

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет комп'ютерних систем і автоматики
(повне найменування інституту, факультету)

Кафедра комп'ютерних систем управління
(повна назва кафедри)

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Автоматизована система управління стоматологічною клінікою. Частина 2. Розробка модуля ведення пацієнтів»

Виконав: студент 2-го курсу, групи 2
АКІТ-20м
спеціальності 151 – Автоматизація та
комп'ютерно-інтегровані технології
(шифр і назва спеціальності)

Сергій Серeda
(ім'я та прізвище)

Керівник: к.т.н., доцент каф. КСУ

Марія Юхимчук
(ім'я та прізвище)

«15» _____ 12 _____ 2021 р.

Опонент: к.т.н., доцент каф. АІТ

Марія Барабан

(ім'я та прізвище)

«16» _____ 12 _____ 2021 р.

Допущено до захисту
Завідувач кафедри КСУ
д.т.н., проф.

Володимир Дубовой
(ім'я та прізвище)

«17» _____ 12 _____ 2021 р.

ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних систем та автоматики

Кафедра комп'ютерних систем управління

Рівень вищої освіти II-й (магістерський)

Галузь знань – 15 Автоматизація та приладобудування

Спеціальність 151 « Автоматизація та комп'ютерно-інтегровані технології

Освітньо-професійна програма Інтелектуальні комп'ютерні системи

ЗАТВЕРДЖУЮ

Завідувач кафедри КСУ

д.т.н., проф. В.М. Дубовой

« 01 » 10 2021 року

З А В Д А Н Н Я НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Середі Сергію Яновичу

(прізвище, ім'я, по батькові)

1. Тема магістерської кваліфікаційної роботи «Автоматизована система управління стоматологічною клінікою. Частина 2. Розробка модуля ведення пацієнтів.»

керівник магістерської кваліфікаційної роботи доцент кафедри КСУ Юхимчук Марія Сергіївна

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затвержені наказом вищого навчального закладу від “24” 09 2021 р. № 277.

2. Строк подання студентом магістерської кваліфікаційної роботи 10.12. 2021 р.

3. Вихідні дані до магістерської кваліфікаційної роботи підтримка ОС – Windows, Ubuntu; максимальний час завантаження – 6 с; авторизація та реєстрація користувачів – так; максимальна кількість запитів до системи – до 1000 шт/с; мови графічного інтерфейсу– українська, англійська; використання кодування під час передачі даних – так.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) вступ, аналіз необхідності та актуальності впровадження автоматизованих систем управління в стоматологічних клініках, дослідження програмних рішень систем автоматизації стоматології, проектування модуля ведення пацієнтів автоматизованої системи управління стоматологічною клінікою, розробка та тестування працездатності модуля ведення пацієнтів автоматизованої системи управління стоматологічною клінікою.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) UML-діаграма обробки даних при реєстрації, UML-діаграма обробки даних при авторизації, UML-діаграма обробки даних при проведенні пошуку пацієнта, UML-

діаграма редагування профілю, вигляд екрану розробленої прикладної програми
– 6 шт.

6. Консультанти розділів магістерської кваліфікаційної роботи

Розділ змістової частини роботи	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний розділ	доцент кафедри ЕПВМ , доцент к.е.н Кавецький В.В		

Дата видачі завдання “ 01 ” 10 2021 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк виконання етапів роботи	Примітка
1	Дослідження актуальності поставленої задачі	04.10.2021р.	
2	Дослідження видів процесів та аналіз завдань, що підлягають автоматизації у сфері надання стоматологічних послуг	02.10.2021р.	
3	Аналіз функцій автоматизованих систем управління стоматологічними клініками та порівняння програм аналогів	13.10.2021р.	
4	Розробка структури програмного забезпечення системи	03.11.2021р.	
7	Оформлення пояснювальної записки, графічного матеріалу і презентації	08.12.2021р.	
8	Графічні матеріали: UML-діаграма обробки даних при реєстрації UML-діаграма обробки даних при авторизації UML-діаграма обробки даних при проведенні пошуку пацієнта UML-діаграма варіантів використання вигляд екранів розробленого web-додатку UML-діаграма редагування профілю	01.12.2021 р. 01.12.2021 р. 01.12.2021 р. 01.12.2021 р. 03.12.2021 р. 03.12.2021 р.	
9	Захист МКР	14.12.2021 р.	

Студент _____ Сергій Середа

(Підпис)

Керівник роботи _____ Марія Юхимчук

(Підпис)

АНОТАЦІЯ

В даній магістерській дипломній роботі розглянуто автоматизування ведення пацієнтів в стоматологічній клініці. В ході виконання роботи було створено власні алгоритми, які являються основною частиною опрацювання даних пацієнтів й самого персоналу клініки. Програмне забезпечення реалізовано на мові програмування Java в середовищі розробки IntelliJIdea, використовуючи систему управління базами даних MySQL Workbench, а також пакети даних з розширенням fxml. Зроблено теоретичну й технічну частину роботи. Було здійснено оптимізацію, й тестування розробленого продукту.

ABSTRACT

In this master's thesis, considered the automation of patient management in a dental clinic. In the course of the work, own algorithms were created, which are the main part of the processing of data of patient and the clinic staff. The software is implemented in the Java programming language in the IntelliJIdea development environment, using the MySQL Workbench database management system, and data packages with the fxml extension. The theoretical and technical part of the work is done. The developed product was optimized and tested.

ЗМІСТ

ВСТУП.....	7
1. ОГЛЯД СТАНУ ПРОБЛЕМИ ТА МЕТОДИ ЇЇ РІШЕННЯ	9
1.1 Сутність об'єкту дослідження	9
1.2 Аналіз існуючого програмного забезпечення, що виконує подібне завдання.....	11
1.3 Критерії роботи систем для роботи з базами даних MySQL.....	16
1.4. Процес ведення пацієнтів.....	16
1.5 Постановка задачі.....	17
1.6 Висновки до розділу	20
2. ДОСЛІДЖЕННЯ ФУНКЦІОНАЛЬНИХ МОЖЛИВОСТЕЙ ПРОГРАМНИХ РІШЕНЬ	19
2.1 Аналіз роботи прикладного програмного забезпечення з базами даних.....	19
2.2 Дослідження функціональних можливостей сучасних програмних рішень.....	20
2.2.1 MySQL workbench.....	21
2.2.2 DBManager.....	22
2.2.3 MyDB Studio	23
2.2.4 SQLyog Ultimate.....	24
2.3 Порівняння існуючих рішень	25
3. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ВЕДЕННЯ ПАЦІЄНТІВ.....	28
3.1 Модулі й структура розробки.....	28
3.2 Схеми послідовності дій в системі.....	28
3.3 Проектування графічного інтерфейсу програми.....	35
3.4 Створення бази даних для застосунку	39
3.5 Візуальне макетування проекту	41
3.6 Застосування графічних пакетів.....	44

3.7 Робота з програмним продуктом.....	47
3.8 Висновки до розділу	52
4. ЕКОНОМІЧНИЙ РОЗДІЛ	53
4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки.....	53
4.2 Розрахунок узагальненого коефіцієнта якості розробки.....	57
4.3 Розрахунок витрат на проведення науково-дослідної роботи	59
4.3.1 Витрати на оплату праці	60
4.3.2 Відрахування на соціальні заходи	63
4.3.3 Сировина та матеріали	63
4.3.4 Розрахунок витрат на комплектуючі.....	65
4.3.5 Спецустаткування для наукових (експериментальних) робіт	65
4.3.6 Програмне забезпечення для наукових (експериментальних) робіт	65
4.3.7 Амортизація обладнання, програмних засобів та приміщень	66
4.3.8 Паливо та енергія для науково-виробничих цілей	68
4.3.9 Службові відрядження	69
4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації.....	70
4.3.11 Інші витрати.....	70
4.3.12 Накладні (загальновиробничі) витрати	70
4.4 Розрахунок економічної ефективності науково-технічної розробки від її впровадження безпосередньо розробником (замовником).....	76
ВИСНОВКИ	81
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	82
ДОДАТКИ	84
Додаток А	88
Додаток Б.....	91
Додаток В	95

ВСТУП

Актуальність. Стоматологія в сучасному світі є надзвичайно важливим розділом медицини, адже вона включає в себе вивчення зубів й їх лікування.

Профілактична стоматологічна допомога вимагає лише незначних кроків, щоб уникнути серйозних проблем із зубами та яснами. Регулярне чищення зубів і використання зубної нитки значною мірою принесе користь здоров'ю ротової порожнини, але регулярне відвідування стоматолога - єдиний вірний спосіб виявлення та лікування проблем. Виконання належної гігієни зубів настільки важливо, тому що це може запобігти цим захворюванням порожнини рота та стоматологічним проблемам.

Через стрімкий розвиток інформаційних технологій, було би необхідно автоматизувати процеси управління. Стоматологічні клініки відносяться до категорій підприємств, де інформаційні системи не завжди мають елементи управління й введенням пацієнтів.

Зручним та корисним рішенням було би використання програмного забезпечення для управління бізнесом. Автоматизація введення клієнтів включає в себе створення якісного програмного забезпечення, яке зберігає необхідну інформацію про клієнтів й лікарів стоматологічної клініки, розклад роботи клініки і прийому пацієнтів, з записуванням і редагуванням даних, виведення потрібної інформації у певному вигляді.

Ефективне й оперативне проведення роботи, а також її аналіз з прийняттям науково обгрунтованих керуючих рішень, можливі саме при автоматизації стоматологічної клініки.

Стоматологічні клініки мають своєчасно обслуговувати клієнтів, знаходити захворювання, тому процес діагностування відноситься до найважливіших задач і видів роботи стоматологічної клініки, так як саме діагностика визначає напрями й види подальшого лікування.

Метою роботи є дослідження методів роботи з базами даних, а також розробка програмного забезпечення для автоматизації та спрощення ведення пацієнтів в стоматологічній клініці.

Для виконання дипломної роботи вирішувалися такі основні завдання:

- дослідження методів занесення інформації в базу даних.
- дослідження програмного забезпечення для побудови інтерфейсу програмного забезпечення.
- побудова алгоритмів, які виконуватимуть поставлену задачу.
- аби продемонструвати програмне забезпечення, написана програма, яка здатна зберігати записи про клієнтів. Присутній інтерфейс, який дозволяє мануально заносити дані про пацієнтів в базу даних MySQL.

Об'єктом дослідження є ведення пацієнтів в стоматологічній клініці.

Предметом дослідження є моделі, принципи та процеси автоматизації й систематизації роботи з даними пацієнтів

Методи дослідження. У роботі застосовані методи й функції для обробки даних в прикладній програмі. Для розробки програмного забезпечення використовувалась мова програмування Java в поєднанні з пакетами розширення fxml.

Новизна результату роботи полягає у тому, що на основі теоретичних положень розроблене алгоритмічне та програмне забезпечення, яке може використовуватись для підвищення ефективності роботи при веденні пацієнтів в стоматологічній клініці.

Практична цінність роботи полягає у програмній реалізації для ведення пацієнтів, а також у створених алгоритмах обробки даних користувача.

1 ОГЛЯД СТАНУ ПРОБЛЕМИ ТА МЕТОДИ ЇЇ ВИРІШЕННЯ

1.1 Сутність об'єкту дослідження

Необхідно дослідити й реалізувати систему ведення пацієнтів стоматологічної клініки, яка надаватиме персоналу клініки можливість зручно вести роботу з пацієнтами. До даної роботи входить занесення даних пацієнта в базу й оформлення медичної картки пацієнта. Здійснення прийому пацієнта може бути тільки тоді, коли відбудеться оформлення персональної інформації й медичної документації пацієнта [1]. Також має скластись план лікування пацієнта, спираючись на занесену інформацію про причину візиту [1]. Не менш важливою частиною оформлення медичної карти пацієнта є інформація про його скарги, перші симптоми даного захворювання, чи пов'язує пацієнт їх з чимось, характер перебігу хвороби та раніше проведене лікування і його ефективність[2].

Програмне забезпечення має зберігати дані про роботу з пацієнтами, бути легкою для використання.

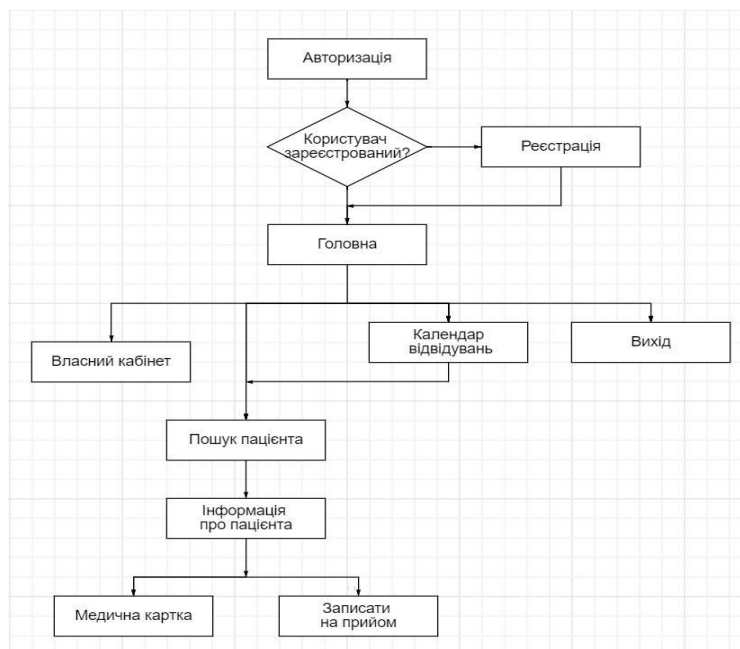


Рисунок 1.1 – Алгоритмічна структура програми

При закінченні розробки програми, персонал клініки зможе самостійно заносити дані що до ведення клієнтів в базу даних.

Для розробки програмного продукту використовувалась мова програмування Java. Графічний інтерфейс побудований за допомогою безкоштовної програми SceneBuilder.

В роботі буде використано та досліджено технології JavaFX та MySQL workbench.

Метою створення програмного забезпечення є безкоштовне вдосконалення ведення пацієнтів стоматологічної клініки, а також його впровадження в медичні установи.

Програмне забезпечення має надати такі переваги:

- мінімізувати комунікацію між працюючим персоналом;
- автоматизація, яка пришвидшить ведення пацієнтів;
- робота з базою даних, що дозволить зберігати інформацію про пацієнтів в зручному форматі;
- зручний контроль роботи клініки;
- фіксування дій персоналу, що дозволить провести аналіз конкретних ситуацій;
- упорядкування роботи: персонал працює за певною схемою, яка прописана в програмі.

Для реалізації роботи було складено такі задачі:

- аналіз предметної області;
- створення програмного модулю, який буде виконувати необхідні функції;
- дослідження сучасних методів розробки;
- створення системи;
- створення інтерфейсу;
- створення бази даних.

1.2 Аналіз існуючого програмного забезпечення, що виконує подібне завдання

Аби скласти вимоги до створеного програмного забезпечення, було розглянуто вже готові програмні рішення, а також складено до них опис. Аналіз програм складений завдяки пошуку в мережі Інтернет. Дослідження даного програмного забезпечення допоможе в розробці свого програмного модулю.

Для огляду візьмемо декілька систем ведення пацієнтів:

SmileStone — це система планування ресурсів стоматологічних технологій, створена керівниками стоматологічних лабораторій. Являється спеціалізованим програмним забезпеченням для управління, яке позбавляє від адміністративної роботи керівників лабораторій і заощаджує багато часу.

Даний програмний продукт дозволяє робити внутрішнє призначення завдань, пов'язаних з підготовкою вхідних задач, також можна керувати з системи. Присутня можливість розділити підготовку роботи на робочі фази і призначити роботу різним співробітникам.

