МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ

КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ АВТОМАТИЗОВАНИХ СИСТЕМ

Кваліфікаційна робота

другий (магістерський)
(рівень вищої освіти)

на тему **Порівняння систем шифрування на прикладі медичної інформаційної системи**

Виконала: студентка 2 курсу, групи 8.1219-пзс
Спеціальності 121 Інженерія програмного забезпечення
(код і назва спеціальності)
освітньої програми Інженерія програмного забезпечення
(код і назва освітньої програми)
Н. С. Касілова
(ініціали та прізвище)

Керівник доцент, к. ф. - м. н. І. А. Скрипник
(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)

Рецензент директор ТОВ «Дісітел» П.О.Лютий
(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)

Запоріжжя
2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІНЖЕНЕРНИЙ НАВЧАЛЬНО – НАУКОВИЙ ІНСТИТУТ

Кафедра _____ програмного забезпечення автоматизованих систем
Рівень вищої освіти _____другий (магістерський)_____
Спеціальність ___121_Інженерія програмного забезпечення_____
(код та назва)
Освітня програма ___Інженерія програмного забезпечення
(код та назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри _____В.Г. Вербицький
“_01_” __вересня_____2020_року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

_____
(прізвище, ім’я, по батькові)

1.Тема роботи ___Порівняння систем шифрування на прикладі медичної інформаційної системи_____

керівник роботи __Скрипник Ірина Анатолівна, канд.фіз.-мат.наук, доцент
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)
затверджені наказом ЗНУ від _“25”_травня_2020_року № 600-с_____

2. Строк подання студентом кваліфікаційної роботи___30.11.2020____
3. Вихідні дані магістерської роботи
1. комплект нормативних документів ;
2. технічне завдання до роботи.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
3. огляд та збір літератури стосовно теми кваліфікаційної роботи;
4. огляд та аналіз існуючих рішень та аналогів;
5. дослідження проблеми шифрування даних медичних інформаційних систем та розробка методів її вирішення;
6. створення програмного продукту та його опис;
7. перелік вимог для роботи програми;
8. дослідження поставленої проблеми та розробка висновків та пропозицій.

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)
___слайдів презентації

6. Консультанти розділів магістерської роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата Завдання прийняв

7. Дата видачі завдання 01.09.2020

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів магістерської роботи	Примітка
1	Аналіз предметної області	02.09-10.09.20	виконано
2	Формулювання основної задачі дипломної роботи та узгодження її з науковим керівником	11.09-12.09.20	виконано
3	Аналіз існуючих методів рішення	13.09-17.09.20	виконано
4	Дослідження принципів роботи ендогастроетерологічного відділення Обласної дитячої лікарні	18.09-24.09.20	виконано
5	Узгодження подальших дій з науковим керівником	25.09-26.09.20	виконано
6	Аналіз теоретичних відомостей	27.09-15.10.20	виконано
7	Проектування інтерфейсу медичної інформаційної системи ендогастроетерологічного відділення	15.10-23.10.20	виконано
8	Узгодження інтерфейсу з науковим керівником	23.10-24.10.20	виконано
9	Реалізація функціоналу інформаційної системи ендогастроетерологічного відділення	25.10-14.11.20	виконано
10	Реалізація ширування даних інформаційної системи ендогастроетерологічного відділення	15.11-16.11.20	виконано
11	Проведення аналізу можливостей розроблених систем шифрування. Оформлення звіту	16.11-24.11.20	виконано

Студентка _____ Н. С. Касілова _____
(підпис) (ініціали та прізвище)

Керівник роботи _____ І. А. Скрипник ___
(підпис) (ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____ І.А. Скрипник _____
(підпис) (ініціали та прізвище)

АНОТАЦІЯ

Касілова Н.С. Порівняння систем шифрування на прикладі медичної інформаційної системи.

Кваліфікаційна робота магістра складається із введення, 4 розділів і висновків, списку використаних джерел з 42 найменувань. Робота містить 100 сторінок тексту, 35 рисунки, 11 таблиць, 7 лістингів коду.

Кваліфікаційна робота для здобуття ступеня вищої освіти магістра за спеціальністю 121 – Інженерія програмного забезпечення, науковий керівник І.А. Скрипник. Інженерний навчально-науковий інститут Запорізького національного університету, 2020.

Мета дослідження полягає у вивченні методів захисту інформації в інформаційних системах, особливо медичних, їх особливостей, а також у створенні власної медичної інформаційної системи, яка буде працювати у режимі онлайн. В системі мають бути такі функції, як: зберігання даних, видалення даних та додавання даних. Головною відмінністю розроблюваної системи від більшості вже існую-чих є те, що система буде зручною у користуванні для різних категорій меди-чного персоналу та більш інформаційно захищеною.

Досліджено існуючі проблеми з області захисту медичних інформаційних систем. Детально проведено порівняння існуючих рішень і задач, які вони вирішують. Класифіковано знайдені види помилок і обрано найменш вивчені та найбільш важливі області для розробки власного рішення проблеми. На основі вивчених даних, створено свою медичну інформаційну систему ендогастроетерологічного відділення Обласної дитячої лікарні. Доведено подальші можливості розвитку шифрування даних в медичних інформаційних системах.

Ключові слова: МЕДИЧНА ІНФОРМАЦІЙНА СИСТЕМА, ЕНДОГАСТРОЕТЕРОЛОГІЧНЕ ВІДДІЛЕННЯ ЛІКАРНІ, СИСТЕМА КЕРУВАННЯ БАЗАМИ ДАНИХ, MySQL, PHP, ШИФРУВАННЯ ДАНИХ, SSL ПРОТОКОЛ.

SUMMARY

Kasilova N.S. Comparison of data encryption systems on the example of medical information system.

The qualification work of the master consists of an introduction, 4 sections and conclusions, a list of used sources of 42 items. The work contains 100 pages of text, 35 figures, 41 tables, 7 code listings.

Qualification work for the degree of Master's degree in specialty 121 - Software Engineering, supervisor I.A. Skrypnyk. ZNU Engineering Institute, 2020.

The aim of the research is to study the methods of information protection in information systems, especially medical, their features, as well as to create their own medical information system that will work online. The system should have such functions as: data storage, data deletion and data addition. The main difference between the developed system and most of the existing ones is that the system will be easy to use for different categories of medical staff and more informationally protected.

The existing problems in the field of protection of medical information systems were researched. A detailed comparison of existing solutions and the problems they solve. The types of errors found are classified, the least studied, and most important areas are selected to develop your own solution to the problem. Based on the studied data, created its own medical information system of the endogastroeterology department of the Regional Children's Hospital. Further possibilities of data encryption development in medical information systems are proved.

Key words: MEDICAL INFORMATION SYSTEM, ENDOGASTROETHEROLOGICAL DEPARTMENT OF HOSPITAL, DATABASE MANAGEMENT SYSTEM, MySQL, PHP, SSL PROTOCOL.

ЗМІСТ

ВСТУП

Актуальність теми

За останній час застосування комп'ютерів в медицині надзвичайно підвищився. Практична медицина стає все більш і більш автоматизованою.

Складні сучасні дослідження в медицині неможливі без застосування обчислювальної техніки. Кількість інформації така величезна, що без комп'ютера користувач був би не здатен її сприйняти і обробити.

Нажаль, в нашій країні робота медичних закладів ще не комп’ютеризована так, як в інших країнах. Деякі медичні заклади ще не мають своєї системи, яка б могла містити картки хворого всіх пацієнтів.

Добре побудована медична система пришвидшує постановку діагнозу людині, що може бути вирішальним в порятунку людського життя. Медична система, як і будь-яка інформаційна система, містить дані. Ці дані потребують особливого захисту, через те, що дані, які містить система, є приватними, такими, про які мають право знати лише пацієнт, його лікар, а також медичний персонал, який його лікує. Втрата, зміна або оприлюднення даних про пацієнта можуть зашкодити його здоров’ю та життю або, навіть, призвести до його смерті. Захист даних – це забезпечення цілісності цих даних, тобто забезпечення гарантії того, що дані не будуть змінені.

Існує багато медичних інформаційних систем, але кожна з них має свої недоліки в захисті інформації та зручності використання медичним персоналом, враховуючи те, що медичний персонал може мати різний вік та рівень освіти. Тому актуальним залишається питання розробки медичної системи, дані якої були б надійно захищенні, а сама система була зручною у користуванні для різних категорій медичного персоналу.

У майбутньому наше життя буде ставати ще більш і більш цифровим. Медицина також не є виключенням. Тож, питання створення надійної, з точ-

ки зору захисту інформації, та зручної, з точки зору користування, медичної інформаційної системи є вкрай актуальним.

Мета і завдання дослідження

Мета дослідження полягає у вивченні методів захисту інформації в інформаційних системах, особливо медичних, їх особливостей, а також у створенні власної медичної інформаційної системи, яка буде працювати у режимі онлайн. В системі мають бути такі функції, як: зберігання даних, видалення даних та додавання даних.

Головною відмінністю розроблюваної системи від більшості вже існуючих є те, що система буде зручною у користуванні для різних категорій медичного персоналу та більш інформаційно захищеною.

Об’єкт дослідження

Інформація, яка використовується при роботі медичного персоналу ендогастроентерологічного відділення лікарні.

Предмет дослідження

Захист інформації медичної інформаційної системи.

Методи дослідження

Для розв’язання представлених завдань використовуються такі методи дослідження:

1. Аналіз джерел про інформаційні системи і засоби захищення інформації в них.

2. Проведення аналогії з-поміж існуючих на ринку медичних інформаційних систем.
3. Синтез різних методів захисту інформації.
4. Порівняльний аналіз програмних продуктів.

Наукова новизна одержаних результатів

Одержані результати є наочним відображення переваг та недоліків методів захисту інформації в медичних інформаційних системах. На основі аналізу цих даних у майбутньому можлива розробка єдиної медичної інформаційної системи, придатної у використанні в медичному закладі, дані якої будуть максимально захищенні, а також зручної у користуванні.

Практичне значення одержаних результатів

На базі отриманих результатів можна зрозуміти, які методи захисту інформації є найбільш ефективними. Також, проаналізувавши дану роботу, можна зробити висновки щодо найбільш вдало побудованої медичної інформаційної системи для користувачів. Ознайомитися з найпоширенішими проблемами захисту інформації в таких інформаційних системах та методами їх вирішення.

Також у результаті було отримано медичну інформаційну систему з даними про пацієнтів, які є максимально захищені.

Апробація результатів

Результати даної роботи опубліковані у збірнику «Молода наука-2020», який укладений за результатами XIII університетської науково-практичної конференції студентів, аспірантів і молодих вчених. А також у XXV науково-

технічній конференції студентів, магістрів, аспірантів, молодих вчених та викладачів.

Глосарій

Інформаційна система — це система, яка має такі функції, як: зберігання, пошук та обробка інформації.

Система — це сукупність пов'язаних між собою та із зовнішнім середовищем елементів або частин, функціонування яких спрямовано на отримання конкретного результату.

Система управління базами даних — це сукупність програмних засобів, які забезпечують керування створенням та використанням баз даних.

Бази даних — це дані, які були зібрані та систематизовані таким чином, щоб можна було їх зберігати, замінювати та обробляти.

Програмне забезпечення — це сукупність програм, які реалізують мету й завдання інформаційної системи та забезпечують функціонування технічних засобів системи.

Маскарад – це виконання якихось дій одним користувачем від імені іншого, що має відповідні повноваження.

Троянський кінь – програма, яка з діями, описаними в її документації, виконує деякі інші дії, що ведуть до порушення безпеки системи та деструктивних результатів.

Правове забезпечення — це сукупність норм, виражених у нормативних актах, які встановлюють і закріплюють організацію цих систем, їх цілі, завдання, структуру, функції та правовий статус інформаційної системи.

Комп'ютерні віруси – це певний тип програмних об'єктів, які володіють рядом властивостей, притаманних живим організмам, – вони народжуються, розмножуються та помирають.

Мережний черв'як – різновид програми-вірусу, яка розповсюджується глобальною мережею і не залишає своєї копії на магнітному носії (хоча є й

інші варіанти „черв'яків“, які зберігаються на фізичних носіях у вигляді файлів).

SSL-сертифікат — це засіб захисту особистої інформації користувачів в інтернеті. Якщо на сайті є SSL-сертифікат, в адресному рядку веб-переглядача з'явиться зелений замок і протокол HTTPS. Це означає, що на цьому сайті безпечно вводити пароль або номер банківської картки.

SQL-ін'єкції – це техніка створення запиту до бази даних зловмисником, яка дозволяє йому вкрасти дані користувачів, отримати доступ до їх паролей, видалити базу даних тощо.

1 ДОСЛІДЖЕННЯ ПРОБЛЕМИ ЗАХИСТУ ІНФОРМАЦІЇ В МЕДИЧНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ

1.1 Загальні відомості про інформаційні системи

Система - це сукупність елементів або частин, взаємопов’язаних між собою та із зовнішнім середовищем, функціонування яких спрямоване на отримання конкретного результату.

ІС — це множина різних елементів та зв’язків між ними, що складають систему в цілому.

У сучасній концепції організації інформаційної системи в кожній предметній області виділено дві частини: забезпечення та функціонування. Кожна з них складається з підсистем.

Програмна частина ІС охоплює підсистеми, що реалізують технологію автоматизованої обробки інформації.

У різних інформаційних системах склад цих підсистем однаковий і охоплює відповідно до національних стандартів: інформація, технології, програмне забезпечення, математика, організація та юридичне забезпечення.

Інформаційна підтримка (ІС). Інформація є важливим елементом комп’ютерної інформаційної системи.

Організація інформаційного забезпечення в інформаційній системі особливо важлива.

Система інформаційного забезпечення передбачає створення єдиного інформаційного фонду, систематизацію та уніфікацію показників та документів та розробку формальних методів опису даних.Інформаційне забезпечення охоплює:

- методичні та інструктивні документи;
- єдину систему класифікації та кодування;

– інформаційну базу, яка у свою чергу поділяється на нормативно-довідкові документи, інформаційні повідомлення та інформаційні масиви.

технічна підтримка. Технічні засоби є основою побудови інтелектуальної власності. Потужність фінансування значною мірою визначає склад завдань, що вирішуються в цій предметній області. Технічна підтримка ІС включає комп'ютерне обладнання, методи зв'язку та офісне обладнання. Іншими словами, технічна підтримка - це сукупність взаємопов’язаних технічних засобів, призначених для збору, обробки, передачі, обміну та відображення інформації, необхідної системі управління.

Технічна підтримка сучасних інформаційних систем - це комплекс різних типів обладнання: комп’ютери, периферійне обладнання, засоби автоматичного зчитування даних, оргтехніка, засоби передачі та обміну даними, мережеве обладнання, мультимедійні засоби тощо.

Програмне забезпечення — це сукупність програм, що реалізує цілі та завдання інформаційної системи та забезпечує роботу технічних засобів системи.

Програмне забезпечення включає набір програм, що реалізують функції та завдання автоматизованих інформаційних технологій та забезпечують стабільну роботу апаратного забезпечення.

Програмне забезпечення включає загальносистемні програми та спеціальні програми.

Загальносистемне програмне забезпечення включає програми, призначені для багатьох користувачів для організації процесу бухгалтерського обліку та вирішення загальних проблем обробки інформації.

Спеціальне програмне забезпечення - це сукупність програм, призначених для створення інформаційних технологій з певною функціональною метою.

Математичне програмне забезпечення - це сукупність економіко-математичних методів, моделей та алгоритмів, що знаходяться в інформаційній системі для обробки інформації.

У теорії комп'ютерних систем існує кілька видів інформаційного забезпечення:

– зовнішнє інформаційне забезпечення, яке охоплює систему показників даної предметної сфери, систему класифікацій, первинні документи;

– внутрішнє забезпечення, що охоплює інформаційну базу даних на машинних носіях.

Правове забезпечення - це сукупність стандартів, висловлених у нормативних документах, що визначають і посилюють організацію та зміцнення систем, їх цілі, завдання, структури, функції та правовий статус ІС.

На етапі розвитку автоматизованих інформаційних систем та інформаційних технологій правове забезпечення охоплює правила, пов'язані з договірними відносинами між розробником та клієнтом у процесі створення інтелектуальної власності та ІТ. правове регулювання різних відхилень у процесі, а також необхідність забезпечення процесу розвитку різних видів інтелектуальної власності та інформаційних ресурсів.

Функціональна частина інформаційної системи полягає у вирішенні проблем домену.

Підсистема - це сукупність системних компонентів, які розділяються за певною ознакою. Кожна функціональна підсистема має свій набір завдань, призначених для виконання функцій управління. Основні принципи виокремлення самостійних функціональних підсистем (комплексів задач):

відносна самостійність кожної з них;

наявність відповідного набору функцій і функціональних задач із чітко виявленою локальною ціллю функціонування;

мінімізація складу елементів, що входять у підсистему [1].

1.2 Захист інформації в інформаційних системах

29 березня 2006 року Кабінет Міністрів України видав постанову №373 про «Правила забезпечення захисту інформації в інформаційних, теле-

комунікаційних та інформаційно-телекомунікаційних системах», згідно з якою захисту в системі підлягає:

– відкрита інформація, яка є власністю держави, і у визначенні Закону України "Про інформацію" належить до статистичної, правової, соціологічної інформації, інформації довідково-енциклопедичного характеру та використовується для забезпечення діяльності державних органів або органів місцевого самоврядування, а також інформація про діяльність зазначених органів, яка оприлюднюється в Інтернеті, інших глобальних інформаційних мережах і системах або передається телекомунікаційними мережами (далі – відкрита інформація);

– конфіденційна інформація, яка є власністю держави або вимога щодо захисту якої встановлена законом, у тому числі конфіденційна інформація про фізичну особу (далі – конфіденційна інформація);

– інформація, що становить державну або іншу передбачену законом таємницю (далі – таємна інформація) [2].

Також ця постанова регулює основні вимоги до забезпечення захисту інформації в системі та організаційні засади захисту інформації.

1.2.1 Вимоги до забезпечення захисту інформації в системі

Під час обробки в системі відкриту інформацію слід зберігати недоторканою, захищати від несанкціонованих дій, які можуть призвести до її випадкових або навмисних змін або знищення.

Усі користувачі повинні отримувати публічну інформацію. Лише ідентифіковані та сертифіковані користувачі, яким надано відповідні повноваження, можуть змінювати або знищувати публічну інформацію.

Спроби змінити або знищити публічну інформацію несанкціонованими користувачами, невстановленими користувачами або несанкціонованими користувачами представленої особи повинні бути заблоковані.

При обробці конфіденційної та конфіденційної інформації повинен бути забезпечений її захист від несанкціонованого та неконтрольованого доступу, зміни, знищення, копіювання, розповсюдження.

Доступ до конфіденційної інформації мають лише ідентифіковані та уповноважені користувачі. Спроба ідентифікованих осіб або користувачів отримати доступ до такої інформації відповідно до Невідомого посвідчення ідентифікатора, наданого під час авторизації.

Система дозволяє користувачеві виконати одну або кілька операцій з обробки конфіденційної інформації або позбавити його цього права.

Вимоги до захисту в інформаційній системі, що містить державну таємницю, визначені законодавством у галузі охорони державної таємниці.

Забезпечення захисту в системі конфіденційної інформації, що не є державною таємницею, здійснюється відповідно до вимог захисту конфіденційної інформації.

Вимоги щодо захисту від несанкціонованого блокування в інформаційній системі визначаються її власником (менеджером), якщо інше не передбачено законом під час обробки цієї інформації або системи. У системі здійснюється обов’язкова реєстрація:

– результатів ідентифікації та автентифікації користувачів;

– результатів виконання користувачем операцій з обробки інформації;

– спроб несанкціонованих дій з інформацією;

– фактів надання та позбавлення користувачів права доступу до інформації та її обробки;

– результатів перевірки цілісності засобів захисту інформації.

Дані реєстрації може аналізувати лише користувач, який уповноважений управляти засобами захисту інформації та контролем захисту інформації в системі (адміністратор безпеки).

Реєстрація здійснюється автоматично, а дані реєстрації захищені від модифікації та знищення користувачами, які не мають повноважень адміністратора безпеки.

Запис спроби несанкціонованої діяльності з інформацією, що становить державну таємницю, а також конфіденційною інформацією про осіб, які відповідно до законодавства віднесені до персональних даних, повинен супроводжуватися повідомленням. (адміністратор безпеки).

Ідентифікація та автентифікація користувачів, надання та позбавлення доступу до інформації та її прав на обробку, контроль цілісності засобів захисту системи виконуються автоматично.

Передача конфіденційної та конфіденційної інформації з однієї системи в іншу повинна здійснюватися в зашифрованому вигляді або через захищені канали зв'язку відповідно до законодавчих вимог щодо технічного та криптографічного захисту інформації.

Порядок підключення систем, в яких обробляється конфіденційна та конфіденційна інформація для глобальних мереж передачі даних, визначається законом.

Система контролює цілісність програмного забезпечення, що використовується для обробки інформації, запобігає несанкціонованим модифікаціям та усуває наслідки такої модифікації. Контролюється також цілісність програмних та технічних засобів захисту інформації. У разі порушення їх цілісності обробка в системі інформації припиняється [2].

