МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

НІКОНЧУК СНІЖАНА ОЛЕКСАНДРІВНА

Допускається до захисту:
В.о. завідувача кафедри
прикладної математики та
кібербезпеки,
_____Луценко А.В
«__»_____20__р.

ОРГАНІЗАЦІЯ БАГАТОРІВНЕВОГО ДОСТУПУ У СИСТЕМІ ЗБЕРІГАННЯ МЕДИЧНИХ ДАНИХ

Спеціальність 125 Кібербезпека
Кваліфікаційна (бакалаврська) робота

Науковий керівник:
Чернов Д.В.,
Канд. техн. наук, доцент, доцент кафедри прикладної математики та кібербезпеки
_____
(підпис)

Оцінка : ___/ ___ / _____
(бали/за шкалою ЄКТС/за національною шкалою)
Голова ЕК: _____
(підпис)

Вінниця – 2025

АНОТАЦІЯ

Нікончук С.О. Організація багаторівневого доступу у системі зберігання медичних даних. Спеціальність: 125 — Кібербезпека. Донецький національний університет імені Василя Стуса, Вінниця, 2025.

У межах бакалаврської роботи розроблено модель багаторівневого доступу до медичних даних з урахуванням специфіки медичних інформаційних систем. Запропоновано архітектуру системи з використанням рольової моделі RBAC, списків контролю доступу (ACL), двофакторної автентифікації та журналювання дій користувачів. Реалізовано програмний застосунок, що демонструє функціонування системи, та сформовано політику безпеки відповідно до вимог законодавства України, GDPR та стандарту ISO/IEC 27001. Проведено оцінку ризиків, зокрема технічних каналів витоку інформації.

Ключові слова: багаторівневий доступ, медичні дані, інформаційна безпека, RBAC, ACL, 2FA, політика доступу.

Структура роботи: анотація, вступ, три розділи, висновки, список використаних джерел, додатки.

45 с., 5 табл., 4 рис., 5 дод., 9 джерел.

ABSTRACT

Nikonchuk S. O. Organization of Multi-Level Access in a Medical Data Storage System. Specialty: 125 — Cybersecurity. Vasyl Stus Donetsk National University, Vinnytsia, 2025.

As part of the bachelor's thesis, a model of multi-level access to medical data was developed, taking into account the specifics of medical information systems. The proposed system architecture is based on the role-based access control model (RBAC), access control lists (ACL), two-factor authentication (2FA), and user activity logging. A software prototype was implemented to demonstrate the system's functionality, and a security policy was developed in compliance with the legislation of Ukraine, the GDPR, and the ISO/IEC 27001 standard. A risk assessment was conducted, including the evaluation of technical channels of information leakage.

Keywords: multi-level access, medical data, information security, RBAC, ACL, 2FA, access policy.

Structure of the thesis: abstract, introduction, three chapters, conclusions, list of references, appendices.

45 p., 5 tables, 4 figures, 5 appendix, 9 sources.

ЗМІСТ

ВСТУП

Сучасний розвиток інформаційних технологій та зростання обсягу медичних даних обумовлюють необхідність забезпечення їхньої конфіденційності, цілісності та доступності. Медична інформація містить персональні дані пацієнтів, історії хвороб, результати аналізів та інші критично важливі відомості, які потребують надійного захисту. Несанкціонований доступ до таких даних може призвести до витоку конфіденційної інформації, фінансових втрат, юридичних наслідків та порушення довіри пацієнтів.

Організація багаторівневого доступу в системах зберігання медичних даних є ключовим елементом захисту, що дозволяє контролювати права доступу користувачів відповідно до їхніх посадових обов'язків і рівня авторизації. Такий підхід передбачає впровадження сучасних методів аутентифікації, розмежування прав доступу та моніторингу активності користувачів. Важливу роль у забезпеченні безпеки відіграє Комплексна система захисту інформації (КСЗІ), яка дозволяє запроваджувати ефективні механізми управління доступом та запобігати несанкціонованому втручанню в медичні системи. Однак ефективне функціонування КСЗІ можливе лише за умови розробки детального плану організації багаторівневого доступу, що визначає принципи та методи його реалізації.

Галузь застосування цієї розробки охоплює медичні установи, страхові компанії, фармацевтичні підприємства та інші організації, що працюють з медичними даними та потребують чіткої системи контролю доступу. План організації багаторівневого доступу є основним документом, що регламентує політику безпеки організації та порядок розподілу доступу до медичної інформації.

Метою бакалаврської роботи є розробка системи багаторівневого доступу до медичних даних, яка дозволить забезпечити захист інформації від внутрішніх і зовнішніх загроз та гарантуватиме дотримання принципів

конфіденційності та доступності. Для досягнення цієї мети необхідно виконати такі завдання:

– проаналізувати основні загрози, пов’язані з доступом до медичних даних;

– дослідити сучасні методи та технології контролю доступу;

– розробити модель багаторівневого доступу для системи зберігання медичних даних;

– визначити порядок впровадження та контролю ефективності запропонованої моделі.

Об’єкт дослідження:

Процес забезпечення захищеного доступу до медичних даних у медичних інформаційних системах.

Предмет дослідження:

Методи, моделі та засоби реалізації багаторівневого контролю доступу до конфіденційної інформації в системах зберігання медичних даних.

Актуальність даної роботи зумовлена зростанням кількості кіберзагроз у сфері охорони здоров’я та необхідністю впровадження ефективних механізмів захисту інформації. Розробка плану організації багаторівневого доступу сприятиме мінімізації ризиків несанкціонованого доступу до медичних даних та забезпеченню їхньої безпеки відповідно до сучасних стандартів інформаційної безпеки.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИМОГ ДО СИСТЕМ ЗБЕРІГАННЯ МЕДИЧНИХ ДАНИХ

Невід'ємною складовою комплексної стратегії інформаційної безпеки є аналіз структури, інформаційних потоків та бізнес-процесів підприємства з медичними даними. Цей аналіз має на меті забезпечити фундаментальне розуміння тонкощів організації, що дозволить виявити потенційні вразливості та ризики в її інформаційних та комунікаційних системах.

1.1 Особливості медичних даних та їх обробки

Лікарня Медичні дані є одним із найбільш чутливих типів інформації, оскільки містять відомості про стан здоров’я, діагнози, результати обстежень, лікування, генетичну інформацію та інші аспекти, що становлять лікарську таємницю. Їх обробка вимагає суворого дотримання принципів конфіденційності, цілісності та доступності, а також законодавчих норм щодо захисту персональних даних.

Особливої актуальності набуває питання безпечного зберігання медичних даних в умовах зростання обсягів цифрової інформації. Ультразвукові, рентгенівські, МРТ- та КТ-зображення є прикладом важливих медичних даних, які повинні бути захищеними від стороннього доступу та спотворення. Згідно з публікацією в журналі Biomedical Engineering and Technology, ефективне управління такими зображеннями відіграє ключову роль у прийнятті клінічних рішень і може бути підсилене технологіями штучного інтелекту.

1.2 Проблеми конфіденційності та безпеки в медичній сфері

Медична інформація — одна з найцінніших і найвразливіших категорій персональних даних. Порушення її конфіденційності або цілісності може мати серйозні соціальні, юридичні та навіть життєві наслідки як для пацієнтів, так і для медичних установ. Основними загрозами витоку медичних даних може бути витік медичних даних як підґрунття для дискримінації. Інформація про

ВІЛ-статус, психічні захворювання, залежності чи онкологічні діагнози, потрапивши у відкритий доступ, може призвести до соціальної стигматизації, втрати роботsи, порушення особистого життя чи репутаційної шкоди.

По-друге, існує загроза фінансового шахрайства. Зловмисники можуть використати викрадені медичні дані для створення фіктивних страхових справ, підробки рецептів, отримання кредитів або відшкодування витрат за неіснуючі послуги. Дані пацієнтів мають високу цінність на «темному ринку» саме через можливість такої незаконної монетизації.

Також, модифікація або втрата медичних даних без належного резервного копіювання може поставити під загрозу здоров'я або навіть життя пацієнтів. Наприклад, зміна інформації про алергію, лікарські призначення чи діагноз здатна призвести до помилок у лікуванні або несумісних процедур.

Варто зазначити, що порушення безпеки інформаційних систем охорони здоров'я здатне паралізувати роботу закладу. Кіберзлочинці, отримавши доступ до медичних баз, можуть зашифрувати їх за допомогою програм-вимагачів, вимагаючи викуп. Такі інциденти вже траплялися в лікарнях Великої Британії, США, Франції та інших країн. У випадку відмови від виплати заклад втрачає дані і змушений відновлювати функціонування з нуля.

Також, недотримання вимог щодо захисту даних є підставою для юридичної відповідальності. Це може спричинити великі штрафи, втрату ліцензії, судові позови з боку постраждалих пацієнтів і суттєві репутаційні втрати.

Таким чином, загрози для конфіденційності, цілісності та доступності медичних даних мають багатовимірний характер. Тому надійна система захисту повинна поєднувати технічні, організаційні та правові механізми безпеки, а також базуватись на принципах багаторівневого доступу.

1.3 Регуляторні вимоги та стандарти

Збереження та обробка медичних даних повинні здійснюватися відповідно до чинного законодавства та міжнародних стандартів інформаційної безпеки. Це обумовлено як високою чутливістю медичної інформації, так і необхідністю

забезпечення безперервності медичних послуг, де навіть незначні порушення можуть мати серйозні наслідки для пацієнтів.

