

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерних наук (або центр післядипломної освіти, або навчально-науковий центр заочної форми навчання)
(повна назва)

Кафедра _____ програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти _____ перший (бакалаврський)

Програмна система обліку клієнтів стоматологічної поліклініки
(тема)

Виконав:

студент 4 курсу, групи ПЗПП-22-1

_____ Веленський А.В.

(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного забезпечення

(код і повна назва спеціальності)

Тип програми _____ освітньо-професійна

Освітня програма Програмна інженерія

(повна назва освітньої програми)

Керівник _____ доц. Вечур О.В.

(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри

_____ (підпис)

_____ доц. Дудар З.В.

(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Центр післядипломної освіти

Кафедра	програмної інженерії
Рівень вищої освіти	перший (бакалаврський)
Спеціальність	121 – Інженерія програмного забезпечення
Тип програми	Освітньо-професійна
Освітня програма	Програмна Інженерія

(шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«____» _____ 2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

студентові _____ Веленському Андрію Вікторовичу
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Програмна система обліку клієнтів стоматологічної поліклініки

Затверджена наказом по університету від 17.06.2024 р. № 588 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 14.07.2024 р.

3. Вихідні дані до роботи Розробити програмний додаток з базою даних стоматологічної поліклініки, яка містить мінімум 10 сутностей, по 5 мінімум атрибутів кожна

4. Перелік питань, що потрібно опрацювати в роботі

Вступ, аналіз предметної галузі, формування вимог до програмної системи, архітектура та проектування програмного забезпечення, опис прийнятих програмних рішень, тестування розробленого програмного забезпечення, висновки, додатки.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі	17.06.2024	<i>виконано</i>
2	Створення специфікації ПЗ	18.06.2024	<i>виконано</i>
3	Проектування ПЗ	24.06.2024	<i>виконано</i>
4	Розробка ПЗ	30.06.2024	<i>виконано</i>
5	Тестування ПЗ	05.07.2024	<i>виконано</i>
6	Оформлення пояснювальної записки	10.07.2024	<i>виконано</i>
7	Підготовка презентації та доповіді	12.07.2024	<i>виконано</i>
8	Попередній захист	14.07.2024	<i>виконано</i>
9	Нормоконтроль, рецензування	17.07.2024	<i>виконано</i>
10	Здача роботи у електронний архів	20.07.2024	<i>виконано</i>
11	Допуск до захисту у зав. кафедри	21.07.2024	<i>виконано</i>

Дата видачі завдання 17 червня 2024р.

Студент (ка) _____ Веленський А.В.
(підпис)

Керівник роботи _____ доц. Вечур О.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка до кваліфікаційної роботи бакалавра, 57 стор., 10 рис., 4 табл., 7 джерел.

БАЗА ДАНИХ, СИСТЕМА ОБЛІКУ, C#, SQL.

Об'єкт розробки – програмна система обліку клієнтів стоматологічної клініки.

Мета розробки – створення програмної системи обліку клієнтів стоматологічної клініки.

Метод рішення – середовище розробки Microsoft Visual Studio, мови програмування C# та SQL.

Explanatory note to the bachelor's thesis, 58 pages, 10 figures, 4 tables, 7 sources.

DATABASE, ACCOUNTING SYSTEM, C#, SQL.

The object of development is a software system for accounting of clients of a dental clinic.

The purpose of the development is to create a software system for accounting of dental clinic clients.

The solution method is the Microsoft Visual Studio development environment, C# and SQL programming languages.

Я, Веленський Андрій Вікторович, студент гр. ПЗПП-22-1, здобувач вищої освіти на першому (бакалаврському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Програмна система обліку клієнтів стоматологічної поліклініки», що буде представлена до екзаменаційної комісії для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIAr KhNURE. Усі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови до допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

Вступ.....	7
1 Аналіз предметної галузі.....	10
1.1 Аналіз предметної галузі.....	10
1.2 Виявлення та вирішення проблем.....	11
1.3 Постановка задачі.....	20
2 Формування вимог до програмної системи.....	22
3 Архітектура та проєктування програмного забезпечення.....	25
3.1 UML проєктування ПЗ.....	25
3.2 Проєктування архітектури ПЗ.....	29
3.3 Проєктування структури зберігання даних.....	33
4 Опис прийнятих програмних рішень.....	38
5 Тестування програмного забезпечення.....	46
6 Впровадження програмного забезпечення.....	48
Висновки.....	50
Перелік джерел посилання.....	51
Додаток А.....	52
Додаток Б.....	53

ВСТУП

Стоматологічні послуги стали невід'ємною частиною життя сучасної людини. Збереження здоров'я зубів та профілактика стоматологічних захворювань є важливим завданням для кожного пацієнта. У зв'язку з цим, стоматологічні клініки відіграють ключову роль у забезпеченні якісного медичного обслуговування населення.

Проте, ефективне управління клінікою та обслуговування пацієнтів неможливе без належної організації обліку клієнтів та їх даних. На сьогоднішній день багато стоматологічних клінік все ще використовують застарілі методи ведення документації та обліку пацієнтів, такі як паперові картки та журнали. Такий підхід є неефективним, трудомістким і схильним до помилок. Він ускладнює пошук та аналіз даних, що призводить до зниження якості обслуговування пацієнтів та ускладнень в управлінні клінікою.

Впровадження сучасної програмної системи обліку клієнтів стоматологічної поліклініки є нагальною потребою для підвищення ефективності роботи клініки та забезпечення високого рівня обслуговування пацієнтів. Така система дозволить автоматизувати процеси ведення медичних карток, запису на прийом, обліку послуг та розрахунків з пацієнтами. Це значно спростить роботу персоналу, зменшить навантаження на адміністраторів та лікарів, забезпечить швидкий доступ до інформації про пацієнтів, їх історії лікування та призначень.

Система обліку клієнтів допоможе уникнути помилок, пов'язаних з людським фактором, та забезпечить надійне зберігання конфіденційних даних пацієнтів відповідно до вимог законодавства. Одним із ключових переваг впровадження програмної системи обліку клієнтів є можливість аналізувати статистичні дані про діяльність клініки, що дозволить керівництву приймати більш обґрунтовані рішення щодо покращення якості обслуговування, оптимізації витрат та розширення спектру послуг.

Загалом, впровадження сучасної програмної системи обліку клієнтів стоматологічної поліклініки є необхідним кроком для забезпечення ефективного функціонування клініки, підвищення якості обслуговування пацієнтів та забезпечення конкурентоспроможності на ринку стоматологічних послуг.

Створення об'єктно-орієнтованої бази даних, що міститиме докладну інформацію про кожного пацієнта, призначені процедури та лікування, дозволить забезпечити точність та доступність потрібної інформації у будь-який момент часу. Такий підхід до зберігання та систематизації інформації сприятиме ефективній координації медичного персоналу та покращенню якості надання медичних послуг.

Крім того, можливість генерування місячних звітів про відвідування та проведене лікування надасть можливість аналізу та оптимізації роботи медичного закладу, що є важливим кроком у покращенні медичної допомоги та задоволеності пацієнтів.

У даній роботі було поставлено мету – розробити систему обслуговування пацієнтів медичного закладу. В рамках даної роботи необхідно вирішити наступні задачі:

- провести аналіз сучасного стану систем обслуговування пацієнтів у медичних закладах;
- визначити основні проблеми та недоліки існуючих підходів до організації обліку та планування візитів пацієнтів;
- розробити концептуальну модель програмної системи обслуговування пацієнтів;
- визначити основні компоненти та їх взаємодії;
- розробити інтерфейс для адміністраторів та користувачів системи;
- реалізувати можливість введення, редагування та видалення інформації про пацієнтів, процедури та кабінети;

- провести тестування програмної системи на наявність можливих помилок та недоліків;

- вирішити виявлені проблеми та вдосконалити роботу системи;

- визначити можливі сфери застосування розробленої програмної системи в медичних закладах;

- оцінити потенційні переваги та внесок у поліпшення організації надання медичних послуг.

Об'єктом дослідження є процес розробки системи обслуговування пацієнтів медичного закладу. Предмет дослідження – технології розробки об'єктно-орієнтованих баз даних.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Аналіз предметної галузі

Здоров'я населення є однією з найважливіших складових благополуччя суспільства. Медичні заклади відіграють важливу роль у забезпеченні якісної медичної допомоги та підтримки пацієнтів на різних етапах їхнього лікування та реабілітації. Однак в умовах різноманітних медичних процедур та медичних втручань, точне планування та систематичний облік стає критичним для забезпечення ефективного надання медичної допомоги кожному пацієнту.

Традиційно, в багатьох медичних установах використовуються ручні методи ведення документації та організації записів пацієнтів. Однак такий підхід може призводити до недоліків у точному контролі за процесом лікування, а також до затримок у наданні медичних послуг через виникнення непередбачуваних ситуацій. Автоматизація обробки та аналізу даних може вирішити ці проблеми.

Інформаційні технології можуть значно полегшити ведення обліку пацієнтів та планування лікування. Розробка програмної системи, яка забезпечить автоматизовану обробку даних, стане ключовим кроком у вдосконаленні надання медичних послуг та підвищенні якості медичного обслуговування.

Дана програмна система дозволить зберігати та аналізувати дані кожного пацієнта, що включають інформацію про призначені процедури, рецепти, результати обстежень та інші медичні дані. Такий підхід дозволяє уникнути втрати важливої інформації та забезпечити її легкий доступ для медичного персоналу.

Однією з важливих переваг автоматизації даних є можливість в реальному часі моніторити стан пацієнтів та вчасно реагувати на зміни. Наприклад, система може нагадати про призначення, вказати на можливі взаємодії препаратів, а також допомогти в управлінні часом прийому препаратів.

Додатковим плюсом автоматизації даних є можливість генерувати звіти та статистику щодо надання медичних послуг. Це дозволяє аналізувати та вдосконалювати роботу медичного закладу, а також забезпечує підвищення ефективності його функціонування.

Автоматизація обробки даних у медичних закладах є важливим кроком у покращенні надання медичної допомоги та забезпеченні якості медичного обслуговування. Точне планування та контроль за станом пацієнтів стає більш ефективним та надійним завдяки впровадженню сучасних інформаційних технологій.

1.2 Виявлення та вирішення проблем

Існуючі комп'ютерні системи організації обліку та планування візитів пацієнтів стикаються з рядом ключових проблем та недоліків. По-перше, часто спостерігається низька ефективність їхньої інтерфейсної частини. Це може призводити до заплутаності та ускладнення роботи медичного персоналу.

Додатковим недоліком є обмежені можливості редагування даних в існуючих системах. Адміністраторам часто важко вносити необхідні зміни в інформацію про пацієнтів чи назначені процедури. Це може призвести до недооцінки необхідного лікування та ускладнити контроль за станом пацієнтів.

Крім того, існуючі системи можуть бути недостатньо гнучкими у відображенні різних форматів даних. Наприклад, можуть виникати труднощі у коректному відображенні та аналізі даних, що представлені у вигляді складних графіків чи таблиць.

Деякі системи можуть також виявляти проблеми у забезпеченні безпеки та конфіденційності медичних даних. Ці програмні додатки можуть бути вразливими перед загрозами з боку несанкціонованих доступів, що може призвести до просочення конфіденційної інформації.

Додатковим недоліком існуючих систем може бути їхня недостатня масштабованість. В разі збільшення потоку пацієнтів чи розширення медичного закладу, такі системи можуть виявитися неефективними.

Також можуть виникати проблеми із сумісністю існуючих систем з іншими медичними програмами та обладнанням. Це може призвести до недооцінки повного спектру можливостей та обмежити використання сучасних технологій у медичній практиці.

Зведемо ряд поширених проблем існуючих автоматизованих систем медичних закладів у таблицю 1.1.

Таблиця 1.1 – Деякі поширені проблеми існуючих автоматизованих систем медичних закладів

Проблема	Опис
Низька ефективність інтерфейсу	В існуючих системах часто відзначається неінтуїтивний інтерфейс, що ускладнює роботу медичного персоналу та може викликати заплутаність при спробі користуватися системою.
Обмежені можливості редагування даних	Адміністраторам може бути важко вносити необхідні зміни в інформацію про пацієнтів та призначені процедури через обмежені можливості редагування, що може призвести до недооцінки необхідного лікування та ускладнити контроль за станом пацієнтів.
Недостатня гнучкість у відображенні даних	Існуючі системи можуть мати обмежену здатність коректно відображати та аналізувати дані у різних форматах, що в свою чергу може призвести до

	труднощів у розумінні та використанні важливої інформації.
Проблеми у забезпеченні безпеки даних	Деякі системи можуть виявляти вразливість перед загрозами з боку несанкціонованих доступів, що може призвести до витоку конфіденційної інформації.
Недостатня масштабованість системи	У випадку збільшення потоку пацієнтів чи розширення медичного закладу, існуючі системи можуть виявитися неефективними та недостатньо гнучкими у роботі з великим обсягом даних.
Проблеми сумісності з іншими системами	Виникнення труднощів у сумісності з іншими медичними програмами та обладнанням може обмежити можливості системи та суттєво ускладнити використання сучасних технологій у медичній практиці.