1.2.2 Організаційні засади захисту інформації

Захист інформації на всіх етапах створення та функціонування системи здійснюється відповідно до плану захисту даних у системі, розробленого службою захисту інформації..

План захисту інформації в системі містить:

– завдання захисту, класифікацію інформації, яка обробляється в системі, опис технології обробки інформації;

– визначення моделі загроз для інформації в системі;

– основні вимоги щодо захисту інформації та правила доступу до неї в системі;

– перелік документів, згідно з якими здійснюється захист інформації в системі;

– перелік і строки виконання робіт службою захисту інформації [2].

1.3 Основні загрози цілісності інформації в медичних інформаційних системах

За метою впливу розрізняють три основні типи загроз безпеці інформаційних систем:

– загрози порушення конфіденційності інформації;

– загрози порушення цілісності інформації;

– загрози порушення працездатності системи (відмови в обслуговуванні).

Погрози порушення конфіденційності спрямовані на розголошення конфіденційної або конфіденційної інформації. Коли ці загрози реалізуються, інформація стає відомою людям, які не повинні мати до неї доступу. З точки зору комп'ютерної безпеки, ризик порушення конфіденційності виникає у разі отримання несанкціонованого доступу до певної конфіденційної інформації, що зберігається в комп'ютерній системі або передається від однієї комп'ютерної системи до іншої.

Погрози цілісності інформації, що зберігається в комп'ютерній системі або передається по каналу зв'язку, призначені для їх зміни або спотворення, що призведе до втрати якості або повного знищення. Цілісність інформації може бути порушена, зокрема, зловмисником, а також впливом зовнішнього середовища, що оточує систему. Ці загрози особливо важливі у випадку систем передачі інформації - комп’ютерних мереж та телекомунікаційних сис-

тем. Цю інформацію не слід навмисно порушувати шляхом дозволених змін, які вносять уповноважені особи з розумною метою.

Небезпека інвалідності (відмова в обслуговуванні) спрямована на створення таких ситуацій, коли певні цілеспрямовані дії або знижують ефективність ІВ, або блокують доступ до її ресурсів. Наприклад, якщо один користувач системи намагається отримати доступ до послуги, а інший вживає заходів, щоб заблокувати цей доступ, перший користувач отримає відповідь від служби. Блокування доступу до ресурсу може бути постійним або тимчасовим.

Порушення конфіденційності та цілісності інформації, а також доступності та цілісності деяких ресурсів ІС можуть бути спричинені різними небезпечними наслідками для інформаційних систем. Сучасні автоматизовані системи обробки інформації - це складні системи, що складаються з великої кількості елементів з різним ступенем автономності, які взаємопов’язані та обмінюються даними. Практично кожен елемент може зазнати впливу зовнішнього впливу або зламатися. Елементи інформаційної системи можна розділити на такі групи:

- апаратні засоби;
- програмне забезпечення;
- дані;
- персонал.

Небезпечні впливи для ІС можна розділити на випадкові та шкідливі. Аналіз досвіду проектування, виготовлення та експлуатації інформаційної системи показує, що інформація піддається різним випадковим впливам на всіх етапах функціонування інформаційної системи. Причинами для несподіваних впливи можуть бути:

- помилки в програмному забезпеченні;
- помилки в роботі обслуговуючого персоналу та користувачів;
- завади в лініях зв’язку, спричинені впливом зовнішнього середовища.

Навмисні загрози пов'язані з навмисними діями зловмисника. Порушником може бути робітник, відвідувач, конкурент, робітник тощо. Дії злочинця можуть бути мотивовані різними мотивами: незадоволеність працівника своєю кар'єрою, матеріальна зацікавленість, цікавість, конкурентна боротьба, прагнення до самоствердження тощо. Враховуючи можливість найбільш небезпечної ситуації, спричиненої діями злочинця, ми можемо скласти гіпотетичну модель потенційного правопорушника:

- кваліфікація порушника може бути на рівні розробника даної системи;
- порушником може бути як стороння особа, так і законний користувач системи;
- порушнику відома інформація про принципи роботи системи;
- порушник вибирає найслабшу ланку в захисті.

Можна виділити такі приклади навмисних загроз:

- несанкціонований доступ сторонніх осіб, що не належать до числа співробітників, та ознайомлення з конфіденційною інформацією;
- ознайомлення співробітників з інформацією, до якої вони не повинні мати доступ;
- несанкціоноване копіювання програм і даних;
- викрадення носіїв інформації, що містять конфіденційну інформацію;
- викрадення роздрукованих документів;
- навмисне знищення інформації;
- несанкціонована модифікація співробітниками фінансових документів, звітності та баз даних;
- фальсифікація повідомлень, що передаються по каналах зв’язку;
- відмова від авторства повідомлення, переданого каналом зв’язку;
- відмова від факту отримання інформації;
- пошкодження інформації, викликане впливом вірусів;
- пошкодження архівної інформації, розміщеної на змінних носіях;

- викрадення обладнання.

Несанкціонований доступ - найпоширеніший і універсальний тип порушень. Суть несанкціонованого доступу полягає в тому, щоб користувачі (порушники) отримали доступ до об’єкта з порушенням правил розмежування доступу, встановлених відповідно до політики безпеки, прийнятої організацією.

Несанкціонований доступ використовує будь-які помилки безпеки, і це можливо при нераціональному виборі системи безпеки, неправильній установці та налаштуванні. Несанкціонований доступ може здійснюватися за допомогою звичайних засобів ІС, а також спеціально створеного обладнання та програмного забезпечення.

Ось список основних каналів несанкціонованого доступу, за допомогою яких зловмисник може отримати доступ до ІС-компонентів та здійснити крадіжку, модифікацію та / або пошкодження інформації::

- усі штатні канали доступу до інформації (комп’ютери користувачів, оператора, адміністратора системи; засоби відображення та документування інформації; канали зв’язку) при їх використанні порушниками, а також законними користувачами за межами їх повноважень;
- технологічні пульти управління;
- лінії зв’язку між апаратними засобами ІС;

Із всього розмаїття способів та прийомів несанкціонованого доступу зупинимося на найбільш розповсюджених та зв’язаних між собою порушеннях:

- перехоплення паролів;
- «маскарад»;
- незаконне використання привілеїв.

Злом паролів виконується спеціально розробленими програмами. Коли зареєстрований користувач намагається увійти, програма зупинки є такою ж, як введення імені користувача та пароля на екрані, які надсилаються власнику програми захоплення, після того, як відображається повідомлення про по-

милку та відновлення контролюв операційній системі. Користувач вважає, що помилився під час введення пароля. Ви входите знову і отримуєте доступ до системи. Власник програми-перехоплювача, отримавши логін та пароль законного власника, може використовувати їх на свою користь. Є й інші способи блокування мережі.

"Маскарад" - це виконання дій одним користувачем від імені іншого користувача з відповідними повноваженнями. Метою "маскараду" є надання певних дій іншому користувачеві або надання іншим користувачам повноважень та привілеїв. Приклад реалізації "Маскараду":

- вхід в систему під іменем та паролем іншого користувача (такому „маскараду“ передує перехоплення паролю);
- передача повідомлень в мережі від імені іншого користувача.

«Маскарад» особливо небезпечний в електронних платіжних банківських системах, де неправильна ідентифікація клієнта через "маскарад" зловмисника може завдати значної шкоди законному клієнту банку.

Незаконне використання привілеїв. Більшість систем безпеки визначають певні набори привілеїв для виконання зазначених функцій. Кожен користувач отримає свої привілеї: звичайні користувачі - мінімум, адміністратори - максимум. Несанкціоноване позбавлення привілеїв, наприклад, шляхом "маскараду", змушує злочинця здійснювати певні дії, минаючи систему захисту. Слід зазначити, що привілеї можуть бути незаконно введені або через помилки в системі безпеки, або через недбалість адміністратора та системні привілеї.

Загрози комп’ютерним мережам можна розглядати окремо. Головною особливістю будь-якої комп’ютерної мережі є те, що її компоненти розгортаються у просторі. Зловмисник може використовувати пасивні та активні методи вторгнення для проникнення в комп’ютерну мережу. У разі пасивного вторгнення (перехоплення інформації) зловмисник лише спостерігає за проходженням інформації через канал зв'язку, не втручаючись у потік інформації чи інформаційний зміст. Як правило, зловмисник може ідентифікувати

пункти призначення та ідентифікатори або просто факт повідомлення, його довжину та частоту обміну, якщо вміст повідомлення неможливо розпізнати - для аналізу трафіку (потоку повідомлень) на цьому каналі.

Під час активного вторгнення зловмисник прагне замінити інформацію, передану в повідомленні. Він може вибірково змінювати або модифікувати повідомлення, затримувати або змінювати порядок повідомлень. Зловмисник може також скасувати та затримати всі повідомлення, що передаються по каналу. Такі дії можуть кваліфікуватися як відмова у передачі повідомлення.

Комп’ютерні мережі характеризуються тим, що крім звичних локальних атак, що здійснюються в рамках однієї системи, на мережеві об’єкти здійснюються так звані віддалені атаки. Зловмисник може знаходитися на відстані тисяч миль від об’єкта, що атакується, не лише певного комп’ютера, а й інформації, що передається через мережеві канали зв'язку. Віддалена атака визначається як зловмисний інформаційний вплив на розподілену комп'ютерну мережу, що здійснюється програмно за допомогою каналів зв'язку.

У таблиці 1.1 показані основні шляхи реалізації загроз безпеці ІС при впливі на її компоненти. Ця таблиця дає тільки загальну картину того, що може відбутися з системою, а конкретні обставини та особливості повинні розглядатися окремо.

Таблиця *1.1*

Шляхи реалізації загроз безпеці ІС

Об’єкти впливу	Порушення конфіденційності інформації	Порушення цілісності інформації	Порушення працездатності системи

Апаратні засоби	Несанкціонований доступ – підключення; використання ресурсів; викрадення носіїв	Несанкціонований доступ – підключення; використання ресурсів; модифікація, зміна режимів	Несанкціонований доступ – зміна режимів; виведення з ладу; пошкодження
Програмне забезпечення	Несанкціонований доступ – копіювання; викрадення; перехоплення	Несанкціонований доступ – впровадження «троянських коней», «вірусів», «черв'яків»	Несанкціонований доступ – спотворення; знищення; підміна
Дані	Несанкціонований доступ – копіювання; викрадення; перехоплення	Несанкціонований доступ – спотворення; модифікація	Несанкціонований доступ – спотворення; знищення; підміна
Персонал	Розголошення; передача відомостей про захист; халатність	«Маскарад»; вербування; підкуп персоналу	Покидання робочого місця; фізичне усунення

У таблиці 1.1 наведено специфічні назви та терміни: «троянський кінь», «вірус», «черв'як». Хоча ці назви мають жаргонний відтінок, вони уже ввійшли в загальноприйнятий комп'ютерний лексикон. Дамо коротку характеристику цих розповсюджених загроз безпеці ІС.

«Троянський кінь» - це програма, яка, крім дій, описаних у її документах, є ще одним заходом, який загрожує безпеці системи та призводить до руйнівних результатів. Аналог такої програми з давньогрецьким «троянським конем» є цілком виправданим, оскільки в обох випадках безперечна оболонка становить серйозну загрозу. Термін "троянський кінь" вперше застосував Ден Едвардс, який згодом став членом Агенції національної безпеки США. Тро-

янський кінь використовує обман, щоб заплутати користувача в управлінні програмами із прихованими загрозами. Зазвичай кажуть, що така програма буде виконувати деякі корисні функції. Зокрема, такі програми одягнені як деякі корисні утиліти.

Загроза "Троянський кінь" - це блок додаткових команд, побудований за допомогою джерела, який потім передається користувачам. Цей командний блок може працювати будь-якою мовою (дати, стан системи) або зовнішньою командою. Користувач, який розробляє таку програму, загрожує як її ресурсам, так і ІВ в цілому. Ось деякі руйнівні функції, реалізовані троянськими конями:

- Знищення інформації. Вибір об’єктів та способів знищення визначається фантазією та цілями автора зловмисної програми.
- Перехоплення та передача інформації. Зокрема, відомі програми, які здійснюють перехоплення паролів, що набираються на клавіатурі.
- Цілеспрямована модифікація тексту програми, яка реалізує функції безпеки та захисту системи.

Загалом, «троянський кінь» завдає збитків ІС, викрадаючи інформацію та чітко пошкоджуючи системне програмне забезпечення. Троянський кінь - одна з найнебезпечніших загроз безпеці ІС. Радикальним способом захисту від цієї загрози є створення закритого середовища для реалізації програм, які повинні зберігатися та захищатися від несанкціонованого доступу. Однак установка нового програмного забезпечення на ваш комп’ютер має бути дозволена лише адміністраторами, чого зазвичай важко досягти.

Комп’ютерні 'віруси” - це тип програмного об’єкта, який має ряд властивостей, властивих живим організмам - вони народжуються, розмножуються та вмирають. Термін "вірус" стосовно комп'ютерів був введений Фредом Коеном з Університету Південної Каліфорнії. Історично першим визначенням Коена було: 'Комп’ютерний вірус” - це програма, яка може заражати інші програми, модифікуючи їх, включаючи можливо змінену копію, тоді як остання зберігає свою здатність до відтворення. Ключовими поняттями в іде-

нтифікації комп’ютерного вірусу є здатність вірусу самовідтворюватися та модифікувати код заражених програм.

Мережевий хробак - це тип вірусної програми, яка поширюється у глобальній мережі і не залишає копії на магнітному носії (хоча існують і інші версії «хробаків», які зберігаються на фізичному носії як файли). Перші версії «хробаків» були розроблені для пошуку в мережі інших комп’ютерів із вільними ресурсами, щоб забезпечити можливість виконання розподілених обчислень. При правильному використанні черв’ячна технологія може бути надзвичайно корисною. Наприклад, "хробак" World Wide Web Worm формує індекс пошуку веб-сайтів. Однак «хробак» легко перетворюється на шкідливе програмне забезпечення.

Мережеві хробаки - найнебезпечніший тип шкідливого програмного забезпечення, оскільки вони можуть атакувати будь-яку з величезної кількості комп’ютерів, підключених до Всесвітньої павутини або інших мереж. Для захисту від «хробака» використовуються засоби блокування несанкціонованого доступу до внутрішньої мережі.

Слід зазначити, що "троянський кінь", комп'ютерні віруси та мережеві "хробаки" є одними з найнебезпечніших загроз для ІС. Щоб захистити себе від зловмисного програмного забезпечення, вам потрібно виконати ряд кроків:

- виключення несанкціонованого доступу до виконуваних файлів;
- тестування нових програм;
- контроль цілісності виконуваних файлів та системних областей;
- створення замкнутого середовища виконання програм [3].

1.4 Організація захисту медичної інформації

Висока концентрація наборів медичних даних, відсутність елементарного контролю за їх зберіганням та відносно низький рівень надійності тех-

нічних засобів створюють серйозні проблеми у забезпеченні зберігання інформації.

Інформація, зібрана та оброблена під час роботи медичної інформаційної системи, є досить вразливою і може бути знищена та використана без дозволу. Велика кількість різних компонентів, операцій, ресурсів та об’єктів медичної інформаційної системи створює дуже привабливе середовище для різних вторгнень та несанкціонованих операцій.

Для створення безпечної інформаційної системи охорони здоров’я необхідно створити системи захисту, що є регулярним процесом, що здійснюється на всіх етапах життєвого циклу медичної інформаційної системи.

Існує кілька етапів створення систем захисту медичної інформаційної системи:

1. Визначення інформаційних та технічних ресурсів і об'єктів медичної інформаційної системи охорони здоров'я, що підлягають захисту.

2. Виявлення безлічі потенційно можливих погроз і каналів витоку інформації.

3. Проведення оцінки вразливості і ризиків ресурсів медичної інформаційної системи при наявній безлічі погроз і каналів витоку.

4. Визначення вимог до системи захисту інформації.

5. Здійснення вибору засобів захисту інформації та їхніх характеристик.

6. Впровадження і організація використання обраних мір, способів і засобів захисту.

7. Здійснення контролю цілісності і управління системою захисту.

Специфічними особливостями рішення задачі створення систем захисту є: неповнота і невизначеність вихідної інформації про склад медичної інформаційної системи і характер погроз; багатокритеріальність задачі, пов'язана з необхідністю обліку великої кількості показників (вимог) системи захисту інформації (СЗІ); наявність як кількісних, так і якісних показників,

які необхідно враховувати при рішенні задач розробки і впровадження СЗІ; неможливість застосування класичних методів оптимізації [4].

1.5 Методи захисту інформації в інформаційних системах

Універсального методу захисту даних в інформаційних системах не існує.

Керування доступом. Керування доступом – це метод захисту інформації, при якому доступ до даних інформаційної системи має тільки певна група осіб, яка володіє ключами та паролями, необхідними для доступу в цю систему. Керування доступом передбачає:

1. Ідентифікацію користувачів та персоналу системи (надання кожному окремому об’єкту персональних ідентифікаторів).
2. Розпізнавання, або встановлення дійсності окремого користувача за тим ідентифікатором, який він надав.
3. Реєстрація користувача.
4. Реагування. Наприклад, у разі неправильно введених даних ідентифікації користувачу забороняють доступ.

Криптографічні методи шифрування. Головна ідея цього методу захисту інформації полягає в тому, що незахищене (відкрите) повідомлення, яке передається, може бути перехоплено зловмисником під час передачі. Для того, щоб цього запобігти, це повідомлення шифрується, тобто перетворюється на шифрований текст. Коли повідомлення потрапляє до отримувача, він його розшифровує, використовуючи свої ключі для розшифрування, тим самим отримуючи первинне повідомлення.

Шифрування може бути симетричним та асиметричним. Різниця в симетричному та асиметричному шифруванні полягає в тому, що для шифру-

вання та розшифрування повідомлення в симетричному методі шифрування використовується один і той самий ключ, а в асиметричному методі використовується два ключа: відкритий та закритий, тобто для шифрування повідомлення використовується один ключ, а для розшифрування інший.

Існує багато алгоритмів як симетричного, так і асиметричного шифрування. Прикладами алгоритмів симетричного шифрування можуть слугувати такі алгоритми шифрування, як: AES, DES, ГОСТ 28147-89 та інші. Прикладами алгоритмів асиметричного шифрування можуть слугувати такі алгоритми: RSA, DSA, схема Ель-гамаля, Діффі-Хеллмана та інші.

Цифровий підпис. Технологія шифрування за допомогою цифрового підпису заснована на асиметричному методі шифрування даних. Спочатку за допомогою хеш-функції (хеш-функція – це функція, яка перетворює великий об'єм даних в дані фіксованого розміру, тобто хеш) відбувається отримання хешу. Після цього отриманий хеш шифрується закритим ключем користувача. Далі користувач передає іншому користувачу дані, підписані цифровим підписом. Існує багато алгоритмів для шифрування даних за допомогою ЕЦП: RSA, DSA, Ель-Гамаля та інші [5]. В RSA для підпису потрібно використати секретний ключ:

$$s = m^d mod\ n,$$

де n=p*q, p та q два великих секретних простих числа, обраних адресатом, а d>p/2 та d>q/2.

Тепер використавши публічний ключ можна отримати початкове повідомлення:

$$\mathrm{m} = s^e mod\ n.$$

Таким чином, отримавши повідомлення та підписавши його, а потім розшифрувавши за допомогою відкритого ключа користувача, ми можемо бути впевнені, що лише він міг підписати його.

У алгоритму є свої переваги: він тестувався протягом багатьох років, і не було знайдено методів, які би суттєво зменшували криптостійкість алгоритму. Але у порівнянні з асиметричними алгоритмами на базі еліптичних

кривих у скінченному полі алгоритм на базі скінченного поля на просторі цілих чисел має такі недоліки: для його реалізації потрібно більший ключ і відповідно більші затрати часу [41].

