

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Проектування та дослідження системи зберігання та
обміну медичними даними з використанням
хмарних технологій
(тема)

Виконав:

студент II курсу, групи СПм-22-4
Сальніков С.С.
(прізвище, ініціали)

Спеціальність
123 «Комп'ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма
Системне програмування
(повна назва освітньої програми)

Керівник: доц. Шматко О.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ
(підпис) Коваленко А.А.
(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління _____

Кафедра _____ електронних обчислювальних машин _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 123 «Комп'ютерна інженерія» _____
(код і повна назва)

Тип програми _____ освітньо-наукова _____
(освітньо-професійна або освітньо-наукова)

Освітня програма _____ Системне програмування _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту _____ Сальнікову Сергію Сергійовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Проектування та дослідження системи зберігання та обміну медичними даними з використанням хмарних технологій

затверджена наказом по університету від “ 01 ” квітня 2024 р. № 257 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 15 червня 2024 р.

3. Вхідні дані до роботи _____

1)Набір електронних медичних даних пацієнтів;

2)Середа розробки IDLE Python;

3) Мова програмування Python.

4. Перелік питань, що потрібно опрацювати у роботі _____

1) Аналіз предметної області та огляд існуючих рішень;

2) Вибір технологій та інструментів розробки;

3) Розробка програмних компонентів для системи зберігання та обміну медичними

4) Дослідження системи зберігання та обміну медичними даними;

5) Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп’ютерних ілюстрацій (слайдів) 20 слайдів

6. Консультанти розділів роботи (заповнюється 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	Аналіз проблеми та огляд існуючих рішень	02.04.24-08.04.24	
2	Дослідження технології блокчейн	09.04.24-16.04.24	
3	Вибір технологій та інструментів для розробки	17.04.24-22.04.24	
4	Розробка програмних модулів	23.04.24-06.05.24	
5	Запуск та тестування програмних модулів	07.05.24-23.05.24	
6	Оформлення матеріалів кваліфікаційної роботи	24.05.24-03.06.24	
7	Подання кваліфікаційної роботи керівникові та її попередній захист	04.06.24-07.06.24	
8	Подання кваліфікаційної роботи на	08.06.24-12.06.24	

Дата видачі завдання 01 квітня 2024 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

доц. Шматко О.В.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи містить: 91 сторінки, 20 рисунків, 3 таблиці, 31 джерело.

МЕДИЧНІ ІНФОРМАЦІЙНІ СИСТЕМИ, БЛОКЧЕЙН, ETHEREUM, SMART-CONTRACT.

Об'єкт дослідження включає в себе системи передачі медичної інформації, які забезпечують обмін даними між медичними закладами, спеціалістами та пацієнтами, забезпечуючи конфіденційність, цілісність та доступність цих даних.

Предметом дослідження є програмні компоненти, необхідні для створення і підтримки захищених систем передачі медичної інформації. Ці компоненти включають в себе програмне забезпечення для шифрування, аутентифікації, авторизації, а також механізми для забезпечення відмовостійкості та відновлення даних.

Метою даного дослідження є підвищення конфіденційності медичних даних, їх цілісності та доступності, а також забезпечення надійного обміну цими даними між медичними закладами та іншими учасниками за рахунок проєктування та розробки програмних компонентів для створення захищених систем передачі медичної інформації.

Розробка таких систем може мати значний вплив на ефективність медичного обслуговування, зокрема швидкість доступу до медичної інформації, точність діагностичних даних та загальну координацію лікування пацієнтів.

ABSTRACT

Master's thesis: 91 pages, 20 figures, 3 tables, 31 sources.

MEDICAL INFORMATION SYSTEMS, BLOCKCHAIN, ETHEREUM,
SMART CONTRACT.

The object of research includes medical information transmission systems that provide data exchange between medical institutions, specialists and patients, ensuring the confidentiality, integrity and availability of this data.

The subject of the study is the software components required to create and maintain secure health information transmission systems. These components include software for encryption, authentication, authorization, as well as mechanisms for fault tolerance and data recovery.

The purpose of this study is to increase the confidentiality of medical data, its integrity and availability, as well as to ensure the reliable exchange of this data between medical institutions and other participants by designing and developing software components to create secure health information transmission systems.

The development of such systems can have a significant impact on the efficiency of medical care, including the speed of access to medical information, the accuracy of diagnostic data, and the overall coordination of patient care.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	8
ВСТУП	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	11
1.1 Загальний опис предметної області	11
1.2 Огляд технологій електронного документообігу	13
1.3 Аналіз існуючих систем	15
1.4 Постановка задачі дослідження.....	18
2 ДОСЛІДЖЕННЯ ТЕХНОЛОГІЇ БЛОКЧЕЙН ТА ЇЇ ЗАСТОСУВАННЯ ДЛЯ ОБМІНУ МЕДИЧНИМИ ДАНИМИ	21
2.1 Основні поняття	21
2.1.1 Ethereum	22
2.1.2 Смарт-контракти	23
2.1.3 Блок та ланцюг блоків	24
2.2 Алгоритм консенсусу PoW	24
2.3 Інтернет речей IoT.....	25
2.4 Метод обміну та управління даними на основі блокчейн	28
2.5 Система обміну даними.....	36
3 ПРОЄКТУВАННЯ ПРОГРАМНИХ КОМПОНЕНТІВ СИСТЕМИ ЗБЕРІГАННЯ ТА ОБМІНУ МЕДИЧНИМИ ДАНИМИ	38
3.1 Моделювання процесів.....	38
3.2 Системна архітектура	39
3.3 Представлення моделі системи.....	42
3.3.1 Дані, зібрані з вузлів Інтернету речей.....	42
3.3.2. Створення блокчейн для кожного пацієнта в мережі Ethereum.....	43
3.3.3 Функція дозвілу лікаря на доступ до блоку	44
3.4 Смарт-контракт	44
3.5 Моделювання програмних компонентів системи.....	45

3.5.1 Діаграма класів	45
3.5.2 Діаграма послідовності.....	48
3.5.3 Діаграма варіантів використання	50
3.5.4 Діаграма активності	51
3.5.5 Діаграма розгортання	53
3.6 Інструментальні засоби розробки.....	54
3.6.1 Ganache.....	54
3.6.2 MetaMask.....	54
3.6.3 Web3	55
3.6.4 Truffle.....	55
3.6.5 VS Code	56
4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ	57
4.1 Розробка алгоритмів	57
4.1.1 Робота з даними пацієнта та лікаря.....	57
4.1.2 Процес отримання доступу до пропонованої системи (серверна частина)	65
4.2 Аналіз отриманих результатів	70
ВИСНОВКИ.....	76
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	78
ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	82

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

HER – електронні медичні записи

API – інтерфейс програмування додатків

RC – договір реєстратора

PPR – договір про взаємодію між пацієнтом і постачальником послуг

SC – підсумковий контракт

KSI – інфраструктура підпису без ключа

DeFi – децентралізовані фінансові сервіси

FHIR – ресурси швидкої взаємодії в сфері охорони здоров'я

URL – унікальний ідентифікатор посилання

URI – уніфікований ідентифікатор ресурсу

BTC – біткоїн

ETH – етеріум

EVM – віртуальна машина етеріума

PoW – алгоритм доказу роботи

PoS – алгоритм доказу власності

UML – уніфікована мова моделювання

ВСТУП

На сьогоднішній день безпека особистих даних стала однією з найбільш актуальних та важливих проблем для кожного [1]. Навіть якщо інформація не вважається значущою, ніхто не бажає її поширювати. Безпека цих даних стає особливо важливою для пацієнтів у критичному стані.

Якщо зловмисник отримає доступ до цих даних, він може завдати пацієнту шкоди по-різному. Цього не повинно траплятися. Якщо дані пацієнта будуть змінені, лікар може надати неправильне лікування.

Саме тому забезпечення безпеки є критичним. Наприклад, якщо хтось намагається втрутитися у важливі дані, зміна ідентифікатора призведе до зміни хеш-коду блоку [2]. Це ускладнить завдання зловмиснику, оскільки він буде змушений змінювати хеш-код кожного блоку по черзі.

Пацієнти в критичному стані не можуть легко відвідувати лікарів, що створює для них труднощі. Тому система дистанційного моніторингу пацієнтів була б дуже корисною для них. Це також має велике значення для літніх людей, які не можуть безпосередньо звернутися до лікарів.

Впровадження технології блокчейн значно покращило систему безпеки [3]. Зараз єдину базу даних використовується для зберігання конфіденційних даних пацієнтів та інших важливих даних, що робить їх більш безпечними. Технологія блокчейн отримала значний розповсюдження в сфері охорони здоров'я через свою важливість у вирішенні проблем взаємодії та безпеки з системами електронних медичних записів.

Завдяки своїй децентралізованості, блокчейн заслуговує довіри. Він продемонстрував свій потенціал у різних системах електронної охорони здоров'я, зокрема у відкритому обміні електронними медичними записами та спільному управлінні даними між різними системами охорони здоров'я [4].

Люди все більше зацікавлені в електронній охороні здоров'я, яка представляє собою нову технологію в медичній науці. Використання Інтернету для надання медичних послуг та інформації про пацієнтів має

велике значення.

Охорона здоров'я в Україні потребує змін, і технологія блокчейн може допомогти у їх реалізації, забезпечуючи відповідність вимогам ефективності, безпеки, новаторства і економічної вигоди. Але поки що відсутні ефективні методи управління медичними даними на основі блокчейн-технологій.

Тому розробка таких методів є актуальною проблемою. Наше дослідження охоплює системи передачі медичної інформації, що забезпечують обмін даними між медичними закладами, спеціалістами та пацієнтами, забезпечуючи конфіденційність, цілісність та доступність цих даних.

Об'єкт дослідження включає в себе системи зберігання та обміну медичної інформації на основі хмарних технологій, які забезпечують обмін даними між медичними закладами, спеціалістами та пацієнтами, забезпечуючи конфіденційність, цілісність та доступність цих даних.

Предметом дослідження є програмні компоненти, необхідні для створення і підтримки захищених систем зберігання та обміну медичної інформації. Ці компоненти включають в себе програмне забезпечення для шифрування, аутентифікації, авторизації, а також механізми для забезпечення відмовостійкості та відновлення даних.

Мета нашого дослідження - підвищення конфіденційності, цілісності та доступності медичних даних через створення захищених систем зберігання та обміну цих даних. Для цього необхідно вирішити кілька ключових завдань, таких як аналіз існуючих підходів до використання технології блокчейн у медичній сфері, проектування архітектури системи передачі медичної інформації на основі блокчейн, розробка смарт-контрактів та механізмів шифрування та автентифікації.

Вирішення цих задач допоможе нам створити систему, яка забезпечить надійний та безпечний обмін медичною інформацією на основі блокчейн, що покращить медичну інфраструктуру та захистить пацієнтів і медичні установи.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Загальний опис предметної області

Діагностичні дані пацієнтів збільшуються, кількість МІС зростає. Як результат, кількість інформації про пацієнта збереженої на цифрових носіях – збільшується. Потреба у функціональному змісті цих систем також зростає.

Незважаючи на те, що системи ІМС зараз широко застосовуються в Україні, їх область застосування наразі включає лише первинну медицину. Значна частина лікарів досі використовують паперові носії у своїй роботі. У 2018 році в Україні буде впроваджена електронна система охорони здоров'я eHealth, яка гарантує права пацієнтів щодо якості медичних послуг та оптимізує взаємодію між пацієнтом та лікарем шляхом автоматизації, медичного обліку та електронного управління медичною інформацією.

eHealth складається з центральної бази та великого розмаїття МІС. Центральна база даних України забезпечує прозорість витрат охорони здоров'я та можливість руху інформації без паперів з поступовим переходом до електронного обліку, включаючи електронні рецепти, електронні картки та електронні довідки, створювати нові електронні послуги, створення ділового середовища, створення інноваційних продуктів у медицині та просування медичного ринку ІТ загалом.

Завдяки швидкому доступу до всієї інформації про пацієнта, лікар отримує цілісне уявлення про ваше здоров'я, а повний анамнез в електронній формі допомагає поставити правильний діагноз. Більшість медичних послуг можна отримати, не виходячи з дому. Модель фінансування медичних закладів з допомогою системи eHealth кардинально змінилася. Діє принцип – «гроші йдуть за пацієнтом». Система дозволяє контролювати ефективність, з якою витрачаються державні кошти. Перш за все, eHealth охопить основну допомогу: лікарі загальної практики, терапевти та педіатри. Пацієнти підписують заяви від вибраних лікарів, а лікарі реєструють їх у системі. Це

допоможе державі оплатити роботу лікаря з кожним пацієнтом. У 2019 році Україна перебувала на фінішній прямій впровадження першої частини електронної системи охорони здоров'я, включаючи завершення реєстрації медичних установ та майже підписання заяв лікарів та пацієнтів. Планується відновити репутацію медичного персоналу на додаток до збільшення заробітної платні, завдяки:

- оновлення парку машин;
- проведення додаткових тендерів для закупівлі інфраструктурного обладнання;
- автоматизація диспетчерських служб.

Як наслідок медичної реформи Україна повинна охоплюватись розгалуженою мережею сучасних центрів контролю. Це означає, що на дзвінки відповідають швидше, а служби швидкої допомоги приїждять вчасно.

Існують особливості, які сповільнюють впровадження системи, зокрема відсутність унікального ідентифікатора для пацієнтів, неповна взаємодія між національними реєстрами, неповна кількість реєстрів та відсутність фахівців з автоматизації та управління змінами. Наразі використовуються та впроваджуються лише певні елементи eHealth. Досі не існує розробленої та затвердженої концепції та національної стратегії щодо електронної системи охорони здоров'я, архітектури та мандату, яка була чітко затверджена на правовому рівні. Не існує міжнародних узгоджених та затверджених числових стандартів щодо зберігання, обробки та передачі медичних даних, класифікаторів та каталогів.

Слід зазначити, що важливо захищати особисті дані пацієнтів. Вся інформація або спектр інформації про пацієнта, яка занесена до декларації про вибір лікаря, є особистими даними пацієнта (ім'я, дата народження, реєстраційний номер податкового рахунку, номер та порядковий номер). Закон України "Про захист персональних даних" від 1 червня 2010 р. № 2297-VI [1] (далі: закон) регулює правовідносини щодо захисту та обробки персональних даних і спрямований на захист основних прав та свобод.

Відповідно до закону, персональні дані - це інформація або сукупність інформації про особу, яку можна конкретно ідентифікувати. Коли електронна лікарня та електронна медична карта починають працювати в системі, персональні дані обробляються для забезпечення процесу лікування та покращення роботи електронної системи охорони здоров'я. Підписуючи декларацію з лікарем, людина погоджується, що їхні персональні дані є доступними в електронній системі тим лікарям, до яких вони звертатимуться за медичною допомогою. Персональні дані пацієнтів повинні підпадати під дію Закону про медичну конфіденційність та забезпечувати захист цих персональних даних. Медичні працівники зобов'язані жодним чином не розголошувати довірені їм або відомі у зв'язку з виконанням професійних чи службових чи трудових зобов'язань особисті дані, якщо це не передбачено законом. Доступ до даних про пацієнта, що містяться в електронній системі охорони здоров'я, можливий лише за письмовою згодою пацієнта (його законного представника) або на бланку, що дозволяє зробити висновок за згодою.

Без згоди доступ до інформації про пацієнтів можливий лише у таких випадках:

- загроза життю пацієнта;
- згідно рішення суду.

1.2 Огляд технологій електронного документообігу

З початку бурхливого впровадження технологій у світі головну роль відіграли принципи створення, зберігання та передачі даних. Електронні системи управління документами можуть заощадити значний час порівняно з паперовими документами та дати можливість автоматизованої обробки даних. У галузі медичної допомоги ми можемо побачити залежність того, що одужання пацієнта безпосередньо залежить від своєчасного і правильного діагнозу. Слід зазначити, що лікар приділяє значно більше уваги пацієнтам за

наявності електронних систем управління документами, оскільки лікар вже не витрачає багато часу на роботу з документами, а швидше дбає про стан своїх пацієнтів. Електронні системи управління підвищують ефективність. Закон сьогодні встановлює загальні вимоги до інформації та документів в електронній системі охорони здоров'я. Створення, введення, відображення інформації та документів у центральній базі даних, а також внесення змін та доповнень здійснюються користувачами відповідно до прав доступу. Інформація та документи створюються та вносяться до центральної бази даних українською мовою. Рух документів в електронній системі охорони здоров'я здійснюється відповідно до вимог законодавства України «Про електронні документи та електронний документообіг». Всі електронні документи, внесені до центральної бази даних, повинні супроводжуватися електронним підписом автора або підписом, який відповідає власноручному підпису відповідно до закону.

Інформація під час обробки та обміну в електронній системі охорони здоров'я повинна залишатися цілісною. Це забезпечується захистом від несанкціонованих дій, які можуть призвести до випадкової чи навмисної зміни чи знищення, зокрема шляхом накладення на електронний підпис автора або підпису, який відповідає власноручному підпису, відповідно до закону. Державні класифікатори, номенклатури та затверджені каталоги використовуються для введення інформації та документів у центральну базу даних у встановленому законодавством порядку, зокрема класифікаціях та спеціальних списках. Адміністратор надає технічну підтримку для використання таких класифікацій, номенклатур, каталогів та списків. Кожен документ та інформація, внесені до центральної бази даних, автоматично реєструються в єдиному реєстрі. Ідентифікатор особи - це унікальний реєстраційний номер у демографічному реєстрі держави [2]. Слід зазначити, що переваги використання електронного управління документами є вагомими.

Сильними аргументами для електронного управління документами є:

- швидкий пошук медичних даних;
- протидія халатності персоналу, що може призвести до втрати медичних даних;
- шифрування підвищує безпеку;
- зберігання інформації в базі.

Однак системи електронного документообігу мають свої недоліки.

Основними недоліками є:

- значні початкові витрати на впровадження СЕД;
- термін запуску СЕД;
- необхідність підвищення кваліфікації персоналу для роботи з СЕД;
- можливість втрати інформації.

В цілому будь – яка сфера, яка отримує доступ до системи електронного документообігу має значний стрибок у продуктивності праці. Це перехід на якісно інший рівень. Так, впровадження системи дороговартісне, але роблячи крок до впровадження цієї технології , установа швидко окупить її з часом.

1.3 Аналіз існуючих систем

Гіганти цифрової індустрії розуміючи невідворотність проникання інформаційних технологій в заклади охорони здоров'я, вирішили прийняти участь у боротьбі за частину ринку. На цей час впровадженими є Google Health, Microsoft Health Vault та Dossia (організовані Intel, Wal-Mart та AT&T). Співпраця цих важковаговиків дозволить об'єднати усю медичну інформацію та створити єдиний стандарт роботи з діагностичними даними пацієнтів. В подальшому це об'єднання при значних дослідженнях та витратах ресурсів дасть можливість запустити систему штучного інтелекту зі величезною базою знань. Використання цієї технології важко уявити, вона виходить за межі охорони здоров'я як такої.

Медична мережа дозволяє інтегрувати дані з різних систем в особисту

картку пацієнта. Компанії поставили перед собою завдання створити інтегроване мережеве – середовище, в якому зберігається первинна діагностична інформація пацієнта.

Стандарт HL7 підтримується Google Health та Microsoft Health Vault. Це дозволяє вести обмін інформацією між базами різних систем.

На ринку України є державні та приватні медичні центри. Кожен медичний центр підтримує власну базу відвідувань кожного пацієнта, особливо це стосується приватних закладів. Разом з тим пацієнти продовжують мати власні паперові картки з медичною документацією. Звести всю інформацію до єдиної бази даних або хмари здається неможливим. Кожна процедура обміну медичними даними змушує гаяти час, що зменшує шанси пацієнта на виживання під час надання швидкої медичної допомоги. Оптимізація та об'єднання всіх історій хвороб пацієнта, є важливим в першу чергу для держави. Здорова нація – успішна країна.

Усі медичні центри, що працюють в Україні, повинні мати доступ до зведеної бази даних. Наступним завданням, з яким потрібно зіткнутися, є доступ до Інтернету з розумною швидкістю всіх медичних установ. Зрештою, обмін діагностичними даними у великих кількостях при низькій швидкості зв'язку або великому навантаженні мережі може призвести до значної трати часу.

Ще одним вирішенням проблеми може стати впровадження стандартів зберігання на основі CDA (Clinical Document Architecture). Це дозволить мати в базі тільки найважливішу інформацію та посилання на медичні картки в інших центрах охорони здоров'я. Ця мода менше залежить від каналу зв'язку.

