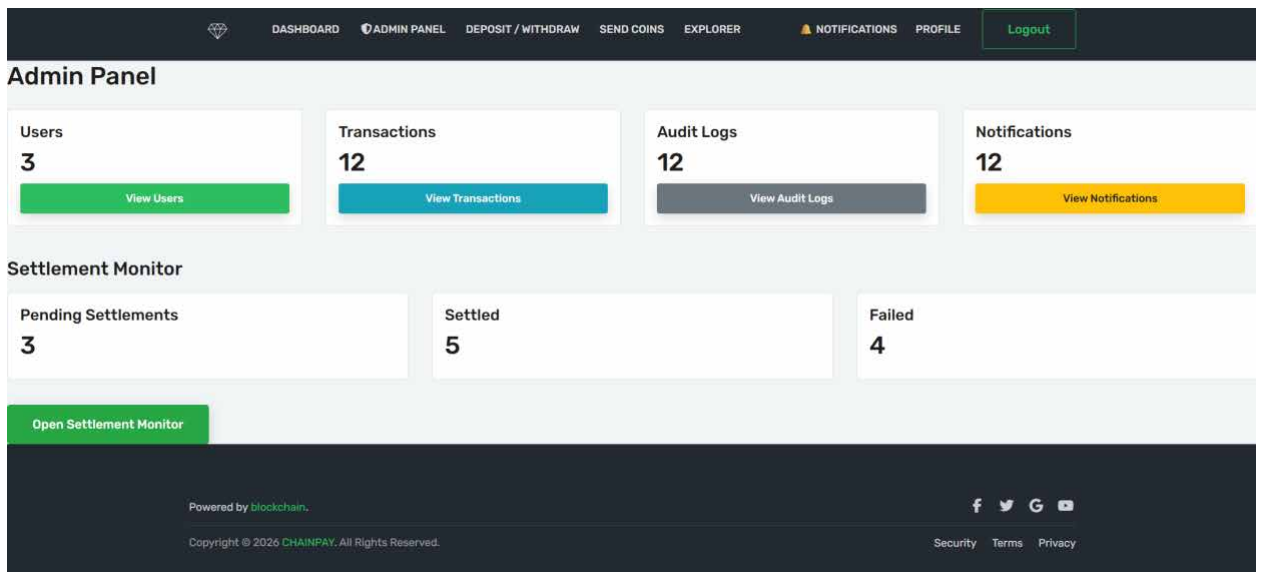


Рис. 4.12 Моніторинг роботи системи для розрахунків і транзакцій.

Під час тестування системи, визначено апаратні та програмні вимоги, описано склад інсталяційного пакету та рекомендації щодо розгортання системи.

Результати дослідження підтвердили можливість стабільного функціонування blockchain payment system в локальному та cloud-середовищі.

ВИСНОВКИ

У результаті виконання бакалаврської кваліфікаційної роботи на тему «Blockchain Payment System for Transaction and Settlement» було здійснено повний цикл розробки сучасної інформаційної системи для виконання цифрових фінансових операцій із використанням blockchain-технологій. Під час виконання роботи було проведено аналіз предметної області, проектування інформаційного забезпечення, розробку програмної частини системи, а також сформовано рекомендації щодо впровадження та експлуатації програмного продукту.

Проведений аналіз сучасних платіжних систем та blockchain-технологій дозволив визначити основні проблеми централізованих фінансових сервісів, серед яких залежність від центральних серверів, ризики несанкціонованого доступу до інформації, недостатній рівень прозорості транзакцій та складність масштабування систем. У результаті аналізу були визначені ключові напрями автоматизації transaction processing та settlement operations, сформульовано функціональні та нефункціональні вимоги до системи, а також побудовано UML-діаграми, що описують взаємодію користувачів і компонентів blockchain payment system.

У межах проектування інформаційного забезпечення було створено логічну та фізичну моделі бази даних, побудовано ER-діаграму та визначено структуру основних таблиць системи. Для зберігання інформації про користувачів, транзакції, blockchain blocks, settlement operations, audit logs та notifications використано систему управління базами даних MySQL. Також були реалізовані SQL-скрипти для автоматичної ініціалізації структури бази даних та створення необхідних зв'язків між сутностями системи.