Одним із ключових нормативних актів у сфері захисту персональних даних є GDPR (General Data Protection Regulation, Регламент ЄС №2016/679), що набув чинності у 2018 році. Він встановлює суворі правила зберігання, обробки та передачі персональних даних фізичних осіб, включаючи медичну інформацію. Згідно з GDPR, медичні дані належать до особливо захищеної категорії, а їх обробка вимагає чіткої правової підстави, згоди суб'єкта або наявності важливих суспільних інтересів (наприклад, громадське здоров'я, наукові дослідження). З його вимог можна зазначити реалізацію принципу «privacy by design» - конфіденційність повинна бути врахована ще на етапі проектування системи, також пацієнт повинен мати доступ до своїх даних, їх виправлення, або повне видалення. GDPR вимагає про повідомлення пацієнтів та персоналу про витік даних протягом 72 годин з моменту витоку. Також нормативний акт надає зобов'язання мати внутрішні політики безпеки та проводити оцінку впилву на захист даних у разі ризикованих обробок.

Іншим важливим стандартом є ISO/IEC 27001 — міжнародний стандарт системи управління інформаційною безпекою (ISMS). Він визначає вимоги до розробки, впровадження, моніторингу та вдосконалення політик і процедур безпеки. Організації, які мають сертифікат відповідності ISO 27001, вважаються такими, що надійно захищають інформацію. В медичній галузі це дозволяє забезпечити довіру пацієнтів та партнерів, а також зменшити ризик інцидентів.

В Україні головним нормативним актом є Закон України «Про захист персональних даних», який встановлює базові вимоги до обробки всіх категорій персональних даних, включаючи медичні. Оскільки медичні дані належать до категорії з обмеженим доступом, їх обробка повинна здійснюватись лише за згодою суб'єкта даних або на підставі спеціального закону.

Окрім цього, Концепція розвитку електронної охорони здоров'я (E-health), затверджена Міністерством охорони здоров'я, передбачає створення Єдиного

медичного інформаційного простору. У межах цієї системи працює Центральна база даних ЕСОЗ, яка інтегрує численні медичні інформаційні системи (МІС), реєстри пацієнтів, декларації, електронні рецепти, медичні висновки тощо. Такі системи повинні відповідати вимогам технічного захисту інформації та проходити тестування на відповідність критеріям безпеки.

Практичне значення регуляторних вимог можна віднести до розробників медичних систем і адміністраторів, вони визначають:

- архітектуру програмного забезпечення (наприклад, модульність, підтримка шифрування);
- порядок дій у разі витоку чи інциденту;
- функціонал системи доступу та реєстрації активності користувачів;
- необхідність залучення спеціалістів з кібербезпеки на етапі проектування та впровадження.

1.4 Потреба в багаторівневому доступі

У сфері охорони здоров’я інформаційні системи повинні обслуговувати широкий спектр користувачів, кожен з яких має різні функції, обов’язки та, відповідно, потреби в доступі до даних. Надання всім користувачам повного доступу створює загрозу витоку даних, порушення лікарської таємниці, а також збільшує ймовірність помилок. Саме тому багаторівнева система доступу є ключовою складовою безпеки.

Категорії користувачів, що у типовій медичній системі мають доступ до певної інформації можна побачити у таблиці 1.1.

Таблиця 1 – Категорії користувачів у медичній системі

Категорія персоналу	Роль у системі	Обмеження доступу
Адміністративний персонал	Доступ до статистичних даних, фінансової звітності, управлінських звітів, організаційної структури	Без доступу до персональних медичних записів пацієнтів
Медичний персонал (лікарі, головний лікар, керівники відділень)	Доступ до електронних медичних карток, історії хвороби, результатів обстежень, призначень	Обсяг доступу визначається відповідно до функціональної ролі

Таблиця 1 – Категорії користувачів у медичній системі

Категорія персоналу	Роль у системі	Обмеження доступу
Фахівці IT-відділу (системні адміністратори, фахівці з кібербезпеки)	Доступ до серверів, систем автентифікації, журналів подій, резервних копій, інтерфейсів управління системою	Не мають доступу до вмісту медичних карток
Кадрова служба (керівник, працівники відділу кадрів)	Доступ до особових справ персоналу, даних про графіки, навантаження, дисципліну	Без доступу до медичних і фінансових даних
Фінансовий відділ (головний бухгалтер)	Доступ до інформації про виплати, кошторис, закупівлі, облікові дані	Без доступу до клінічної (медичної) інформації
Охорона та служба безпеки (керівник служби, охоронці)	Доступ до систем відеоспостереження, журналів подій, обліку входу/виходу	Без доступу до будь-яких медичних, фінансових або кадрових даних

На основі цього можна скласти матрицю доступу, що відобразить, які саме категорії персоналу мають доступ до відповідних інформаційних блоків, що в майбутньому дозволить формалізувати політику безпеки, виявити надмірний, або зайвий доступ, а також налаштувати систему так, щоб відповідати принципу мінімальних привілеїв згідно GDPR – кожен користувач має лише той доступ, який йому необхідний для виконання службових обов’язків. Це й дозволить захистити персональні дані пацієнтів, забезпечить юридичну відповідність GDPR, зменшить ризики внутрішніх порушень та покращить контроль за доступом до інформації.

2 РОЗРОБКА БАГАТОРІВНЕВОЇ СИСТЕМИ ДОСТУПУ

2.1 Класифікація інформації що обробляється в АС

Медичні інформаційні системи (МІС) оперують великими обсягами різнорідних даних, які вимагають чіткого структурування та розмежування за рівнем чутливості й доступу. Ефективна класифікація таких даних є основою для реалізації багаторівневої системи доступу в медичних закладах.

У межах медичного закладу, інформація умовно поділяється на три основні категорії за ступенем критичності:

- Високочутлива інформація;
- Внутрішньо-службова інформація;
- Адміністративно-фінансова інформація.

Високочутлива інформація включає в себе електронні медичні картки пацієнтів, персональні дані пацієнтів, історію хвороб, результати діагностичних досліджень, записи про лікування, психіатрічні, онкологічні та інші делікатні медичні дані. Ця інформація підпадає під найвищий рівень захисту, оскільки її витік може спричинити серйозні етичні, правові та соціальні наслідки[1].

До внутрішньо-службової інформації належать облікові записи співробітників у системі, службові переписки, дані про навантаження та графіки роботи, а також аудиторські лог-файли дій користувачів. До цієї інформації доступ мають лише обмежені категорії персоналу, такі як: ІТ-відділ, керівники. Така інформація повинна бути захищена від модифікації, або втрати.

Адміністративно-фінансова інформація тримає в собі бюджетні та платіжні документи, інформацію про закупівлі, звітність, фінансові операції, а також персональні дані співробітників, пов’язані з оплатою праці (ІПН, банківські реквізити).

Дані різних рівнів зберігаються як у локальних системах (сервери, NAS-сховища), так і в хмарних рішеннях, пов’язаних із ЕСОЗ та іншими зовнішніми сервісами. Саме розподіленість джерел та складність взаємодії між модулями підвищують вимоги до якісної реалізації системи розмежування доступу[2].

Для забезпечення надійного захисту цих даних на практиці застосовується градація рівнів доступу, яка дозволяє визначити, які дії (читання, запис, створення) може виконувати кожен користувач у межах своєї ролі. Ця логіка реалізується через рольову модель доступу, що буде розглянута в наступних підрозділах.

2.2 Користувачі системи та їхні рівні доступу

Раніше було визначено перелік ключових категорій персоналу, що взаємодіють із медичною інформаційною системою. Для кожної ролі визначено функціональні обов'язки та рівень доступу до різних типів даних. На основі цієї класифікації формується багаторівнева система доступу, що забезпечує ізоляцію критичних даних між підрозділами, прозорість у розмежуванні обов'язків, контроль за доступом до конфіденційної інформації та підзвітність дій користувачів.

Для реалізації цього підходу було використано матрицю доступу, яка визначає, які саме дії може виконувати кожна роль у відношенні до конкретного класу даних. Система розрізняє три основні типи дій:

- Читання – перегляд інформації;
- Запис – редагування або оновлення інформації;
- Створення – внесення нових записів.

Наступним кроком визначено структурні підрозділи на досліджуваному підприємстві та виділено наступні групи:

- Адміністративний персонал
- Медичний персонал
- Фахівці ІТ-відділу
- Кадрова служба

В інформаційній мережі лікарні відбувається обробка великого обсягу інформації що підлягає захисту. До цієї інформації належать:

- Електронні медичні картки пацієнтів
- Бази персональних даних пацієнтів

– Результати лабораторних досліджень та діагностичних обстежень
– Дані про перебіг лікування, призначення, медичні протоколи
– Облікові записи співробітників у медичних системах
– Фінансова документація
– Персональні дані співробітників

Модель розмежування доступу базується на принципі мінімальних необхідних прав (principle of least privilege) [3], що є основоположним у сфері інформаційної безпеки. Такий підхід знижує ризик як навмисних, так і випадкових витоків даних, забезпечує відповідність міжнародним стандартам безпеки, таким як ISO/IEC 27001 та вимогам GDPR [4].