Хоча система MIS працює не у всіх сферах медичних установ України перед лікарями стоїть складне завдання заповнити десятки форм. Переважна більшість медичних інформаційних систем призначені для лікарів загальної практики, хоча також існує великий потік медичних даних від лікарів-спеціалістів, який потрібно збирати та обробляти. На сьогодні існує кілька

десятків різних систем, найбільш відомою серед них є "Helsi" [5] (рисунок 1.1).

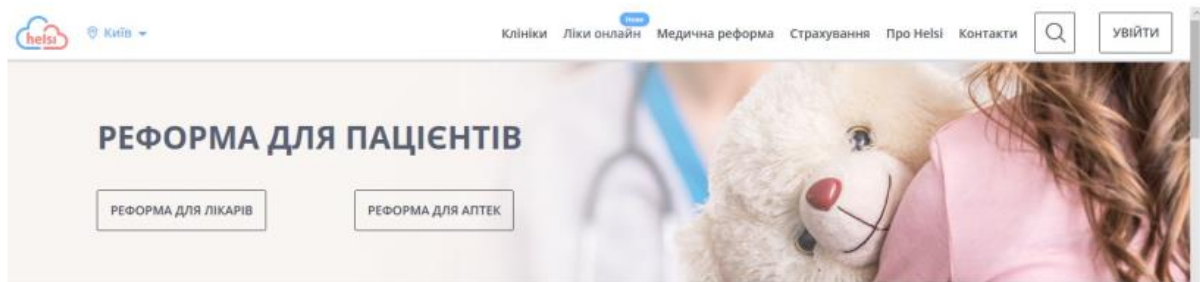


Рисунок 1.1 – Helsi.me

Ця система дуже відома оскільки переважна більшість населення користується державними центрами охорони здоров'я. Доступ до системи пацієнт отримує після підписання декларації з персональним лікарем.

Система має простий інтуїтивний інтерфейс. У випадку якщо клієнт забув пароль, відновити доступ можна за допомогою одноразового паролю через смс. Маю відзначити система іноді самостійно надсилає одноразовий пароль під час входу, як елемент додаткового захисту. Недоліком цього є те, що пацієнт матиме труднощі з входом до системи, якщо загубить телефон.

Система працює з електронним рецептом та створено напрямок страхової медицини. Страхова медицина дозволить залучити додаткові кошти та побороти проблеми з недофінансуванням в галузі охорони здоров'я. За рахунок цього кошти матимуть цільове призначення та відповідно будуть рости і розвиватися ті сервіси та послуги якими користуватимуться пацієнти. Державі лишиться фінансування стратегічних та перспективних напрямків розвитку галузі за потреби. Кожен пацієнт має власний кабінет (рисунок 1.2) де він може.

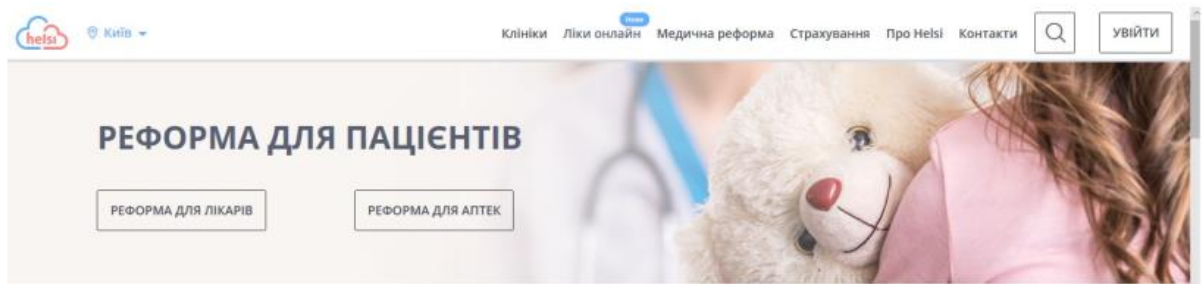


Рисунок 1.2 – Власний кабінет Helsi.me

- редагувати власний профіль;
- переглянути історію візитів за весь час;
- працювати з пов'язаними особами(наприклад : сім'я);
- замовляти ліки ;
- користуватися страховим полісом.

Єдиним на мою думку суб'єктивним недоліком Helsi є обмеженість співпраці зі страховими компаніями. Свій страховий поліс я не зміг долучити, аби протестувати таку можливість. В перспективі я впевнений це питання буде закрито

1.4 Постановка задачі дослідження

Розроблювана система повинна відповідати певним вимогам як по її функціональності, так і по надійності. Тобто ми маємо два набори вимог: системні та функціональні.

До системних вимог можна віднести наступне :

- створення інтуїтивно зрозумілого графічного дизайну;
- авторизація та політика безпеки;
- заповнення зовнішніх форм статистики та збереження діагностичних даних пацієнтів для подальшої обробки;
- тестування та обробка помилок;

- виконання робіт на стороні замовника перед першим запуском;

Технічні та апаратні вимоги:

- ОС Windows 7,8,10;
- вимоги до оперативної пам'яті 4 гб або більше;
- вільне місце на жорсткому диску з базою не менше 200 гб;
- встановлена JVM;
- підготовлена до роботи база даних MS SQL;

Функціональність:

1) Навігація:

- внесення даних до бази та заповнення статистичних звітів;
- пошук у базі даних та аналіз діагностичних даних певним чином;18
- вихід з програми;

2) Внесення даних до бази та заповнення статистичних звітів:

- додавання до бази даних звернень пацієнтів про кожний конкретний

такий випадок;

- автоматизація заповнення та надсилення статистики через сайт;

3) Пошук:

- створення запиту для аналізу інформації в базі даних;
- виведення результату запиту в зручній формі;
- збереження частини результатів роботи користувача за потреби.

Нефункціональні вимоги

а) Безпека:

- дотримання стандартів безпеки та нормативних вимог для захисту

медичних даних;

- шифрування даних з використанням сильних алгоритмів і дотримання нормативів щодо захисту персональних даних;

- доступність:

- гарантована доступність системи не менше 99,9% часу;

- б) резервування і механізми відновлення для забезпечення безперервної роботи системи;

в) Продуктивність:

ефективна обробка та передача медичних даних для забезпечення оперативності та мінімізації затримок. Час відповіді на запит не більше 5 секунд;

г) Відповідність законодавству:

дотримання місцевих і міжнародних законів і стандартів в області медичної інформації та захисту даних;

г) Зручність використання:

інтуїтивно зрозумілий користувальницький інтерфейс для медичних працівників і пацієнтів;

підтримка різних пристроїв і платформ.

Головне призначення запропонованої системи полягає в створенні буферної бази, що допоможе уникнути помилок при заповненні статистичних форм. Це також дозволить зберігати частину результатів і не починати все з початку. Лікарі та молодший медичний персонал зможуть більше часу приділяти пацієнтам а не статистиці. Також враховано факт, що успіх ІТ – проектів в галузі охорони здоров'я сильно залежить від сприйняття користувачем зручності інтерфейсу програмної системи[3], оскільки не всі користувачі маю однаковий досвід роботи з програмними системами.

2 ДОСЛІДЖЕННЯ ТЕХНОЛОГІЇ БЛОКЧЕЙН ТА ЇЇ ЗАСТОСУВАННЯ ДЛЯ ОБМІНУ МЕДИЧНИМИ ДАНИМИ

2.1 Основні поняття

Технологія блокчейн-це сукупність блоків, які з'єднані в незмінну і децентралізовану мережу. У IoT блокчейн працює без допомоги централізованої бази даних. Блокчейн - це така технологія, в якій дані зберігаються блок за блоком, щоразу, коли створюються нові блоки, вони додаються в кінці ланцюга в лінійній процедурі.

Блокчейн - мережа повинна перевірити дійсність транзакції після того, як вона була сертифікована. Надаючи підтвердження транзакції за допомогою комп'ютера, транзакція прив'язується до блокчейну у вигляді блоку (рисунок 2.1). Кожен блок на блокчейн-мережа має чіткий хеш. Хеш розробляється з використанням алгоритму і має вирішальне значення для управління блокчейном.

У випадку маніпулювання будь-якою інформацією блоку, це миттєво змінює хеш-значення блоку. Однак це також призводить до зміни хешу наступного блоку. Це постійний процес, тому при зміні або видаленні будь-яких даних хеш всіх інших блоків змінюється автоматично. Щоб змінити один блок в блокчейні, зловмиснику доведеться змінити кожен окремий блок після поточного. Для перерахунку всіх цих хешів знадобиться величезна кількість обчислювальної потужності. Таким чином, ніхто не може змінити жодної речі в блокчейн-мережі без попереднього повідомлення.

У блокчейн-системі існують два варіанти вузлів: перший - це той, який хоче додати нові блоки в Мережу, а інший - той, який отримує ці нові блоки для створення ланцюжка блоків. Блокчейн складається з блоків, а блоки мають заголовки блоків. І заголовок кожного блоку має свій унікальний хеш, хеш попереднього блоку, корінь Merkle, одноразовий номер, тимчасову мітку [5].

На рисунку 2.1 показано повне уявлення про блок мережі блокчейн.

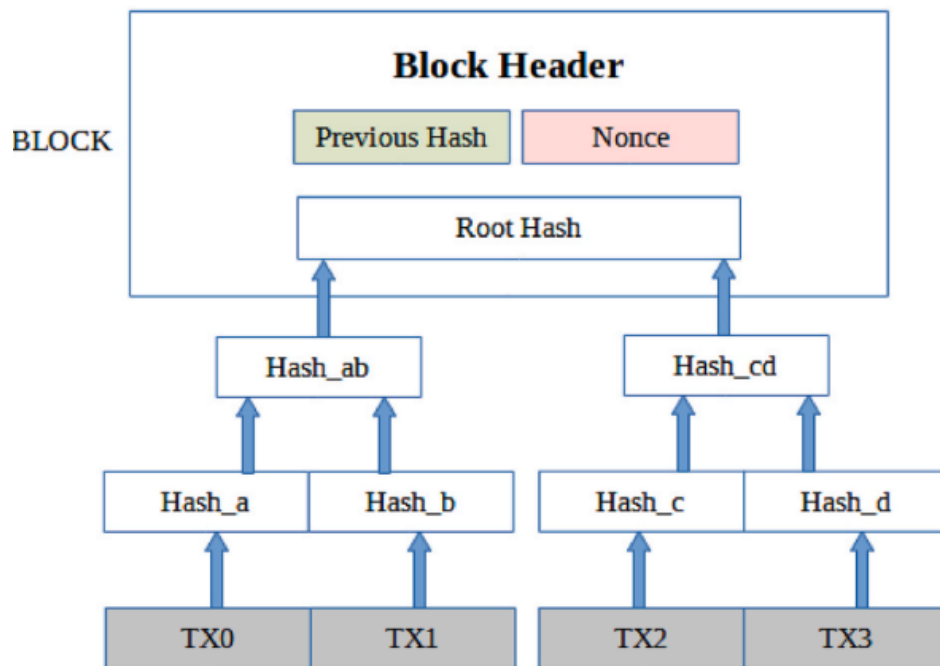


Рисунок 2.1 – Блок в блокчейн

Блоки в блокчейні з'єднані один за іншим лінійним чином, це виглядає як ланцюжок блоків, тому вона і названа блокчейном. Всі транзакції в одному блоці перетворюються в хеш, який називається унікальним хешем блоку. Оскільки блокчейн являє собою ланцюжок блоків, хеш попереднього блоку також зберігається в поточному блоці [6-8].

Зміна однієї транзакції в блоці без згоди Майнера призводить до зміни хешу блоку. Отже, вся система виходить з ладу.

2.1.1 Ethereum

Ethereum-це децентралізована мережа, побудована на технології блокчейн. Вперше він був розгорнутий на популярному блокчейні криптовалюти. Ethereum був створений з метою створення платформи смарт-

контрактів з відкритим вихідним кодом і функціональністю блокчейна. Ця технологія також використовує однорангову мережу для розповсюдження себе. Ця мережа також використовує ефіри, які є її власною криптовалютою. Ethereum також надає програмістам мову під назвою Solidity, яка дозволяє їм створювати власний блокчейн. Він був створений для смарт-контрактів Ethereum, які є останньою функцією. Транзакції-це засіб, за допомогою якого зовнішні організації підключаються до Ethereum. Зовнішні користувачі можуть використовувати його для зміни статусу документа або набору даних у мережі Ethereum Blockchain. Є кілька елементів, що складають транзакцію Ethereum, таких як автор-відправник з 20-байтовою адресою та одержувач, який також має 20-байтову адресу. Незважаючи на те, що існує певна вартість (кількість грошей, переведених з джерела в пункт призначення, і газ), кожна транзакція в мережі блокчейн вимагає оплати автором певних витрат. Газ-це назва, дана цій вартості. Обмеження та витрати на газ включені в кожную транзакцію, а ціна газу - це сума грошей, яку автор транзакції готовий заплатити за газ. На цю угоду можна витратити багато газу [4].

2.1.2 Смарт-контракти

Смарт-контракт - це набір команд, які можуть бути використані для виконання будь-якої транзакції в блокчейне. Коли користувачі надсилають транзакції, запускається цей фрагмент коду. Вони працюють безпосередньо на блокчейні, що робить їх несприйнятливими до маніпуляцій і модифікації. Smart-contracts використовує мову програмування Solidity для програмування будь-якої форми діяльності на блокчейні. Програмісти можуть скомпілювати необхідні операції після того, як вони були запрограмовані. Потім після компіляції вони можуть бути запуснені та розгорнуті на блокчейні Ethereum. JavaScript - це мова програмування, яка реалізує мову Solidity Ethereum для написання коду смарт-контракту.

2.1.3 Блок та ланцюг блоків

IPFS - це технологія розподіленого зберігання даних за допомогою однорангової мережі. Оскільки дані IPFS захищені від змін і забезпечують безпечне зберігання даних, будь-яка спроба змінити дані, збережені в IPFS, може бути виконана лише шляхом зміни ідентифікатора. Отже, він забезпечує криптографічну ідентифікацію для захисту даних від маніпуляцій. Кожен файл даних, що зберігається в IPFS, містить криптографічно згенероване хеш-значення. Він має лише одне значення і використовується для ідентифікації файлів даних, що зберігаються в IPFS. Протокол IPFS використовує однорангове з'єднання (P2P), яке включає об'єкт IPFS, що містить дані та зв'язки. Дані - це масив неорганізованих двійкових значень, тоді як посилання - це неупорядковане двійкове значення. Протокол IPFS функціонує наступним чином: файлам IPFS присвоєно унікальний криптографічний хеш у мережі IPFS дублікати файлів не допускаються.

2.2 Алгоритм консенсусу PoW

Доказ роботи використовується для транзакцій Майнінг. У блоці немає можливості оновити будь-які дані. Однак може виявитися можливим згенерувати всі хеші блоків і будь-яким чином зібрати ланцюжок заново. При використанні підходу до інтелектуального аналізу транзакцій буде потрібно, якщо взагалі можливо, дуже високопродуктивний процесор. За наших обставин ми використовуємо цей метод для перевірки обчислювальної потужності перед транзакціями. Відправник повинен обробити транзакцію після її завершення, коли вузол створює новий блок, що приймають вузли виконують хеш-функцію для вмісту блоку і одноразовий номер, який відомий як Майнінг.

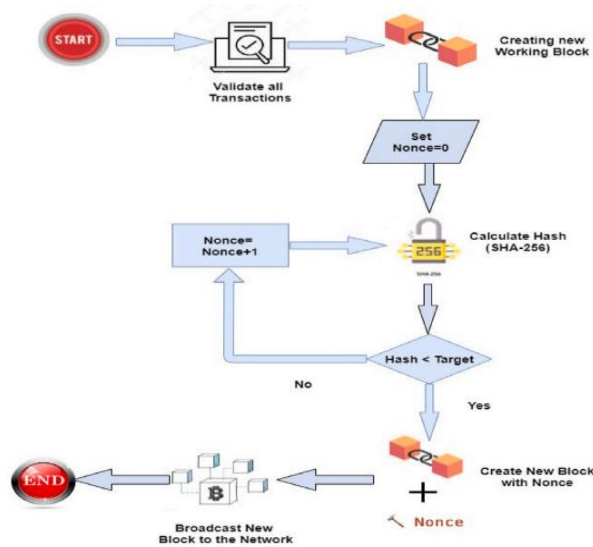


Рисунок 2.2 – Алгоритм PoW

У криптографії одноразовий номер – це довільне число, яке може бути використано тільки один раз в криптографічному повідомленні [9]. Зазвичай це рандомізоване або квазізипадкове число, що використовується в схемі ідентифікації для запобігання використанню попередніх взаємодій при повторних атаках [10]. Зазначене значення має бути вище обчисленого хеш-значення. Якщо ця вимога не виконується, генерується nonce, і хеш - функція виконується повторно за допомогою нового nonce.

Новий блок тепер містить підтвердження своєї роботи, яке буде перевірено інші вузли в мережі, коли майнер завершує свої обов'язки. Однак майнінг - це не проста операція, оскільки комп'ютери повинні виконувати складні програми для вирішення складних математичних виразів. На рисунку 2.2 представлено алгоритм Pow.

2.3 Інтернет речей IoT

Інтернет речей (IoT) - це компактна мережа датчиків і інших пристроїв для зв'язку один з одним через Інтернет.

Інтернет медичних речей (IoMT) - це частина програми ІОТ, яка збирає та аналізує дані для тестування та моніторингу [11]. Ця інфраструктура дозволяє створити цифрову систему охорони здоров'я, яка з'єднує доступні ресурси пацієнтів та медичні послуги. Віддалений моніторинг пацієнтів в даний час можливий тільки з розвитком Інтернету речей. Ці засоби моніторингу стану здоров'я можуть бути самими різними - від датчиків, що носяться, до різних складних пристроїв..

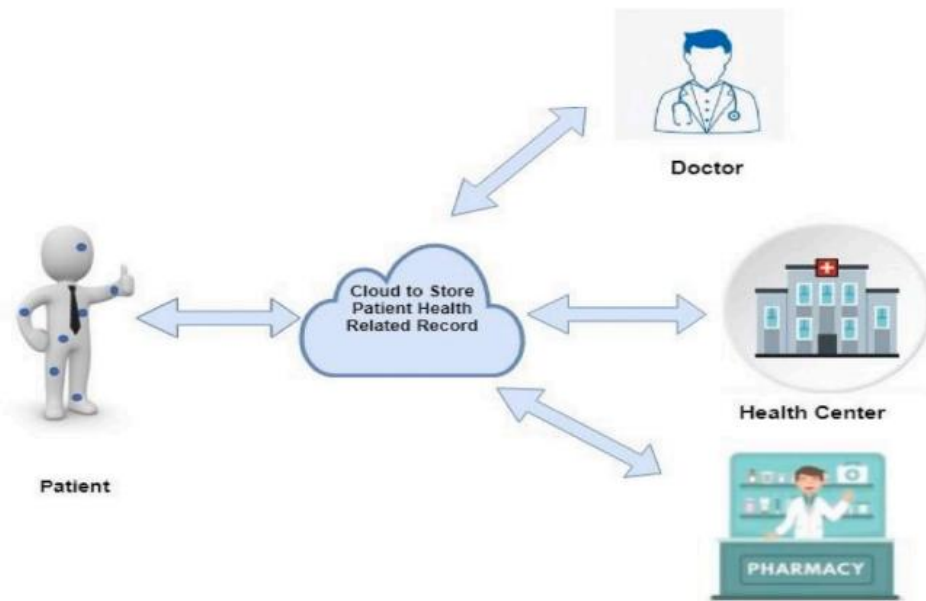


Рисунок 2.3 – Інтернет медичних речей (IoMT)

Недавня статистика показує, що в секторі охорони здоров'я існує близько третини пристроїв Інтернету речей, і аналітики прогнозують, що протягом 2025 року їх кількість збільшиться, що є ознакою цифровізації нашого сектору охорони здоров'я [12]. Стрімкий розвиток цього сектора не тільки робить наше життя комфортним, але і збільшує середню тривалість життя. Але з часом рішення IoMT стикаються з труднощами, пов'язаними з безпекою та крихкістю конфіденційності.

На рисунку 2.3 показано процедура обміну даними через Інтернет речей. В даний час даними про стан здоров'я пацієнтів можна легко

обмінюватися між лікарями, аптеками та іншими третіми особами у співпраці з хмарними обчисленнями.

