

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики та комп'ютеризованих технологій
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та
робототехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)
Розроблення підсистеми автоматизації для взаємодії з постачальниками на
промисловому підприємстві
(тема)

Виконав:

здобувач 2 року навчання,

групи КІТПВМ-24-2

Ярослав ОЛІНКЕВИЧ

(власне ім'я, прізвище)

Спеціальність 174 Автоматизація, комп'ютерно-
інтегровані технології та робототехніка

(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Комп'ютерно-інтегровані
технологічні процеси та виробництва

(повна назва освітньої програми)

Керівник доц. Софія ХРУСТАЛЬОВА.

(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри _____

(підпис)

2025 р.

Ігор НЕВЛЮДОВ

(власне ім'я, прізвище)

Я, Олінкевич Ярослав Віталійович, як здобувач вищої освіти ХНУРЕ, розумію і підтримую політику закладу з академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Я не використовував штучний інтелект для підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

«08» грудня 2025 р.



Ярослав ОЛІНКЕВИЧ

Харківський національний університет радіоелектроніки

Факультет _____ АКТ _____

Кафедра _____ КІТАР _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність 174 Автоматизація, комп'ютерно-інтегровані технології _____

та робототехніка _____

(код і повна назва)

Тип програми _____ освітньо-професійна _____

Освітня програма Комп'ютерно-інтегровані технологічні процеси та виробництва _____

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

« _____ » _____

20 ____ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачу _____ Олінкевичу Ярославу Віталійовичу _____

(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення підсистеми автоматизації для взаємодії з постачальниками на промисловому підприємстві

затверджена наказом університету від 10 листопада 2025 р. № 1029Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 22 грудня 2025 р.

3. Вихідні дані до роботи. Форма діалогу: вебдодаток. Перелік програмних засобів, використовуваних у роботі: Visual Studio Code, JavaScript, Node.js, MongoDB, MongoDB Atlas, Ethereum, Hardhat, Materialize CSS, ethers.js, OpenZeppelin ERC-20. Технічне обладнання: комп'ютер або ноутбук з веббраузером Google Chrome.

4. Перелік питань, що потрібно опрацювати в роботі 4.1 Вступ; 4.2 Огляд і аналіз існуючих SRM-систем; 4.3 Розроблення вимог до підсистеми; 4.4 Програмна реалізація підсистеми; 4.5 Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Демонстраційний матеріал, представлений у форматі презентації PowerPoint (*.pptx) – 13 сторінок формату А4

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Огляд і аналіз існуючих SRM-систем	05.09 – 15.09.2025	Виконано
2	Розроблення вимог до підсистеми	15.09 – 01.10.2025	Виконано
3	Програмна реалізація підсистеми	01.10 – 23.11.2025	Виконано
4	Тестування роботи підсистеми автоматизації	24.11 – 29.11.2025	Виконано
5	Оформлення пояснювальної записки	30.11 – 16.12.2025	Виконано
6	Подання роботи на нормоконтроль	17.12.2025	Виконано
7	Подання роботи на перевірку Інтернет-сервісом StrikePlagiarism	18.12.2025	Виконано
8	Подання роботи на рецензування	19.12.2025	Виконано
9	Подання роботи на підпис зав. кафедри	22.12.2025	Виконано
10	Подання роботи до ЕК	-	Виконано

Дата видачі завдання 01 вересня 2025 р.

Здобувач _____
 Керівник роботи _____
 (підпис)



(підпис)

(підпис)

Ярослав ОЛІНКЕВИЧ
 доц. Софія ХРУСТАЛЬОВА
 (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка містить: 100 с., 3 табл., 35 рис., 3 дод., 31 джерел.

АВТОМАТИЗАЦІЯ, ПОСТАЧАЛЬНИК, СМАРТ-КОНТРАКТ, ESCROW, БЛОКЧЕЙН, SRM, ERP, ПІДСИСТЕМА.

Мета дослідження – підвищення ефективності взаємодії з постачальниками на промисловому підприємстві за рахунок розроблення SRM-рішення на базі браузерного додатку.

Об'єкт дослідження – процес взаємодії між менеджерами із закупівель промислового підприємства та зовнішніми постачальниками в рамках закупівельної діяльності.

Предмет дослідження – програмний засіб для автоматизації взаємодії з постачальниками на промисловому підприємстві.

Методи дослідження: аналіз предметної області та існуючих SRM-рішень, виділення базових вимог до розробки підсистеми SRM, моделювання логіки роботи підсистеми за допомогою UML-діаграм, розроблення алгоритмів роботи системи, конфігурація та розгортання тестового середовища блокчейну, розроблення серверної та клієнтської частин проекту та їхня інтеграція з блокчейном.

У роботі розглянуто актуальні питання за темою «Розроблення підсистеми автоматизації для взаємодії з постачальниками на промисловому підприємстві», запропоновано рішення для автоматизації процесів закупівельної діяльності на підприємстві. За результатами роботи було опубліковано наукову статтю в науково-технічному журналі.

ABSTRACT

The explanatory note contains: 100 pages, 3 tables, 35 figures, 3 appendices, 31 sources.

AUTOMATION, SUPPLIER, ESCROW, BLOCKCHAIN, SRM, ERP, SUBSYSTEM, SMART CONTRACT.

The purpose of the study is to improve the efficiency of interaction with suppliers at an industrial enterprise by developing an SRM solution based on a browser application.

The object of the study is the process of interaction between procurement managers of an industrial enterprise and external suppliers within the framework of procurement activities.

The subject of the study is a software tool for automating interaction with suppliers at an industrial enterprise.

Research methods: analysis of the subject area and existing SRM solutions, identification of basic requirements for the development of the SRM subsystem, modeling of the subsystem's logic using UML diagrams, development of system operation algorithms, configuration and deployment of a blockchain test environment, development of the server and client parts of the project and their integration with the blockchain.

The work considers topical issues on the theme “Development of an automation subsystem for interaction with suppliers at an industrial enterprise” and proposes solutions for automating procurement processes at the enterprise. Based on the results of the work, a scientific article was published in a scientific and technical journal.

ЗМІСТ

Перелік умовних скорочень	9
Вступ.....	10
1. Огляд і аналіз існуючих методів, засобів та систем автоматизації для взаємодії з постачальниками.....	12
1.1 Аналіз поточного стану питання взаємодії з постачальниками на промислових підприємствах	12
1.2 Оцінювання економічного ефекту та операційної результативності впровадження SRM-системи на промисловому підприємстві	13
1.3 Аналіз існуючих програмних рішень в області автоматизації взаємодії з постачальниками	15
1.3.1 Архітектурний аналіз SAP Ariba	16
1.3.2 Архітектурний аналіз Oracle ERP Cloud.....	18
1.3.3 Узагальнення виявлених недоліків централізованих SRM-систем	19
1.4 Постановка задачі.....	21
2 Розроблення вимог, структурної схеми та алгоритму роботи підсистеми для автоматизації взаємодії з постачальниками	23
2.1 Концептуальна модель гібридної Blockchain-SRM підсистеми.....	23
2.2 Формулювання функціональних вимог до розроблюваної підсистеми автоматизації взаємодії з постачальниками	26
2.2.1 Окреслення системних вимог до компонентів підсистеми автоматизації взаємодії з постачальниками.....	26
2.2.2 Розроблення вимог до клієнтської частини розроблюваної підсистеми автоматизації взаємодії з постачальниками	27
2.3 UML-моделювання логіки роботи підсистеми автоматизації взаємодії з постачальниками	27

2.3.1 Розроблення діаграми прецедентів для підсистеми автоматизації взаємодії з постачальниками.....	29
2.3.2 Розроблення діаграми компонентів для підсистеми автоматизації взаємодії з постачальниками.....	34
2.3.3 Розроблення діаграм послідовностей.....	36
2.4 Охорона праці.....	40
3 Реалізації підсистеми автоматизації у вигляді програмного засобу.....	45
3.1 Обґрунтування вибору технологій для розроблення SRM-підсистеми.....	45
3.2 Розроблення алгоритмів поведінки акторів під час взаємодії з підсистемою.....	47
3.2.1 Розроблення алгоритму поведінки менеджера з закупівель.....	47
3.2.2 Розроблення алгоритму поведінки представника постачальника.....	49
3.3 Розроблення логічної структури даних.....	50
3.4 Опис механізму роботи blockchain-модулю та його інтеграція з backend-сторонаю проекту.....	54
3.5 Розроблення користувацьких інтерфейсів.....	57
Висновки.....	68
Перелік джерел посилання.....	70
Додаток А Апробація наукових результатів дослідження.....	75
Додаток Б Лістинг програми.....	85
Додаток В Демонстраційний матеріал.....	99

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ДСТУ – державний стандарт України;

НПАОП – нормативно-правовий акт з охорони праці;

ПУЕ – правила улаштування електроустановок;

СУБД – система управління базами даних;

API – Application Programming Interface;

BSON – Binary JSON;

BPMS – Business Process Management Systems;

CLM – Contract Lifecycle Management;

CSS – Cascading Style Sheets;

ERP – Enterprise Resource Planning;

HTML – HyperText Markup Language;

JSON – JavaScript Object Notation;

KPI – Key Performance Indicators;

P2P – Procure-To-Pay;

S2C – Source-to-Contract;

SRM – Supplier Relationship Management;

UI – User Interface;

UML – Unified Modeling Language.

ВСТУП

Актуальність обраної теми зумовлена необхідністю трансформації традиційних процесів управління ланцюгами постачання промислових підприємств відповідно до викликів глобальної цифрової економіки та цілей сталого розвитку ООН. Сучасні централізовані SRM-системи, незважаючи на високий рівень функціональності, демонструють системні архітектурні обмеження у забезпеченні довіри між контрагентами, атомарності фінансових операцій та прозорості транзакційних процесів. Застосування технології розподілених реєстрів для автоматизації критичних бізнес-процесів взаємодії з постачальниками відкриває можливості для створення принципово нової моделі міжкорпоративної співпраці, заснованої на криптографічних гарантіях незмінності даних та програмованій бізнес-логіці смарт-контрактів. Запропонований підхід безпосередньо узгоджується з Ціллю 9 «Індустріалізація, інновації та інфраструктура» через модернізацію промислової інфраструктури та забезпечення доступу малих підприємств до інноваційних інструментів автоматизації, а також з Ціллю 12 «Відповідальне споживання і виробництво» через забезпечення повної прозорості походження матеріалів та стимулювання відповідальної поведінки всіх учасників процесу закупівлі.

Мета дослідження – підвищення ефективності взаємодії з постачальниками на промисловому підприємстві за рахунок розроблення SRM-рішення на базі браузерного додатку.

Об'єкт дослідження – процес взаємодії між менеджерами із закупівель промислового підприємства та зовнішніми постачальниками в рамках закупівельної діяльності.

Предмет дослідження – програмний засіб для автоматизації взаємодії з постачальниками на промисловому підприємстві.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- здійснити огляд і аналіз існуючих методів, засобів та автоматизованих систем для взаємодії з постачальниками;
- розробити вимоги, структурну схему та алгоритм роботи підсистеми для автоматизації взаємодії з постачальниками;
- реалізувати підсистему автоматизації у вигляді програмного засобу;
- виконати тестування роботи підсистеми автоматизації;
- оформити звіт з практики згідно ДСТУ 3008:2015, методичними вказівками з підготовки та захисту кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка освітньо-професійної програми «Комп'ютерно-інтегровані технологічні процеси і виробництва», а також загальними рекомендаціями [1-7]. За результатами роботи було опубліковано наукову статтю у фаховому журналі, наведено її текст у Додатку А [8].

1 ОГЛЯД І АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ, ЗАСОБІВ ТА СИСТЕМ АВТОМАТИЗАЦІЇ ДЛЯ ВЗАЄМОДІЇ З ПОСТАЧАЛЬНИКАМИ

1.1 Аналіз поточного стану питання взаємодії з постачальниками на промислових підприємствах

Цільовою предметною областю є малі та середні виробничі підприємства дискретного циклу, які спеціалізуються на виготовленні складної техніки або обладнання. Особливості такого типу виробництва – залежність від широкого асортименту сировини та необхідність жорсткого контролю ланцюгів постачання. Часта трансформація глобальних ланцюгів поставок, посилення залежності виробничих процесів від надійності партнерських зв'язків та необхідність забезпечення безперервності операційної діяльності висувають ефективне управління виробничими ресурсами в число найпріоритетніших задач для сучасних підприємств такого типу. Актуальність цієї проблематики зумовлена низкою об'єктивних чинників:

- динамічність постачальницьких мереж, що характеризується постійними змінами кількості та складу партнерів, вимагає від підприємств гнучкості та додаткових витрат для ефективного моніторингу та оцінювання діяльності постачальників;

- зростання вимог до прозорості бізнес-процесів та необхідність дотримання міжнародних стандартів корпоративного управління створюють потребу в систематизації взаємодії з контрагентами;

- волатильність цін на сировину та матеріали, а також непередбачувані логістичні виклики роблять критично важливим оперативне реагування на зміни умов співпраці;

– посилення регуляторних вимог щодо походження товарів, етичності виробництва та дотримання екологічних норм постачальниками зобов'язує підприємства здійснювати ретельний контроль усього ланцюга створення вартості (Value Creation Cycle);

– глобальна та стрімка цифровізація економіки й поширення електронних форматів комунікації відкривають можливості для автоматизації документообігу та підвищення оперативності прийняття рішень у сфері закупівель.

За таких умов створення ефективної системи управління взаємодією з постачальниками дозволяє промисловим підприємствам досягти суттєвих конкурентних переваг: оптимізувати витрати на матеріально-технічне забезпечення виробництва; мінімізувати ризики зриву термінів поставок та дефіциту ресурсів; сформувати базу надійних партнерів із перевіреною репутацією; забезпечити гнучкість виробничих графіків завдяки диверсифікації джерел постачання; підвищити якість вхідної сировини через систематичний аудит постачальників.

1.2 Оцінювання економічного ефекту та операційної результативності впровадження SRM-системи на промисловому підприємстві

Сучасний ринок програмного забезпечення насичений різноманітними рішеннями класу BPMS, що призначені для автоматизації господарських процесів компаній. Для вирішення задач взаємодії з постачальниками застосовуються SRM-системи, які зазвичай реалізуються як функціональні модулі глобальних ERP-платформ – комп'ютерних систем для автоматизації складних, багатоступінчатих бізнес-процесів з оперативного планування, виробництва та обслуговування клієнтів – або як автономні сервіси, що з'єднуються з ERP-системами через інтеграційні API-сервіси [9]. SRM-система є програмним рішенням, призначення якої полягає в автоматизації процесів закупівельної діяльності підприємства. Її

застосування дозволяє підвищити керованість ланцюгів постачання через ефективний вибір контрагентів і аналіз результатів ефективності співпраці та забезпечити прозорість процесів закупівлі, а кінцевою метою використання такого ПЗ є максимізація доходів підприємства при одночасному зниженні його витрат [10]. Функціональну структуру SRM-системи зображено на рисунку 1.1.



Рисунок 1.1 – Компоненти системи управління взаємовідносинами з постачальниками [8]

Функціонал SRM-системи може охоплювати: створення єдиного сховища інформації про партнерів із доступом до історії взаємодії з ними, розподіл контрагентів за критеріями результативності та стратегічного значення для підприємства, дослідження їхньої фінансової стабільності та репутації для виявлення потенційних проблем; автоматизацію процесів придбання ресурсів через об'єднання з системою ERP, що усуває можливі помилки, спричинені людським фактором; зіставлення пропозицій від різних учасників ринку, відстеження дотримання домовленостей на всіх стадіях постачання та синхронізацію з бухгалтерськими інструментами для точного фінансового обліку; здійснення

систематичної перевірки партнерів за параметрами пунктуальності поставок, відповідності поставленої продукції та/або послуг очікуванням підприємства, формування оціночних реєстрів для виваженого відбору контрагентів для подальшої співпраці та підготовку аналітичних матеріалів про фінансові показники й потенційні небезпеки; зміцнення партнерських зв'язків завдяки прозорим каналам співпраці, швидкому врегулюванню суперечок та покращенню організації бізнес-процесів, що приносить економічні переваги обом учасникам; гарантування законності домовленостей через перевірку їхньої відповідності нормативним актам та зменшення небезпек, спричинених недоброчесністю партнерів.