Існуючі проблеми містять загальні алгоритми вирішення, пов'язані не лише із програмними можливостями автоматизації баз даних і практичними якостями розробника, а і з умінням створювати та тестувати нові ефективні моделі організації даних.

Для поліпшення ефективності роботи з системою, необхідно ретельно продумати та впровадити інтуїтивно зрозумілий та зручний інтерфейс. Це дозволить медичному персоналу швидше та ефективніше виконувати свої обов'язки, сприяючи точнішому плануванню візитів пацієнтів.

Щоб зробити інтерфейс більш ефективним, слід врахувати кілька ключових аспектів, що відображено у вигляді послідовного алгоритму дій на рисунку 1.1:

- а) **Мінімізація зайвого:** Уникайте надмірного завантаження інформацією на одну сторінку. Ретельно відберіть та розташуйте основні елементи інтерфейсу так, щоб вони були легко доступні та не викликали заплутання.
- б) **Чітка навігація:** Розробіть логічну та інтуїтивно зрозумілу структуру навігації. Важливі функції повинні бути легко доступні та не потребувати зайвих кроків для їх виконання.
- в) **Групування та категоризація:** Розділіть інформацію на логічні групи, щоб полегшити її сприйняття. Використовуйте візуальні елементи, такі як рамки, кольори або іконки, для виділення різних секцій.
- г) **Відповідність стандартам дизайну:** Дотримуйтесь загальноприйнятих стандартів дизайну інтерфейсу, які роблять його більш зрозумілим для користувачів.
- д) **Застосування відгуків на дії:** Включіть анімації, повідомлення або зміни кольорів, щоб підтверджувати користувачеві, що його дії виконані успішно.
- е) **Адаптація до різних пристроїв:** Впевніться, що інтерфейс пристосований до різних типів пристроїв (комп'ютер, планшет, смартфон) і коректно відображається на різних розмірах екрану.
- ж) **Тестування з користувачами:** Проводьте тестування прототипів інтерфейсу з реальними користувачами для отримання їхнього фідбеку та виявлення можливих недоліків.
- и) **Надання пояснень та допомоги:** Забезпечте можливість користувачам отримати докладні інструкції або допомогу у використанні системи, якщо вони цього потребують.
- к) **Моніторинг використання:** Використовуйте аналітику для відстеження того, як користувачі взаємодіють з інтерфейсом та виявлення можливих проблемних місць.

л) Постійне вдосконалення: Слід постійно вносити зміни та вдосконалювати інтерфейс відповідно до потреб користувачів та змін в бізнес-процесах.

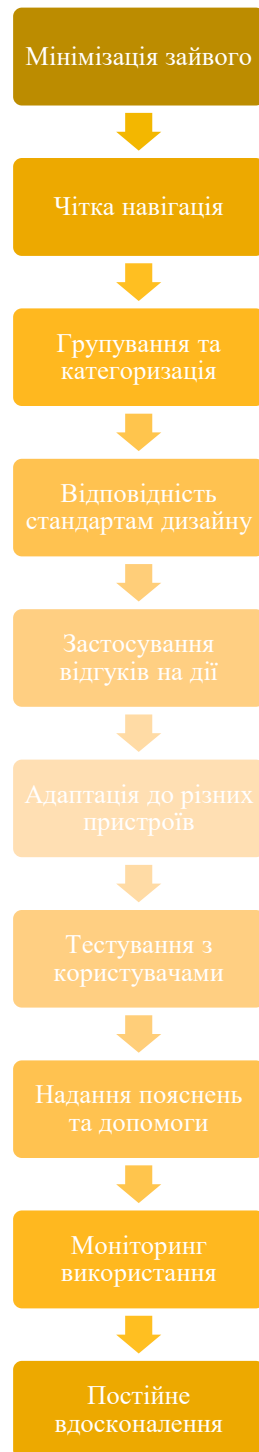


Рисунок 1.1 – Вирішення проблеми інтерфейсу в системах автоматизації медичних закладів

Обмежені можливості редагування даних у системі автоматизації медичних закладів можна подолати за допомогою кількох ключових підходів.

По-перше, важливо визначити рівні доступу для різних користувачів системи. Адміністраторам слід надати повний доступ до редагування та внесення змін у всі дані. Лікарям та медичному персоналу слід надати обмежений доступ, який дозволяє редагувати тільки власні дані та дані пацієнтів, з якими вони працюють.

Додатково, можна впровадити систему аудиту, яка відстежує всі зміни у базі даних. Це дозволить виявляти та вирішувати можливі конфлікти та недорозуміння.

Крім того, важливо надати можливість для внесення коментарів чи підказок під час редагування даних. Це дозволить уточнити або пояснити певні аспекти для інших користувачів.

Необхідно також враховувати можливість відкату змін. Відкат змін - це функція, яка дозволяє повернутися до попереднього стану бази даних в разі виявлення помилки або неправильно внесених даних. Це особливо важливо для уникнення негативних наслідків в результаті помилкових змін.

Нарешті, система повинна враховувати принципи захисту даних та використовувати сучасні методи шифрування для забезпечення конфіденційності та цілісності інформації. Серед сучасних методів шифрування, які використовуються для захисту даних, можна виділити AES (Advanced Encryption Standard) та RSA (Rivest–Shamir–Adleman). AES використовує симетричний ключ для шифрування та розшифрування даних, тоді як RSA використовує асиметричні ключі для цих цілей. Обидва методи вважаються надійними та широко використовуються в інформаційній безпеці.

Для вирішення проблеми гнучкості у відображенні даних в інформаційній системі медичних закладів можна вжити заходи, зображені на структурній схемі 1.2.

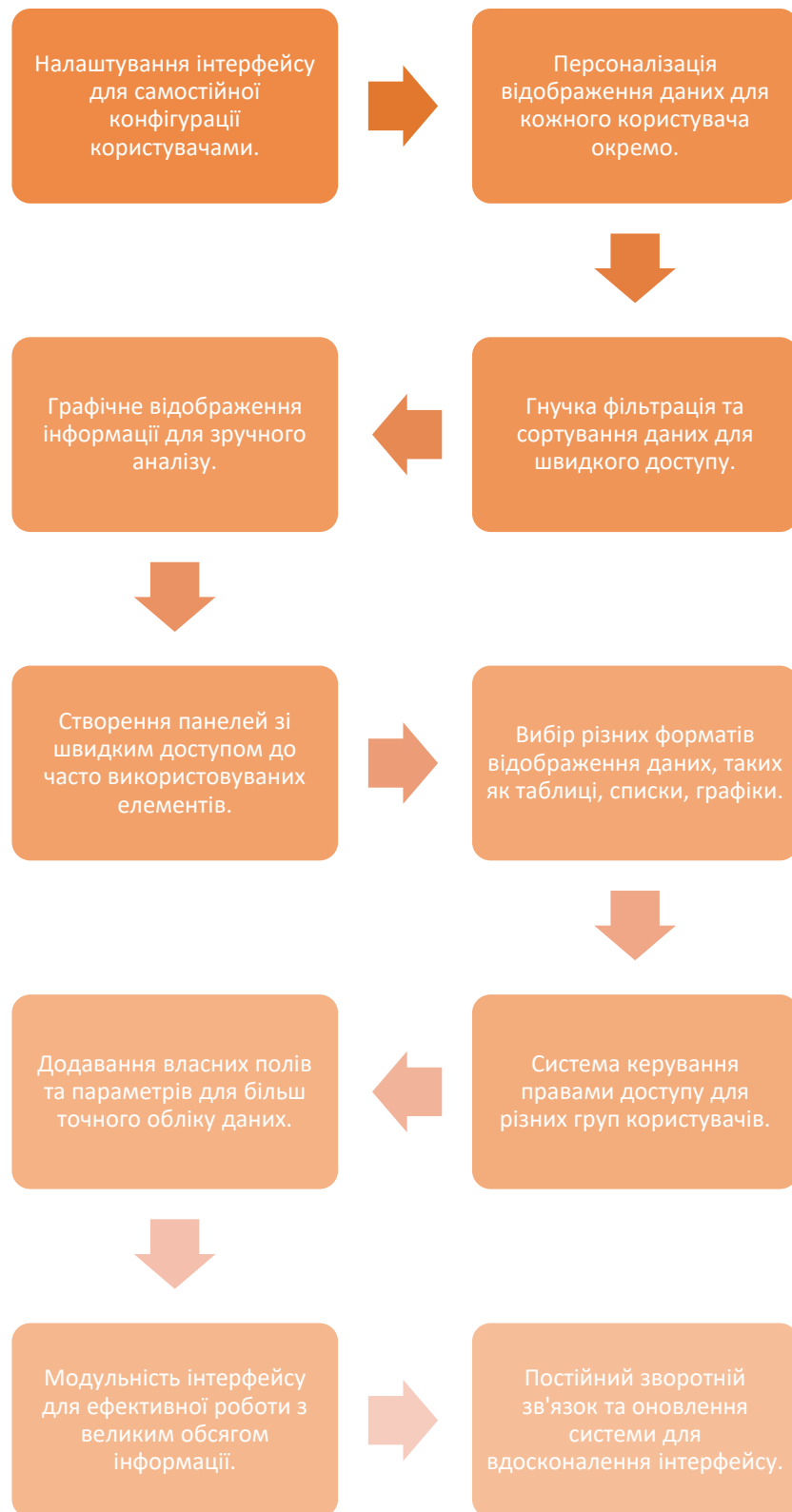


Рисунок 1.2 – Вирішення проблеми гнучкості у відображенні даних в інформаційній системі медичних закладів

Проблема у забезпеченні безпеки даних у медичних інформаційних системах є критичною і вимагає комплексного підходу. Перш за все, важливо встановити надійну систему аутентифікації та авторизації, яка гарантує, що лише авторизовані користувачі отримують доступ до конфіденційних медичних даних. Для цього можна використовувати криптографічні методи, такі як шифрування та хешування паролів, щоб захистити авторизаційні дані.

Другий аспект у розв'язанні проблеми безпеки даних – це впровадження системи аудиту. Завдяки цій системі буде відстежуватися кожна дія користувачів у системі, включаючи перегляд, редагування та видалення даних. Це надає можливість виявити недоречності або недоліки у роботі з даними та прийняти відповідні заходи.

Третім кроком є регулярне проведення аудиту безпеки. До заходів на цьому кроці відноситься аналіз системи на предмет потенційних вразливостей, оновлення заходів захисту та впровадження нових технологій у сфері кібербезпеки. Крім того, важливо організувати навчання та підвищення кваліфікації персоналу з питань безпеки даних.

Не менш важливою складовою вирішення проблеми є встановлення системи резервного копіювання даних. Регулярне створення резервних копій дозволяє відновлювати дані у випадку втрати чи пошкодження основної бази даних.

Розглянемо детальніше проблему недостатньої масштабованості системи. Для початку, необхідно провести аналіз можливостей розширення та вдосконалення системи. Це може включати в себе оцінку та вибір нових технологій, які дозволять покращити продуктивність та масштабованість.

Далі, рекомендується впровадження архітектури, яка дозволяє легко додавати нові функції та модулі у майбутньому. Це може бути реалізовано через використання мікросервісної архітектури або контейнеризації.

Також важливо вдосконалити систему моніторингу та аналізу продуктивності для вчасного виявлення та вирішення можливих проблем. Для цього можна використовувати спеціалізовані інструменти та платформи моніторингу.

Для вирішення проблеми сумісності з іншими системами у контексті інформаційних систем медичних закладів, можна вжити наступні заходи:

- Стандартизація даних: Важливо використовувати загальноприйняті стандарти обміну медичною інформацією, такі як HL7 або FHIR. Це дозволить забезпечити сумісність з іншими медичними системами та обміну даними без перешкод.