У роботі [13] автори використовували блокчейн для розширення можливостей хмарних обчислень. Вони створили систему радіаційної онкології, призначену в основному для онкологічних хворих. Прототип побудований на базі Hyperledger Fabric як реалізація технології блокчейн з відкритим вихідним кодом. Тут дані завантажуються в хмару, але в цьому прототипі немає реалізації Інтернету речей, тому всі дані пацієнта повинні бути відправлені вручну. Їх архітектура складається з інтерфейсу користувача та серверної частини, що складається зі служби членства та центру сертифікації, мережі вузлів, засобів балансування навантаження для перенаправлення користувача на будь-який з довірених вузлів мережі, окремого хмарного сховища даних пацієнта і сертифікатів.

У роботі [14] автори запропонували захищену електронну медичну карту (EHR) спільне використання мобільних хмарних систем електронної охорони здоров'я з використанням блокчейна, вони використовують блокчейн-платформу Ethereum для побудови системи електронної охорони здоров'я, їх прототипом в основному є мобільний моніторинг пацієнтів. Крім того, вони використовували систему AWS для хмарної системи, що дуже вражає, але якби вони використовували приватний блокчейн, їх система безпеки стала б більш безпечною, ніж нинішня.

У [15] автори зробили досить подібну роботу, порівнявши дві вищенаведені. Вони створили прототип управління доступом до даних електронної охорони здоров'я за допомогою блокчейну Ethereum. Ethereum - це, в основному, публічний блокчейн, який використовується для управління доступом до даних.

Вони також використовували мову solidity для програмування смарт-контрактів і IPFS як база даних для зберігання файлів. Щоб завершити збір даних, вони використовували багато датчиків і Raspberry pi.

2.4 Метод обміну та управління даними на основі блокчейн

В процесі управління медичними даними з використанням блокчейн-технологій медичні дані будуть розташовуватися в блоках блокчейна. Тому, по-перше, розглянемо структуру блокчейн-блоку. Враховуючи, що заголовок блокчейн- блоку містить дату і час, версію, метадані, цифрові підписи сторін, власний зашифрований код, хеш попереднього блоку, ми представляємо такий блок наступним чином. Представимо блокчейн-блок (рис.2.4) у формалізованому вигляді - у вигляді кортежу його елементів:

$$B = \langle H, D \rangle = \langle (d, t, v, md, ec, ds, hb), D \rangle \quad (2.1)$$

де H - заголовок блоку,

D - дані блоку,

d - дата,

t - час,

v - версія,

md - метадані,

ec - власний зашифрований код,

ds - цифрові підписи сторін,

hb - хеш попереднього блоку.

Блокчейн - це блок. Кожен блок пов'язаний з попереднім і наступним блоками. Блоки нерозривно пов'язані один з одним, що виключає можливість зміни блоку або вставка між двома блоками. Маючи це на увазі, ми представляємо фрагмент блокчейна в наступному вигляді (2.2).

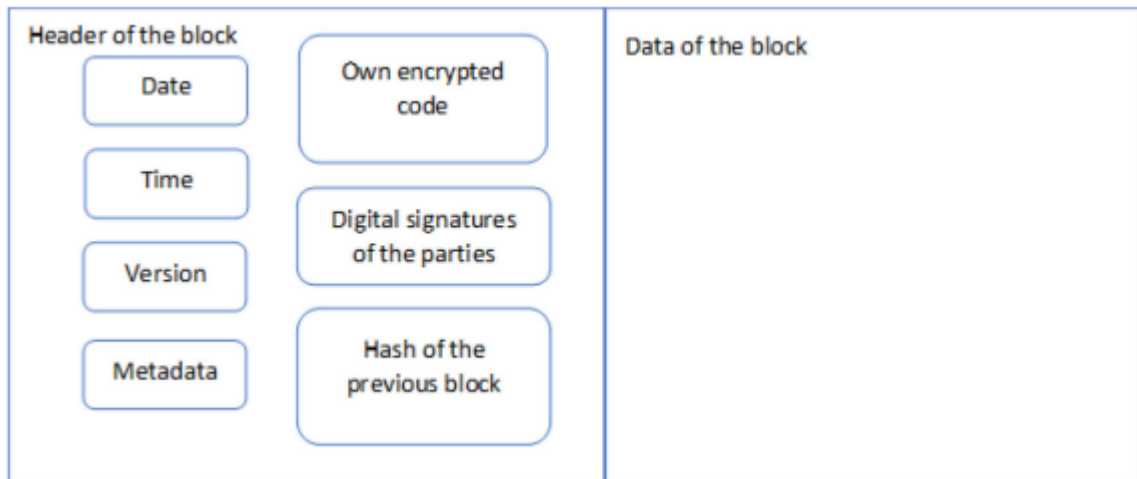


Рисунок 2.4 – Блок мережі блокчейн

Потім ми представляємо наступний фрагмент блокчейна у формалізованому вигляді-у вигляді набору кортежів, які представляють блоки блокчейна:

$$\begin{aligned}
 BCH = \{B_1, B_2, \dots, B_{n-1}, B_n, B_{n+1}, \dots\} = \{ < (d_1, t_1, v_1, md_1, ec_1, ds_1), D_1 > \\
 < (d_2, t_2, v_2, md_2, ec_2, ds_2, hb_1), D_2 > \dots, \\
 < (d_{n-1}, t_{n-1}, v_{n-1}, md_{n-1}, ec_{n-1}, ds_{n-1}, hb_{n-2}), D_{n-1} > \\
 < (d_n, t_n, v_n, md_n, ec_n, ds_n, hb_{n-1}), D_n > \\
 < (d_{n+1}, t_{n+1}, v_{n+1}, md_{n+1}, ec_{n+1}, ds_{n+1}, hb_n), D_{n+1} >, \\
 \dots\}
 \end{aligned} \quad (2.2)$$

де номер елемента hb вказує на те, що це хеш попереднього блоку.

З огляду на особливості побудови блокчейна, слід оцінити достатність медичних даних, перш ніж вводити їх в розділ "дані блоку" блокчейн-блоку.

Враховуючи роботи [3, 4], головною умовою для цього є розуміння того, які обов'язкові елементи повинні бути представлені у цих даних, оскільки достатність медичних даних буде означати наявність всіх необхідних інформаційних елементів.

Враховуючи, що дані блоку D є набором записів: $D = \{d_1, d_2, \dots, d_k\}$, і окремий запис в наборі медичних даних сприймається як система з трьох

елементів (тип, час і якість) $d_i = \langle dtp_i, dtm_i, dq_i \rangle$, потім доступний набір медичних даних, які планується ввести в блок блокчейна може бути представлений у вигляді матриці.

$$D_{av} = \begin{bmatrix} dtp_1 & dtm_1 & dq_1 \\ dtp_2 & dtm_2 & dq_2 \\ \dots & \dots & \dots \\ dtp_i & dtm_i & dq_i \\ \dots & \dots & \dots \\ dtp_k & dtm_k & dq_k \end{bmatrix} \quad (2.3)$$

де dtp_i - тип даних (тип даних може бути динамічним - наприклад, аналіз крові, або статичним-геном, відбитки пальців);

dtm_i - час отримання даних, дата, коли ці дані були отримані;

dq_i - якість даних, яка враховує "термін придатності" інформації або аналізу - чим він вищий, тим цінніше інформація; наприклад, результат тесту на холестерин дійсний протягом шести місяців, а генетичний тест-все життя;

$i=1..k$, k -кількість записів у наборі медичних даних, які готуються до введення в блокчейн-блок.

Елементи dtp_i , dtm_i , dq_i можуть бути відсутні, якщо медичних даних у наборі недостатньо. Враховуючи, що всі елементи медичних даних, перераховані в матриці, є обов'язковими з точки зору адекватності медичних даних, правила визначення достатності медичних даних наступні:

якщо в i -му записі даних заданого D_{av} є інформація про тип даних (доступний елемент dtp_i), то: $sntp = snip + 1$ і $b[i, 1]=1$; в іншому випадку, $b[i, 1]=0$;

якщо в i -му записі даних встановленого D_{av} є інформація про час даних (доступний елемент dtm_i), то: $sntm = sntm + 1$ і $b[i, 2] = 1$; в іншому випадку, $b[i, 2]=0$;

- якщо в i -му записі даних заданого D_{av} є інформація про якість даних (доступний елемент dq_i), то: $sdq = sdq + 1$ і $b[i, 3]=1$; в іншому випадку, $b[i,$

$3]=0$.

Враховуючи особливості формування матриці B (2.1), правила формування рекомендацій по доповненню набору медичних даних наступні:

якщо $b[i, 1]=0$, то Користувачеві рекомендується доповнити i -й запис даних блоку інформацією про тип даних;

якщо $b[i, 2]=0$, Користувачеві рекомендується доповнити i -й запис даних блоку інформацією про час передачі даних;

- якщо $b[i, 3]=0$, Користувачеві рекомендується доповнити i -й запис даних блоку інформацією про якість даних.

Метод оцінки достатності медичних даних складається з наступних етапів:

а) аналіз набору медичних даних D_{av} перед введенням їх в розділ "дані блоку" в блокчейн з використанням кожного з розроблених правил визначення достатності медичних даних і підрахунку лічильників sdt_p , sdt_m , sdq ;

б) Розрахунок кількісної оцінки достатності медичних даних за формулою

$$sfm = \frac{1}{3} \left(\frac{sdt_p}{k} + \frac{sdt_m}{k} + \frac{sdq}{k} \right) \quad (2.4);$$

в) якщо $sdt_p=k$, $sdt_m=k$ і $sdq=k$ і $sfmd=1$, то приймається рішення про достатність медичних даних в наборі D_{av} (оскільки всі елементи набору медичних даних є обов'язковими з точки зору достатності медичних даних і доступні в наборі медичних даних, які готуються для введення в D_{av}) і введення даних набору D_{av} в розділі "дані блоку" блоку блокчейна;

г) якщо $sdt_p < k$ або $sdt_m < k$ або $sdq < k$ або $sfmd < 1$, то приймається рішення про недостатність медичних даних в наборі D_{av} ; такий набір медичних даних необхідно доповнити перед введенням цих даних в розділ "дані блоку" блокчейн-блоку.

Користувачеві надаються рекомендації щодо доповнення набору

медичних даних - відповідно до Правил формування рекомендацій щодо доповнення набору медичних даних - в якості керівництва про те, яку інформацію слід додати до набору медичних даних D_{av} ; відбувається доповнення до набору медичних даних; відбувається повернення до кроку 1 методу.

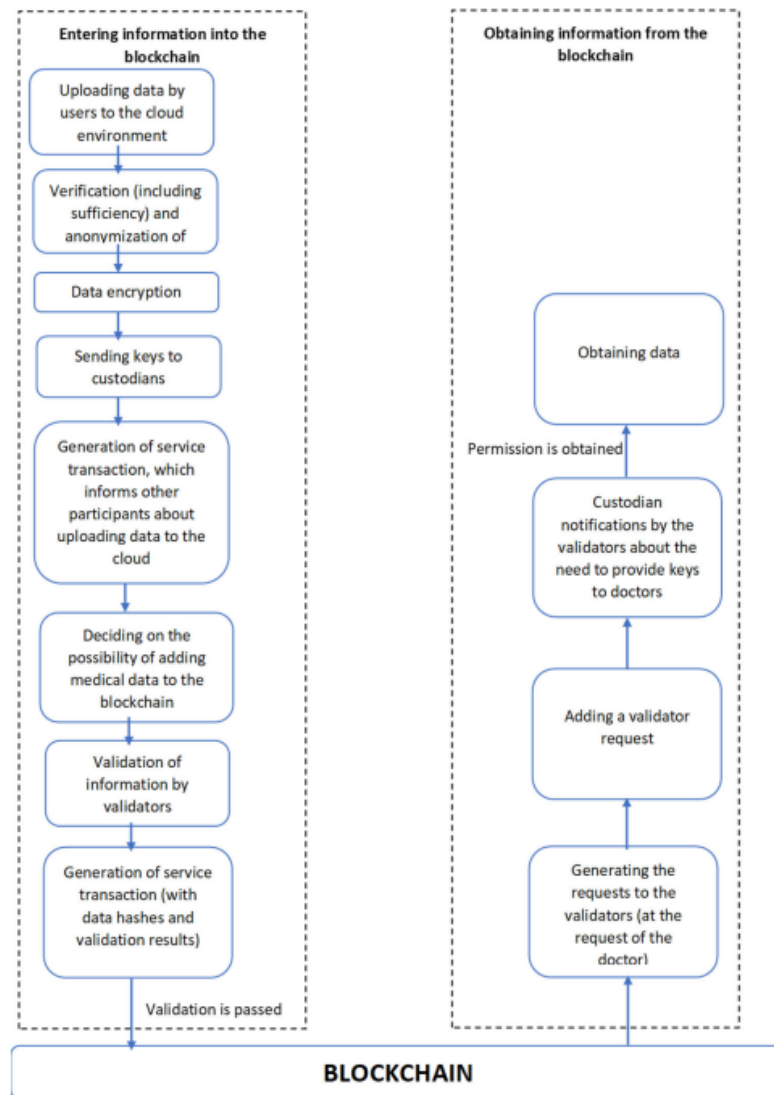


Рисунок 2.5 – Схема обробки транзакцій в блокчейн

Розроблений метод оцінки достатності медичних даних забезпечує можливість аналізу сукупності медичних даних на предмет їх достатності; можливість кількісної оцінки достатності медичних даних; в разі

недостатності медичних даних в наборі формування рекомендацій по доповненню набору медичних даних - в якості керівництва, яку інформацію слід додати в набір медичних даних, щоб мати можливість завантажувати їх в розділ "блок даних" блокчейна.

Тепер розглянемо особливості транзакцій при роботі з медичними даними. Медична екосистема надає чотири групи користувачів: користувачі (пацієнти) - самі пацієнти або уповноважені треті особи (наприклад, медичні центри), які завантажують медичні дані в систему; валідатори - перевіряють якість і справжність даних, завантажуваних користувачами; клієнти (лікарі) – вивчають дані пацієнтів, складають медичні звіти; фармацевтичні та дослідницькі компанії, які можуть отримати доступ до знеособлених даних користувача.

Метод виконання транзакцій з медичними даними складається з наступних кроків (рис.2.5):

Етап I-введення інформації в блокчейн:

- завантаження даних користувачами (пацієнтами) в хмарне середовище;
- анонімізація (видалення ідентифікатора пацієнта) даних;
- шифрування даних за допомогою симетричного шифрування;
- відправка ключів зберігачам за прямими аутентифікованими канал зв'язку;
- генерація сервісної транзакції (з відкритим ключем, інформацією про тип даних і посиланням на дані в хмарному середовищі), яка інформує інших учасників процесу про завантаження даних у хмару;
- прийняття рішення про можливість додавання медичних даних у блокчейн з використанням алгоритму консенсусу (блок порівнюється з кожним екземпляром реєстру; якщо всі екземпляри реєстру однакові, транзакція стверджується);
- перевірка інформації валідаторами;
- генерація сервісної транзакції (з хешами даних і результатами

перевірки);

- якщо дані верифіковані, інформація записується в блокчейн, і інформація про пацієнта стає доступною іншим учасникам, в іншому випадку користувачеві відмовляють у передачі інформації в блокчейн.

Етап II-отримання інформації з блокчейна:

- якщо лікарю потрібні певні дані для дослідження або постановки діагнозу, то він генерує запит до валідаторів

- додавання запиту валідатора в блокчейн

- повідомлення зберігача від валідаторів про те, що для розшифровки хмарних даних лікарю необхідно відправити криптографічні ключі

- отримання даних - якщо отримано дозвіл від валідаторів і зберігачів.

Тому був розроблений метод виконання транзакцій з медичними даними, який складається з етапів:

- введення інформації в блокчейн і отриманні

- інформації з блокчейна.

Розроблений метод виконання транзакцій з медичними даними забезпечує можливість прийняття рішення про додавання транзакцій і медичних даних від пацієнта в блокчейн; можливість прийняття рішення про видачу даних з блокчейна за запитом лікарів.

Грунтуючись на аналізі архітектур відомих систем управління медичними даними, заснованих на блокчейн-технологіях, розробимо архітектуру управління медичними даними. Система, заснована на розроблених методах оцінки достатності медичних даних і виконання транзакцій з медичними даними. Архітектура системи управління медичними даними представлена на рисунку 2.6

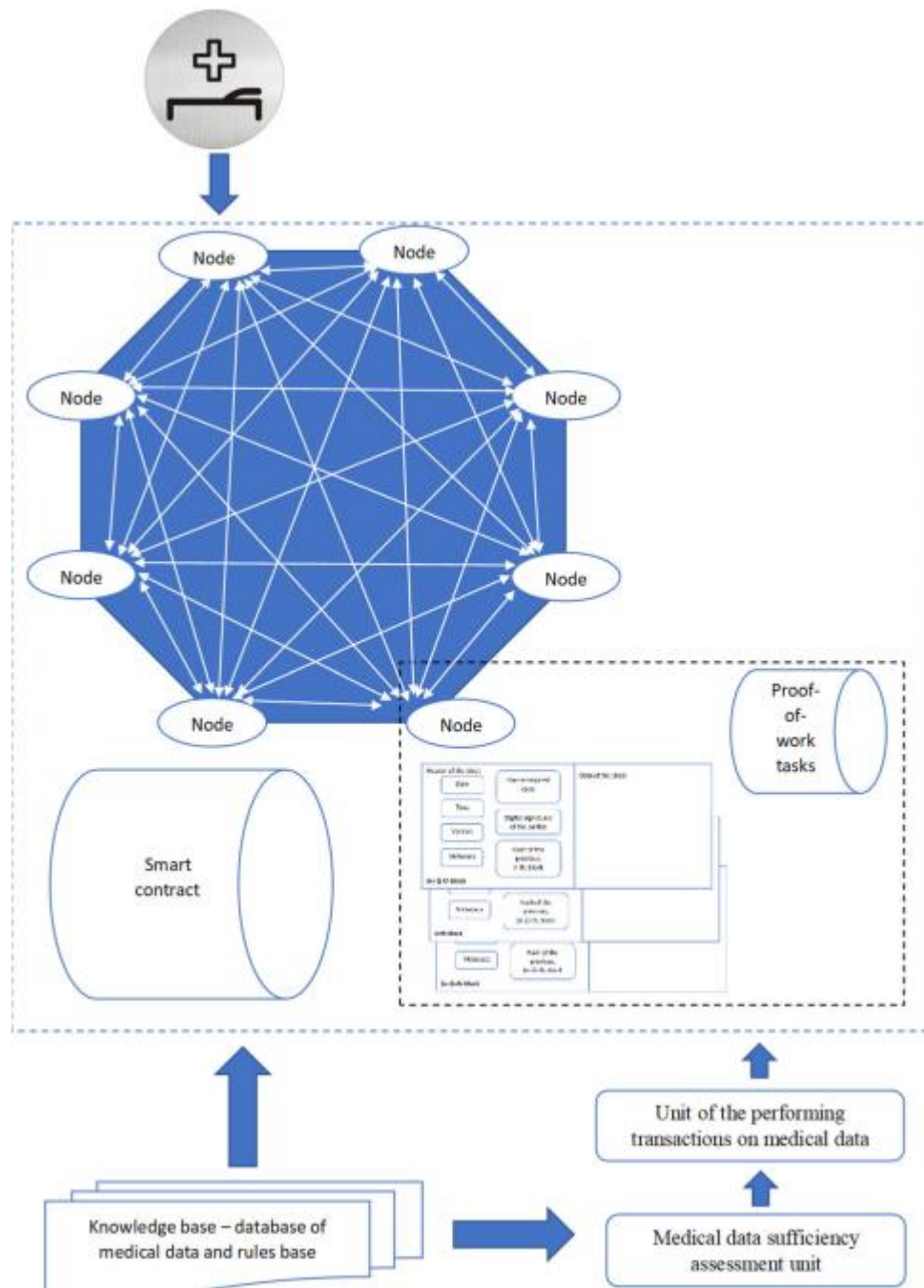


Рисунок 2.6 – Архітектура системи

Грунтуючись на аналізі архітектур відомих систем управління медичними даними, заснованих на блокчейн-технологіях, розробимо архітектуру управління медичними даними. Система, заснована на розроблених методах оцінки достатності медичних даних і виконання транзакцій з медичними даними. Архітектура системи управління медичними даними представлена на рисунку 2.6

2.5 Система обміну даними

Традиційний метод використання технології Інтернету речей полягає в зберіганні всіх даних в хмарі. Безпека цих хмарних даних в даний час є джерелом занепокоєння. Користувачі повинні бути в курсі того, хто має доступ до своїх хмарних даних. Ми хочемо також зберігати дані пацієнтів в хмарі, але з дотриманням надійних процедур безпеки. Деякі дані є одночасно важливими та конфіденційними, і вони повинні надійно зберігатися в хмарі, а пацієнти повинні мати відповідний контроль над тим, хто може отримати до них доступ. В результаті необхідно забезпечити безпеку цих даних в хмару. Ми використовуємо добре відому блокчейн технологію для цього аспекту безпеки. Існує безліч доступних блокчейн-систем; ми вибрали Ethereum через його чудові стандарти безпеки. Нижче наведені деякі з найбільш важливих причин для використання

Технологія Ethereum:

Ethereum-це велика усталена мережа, яка протягом багатьох років піддавалася випробуванням завдяки численним заходам і значної цінності. В в результаті це безпечно;

ключова причина, по якій ми вибираємо Ethereum для нашої системи, полягає в тому, що він володіє безліччю функціональних можливостей, відмінно підходить для використання смарт-контрактів і може безпечно зберігати дані децентралізованим способом;

- він надзвичайно популярний і має сильну спільноту, яка постійно шукає нові методи для поліпшення технології.