1.3 Аналіз існуючих програмних рішень в області автоматизації взаємодії з постачальниками

Незважаючи на численні переваги, впровадження SRM-системи супроводжується певними викликами та обмеженнями. До основних недоліків належать: високі початкові інвестиції на придбання програмного забезпечення, складність його інтеграції з існуючою програмною екосистемою й налаштування під вимоги підприємства; ризик опору з боку співробітників через необхідність зміни усталених робочих процесів, вивчення технічної документації, проходження навчання тощо; потреба в регулярному оновленні та технічній підтримці системи протягом еволюції ринкового середовища. Ці фактори обумовлюють важливість ретельного аналізу існуючих на ринку SRM-рішень шляхом порівняння їх функціональних можливостей, вартості ліцензії/оренди та підтримки, а також гнучкості готового програмного рішення для адаптації під специфічні потреби бізнесу для вибору оптимального варіанта.

Сучасні SRM-системи, такі як SAP Ariba та Oracle ERP Cloud, досягли високого рівня автоматизації, охоплюючи стратегічний сорсинг, управління контрактами (CLM), закупівлі та обробку рахунків-фактур. Однак незважаючи на значні переваги, архітектура цих систем залишається централізованою, що створює єдину точку довіри та залежність від внутрішніх політик компаній для забезпечення цілісності даних. SRM-система традиційно є централізованим програмним рішенням, що зберігає всі дані щодо постачальників в одному місці [11]. Таким чином, виконання контрактів у них залежить від встановлених внутрішніх бізнес-правил, схвалення уповноваженими особами та юридичних документів, що зберігаються у реляційних базах даних (RDBMS). З цього випливає ряд архітектурних недоліків:

- керування контрактами перетворює юридичні документи на електронні угоди з метаданими, що зберігаються всередині бази даних компанії-власника SRM, але ці документи не є самовиконуваними програмними активами, через що виникають проблеми довіри, оскільки критичні транзакційні записи не є незмінними [12];

- для гарантованого умовного блокування коштів (Escrow), система змушена залучати традиційні банківські ескроу-рахунки або довірених третіх сторін.

1.3.1 Архітектурний аналіз SAP Ariba

SAP Ariba є визнаним світовим лідером у сфері рішень Source-to-Contract (S2C) та Procure-to-Pay (P2P) [13]. Платформа функціонує на базі Ariba Network – найбільшої бізнес-мережі. Ключові функціональні можливості включають Управління життєвим циклом контрактів (CLM) та Управління рахунками-фактурами (Invoice Management).

Однак, високий рівень автоматизації P2P-процесів у SAP Ariba не означає їхньої повної автономності. Критичне архітектурне обмеження полягає у процесі

обробки виключень (Invoice Exceptions). Такі виключення виникають у разі невідповідності даних у рахунку-фактурі даним у замовленні на закупівлю або даних про отримання. У цих випадках система створює документ примирення рахунку (Invoice Reconciliation document) та вимагає ручного розгляду та зіставлення для вирішення невідповідності, перш ніж платіж буде ініційовано. Це означає, що платіж не є атомарною операцією, а підтвердження доставки та виконання платежу відбуваються як окремі, послідовні процеси (рис. 1.2).

The screenshot shows the SAP logo in the top left corner. In the top right, there is a language selection menu with a globe icon, the text 'English', and a downward arrow. Below the logo, the text 'FAQ KB0408430' is displayed. The main title of the FAQ is 'How to resolve an Item Unmatched invoice exception with manual match?'. Underneath, there is a section header 'Symptom' followed by the text 'How to resolve an **Item Unmatched** invoice exception with manual match?'. Another section header 'Resolution' is followed by the text 'On the invoice reconciliation, select the line items with the **Item Unmatched** exception and click **Manual Match**.' Below this, it says 'For each unmatched line, perform the following steps:' followed by a numbered list:

1. Select **Unmatched** from the **Manually Match To** pull-down menu.
2. Click **select** to display available purchase order or contract lines.
3. For the purchase order or contract line that you want to match to the invoice line item, click **Select**.
4. Click **OK**.
5. Make any other edits to the line as needed.

Рисунок 1.2 – Інструкція від SAP для кінцевих користувачів щодо обробки виключень типу Invoice Exceptions

SAP, хоча і є контриб'ютором Hyperledger Foundation, міжнародного проєкту з розробки блокчейну (децентралізованої розподіленої бази даних) з відкритим вхідним кодом, застосовує блокчейн переважно для додаткових функцій, таких як Traceability (відстеження походження) та підвищення прозорості ланцюга

постачання. Це підтверджується галузевими дослідженнями, де фокус інтеграції блокчейну в SAP знаходиться у сфері SCM (Supply Chain Management) для забезпечення незмінності записів про рух товару [14]. Ці рішення не застосовуються для базового процесу P2P-оплати. P2P-процес у SAP Ariba, як і раніше, залежить від централізованих фінансових модулів для виконання платежу. Таким чином, блокчейн у SAP фіксує нефінансові дані про товар, але не виконує атомарну фінансову транзакцію Escrow, залишаючи проблему ручного вирішення Invoice Exceptions актуальною для основного функціоналу Ariba.

1.3.2 Архітектурний аналіз Oracle ERP Cloud

Компанія Oracle є визнаним лідером в галузі Source-to-Pay (S2P) рішень [15]. Oracle Fusion Cloud Procurement, що є інтегрованим у глобальну ERP-систему модулем у складі пакета SCM, автоматизує бізнес-процеси, забезпечує стратегічний сорсинг та управління постачальниками. Платежі ініціюються на основі внутрішніх бізнес-правил та потребують схвалення через налаштовані корпоративні робочі процеси [16].

Архітектура Oracle ERP Cloud SCM покладається на централізовані реляційні СУБД для роботи з основним обсягом даних про транзакції. Зберігання критичних транзакційних даних у реляційній СУБД призводить до дефіциту архітектурної довіри. Технічно адміністратор центральної бази даних має можливість модифікувати історичні записи, і довіра до історичних даних залишається організаційною.

Oracle пропонує Oracle Blockchain Platform (OBP), побудовану на основі відкритого коду Hyperledger Fabric, як керований хмарний сервіс. Ця платформа може інтегруватися з Oracle SCM Cloud та ERP Cloud. OBP використовується для створення незмінного реєстру, відстеження походження продукції та забезпечення прозорості ланцюга постачання. Однак блокчейн залишається опціональним

додатковим модулем, а не базовою архітектурою P2P-системи, і стандартні сценарії закупівлі та оплати продовжують функціонувати на централізованій основі [17].

1.3.3 Узагальнення виявлених недоліків централізованих SRM-систем

Проведений архітектурний аналіз провідних SRM-рішень виявив системні обмеження, притаманні централізованій парадигмі побудови систем управління постачальниками. Ці обмеження можна класифікувати за трьома ключовими категоріями: архітектурні, операційні та економічні.

До архітектурних недоліків належить фундаментальна проблема довіри, що впливає зі зберігання критичних транзакційних даних у централізованих реляційних базах даних. Технічно адміністратори цих систем мають можливість модифікувати історичні записи, що робить незмінність даних організаційною політикою, а не технологічною гарантією. Це створює єдину точку довіри та робить систему вразливою до внутрішніх зловживань або компрометації облікових записів привілейованих користувачів. Крім того, відсутність криптографічних механізмів верифікації призводить до того, що у випадку суперечок між сторонами немає технічно незаперечного доказу послідовності подій та стану транзакцій.

Операційні обмеження проявляються у відсутності атомарності критичних бізнес-процесів. Процес закупівлі в існуючих системах реалізовано як послідовність окремих етапів: створення замовлення, підтвердження поставки, узгодження рахунку-фактури та ініціювання платежу. Кожен із цих етапів може бути перерваний або потребувати ручного втручання, що створює проміжні стани невизначеності. Особливо критичною є проблема обробки виключень типу Invoice Exceptions у SAP Ariba, коли система виявляє невідповідність між даними замовлення та рахунку-фактури і вимагає ручного втручання для вирішення конфлікту. Це порушує автоматизацію процесу та створює затримки у розрахунках з постачальниками.

Механізм умовного блокування коштів Escrow в обох проаналізованих системах реалізовано через залучення зовнішніх фінансових інституцій або довірених третіх сторін. Це створює додаткові витрати, збільшує час виконання транзакцій та вводить додаткові точки можливого збою процесу. Постачальник змушений чекати на завершення багатоступінчатих процедур узгодження та схвалення перед отриманням оплати, що негативно впливає на його оборотний капітал.

Незважаючи на те, що обидві компанії декларують інтеграцію з блокчейн-технологіями, фактичне застосування блокчейну обмежується допоміжними функціями. Базові P2P-процеси залишаються централізованими, а блокчейн використовується лише для фіксації нефінансових даних про рух товарів. Таким чином, потенціал технології розподілених реєстрів для вирішення фундаментальних проблем довіри та автоматизації залишається нереалізованим у контексті основного функціоналу SRM-систем.

Економічні недоліки централізованих рішень пов'язані з бізнес-моделлю провідних постачальників. Ліцензування та впровадження SAP Ariba або Oracle ERP Cloud вимагає значних початкових інвестицій, що робить ці рішення складнодоступними для малих та середніх підприємств. Додаткові витрати виникають через необхідність залучення спеціалізованих консультантів для налаштування системи, тривалих циклів впровадження та регулярних платежів за технічну підтримку та оновлення. Монополізація ринку кількома великими постачальниками створює ситуацію vendor lock-in, коли підприємство стає критично залежним від обраного рішення і міграція на альтернативну платформу пов'язана з неприйнятно високими витратами.

Важливо зазначити, що провідні постачальники корпоративного програмного забезпечення усвідомлюють потенціал технології блокчейн, про що свідчить їхня дотичність до Hyperledger та розробка блокчейн-модулів для відстеження

походження товарів. Однак повноцінна інтеграція блокчейну як базової архітектури для P2P-процесів стримується кількома об'єктивними факторами: по-перше, існуюча бізнес-модель цих компаній побудована на централізованому контролі та довгострокових ліцензійних угодах, що забезпечує стабільний дохід; по-друге, впровадження децентралізованої архітектури вимагає координації між конкуруючими учасниками ринку (постачальники також мають бути учасниками блокчейн-мережі); по-третє, юридична невизначеність правового статусу смарт-контрактів у багатьох юрисдикціях створює регуляторні ризики; по-четверте, консервативність корпоративних клієнтів, які віддають перевагу перевіреним технологіям перед інноваційними рішеннями з обмеженою історією промислової експлуатації.

Виявлені архітектурні та операційні недоліки обґрунтовують доцільність дослідження альтернативного підходу до побудови SRM-систем на основі технології розподілених реєстрів. При цьому слід розуміти, що запропоноване рішення не претендує на повне усунення економічних бар'єрів впровадження корпоративних систем, а зосереджується на вирішенні фундаментальних технологічних обмежень централізованих архітектур, особливо актуальних для сценаріїв міжкорпоративної співпраці, де довіра між учасниками є критичним фактором.

1.4 Постановка задачі

Основною задачею кваліфікаційної роботи є розробка гібридної комп'ютерної веб-орієнтованої підсистеми для автоматизації процесів взаємодії з постачальниками на промисловому підприємстві. Для того, щоби підсистема вважалася SRM-подібною, її функціонал має відповідати базовому набору

інструментів, притаманних SRM-системам, які описано в пункті 1.2, а також застосовувати технологію blockchain як базовий механізм для укладання угод.

Планується, що підсистема складатиметься з трьох компонентів: веб-інтерфейсу для менеджерів із закупівель, веб-інтерфейсу для представників постачальників, а також блокчейн-модуля з реалізацією логіки формування смарт-контрактів.

Пропонована підсистема підтримуватиме наступний функціонал:

- безшовну інтеграцію з корпоративною ERP-системою для обміну даними;
- традиційний список SRM-функцій, що охоплюють управління каталогом постачальників та інструменти для аналітики ефективності взаємодії з постачальниками;
- створення та розгортання смарт-контрактів як самовиконуваних угод;
- програмну реалізацію механізму умовного блокування коштів через механізм ERC-20 Allowance;
- атомарне виконання операції «поставка-оплата» після криптографічно підтвердженого факту поставки.

2 РОЗРОБЛЕННЯ ВИМОГ, СТРУКТУРНОЇ СХЕМИ ТА АЛГОРИТМУ РОБОТИ ПІДСИСТЕМИ ДЛЯ АВТОМАТИЗАЦІЇ ВЗАЄМОДІЇ З ПОСТАЧАЛЬНИКАМИ

2.1 Концептуальна модель гібридної Blockchain-SRM підсистеми

На основі виявлених архітектурних обмежень централізованих SRM-систем, зокрема проблем довіри до історичних даних, відсутності атомарності фінансових операцій та необхідності залучення третіх сторін для гарантування виконання контрактів, пропонується гібридна модель підсистеми управління взаємовідносинами з постачальниками, яка поєднує традиційний функціонал бізнес-автоматизації з децентралізованим виконанням критичних транзакцій через механізм смарт-контрактів.

Ключова ідея запропонованого рішення полягає у розподілі функціональних обов'язків між централізованим прикладним рівнем та децентралізованим транзакційним рівнем. Централізований компонент зберігає за собою традиційні SRM-функції: управління каталогом постачальників, аналітику ефективності співпраці, формування звітності та інтеграцію з ERP-системою підприємства. Водночас критичні операції, що вимагають незмінності, прозорості та автоматичного виконання без участі довірених посередників, переносяться на рівень блокчейн-платформи.

Архітектурна новизна запропонованої підсистеми полягає у застосуванні смарт-контрактів як самовиконуваних програмних активів, що реалізують логіку угод між підприємством та постачальниками. На відміну від традиційних електронних договорів, що зберігаються у реляційних базах даних у вигляді документів з метаданими, смарт-контракти є виконуваним програмним кодом,

розгорнутим у децентралізованій мережі. Це забезпечує атомарність виконання бізнес-логіки: умови контракту перевіряються та виконуються автоматично без можливості односторонньої зміни або скасування після розгортання.

Фундаментальною перевагою такого підходу є вирішення проблеми довіри через криптографічні гарантії незмінності. Кожна транзакція, зафіксована у блокчейні, захищена криптографічним хешуванням та підтверджена консенсусом вузлів мережі, що робить будь-яку спробу модифікації історичних записів технічно неможливою.

Критично важливим елементом запропонованої архітектури є реалізація механізму умовного блокування коштів безпосередньо на рівні смарт-контракту. Замість залучення банківських ескроу-рахунків або довірених третіх сторін, кошти резервуються програмним шляхом при створенні замовлення та автоматично вивільнюються після криптографічно підтвердженого факту поставки. Це забезпечує атомарність операції «поставка-оплата»: або обидві дії виконуються успішно, або жодна з них не відбувається, що усуває проблему Invoice Exceptions.

Запропонована модель передбачає використання локальної Ethereum-мережі (Hardhat), що імітує поведінку публічного блокчейну. Для промислового впровадження рекомендується використання дозвоільного блокчейну (наприклад, Hyperledger Besu або Quorum) з контролем доступу на рівні вузлів мережі.

Для забезпечення конфіденційності фінансової інформації у смарт-контракті зберігаються лише криптографічні хеші специфікацій та їхні ідентифікатори, тоді як детальні дані про матеріали, ціни та умови співпраці залишаються у централізованій базі даних з контрольованим доступом.

Інтеграція блокчейн-компоненту з існуючою корпоративною інфраструктурою здійснюється через прикладний програмний інтерфейс (API), що

дозволяє ERP-системі ініціювати створення смарт-контрактів, відстежувати їхній стан та отримувати індикацію про завершені транзакції. Така архітектура забезпечує поступове впровадження технології без необхідності повної заміни існуючих систем, що знижує бар'єр входу для промислових підприємств.

