КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет комп'ютерних наук та кібернетики
Кафедра інтелектуальних програмних систем

Кваліфікаційна робота

на здобуття ступеня бакалавра

за спеціальністю 121 Інженерія програмного забезпечення
на тему:

ЗАСТОСУВАННЯ БЛОКЧЕЙНУ ДЛЯ ЗБЕРІГАННЯ ТА ОБМІНУ МЕДИЧНОЇ ІНФОРМАЦІЇ

Виконав: студент 4 курсу бакалаврату
Нікіта САЗОНОВ _____
(підпис)

Науковий керівник:
Доцент, кандидат фізико-математичних наук
Лариса КАТЕРИНИЧ _____
(підпис)

Засвідчую, що в цій роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Роботу розглянуто й допущено до захисту на засіданні кафедри інтелектуальних програмних систем

«__» _____ 2023р.,
протокол №__
Завідувач кафедри
Олександр ПРОВОТАР _____
(підпис)

Київ – 2023

РЕФЕРАТ

Обсяг роботи 55 сторінок, 12 ілюстрацій, 22 джерела посилань.

WEB-3, АНОНІМНІСТЬ, БЕЗПЕКА, БЛОКЧЕЙН, ДЕЦЕНТРАЛІЗОВАНА АРХІТЕКТУРА, ЗБЕРІГАННЯ ТА ОБМІН ІНФОРМАЦІЇ, КЛІЄНТ-СЕРВЕРНА АРХІТЕКТУРА, МЕДИЧНА ІНФОРМАЦІЯ, НАДІЙНІСТЬ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ПРИВАТНА ІНФОРМАЦІЯ.

Об’єктом кваліфікаційної роботи є процес збереження та редагування та медичної інформації лікарями та її перегляд та обмін пацієнтами. Предметом роботи є програмний засіб для зберігання та обміну медичної інформації в централізованій мережі.

Метою роботи є створення додатку для зберігання та обміну медичної інформації на централізований мережі блокчейн.

Інструменти розроблення: безкоштовне, вільно поширюване середовище розробки Microsoft Visual Studio Code, мова програмування TypeScript, використання бібліотек React, web3-react, CSS бібліотеки Tailwind, програмного середовища розробки Hardhat, мова програмування смарт-контрактів Solidity, бібліотеки смарт-контрактів OpenZeppelin Contracts.

Результати роботи: виконано загальний огляд типів медичної інформації, проблем її зберігання та обміну, архітектуру існуючих додатків, її переваги та недоліки. Також було досліджено та описано технологію блокчейн, її особливості, переваги та недоліки в порівнянні з клієнт-серверною архітектурою. Розроблено програмний продукт для зберігання та обміну медичної інформації на блокчейні.

Додаток для зберігання та обміну медичної інформації на блокчейні може застосовуватися у медичній сфері для реєстрації пацієнтів, поставлення їм діагнозів, перегляду вже поставлених діагнозів. Перевагами побудови даного додатку на блокчейні є підвищена стійкість до спроб зупинити роботу додатку, взлому акаунтів користувачів, а також більша прозорість до внесення даних.

ЗМІСТ

ВСТУП

Оцінка сучасного стану об’єкта розробки. В останні роки були зроблені певні кроки вперед у використанні електронних систем зберігання та обміну медичних даних, все ж існують значні виклики та недоліки, які потребують подальшої уваги та вдосконалення.

З одного боку, наявність медичних електронних систем (МЕС) дозволяє зберігати медичну інформацію в електронному форматі, що сприяє зручності та доступності обробки інформації. ЕМС можуть містити дані про пацієнтів, медичні записи, результати лабораторних досліджень та іншу важливу інформацію. Вони дозволяють швидкий доступ до даних, сприяють координації між медичними працівниками та поліпшують якість надання медичних послуг.

З іншого боку, існують певні виклики та недоліки в сучасних системах зберігання та обміну медичної інформації. Одним з головних завдань є забезпечення безпеки та конфіденційності медичних даних. Існують загрози щодо несанкціонованого доступу до даних, кібератак, втрати чи пошкодження інформації. Крім того, існує проблема відсутності єдиного стандарту обміну даними між різними медичними системами, що ускладнює обмін інформацією між різними установами та лікарями.

Актуальність роботи та підстави для її виконання. Сучасна медицина виявляє все більший інтерес до використання новітніх технологій для покращення якості надання медичних послуг та збереження медичної інформації. Одним з перспективних рішень, що пропонується для реалізації цих цілей, є технологія блокчейну. Блокчейн, як розподілена мережа, має потенціал стати надійним та безпечним механізмом для зберігання та обміну медичної інформації.

Застосування блокчейну для зберігання та обміну медичних даних відкриває нові перспективи для поліпшення ефективності та безпеки медичного обслуговування. У традиційних системах зберігання медичної інформації виникають проблеми, пов'язані з недостатньою захищеністю даних, обмеженим доступом до інформації, недостовірністю та проблемами конфіденційності.

Мета й завдання роботи. Метою роботи «Застосування блокчейну для зберігання та обміну медичної інформації» є дослідження можливості, переваг та недоліків застосування технології блокчейн у медичній сфері для зберігання та обміну інформації та розробка програмного забезпечення для цього.

Для досягнення цієї мети були окреслені наступні завдання:

1. Визначити, яка інформація називається медичною
2. Дослідити специфіку зберігання та обміну медичної інформації
3. Проаналізувати існуючі реалізації зберігання та обміну медичних даних
4. Проаналізувати переваги та недоліки найбільш ефективної реалізації
5. Дослідити технологію блокчейн, її переваги та недоліки
6. Дослідити, які проблеми блокчейн може вирішити у сфері зберігання та обміну медичної інформації
7. Дослідити особливості розробки децентралізованих додатків
8. Розробити програмне забезпечення з використанням блокчейну для зберігання та обміну медичної інформації

Об'єкт, методи й засоби розроблення

Об’єктом кваліфікаційної роботи є процес збереження та редагування та медичної інформації лікарями та її перегляд та обмін пацієнтами.

Об’єктом розроблення додатку для зберігання та обміну медичної інформації на блокчейні є спосіб збереження даних, процес їх редагування та поширення, що базуються на децентралізованій архітектурі блокчейну.

Під час розробки даного програмного продукту спочатку було чітко окреслено технічне завдання з вказанням функціоналу та критеріїв до додатку, далі було обрано та описано архітектурні компоненти системи. Після цього були обрані технічні засоби розробки.

В якості інструменту створення програмного засобу було обрано Microsoft Visual Studio Code – інтегроване середовище розробки (IDE), яке є безкоштовним, вільно поширюваним, з відкритим вихідним кодом.

Для компоненти графічного інтерфесу було обрано мову програмування TypeScript, бібліотеки React та keact-web3 та CSS бібліотеку Tailwind.

Для розробки смарт-контрактів використовувалася мова програмування Solidity, програмне середовище для розробки смарт-контрактів Hardhat, бібліотека смарт-контрактів OpenZeppelin Contracts.

Можливі сфери застосування

Дослідження, що описані в цій науковій роботі, дозволять оцінити потенціал блокчейну для зберігання та обміну медичної інформації та визначити переваги і виклики, пов'язані з його впровадженням у сферу охорони здоров'я. Реалізований додаток можна використовувати у медичній сфері для зберігання та обміну медичними даними пацієнтів. Результати цього дослідження можуть послужити основою для подальшого розвитку та удосконалення систем зберігання та обміну медичних даних, забезпечуючи більшу безпеку, ефективність та доступність медичного обслуговування.

РОЗДІЛ 1 МЕДИЧНА ІНФОРМАЦІЯ, ЇЇ ЗБЕРІГАННЯ ТА ОБМІН

1.1 **Медична інформація**

Перш ніж аналізувати способи зберігання та обміну медичної інформації, потрібно визначити, що є інформацією, та яка інформація є медичною.

1.1.1 Визначення інформації

Поняття інформації є одним із фундаментальних у сучасній науці. Це поняття походить від латинського слова "informatio", що означає роз'яснення, пояснення якогось факту, події, явища. Поряд з матерією та енергією інформація вважається найважливішою складовою світу, в якому ми живемо.

Видатні вчені в галузі теорії інформації та управління дають найрізноманітніші визначення. Засновник кібернетики, Норберт Вінер, визначає інформацію як зміст повідомлення, отриманого системою (організмом, машиною) із зовнішнього світу. Філософ і лінгвіст Бріллюен, розвиваючи наукову концепцію Вінера, визначає інформацію як міру зменшення ентропії (де ентропія - міра невизначеності), тобто інформація є засобом внесення визначеності, впорядкованості, організації. Клод Шеннон, засновник сучасної теорії інформації, трактує інформацію як особливим чином закодовані сигнали, що передаються каналами зв'язку. [1]

У звичному розумінні термін "інформація" асоціюється з якимись відомостями, даними, знаннями тощо. Інформація може бути поточними даними про цінності в деяких сферах діяльності, систематизованими відомостями про основні причинно-наслідкові зв'язки, які містяться в знанні як понятті більш загального класу, по відношенню до якого інформація є підпорядкованою. Не можна не зауважити, що інформація бере участь у процесі передачі знань, є сигналом або повідомленням. Таким чином, ще один спосіб визначити інформацію - це відомості (дані), передані однією особою іншій або одержані в результаті власних досліджень чи вивчення.

Інформацію можна розрізняти:

– за структурно-метричними властивостями - топологічна, параметрична, абстрактна.

– за галузями знань - наукова, технічна, біологічна, економічна, медична тощо.

– за типом сприйняття - слухова, зорова, смакова тощо.

Дії будь-якого автоматичного пристрою, поведінка живої істоти, та й саме життя пов'язані з передачею, зберіганням і обробленням інформації. Накопичення великої кількості інформації в певній сфері діяльності людини призводить до серйозних проблем її сприйняття та обробки.

Протиріччя між скромними людськими можливостями і зростаючим потоком інформації можна вирішити за допомогою комп'ютерів. Їхнє використання полегшує процес збирання, класифікації, відтворення та передачі інформації, що значним чином підвищує ефективність людської діяльності майже у всіх сферах.

1.1.2 Визначення медичної інформації

Медичною інформацією можна називати будь-яку інформацію, яка пов'язана з медициною, охороною здоров'я, хворобами, медичними процедурами, лікуванням, діагностикою, профілактикою, лікарськими засобами та іншими аспектами здоров'я і медицини. Вона може включати медичні записи, результати лабораторних досліджень, діагностичні звіти, рецепти, історію хвороби пацієнта, клінічні дослідження, медичну статистику тощо.

Інформатизація та бурхливий розвиток інформаційних процесів в системі охорони здоров'я в 70-х роках XX століття спочатку за кордоном, а потім і в нашій країні призвели до формування самостійної науки - медичної інформатики.

Медична інформатика - галузь науки, що швидко розвивається, у центрі уваги якої - біомедична інформація (дані і знання, їх зберігання, передача і обробка, використання для вирішення проблем або прийняття рішень). Вона вивчає закономірності і методи отримання, зберігання, обробки і використання знань в медичній науці і практиці з метою розширення горизонтів і можливостей пізнання, профілактики і лікування захворювань, охорони і поліпшення здоров'я людини. Це

наукова дисципліна, яка містить систему знань про інформаційні процеси в медицині, системі охорони здоров'я та суміжних дисциплінах, обґрунтовує та визначає шляхи і засоби раціональної організації та використання інформаційних ресурсів для цілей охорони здоров'я.

Медична інформатика сьогодні - це цілий комплекс наукових напрямків, які відрізняються один від одного як за своїм поглядом, так і за методами, що використовуються в них. І сьогодні триває дискусія про те, який метод краще для медицини - теоретичний або експериментальний: це здорове протистояння поглядів емпіричних досліджень і результатів наукових досліджень. Теоретичні припущення були переважно основою раціональної практичної медицини. Якщо колись медицину вважали мистецтвом, то тепер все більше звертаються до її теоретичного обґрунтування, надають перевагу розробці формальних теоретичних методів, які б впроваджувалися в медичну практику. Водночас розвиваються і медичні знання, в тому числі на молекулярному та генетичному рівнях. [2]

Більшість медичних даних фіксується в різних документах (наприклад, історія хвороби, направлення на дослідження, результати аналізів, рецепт, звіт про діяльність медичного закладу, реферат статті медичного журналу тощо). Але звичайні медичні документи не придатні або мало придатні для автоматизованої обробки.

Медичний документ, як правило, має складну структуру: багато розділів, параграфів, таблиць тощо. Вони створюються за допомогою стандартизованих історій хвороби, етапних епікризів, карт певних видів досліджень, паспортів закладів охорони здоров'я. Всі ці документи мають певну форму, тобто внутрішню будову, яка відображає структуру, зв'язок і спосіб взаємодії елементів об'єкта або явища, інформація про які зафіксована в цьому документі. Фахівець повинен вміти заповнювати відповідні стандартні форми медичних документів.

Як правило, в медичних документах фіксуються такі дані як:

– паспортно-демографічні - відомості про прізвище, ім'я, по батькові пацієнта, рік і місце народження, характер роботи, родичів;

– дані про структуру та функції медичних закладів, що відображають основний процес роботи медичного закладу; для медичного закладу, наприклад, дані про лабораторні та інструментальні методи дослідження, які можливі в цьому закладі;

– статистичні та управлінські дані, які є основою для подальших розрахунків характеристик державної медичної статистики (наприклад, структура закладу) та характеристик, що характеризують роботу лікаря, або відділення та закладу в цілому; до них відносяться характеристики точності постановки діагнозів (за класифікацією ВООЗ), тривалість перебування в стаціонарі, ступінь відновлення працездатності, розбіжності в діагнозах;

– планові показники, дані про господарську та бухгалтерську діяльність медичних закладів.