1.6 Аналіз захисту існуючої медичної інформаційної системи

Прикладом медичної інформаційної системи може служити всім відома та дуже популярна вітчизняна система Helsi. (рисунок 1.1)

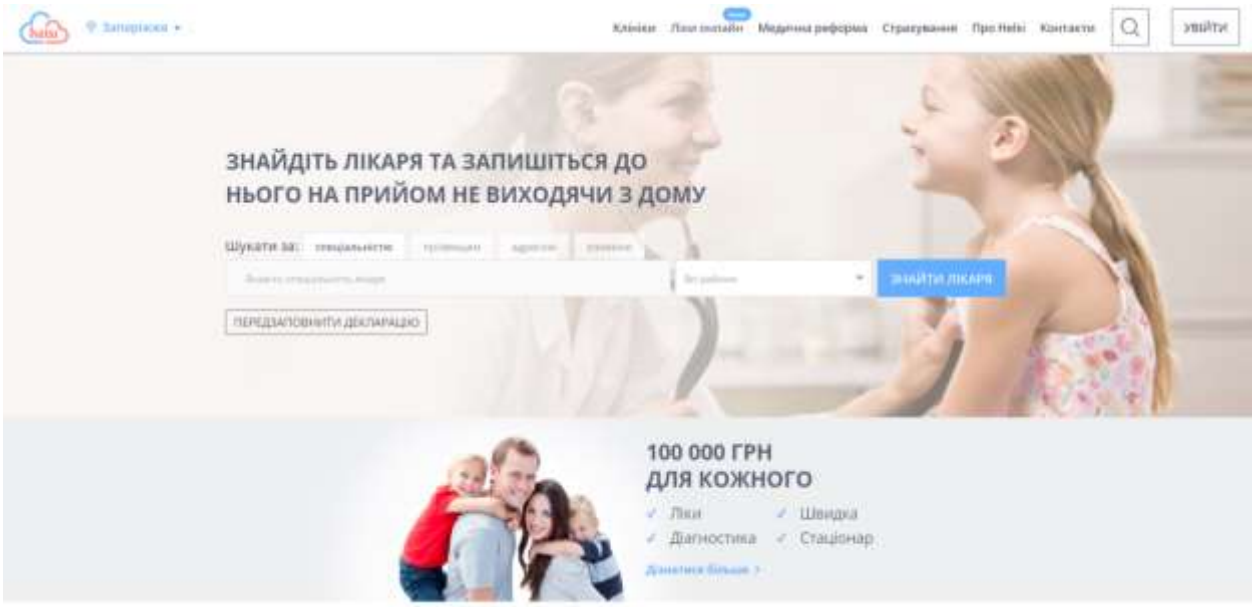

Рисунок *1.1* Медична система Helsi

Вона надійно захищає дані пацієнтів. Всі дані зберігаються у датацентрі, який отримав сертифікат комплексної системи захисту інформації від Державної служби спеціального зв'язку та захисту інформації України згідно з державним стандартом України з захисту інформації [7-8]. Система Helsi має комплексну систему захисту інформації, підтверджену державними службами. Система захисту інформації в цій системі включає ідентифікацію, ав-

тентифікацію, цифровий підпис з використанням одноразових паролів та міток часу.(рисунок 1.2)

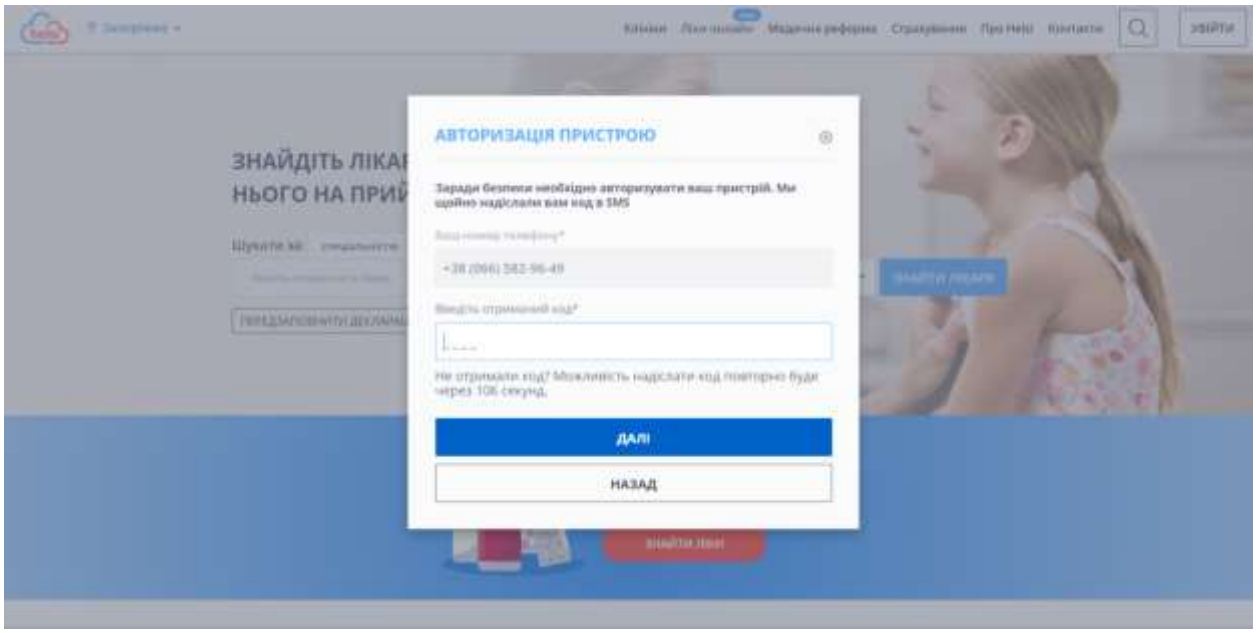

Рисунок *1.2* Авторизація в медичній системі Helsi

Проте ця система порівняно нова та недосконала. Недоліки в медичній системі Helsi пов’язані не з захистом інформації, а з недоліками в медичній реформі, а саме той факт, що укладення договорів із сімейними лікарями відбувалося з використанням номеру телефону, який було використано в подальшому, як логін для входу в систему. Цей недолік полягає в тому, що, якщо один член родини використав номер телефону іншого члена своєї родини для укладання договору зі своїм лікарем, то той інший член родини цього користувача, чий номер було використано, вже не зможе зареєструвати себе в системі. Таким чином головна мета, для якої створювали систему втрачається, через те, що людина не може записати себе до лікаря.

1.7 Забезпечення безпеки та конфеденційності інформації медичної системи Health24

Сервіс Health24 (рисунок 1.3) розміщений у хмарній інфраструктурі GigaCloud. Компанія гарантує високий рівень безпеки своїх хмарних платформ, а її комплексна система захисту інформації (КСЗІ) атестована на відповідність вимогам Державної служби спеціального зв'язку та захисту інформації України.

Криптографічний захист. Мова йде про криптографічні протоколи, шифрування, ключі, вироблення і перевірку електронного цифрового підпису й хешування.

Політика інформаційної безпеки. Права доступу до апаратної частини GigaCloud чітко регламентуються в залежності від посади. Крім того, жоден працівник компанії не має доступу до інформації, що зберігається на серверах. Управління інформаційною безпекою у хмарі підтверджене сертифікатом ISO 27001. Це міжнародний стандарт, що встановлює вимоги до системи менеджменту інформаційної безпеки та підтверджує здатність компанії захищати свої інформаційні ресурси [37].

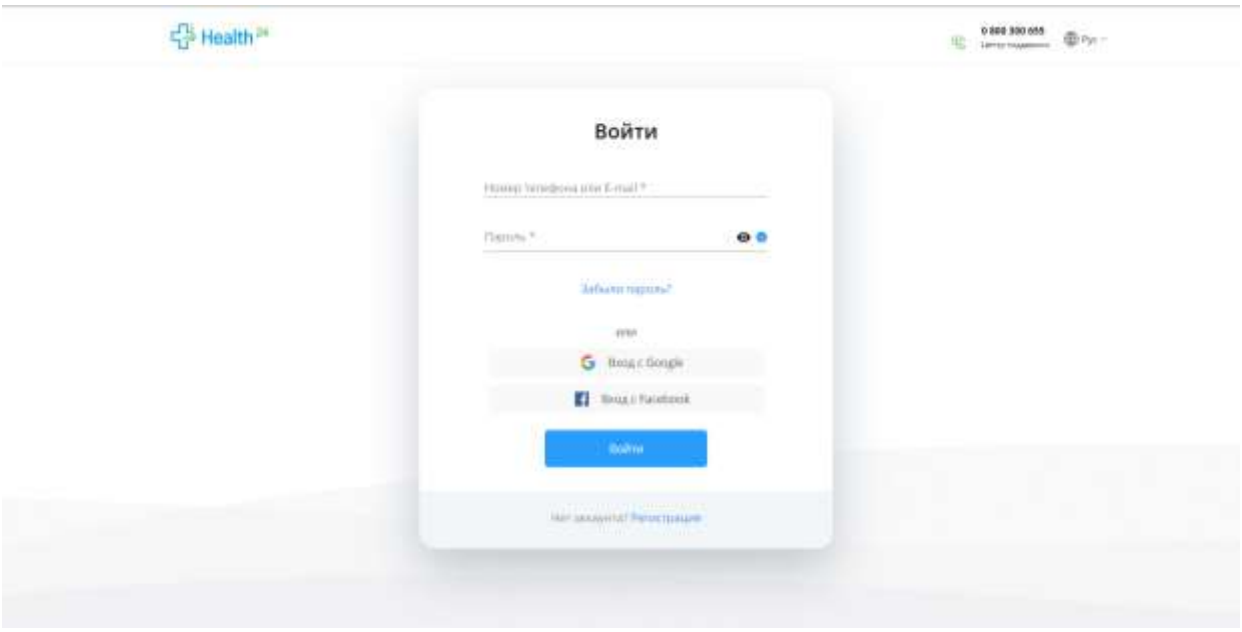

Рисунок *1.3* Авторизація користувача в медичній системі Health24

1.7.1 Комплекс засобів для захисту персональних даних системи Health24

Персональні медичні записи захищені від несанкціонованого доступу з боку інших учасників системи за допомогою:

1. Ідентифікації та аутентифікація облікового запису користувача.
2. Розмежування прав доступу до електронної медичної карти: Архітектура побудови системи, при якій облікові дані, медична інформація та персональні дані зберігаються в окремих місцях. Дозволяє реалізувати настроювані схеми обмеження прав доступу до інформації.
3. Протоколів шифрування SSL і TLS: Для забезпечення безпечного обміну даними в систему додатково вбудована підтримка протоколів шифрування SSL (2.0 та 3.0) і TLS. Всі дані, що знаходяться в процесі передачі, шифруються з використанням протоколу SSL/TLS[37].
4. Шифрування даних за допомогою електронного цифрового підпису.

2 ДОСЛІДЖЕННЯ МЕТОДІВ ЗАХИСТУ В МЕДИЧНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ

2.1 Захист інформаційної системи, як захист бази даних

Інфомаційні системи являють собою базу баних. Тож, коли йде мова про захист інформації в інформаційних системах, то цей захист можна розглядати, як захист даних у базі даних.

База даних - це комп'ютерна система, яка використовується для зберігання та отримання інформації. Реляційна система управління базами даних(далі СУБД) - це комп'ютеризована система управління базами даних, яка використовує реляційні методи для зберігання та отримання даних. Найпоширеніший тип бази даних - це реляційна база даних, яка є електронною таблицею, в якій певні дані можуть бути реорганізовані та доступні різними способами.

Незалежно від конкретної архітектури, в базі даних запитувач (наприклад, програма чи операційна система) повинен отримати доступ до зазначеної бази даних і вимагати доступу до неї. Такі запити можуть включати, наприклад, простий запит до каталогу або комбінацію транзакцій та транзакцій, що використовуються для читання, модифікації та додавання записів до бази даних. Ці запити виконуються з використанням мов запитів високого рівня, таких як SQL (мова структурованих запитів). Наприклад, SQL використовується для створення інтерактивних запитів для отримання інформації та оновлення баз даних, таких як DB2 International Business Machines (IBM) або Microsoft SQL Server та продуктів баз даних Sybase, Computer Associates та Oracle. Термін "запит" відноситься до набору команд для отримання даних із збереженої бази даних.

Одне з важливих питань для баз даних - це безпека. Бази даних часто містять конфіденційний або іншим чином конфіденційний матеріал, який вимагає певної форми захисту для запобігання доступу. Наприклад, медичні

записи вважаються надзвичайно особистими та конфіденційними. Тому доступ до медичних карток, як правило, обмежений певними користувачами. Інші приклади конфіденційних документів включають, але не обмежуючись ними, номери кредитних карток та персональні ідентифікаційні номери (ПІН-коди), що використовуються у фінансових операціях, та документи працівників. З цією метою в звичайних системах управління базами даних часто застосовуються профілі користувачів, які визначають рівень дозволів. Чи може користувач отримати доступ до певних даних, буде залежати від рівня дозволу користувача, вказаного в його профілі.

Однак через нав'язливі методи вторгнення (відстеження, маніпуляції та інші форми прослуховування) зловмисники все ще можуть отримати доступ до конфіденційної інформації, перехоплюючи запити до бази даних або запити конфіденційної інформації. Ця проблема ускладнюється тим, що мови високого рівня, що використовуються для генерації запитів, як правило, дуже легко читаються (наприклад, для інтерпретації та усунення несправностей). Іншими словами, оскільки запити та результати часто передаються через мережу як високочитабельні, вміщений у них чутливий матеріал може бути легко ідентифікований, якщо його перехопить зловнисник [9].

2.2 Протоколи захисту

Одним із способів захисту конфіденційних матеріалів в операціях з базами даних є використання протоколів, що використовуються для безпечної передачі даних через Інтернет, наприклад, рівень захищених сокетів (SSL) або протокол безпечної передачі гіпертексту (S-HTTP). Такі протоколи використовують метод «все» або «нічого», тобто шифрування всіх документів або транзакцій протягом усього сеансу. Однак, оскільки багато запитів до бази даних повертають великі обсяги даних (можливо, тисячі записів результатів), шифрування всього набору результатів може перевантажити системні ресурси. Зокрема, коли необхідно надати невелику частину результатів (наприклад,

номер кредитної картки з 16 символів, ідентифікаційний номер пацієнта тощо), шифрування всіх результатів буде марним.

Відповідно, існує потреба в вдосконаленому методі захисту конфіденційної інформації в транзакції бази даних [9].

2.2.1 Протокол захисту інформації SSL

SSL-сертифікат — це засіб захисту особистої інформації користувачів в інтернеті. Якщо на сайті є SSL-сертифікат, в адресному рядку веб-переглядача з’явиться зелений замок і протокол HTTPS. Це означає, що на цьому сайті безпечно вводити пароль або номер банківської картки [38]. HTTPS також дозволяє отримувати точніші дані про відвідуваність сайту. Коли користувач переходить з HTTPS на HTTP, інформація про перехід втрачається. Отже, якщо ваш сайт працює за протоколом HTTP, а користувач перейшов із сайту HTTPS, це буде відображено у вашій статистиці як прямий перехід.

Які типи SSL-сертифікатів? Сертифікати різняться за своїми характеристиками та рівнем валідації. Типи сертифікатів за рівнем автентифікації:

a. Сертифікати, що підтверджують тільки доменне ім'я (Domain Validation - DV).

b. Сертифікати, що підтверджують домен і організацію (Organization Validation - OV).

c. Сертифікати, з розширеною перевіркою (Extendet Validation - EV).

Сертифікати DV. Тільки сертифікати доменних імен - це найпростіші сертифікати, і їх головною перевагою є те, що вони видаються автоматично. На відміну від інших, вам не потрібно перевіряти жодні документи. Зазвичай, коли ви перевіряєте такий сертифікат, надсилається лист із спеціальним посиланням, на яке ви повинні перейти, щоб підтвердити, що ви є власником домену, на якому видано сертифікат. Головне - надіслати цей лист лише на

електронний лист, вказаний у псевдодескреторі. Електронна пошта, зареєстрована в пошті або в записі домену MX.

Сертифікати OV. Сертифікати, що підтверджують домен та організацію, не може отримати фізична особа, лише юридична особа. У такому сертифікаті вже буде вказано назву організації. Термін видачі таких сертифікатів зазвичай становить від 3 до 10 робочих днів, залежно від центру сертифікації. Це пов’язано з тим, що центр сертифікації перевіряє, чи існує організація насправді та чи володіє вона вказаним доменом. Методи та кількість перевірок залежать від центру сертифікації, тому час прийому знаходиться в цьому діапазоні.

Сертифікати EV. Ці сертифікати мають так звану «green bar» - на вебсайті, який отримав сертифікат, з’являється зелена смужка в адресному рядку браузера із зазначенням назви організації, яка отримала сертифікат. Це найдорожчі сертифікати, і вам найважче отримати їх. Видача займає від 10 до14 днів [39].

2.2.1.1 Типи SSL сертифікатів

Існують такі типи SSL сертифітів:

A. Звичайні SSL сертифікати. Вони є звичайними сертифікатами, які видаються автоматично і лише підтверджують домен. Підходить для всіх сайтів.
B. SGC сертифікати. Сертифікати з підтримкою підвищення рівня шифрування. Підходить для дуже старих браузерів, які підтримували лише 40 або 56 біт шифрування. При використанні цього сертифіката рівень шифрування примусово збільшується до 128 біт. Зараз ці сертифікати не використовують.
C. Wildcard сертифікати. Використовуються, коли потрібно забезпечити шифрування для всіх піддоменах, окрім основного домену. Наприклад: існує домен domain.com, вам потрібно встановити

той самий сертифікат на support.domain.com, forum.domain.com і billing.domain.com.

D. SAN сертифікати. Корисний, якщо ви хочете використовувати один і той же сертифікат для кількох різних доменів, розміщених на одному сервері. Як правило, такий сертифікат складається з 5 доменів.
E. EV сертифікати. Це ті самі сертифікати браузера з розширеним підтвердженням та зеленою смужкою, про які було згадано вище.
F. Сертифікати с підтримкою IDN. англ. IDN - Internationalized Domain Names - багатомовні доменні імена. Доменні імена, адаптовані до національних алфавітів. Іншими словами - сертифікати, що підтримують кирилицю [39].

2.3 Порівняння захисту інформації в медичних інформаційних систмах з використанням медичних інтернет-баз даних PubMed (MEDLINE), CINAHL, ProQuest Nursing та Allied Health Source

Для долсідження засобів захисту інформації в різних медичних інформаційних системах були зібрані зібрані дані з трьох інтернет-баз даних: PubMed (MEDLINE), CINAHL, ProQuest Nursing та Allied Health Source. Проаналізувавши результати, можна зробити висновок, що двома найбільш часто обговорюваними методами безпеки, у згаданих медичних системах, було використання брандмауерів та криптографії. [26]

2.3.1 Використання брандмауерів для захисту мединих інформаційних систем

Найчастіше обговорюваною технікою безпеки було впровадження брандмауерів для захисту інформаційно-технологічної системи організацій охорони здоров'я [9, 10, 11, 12, 13]. Хоча відомо, що брандмауери можуть коштувати дорого і змінюватись залежно від розміру та сфери діяльності органі-

зації, вони виявилися дуже успішними у захисті мережі та інформації в медичних інформаційних системах. Існує кілька різних форм брандмауерів, які можуть бути впроваджені як внутрішньо, так і зовні, щоб захистити медичну організацію від будь-якого різноманіття загроз інформації, яку зберігає медична система.

Першим типом брандмауера, який використовується організаціями є брандмауер фільтрації пакетів. Брандмауер фільтрації пакетів організації фільтрує внутрішні електронні канали та запобігає проникненню у мережу організації зовні[14 , 15]. Це можна порівняти з тим, коли організація обмежує доступ до певних IP адрес. Брандмауер фільтрації пакетів вважається статичним та базовим брандмауером, який слід застосовувати для захисту безпеки електронних медичних записів.

Друга категорія брандмауерів - це брандмауери перевірки стану. Хоча ця форма брандмауерів подібна до брандмауерів фільтрації пакетів, вони відрізняються тим, що брандмауери перевірки стану набагато динамічніші в тому сенсі, що вони можуть перевірити та встановити кореляцію вхідних електронних каналів із раніше відфільтрованими електронними каналами [14]. Брандмауери перевірки стану є більш складними, ніж попередня категорія брандмауерів, і їх слід застосовувати в організаціях, які бажають бачити складну кореляцію зв’язків внутрішньої та зовнішньої IP-адрес. Цей тип системи вимагає часу і може коштувати дорого, що, можливо, не найкраще підходить для всіх організацій охорони здоров’я, які прагнуть захисту своєї інформації.

Третя категорія брандмауерів - це шлюз рівня додатків. Цей тип брандмауера виступає воротарем мережі організації при скануванні веб-сторінки IP на наявність загроз до пересилання сторінки кінцевому користувачеві. У цьому типі брандмауера доступ до зовнішніх мережевих з'єднань здійснюється через шлюз, щоб запобігти зовнішньому проникненню у внутрішню мережу організації [14]. Шлюзи рівня додатків мали успіх у забезпеченні захисту електроних медичних записів, оскільки хакери не можуть безпосеред-

ньо увійти в систему, щоб отримати захищену інформацію про здоров'я. Ця категорія брандмауерів, як правило, є складною і дорогою для організації; отже, необхідно провести повний внутрішній та зовнішній аналіз організації, щоб визначити придатність та життєздатність брандмауера для кожного конкретного відділу, а також організації в цілому.

Остання категорія брандмауерів - це перекладач мережевих адрес (ПМА). Основною функцією ПМА є приховування IP-адреси організації в інтрамережі від хакерів або зовнішніх користувачів, які прагнуть отримати доступ до реальної IP-адреси інтрамережі [14]. Цей тип брандмауера створює бар’єр між внутрішньомережевими організаціями та локальною мережею. Незважаючи на те, що перекладачі мережевих адрес можуть бути дорогими та складними, вони дуже ефективні у захисті медичної інформації в межах електроних медичних записів. Хоча самі брандмауери вважаються важливими для безпеки електроних медичних записів, також життєво важливо, щоб під час впровадження дотримувались чотири фази стратегій безпеки брандмауера:

1) управління послугами;
2) управління напрямком;
3) управління користувачем;
4) контроль поведінки. [16]

Загалом для організації важливо виконати повну оцінку потреб, оцінку бюджету та оцінку загроз, як внутрішніх, так і зовнішніх для організації, перед тим, як застосовувати будь-який тип брандмауера. Якщо організація цього не робить або не виконує чотири етапи стратегії безпеки, це може нанести шкоду безпеці електронних медичних карт пацієнта та інформаційній системі організації в цілому [9, 10, 11, 12, 13].