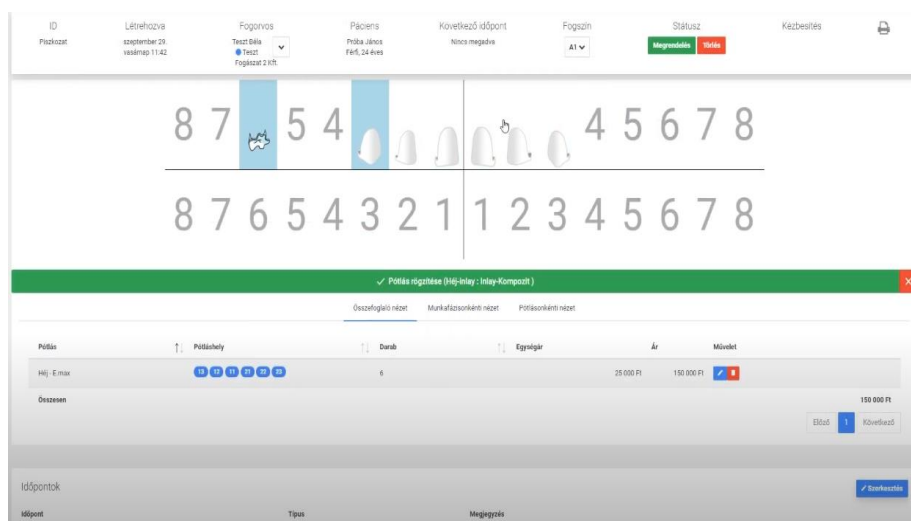


Рисунок 1.1 – Інтерфейс програми SmileStone

Dentaltap — це стоматологічне програмне забезпечення для стоматологів, приватних клінік і мережевих стоматологічних клінік.

Завдяки цій програмі лікарі-стоматологи всіх спеціалізацій зможуть автоматизувати лікувальні процеси, а реєстрація клініки організує всі бізнес-процеси[3]. До переваг даної системи можна віднести:

- використання програмного забезпечення на мобільних пристроях в режимі реального часу;
- гнучка конфігурація інструментів платформи для певних завдань;
- автоматизація планів лікування;
- шаблони записок, бланків та документів;
- багатоканальне спілкування з пацієнтами через смартфон;
- швидка та безкоштовна інтеграція з партнерськими рішеннями;

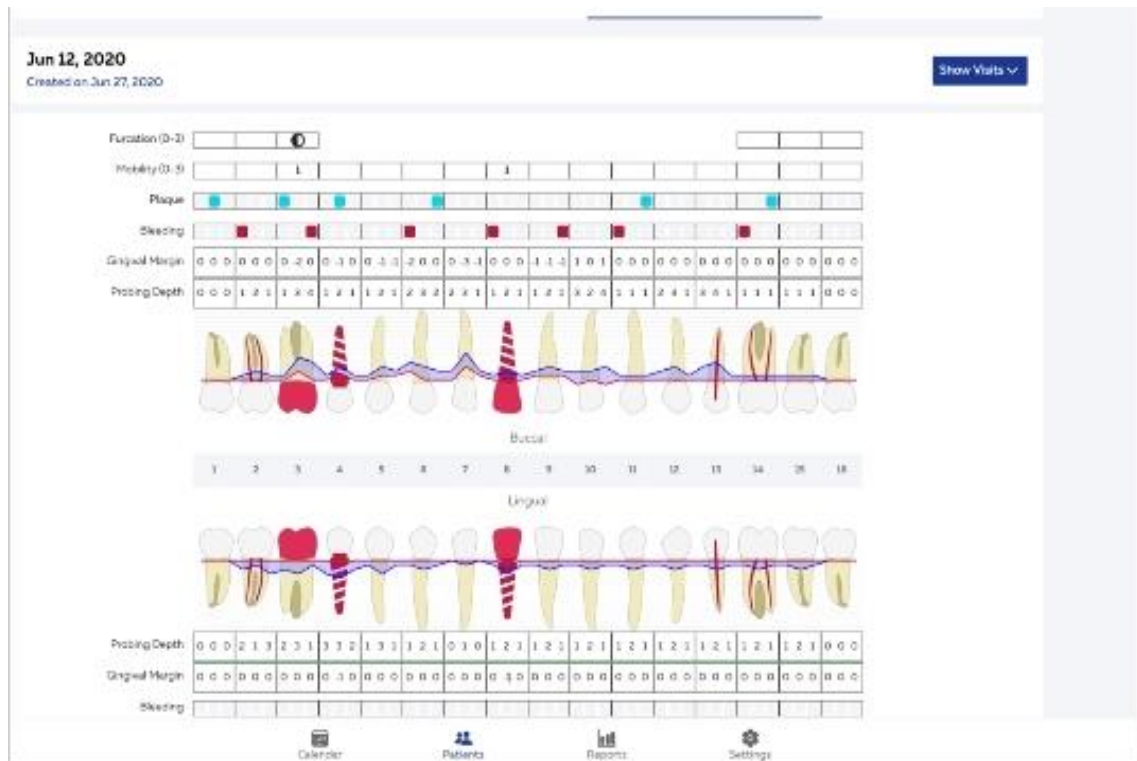


Рисунок 1.2 – Інтерфейс програми Dentaltap

Dental Pro — хмарне програмне забезпечення для керування стоматологічними процесами. Дане програмне забезпечення робить клінічні та адміністративні процедури більш простими та надає потужні інструменти для керування стоматологічною клінікою, незалежно від того, чи це одна клініка чи цілий ланцюг. Перевагами даної системи є те, що є можливість працювати з картою пацієнта, й інтерфейс зрозумілий користувачу[4]. Існує також варіант в мобільному інтерфейсі.

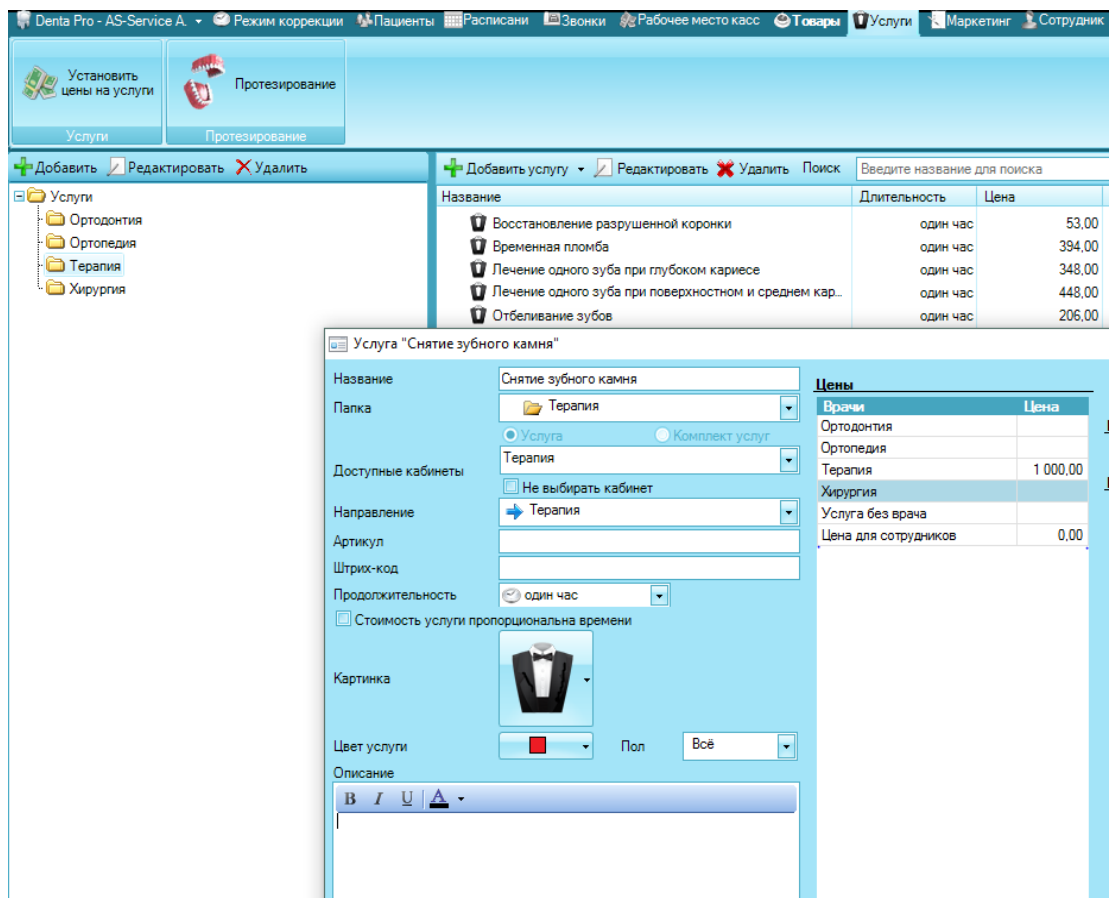


Рисунок 1.3 – Інтерфейс програми Dental Pro

hCue — це система управління пацієнтами, й системою стоматологічних клінік. Деталі минулих пацієнтів можна швидко отримати, знаючи лише одну деталь. Це може бути номер повернення, ідентифікатор пацієнта, дата повернення, ім'я пацієнта, мобільний телефон пацієнта, ідентифікатор

електронної пошти тощо. Даний програмний продукт дозволяє перемикатися між декількома клініками на панелі інструментів. Це дає зручний доступ до керування стоматологічними клініками як єдиного центру. Також перевагами даного програмного забезпечення є зручне керування прийомами. Доступна дана функція цілодобово і без вихідних через мережу інтернет[5].

Сповіднення надсилаються Пацієнтам за допомогою електронної пошти, або сповіщень у додатку.

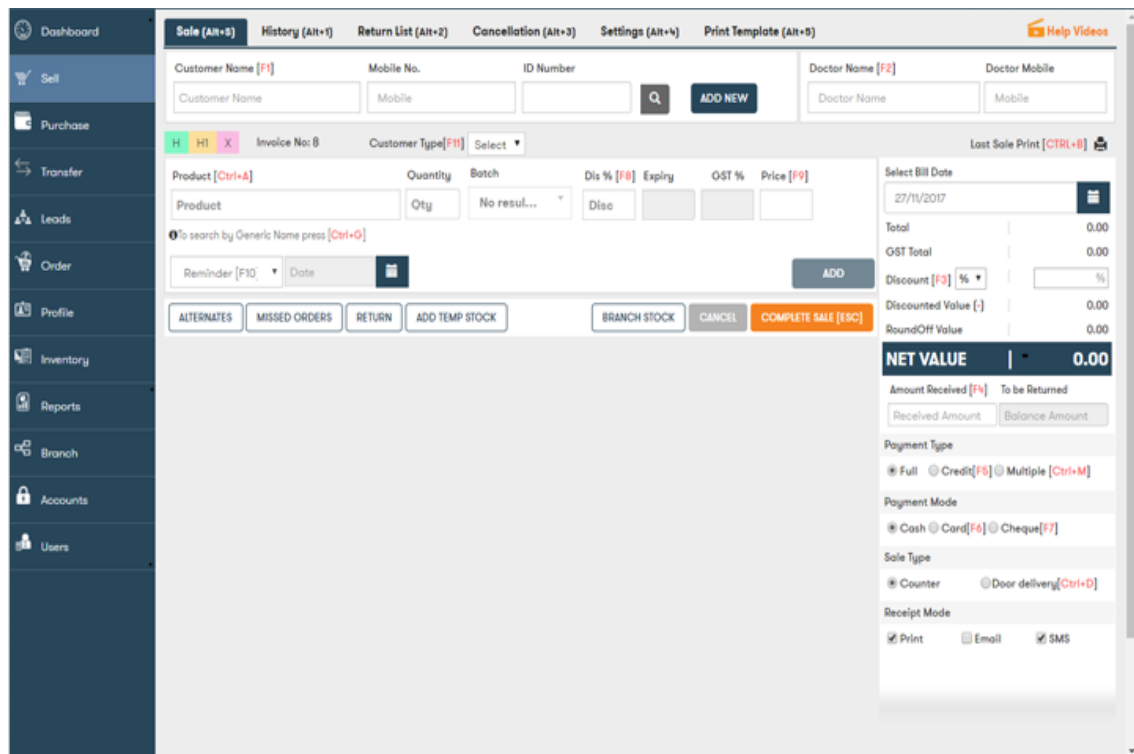


Рисунок 1.4 – Інтерфейс програми hCue

Кожна з розглянутих систем відрізняється різними функціями, способами виконання роботи. Окрім того, частина з них може бути у вигляді як десктопного, так і мобільного додатку. Різницею між цими двома варіантами є те, що розрахунки в десктопному варіанті виконуються на персональному комп'ютері, а при використанні мережевого додатку - на серверах. Краще,

щоб розрахунки відбувались на десктопі, тому що в такому разі відбувається менше навантаження при значному обсягу користувачів на базу даних, коли у веб варіанті даний показник гірший. Однак бувають випадки коли необхідно використати додаток не з робочого пристрою, а з будь-якого іншого місця. Тоді з'являється потреба завантажити додаток в операційну систему. Навідміну від цього, інтернет-додаток може бути використаний з будь-якого комп'ютера, де є присутній інтернет браузер, в тому числі на смартфоні. Варто зауважити, що перехід на новіші версії в онлайн додатку набагато простіший, коли для десктопних необхідно самостійно оновити.

Провевши огляд додатків можна зазначити, що більша частина подібних систем є важкою для користування, пропонують зайві компоненти, які для стоматологічних клінік не є потрібними. Існування таких компонентів може здійснювати вплив на кінцеву вартість продукту.

Варто зазначити, що дані програмні продукти потребують додаткове налаштування й адаптування для функціонування в системі стоматологічних клінік. Саме тому варто створити простий додаток, який буде реалізован по рекомендаціям стоматологів. Такий програмний продукт перш за все має працювати з інформацією клієнтів стоматологічної клініки, саме тому дії при додаванні нових даних про пацієнтів необхідно реалізувати якомога простішими. Важливим також є реалізування зручного пошуку, перегляду й вводу даних. За для того, щоб додаток був зручним для використання, слід чітко структурувати дані, а сам додаток повинен мати зрозумілий для користувача інтерфейс. Саме це впливає на те, на скільки програмне забезпечення буде простим для користувача.

1.3 Критерії роботи систем для роботи з базами даних MySQL

Для того, щоб визначити ефективність роботи програного забезпечення БД, слід використати ряд критеріїв.

- модель бази даних у графічному вигляді. Для ефективнішої розробки проекту інколи необхідно використовувати графічну модель баз даних, так як це дає більш наглядний опис структури даних [6];
- багатоплатформність. Це властивість, яка дозволяє працювати програмі на декількох платформах. Дозволяє скоротити витрати й час на розробку програмного забезпечення [6];
- наявність редактору SQL запитів. При відправці даних на сервер, бажано отримувати відповідь в зручній формі, у вигляді таблиць [6];
- функціональний механізм створення зв'язків між полями. Дана можливість дозволяє створювати зв'язок між полями з функцією створення таблиці зв'язків[6];
- доступність. Даний критерій є не менш важливим, так як багато програмних продуктів потребують значних фінансових витрат. Аби продукт користувався більшим попитом, він стає безплатним.

1.4. Процес ведення пацієнтів.

Програмне забезпечення Dental control використовуватиметься для автоматичної реєстрації клієнтів стоматологічної клініки, покращення роботи й інформативності діагностичних досліджень, ведення документації.

База даних зберігається у вигляді файлу або набору файлів. Інформація у цих файлах може бути розбита на записи, кожен з яких складається з одного або декількох полів. Поля є базовими одиницями зберігання даних, і кожне поле зазвичай містить інформацію, що відноситься до одного аспекту або атрибуту

сутності, описаної базою даних. Записи також організовані у таблиці, які містять інформацію про взаємозв'язки між різними її полями. Хоча база даних застосовується вільно до будь-якої колекції інформації в комп'ютерних файлах, база даних у строгому розумінні надає можливості перехресного посилання. Використовуючи ключові слова та різні команди сортування, користувачі можуть швидко шукати, впорядковувати, групувати та вибирати поля у багатьох записах, щоб отримувати або створювати звіти про певні сукупності даних.