Таблиця 2.1 – Порівняльний аналіз архітектурних особливостей існуючих SRM-систем та запропонованої гібридної Blockchain-SRM підсистеми

Критерій аналізу	SAP Arriba	Oracle ERP Cloud	Запропонована Blockchain-SRM підсистема
Архітектура бази даних	Централізована реляційна СУБД	Централізована реляційна СУБД	Гібридна (NoSQL + розподілений реєстр)
Основа довіри та імутабельності	Організаційна політика	Організаційна політика	Математична гарантія (криптографічна незмінність)
Механізм Escrow для оплати	Традиційний банківський/треті особи	Традиційний банківський/треті особи	Програмний Escrow (ERC-20 Allowance)
Обробка виключень	Ручне втручання (Manual Match)	Ручне схвалення/налаштований workflow	Автономна логіка в Smart Contract (програмне вирішення)
Тип блокчейн-інтеграції	Опціональний	Опціональний	Базова архітектура на Ethereum/Solidity
Атомарність транзакцій	Ні (послідовні, роздільні процеси)	Ні (послідовні, роздільні процеси)	Так (забезпечується смарт-контрактом)

2.2 Формулювання функціональних вимог до розроблюваної підсистеми автоматизації взаємодії з постачальниками

2.2.1 Окреслення системних вимог до компонентів підсистеми автоматизації взаємодії з постачальниками

Для досягнення мети поставленої задачі з автоматизації господарських процесів підприємства, необхідно, щоби всі компоненти підсистеми відповідали наступним системним вимогам:

- побудову підсистеми на основі гібридної архітектури, що включає три рівні (клієнтська частина, сервер застосунків, сервер бази даних) та децентралізований транзакційний рівень (blockchain-модуль);
- база даних має бути нереляційною (NoSQL) та побудованою на базі СУБД MongoDB;
- клієнтська частина має являти собою користувацький інтерфейс (UI) у вигляді HTML-сторінок, що підтримується сучасними браузером;
- UI має бути інтуїтивно зрозумілим та мати інтерактивні компоненти, що ініціюють виконання логічних операцій всередині підсистеми;
- blockchain-модуль повинен функціонувати в EVM-сумісному тестовому середовищі (Hardhat) та реалізовувати логіку управління фінансовими операціями на основі стандарту ERC-20;
- підсистема повинна дозволяти обом сторонам (менеджеру з закупівель та постачальнику) підписувати угоду та ініціювати зміни статусу контракту, автоматично здійснювати переказ токенів з рахунку компанії на рахунок постачальника;
- підсистема має розраховувати ключові показники ефективності (KPI) постачальників на основі даних, отриманих внаслідок укладання угод.

2.2.2 Розроблення вимог до клієнтської частини розроблюваної підсистеми автоматизації взаємодії з постачальниками

UI запропонованої SRM-підсистеми має складатися з двох модулів: кабінету представника компанії-замовника – менеджера з закупівель – та кабінету представника постачальника. Ці модулі мають бути автономними, але функціонально пов'язаними в backend-частині проєкту для взаємодії один з одним. Для отримання доступу до свого кабінету, діюча особа має отримати від адміністраторів ERP-системи попередньо зареєстровані в БД дані для входу в систему та здійснити процедури автентифікації та авторизації. Після цього на відповідних сторінках інтерфейсу особі відкриється можливість здійснення ряду операцій, що знаходиться в рамках її повноважень.

2.3 UML-моделювання логіки роботи підсистеми автоматизації взаємодії з постачальниками

UML (Unified Modeling Language) – уніфікована мова моделювання, що використовується розробниками програмного забезпечення для візуалізації процесів та роботи систем [18].

Основна мета UML полягає в тому, щоб надати розробникам програмного забезпечення універсальний інструмент для моделювання систем незалежно від конкретної мови програмування чи платформи. UML дозволяє створювати абстрактні моделі системи на різних рівнях деталізації. Це особливо важливо для складних проєктів, де потрібна чітка візуалізація структури та поведінки системи ще до початку написання коду.

UML також відіграє важливу роль на етапі планування та проєктування. Створення моделей системи до написання коду допомагає виявити потенційні проблеми в архітектурі, знайти неоптимальні рішення та зрозуміти складність

майбутньої системи. Це значно дешевше, ніж виправляти помилки на етапі розробки або після випуску продукту. Моделювання дозволяє експериментувати з різними архітектурними підходами, порівнювати альтернативні рішення та обирати найкращий варіант без витрат на реальну імплементацію.

UML складається з різних типів діаграм, кожна з яких призначена для моделювання певних аспектів системи. Структурні діаграми показують статичну структуру системи – які елементи існують та як вони пов'язані між собою. Діаграма класів, наприклад, є однією з найпопулярніших структурних діаграм, яка показує класи системи, їхні атрибути, методи та відношення між класами (асоціація, агрегація, композиція, успадкування). Це надзвичайно корисно при проектуванні об'єктно-орієнтованих систем, оскільки дозволяє побачити всю ієрархію класів та їх взаємозв'язки на одній діаграмі.

Типи діаграм UML наведено на рис. 2.1

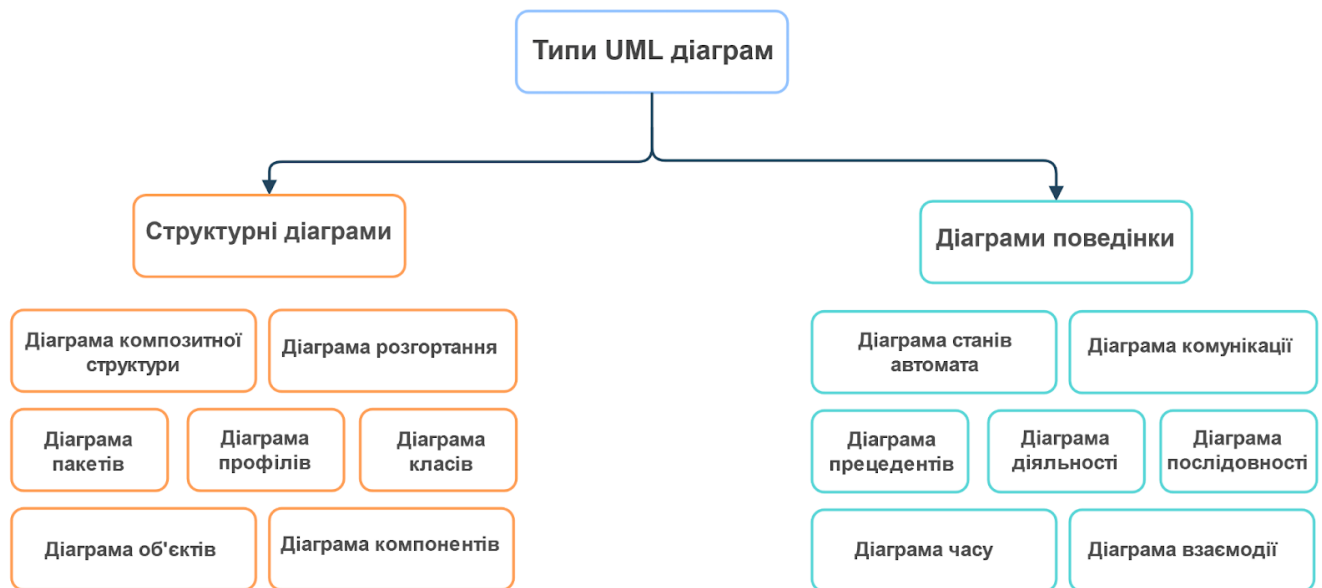


Рисунок 2.1 – Ієрархія діаграм UML [18]

2.3.1 Розроблення діаграми прецедентів для підсистеми автоматизації взаємодії з постачальниками

Діаграма варіантів використання або діаграма прецедентів або ж Use Case діаграма – це графічне представлення функціональних вимог системи, яке показує, що система повинна робити, але не те, як вона це робитиме.

Вона описує систему з точки зору її зовнішніх користувачів (акторів) та цілей, які вони можуть досягти за допомогою цієї системи. По суті, вона моделює межі системи та її взаємодію із зовнішнім світом. Діаграма прецедентів складається з таких елементів:

- актор, який є сутністю, що взаємодіє з системою, знаходячись ззовні неї;
- прецедент, що є функцією, яку виконує система, щоби принести користь акторові;
- межа системи, що є контейнером для прецедентів системи та вказує на фактичні межі системи, що моделюється.

У запропонованій підсистемі акторами є менеджер із закупівель та представник постачальника. Прецедентами є створення замовлення, підписання контракту, здійснення атомарної оплати, відмітка статусу доставки, скасування угоди, перегляд деталей контракту, оцінка ефективності угоди.

Розроблену діаграму прецедентів зображено на рисунку 2.2. Таблиці 2.1-2.2 містять опис елементів діаграми.



Рисунок 2.2 – Діаграма прецедентів для підсистеми автоматизації взаємодії з постачальниками

Таблиця 2.1 – Описи прецедентів

Назва прецеденту	Опис
Створення замовлення	Цей прецедент охоплює процес формування нової специфікації, вибору постачальника та ініціації розгортання смарт-контракту в блокчейн-мережі.
Підписання контракту	Прецедент забезпечує криптографічне підтвердження обома акторами (Менеджером із закупівель та Постачальником) умов угоди, фіксуючи незмінний статус контракту в блокчейні..

Продовження таблиці 2.1.

1	2
Здійснення атомарної оплати	Охоплює процес підтвердження виконання угоди компанією-замовником, що автоматично і миттєво призводить до переказу коштів постачальнику.
Відмітка статусу доставки	Прецедент дозволяє Постачальнику зафіксувати у блокчейні факт відправлення товару або його фактичної доставки, що запускає фінальний етап оплати.
Скасування угоди	Прецедент дозволяє одній зі сторін припинити дію контракту до етапу відправлення замовлення, знімаючи резервування коштів з рахунку компанії.
Перегляд деталей контракту	Прецедент охоплює процес відображення актуальної, верифікованої інформації про умови, статус, хеші транзакцій та часові мітки контракту з блокчейну.
Оцінка ефективності угоди	Прецедент дозволяє менеджерів переглянути показники ефективності постачальника, зокрема, точність дотримання термінів поставки, на основі даних, зафіксованих у блокчейні.

Таблиця 2.2 – Описи акторів

Назва актора	Опис
Менеджер із закупівель	Це особа, що відповідає за планування, ініціацію та фінансове завершення процесу придбання матеріалів для виробничого підприємства. Вона створює специфікації, ініціює розгортання смарт-контрактів, контролює дотримання

Продовження таблиці 2.2

1	2
	постачальниками термінів, здійснює атомарну оплату (Escrow) та проводить аналіз ефективності співпраці.
Представник постачальника	Це зовнішня діюча особа, яка взаємодіє з підсистемою від імені компанії-постачальника. Вона має унікальну блокчейн-ідентифікацію, підписує смарт-контракти, фіксує ключові етапи виконання замовлення (відправлення та доставка) та отримує оплату токенами після підтвердження виконання.

Центральним прецедентом розроблюваної SRM-підсистеми є Підписання контракту. Нижче наведено детальний опис прецеденту №2.

Прецедент 2. Підписання контракту.

Масштаб: підсистема управління взаємовідносинами з постачальником.

Рівень: задача рівня користувача/критична системна транзакція.

Головний актор: представник постачальника.

Зацікавлені особи та їхні вимоги:

– менеджер із закупівель. хоче отримати незмінне підтвердження згоди постачальника. Хоче, щоб необхідна сума для оплати була зарезервована до початку роботи;

– представник постачальника. Хоче, щоб умови угоди, ціна та терміни були незворотно зафіксовані. Хоче гарантію, що оплата технічно забезпечена через механізм Allowance;

– аудитори (керівники компанії-замовник, компанія постачальник). Хочуть мати незаперечний доказ факту укладання угоди для подальшого аудиту.

Передумови: замовлення створено, смарт-контракт розгорнуто, ініційований процес резервування токенів.

Результати: смарт-контракт переходить у стан підписаного. У блокчейні зафіксовано незмінну часову мітку підпису. Токени, необхідні для оплати, зарезервовані для використання цим контрактом.

Основний успішний сценарій:

а) менеджер із закупівель розгортає смарт-контракт, ініціюючи фінансову гарантію;

1) менеджер із закупівель скасовує угоду, знімаючи утримання коштів на рахунку компанії та завершуючи контракт;

б) представник постачальника отримує сповіщення про готовність угоди до підписання;

в) представник постачальника підписує контракт;

1) представник постачальника скасовує угоду, знімаючи утримання коштів на рахунку компанії та завершуючи контракт;

г) представник постачальника ставить відмітку, що замовлення в транзиті;

д) представник постачальника ставить відмітку, що замовлення доставлено;

е) менеджер із закупівель приймає замовлення, завершуючи його та ініціюючи переказ токенів на гаманець компанії-постачальника;

ж) менеджер із закупівель проводить оцінку діяльності постачальника.

Розширення (або альтернативні потоки):

а) відсутність Escrow:

1) смарт-контракт блокує підпис постачальника, якщо Менеджер із закупівель не надав достатньої кількості Allowance (токенів);

2) система повідомляє Менеджера про необхідність поповнити/підтвердити резерв;

б) помилка автентифікації:

1) система відхиляє спробу підпису, якщо криптографічний ключ постачальника не відповідає адресі, вказаній у смарт-контракті.

Частота використання: постійно, для кожного нового замовлення.

Відкриті питання: як завершити життєвий цикл контракту, якщо якась із сторін не дотрималася умов співпраці; як здійснюється узгодження умов співпраці під час життєвого контракту.

2.3.2 Розроблення діаграми компонентів для підсистеми автоматизації взаємодії з постачальниками

Діаграми компонентів показують структуру компонентів архітектури системи та те, як вони з'єднуються та взаємодіють між собою [19]. Цей вид діаграм надає високорівневе уявлення про архітектуру системи, показуючи, як система розділена на логічні або фізичні компоненти (наприклад, модулі, бібліотеки, виконавчі файли, бази даних). Діаграма компонентів частіше за все складається з таких елементів:

– компонент, що є замінюваним та автономним блоком системи, який виконує певний набір функцій і взаємодіє з іншими частинами інтерфейсу через точки контакту;

– інтерфейс, що є послугою, яку компонент надає іншим компонентам;

– зв'язок, що може бути залежністю одного компонента від іншого, чи з'єднанням – тобто зв'язком, що об'єднує інтерфейс одного компонента з інтерфейсом іншого, ілюструючи зв'язок типу «клієнт-сервер».

Розроблену для підсистеми автоматизації взаємодії з постачальниками зображено на рисунку 2.3.

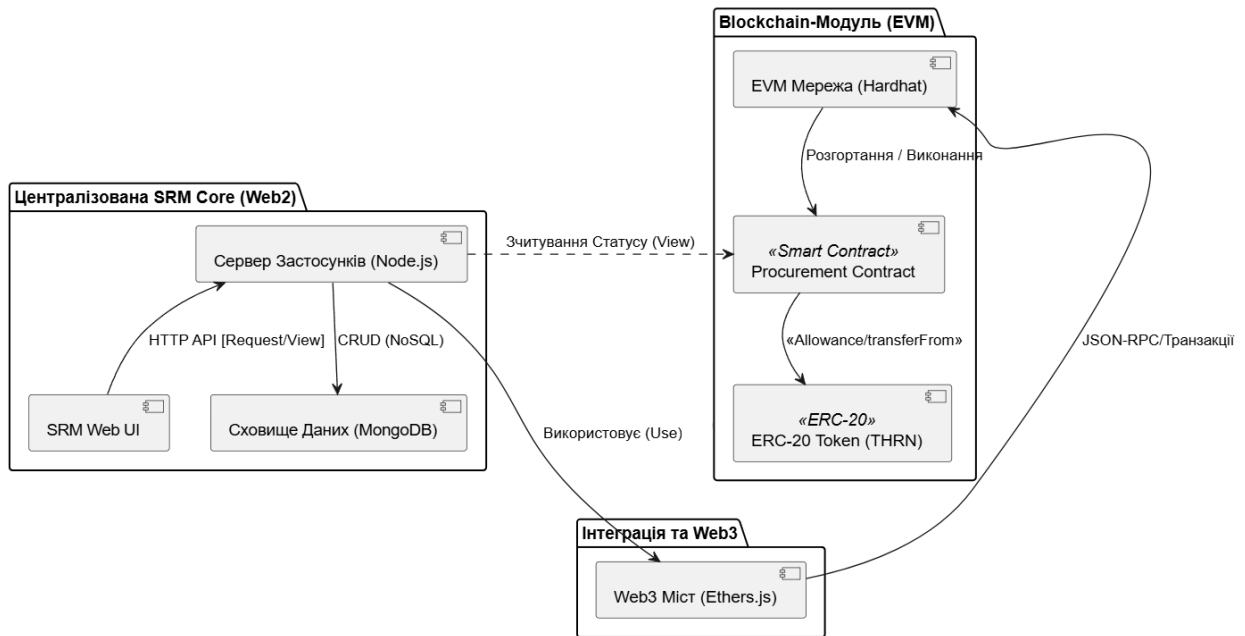


Рисунок 2.3 – Діаграма компонентів для підсистеми автоматизації взаємодії з постачальниками

Дана діаграма відображає гібридну архітектуру розроблюваної SRM-системи, яка поєднує в собі централізований рівень для управління даними з децентралізованим транзакційним рівнем для виконання смарт-контрактів. Архітектура поділена на три функціональні області: ядро централізованої SRM, інтеграційний прошарок (міст між SRM системою та blockchain-модулем) та blockchain-модуль Ethereum (EVM, Ethereum Virtual Machine).