2.3.2 Застосування криптографії для захисту медичних інформаційних систем

Застосування криптографії також забезпечило безпеку охоронюваної медичної інформації в електронних системах медичних карт. Зокрема, шифрування покращило безпеку електронних медичних карт під час обміну медичною інформацією. Процес обміну інформацією про стан здоров'я має набір специфікацій, передбачених значущими критеріями використання, який вимагає, щоб процес обміну реєструвався організаціями, коли шифрування вмикається або забороняється [17, 18, 26].

Одним із криптографічних методів для захисту даних електронних медичних карт є використання дешифрування [16]. Наприклад, розшифровка забезпечує безпеку електронних медичних карт при перегляді пацієнтами. Цифрові підписи - це рішення для запобігання зміні даних в медичних інформаційних систмах, коли пацієнти переглядають особисту інформацію. Цей метод зарекомендував себе як запобіжний захід порушень безпеки [19, 20]. Методи шифрування та дешифрування також є успішними, коли використовуються для захисту медичної інформації, до якого здійснюється доступ через мобільні агенти. Забезпечуючи мобільні агенти для передачі пацієнтами між установами, електронні медичні картки не тільки більш безпечні, але й більш доступні [21, 26].

Іншою формою криптографії є використання імен користувачів та паролів. Використання імен користувачів та паролів може врешті-решт запобігти порушенням безпеки, просто включаючи особисту конфіденційність щодо паролів та вимагаючи від користувачів часто змінювати особисті паролі [12, 22, 23]. Пароль не повинен містити значущих імен або дат для особи, намагаючись уникнути ймовірності того, що хакер міг би спекулювати паролем. Використання імен користувачів та паролів також є корисним методом безпеки для провайдерів при встановленні контролю доступу на основі ролей. Рольові засоби керування доступом обмежують інформацію для користувачів на основі облікових даних імені користувача та пароля, призначених

системним адміністратором. Цей прийом захисту захищає інформацію в медичних інформаційних системах від внутрішніх порушень або загроз [24]. Також важливо, щоб працівник пам’ятав виходити з системи після кожного використання, щоб уникнути залишення медичної інформації видимою для неавторизованого персоналу [12, 26].

2.3.3 Додаткові методи захисту медичних інформаційних систем

На додаток до брандмауерів та криптографії, серед інших відомих методів безпеки є хмарні обчислення, антивірусне програмне забезпечення, програми початкової оцінки ризиків та ідентифікація радіочастот. З розвитком технологій хмарні обчислення стають все більш досліджуваними для спрощення та інтеграції в системи EHR. Інфраструктури, які створюють хмарні обчислення, дозволяють електронно передавати та обмінюватися інформацією через "оренду" хмарного сховища, програмного забезпечення та обчислювальних потужностей. Орендуючи хмарне сховище організації охорони здоров’я можуть скоротити витрати на впровадження медичних інформаційних систем за рахунок зміни власника системи та її коштовного обслуговування обслуговування, одночасно інтегруючи криптографічні методи для забезпечення безпечного доступу до хмари [25]. Хоча хмарні обчислення представляють перспективну платформу, антивірусне програмне забезпечення залишається постійно використовуваним захисним заходом безпеки.

Контроль доступу (технічна гарантія) - це техніка, яка перешкоджає або обмежує доступ до електронного ресурсу. Метою контролю доступу є обмеження доступу лише уповноваженими сторонами. Заклад охорони здоров’я збирає, обробляє та зберігає такі дані пацієнтів, які є дуже чутливими. Цей захист може мати форму контролю за доступом на основі ролей, контролю доступу на основі атрибутів та ідентифікації користувача. Рольовий стосується ролі людини в медичному закладі. Наприклад, коли людина починає працювати у закладі охорони здоров’я, він / вона має доступ до даних пацієнта, але лише даних своїх пацієнтів. Якщо цей постачальник

також працює в певному комітеті в лікарні, тоді створюється інший набір привілеїв, щоб забезпечити доступ до ресурсів комітету. При доступі до інших даних створюється журнал, який періодично перевіряється. Коли адміністратор на стійці реєстрації починає працювати у закладі, він / вона не має підстав отримувати доступ до клінічних даних, але йому може знадобитися доступ до адміністративних даних, таких як адреса та номер телефону, залежно від ролі, яку людина відіграє в організації. Інші назви для цього - засоби управління медіа, аутентифікація сутності, шифрування, брандмауер, аудіозаписи, перевірка вірусів та фільтрація пакетів.

Адміністративні гарантії - це захисні заходи, які як правило, мають форму політики, практики та процедур в установі для регулярної перевірки на наявність уразливих місць та постійного поліпшення стану безпеки організації. Інші назви цього контролю - це аналіз та управління ризиками, оцінка безпеки системи, персонал, вибраний на певні ролі, непередбачені ситуації, безперервність бізнесу та планування відновлення після катастрофи[26].

3 ПРОЕКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ЕНДОГАСТРОЕТЕРОЛОГІЧНОГО ВІДДІЛЕННЯ ЛІКАРНІ

3.1 Принцип роботи відділення лікарні

Коли пацієнт потрапляє до відділення, в нього вже є картка стаціонарного хворого, яку йому склали у приймальному відділенні. Ця картка має свій особистий номер. Всі подальші направлення на обстеження, які буде проходити пацієнт, будуть мати цей номер. У цій картці позначено: його прізвище, ім’я та по батькові, вік, стать, перенесенні інфекційні захворювання, прізвище, ім’я та по батькові його батьків та їх місце роботи, його зріст та вігу. Під час оформлення пацієнта до відділення медсестра пояснює йому та його батькам: правила перебування у відділенні, які продукти можна тримати у відділенні, розклад відділення.

Після цього пацієнт дізнається номер палати, в якій буде знаходитись. Потім його оглядає лікар. За результатами огляду пацієнту призначаються різні обстеження та аналізи, консультації вузьких спеціалістів, номер дієтичного стола, також лікар вже може призначити йому первинне чи невідкладне лікування.

Після того, як будуть готові результати всіх аналізів та обстежень, лікар призначає пацієнту лікування, медичні та фізіотерапевтичні процедури, а також консультації вузьких спеціалістів, які, в свою чергу, також можуть робити призначення пацієнту.

3.2 Вибір засобів реалізації

Для реалізації інформаційної системи ендогастроентерлогічного відділення використано бази даних MySQl, а також мову програмування PHP.

База даних (БД) – іменована сукупність даних, яка відображає стан об'єктів та їх відносин у даній предметної області, або інакше БД – це сукуп-

ність взаємопов'язаних даних при такій мінімальній надмірності, яка допускає їх використання оптимальним чином для одного або декількох додатків в певній предметній області. БД складається з безлічі пов'язаних файлів.

Система управління базами даних (СУБД) – сукупність мовних і програмних засобів, призначених для створення, ведення і спільного використання БД багатьма користувачами.

Автоматизована інформаційна система (АІС) – це система, що реалізує автоматизований збір, обробку, маніпулювання даними, що функціонує на основі ЕОМ та інших технічних засобів і включає відповідне програмне забезпечення (ПО) і персонал. Надалі в цій якості буде використовуватися термін інформаційна система (ІС), який має на увазі поняття автоматизована[14].

У цьому відмінність баз даних від звичайних комп'ютерних файлів. СУБД пов'язує користувачів і фізичне подання даних (тобто те, як дані реально зберігаються). Всі призначені для користувача запити, незалежно від того, від людини або комп'ютерної програми вони виходять, обробляються СУБД. Головна функція СУБД – приховування програмного коду від користувачів БД. З іншого боку, СУБД дозволяє відображати дані на більш високому, ніж програмне забезпечення, рівні, наприклад, призначений для користувача запит «Отримати дані про пацієнта Коваленко» може бути написаний на мові більш високого рівня.

СУБД також встановлюють кількість і інформації, я

ка може бути доступна конкретному користувачеві. Наприклад, лікаря і медсестрі лікарні потрібні різні уявлення бази даних. Коли користувач бажає отримати доступ до бази даних, він робить запит, використовуючи спеціальну мову для маніпуляції даними, який розуміється СУБД. СУБД отримує запит і перевіряє його на синтаксичні помилки.

Далі СУБД вивчає зовнішню і абстрактну будову, а також взаємні відображення між зовнішньою і внутрішньою схемою. Потім СУБД виконує необхідні операції з збереженої базою даних. У загальному випадку, поля запи-

су можуть використовуватися в різних таблицях, що містяться в базах даних. Кожний віртуальний запис може, наприклад, містити дані з різних фізичних записів реальної БД. СУБД повинен відшукувати кожну із запитаних записів і складати з неї таблицю, що відповідає запитам користувача. В цьому випадку користувачам не потрібно знати фізичну будову БД, яке може змінюватися. При цьому їх віртуальне уявлення мінятися не буде[15].

3.2.1 Мова програмування PHP та СУБД MySQL

PHP (англ. Hypertext Preprocessor) - мова програмування, призначена для створення HTML-сторінок з веб-сервера.

PHP - одна з найпоширеніших мов, що використовуються у веб-розробці поряд з Java. NET, Perl, Python та Ruby.

Веб-сервер інтерпретує PHP-код у HTML-код і передає його клієнту. Ви можете вставити PHP-код безпосередньо в HTML-код. На відміну від JavaScript, користувач не бачить PHP-код, оскільки браузер клієнта отримує готовий HTML-код.

Давайте проаналізуємо, що відбувається, коли користувач входить на сервер. Він пише адресу сторінки або посилання на адресний рядок браузера, тобто створює запит. Що робить сервер? Зазвичай викликають HTML-сторінку - текстовий файл. Браузер відправляє запит на сервер, отримує код сторінки, крім того завантажує файли, зареєстровані в коді сторінки і необхідні для його оформлення та відображення для належного функціонування, і відображає результат у вікні. Такі сторінки є статичними.

Рисунок *3.1* Принцип роботи статичного сервера

Тепер розглянемо випадок, коли ви викликаєте сторінку, написану на PHP. У цьому випадку трапляється зовсім інше. При зверненні до сервера Web машина обробляє дані, отримані з файлу *.php, після чого сервер відправляє результат опрацювання цих даних у браузер користувача. Такі сторінки називають динамічними [35].

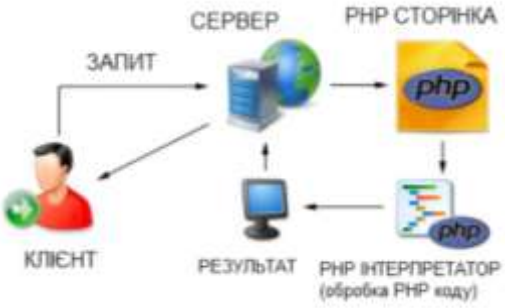

Рисунок *3.2* Принцип роботи сервера, коли сторінка написана на мові PHP

MySQL - це система управління реляційними базами даних. У реляційній базі даних дані зберігаються в окремих таблицях, завдяки чому досягається виграш у швидкості й гнучкості. Таблиці зв'язуються між собою за допомогою відносин, завдяки чому забезпечується можливість поєднувати при виконанні запиту дані з декількох таблиць. SQL як частина системи MySQL можна охарактеризувати як мова структурованих запитів, що використовується для доступу до баз даних.

MySQL - це ПЗ з відкритим кодом. Застосовувати його і модифікувати може будь-хто. Таке ПЗ можна отримувати за допомогою Internet і використовувати безкоштовно. При цьому кожен користувач може вивчити вихідний код і змінити його у відповідності зі своїми потребами.

MySQL складається з двох частин:

а) серверної;

б) клієнтської.

Сервер MySQL постійно працює на комп'ютері. Клієнтські програми (наприклад, скрипти PHP) посилають серверу MySQL SQL-запити через механізм сокетів (тобто за допомогою мережевих засобів), сервер їх обробляє і запам'ятовує результат. Тобто скрипт (клієнт) вказує, яку інформацію він хоче отримати від сервера баз даних. Потім сервер баз даних посилає відповідь (результат) клієнтові (скрипту).

Структура MySQL трирівнева: бази даних - таблиці - записи. Бази даних і таблиці MySQL фізично представляються файлами з розширеннями frm, MYD, MYI. Логічно таблиця являє собою сукупність записів. А запису - це сукупність полів різного типу. Ім'я бази даних MySQL унікально в межах системи, а таблиці - в межах бази даних, поля - в межах таблиці. Один сервер MySQL може підтримувати одразу декілька баз даних, доступ до яких може розмежовуватись логіном і паролем.

База даних з точки зору MySQL - це звичайний каталог, що містить файли певного формату - таблиці. Таблиці складаються із записів, а записи, у свою чергу, складаються з полів. Поле має два атрибути - ім'я і тип.

Тип поля може бути:

а) ціле число;

б) дійсним;

в) рядок;

г) бінарний;

д) дата і час;

е) перерахування;

ж) множини [36].

Технології PHP MySQL визнані найбільш надійними у всій світовій практиці. PHP є швидкісний, і, разом з тим, дуже простий крос-платформний мова програмування, найбільш часто вживаний для сайтобудування, і тому можливість техпідтримки хостингу, зробленого на ньому, дуже важлива.

PHP хостинги надають своїм користувачам чимало переваг перед іншими аналогами, які у багато разів підвищуються завдяки підтримки MySQL. Вони стають повноцінними управлінськими системами баз даних, високонадійними і одночасно простими для використання користувачами серверів, чим обумовлена їх велика популярність.

Подібний хостинг гарантує швидкісний пошук незалежно від обсягу записів або кількості даних, які доводиться обробляти. Це друга важлива відмінність від хостингу такого типу. Крім того, технологічні розробки MySQL і PHP відмінно взаємодіють фактично з кожної операційною системою, їх легко встановити і налаштувати, щоб ефективно і успішно користуватися наданими ними широкими можливостями.

Цікавим нюансом є також і те, що подібні хостинги, встановлені на операционках UNIX і Linux, набагато дешевші і стабільні, ніж працюють з Windows платні аналоги. Саме тому більшість сучасних користувачів вибирають саме хостинги, які функціонують з PHP MySQL, внаслідок високого рівня їх надійності та простоти[16].

3.3 Особливості інформаційної системи ендогастроентерлогічне відділення

Ендогастроентерлогічне відділення спеціалізується на діагностуванні та лікуванні пацієнтів з ендокринними «цукровий діабет, ожиріння, затримка зросту, захворювання, щитовидної залози, тощо» і гастроентерологічні захворювання «хронічний гастродоуденіт, виразкова хвороба шлунку, дванадцятиперстної кішки, коліти».

Інформаційна система направлена на користування:

– докторам відділення, які можуть переглядати дані про пацієнтів та робити їм призначення;

– медсестрам відділення. Які зможуть переглядати призначення лікарів, додавати нових пацієнтів, видати тих кого виписали, а також додавати інформацію про медикаменти.

З точки зору лікарів – це сайт, який надає можливість отримати необхідну інформацію, а також додати нову інформацію.

З точки зору медсестер – це інформаційна система, яка надає можливість зручно вносити дані, поновлювати та видаляти їх, а також надійно зберігати.

Сфера використання інформаційної системи – сайт, який працює на будь-яку Обласну дитячу лікарню.

З цього веб-сайту користувачі зможуть отримати таку інформацію, як:

а) ім’я пацієнта;

б) його вік;

в) діагноз вік;

г) його місце проживання;

д) хто його лікар;

е) коли його поклали до лікарні;

ж) коли його виписали з лікарні;

з) аналізи які йому робили або зроблять;

и) коли були отриманні результати тих аналізв;

к) ін’єкції призначенні пацієнту;

л) коли йому їх зробили;

м) доза ін’єкцій;

н) які вузькі спеціалісти обстежували або будуть обстежувати пацієнта;

о) які фізіотерапевтичні процедури призначенні пацієнту;

п) які ультразвукові, рентгенологічні та функціональні обстеження були призначенні пацієнту;

р) коли прийшли результати цих обстежень;

с) якому дієтстолу належить пацієнт;

т) якими бюджетними медикаментами він користується.

Вхідні дані: описуються, наприклад, при вступі нового пацієнта, що необхідно занести до бази даних. Вхідні дані містять наступну інформацію: код_пацієнта, код_відділення, в якому лікується пацієнт, його прізвище ім’я по батькові, його діагноз, його місце проживання, з селі він або з міста, коли він потрапив до відділення, коли пацієнта з нього виписали, його стать, дата народження та ім’я його лікаря.

Випишемо усі елементи даних, що використовуються або створюються.

Основні операції у даній предметній області можуть бути наступними:

- операція 1 Занесення нового пацієнта до бази даних(таблиця 3.1);
- операція 2 Занесення нового аналіза до бази даних (таблиця 3.2);
- операція 3 Занесення нової ін’єкції до бази даних (таблиця 3.3).
-

Таблиця *3.1*

Характеристики для Операції 1

Назва операції	Використовувані об'єкти	Скільки екземплярів бере участь	Тип доступу (R-читання, W- запис)
ерація 1	Пацієнт	1	W
	Діагноз	1	W
	Лікар	1	W

Таблиця *3.2*

Характеристики для Операції 2

Назва операції	Використовувані об'єкти	Скільки екземплярів бере участь(M – будь-яка кількість)	Тип доступу (R- читання, W- запис)
Операція 2	Пацієнт	1	R
	Аналіз	M	W

Таблиця *3.2*

Характеристики для Операції 3

Назва операції	Використовувані об'єкти	Скільки екземплярів бере участь	Тип доступу (R- читання, W- запис)
Операція 3	Пацієнт	1	W
	Ін’єкція	M	R

Перетворення інформаційних вимог у форму, зручну для проведення аналізу.

Елементи даних, що використовуються або створюються (таблиця 3.4).

Таблиця *3.3*

Визначення використовуваних елементів даних.

№	Найменування	Визначення
1	Номер пацієнта відділен-ня(код_пацієнта)	Однозначно визначає кожного пацієнта
2	Номер відділення, де знаходиться паці-єнт(код_відділення)	Визначає відділення, де лікують пацієнта
3	ПІП	Прізвище ім’я та по батькові пацієнта
4	Діагноз	Діагноз за яким пацієнт знаходиться у відділенні
5	Місце_прживання	Місце, де приживає Пацієнт
6	Да-та_початку_лікування	Дата, коли пацієнт ліг до лікарні
7	Дата_виписки	Дата, коли пацієнта виписали зі лікарні
8	Номер_палати	Палата, в якій перебуває пацієнт
9	ПІП_лікаря	Прізвище ім’я та по батькові лікаря, який лікує пацієнта
10	Дата_народження	Дата, коли пацієнт народився
11	Стать	Інформація про те, якої статі пацієнт
12	Дата_призначення	Залежно від таблиці; дата, коли пацієнту при-значили аналіз, ін’єкцію, обстеження, кон-сультацію, медикамент або процедуру
13	Дата_відміни	Залежно від таблиці; дата, коли пацієнту відмінили ін’єкцію, консультацію, медика-мент або процедуру
14	Дата_результату	Дата, коли був отримання результату аналізу

Продовження таблиці 3.4

15	Вік	Скільки пацієнту повних років
16	Номер_дієтстола	Номер дієтстола, призначеного пацієнту згідно з його діагнозу та за яким він має харчуватися у відділенні
17	Тип_ін’єкції	Яку саме ін’єкцію треба зробити пацієнту
18	Доза	Залежно від таблиці; яку саме дозу ін’єкції або препарату потрібно надати пацієнту
19	Спеціаліст	ПІП вузького спеціаліста, який буде оглядати хворого
20	Час_прийому	Коли пацієнт має приймати медикаменти
21	Кількість_прийому_ на_добу	Скільки саме медикаменту потрібно прийняти пацієнту
22	Реєстрація виконання	Залежно від таблиці; виконано, чи ні процедура, ін’єкція або обстеження
23	Назва_обстеження	Яке саме обстеження повинно зробити пацієнту

3.4 Концептуальне проектування предметної області. Побудова ER–діаграми

На рисунку 3.3 зображена ER – діаграма предметної області.