- Використання API і інтеграційних платформ: Розробка API (Application Programming Interface) та інтеграційних платформ спрощує підключення і взаємодію з іншими системами. Це дозволяє обмінюватися даними та використовувати функціонал інших систем безперешкодно [1].

- Узгодження інтерфейсів: При інтеграції з іншими системами, важливо узгодити структуру та формат даних, які передаються та отримуються. Це забезпечить вірну інтерпретацію і обробку інформації.

- Тестування та верифікація сумісності: Перед впровадженням інтеграції, необхідно провести тестування взаємодії систем. Це дозволить виявити та виправити можливі конфлікти чи невідповідності.

- Моніторинг та підтримка: Після впровадження інтеграції, важливо здійснювати постійний моніторинг взаємодії систем та надавати технічну підтримку для вирішення можливих проблем.

1.3 Постановка задачі

Постановка задачі на роботу "Програмна система обліку клієнтів стоматологічної поліклініки" передбачає створення комплексного програмного рішення для ефективного управління даними та процесами в стоматологічній установі. Система базується на трьох ключових компонентах: база даних, бізнес-логіка та графічний інтерфейс користувача.

Основою системи виступає реляційна база даних, реалізована на платформі Microsoft SQL Server [2]. Ця база даних забезпечує структуроване зберігання та організацію інформації, пов'язаної з пацієнтами, медичним персоналом, медичними записами та іншими аспектами роботи стоматологічної поліклініки. Реляційна модель даних дозволяє ефективно встановлювати зв'язки між різними сутностями, такими як пацієнти, лікарі, медичні картки, призначення, процедури тощо. Кожна сутність представлена окремою таблицею з визначеними атрибутами та типами даних. Первинні ключі забезпечують унікальну ідентифікацію записів у кожній таблиці, що є критичним для підтримки цілісності даних та ефективного пошуку інформації.

Бізнес-логіка системи відповідає за обробку даних та реалізацію медичних алгоритмів. Цей компонент забезпечує виконання складних обчислень, таких як аналіз показників здоров'я пацієнтів, генерація звітів про стан ротової порожнини, формування рекомендацій щодо лікування на основі історії хвороби. Крім того, бізнес-логіка відповідає за взаємодію з базою даних, формуючи та виконуючи SQL-запити для отримання, оновлення та збереження інформації. Цей рівень абстракції дозволяє відокремити логіку роботи з даними від презентаційного шару, що покращує модульність та масштабованість системи.

Графічний інтерфейс користувача реалізується за допомогою технології Windows Forms, яка надає широкий спектр інструментів для створення інтуїтивно зрозумілого та функціонального інтерфейсу. Інтерфейс включає форми для

введення даних про нових пацієнтів, перегляду та редагування медичних карток, планування візитів, реєстрації процедур та призначень. Важливим елементом інтерфейсу є система візуалізації даних, яка може включати графіки динаміки лікування, діаграми статистики захворювань, календар прийомів тощо. Взаємодія користувача з інтерфейсом ініціює відповідні процеси в бізнес-логіці, які, в свою чергу, призводять до операцій з базою даних [3].

Система передбачає детальну структуру даних для кожної сутності. Наприклад, сутність "Пацієнт" містить такі атрибути як ідентифікатор, ім'я, вік та інформацію про захворювання. Сутність "Лікар" включає ідентифікатор, ім'я, спеціальність та місце роботи. Особлива увага приділяється медичним записам, які зберігаються в сутності "Медична картка" і містять інформацію про пацієнта, лікаря та дату візиту. Система також враховує специфіку стоматологічної практики, включаючи сутності для процедур, розкладу прийомів та технічного персоналу.

Важливим аспектом системи є забезпечення безпеки та конфіденційності даних пацієнтів. Це досягається шляхом впровадження механізмів аутентифікації та авторизації користувачів, а також шифрування чутливої інформації в базі даних.

Реалізація даної системи дозволить значно підвищити ефективність роботи стоматологічної поліклініки, забезпечити швидкий доступ до інформації про пацієнтів, оптимізувати процеси планування та проведення лікування, а також поліпшити якість обслуговування клієнтів. Система надасть можливість генерувати аналітичні звіти, що допоможе в прийнятті управлінських рішень та плануванні розвитку медичного закладу.

2 ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ

Формування вимог до програмної системи обліку клієнтів стоматологічної поліклініки є критичним етапом розробки, який визначає функціональність, ефективність та успішність майбутнього програмного продукту. Процес формування вимог починається з детального аналізу потреб користувачів системи, включаючи адміністративний персонал, лікарів, медсестер та технічних спеціалістів. Важливо враховувати специфіку роботи стоматологічної поліклініки, особливості ведення медичної документації та вимоги законодавства щодо захисту персональних даних пацієнтів.

На основі зібраної інформації формуються функціональні вимоги, які описують конкретні можливості та функції системи. Ці вимоги повинні охоплювати всі аспекти роботи поліклініки, включаючи реєстрацію пацієнтів, ведення медичних карток, планування прийомів, облік процедур та матеріалів, формування звітності. Особлива увага приділяється інтеграції різних модулів системи для забезпечення безперебійного обміну даними між ними.

Не менш важливими є нефункціональні вимоги, які визначають якісні характеристики системи. Сюди входять вимоги до продуктивності (швидкодія, час відгуку), надійності (стабільність роботи, збереження даних), безпеки (захист від несанкціонованого доступу, шифрування даних), масштабованості (можливість розширення функціональності) та зручності використання (інтуїтивно зрозумілий інтерфейс, адаптивність до потреб користувачів).

У процесі формування вимог важливо забезпечити їх чіткість, однозначність та вимірюваність. Кожна вимога повинна бути сформульована таким чином, щоб її можна було перевірити та оцінити після реалізації. Також необхідно встановити пріоритети вимог, визначивши які функції є критичними для першої версії системи, а які можуть бути реалізовані в подальших оновленнях.

В таблиці 2.1 можна побачити функціональні вимоги додатку.

Таблиця 2.1 – Функціональні вимоги додатку

ID	Вимога	Опис	Пріоритет
F1	Реєстрація пацієнтів	Система повинна забезпечувати можливість додавання нових пацієнтів з введенням основних даних (ПІБ, дата народження, контактна інформація)	Високий
F2	Ведення медичних карток	Створення та редагування електронних медичних карток пацієнтів з історією візитів, діагнозами та призначеннями	Високий
F3	Планування прийомів	Функціонал для створення розкладу прийомів лікарів з можливістю запису пацієнтів на конкретний час	Високий
F4	Облік процедур	Реєстрація проведених процедур з можливістю вказання використаних матеріалів та їх кількості	Середній
F5	Фінансовий облік	Ведення обліку оплат за послуги, формування рахунків та звітів про фінансову діяльність	Середній
F6	Генерація звітів	Можливість формування різноманітних звітів (статистика відвідувань, ефективність лікування, фінансові показники)	Низький
F7	Управління персоналом	Облік інформації про співробітників, їх спеціалізації, графіки роботи	Низький
F8	Інтеграція з медичним обладнанням	Можливість імпорту даних з діагностичного обладнання (рентген, сканери)	Низький

В таблиці 2.2 можна побачити нефункціональні вимоги додатку.

Таблиця 2.2 – Нефункціональні вимоги додатку

ID	Вимога	Опис	Пріоритет
NF1	Продуктивність	Система повинна забезпечувати час відгуку не більше 2 секунд для 95% запитів при одночасній роботі до 50 користувачів	Високий
NF2	Надійність	Система повинна бути доступна 99,9% часу роботи поліклініки. Щоденне резервне копіювання даних	Високий
NF3	Безпека	Шифрування персональних даних пацієнтів. Багаторівнева система аутентифікації користувачів. Журналювання всіх дій з чутливими даними	Високий
NF4	Масштабованість	Можливість збільшення кількості користувачів до 200 без значного падіння продуктивності. Підтримка роботи з кількома філіями поліклініки	Середній
NF5	Зручність використання	Інтуїтивно зрозумілий інтерфейс. Час навчання нового користувача не повинен перевищувати 4 години	Середній
NF6	Підтримка та оновлення	Можливість віддаленого оновлення системи без переривання роботи. Технічна підтримка 24/7	Середній
NF7	Сумісність	Підтримка роботи з актуальними версіями Windows (10, 11). Можливість експорту даних у формати PDF, Excel	Низький
NF8	Локалізація	Підтримка української та англійської мов інтерфейсу	Низький
NF9	Відповідність стандартам	Відповідність вимогам GDPR та національним стандартам захисту персональних даних	Високий
NF10	Ергономічність	Адаптивний дизайн для роботи на різних пристроях (ПК, планшети). Підтримка роботи з клавіатурою для швидкого введення даних	Низький

3 АРХІТЕКТУРА ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 UML-проектування ПЗ

Діаграма прецедентів UML є інструментом, що відображає взаємодію між акторами і прецедентами, який є частиною моделі прецедентів, описуючи систему на концептуальному рівні. Прецедент визначає можливості системи, через які користувач може досягти конкретного, вимірюваного і потрібного йому результату. Він відображає один з варіантів використання системи та описує типовий спосіб взаємодії користувача з нею.

Головна мета діаграми прецедентів полягає в описі функціональності та поведінки системи, щоб замовник, кінцевий користувач та розробник могли спільно обговорювати проєктовану або існуючу систему. Для представлення моделі прецедентів на діаграмі використовуються наступні символи:

- Рамки системи: прямокутник з назвою у верхній частині та еліпсами (прецедентами) всередині.
- Актор: символ людини, що позначає набір ролей користувача.
- Прецедент: еліпс з написом, що позначає дії, які виконує система, та призводять до результатів, спостережуваних акторами. Ім'я прецеденту зазвичай пов'язане з неперервним сценарієм, що ілюструє поведінку системи.

З допомогою діаграм прецедентів можна визначити і описати основні можливості системи та взаємодію з користувачами, що сприяє ясному розумінню функціональних вимог до проєкту або існуючої системи (рис.3.1).

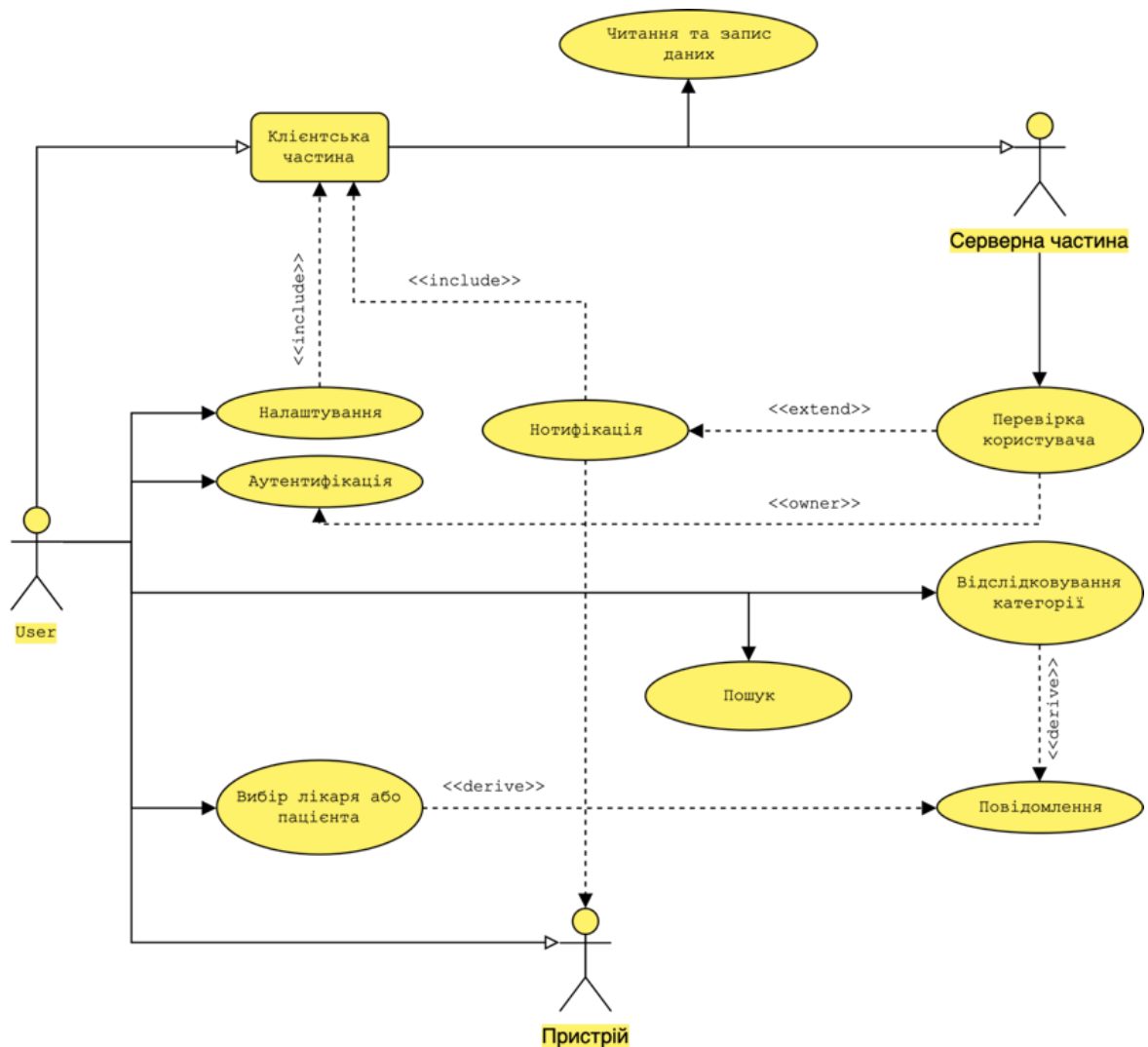


Рисунок 3.1 – Діаграма прецедентів стоматологічної клініки

На діаграмі виділено трьох основних акторів: Користувач, Серверна частина та Пристрій. Користувач взаємодіє з системою через клієнтську частину, яка включає в себе кілька ключових прецедентів.