Централізований рівень відповідає за:

- клієнтську частину (SRM Web UI), яка ініціює всі запити через HTTP API;
- сервер застосунків (Node.js), що керує автентифікацією, бізнес-логікою та зберігає некритичні дані. Залежить від сховища даних (MongoDB) для операцій читання/запису (CRUD);

– сховище даних. Зберігає всі некритичні дані, такі як базова інформація про постачальників, деталі замовлень, хеші завершених транзакцій та дані для UI.

Інтеграційний прошарок відповідає за:

– Web3 Міст (Ethers.js) для зв'язку Web2-середовища та Web3-технологіями. Є бібліотекою, яку використовує сервер застосунків для підписання та відправки транзакцій до блокчейн-мережі, а також для зчитування поточного статусу смарт-контракту.

Децентралізований транзакційний рівень відповідає за:

– EVM мережу (Hardhat), що є тестовим середовищем, яке емулює віртуальну машину Ethereum. Забезпечує виконання смарт-контрактів та ведення розподіленого, незмінного реєстру;

– смарт-контракт, який є ядром Escrow-логіки. Цей компонент містить бізнес-логіку життєвого циклу контракту та контролює механізм оплати;

– ERC-20 Token, що є компонентом цифрового активу, який використовується для оплати.

Ключова взаємодія, що демонструє атомарність, відбувається між сервером застосунків, Web3 мостом та EVM-мережею для ініціації транзакції, та між смарт-контрактом і ERC-20 Token для списання коштів.

2.3.3 Розроблення діаграм послідовностей

Діаграма послідовності показує, як набір об'єктів взаємодіє в процесі з плином часу. У ньому відображаються повідомлення, які передають учасникам і об'єктам у системі, а також порядок їх виникнення [20]. Діаграма послідовності складається з таких елементів:

– лінії життя, вертикальні пунктирні лінії, які представляють окремі учасники взаємодії (об'єкти, актори, компоненти тощо). Вони показують, що об'єкт існує протягом певного періоду часу в контексті послідовності;

- повідомлення, горизонтальні стрілки, що йдуть від однієї лінії життя до іншої й представляють комунікацію між об'єктами;
- фрагменти активації, прямокутники, накладені на лінію життя, що позначають період часу, протягом якого об'єкт виконує операцію;
- комбіновані фрагменти, області, що дозволяють моделювати альтернативну логіку керування та алгоритми.

Нижче наведено діаграми послідовностей для опису повного життєвого циклу закупівлі матеріалів у розроблювальній підсистемі SRM (рис. 2.4-2.8).

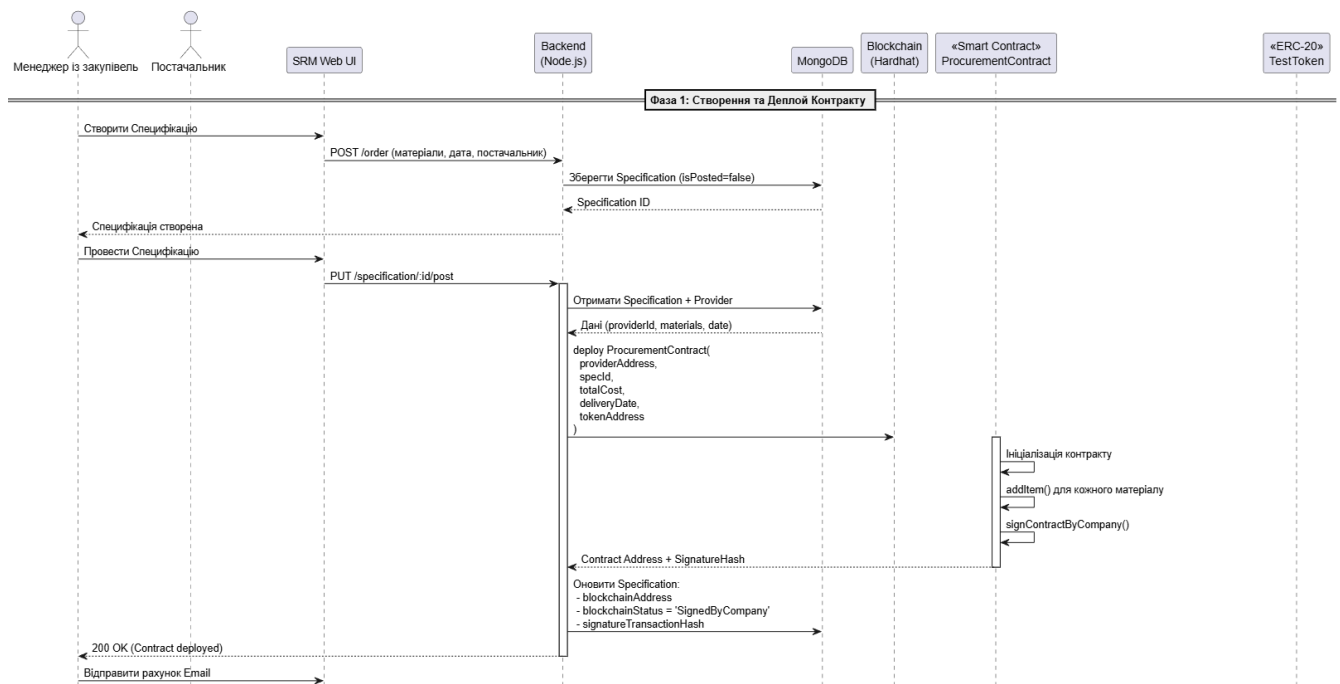


Рисунок 2.4 – Фаза 1: створення та деплой контракту

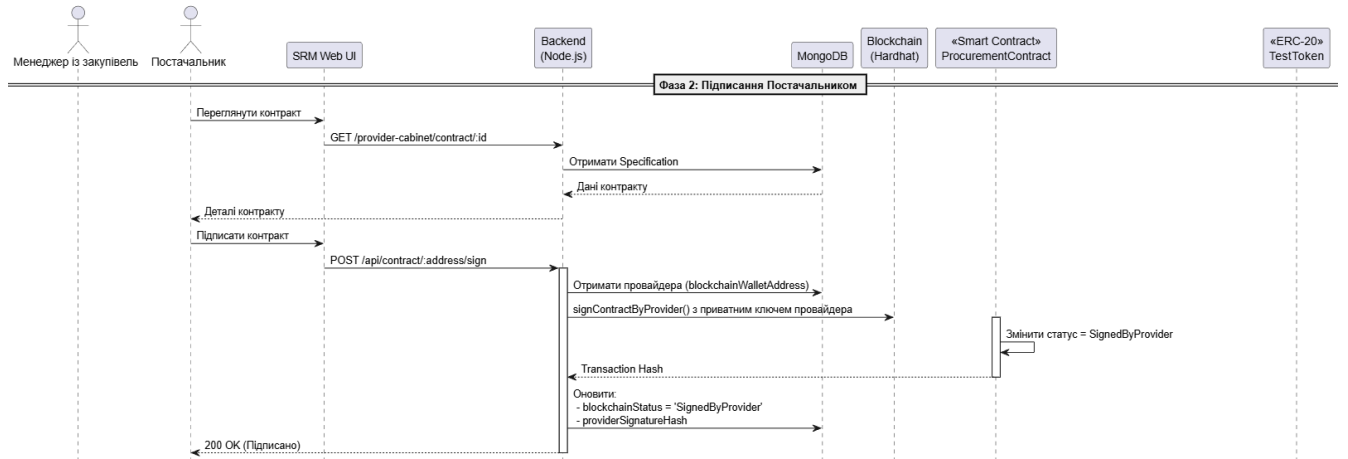


Рисунок 2.5 – Фаза 2: підписання контракту постачальником

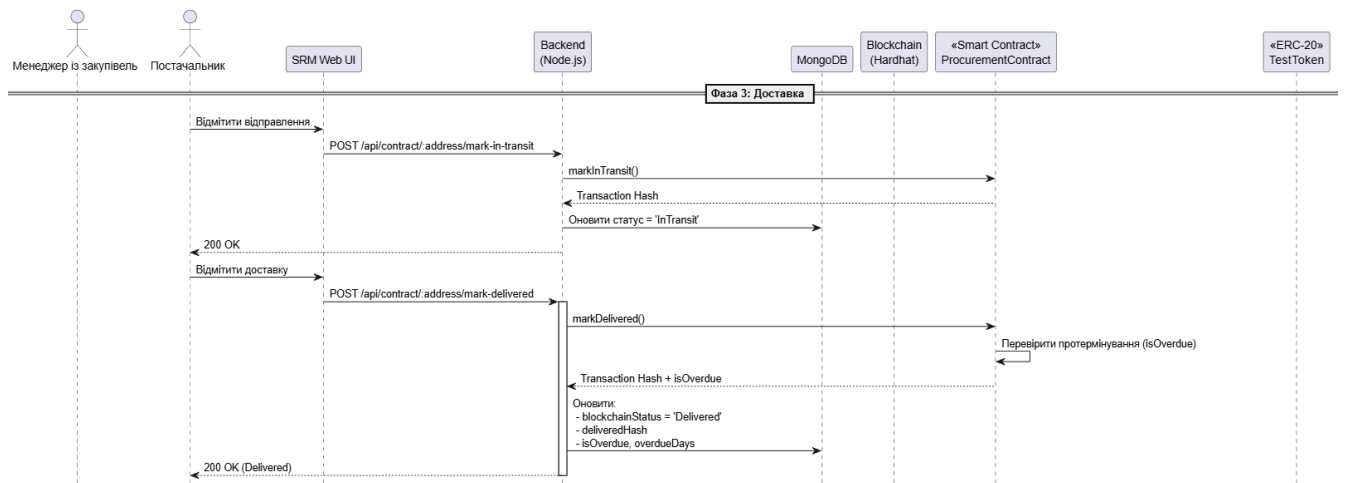


Рисунок 2.6 – Фаза 3: доставка

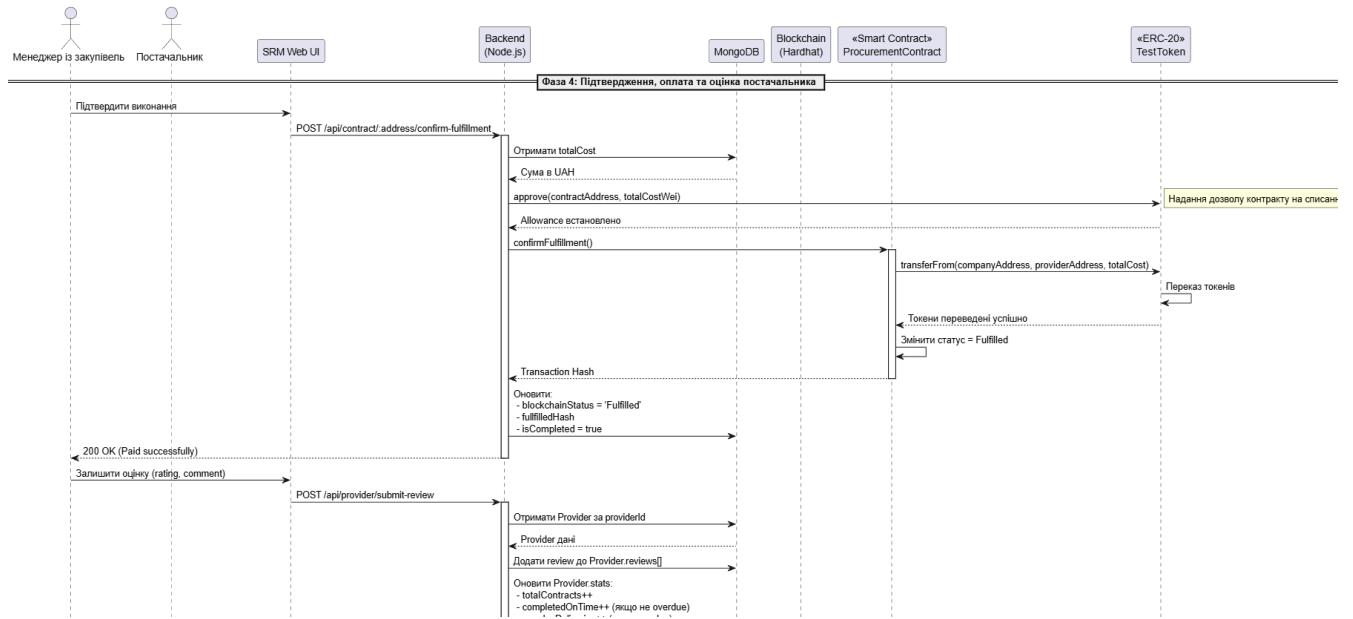


Рисунок 2.7 – Фаза 4: підтвердження, оплата та оцінка постачальника

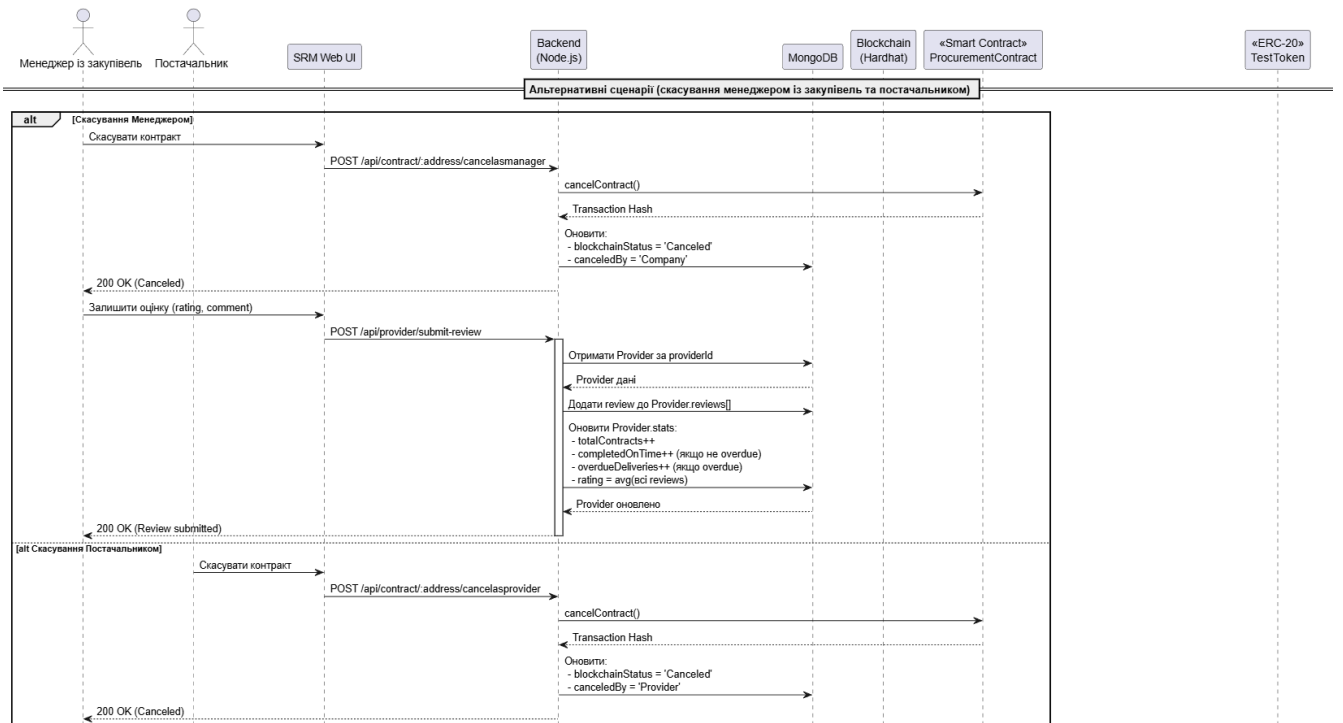


Рисунок 2.8 – Альтернативні сценарії (скасування менеджером із закупівель та постачальником)

2.4 Охорона праці

SRM-підсистема потребує серверного обладнання для належного функціонування. Серверне обладнання живиться від мережі змінного струму напругою 220 В, що створює потенційну небезпеку ураження електричним струмом.