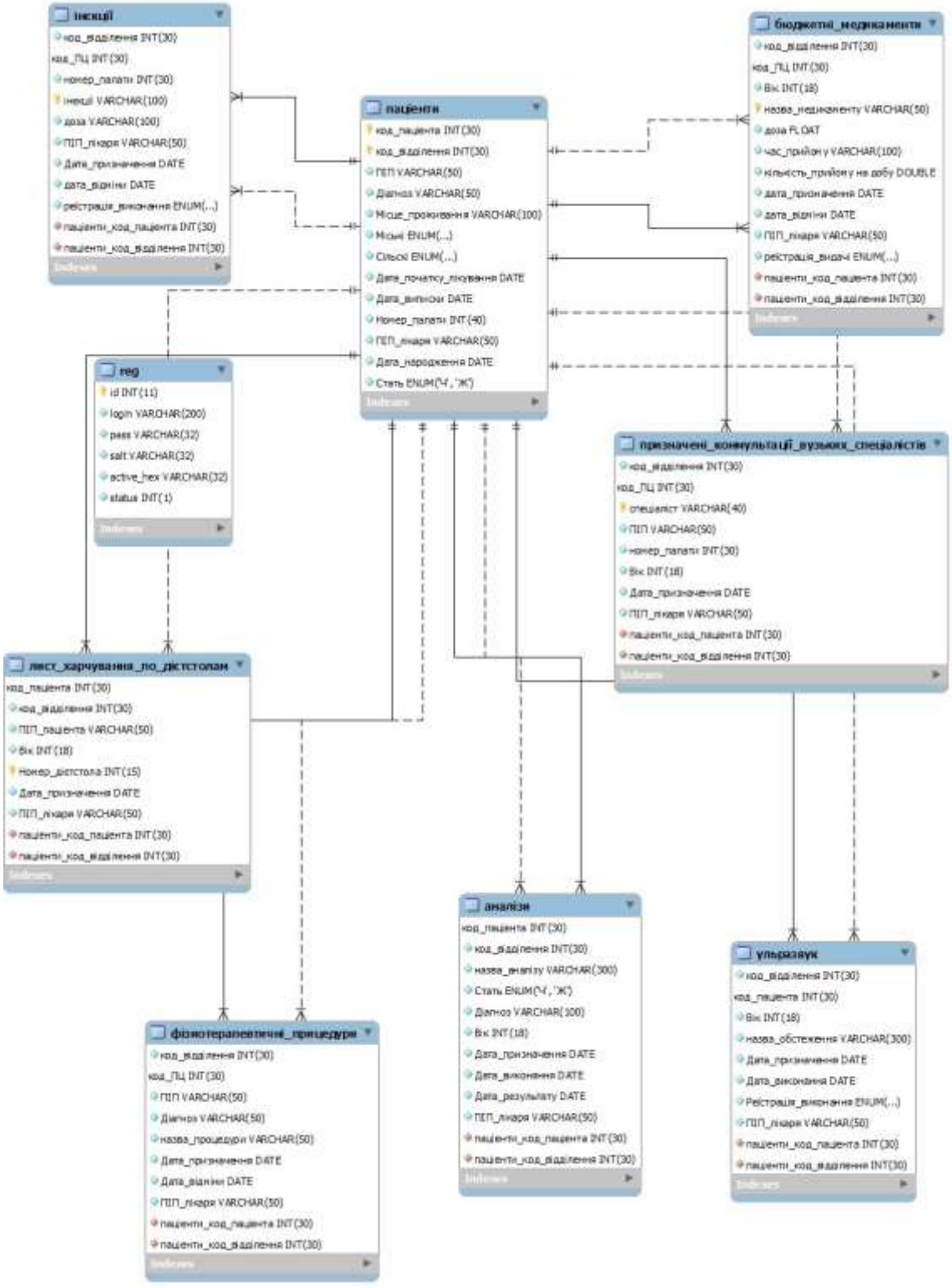

Рисунок 3.3 ER-діаграма предметної області

Пацієнти (код_пацієнта, код_відділення, ПІП, Діагноз, Місце_проживання, Міські, Сільські, Дата_початку_лікування, Дата_виписки, номер_палати, ПІП_лікаря, дата_народження, стать).

Аналізи (код_пацієнта, код_відділення, назва_аналізу, стать, діагноз, вік, дата_призначення, дата_виконання, дата_результату, ПІП_лікаря).

Лист_харчування_по_дієтстолам (код_відділення, код_пацієнта, Вік, номер_дієтстола, Дата_призначення, ПІП_лікаря).

Ін'єкції (код_відділення, код_пацієнта, номер_палати, ін'єкції, доза, дата_призначення, дата_відміни, ПІП_лікаря).

Призначення_консультацій_візьких_спеціалістів (код_відділення, код_пацієнта, спеціаліст, ПІП, номер_палати, вік, дата_призначення, ПІП_лікаря).

Процедури_фізиотерапевтичні (код_відділення, код_пацієнта, номер_палати, ПІП, Вік, діагноз, дата_призначення, дата_відміни, назва_процедури, риєстрація_виконання, ПІП_лікаря).

Бюджетні_медикаменти(код_відділення, код_пацієнта, вік, назва_медикаменту, доза, час_прийому, кількість_прийому_на_добу, дата_призначення, дата_відміни, ПІП_лікаря, реїстрація_виконання).

Ультразвукові,_рентгенологічні_та_функціональні_обстеження(код_відділення, код_пацієнта, назва_медикаменту, доза, час_прийому, кількість_прийому_на_добу, дата_призначення, дата_відміни, ПІП_лікаря, реїстрація_виконання).

Створення таблиць:

Таблиця **пацієнти** *(Кількість записів:30*

```
CREATE TABLE `пацієнти` (
  `код_пацієнта` int(30) NOT NULL,
  `код_відділення` int(30) NOT NULL,
  `ПІП` varchar(50) NOT NULL,
  `Діагноз` varchar(50) NOT NULL,
```

```
  `Місце_проживання` varchar(100) NOT NULL,
  `Міські` enum('Так','Ні') NOT NULL,
  `Сільські` enum('Так','Ні') NOT NULL,
  `Дата_початку_лікування` date NOT NULL,
  `Дата_виписки` date NOT NULL,
  `Номер_палати` int(40) NOT NULL,
  `ПІП_лікаря` varchar(50) NOT NULL,
  `Дата_народження` date NOT NULL,
  `Стать` enum('Ч','Ж') NOT NULL,
  PRIMARY KEY (`код_пацієнта`,`код_відділення`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Вигляд таблиці пацієнти представлено на рисунку 3.4

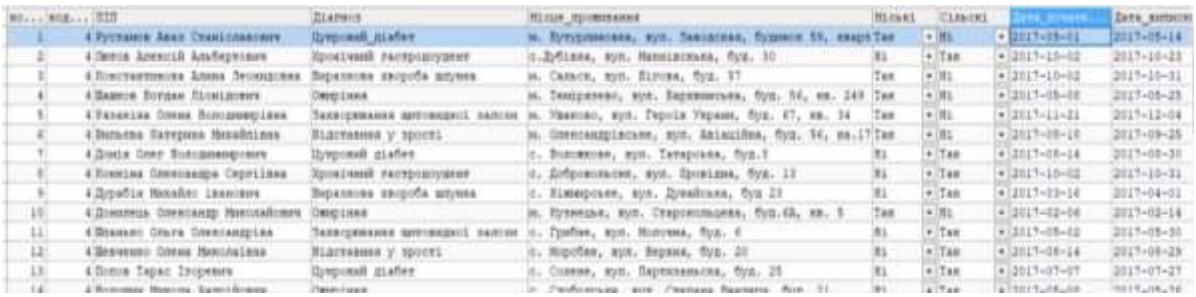

Рисунок *3.4* Таблиця пацієнти

Таблиця **аналізи** (Кількість записів: 30)

```
CREATE TABLE `аналізи` (
  `код_пацієнта` int(30) NOT NULL,
  `код_відділення` int(30) NOT NULL,
`назва_аналізу` varchar(300) NOT NULL,
`Стать` enum('Ч','Ж') NOT NULL,
  `Діагноз` varchar(100) NOT NULL,
  `Вік` int(18) NOT NULL,
  `Дата_призначення` date NOT NULL,
  `Дата_виконяння` date NOT NULL,
```

`Дата_результату` date NOT NULL,

`ПІП_лікаря` varchar(50) NOT NULL,

PRIMARY KEY (`код_пацієнта`),

CONSTRAINT `аналізи_ibfk_1` FOREIGN KEY (`код_пацієнта`) REFERENCES `пацієнти` (`код_пацієнта`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

Вигляд таблиці аналізи представлено на рисунку 3.5

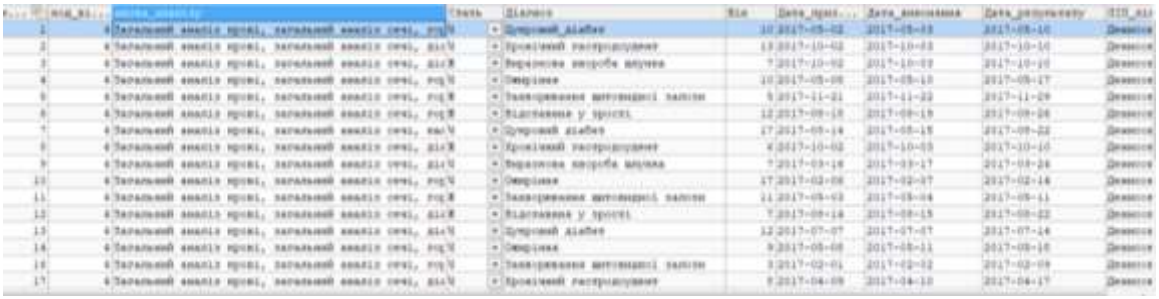

Рисунок *3.5* Таблиця аналізи

Таблиця **призначення консультацій вузьких спеціалістів** *(Кількість записів: 30)*

CREATE TABLE `призначені_кон сультації_вузьких_спеціалістів` (

`код_відділення` int(30) NOT NULL,

`код_ПЦ` int(30) NOT NULL,

`спеціаліст` varchar(40) NOT NULL,

`ПІП` varchar(50) NOT NULL,

`номер_палати` int(30) NOT NULL,

`Вік` int(18) NOT NULL,

`Дата_призначення` date NOT NULL,

`ПІП_лікаря` varchar(50) NOT NULL,

PRIMARY KEY (`код_ПЦ`,`спеціаліст`),

CONSTRAINT`призначені_конмультації_вузьких_спеціалістів_ibfk_1` FOREIGN KEY (`код_ПЦ`) REFERENCES `пацієнти` (`код_пацієнта`)) ENGINE=InnoDB DEFAULT CHARSET=utf8;

Вигляд таблиці призначені консультації вузьких спеціалістів представлено на рисунку 3.6

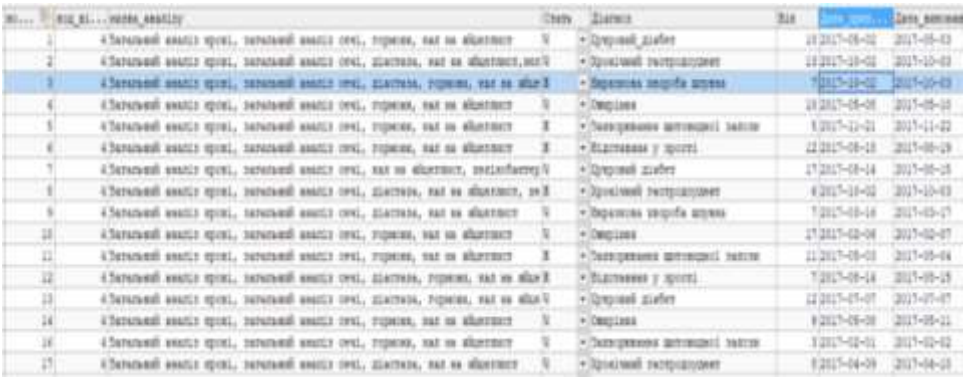

Рисунок *3.6* Таблиця призначені консультації вузьких спеціалістів

Таблиця **фізиотерапевтичні прицедури** *(Кількість записів:30)*

CREATE TABLE `фізиотерапевтичні_прицедури` (

`код_відділення` int(30) NOT NULL,

`код_ПЦ` int(30) NOT NULL,

`ПІП` varchar(50) NOT NULL,

`Діагноз` varchar(50) NOT NULL,

`назва_процедури` varchar(50) NOT NULL,

`Дата_призначення` date NOT NULL,

`Дата_відміни` date NOT NULL,

`ПІП_лікаря` varchar(50) NOT NULL,

PRIMARY KEY (`код_ПЦ`),

CONSTRAINT `процедури_фізиотерапевтичні_ibfk_1` FOREIGN KEY (`код_ПЦ`) REFERENCES `пацієнти` (`код_пацієнта`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

Вигляд таблиці фізіотерапевтичні процедури представлено на рисунку 3.7

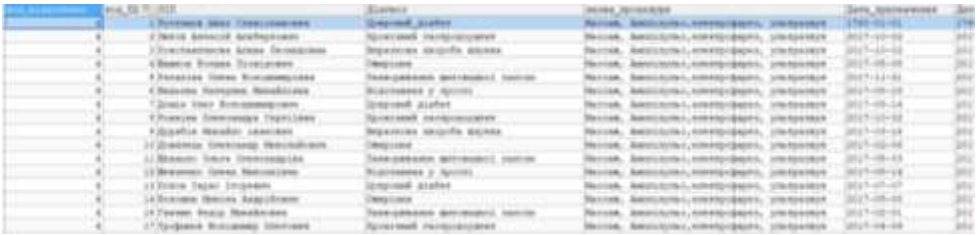

Рисунок *3.7* Таблиця фізиотерапевтичні процедури

Таблиця **ін'єкції** (Кількість записів:7)

CREATE TABLE `інєкції` (

`код_відділення` int(30) NOT NULL,

`код_ПЦ` int(30) NOT NULL,

`номер_палати` int(30) NOT NULL,

`інєкції` varchar(100) NOT NULL,

`доза` varchar(100) NOT NULL,

`ПІП_лікаря` varchar(50) NOT NULL,

`Дата_призначення` date NOT NULL,

`дата_відміни` date NOT NULL,

`реїстрація_виконання` enum('Виконано','Не_Виконано') NOT NULL,

PRIMARY KEY (`код_ПЦ`,`інєкції`),

CONSTRAINT `інєкції_ibfk_1` FOREIGN KEY (`код_ПЦ`) REFERENCES `пацієнти` (`код_пацієнта`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

Вигляд таблиці ін'єкції представлено на рисунку 3.8

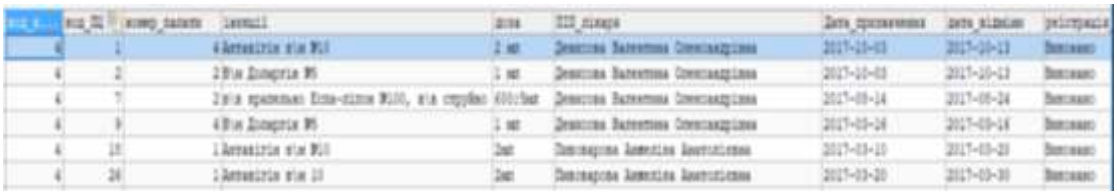

Рисунок *3.8* Таблиця ін'єкції

Таблиця **ульразвукові, регінологічні та функціональні обстеження`** (Кількість записів:30)

```
CREATE TABLE `ульразвук` (
 `код_відділення` int(30) NOT NULL,
  `код_пацієнта` int(30) NOT NULL,
  `Вік` int(18) NOT NULL,
`назва_обстеження` varchar(300) NOT NULL, `
Дата_призначення` date NOT NULL,
 `Дата_виконання` date NOT NULL,
  `Реїстрація_виконання` enum('Виконано','Не_Виконано') NOT NULL,
  `ПІП_лікаря` varchar(50) NOT NULL,
 PRIMARY KEY (`код_пацієнта`), CONSTRAINT `ульразвук_ibfk_1` FOREIGN KEY (`код_пацієнта`) REFERENCES `пацієнти` (`код_пацієнта`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Вигляд таблиці ульразвукові, регінологічні та функціональні обстеження представлено на рисунку 3.9

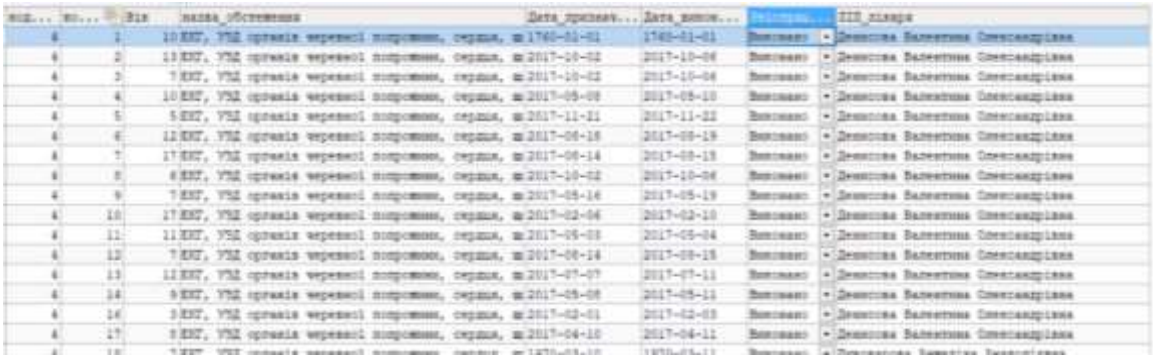

Рисунок *3.9* Таблиця ульразвукові, регінологічні та функціональні обстеження

Таблиця **лист харчування по дієтстолам** (*Кількість записів:30*)

```
CREATE TABLE `лист_харчування_по_дієтстолам` (
  `код_пацієнта` int(30) NOT NULL,
  `код_відділення` int(30) NOT NULL,
```

`ПІП_пацієнта` varchar(50) NOT NULL,

`Вік` int(18) NOT NULL,

`Номер_дієтстола` int(15) NOT NULL,

`Дата_призначення` date NOT NULL,

`ПІП_лікаря` varchar(50) NOT NULL,

PRIMARY KEY (`код_пацієнта`,`Номер_дієтстола`),

CONSTRAINT `лист_харчування_по_дієтстолам_ibfk_1` FOREIGN KEY (`код_пацієнта`) REFERENCES `пацієнти` (`код_пацієнта`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

Вигляд таблиці лист харчування по дієт-столам представлено на рисунку 3.10.

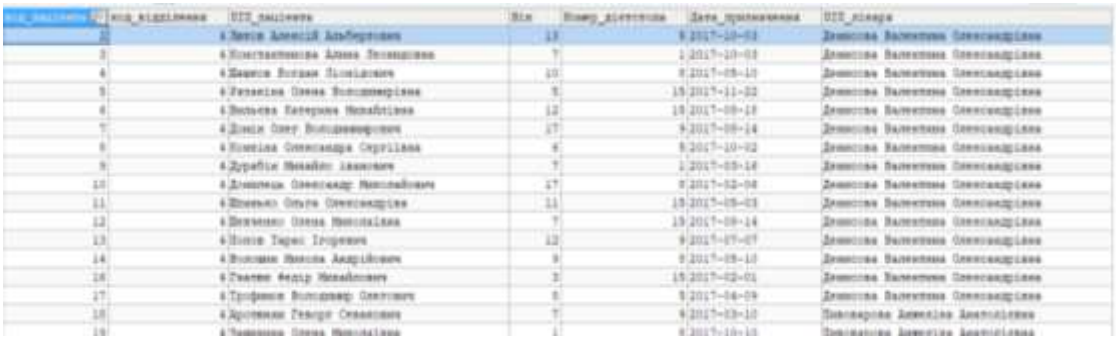

Рисунок *3.10* Таблиця лист харчування по дієтстолам

Таблиця **бюджетні медикаменти**(Кількість записів:30)

CREATE TABLE `бюджетні_медикаменті` (

`код_відділення` int(30) NOT NULL,

`код_ПЦ` int(30) NOT NULL,

`Вік` int(18) NOT NULL,

`Назва_медикаменту` varchar(50) NOT NULL,

`Доза` float NOT NULL,

`Час_прийому` varchar(100) NOT NULL,

`Кількість_прийому_на_добу` double NOT NULL,

`Дата_призначення` date NOT NULL,

`Дата_відміни` date NOT NULL,

`ПІП_лкаря` varchar(50) NOT NULL,

`Реїстрація_видачі` enum('Видано','Не_Видано') NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

Вигляд таблиці бюджетні медикаменти представлено на рисунку 3.11

Рисунок *3.11* Таблиця бюджетні медикаменти

3.5 Розробка системи запитів

Запит SQL для вибірки з таблиць виглядає так:

SELECT [предикат] { * | таблиця.* | [таблиця.]поле_1

[AS псевдонім_1] [, [таблиця.]поле_2 [AS псевдонім_2] [,

...]]}

FROM вираз [, ...] [IN зовнішняБазаДаних]

[WHERE...]

[GROUP BY...]

[HAVING...]

[ORDER BY...]

де:

– Предикат –ALL | DISTINCT[ROW] | [TOP n [PERCENT]].

– ALL –означає, що нужно обрати всі рядки навіть втому числі ті, що повторюються.

– DISTINCT[ROW] – обрати тільки різні рядки

– TOP n [PERCENT] – обираємо перших n кількіть записів.

– TOP n PERCENT – обираємо перших n відсотків записів.

– Таблиця – та таблиця, з якой відбираються записи.

– Поле_і — поля, з яких відбираються записи. Якщо ви включите кілька полів, вони будуть вилучені в зазначеному порядку. Поля, відібрані за допомогою окремого запиту, та вирази на основі результатів окремих запитів також можуть використовуватися як поля.

– Псевдонім_і – імена, які замінять вхідні назви стовпчиків і стануть їх заголовками.

– Вираз –таблиці, одна або декілька, чиї дані відбирають або визари з оператором INNER JOIN, RIGHT JOIN, LEFT JOIN

Для кожного SELECT має існувати речення FROM. Те, як будуть слідувати назви таблиць у виразі не має значення.

WHERE вказує, які записи таблиць, імена яких зазначені в FROM мають стати результатом функції SELECT.