Таблиця 2 – Матриця доступу

Роль / Дані	Електронні медичні картки пацієнтів	Бази персональних даних пацієнтів	Результати лабораторних досліджень	Дані про лікування, призначення	Облікові записи співробітників	Фінансова документація	Персональні дані співробітників
Генеральний директор	r	r	-	-	r	rwc	r
Заступник директора	r	r	-	-	r	rw	r
Адміністратор	-	-	-	-	rwc	-	-
Головний лікар	rwc	rw	rwc	rwc	r	-	r
Керівник відділення	rw	r	rw	rw	-	-	-
Керівник IT-відділу	-	r	-	-	rwc	r	rw
Системний адміністратор	-	r	-	-	rwc	rw	rw
Фахівець з кібербезпеки	-	r	-	-	rw	r	rw

Таблиця 2 – Матриця доступу

Роль / Дані	Електронні медичні картки пацієнтів	Бази персональних даних пацієнтів	Результати лабораторних досліджень	Дані про лікування, призначення	Облікові записи співробітників	Фінансова документація	Персональні дані співробітників
Керівник служби охорони	-	-	-	-	r	-	-
Охоронець	-	-	-	-	r	-	-
Керівник відділу кадрів	-	-	-	-	rwc	-	rwc
Працівник відділу кадрів	-	-	-	-	rw	-	rw
Лікар	rwc	r	rwc	rwc	-	-	-
Головний бухгалтер	-	-	-	-	-	rwc	r

Як видно з таблиці, рівень доступу персоналу до інформаційних ресурсів чітко розмежовано відповідно до посадових обов'язків. Це дозволяє забезпечити захист найбільш критичних даних, таких як електронні медичні картки, результати обстежень, а також персональна та фінансова інформація як пацієнтів, так і працівників. Правильно сформована матриця доступу є основою для подальшої реалізації рольової моделі управління доступом (RBAC).

2.3 Рольова модель управління доступом

Реалізація багаторівневої системи доступу в медичній установі базується на рольовій моделі керування доступом (RBAC), яка найбільш ефективно відповідає потребам галузі охорони здоров'я.

На практиці це означає, що всі дії користувачів у медичній інформаційній системі обумовлені їхньою посадою та функціональними обов'язками. Наприклад, лікар може створювати, читати та редагувати записи про лікування

лише своїх пацієнтів, головний бухгалтер має доступ лише до фінансових модулів і не бачить жодних медичних карток, Адміністратор МІС має змогу керувати обліковими записами, але не взаємодіє з медичними чи персональними даними.

Ролі ж формуються на основі внутрішнього регламенту закладу та узгоджуються з ІТ-відділом. Це дозволяє швидко додати нових співробітників до системи, оновлювати повноваження при зміні посади або переведенні, виключити ситуації, коли новий або звільнений працівник має надлишкові або застарілі права.

Для кожної ролі заздалегідь визначено набір дозволів на виконання конкретних дій у межах системи, це читання, запис або створення. Ця логіка вже реалізована у матриці доступу, поданій у попередньому підрозділі, де чітко окреслено, хто з персоналу має доступ до кожного з ключових інформаційних блоків системи. Такий підхід дозволяє дотримуватись принципу мінімального доступу і забезпечує відповідність чинному законодавству та вимогам електронної системи охорони здоров'я (ЕСОЗ) [1].

На етапі розробки система розглядається як гнучка та масштабована: нові ролі можна додавати без перебудови архітектури, а дозволи – оновлювати централізовано через модулі адміністрування. Для цього застосовуються засоби медичних інформаційних систем, які вже мають вбудовану підтримку RBAC, або стороннє ПЗ із сумісністю до стандартів HL7, FHIR тощо. [5].

Таким чином, рольова модель не лише полегшує управління доступом, а й забезпечує надійну, контрольовану та перевірювану схему захисту даних у медичному закладі.

2.4 Розробка структури моделі системи багаторівневного доступу.

Ефективна реалізація багаторівневої системи доступу до медичних даних неможлива без належної технічної інфраструктури, яка забезпечує надійність, безперервність і захищеність обміну інформацією між користувачами системи.

У медичному закладі система доступу реалізується через інтеграцію програмно-апаратних засобів, що формують єдине захищене середовище. До її складу входять:

- Серверне середовище;
- Клієнтські пристрої;
- Засоби автентифікації та авторизації;
- Резервне копіювання та NAS-сховища;

Серверне середовище включає в себе сервери медичної інформаційної системи, де зберігаються бази даних пацієнтів, призначень, медичних висновків, фінансових і кадрових даних. Сервери розміщені у спеціалізованому серверному приміщенні з обмеженим фізичним доступом.

Клієнтські пристрої включають в себе пристрої з різних груп персоналу, це можуть бути робочі станції лікарів і медсестер, що підключені через локальну мережу, планшети для виїзного персоналу, які працюють через VPN та двохфакторну автентифікацію, робочі ПК адміністрації, бухгалтерії та IT-відділу. Між клієнтськими пристроями і серверами встановлено логічне сегментування мережі (VLAN), що дозволяє розділити трафік між підрозділами та зменшити ризик несанкціонованого доступу.

Засоби автентифікації та авторизації побудовані так, що усі користувачі проходять аутентифікацію через єдиний вхід з логіном і паролем. Додатково для адміністративного та медичного персоналу застосовується двохфакторна автентифікація (2FA) на базі апаратних або програмних токенів [Стаття_2]. Механізм авторизації реалізовано згідно з RBAC-моделлю, описаною раніше. Кожному користувачу призначається роль, яка визначає його рівень доступу до даних у МІС.

Резервне копіювання та NAS-сховища використовуються для забезпечення безперервності доступу до критичних даних та відновлення в разі збою використовуються NAS-сервери резервного копіювання, які автоматично зберігають образи баз даних та важливі документи. Резервні копії зберігаються за розкладом і перевіряються IT-відділом на цілісність.

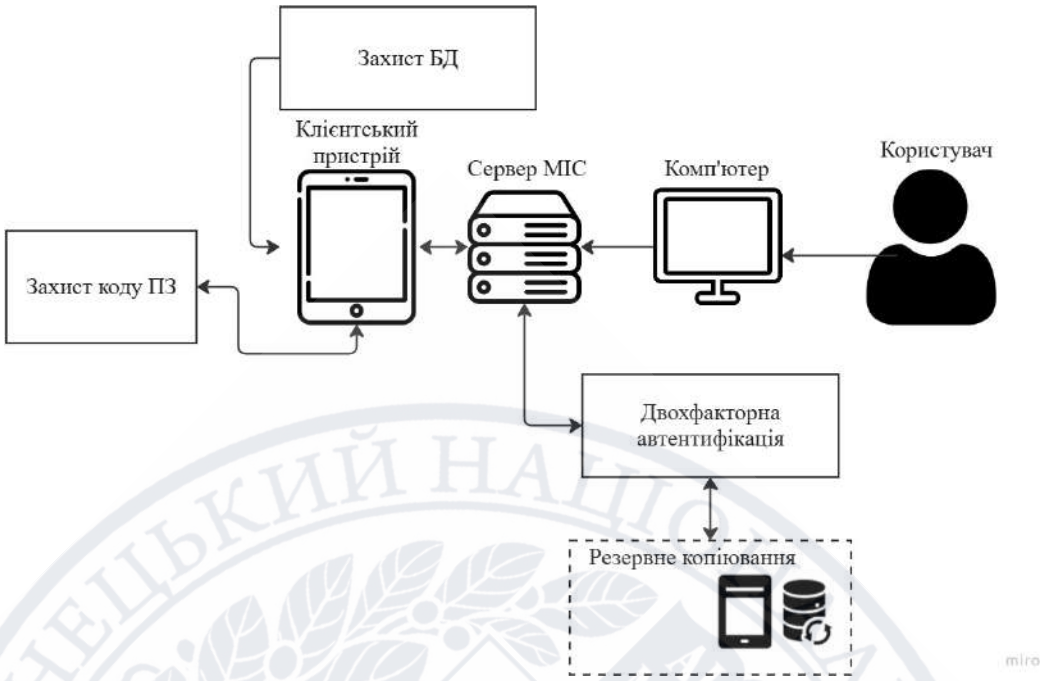


Рисунок 2.1 – Архітектура системи захисту

Таким чином, технічна реалізація багаторівневого доступу в медичному закладі охоплює цілий комплекс рішень — від фізичних серверів і клієнтських пристроїв до логічних механізмів контролю доступу, журналювання та резервного копіювання [6]. Саме така інтегрована архітектура дозволяє гарантувати дотримання політики безпеки в умовах реальної лікарняної інфраструктури.[2]

2.5 Програмне забезпечення доступу та захисту

У межах багаторівневої системи доступу програмне забезпечення виконує ключову роль: саме воно забезпечує ідентифікацію, автентифікацію, авторизацію, моніторинг, логування та шифрування даних. В медичних установах, де опрацьовуються високочутливі дані, від вибору та налаштування ПЗ залежить не лише зручність, а й відповідність системи законодавчим та етичним нормам.

Медичні інформаційні системи є основою функціонування медичного закладу, та виконує роль прикладного середовище для роботи медичного персоналу, саме у МІС реалізується створення та зберігання електронних медичних карток, робота з електронними рецептами, направленнями,

заключеннями, також взаємодія з центральною базою даних ЕСОЗ і підтримка доступу згідно з ролями [7]. Усі МІС, що інтегруються з ЕСОЗ, проходять державну перевірку на відповідність технічним вимогам безпеки [2].