Програмна частина blockchain payment system була реалізована мовою програмування C# у середовищі Visual Studio 2022 із використанням платформи ASP.NET Core. Архітектура системи побудована відповідно до принципів MVC та

Clean Architecture, що забезпечило розділення бізнес-логіки, API та рівня доступу до даних. Для взаємодії між frontend та backend було реалізовано REST API, яке забезпечує роботу з транзакціями, гаманцями користувачів, settlement operations, audit logs та системою сповіщень.

У процесі розробки було реалізовано основні blockchain-механізми, які забезпечують перевірку транзакцій, генерацію hash-значень, blockchain validation та ведення distributed ledger. Для забезпечення автоматизованої обробки фінансових операцій було створено Settlement Engine, який виконує фіналізацію транзакцій, оновлення балансів користувачів та зміну статусів transaction processing. Крім цього, реалізовано Clearing Queue для асинхронної обробки черги транзакцій та балансування навантаження системи.

Важливою складовою розробленої системи є реалізація механізмів безпеки. У blockchain payment system було впроваджено JWT authentication, Two-Factor Authentication, BCrypt password hashing та Role-Based Authorization. Для реалізації двофакторної автентифікації використовувалися QR-коди та інтеграція з Microsoft Authenticator, що дозволило значно підвищити рівень захисту користувацьких акаунтів і фінансових операцій.

Для журналювання системних подій та моніторингу фінансових операцій використано бібліотеку Audit.NET. Реалізована система аудиту дозволяє фіксувати всі критичні дії користувачів, зміни статусів транзакцій, settlement events та security events. Це забезпечує прозорість роботи системи та спрощує контроль за виконанням фінансових операцій.

Також у роботі було створено адміністративну панель та dashboard користувача, які забезпечують зручне управління системою, перегляд transaction history, audit logs, notifications та settlement operations. Для реалізації системи сповіщень використовувалися notification services та SignalR, що дозволяє надсилати користувачам повідомлення в режимі реального часу.

У четвертому розділі було проведено тестування blockchain payment system. Виконано модульне, інтеграційне та безпекове тестування системи. Перевірено коректність роботи REST API, Settlement Engine, Clearing Queue, JWT authentication та blockchain transaction processing. Результати тестування підтвердили стабільність роботи системи, коректність виконання транзакцій та ефективність реалізованих механізмів безпеки.

Для впровадження та експлуатації системи були визначені вимоги до апаратного та програмного забезпечення, а також сформовано структуру інсталяційного пакету. Для cloud deployment використовувалися AWS EC2, AWS RDS, Docker та Nginx. Це дозволяє масштабувати blockchain payment system та забезпечувати її стабільну роботу у production-середовищі.

Результати розробки blockchain payment system були представлені у вигляді тез доповіді на студентській науково-практичній конференції з інформаційних технологій та програмної інженерії. Тези доповіді на тему «Blockchain Payment System for Transaction and Settlement» наведені у додатках до роботи.

У результаті виконання бакалаврської кваліфікаційної роботи всі поставлені цілі та задачі були повністю досягнуті. Розроблена blockchain payment system забезпечує:

- безпечне виконання цифрових фінансових транзакцій;
- автоматизацію settlement-процесів;
- blockchain transaction processing;
- асинхронну обробку transaction queue;
- аудит фінансових операцій;
- двофакторну автентифікацію користувачів;
- централізований контроль статусів транзакцій;
- масштабування системи у cloud-середовищі;
- високий рівень безпеки та надійності.

Створене програмне забезпечення є функціональним, практичним та готовим до використання у сфері fintech та blockchain payment solutions. Отримані результати підтверджують ефективність використання blockchain-технологій для побудови сучасних платіжних систем із високим рівнем захисту, прозорості та автоматизації фінансових процесів.