GROUP BY групує(об’єднує) в одну групу ті записи полів таблиці, в яких є однакові значення. Якщо в SELECT є статистична функція SQL, тоді буде обчислено підсумкове значення кожної групи.

HAVING визначає, які згруповані записи відображаються при використанні інструкції SELECT із реченням GROUP BY. Після того, як записи будуть організовані за допомогою GROUP IN, оператор HAVING вибере групи записів, які відповідають критеріям відбору, зазначеним у операторі HAVING.

Порядок ORDER BY перелічує записи, отримані в результаті зростання (ASC) або спадання(DESC) запиту на основі значення зазначеного поля або областей.

Запит SQL для додавання одного запису виглядає так::

INSERT INTO призначення [(поле_1[, поле_2[, ...]])]

VALUES (значення_1[, значення_2[, ...])

де:

– призначення – назва таблиці або запиту, до яких додаються записи;

– поле_і – назва поля таблиці;Ви можете вказати лише поля, де введено значення, але обов’язково вказати всі ключові поля;

– значення_і – Значення, яке отримане з допустимих значень для поля_і.

Для додавання кількох записів:

INSERT INTO призначення [(поле_1[, поле_2[, ...]])]

SELECT [джерело.]поле_1[, поле_2[, ...]

FROM вираз

[WHERE…]

де:

– призначення – назва таблиці або запиту, до яких додаються записи;джерело - назва таблиці або запиту, з якого копіюються записи;

– поле_і – Ім'я поля, що використовується для додавання даних (якщо вони слідують за цільовим параметром); якщо дані слідують за параметром джерела, ім'я поля, з якого отримуються дані; вираз - назва однієї або декількох таблиць, в які вставляються дані;

– значення _і – Значення, додане до вказаного поля нового запису. Кожне значення буде вставлено в поле, що займає однакову позицію у списку: значення_1 вставляється в поле_1 нового запису, значення_2 вставляється в поле_2 тощо. Кожне значення текстового поля має бути в лапках ' ';

– для розділення значень використовуйте коми.

Запит SQL для створення таблиці шляхом додавання до неї записів:

SELECT поле_1[, поле_2[, ...]]

INTO новаТаблиця

FROM джерело

[WHERE…]

де:

– Поле_і – назви полів для копіювання в нову таблицю.

– НоваТаблиця – Назва таблиці, яку вони створюють. Ця назва повинна відповідати стандартним правилам іменування. Якщо нова таблиця відповідає імені існуючої таблиці, тоді буде перехоплена помилка

– Джерело – Ім'я існуючої таблиці, в якій відбираються записи. Це може бути: таблиця, запит або кілька таблиць.

Інструкція SQL на видалення записів має такий вигляд:

DELETE [таблиця.*]

FROM таблиця

WHERE умоваВідбору

де:

– таблиця – назва таблиці, з якої видаляються записи;

– умоваВідбору – вираз, який ідентифікує записи, які потрібно видалити.

Інструкція SQL на поновлення записів в таблиці виглядає так:

UPDATE таблиця

SET новеЗначення

WHERE умоваВідбору;

де:

– таблиця – назва таблиці, в якій дані повинні бути змінені;

– новеЗначення – вираз, значення, яке потрібно вставити в поле оновлених записів, яке вказано в запиті.

– умоваВідбору – вираз, що вибирає значення запису, який потрібно змінити.

При виконанні цієї інструкції буде змінені тільки записи, що відповідають вказаній умові[7].

Показати номер палати тих пацієнтів, чиїм лікарем є Денисова Валентина Олександрівна:

SELECT `аналізи`.`код_пацієнта`,`пацієнти`.`Номер_палати`

FROM `аналізи`, `пацієнти`

WHERE `аналізи`.`код_пацієнта`=`пацієнти`.`код_пацієнта`

AND `пацієнти`.`ПІП_лікаря`= 'Денисова Валентина Олександрівна';

Результат запиту можна побачити на рисунку 3.12.

код_паці...	Номер_палати
1	4
2	2
3	2
4	10
5	3
6	7
7	2
8	6
9	4
10	4
11	6
12	1
13	3
14	7
16	9
17	8

Рисунок *3.12* Запит на вибір даних №1

Додати запис до таблиці пацієнти

INSERT INTO `пацієнти`(`код_пацієнта`,`код_відділення`,`ПІП`, `Діагноз`,`Місце_проживання`,`Дата_початку_лікування`,`Дата_виписки`,`Номер_палати`,`ПІП_лікаря`,`Дата_народження`,`Стать`)

VALUES (31,4,'Зозуля Тетяна Василівна','Відставання_у_зрості','м. Бутурлиновка, вул. Заводская, будинок 59, квартира29','2017-05-01','2017-05-14',4,'Денисова Валентина Олександрівна','2007-07-12','Ж');Результат запиту можна побачити на рисунку 3.13

Рисунок *3.13* Запит на введення даних №2

Показати код тих пацієнтів, у яких діагноз «Ожиріння» або «Відставання у зрості»

```
SELECT `код_пацієнта`
FROM `пацієнти`
WHERE `Діагноз` = 'Ожиріння'
OR `Діагноз` = 'Відставння у зрості';
```

Результат запиту можна побачити на рисунку 3.14

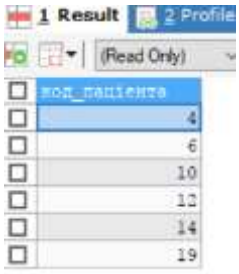

Рисунок *3.14* Запит на введення даних №3

3.6 Шифрування сховища бази даних

Ви можете встановити з'єднання через SSL для шифрування зв'язку клієнт / сервер для підвищення безпеки, або ви можете використовувати ssh для шифрування мережевого з'єднання між клієнтами і сервером бази даних. Якщо будь-який з них використовується, то моніторинг вашого трафіку і отримання інформації про вашу бази даних буде важким для потенційного зловмисника.

У разі, якщо зломщик отримав безпосередній доступ до БД (в обхід веб-сервера), він може отримати дані, що цікавлять або порушити їх цілісність, якщо інформація не захищена на рівні самої БД. Шифрування даних - хороший спосіб запобігти такій ситуації, але лише незначна кількість БД надають таку можливість.

Найбільш просте рішення цієї проблеми - встановити спочатку звичайний програмний пакет для шифрування даних, а потім використовувати його в ваших PHP-скриптах. PHP може вам допомогти з цим завданням за допомогою таких розширень як Mcrypt і Mhash, що реалізують досить велике число алгоритмів шифрування. При такому підході скрипт спочатку шифрує зберi-

гаються дані, а потім дешифрує їх при запиті. Нижче наведені приклади того, як працює шифрування даних в PHP-скриптах [13].

3.6.1 Хешування

Хешування - це перетворення масиву вхідних даних довільної довжини у вихідні біти фіксованої довжини. Ці перетворення називаються хеш-функціями або функціями згортки, а їх результати називаються хешем, хеш-кодом або дайджестом повідомлення. Хеш використовується для порівняння даних: якщо два масиви мають різні хеш-коди, масиви гарантовано будуть різними; Якщо вони однакові, таблиці однакові. Взагалі, не існує однозначної відповідності між вихідними даними та хеш-кодом, оскільки кількість значень хеш-функцій менше, ніж варіанти вхідного масиву; Є багато масивів, що дають однакові хеш-коди - тобто колізії. Ймовірність колізії відіграє значиму роль в оцінці хеш-функцій. Існує велика кількість алгоритмів хешування з різними характеристиками (криптостійкість, обчислювальна складність, розрядність тощо). Вибір хеш-функції визначається специфікою проблеми [40]. Якщо ви працюєте із прихованими службовими даними або, навіть ,вимагаєте їх зашифрованого уявлення (тобто не вимагаєте, щоб це відображалося), тоді використовуйте хешування. Відомий приклад хешування- це зберігання криптографічного хешу з пароля в базі даних замість збереження вихідного значення.

Функція password дозволяє легко видаляти конфіденційні дані та працювати з цими хешами. password_hash () дозволяє розділити заданий рядок з найпотужнішими алгоритмами, а password_verify () перевіряє, що даний пароль відповідає хешу, що зберігається в базі даних. У лістингу 3.1 ми наведемо приклад хеш-полів із паролем. [13]

Лістинг *3.1 Хешування полів з паролями*

```
<?php
   // Зберігаємо хеш пароля
```

```
$query  = sprintf("INSERT INTO users(name,pwd) VALUES('%s
','%s');",
            pg_escape_string($username),
            password_hash($password, PASSWORD_DEFAULT));
$result = pg_query($connection, $query);

// перевірка введеного користуачем пароля на коректність
$query = sprintf("SELECT pwd FROM users WHERE name='%s';"
,
            pg_escape_string($username));
$row = pg_fetch_assoc(pg_query($connection, $query));

if ($row && password_verify($password, $row['pwd'])) {
    echo 'Вітаємо, ' . htmlspecialchars($username) . '!';
} else {
    echo 'Помилка авторизації, ' . htmlspecialchars($user
name) . '.';
}
?>
```

3.7 Захист від SQL – ін’єкцій

SQL- ін’єкція - це техніка запиту до бази даних зловмисника, яка дозволяє йому викрадати дані користувачів, отримувати доступ до їх паролів, видаляти їх з бази даних тощо. В даний час веб-додатки управляються базами даних. SQL-ін'єкції відбуваються, коли ви просите користувача ввести ім'я користувача та пароль, наприклад, а замість цього користувач вводе повідомлення SQL, яке запускається у вашій базі даних без вашого відома. Наприклад: (рис. 3.15)

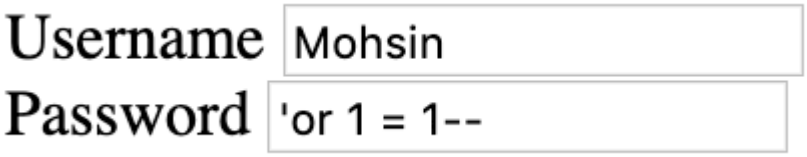

Рисунок *3.15* Приклад SQL-ін’єкції

З боку сервера хакерська атака, зображена на рисунку 3.15 матиме наступний вигляд виконання, яке зображене на лістінгу 3.2

Лістинг *3.2 Вигляд харекрські атаки з використанням SQL-ін’єкції з боку сервера*

```
SELECT * FROM USER_TABLE
WHERE Username= ‘Brian’ and Password=‘ ’ or 1 = 1 – ‘
```

1 = 1 завжди істинно, тому цей запит поверне всю інформацію про користувачів із таблиці.

Наведемо ще один приклад хакерської атаки з використанням SQL-ін’єкцій.(рис. 3.16)

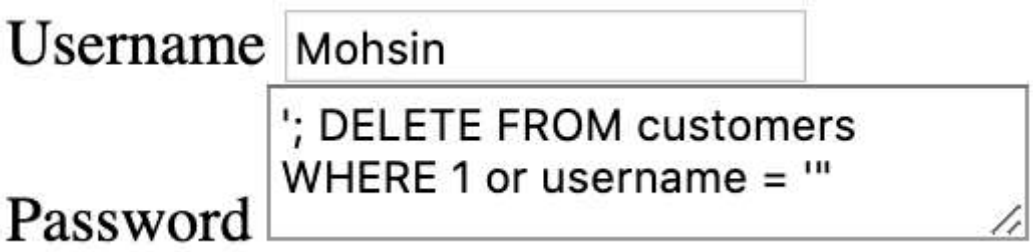

Рисунок *3.16* Приклад SQL-ін’єкції

З боку сервера хакерська атака, зображена на рисунку 3.16 матиме наступний вигляд виконання, яке зображене на лістінгу 3.3

Лістинг *3.3 Вигляд харекрські атаки з використанням SQL-ін’єкції з боку сервера*

```
SELECT * FROM customers
WHERE username = ‘Brian’ and Password= ‘ ’ ;
```

```
DELETE FROM customers WHERE 1 or username = ‘ ‘
```

Якщо хакер може запустити цей запит на сервері, таблиця клієнта буде порожньою, а всі записи зникнуть. [10]

Тож, як убезпечити інформаційну системи від атак з використанням SQL-ін’єкцій? Існує декілька методів захисту від SQL-ін’єкцій.

3.7.1 Захист від SQL – ін’єкцій за допомогою використання функції mysql_real_escape_string()

Перший метод захисту бази даних від ін'єкції SQL - використання функції mysql_real_escape_string (). mysql_real_escape_string - це рядок, який буде використовуватися в запиті SQL і повертатиме той самий рядок із усіма спробами інсталяції ін'єкції SQL, яких уникати. По суті, він замінить ті небезпечні лапки ('), які користувач може ввести за допомогою зворотної риски (\).На лістингу 3.4 можна побачити, як лапки ('), показані на рисунку 3.2 та 3.3 перетворюються на зворотню косю лінією (\).

Лістинг *3.4 Результат використання функції mysql_real_escape_string().*

```
SELECT * FROM customers
WHERE username = ‘Brian’ and Password= ‘\’ OR 1\’’

SELECT * FROM customers
WHERE username = ‘Brian’ and Password= ‘\’;
DELETE FROM customers WHERE 1 or username = \’’ [12]
```

3.7.2 Використання параметризованих запитів

Хороший спосіб запобігти атакам з боку SQL-ін'єкцій - використовувати параметризовані запити. Це означає, що вам потрібно визначити код SQL, який буде виконуватися за допомогою заповнювачів значення параметра,

програмно додавати значення параметрів і запускати запит. Він включає три кроки:

-Підготовка: шаблон оператора SQL буде створений та надісланий до бази даних. Деякі значення все ще невизначені, вони називаються параметрами (позначені "?").

-Аналіз бази даних, компіляція та оптимізація запиту за шаблоном оператора SQL та збереження результату без виконання результату.

-Виконання: Додаток прив'язує значення до параметра, і база даних виконується оператором. Оператор може виконувати програми з різними значеннями стільки разів, скільки потрібно.

В лістингу 3.5 наведемо приклад SQL запиту

Лістинг *3.5 Приклад запиту SQL*

```
$conn = new mysqli($servername, $username, $password, $dbname);
$stmt = $conn->prepare("INSERT INTO MyGuests (username, password) VALUES (?, ?)");$stmt->bind_param("ss", $_POST['username'];
$_POST['password']);$stmt->execute();$stmt->close();[12]
```

Далі в лістінгу 3.6 покажемо, як буде описаний лістінг 4 з використанням методу параметризованих запитів.

Лістинг *3.6 Використання параметризованих заппитів*

```
$conn->prepare("INSERT INTO MyGuests (username, password) VALUES (?, ?)");

$stmt->bind_param("ss", $username, $password);
```

(?) показує, де змінити значення параметра в операторі SQL. Це може бути ціле число, рядок, double або bob. Метод Bind покаже вам, які параметри

є в SQL. Аргумент "ss" покаже вам тип параметрів у SQL. Параметри можуть бути чотирьох типів:

- i – Integer;
- s – String;
- d – Double;
- b – BLOB;

Потім значення присвоюються параметрам і виконується запит.

Використовуючи параметризований запит, можна запобігти введення SQL-ін’єкцій, оскільки значення параметрів (навіть якщо вони вводяться SQL) не будуть виконуватися через те, що вони не є частиною оператора запиту і це буде розглядатися як введення користувачем у вигляді рядоку, а не код SQL. Іншими словами, запит буде шукати користувача з цим паролем замість того, щоб виконувати несподіваний код SQL [12].

3.8 Клієнтська частина медичної інформаціної системи едногастроетерологічного відділення лікарні

З боку клієнтської частини застосунок має декілька сторінок. Головна сторінка – це поле для ідентифікації користувача, де користувач має ввести свій пароль та логін, аби побачити дані інформаційної системи та мати змогу їх редагувати. Створено два різних вида користувачів: з правами аміністратора та звичаний користувач. Можливості редагвання даних в інформаційній системі для обох користувачів різні. Звичайний може може: реадгувати список пацієнтів, додавати їм аналізи, медикаменти, ультразвукові дослідження тощо. Аміністратор може переглядати список пацієнтів, але не може редагувати його, але може вносисти та переглядати нових користувачів інформаційної системи, чого не може робити користувач, який не має прав адміністратора.

3.8.1 Автентифікація користувача

Коли користувач відкриває інформаційну систему йому спершу потрібно ввести свій логін та пароль, аби мати змогу переглядати та змінювати дані системи. Тож, на сторнінці автентифікації користувача є два поля:

1. Для вводу логіну, який є адресою електроної пошти користувача.
2. Для вводу пароля.

На рисунку 3.17 показано вид сторінки автентифікації користувача.

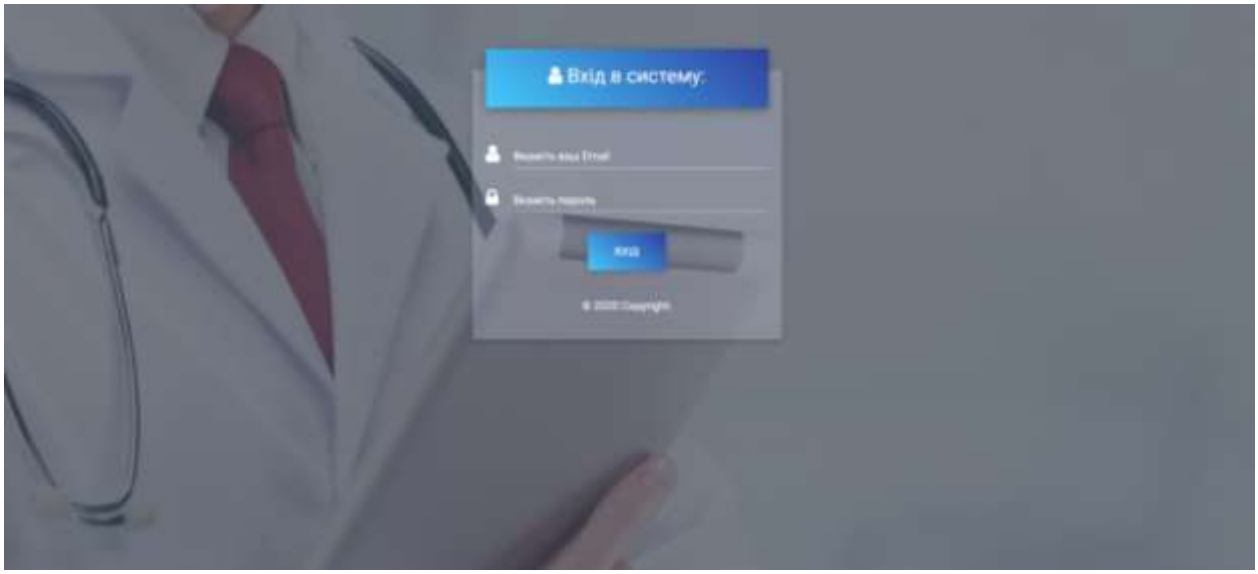

Рисунок *3.17* Сторінка автентифікації користувача

В системи є два типа користувачів:

1) Користувач з правами адміністратора.
2) Звичайний користувач, який не володіє правами аміністратора.

3.8.2 Головна сторінка

При розрбці данної інформаційної системи мною було прийнято рішення, що у системи буде два типи користувачів:

а) перший буде мати права адміністраторів системи і мати змогу перглядати інформацію про свої пацієнтів, додавати медикаменти до списку та переглядати список медикаментів та додаваи або видаляти користувачів:

б) другий буде звичайним користувачем та не матиме можливості заности в систему нових корситувачів або діагноз пацієнта, але на відміну від користувача з правами адміністратора, звичайний користувач матиме змогу додавати пацієнтів, ультразвукові обстеження, процедури.

На головній сторінці інформаційної системи для користувача, який заходить з правами аміністратора розташовано такі кнопоки:

а) для перегляду даних: пацієнти, ін’єкції, ультразвукові та рентгенологічні обстеження, фізіотерапевтичні процедури, консультації вузьких спеціалістів, медикаменти, лист харчування по дієтстолам;

б) для додавання даних: додати користувача (рисунки 3.18, 3.19).