Системи автентифікації та керування обліковими записами служать для ідентифікації персоналу, та використовують уніфіковані облікові записи, збереження в системі МІС, або зовнішньому сервісі, наприклад LDAP або Active Directory. Додатково впроваджується двохфакторна автентифікація для критичних ролей, таких як адміністратор, головний лікар, керівник відділення. Для цього використовуються OTP-додатки, такі як: GoogleAuthenticator або FreeOTP, також використовуються апаратні токени та SMS-коди.

Для моніторингу та логування, фіксації усіх дій користувачів, спроб входу або перегляду критичних даних використовуються спеціалізовані сервіси моніторингу, такі як:

- Zabbix – моніторинг інфраструктури;
- NetFlow – аналіз мережевої активності;
- Syslog – централізоване зберігання подій.

Засоби захисту інформації необхдіні для запобігання несанкціонованому доступу та витоку даних, для цього можна використати антивірусне ПЗ, таке як Avast або ESET, також VPN-клієнти з підтримкою 2FA для віддаленого доступу, системи резервного копіювання, такі як Synology або Veeam, які захищають бази даних від втрати чи пошкодження, а також шифрування даних для передачі по мережі (TSL/SSL).

Багато з перелічених засобів вже інтегровані в типові МІС, однак для додаткової безпеки рекомендовано їх поєднувати з незалежними рішеннями, що відповідають міжнародним стандартам безпеки (наприклад, NIST або ISO/IEC 27001) [8]. Таким чином, програмне забезпечення виступає центральною ланкою у впровадженні багаторівневого доступу, забезпечуючи як технічну реалізацію політик безпеки, так і зручність взаємодії персоналу з інформаційною системою лікарні.

2.5.1 Основні вимоги до реалізації контролю доступу

Алгоритм

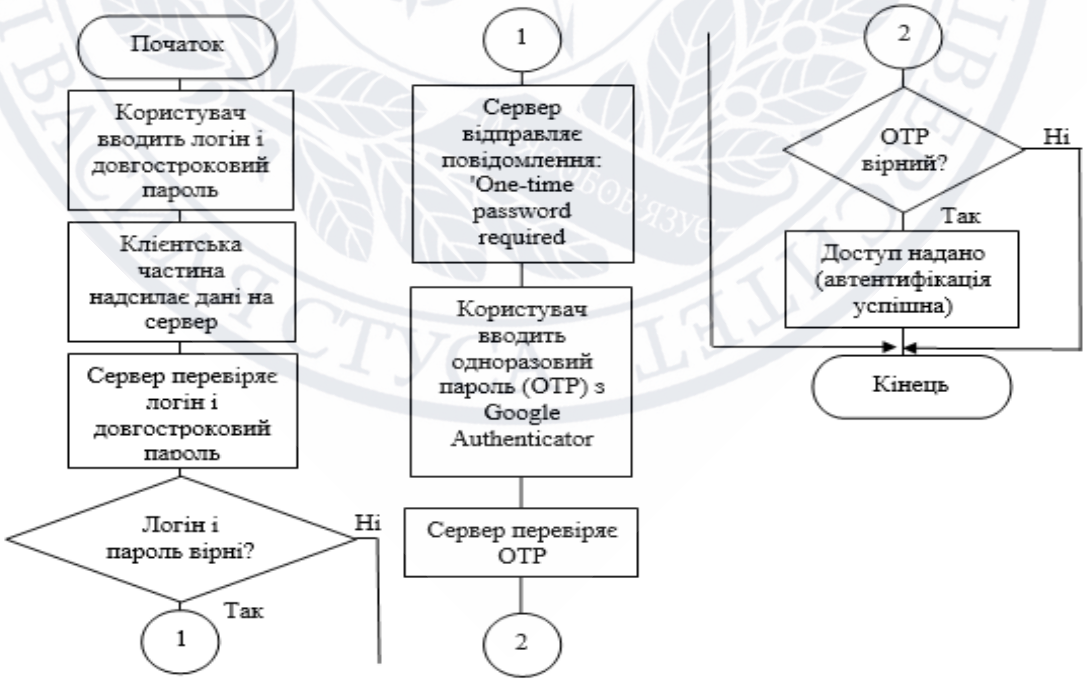
Алгоритм автентифікації за довгостроковим паролем і одноразовим паролем, отриманим за допомогою програми Google Authenticator виглядає наступним чином:

1. Ввод логіну і довгострокового паролю на клієнтській частині.
2. Клієнтська частина відправляє дані, введені суб'єктом, на серверну частину.
3. Серверна частина обробляє запит і перевіряє справжність введеного логіна і довготривалого пароля.
4. Якщо введені дані вірні, то перехід на крок 5.

Інакше, на клієнтську частину відправляється відповідь з маркером помилки.

5. Серверна частина відправляє на клієнтську частину повідомлення: "One-time password required"
6. Ввод одноразового паролю з програми Google

Блок-схема наведеного парою представлена на рс.2.2



На рис. 2.2 представлено блок схему алгоритму

Authenticator на клієнтській частині.

7. Клієнтська частина відправляє одноразовий пароль, логін і довготривалий пароль на серверну частину.

8. Серверна частина обробляє запит і перевіряє справжність введеного логіна, довготривалого пароля і одноразового пароля за допомогою секрету.

9. Якщо отриманий одноразовий пароль не збігається зі згенерованим за допомогою секрету одноразовим паролем, то перехід на крок 10.Інакше, перехід на крок 11.

10. Серверна частина відправляє на клієнтську частину маркер помилки і повідомлення: "Incorrect code"

11. Серверна частина генерує ідентифікатор сесії, зберігає його в базі даних і відправляє на клієнтську частину.

12. Клієнтська частина отримує відповідь від серверної частини і, якщо немає помилки, то встановлює секрет в наступні запити на серверну частину.

Програмна реалізація

Мова і засоби розробки

Як мова розробки програми була обрана платформа мови програмування JavaScript NodeJS, а в якості фреймворка для запуску веб-сервера був обраний модуль Express. Цей вибір обгрунтований наступними факторами:

1. Кросплатформеність, що дозволяє використовувати модуль в продуктах на різних ОС.

2. Наявність безлічі модулів (бібліотек), що дозволяють зменшити кількість вихідного коду і час розробки;

Як середовище розробки був обраний інструмент Visual Studio Code IDE 1.40.2, який є найбільш актуальною версією програми на момент написання роботи. Основною перевагою даного середовища є підтримка безлічі плагінів для зручної роботи, а також вільний доступ.

Ресурси

1. Node JS (LTS)

2. Google Authenticator

Після установки перерахованих вище ресурсів займемося створенням API для додатка.

Порядок розробки

Крок 1: серверний додаток

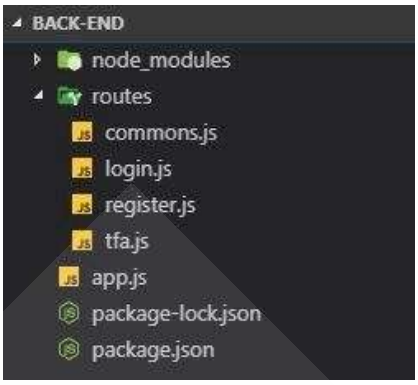
Для створення API-сервісів ми будемо використовувати невеликий фреймворк для Node.js, який називається Express.js. Створимо папку 'back- end' для нашого серверного додатка. Потім перейдемо в неї в терміналі командного рядка і встановимо необхідні залежності.

```
> mkdir back-end
> cd back-end
> npm init -y
> npm install --save express body-parser cors qrcode speakeasy
```

Ми створили папку 'back-end' і ініціалізували проект Node.js, встановивши наступні залежності:

1. express - це невеликий настроюється фреймворк для створення API сервісів.
2. body-parser - для аналізу методів HTTP.
3. cors - пакет використовується для інтеграції клієнтської частини веб-додатки з API сервісами.
4. qrcode - відповідає за генерацію QR-код у вигляді зображень base64.
5. speakeasy - генератор секретних ключів за алгоритмом T-OTP, який використовує Google Authenticator.

Тепер створимо кілька API-сервісів, в яких головним виконуваним файлом буде app.js. Для спрощення вивчення матеріалу ми опустимо код, який відповідає за взаємодію з базою даних.