Враховуючи всі ці переваги, ми створили смарт-контракт на базі Ethereum для спілкування пацієнта з лікарем. Цей смарт-контракт дозволяє лікарям просто відстежувати своїх пацієнтів. Весь процес може бути розбитий на декілька етапів. У цьому розділі ми розглянемо кожен з цих пунктів по черзі. На рисунку 2.7 зображено візуальне представлення центральної концепції нашої системи, яка розділена на дві частини: IoT і

блокчейн. Датчики збирають дані пацієнта, які надійно шифруються і зберігаються в хмарі.

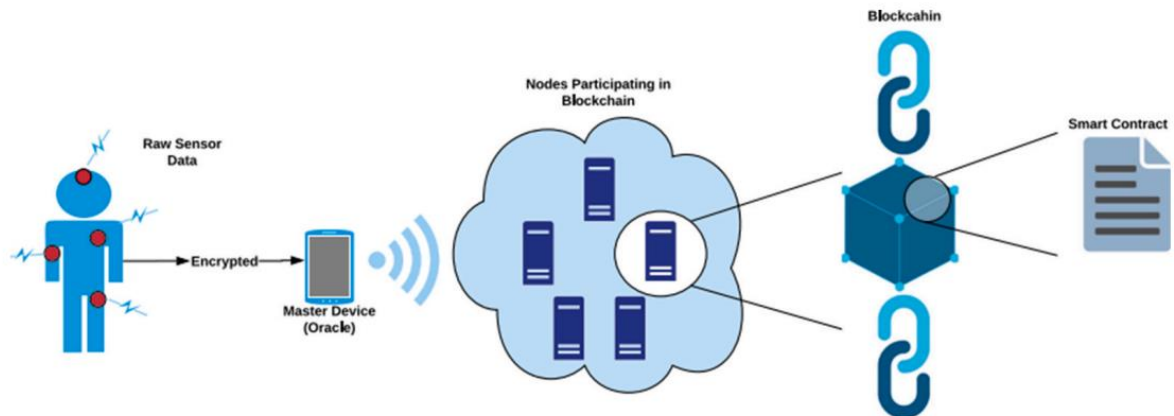


Рисунок 2.7 – Системна архітектура

Потім хмарна адреса додається до блокчейну. Як результат, система набагато безпечніша, ніж існуючі медичні рішення Інтернету речей. Не зберігаючи всі значення датчиків у блокчейні, ми зберігаємо адресу в блокчейні. Це робить систему більш економічною.

3 ПРОЄКТУВАННЯ ПРОГРАМНИХ КОМПОНЕНТІВ СИСТЕМИ ЗБЕРІГАННЯ ТА ОБМІНУ МЕДИЧНИМИ ДАНИМИ

3.1 Моделювання процесів

Проектування моделі процесів.

IDEF0 (Integration Definition for Function Modeling) – нотація опису бізнеспроцесів. Заснована на методології SADT.

SADT (Structured Analysis and Design Technique, технологія структурного аналізу і проектування) - графічні позначення і підхід до опису систем. Розробка SADT почалася в 1969 році і була випробувана на практиці в компаніях різних галузей (аерокосмічна галузь, телефонія і т.д.). Публічно з'явилася на ринку в 1975 р і отримала дуже широке поширення в світі.

IDEF0 є результатом програми комп'ютеризації промисловості, яка була запропонована ВВС США. Автоматизація діяльності підприємств зажадала відповідних методик та інструментів. Перед тим, як розробляти програмні забезпечення, необхідно було чітко і зрозуміло описати бізнес-процеси (не можна автоматизувати хаос). Нотація може бути використана для моделювання широкого кола автоматизованих і неавтоматизованих систем.

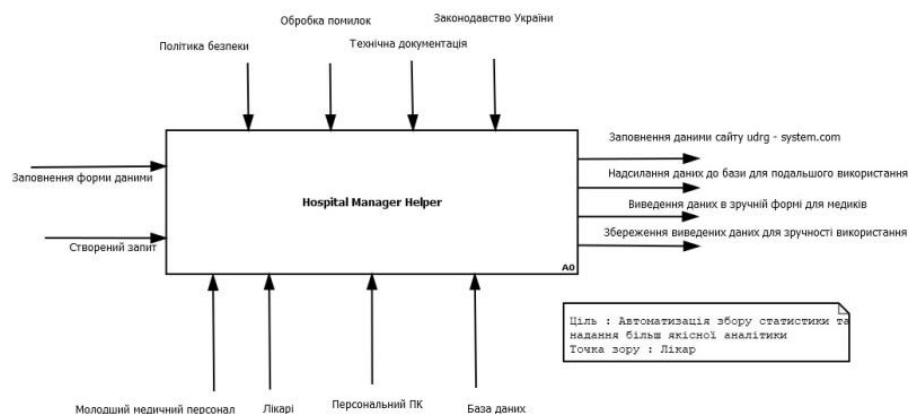


Рисунок 3.1 – Модель системи нотації IDEF0

Отримати бачення про ціле як сукупність частин можливо лише при розгляді архітектури, а не кодуванні. Розглянемо модель проекту, діаграму IDF0, рисунок 3.1. Кожна модель має своє призначення. У цьому випадку збирання статистик та надання кращої аналітики автоматизовані. Модель Hospital Manager Helper має чотири сторони, які відповідають за вхід, вихід, контроль та механізм. З лівого боку модель отримує форму даних та запит. Знизу приєднується механізм, за рахунок чого здійснюється виконання. Механізмом виступає медичний персонал, ПК та база даних. Згори на модель накладається контроль. Всі роботи контролюються вказівками щодо безпеки, поводження з помилками, технічною проектною документацією та законодавством України. З правої сторони моделі виходить автоматично заповнені статистичні форми, зберігання діагностичної інформації в базі, вивід інформації у зручній формі для подальшого використання та додаткового можливості збереження частини результатів за необхідності.

3.2 Системна архітектура

На рисунку 3.2 показана схема роботи системи всякий раз, коли пацієнт вибирає перегляд медичних записів за допомогою MetaMask або децентралізованого веб-сайту системи охорони здоров'я. Отримуючи доступ до приватного ключа з гаманця Ethereum, користувач автоматично входить в систему.

Ethereum wallet - це гаманець для холодного зберігання. В результаті, по порівнянню з іншими гарячими гаманцями, небезпека компрометації досить низька. Крім того, якщо гаджет втрачено, пацієнтам можна просто видати новий, не піддаючись штрафу за втрату своїх медичних записів. Гаманець можна використовувати таким же чином можна підписати будь-який документ або для яких - або потреб у перевірці. Цей гаманець також можна використовувати для проведення багатосторонньої верифікації пацієнта. Він може бути використаний для створення системи контролю доступу до

записів на основі ролей, а також системи розподіленої ідентифікації власності на основі блокчейна. У випадку невідкладної медичної допомоги може бути реалізований аналогічний багатосторонній механізм отримання дозволів для отримання доступу до записів пацієнт.

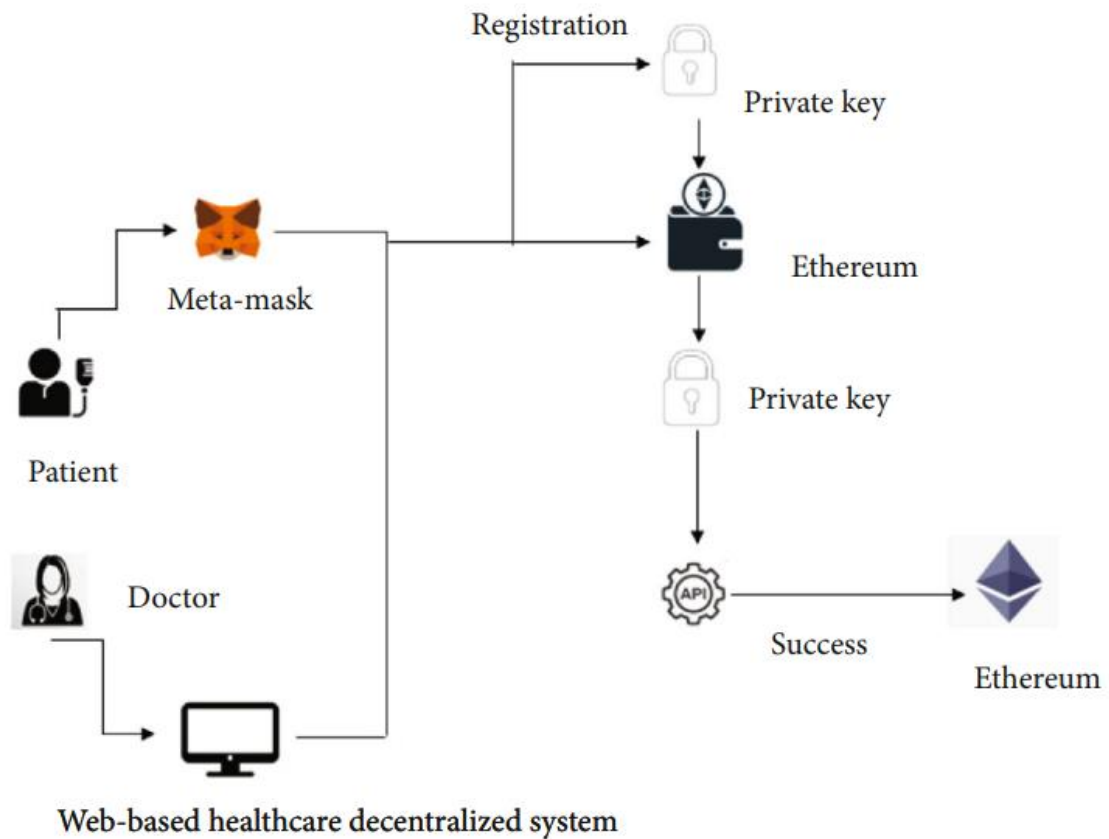


Рисунок 3.2 – Схема протоколу системи EHR

На рисунку 3.3 показана структурна схема. Дизайн, який ми пропонуємо, складається з чотирьох основних компонентів: Користувач додаток, протокол рукоштовання блокчейна, хмара і загальнодоступна блокчейн-мережа. Система являє собою віртуальне представництво, яке служить двом цілям. По-перше, він надає користувачам доступ до інтерфейсів додатків. Лікарі та системні адміністратори - це два типи користувачів у нашій системі.

У кожного користувача є своя функція. В результаті користувацький додаток надає різні інтерфейси користувача залежно від

ролі користувач. По-друге, на основі введених користувачем даних користувальницький додаток створює початкову транзакцію. З метою підтвердження транзакція відправляється в протокол рукостискання блокчейна. Нарешті, користувальницький інтерфейс встановлює зв'язок між користувачами і протоколом рукостискання блокчейна.

Фундаментальним компонентом запропонованої архітектури є протокол blockchain handshake (BH). Цей компонент з'єднує сервер бази даних, блокчейн-мережу і хмарну систему медичних записів, яка діє як оболонка. Ця запропонована Архітектура використовує блокчейн-мережу Ethereum. Розподілена бухгалтерська книга, що з'єднує блокчейн-вузли, відома як публічна блокчейн-мережа. Блокчейн-вузли - це майнери, які відповідають за оновлення блокчейна на основі методу прийняття рішення.

В якості альтернативи блокчейн-вузли приймають транзакції і використовують смарт-контракти мережі для їх аутентифікації.

У запропонованому проекті хмара надає дві послуги, які аналогічні тим, які надаються існуючими хмарними сервісами.

Системи адміністрування EHR. Система адміністрування EHR розміщується в якості початкової служби. Зберігання даних-це наступна послуга. Всі медичні записи можуть бути збережені в базі даних. Система адміністрування EHR приймає транзакції з протоколу рукостискання блокчейна, виконує всі обов'язки, пов'язані з ними, і, нарешті, зберігає їх в хмарній базі даних. У відповідь на запити користувачів про доступ хмара надає необхідні дані.

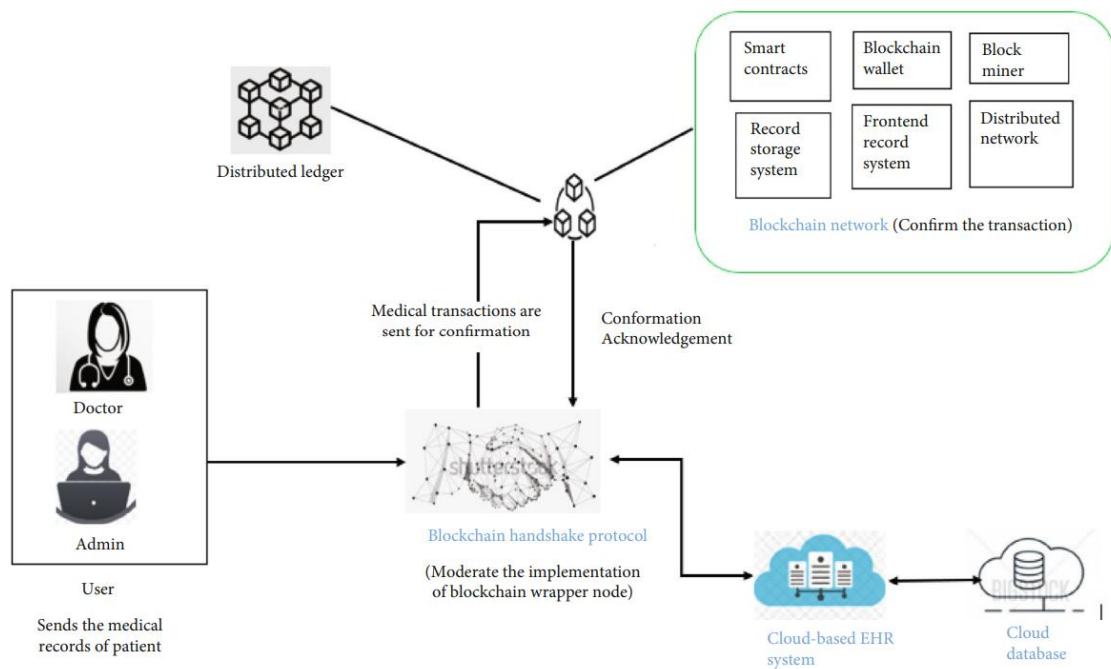


Рисунок 3.3 – Схема взаємодії компонентів системи

3.3 Представлення моделі системи

У цій роботі була використана системна парадигма, що дозволяє здійснювати віддалений моніторинг пацієнтів, залишаючись при цьому захищеною від витоку даних. Оскільки важливі дані пацієнта можуть бути легко змінені будь-ким, ми хочемо надати пацієнтам безпечну та надійну платформу. Нижче наведені кроки, використані для створення платформи.

3.3.1 Дані, зібрані з вузлів Інтернету речей

Вузли Інтернету речей збирають достовірні дані пацієнта. Вони вигадані з різних датчиків (датчик температури (LM35), датчик вимірювання серцебиття і т. д.) і медичного обладнання, які розміщені навколо пацієнта. Ми регулярно збираємо дані з цих вузлів. Для збору даних з оточення пацієнта ми використовуємо різні датчики (датчик температури (LM35), датчик вимірювання серцебиття), а Raspberry Pi виступає в якості головного вузла. Цей головний вузол отримує дані з датчиків. Потім дані попередньо

обробляються на цьому вузлі, і будь-яке сміття і зайві дані видаляються з даних [16].

Після цього дані шифруються за допомогою методу асиметричного шифрування з використанням відкритого ключа лікаря і збереженого зашифрованого значення в хмарі. Тільки пацієнт і призначені лікарі мають доступ до його або її даних.

3.3.2. Створення блокчейн для кожного пацієнта в мережі Ethereum

Як ми всі знаємо, кожна зміна вмісту блоку в мережі блокчейн впливає на всі блоки, що знаходяться перед ним. Отже, якщо ми створимо єдиний блокчейн для всіх пацієнтів в одній мережі Ethereum, і якщо кожен пацієнт оновить свої дані у своєму блоці, це вплине на кожен попередній вузол у ланцюжку. Оскільки кількість пацієнтів зростає з кожним днем, це може стати непомірно дорогим.

Пацієнт має право тільки на своєму блоці [18,19]. Ніхто не може отримати доступ до його інформації без його дозволу.

Тільки призначені лікарі можуть отримати доступ до цього блоку за допомогою особистого ключа лікаря.

Коли пацієнтам не потрібні дані з вузлів Інтернету речей, ми можемо безпосередньо завантажувати особисту інформацію пацієнта в базу даних MySQL. Він може просто ввести інформацію, яку він хоче передати своєму лікарю. Ці дані негайно завантажуються в базу даних після введення значень або інформація на веб-сайті. Щоб читати і писати, як пацієнти, так і лікарі повинні відповідати необхідним вимогам. Зберігайте хмарну адресу пацієнта в блокчейн.

Після введення точної інформації на нашому веб-сайті пацієнт повинен завантажити свою хмарну адресу у свій блок. Ці блоки послідовно зберігаються в блокчейн-мережі. Потім вся система перетворюється на компактну і безпечну систему електронної охорони здоров'я. Пацієнти

повинні увійти в свої облікові записи і ввести Хмарний адресу, коли їм потрібно завантажити дані в блокчейн. Він також повинен відповідати вимогам щодо повноважень на читання та запис свого блоку. Смарт-контракт розробляється на мові програмування Solidity в середовищі IDE remix online і потім публікується в блокчейн-мережі Ethereum [17]. Хмарні адреси в першу чергу зберігаються для кожного пацієнта в його власному блокчейне Ethereum. Пропонована модель не тільки зберігає всі дані пацієнтів через високу вартість плати за газ; тому ми просто зберігаємо хмарну адресу даних Інтернету речей.

Доступ до нього має лише власник цього блоку; в іншому випадку транзакція на Ethereum неможлива.

3.3.3 Функція дозвілу лікаря на доступ до блоку

Інша функція дозволяє пацієнту надати довіреному лікарю доступ до даними цього блоку. Якщо лікар має дійсний обліковий запис Ethereum і дозвіл пацієнта, він може читати і записувати в блок. Крім того, пацієнт вказує часовий інтервал для подальшого захисту, і лікар зобов'язаний отримати доступ до цього блоку протягом цього періоду часу. Лікар використовує свій особистий ключ для розшифровки. Обчисліть значення хмарного адреси і витягніть необхідну інформація. Потім він дає пацієнту відповідні поради, зміна змінних рекомендацій блоку. Потім пацієнт отримує цю пораду на сторінці рекомендацій веб-сайту.

3.4 Смарт-контракт

Безпека даних вузлів Інтернету речей є нашою першочерговою турботою, тому ми використовуємо методи асиметричного шифрування даних для забезпечення безпеки, яка робить систему цілком захищений від атак. Але ми розгорнули нашу систему на блокчейн Ethereum, щоб зробити

його більш безпечним, оскільки хмарними даними також можна маніпулювати будь-яким способом. Ці дані зберігаються в хмарі для віддаленого доступу до лікаря і повинні бути захищені від атак зловмисників. У цьому випадку пацієнти можуть дозволити одному лікарю отримати доступ до хмарних даних через смарт-контракт. Ми укладаємо смарт-контракт, який забезпечує безпечну перевірку дозволів і надає тільки бажаним лікарям доступ до хмарних даних конкретного пацієнта.

Псевдокод смарт-контракту описаний нижче, щоб дати чітке уявлення про те, що лікар може отримати доступ до захищеної інформації пацієнта лише за допомогою особистого ключа. Приватний ключ розшифровує зашифровані дані в хмарі. Потім лікар дає рекомендації пацієнтам, і ці дані можна бачити лише після входу у свій обліковий запис на веб-сайті.