Записи та файли баз даних мають бути організовані, щоб дозволити отримати інформацію. Запити - це основний спосіб, яким користувачі отримують інформацію з бази даних. Потужність СУБД впливає з її здатності визначати нові відносини з базових, заданих таблицями, і використовувати їх для отримання відповідей на запити. Як правило, користувач надає рядок символів, а комп'ютер здійснює пошук у базі даних відповідної послідовності та надає вихідні матеріали, у яких з'являються ці символи.

1.5 Постановка задачі

Після проведення аналізу подібних додатків було складено необхідні вимоги, що реалізовуватимуться в створеному програмному забезпеченні:

- легкий інтерфейс, який буде зрозумілий користувачу, з можливістю легко засвоїти програми;
- надійне зберігання персональних даних персоналу й пацієнтів;
- обмежений доступ до редагування даних про пацієнтів;
- зрозуміле відображення розкладу обслуговування пацієнтів;
- створення швидкого й легкого пошуку пацієнту, який знаходиться в базі даних;
- можливість назначити дату й час прийому пацієнта;
- реалізація статусу, який дозволить показати стан пацієнта.

Дані задачі дозволять реалізувати досить комфортну експлуатацію програмного продукту, а також реалізувати головну ідею — вдосконалення ведення пацієнтів стоматологічної клініки.

1.6 Висновки до розділу

Було досліджено роботу стоматологічних клінік, їх особливості діяльності. Проведено аналіз роботи подібного програмного забезпечення, визначено критерії роботи баз даних.

Також створена концепція системи ведення пацієнтів, яка показує роботу персоналу клініки з системою, була побудована структура бази даних.

2 ДОСЛІДЖЕННЯ ФУНКЦІОНАЛЬНИХ МОЖЛИВОСТЕЙ ПРОГРАМНИХ РІШЕНЬ

2.1 Аналіз роботи прикладного програмного забезпечення з базами даних

Дана розробка являється прикладною програмою. Прикладна програма призначена для виконання певних завдань користувача та розрахована на безпосередню взаємодію з користувачем. В більшості операційних систем прикладні програми не можуть звертатися до ресурсів комп'ютера напряму, а взаємодіють з обладнанням при роботі операційної системи. До прикладного програмного забезпечення відносяться програми, написані для користувачів або самими користувачами, для надання комп'ютеру конкретної роботи [7]. Програми обробки замовлень або створення списків — приклад прикладного програмного забезпечення.

Для того, щоб зберігати й редагувати дані, прикладна програма має з'єднуватись з базою даних. База даних забезпечує зберігання інформації та вважається групою даних, складеними за певними правилами, що має загальні правила опису, зберігання та редагування даних.

Додаток являє собою програму або комплекс програм, які використовують базу даних і забезпечують автоматизацію обробки інформації з деякої предметної області. Додатки можуть створюватися як у середовищі систем управління базами даних, так і без них — за допомогою системи програмування, наприкладі, C++ або Java, користуючись засобами доступу до баз даних.

Для роботи з базою даних у багатьох випадках можна використовувати тільки засоби систем управління базами даних, наприклад створені запити та звіти.

Перевагою використання баз даних в інформаційних системах є забезпечення незалежності даних від прикладних програм. Немає необхідності займатися питаннями розміщення даних у пам'яті і методами доступу до них[8].

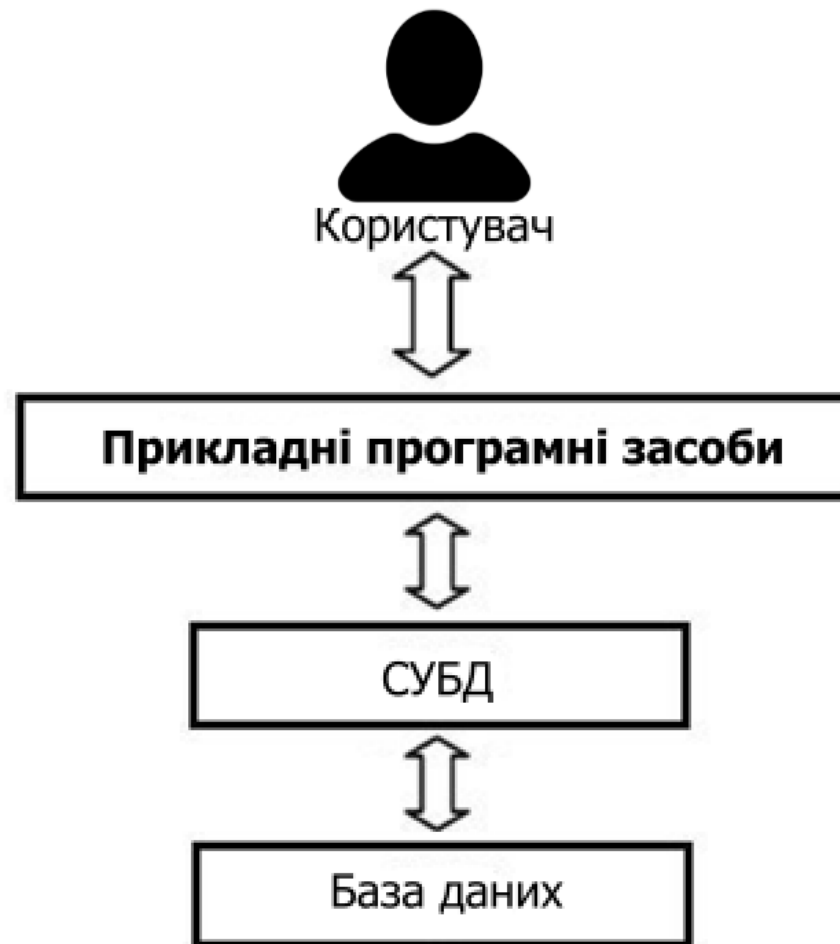


Рисунок 2.1 – Взаємодія користувача з базою даних

2.2 Дослідження функціональних можливостей сучасних програмних рішень

За для того щоб підвищити швидкість й ефективність створення програмного забезпечення, можна використовувати готові фреймворки й

програми, які значно полегшують роботу, автоматично виконуючи необхідні дії для розробника.

2.2.1 MySQL workbench

MySQL Workbench - це єдиний візуальний інструмент для архітекторів баз даних, розробників та адміністраторів баз даних. MySQL Workbench пропонує моделювання даних, розробку SQL та комплексні засоби адміністрування для конфігурації сервера на дошці адміністрування (рисунок 2.1), адміністрування користувачів, резервного копіювання та багато іншого. MySQL Workbench доступний для більшості операційних систем [9].

MySQL Workbench дозволяє адміністратору баз даних, розробнику або архітектору даних візуально проектувати, моделювати, генерувати та керувати базами даних. Він включає в себе все, що потрібне моделювачу даних для створення складних моделей ER, а також забезпечує ключові функції для виконання складних завдань з управління змінами та документації, які зазвичай вимагають багато часу та зусиль, які розробник витрачає на налаштування бази даних.

Він пропонує візуальні інструменти для створення, виконання та оптимізації запитів SQL. Редактор SQL забезпечує виділення синтаксису кольору, автоматичне заповнення, повторне використання фрагментів SQL та історію виконання SQL. Панель з'єднань з базами даних дозволяє розробникам легко керувати стандартними з'єднаннями з базами даних, включаючи MySQL Fabric. Переглядач об'єктів забезпечує миттєвий доступ до схеми бази даних та об'єктів[9].

Також є можливість забезпечувати візуальну консоль для легкого адміністрування середовищ MySQL та кращого огляду баз даних. Розробники та адміністратори баз даних можуть використовувати візуальні інструменти для

налаштування серверів, адміністрування користувачів, виконання резервного копіювання та відновлення, перевірки даних аудиту та перегляду стану бази даних.

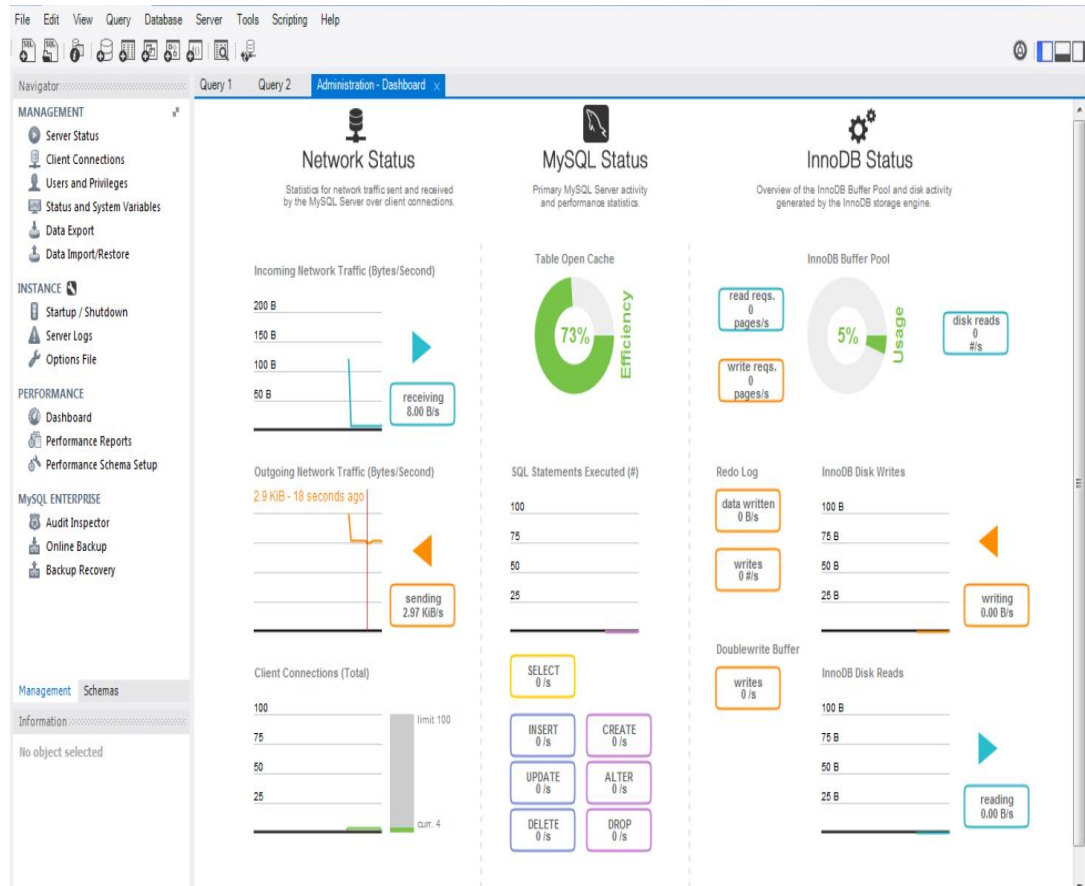


Рисунок 2.2 – Дошка адміністрування в MySQL workbench

2.2.2 DBManager

DBManager – це додаток для управління даними. Присутня вбудована підтримка MySQL (рисунок – 2.2). Він випускається у двох виданнях, тому є можливість вибрати той, який відповідатиме потребам користувача: стандартний та розширений [10]. Стандартне видання є повністю функціональним, і деякі функції навіть відсутні в будь-якому іншому безкоштовному програмному забезпеченні для управління базами даних. DBManager спеціально розроблений

для особистого користування, він має мінімальний набір функцій, що робить його простим для нових користувачів бази даних [10]. Дане програмне забезпечення можна безкоштовно використовувати для некомерційного використання. Для комерційного використання можна придбати ліцензію, яка надаватиме додаткові функції.

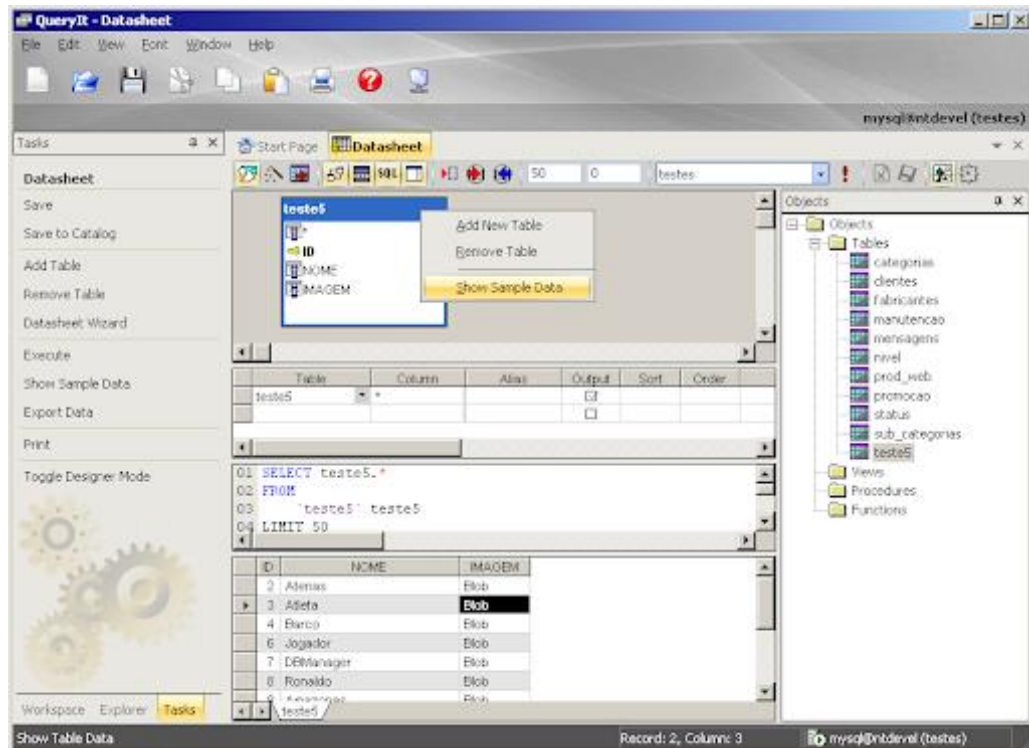


Рисунок 2.3 – Підтримка MySQL в DBManager

2.2.3 MyDB Studio

MyDB Studio - безкоштовний інструмент для адміністрування БД MySQL, який дозволяє створювати, редагувати і видаляти записи, таблиці і бази даних (рисунок – 1.3). Працює виключно на платформі Windows [11].