Інформацію про спостережувані об'єкти, процеси чи явища отримують, вивчаючи різні фізичні величини. Наприклад, стан організму можна описати системою таких параметрів, як температура тіла, частота пульсу, тиск, дані кардіограми тощо. Деякі величини можуть набувати будь-яких значень у певному діапазоні. Їх називають неперервними, а інформацію, яку вони містять, - неперервною. Неперервними величинами є, наприклад, криві зміни маси тіла, температури, відстані тощо. Багато величин можуть набувати лише цілих значень. Їх називають дискретними, а інформацію, яку вони містять, - дискретною. Приклади дискретних величин: кількість електронів в атомі, частота пульсу, кількість пацієнтів у відділенні. Таким чином, незважаючи на різноманітність видів, інформація виявляється лише у двох формах - безперервній і дискретній.

Зазвичай медичними даними вважають лише ті, що отримані шляхом вимірювання характеристик пацієнта. Кількість характеристик пацієнта, хворої або здорової людини є значною. Різноманітні медичні дані за обсягом інформації, що міститься в них, можна розділити на наступні типи:

– якісні ознаки (наявність болю, температура, колір шкіри, перкусійні та аускультативні явища);

– поодинокі числові дані (вага, артеріальний тиск, температура тіла, кількість лейкоцитів, ШОЕ);

– динамічні дані (електрограми - ЕКГ, ЕЕГ, ЕГГ; реограми РКГ, РЕГ, фонокардіограма);

– статичні знімки (рентгенограма, авторентгенограма);

– динамічні зображення (поле біопотенціалів, електрокардіограма).

Медичні дані характеризуються специфічними особливостями: нечіткість, а іноді й неузгодженість термінології;

– велика кількість якісних ознак, які суб'єктивно оцінюють стан пацієнта;

– відсутність уніфікованих алгоритмів опису стану пацієнта, процесів діагностики та лікування;

– недостатній рівень стандартизації медичної документації;

– значна варіабельність медичних даних, малі вибірки з невідомими законами розподілу, що значно ускладнює статистичні розрахунки та побудову відповідних оцінок.

1.2 Існуючі способи зберігання та обміну медичної інформації

Важливою умовою для ефективного використання зібраної медичної інформації є умови її зберігання, яким чином ці дані згруповані та відображені. Очевидно, що чітко структурованою за логічними принципами інформацією легше оперувати, запам’ятовувати та відтворювати. Ба більше, від організації інформації залежить швидкість її сприйняття, що є важливим фактором у медичній сфері. Неправильно сприйняте твердження, симптом чи діагноз може призвести до негативних наслідків, що ускладнять або навіть унеможливлять лікування.

Історично можна виділити два способи збереження інформації, у тому числі медичною, – на паперових носіях та в електронних системах. Розглянемо більш детально кожен з цих способів.

1.2.1 Паперовий документообіг

Історія медичної документації йде паралельно з історією медицини. Записи так само необхідні для медичної практики, як і ліки для ефективного лікування, і це можна простежити з давніх часів. Перші записи були примітивними за формою

і дуже відрізнялися від сучасних медичних, але вони слугували для фіксації досягнень медицини для наступних поколінь.

Одним з найстаріших прикладів паперових медичних документів є папіруси, що були використані в Давньому Єгипті близько 16-18 століттях до н.е. Вони містили медичні записи, рецепти та іншу інформацію про хвороби та лікування. Інші культури, такі як стародавні греки, римляни та китайці, також використовували папіруси та інші форми паперу для медичної документації.

У середньовіччі монастирі були центрами зберігання та передачі медичної інформації. Монахи складали рукописи зі знаннями про медицину, давали рекомендації щодо лікування та вести пацієнтську документацію. Ці рукописи і картки служили основними джерелами медичних знань у ті часи.

У 19-20 століттях, з розвитком систематичної медицини та наукових досліджень, паперовий медичний документообіг став більш організованим і структурованим. Були розроблені спеціальні форми та формати для медичних записів, які допомагали зберігати та передавати інформацію з більшою чіткістю та стандартизацією

Перші лікарняні записи, написані чорнилом, можна прочитати і сьогодні. До двадцятого століття ведення записів у лікарнях було модою; лише на початку цієї епохи медичні записи отримали серйозну увагу з боку інших типів лікарень, а особливо з боку медичних і лікарняних асоціацій.

У 1905 році самі лікарі почали замислюватися над цінністю та необхідністю адекватної медичної документації. Найбільше поліпшення почалося з початком руху за стандартизацію лікарень у 1918 році, а нові досягнення були отримані після організації працівників медичної документації та впровадження нормативних актів у сфері охорони здоров'я. [3]

Особливості зберігання медичної інформації на паперових носіях полягають у фізичному зберіганні медичних документів у вигляді папок, карток або спеціальних форм. Зазвичай ці документи зберігаються в лікарських або стоматологічних кабінетах, поліклініках, лікарнях, архівах тощо. Зберігання медичної інформації на паперових носіях вимагає фізичного простору для

зберігання документів і додаткових заходів для забезпечення їх безпеки та конфіденційності.

Переваги зберігання медичної інформації на паперових носіях полягають у простоті використання та доступу до даних. Ця система не вимагає складних технологічних рішень або спеціалізованого програмного забезпечення. Вона може бути особливо корисною в місцях з обмеженим доступом до електронних систем або в регіонах з нестабільним електропостачанням. Збереження медичної інформації на паперових носіях також може бути дешевшим на початкових етапах в порівнянні з інвестиціями у електронні системи.

Проте, зберігання медичної інформації на паперових носіях також має свої недоліки. Ця система потребує багато фізичного простору для зберігання документів, що може стати проблемою для медичних установ з обмеженими ресурсами. Зберігання і підтримка в актуальному стані паперової медичної інформації може бути часо- та працезатратним процесом. Крім того, ризик втрати або пошкодження документів, наприклад, в результаті пожежі, повені або неправильного зберігання, може призвести до втрати важливих медичних даних.

З огляду на швидкі технологічні зміни, багато медичних установ поступово переходять до електронних систем зберігання медичної інформації, що дозволяє поліпшити доступність, безпеку та ефективність обробки даних.

1.2.2 Медичні інформаційні системи

Після створення першого цифрового електронного комп'ютера в 1943 році в суспільстві набула популярності ідея про те, що незабаром комп'ютери будуть широко використовуватися в обробці інформації, в першу чергу в медицині. Протягом наступного десятиліття лікарі постійно чули про можливий революційний вплив нових технологій, які змінять всю систему медичної допомоги.

Медична інформатика сприяла глибшому розумінню того, що на сучасному рівні вже неможливо управляти медичними знаннями за допомогою традиційних методів ведення записів на папері. Свою роль у цьому зіграло і переконання, що

процес прийняття кваліфікованого медичного рішення (постановка діагнозу, вибір методу лікування) для сучасної медицини так само важливий, як і збір фактичного матеріалу, на якому ґрунтується вибір рішення лікарем або планування наукового дослідження.

Останнє призвело до появи принципово нового напрямку - теорії медичних інформаційних систем. У сучасному розумінні медична інформаційна система - це сукупність методологічних прийомів, технічних засобів і алгоритмів управління, призначених для збору, зберігання, обробки і передачі інформації в медичних установах. Одними з перших прикладів використання обчислювальної техніки в лікарнях були автоматизовані системи для допомоги лікарю в прийнятті рішень. Однак не всі програми, розроблені для використання комп'ютерів у медицині, переслідували саме цю мету. Деякі з програм були також присвячені вивченню можливості створення єдиної лікарняної інформаційної системи. Ці початкові проекти, можливо, не були дуже амбітними, оскільки вони були спрямовані на досягнення практичних результатів у короткі терміни. Однак труднощі, з якими зіткнулися їхні розробники, були досить значними.

Очевидно, що практично побудована комп'ютерна система може підвищити продуктивність і ефективність роботи медичних працівників і потенційно допомогти зменшити витрати на оплату праці в лікарні. Фрідман і Мартін запропонували модель медичної інформаційної системи (МІС), що базується на шести різних компонентах: програма управління, фінансові процедури, комунікації та мережа, управління лікарнею, медична документація та медична підтримка [4].

Основою для побудови МІС є уніфікована медична документація і, перш за все, історія хвороби. Історія хвороби підсумовує те, що сталося з пацієнтом у минулому, і документує спостереження, діагностичні висновки та плани медичного персоналу. У певному сенсі, це зовнішня пам'ять, до якої медичні працівники можуть звернутися, коли їм потрібно знайти інформацію про пацієнта через деякий час.

Госпітальна історія хвороби є основним механізмом, який забезпечує безперервність лікування під час перебування пацієнта в стаціонарі. У свою чергу,

амбулаторна історія хвороби допомагає забезпечити безперервність лікування від одного візиту пацієнта до іншого. Зі збільшенням тривалості життя та старінням населення фокус амбулаторної допомоги зміщується в бік профілактики та лікування хронічних захворювань, а не лікування гострих. Амбулаторна історія хвороби дозволяє медичним працівникам переглядати дані, зібрані за досить тривалий період часу, і таким чином вивчати перебіг захворювань пацієнта.

Автоматизовані системи ведення історії хвороби надають більшість звітів про прогресування хвороби. При цьому дані організовані відповідно до часу їх збору. Таким чином, акцент робиться на зміні стану пацієнта в часі. При цьому, пошук необхідної інформації про стан пацієнта в МІС відбувається приблизно в чотири рази швидше, ніж у звичайній історії хвороби. Підкреслимо, що при цьому МІС дозволяє відображати більшу частину історії хвороби у вигляді звітів (епікризів) або у вигляді компактних і більш наочних документів. [5]

Сучасні умови розвитку медицини диктують вимоги до інтеграції амбулаторної практики та стаціонарного лікування з метою максимізації обсягу діагностичних досліджень і лікувальних заходів, що проводяться в амбулаторних умовах, усунення дублювання та гарантування надання кваліфікованої медичної допомоги в будь-яких умовах. Пацієнт потрапляє до стаціонару лише для проведення обстежень, які можуть бути небезпечними в амбулаторних умовах, або для інтенсивної терапії. Таке поєднання амбулаторної та стаціонарної допомоги вимагає, щоб вся необхідна інформація була доступна саме там, де в цей момент обстежується пацієнт. Консолідація інформації, отриманої з різних джерел, є вкрай необхідною. Мається на увазі оцінка безперервності даних за змістом порівнянності. Таким чином, всі дані, отримані з системи-джерела, повинні містити часові мітки, а процес збору даних повинен включати перевірку часових міток при формуванні затвердженого плану лікування пацієнта. Найкраще цього можна досягти за допомогою індивідуального медичного електронного паспорта (МЕП), що містить інформацію про стан здоров'я пацієнта.

Донедавна самі пацієнти переважно зберігали інформацію про обстеження чи лікування в архаїчних паперових медичних картках. Незручність останніх

особливо гостро відчувається сьогодні. З одного боку, лікар повинен витрачати час, іноді вимірюваний годинами, на пошук і ознайомлення з необхідними даними об'ємних історій хвороби, з іншого боку, пацієнт повинен стежити за збереженням і заповненням традиційної медичної документації, що ускладнюється при переїзді (наприклад, на інше місце проживання, в санаторій). Проблема стає ще гострішою, якщо включаються дані візуалізації (рентген, КТ, МРТ).

Прийняття МЕП як єдиного сховища особової медичної інформації хоч і вимагає не тільки великих змін у підході до реєстрації даних, а й прийняття законодавчої бази, але вирішує більшість незручностей використання паперового варіанту МЕП, описаного вище. Надалі в роботі будуть розглядатись особливості реалізації системи МЕП, а також її переваги і недоліки.

1.3 Існуючі МІС та їх архітектура

У 2018 році в парламенті був зареєстрований законопроєкт про реформи системи охорони здоров'я в Україні. У рамках реалізації реформи було створено "Національну службу здоров'я України" (НСЗУ). Відповідно до Закону, серед її основних функцій є забезпечення роботи електронної системи охорони здоров'я (eHealth).

Електронна інформаційна система охорони здоров'я eHealth забезпечує автоматизацію обліку медичних послуг та управління медичною інформацією в електронному вигляді. Структура eHealth включає центральну базу даних та електронні медичні інформаційні системи, з автоматичним обміном даними через відкритий програмний інтерфейс (API). Після створення державного центрального компонента eHealth з'явилося багато медичних інформаційних систем. Наразі їх понад 20, з найбільш популярних і досконалих слід відзначити: Helsi, Doctor Eleks, Emcimed, PBCH, MedStar.

Система Helsi - медична інформаційна система, яка є однією з найбільших в Україні та охоплює велику частку ринку. До переліку партнерів Helsi входять лікарі, пацієнти, медичні заклади з усієї України, фармацевтичні та страхові компанії. За офіційними даними Національної служби здоров'я України через

систему Helsi укладено 15,6 млн декларацій між лікарями та пацієнтами, що становить приблизно 44% від загального обсягу укладених декларацій.