Лікарю дозволяється отримувати доступ до певних блоків для пацієнтів лише деякий час проміжку часу після початку прийому.

3.5 Моделювання програмних компонентів системи

3.5.1 Діаграма класів

Розробка структури класів є складовою частиною технічного завдання будь – якого проекту. Діаграми класів дають розробнику зрозуміти приблизну структуру класів та їх залежність один від одного. Кваліфікований підхід до створення документації дозволяє знайти існуючі та уникнути багатьох помилок в подальшому.

Об'єктно-орієнтований підхід робить крок вперед, пропонуючи програмісту можливість представляти завдання у своєму просторі. Цей підхід має загальний характер і не підпадає під будь – які обмеження чи характер проблеми. Проблеми вирішуються за допомогою об'єктів у межах простору завдання. Це гнучка і потужна абстракція. За допомогою ООП ви можете описати завдання в контексті самої задачі, а не в контексті ПК, на якому

працює програма. Однак все ж є зв'язок з ПК. Кожен об'єкт схожий на невеликий ПК: у ньому є стани та операції, які він може виконувати. Ця аналогія пов'язана із зовнішнім світом, нас оточують об'єкти із властивостями та поведінкою.

Створений та перевірений клас - корисний блок коду. Код для багаторазового використання - вражаюча перевага об'єктно-орієнтованих мов. Новий клас може містити будь-яку кількість інших типів об'єктів, необхідних для досягнення необхідної функціональності. Якщо ви створюєте новий клас із існуючих класів, цей метод є композицією.

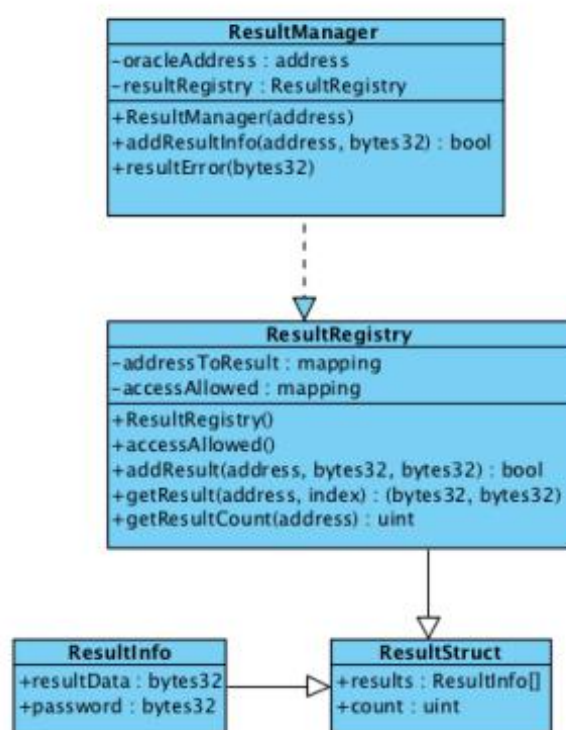


Рисунок 3.4 – Основні класи

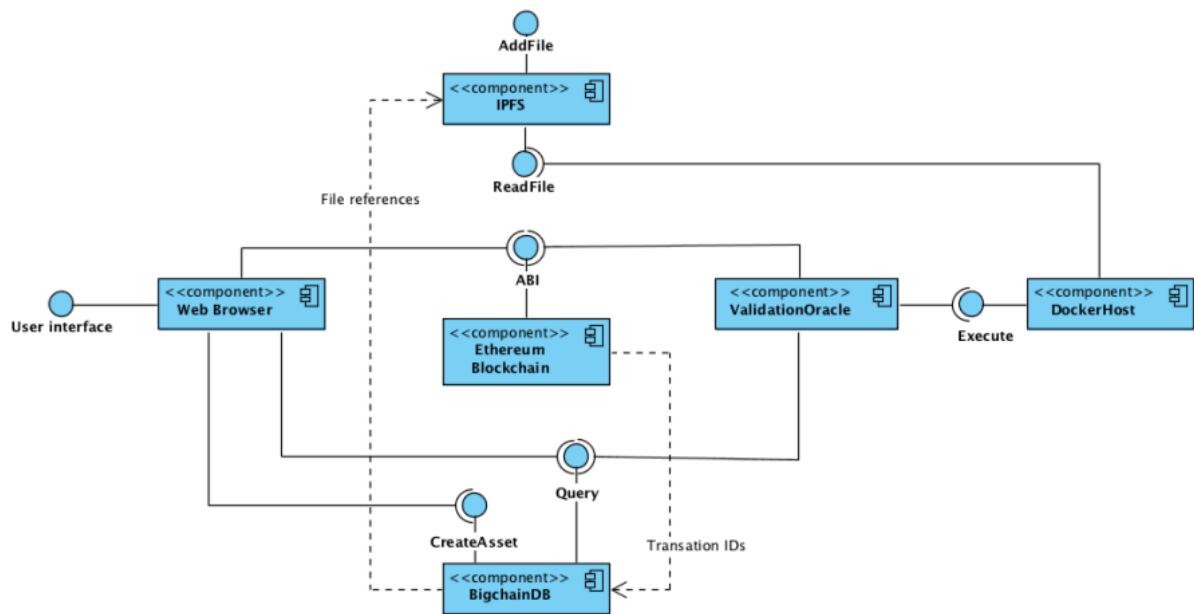


Рисунок 3.5 – Складові основних класів

Веб-браузер - цей компонент надає веб-інтерфейс для взаємодії користувачів з додатком. Користувачам потрібен плагін для браузера Ethereum wallet, а саме MetaMask [38] для управління своїми обліковими записами. Веб-інтерфейс надає функціональність для зберігання наборів даних, програмного забезпечення та контейнерів у блокчейні, а також для запуску виконання обчислень в автономному контейнері. Він також відображає результати обчислень для публічної адреси Ethereum, підключеної до система.

BigchainDB - BigchainDB надає API для зберігання ресурсів і запитів до бази даних. Він використовується додатком для зберігання метаданих ресурсів, таких як імена файлів, специфікації файлів, IPFS-адреси та вартість використання.

Блокчейн Ethereum - система смарт-контрактів розгорнута на блокчейні Ethereum. Цей компонент забезпечує двійковий інтерфейс програми (ABI) користувачам для виклику методів контракту і читання або запису інформації в блокчейн.

IPFS - цей модуль надає API для додавання та читання файлів із

мережі IPFS. Постачальники наборів даних, програмного забезпечення та контейнерів додають свої файли до мережі IPFS вручну перед взаємодією з Додатком, а потім зберігають адреси IPFS в BigchainDB за допомогою веб-інтерфейсу.

ValidationOracle - смарт-контракти не можуть отримати доступ до даних за межами блокчейна, тому існує необхідність в каналах передачі даних, званих oracles, призначених для відправки зовнішніх даних в смарт-контракти. У розробленій системі ValidationOracle грає роль оракула, отже, це важливий компонент програми.

Однією з його основних завдань є перевірка інформації перед її відправкою в автономний контейнер. Щоб досягти цього, він обчислює контрольну суму даних і порівнює її з контрольною сумою, що зберігається в блокчейні. Далі, він витягує статус контейнера, який призначений для виконання обчислення. Виконання запускається лише в тому випадку, якщо значення контрольної суми рівні і контейнер запущений. В у цьому випадку oracle встановив статус обчислення на SUCCEEDED, викликавши контракт ComputationManager, в іншому випадку буде анульований. Іншою важливим завданням цього компонента є отримання вихідних даних обчислень з контейнера і запис їх в блокчейн шляхом виклику контракту ResultManager.

DockerHost - для обчислення виконуються в контейнерах Docker. Провайдери розміщують запущені контейнери на своїх комп'ютерах і пропонують їх за певну плату для виконання обчислень. Файли, що стосуються контейнерів, такі як їх специфікації, а також скрипти, які керують виконанням обчислень, розміщуються в IPFS.

3.5.2 Діаграма послідовності

Представлення процесу містить взаємодію між учасниками та різними компонентами системи. У цьому розділі описується процес для двох різних учасників, постачальників і окремих користувачів.

Процес взаємодії з системою з точки зору постачальника з метою надання набору даних, програмного забезпечення та / або контейнерних ресурсів проілюстрований на рисунку 3.6. На Крок 1 постачальники зберігають свої файли в IPFS. Набори даних повинні бути зашифровані постачальником, перш ніж будуть додані до IPFS. Як тільки ці файли додаються до мережі IPFS, їх метадані, а також IPFS-адреса зберігаються в BigchainDB за допомогою веб-інтерфейсу, як показано на кроці 4. Контрольна сума цієї інформації обчислюється в інтерфейсі на кроці 6 і він зберігається в блокчейні разом з Ідентифікатором транзакції Bigchain DB.

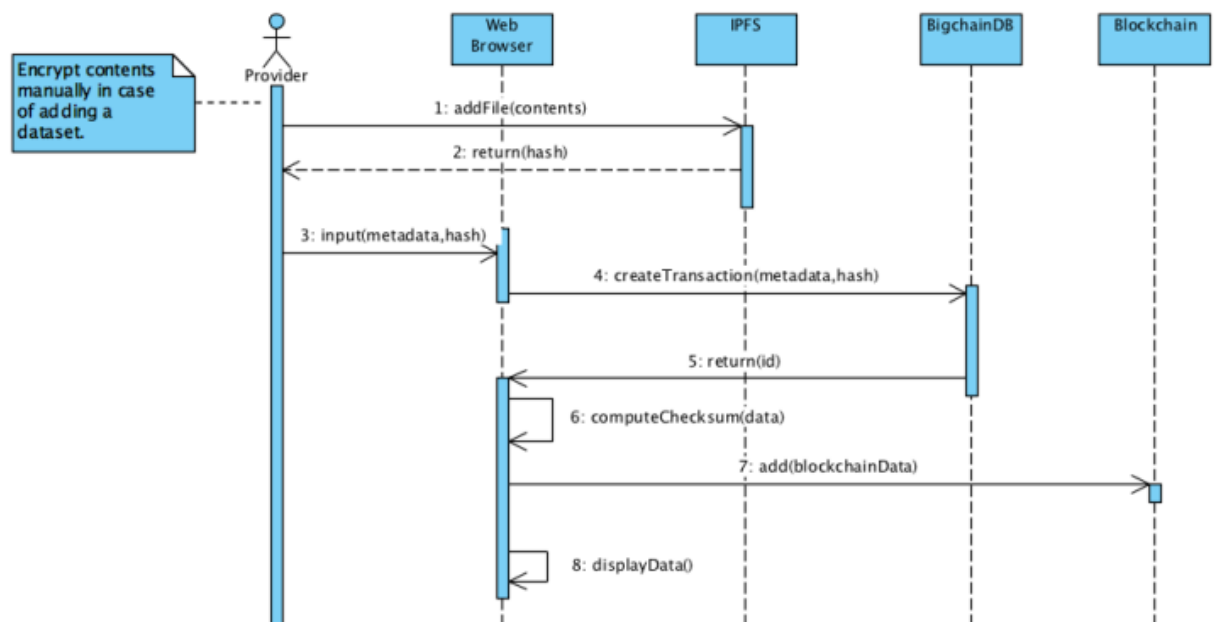


Рисунок 3.6 – Діаграма послідовності

Крок 7. Як тільки цей процес завершується, завантажені ресурси стають доступними для користувач. Процес взаємодії з системою в якості індивідуального користувача, а також всі кроки, необхідні для виконання обчислень, проілюстровані на рисунку 3.6. Постачальники слідує тому ж процесу, якщо вони хочуть діяти як індивідуальні користувачі та замовляти обчислення.

На кроці 1 служба oracle запущена і очікує нових обчислювальних

подій. За допомогою веб-інтерфейсу рівні користувачі вибирають, який набір даних, контейнер та програмне забезпечення вони хочуть використовувати, і інформація про обчислення зберігається в блокчейні на кроці 3, включаючи суму в ефірі за використання цих ресурсів. Користувачі також надають свій відкритий ключ, який потрібен контейнеру в процесі шифрування обчисленого результату. На кроці 4 oracle отримує повідомлення про надходження нового обчислення, і на наступному кроці він зчитує значення контрольної суми з блокчейна. Контрольна сума обчислювальних даних обчислюється в oracle шляхом запиту до BigchainDB, і це у порівнянні зі значенням блокчейна. Крім того, на кроці 10 перевіряється статус вибраного контейнера. Якщо контрольні суми рівні і вибраний контейнер запущений, він отримує повідомлення про початок виконання обчислення, як показано на кроці 12. У цьому випадку обчислений результат шифрується одноразовим паролем, цей пароль шифрується відкритим ключем користувача, а зашифровані Тексти зберігаються в IPFS. На кроці 14 oracle отримує IPFS-адреси, а на кроці 15 зберігає ці адреси в блокчейні.

Потім відповідні провайдери повинні отримати вказану суму в ефірі за пропозиція своїх ресурсів. З іншого боку, як показано на кроці 19, отримано повідомлення про помилку, і кошти повертаються ініціатору обчислення. В обох випадках інтерфейс повідомляє користувача IPFS-адресами, що містять результат, або повідомлення про помилку, як показано на кроках 17 і 22.

3.5.3 Діаграма варіантів використання

Діаграма варіантів використання показана на рисунку 3.7. У прецеденті використання цього додатка є три ключові сутності: адміністратор, пацієнт і лікар.

Тепер введення відомостей про профіль, який є унікальною функцією, надає доступ до всіх трьох організацій, також включений в список дій. Пацієнт має доступ до трьох операціями з десяти, в той час як лікар має

доступ до трьох операціями з десяти. Тільки адміністратор має доступ до всіх десяти діям, що дозволяє йому переглядати і відстежувати всю інформацію. Єдина операція, яка може змінити Дані блоку один раз це було відновлено шляхом запису історії хвороби пацієнта, що може зробити тільки лікар.

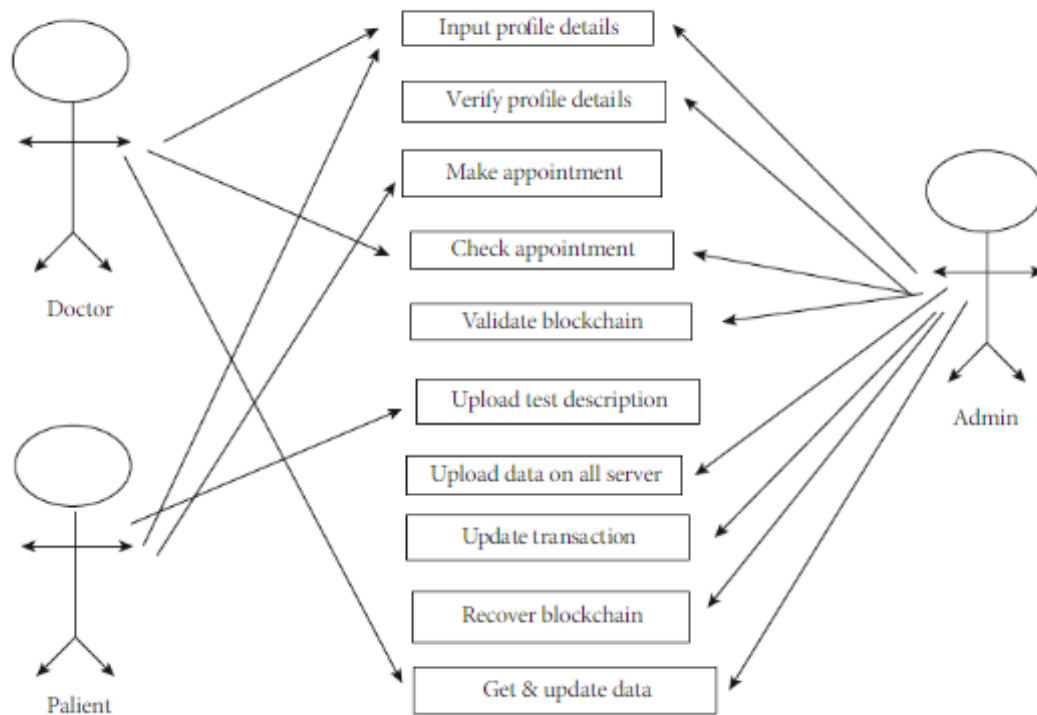


Рисунок 3.7 –Діаграма варіантів використання

3.5.4 Діаграма активності

Блок-схема запропонованої системи. На рисунку 3.8 показано процес створення медичної карти. По-перше лікар системи надасть медичну карту. Після цього лікар записує результати обстеження кожного пацієнта. Транзакція метаданих для цієї медичної карти буде оброблена.

Частина даних, яка називається метаданими транзакцій, додається до транзакції після її обробки. Незалежно від того, успішна транзакція чи ні, всі транзакції, які реєструються в бухгалтерській книзі, містять метадані. Інформація про транзакцію містить детальний опис укладення угоди. Після цього медичне дос'є буде завантажено в мережу IPFS. IPFS (інтерпланетна

файлова система) - це система документообігу, яка дозволяє виконувати транзакції з мінімальними витратами ресурсів і часу. Ми отримуємо адресу вмісту після завантаження файлу в мережу IPFS.

Транзакція Ethereum - це наступний етап. Ganashe необхідний для транзакцій Ethereum, оскільки він надає адреси та секретні ключі. Адреси зберігаються у файлі, і транзакції видно всім. Для здійснення транзакції використовуються закриті ключі, які розблокують ці адреси. Віртуальна машина Ethereum (EVM) використовується для обробки транзакції Ethereum. EVM в основному використовується для проведення взаємодій зі смарт-контрактами, в яких всі вузли повинні погоджуватися з тим, що транзакція відбулася кожен раз, коли хтось взаємодіє з контрактом. Після цього EVM виконує контракт негайної відправки відповідно до правил транзакції. Ethereum зберігає запис усіх попередніх транзакцій і історію блокчейна, яка зберігається і підтверджується через згоду. З іншої сторони, оператори вузлів Ethereum відстежують всі взаємодії зі смарт-контрактами, які відбуваються в мережі Ethereum. У цій ситуації майнер буде перетворено на бота, який буде обробляти транзакції автоматично, коли транзакція буде отримана.

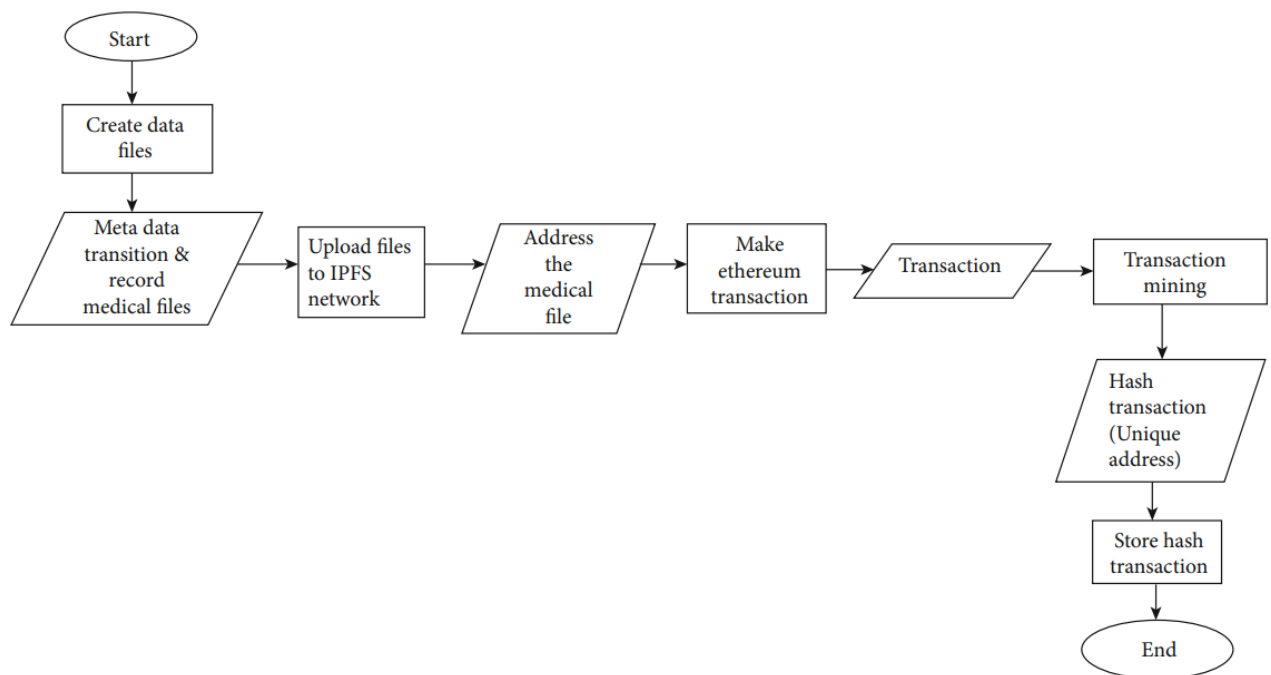


Рисунок 3.8 – Блок-схема створення медичної карти

3.5.5 Діаграма розгортання

Фізичний вигляд описує, як підключаються програмне та апаратне забезпечення/

Додатки BlackboxChain і як встановлюється система під час її роботи. Рисунок 3.9 ілюструє UML-схему розгортання програми.