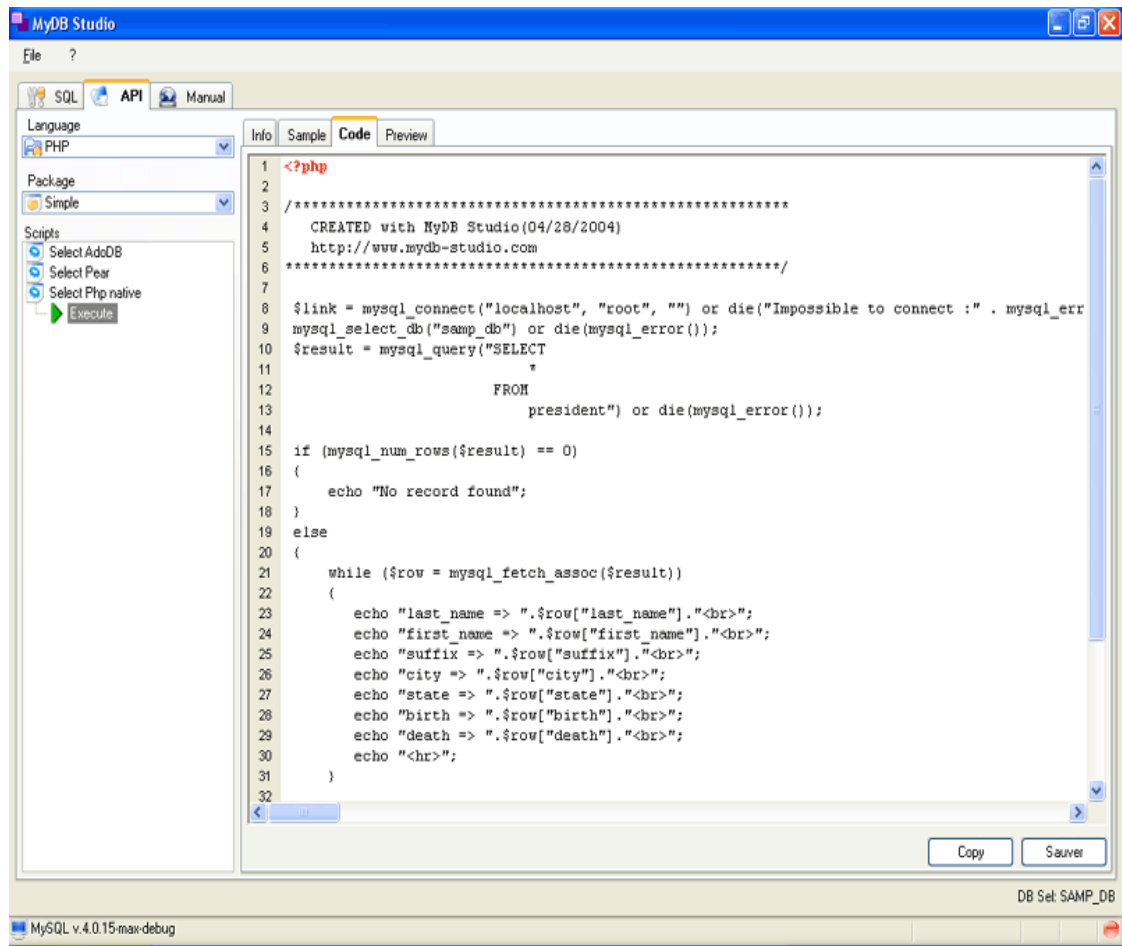


Рисунок 2.4 – Приклад роботи MyDB Studio з MySQL

MyDB Studio дозволяє SSH -тунелювання для захисту з'єднань, що дозволяє підключатися до власного хоста, не дозволяючи підключення по віддаленому доступу.

Адміністратор може надати право користувачам перезавантажити сервер, очистити внутрішні кеші використання MySQL, переглянути/видалити процес сервера, експорт змінних сервера в XML, Excel, Word, CSV [11].

2.2.4 SQLyog Ultimate

SQLyog Ultimate дозволяє розробникам баз даних, адміністраторам та архітекторам візуально порівнювати, оптимізувати та документувати схеми.

SQLyog Ultimate містить інструмент для автоматизації та планування синхронізації даних між двома хостами MySQL(рисунок – 2.4). Присутня можливість створення файлу визначення завдання за допомогою інтерактивного майстра. Інструмент не вимагає інсталяції на хостах MySQL. Для запуску інструменту можна використовувати будь-який хост [11].

SQLyog дає можливість для інтерактивної синхронізації даних.

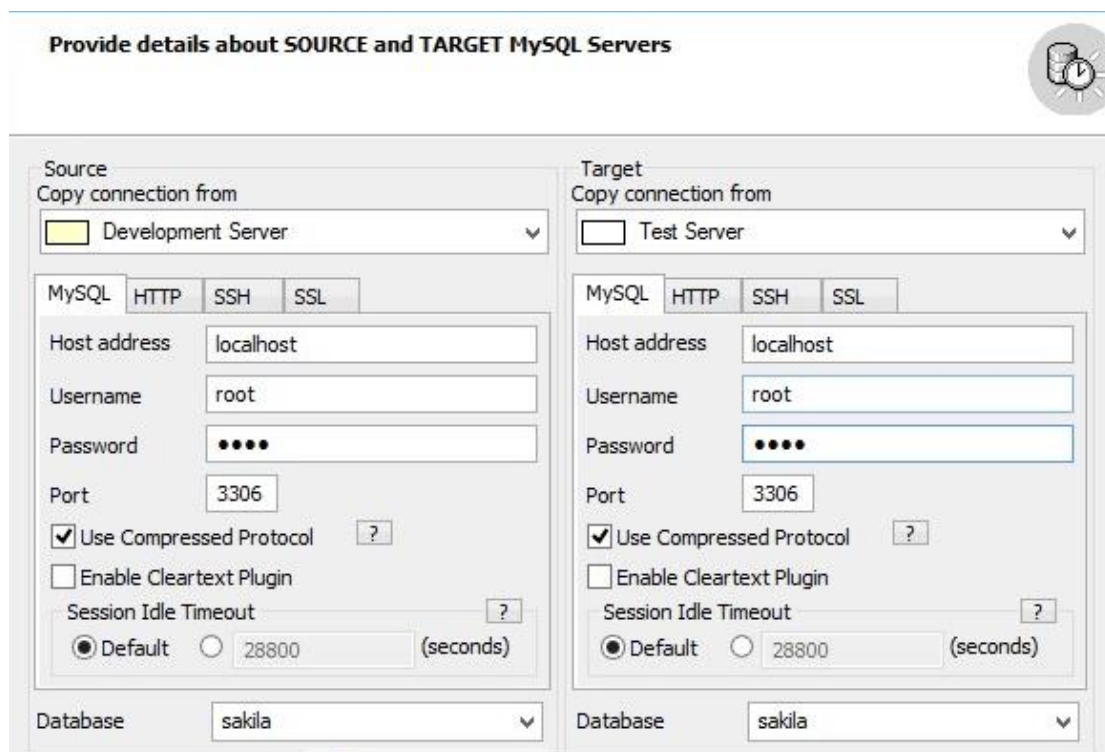


Рисунок 2.5 – Приклад синхронізації хостів в SQLyog Ultimate

2.2 Порівняння існуючих рішень

Кожне рішення має свої особливості. Але для того щоб виконати об'єктивне порівняння, розглянемо переваги й недоліки кожної з програм по запропонованим критеріям:

- модель бази даних у графічному вигляді. Для ефективнішої розробки проекту інколи необхідно використовувати графічну модель баз даних, так

як це дає більш наглядний опис структури даних. Серед розглянутих технологій, графічна модель представлена тільки в MySQL workbench (рисунок 2.5), коли в MyDB Studio, DBManager й SQLyog Ultimate вона відсутня.

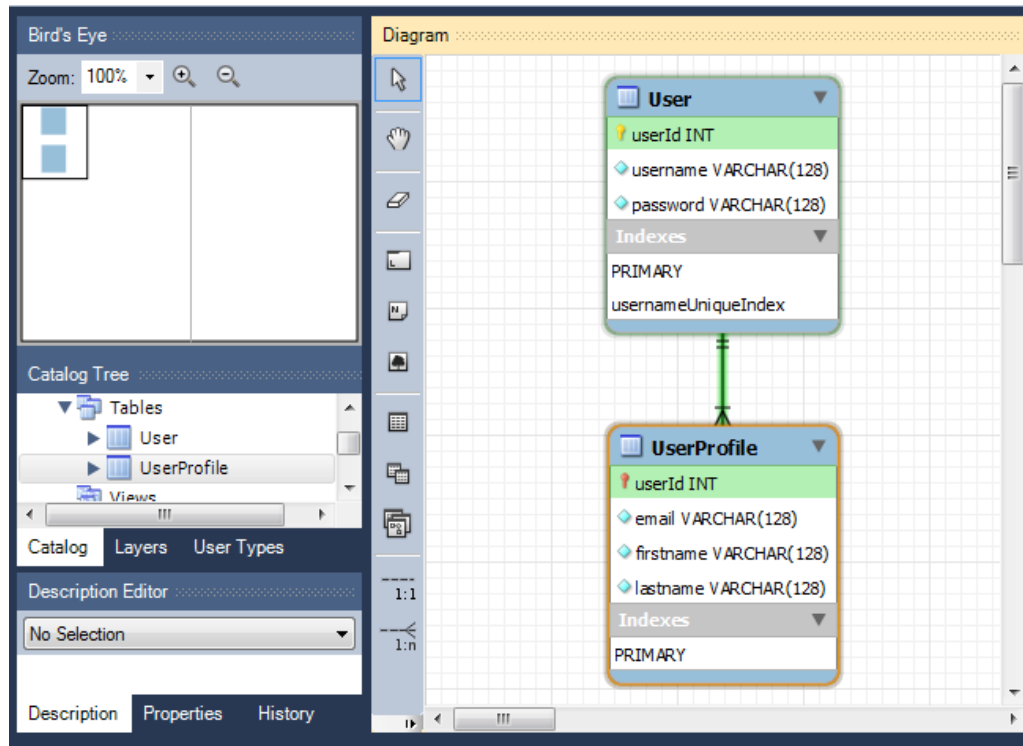


Рисунок 2.6 – Модель бази даних у графічному вигляді MySQL workbench

- багатоплатформність. Це властивість, яка дозволяє працювати програмі на декількох платформах. Дозволяє скоротити витрати й час на розробку програмного забезпечення. Дана властивість присутня в таких розглянутих продуктах, як MySQL workbench, й SQLyog. MyDB Studio й DBManager працюють виключно на платформі Windows;
- наявність редактору SQL запитів. При відправці даних на сервер, бажано отримувати відповідь в зручній формі, у вигляді таблиць. З розглянутих програмних продуктів, даний редактор наявний тільки в програмному забезпеченні MySQL workbench (рисунок 2.6).

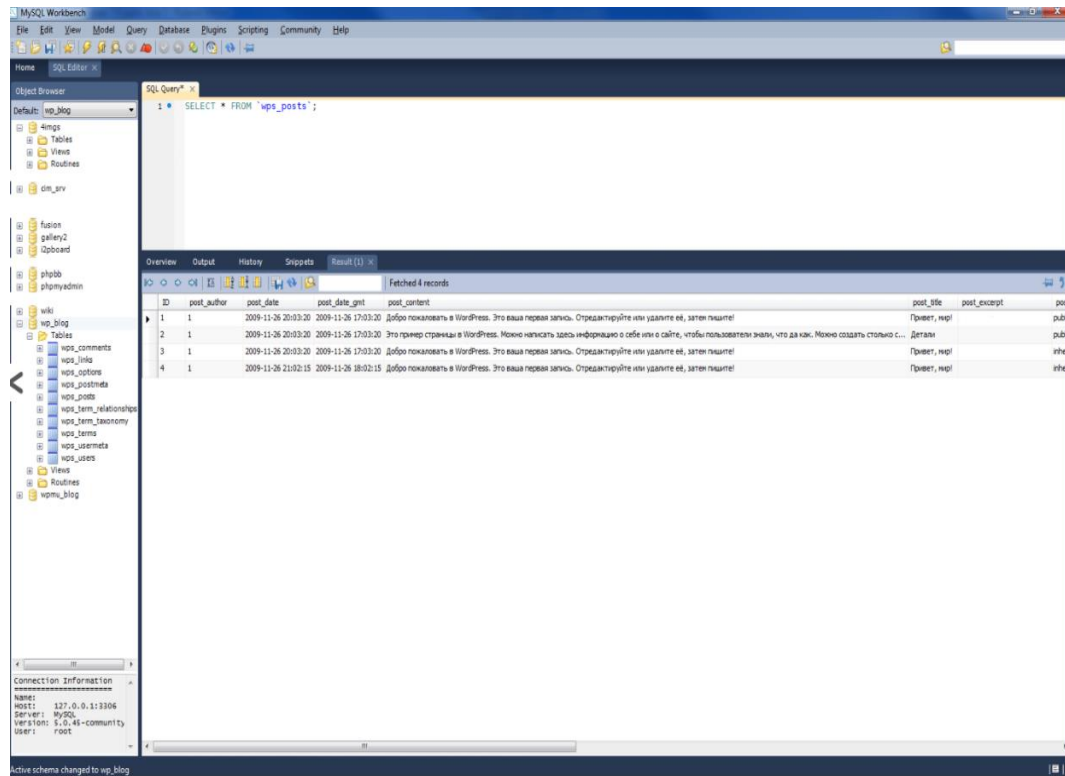


Рисунок 2.7 – Редактор запитів SQL в MySQL Workbench

- функціональний механізм створення зв'язків між полями. Дана можливість дозволяє створювати зв'язок між полями з функцією створення таблиці зв'язків. Цей механізм присутній в таких програмах, як MySQL Workbench й SQLyog.
- доступність. Даний критерій є не менш важливим, так як багато програмних продуктів потребують значних фінансових витрат. Аби продукт користувався більшим попитом, він стає безплатним. На даний момент безплатними програмними продуктами є MySQL Workbench, SQLyog й MyDB Studio.

3 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ СИСТЕМИ ВЕДЕННЯ ПАЦІЄНТІВ

3.1 Модулі й структура розробки

За для легкого використання розробленого програмного продукту, та спрощення користування даними, необхідно побудувати продуману й чітку структуру.

Для реалізації системи ведення пацієнтів було розроблено такий функціонал: сторінка авторизації, реєстрація, головна сторінка, власний кабінет, пошук пацієнта, календар відвідувань, запис на прийом.

Реєстрація дозволяє користувачу зареєструвати себе в системі, що необхідно для того, щоб працювати з даними від свого імені.

Екран авторизації відповідно дозволяє користувачу потрапити в свій профіль, який захищений від інших користувачів.

Головна сторінка містить в собі такі модулі, як:

- пошук пацієнта;
- власний кабінет;
- календар відвідувань;
- вихід з системи.

Модуль пошуку пацієнта дозволяє знайти пацієнта за допомогою його мобільного телефону, або безпосередньо самим ім'ям. Містить в собі такі поля, як:

- введення ім'я або номеру телефону пацієнта;
- результат пошуку.

Модуль власного кабінету дозволяє переглянути й редагувати дані про себе. Містить такі поля:

- прізвище;

- ім'я;
- номер телефону;
- посада;
- e-mail.

Модуль перегляду відвідувань дає змогу переглянути інформацію що до прийомів на конкретний день й час у зручній для користувача формі. Містить такі поля, як:

- фільтр;
- дні прийому;
- година прийому;
- прізвище й ім'я пацієнта.

Також даний продукт має фізичну форму, тобто складається з файлової системи. Файли розташовані в певному порядку у спеціально створених пакетах й папках. При написанні програми використовувалась звичайна організація файлової системи в середовищі розробки IntelliJ Idea. Власно, маємо такі частини, як:

- пакет `com.example.demo` з `java`-файлами, які містять основний код програми, й вважаються головною логічною частиною. Всі ці файли виконуються в головному класі під назвою `Main`;

- пакет `resources` з `FXML`-файлами, які з'єднані з класами-контролерами. Саме в них побудован візуальний інтерфейс програми;

- пакет `assets` з `png`-файлами. Дані файли використовуються для зображення кнопок й іконок програмного забезпечення (рисунок 3.1).