Основні прецеденти, доступні користувачу, включають "Налаштування", "Автентифікація", "Вибір лікаря або пацієнта", та "Пошук". Прецедент "Налаштування" дозволяє користувачу конфігурувати систему під свої потреби. "Автентифікація" забезпечує безпечний вхід до системи. "Вибір лікаря або

пацієнта" дає можливість обрати конкретного лікаря чи пацієнта для подальшої роботи. "Пошук" дозволяє знаходити необхідну інформацію в системі.

Серверна частина системи відповідає за обробку та зберігання даних. Вона включає прецедент "Перевірка користувача", який пов'язаний з автентифікацією та забезпечує контроль доступу до системи. Також на серверній частині реалізовано прецедент "Читання та запис даних", який відповідає за операції з базою даних.

Важливим елементом системи є прецедент "Нотифікація", який розширюється прецедентом "Повідомлення". Це дозволяє системі інформувати користувачів про важливі події чи зміни.

Прецедент "Відслідковування категорій" пов'язаний з клієнтською частиною та дозволяє класифікувати та організувати інформацію в системі.

Діаграма також показує, що система може взаємодіяти з зовнішніми пристроями, що розширює її функціональність та дозволяє інтегрувати додаткове обладнання.

Діаграми діяльності в контексті інформаційних систем медичних закладів використовуються для візуалізації послідовності дій та процесів, що відбуваються в системі. Ці діаграми дозволяють докладно описати кожен крок у конкретному процесі, зокрема, реєстрацію пацієнтів, назначення та планування обстежень, збереження медичних зображень та інші операції, які відбуваються в медичному закладі.

Діаграми діяльності враховують взаємодію різних учасників системи, таких як реєстратори, лікарі, адміністратори тощо. Вони надають можливість визначити кожен етап процесу, його умови та результати. Крім того, вони допомагають ідентифікувати можливі проблемні ситуації та розробити ефективні стратегії для їх вирішення.

Діаграми діяльності є потужним інструментом для аналізу та оптимізації робочих процесів в медичних закладах. Вони сприяють покращенню продуктивності та ефективності надання медичних послуг, а також допомагають у вдосконаленні організації роботи персоналу та взаємодії між відділами та співробітниками медичного закладу (рис.3.2).

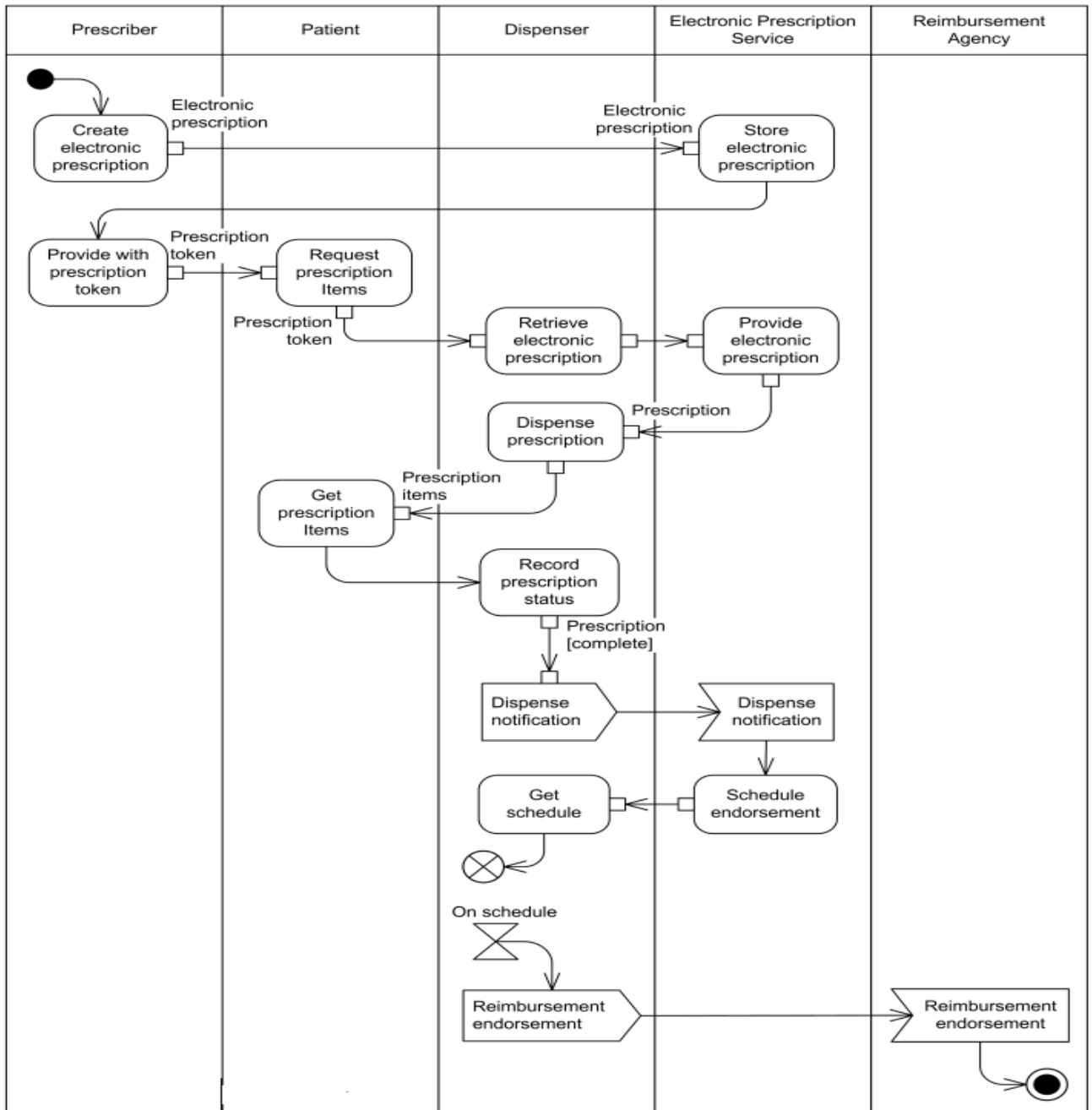


Рисунок 3.2 – Діаграма діяльності медичного закладу

3.2 Проектування архітектури ПЗ

ER (сутність-зв'язок) – це техніка моделювання, яка використовується в інженерії програмного забезпечення та проектуванні баз даних для візуального представлення логічної структури бази даних. У діаграмах ER сутності зображаються у вигляді прямокутників, атрибути – у вигляді овалів, а зв'язки – у вигляді ромбів.

Сутності представляють реальні об'єкти або концепції, атрибути описують властивості цих сутностей, а зв'язки визначають, як сутності пов'язані між собою.

ER-моделювання допомагає ідентифікувати та визначати сутності, атрибути та зв'язки, а також виявляти потенційні проблеми, такі як аномалії даних чи неоднозначності.

Нотація Баркера, або нотація для моделювання систем (англ. Barker notation), є однією з варіацій нотації, що використовується в об'єктно-орієнтованому моделюванні для роботи з діаграмами класів та іншими моделями. Ця нотація була розроблена Девідом Баркером та використовується для опису бізнес-процесів та даних в рамках моделювання [4].

Нотація Баркера зазвичай використовується в контексті Entity-Relationship (ER) моделювання, де вона дозволяє визначити сутності, атрибути та їх взаємозв'язки. У цій нотації сутності зображаються у вигляді прямокутників, а атрибути – у вигляді овалів, які приєднуються до відповідних сутностей. Взаємозв'язки між сутностями позначаються стрілками з вказанням типу взаємозв'язку.

ER-модель нашої інформаційної системи зображено на рисунку 3.3.

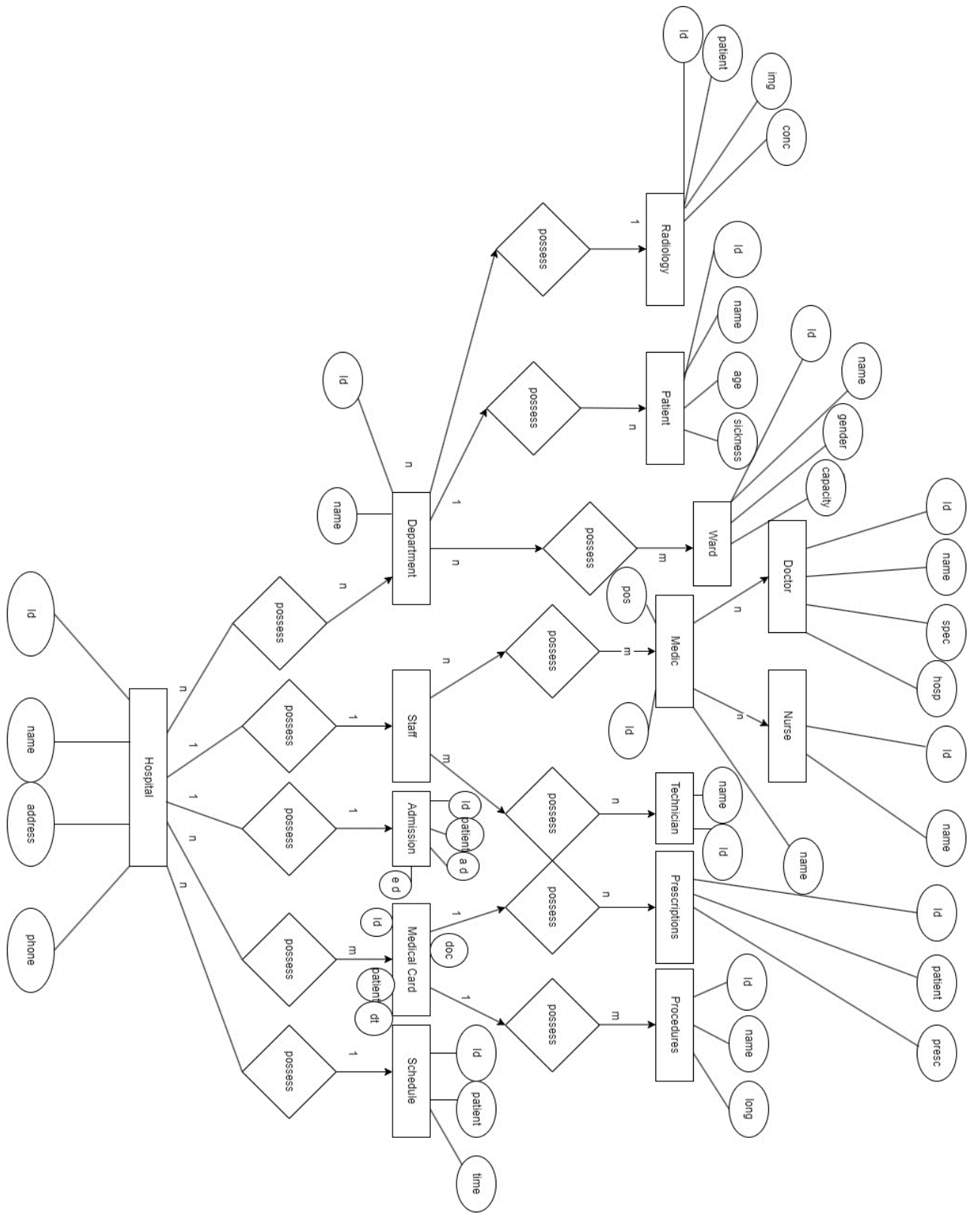


Рисунок 3.3 – ER-модель ІС стоматології за нотацією Баркера

Опишемо основні сутності, наявні в базі даних стоматологічної клініки.