Веб-інтерфейс-інтерфейс запущений в браузері на комп'ютері. Він також може бути розміщений на IPFS, щоб бути децентралізованим і загальнодоступним. Він взаємодіє з блокчейном Ethereum за допомогою віддалених викликів процедур (RPC) і з вузлом BIGCHAINDB по протоколу HTTP.

EVM-смарт-контракти, використовувані системою, розгортаються в EVM, який діє як серверна частина програми.

Кластер BigchainDB - цей компонент містить набір підключених вузлів BighchainDB. Oracle та інтерфейсний сервер підключаються до одного з цих вузлів, щоб запит бази даних або додавання даних користувачів до мережі.

IPFS-запущений демон IPFS необхідний додаткам для взаємодії з іншою частиною мережі IPFS. Контейнери Docker підключаються до запущеного демона IPFS через протокол HTTP для додавання та читання файлів у мережі.

ValidationOracle.js-це Node.js [41] процес, який розгортається на комп'ютері. Він використовує RPC для спілкування з EVM та HTTP для підключення до вузлу BigchainDB і DockerHost.

DockerHost - цей пристрій описує набір запущених контейнерів Docker, розміщених на комп'ютер. Кожен контейнер ізольований від навколишнього середовища і взаємодіє з демоном IPFS і ValidationOracle.JS процес з використанням HTTP.

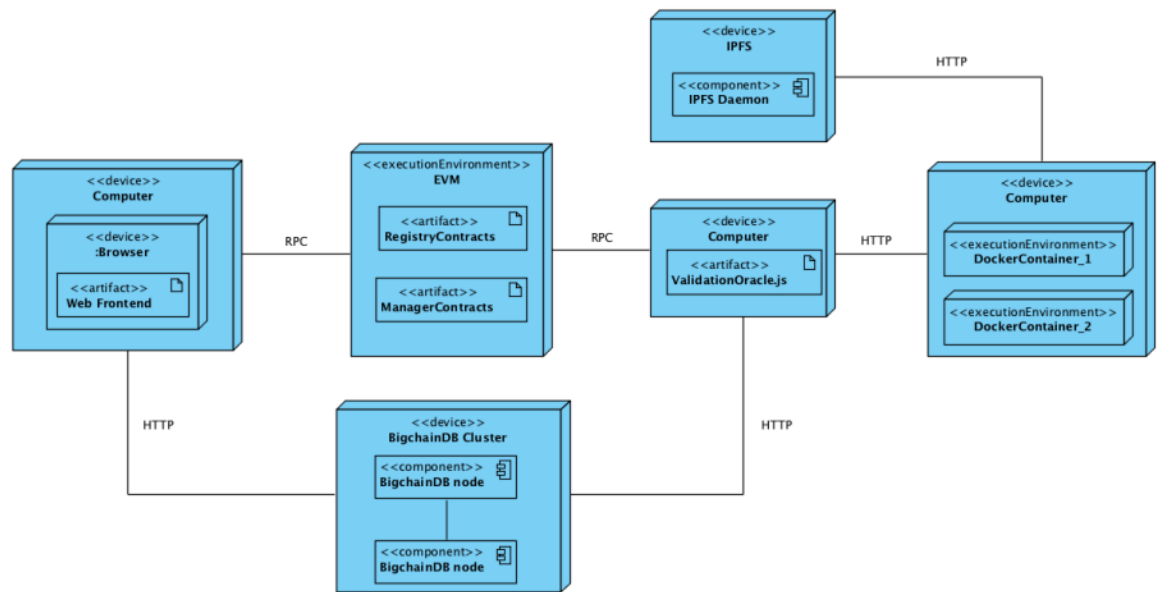


Рисунок 3.9 – Діаграма розгортання

3.6 Інструментальні засоби розробки

3.6.1 Ganache

Це локальний блокчейн Ethereum для швидкого створення децентралізованих програм. Ganache можна використовувати для розгортання, розробки та тестування в передбачуваному та безпечному середовищі протягом усього циклу розробки. Він працює в обох напрямках: як настільна програма і як інструмент командного рядка (Ethereum).

3.6.2 MetaMask

Це точка входу, яка дозволяє переглядати децентралізовану мережу майбутнього у вашому браузері прямо зараз. Це дозволяє запускати децентралізовані програми Ethereum без необхідності запускати повний вузол Ethereum у вашому браузері.

3.6.3 Web3

Перевірка транзакцій повинна виконуватися в ланцюжку, щоб взаємодіяти з модулями в ланцюжку. Щоб створити та підтвердити транзакцію, учасник мережі іншого автономного фреймворку повинен передати її через однорангове з'єднання (p2p), яке є реальною мережею. Він також включає бібліотечну колекцію, яка полегшує взаємодію між вузлами Ethereum та компонентами всередині ланцюга. Він використовується на стороні сервера для Node.js.Web3 використовує з'єднання через протокол передачі гіпертексту (HTTP) для підключення до мережі Ethereum через вузол Ethereum. Це може бути вузол в ЕТН-гаманцях з локальної системи. MetaMask-це розширення для браузера, яке дозволяє працювати з обліковими записами Ethereum і може використовуватися для інтеграції Ethereum з веб-сайтом. MetaMask-це гаманець браузера Ethereum, який підключає brows до класу провайдерів Web3. Постачальник Web3-це структура даних, яка надає посилання на загальнодоступні вузли Ethereum. Користувач може використовувати, зберігати та підтримувати відкриті та приватні ключі, унікальні для свого облікового запису, за допомогою MetaMask. Комбінація Ethereum, MetaMask і web3.js, а також веб-інтерфейс, що забезпечує взаємодію між серверною частиною і інтерфейсом.

3.6.4 Truffle

Це потужне середовище розробки віртуальних машин Ethereum, яке використовує блокчейни, а також конвеєр активів та тестову платформу для того самого. Він володіє деякими функціями, такими як обчислення, реалізація та обслуговування смарт-контрактів, а також управління бінарними залежностями. У ньому також є повністю автоматизоване середовище для тестування смарт-контрактів і платформа розгортання і міграції, яку можна створювати за сценарієм і розширювати. Це може створити прямий зв'язок з контрактом і конвеєр з тісною інтеграцією.

Середовище Truffle використовується для запуску програм. Сервер і внутрішня частина веб-сайту управляються за допомогою мови програмування Solidity і Node.js. Є два інструменти, Truffle і Ganache, які використовуються для генерації локальних блокчейнів Ethereum для побудови системи. Віртуальний інтерфейс Ethereum, MetaMask (як гаманець), Truffle (як IDE), Yarn (інтерфейс командного рядка), Ganache (створення облікового запису) і local Web3 (веб-інтерфейс) використовуються для створення ланцюжка блоків і доступу до системи або її використання

3.6.5 VS Code

Visual Studio Code від Microsoft-це редактор для Windows, Linux і macOS. Усунення несправностей, управління Git, GitHub, підкреслення синтаксису, розумне завершення коду, приклади та виправлення помилок доступні мовами

.

4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

4.1 Розробка алгоритмів

Процес отримання доступу до даних в розробленій системі представлено в цьому розділі. Ця система побудована з використанням Truffle і Ganache, двох простих у використанні інструментів для створення локального блокчейна Ethereum. Сервер і внутрішня частина системи управлялися з використанням мови Solidity і Node.js .

Для створення блокчейна і доступу до системи використовуються віртуальний інтерфейс Ethereum, MetaMask (в якості гаманця), Truffle (в якості IDE), використовуються Yarn (інтерфейс командного рядка), Ganache (створення облікового запису) та Local Web3 (веб-інтерфейс).

4.1.1 Робота з даними пацієнта та лікаря

Панель адміністратора. Інтерфейс адміністратора цієї системи представлено на рисунку 4.1.

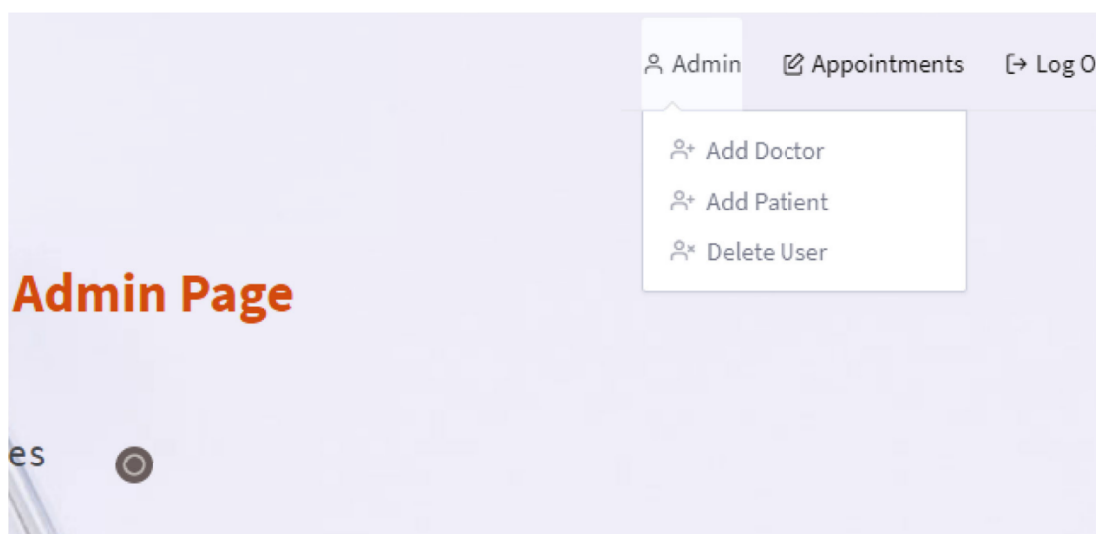


Рисунок 4.1 – Панель адміністратора

Адміністратори можуть додавати лікарів та пацієнтів до системи, а також видаляти поточних користувачів за допомогою інтерфейсу адміністратора (лікар і пацієнт). Адміністратор має свій власний незмінний і недоступний обліковий запис в цій системі, який має повноваження створювати і видаляти користувачів (лікаря і пацієнт). З метою автентичності і безпеки ніхто інший не має до цього доступу. Адміністратор може переглядати записи на прийом і повідомляти лікарів про їх списки пацієнтів. Він також може планувати прийом в залежності від кількості пацієнтів. Він може також надати інформацію про призначені зустрічі та інші відповіді на питання і призначення.

Додавання лікаря. На рисунку 4.2 показаний процес додавання лікаря. На рисунку 4.2.a показаний модуль реєстрації лікаря, де потрібно заповнити дані лікаря, такі як ім'я; дата народження; ідентифікатор електронної пошти; номер мобільного телефону; ідентифікатор лікаря, який є адресою облікового запису в Ganache Ethereum; Місто; Держава; і спеціальність. На рисунку 4.2.b показано повідомлення про підтвердження від MetaMask, який використовується в якості смарт-контракту. За допомогою MetaMask буде збережена вся інформація про лікаря, і система отримає повідомлення з підтвердженням аутентифікації. Але, якщо користувач використовує невірну інформацію або адресу облікового запису, що вже існує, система видасть повідомлення про відмову в доступі. Нарешті, цей смарт-контракт забезпечує безпеку даних лікаря.

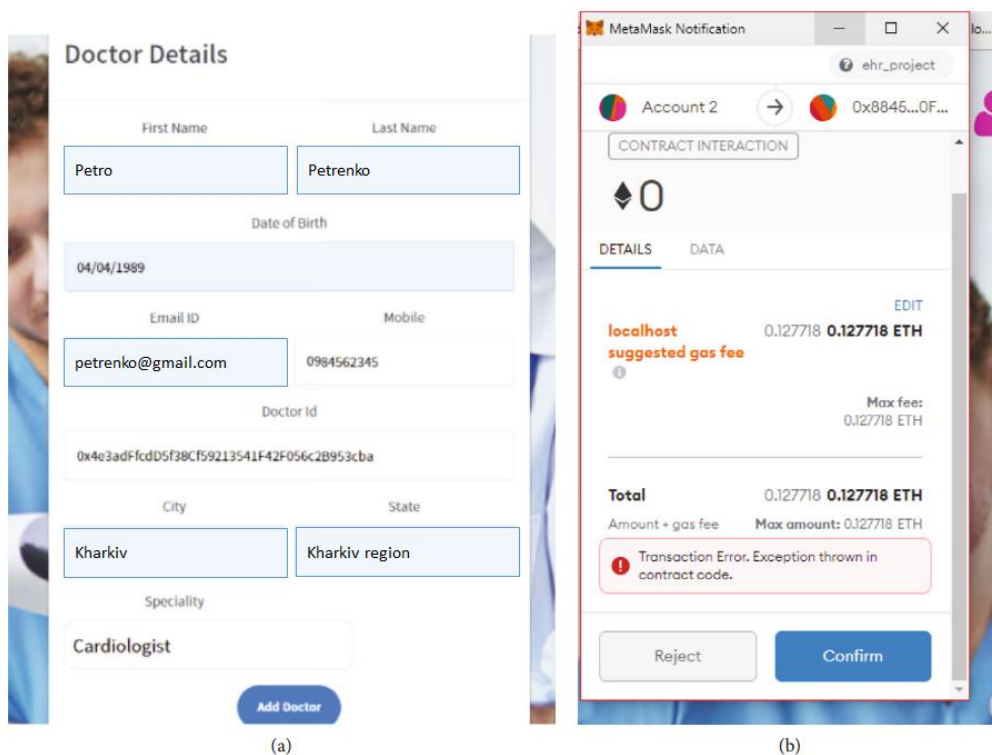


Рисунок 4.2 – Додавання даних про лікаря

Додавання пацієнта. На рисунку 4.3 описаний метод додавання пацієнт. На рисунку 4.3.a показаний модуль реєстрації пацієнтів, що вимагає від вас введення інформації про пацієнтів, наприклад імена, адреси електронної пошти, номери телефонів та ідентифікатори пацієнтів, які є адресою облікового запису Ganache Ethereum.

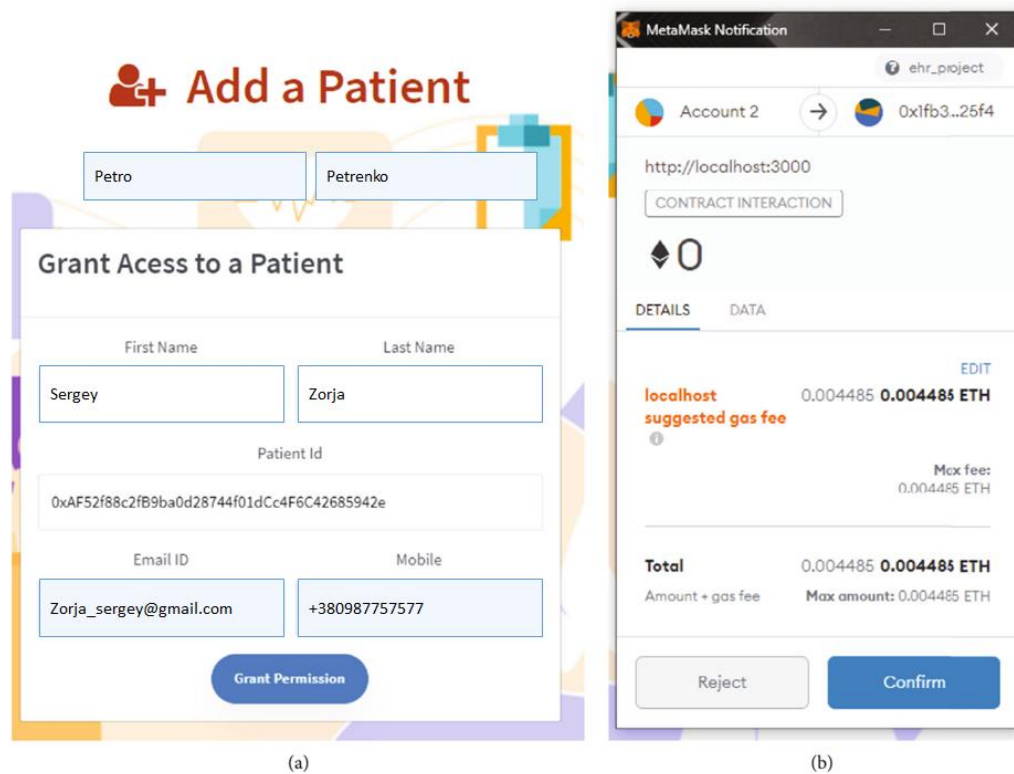


Рисунок 4.3 – Додавання даних про пацієнта

Після заповнення заповнивши всі прогалини необхідною інформацією, Користувач потрібно натиснути кнопку "Додати пацієнта", щоб зберегти дані, а потім переходити до наступного процесу. Підтверджуюче повідомлення від MetaMask, яке використовується в якості смарт-контракту, показано на рисунку 4.3.b. Вся інформація про пацієнта буде збережена за допомогою MetaMask, і система отримає повідомлення з підтвердженням аутентифікації. Щоб зберегти всі дані, повідомлення MetaMask відображає на екрані детальний курс валюти Ethereum. Він зберігає всі дані в форматі валюти Ethereum. Система відобразить повідомлення про заборону доступу, якщо користувач введе невірну інформацію або та ж адреса облікового запису. Нарешті, безпека самого дані пацієнта захищені цим смарт-контрактом.

Видалення користувача. На рисунку 4.4 показано процес видалення користувача з системи. На рисунку 4.4.a показано видалення користувача, при якому тільки адміністратор має право видалити лікаря і пацієнта як

користувача зі списку. Щоб видалити будь-якого користувача, ви повинні вибрати тип користувача та ідентифікатор користувача, наприклад лікаря або пацієнта. На рисунку 4.4.b показано повідомлення про підтвердження від MetaMask, яка використовується в якості смарт-контракту. За допомогою MetaMask вся інформація про пацієнта та лікаря буде видалена, і система отримає повідомлення з підтвердженням для перевірки автентичності. Крім того, адміністратор також може стерти важливі дані лікаря та пацієнта.

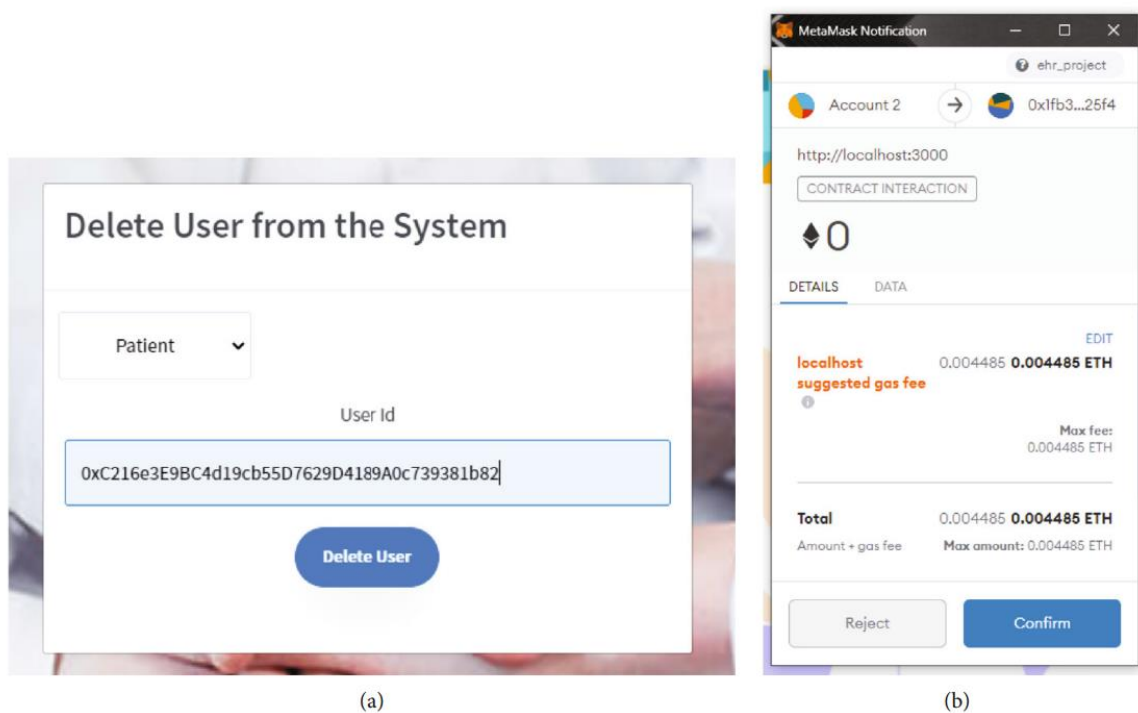


Рисунок 4.4 – Видалення даних про пацієнта або лікаря

На рисунку 4.5 показана панель лікаря. На панелі лікаря лікар може переглядати і редагувати свою інформація. Крім того, лікарі можуть додавати записи пацієнтів, а також оновлювати їх. Коли консультація лікаря з пацієнтом закінчується, лікар має можливість видалити запис пацієнта.