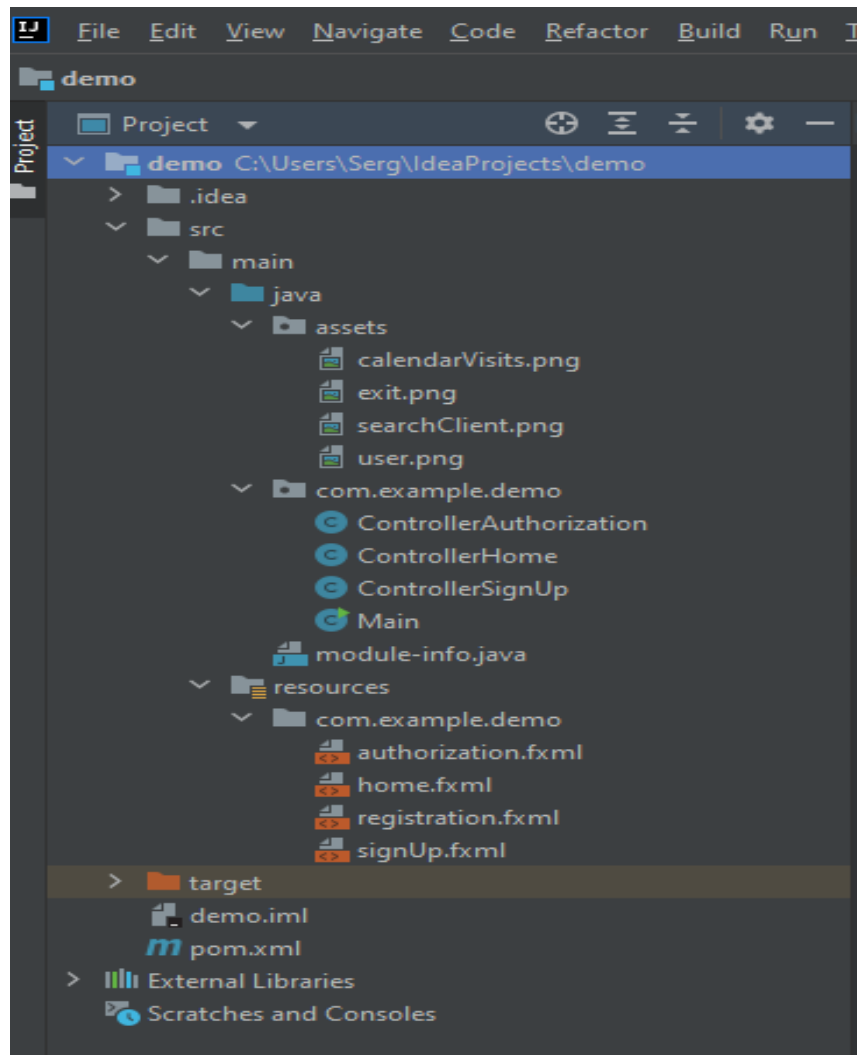


Рисунок 3.1 – Ієрархія файлів проекту

3.2 Схеми послідовності дій в системі

Схеми послідовності дій можна застосувати для уявлення взаємодії користувача з системою, й її внутрішніми процесами.

Дані послідовності відбуваються по принципу роботи прикладного програмного забезпечення.

Нижче буде описано такі послідовності, як: авторизація, реєстрація користувача, пошук пацієнтів, редагування профілю, запис пацієнта на прийом.



Рисунок 3.3 – Схема послідовності «Авторизація»

Авторизація відбувається в такій послідовності:

- 1) користувач вводить логін й пароль в форму авторизації;
- 2) відбувається відправка даних в контролер для обробки;
- 3) в контролері відбувається перевірка коректності заповнення;
- 4) контролер відправляє запит даних користувача до таблиці бази даних користувачів;
- 5) з таблиці приходить відповідь у контролер, й користувач авторизується;

Після даної послідовності відбувається перехід на екран головного меню, де користувач може обрати основні дії для ведення пацієнтів, або почати роботу з персональними даними.



Рисунок 3.4 – схема послідовності «Реєстрація нового користувача»

Реєстрація нового користувача відбувається в такій послідовності:

- 1) користувач переходить до екрану реєстрації;
- 2) користувач заповнює форму з необхідними полями;
- 3) відбувається відправка даних для обробки в контролер;
- 4) відбувається перевірка коректності заповнення;
- 5) внесення даних до таблиці користувачів;
- 6) в контролер надходить відповідь про результат операції.

Після даної послідовності, користувач потрапляє на головну сторінку.



Рисунок 3.5 – схема послідовності «Пошук пацієнта»

Пошук пацієнта відбувається в такій послідовності:

- 1) користувач переходить до форми авторизації;
- 2) користувач авторизується;
- 3) відкривається головне меню;
- 4) перехід до форми пошуку пацієнтів;
- 5) запит інформації про пацієнта до контролера;
- 6) формування результату контролера за допомогою таблиці пацієнтів;

7) повернення результату з таблиці пацієнтів в контролер.

Після даної послідовності відбувається відображення результату.

Редагування профілю пацієнта відбувається в такій послідовності:

- 1) користувач переходить до форми пошуку пацієнта;
- 2) проводить пошук пацієнта;
- 3) потрапляє в профіль пацієнта;
- 4) переходить в редагування профілю пацієнта;
- 5) форма редагування профілю робить запит даних до контролера;
- 6) контролер формує дані пацієнта;
- 7) повернення результату з таблиці пацієнтів в контролер.

Після даної послідовності відбувається збереження даних, й відправка користувача в профіль пацієнта.



Рисунок 3.6 – схема послідовності «Редагування профілю пацієнта»

3.3 Проектування графічного інтерфейсу програми

Графічний інтерфейс програми є важливою складовою програмного забезпечення, так як саме завдяки йому користувач може взаємодіяти з модулями програми, проводячи пошук інформації, її редагування.

Для того щоб побудувати графічний інтерфейс, спочатку необхідно зобразити його каркас, по якому і буде зроблено графічну складову [12].

Проектування графічного інтерфейсу є важливою складовою розробки програмного забезпечення, так як це надає ряд переваг:

- зменшення обсягу роботи при розробці дизайну;
- розробник зменшує кількість роботи, коли створює дизайн;
- стає можливо чітко уявити, як саме має виглядати майбутній інтерфейс;
- можливість оперативно позбутись елемента, який згодом може стати зайвим;
- розробник зменшує кількість роботи, коли створює дизайн
- зменшення кількості часу, який розробник витрачає на розробку чіткого й зрозумілого інтерфейсу;

Нижче наведено приклади проектування основних елементів графічного інтерфейсу розробленої програми:

DENTAL CONTROL

Авторизація

Ім'я користувача

Пароль

Вхід
або
Зареєструватися

Рисунок 3.7 – проектування екрану авторизації

DENTAL CONTROL

Реєстрація

Прізвище

Ім'я

Логін

Пароль

Зареєструватися

Рисунок 3.8 – проектування сторінки реєстрації



Рисунок 3.9 – проектування головної сторінки



Рисунок 3.10 – проектування пошуку пацієнта

DENTAL CONTROL

Медична картка пацієнта
«Ім'я» «Прізвище»

ФОТО

Статус: _____

Візити пацієнта

Скарги

Діагноз

Вийти в головне меню

Рисунок 3.11 – проектування медичної картки пацієнта

DENTAL CONTROL

Записати на прийом
Пацієнта «Прізвище» «Ім'я»

Місце прийому: _____

Причина: _____

Час: _____

Дата: _____

Вийти в головне меню

Рисунок 3.12 – проектування запису на прийом

3.4 Створення бази даних для застосунку

Щоб організувати й керувати даними, необхідно додати базу даних, яка була створена за допомогою системи управління базами даних MySQL Workbench.

Перевагою даної системи є панель керування продуктивністю та звітів, які дозволяють легко переглядати загальну продуктивність сервера, а різні звіти надають представлення про точки доступу, оператори SQL, мережу, платформу даних тощо. Він має змогу надати багато функцій, які можна використовувати для проектування та моделювання. Можна створювати складні моделі, виконувати зворотний і прямий інжиніринг, а також можна легко вносити зміни та керувати документами, які можна використати щоб розробити свою базу даних.

Запити SQL створені та оптимізовані, здатні виконуватись за допомогою візуальних інструментів, наданих робочим середовищем MySQL. Інші функції, які допомагають і спрощують завдання розробки та виконання запитів, включають автозаповнення, виділення синтаксису різними кольорами, надання історії виконання запитів і повторне використання фрагментів SQL. Панель підключень може зберігати та керувати різними з'єднаннями з базою даних для баз даних, включаючи структуру MySQL. До схеми та об'єктів бази даних можна отримати миттєвий доступ за допомогою браузера об'єктів [13].

Існує також візуальна консоль, яка надається саме в MySQL Workbench, яку можуть використовувати адміністратори та розробники баз даних, щоб оглянути все середовище. Інші доступні інструменти можуть виявитися корисними для конфігурації сервера, адміністрування користувачів, аудиту даних для перевірки, перегляду справності бази даних і резервного копіювання, а також відновлення даних.

Ефективність додатків баз даних можна проаналізувати та покращити за допомогою набору інструментів, наданого робочим середовищем MySQL. Існує можливість з'ясувати точки доступу, а також запити та оператори SQL.

Показники продуктивності можна переглянути та проаналізувати за допомогою панелі керування продуктивністю робочого столу. Інші місця, де можна оптимізувати запити, також пропонуються та пояснюються візуально на робочому місці. Основна частина програми, яка включає пошук і зберігання даних, заснована на виконанні запиту та його продуктивності.

Інструмент Workbench надає рішення та функції, які можна використовувати для міграції даних на різні платформи, такі як Sybase ASE, Microsoft Access, PostgreSQL, Microsoft SQL Server, а також об'єкти, дані, таблиці реляційної бази даних до MySQL. Існуючі програми можуть бути легко конвертовані для роботи на будь-якій іншій платформі, наприклад, Windows, Linux тощо. Дані також можна перенести зі старої версії MySQL до новішої за допомогою робочого середовища.

Розглянемо створення таблиці баз даних. Для того, щоб створити макет таблиці, необхідно ввести таку команду «CREATE TABLE». «VARCHAR» є хорошим вибором для стовпців імені, та прізвища користувача, оскільки значення стовпців відрізняються за довжиною. Довжини в цих стовпцях не обов'язково мають бути однаковими. Зазвичай можна вибрати будь-яку довжину від 1 до 65535, що здається найбільш кращим варіантом. Якщо згодом виявиться невдалий вибір, і стане зрозуміло, що потрібне довше поле, MySQL пропонує оператор «ALTER TABLE». Для стовпців дати та часу для прийому пацієнта, доцільно буде використовувати тип даних «DATE». Після створення таблиці, маємо результат на рисунку 3.13.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idusers	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
name	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
lastName	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
login	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
password	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
date	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
time	TIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Рисунок 3.13 – Поля даних користувачів

3.5 Візуальне макетування проекту

Для розробки візуального макету використовувався фреймворк SceneBuilder.

SceneBuilder дозволяє користувачам швидко проектувати користувацькі інтерфейси додатків JavaFX без кодування. Користувачі можуть перетягувати компоненти інтерфейсу користувача до робочої області, змінювати їх властивості, застосовувати таблиці стилів, а код FXML для макету, який вони створюють, автоматично створюється у фоновому режимі. Результатом є файл FXML, який потім можна об'єднати з проектом Java, прив'язавши інтерфейс користувача до логіки програми [14].

Зробити адаптивним графічний інтерфейс дуже важливо. Сьогодні розміри пристрою варіюються від портативних до настінних телевізорів. Використовуючи правильні контейнери та конфігурації компонентів, можна швидко розробити графічний інтерфейс JavaFX за допомогою конструктора сцен.

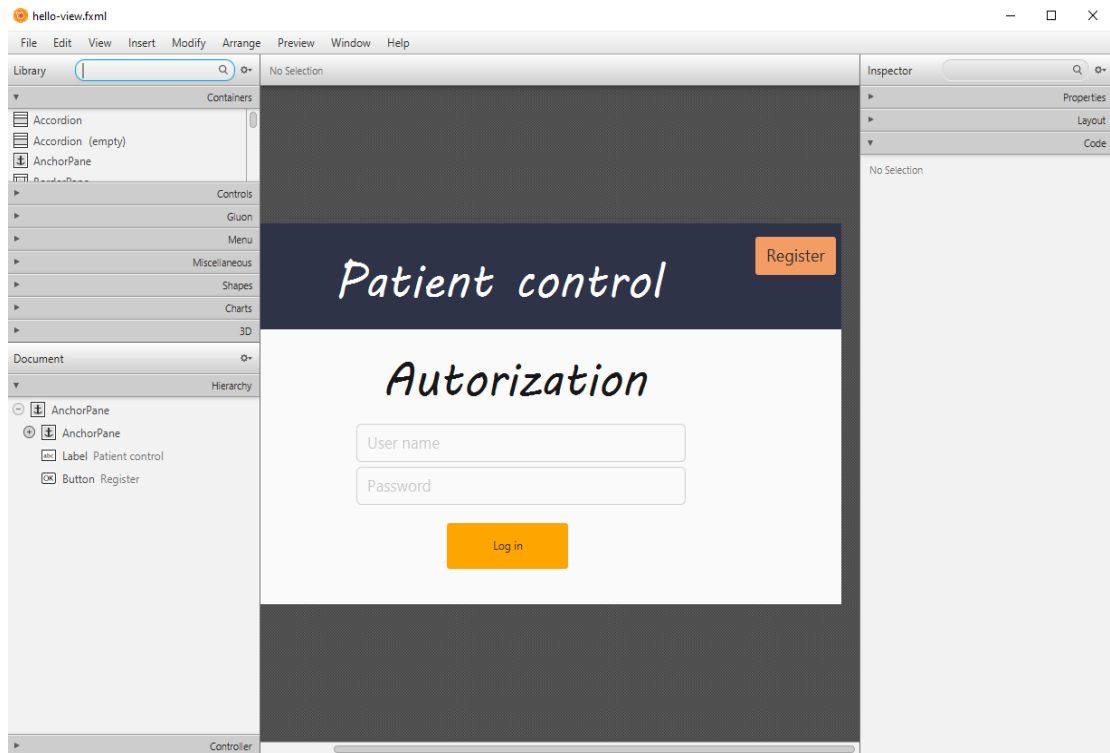


Рисунок 3.14 – приклад інтерфейсу SceneBuilder

Перевагами SceneBuilder є:

- Scene Builder дозволяє легко розміщувати елементи керування, діаграми, фігури та контейнери JavaFX, що дозволяє швидко створювати прототипи користувацьких інтерфейсів. Анімації та ефекти можна легко застосувати для більш складних інтерфейсів користувача[9];
- створює FXML, мову розмітки на основі XML, яка дозволяє користувачам визначати інтерфейс користувача програми, окремо від логіки програми. Також можна відкривати та редагувати наявні файли FXML, створені іншими користувачами;
- можна використовувати в поєднанні з будь -якою IDE Java, але більш тісно інтегрований з IDB NetBeans. Є можливість прив'язати інтерфейс користувача до вихідного коду, який оброблятиме події та дії, що виконуються з кожним елементом, за допомогою простого процесу, запускати свою програму в NetBeans, і будь -які зміни до FXML у NetBeans

також відобразатимуться в проєкті Scene Builder;

- під час створення проєкту можливо переглянути, як справді виглядатиме інтерфейс користувача при його розгортанні, без зайвих обтяжень у меню та палітрах інструменту;

Scene Builder написаний як додаток JavaFX, підтримується у Windows, Mac OS X та Linux. Цей продукт упакований як автономний додаток, що означає, що він поставляється в комплекті з власною приватною копією JRE.

За допомогою таблиць стилів можна застосувати свій вигляд до макета графічного інтерфейсу. Це так само просто, як вибрати компонент графічного інтерфейсу та вказати на файл CSS в панелі властивостей. Аналізатор CSS дозволяє зрозуміти, як конкретні правила CSS можуть впливати на аспекти компонента JavaFX[15].

Аби побудувати інтерфейс, для початку необхідно зайти в панель «Containers» й обрати там необхідні блоки (рисунок Б.4).



Рисунок 3.15 – Список наявних блоків «Containers»

Для прикладу, при розробці сторінки авторизації використовувались такі блоки:

- label;
- textField;
- passwordField;
- button;
- textField;
- text;

Як результат, отримано сторінку авторизації користувача(рисунок 3.13).

The image shows a user authentication interface for 'DENTAL CONTROL'. At the top, there is a dark blue header with the text 'DENTAL CONTROL' in white. Below this is a light blue border containing the title 'Авторизація' in black. There are two input fields: 'Ім'я користувача' and 'Пароль'. Below the fields are two orange buttons: 'Вхід' and 'Зареєструватися', with the word 'або' centered between them.

Рисунок 3.16 – сторінка авторизації користувача

3.6 Застосування графічних пакетів

Для застосування графічних пакетів, використовувався набір JavaFX. Цей набір графічних пакетів дозволяє розробникам проектувати, створювати, тестувати, налагоджувати та розгортати клієнтські програми, які послідовно працюють на різних платформах.