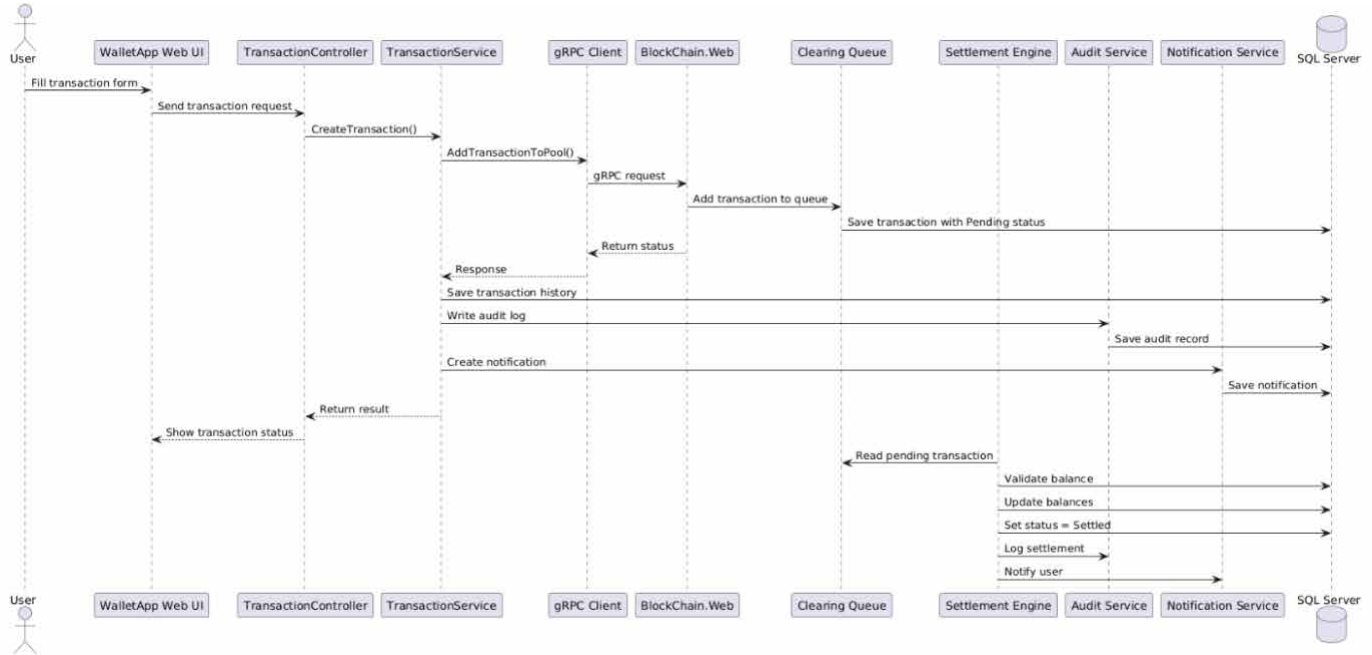
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System – [Електронний ресурс] - режим доступу: <https://bitcoin.org/bitcoin.pdf>
2. Antonopoulos A. M. Mastering Bitcoin: Programming the Open Blockchain. – O'Reilly Media , 2017. – С. 25–78, С.143–201.
3. Bashir I. Mastering Blockchain. – Packt Publishing, 2023. - С. 55–146, С.302–388.
4. Narayanan A., Bonneau J., Felten E., Miller A., Goldfeder S. Bitcoin and Cryptocurrency Technologies. – Princeton University Press, 2016. – С. 33–95, С. 210–248.
5. Microsoft Corporation. ASP.NET Core Documentation – [Електронний ресурс] - режим доступу : <https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-10.0>
6. Microsoft Corporation. Entity Framework Core Documentation – [Електронний ресурс] – режим доступу: <https://learn.microsoft.com/en-us/ef/core/>
7. SQL Server technical documentation – [електронний ресурс] – режим доступу: <https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver17>
8. Freeman A. Pro ASP.NET Core MVC 2. – Apress, 2018. - С. 112–245, С.412–538.
9. Fowler M. Patterns of Enterprise Application Architecture. – Addison-Wesley, 2002. - С. 45–118, 283–347.
10. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. – Addison-Wesley, 1994.- С. 13–30, С.135–212.