Система Helsi розроблена у вигляді веб-додатку, що складається з трьох частин: helsi.me, helsi.pro, reform.helsi.me. Helsi.me - це частина системи, призначена для пацієнтів, helsi.pro - частина системи, призначена для лікарів та медичного персоналу (прийом ведеться у форматі eHealth, діагнози кодуються через систему ICPC2 або МКХ-11), reform.helsi.me - частина системи, призначена для адміністрації медичного закладу. [6]

Таким чином, через наявність великої кількості даних, поміщених у декілька баз даних, а також через необхідність створення відкритого програмного інтерфейсу, можна побачити, що більшість наявних МІС використовують клієнт-серверну архітектуру (Рис. 1.1). Клієнт виконує запит і отримує відповідь, яка формується серверною частиною. А серверна частина займається обробкою запиту, формуванням відповіді та відправкою відповіді клієнту.

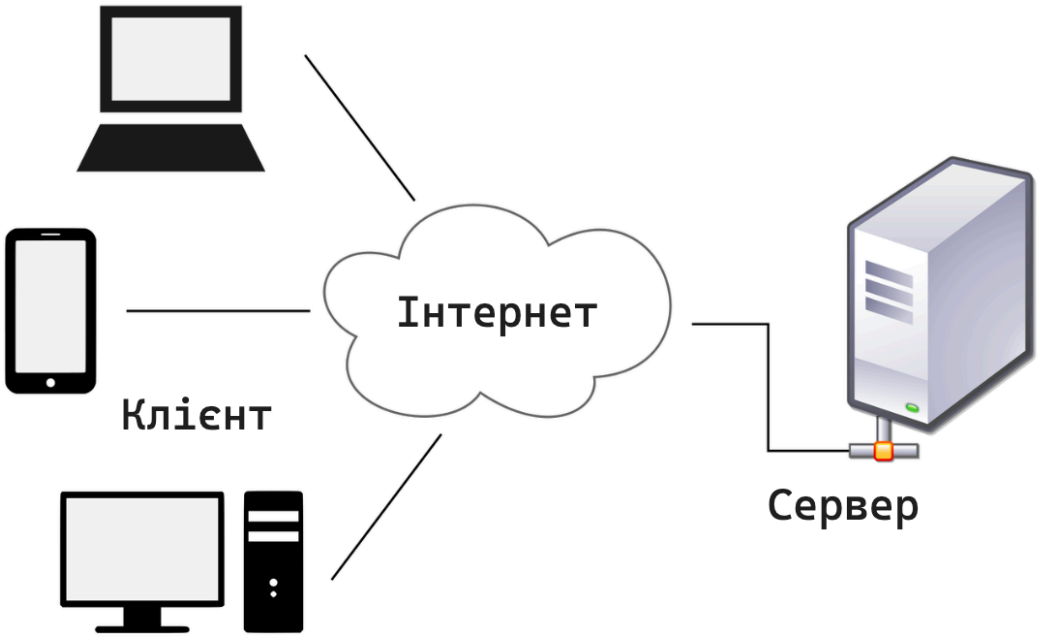

Рисунок 1.1 – Модель клієнт-серверної архітектури

Ця модель включає в себе два окремих процеси, тобто запущені програми - одна з них працює на клієнтській машині, а інша - на серверній машині. Клієнтська

частина, або просто клієнт, є "обличчям" програми, тобто тим, що бачить користувач, а серверна частина служить центральним вузлом обчислень - мозком, який зберігає, обчислює і видає значні обсяги даних.

За більшості умов один сервер може обслуговувати велику кількість (сотні або тисячі) клієнтів одночасно. Спілкування відбувається у формі клієнтського процесу, який надсилає повідомлення через мережу серверному процесу. Потім клієнтський процес чекає на відповідь. Коли серверний процес отримує запит, він виконує запитувану роботу або шукає запитувану інформацію і повертає відповідь.

Така архітектура дозволяє розділити зони відповідальності між двома підсистемами і зробити їх більш незалежними. Немає необхідності переписувати серверну частину для різних клієнтських реалізацій і навпаки, при зміні внутрішньої логіки роботи сервера - всі клієнти можуть продовжувати користуватися ним до тих пір, поки дотримується встановлений API.

1.4 Особливості та проблеми централізованого підходу до зберігання та обміну медичної інформації

У сучасному світі більшість сервісів мають централізовану архітектуру, тобто це системи, в якій контроль, управління та повноваження щодо прийняття рішень зосереджені в центральному місці або організації. Такий підхід передбачає консолідацію ресурсів, функцій і процесів прийняття рішень у центральній точці, як правило, центральному сервері або адміністративному центрі.

За централізованого підходу дані, додатки та інфраструктура, як правило, знаходяться в центральному центрі обробки даних або сервері, а користувачі отримують доступ до цих ресурсів віддалено. Така централізація забезпечує кращий контроль, безпеку та стандартизоване управління інформаційними ресурсами. Це також полегшує обслуговування, оновлення та усунення несправностей, оскільки зміни можна впроваджувати централізовано.

Таким чином, можна зробити висновок, що оскільки більшість МІС побудовані на клієнт-серверній архітектурі, вони є централізованими. Але за

рахунок розміщення всіх ресурсів в одному місці, такий підхід має певні недоліки, і ось деякі з них:

– Єдина точка відмови: Коли централізована система або сервер зазнає збою або відключення, це може порушити роботу всієї системи і вплинути на всіх підключених користувачів. Ця вразливість збільшує ризик простою та потенційної втрати продуктивності.

– Підвищена залежність від мережі: При централізованому підході користувачі значною мірою покладаються на мережеву інфраструктуру для доступу до ресурсів. Якщо мережеве з'єднання повільне, нестабільне або недоступне, це може серйозно вплинути на якість роботи користувачів і продуктивність.

– Проблеми з масштабуванням: Централізовані системи можуть стикатися з проблемами масштабованості зі збільшенням кількості користувачів або ресурсів. Додавання нових користувачів або розширення системи може вимагати значних інвестицій в модернізацію інфраструктури або додаткові ресурси.

– Обмежена автономія та гнучкість: При централізованому підході прийняття рішень і контроль зосереджені в центральному органі влади. Це може обмежувати окремі відділи або користувачів у прийнятті незалежних рішень або налаштуванні IT-рішень відповідно до їхніх конкретних потреб.

– Вищі витрати на обслуговування та підтримку: Управління та обслуговування централізованої інфраструктури може бути складним і дорогим. Для цього часто потрібен спеціалізований персонал, регулярні оновлення, резервні копії та заходи безпеки, що призводить до вищих операційних витрат.

– Потенційні ризики для безпеки: Централізована система стає вигідною мішенню для зловмисних атак. Якщо центральний сервер або центр обробки даних скомпрометований, це може призвести до витоку конфіденційних даних і перебоїв у наданні послуг усім користувачам, підключеним до системи.

– Відсутність локалізованої продуктивності: При централізованому підході користувачі, які отримують доступ до ресурсів віддалено, можуть зіткнутися з проблемами продуктивності через затримку в мережі. Такі дії, як передача даних

або співпраця в режимі реального часу, можуть бути повільнішими або менш оперативними.

При роботі з персональними даними, як от з медичними електронними паспортами, що містять приватну інформацію про стан здоров'я людини, надзвичайно важливою є безпека та анонімність. Взлом центрального серверу та витік даних про здоров'я людей є абсолютно неприпустимим. Цього можна уникнути за рахунок додавання анонімності, тобто дані особи асоціюються не з цією особою, а з якимись іншими даними, які неможливо пов'язати з цією особою. Таку можливість і надає блокчейн.

Також блокчейн дозволяє зменшити фінансові та часові витрати на масштабування системи, коли існуюча інфраструктура не здатна задовольнити потреби великої кількості користувачів. Завдяки децентралізації блокчейну, проблема єдиної точки відмови теж зникає – оскільки виконавчі одиниці розподілені бо вузлах блокчейну, зловмиснику буде набагато важче зламати систему.

Отже, застосування блокчейну для зберігання та обміну медичної інформації є перспективним, оскільки може покращити певні проблеми централізованої архітектури у цій сфері.

РОЗДІЛ 2 ВИКОРИСТАННЯ БЛОКЧЕЙНУ ДЛЯ ЗБЕРІГАННЯ ТА ОБМІНУ МЕДИЧНОЇ ІНФОРМАЦІЇ

2.1 Історія блокчейну

Перш ніж вивчати основи технології блокчейн та її застосування, варто коротко ознайомитися з її історією, адже вона бере свій початок ще у минулому тисячолітті, та містить багато важливих моментів.

У 1991 році вчені-дослідники Стюарт Хейбер і В. Скотт Сторнетта представили технологію блокчейн. Ці вчені прагнули отримати практичне рішення для позначення часу на підписаних цифрових документах, щоб їх не можна було підробити або змінити дату їх підписання. Обидва вчені разом розробили систему за допомогою використання криптографії, у якій документи з позначкою часу підписання зберігаються у вигляді ланцюжка блоків.

Після цього в 1992 році була створена компанія, яка використовувала технологію Merkle trees і систему, розроблену Стюартом Хабером і В. Скоттом Сторнеттою, з деякими додатковими функціями. Таким чином, технологія блокчейн стала ефективною для зберігання декількох документів, які збиралися в один блок. Використовувався захищений ланцюжок блоків, який зберігає кілька записів даних у послідовності. Однак ця компанія збанкрутувала, коли в 2004 році вийшов патент на одну з технологій.

У 2000 році Стефан Конст опублікував свою теорію криптографічних захищених ланцюжків, а також ідеї щодо її реалізації.

У 2004 році криптографічний активіст Хел Фінні представив систему цифрових грошей, відому як "Повторно використовуване підтвердження роботою" (“Reusable Proof of Work”). Цей крок став переломним в історії блокчейну та криптографії. Така система допомагає іншим вирішити проблему повторного витрачення одних і тих самих грошей, зберігаючи право власності на токени, зареєстровані на довіреному сервері.

Після цього 2008 року Сатоші Накамото концептуалізував поняття розподіленого блокчейну (decentralized blockchain) у своїй доповіді "Міжкористувальницька електронна система грошей" (“A peer to peer electronic cash system”). Він модифікував модель Merkle tree і створив систему, яка є більш безпечною і містить захищену історію обміну даними. Його система працює на основі міжкористувальницької мережі зі збереженням позначок часу. Вона стала настільки популярною, що саме блокчейн став найбільше асоціюватися з криптографією.

Після цього стало очевидно, що технологія блокчейну є дуже перспективною, тому її розробка стала більш стабільною, і мала попит у різних сферах. У 2009 році Сатоші Накамото випускає технічний документ, що описував основи та економічну доцільність біткоїна (Bitcoin White paper). Доволі цікава історія у свій час стала доказом неймовірної безпеки блокчейну. Джеймс Хауеллс був IT-працівником у Великій Британії, у 2009 році запускає певне програмне забезпечення, яке підтримує блокчейн, за що починає отримувати біткоїн, і припиняє це робити у 2013 році. Він витратив приблизно $17 000 на інфраструктуру, а після того, як припинив, продав частини свого ноутбука за ненадібністю, але залишив диск з собою, щоб коли він знову захоче працювати з біткоїнами, він міг його використати. Однак під час прибирання свого будинку в 2013 році він ненароком викинув свій диск разом зі сміттям. Якщо прорахувати вартість його біткоїнів зараз, то можна отримати величезну суму у майже $127 мільйонів. Біткоїни ще досі не виведені з його блокчейн-акаунту, який залишився на цьому диску. Багато людей у світі протягом довгого часу займалися пошуком цього чарівного диску з біткоїнами, але, на жаль, ще ніхто не мав успіху [7].

2014 рік став переломним для технології блокчейн. Технологія відокремлюється від валюти і народжується Blockchain 2.0. Фінансові установи та інші галузі почали переорієнтовуватися з простої цифрової валюти на розвиток технологій блокчейн [8].

У 2015 році було запущено Ethereum Frontier Network, що дозволило розробникам створювати певні додатки, які користувачі можуть запускати на

блокчейні (смарт-контракти). Того ж року Linux Foundation запустив проект Hyperledger [9].

У 2016 році слово Blockchain прийнято як єдине слово замість двох різних концепцій, як це було в оригінальній статті Накамото. Того ж року знайдено помилку в коді Ethereum DAO, чим скористалися зловмисники і викрали майже $50 мільйонів, що призвело до розділення (hard fork) мережі на Ethereum та Ethereum Classic [10]. Також біржа біткоїнів Bitfinex була зламана, в результаті чого було викрадено 120 000 біткоїнів.

У 2017 році Японія стала першою країною, що визнала біткоїн законною валютою. Компанія Block.one представила операційну систему Blockchain EOS, призначену для підтримки комерційних децентралізованих додатків.

У 2018 році біткоїну виповнилося 10 років. Але впродовж цього року його вартість продовжувала падати, досягнувши на кінець року позначки $3800. Онлайн-платформи, такі як Google, Twitter та Facebook, заборонили рекламу криптовалют.

У 2019 році кількість транзакцій у мережі Ethereum перевищила 1 мільйон на день. Amazon оголосив про загальну доступність сервісу Amazon Managed Blockchain на AWS.

Попит на стейблкоїни (stablecoins) зріс у 2020 році, оскільки вони обіцяли більшу стабільність, ніж традиційні криптовалюти. Того ж року Ethereum запустив Beacon Chain в рамках підготовки до запуску Ethereum 2.0.

2022 року Ethereum перейшов від механізму консенсусу Proof of Work (PoW) до Proof of Stake (PoS). Оригінальна мережа Ethereum об'єдналася з блокчейном Beacon Chain. Енергоспоживання Ethereum скоротилося приблизно на 99,95% [11].

2.2 Технологія блокчейну

Отже, ця технологія була введена Накамото у 2009 році, для його популярного винаходу цифрової валюти, тобто біткойну. Накамото використовував технологію блокчейн для вирішення проблеми можливості

подвійних витрат біткоїна, але незабаром ця нова технологія почала використовуватися в багатьох інших програмах.