Щоб розробити програми на стороні клієнта з багатьма функціями, програмісти звикли залежати від різних бібліотек, додаючи такі функції, як медіа, елементи керування інтерфейсом, веб, 2D та 3D тощо. Ця технологія включає всі ці функції в єдину бібліотеку. На додаток до них, розробники також можуть отримати доступ до існуючих функцій бібліотеки Java, таких як Swing[16].

JavaFX пропонує багатий набір API, а також використовує сучасний графічний процесор через апаратну графіку. Даний продукт також надає інтерфейси, за допомогою яких розробники можуть поєднувати графічну анімацію та управління UI.

З JavaFX, немає необхідності вивчати додаткові технології, оскільки попереднє знання будь-якої технологій буде достатньо хорошим для розробки RIA з використанням JavaFX.

Код програми JavaFX може посилатися на API з будь-якої бібліотеки Java. Наприклад, програми можуть використовувати бібліотеки API Java для доступу до власних системних можливостей і для підключення до програм на основі проміжного програмного забезпечення на основі сервера [16].

Зовнішній вигляд додатків JavaFX можна налаштувати. Каскадні таблиці стилів (CSS) відокремлюють зовнішній вигляд і стиль від реалізації, щоб розробники могли зосередитися на кодуванні. Графічні дизайнери можуть легко налаштувати зовнішній вигляд та стиль програми за допомогою CSS. Якщо присутній фон веб-дизайну, або якщо необхідно відокремити інтерфейс користувача (UI) і логіку сервера, то є можливість розробити презентаційні

аспекти інтерфейсу на мові сценаріїв FXML та використовувати код Java для програмної логіки. Якщо надається перевага розробці інтерфейсів користувача без написання коду, використовується JavaFX Scene Builder [16]. (рисунок 3.17).

```
<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="399.0" prefWidth="709.0"
style="-fx-background-color: #2E3348;" xmlns="http://javafx.com/javafx/16"
xmlns:fx="http://javafx.com/fxml/1"
fx:controller="com.example.patientcontrol.HomeController">
```

Рисунок 3.17 – фрагмент коду, де відбувається об’єднання файлу FXML з Java

Створюючи новий проект, можна підключити JavaFX, що є вбудованою функцією IntelliJ IDEA. Основним виконувачем програми являється клас HelloApplication, в якому вмикається зв’язок з FXML файлом (рисунок 3.18).

```
package com.example.patientcontrol;

import ...

public class HelloApplication extends Application {
    @Override
    public void start(Stage stage) throws IOException {
        FXMLLoader fxmlLoader = new FXMLLoader(HelloApplication.class.getResource("hello-view.fxml"));
        Scene scene = new Scene(fxmlLoader.load(), w: 700, h: 400);
        stage.setTitle("Hello, dantist!");
        stage.setScene(scene);
        stage.show();
    }
}
```

Рисунок 3.18 – реалізація з’єднання Java з FXML файлом програми

Для того, щоб відбувся зв’язок з FXML файлом, також необхідно підключити такі бібліотеки:

- 1) `javafx.application.Application;`
- 2) `javafx.fxml.FXMLLoader;`
- 3) `javafx.scene.Scene;`
- 4) `javafx.stage.Stage.`

Далі заносимо FXML файл, який являється інтерфейсом програми в окремий пакет проекту (рисунок 4.2) .

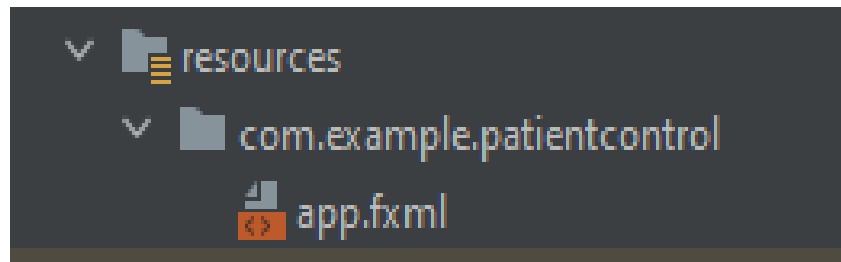


Рисунок 3.19 – пакет з файлами fxml

Далі необхідно створити клас-контролер, який керуватиме FXML файлом [9]. Створюємо екземпляр контролера самостійно, після чого завантажуюмо за допомогою `FXMLLoader`. Також необхідно вказати контролер в самому `SceneBuilder` (рисунок 4.3).

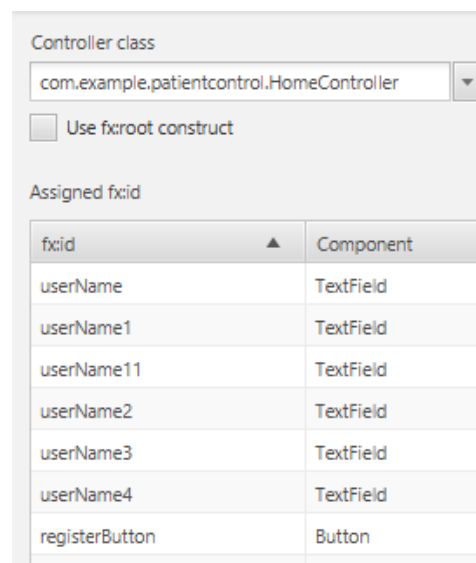


Рисунок 3.20 – з'єднання FXML файлу з контролером

3.7 Робота з програмним продуктом

Для початку користувачу необхідно запустити файл PatientControl.jar, після чого відкривається вікно авторизації. Туди користувач має ввести свій логін та пароль, при умові, що даний користувач вже зареєстрован.

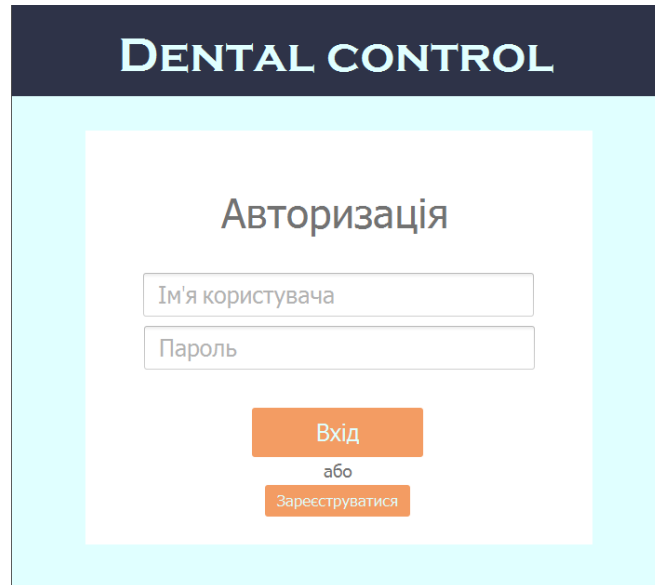


рисунок 3.21 – вікно авторизації

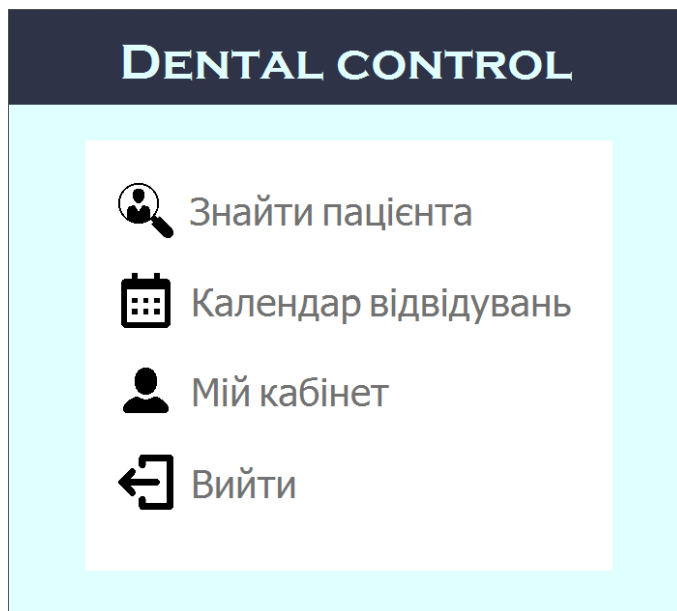


Рисунок 3.22 – головне меню

В даному випадку знаходяться наступні пункти меню:

- знайти пацієнта;
- календар відвідувань;
- мій кабінет;
- вийти.

Користувач має натиснути на кнопку «Знайти пацієнта», після чого він потрапляє на екран пошуку пацієнта(рисунок 3.19). Далі щоб здійснився пошук, користувач має ввести в поле пошуку прізвище, ім'я, або мобільний номер пацієнта. Якщо такий пацієнт існує, має впливсти список пацієнтів з схожими даними, або безпосередньо, сам профіль пацієнта.

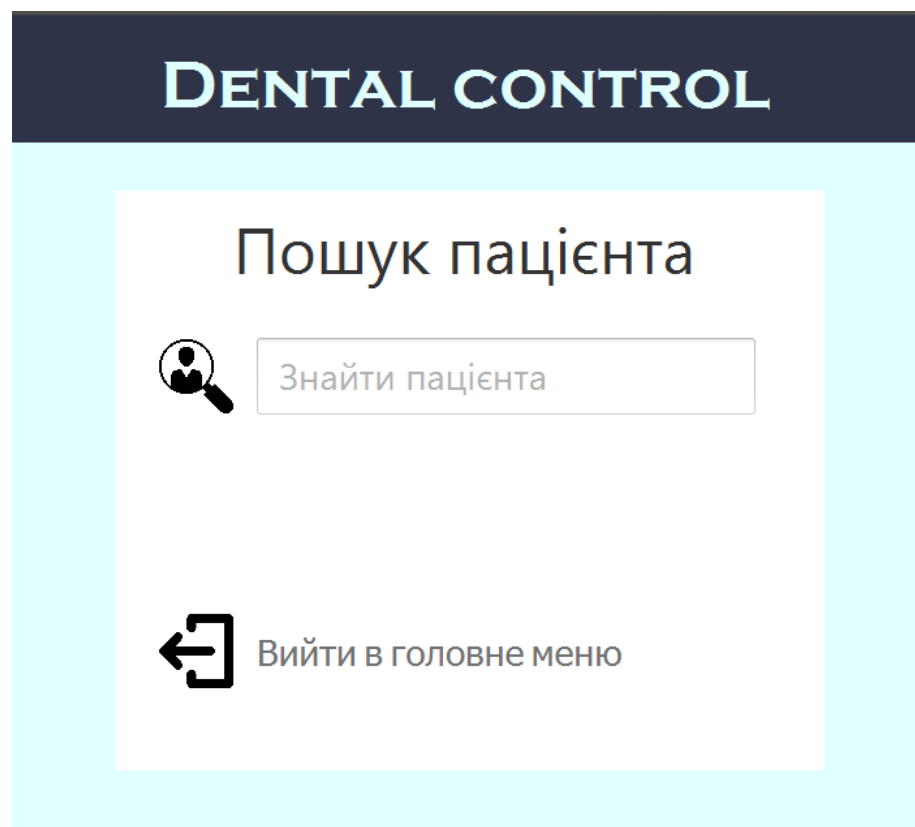
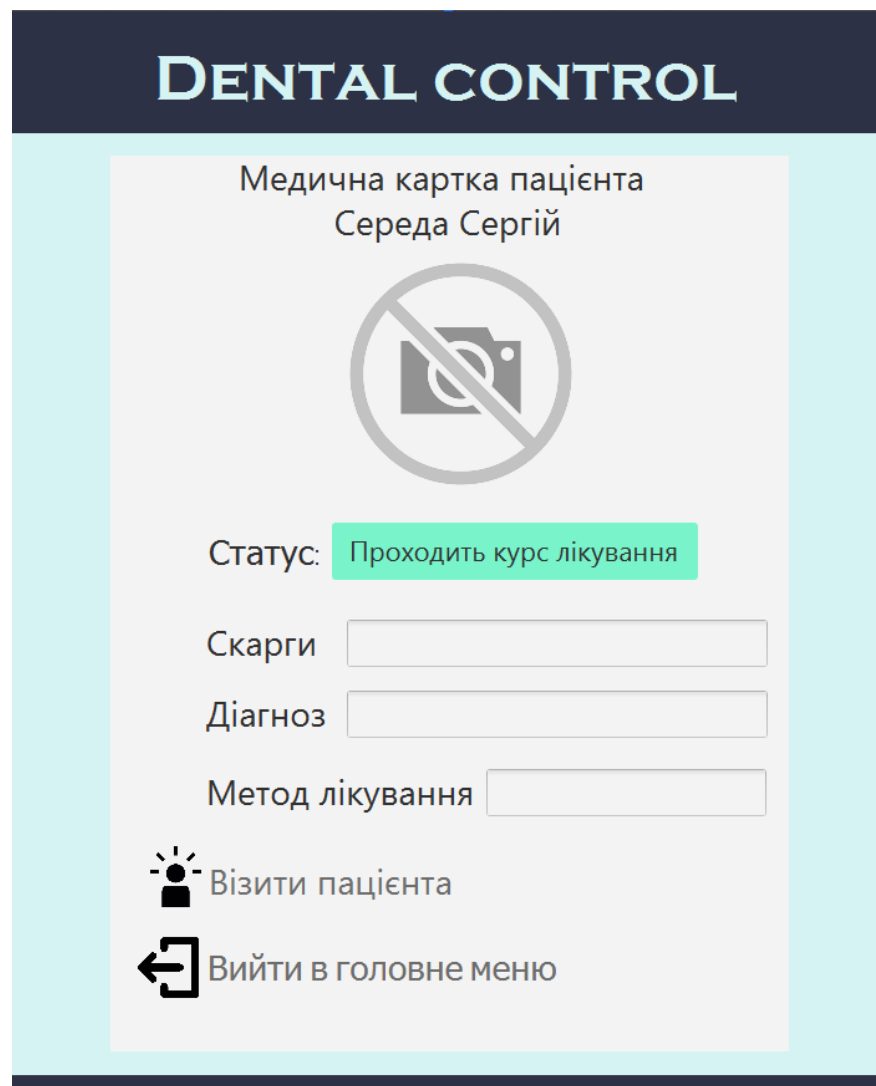


Рисунок 3.23 – екран пошуку пацієнта

Натиснувши на запропонований результат, користувач потрапляє на екран медичної картки пацієнта (рисунок 3.20). Далі треба заповнити дані поля потрібними даними:

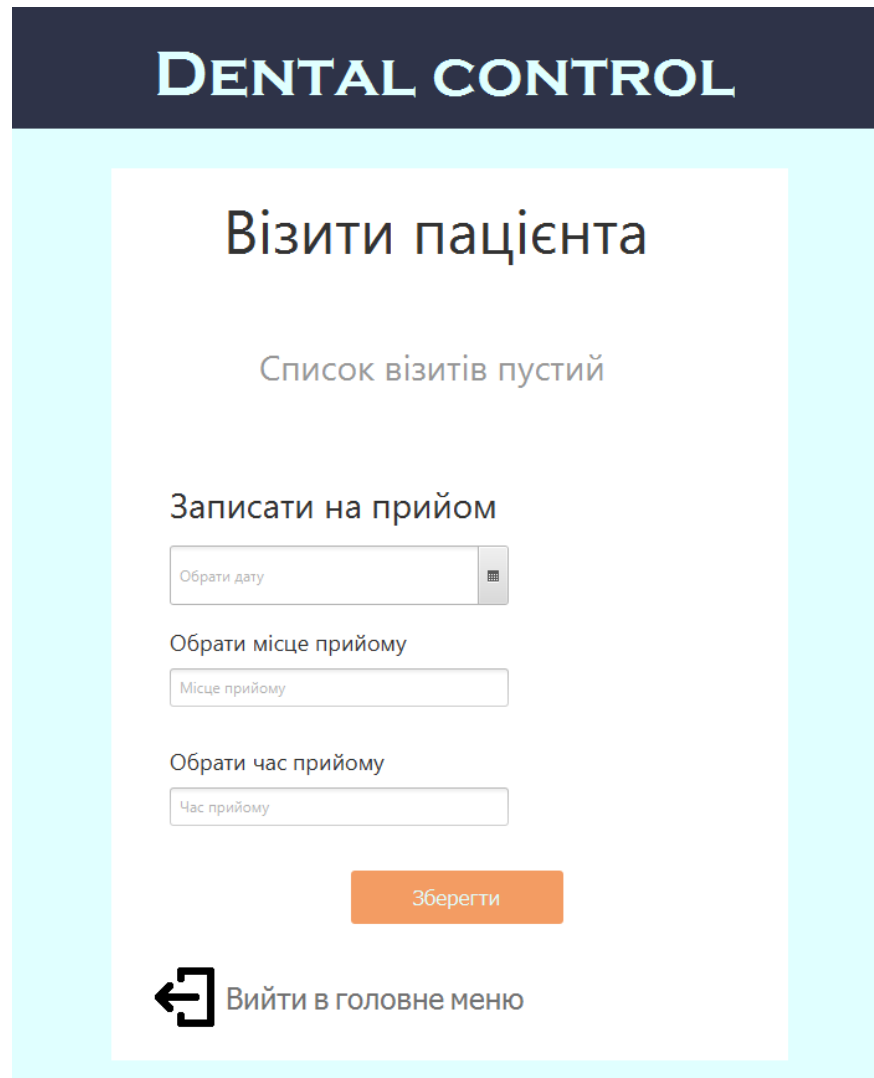
- скарги;
- календар відвідувань;
- мій кабінет;
- вийти.