Таблиця 3.1 – Сутності бази даних медичного закладу та їх атрибути

Сутність	Атрибут	Тип даних	Ключ
Admission	Id	INT	Первинний ключ
	Patient	NVARCHAR(MAX)	
	Acceptance date	DATE	
	Extraction date	DATE	
Doctor	Id	INT	Первинний ключ
	name	NVARCHAR(MAX)	
	specialty	NVARCHAR(MAX)	
	hospital	NVARCHAR(MAX)	
Hospital	Id	INT	Первинний ключ
	name	NVARCHAR(MAX)	
	address	NVARCHAR(MAX)	
	phone	NVARCHAR(MAX)	
Medic	Id	INT	Первинний ключ
	name	NVARCHAR(MAX)	
	position	NVARCHAR(MAX)	
Medical card	Id	INT	Первинний ключ
	Patient	NVARCHAR(MAX)	
	Doctor	NVARCHAR(MAX)	
	Datetime	DATETIME	
Nurse	Id	INT	Первинний ключ
	name	NVARCHAR(MAX)	
Prescription	Id	INT	Первинний ключ
	Patient	NVARCHAR(MAX)	
	Prescription	NVARCHAR(MAX)	
Patient			
	Id		Первинний ключ
	name		

	age		
	sickness		
Procedures	Id	INT	Первинний ключ
	name	NVARCHAR(MAX)	
	longation	INT	
Schedule	Id	INT	Первинний ключ
	Patient	NVARCHAR(MAX)	
	Time	DATETIME	
Staff	Id	INT	Первинний ключ
	name	NVARCHAR(MAX)	
	birthdate	DATE	
	gender	BIT	
	home address	NVARCHAR(MAX)	
	phone	NVARCHAR(MAX)	
	joined	DATE	
	education	NVARCHAR(MAX)	
	certification	BIT	
	languages	NVARCHAR(MAX)	
Technician	Id	INT	Первинний ключ
	name	NVARCHAR(MAX)	
Ward	Id	INT	Первинний ключ
	name	NVARCHAR(MAX)	
	patients gender	BIT	
	capacity	INT	

В базі даних медичного закладу використані наступні типи даних:

- a) INT (Integer): Цей тип даних призначений для зберігання цілих чисел. Він може бути використаний для представлення чисел без дробової частини (наприклад, 1, 2, -3, 0).

- б) NVARCHAR(MAX): Цей тип даних використовується для зберігання рядків (тексту). Параметр "MAX" означає, що розмір рядка може бути дуже великим, необмеженим до певного конкретного числа символів.
- в) NCHAR(10): Цей тип даних також призначений для зберігання рядків, але в цьому випадку розмір рядка фіксований і дорівнює 10 символам. Наприклад, якщо ви зберігаєте ім'я департаменту, де ім'я не може бути довше 10 символів.
- г) DATE: Цей тип даних використовується для зберігання дати без часу. Він представляє дату у форматі "рік-місяць-день".
- д) DATETIME: Цей тип даних призначений для зберігання об'єкта, що включає дату та час. Він має формат "рік-місяць-день година:хвилина:секунда".
- е) BINARY(50): Цей тип даних використовується для зберігання бінарних даних, наприклад, зашифрованих або хешованих значень. У даному випадку, вказано розмір поля - 50 байт.
- ж) BIT: Цей тип даних використовується для зберігання булевих значень (1 або 0, true або false).
- и) IMAGE: Цей тип даних використовується для зберігання зображень або інших бінарних даних великого розміру.

3.3 Проектування структури зберігання даних

Реляційна база даних – це база даних, яка використовується для зберігання та організації доступу до взаємопов'язаних елементів інформації. Реляційні бази даних ґрунтуються на реляційній моделі – інтуїтивно зрозумілому, наочному поданні інформації у вигляді таблиць. Кожен рядок у таблиці такої бази даних являє собою запис унікальним ідентифікатором – ключем. Стовпці таблиць мають атрибути даних, що дає змогу встановлювати зв'язки між елементами

даних. Саме цьому, реляційна модель бази даних є найбільш ефективною для вирішення задачі даного курсового проекту.

У Visual Studio, робота з базами даних зазвичай відбувається через вбудовану підтримку для SQL Server та відповідні інструменти. Це дає можливість ефективно управляти та маніпулювати даними.

- а) Створення таблиці: У Visual Studio можна використовувати SQL-запити для створення таблиць, подібно до тих, що ви надали раніше. Система автоматично перевіряє синтаксис та правильність команд перед виконанням.
- б) Використання обмежень: Можна встановити обмеження (constraints) на поля таблиці. Наприклад, визначити, що конкретне поле не може бути NULL або має унікальні значення. Це дозволяє уникнути некоректних даних.
- в) Тригери та збережені процедури: У Visual Studio можна використовувати тригери та збережені процедури для автоматизації операцій з даними. Наприклад, перед вставкою нового запису можна перевірити його коректність.

Використання ідентифікатора (часто автоінкрементного) як первинного ключа є стандартною практикою в базах даних. Це має декілька переваг:

- Унікальність та однозначність: Ідентифікатори завжди унікальні та легко ідентифікують конкретний запис.

- Швидкий доступ до даних: Оскільки ідентифікатори найчастіше є цілими числами, то пошук за ними швидше, ніж за рядковими значеннями.

- Ефективність при об'єднанні таблиць: У разі, коли потрібно об'єднати таблиці, ідентифікатори дозволяють ефективно встановлювати зв'язки між ними.

- Легкість у підтримці: Ідентифікатори полегшують внесення змін та виправлення даних, оскільки вони сталі та не змінюються.

- Масштабованість: У випадку, коли таблиця має велику кількість записів, ефективність роботи з ідентифікаторами набагато вища.

В кожній з наведених нижче таблиць користувач може додавати, редагувати, видаляти та оновляти дані (рис.3.4).

	Id	Patient	Acceptance date	Extraction date
	1	Ivan Georgiev	14.06.2023 00:00:00	19.06.2023 00:00:00
	2	Petar Kolev	22.03.2023 00:00:00	27.03.2023 00:00:00
	3	Dimitar Petrov	14.06.2023 00:00:00	22.06.2023 00:00:00
	4	Ralitza Trifonova	26.09.2023 00:00:00	27.09.2023 00:00:00
	5	Atanas Dimitrov	30.11.2022 00:00:00	15.12.2022 00:00:00

	Id	name	address	phone
▶	1	Varna_Med_Hospital	Tzarigrad Highway, 76	222
	2	Pyrogov_Hospital	Khan Kubrat, 96	333
	3	Hospital_St_Anna	Oborishte, 12	444
	4	Hospital_Borimed	Krivica, 15	555
	5	Hospital_Altamedica	Rakovski, 46	666

	Id	name	position
▶	1	Lidia Filipova	doctor
	2	Hristo Mihailov	technician
	3	Ivan Penev	nurse
	4	Boiana Nedialk...	nurse
	5	Rangel Krustev	doctor

	Id	Patient	Doctor	Datetime
▶	1	Ivan Georgiev	Aleksandar Ivanov	14.06.2023 00:00:00
	2	Petar Kolev	Lidia Filipova	22.03.2023 00:00:00
	3	Dimitar Petrov	Rangel Krustev	14.06.2023 00:00:00
	4	Ralitza Trifonova	Boris Kirov	26.09.2023 00:00:00
	5	Atanas Dimitrov	Vladimir Tzanev	30.11.2022 00:00:00

	Id	name
▶	1	Ivan Penev
	2	Boiana Nedialk...
	3	Todor Kanev
	4	Ivanka Boiadjieva
	5	Tzvetana Kirilova

	Id	Patient	Prescription
▷	1	Ivan Georgiev	injections
	2	Petar Kolev	drips
	3	Dimitar Petrov	insuline
	4	Ralitz Trifonova	special diet
	5	Atanas Dimitrov	azithromycin

	Id	name	longation
▷	1	injections	2
	2	drips	3
	3	physiotherapy	2
	4	radiotherapy	1
	5	massage	2

	Id	Patient	Image	Conclusion
	1	Ivan Georgiev	NULL	arthritis
	2	Petar Kolev	NULL	covid-19
	3	Dimitar Petrov	NULL	pneumonia
	4	Ralitz Trifonova	NULL	diarrhea
	5	Atanas Dimitrov	NULL	covid-19

	Id	Patient	Time
	1	Ivan Georgiev	23.09.2023 00:00:00
	2	Petar Kolev	19.08.2022 00:00:00
	3	Dimitar Petrov	17.05.2023 00:00:00
	4	Ralitz Trifonova	23.10.2022 00:00:00
	5	Atanas Dimitrov	15.03.2023 00:00:00

	Id	name	birthdate	gender	home address	phone	joined	education	certification	languages
▷	1	Lidia Filipova	13.06.1983	female	Kniaz Boris, 82	122	12.09.2020	VNMU	True	English
	2	Hristo Mihailov	27.02.1989	male	Ropotamo,33	543	13.09.2020	med college	False	Bulgarian
	3	Ivan Penev	19.02.1995	male	Slivniza, 46	888	18.09.2019	Varna college	True	German
	4	Boiana Nedialkova	25.03.1998	female	Benkovski, 74	667	27.08.2023	VNMU	True	English

	Id	name
	1	Tzvetana Kirilova
	2	Mihail Jeliaskov
	3	Hristina Todorova
	4	Dobromira Hristozova
	5	Filip Iordanov

Рисунок 3.4 – Комплексний вигляд таблиць даних БД медичного закладу

Create (Створити): Ця операція дозволяє створити новий запис або об'єкт у базі даних. Це може бути новий користувач, товар, завдання або будь-який інший об'єкт, який система потребує.

Read (Читати): Ця операція дозволяє отримати дані з бази даних. Користувач може вибрати конкретний запис або набір записів за певними критеріями, щоб переглянути їх.

Update (Оновити): Ця операція дозволяє змінити існуючі дані в базі даних. Наприклад, можна оновити інформацію про користувача, виправити помилки або змінити статус об'єкта.

Delete (Видалити): Ця операція дозволяє видалити існуючий запис або об'єкт з бази даних. Видалення може бути тимчасовим або назавжди, залежно від налаштувань системи.

При введенні даних які несумісні з встановленим за замовчуванням форматі система видає помилку, а при залишанні комірки даних пустою – пише значення NULL.

4 ОПИС ПРИЙНЯТИХ ПРОГРАМНИХ РІШЕНЬ

Visual Studio є інтегрованою середовищем розробки (IDE), яка надає багатий набір інструментів для розробки, тестування та налагодження ПЗ. У поєднанні з Windows Forms, що є бібліотекою для створення графічного інтерфейсу користувача, вони забезпечують ефективний спосіб взаємодії з користувачем в рамках Windows-платформи. Це особливо важливо для медичних додатків, оскільки вони повинні бути інтуїтивно зрозумілими та легкими у використанні для медичного персоналу.

Microsoft SQL Server є однією з найпоширеніших та надійних систем керування реляційними базами даних [5]. Вона надає високий рівень безпеки, ефективності та масштабованості, що є критичними для забезпечення надійності та швидкодії роботи медичних додатків. Microsoft SQL Server має багатий набір інструментів для адміністрування та моніторингу баз даних, що дозволяє ефективно керувати великим обсягом даних, які накопичуються в медичних установах.

Загалом, комбінація Visual Studio, Windows Forms та Microsoft SQL Server надає рішення, яке поєднує найкращі практики для розробки графічного інтерфейсу користувача та управління базою даних. Це забезпечує надійність, ефективність та зручність у використанні для медичного персоналу, що є критичними аспектами для успішної розробки ПЗ для лікарень.

Механізм роботи бази даних та програмного додатку для системи обліку клієнтів стоматологічної клініки "Daidamony Dental Center" базується на тісній інтеграції реляційної бази даних з графічним інтерфейсом користувача. Система використовує архітектуру клієнт-сервер, де клієнтська частина реалізована як Windows Forms додаток, а серверна частина представлена системою управління базами даних, ймовірно, Microsoft SQL Server.