Фактично, блокчейн – це особливий тип бази даних з певними правилами додавання даних. Після того, як дані збережені, їх практично неможливо змінити або видалити.

2.2.1 Блоки

З плином часу, дані додаються до структур, які називаються блоками. Кожен блок будується поверх попереднього і включає фрагмент інформації, пов'язаний з попереднім (Рис. 2.1). Така система була створена з тією метою, щоб будь-який користувач, переглянувши крайній блок, легко зміг перевірити, правильність його порядку. Якщо пройти весь шлях по "ланцюгу", ми досягнемо найпершого блоку під назвою генезис-блок (з англ. “genesis” – початок).

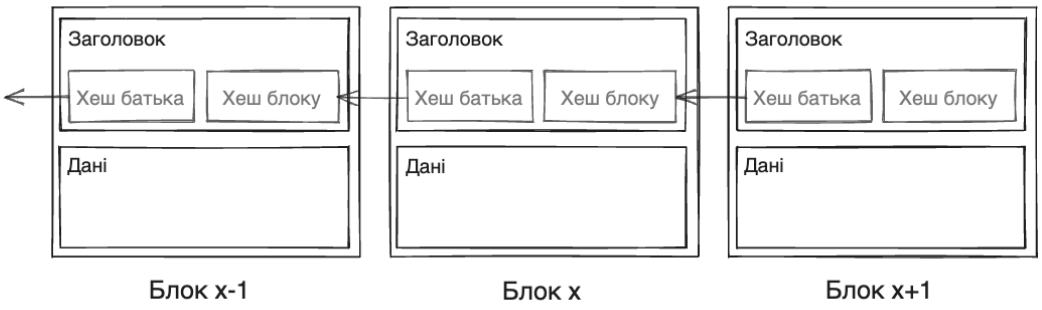

Рисунок 2.1 – Блокчейн складається з послідовних блоків, де кожний наступний базується на попередньому

Додані до блокчейну блоки не можуть бути змінені, оскільки задля зміни інформації на блокчейні з цією зміною мають погодитися всі вузли (nodes), чого вони не можуть зробити, адже блок вже був доданий, дані вже поширилися мережею вузлів [12].

Хешування – це клей, що скріплює блоки, і воно полягає в тому, що беруться дані будь-якого розміру і подаються на вхід до математичної функції, яка видає результат (хеш) завжди однакової довжини.

Хеші, які використовуються в блокчейнах, цікаві тим, що ймовірність того, що знайдуться два фрагменти даних, що дають однаковий результат, астрономічно мала. Тобто будь-яка невелика модифікація наших вхідних даних дасть зовсім інший вихід (Табл. 1).

Дані	SHA256
Taras Shevchenko	c3d2262c05d1a8238cf3f9b8c39dc1f67755819e5effa294a8a2834e3d604359
Taras shevchenko	051b6b59df97bea13b881d1153a7fae8358854bc0030100576a303dce7212e52
taras shevchenko	b52fd8195d75866063b4d049ff58e23d72cbd434cd2c9da0ae4c77725926101a

Таблиця 1 – Результати хешування фунцією SHA256 майже однакових даних зовсім різні

Факт того, що немає жодних відомих співпадінь з SHA256 (тобто двох різних вхідних даних, які дають той самий результат), є неймовірно цінним у контексті блокчейну. Це означає, що кожен блок може посилатися на попередній, включаючи його хеш, і будь-яка спроба редагування старіших блоків відразу стане очевидною [12].

Коли люди говорять про блокчейн-технологію, швидше за все, мається на увазі не тільки сама база даних, але й екосистеми, побудовані навколо блокчейнів.

Як автономні структури даних, блокчейни справді корисні лише у нішевих додатках. У поєднанні з іншими технологіями та теорією ігор, блокчейн може діяти як розподілений реєстр, який ніким не контролюється.

Це означає, що ніхто не має права редагувати записи поза правилами системи. У цьому сенсі можна стверджувати, що реєстр одночасно належить всім: учасники дійдуть згоди у тому, як він виглядає у будь-який момент.

2.2.2 Міжкористувальницька мережа (P2P)

P2P-мережа (Peer-to-peer) – це рівень користувачів, у якому немає адміністратора. Тому замість того, щоб надсилати запит на центральний сервер, коли користувач хоче обмінятися інформацією з іншим користувачем, він надсилає її напряму своїм колегам.

Розглянемо приклад (Рис. 2.1). Ліворуч розташована централізована структура, в якій учаснику A необхідно надіслати повідомлення через сервер, щоб передати його учаснику F. Однак праворуч всі учасники підключені напряму, без будь-якого посередника.

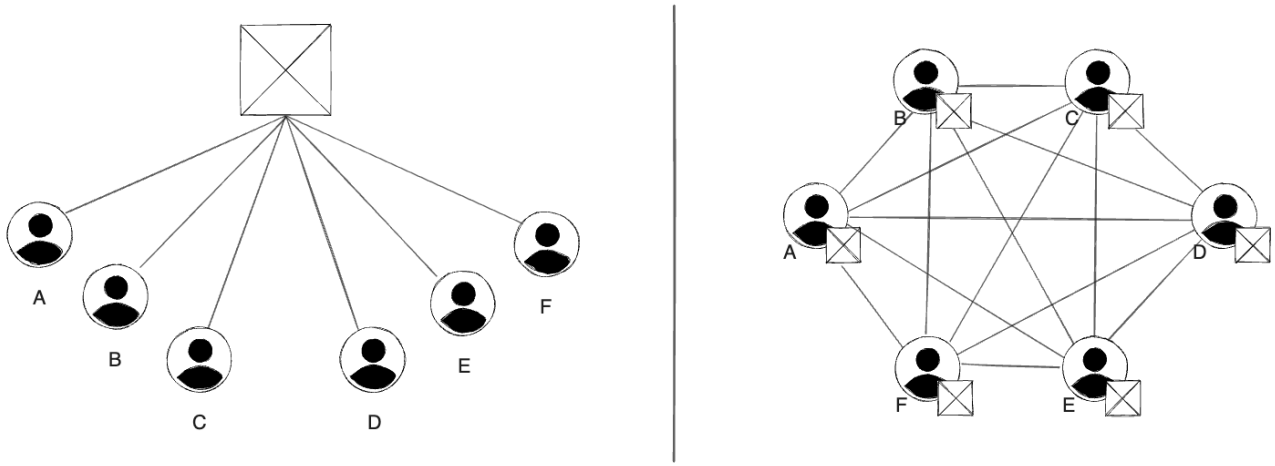

Рисунок 2.1 – Централізована та децентралізована мережа користувачів

Фактично, весь блокчейн зберігається на комп'ютері вузла. Якщо хтось залишить мережу, решта вузлів, як і раніше, зможуть отримати доступ до блокчейну та обмінюватися інформацією один з одним. Коли в ланцюг додається новий блок, дані розповсюджуються через мережу, щоб кожен міг оновити свою власну копію реєстру.

2.2.3 Блокчейн-транзакція

У централізованих банківських системах для пересилання коштів іншій людині, клієнт має надіслати запит центральному серверу банку, що спочатку перевіряє баланс клієнта, і лиш за умови достатньої кількості коштів оновлює записи для відправника і отримувача у внутрішній базі даних.

Насправді, у блокчейні відбувається схожий процес, оскільки це теж база даних, ключова відмінність полягає в тому, що немає центральної сторони, яка виконує перевірки та оновлює баланс - це повинні робити всі вузли мережі.

Якщо Аліса хоче відправити 5 BTC Бобу, вона передає повідомлення про це до мережі. Воно не буде додано у блокчейн відразу – його побачать вузли, але для підтвердження транзакції повинні бути виконані інші дії.

Як тільки ця транзакція буде додана до блокчейну, всі вузли зможуть перевірити факт її виконання і оновлять свою копію блокчейну. Тепер Аліса не може відправити ті ж 5 BTC Керол, тому що мережа знає, що вона вже витратила їх у більш ранній транзакції.

Не існує концепції імен користувачів та паролів – блокчейн використовує криптографію з публічним ключем для підтвердження права власності на кошти. Перш за все, щоб отримати кошти, Бобу потрібно згенерувати приватний ключ. Далі зі свого приватного ключа він має отримати публічний, а з нього іншими операціями (зазвичай, хешуванням) – свої публічні адреси, які і використовуються для надсилання коштів. Ділитися публічними адресами безпечно, оскільки неможливо за відносно невеликий (сотні чи тисячі років) проміжок часу отримати приватний ключ, з якого і створені ці адреси.

Далі Аліса, знаючи публічну адресу Боба, створює транзакцію в якій передає частину своїх коштів Бобу. Після чого аби довести факт власності над коштами, Аліса генерує цифровий підпис, використовуючи свій приватний ключ. Будь-хто може взяти підписане Алісою повідомлення, порівняти його з її публічним ключем і впевнено сказати, що вона має право власності на ці кошті і може відправляти їх Бобу.

Варто зауважити, що для виконання транзакцій користувачі мають платити комісію, що зазвичай виражається у криптовалюті блокчейну. Це необхідно для уникнення перевантаження розподіленої мережі інформацією від користувачів. Тобто якщо хтось захоче надіслати величезну кількість транзакцій, аби збільшити навантаження на вузли та час очікування, або навіть вивести мережу з ладу, для цього потрібно буде витратити величезну кількість фінансів, оскільки ціна комісії збільшується із завантаженістю мережі – за аналогією ринкових відносин, коли великий попит змушує ціну на товар зростати [12].

2.2.4 Вузли та алгоритми консенсусу

Вузли – це, фактично, підключені до мережі машини, які зберігають копії блокчейну та обмінюються інформацією з іншими машинами. Користувачам не потрібно обробляти всі транзакції та вносити будь-які зміни в блокчейн вручну – за це відповідає програмне забезпечення вузла блокчейну, яке потрібно запустити до підключення до мережі.

Вузли пов’язані у загальну мережу та зберігають копії блокчейну, передають один одному інформацію про транзакції та нові блоки. Але яким же чином нові блоки додаються до блокчейну?

У децентралізованій мережі відсутній центральний, чи головний, учасник, але яким же чином всі вузли розуміють, який саме вузол має додати новий блок? Для цього потрібна система, яка робить обман інших вузлів економічно невигідним, але в той же час винагороджує за чесні дії. Будь-який розумний учасник пригнитиме діяти економічно вигідно для себе.

Оскільки одним з пріоритетів децентралізованої мережі є забезпечення рівноправності всіх учасників, то і створення блоків має бути доступним для всіх. Протоколи часто забезпечують таку рівність, вимагаючи від користувача здійснити свій вклад у гру, тобто передбачає деякий ризик, що дозволить їм брати участь у створенні блоку, і якщо вони створять дійсний блок, їм буде виплачено винагороду. Однак, якщо вузли спробують обдурити мережу, решта учасників дізнається про це, і початковий вклад учасника в мережу буде втрачено [13].

Протоколи, що забезпечують дані правила, називають алгоритмами згоди, або консенсусу (consensus algorithms), бо вони дозволяють учасникам мережі досягти згоди відносно наступного блоку, що буде додано до блокчейну.

На сьогоднішній день найрозповсюдженішими є два алгоритми консенсусу:

– Майнінг (mining) – алгоритм підтвердження виконаною роботою (Proof of Work), що передбачає використання великих обчислювальних потужностей для вирішення викладеної в протоколі задачі. Ця задача вимагає, щоб вузли хешували транзакції та іншу інформацію, включену до блоку. Але щоб хеш вважався дійсним, він повинен відповідати певним правилам. Оскільки неможливо передбачити, яким буде той чи інший результат алгоритму хешування, майнери повинні хешувати злегка змінені дані, поки не знайдуть правильне рішення.

– Стейкінг (staking) – алгоритм підтвердження вкладеним капіталом (Proof of Stake). Тоді як у Proof of Work причиною діяти чесно для вузла є капітал, вкладений в обладнання для майнінгу та електрику, то у Proof of Stake – зовнішні витрати відсутні. Замість майнерів представлені валідатори, які пропонують та голосують за блоки. Вони можуть використовувати звичайний комп'ютер для створення нових блоків, але вони повинні заморозити значну частину своїх коштів, щоб отримати цей привілей. Стейкінг здійснюється із заздалегідь визначеною сумою криптовалюти блокчейну відповідно до правил кожного протоколу. Існує багато варіацій цього протоколу, але у всіх як тільки валідатор заморожує свої кошти і готовий до стейкінгу, він може бути випадково обраним для вибору наступного блоку. За умови правильного виконання вузли отримають винагороду.

2.3 Переваги блокчейну

Одна з безпосередніх переваг, зазначених у "Bitcoin whitepaper" [14], полягає в передачі платежів без посередника. Деякі блокчейни пішли ще далі, дозволяючи користувачам надсилати у транзакції інші види інформації. Усунення контрагентів означає менший ризик для залучених користувачів і призводить до нижчих комісій, оскільки посередник не отримує частку.

Як згадувалося раніше, публічна блокчейн-мережа також інклюзивна – тут немає бар'єру для входу, оскільки відсутній регулюючий орган. Єдина умова взаємодії з іншими вузлами мережі – наявність підключення до Інтернету.

Також, однією з найважливіших якостей блокчейнів є їх високий рівень стійкості до цензури, оскільки для виведення з ладу централізованої служби, зловмиснику необхідно лише атакувати сервер. Але в P2P-мережі кожен вузол працює як окремий сервер, тому для придушення блокчейну зловмиснику необхідно зупинити всі вузли мережі.