The screenshot shows a mobile application interface for a dental clinic. At the top, there is a dark blue header with the text "DENTAL CONTROL" in white. Below the header, the main content area has a light blue background. In the center, there is a white card titled "Медична картка пацієнта" (Medical card of the patient) for "Сергея Сергій" (Sergiy Serhiy). Below the title, there is a grey camera icon with a diagonal slash through it, indicating that photo uploads are disabled. The card contains several fields: "Статус:" (Status) with a green highlight and the text "Проходить курс лікування" (Undergoing treatment); "Скарги" (Complaints) with an empty text input field; "Діагноз" (Diagnosis) with an empty text input field; and "Метод лікування" (Treatment method) with an empty text input field. At the bottom of the card, there are two navigation options: "Візити пацієнта" (Patient visits) with a person icon and "Вийти в головне меню" (Exit to main menu) with a back arrow icon.

Рисунок 3.24 – медична картка пацієнта

Після заповнення даних користувач може перейти в розділ візитів пацієнта, де може вказати дату, час, та назначене місце візиту (рисунок 3.21).



DENTAL CONTROL

Візити пацієнта

Список візитів пустий

Записати на прийом

Обрати дату

Обрати місце прийому

Обрати час прийому

Зберегти

Вийти в головне меню

Рисунок 3.25 – вікно візитів пацієнта

Після заповнення полів, натиснути кнопку «Зберегти».

Після даної послідовності дій, користувач може записати пацієнта на новий прийом до лікаря.

3.8 Висновки до розділу

В даному розділі було реалізоване створення програми для ведення пацієнтів стоматологічної клініки за допомогою технології JavaFX. Також покроково описано створення програмного забезпечення для використання, а також перевірено графічний інтерфейс для користувача.

4 ЕКОНОМІЧНИЙ РОЗДІЛ

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

Магістерська кваліфікаційна робота за темою «Автоматизована система управління стоматологічною клінікою. Розробка модуля ведення пацієнтів» відноситься до науково-технічних робіт, які плануються для використання безпосередньо самим розробником (замовником), тобто її результатами буде користуватися тільки одна особа – розробник (або замовник). У цьому випадку нам потрібно довести ефективність інвестицій, вкладених у цей проект самим розробником (замовником).

Для цього випадку необхідно виконати такі етапи робіт:

- 1) провести технологічний аудит власної науково-технічної розробки, тобто встановити її науково-технічний рівень;
- 2) розрахувати витрати на здійснення науково-технічної розробки;
- 3) провести розрахунок економічної ефективності науково-технічної розробки у випадку її впровадження розробником (замовником) на власному підприємстві та обґрунтувати економічну доцільність впровадження розробником (замовником) розробленого науково-технічного проекту.

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Автоматизована система управління стоматологічною клінікою. Розробка модуля ведення пацієнтів» є оцінювання науково-технічного рівня та

рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в табл. 4.1 [17].

Таблиця 4.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція не підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено на працездатність продукту в реальних
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має	Ринок малий, але має	Середній ринок з позитивною	Великий стабільний	Великий ринок з позитивною

Продовження таблиці 4.1

7	Активна конкуренція	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як знайти технічної, так і з комерційної	Необхідно фахівців або витратити	Необхідне незначне навчання фахівців та	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної
9	Потрібні значні фінансові ресурси, які	Потрібні незначні фінансові ресурси.	Потрібні значні фінансові ресурси. Джерела	Потрібні незначні фінансові ресурси.	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовують ся у військово	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін	Термін реалізації ідеї від 3-х до 5-ти років. Термін	Термін реалізації ідеї менше 3-х років. Термін	Термін реалізації ідеї менше 3-х років. Термін
12	Необхідна розробка регламентних документів та отримання великої	Необхідно отримання великої кількості дозвільних документів на	Процедура отримання дозвільних документів для виробництва та реалізації	Необхідно тільки повідомлення відповідним органам про виробництво	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці.

Таблиця 4.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	4	4	4
2. Ринкові переваги (наявність аналогів)	4	3	4
3. Ринкові переваги (ціна продукту)	4	4	3
4. Ринкові переваги (технічні властивості)	2	2	2
5. Ринкові переваги (експлуатаційні витрати)	4	4	4
6. Ринкові перспективи (розмір ринку)	4	4	3
7. Ринкові перспективи (конкуренція)	4	4	4
8. Практична здійсненність (наявність фахівців)	4	3	3
9. Практична здійсненність (наявність фінансів)	4	3	4
10. Практична здійсненність (необхідність нових матеріалів)	3	3	3
11. Практична здійсненність (термін реалізації)	3	4	4
12. Практична здійсненність (розробка документів)	3	3	3
Сума балів	43	41	41
Середньоарифметична сума балів $СБ_c$	41,7		

За результатами розрахунків, наведених в таблиці 4.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в табл. 4.3 [17].

Таблиця 4.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів СБ , розрахована на основі висновків	Науково-технічний рівень та комерційний потенціал розробки
41...48	Високий
31...40	Вище середнього
21...30	Середній
11...20	Нижче середнього
0...10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Автоматизована система управління стоматологічною клінікою. Розробка модуля ведення пацієнтів» становить 41,7 бала, що, відповідно до таблиці 4.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий).

4.2 Розрахунок узагальненого коефіцієнта якості розробки

Окрім комерційного аудиту розробки доцільно також розглянути технічний рівень якості розробки, розглянувши її основні технічні показники. Ці показники по-різному впливають на загальну якість проектної розробки.

Узагальнений коефіцієнт якості (B_n) для нового технічного рішення розрахуємо за формулою [16]:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i, \quad (4.1)$$

де k – кількість найбільш важливих технічних показників, які впливають на якість нового технічного рішення;

α_i – коефіцієнт, який враховує питому вагу i -го технічного показника в загальній якості розробки. Коефіцієнт α_i визначається експертним шляхом

і при цьому має виконуватись умова $\sum_{i=1}^k \alpha_i = 1$;

β_i – відносне значення i -го технічного показника якості нової розробки.

Відносні значення β_i для різних випадків розраховуємо за такими формулами:

- для показників, зростання яких вказує на підвищення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ni}}{I_{ai}}, \quad (4.2)$$

де I_{ni} та I_{na} – чисельні значення конкретного i -го технічного показника якості відповідно для нової розробки та аналога;

- для показників, зростання яких вказує на погіршення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ai}}{I_{ni}}; \quad (4.3)$$

Використовуючи наведені залежності можемо проаналізувати та порівняти техніко-економічні характеристики аналогу та розробки на основі отриманих наявних та проектних показників, а результати порівняння зведемо до таблиці

Таблиця 4.4 – Порівняння основних параметрів розробки та аналога.

Показники (параметри)	Одиниця вимірю- вання	Аналог	Проектований пристрій	Відношення параметрів нової розробки до аналога	Питома вага показника

Продовження таблиці 4.4

Кількість підтримуваних баз даних	шт.	2	3	1,5	0,1
Дружність інтерфейсу	бал	6	8	1,33	0,15
Рівень інтегрованості в систему	бал	7	9	1,29	0,3
Рівень захисту від несанкціонованого доступу	бал	6	9	1,5	0,25
Кількість спеціальних функцій лікувального супроводу пацієнтів	шт.	4	8	2	0,2

Узагальнений коефіцієнт якості (B_n) для нового технічного рішення складе:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i = 1 \cdot 0,15 + 1,33 \cdot 0,1 + 2 \cdot 0,3 + 1,11 \cdot 0,25 + 2 \cdot 0,2 = 1,51.$$

Отже за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,51 рази.

4.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Автоматизована система управління стоматологічною клінікою. Розробка модуля ведення пацієнтів», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

4.3.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників (Z_o) розраховуємо у відповідності до посадових окладів працівників, за формулою [17]:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.4)$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, $T_p=22$ дні.

$$Z_o = 12000,00 \cdot 21 / 22 = 11454,55 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.5 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проекту	12000,00	545,45	21	11454,55
Інженер-програміст	11500,00	522,73	21	10977,27
Консультант (лікар-стоматолог)	10200,00	463,64	5	2318,18
Інженер-проектувальник автоматизованих систем управління	11450,00	520,45	15	7806,82
Консультант-реєстратор (рецепція)	9250,00	420,45	5	2102,27
Всього				34659,09

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників (Z_p) за відповідними найменуваннями робіт НДР на тему «Автоматизована система управління стоматологічною клінікою. Розробка модуля ведення пацієнтів» розраховуємо за формулою:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \quad (4.5)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (4.6)$$

де M_M – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), прийmemo $M_M=2379,00$ грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) [17];

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p – середнє число робочих днів в місяці, приблизно $T_p = 22$ дн;

$t_{зм}$ – тривалість зміни, год.

$$C_l = 2379,00 \cdot 1,10 \cdot 1,65 / (22 \cdot 8) = 24,53 \text{ грн.}$$

$$Z_{pl} = 24,53 \cdot 8,00 = 196,27 \text{ грн.}$$

Таблиця 4.6 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника, грн
Установка офісного обладнання	8,00	2	1,10	24,53	196,27
Підготовка робочого місця розробника модуля ведення пацієнтів	4,00	2	1,10	24,53	98,13

Продовження таблиці 4.6

Формування бази досліджень	10,00	4	1,50	33,45	334,55
Тренування системи	3,50	3	1,35	30,11	105,38
Всього					923,91

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$Z_{\text{дод}} = (Z_o + Z_p) \cdot \frac{H_{\text{дод}}}{100\%}, \quad (4.7)$$

де $H_{\text{дод}}$ – норма нарахування додаткової заробітної плати. Прийmemo 10%.

$$Z_{\text{дод}} = (34659,09 + 923,91) \cdot 10 / 100\% = 3558,30 \text{ грн.}$$

4.3.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$Z_n = (Z_o + Z_p + Z_{\text{дод}}) \cdot \frac{H_{\text{зн}}}{100\%} \quad (4.8)$$

де $H_{\text{зн}}$ – норма нарахування на заробітну плату. Приймаємо 22%.

$$Z_n = (34659,09 + 923,91 + 3558,30) \cdot 22 / 100\% = 8611,09 \text{ грн.}$$

4.3.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Автоматизована система управління стоматологічною клінікою. Розробка модуля ведення пацієнтів».

Витрати на матеріали (M), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n H_j \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_{ej}, \quad (4.9)$$

де H_j – норма витрат матеріалу j -го найменування, кг;

n – кількість видів матеріалів;

C_j – вартість матеріалу j -го найменування, грн/кг;

K_j – коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$);

B_j – маса відходів j -го найменування, кг;

C_{ej} – вартість відходів j -го найменування, грн/кг.

$$M_1 = 3,00 \cdot 96,00 \cdot 1,11 - 0,000 \cdot 0,00 = 319,68 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.7 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Папір канцелярський офісний (А4)	96,00	3,00	0,000	0,00	319,68
Папір для заміток (А5)	45,00	3,00	0,000	0,00	149,85
Начиння канцелярське	150,00	4,00	0,000	0,00	666,00
Органайзер офісний	95,00	4,00	0,000	0,00	421,80

Продовження таблиці 4.7

Картридж для принтера	810,00	2,00	0,000	0,00	1798,20
Диск оптичний	11,50	4,00	0,000	0,00	51,06
FLASH-пам'ять	320,00	1,00	0,000	0,00	355,20
Всього					3761,79

4.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі (K_8), які використовують при проведенні НДР на тему «Автоматизована система управління стоматологічною клінікою. Розробка модуля ведення пацієнтів відсутні.

4.3.5 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення. В даному дослідженні витрати на спецустаткування не передбачені.

4.3.6 Програмне забезпечення для наукових (експериментальних) робіт

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{\text{прог}} = \sum_{i=1}^k C_{\text{прог},i} \cdot C_{\text{прог},i} \cdot K_i, (4.10)$$

де $C_{\text{прог}}$ – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{прз.i}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($K_i = 1, 10 \dots 1, 12$);

k – кількість найменувань програмних засобів.

$$B_{прз} = 5310,00 \cdot 1 \cdot 1,1 = 5841,00 \text{ грн.}$$

Отримані результати зведемо до таблиці:

Таблиця 4.8 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
ОС Windows 10	1	5310,00	5841,00
Прикладний пакет Microsoft Office 2016	1	5120,00	5632,00
Всього			11473,00

4.3.7 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_{б}}{T_{в}} \cdot \frac{t_{вик}}{12}, \quad (4.11)$$

де $Ц_{б}$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_{в}$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{обл} = (23200,00 \cdot 1) / (2 \cdot 12) = 966,67 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.9 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Програмно-аналітичний комплекс	23200,00	2	1	966,67
Графічно-обчислювальний комплекс обробки даних	24100,00	2	1	1004,17
Програмне забезпечення розробки прикладного модуля реєстрації пацієнтів	10200,00	2	1	425,00
Обладнання виводу інформації	6500,00	4	1	135,42
Місце оператора спеціалізоване	7640,00	4	1	159,17
Офісна оргтехніка	9250,00	5	1	154,17

Продовження таблиці (4.9)

Приміщення дослідницької лабораторії	300000,00	20	1	1250,00
Всього				4094,58

4.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{eni}}{\eta_i}, (4.12)$$

де W_{yi} – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

C_e – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), прийmemo $C_e = 4,25$ грн;

K_{eni} – коефіцієнт, що враховує використання потужності, $K_{eni} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$$B_e = 0,32 \cdot 164,0 \cdot 4,25 \cdot 0,95 / 0,97 = 223,04 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.10 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Програмно-аналітичний комплекс	0,32	164,0	223,04
Графічно-обчислювальний комплекс обробки даних	0,50	160,0	340,00

Продовження таблиці 4.10

Обладнання виводу інформації	0,05	30,0	6,38
Місце оператора спеціалізоване	0,15	160,0	102,00
Офісна оргтехніка	0,42	10,0	17,85
Всього			689,27

4.3.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему «Автоматизована система управління стоматологічною клінікою. Розробка модуля ведення пацієнтів» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20...25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cv} = (Z_o + Z_p) \cdot \frac{H_{cv}}{100\%}, (4.13)$$

де H_{cv} – норма нарахування за статтею «Службові відрядження», прийmemo $H_{cv} = 20\%$.