База даних зберігає всю інформацію про пацієнтів, їхні візити, діагнози, лікування та фінансові операції у взаємопов'язаних таблицях. Основною таблицею є "Patient", яка містить поля для зберігання особистої інформації пацієнтів, такі як ім'я, прізвище, вік, контактні дані та унікальний ідентифікатор. Ця таблиця пов'язана з іншими таблицями, які зберігають дані про медичну історію, проведені процедури та фінансові транзакції.

Програмний додаток взаємодіє з базою даних через спеціалізований шар доступу до даних, який використовує SQL-запити для отримання, додавання, оновлення та видалення інформації. Коли користувач відкриває форму для введення даних нового пацієнта, додаток створює новий запис у таблиці "Patient", генеруючи унікальний ідентифікатор. При збереженні даних виконується SQL-запит INSERT, який додає новий рядок до таблиці.

Для пошуку та відображення інформації система використовує складні SQL-запити, які можуть включати з'єднання кількох таблиць, фільтрацію та сортування даних. Наприклад, коли користувач шукає пацієнтів з певним діагнозом, система формує запит, який об'єднує таблиці пацієнтів та медичних записів, фільтруючи результати за вказаним критерієм.

Особливу увагу варто приділити конструктору запитів, який надає користувачам гнучкий інструмент для формування складних вибірок даних. Цей компонент дозволяє візуально будувати SQL-запити, вибираючи необхідні поля, встановлюючи умови фільтрації та визначаючи порядок сортування. Побудований запит автоматично перетворюється на SQL-код, який потім виконується на сервері бази даних. Результати запиту повертаються до клієнтського додатку та відображаються у зручному для користувача форматі, зазвичай у вигляді таблиці або звіту.

Система також забезпечує механізми для підтримки цілісності даних. Наприклад, при видаленні запису пацієнта, всі пов'язані з ним дані у інших таблицях також видаляються або позначаються як неактивні, залежно від бізнес-

логіки додатку. Це реалізується за допомогою тригерів або каскадних операцій, налаштованих у схемі бази даних.

Для забезпечення швидкодії при роботі з великими обсягами даних, система може використовувати індекси на ключових полях таблиць, що значно прискорює операції пошуку та сортування. Крім того, для оптимізації продуктивності можуть застосовуватися механізми кешування часто використовуваних даних на рівні додатку [6].

Безпека даних забезпечується на кількох рівнях. На рівні бази даних використовуються механізми аутентифікації та авторизації, які обмежують доступ до даних відповідно до ролі користувача. На рівні додатку реалізовано систему логінів, яка контролює доступ до різних функцій системи. Важлива інформація, така як паролі користувачів, зберігається у зашифрованому вигляді.

Опишемо основні команди, що використовуються для створення запитів у рамках маніпулювання даними медичного закладу.

INSERT INTO (ВСТАВИТИ У): Команда **INSERT INTO** використовується для вставки нових записів в таблицю бази даних. Ця операція дозволяє додавати нові дані у вказані стовпці таблиці. Важливо вказати правильні значення для кожного стовпця у відповідності до їх типів даних.

UPDATE (ОНОВИТИ): Команда **UPDATE** використовується для зміни вже існуючих записів в таблиці. Ця команда дозволяє оновити один чи кілька стовпців вибраних записів на нові значення відповідно до вказаної умови.

DELETE FROM (ВИДАЛИТИ З): Команда **DELETE FROM** використовується для видалення одного чи кількох записів з таблиці бази даних. Ця команда може бути застосована з умовою, щоб видалити тільки певні записи.

SELECT (ВИБРАТИ): Команда **SELECT** використовується для вибору даних з таблиці. Її можна доповнити різними опціями, які дозволяють відбирати конкретні стовпці, сортувати, групувати та використовувати агрегатні функції.

JOIN (З'ЄДНАННЯ): Команда JOIN використовується для поєднання даних з двох або більше таблиць за допомогою спільного стовпця. Ця команда дозволяє об'єднати дані, які відповідають певній умові.

GROUP BY (ГРУПУВАТИ ЗА): Команда GROUP BY використовується для групування результатів запиту за певним стовпцем. Це дозволяє використовувати агрегатні функції (наприклад, COUNT, SUM) для обчислення значень в кожній групі.

HAVING: Команда HAVING використовується для фільтрації результатів, які виводяться після виконання GROUP BY. Вона дозволяє вказати умови для відбору груп.

ORDER BY (СОРТУВАТИ ЗА): Команда ORDER BY використовується для сортування результатів за певними стовпцями. Це може бути використано для виведення даних у певному порядку.

COUNT, SUM, AVG, MAX, MIN (ПІДРАХУНОК, СУМА, СЕРЕДНЄ, МАКСИМУМ, МІНІМУМ): Ці агрегатні функції використовуються для виконання різних обчислень над даними. Наприклад, COUNT підраховує кількість записів у вибірці, а AVG обчислює середнє значення.

SUBQUERIES (ВКЛАДЕНІ ЗАПИТИ): Вкладені запити використовуються для вибору даних з іншого запиту. Це дозволяє вибирати дані, які відповідають певній умові, використовуючи результати іншого запиту.

Технологія DataGridView є потужним інструментом у розробці програм, що взаємодіють з базами даних (БД) через графічний інтерфейс. Вона допомагає спростити та зробити більш ефективним процес відображення та редагування даних в програмі.

Основні опції DataGridView, які я використав в програмі:

а) Відображення та Редагування Даних: DataGridView надає зручний та ефективний спосіб відображення даних з таблиць БД у графічному

інтерфейсі. Кожен рядок таблиці відображає окремий запис, а кожний стовпець - окремий атрибут запису.

б) Синхронізація з БД: DataGridView дозволяє легко зв'язувати дані з таблиць БД. Це означає, що зміни, внесені в DataGridView, автоматично оновлюють відповідні дані у базі даних.

в) Фільтрація та Сортування: За допомогою DataGridView можна легко фільтрувати та сортувати дані. Це дозволяє користувачам швидко знаходити необхідну інформацію великих наборів даних.

г) Редагування На Місці (In-Place Editing): DataGridView надає можливість редагувати дані прямо в таблиці. Це дозволяє користувачам зручно вносити зміни без потреби відкривати окремі вікна редагування.

д) Вибірка та Вставка Даних: Користувач може вибирати рядки для подальшого аналізу або видалення. Також можна вставляти нові записи безпосередньо у DataGridView, що автоматично оновить відповідну таблицю в БД.

ж) Навігація: DataGridView надає зручні засоби навігації між різними записами та сторінками даних. Користувач може легко перемикатися між різними сторінками довгих списків.

и) Забезпечення Доступу до Даних в Реальному Часі: Оскільки DataGridView автоматично оновлює дані з бази даних при їх зміні, користувачі можуть бачити оновлену інформацію в реальному часі.

к) Обробка Подій: DataGridView надає можливість обробляти різні події, такі як вибір рядка, редагування комірки, додавання або видалення запису. Це дає можливість виконувати додаткові дії при взаємодії з даними.

На даних рисунках бачимо реєстраційну форму користувача, а також набір вікон з таблицями для редагування та введення даних, а також можливістю виконання запитів. Конструювати запити можна з використанням штучного інструментарію Visual Studio.



Рисунок 4.1 – Головна сторінка програмного додатку

Головна сторінка програми Daidamony Dental Center представляє собою лаконічний інтерфейс з логотипом клініки та набором функціональних кнопок, які забезпечують швидкий доступ до основних операцій системи. Користувачі мають можливість додавати нових пацієнтів, здійснювати пошук у базі даних, редагувати наявну інформацію, переглядати список пацієнтів, керувати ідентифікаторами та отримувати доступ до фінансової інформації, що робить навігацію по системі інтуїтивно зрозумілою та ефективною для щоденного використання медичним персоналом (рис.4.1).

Daidamony Dental Center

Patient's information

First Name Last Name Age ID

Date (yyyy-mm-dd) Tel. Occupation Address

Add

Update

Delete

Clear

Patient's ID Date (yyyy-mm-dd) Medical History Diagnosis

Tooth Operation Cost Paid

Insert

Update

Clear

Delete




Рисунок 4.2 – Сторінка додавання інформації про пацієнта

Сторінка додавання інформації про пацієнта містить розширений набір полів для введення детальних даних, включаючи особисту інформацію, контактні дані, медичну історію та деталі лікування. Форма дозволяє вносити ім'я, прізвище, вік, ідентифікаційний номер пацієнта, дату візиту, телефон, професію та адресу, а також містить поля для запису медичної історії, діагнозу, інформації про конкретний зуб, проведену операцію, вартість послуг та стан оплати. Додатково, на цій сторінці розміщені кнопки для додавання нового запису, оновлення існуючої інформації, видалення запису та очищення форми, що забезпечує повний цикл управління даними пацієнта в рамках єдиного інтерфейсу (рис.4.2-4.4).