Згідно з НПАОП 40.1-1.21-98 "Правила безпечної експлуатації електроустановок споживачів" [21], приміщення серверної відноситься до приміщень без підвищеної небезпеки, оскільки характеризується нормальною температурою повітря, відносною вологістю не більше 60% та відсутністю струмопровідного пилу.

Згідно з ПУЕ [22], основними чинниками ураження електричним струмом є величина струму, що проходить через тіло людини, та час його дії. Небезпечним вважається змінний струм силою приблизно 10 мА, смертельним – приблизно 100 мА.

Для оцінки потенційної небезпеки використовується формула струму через тіло людини при непрямому дотику:

$$I_{\text{л}} = \frac{U}{R_{\text{л}}}, \quad (2.1)$$

де $I_{\text{л}}$ – струм через тіло людини, А;

U – напруга дотику, В;

$R_{\text{л}}$ – опір тіла людини, Ом (приймається 1000 Ом).

Якщо розрахований струм перевищує 10 мА, необхідні додаткові засоби захисту.

Основним засобом захисту від ураження електричним струмом є захисне заземлення – навмисне електричне з'єднання з заземлювальним пристроєм металевих неструмопровідних частин, які можуть опинитися під напругою.

Виникає необхідність перевірити відповідність системи електробезпеки вимогам з електрозахисту приміщень [23].

Згідно з ДСТУ EN 62305-3:2021 «Блискавкозахист. Частина 3. Фізичні пошкодження будівель (споруд) та небезпека для життя» [24], опір заземлювального пристрою для електроустановок напругою до 1000 В не повинен перевищувати 4 Ом.

Для визначення опору одиночного вертикального заземлювача (стержня) використовується формула:

$$R_B = \frac{\rho}{2\pi l} \left[\ln \frac{2l}{d} + \frac{1}{2} \ln \frac{4t+l}{4t-l} \right], \quad (2.2)$$

де R_B – опір одиночного вертикального заземлювача, Ом;

ρ – питомий опір ґрунту, Ом·м;

l – довжина заземлювача, м;

d – діаметр заземлювача, м;

t – відстань від поверхні землі до середини заземлювача, м.

Якщо опір одного заземлювача перевищує допустиме значення, розраховується необхідна кількість заземлювачів за формулою:

$$n = \frac{R_B}{R_z \cdot \eta_B}, \quad (2.3)$$

де n – кількість вертикальних заземлювачів;

R_3 – нормований опір заземлювального пристрою, Ом;

η_B – коефіцієнт використання заземлювачів.

Горизонтальний заземлювач (сполучна смуга) з'єднує вертикальні заземлювачі. Його опір розраховується за формулою:

$$R_\Gamma = \frac{\rho}{2\pi L_\Gamma} \ln \frac{2L_\Gamma^2}{bt}, \quad (2.4)$$

де L_Γ – довжина горизонтального заземлювача, м;

b – ширина смуги, м;

t – глибина закладання, м.

Загальний опір заземлювального пристрою з паралельно з'єднаних заземлювачів:

$$R_{\text{заг}} = \frac{R_B \cdot R_\Gamma}{R_B \cdot \eta_\Gamma + R_\Gamma \cdot \eta_B}, \quad (2.5)$$

де η_Γ – коефіцієнт використання горизонтального заземлювача.

Для електроустановок напругою до 1000 В із глухозаземленою нейтраллю застосовується захисне зануління – з'єднання металевих частин обладнання з нульовим захисним провідником.

Ефективність зануління перевіряється за умовою:

$$I_{\text{кз}} \geq k \cdot I_{\text{ном}}, \quad (2.6)$$

де $I_{кз}$ – струм однофазного короткого замикання, А;

$I_{ном}$ – номінальний струм спрацювання захисту, А;

k – коефіцієнт кратності.

Отримане значення за формулою (2.6) гарантує, що при замиканні фази на корпус обладнання виникне достатньо великий струм короткого замикання, який спричинить швидке спрацювання захисних апаратів.

Струм короткого замикання визначається за формулою:

$$I_{кз} = \frac{U_{\phi}}{Z_{пет}}, \quad (2.7)$$

де U_{ϕ} – фазна напруга мережі, В;

$Z_{пет}$ – повний опір петлі фаза-нуль, Ом.

Отже, чим менший опір проводів та трансформатора, тим більший струм і швидше спрацює захист. Опір петлі фаза-нуль складається з опорів:

$$Z_{пет} = Z_{т} + Z_{\phi} + Z_{н}, \quad (2.8)$$

де $Z_{т}$ – опір обмотки трансформатора, Ом;

Z_{ϕ} – опір фазного проводу, Ом;

$Z_{н}$ – опір нульового проводу, Ом.

Формула 2.8 дозволяє розрахувати повний опір ланцюга КЗ та вибрати необхідний переріз проводів і номінал захисних апаратів. Опір проводу визначається за формулою:

$$R = \frac{\rho \cdot l}{S}, \quad (2.9)$$

де ρ – питомий опір матеріалу проводу, Ом·мм²/м;

l – довжина проводу, м;

S – переріз проводу, мм².

Таким чином, зі збільшенням довжини лінії та зменшенням перерізу проводів опір зростає, а струм КЗ зменшується. Тому для віддалених споживачів потрібні проводи більшого перерізу або додаткові заходи захисту.

3 РЕАЛІЗАЦІЯ ПІДСИСТЕМИ АВТОМАТИЗАЦІЇ У ВИГЛЯДІ ПРОГРАМНОГО ЗАСОБУ

3.1 Обґрунтування вибору технологій для розроблення SRM-підсистеми

В основі задачі розроблення гібридної SRM-підсистеми лежить використання сучасного технологічного стеку, який забезпечуватиме швидкість розробки, гнучкість вебдодатка та можливість надійної взаємодії з технологією blockchain.

Для розроблення серверної та клієнтської частини було обрано JavaScript-орієнтований стек через універсальність та високу продуктивність доступних інструментів в асинхронних операціях.

JavaScript (JS) – це високорівнева мультипарадигменна кросплатформенна скриптова мова програмування, яка широко використовується для створення інтерактивних застосунків і сайтів [25]. З боку фронтенду вона дозволяє реалізовувати просунуті анімації, впроваджувати складені інтерактивні форми, маніпулювати контентом поза можливостями базових HTML та CSS тощо. Завдяки таким платформам, як Node.js, на JavaScript стає можливим створення й бекенд частин проєктів та мобільних додатків без необхідності опанування нового синтаксису. В контексті поточного проєкту JavaScript було обрано як єдину мову для всієї гібридної підсистеми (за винятком розгортання смарт-контрактів).

Нижче представлено опис технологій централізованого Off-Chain стеку.

Node.js являє собою кросплатформенне середовище для виконання JavaScript-коду, що дозволяє запускати JS-код поза браузером, на сервері, для створення веб-застосунків, написання API тощо [26]. Підтримує технологію npm – масштабну екосистему пакетів та бібліотек для швидкого завантаження та імплементації допоміжних інструментів і технологій, що дозволяють масштабувати стандартний

Node.js функціонал та полегшити розробку. Для створення SRM-підсистеми Node.js необхідний у першу чергу як базова платформа для розробки бекенд-сервера та API-інтерфейсів.

Express.js – це веб-фреймворк для маршрутизації для середовища Node.js, що спрощує розробку вебдодатків та API, надаючи необхідні інструменти для роутингу та обробки HTTP-запитів [27]. Є базовою технологією при використанні Node.js.

Handlebars є шаблонізатором для генерації динамічного HTML-контенту на боці сервера [28]. Завдяки цій технології можлива генерація HTML-інтерфейсів для особистих кабінетів ключових акторів.

MongoDB – нереляційна (NoSQL) документоорієнтована СУБД. Зберігає дані у форматі, подібному до JSON (BSON), забезпечуючи гнучкість схеми та горизонтальне масштабування [29]. В проєкті необхідна в якості сховища для Off-Chain даних (користувачі, статуси контрактів, хеші транзакцій тощо).

Далі наведено опис технологій децентралізованого On-Chain стеку.

Solidity – об'єктно-орієнтована мова програмування, розроблена спеціально для написання смарт-контрактів, які виконуються на Ethereum Virtual Machine, яка, в свою чергу, працює на основі блокчейн-платформи Ethereum [30]. В проєкті необхідна для створення логіки смарт-контракту з Escrow.

ERC-20 Token – стандарт для взаємозамінних токенів у мережі Ethereum. Визначає набір функцій (як-от approve та transferFrom), критичних для фінансових гарантій та атомарної оплати.

Hardhat – це середовище розробки для Ethereum, яке надає інструменти для компіляції, налагодження та створення локального блокчейн-середовища.

Ethers.js є JavaScript-бібліотекою, що виступає в ролі мосту між серверним Node.js додатком та блокчейн-мережею, дозволяючи Web-додаткам викликати функції смарт-контракту [31].

3.2 Розроблення алгоритмів поведінки акторів під час взаємодії з підсистемою

Розроблення алгоритмів поведінки є необхідним для забезпечення детермінованості системи. Це дозволяє мінімізувати ризик виникнення логічних помилок та забезпечити передбачувану реакцію інтерфейсу на будь-які дії користувача, включно з обробкою виняткових ситуацій.

3.2.1 Розроблення алгоритму поведінки менеджера з закупівель

Алгоритм поведінки менеджера з закупівель наведено на рисунку 3.1.

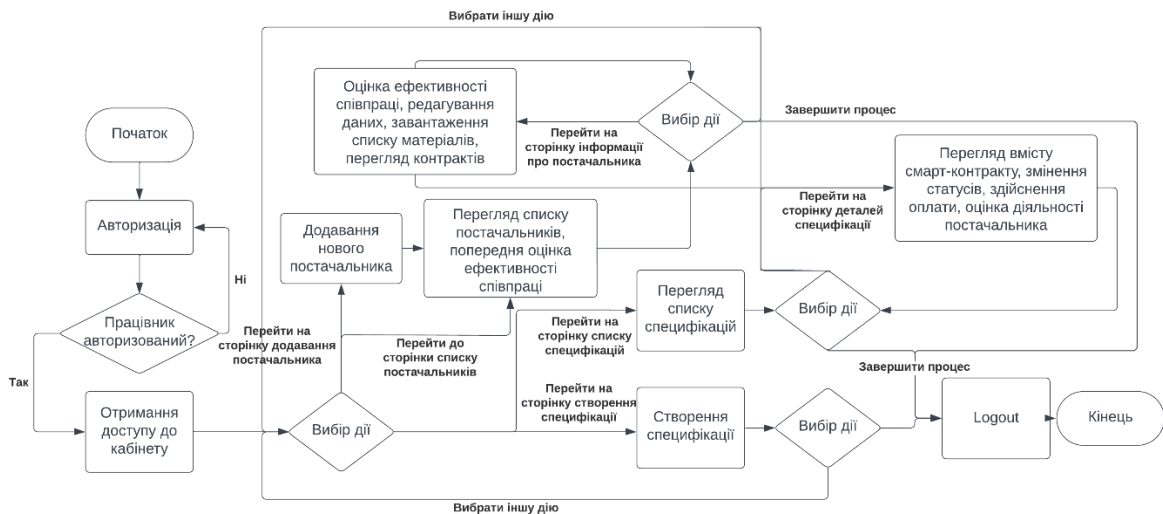


Рисунок 3.1 – Схема поведінкового алгоритму «Менеджер із закупівель»

Обґрунтування елементів схеми:

– блоки «Авторизація» та «Отримання доступу до кабінету». Працівник проходить процедуру авторизації, використовуючи особисті дані в вигляді адреси корпоративної поштової скриньки та пароля, що були попередньо створені для

нього адміністраторами ERP-системи. За успішної автентифікації працівник буде спрямований до особистого кабінету, що містить SRM-функціонал, та буде вільний в виборі доступних дій;

– блок «Додавання нового постачальника». Функціонал цієї сторінки передбачає додавання нового постачальника до бази даних за допомогою введення відомих реквізитів постачальника. За допомогою цих реквізитів підсистема налаштовує зв'язок між кабінетами менеджера з закупівель та постачальника. Оскільки проєкт, що розроблюється, є прототипом, для зручності його створення адреса публічного криптогаманця постачальника на цьому етапі присвоюється автоматично, використовуючи надані мок-дані фреймворком Hardhat для тестування мережі в локальному режимі. В інших випадках для коректної роботи механізму оплати унікальний публічний криптогаманець постачальника має бути привласнений вручну;

– блок «Перегляд списку постачальників, попередня оцінка ефективності співпраці». Сторінка пропонує презентацію анкет постачальників в вигляді карток для попереднього перегляду базової інформації та показників ефективності співпраці. Кожна картка веде до сторінки детальної інформації про постачальника;

– блок «Оцінка ефективності співпраці, редагування даних, завантаження списку матеріалів, перегляд контрактів». Сторінка являє собою детальну інформацію про постачальника, що включає розширену інформацію про показники ефективності співпраці, можливість редагування та видалення інформації про постачальника, можливість додавати та оновлювати попередньо отримані від постачальника списки пропонованих матеріалів, можливість переглядати історію контрактів з переходом до детальної інформації про них;

– блок «Створення специфікації». Інтерфейс надає можливість створення специфікації замовлення з указанням побажань щодо замовлення;

– блок «Перегляд списку специфікацій». Сторінка пропонує попередній перегляд створених специфікацій та активних контрактів з можливістю переходу до сторінки деталей;

– блок «Перегляд вмісту смарт-контракту, змінення статусів, здійснення оплати, оцінка діяльності постачальника». На сторінці менеджер із закупівель може ознайомитися з деталями специфікації чи смарт-контракту та здійснити Off- та On-chain зміни;

– блок «Logout». Завершення поточної сесії користувача та повернення на сторінку входу.

3.2.2 Розроблення алгоритму поведінки представника постачальника

Алгоритм поведінки представника постачальника представлений на рисунку 3.2.

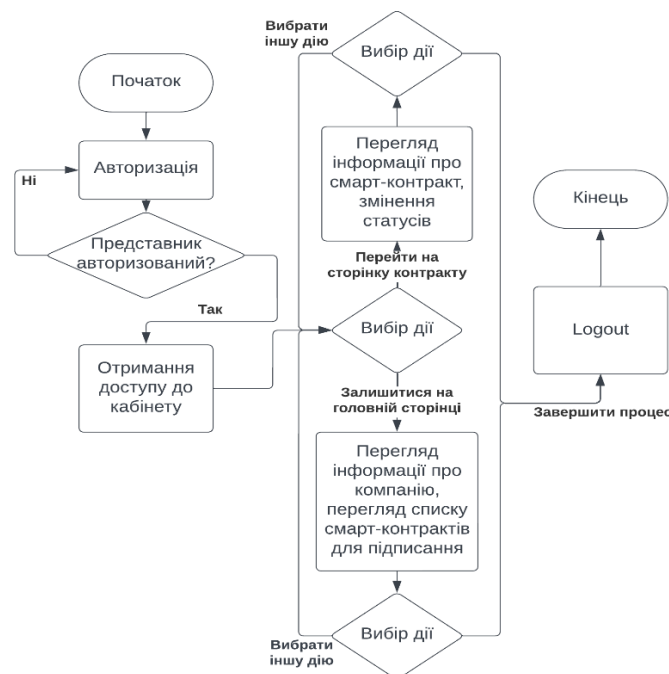


Рисунок 3.2 – Схема поведінкового алгоритму «Представник постачальника»

Обґрунтування елементів схеми:

– блок «Перегляд інформації про компанію, перегляд списку смарт-контрактів для підписання». Сторінка, на яку представник постачальника потрапляє після процедури авторизації. Вона дозволяє переглянути в списку смарт-контрактів інформацію про кожен у стислому вигляді та перейти до її деталей. Також на сторінці відображається публічна інформація про постачальника;

– блок «Перегляд інформації про смарт-контракт, змінення статусів». Інтерфейс дозволяє ознайомитися з деталями смарт-контракту й здійснити On-Chain зміни.

3.3 Розроблення логічної структури даних

Логічна структура даних – це рівень абстракції в проєктуванні бази даних, що описує організацію даних у термінах обраної моделі даних, незалежно від фізичного способу їх зберігання та технічних характеристик апаратного забезпечення. Вона визначає сукупність сутностей, їхніх атрибутів, типів даних та взаємозв'язків між ними, забезпечуючи семантичну цілісність підсистеми.