Така система, як Bitcoin, має понад 10 000 видимих вузлів по всьому світу, що унеможливлює переривання роботи мережі навіть для зловмисника з великою кількістю ресурсів. Слід зазначити, що існує також багато прихованих вузлів, які не висвітлюються на загал задля більшої безпеки блокчейну [15].

2.4 Недоліки блокчейну

Блокчейни не є панацеєю від усіх проблем. Оптимізовані для переваг, описаних у попередньому розділі, їм не вистачає розвитку в інших областях. Найбільш очевидною перешкодою для масового впровадження блокчейнів є їх низька масштабованість.

Оскільки всі учасники повинні синхронізуватись, нова інформація не може додаватися дуже швидко, оскільки вузли не встигатимуть за її оновленням. Тому розробники, зазвичай, навмисно обмежують швидкість оновлення блокчейну, щоб система залишалася децентралізованою.

Для користувачів мережі це може проявлятись у тривалих періодах очікування, якщо надто багато людей намагаються здійснити транзакції. Блоки можуть містити обмежений об'єм даних, і вони не додаються у ланцюг миттєво. Якщо транзакцій більше, ніж може поміститися в блоці, будь-які додаткові повинні чекати наступного блоку.

Ще один можливий недолік децентралізованих блокчейн-систем полягає у складності їх підтримки та оновлення. При створенні власного програмного

забезпечення, розробники можуть додавати нові функції на свій розсуд, їм не потрібно працювати з іншими учасниками або питати дозволи на внесення змін.

У середовищі з мільйонами користувачів вносити зміни значно складніше. Змінити налаштування програмного забезпечення одного вузла доволі просто, але як наслідок, цей вузол виявиться відокремленим від мережі, оскільки він відрізняється від інших.

Розглянемо наочний приклад. Припустимо, розробник вузла хоче змінити правило підтримуваного розміру блока з 1 МБ до 2 МБ. Тоді цей вузол буде надсилати іншим блоки великого розміру, але інші учасники сприйматимуть його помилковим, оскільки для них все ще діє правило розміру блоку 1 МБ. Таким чином, вони не додадуть його до своєї копії блокчейну.

Єдиний спосіб застосувати зміни – змусити їх прийняти більшу частину вузлів мережі. Перш ніж це стане можливим, запропоновані модифікації, найвірогідніше, пройдуть місяці або навіть роки інтенсивних обговорень на форумах, перш ніж більшість учасників буде згодна прийняти зміни.

2.4 Особливості розробки додатків на блокчейні

2.4.1 Децентралізовані додатки

З моменту створення Bitcoin (BTC) більше десяти років тому, блокчейни еволюціонували і відкрили безліч нових функцій та варіантів використання окрім криптовалют. Одним із цих нових напрямків є створення децентралізованих додатків (decentralized applications – dApps) для використання блокчейн-технології та покращення багатьох традиційних секторів і послуг.

Децентралізовані додатки – це цифрові програми на основі смарт-контрактів, які працюють на блокчейнах, а не на централізованих серверах. Вони пропонують широкий спектр послуг і функцій, від ігор до фінансів, соціальних мереж та багато іншого [16].

DApps мають наступні характеристики:

– Відкритий вихідний код: оскільки взаємодія з децентралізованими додатками зазвичай включає фінанси, то користувачі мають бути впевнені у їх чіткості роботи

та безпеці, тому вихідний код таких програм знаходиться у відкритому доступі задля того, щоб кожен міг його перевірити, використовувати, копіювати та змінювати.

– Децентралізований та криптографічний захист: задля безпеки даних вся інформація dApps захищена криптографією та зберігається на публічному децентралізованому блокчейні, що підтримується багатьма вузлами. Таким чином, будь-хто може перевірити достовірність слів розробників, точність роботи додатку, а також спроби взлому додатку і швидкій реакції на це.

– Токенізована система: доступ до dApps можна отримати за допомогою криптографічного токена, тобто криптовалюти, наприклад ETH, або інших.

2.4.2 Смарт-контракти

Простіше кажучи, смарт-контракт працює як детермінована програма. Вона виконує певне завдання та задає "коли і як виконуються певні умови". Таким чином, система смарт-контрактів часто слідує твердженням "якщо… то…". Але, незважаючи на популярну термінологію, смарт-контракти не є ні юридичними контрактами, ні розумними (смарт). Це просто фрагмент коду, що працює у розподіленій системі (блокчейні).

У мережі Ethereum смарт-контракти відповідають за виконання блокчейн-операцій та керування ними, коли користувачі (адреси) взаємодіють один з одним. Будь-яка адреса, яка не є смарт-контрактом, називається зовнішнім акаунтом (externally owned account – EOA). Таким чином, смарт-контракти контролюються комп'ютерним кодом, а EOA – користувачами [16].

По суті, смарт-контракти Ethereum складаються з коду контракту та двох публічних ключів. Перший публічний ключ надається автором контракту. Інший ключ представляє сам контракт, діючи як цифровий ідентифікатор, унікальний для кожного смарт-контракту.

Розгортання будь-якого смарт-контракту здійснюється через транзакцію в блокчейні, і вона може бути активована лише під час виклику EOA (або іншими

смарт-контрактами). Однак перший тригер завжди викликає EOA (користувач) [16].

Смарт-контракт має наступні характеристики:

– Розподіленість. Смарт-контракти створюються та розподіляються у всіх вузлах розподіленої мережі. Це одна з основних відмінностей від інших рішень, що базуються на централізованих серверах.

– Детермінованість. За умови однакових вхідних даних, результат роботи смарт-контракту завжди буде однаковим. Крім того, результат завжди буде однаковим незалежно від того, хто їх виконує.

– Автономність. Смарт-контракти можуть автоматизувати всі види завдань, працюючи як програма, що виконується самостійно.

– Незмінність. Смарт-контракти не можна змінити після створення. Їх можна "видалити" лише в тому випадку, якщо раніше була реалізована певна функція, але тоді всі його дані будуть втрачені. Таким чином ми можемо сказати, що смарт-контракти можуть надавати захист від несанкціонованого доступу до коду.

– Система "без довіри". Дві або більше сторін можуть взаємодіяти через смарт-контракти, не знаючи та не довіряючи одна одній. Крім того, блокчейн-технологія гарантує точність даних.

– Прозорість. Оскільки смарт-контракти засновані на публічному блокчейні, їхній вихідний код не тільки незмінний, але й доступний всім.

2.5 Переваги та обмеження децентралізованого підходу до побудови додатку для зберігання та обміну медичної інформації

Оскільки блокчейн фактично є безпечним сховищем фінансових транзакцій, його також можна застосовувати для зберігання медичних даних. Оскільки блокчейн є розподіленою системою, що використовує криптографію при зберіганні та перевірці даних, то потайки змінити дані, не маючи схвалення інших вузлів мережі, не вийде. Саме тому незмінність є однією з особливостей, які дозволяють створювати надійні бази даних медичних записів.

Крім того, мережева архітектура архітектура, що використовується в блокчейні, дозволяє синхронізувати між собою всі копії медичних даних пацієнта при внесенні оновлень, навіть якщо вони зберігаються на різних комп'ютерах. Фактично, кожен вузол мережі зберігає копію всього блокчейну, і вони регулярно обмінюються даними, щоб забезпечити їхню актуальність та достовірність. Таким чином, децентралізація і розподіл даних також є важливими аспектами.

Серед переваг блокчейну для зберігання та обміну медичної інформації можна виділити наступні [17]:

– Підвищення безпеки. Як уже згадувалося, одним з найважливіших варіантів використання блокчейну в галузі охорони здоров'я є використання технології для створення безпечної та уніфікованої розподіленої бази даних. Завдяки незмінності блокчейну, пошкодження даних більше не є проблемою. Технологія блокчейн може бути використана для ефективної реєстрації та відстеження медичних даних тисяч пацієнтів.

На відміну від традиційних баз даних, які покладаються на централізований сервер, використання розподіленої системи дозволяє обмінюватися даними з вищим рівнем безпеки, а також скоротити адміністративні витрати, які накладає поточна система. Децентралізована природа блокчейнів також робить їх менш вразливими до технічних збоїв і зовнішніх атак, які часто компрометують цінну інформацію. Безпека, яку забезпечують мережі блокчейн, може бути особливо корисною для лікарень, які часто стикаються з хакерськими вторгненнями та атаками з вимогами викупу.

– Інтероперабельність. Ще однією перевагою зберігання медичних записів на основі блокчейну є їхня здатність підвищувати простоту обміну між клініками, лікарнями та іншими постачальниками медичних послуг. Технологічні відмінності в системах зберігання даних часто ускладнюють обмін документами між організаціями. Однак блокчейн може вирішити цю проблему, дозволяючи уповноваженим сторонам отримати доступ до єдиної бази даних файлів пацієнтів або навіть записів про розподіл ліків. Таким чином, замість того, щоб намагатися

взаємодіяти з внутрішніми сховищами один одного, постачальники послуг можуть працювати разом над єдиним сховищем.

– Доступність та прозорість. На додаток до спрощення процесу обміну записами, блокчейн може також надати пацієнтам підвищений рівень доступності та прозорості їхньої власної медичної інформації. Постійний облік змін, внесених до документів пацієнта, може забезпечити надійність, достовірність та актуальність даних, а за умови належного використання, такі перевірки можуть гарантувати безпеку як від людських помилок, так і від навмисної фальсифікації.

– Захист від страхового шахрайства. Блокчейн також може бути використаний для боротьби з шахрайством у сфері медичного страхування - проблемою, яка, за оцінками, коштує американській системі охорони здоров'я близько 68 мільярдів доларів щороку. Незмінні записи, що зберігаються в блокчейні і передаються страховій компанії, можуть запобігти деяким з найпоширеніших видів шахрайства, включаючи виставлення рахунків за процедури, які ніколи не відбувалися, і стягнення плати за непотрібні послуги.

Незважаючи на те, що блокчейн пропонує багато переваг як пацієнтам, так і лікарям, він все ще має подолати деякі перешкоди, перш ніж досягне широкого впровадження в медичному секторі, а саме:

– Відповідність вимогам. Якщо взяти за приклад США, то медичні компанії, які зацікавлені у впровадженні технології блокчейн, зобов'язані дотримуватися існуючих нормативно-правових актів щодо захисту даних. По суті, такі закони визначають стандарти зберігання, обміну та захисту даних у секторі охорони здоров'я. Для того, щоб повністю відповідати цим вимогам, американським компаніям необхідно розгортати індивідуальні системи блокчейн-записів з підвищеним рівнем конфіденційності та обмеженим доступом [17].

– Початкові витрати і швидкість. З боку провайдерів рішення на основі блокчейну, ймовірно, вимагатимуть великих початкових інвестицій, що, безумовно, перешкоджає їх широкому впровадженню. Крім того, розподілені системи, як правило, значно повільніші за централізовані з точки зору кількості транзакцій в секунду. Велика мережа блокчейн з великою кількістю вузлів,

ймовірно, потребуватиме більше часу для передачі та синхронізації даних порівняно з централізованими системами. Це особливо стосується величезних баз даних, які з часом повинні будуть зберігати і відстежувати інформацію про мільйони пацієнтів. Проблема є ще гіршою для великих файлів зображень, таких як комп'ютерна томографія або МРТ-скани.

РОЗДІЛ 3 РОЗРОБКА ДОДАТКУ ІЗ ЗАСТОСУВАННЯМ БЛОКЧЕЙНУ ДЛЯ ЗБЕРІГАННЯ ТА ОБМІНУ МЕДИЧНОЇ ІНФОРМАЦІЇ

3.1 Технічне завдання

Запорукою швидкої та ефективної розробки будь-якого програмного забезпечення є визначення чіткого технічного завдання. Саме у ньому замовник заздалегідь описує бажаний функціонал, критерії до швидкості, масштабованості, вартості запуску, підтримки та обслуговування створюваного продукту. Отже, перш ніж переходити до архітектури і розробки, опишемо технічне завдання.

Під медичною інформацією, що буде зберігатися та якою користувачі будуть обмінюватися, будемо розуміти медичні електронні паспорти (МЕП), що раніше згадувалися у роботі. До МЕП входять:

– Загальні дані пацієнта – ПІБ, вік, стать, зріст, вага тощо.

– Вхідні медичні дані пацієнта – група крові, наявність хронічних захворювань тощо.

– Медична карта пацієнта – результати аналізів, діагнози тощо.

Створюваний додаток для зберігання та обміну медичної інформації має реалізовувати наступний функціонал:

– Реєстрація у системі, внесення загальних та вхідних медичних даних пацієнта його лікарем.

– Оновлення медичної карти пацієнта його лікарем.

– Зміна та видалення даних з медичної карти пацієнта його лікарем.

– Перегляд власних медичних даних пацієнтом.

– Зміна ролей адміністратором

– Видалення даних про пацієнта адміністратором.

Можна зрозуміти, що додаток підтримуватиме 3 категорії користувачів: пацієнти, звичайні лікарі та адміністратори. З міркувань безпеки, пацієнти не можуть змінювати свої, і звичайно, чужі МЕП, лише лікарі мають на це право. Задля унеможливлення навмисного видалення пацієнта з системи, це можуть робити лише адміністратори.

Очевидно, що одним з найважливіших параметрів для користувачів є швидкість роботи, тим паче, у медичній сфері, адже неприпустимими є ситуації, коли лікарі не встигають додавати дані про нових пацієнтів, або пацієнти не можуть вчасно їх переглянути. У медичній сфері важливими можуть бути лічені хвилини чи навіть секунди.