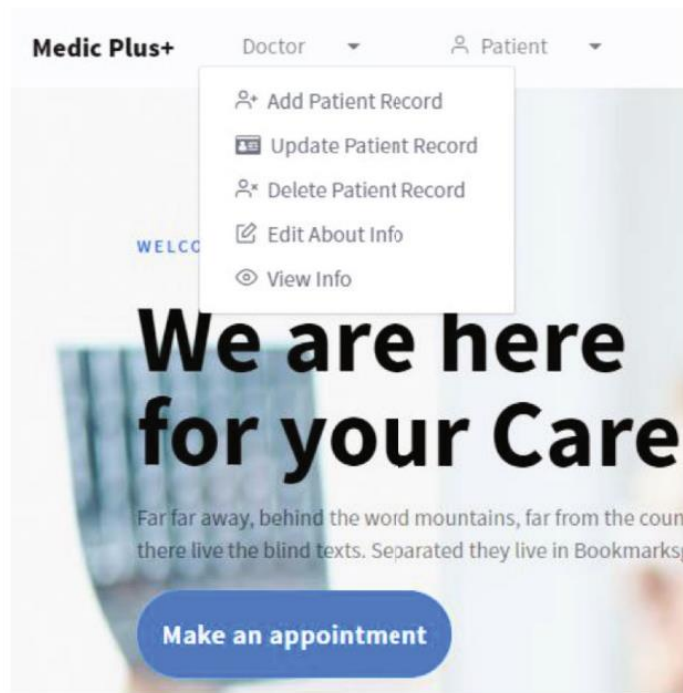


Рисунок 4.5 – Панель лікаря

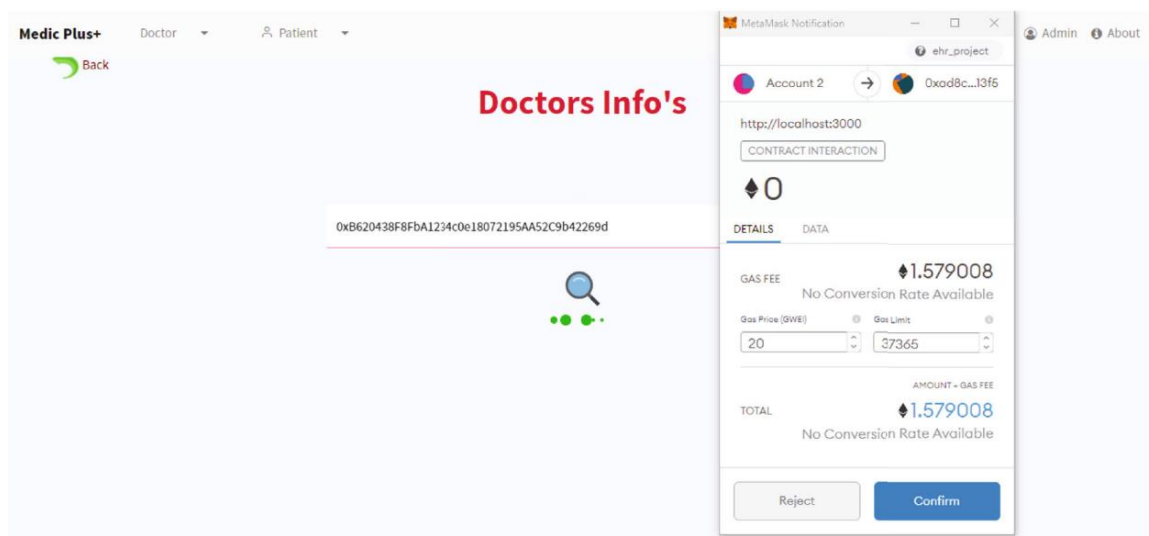


Рисунок 4.6 – Пошук лікаря

Перегляд і редагування інформації лікарем. На рисунку 4.6 показано, як лікар може переглядати і редагувати свою особисту інформацію. На рисунку 4.6 показано повідомлення про підтвердження через MetaMask після пошуку ідентифікатора лікаря. Лікар також може редагувати свою інформацію. Після оновлення вся інформація буде збережена і захищена за допомогою маски мережі.

Припис пацієнту. На рисунку 4.7 показана медична карта пацієнта, з якою може оперувати тільки лікар. По суті, це рецепт пацієнта. Лікар може завантажити медичну карту пацієнта, яка буде включати особисту інформацію пацієнта, а також медичні записи.

Лікар заповнює порожні поля такою інформацією, як ім'я, рік народження, номер телефону та адреса електронної пошти. Крім того, лікар може залишити коментар для пацієнта, щоб встановити теплі стосунки. Частина рецепта наведена на рисунку 4.7, який включає перелік основних медичних ліків та аналізів. Після того, як пацієнту був поставлений діагноз, лікар може додати назви ліків, дози, клінічні тести і побічні зауваження, щоб мотивувати пацієнта. Якщо пацієнту потрібен новий препарат або новий клінічний тест, він може оновити свою медичну карту в залежності від свого рецепта.

The screenshot displays a medical record interface with two main sections: 'Medications' and 'Recommended Clinical Test'.

Medications Section:

#	BRAND NAME	DOSAGE	FREQUENCY	NO.OF DAYS	REMARKS	ACTION
1	Rocal -D	20	0-1-1	4	Continue	Delete
2	Carbizol	5	0-0-1	4	Continue	+ Add

Recommended Clinical Test Section:

#	CLINICAL TEST	ACTION
1	Complete Blood Count	Delete

Рисунок 4.7 – Медична карта пацієнта

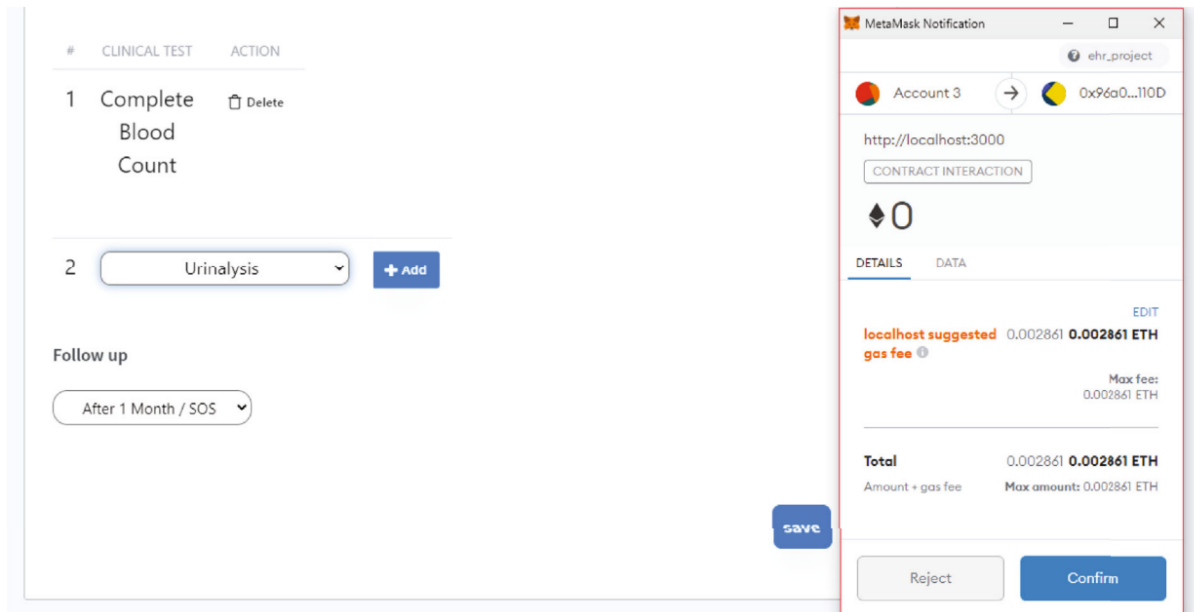


Рисунок 4.8 – Запис даних у смарт-контракт

На рисунку 4.8 показаний варіант збереження рецепта, а також смарт-контракт (MetaMask). Яку б інформацію лікар не додав, не змінив або не видалив з медичної карти пацієнта, все це буде збережено і захищено за допомогою смарт-контрактів, заснованих на гаманці Ethereum (ETH wallet).

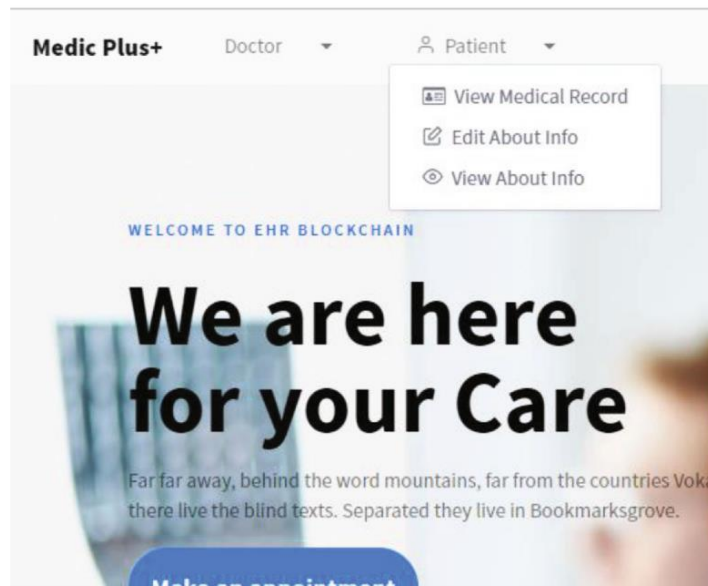


Рисунок 4.9 – Панель пацієнта

Панель пацієнта. На рисунку 4.9 показана панель пацієнта. Пацієнт

може переглядати та змінювати свою особисту інформацію на панелі пацієнта, якщо відбулися якісь зміни.

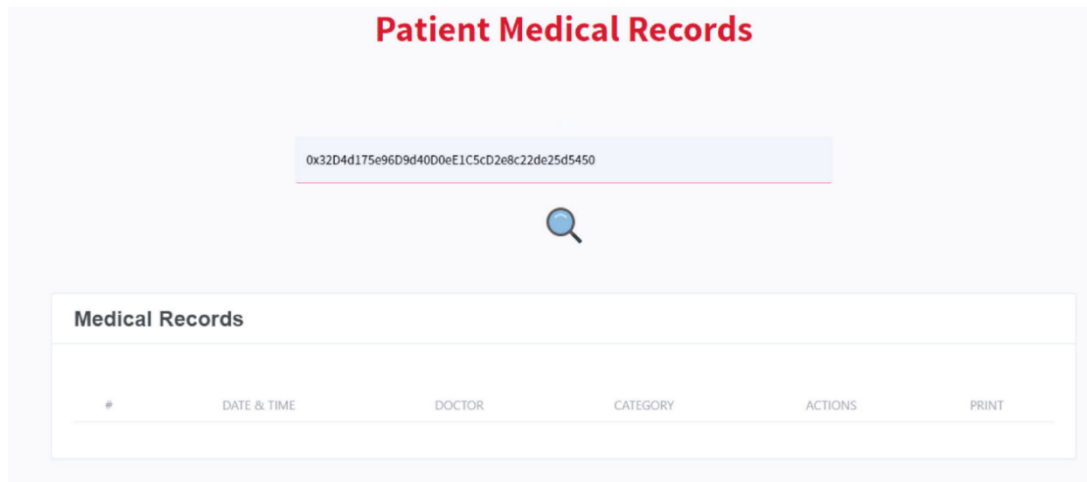


Рисунок 4.10 – Записи в медичній карті пацієнта

Тепер пацієнти можуть переглядати свої медичні записи, завантажені лікарями (рис.4.10). Пацієнт може ознайомитися з детальним призначенням лікаря, але не може вносити будь-які корективи в зв'язки з будь-яким правопорушенням.

4.1.2 Процес отримання доступу до пропонованої системи (серверна частина)

Рисунок 4.11 ілюструє персональний блокчейн Ethereum, представлений Ganache. Він був використаний для тестування і розгортання системи. Для тестування та локальної розробки Ganache пропонує кілька віртуальних облікових записів зі 100 ЕТН. Він забезпечує можливості, порівнянні з Ganache, при розгортанні в основній мережі Ethereum. Деякі з підроблених транзакцій виконуються віртуальними обліковими записами в додатку разом з хешами транзакцій і адресою контракту, на який вони були розгорнуті. Значення валюти кожної транзакції також відображається у стовпці.

Першим кроком на серверній частині є завантаження і установка Ganache з набору Truffle Suite.

Truffle Suite - це середовище розробки світового класу для децентралізованих додатків (dapps) і смарт-контрактів на блокчейні.

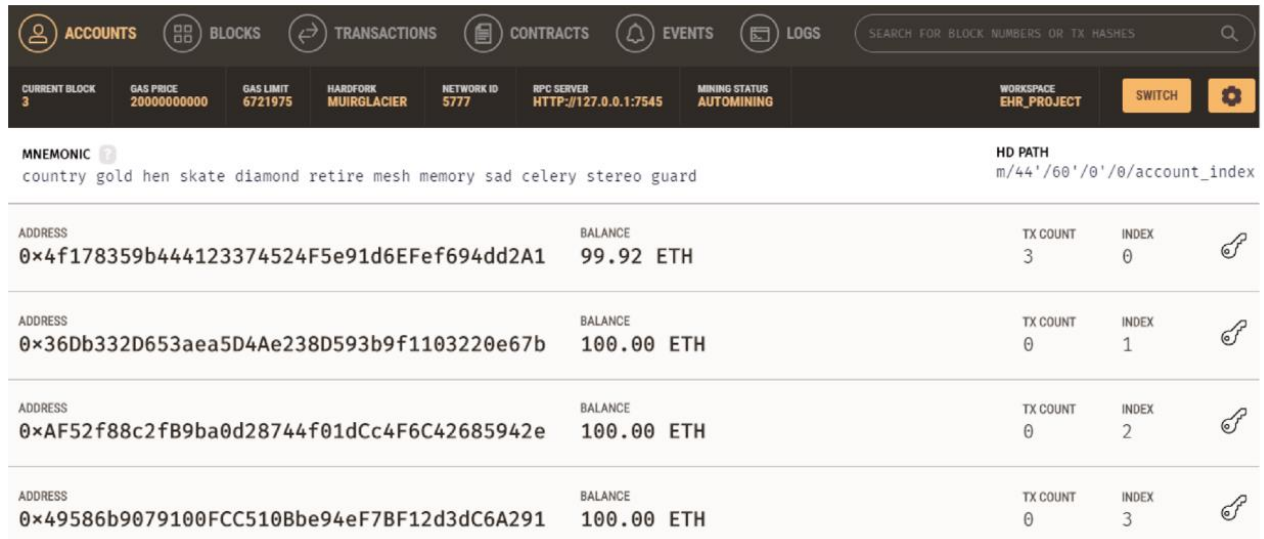


Рисунок 4.11 – Персональний блокчейн Ganache

Після завершення встановлення Ganache необхідно створити нову робочу область з назвою `ehr_project`. Потім додати Truffle з `truffle-config.js`. Таким чином, будуть показані захищені унікальні адреси, Індекс, баланс та відповідні приватні ключі для створення облікових записів для доступу до системи. Ganache збереже конфіденційність облікових записів і серійні номери в безпеці.

Підключення до сервера за допомогою Смарт-контракту. На рисунку 4.21 показано підключення системи за допомогою смарт-контрактів.

MetaMask використовується в якості смарт-контракту для підключення до системи. Створюються контракти, які надають метадані по назвах записів, доступі та цілісності даних. Додаються криптографічно підписані інструкції для управління цими характеристиками у блокчейн-транзакціях цієї системи. Використовуються лише операції, що забезпечують зміну даних функціональними можливостями переходу стану контракту для виконання

політик. До тих пір, поки медична карта може зберігатися в електронному вигляді, ці контракти можуть бути створені для забезпечення дотримання будь-якого набору правил, що контролюють її.

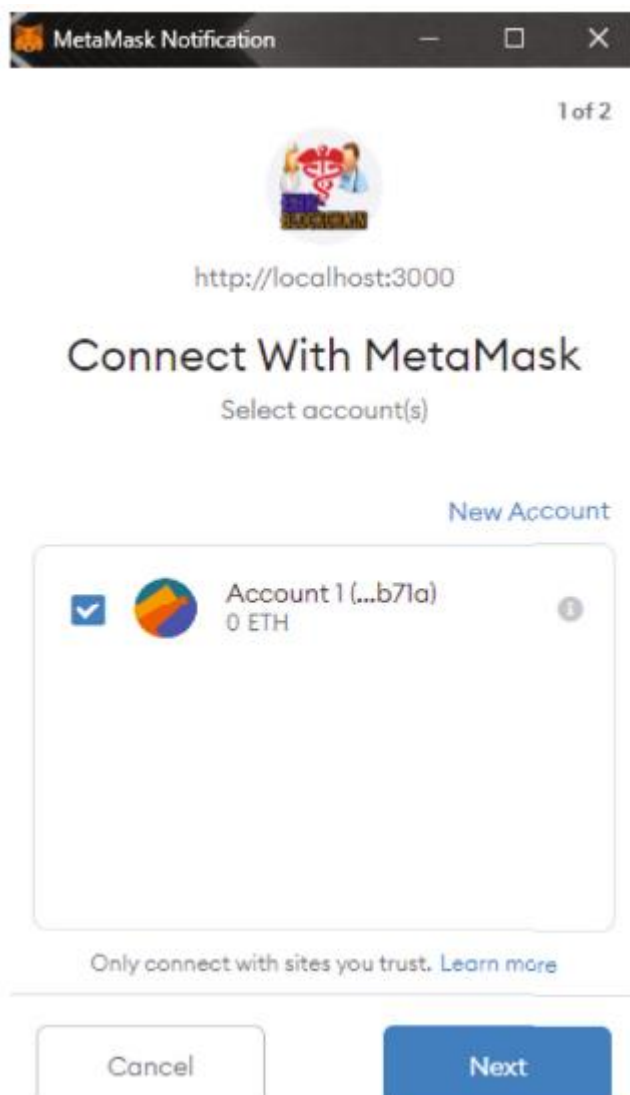


Рисунок 4.12 – Підключення до гаманця MetaMask.

Цей смарт-контракт, заснований на технології блокчейн, може бути розроблений таким чином, щоб включати в себе всі умови, такі як обробка різних дозволів та доступ до даних. Можна помітити, що в цій схемі задіяний ряд зацікавлених сторін, кожен з них виконує різні завдання. Це полегшить спілкування лікарів і пацієнтів. Смарт-контракти включають в себе правила авторизації даних. Це також може допомогти у відстеженні всіх дій,

пов'язаних з унікальним ідентифікатором, від точки походження до точки відправки. Були створені та пояснені різні ситуації, а також усі функції і процедури, які включені в смарт-контракти.

Ця функція може контролюватися і затверджуватися безпосередньо за допомогою смарт-контракту. Для цього немає необхідності в централізованому органі влади, що значно знижує витрати на адміністрування. Для підвищення ефективності всі дані медичної карти зберігаються в локальній базі даних.