Як було зазначено в п.п. 2.1.1, логічна структура бази даних розроблюваної SRM-підсистеми реалізована з використанням документо-орієнтованої СУБД MongoDB. Вибір нереляційної моделі даних зумовлений простотою використання, необхідністю гнучкого зберігання різномірних даних про контракти, а також потребою в швидкому масштабуванні та інтеграції з даними блокчейн-мережі. Отже, база даних складається з чотирьох основних колекцій (Collections), взаємозв'язки між якими реалізовані на логічному рівні через посилання на унікальні ідентифікатори (`_id` та `providerId`). Графічне представлення структури наведено на рисунку 3.3.

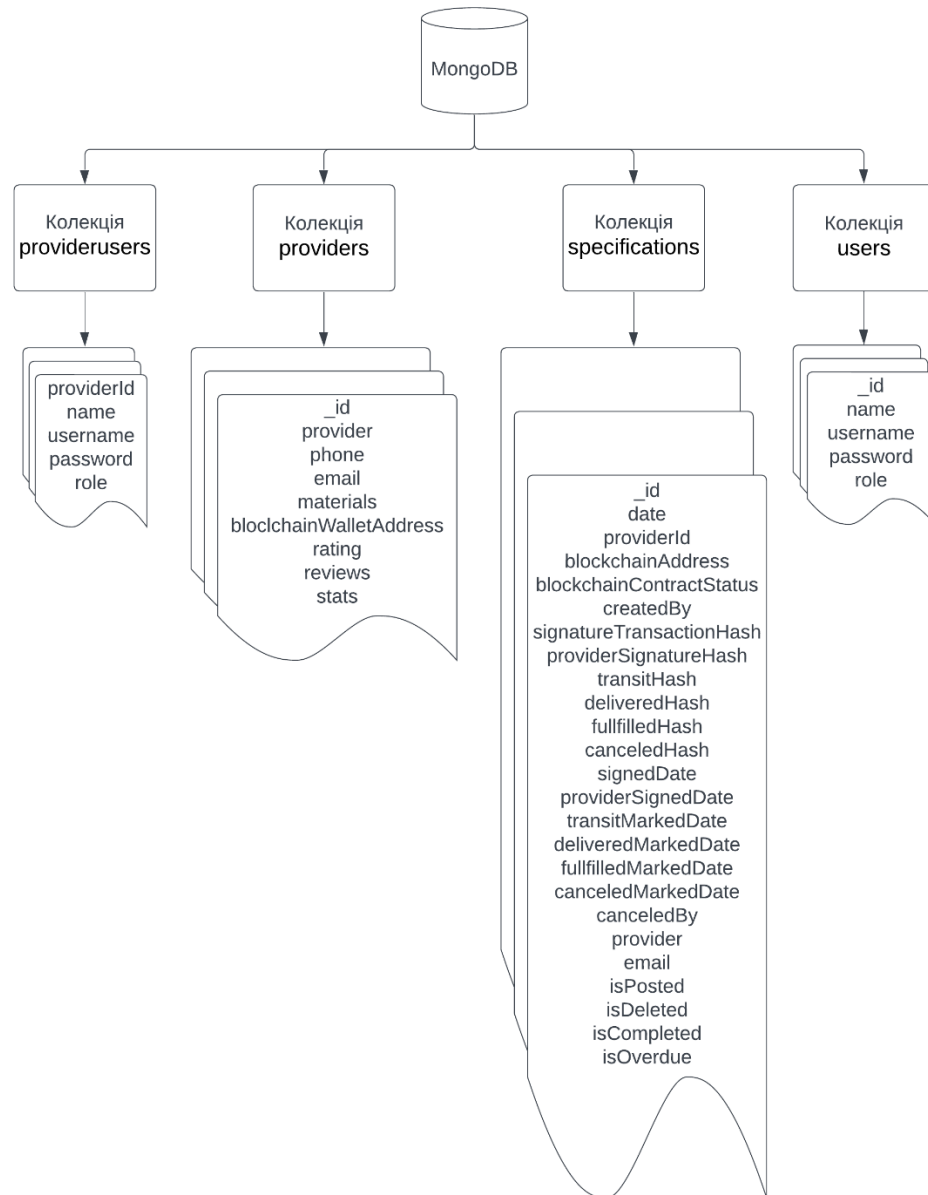


Рисунок 3.3 – Структура бази даних SRM-підсистеми

Коллекція users зберігає облікові дані внутрішніх співробітників компанії-замовника, зокрема – менеджера з закупівель (рис. 3.4). Призначена для автентифікації та авторизації менеджерів із закупівель з використанням ключових атрибутів.

```

_id: ObjectId('66537f04902fcf4fdfe8ff7...
username : "planner@gmail.com"
password : "planner"
role : "production_planner"
name : "Данилов Даниїл Данилович"

```

Рисунок 3.4 – Приклад екземпляру колекції users у MongoDB Atlas

Колекція providers містить юридичну та контактну інформацію про контрагентів (постачальників) (рис. 3.5).

```

_id: ObjectId('68e6cfad150f625fdb94d3ee')
provider : "Dunno"
phone : "0670777095"
email : "dunno@dunno.ua"
blockchainWalletAddress : "0x70997970C51812dc3A010C7d01b50e0d17dc79C8"
▸ materials : Array (5)
  __v : 20
  rating : 3.8333333333333335
▸ reviews : Array (6)
▸ stats : Object

```

Рисунок 3.5 – Приклад екземпляру колекції providers у MongoDB Atlas

Є критично важливою для інтеграції з блокчейном, оскільки зберігає адресу унікального криптогаманця. Ключовими атрибутами є: blockchainWalletAddress – публічна адреса гаманця Ethereum для взаємодії зі смарт-контрактами; rating, reviews, stats – дані для розрахунку KPI та репутації постачальника; materials – перелік номенклатури, яку постачає компанія.

Колекція providerusers містить облікові дані представників постачальників. Ключові атрибути: providerId (посилання на батьківську компанію з колекції providers), облікові дані username та password (рис. 3.6).

```

_id: ObjectId('68e6c9e47aaddf7231e70620')
name: "Абдул Альхазред"
username: "alhazred@dunno.ua"
password: "dunno"
role: "provider"
providerId: "68e6cfad150f625fdb94d3ee"

```

Рисунок 3.6 – Приклад екземпляру колекції providerusers у MongoDB Atlas

Колекція specifications є центральною сутністю підсистеми, яка відображає повний життєвий цикл закупівлі – від створення заявки до закриття смарт-контракту в блокчейні (рис. 3.7).

```

_id: ObjectId('6935aa4a3ac41eb946fc81b5')
date: "31.12.2025"
providerId: ObjectId('68e6cfad150f625fdb94d3ee')
blockchainAddress: "0x959922bE3CAee4b8Cd9a407cc3ac1C251C2007B1"
blockchainContractStatus: "Fulfilled"
createdBy: Object
signatureTransactionHash: "0x4bec8072f82920926596f1de635d1f154168a76fb116f55c4d29fd4d16d1c999"
providerSignatureHash: "0x41e02b5fd617b3e29d9dd8ac95d49de3eb8786f30a4ff6a4e9cb5273689b82f5"
transitHash: "0x7c06edc8d1fabe7acbdbbdeef19fd5952d0666ccdd139819e5a2e94807c00614"
deliveredHash: "0xfeacce40235a3f6e0c2744e8637fe010dc708981626cebe51841808c1a398f20"
fulfilledHash: "0x849074df984d435b9145414e8fd6831511db1b9965a70260e2c2771a024fe304"
canceledHash: null
signedDate: 2025-12-07T16:25:36.134+00:00
providerSignedDate: 2025-12-07T17:00:53.904+00:00
transitMarkedDate: 2025-12-07T17:00:57.335+00:00
deliveredMarkedDate: 2025-12-07T17:01:03.289+00:00
fulfilledMarkedDate: 2025-12-07T17:02:00.970+00:00
canceledMarkedDate: null
canceledBy: null
provider: "Dunno"
email: "dunno@dunno.ua"
isPosted: true
isDeleted: false
isCompleted: true
isOverdue: false

```

Рисунок 3.7 – Приклад екземпляру колекції specifications у MongoDB Atlas

Ключовими атрибутами є: providerId – логічний зв'язок із постачальником; blockchainAddress – адреса розгорнутого смарт-контракту в мережі Ethereum; blockchainContractStatus – поточний стан угоди; атрибути аудиту транзакцій – група полів для зберігання хешів транзакцій, що підтверджують юридично значущі дії (signatureTransactionHash, providerSignatureHash, transitHash, deliveredHash,

fulfilledHash, canceledHash); часові мітки – поля для фіксації часу подій (signedDate, deliveredMarkedDate тощо).

3.4 Опис механізму роботи blockchain-модулю та його інтеграція з backend-сторonoю проєкту

Децентралізований рівень SRM-підсистеми базується на платформі Ethereum Virtual Machine (EVM) і використовує два ключові смарт-контракти:

- TestToken.sol (ERC-20), що є цифровим активом (THRN) і який використовується для забезпечення фінансової гарантії (Escrow) та здійснення атомарної оплати;

- ProcurementContract.sol, що є центральним елементом блокчейн-проєкту. Він фіксує всі умови контракту (вартість, матеріали, учасники) та керує його життєвим циклом (від Created до Fulfilled). Контракт використовує перерахування (enum State) для незмінного відстеження прогресу контракту: SignedByCompany, SignedByProvider, Delivered, Fulfilled та Canceled. Критична функція confirmFulfillment() викликає token.transferFrom(companySigner, providerSigner, totalCostWei), що гарантує, що оплата відбувається лише якщо статус контракту є Delivered і лише якщо компанія-замовник раніше надала дозвіл (Allowance).

Інтеграція blockchain-модуля з backend-сторonoю реалізована за допомогою Ethers.js, який виступає в ролі мосту між централізованим сервером Node.js та EVM-блокчейном.

Життєвий цикл смарт-контракту зображений на рисунку 3.8.

```

enum State {
    Created, // 0: Контракт створено та розгорнуто
    SignedByCompany, // 1: Компанія підписала контракт
    SignedByProvider, // 2: Постачальник підписав контракт
    InTransit, // 3: Товар відправлено
    Delivered, // 4: Постачальник відмітив доставку
    Fulfilled, // 5: Компанія підтвердила виконання (відбувається оплата)
    Canceled // 6: Контракт скасовано
}

```

Рисунок 3.8 – Життєвий цикл смарт-контракту

Взаємодія з блокчейном у розроблюваному проєкті розділена на два чіткі, послідовні етапи: ініціалізація (виконується автономно) та керування життєвим циклом (виконується основним backend-сервером).

Ініціалізація є передумовою для функціонування SRM-системи. Вона виконується в окремому терміналі і не є частиною постійної роботи backend-сервера. Вона включає: виконання команди `prx hardhat node`, що створює локальну тестову EVM-мережу (<http://127.0.0.1:8545>) та ініціалізує тестові гаманці з ETH-балансами, а також виконання команди `node deploy.js` – скрипта для Ethers.js, що виконує транзакції для створення незмінних ресурсів. Серед цих ресурсів: розгортання `TestToken.sol`; розгортання `ProcurementContract.sol` (передаючи в конструктор адресу постачальника, загальну вартість та адресу токена); ініціалізація фінансової гарантії Escrow шляхом виклику функції `token.approve(procContractAddress, allowanceAmount)`, що надає новому смарт-контракту дозвіл на списання необхідної суми токенів від імені компанії; підпис компанією шляхом виклику `contract.signContractByCompany()` для встановлення статусу контракту `SignedByCompany`. Після цих дій адреса розгорнутого контракту (`procContractAddress`) та хеш початкової транзакції повертаються до основного Backend-сервера та зберігаються в MongoDB (поле `blockchainAddress`).

Всі подальші транзакційні дії являють собою основний життєвий цикл контракту й керуються основним backend-сервером через API-маршрути за допомогою інтегрованої в Express.js бібліотеки Ethers.js.

Фінальною, найкритичнішою операцією є атомарна оплата під час фази Fulfilled. Менеджер із закупівель, знаходячись на сторінці деталей контракту, натискає кнопку підтвердження виконання замовлення. Node.js відправляє транзакцію з гарантії компанії, викликаючи: `contractInstance.confirmFulfillment()`. Смарт-контракт перевіряє, чи статус Delivered. Якщо так, він виконує рядки з рисунку 3.9, здійснюючи тим самим переказ токенів з рахунку компанії-замовника на електронний гаманець постачальника.

```
bool success = token.transferFrom(companySigner, providerSigner, totalCostWei);  
require(success, "PC: Token transfer failed. Check Company allowance or balance.");
```

Рисунок 3.9 – Функціонал здійснення атомарної оплати

Функція `transferFrom` атомарно виконує дві дії: переказ токенів та оновлення статусу контракту на Fulfilled. Якщо переказ токенів неможливий (наприклад, недостатньо Allowance або балансу), транзакція відкочується, а статус контракту (та Escrow) залишається без змін. Це усуває ризик неотримання оплати.

Приклад backend-логіки роутеру `confirm-filfillment`, що поєднує взаємодію з blockchain-модулем та базою даних наведено на рисунку 3.10.

```

// 1. ПІДГОТОВКА КОНТРАКТУ
const procurementArtifact = loadContractArtifact('ProcurementContract');

const contractInstance = new ethers.Contract(
  contractAddress, // Адреса контракту, що виконується
  procurementArtifact.abi, // Інтерфейс (що можна викликати)
  deployerWallet // Гаманець, що підписує транзакцію (компанія)
);

// 2. ВИКЛИК confirmFulfillment()
// Ця функція у Solidity запускає переказ токенів (token.transferFrom)
const tx = await contractInstance.confirmFulfillment({ nonce: nonce++ });

// 3. ОЧІКУВАННЯ ПІДТВЕРДЖЕННЯ БЛОКЧЕЙНУ
const receipt = await tx.wait();

// 4. СИНХРОНІЗАЦІЯ З MongoDB
await Specification.findByIdAndUpdate(specId, {
  blockchainContractStatus: 'Fulfilled',
  fullfilledHash: tx.hash,
  isCompleted: true
});

```

Рисунок 3.10 – Логіка ініціації атомарної оплати

3.5 Розроблення користувацьких інтерфейсів

Цей пункт демонструє кінцеві результати розробки, акцентуючи увагу на дизайні та функціональності користувацьких інтерфейсів (UI), які є точкою взаємодії між акторами й гібридною архітектурою системи.

UI актора «Менеджер із замовлень» представлено на рисунках 3.11-3.20.

Рисунок 3.11 – Сторінка створення екземпляра постачальника

Рисунок 3.11 демонструє можливість додавання нового постачальника до бази даних шляхом заповнення форми.

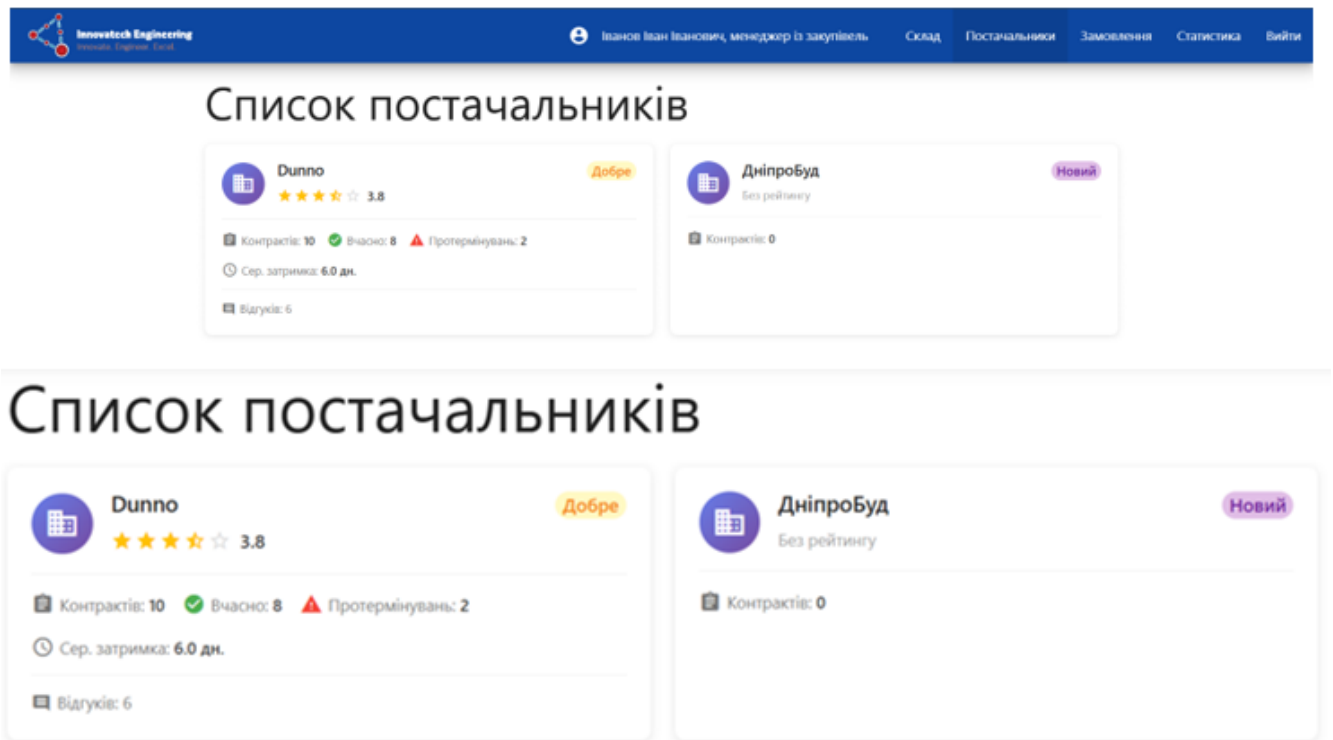


Рисунок 3.12 – Сторінка списку постачальників

Рисунок 3.12 демонструє список постачальників у вигляді карток з можливістю попереднього перегляду показників ефективності.