Також не менш значимим є витримуване навантаження додатком. Зі збільшенням кількості медичних закладів, що використовують реалізовуваний програмний продукт, зростає кількість пацієнтів та лікарів, а отже і одночасна кількість користувачів і запитів до додатку. Важливо, аби цей факт мінімально впливав на швидкодію чи непрацездатність додатку.

Важливо, або вартість розгортання, роботи та підтримки додатку для зберігання та обміну медичної інформації не була великою, адже наявне фінансування медичних закладів України не дозволяє покривати великі витрати на системи зберігання та обміну медичними даними. Очевидно, що за великої вартості користування чи підтримки децентралізованого додатку, медичні заклади перейдуть на централізовані (клієнт-серверні) аналоги.

Ще одним параметром є доступність та зрозумілість інтерфейсу для пацієнтів та лікарів. Від цього залежить швидкість взаємодії користувачів з додатком та емоційний штамп, що ця взаємодія лишає на них.

Не менш важливим фактором є наявність тестів різних видів для створеного функціоналу. Неодноразово траплялися випадки виявлення помилок у програмному коді, які можна було запобігти, якби програмний продукт був протестований.

3.2 Архітектура та особливості розробки додатку для зберігання та обміну медичної інформації

Архітектура додатку є однією з найважливіших і життєво важливих частин будь-якого додатку, оскільки саме за її правилами розробляється система. У цьому підрозділі описано модулі, архітектуру на різні елементи, які об’єднуються в реалізації для створення додатку. Як було визначено раніше, створюваний додаток

має бути побудований на децентралізованій системі, і бути захищеним від несанкціонованого доступу, а також мати конфіденційну систему для зберігання та обміну медичною інформацією.

Оскільки блокчейн надає можливість лише зберігати, обробляти та відтворювати дані, і не немає графічного інтерфейсу, однією з компонент створюваного додатку має бути графічний інтерфейс користувача.

Запропонована архітектура додатку (Рис. 3.1) складається з трьох компонентів, які, об’єднані разом, забезпечать реалізацію вищеописаних функцій. Надалі ми більш детально опишемо кожну з цих компонент.

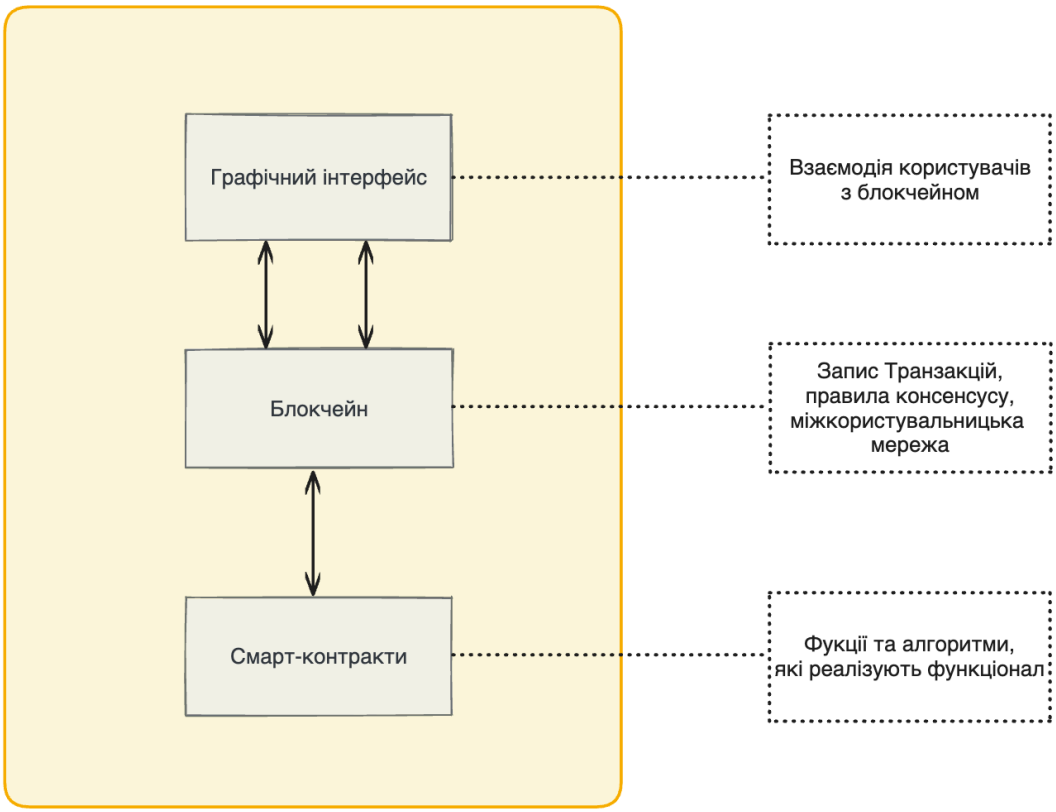

Рисунок 3.1 – Компоненти архітектури додатку

3.3 Графічний інтерфейс користувача

Користувачі - це особи, які ефективно використовують систему та її ресурси. Користувачі мають різні ролі та функції, завдяки чому їх можна ідентифікувати в системі.

Користувачами цієї системи можуть бути пацієнти, лікарі, адміністративний персонал. Основним завданням цих користувачів буде взаємодія з системою та виконання базових завдань, таких як створення, читання, оновлення та видалення медичної документації. Доступ до графічного інтерфейсу надає браузер користувача, таким чином, можна сказати, що ця компонента є веб-додатком.

В залежності від ролі користувача, йому буде показано різний графічний інтерфейс, оскільки різні ролі мають різні функції. Користувач, відповідно до призначеної йому ролі, може використовувати цей графічний інтерфейс для взаємодії з іншим рівнем системи, тобто з рівнем блокчейну.

3.4 Блокчейн

Наступна компонента архітектура - це рівень блокчейну, тобто правила і алгоритми, які забезпечують його функціонування, розподіленість та безпеку. Можна виділити три елементи блокчейну:

– Активи: транзакція - це процес, за допомогою якого зовнішній користувач може оновити стан запису або інформації, що зберігається в мережі блокчейну. Ці транзакції розглядаються блокчейном як активи, оскільки вони є частиною інформації, яку користувач може надіслати іншому користувачеві або просто зберегти її для подальшого використання.

– Правила управління: Технологія блокчейн дотримується певних правил алгоритмів для здійснення та обчислення транзакцій, у тому числі консенсусу, щоб зробити блокчейн захищеним від несанкціонованого втручання.

Мережа: Блокчейн є міжкористувальницькою мережею, у якій всі вузли мають однаковий статус та з’єднані між собою. Вузлами, фактично, є сервери, на

яких запущене певне програмне забезпечення, яке обробляє транзакції та виконує правила мережі.

3.4.1 Смарт-контракти

Третя архітектурна компонента - це смарт-контракти, які є кодом, що запускається, коли користувач транзакцією викликає певну функцію. Саме смарт контракти дозволяють зберігати, читати, змінювати та видаляти інформацію на блокчейні. Для створюваного додатку смарт контракт має включати наступні функції:

– **Додавання записів** створить медичну картку пацієнта в додатку. Вона містить поля імені, супутніх захворювань, групи крові тощо.

– **Оновлення записів** призводить до оновлення медичної інформації пацієнта.

– **Перегляд записів** дозволяє користувачеві переглядати медичні записи пацієнта. Функція перегляду записів використовується як лікарями, так і пацієнтами. Пацієнт може переглядати свої записи за допомогою системи, яка аутентифікує, що пацієнт переглядає тільки свої власні медичні записи. Для цього система використовує публічну адресу облікового запису пацієнта, щоб гарантувати, що пацієнту будуть показані тільки відповідні медичні записи.

– **Видалення записів** дозволить користувачеві видалити запис про будь-якого пацієнта. Користувачами тут будуть лікарі, яким надається право видаляти будь-які записи про пацієнта, що зберігаються в блокчейні.

– **Надання доступу** для кожної з вищезгаданих транзакцій, певний користувач повинен мати доступ до них, тобто, тільки лікар або медперсонал можуть вносити зміни в записах пацієнта або додавати їх. Таким чином, додавання та оновлення записів буде доступне лише цим особам. Більше того, пацієнт зможе переглядати свою медичну картку, але не матиме доступу до їх додавання чи оновлення.

3.4.2 Технології розробки

Важливим аспектом розробки програмного продукту є обрані технологічні засоби, адже вони прямо впливають на швидкість і якість реалізації завдання.

Для реалізації додатку обрано середовище розробки Microsoft Visual Studio Code. Це безкоштовний кросплатформовий редактор з відкритим вихідним кодом, який працює на більшості сучасних операційних системах, що дозволяє використовувати його більшій кількості розробників. VS Code підтримує безліч мов програмування і додаткових функцій, що є можливим завдяки величезній кількості розширень, які можна безкоштовно і швидко інсталювати у середовище розробки.

Ще однією зручною особливістю VS Code є підтримка технології розумного аналізу та доповнення, так званого IntelliSense, що надається для багатьох мов програмування і відображає багато корисної інформації, аналізуючи внутрішню реалізацію, аби користувачі могли краще зрозуміти код на основі виведення типів, анотацій JSDoc або файлів оголошень. Він забезпечує інтелектуальне завершення коду, інформацію про параметри, пошук за посиланнями та багато інших розширених функцій мови.

Для реалізації компоненту графічного інтерфейсу використовуються наступні технології:

– Typescript - це строго типізована мова для веб-розробки, що є надбудовою над JavaScript, яка допомагає значно зменшити кількість помилок через використання різних типів. Також додає можливість використання швидкого вибору пропозицій зміни коду.

– React – це бібліотека JavaScript із відкритим вихідним кодом для створення інтерфейсів користувача на основі компонентів. Його підтримують Meta (раніше Facebook) і спільнота окремих розробників і компаній. React можна використовувати як основу для розробки односторінкових, мобільних або серверних додатків із такими фреймворками, як Next.js.

– Tailwind CSS - це першокласний CSS-фреймворк для швидкого створення користувацьких інтерфейсів. Він є низькорівневим фреймворком CSS з безліччю налаштувань, який надає усі так звані будівельні блоки, необхідні для створення дизайнів на будь-який смак без необхідності перевизначення стилів.

– React-web3 – Typescript бібліотека для контролю та налаштування взаємодії користувачів з блокчейном.

Також для роботи з додатком у користувача має бути встановлений блокчейн гаманець для браузера MetaMask. Він використовується для підпису транзакцій, тобто доказу, що користувач належить до певного класу.

Для реалізації, тестування та розгортання смарт-контрактів використовуються технології:

– Solidity - це мова програмування зі статичною типізацією, призначена для розробки смарт-контрактів, які працюють на віртуальній машині Ethereum (EVM) або сумісних віртуальних машинах. Solidity використовує синтаксис, подібний до ECMAScript, що робить її знайомою для існуючих веб-розробників; однак, на відміну від ECMAScript, вона має статичну типізацію та варіативні типи змінних повернення [18].

– Hardhat - це програмне середовище розробки, яке допомагає розробникам тестувати, компілювати, розгортати та налагоджувати додатки на блокчейні Ethereum. Воно відіграє важливу роль у підтримці кодерів та розробників в управлінні завданнями, які є важливими для розробки смарт-контрактів та dApp. Hardhat написаний та застосовується в поєднанні з Typescript [19].

– OpenZeppelin contracts - бібліотека для безпечної розробки смарт-контрактів, побудована на міцному фундаменті перевіреного спільнотою коду [20]. OpenZeppelin Contracts допомагає мінімізувати ризики, використовуючи перевірені часом смарт-контракти для Ethereum та інших блокчейнів. Вона включає в себе найбільш використовувані реалізації стандартів ERC.

3.5 Реалізація та функціонал створеного додатку

3.5.1 Смарт-контракти

Як пояснювалося раніше, смарт-контракти є важливою частиною децентралізованих додатків, оскільки саме вони виконують операції над даними. Створюваний додаток для зберігання та обміну медичної інформації включає два смарт-контракти: CryptoHealth та AccessControl.

Ці контракти використовуються для надання доступу користувачам до децентралізованого додатку та виконання CRUD-операцій над записами пацієнтів.

Контракт CryptoHealth створений виключно для реалізації функціоналу додатку, тобто виконує операції CRUD, вимагаючи наявності певних ролей для доступу до функцій. Другий контракт, згаданий вище, а саме AccessControl, є заздалегідь створеним смарт-контрактом бібліотеки OpenZeppelin Contracts. Ця бібліотека містить велику кількість смарт-контрактів, що реалізовують найрізноманітніший функціонал, та які можна використовувати для створення власних смарт-контрактів [20]. AccessControl надає гранульований механізм визначення ролей, що і стало основною причиною вибору цього смарт-контракту.

Опишемо функції, що присутні в контрактах AccessControl та CryptoHealth. Для цього будемо використовувати псевдокод. Позначатимемо блокчейн-адресу користувача акаунтом, а акаунт, що надіслав транзацію – msg.sender.