Структура папок для серверної частини

API-сервіси реалізують функціонал входу в систему, реєстрації та TFA (двохфакторну автентифікацію):

Сервіс входу в систему

Включає в себе базову функціональність для входу за допомогою логіна, пароля і коду автентифікації.

```
import { Component, OnInit } from '@angular/core';
import { LoginServiceService } from 'src/app/services/login-service/login-service.service';

@Component({
const express = require('express');
const speakeasy = require('speakeasy');
const commons = require('./commons');
const router = express.Router();

router.post('/login', (req, res) => {
    console.log(`DEBUG: Received login request`);

    if (commons.userObject.uname && commons.userObject.upass) {
        if (!commons.userObject.tfa || !commons.userObject.tfa.secret) {
            if (req.body.uname == commons.userObject.uname && req.body.upass == commons.userObject.upass) {
                console.log(`DEBUG: Login without TFA is successful`);

                return res.send({
                    "status": 200,
                    "message": "success"
                });
            }
            console.log(`ERROR: Login without TFA is not successful`);

            return res.send({
                "status": 403,
                "message": "Invalid username or password"
            });

        } else {
            if (req.body.uname != commons.userObject.uname || req.body.upass != commons.userObject.upass) {
                console.log(`ERROR: Login with TFA is not successful`);

                return res.send({
                    "status": 403,
                    "message": "Invalid username or password"
                });
            }
            if (!req.headers['x-tfa']) {
                console.log(`WARNING: Login was partial without TFA header`);

                return res.send({
                    "status": 206,
                    "message": "Please enter the Auth Code"
```

```
            });
        }
        let isVerified = speakeasy.totp.verify({
            secret: commons.userObject.tfa.secret,
            encoding: 'base32',
            token: req.headers['x-tfa']
        });

        if (isVerified) {
            console.log(`DEBUG: Login with TFA is verified to be successful`);

            return res.send({
                "status": 200,
                "message": "success"
            });
        } else {
            console.log(`ERROR: Invalid AUTH code`);

            return res.send({
                "status": 206,
                "message": "Invalid Auth Code"
            });
        }
    }
}

return res.send({
    "status": 404,
    "message": "Please register to login"
});
});

module.exports = router;
```

Не будемо використовувати базу даних для зберігання даних користувачів. Тому реалізуємо це на стороні сервера.

```
let userObject = {};

module.exports = {
    userObject
}
```

Сервіс реєстрації

Реєстрація користувача в додатку буде полягати в додаванні логіна і пароля в об'єкт userObject. А також у видаленні існуючої в ньому інформації. Модулі входу і реєстрації створювалися винятково для демонстрації, тому додаток буде підтримувати тільки одного користувача.

```
const commons = require('./commons');
const router = express.Router();

router.post('/register', (req, res) => {
    console.log(`DEBUG: Received request to register user`);

    const result = req.body;

    if ((!result.uname && !result.upass) || (result.uname.trim() == "" || result.upass.trim() == "")) {
        return res.send({
            "status": 400,
            "message": "Username/ password is required"
        });
    }

    commons.userObject.uname = result.uname;
    commons.userObject.upass = result.upass;
    delete commons.userObject.tfa;

    return res.send({
        "status": 200,
        "message": "User is successfully registered"
    });
});

module.exports = router;
```

сервіс TFA

Сервіс призначений для реалізації двохфакторної автентифікації поряд з верифікацією коду T-OTP, згенерованого Google Authenticator. Він буде включати в себе функціональність для отримання налаштувань TFA, а також включення або відключення TFA для userObject.

```
const express = require('express');
const speakeasy = require('speakeasy');
const QRCode = require('qrcode');
const commons = require('./commons');
const router = express.Router();

router.post('/tfa/setup', (req, res) => {
    console.log(`DEBUG: Received TFA setup request`);

    const secret = speakeasy.generateSecret({
        length: 10,
        name: commons.userObject.uname,
        issuer: 'NarenAuth v0.0'
    });
    var url = speakeasy.otpauthURL({
        secret: secret.base32,
        label: commons.userObject.uname,
        issuer: 'NarenAuth v0.0',
        encoding: 'base32'
    });

    QRCode.toDataURL(url, (err, dataURL) => {
        commons.userObject.tfa = {
            secret: '',
            tempSecret: secret.base32,
            dataURL,
            tfaURL: url
        };
        return res.json({
            message: 'TFA Auth needs to be verified',
            tempSecret: secret.base32,
            dataURL,
            tfaURL: secret.otpauth_url
        });
    });
});

router.get('/tfa/setup', (req, res) => {
    console.log(`DEBUG: Received FETCH TFA request`);

    res.json(commons.userObject.tfa ? commons.userObject.tfa : null);
});
```

```
router.delete('/tfa/setup', (req, res) => {
    console.log(`DEBUG: Received DELETE TFA request`);

    delete commons.userObject.tfa;
    res.send({
        "status": 200,
        "message": "success"
    });
});

router.post('/tfa/verify', (req, res) => {
    console.log(`DEBUG: Received TFA Verify request`);

    let isVerified = speakeasy.totp.verify({
        secret: commons.userObject.tfa.tempSecret,
        encoding: 'base32',
        token: req.body.token
    });

    if (isVerified) {
        console.log(`DEBUG: TFA is verified to be enabled`);

        commons.userObject.tfa.secret = commons.userObject.tfa.tempSecret;
        return res.send({
            "status": 200,
            "message": "Two-factor Auth is enabled successfully"
        });
    }

    console.log(`ERROR: TFA is verified to be wrong`);

    return res.send({
        "status": 403,
        "message": "Invalid Auth Code, verification failed. Please verify the system Date and Time"
    });
});

module.exports = router;
```

Згадані вище сервіси включені в один виконуваний файл 'app.js', розташований в кореневій папці. Цей код запустить HTTP-сервер, створений за допомогою express.js на локальному хості з портом 3000.

```
const express = require('express');
const app = express();
const bodyParser = require('body-parser');
const cors = require('cors');
const login = require('./routes/login');
const register = require('./routes/register');
const tfa = require('./routes/tfa');

app.use(bodyParser.json());
app.use(cors());

app.use(login);
app.use(register);
app.use(tfa);

app.listen('3000', () => {
    console.log('The server started running on http://localhost:3000');
```

Ми реалізували серверну частину коду веб-додатки.

```
> ng serve

** Angular Live Development Server is listening on localhost:4200,
```

Наступним кроком буде створення простого додатка, що використовує ці сервіси на Angular 7.

Крок 2: додаток на базі Angular 7

Спочатку потрібно встановити Angular. Після цього ми створимо додаток з назвою 'front-end' і встановимо залежність від 'bootstrap' (посилання на bootstrap.min.css в styles.css), перейшовши в папку front-end.

```
> npm install -g @angular/cli
> ng new front-end
> cd front-end
> npm install --save bootstrap
> ng serve
```

Після цього створимо кілька компонентів і сервісів, які потрібні додатком. Для цілей демонстрації ми створимо LoginService і два guards - 'Auth Guard' і 'Login Guard.'

```
> ng g s services/login-service/login-service --spec=false
> ng g g guards/AuthGuard
> ng g g guards/Login
```

Guards, які ми створюємо, відносяться до типу CanActivate. Сервіс входу в систему буде включати в себе HTTP-запити до сервісів, створеним на стороні сервера.

AuthGuard обмежить навігацію користувача тільки домашньою сторінкою, якщо той не увійшов в систему.

```
import { Injectable } from '@angular/core';
import { CanActivate, ActivatedRouteSnapshot, RouterStateSnapshot, Router } from '@angular/router';
import { LoginServiceService } from 'src/app/services/login-service/login-service.service';

@Injectable({
  providedIn: 'root'
})
export class AuthGuardGuard implements CanActivate {

  constructor(private _loginService: LoginServiceService, private _router: Router) {
  }
  canActivate(next: ActivatedRouteSnapshot, state: RouterStateSnapshot) {
    if (this._loginService.getAuthStatus()) {
      return true;
    }

    this._router.navigate(['/login'])

    return false;
  }
}
```

LoginGuard не дозволить користувачеві заходити на сторінку авторизації, якщо користувач вже увійшов в систему.

```
import { Injectable } from '@angular/core';
import { CanActivate, ActivatedRouteSnapshot, RouterStateSnapshot, Router } from '@angular/router';
import { LoginServiceService } from 'src/app/services/login-service/login-service.service';

@Injectable({
  providedIn: 'root'
})
export class LoginGuard implements CanActivate {
  constructor(private _loginService: LoginServiceService, private _router: Router) {
  }
  canActivate(
    next: ActivatedRouteSnapshot,
    state: RouterStateSnapshot) {
    if (!this._loginService.getAuthStatus()) {
      return true;
    }

    this._router.navigate(['/home'])
    return false;
  }

}
```

Основи для нашого застосування, (служби і guards). Розробка компонентів.

```
> ng g c components/header --spec=false
> ng g c components/home --spec=false
> ng g c components/login --spec=false
> ng g c components/register --spec=false
```

Після створення необхідних компонентів додатка ми налаштуємо маршрутизацію для додатка, зв'язавши відповідні guards для активації маршрутів. Також, видалимо код, доданий за замовчуванням в app.component.html, і вставимо компонент загального заголовка і висновок маршрутизатора.

```
<app-header></app-header>
<router-outlet></router-outlet>
```

Компонент заголовку

Це загальний компонент для інших компонентів, який включає в себе панель навігації програми. Видимість посилань в заголовку контролюється методом getAuthStatus () сервісу LoginService. У фоновому режимі запросимо файл * .ts для компонента заголовка.