- 11.OWASP Foundation. OWASP Top 10 Web Application Security Risks. – [Электронный ресурс] – режим доступа: <https://owasp.org/www-project-top-ten/>
- 12.Audit.NET Documentation. – [Электронный документ] – режим доступа: <https://github.com/thepirat000/Audit.NET>
- 13.JWT.IO. JSON Web Tokens Introduction. – [Электронный ресурс] – режим доступа: <https://jwt.io>
- 14.Docker Documentation. – [Электронный ресурс] – режим доступа: <https://docs.docker.com>
- 15.Amazon Web Services Documentation. – [Электронный ресурс] – режим доступа: <https://docs.aws.amazon.com>
- 16.Fielding R. Architectural Styles and the Design of Network-based Software Architectures. – University of California, 2000. - С. 76–121.
- 17.UML Specification Version 2.5.1. Object Management Group (OMG) – [Электронный ресурс] – режим доступа: <https://www.omg.org/spec/UML/2.5.1/About-UML>
- 18.Pressman R. Software Engineering: A Practitioner’s Approach. – McGraw-Hill Education, 2014. - С. 55–144, С.603–688.
- 19.MDPI.Assessing Blockchain Consensus and Security Mechanisms against the 51% Attack - [Электронный ресурс] - Режим доступа: <https://www.mdpi.com/2076-3417/9/9/1788>
- 20.Dribbble. Money Transfer and Online Payments Website Homepage– [Электронный ресурс] – режим доступа: <https://dribbble.com/shots/6666395-Money-Transfer-and-Online-Payments-Website-Homepage>