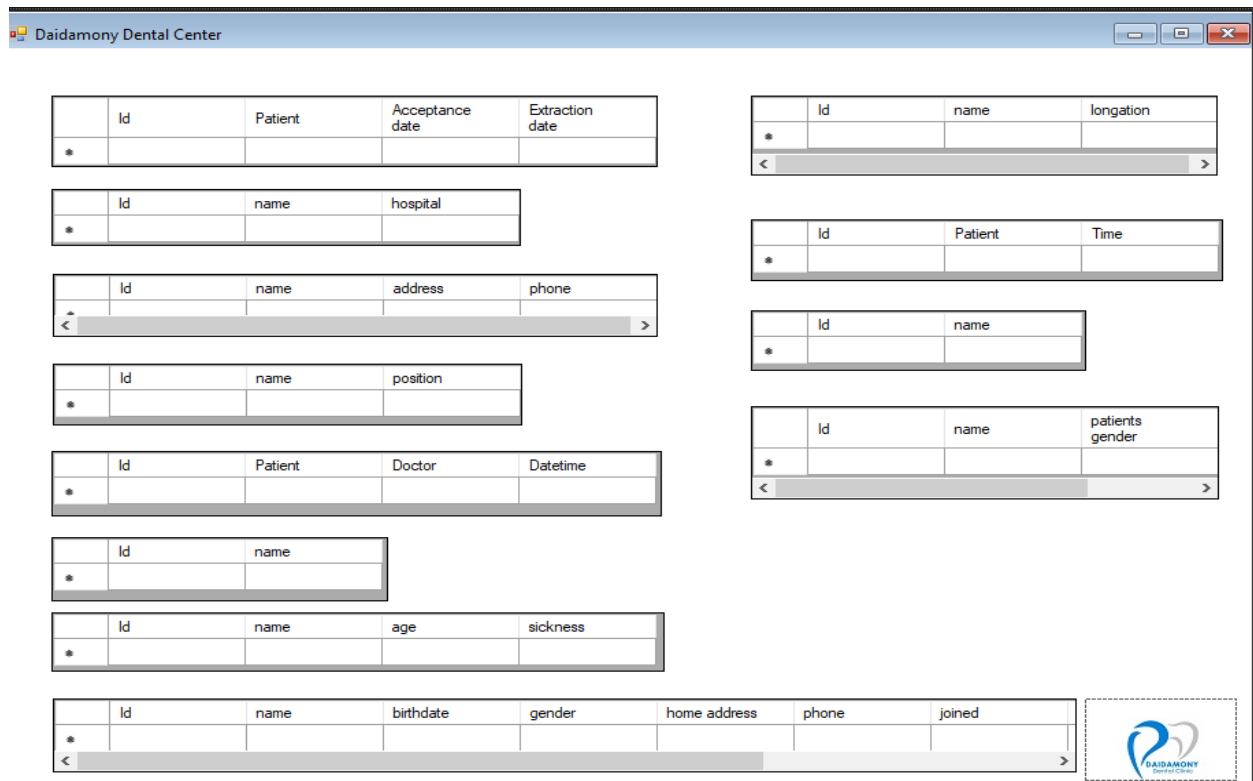


Рисунок 4.3 – Таблиці з інформацією в БД

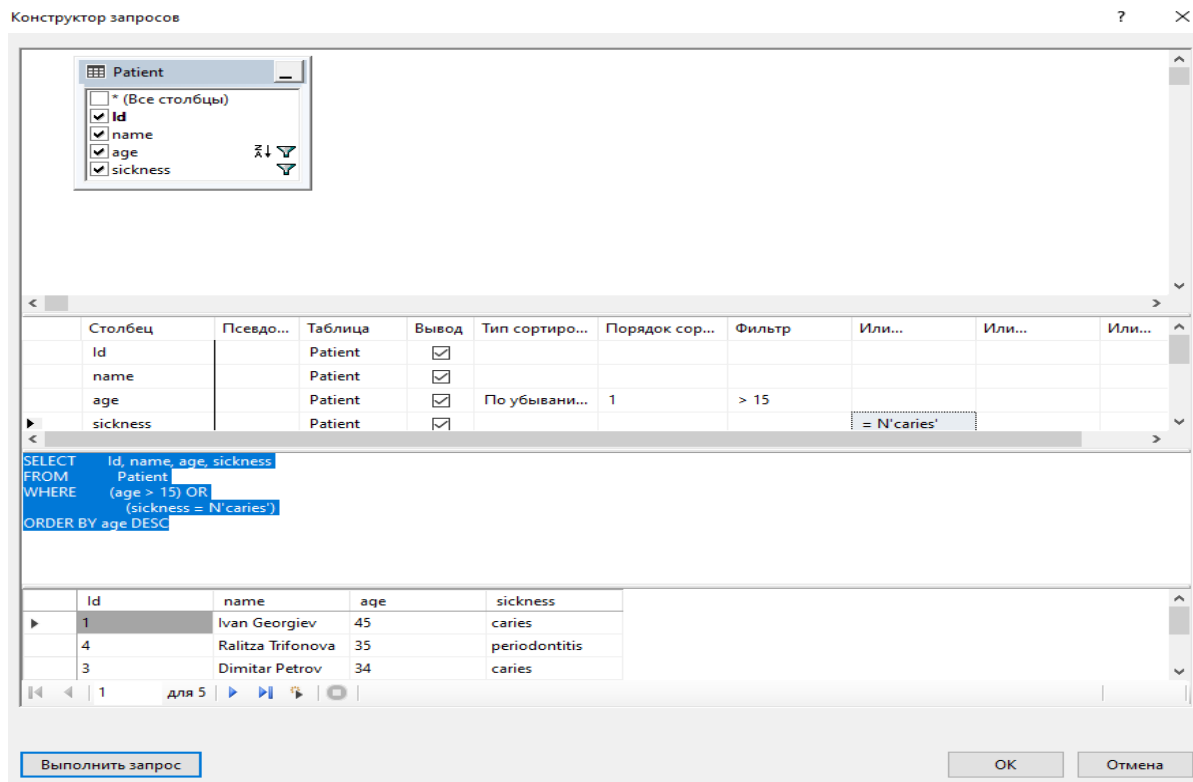


Рисунок 4.4 – Приклад конструювання запиту

5 ТЕСТУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Необхідно описати, як проводилось функціональне тестування бази даних медичного закладу.

а) Підготовка Тестового Середовища:

Початковий етап тестування включає підготовку тестового середовища. Спершу створюється тестова база даних, яка відповідає реальній структурі, але має обмежений обсяг записів. Це забезпечує збереження реалістичності, але уникнення можливих ризиків втрати чутливої інформації.

Після цього встановлюється тестовий сервер та графічний інтерфейс, які імітують реальне робоче середовище. Це може бути здійснено за допомогою віртуальних середовищ, контейнерів або інших інструментів для ізоляції тестового середовища від продуктивного.

б) Виконання Тестових Сценаріїв:

Зміна Даних: Даний крок передбачає спроби внесення змін до різних типів даних через графічний інтерфейс. Це включає в себе зміни особових даних пацієнта, призначень, результатів обстежень тощо. Після кожної зміни проводиться перевірка, чи відображаються зміни коректно та чи вони правильно оновлюються в базі даних. **Додавання Нових Записів:** Додавання нових пацієнтів, лікарів, відділень, призначень та іншої інформації через графічний інтерфейс є необхідним кроком тестування. Після кожного додавання проводиться перевірка, чи нові дані відображаються у відповідних таблицях та чи коректно вони додаються до бази даних. **Видалення Записів:** Видалення обраних записів (пацієнтів, лікарів, призначень тощо) через графічний інтерфейс важливо для перевірки коректності обробки видалень. Після кожного видалення перевіряється, чи відповідні дані видаляються з таблиць та бази даних. **Виконання Запитів:** Тестування системи на відповідність її реакції на різноманітні запити до бази даних включає в себе введення та виконання

різноманітних запитів, наприклад, вибірку всіх пацієнтів з певними характеристиками або створення звіту про надходження пацієнтів за конкретний період. Обробка Помилки та Виключень: Важливо перевірити, як система реагує на неправильні введення, невірні запити або внутрішні помилки. Це включає в себе перевірку повідомлень про помилки, коректність відображення, а також чи враховано та оброблено неправильні вхідні дані.

в) Тестування на SQL-Ін'єкції:

SQL-ін'єкції є потенційно небезпечними атаками, де зловмисник намагається вставити SQL-код у введені дані, з метою виконання небажаних запитів до бази даних. Для тестування цієї уразливості, намагаємося ввести шкідливий SQL-код через поля введення.

Приклад SQL-Ін'єкції:

Припустимо, є поле для пошуку пацієнта за ім'ям, і користувач вводить ' OR '1'='1'--'. У разі недостатнього захисту, це може призвести до вибору всіх записів у базі даних, оскільки умова '1'='1' завжди виконується.

Заходи Захисту:

Використання Параметризованих Запитів:

Приклад (C#):

```
SqlCommand cmd = new SqlCommand("SELECT * FROM Patients WHERE
Name = @Name", connection);
```

```
cmd.Parameters.AddWithValue("@Name", userInput);
```

Екранування Введених Даних:

Приклад (C#):

```
string safeInput = userInput.Replace("'", "");
```

Загальне тестування підтвердило повну коректність і працездатність системи, враховуючи нюанси взаємодії баз даних з графічним інтерфейсом.

6 ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Впровадження програмного забезпечення для обліку клієнтів стоматологічної поліклініки "Daidamony Dental Center" є комплексним процесом, який вимагає ретельного планування та поетапної реалізації. Цей процес починається з детального аналізу потреб клініки, існуючих бізнес-процесів та технічної інфраструктури. На основі цього аналізу розробляється план впровадження, який враховує особливості роботи конкретного медичного закладу.

Перший етап впровадження передбачає підготовку технічної бази. Це включає налаштування серверного обладнання, встановлення необхідного програмного забезпечення, включаючи систему управління базами даних та саму програму обліку клієнтів. Важливим аспектом є забезпечення надійного та безпечного мережевого з'єднання між робочими станціями та сервером бази даних.

Наступним кроком є налаштування програмного забезпечення відповідно до специфіки роботи клініки. Це може включати кастомізацію інтерфейсу, налаштування шаблонів звітів, встановлення параметрів безпеки та прав доступу для різних категорій користувачів. На цьому етапі також проводиться інтеграція з існуючими системами, якщо такі є, наприклад, з бухгалтерським програмним забезпеченням або системою електронного документообігу.

Особлива увага приділяється міграції даних з попередніх систем, якщо такі використовувалися. Цей процес вимагає ретельного планування та виконання, щоб забезпечити цілісність та актуальність перенесеної інформації. Дані можуть включати історії хвороб пацієнтів, фінансові записи, графіки прийомів та іншу важливу інформацію. Після міграції обов'язково проводиться верифікація даних для виявлення та виправлення можливих помилок.

Важливим етапом впровадження є навчання персоналу роботі з новою системою. Це включає проведення тренінгів для різних категорій користувачів: адміністраторів, лікарів, медсестер, технічного персоналу. Навчання охоплює всі аспекти роботи з програмою: від базових операцій введення даних до складних аналітичних запитів та генерації звітів. Для забезпечення ефективності навчання розробляються детальні інструкції та проводяться практичні заняття.

Після завершення основних етапів впровадження проводиться тестова експлуатація системи. Під час цього періоду користувачі працюють з програмою в реальних умовах, але під пильним наглядом технічних спеціалістів. Це дозволяє виявити можливі проблеми або незручності в роботі системи та оперативно їх усунути. Тестова експлуатація також дає можливість користувачам адаптуватися до нової системи та надати зворотний зв'язок для її покращення.

Паралельно з тестовою експлуатацією проводиться оптимізація роботи системи. Це може включати налаштування продуктивності бази даних, оптимізацію SQL-запитів, покращення швидкодії інтерфейсу користувача. На цьому етапі також можуть вноситися корективи в бізнес-процеси клініки для максимально ефективного використання можливостей нової системи.

Важливим аспектом впровадження є забезпечення безпеки даних. Це включає налаштування системи резервного копіювання, впровадження політик безпеки, налаштування системи моніторингу та аудиту дій користувачів. Особлива увага приділяється захисту персональних даних пацієнтів відповідно до вимог законодавства.

Після успішного завершення тестової експлуатації та внесення необхідних коректив система вводиться в повноцінну експлуатацію. На даному етапі важливо забезпечити постійну технічну підтримку користувачів, оперативне вирішення можливих проблем та регулярне оновлення системи.

ВИСНОВКИ

Розроблена програмна система надає стоматологічній клініці зручний та ефективний інструмент для управління пацієнтами та медичним персоналом. Система дозволяє легко реєструвати та відстежувати основні дані пацієнтів, а також призначення, проведення процедур та контролю лікування. Можливість швидкого доступу до медичних карток пацієнтів та їх історії сприяє збільшенню продуктивності роботи медичного персоналу.

Система надає можливість надання лікарських призначень та контролю над видачею рецептів. Стоматологи можуть легко створювати та відстежувати призначення, а також контролювати видачу ліків. Це дозволяє забезпечити правильне лікування пацієнтів та уникнути непорозумінь у видачі медикаментів.

Програмна система дозволяє ефективно планувати та відстежувати стоматологічні заходи, призначення, обстеження та процедури. Система надає можливість контролювати навантаження медичного персоналу та забезпечувати своєчасну надання послуг пацієнтам.

Система дозволяє збирати та аналізувати дані про здоров'я пацієнтів. Це надає можливість моніторити стан пацієнтів, виявляти та передбачати тенденції в їх здоров'ї. Аналітичні звіти сприяють прийняттю важливих медичних рішень та плануванню лікування.

Розроблена система включає в себе заходи забезпечення безпеки даних, включаючи захист від SQL-ін'єкцій, використання параметризованих запитів та екранування введених даних. Розроблена програмна система для стоматологічної клініки з наданим набором функціональних можливостей виявилася дієвим інструментом для управління пацієнтами, медичним персоналом та медичною документацією. Її впровадження в практику дозволяє оптимізувати роботу стоматологічної клініки, покращити якість надання медичних послуг та забезпечити надійність та безпеку обробки медичних даних.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Як відкрити захищену базу даних Access через ADO за допомогою OLE. URL: <https://support.microsoft.com/en-us/kb/191754>
2. PHP + MySQL. URL: <http://sites.znu.edu.ua/webprog/lect/1222.ukr.html>
3. Берко А. Ю., Верес О. М., Пасічник В. В. Системи баз даних та знань. Книга 2. Системи управління базами даних та знань: навч. посіб. Львів: "Магнолія-2006", 2012. 584 с.
4. Введення в програмування Access. URL: <http://office.microsoft.com/ru-ru/access-help/НА010341717.aspx?CTT=1#BMlearnaboutmacros> (дата звернення: 22.12.2018).
5. Гайна Г. А. Основи проектування баз даних: навч. посіб. К.: Кондор, 2008. 200 с.
6. ДСТУ 2874-94. Системи оброблення інформації. Бази даних. Терміни та визначення. К.: Держстандарт України, 1995. 29 с.

ДОДАТОК А

Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ



Дата звіту 7/14/2024

Дата редагування ---



Звіт не був оцінений.