Функції «Надати Роль», «Має Роль» та «Вимагати Роль» (Рис. 3.2) є основними у контракті AccessControl, та використовуються у CryptoHealth. Вони реалізують функціонал видачі ролей користувачам, а також диференціації користувачів за цими ролями.

```
function Надати Роль (акаунт, роль) {
    додати нову роль та акаунт у внутрішньому словнику ролей
}
```

```
function Має Роль (акаунт, роль) {
    if (акаунт має роль) {
        return true
    } else {
        return false
    }
}
```

```
function Вимагати Роль (акаунт, роль) {
    if(! Має Роль (акаунт, роль)) {
        Помилка: акаунт не має вказаної ролі
    }
}
```

Рисунок 3.2 – Функції контракту AccessControl

Функції «Додати Пацієнта» і «Отримати Дані Пацієнта» (Рис. 3.3) вимагають акаунт, що надіслав транзакцію бути доктором, тобто мати роль доктора, а також відсутність і наявність реєстрації вказаного пацієнта відповідно. Також для отримання даних пацієнта акаунтом, що надіслав транзакцію, може бути сам пацієнт. Таким чином, лише пацієнт та лікарі мають доступ до даних пацієнта.

```
function Додати Пацієнта (акаунт, дані пацієнта) {
    Вимагати Роль (msg.sender, доктор))

    if (акаунт не зареєстрований) {
         пов'язати дані пацієнта з акаунтом
    } else {
        Помилка: пацієнт вже зареєстрований
    }
}
```

```
function Отримати Дані Пацієнта (акаунт) {
    if (msg.sender == акаунт ||
        Має Роль (msg.sender, доктор)) {
        if (акаунт зареєстрований) {
             return Дані Пацієнтів [акаунт]
        } else {
            Помилка: пацієнт ще не зареєстрований
        }
    } else {
        Помилка: тільки пацієнт чи доктор мають доступ
    }
}
```

Рисунок 3.3 – Функції «Додати Пацієнта» та «Отримати Дані Пацієнта»

Функції «Змінити Пацієнта» і «Видалити Пацієнта» (Рис. 3.4) вимагають, аби пацієнт був зареєстрований, а роль доктора та адміністратора відповідно у акаунта, який надсилає транзакцію. Така система дозволяє лікарям додавати, переглядати та змінювати пацієнтів, а адміністраторам надає можливість їх видаляти.

Таким чином, смарт-контракт CryptoHealth реалізує набір створення, читання, редагування та видалення пацієнтів (CRUD операції), дозволяючи операції зміни лише докторам, а операцію видалення – адміністраторам.

Варто зауважити, що помилки, які повертають смарт-контракти, можуть бути використані іншими компонентами додатку для пояснення користувачу причини неуспішного виконання функції.

```
function Змінити Пацієнта (акаунт, оновлені дані пацієнта) {
    Вимагати Роль (msg.sender, доктор))

    if (акаунт зареєстрований) {
         пов'язати оновлені дані пацієнта з акаунтом
    } else {
        Помилка: пацієнт ще не зареєстрований
    }
}
```

```
function Видалити Пацієнта (акаунт) {
    Вимагати Роль (msg.sender, адміністратор))

    if (акаунт зареєстрований) {
         видалити зв'язок між даними пацієнта та акаунтом
    } else {
        Помилка: пацієнт ще не зареєстрований
    }
}
```

Рисунок 3.4 – Функції «Змінити Пацієнта» та «Видалити Пацієнта»

Також наявні функції «Додати Діагноз» (Рис. 3.5), «Отримати Історію Діагнозів» і «Отримати Список Активних Діагнозів». Для додавання діагнозу, відправник транзакції має бути доктором, а для читання так само – доктором або пацієнтом, дані якого запитуються. При додаванні діагнозу, він додається в історію діагнозів, а також якщо він активний – то додається до списку активності діагнозів, якщо негативний – видаляється.

```
function Додати Діагноз (акаунт, хвороба, активний) {
    Вимагати Роль (msg.sender, доктор))

    if (акаунт зареєстрований) {
         if (активний) {
            додати хворобу до списку активних
         } else {
            видалити хворобу зі списку активних
         }
    } else {
        Помилка: пацієнт ще не зареєстрований
    }
}
```

Рисунок 3.5 – Функція «Додати Діагноз»

Необхідно зауважити, що функції створених смарт-контрактів випромінюють відповідні події (emit events) [21], аби інші компоненти додатку могли реагувати на успішне виконання функцій. У цих подіях вказується інформація, що стосується виконання функцій, наприклад, акаунт пацієнта, його дані, змінені дані тощо. Враховуючи вищезгадану прозорість блокчейну, стає очевидним, що при випроміненні подій, будь-який користувач зможе їх продивлятися і пов’язувати акаунт пацієнта з його даними, що анулює перевагу анонімності. Для вирішення цієї проблеми, створюваний додаток має запускатися на приватному блокчейні, підключатися до якого і передивлятися події може обмежене коло користувачів [22].

3.5.2 Графічний інтерфейс

Графічний інтерфейс додатку для збереження та обміну медичної інформації розроблено як web-додаток, з використанням технології React. Таким чином, відображуваний інтерфейс залежить від інформації стану додатку. Наприклад, якщо користувач ще не під’єднав свій web3 гаманець, тобто додаток не має інформації про нього у своєму стані, то буде початкову сторінку.

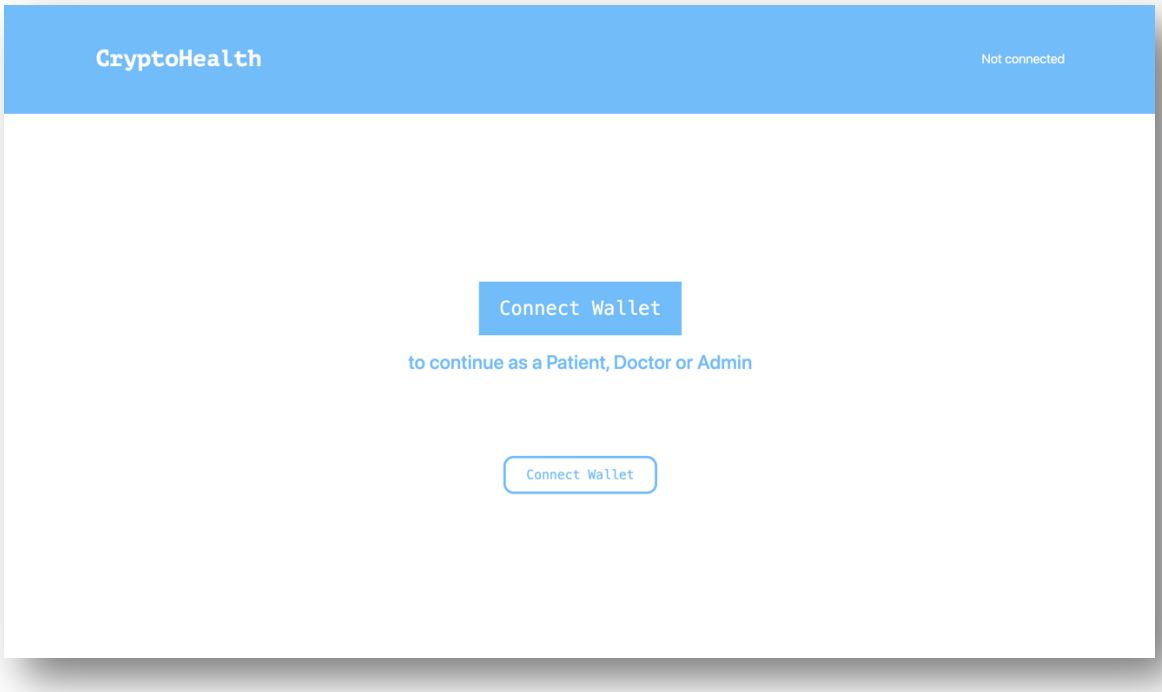

Рисунок 3.6 – Початкова сторінка додатку

При зміні стану і відображені сторінки з інформацією про пацієнта чи лікаря, виконується запит на цю інформацію на блокчейн, використовуючи під'єднаний гаманець користувача.

Графічна компонента додатку складається з декількох сторінок, перехід між якими можливий лише при зміні під'єднаного гаманця і, відповідно, ролі користувача. При відкритті додатку, відображається початкова сторінка (Рис. 3.6), на якій користувач має під'єднати свій web3 гаманець. Далі, в залежності від ролі під'єднаного акаунту, відображається відповідна сторінка.

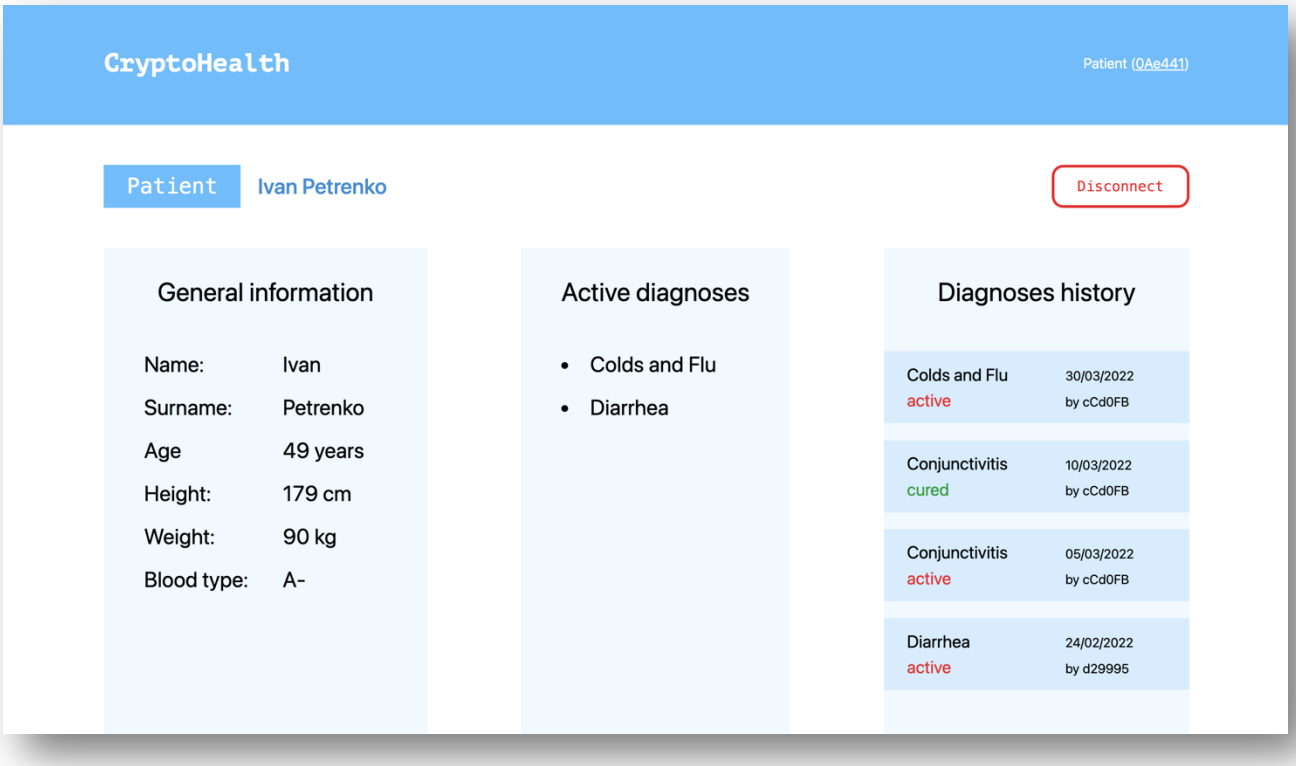

Рисунок 3.7 – Сторінка пацієнта

Якщо під'єднаний акаунт – пацієнт, відображається сторінка з інформацією про нього – загальна інформація, включаючи прізвище та ім'я, активні діагнози та історія діагнозів (Рис. 3.7). Ці дані доступні після виклику відповідних функцій смарт-контракту CryptoHealth, адреса якого доступна в конфігурації додатку. Оскільки запит робиться з під'єднаної адреси і про цього ж пацієнта, то дані повертаються успішно. Якщо ж пацієнт захоче зробити запит на смарт-контракт з

бажанням дізнатися інформацію про іншого пацієнта – функція поверне помилку. Також варто зауважити, що у верхній частині сторінки також відображено роль акаунту та перші 6 символів адреси. Немає сенсу повністю відображати адресу акаунту, оскільки ці дані займають багато місця та не несуть інформаційного навантаження, оскільки користувач і так знає свою адресу. Її перші символи відображені для запевнення користувача у тому, що він під'єднав потрібну адресу, оскільки користувачі можуть мати декілька адрес.

Також, на кожній сторінці у прямокутнику зліва продубльовано роль акаунта, а справа розміщена кнопка для відключення акаунта, натискання на яку поверне користувача на початковий екран. Ця функція дозволяє користувачу перепідключити потрібний акаунт, якщо їх у нього декілька.

Якщо підключений акаунт має роль лікаря, відображається відповідна сторінка з рядком для пошуку пацієнта за адресою. Якщо пацієнт не зареєстрований, лікар може додати його у систему. Також, для цієї функції розміщена окрема кнопка праворуч від рядка пошуку пацієнта.