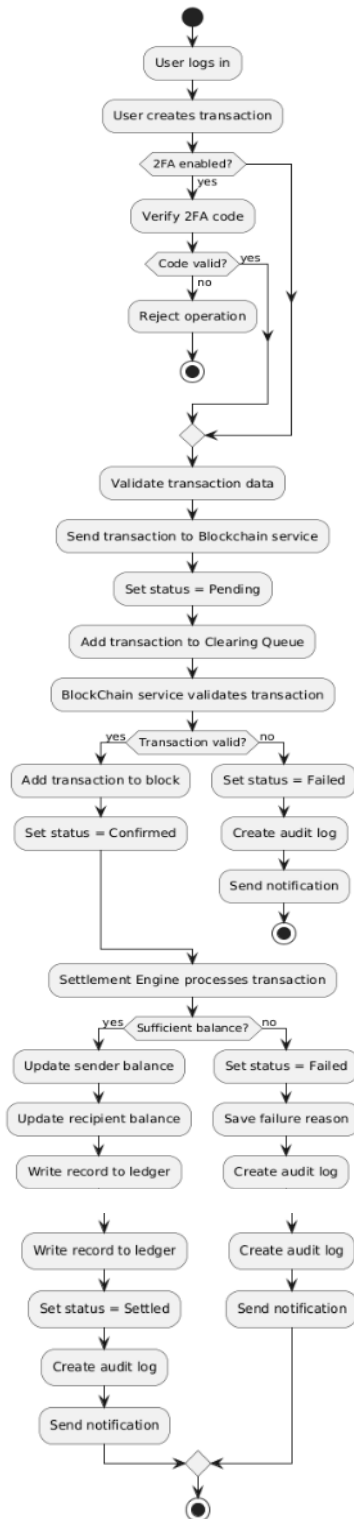
Додаток А

Діаграма послідовності



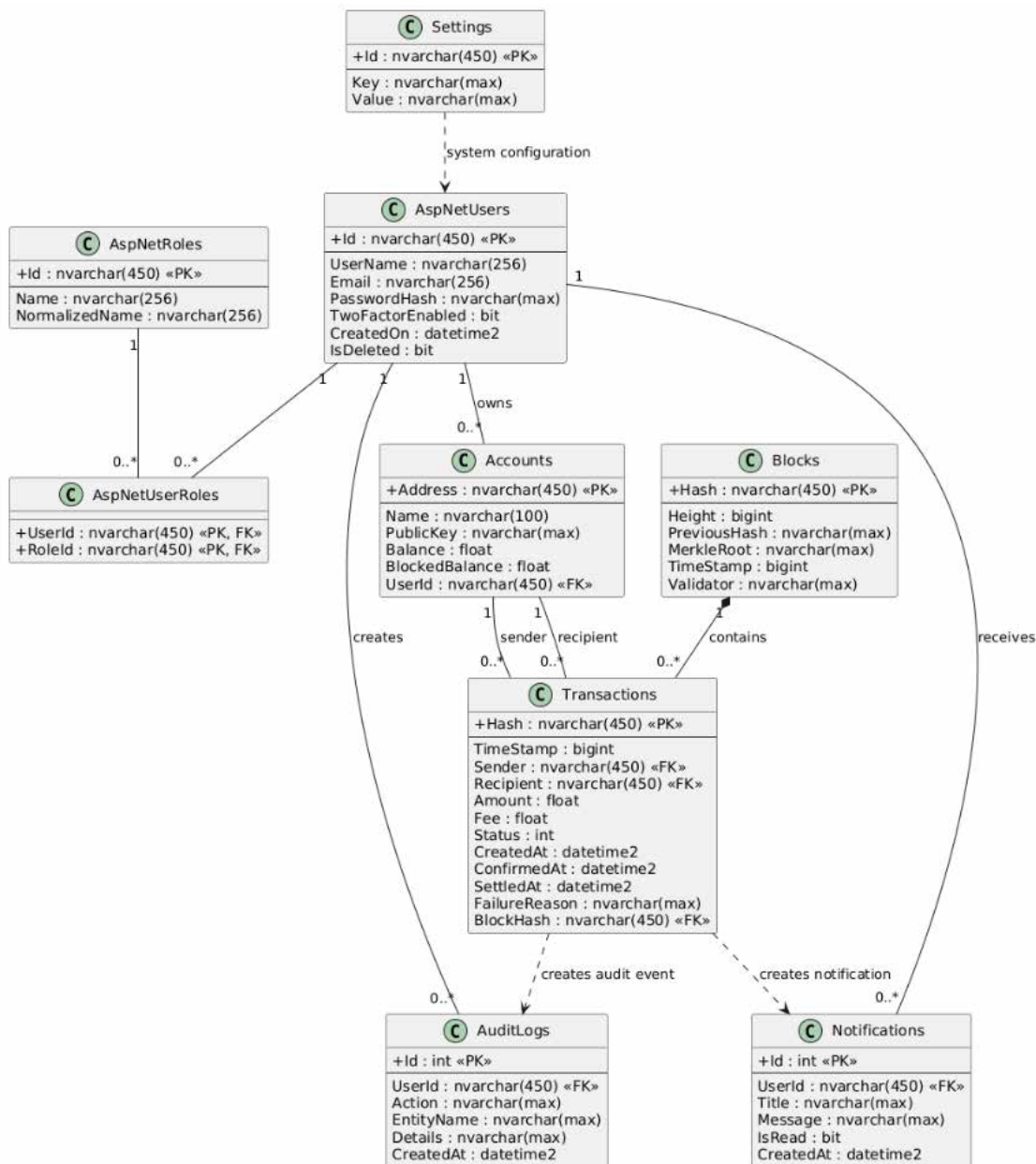
Додаток Б

Діаграма активності



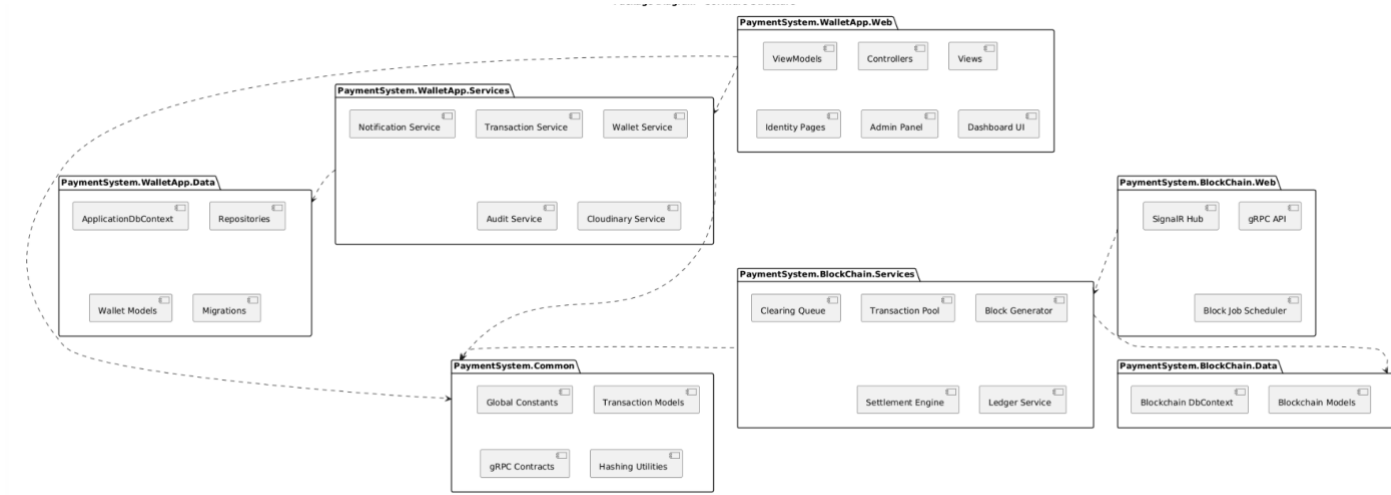
Додаток В

Фізична модель бази даних



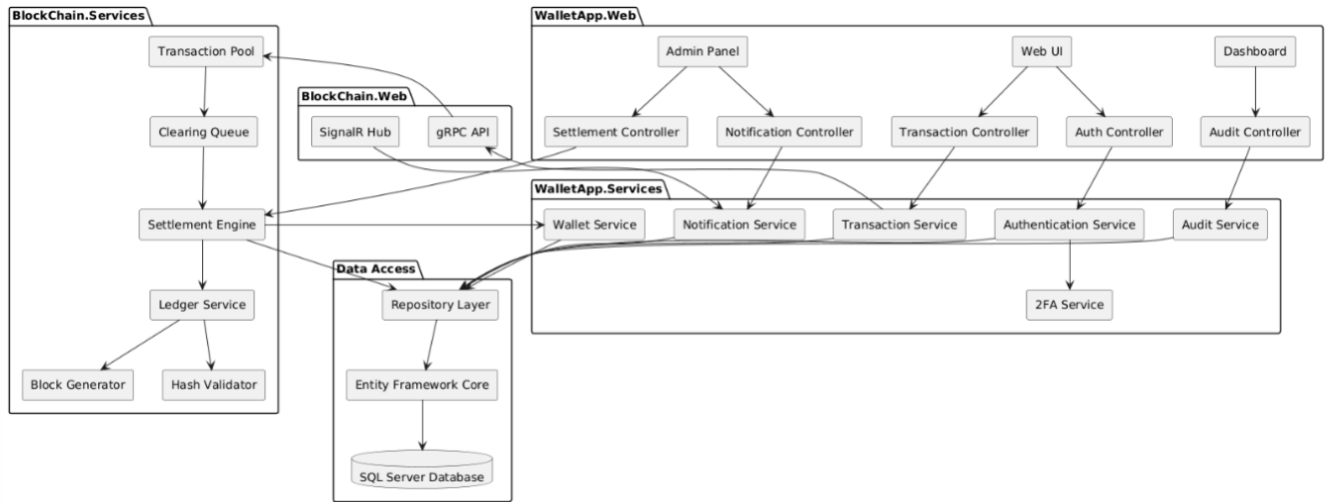
Додаток Г

Діаграма пакетів



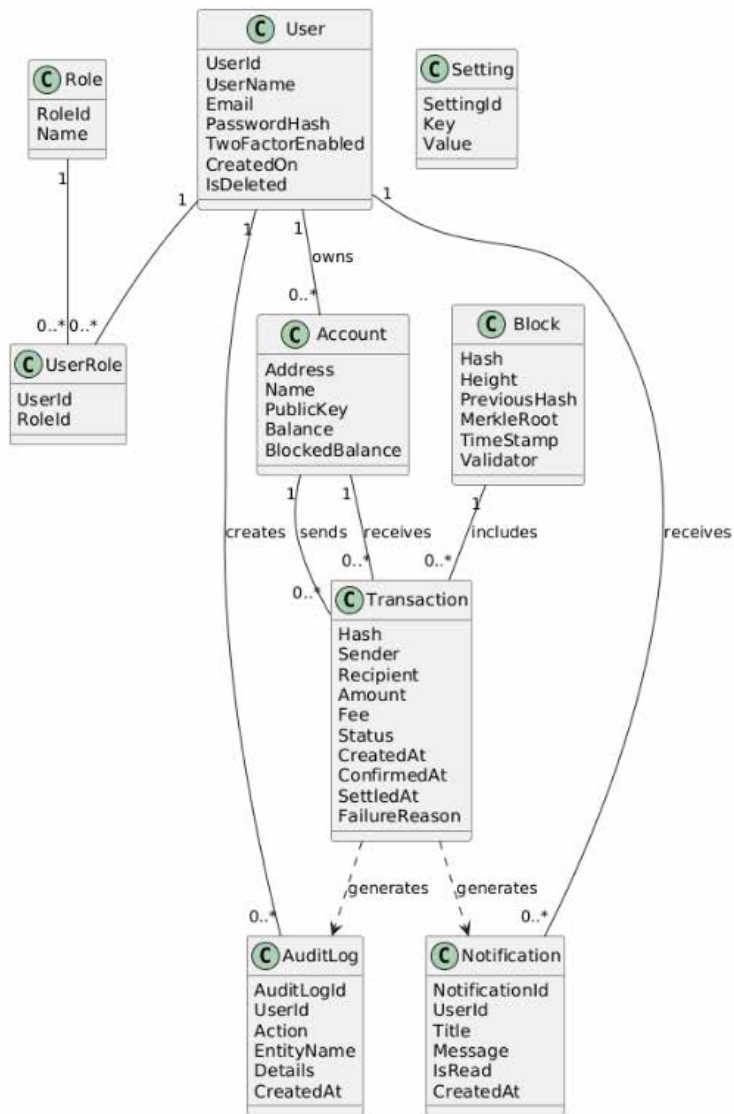
Додаток Д

Діаграма компонентів



Додаток Е

Логічна модель бази даних



Додаток Є

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

Декларація про академічну доброчесність та використання ШІ

ДЕКЛАРАЦІЯ ЗДОБУВАЧА

Я, Некрашевський Микола Андрійович, здобувач НУБіП України, усвідомлюючи відповідальність за порушення академічної доброчесності, цією заявою підтверджую, що:

1. Моя бакалаврська кваліфікаційна робота (проект) на тему «Програмне забезпечення для транзакцій і розрахунків на основі блокчейну» є результатом моєї особистої праці.
2. Усі запозичення з текстів інших авторів мають відповідні посилання згідно з чинними стандартами.
3. Щодо використання інструментів ШІ зазначаю, що (обрати необхідне):
☐ інструменти генеративного ШІ не використовувалися.
☒ інструменти ШІ (вказати назву: ChatGPT, Gemini, Claude, DeepL, _____ ChatGPT _____ інші (вказати назву)) були використані для:
☐ перекладу іншомовних джерел;
☒ стилістичного редагування та перевірки граматики;
☒ допомоги у написанні/відлагодженні програмного коду;

Я розумію, що надання недостовірної інформації може призвести до скасування результатів захисту та відрахування з числа здобувачів НУБіП України.

«___» _____ 2026 р. _____ **Микола НЕКРАШЕВСЬКИЙ**
(підпис)