Компонент входу в систему

Призначений для отримання логіну, паролю та коду AuthCode (якщо включений TFA) від користувача і його перевірки на стороні сервера. Якщо

дані вірні, то користувач буде переміщений в HomeComponent. Ми також будемо перевіряти статус, отриманий від серверної частини коду, для відображення повідомлень користувачеві.

```
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router'
import { LoginServiceService } from 'src/app/services/login-service/login-service.service';

@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.css']
})
export class LoginComponent implements OnInit {

  tfaFlag: boolean = false
  userObject = {
    uname: "",
    upass: ""
  }
  errorMessage: string = null
  constructor(private _loginService: LoginServiceService, private _router: Router) {
  }

  ngOnInit() {
  }

  loginUser() {
    this._loginService.loginAuth(this.userObject).subscribe((data) => {
      this.errorMessage = null;
      if (data.body['status'] === 200) {
        this._loginService.updateAuthStatus(true);
        this._router.navigateByUrl('/home');
      }
      if (data.body['status'] === 206) {
        this.tfaFlag = true;
      }
      if (data.body['status'] === 403) {
        this.errorMessage = data.body['message'];
      }
      if (data.body['status'] === 404) {
        this.errorMessage = data.body['message'];
      }
    })
  }
}
```

Компонент реєстрації

Ми зареєструємо одного користувача в усьому додатку.Якщо загубиться логін і пароль для додатка, або секретний ключ TFA, просто потр ввести нове ім'я користувача та пароль на сторінці.

Сторінка входу і реєстрації

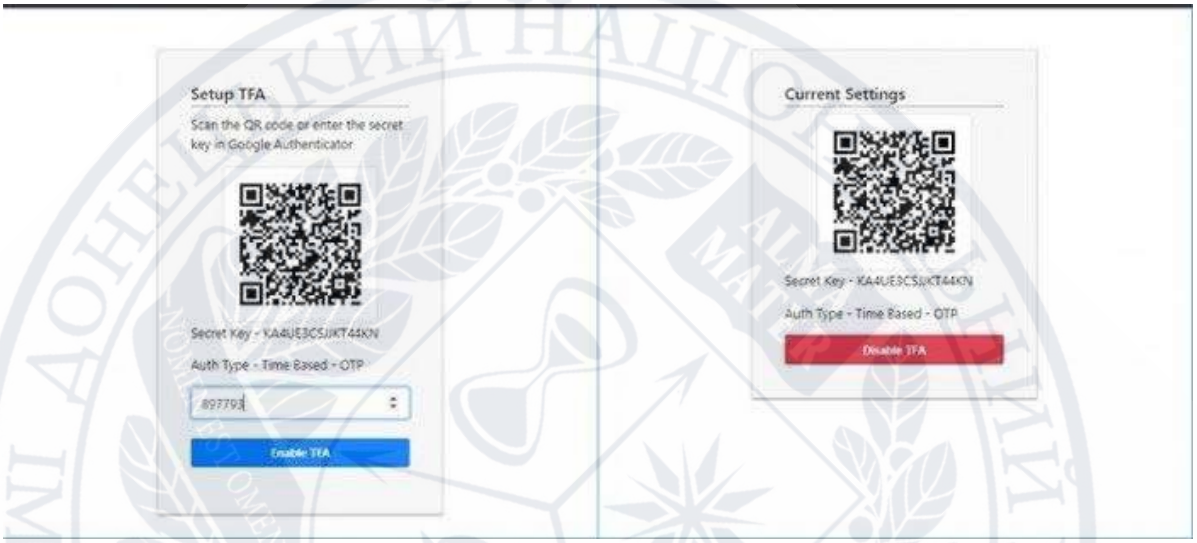
Як тільки користувач зареєструвався і увійшов в систему з логіном і паролем, йому буде надана можливість включення і відключення двохфакторної автентифікації в HomeComponent.

Компонент Home

Дозволить користувачеві налаштовувати і перевіряти TFA. Як тільки користувач потрапить на цю сторінку, він зможе відсканувати QR-код в

додатку Google Authenticator. Після сканування T-OTP (елемент TFA), пов'язаний з userObject, буде включений в додаток Google Authenticator. AuthCode буде відображатися в додатку на тимчасовій основі. Такий самий пароль необхідний для перевірки і включення TFA для userObject.

Якщо користувач увімкнув TFA, то відобразяться поточні настройки з QR-кодом і секретним ключем. А також опція відключення TFA, пов'язаного з userObject.



Установка і поточні налаштування TFA

Функції тимчасової роботи (middleware) - це функції, які мають доступ до об'єкта запиту (request), об'єкту відповіді (response) і до наступної функції тимчасової роботи в циклі "запит-відповідь" додатка. Наступна функція тимчасової роботи, як правило, позначається змінної next.

Функції тимчасової роботи можуть виконувати такі завдання:

1. Виконання будь-яких команд.
2. Внесення змін до об'єкти запитів і відповідей.
3. Завершення циклу "запит-відповідь".
4. Виклик наступної функції тимчасової роботи з стека функцій.

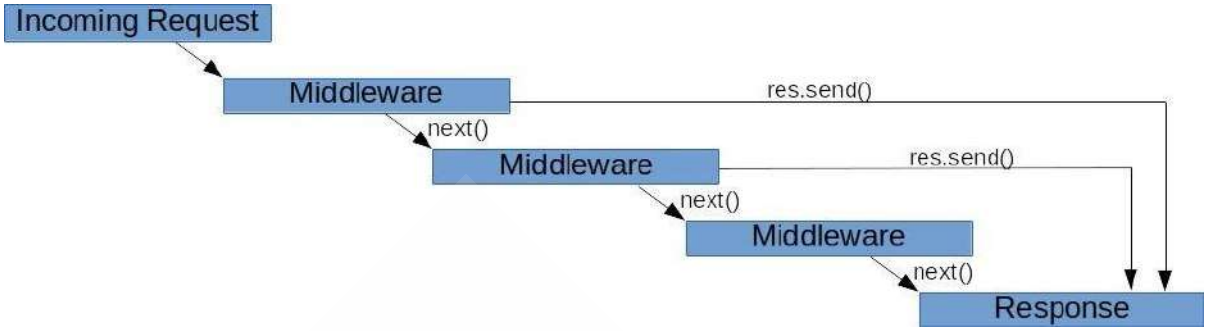


Рис. 2.3 Стек функцій тимчасової роботи сервера

2.6 Оцінка ризиків при неправильно реалізованому доступі

У процесі впровадження багаторівневої системи доступу в медичній інформаційній системі важливо враховувати ризики, що виникають у разі помилок у конфігурації доступу, відсутності належного моніторингу або нехтування організаційними заходами безпеки. Наслідками можуть бути як несанкціонований доступ до конфіденційних даних, так і їх втрата чи модифікація.

Найбільш ймовірні ризики вказані у таблиці 3.

Таблиця 3 – Можливі ризики у системі з медичними даними

Ризик	Причина	Наслідок	Джерело	Порушений параметр
Ризик несанкціонованого перегляду медичних записів	Надлишкові права	Несанкціоноване ознайомлення з медичними записами пацієнтів інших відділень або лікарів	Внутрішній співробітник	Конфіденційність
Ризик втрати контролю над обліковим записом	Відсутність F2A	Викрадення акаунта	Зовнішній порушник	Конфіденційність, цілісність
Ризик модифікації даних	Спільні облікові записи	Непідконтрольні зміни в даних	Внутрішні співробітники	Цілісність
Неконтрольований витік	Технічні канали витоку інформації	Витік інформації через радіоканали, акустичні канали, віброакустичні канали, оптичні канали, мовні.	Зовнішній порушник	Конфіденційність, цілісність

Ризик втрати даних	Відсутність захисту резервних копій	Втрата даних без можливості повернення	Стихійні лиха, зовнішні та внутрішні порушники, пожежі, технічні збої	Доступність, цілісність

Наступним кроком можна зробити оцінку ризиків витоку даних у медичному закладі. Методом обраним для оцінки буде експертна оцінка, у якій буде три рівня ризику: високий, середній та низький, кожен з яких буде означати величину фінансових збитків, де низький – збитки до 20 тис. грн.,

середній – збитки до 70 тис. грн., високий – збитки від 70 тис. грн.

Таблиця 4 – Експертна оцінка ризиків

Об’єкт загрози	Джерело загрози	Канал витоку (ТКВІ)	Порушена складова безпеки	Рівень ризику
ЕМК	Стороння особа	Радіоканал	Конфіденційність	Високий
Паролі користувачів	Спостереження	Акустичний канал	Конфіденційність	Середній
Дані обстеження	Медичне обладнання	Віброакустичний	Конфіденційність	Середній
Екрани з даними	Співробітники	Оптичний канал	Конфіденційність	Низький
Розмови лікарів	Внутрішній персонал	Мовний канал	Конфіденційність	Високий
Резервні копії	ІТ-персонал	Несанкціонований доступ	Доступність, цілісність	Середній
Облікові записи	Внутрішня помилка	Відсутність 2FA	Конфіденційність, підзвітність	Високий

З огляду на чутливість медичних даних, ризики, пов’язані з неналежною реалізацією багаторівневого доступу, мають системний характер. Вони охоплюють як технічні, так і організаційні загрози, що вимагає комплексного підходу: правильного розмежування доступу, технічного захисту каналів передачі та носіїв, а також регулярного аналізу й усунення вразливостей.

3. ПОЛІТИКА БЕЗПЕКИ

3.1 Загальні принципи політики безпеки в медичній інформаційній системі

Політика безпеки доступу до інформаційних ресурсів медичного закладу встановлює загальні принципи, вимоги та обмеження, що регулюють порядок взаємодії користувачів із електронною системою зберігання медичних даних. Її мета — забезпечити конфіденційність, цілісність, доступність та підзвітність даних пацієнтів, персоналу та всієї медичної інформації, яка обробляється в межах автоматизованої системи закладу охорони здоров'я. Дана політика поширюється на всіх користувачів які мають доступ до ЕМК, персональних даних, результатів діагностичних та лабораторних досліджень, даних про лікування, призначення, направлення, фінансових і кадрових записів.