Innovatich Engineering
Innovate. Engage. Grow.

Іванов Іван Іванович, менеджер із закупівель Склад Постачальники Замовлення Статистика Вийти

Dunno ★ 3.8

ID: 68e6cfad150f625fdb94d3ee
 Телефон: 0670777095
 Email: dunno@dunno.ua
 Гаманець: 0x70997970c51812dc3a010c7d01b50e0d17dc79c8

5

Всього контрактів

4

Виконано вчасно

1

Протермінувань

13.0

Сер. затримка (дн.)

Прийнятно. ~80% контрактів виконано вчасно

Аналітика виконання

Співвідношення виконання

Вчасно Із затримкою Скасовано

Динаміка оцінок

3 МІС 6 МІС РІК

Місяць	Оцінка
лип. 25 р.	5.0
серп. 25 р.	2.0
вер. 25 р.	2.0
жовт. 25 р.	4.0
лист. 25 р.	4.0
груд. 25 р.	4.0

ІНФОРМАЦІЯ
ВІДГУКИ (6)
МАТЕРІАЛИ (5)
ІСТОРІЯ КОНТРАКТІВ

Редагувати інформацію

Dunno

0670777095

dunno@dunno.ua

0x70997970c51812dc3a010c7d01b50e0d17dc79c8

РЕДАГУВАТИ ВИДАЛИТИ

Завантажте CSV-файл з матеріалами, отриманий від цього постачальника.

ФАЙЛ Оберіть файл

ЗАВАНТАЖИТИ ▶

ОЧИСТИТИ ТАБЛИЦЮ 🗑️

Рисунок 3.13 – Сторінка деталей постачальника

Рисунок 3.13 демонструє сторінку деталей постачальника з наведенням показників ефективності, в тому числі інтерактивних кругової діаграми та графіку,

а також додаткову секцію з чотирма вкладками, перша серед яких пропонує функціонал редагування даних постачальника, а також завантаження списку матеріалів від контрагента.

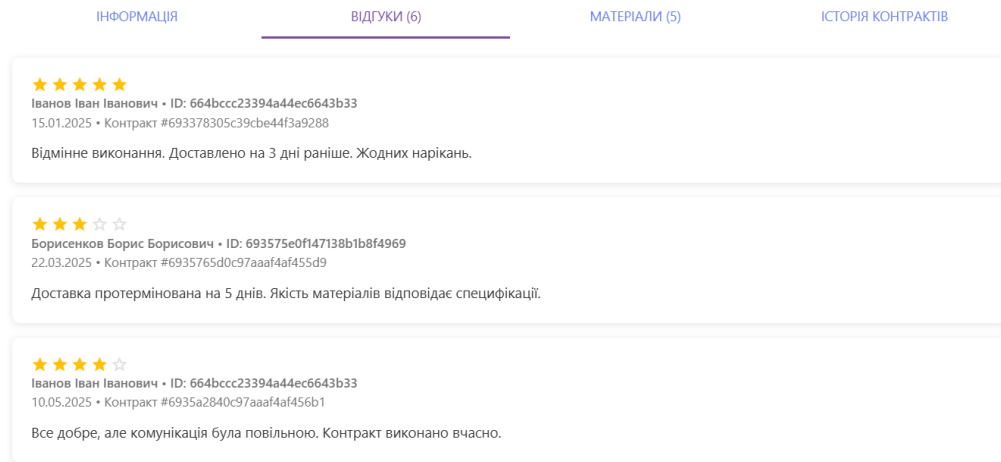


Рисунок 3.14 – Вкладка з відгуками менеджерів

Рисунок 3.14 демонструє вкладку з відгуками персоналу компанії, залишеними про постачальників після завершення угод.

ІНФОРМАЦІЯ ВІДГУКИ (6) МАТЕРІАЛИ (5) ІСТОРІЯ КОНТРАКТІВ

Категорія	Матеріал	Ціна (UAH)
Метал	Сталевий лист	100
Пластик	Поліпропіленова труба	50
Скло	Скляна панель	80
Дерево	Рейка	20
Провідники	Алюміній	10

Рисунок 3.15 – Вкладка з доданими матеріалами

Рисунок 3.15 демонструє вкладку з попередньо завантаженими матеріалами від контрагента, які надалі використовуються під час створення специфікації.

ІНФОРМАЦІЯ	ВІДГУКИ (6)	МАТЕРІАЛИ (5)	ІСТОРІЯ КОНТРАКТІВ				
<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> 5 Всього контрактів </div> <div style="text-align: center;"> 4 Завершено </div> <div style="text-align: center;"> 1 Протерміновано </div> <div style="text-align: center;"> 1 Скасовано </div> </div>							
ID	Менеджер	Дата подання	Очікування до	Вартість (UAH)	Статус	Доставка	Дії
693d476f8d...	664bccc23394...	13.12.2025	31.12.2025	1230.00	Замовлення прийнято	Вчасно	ДЕТАЛІ
693d471d8d...	664bccc23394...	13.12.2025	31.12.2025	1230.00	Замовлення прийнято	Вчасно	ДЕТАЛІ
693d46d68d...	664bccc23394...	13.12.2025	01.12.2025	1230.00	Замовлення прийнято	+13 дн.	ДЕТАЛІ
693a182463...	664bccc23394...	11.12.2025	31.12.2025	2460.00	Скасовано	—	ДЕТАЛІ

Рисунок 3.16 – Вкладка з історією контрактів

Рисунок 3.16 демонструє вкладку з повною історією контрактів за участі постачальника. За допомогою кнопки «Деталі» можливим є перехід до деталей контракту.

Innovatech Engineering		Іванов Іван Іванович, менеджер із закупівель		Склад	Постачальники	Замовлення	Статистика	Вийти
Список специфікацій								
В	Duplo	31.12.2025	Замовлення прийнято	Доставлено вчасно	✓ Проведено	✓ Виконано		
В	Duplo	31.12.2025	Скасовано		✓ Проведено	■ Не виконано		
В	Duplo	01.12.2025	Замовлення прийнято	Протермінована доставка	✓ Проведено	✓ Виконано		
В	Duplo	31.12.2025	Замовлення прийнято	Доставлено вчасно	✓ Проведено	✓ Виконано		
В	Duplo	31.12.2025	Замовлення прийнято	Доставлено вчасно	✓ Проведено	✓ Виконано		

Рисунок 3.17 – Сторінка списку специфікацій

Рисунок 3.17 демонструє сторінку зі списком специфікацій з ярликами-статусами про стани контрактів.

Innovatch Engineering
Innovation. Technology. Energy.

 Баланс компанії: 996310.0 UAH
 Іванов Іван Іванович, менеджер із закупівель
 Склад
 Постачальники
 Замовлення
 Статистика
 Вийти

Специфікація

ID: 693dd76f8d8a012636be3341 ✓ Проведено ✓ Виконано

✓ Доставка виконана вчасно

Статус контракту (Blockchain): Замовлення прийнято

Адреса смарт-контракту: 0x00Cd1Bf9A1b36cE34237eEaFeF220932846BC082

Деталі операцій

✓ ****Підписання компанією****

Хеш транзакції: 0x47637efffe82c89Faa41b83bd04b4e5dd9a072994e164f7c13032dfc82f1ab43 ETHERSCAN

Дата підписання: Sat Dec 13 2025 23:15:32 GMT+0200 (Eastern European Standard Time)

✓ ****Підписання постачальником****

Хеш транзакції: 0x3dfdd4caba975eb1c01d4f04f1f7ec97982516fd624de33cbf1463c5e96ab93ca ETHERSCAN

Дата підписання: Sat Dec 13 2025 23:15:45 GMT+0200 (Eastern European Standard Time)

🚚 ****Відмітка "В транзиті"*****

Хеш транзакції: 0x586fd1c079db0af947b2ebb0a60ac8565c467cd76b349c95ee1be49408f7e9b5 ETHERSCAN

Дата підписання: Sat Dec 13 2025 23:15:48 GMT+0200 (Eastern European Standard Time)

📍 ****Відмітка "Доставлено"*****

Хеш транзакції: 0xd01662ce89035b61c0c7f26661065d656e7ebe43a96a60c15fd3738598fdc9e4 ETHERSCAN

Дата доставки: Sat Dec 13 2025 23:15:51 GMT+0200 (Eastern European Standard Time)

💰 ****Оплата та завершення****

Хеш транзакції: 0x4f15b353872211629d19f3a529939b834de36c61ea46fa59b22a30a1e0a95e2e ETHERSCAN

Дата завершення: Sat Dec 13 2025 23:16:10 GMT+0200 (Eastern European Standard Time)

Заявку подано	Бажана дата доставки	Постачальник	Ел. пошта	Матеріал	Кількість (од.)	Вартість (UAH)
13.12.2025	31.12.2025	Dunno	dunno@dunno.ua	Алюміній	123	1230

Сумарна вартість: 1230.00 UAH

Коментар до замовлення: ТЕРМІНОВО! До 31.12!

ПОВЕРНУТИСЯ
ОФОРМИТИ НА БЛАНКУ
ВИДАЛИТИ З БАЗИ ДАНИХ

Рисунок 3.18 – Сторінка деталей смарт-контракту (завершена угода)

Рисунок 3.18 демонструє сторінку з деталями вчасно завершеного смарт-контракту. Пропонується оформлення деталей угоди на pdf-бланку та видалення контракту з бази даних. Варто зазначити, що функція видалення специфікації є доступною тільки перед проведенням та після повного завершення угоди з оплатою. Спадаючий список деталей операцій дозволяє переглянути кожну зміну, внесену акторами в блокчейн. Кнопка «ETHERSCAN» дозволяє подивитися деталі транзакції в онлайн-оглядачі Ethereum (не працює в локальному середовищі).

Специфікація ID: 693dd6d68d8a012636be321b ✓ Проведено ✓ Виконано

⚠ ДОСТАВКА ПРОТЕРМІНОВАНА
 Бажана дата: 01.12.2025
 Фактична дата: 13.12.2025
 Протермінованих днів: 13

Статус контракту (Blockchain): Замовлення прийнято

Адреса смарт-контракту: 0xcF7Ed3AccA5a467e9e704C703E8087F634F80Fc9

Деталі операцій

Заявку подано	Бажана дата доставки	Постачальник	Ел. пошта	Матеріал	Кількість (од.)	Вартість (UAH)
13.12.2025	01.12.2025	Dunno	dunno@dunno.ua	Алюміній	123	1230

Сумарна вартість: 1230.00 UAH

Коментар до замовлення: ТЕРМІНОВО! До 31.12!

[ПОВЕРНУТИСЯ](#) [ОФОРМИТИ НА БЛАНКУ](#) [ВИДАЛИТИ З БАЗИ ДАНИХ](#)

Рисунок 3.19 – Сторінка деталей смарт-контракту (протермінована угода)

Рисунок 3.19 демонструє сторінку з деталями завершеного, але протермінованого смарт-контракту. Інтерфейс надає інформацію щодо деталей протермінування і також пропонує оформлення деталей угоди на pdf-бланку та видалення контракту з бази даних.

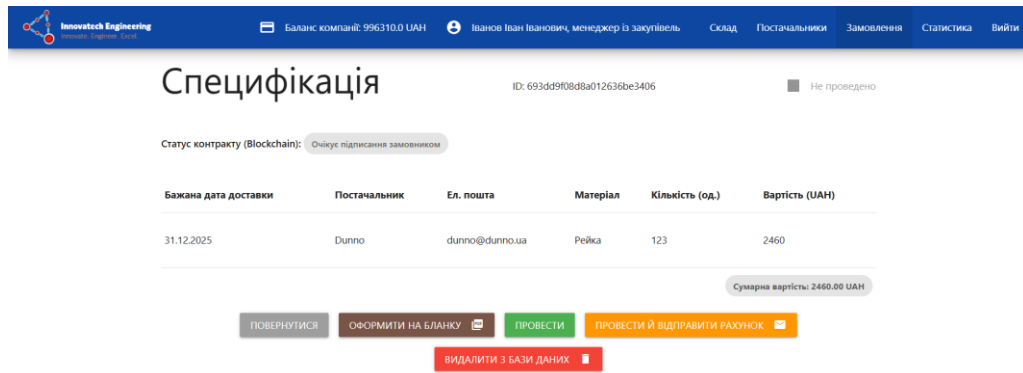


Рисунок 3.20 – Сторінка деталей специфікації (до фіксації в блокчейні)

Рисунок 3.20 демонструє сторінку з деталями щойно створеної специфікації, яку ще не було проведено і, відповідно, зафіксовано в блокчейні. Інтерфейс пропонує провести специфікацію та зафіксувати статус підписання контракту в блокчейні, або ж зробити аналогічну дію з відправленням pdf з деталями замовлення на пошту постачальникові.

UI актора «Представник постачальника» представлено на рисунках 3.11-3.20.

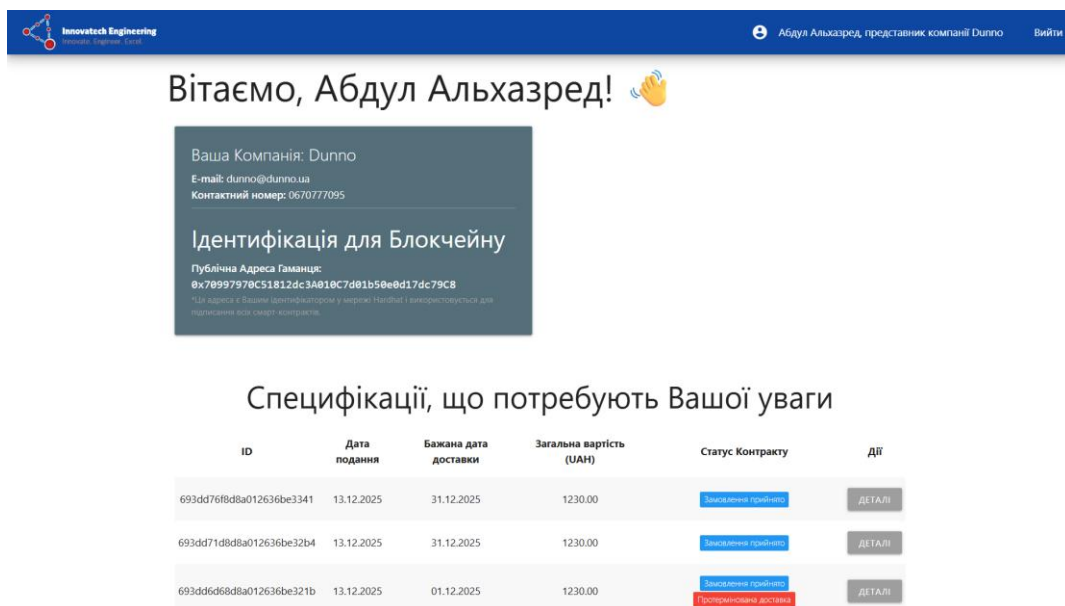


Рисунок 3.21 – Особистий кабінет актора «Представник постачальника»

Рисунок 3.21 демонструє сторінку з публічною інформацією про постачальника, а також секцію зі списком останніх угод – активних чи завершених. Індикація в вигляді ярликів свідчить про поточний стан контракту. Кнопка «Деталі» дозволяє перейти до сторінки з детальною інформацією про контракт.