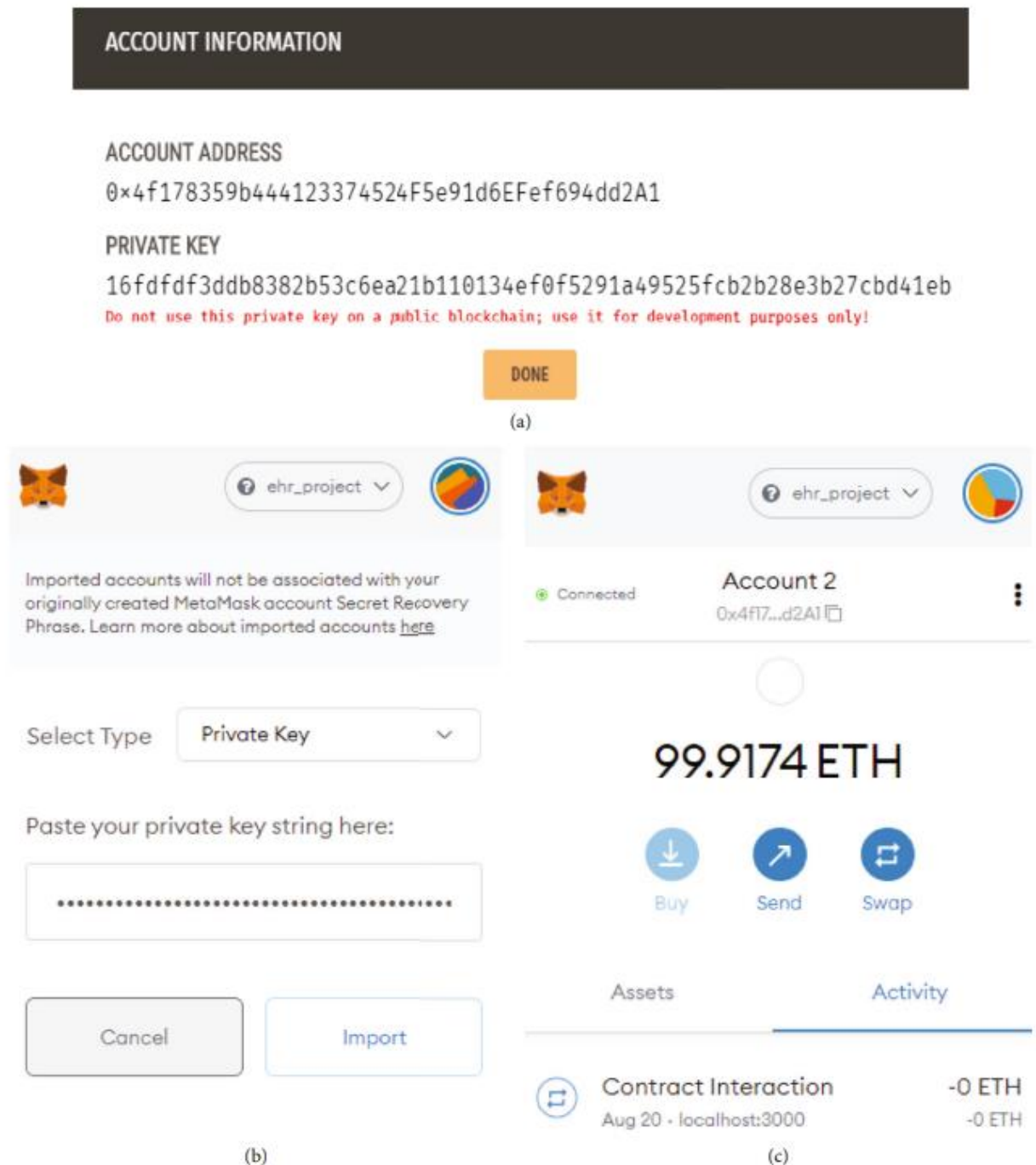


Рисунок 4.13 – а) Ganache Ethereum (адреса облікового запису), б) додавання приватного ключа та с) створення облікового запису

Створення облікового запису за допомогою Смарт-контракту. На рисунку 4.13 показано покроковий процес створення облікового запису, де

На рисунку 4.13.а показана інформація про обліковий запис Ganache Ethereum, який дозволяє користувачам використовувати браузер для доступу до децентралізованої системи без наявності повноцінного вузла блокчейна.

На рисунку 4.13.b показано імпортування приватних ключів через MetaMask, який гарантує можливість реалізації контрактів швидко. Гаманець Ethereum, призначено для вирішення проблеми неправильного використання криптографічних ключів. Захищений паролем Ethereum wallet - це частина програмного забезпечення, яке може використовуватися для зберігання секретних ключів і для підпису, авторизації та управління транзакціями з використанням електронних медичних карт. Секретні ключі адміністратора, лікаря і пацієнта будуть зберігатися в метамаск-гаманці, який може бути використаний скрізь, де потрібен дозвіл на використання закритого ключа. На рисунку 4.13.c показано ідентифікатор облікового запису із Ganache на гаманець Ethereum, де гаманець MetaMask потім підключиться до системи і буде функціонувати відповідно до ehr_project, а адміністратор отримає миттєвий доступ для входу через цей обліковий запис.

4.2 Аналіз отриманих результатів

Продемонструємо результати роботи вузлів системи за допомогою датчиків пульсу та датчиків визначення температури і вимірювання серцебиття з регулярним інтервалом. Також порівнюємо температуру, серцебиття і значення датчиків в реальному часі. Апаратні компоненти об'єднані з raspberry pi і Arduino Uno, щоб створити корисний інструмент моніторингу стану здоров'я.

Для вимірювання показників здоров'я в системі використовуються різні апаратні компоненти. Всі датчики, які підключені до головного вузлу, роблять систему більш функціональною. Пропонована система використовує біологічні датчики, перераховані нижче.

У цій роботі ми використовуємо датчик температури LM75. Він вставляється палець користувача для визначення температури тіла користувача.

Вихідний діапазон датчика LM75 становить від -40°C до 110°C .

Датчик серцебиття. Для вимірювання серцебиття використовується датчик пульсу. Кожну хвилину потрібно безперервне зчитування серцебиття користувача. Він має напругу +5 В або +3,3 Робоча напруга В і споживаний струм 4 мА.

Для оцінки дослідження та вивчення його ефективності використовується кілька експериментів. Під час проектів у пацієнтів вимірювалися температура тіла і серцебиття. Система проходить серію випробувань.

Показники здоров'я пацієнтів відображаються при оцінці якості дослідження.

Температура тіла пацієнта вимірюється за допомогою датчика температури тіла LM35, як показано у другому стовпці таблиці 4.1. У таблиці порівнюється температура, розрахована нашою системою, з реальною температурою різних користувачів, яка визначається термометром.

Середня точність обчислюється за допомогою (4.1).

$$Av = \frac{1}{n} \sum_n \frac{T_s}{T_t} \quad (4.1)$$

Av - середня точність.

n - кількість користувачів

T_s - Температура, виміряна системою

T_t - Температура, виміряна термометром вручну

Крім того, серцебиття пацієнта вимірюється за допомогою датчика пульсу, як показано у другому стовпці таблиці 4.2. У таблиці порівнюється серцебиття, розраховане нашою системою, з реальним серцебиттям різних користувачів. Середня точність обчислюється за допомогою (4.1).

Таблиця 4.1 – Порівняння вимірної температури з фактичною температурою

Користувач	Температура за датчиком LM75 ⁰ C	Температура за термометром ⁰ C	Середня точність
Користувач1	35,2	36,3	98.24%
Користувач2	38,5	38,1	
Користувач3	36,8	37,3	
Користувач4	34,6	36,6	
Користувач5	34,3	35,5	

Таблиця 4.2 – Порівняння між вимірним серцебиттям і фактичним серцебиттям

Користувач	Датчик пульсу (ударів в хвилину)	Серцебиття користувачів	Середня точність
Користувач1	75	77	97.23%
Користувач2	80	83	
Користувач3	91	93	
Користувач4	83	85	
Користувач5	93	96	

Порівняння продуктивності запропонованого методу та існуючих методів. В запропонованій системі використовується блокчейн заради безпеки даних і прозорість. У цій архітектурі будь-які спроби незаконного використання заборонені. Таким чином, система вважається однією з найбезпечніших платформ порівняно з іншими платформами електронної охорони здоров'я.

У цій частині ми порівнюємо архітектуру нашої системи з архітектурою існуючих систем охорони здоров'я. Порівняння базується на ключових характеристиках різних систем охорони здоров'я. Порівняння показано в таблиці 4.3.

Таблиця 4.3 – Порівняльна таблиця між запропонованим методом та існуючими методами

Параметри	Система моніторингу глюкози	Secure HER	healthcare provisioning ecosystem	Запропонована система
Дистанційне спостереження за пацієнтом	Ні	Ні	Так	Так
Дистанційне виявлення захворювань пацієнта	Ні	Ні	Ні	Ні
Використання датчиків	Так	Ні	Так	Так
Рекомендації щодо автентифікації користувача	Ні	Ні	Ні	Так
Зберігання даних в блокчейн	Так	Так	Так	Так
Використання криптографічних функцій	Ні	Так	Ні	Так
Використання файлової системи IPFS	Так	Ні	Так	Так

Спожитий газ за кожну транзакцію. Кожна транзакція в мережі Ethereum вимагає витрат на газ.

Іншими словами, газ - це свого роду готівкові гроші, які можуть бути використані для проведення будь-якої транзакції. Оскільки плата за газ в мережі Ethereum настільки висока, ми зберігаємо там не всі типи даних пацієнтів. В результаті ми вирішили зберегти в Ethereum тільки окремі хмарні адреси, що тягне за собою невелику плату за газ. З іншого боку, наша система є більш ефективною, ніж будь-яка інша система. Графік залежності плати за газ від розміру вхідних даних у байтах показано на рис. 4.14. На цьому графіку ми можемо наочно уявити, що при збільшенні розміру даних плата за газ автоматично збільшується. Таким чином, розмір даних, що зберігаються в блокчейн, повинен бути якомога менше.

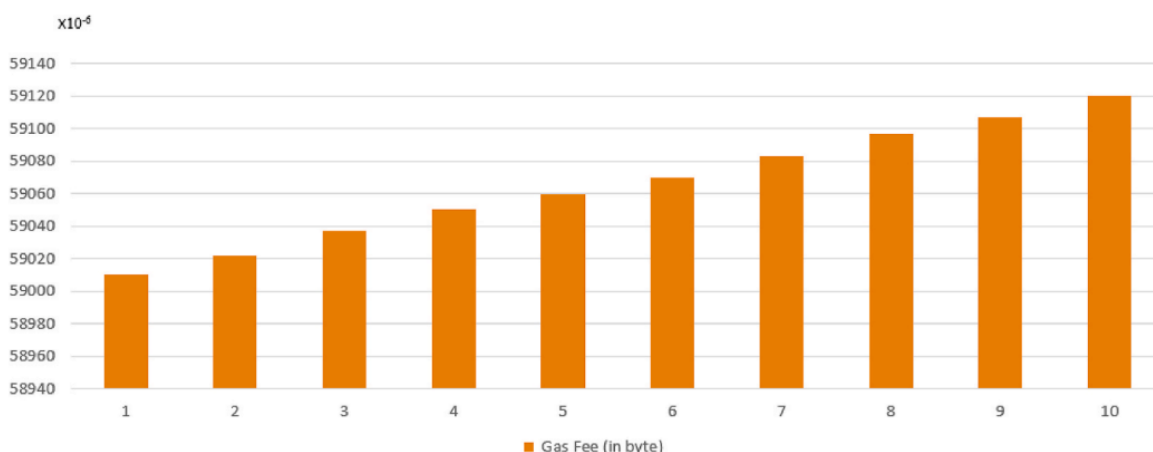


Рисунок 4.14 – Газ, який використовується під час транзакції в Ethereum

У секторі охорони здоров'я безпека даних сьогодні набуває все більшого визнання. Отож, щоб подолати цю проблему, ми запропонували технологію блокчейн з метою забезпечення конфіденційності. Ця ініціатива в першу чергу призначена для віддаленого догляду за пацієнтами, а також для пріоритетного захисту конфіденційних записів пацієнтів. Для забезпечення безпеки даних ми впровадили блокчейн Ethereum, який вважається одним з найбільш ефективних методів, використовуваних для забезпечення безпеки даних. Ми також використовували асиметричну криптографію для методу хешування, який робить транзакції більш безпечними, ніж інші системи.

У майбутньому до системи можуть бути додані нові датчики, такі як датчик акселерометра, датчики точного визначення місця розташування пацієнтів та інші.

ВИСНОВКИ

Традиційний підхід до ведення медичної документації характеризується неефективністю, оскільки вимагає істотного виділення місця для зберігання великої колекції даних медичних аналізів для кожного пацієнта. Відсутність структури в даних попередніх систем робило передачу інформації неможливою. Враховуючи значну кількість даних, що генеруються в секторі охорони здоров'я, важливо розглянути можливість вдосконалення наших підходів до управління даними, одночасно забезпечуючи безпеку та конфіденційність даних.

У зв'язку з імперативним характером конфіденційності даних будуть внесені подальші зміни. Впровадження цієї зміни породжує багато проблем, які потребують уваги, і блокчейн ефективно вирішує основні проблеми. Вищезгаданий метод дозволяє пацієнтам легко надавати або відкликати згоду на певні записи відповідним органам простим дотиком. Впровадження цієї автоматизації було значно полегшено в результаті використання Ethereum і смарт-контрактів. Пропонований гаманець функціонує як посередник для полегшення безпечного і простого входу в блокчейн, а також забезпечує легке управління секретними ключами. Крім того, це може служити засобом підключення для пацієнтів, які можуть мати застереження щодо передачі своїх медичних даних в електронні медичні записи (EHRs). Криптографічні методи шифрування, використовувані системою, характеризуються своєю надійністю і стійкістю до несанкціонованого доступу, що забезпечує як безпеку, так і надійність. Було встановлено, що система успішно виконала більшість завдань проекту, зокрема, безпечний обмін аутентифікаційними даними для медичних звітів з використанням технології блокчейн. Очікується, що реалізація проекту буде ефективно задовольняти вимогам користувачів. Ефективне впровадження технології блокчейн полегшило завершення аутентифікації, обміну даними та заходів безпеки для медичних звітів. Система також вирішує проблеми, пов'язані з прямою передачею

захворювань в медичних установах, наприклад, з нинішньою пандемією Covid-19. В першу чергу це досягається за рахунок використання фізичних копій медичних записів і визнання підвищеного ризику, пов'язаного з подальшою передачею інфекції від людини до людини.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Cresswell K, Majeed A, Bates D, Sheikh A (2012) Computerised decision support systems for healthcare professionals: an interpretative review. *J Innov Health Inform* 20(2):115–128. <https://doi.org/10.14236/jhi.v20i2.32>
2. Drozd J, Drozd A, Al-dhabi M (2015) A resource approach to on-line testing of computing circuits. In: Paper presented at the IEEE East-West design & test symposium, Batumi, Georgia, 26–29 September 2015, pp 276–281. <https://doi.org/10.1109/EWDTS.2015.7493122>
3. Hovorushchenko T (2018) Information technology for assurance of veracity of quality information in the software requirements specification. *Adv Intell Syst Comput* 689:166–185. https://doi.org/10.1007/978-3-319-70581-1_12
4. Hovorushchenko T, Herts A, Hnatchuk Y (2019) Concept of intelligent decision support system in the legal regulation of the surrogate motherhood. In: *CEUR-WS*, vol 2488, pp 57–68
5. Collen M, Ball M (2015) Medical informatics: past and future. History of medical informatics in the United States, pp 725–748. https://doi.org/10.1007/978-1-4471-6732-7_18
6. Zheng Z, Xie S, Dai H, Chen X, Wang H (2018) Blockchain challenges and opportunities: a survey. *Int J Web Grid Serv* 14(4):352–375. <https://doi.org/10.1504/IJWGS.2018.095647>
7. Zeng Y, Zhang Y (2019) Review of research on blockchain application development method. *J Phys Conf Ser* 1187:052005. <https://doi.org/10.1088/1742-6596/1187/5/052005>
8. Xu Z, Zou C (2021) What can blockchain do and cannot do? *China Econ J* 14(1):4–25. <https://doi.org/10.1080/17538963.2020.1748968>
9. Pal O, Alam B, Thakur V, Singh S (2021) Key management for blockchain technology. *ICT Express* 7(1):76–80. <https://doi.org/10.1016/j.icte.2019.08.002>

10. Li W, He M, Sang H (2021) An overview of blockchain technology: applications, challenges and future trends. In: Paper presented at the IEEE international conference on electronics information and emergency communication, Beijing, China, 18–20 June 2021, pp 31–39. <https://doi.org/10.1109/ICEIEC51955.2021.9463842>
11. Kuo T, Rojas H, Ohno-Machado L (2019) Comparison of blockchain platforms: a systematic review and healthcare examples. *J Am Med Inform Assoc* 26(5):462–478. <https://doi.org/10.1093/jamia/ocy185>
12. Zaitseva E, Levashenko V (2016) Construction of a reliability structure function based on uncertain data. *IEEE Trans Reliab* 65(4):1710–1723. <https://doi.org/10.1109/TR.2016.2578948>
13. Levashenko V, Zaitseva E, Kvassay M, Deserno T (2016) Reliability estimation of healthcare systems using fuzzy decision trees. In: Paper presented at the Federated Conference on Computer Science and Information Systems, Lviv, Ukraine, 23–25 September 2016, pp 331–340. <https://doi.org/10.15439/2016F150>
14. Zaitseva E, Levashenko V, Rabcan J, Krsak E (2020) Application of the structure function in the evaluation of the human factor in healthcare. *Symmetry* 12(1):93. <https://doi.org/10.3390/sym12010093>
15. Transform healthcare outcomes with the simplicity of IBM Blockchain. <https://www.ibm.com/downloads/cas/DQPLDP8N>
16. Syed T, Alzahrani A, Jan S, Siddiqui M, Nadeem A, Alghamdi T (2019) A comparative analysis of blockchain architecture and its applications: problems and recommendations. *IEEE Access* 7:176838–176869. <https://doi.org/10.1109/ACCESS.2019.2957660>
17. Zheng Z, Xie S, Dai H, Chen X, Wang H (2017) An overview of blockchain technology: architecture, consensus, and future trends. In: Paper presented at the IEEE international congress on big data, Honolulu, Hawaii, 25–30 June 2017, pp 557–564. <https://doi.org/10.1109/BigDataCongress.2017.85>
18. Li X, Jiang P, Chen T, Luo X, Wen Q (2017) A survey on the security of blockchain systems. *Future Gener Comput Syst Int J Esci* 107:841–853.

<https://doi.org/10.1016/j.future.2017.08.020>

19. Alexander C, Wang L (2019) Cybersecurity, information assurance, and big data based on blockchain. In: Paper presented at the IEEE Southeast conference, Huntsville, USA, 11–14 April 2019

20. Westphal E, Seitz H (2021) Digital and decentralized management of patient data in healthcare using blockchain implementations. *Front Blockchain* 4:732112. <https://doi.org/10.3389/fbloc.2021.732112>

21. Kim S, Huh J (2020) Artificial neural network blockchain techniques for healthcare system: focusing on the personal health records. *Electronics* 9(5):763. <https://doi.org/10.3390/electronics9050763>

22. Chen Y, Meng L, Zhou H, Xue G (2021) A blockchain-based medical data sharing mechanism with attribute-based access control and privacy protection. *Wirel Commun Mobile Comput* 2021:6685762. <https://doi.org/10.1155/2021/6685762>

23. Zhang L, Peng M, Wang W, Su Y, Cui S, Kim S (2021) Secure and efficient data storage and sharing scheme based on double 123 16 *Journal of Reliable Intelligent Environments* (2023) 9:5–16 blockchain. *CMC Comput Mater Continua* 66(1):499–515. <https://doi.org/10.32604/cmc.2020.012205>

24. Chen F, Huang J, Wang C, Tang Y, Huang C, Xie D, Wang T, Zhao C (2021) Data access control based on blockchain in medical cyber physical systems. *Secur Commun Netw* 2021:3395537. <https://doi.org/10.1155/2021/3395537>

25. Cao Y, Sun Y, Min J (2020) Hybrid blockchain-based privacy-preserving electronic medical records sharing scheme across medical information control system. *Meas Control* 53(7–8):1286–1299. <https://doi.org/10.1177/0020294020926636>

26. Kim T, Lee S, Chang D, Koo J, Kim T, Yoon K, Choi I (2021) DynamiChain: development of medical blockchain ecosystem based on dynamic consent system. *Appl Sci Basel* 11(4):1612. <https://doi.org/10.3390/app11041612>

27. Zou R, Lv X, Zhao J (2021) SPChain: blockchain-based medical data sharing and privacy-preserving eHealth system. *Inf Process Manag* 58(4):102604.

<https://doi.org/10.1016/j.ipm.2021.102604>

28 Hovorushchenko T, Hnatchuk Y, Herts A, Moskalenko A, Osyadlyi V (2021) Theoretical and applied principles of information technology for supporting medical decision-making taking into account the legal basis. In: CEUR-WS, vol 3038, pp 172–181