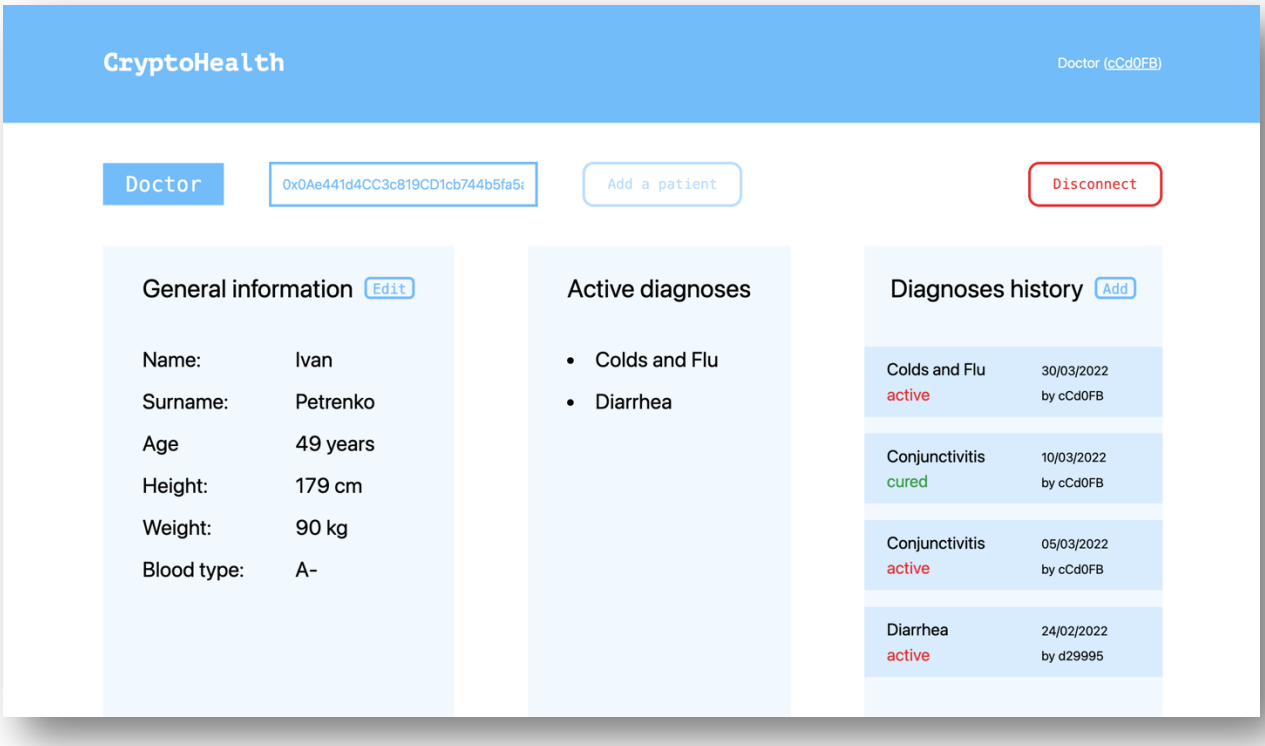

Рисунок 3.8 – Сторінка доктора з обраним пацієнтом

При введенні адреси зареєстрованого пацієнта, відображаються його дані, з можливістю їх редагування (Рис. 3.8). При натисканні на відповідну іконку, з’являється вспливаюче вікно, у якому лікарю необхідно змінити дані пацієнта. Ліворуч, у блоці з історією діагнозів, лікар може додати діагноз, ввівши потрібну інформацію у вспливаючому вікні.

Відображені діагнози посортовані за спаданням дат, тобто згори найновіші. Також наявна назва діагнозу, статус (діагностовано чи вилікувано) та адресу лікаря, що додав цей запис в історію діагнозів.

Якщо ж підключений акаунт є адміністратором, відображається сторінка, схожа на лікарську (Рис. 3.9). Зверху розташоване поле пошуку акаунту за адресою, при введенні якої відображаються його роль, можливі ролі та загальна інформація. Адміністратор може змінити роль обраного акаунту, а також видалити інформацію про нього.

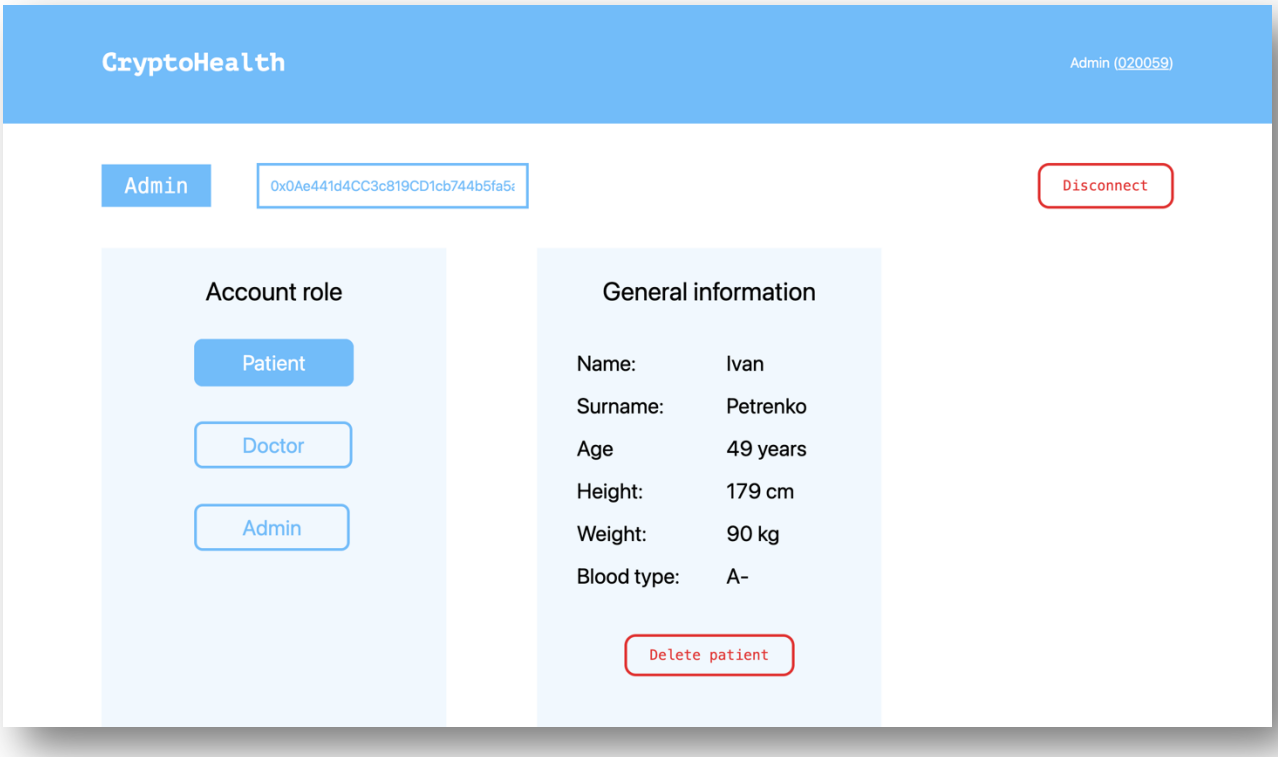

Рисунок 3.9 – Сторінка адміністратора з обраним акаунтом

ВИСНОВКИ

Відповідно до поставленої мети та завдань роботи, отримано такі основні результати:

1. Визначено медичну інформацію, досліджено специфіку її зберігання та обміну

2. Проаналізовано паперовий та електронні варіанти збереження та обміну медичної інформації, окреслено їх переваги та недоліки

3. Досліджено клієнт-серверних підхід до побудови медичних інформаційних систем, описано переваги та недоліки такого архітектурного підходу

4. Проаналізовано переваги та недоліки технології блокчейн, у тому числі для зберігання та обміну медичної інформації

5. Розглянуто особливості розробки розподілених додатків

6. Розроблено програмне забезпечення для зберігання та обміну медичної інформації

На основі результатів дослідження можна зробити такі висновки:

1. Медична інформація – це будь-яка інформація, що стосується здоров'я людини, медичного діагнозу, лікування, медичних процедур, медичної історії пацієнта та іншої медичної документації. Вона включає в себе дані, що отримані від пацієнтів, медичного персоналу, медичних установ, дослідницьких організацій та інших джерел.

2. Існує два основних способи зберігання медичної інформації: паперовий документообіг та медичні електронні системи. Незважаючи на простоту та доступність паперового варіанту, впродовж останніх років електронний стає все більш поширеним за рахунок більшої кількості доступних функцій, автоматичної обробки інформації, більш легкого обміну даними та їх стандартизованості.

3. Найбільш поширеним архітектурним підходом до побудови медичних інформаційних систем є клієнт-серверна архітектура. Її основними перевагами є відносна легкість створення та підтримки за рахунок розділення обов’язків між клієнтом та сервером, легка масштабованість за рахунок додавання серверів, а

також централізоване керування. Її недоліками є наявність одиничної точки відмови, відносно низька надійність та захист від втрат чутливої інформації та обмежена масштабованість при різкому зростанні кількості клієнтів.

4. Блокчейн є розподіленою базою даних, що запущена на сотнях вузлів мережі, що забезпечують її функціонування, та підтримує непідробну історію змін даних. Перевагами блокчейну є відсутність єдиної точки відмови, анонімність, легка масштабованість та покращена безпека. Її недоліками може бути вартість запуску та підтримки системи і складність застосування технології.

5. Тим не менш, технологія блокчейн є перспективною для застосування у сфері зберігання та обміну медичної інформації, оскільки дозволяє покращити безпеку та анонімність даних, спрощує процес обміну медичними даними та унеможливлює зупинку роботи системи через зловмисні атаки.

6. Створений додаток можна застосовувати у сфері зберігання та обміну медичної інформації у медичних закладах з наявним підключенням до мережі інтернет. Даний додаток може використовуватися пацієнтами для перегляду діагнозів, лікарями для перегляду та редагування інформації про пацієнтів та адміністраторами для видачі ролей.

7. Варто зазначити, що до створеного додатку можна додавати функціонал і продовжувати його розвивати. Таким чином, можна додати необхідність пацієнта давати дозвіл на перегляд його даних навіть лікарям, підтримувати можливість додавати результати аналізів тощо.

8. Загалом, використання технології блокчейн для зберігання та обміну медичної інформації є перспективним, оскільки дозволяє уникнути певних вагомих недоліків клієнт-серверного архітектурного підходу до розробки медичних інформаційних систем.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Енциклопедія кібернетики : у 2 т. / За ред. В. М. Глушкова. – К. : Головна редакція Української радянської енциклопедії, 1973. – Т. 1. – 584 с
2. Мінцер О. П. Інформаційна основа медицини третього тисячоліття / О. П. Мінцер / Медичний всесвіт. - 2002. - Т. 2, № 1-2. С. 150 – 160
3. Lorkowski J. Medical Records: A Historical Narrative / J. Lorkowski, M. Pokorski. // Biomedicines. – 2022. – С. 2–11.
4. Інформаційні технології у сфері охорони здоров’я : монографія / Л.Б. Ліщинська, С.А. Яремко, К.В. Копняк, І.О. Гулівата, Л.П. Гусак ; за заг. ред. Л.Б. Ліщинської. – Вінниця : видавничоредакційний відділ ВТЕІ КНТЕУ, 2018. – 240 с.
5. Мінцер О. П. Інформатика та охорона здоров’я / О. П. Мінцер // Медична інформатика та інженерія. - 2010. - № 2. - С. 8-22.
6. Розробка медичної інформаційної системи для медичних закладів первинної ланки / О. І. Панасюк, В. Л. Плескач, В. В. Стаценко, В. А. Хомазюк // Технології та інжиніринг. - 2021. - № 6. - С. 9-18.
7. López-Sorribes S. A Bibliometric Review of the Evolution of Blockchain Technologies / S. López-Sorribes, J. Rius-Torrentó, F. Solsona-Tehàs. // Censors. – 2023. – №23.
8. von Haller Grønbæk M. Blockchain 2.0, smart contracts and challenges / Martin von Haller Grønbæk. // Bird&Bird. – 2016.
9. Tual S. The Thawing Frontier [Електронний ресурс] / Stephan Tual. – 2015. – Режим доступу до ресурсу: https://blog.ethereum.org/2015/08/04/the-thawing-frontier.
10. Understanding a Revolutionary and Flawed Grand Experiment in Blockchain: The DAO Attack / [M. Mehar, C. Shier, A. Giambattista та ін.]. // Journal of Cases on Information Technology. – 2017.
11. Neureuter J. The Ethereum Merge / J. Neureuter, D. Grey. // Fidelity: Digital Assets. – 2022.

12. An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends / [Z. Zheng, S. Xie, H. Dai та ін.]. // 6th IEEE International Congress on Big Data. – 2017.
13. Michalski R. Revealing the Character of Nodes in a Blockchain With Supervised Learning / R. Michalski, D. Dziubaltowska, P. Macek. // IEEE Access. – 2020.
14. Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System [Електронний ресурс] / Satoshi Nakamoto. – 2008. – Режим доступу до ресурсу: https://bitcoin.org/bitcoin.pdf.
15. Strebko J. The Advantages and Disadvantages of the Blockchain Technology / J. Strebko, A. Romanovs. // IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE). – 2018. – №2018.
16. An overview of smart contract: Architecture, applications, and future trends / [Z. Zheng, S. Xie, H. Dai та ін.]. // IEEE Intelligent Vehicles Symposium (IV). – 2018. – №2018.
17. Pirtle C. Blockchain for Healthcare: The Next Generation of Medical Records? / C. Pirtle, J. Menachem Ehrenfeld. // Journal of Medical Systems. – 2018. – №42.
18. Solidity Docs [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: https://docs.soliditylang.org/.
19. Hardhat Docs [Електронний ресурс] – Режим доступу до ресурсу: https://hardhat.org/docs.
20. OpenZeppelin Contract Docs [Електронний ресурс] – Режим доступу до ресурсу: https://docs.openzeppelin.com/contracts/4.x/.
21. Altaş H. Data immutability and event management via blockchain in the Internet of things / H. Altaş, U. Can Çabuk, G. Dalkılıç. // Turkish Journal of Electrical Engineering and Computer Sciences. – 2022. – №30.
22. Ncube T. Private Blockchain Networks: A Solution for Data Privacy / T. Ncube, N. Dlodlo, A. Terzoli. // IMITEC 2020. – 2020.