Контракт на закупівлю
№ 693de0298d8a012636be343c

Деталі контракту
 Постачальник: Dunno
 E-mail: dunno@dunno.ua
 Бажана дата доставки: 31.12.2025
 Статус контракту (Blockchain): Очікує оформлення постачальником
 Адреса смарт-контракту: 0x6881D87F95878FE058998F1966F4baba5De1aed

Деталі операцій

****Підписання компанією****
 Хеш транзакції: 0xf2bfea58a8cc7c358f5a5bfd6151e7835b36906331d8c5b8f82c48983333018
 Дата підписання: Sat Dec 13 2025 23:52:48 GMT+0200 (Eastern European Standard Time)

Матеріали (1 позицій)

Матеріал	Кількість	Вартість позиції (UAH)
Скляна панель	12323 шт.	985840

Коментар до замовлення: ТЕРМІНОВО! До 31.12!
 Загальна вартість: 985840.00 UAH

*Увага: Підписання та скасування є незворотними діями в блокчейні.

Рисунок 3.22 – Сторінка детальної інформації про угоду (стан – підписано компанією-замовником)

Рисунок 3.22 демонструє інтерфейс з детальною інформацією про угоду, а також дозволяє представникові змінити статус контракту шляхом натискання кнопок «Підписати смарт-контракт» та «Скасувати смарт-контракт». Дія

«Скасувати смарт-контракт» є доступною лише до фіксування постачальником факту відправлення замовлення. Після натискання кнопки «Підписати смарт-контракт», її замінює нова активна кнопка «Підтвердити відправлення» (рис. 3.23), а після використання останньої – з’являється кнопка «Підтвердити доставку» (рис. 3.24). «Підтвердити доставку» є останньою кнопкою, що дозволяє представнику постачальника зафіксувати зміни в смарт-контракті. Після її використання доступною стане лише кнопка повернення до списку угод (рис. 3.25).

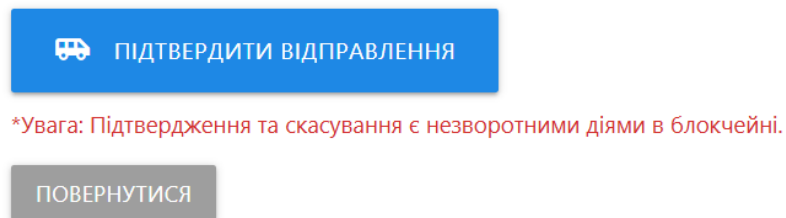


Рисунок 3.23 – Кнопка для підтвердження відправлення замовлення

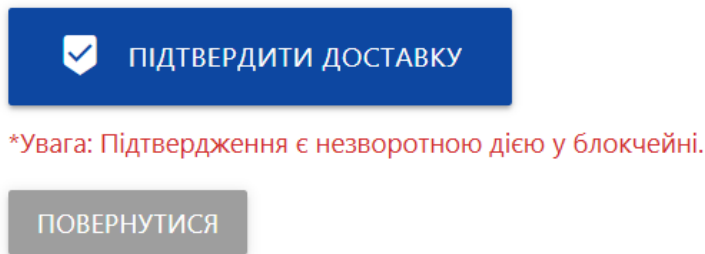


Рисунок 3.24 – Кнопка для підтвердження факту завершення процедури доставки

Контракт на закупівлю

№ 693de0298d8a012636be343c

Деталі контракту

Постачальник: Dunno
 E-mail: dunno@dunno.ua
 Бажана дата доставки: 31.12.2025
 Статус контракту (Blockchain): Доставлено
 Адреса смарт-контракту: 0x6881D87F95878FE05E998F19b66F4baba5De1aed

Деталі операцій

****Підписання компанією****

Хеш транзакції: 0xf2bfea58a8bc7c358f5a5bdf6151e7835b36906331d8c5b8f82c4898333018 [ETHERSCAN](#)

Дата підписання: Sat Dec 13 2025 23:52:48 GMT+0200 (Eastern European Standard Time)

****Підписання постачальником****

Хеш транзакції: 0x6c7fcfaeb0e6ba41cd77b6afaa076e1167f645d9fa5b6e3cf0c35c0398963bf9 [ETHERSCAN](#)

Дата підписання: Sat Dec 13 2025 23:54:22 GMT+0200 (Eastern European Standard Time)

****Відмітка "В транзиті"*****

Хеш транзакції: 0x3ae864daa93c587509096853bf534f019b6703d91733397f80d73f4ad69c530a [ETHERSCAN](#)

Дата підписання: Sat Dec 13 2025 23:54:37 GMT+0200 (Eastern European Standard Time)

****Відмітка "Доставлено"*****

Хеш транзакції: 0x8b1ff539a4329de36cf49ad690c2bc6aca48a516d36137cd7e5492e1e1949573 [ETHERSCAN](#)

Дата підписання: Sat Dec 13 2025 23:54:55 GMT+0200 (Eastern European Standard Time)

Матеріали (1 позицій)

Матеріал	Кількість	Вартість позиції (UAH)
Скляна панель	12323 шт.	985840

Коментар до замовлення: ТЕРМІНОВО! До 31.12!

Загальна вартість: 985840.00 UAH

[ПОВЕРНУТИСЯ](#)

Рисунок 3.25 – Сторінка деталей замовлення після фінальних змін, внесених представником постачальника

ВИСНОВКИ

Мета кваліфікаційної роботи полягала в розробці гібридної SRM-підсистеми на базі браузерного додатка, призначенням якого є забезпечення надійного, прозорого та автоматизованого середовища для управління контрактами із закупівлі та взаємодії з постачальниками за допомогою технології блокчейну.

В ході виконання роботи проведено ґрунтовний аналіз предметної області та проаналізовано існуючі програмні рішення, що дозволило сформулювати та обґрунтувати основні вимоги до розроблюваної SRM-підсистеми. Розглянуто концептуальну модель blockchain-SRM системи, а її архітектурне рішення піддалося порівняльному аналізу з наявними на ринку, підтверджуючи доцільність впровадження децентралізованих механізмів.

На етапі проєктування окреслені системні вимоги до компонентів підсистеми, розроблено вимоги до клієнтської частини, а також проведено UML-моделювання. Результатом моделювання стало створення діаграм прецедентів, послідовності та компонентів, що стали ключовим елементом формалізації логіки життєвого циклу контракту. Виконано обґрунтування вибору технологічного стеку, необхідного для реалізації гібридної архітектури. Створення схем поведінкових алгоритмів дозволило структуровано описати очікуваний результат роботи підсистеми та мінімізувати кількість помилок під час розроблення. Додатково спроектовано логічну структуру даних у MongoDB, що необхідна для з'єднання всіх шарів SRM-підсистеми та їхньої ефективної взаємодії з блокчейн-модулем.

Фінальним етапом розроблення підсистеми стало створення інтуїтивно зрозумілих користувацьких інтерфейсів. Ці інтерфейси, у поєднанні з іншими частинами проєкту, утворили повністю функціональну та готову до використання SRM-підсистему з можливістю створення смарт-контрактів та об'єктивної оцінки їх ефективності в довготривалому періоді. Таким чином розроблена підсистема демонструє високий потенціал для оптимізації процесів закупівлі сировини, мінімізації фінансових ризиків через гарантію Escrow, підвищення прозорості та доходів підприємства.

У результаті проведеного дослідження було підготовлено та опубліковано наукову статтю у фаховому журналі, зміст якої узгоджується з Ціллю 9 ООН – «Індустріалізація, інновації та інфраструктура».

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. структура та правила оформлення. Введ. 2015-06-22. К. Держстандарт України, 2017. 29 с.

2. Методичні вказівки з підготовки та захисту кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка, освітньо професійних програм: «Комп'ютерно-інтегровані технологічні процеси і виробництва», «Комп'ютеризовані та робототехнічні системи» / Упоряд. І. Ш. Невлюдов, Р. В. Артюх, В. В. Безкоровайний, Н. П. Демська, В. В. Євсєєв, О. І. Филипенко, О. М. Цимбал. Харків: ХНУРЕ, 2024. 57 с.

3. Положення про академічну доброчесність [Електронний ресурс]: Наказ ХНУРЕ від 02 лютого 2021 р. №50. – Режим доступу: https://nure.ua/wp-content/uploads/Main_Docs_NURE/polozhennja-pro-akademichnu-dobrochesnist.pdf (дата звернення: 27.10.2025).

4. Стандарт вищої освіти за спеціальністю 151 «Автоматизація та комп'ютерно-інтегровані технології» галузі знань 15 «Автоматизація та приладобудування» для другого (магістерського) рівня вищої освіти, затверджений наказом МОН України № 1022 від 10.08.2020р. «Про затвердження стандарту вищої освіти за спеціальністю 151 Автоматизація та комп'ютерно-інтегровані технології для другого (магістерського) рівня вищої освіти» Режим доступу: <https://mon.gov.ua/static-objects/mon/sites/1/vishcha-osvita/zatverdzeni%20standarty/2020/08/10/151-avtomatizatsiya-ta-kit-magistr.pdf> (дата звернення: 27.10.2025).

5. Основи наукових досліджень : підручник / І. Ш. Невлюдов, Ю. М. Олександров, А. О. Андрусевич, О. О. Чала ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Prague : OKTAN PRINT, 2024. – 468 с. DOI

<https://doi.org/10.46489/ONDNP> Режим доступу на ресурсі бібліотеки ХНУРЕ
<https://openarchive.nure.ua/handle/document/28574> (дата звернення: 30.10.2025).

6. Невлюдов І.Ш. Автоматизована система керування технологічними процесами в SCADA системі TRACE MODE 6: Навчальний посібник / І.Ш.Невлюдов, А.О. Андрусевич, В.В. Євсєєв, С.С. Максимова, М.Г. Стародубцев, В.В.Невлюдова. – Кривий Ріг: Криворізький коледж НАУ, 2018. – 320 с.

7. ДСТУ 3008: 2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання / Нац. стандарт України. – Вид. офіц. – [Чинний від 2017 – 07 – 01]. – Київ: ДП «УкрНДНЦ», 2016. – 26 с.

8. Олінкевич Я.В. SRM-система в сучасному підприємстві: ефективне управління бізнес-процесами // Автоматизація та приладобудування («Automation and Development of Electronic Devices» ADED-2024): збірник студентських наукових статей / Харківський національний університет радіоелектроніки; [редкол.: І.Ш. Невлюдов та ін.]. – Харків : ХНУРЕ, 2024. Вип. 2. С. 202–206.

9. Олінкевич Я.В. Впровадження ERP-системи на виробництві // Автоматизація та приладобудування («Automation and Development of Electronic Devices» ADED-2023): збірник студентських наукових статей / Харківський національний університет радіоелектроніки ; [редкол.: І.Ш. Невлюдов та ін.]. – Харків : ХНУРЕ, 2023. Вип. 2. С. 219–222.

10. What is SRM and why is it important for your business in 2024?. Weproc. URL: <https://blog.weproc.com/en/supplier-management/what-is-srm/#why-use-srm> (дата звернення: 01.10.2025).

11. Teuchler R. What Is Supplier Relationship Management (SRM)?. Kodiak Hub |End- to- End Supplier Relationship Management. URL: <https://www.kodiakhub.com/blog/what-is-supplier-relationship-management-srm> (дата звернення: 10.10.2025).

12. Chitti P., Murkin J., Chitchyan R. Data management: relational vs blockchain databases. Lecture notes in business information processing. Cham, 2019. P. 189–200. URL: https://doi.org/10.1007/978-3-030-20948-3_17 (дата звернення: 11.10.2025).

13. SAP named a leader in gartner magic quadrant for source-to-pay suites. SAP News Center. URL: <https://news.sap.com/2025/03/sap-a-leader-gartner-magic-quadrant-source-to-pay-suites/> (дата звернення: 11.10.2025).

14. Blockchain integration in SAP for supply chain transparency / V. K. Ravi et al. Integrated journal for research in arts and humanities. 2024. Vol. 4, no. 6. P. 251–278. URL: <https://doi.org/10.55544/ijrah.4.6.22> (дата звернення: 15.10.2025).

15. Oracle Named a Leader in 2025 Gartner® Magic Quadrant™ for Source-to-Pay Suites. URL: <https://www.oracle.com/news/announcement/oracle-named-a-leader-in-2025-gartner-magic-quadrant-for-source-to-pay-suites-2025-03-27/> (дата звернення: 16.10.2025).

16. How can you equip your workforce with a modern ERP and make them more productive?. URL: <https://www.oracle.com/erp/procurement/> (дата звернення: 16.10.2025).

17. Oracle blockchain cloud service FAQ. URL: <https://www.oracle.com/ua/blockchain/cloud-platform/faq/> (дата звернення: 16.10.2025).

18. Як будувати UML-діаграми. Розбираємо три найпопулярніші варіанти. URL: <https://dou.ua/forums/topic/40575/> (дата звернення: 05.11.2025).

19. Understanding UML Component Diagrams | Miro. <https://miro.com/>. URL: <https://miro.com/diagramming/what-is-a-uml-component-diagram/> (дата звернення: 05.11.2025).

20. Створення схеми послідовності UML - Підтримка від Microsoft. Microsoft Support. URL: <https://support.microsoft.com/uk-ua/topic/створення-схеми-послідовності-uml-c61c371b-b150-4958-b128-902000133b26> (дата звернення: 10.11.2025).

21. Про затвердження Правил безпечної експлуатації електроустановок споживачів (ДНАОП 0.00-1.21-98). Офіційний вебпортал парламенту України. URL: <https://zakon.rada.gov.ua/laws/show/z0093-98#Text> (дата звернення: 14.11.2025).

22. ПРАВИЛА УЛАШТУВАННЯ ЕЛЕКТРОУСТАНОВОК : Норм. док. від 21.07.2017 № 476. URL: <https://zakon.isu.net.ua/sites/default/files/normdocs/pue.pdf> (дата звернення: 18.11.2025).

23. Комплекс навчально-методичного забезпечення навчальної дисципліни "Організація керування умовами праці" підготовки першого (бакалаврського) рівня усіх спеціальностей та усіх напрямів університету [Електронний ресурс] / ХНУРЕ; розроб.: Т. Є. Стиценко, Г. В. Пронюк, О. В. Мамонтов. – Харків, 2024. – 164 с.

24. БЛИСКАВКОЗАХИСТ. Частина 3. Фізичні пошкодження будівель (споруд) та небезпека для життя : Нац. стандарт України від 09.12.2021 № 490. URL: https://zakon.isu.net.ua/sites/default/files/normdocs/dstu_en_62305-3_2021_bliskavkozakhist._chastina_3._fizichni_p.pdf (дата звернення: 15.11.2025).

25. Що таке JavaScript і для чого він потрібен. GoIT. URL: <https://goit.global/ua/articles/shcho-take-javascript-i-dlia-choho-vin-potriben/> (дата звернення: 20.11.2025).

26. Node.js – Запускайте JavaScript будь-де. Node.js – Run JavaScript Everywhere. URL: <https://nodejs.org/uk> (дата звернення: 20.11.2025).

27. Express - це фреймворк для веб-застосунків, побудованих на Node.js. Express - 基于 Node.js 平台的 web 应用开发框架 - Express中文文档 | Express中文网 . URL: <https://www.expressjs.com.cn/uk/> (дата звернення: 20.11.2025).

28. Застосування шаблонних двигунів у JavaScript-розробці: Повний посібник для новачків - IT Блог. IT Блог. URL: <https://blog-it.com.ua/zastosuvannya-shablonnyh-dvyguniv-u-javascript-rozrobci/> (дата звернення: 21.11.2025).

29. MongoDB: що це за СУБД, переваги та недоліки. GoIT. URL: <https://goit.global.ua/articles/mongodb-shcho-tse-take-ta-yak-prazui/> (дата звернення: 21.11.2025).

30. Що таке Solidity?. Gate. URL: <https://www.gate.com/uk/learn/articles/what-is-solidity/870> (дата звернення: 22.11.2025).

31. Що таке Ethers.js? Глибоке занурення в бібліотеку JavaScript Ethereum. Gate. URL: <https://www.gate.com/uk/learn/articles/what-is-ethers-js-a-deep-dive-into-ethereum-s-javascript-library/2147> (дата звернення: 22.11.2025).