В основі політики лежить рольова модель управління доступом (RBAC), яка визначає права доступу не індивідуально, а через відповідність посадовим обов'язкам. Такий підхід дає змогу централізовано та ефективно розмежовувати повноваження медичного, адміністративного, технічного та допоміжного персоналу.

До основних принципі реалізації політики безпеки належать:

- Принцип мінімального привілею — кожен користувач має доступ лише до тих ресурсів, які потрібні йому для виконання службових обов'язків.
- Розмежування обов'язків — критично важливі операції (наприклад, створення, затвердження, аудит) не можуть виконуватись однією особою.
- Ідентифікація та автентифікація користувачів — кожен доступ до системи має бути індивідуальним, підтвердженим автентифікацією (із застосуванням 2FA для критичних ролей).
- Журналювання дій та аудит доступів — усі дії користувачів фіксуються в системі моніторингу для подальшого аналізу безпеки.
- Захист каналів передавання та зберігання даних — використання шифрування (TLS), резервного копіювання та контролю доступу до

копій.

- Оцінка ризиків і реагування на інциденти — регулярний аналіз загроз і впровадження плану дій на випадок порушення безпеки.

Зазначені принципи узгоджуються з положеннями законодавства України у сфері захисту персональних даних, а також з міжнародними стандартами ISO/IEC 27001, GDPR та регламентами ЕСОЗ щодо безпеки медичної інформації [2].

3.2 Структура системи багаторівневого доступу

Система багаторівневого доступу до медичних даних у межах автоматизованої інформаційної системи лікарні є сукупністю програмних, апаратних та організаційних засобів, що забезпечують регламентований доступ до ресурсів залежно від ролі користувача. Така система створюється з метою захисту медичної інформації від несанкціонованого доступу, втрати, модифікації або розголошення.

Політика доступу реалізується на основі рольової моделі RBAC, в якій кожна роль пов’язана з певним набором дозволених дій над конкретними типами даних. Архітектурно система побудована таким чином, щоб забезпечити:

- централізоване керування обліковими записами;
- фіксацію усіх подій доступу;
- автоматичне призначення прав залежно від функціональної ролі;
- взаємодію з іншими підсистемами (ЕСОЗ, МІС, кадрові системи).

Структура системи охоплює користувацький рівень, рівень управління доступом, службовий рівень, інтеграційний рівень.

Користувацький рівень включає персонал, який взаємодіє з МІС через робочі станції або мобільні пристрої. Їх доступ залежить від ролі: лікар, медсестра, адміністратор, бухгалтер тощо.

Рівень управління доступом – це серверна частина, де відбувається ідентифікація, автентифікація та авторизація користувачів. Тут реалізовано

зв'язок із обліковими записами, групами, правами.

Службовий рівень – системні адміністратори та фахівці з кібербезпеки, які керують інфраструктурою, оновленнями, журналами подій та резервними копіями.

Інтеграційний рівень – програмні шлюзи, через які відбувається обмін даними з ЦБД ЕСОЗ та іншими медичними системами, з обов’язковою фільтрацією та шифруванням переданих даних.

Система доступу є розширюваною, тобто дозволяє додавати нові ролі або обмеження без переробки всієї архітектури, що важливо для лікарень з динамічною структурою персоналу або підключенням до нових модулів електронного здоров’я [9].

3.2.1 Вимоги до системи багаторівневого доступу

Система багаторівневого доступу до медичних даних у складі інформаційної системи лікарні повинна відповідати низці вимог, які забезпечують її надійність, масштабованість, безпечність та відповідність законодавству і галузевим стандартам. Ці вимоги стосуються як технічної, так і організаційної складової системи.

Загальні вимоги до системи багаторівневого доступу:

- Гнучкість – можливість створення, редагування та скасування ролей і прав доступу без необхідності перезапуску чи зміни архітектури всієї системи.
- Масштабованість – підтримка зростання кількості користувачів і типів інформації без зниження продуктивності.
- Сумісність – здатність взаємодіяти з іншими інформаційними системами охорони здоров’я (ЕСОЗ, eHealth, лабораторні модулі) за допомогою стандартів HL7, FHIR тощо. [5].
- Простота адміністрування – централізоване керування обліковими записами, логами доступу та політиками безпеки.

Функціональні вимоги до системи багаторівневого доступу:

- Забезпечення авторизованого доступу на основі рольової моделі (RBAC);
- Індивідуальна автентифікація кожного користувача (в т.ч. за допомогою 2FA для критичних ролей);
- Підтримка різних типів дій (читання, запис, створення) з можливістю обмеження на рівні типів даних;
- Ведення журналу дій користувачів із подальшим аналізом;
- Автоматичне обмеження сесій за часом неактивності;
- Підтримка швидкої блокування доступу до облікового запису в разі інциденту.

Вимоги до безпеки:

- Використання захищених протоколів зв'язку (TLS/SSL) при обміні даними між клієнтами та серверами;
- Резервне копіювання баз даних із перевіркою цілісності та контролем доступу;
- Фізичний захист серверів та ключових точок доступу;
- Виявлення та попередження спроб несанкціонованого доступу (інтеграція з системами SIEM/IDS).

Відповідність нормативним документам:

- Закон України «Про захист персональних даних»;
- ISO/IEC 27001 — Система управління інформаційною безпекою;
- GDPR — Загальний регламент ЄС про захист даних;
- Стандарти та інструкції МОЗ України щодо роботи з ЕСОЗ.

Таким чином, система доступу повинна бути не лише технічно захищеною, але й підконтрольною, прозорою та регламентованою, з можливістю оперативної адаптації до змін у структурі закладу або вимогах регуляторів.

3.2.2 Мета та завдання політики доступу в медичному закладі

Політика доступу до інформаційної системи медичного закладу є ключовим елементом внутрішньої системи інформаційної безпеки. Її впровадження забезпечує регламентований і контрольований доступ до даних, запобігає

зловживанням правами та сприяє дотриманню законодавчих вимог щодо захисту персональних і медичних відомостей.

Метою політики доступу є забезпечити конфіденційність, цілісність та доступність електронної інформації, що циркулює в межах медичної інформаційної системи, формалізувати правила ідентифікації, автентифікації та авторизації користувачів, унеможливити несанкціонований доступ до пацієнтських, фінансових та службових даних, гарантувати підзвітність дій користувачів, що взаємодіють з критично важливими ресурсами.

Основним завданням політики є класифікація інформаційних ресурсів за ступенем критичності та обмеженням доступу, визначення ролей користувачів, відповідно до їх функціональних обов'язків, регламентація типів дозволених дій, опис механізмів автентифікації та засобів багатофакторного підтвердження, встановлення термінів та умов доступу, обмеження часу сесій, забезпечення моніторингу, журналювання та аудиту дій користувачів, регламент дій у разі виявлення порушень, або підозрілих активностей та забезпечення навчання персоналу щодо безпечної роботи з МІС.

Політика доступу створюється на основі принципів інформаційної безпеки, визначених міжнародними стандартами ISO/IEC 27001, а також з урахуванням вимог GDPR та законодавства України у сфері охорони здоров'я та захисту персональних даних [9].

3.2.3 Архітектура інформаційної системи лікарні

Інформаційна система лікарні є сукупністю взаємопов'язаних апаратних, програмних і комунікаційних компонентів, що забезпечують автоматизацію медичних, адміністративних і управлінських процесів у закладі охорони здоров'я. Вона також виконує роль інфраструктурної основи для реалізації політики багаторівневого доступу до даних.

Структура системи включає в себе такі компоненти:

- Користувацькі пристрої – персональні комп'ютери, ноутбуки, планшети, які використовуються лікарями, медсестрами, адміністраторами, персоналом ІТ та бухгалтерією. Доступ до системи здійснюється з

урахуванням авторизації через логін/пароль, а для критичних ролей — з підтримкою 2FA.

- Медична інформаційна система (МІС) – ядро програмного забезпечення, через яке здійснюється введення, перегляд, обробка та збереження медичних і службових даних. МІС інтегрується з Центральною базою даних (ЦБД) ЕСОЗ через стандартизовані API (FHIR/HL7) [7].
- Серверне середовище – сервери додатків, баз даних та резервного копіювання, на яких розміщуються ЕМК, дані про обстеження і призначення, облікові записи користувачів, логи дій, журнали подій.
- Інтеграційні шлюзи – канали обміну між МІС та іншими зовнішніми або внутрішніми системами: лабораторіями, аптеками, страховими компаніями, eHealth-сервісами.
- Системи контролю доступу – програмні модулі або окремі сервіси, що реалізують механізми RBAC, автентифікації, ведення журналів, обмеження сесій та аудит.
- Засоби захисту інформації – антивірусні рішення, VPN, фаєрволи, системи виявлення вторгнень (IDS), резервне копіювання, шифрування передавання та зберігання даних (TLS, AES тощо).
- Системи моніторингу та адміністрування – платформи для централізованого управління користувачами, перегляду журналів подій, сповіщень про аномалії або спроби порушення політики доступу.