метадані

Заголовок

2024_Б_ПІ_ПЗПп-22-1_Веленський_А_В_Пояснювальна_записка

Автор

Веленський Андрій Вікторович

Науковий керівник / Експерт

Вадим Юрійович Нечволод

підрозділ

Харківський національний університет радіоелектроніки

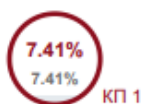
Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв	B	0
Інтервали	A→	0
Мікропробіли		4
Білі знаки	B	0
Парафрази (SmartMarks)	a	19

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



25

Довжина фрази для коефіцієнта подібності 2

7626

Кількість слів

60620

Кількість символів

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

ДОДАТОК Б
Слайди презентації

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Кваліфікаційна робота бакалавра

**Програмна система обліку клієнтів
стоматологічної поліклініки**

1

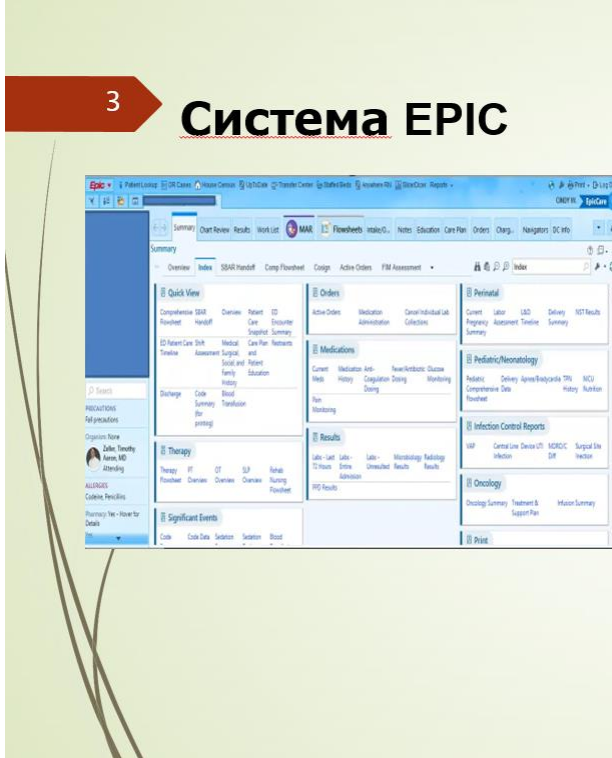
Виконав:
студент гр. ПЗПП-22-1
Веленський А. В.

Науковий керівник:
Вечур О.В.

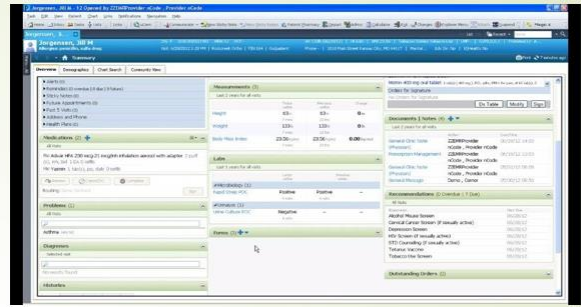
2

Мета роботи

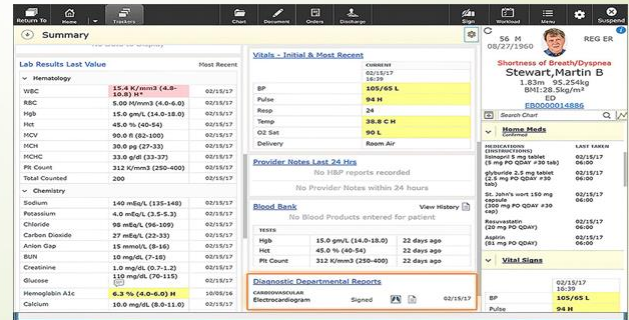
- Автоматизація діяльності стоматологічної поліклініки
 - облік пацієнтів
 - облік лікарів
 - облік рецептів та прийому
 - додавання та редагування даних
 - інтеграція зовнішніми системами



Система CERNER



Система Meditech



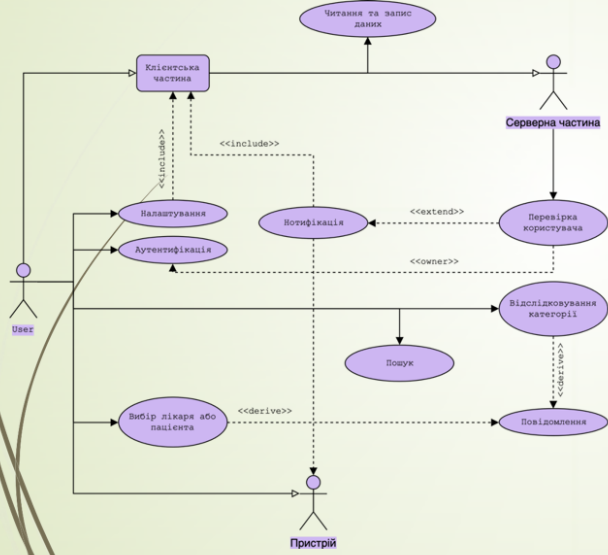
4

Аналіз проблем в існуючих системах

Проблема	Опис
Низька ефективність інтерфейсу	В існуючих системах часто відзначається неінтуїтивний інтерфейс, що ускладнює роботу медичного персоналу та може викликати заплутаність при спробі користуватися системою.
Обмежені можливості редагування даних	Адміністраторам може бути важко вносити необхідні зміни в інформацію про пацієнтів та призначені процедури через обмежені можливості редагування, що може призвести до недооцінки необхідного лікування та ускладнити контроль за станом пацієнтів.
Недостатня гнучкість у відображенні даних	Існуючі системи можуть мати обмежену здатність коректно відображати та аналізувати дані у різних форматах, що в свою чергу може призвести до труднощів у розумінні та використанні важливої інформації.
Проблеми у забезпеченні безпеки даних	Деякі системи можуть виявляти вразливість перед загрозами з боку несанкціонованих доступів, що може призвести до витоку конфіденційної інформації.
Недостатня масштабованість системи	У випадку збільшення потоку пацієнтів чи розширення медичного закладу, існуючі системи можуть виявитися неефективними та недостатньо гнучкими у роботі з великим обсягом даних.
Проблеми сумісності з іншими системами	Виникнення труднощів у сумісності з іншими медичними програмами та обладнанням може обмежити можливості системи та суттєво ускладнити використання сучасних технологій у медичній практиці.

5

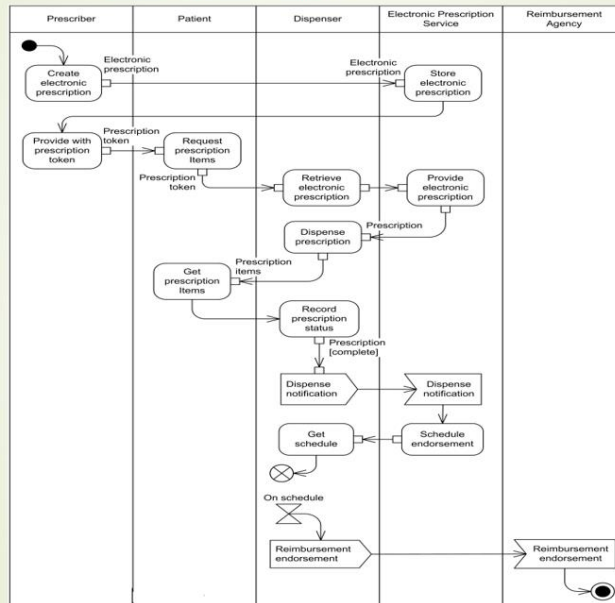
Постановка задачі та опис системи



ID	Вимога	Опис	Пріоритет
F1	Реєстрація пацієнта	Система повинна забезпечувати можливість додавання нових пацієнтів з введенням основних даних (ПІБ, дата народження, контактна інформація)	Високий
F2	Ведення медичних карток	Створення та редагування електронних медичних карток пацієнтів з історією візитів, діагнозами та призначеннями	Високий
F3	Планування прийомів	Функціонал для створення розкладу прийомів лікарів з можливістю запису пацієнтів на конкретний час	Високий
F4	Облік процедур	Реєстрація проведення процедур з можливістю вказання використаних матеріалів та їх кількості	Середній
F5	Фінансовий облік	Ведення обліку оплат за послуги, формування рахунку та звітів про фінансову діяльність	Середній
F6	Генерація звітів	Можливість формування рівномірних звітів (статистика відвідувань, ефективність лікування, фінансові показники)	Низький
F7	Управління персоналом	Облік інформації про співробітників, їх спеціалізації, графік роботи	Низький
F8	Інтеграція з медичним обладнанням	Можливість імпорту даних діагностичного обладнання (рентген, сканери)	Низький

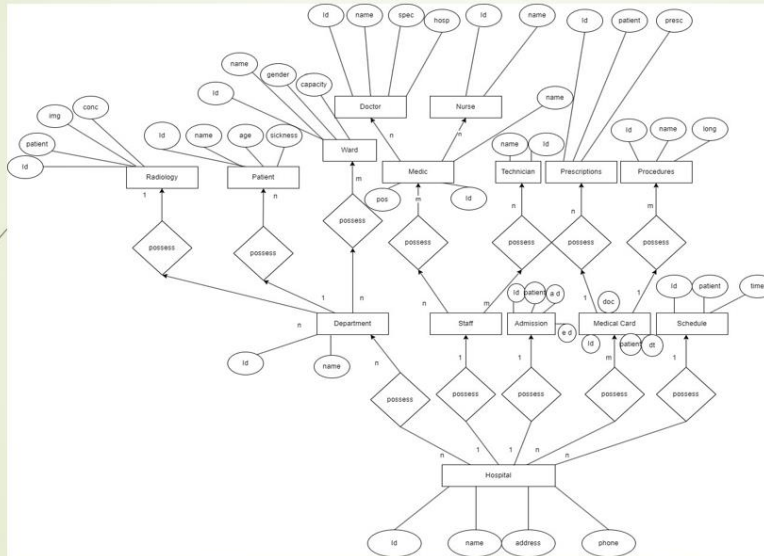
6

Діаграма діяльності



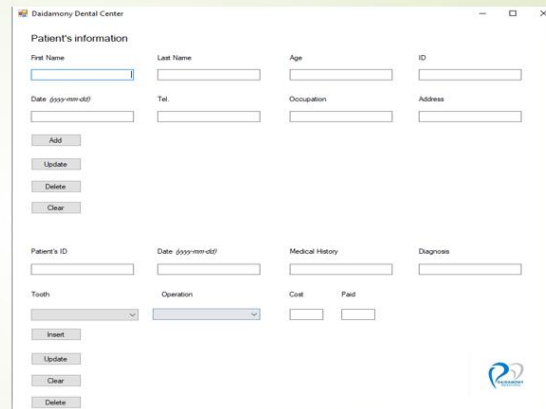
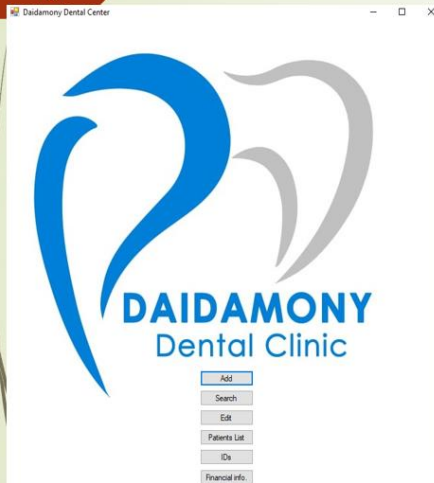
7

Логічна модель



8

Інтерфейс користувача



9

Впровадження

Впровадження програмного забезпечення для обліку клієнтів стоматологічної поліклініки "Daidamony Dental Center" є комплексним процесом, який вимагає ретельного планування та поетапної реалізації. Цей процес починається з детального аналізу потреб клініки, існуючих бізнес-процесів та технічної інфраструктури. На основі цього аналізу розробляється план впровадження, який враховує особливості роботи конкретного медичного закладу.

Перший етап впровадження передбачає підготовку технічної бази. Це включає налаштування серверного обладнання, встановлення необхідного програмного забезпечення, включаючи систему управління базами даних та саму програму обліку клієнтів. Важливим аспектом є забезпечення надійного та безпечного мережевого з'єднання між робочими станціями та сервером бази даних.

Наступним кроком є налаштування програмного забезпечення відповідно до специфіки роботи клініки. Це може включати кастомізацію інтерфейсу, налаштування шаблонів звітів, встановлення параметрів безпеки та прав доступу для різних категорій користувачів. На цьому етапі також проводиться інтеграція з існуючими системами, якщо такі є, наприклад, з бухгалтерським програмним забезпеченням або системою електронного документообігу.

10

Висновки

Розроблена програмна система надає стоматологічній клініці зручний та ефективний інструмент для управління пацієнтами та медичним персоналом. Система дозволяє легко реєструвати та відстежувати основні дані пацієнтів, а також призначення, проведення процедур та контролю лікування. Можливість швидкого доступу до медичних карток пацієнтів та їх історії сприяє збільшенню продуктивності роботи медичного персоналу.