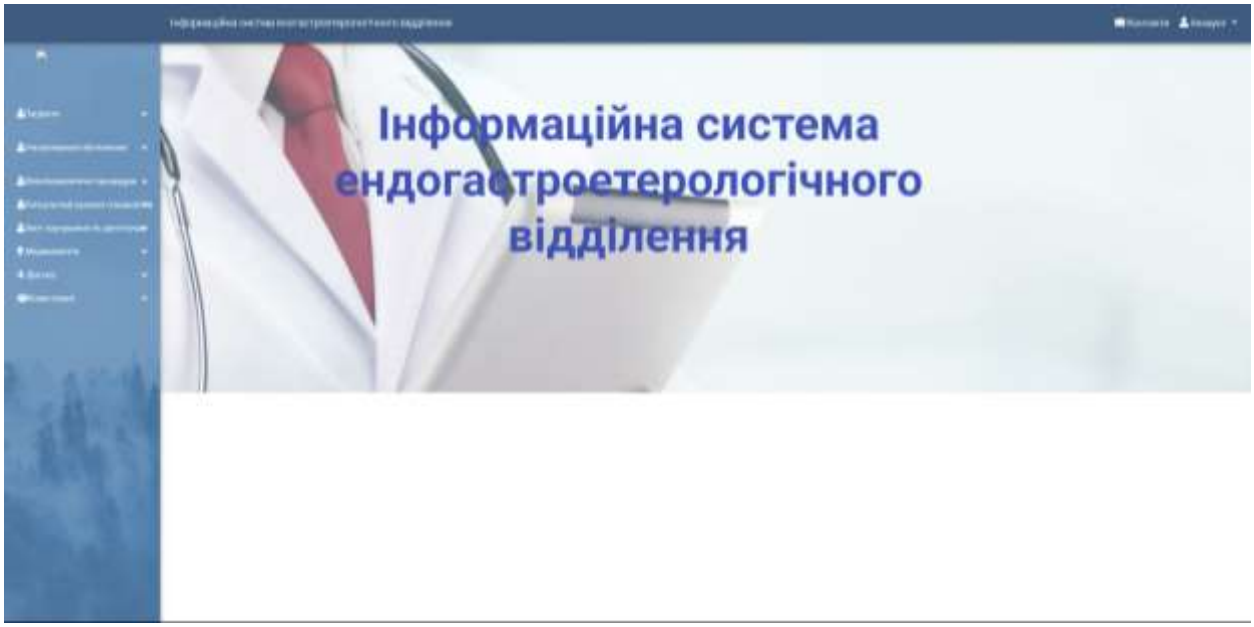

Рисунок *3.18* Головна сторінка для користувача з правами адміністратора

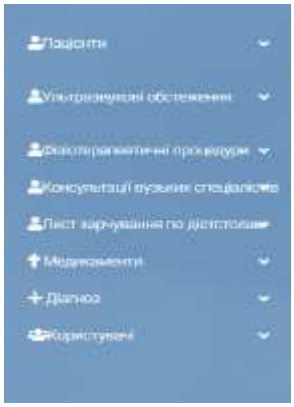

Рисунок *3.19* Кнопки на говній сторінці користувача без прав адміністратора

На головній сторінці інформаційної системи для користувача, який не має прав адміністратора розташовано такі кнопоки:

в) для перегляду даних: пацієнти, ін’єкції, ультразвукові та рентгенологічні обстеження, фізіотерапевтичні процедури, консультації вузьких спеціалістів, бюджетні медикаменти, лист харчування по дієтстолам (рис. 3.1);

г) для додавання даних: додати пацієнта, додати ультрозвукове обстеження, додати процедуру, додати медикамент, додати дієтстол, додати вузького спеціаліста (рисунки 3.20, 3.21).

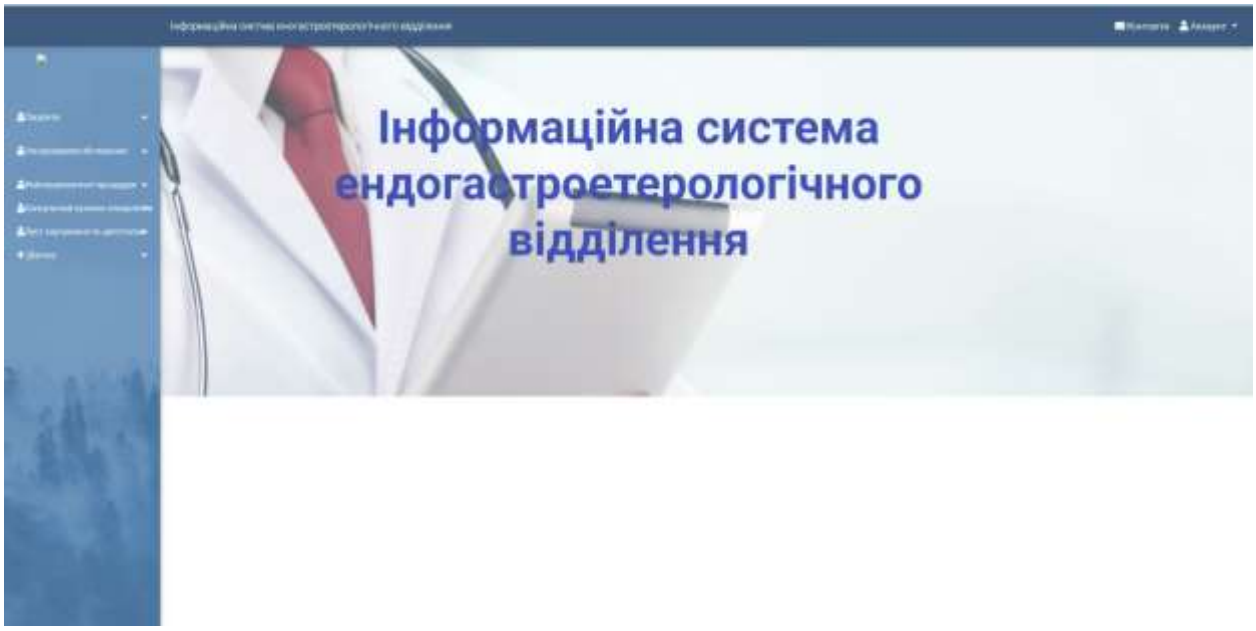

Рисунок *3.20* Головна сторінка для користувача без прав адміністратора

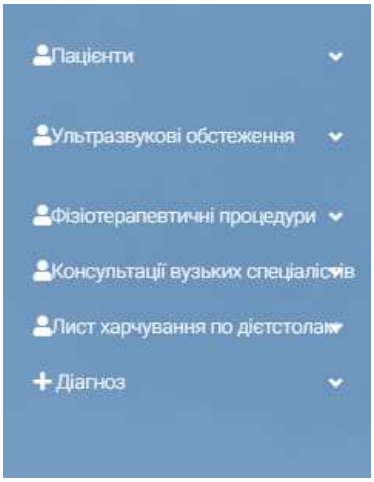

Рисунок *3.21* Кнопки на говній сторінці користувача без прав адміністратора

3.8.3 Перегляд даних

Переглянути данні будь-якої таблиці користувач може, натиснувши на кнопку «перглянути список» для користувача, який володіє правами адміністратора є такі поля «Медикаменти» та «Користувачі», де він може додавати дані. На рисиунку 3.22 показано приклад перегляду списку медикаментів.

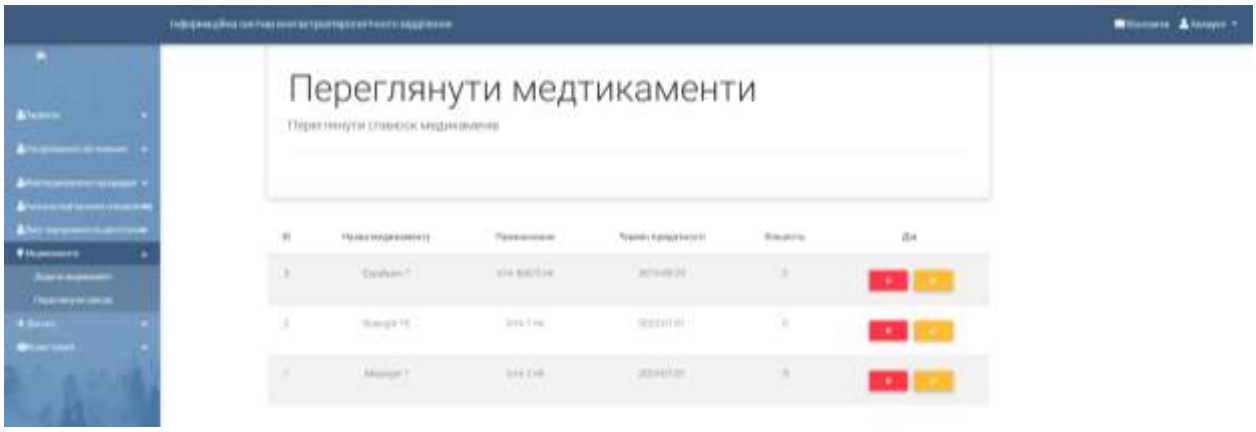

Рисунок *3.22* Перегляд таблиці пацієнти

Після пергляду користувач може видалити медикамет або відредагувати його (рисунки 3.23, 3.24).

Рисунок *3.23* Видалення медикаменту

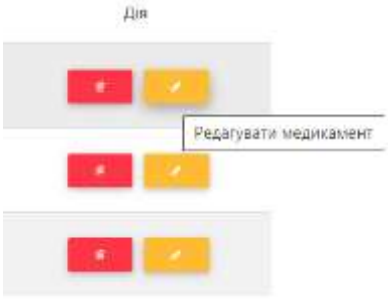

Рисунок *3.24* Зміна медикаменту

Також користувач може переглянути список наявних пацієнтів у відділенні (рисунок 3.25) та список користувачів системи (рисунок 3.26).

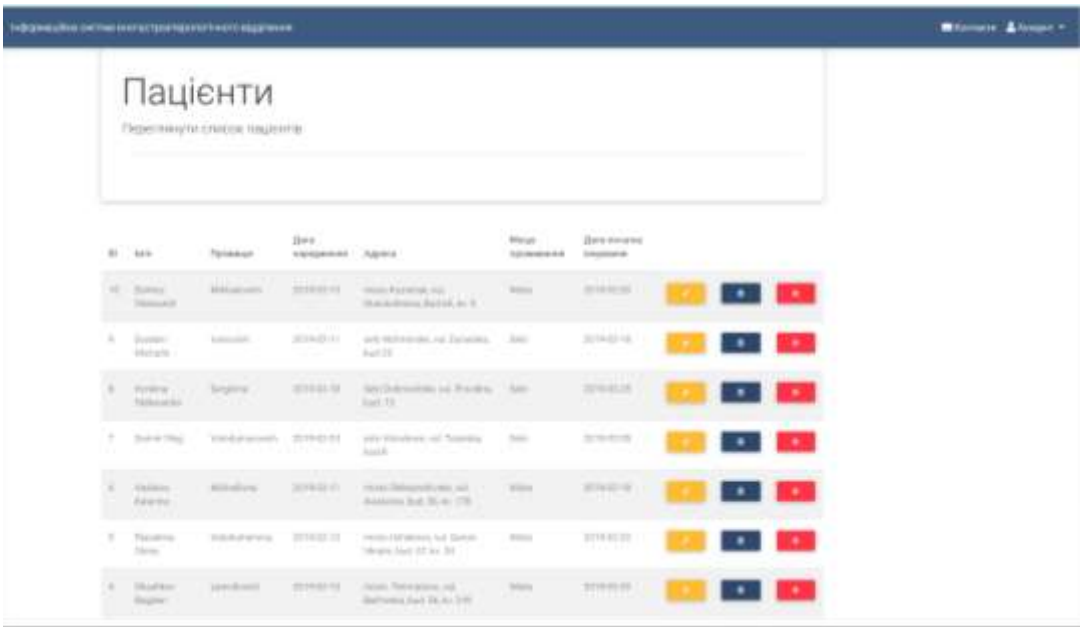

Рисунок *3.25* Вміст таблиці «пацієнти»

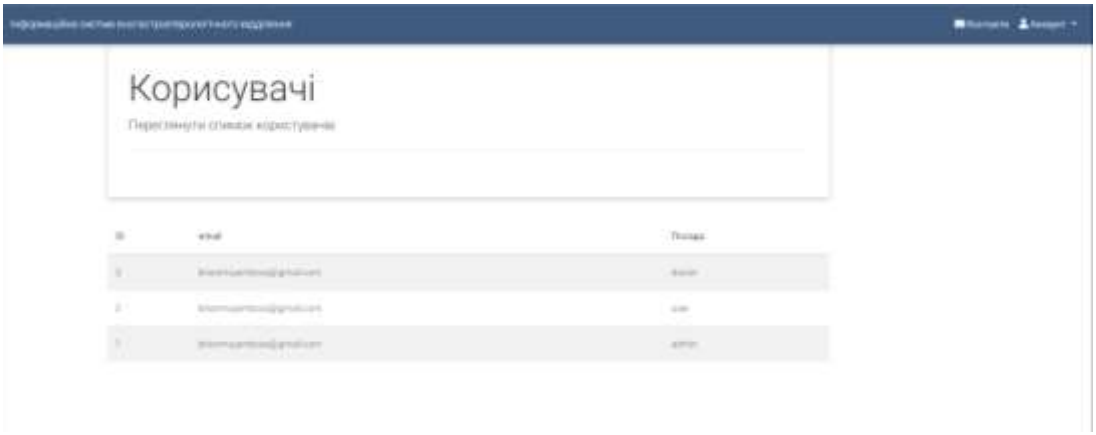

Рисунок *3.26* Перегляд списка користувачів інформаційної системи

Таким чином, як показано вище, можна переглядати вміст таблиць «ін’єкції», «ультразвукові та рентгенологічні обстеження», «фізіотерапевтичні процедури», «консультації вузьких спеціалістів», «медикаменти», «лист харчування по дієтстолам».

3.8.4 Додавання даних

Для того, щоб додати дані користувач має натиснути на одну з кнопок для додачі даних. Наприклад, щоб додати медикамент користувач має натиснути на кнопку «додати медикамент». Після цього він має заповнити всі необхідні поля і натиснути кнопку «Додати медикамент» (рисунки 3.27, 3.28).

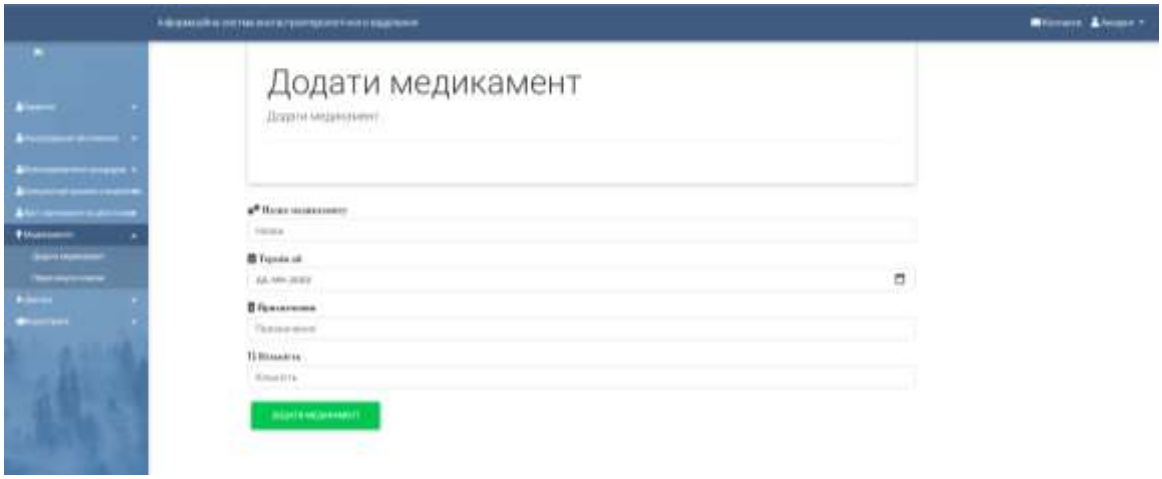

Рисунок *3.27* Заповнення даних таблиці «медикаменти»

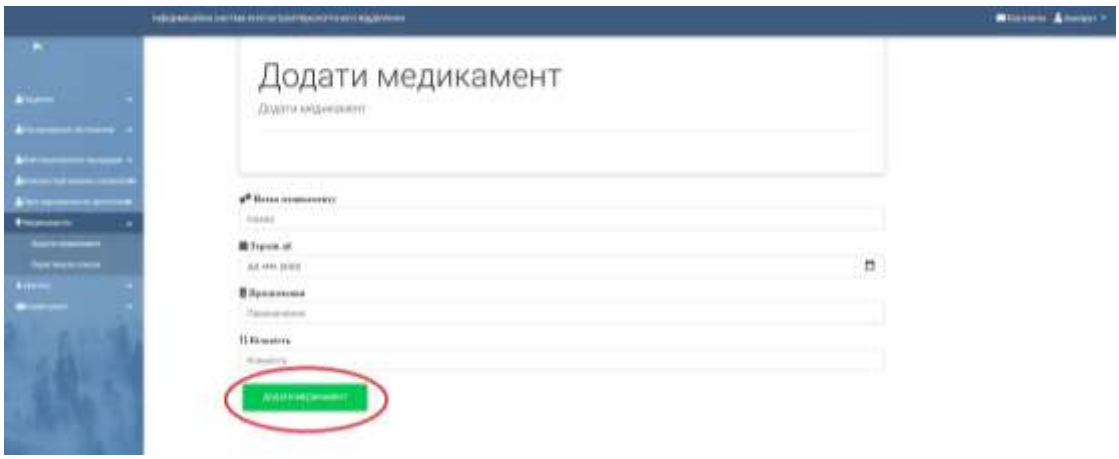

Рисунок *3.28* Додавання медикаменту

Після того, як користувач натискає кнопку «Зберегти» він автоматично переходить до таблиці «ін’єкції», де може подивитися наскільки точно всі данні були додані (рисунок 3.29)

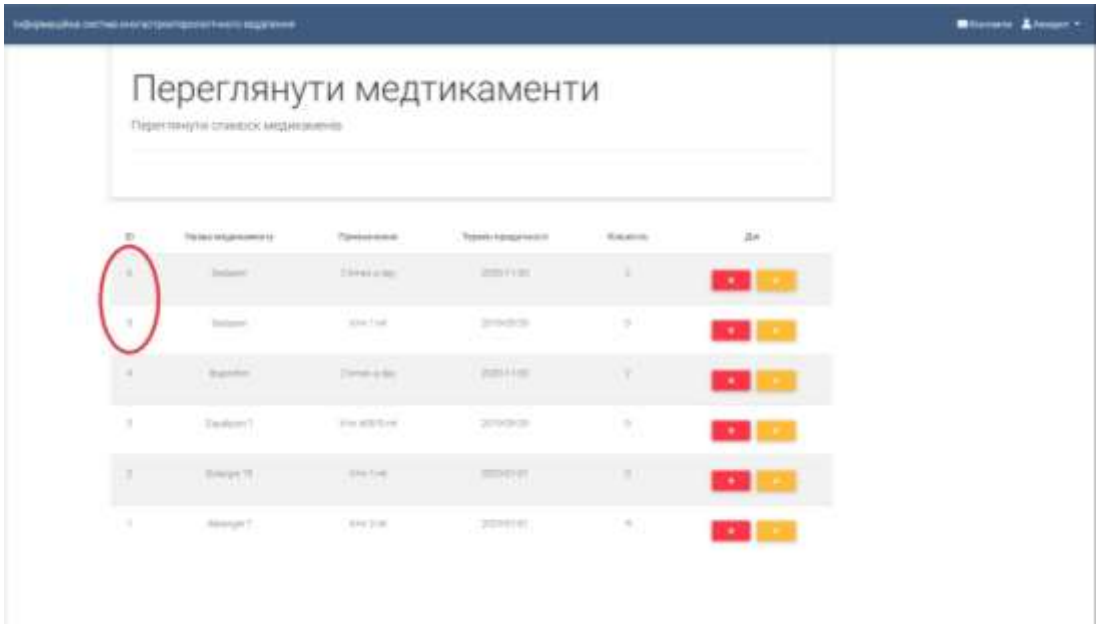

Рисунок *3.29* Результат додавання медикаменту

Так само можна додавати данні в таблиці: «аналізи», «пацієнт», «лист харчування по дієтстолах».

3.8.5 Видалення даних

Для того, щоб видалити данні з таблиці користувачу достатньо натиснути кнопку «Видалити дані». Наведом приклад видалення пацієнта таблиці. (рисунок 3.30)

Рисунок *3.30* Засіб видалення даних з таблиці «Пацієнти»

Результат видалення даних (рисунок 3.31).

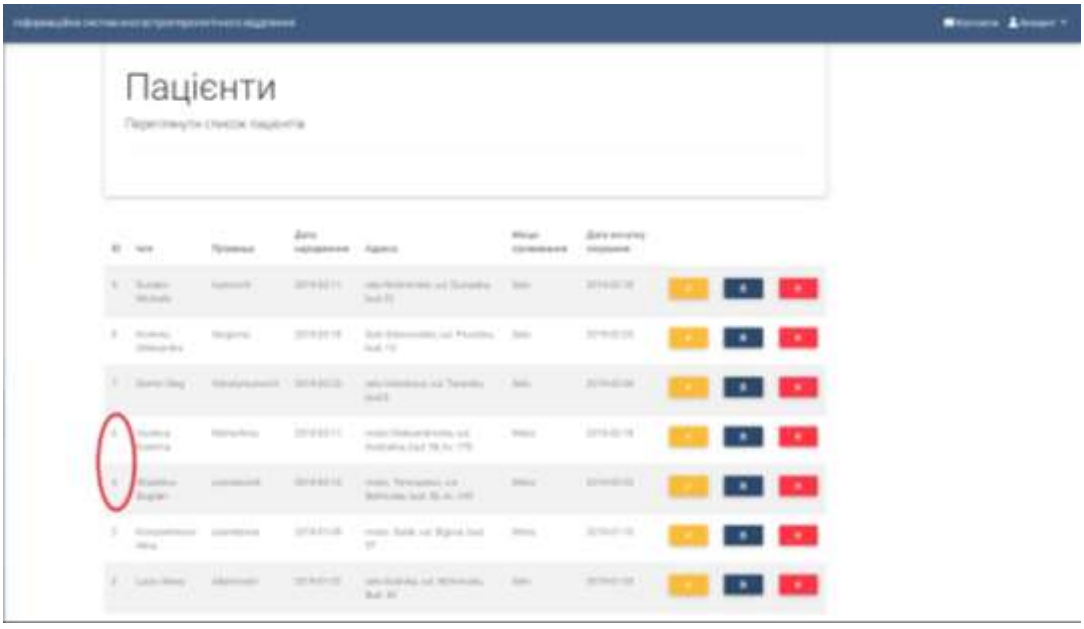

Рисунок *3.31* Результат видалення даних з таблиці «пацієнти»

Користувач, який не має прав адміністратора таким самим чином, як показано вище, може перглядати список, додавати, редагувати та видаляти

дані таблиці «ультразвукові обстеження», «фізотерапевтичні процедури», «консультації вузьких спеціалістів», «лист хврчування по дієтстолам».