Інформаційна система функціонує у розділеному мережевому середовищі, у якому є внутрішній сегмент, для медичного персоналу та адміністрації, демілітаризована зона для підключення зовнішніх систем, лабораторій тощо, та захищені канали VPN, для доступу до МІС із зовнішніх точок.

Архітектура системи будується з урахуванням принципів ізоляції критичних компонентів, захисту каналів взаємодії, гнучкого керування правами доступу та безперервності надання послуг [6].

3.2.4 Вимоги до побудови захищеного середовища доступу

Захищене середовище доступу до інформаційних ресурсів медичного закладу – це умова надійної реалізації політики багаторівневого доступу. Його побудова передбачає дотримання комплексу вимог, які охоплюють фізичний, програмний, мережевий і адміністративний рівні захисту. У фізичному рівні сервери та мережеве обладнання повинні бути розміщені в охоронюваному

приміщенні з обмеженим фізичним доступом. Всі пристрої користувачів повинні бути марковані та закріплені за конкретними працівниками. Має бути запроваджено облік та контроль доступу до серверної кімнати з електронними ключами/картками.

На програмному рівні необхідне встановлення сертифікованого антивірусного ПЗ та регулярне оновлення сигнатур. Наявність вбудованих механізмів контролю доступу в медичній інформаційній системі (МІС), з можливістю налаштування ролей та дозволів. Підтримка журналювання всіх подій входу, редагування, спроб порушення політики. Відновлення доступу до системи в разі помилок або втрати паролів здійснюється лише адміністраторами з фіксацією в журналі подій.

На мережевому рівні обов’язковим є сегментування мережі на окремі віртуальні сегменти VLAN відповідно до функціональних ролей (наприклад, медичний, адміністративний, технічний). Використання міжмережевих екранів (фаєрволів) між зовнішнім трафіком і системою. Захищене з'єднання через VPN для доступу з віддалених точок, обов’язково з 2FA. Забезпечення шифрування передавання даних (TLS) для всіх комунікацій між клієнтами та серверами.

У адміністративному рівні забезпечити призначення відповідальних осіб за контроль доступу. Проведення регулярного аудиту прав доступу та оновлення ролей при зміні посад. Документування усіх процедур надання, зміни, блокування та відкликання доступу. Проведення інструктажів з інформаційної безпеки для співробітників.

Таким чином, побудова захищеного середовища — це системна робота, яка охоплює всі рівні взаємодії користувачів з медичною інформаційною системою та дозволяє забезпечити відповідність міжнародним стандартам, вимогам законодавства та вимогам електронної системи охорони здоров’я [6].

3.2.5 Функціональні можливості системи керування доступом

Система керування доступом у медичній інформаційній системі повинна забезпечувати ідентифікацію та автентифікацію користувачів, з підтримкою

2FA для критичних ролей, авторизацію на основі ролей відповідно до службових функцій, гнучке управління ролями та правами доступу через централізовану панель адміністратора, логування дій користувачів, зберігання історії входів, переглядів та змін у даних, моніторинг активності та аудит, з можливістю формування звітів про доступи, шифрування передавання даних та захист сеансів користувачів. Наявність цих функцій дозволяє реалізувати політику безпеки доступу відповідно до стандартів ISO/IEC 27001 та вимог МОЗ щодо ЕСОЗ [6].

3.2.6 Основні вимоги до реалізації контролю доступу

Контроль доступу до медичних даних має відповідати вимогам. Унікальність облікових записів — відсутність спільних логінів, призначення ролей згідно з посадою працівника, двофакторна автентифікація для користувачів з підвищеним доступом, журналювання дій у системі, обмеження доступу за часом, IP або місцем підключення, захист даних під час зберігання і передавання. Ці вимоги забезпечують виконання принципу мінімального доступу та відповідність законодавчим і технічним нормам [9].

3.2.7 Адміністрування та контроль за дотриманням політики доступу

Адміністрування системи доступу в медичному закладі здійснюється IT-відділом спільно з відповідальними особами за інформаційну безпеку. Основні функції:

- Створення, редагування та блокування облікових записів відповідно до кадрових змін;
- Призначення ролей та контроль за відповідністю доступу посадовим обов’язкам;
- Аналіз логів та подій, виявлення аномальної активності;
- Періодичний аудит політики доступу, перегляд ролей і прав користувачів;
- Звітування керівництву про виявлені порушення або інциденти;
- Навчання персоналу щодо правил роботи з інформаційною системою.

Адміністрування має бути прозорим, з фіксацією всіх змін у журналі подій та регулярною перевіркою виконання політики [6].

ВИСНОВКИ

У межах бакалаврської роботи проведено дослідження захисту інформації в медичних інформаційних системах, з фокусом на організацію багаторівневого доступу. Визначено ключові типи даних, що потребують захисту, сформовано ролі користувачів та класифікацію даних за критичністю.

Розроблена система базується на моделі RBAC з матрицею доступу, що забезпечує чітке розмежування повноважень і мінімізацію надлишкових прав.

Запропонована архітектура включає технічні засоби захисту: 2FA, VPN, сегментацію мережі, антивірусне ПЗ та засоби моніторингу. Система інтегрується з ЕСОЗ і відповідає стандартам HL7, FHIR та ISO/IEC 27001.

Політика безпеки передбачає принцип мінімального привілею, підзвітність, регулярний аудит та навчання персоналу. Проведено оцінку ризиків, зокрема щодо технічних каналів витоку.

Результат — ефективна модель, яка підвищує безпеку медичних даних і відповідає чинним нормативам.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Електронна система охорони здоров'я : навч. матеріал / Вінницький нац. техн. ун-т. 2025. 67 с. (дата звернення: 18.03.2025).

2. Що таке медичні інформаційні системи та які послуги вони надають. 2024. 4 с. (дата звернення: 22.03.2025).

3. What is GDPR? The EU's new data protection law. 2024. URL: https://gdpr.eu/what-is-gdpr/. (дата звернення: 22.03.2025).

4. What is ISO 27001? An easy-to-understand explanation. 27001Academy. 2024. URL: https://advisera.com/27001academy/what-is-iso-27001/. (дата звернення: 22.03.2025).

5. HL7 International. FHIR Overview. 2024. URL: https://www.hl7.org/fhir/overview.html. (дата звернення: 22.03.2025).

6. Куперштейн Л.М., Войтович О.П., Остапенко-Боженова А.В., Прокопчук С.А. Багаторівневий підхід до захисту від несанкціонованого використання додатків в операційній системі Android // PI. 2018. №2. С. 45–50. (дата звернення: 22.03.2025).

7. Електронна система охорони здоров'я України. 2025. URL: https://ehealth.gov.ua/. (дата звернення: 22.03.2025)

8. Pascoe C., Quinn S., Scarfone K. The NIST Cybersecurity Framework (CSF) 2.0. Gaithersburg, MD: National Institute of Standards and Technology, 2024. 41 p. (NIST Cybersecurity White Paper, CSWP 29). URL: https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.29.pdf (дата звернення: 22.03.2025)

9. Боженко А.Ю., Настенко Є.А., Корнієнко Г.А., Дикан І.М., Павлов В.А., Тарасюк Б.А., Городецька О.К., Авер'янова О.А. Розробка системи накопичення та зберігання медичних зображень за допомогою Salesforce // Biomedical Engineering and Technology. 2024. Т. 13, № 1. С. 1–17. ISSN (Online) 2707-8434. (дата звернення: 22.03.2025)

ДОДАТОК А

Медичні інформаційні системи відносяться до 3-го класу оскільки вони редставляють розподілений багатомашинний багатокористувачевий комплекс, який обробляє інформацію різних ступенів обмеження доступу.

Стандартні функціональні профілі захищеності в КС, що входять до складу АС класу 3, з підвищеними вимогами до забезпечення конфіденційності, цілісності і доступності оброблюваної інформації:

3.КЦД.1 = { КД-2, КО-1, КВ-1, ЦД-1, ЦО-1, ЦВ-1, ДР-1, ДВ-1, НР-2, НИ-2, НК-1, НО-2, НЦ-2, НТ-2, НВ-1 }

3.КЦД.2 = { КД-2, КА-2, КО-1, КВ-2, ЦД-1, ЦА-2, ЦО-1, ЦВ-2, ДР-1, ДВ-1, НР-2, НИ-2, НК-1, НО-2, НЦ-2, НТ-2, НВ-1 }

3.КЦД.3 = { КД-2, КА-2, КО-1, КК-1, КВ-3, ЦД-1, ЦА-3, ЦО-2, ЦВ-2, ДР-2, ДС-1, ДЗ-1, ДВ-2, НР-3, НИ-2, НК-1, НО-2, НЦ-3, НТ-2, НВ-2 }

3.КЦД.4 = { КД-3, КА-3, КО-1, КК-1, КВ-3, ЦД-1, ЦА-3, ЦО-2, ЦВ-2, ДР-3, ДС-2, ДЗ-2, ДВ-2, НР-4, НИ-2, НК-1, НО-3, НЦ-3, НТ-2, НВ-2, НА-1, НП-1 }

3.КЦД.5 = { КД-4, КА-4, КО-1, КК-2, КВ-4, ЦД-4, ЦА-4, ЦО-2, ЦВ-3, ДР-3, ДС-3, ДЗ-3, ДВ-3, НР-5, НИ-2, НК-2, НО-3, НЦ-3, НТ-2, НА-1, НП-1, НВ-2, НА-1, НП-1 }