3.8.6 Вихід з системи

Вийти з системи кожен користувач може натиснувши на кнопку у верньому правому кутку «Аккаунт», а потім натиснуваши «Вийти» (рисунок 3.32).

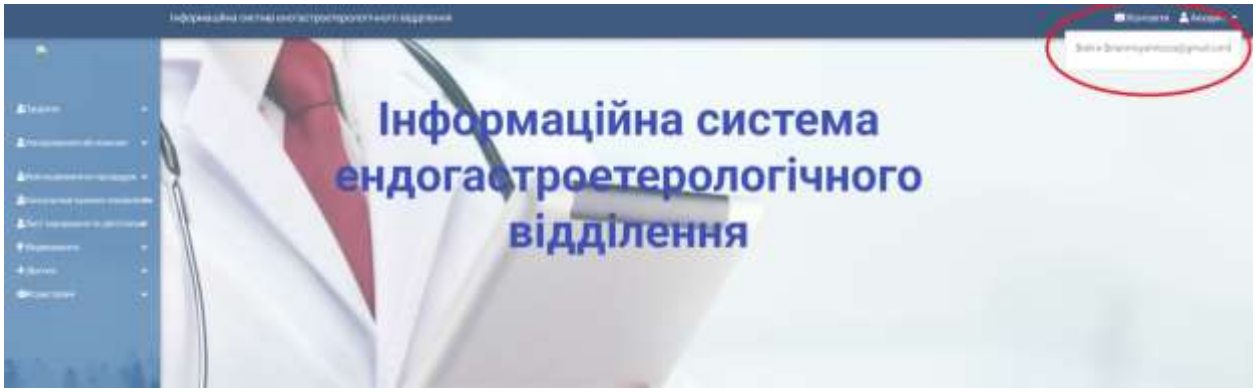

Рисунок *3.32* Вихід з інформаційної системи ендогастроетерологічного відділення

3.9 Серверна частина інформаційної системи ендогастроетерологічного відділення лікарні

Серверна частина програми була створена за мови програмування PHP. Робота серверу полягає у з'єднанні клієнтської частини застосунку з базою даних з метою надати можливість користувачу додавати, переглядати та редагувати дані, які зберігаються в базі даних.

3.9.1 Підключення до бази даних

На лістингу 3.7 наведено код з'єднання з базою даних інформаційної системи едногастроетерологічного відділення Обласної дитячої лікарні.

Лістинг *3.7* – з'єднання з базою даних

```
<?php

class DBconnection{

    protected $db;

 public function getConnection()
    {
       return $this->connect();
    }

  public function connect(){

    $conn = NULL;

        try{
            $conn = new

PDO("mysql:host=localhost;dbname=his", "root", "");

            $conn->setAttribute(PDO::ATTR_ERRMODE,

 PDO::ERRMODE_EXCEPTION);

            } catch(PDOException $e){

                echo 'ERROR: ' . $e->getMessage();

                }
            $this->db = $conn;

            return $this->db;
    }

  }
```

4 ДОСЛІДЖЕННЯ СИСТЕМ ШИФРУВАННЯ МЕДИЧНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

Метою кваліфікаційної роботи було порівняння систем шифрування для медичної інформаційної системи. В цій частині роботи буде проведений аналіз використаних методів шифрування медичної інформаційної системи ендогастроетерологічного відділення лікарні.

Дослідження проводилось шляхом порівняння тих методів шифрування даних, які були використані при розробці інформаційної системи ендогастроетерологічного відділення Обласної дитячої лікарні.

Наразі існує дуже дабато медичних систем, які використовуються безліччю медичних закладів. Всі вони зберігають інформацію про пацієнтів цих медичних установ і ці дані є конфеденційними, тож всі ці інформаційні системи мають закритий код алгоритмів захисту та шифрування даних в них, тому робити дослідження порівняння систем шифрування в них з інформаційною системою ендогастроетерологічного відділення лікарні або іншими медичними інформаційними системами не є доступними. В рамках дослідження було обрано наступні медичні інформаційні системи: Helsi, Health24, інформацію про структуру та шифрування даних яких нам відомо. Також для порівняння була обрана розроблена в цій роботі медична інформаційна система ендосгастроетерологічного відділення Обласної дитячої клінічної лікарні. Вибір заснований на частоті згадки в сучасних науково-дослідних журналах і існуючими розробками останніх років [7, 37].

4.1 Дослідження медичних інформаційних систем

При створенні захисту інформаційної системи ендогастроетерологічного відділення був використан такий метод шифрування, як хешування даних з використанням протоколу SSL. Результати, які були отримані після вкористання зазаначеного методу шифрування були такими ж, як і в інших медич-

них інформаційних системах, тому було прийняте рішення порівняти дані отриманні в медичній інформаційні системі енгостроетерологічного відділення лікарні з даними інших інформаційних систем.

Медичні інформаційні системи були порівнянні за своїми характеристиками. Критеріями порівняння харакеристик медичних інформційних систем було обрано (таблиця 4.1):

а) чи має медична інформаціна система клієнт-серверну архітектуру;

б) який рівень вимог до апаратних засобів в медичній інформаційній системі;

в) наявність механізмів захисту інформації;

г) наявність web – інтерфейсу;

д) використання баз даних;

е) моливість конфігурації системи.

Таблиця *4.1*

Порівняння основних харакеристик МІС

	Helsi	Health24	Медична інформаційна система ендогастроетерологічного відділення ОДКЛ
Наявність клієнт-серверної архітектури	+	+	+
Рівень вимог до апаратних засобів	Низький	Високий	Низький
Наявність механізмів захисту інформації	+	+	+
наявність web – інтерфейсу	+	+	+
використання баз даних	ДВ	+	+
моливість конфігурації системи	ДВ	+	+

*ДВ – дані відсутні

Також були розглянуті алгоритми шифрування даних вище зазначених систем (таблиця 4.2).

Таблиця *4.2*

Порівняння алгоритмів шифрування даних МІС

Медична інформаційна система	Використаний алгоритм шифрування
Helsi	Цифровий підпис на базі еліптичних кривих з використанням одноразових паролей
Health24	Розмежування доступу до даних, цифровий підпис на базі алгоритму RSA, хешування
Медична інформаційна система ендогастроетерологічного відділення ОДКЛ	Розмежування доступу до даних, хешування

В медичних інформаційних системах Helsi та Health24 для захисту інформації використовується метод захисту інформації – цифровий підпис, але алгоритми задання цийорового підпису використовуються різні. Порівняємо їх (таблиця 4.3)

Таблиця *4.3*

Порівняння алгоритмів цифрового підпису МІС

Криптостійкість (bit)	80	112	128	152	256
Епіліптичні криві (довжина публічного ключа bit)	160	224	256	384	512
RSA(довжина публічного ключа bit)	1024	2048	3072	7680	15360

Проаналізуємо властивості методів шифрування медичних інформаційних систем Helsi, Health24 та енлогастроетерологічного відділення ОДКЛ. Медична інформаційна система Helsi використовує цифровий підпис на базі еліптичних кривих(ECDSA). В медичній інформаційній системі Health24 також використорвує цифровий підпис, а також протокол захисту інформації SSL/ TLS, який використовує алгорим RSA для шифрування даних. Медична інформаційна система ендогастроетерологічного відділення ОДКЛ, як і інформаційна система Health24 використовує протокол захисту інформації SSL з алгоритмом RSA для шифрування свої даних. Порівняємо властивості цих методів шифрування даних(таблиця 4.4):

Таблиця *4.4*

Порівняння алгоритмів шифрування в МІС

Алгоритм	ECDSA	RSA
Швидкість шифрування даних	Висока	Низька
Довжина ключа(біт)	160	1024
Час генерації ключа	Секунди	Хвилини
Тип ключа	Асиметричний	Асиметричний
Криптостійкість	Крипостійкість вище, ніж у RSA	2,7•1028 для ключа завдовжки 1300 біт

Під час досліджень були проаналізовані протоколи, які використовувались для шифрування даних медичних інформаційних систем (табл. 4.5).

Таблиця *4.5*

Протоколи шифрування даних МІС

Медична інформаційна система	Використаний сертифікат
Helsi	ДВ
Health24	SSL і TLS
Медична інформаційна система ендо-гастроетерологічного відділення ОД-КЛ	SSL

*ДВ – дані відсутні

Проаналізувавши отримані вище результати, можна скласти характеристику кожної з розглянутих медичних інформаційних систем систем та засобів шифрування даних цих систем.

Медична інформаційна система Helsi. Має клієнт-серверну архітектуру, низький рівень вимог до апаратних засобів, механізми для захисту інформації та web – інтерфейс. Шифрування даних в цій системі проводиться за допомогою цифрового підпису на базі еліптичних кривих з використанням одноразових паролей. Чи використовуються якісь протоколи для шифрування та захисту даних в медичній інформаційній системі Helsi виявити не вдалось. Довжина публічного ключа, який використовується в системі Helsi для електроно – цифрового підпису більша, ніж у системи Health24, але швидкість шифрування даних вища, як і криптостійкість використаного алгоритму шифрування.

Медична інформаційна система Health24. Медична інформаційна система Health24 також має клієнт – сереверну архітектуру, але високий рівень вимог до апаратних засобів, має механізми захисту інформації та web – інтерфейс. Шифрування даних в цій системі проводится з використанням алгоритму цифрового підпису RSA. Також в системі для шфрування і захисту даних використовується протокол SSL/TLS. Довжина публічного ключа, який використовується в системі Health24 для електроно – цифрового підпису мень-

ша, ніж у системи Helsi. Швидкість шифрування даних меньше, як і крипостійкість, але перевагою використаного алгоритму RSA є те, що його включено до багатьох стандіртів і протоклів, у той час, як алгоритм створення цифрового підпису на базі еліптичних кривих новий метод, який ще повністю не розвинули і не дослідили.

Медична інформаційна система ендогастроетерологічного відділення ОДКЛ. Медична інформаційна система ендостроетерлогічного відділення також має клієнт – серверну архітектуру з нізким івнем вимог до апаратних засобів. Медична інформаційна система ендостроетерлогічного відділення предятавляє собою web – застосунок, тому має та web – інтерфейс. Шифрування даних в цій системі відбувається за допомогою використання протоколу шифрування даних. Хешування даних в інформаційній системі ендогастроетерологічного відділення відбувається з використанням алгоритму RSA.

ВИСНОВКИ

Під час написання кваліфікаційної роботи проаналізовано специфіку роботи ендогастроентерологічного відділення лікарні, а також методи та системи шифування медичних інформаційних систем.

Для розробки інформаційної системи були використані всі дані, якими оперують середній та вищій медичний персонал ендогастроентерологічного відділення Запорізької обласної дитячої лікарні.

Результатом кваліфікаційної роботи є розроблений веб-застосунок, в якому було реалізовано захист даних з використанням функції захисту та параметризованих запитів для СУБД MySQL для захисту від MySQL – ін’єкцій, шифрування сховища бази даних, а також використанням SSL – протоколу.

Веб-застосунок представляє собою медичну інформаційну систему конкретного відділення дитячої лікарні, яка зберігає інформацію про пацієнтів, їх лікарів, аналізи та обстеження, які вони пройшли та мають пройти. Для розробки застосунку використовувалась мови програмування JS, PHP та системи керування базами даних MySQL.

В подальшому ця система може бути розширена. Може бути створений захист від випадкової лікарської помилки. Наприклад, якщо лікар випадково вводить завелику дозу препарату для дитини система буде його про це сповіщати.

ПЕРЕЛІК ПОСИЛАНЬ

5. Buklib.net. Загальна характеристика та класифікація інформаційних систем обліку. URL: https://buklib.net/books/22551/ (дата звернення 15.05.2020)

6. Kmu.gov.ua. Правила забезпечення захисту інформації в інформаційних, телекомунікаційних та інформаційно – телекомунікаційних системах URL: https://www.kmu.gov.ua/npas/32791826 (дата звернення 15.05.2020)

7. Studfile.net. Основні загрози безпеці інформаційних систем. URL: https://studfile.net/preview/5118185/page:42/ (дата звернення 18.05.2020)

8. Studfile.net. Захист інформації в медичних інформаційних системах. URL: https://studfile.net/preview/6034038/page:39/ (дата звернення 18.05.2020)

9. Фергюсон Н., Шнайер Б. Практическая криптография. Изд-во : Диалектика, 2018. 420 с.

10. Державний стандарт України. Захист інформації. Технічний захист інформації. Основні положення. ДСТУ 3396.0-96: URL: http://www.dsszzi.gov.ua/dsszzi/control/uk/publish/article?art_id=38883&cat_id=38836 (дата звернення 18.05.2020)

11. Helsi. URL: https://reform.helsi.me/esign (дата звернення 19.05.2020)

12. Урядовий портал. URL: https://www.kmu.gov.ua/news/247952015 (дата звернення 19.05.2020)

13. Collier R. New tools to improve safety of electronic health records. CMAJ. 2014;186(4):251–251.

14. Jannetti MC. Safeguarding patient information in electronic health records. AORN J. 2014;100(3):C7–C8.

15. Hunter, E.S., Electronic health Records in an Occupational Health Setting--Part I. A global overview. Workplace health & safety.

16. Lemke J. Storage and security of personal health information. OOHNA J. 2013;32(1):25–26.

17. Liu V, Musen MA, Chou T. Data breaches of protected health information in the United States. JAMA. 2015;313(14):1471–1473.

18. Liu V, Musen MA, Chou T. Data breaches of protected health information in the United States. J. Am. Med. Assoc. 2015;313(14):1471–1473.

19. U.S. Department of Health and Human Services. Cybersecurity: 10 best practices for the small healthcare environment. URL: https://www.healthit.gov/sites/default/files/basic-security-for-the-small-healthcare-practice-checklists.pdf. (дата звернення 15.05.2020)

20. Wikina SB. What caused the breach? an examination of use of information technology and health data breaches. Perspect. Health Inf. Mana. 2014;2014:1–16.

21. Wang CJ, Huang DJ. The HIPAA conundrum in the era of mobile health and communications. JAMA. 2013;310(11):1121–1122.

22. Wickboldt AK, Piramuthu S. Patient safety through RFID: Vulnerabilities in recently proposed grouping protocols. J. Med. Syst. 2012;36(2):431–435.

23. Jannetti MC. Safeguarding patient information in electronic health records. AORN J. 2014;100(3):C7–C8.

24. Vockley M. Safe and secure? Healthcare in the Cyberworld. J Biomed. Instrum. Technol. 2012;46(3):164–173.

25. Nikooghadam M, Zakerolhosseini A. Secure communication of medical information using mobile agents. J. Med. Syst. 2012;36(6):3839–3850.

26. Chen HM, Lo JW, Yeh CK. An efficient and secure dynamic id-based authentication scheme for telecare medical information systems. J. Med. Syst. 2012;36(6):3907–3915.

27. Sittig D, Singh H. Electronic health records and National Patient-Safety Goals. N. Engl. J. Med. 2012;367(19):1854–1860.

28. Chen YY, Lu JC, Jan JK. A secure EHR system based on hybrid clouds. J. Med. Syst. 2012;36(5):3375–3384.

29. Lee HC, Chang SH. RBAC-matrix-based EMR right management system to improve HIPAA compliance. J. Med. Syst. 2012;36(5):2981–2992.

30. ncbi.nlm.nih.gov. Sequrity techniques for the electronic health records URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5522514/#CR9 (дата звернення 20.10..2020)

31. Google Академія,. Secure database access through partial encryption URL: https://patents.google.com/patent/US7418600B2/en (дата звернення 12.10.2019)

32. Bourabai.kz. Методы защиты от XSS-атак и SQL-инъекций. URL: http://bourabai.kz/php/secure-php.htm (дата звернення 20.10..2020)

33. Spy-soft.net. Топ-5 защит от XSS-атак. URL: https://spy-soft.net/zashhita-ot-xss/ (дата звернення 20.10.2020)

34. Docs.microsoft.com. What's the Right Way to Prevent SQL Injection in PHP Scripts? URL: https://docs.microsoft.com/ru-ru/archive/blogs/brian_swan/whats-the-right-way-to-prevent-sql-injection-in-php-scripts (дата звернення 20.10.2020)

35. Php.net. Шифрование хранилища базы данных. URL: https://www.php.net/manual/ru/security.database.storage.php (дата звернення 20.10.2020)

36. Бураков П. В., Петров В. Ю. Введение в системы баз данных: учебное пособие, Санкт-Петербург: Санкт-Петербургский государственный университет информационных технологий, механики и оптики, 2010, 128 с.

37. Allbest. Проектирование базы данных для медицинского учреждения. URL: https://knowledge.allbest.ru/programming.html (дата звернення 03.11.2020)

38. Уроки программирования для начинающих. Хостинг PHP MySQL – в чем отличия и преимущества? URl: http://www.programm-school.ru/hosting/hosting-php-mysql-v-chem-otlichiya-i-preimushchestva (дата звернення 03.11.2020)

39. kievoit.ippo.kubg.edu.ua. Мова програмування PHP, середовище програмування. URl:

http://www.kievoit.ippo.kubg.edu.ua/kievoit/2016/43_PHP/index.html (дата звернення 03.11.2020)

40. sites.znu.edu.ua. PHP+MySQL. URl: http://sites.znu.edu.ua/webprog/lect/1222.ukr.html (дата звернення 03.11.2020)

41. mis.h24.ua. Забезпечення безпеки та конфеденційності інформації системи URL: https://mis.h24.ua/FAQ/article/145 (дата звернення 20.10.2020)

42. hostiq.ua. Що таке SSL сертифікат? URL: https://hostiq.ua/ukr/info/what-is-ssl/ (дата звернення 20.10.2020)

43. css.in.ua. Що таке SSL сертифікат і нащо він потрібен? URL: https://css.in.ua/article/shcho-take-ssl-sertyfikat-i-navishcho-vin-potriben_4 (дата звернення 20.10.2020)

44. wiki.tntu.edu.ua.Хешування.URL:https://wiki.tntu.edu.ua/%D0%A5%D0%B5%D1%88%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F (дата звернення 20.10.2020)

45. Альбрехт Й. О. Система аутентифікації на базі еліптичних кривих з використанням векторних операцій, КПІ ім. І. Сікорського, 2018.

46. Касілова Наталя, студентка магістратури ФЕЕІТ ІІ ЗНУ. Наук. кер.: к.ф.-м.н., доц. Скрипник І.А. «ПОРІВНЯННЯ СИСТЕМ ШИФРУВАННЯ НА ПРИКЛАДІ МЕДИЧНОЇ СИСТЕМИ». Збірник наукових праць студентів, аспірантів і молодих вчених «Молода наука-2020» : у 5 т. Запорізький національний університет. Запоріжжя: ЗНУ, 2020. Т.5. С. 71-73.

47. Касілова Наталя, студентка магістратури ФЕЕІТ ІІ ЗНУ. Наук. кер.: к.ф.-м.н., доц. Скрипник І.А. «ЗАХИСТ ДАНИХ МЕДИЧИНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ». Збірник наукових праць студентів, магістрів, аспірантів, молодих вчених та видкладачів «XXV науково-технічної конференції» : Запорізький національний університет. Запоріжжя: ЗНУ, 2020. С.164 .

Декларація

академічної доброчесності

здобувача ступеня вищої освіти ЗНУ

Я, Касілова Наталя Сергіївна, студентка _2_ курсу, форми навчання денної, Інженерного навчально-наукового інституту, спеціальність 121 Інженерія програмного забезпечення, адреса електронної пошти sp114-12@stu.zsea.edu.ua, — підтверджую, що написана мною кваліфікаційна робота на тему «**Порівняння методів штучного інтелекту при розпізнаванні пісенної і розмовної мови**» відповідає вимогам академічної доброчесності та не містить порушень, що визначені у ст.42 Закону України «Про освіту», зі змістом яких ознайомлений.

- заявляю, що надана мною для перевірки електронна версія роботи є ідентичною її друкованій версії;

згоден/згодна на перевірку моєї роботи на відповідність критеріям академічної доброчесності у будь-який спосіб, у тому числі за допомогою інтернет-систем, а також на архівування моєї роботи в базі даних цієї системи.

Дата 30.11.2020 Підпис _____ _____ Касілова Наталя Сергіївна
(студентка)

Дата 30.11.2020 Підпис _____ __ Скрипник Ірина Анатолівна
(науковий керівник)