$$B_{cv} = (34659,09 + 923,91) \cdot 20 / 100\% = 7116,60 \text{ грн.}$$

4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» розраховуємо як 30...45% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cn} = (Z_o + Z_p) \cdot \frac{H_{cn}}{100\%}, (4.14)$$

де H_{cn} – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», прийmemo $H_{cn} = 30\%$.

$$B_{cn} = (34659,09 + 923,91) \cdot 30 / 100\% = 10674,90 \text{ грн.}$$

4.3.11 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_e = (Z_o + Z_p) \cdot \frac{H_{ie}}{100\%}, (4.15)$$

де H_{ie} – норма нарахування за статтею «Інші витрати», прийmemo $H_{ie} = 50\%$.

$$I_e = (34659,09 + 923,91) \cdot 50 / 100\% = 17791,50 \text{ грн.}$$

4.3.12 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{нзв} = (З_o + З_p) \cdot \frac{H_{нзв}}{100\%}, (4.16)$$

де $H_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», прийmemo $H_{нзв} = 100\%$.

$$B_{нзв} = (34659,09 + 923,91) \cdot 100 / 100\% = 35583,00 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи на тему «Автоматизована система управління стоматологічною клінікою. Розробка модуля ведення пацієнтів» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = З_o + З_p + З_{доо} + З_n + M + K_в + B_{стел} + B_{прг} + A_{обл} + B_e + B_{св} + B_{сн} + I_в + B_{нзв}. (4.18)$$

$$B_{заг} = 34659,09 + 923,91 + 3558,30 + 8611,085483 + 3761,79 + 0,00 + 0,00 + 11473,00 + 4094,58 + 689,27 + 7116,60 + 10674,90 + 17791,50 + 35583,00 = 138937,02 \text{ грн.}$$

Загальні витрати $ЗВ$ на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ЗВ = \frac{B_{заг}}{\eta}, (4.17)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, прийmemo $\eta = 0,9$.

$$ЗВ = 138937,02 / 0,9 = 154374,46 \text{ грн.}$$

4.4 Розрахунок економічної ефективності науково-технічної розробки від її впровадження безпосередньо розробником (замовником)

При виконанні даної роботи за темою «Автоматизована система управління стоматологічною клінікою. Розробка модуля ведення пацієнтів» розглядається ситуація, коли замовник певної науково-технічної розробки використовує її тільки на своєму підприємстві (чи в організації) і не виводить її на ринок. У цьому випадку позитивним результатом від впровадження цієї науково-технічної розробки може бути покращення певних економічних та фінансових показників діяльності підприємства.

Для визначення величини майбутнього економічного ефекту та ефективності розробки визначимо певні характеристики підприємства.

Таблиця 4.11 – Вихідні дані замовника

Показник	Рік розробки	1-й рік	2-й рік	3-й рік	4-й рік
Чисельність працівників, які виконують визначені функції вручну, осіб	4	-	-	-	-
Середня заробітна плата працівника, який виконує відповідну функцію вручну, грн	10000,00	-	-	-	-
Приблизні витрати на розробку автоматизованої системи управління, грн	154374,46	-	-	-	-

Продовження таблиці 4.11

Економія чисельності працівників виконання виробничої чи управлінської функції яких було автоматизовано у році що аналізується, осіб	-	1	1	1	1
Кількість функцій, які виконуються вручну у році до впровадження результатів нової науково-технічної розробки, шт	8000	-	-	-	-
Прогнозоване зростання кількості виробничих чи інформаційно-технічних управлінських функцій, виконання яких автоматизується, у році що аналізується (відносно року до впровадження даної розробки), шт	-	2000	2000	2000	2000

В даному випадку майбутній економічний ефект та ефективність буде формуватися на основі використання таких показників: $\Delta\Pi_y$ – зростання прибутку підприємства внаслідок зниження витрат на оплату праці працівників,

які виконують окремі виробничі чи інформаційно-технічні управлінські функції, грн. Причому $\Delta\Pi_{я}$ може бути визначено як:

$$\Delta\Pi_{я} = \frac{ЧП \cdot ЗП \cdot 12}{N} - \frac{(0,2...0,6) \cdot ЗВ}{\Delta N_i}, \quad (4.18)$$

де $ЧП$ – чисельність працівників, які виконують визначені функції вручну, прийmemo 4 особи; $ЗП$ – середня заробітна плата працівника, який виконує відповідну функцію вручну, прийmemo 10000,00 грн; $ЗВ$ – приблизні витрати на розробку автоматизованої системи управління, прийmemo 154374,46 грн; N – кількість функцій, які виконуються вручну у році до впровадження результатів нової науково-технічної розробки, 8000 шт; ΔN_i – прогнозоване зростання кількості виробничих чи інформаційно-технічних управлінських функцій, виконання яких автоматизується, у році що аналізується (відносно року до впровадження даної розробки), шт.

Зростання прибутку підприємства в 1-й рік впровадження розробки
 $\Delta\Pi_{я} = 4 \cdot 10000,00 \cdot 12 / 8000 - 0,5 \cdot 154374,46 / 2000 = 21,41$ грн/функц.

Зростання прибутку підприємства в 2-й рік впровадження розробки
 $\Delta\Pi_{я} = 4 \cdot 10000,00 \cdot 12 / 8000 - 0,5 \cdot 154374,46 / 4000 = 40,70$ грн/функц.

Зростання прибутку підприємства в 3-й рік впровадження розробки
 $\Delta\Pi_{я} = 4 \cdot 10000,00 \cdot 12 / 8000 - 0,5 \cdot 154374,46 / 6000 = 47,14$ грн/функц.

Зростання прибутку підприємства в 4-й рік впровадження розробки
 $\Delta\Pi_{я} = 4 \cdot 10000,00 \cdot 12 / 8000 - 0,5 \cdot 154374,46 / 8000 = 50,35$ грн/функц.

$\Pi_{я}$ – прибуток, який отримує підприємство від автоматизації виконання окремої виробничої чи інформаційно-технічної управлінської функції у кожному із років після впровадження науково-технічної розробки, грн. Даний прибуток можна приблизно оцінити виходячи з формули:

$$P_{\text{я}} = \frac{\Delta\text{ЧП} \cdot 3\text{П} \cdot 12}{N}, (4.19)$$

де $\Delta\text{ЧП}$ – економія чисельності працівників виконання виробничої чи управлінської функції яких було автоматизовано у році що аналізується, осіб;

Прибуток який отримує підприємство від автоматизації функції в 1-й рік
 $P_{\text{я}} = 1 \cdot 10000,00 \cdot 12 / 8000 = 15$ грн/функц.

Прибуток який отримує підприємство від автоматизації функції в 2-й рік
 $P_{\text{я}} = 1 \cdot 10000,00 \cdot 12 / 8000 = 15$ грн/функц.

Прибуток який отримує підприємство від автоматизації функції в 3-й рік
 $P_{\text{я}} = 1 \cdot 10000,00 \cdot 12 / 8000 = 15$ грн/функц.

Прибуток який отримує підприємство від автоматизації функції в 4-й рік
 $P_{\text{я}} = 1 \cdot 10000,00 \cdot 12 / 8000 = 15$ грн/функц.

Збільшення чистого прибутку підприємства $\Delta\Pi_i$ для кожного із років, протягом яких очікується отримання позитивних результатів від можливого впровадження науково-технічної розробки, розраховуємо за формулою:

$$\Delta\Pi_i = (\Delta P_{\text{я}} \cdot N + P_{\text{я}} \cdot \Delta N)_i, (4.20)$$

де $\Delta P_{\text{я}}$ – покращення основного якісного показника від впровадження на підприємстві результатів науково-технічної розробки у році що аналізується;

N – основний кількісний показник, який визначає обсяг діяльності підприємства у році до впровадження результатів нової науково-технічної розробки;

$P_{\text{я}}$ – основний якісний показник, який визначає результати діяльності підприємства у кожному із років після впровадження науково-технічної розробки;

ΔN – зміна основного кількісного показника діяльності підприємства в результаті впровадження науково-технічної розробки у році що аналізується.

Збільшення чистого прибутку підприємства в 1-й рік впровадження
 $\Delta\Pi_i = 21,41 \cdot 8000 + 15 \cdot 2000 = 201251,07$ грн.

Збільшення чистого прибутку підприємства в 2-й рік впровадження
 $\Delta\Pi_i = 40,70 \cdot 8000 + 15 \cdot (2000 + 2000) = 385625,54$ грн.

Збільшення чистого прибутку підприємства в 3-й рік впровадження
 $\Delta\Pi_i = 47,14 \cdot 8000 + 15 \cdot (2000 + 2000 + 2000) = 467083,69$ грн.

Збільшення чистого прибутку підприємства в 4-й рік впровадження
 $\Delta\Pi_i = 50,35 \cdot 8000 + 15 \cdot (2000 + 2000 + 2000 + 2000) = 522812,77$ грн.

Приведена вартість збільшення всіх чистих прибутків $ПП$, що їх може отримати замовник від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, (4.21)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,12$;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання замовником додаткових чистих прибутків у цьому році.

$$\begin{aligned} III &= 201251,07/(1+0,12)^1 + 385625,54/(1+0,12)^2 + 467083,69/(1+0,12)^3 + \\ &+ 522812,77/(1+0,12)^4 = 179688,46 + 307418,32 + 332460,95 + 332256,97 = \\ &= 23456837,10 \text{ грн.} \end{aligned}$$

Величина початкових інвестицій PV , які замовник має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{инв} \cdot 3B, (4.22)$$

де $k_{инв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{инв} = 2$;

$3B$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 154374,46 грн.

$$PV = k_{инв} \cdot 3B = 2 \cdot 154374,46 = 308748,93 \text{ грн.}$$

Абсолютний економічний ефект $E_{абс}$ для розробника від можливого впровадження науково-технічної розробки становитиме:

$$E_{абс} = III - PV (4.23)$$

де III – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 1151824,69 грн;

PV – теперішня вартість початкових інвестицій, 308748,93 грн.

$$E_{абс} = III - PV = 1151824,69 - 308748,93 = 843075,76 \text{ грн.}$$

Внутрішня економічна дохідність інвестицій E_g , які можуть бути вкладені розробником у впровадження та комерціалізацію науково-технічної розробки:

$$E_g = T_{жс} \sqrt{1 + \frac{E_{абс}}{PV}} - 1, (4.24)$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, 843075,76 грн;

PV – теперішня вартість початкових інвестицій, 308748,93 грн;

$T_{ж}$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримання позитивних результатів від її впровадження, 4 роки.

$$E_{\epsilon} = T_{ж} \sqrt[4]{1 + \frac{E_{абс}}{PV}} - 1 = (1 + 843075,76/308748,93)^{1/4} = 0,390.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій τ_{min} :

$$\tau_{min} = d + f, (4.25)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2021 році в Україні $d = 0,1$;

f – показник, що характеризує ризикованість вкладення інвестицій, приймемо 0,08.

$\tau_{min} = 0,1 + 0,08 = 0,18 < 0,390$ свідчить про те, що внутрішня економічна дохідність інвестицій E_{ϵ} , які можуть бути вкладені розробником у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Автоматизована система управління стоматологічною клінікою. Розробка модуля ведення пацієнтів» доцільно.

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені розробником у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_g}, (4.26)$$

де E_g – внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = 1 / 0,390 = 2,57 \text{ р.}$$

$T_{ок} < 3$ -х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати розробника профінансувати впровадження даної розробки для застосування в діяльності підприємства.

Висновки до розділу

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Автоматизована система управління стоматологічною клінікою. Розробка модуля ведення пацієнтів» становить 41,7 бала, що свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий).

При оцінюванні за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,51 рази.

Також термін окупності становить 2,57 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати розробника до впровадження даної розробки при отриманні ефекту в розмірі 843075,76 грн.

Отже, можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Автоматизована система управління стоматологічною клінікою. Розробка модуля ведення пацієнтів».

ВИСНОВКИ

В ході виконання даної роботи було розроблено програму, яка дозволяє полегшити ведення пацієнтів стоматологічної клініки.

Додаток було написано на мові програмування Java з використанням технології JavaFX та додатку SceneBuilder.

Було визначено, що пакети з розширенням fxml являються оптимальним варіантом для побудови графічної частини програмного забезпечення. За допомогою даних пакетів можна відділити визначення інтерфейсу від програмної логіки. Також дана технологія дозволила змінити визначення файлу fxml без перекомпіляції додатків.

В першому розділі проведено дослідження функціональних можливостей сучасних програмних рішень, а саме систем Smilestone, Dentaltap, Dental Pro, hCue.

В другому розділі було проведено аналіз програмних засобів, що використовувались в ході розробки автоматизованої системи, а також проведено тестування працездатності заявлених функцій системи ведення пацієнтів.

В третьому розділі було виконано проектування автоматизованої системи управління стоматологічної клініки, для чого розроблено структуру бази даних системи, а також UML-діаграма взаємодії користувача з базою даних й UML-діаграма алгоритмічної структури програми.

В четвертому розділі було проведено дослідження рівня комерційного потенціалу розробки, згідно результатів якого можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Автоматизована система управління стоматологічною клінікою. Частина 2. Розробка модуля ведення пацієнтів»

Для роботи з даним програмним забезпеченням необхідний будь-який комп'ютер та система, де підтримується робота Java додатків.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Правила внутрішнього розпорядку стоматологічної клініки [Електронний ресурс] Режим доступу: <https://artsmilekiev.com.ua/informatsiia/pravy-la-vnutris-hnoho-rozporiadku>
2. Інструкція щодо заповнення форми первинної облікової документації [Електронний ресурс] Режим доступу: <https://zakon.rada.gov.ua/laws/show/z0-678-12#Text>
3. Dental Pro [Електронний ресурс] Режим доступу: <https://dental-pro.online-/features/>
4. dentaltap [Електронний ресурс] Режим доступу: <https://dentaltap.com/ru/>
5. hCue [Електронний ресурс] Режим доступу: <https://www.wolterskluwer.com/-en/solutions/ct-corporation/hcue-entity-management>
6. Кристофер Д. Д. Введення в системи баз даних. Вільямс, 2010. 655 с.
7. Прикладна програма [Електронний ресурс] Режим доступу: https://dbn.co.ua/-blog/prikladna_programa/2016-12-20-14657
8. Visual Database Design with MySQL Workbench [Електронний ресурс] Режим доступу: <https://www.section.io/engineering-education/visual-database-design-with-mysql-workbench/>
9. DBManager [Електронний ресурс] Режим доступу: <https://www.softpedia.com-/get/Internet/Servers/DatabaseUtils/DBManager.shtml>
10. Обзор инструментов для работы с MySQL [Електронний ресурс] Режим доступу: <https://www.webmasters.by/-articles/review-po/96-mysql-develoment-tools.html>

11. GUI [Електронний ресурс] Режим доступу: <https://www.computerhope.com/jargon/g/gui.htm>SQLyog
12. Створення баз даних [Електронний ресурс] Режим доступу: <https://sites.google.com/site/databaseinform11/stvorennia-vlasnoie-bazi-danih>
13. IntelliJ IDEA [Електронний ресурс] Режим доступу: <https://www.jetbrains.com/ru-ru/idea/>
14. Scene Builder [Електронний ресурс] Режим доступу: <https://code.m-akery.com/h/uk/library/javafx-tutorial/part1/>
15. JavaFX Tutorial: FXML and SceneBuilder [Електронний ресурс] Режим доступу: <https://www.vojtechruzicka.com/javafx-fxml-scene-builder/>
16. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.
17. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепка – Вінниця : ВНТУ, 2016. – 113 с.
18. JavaFX Scene Builder [Електронний ресурс] Режим доступу: <https://www.oracle.com/java/technologies/javase/javafxscenebuilder-info.html>
19. Рудаков А. В. Технология разработки программных продуктов : учебник для студ. сред. проф. образования / А. В. Рудаков. 7-е изд., стер. М. : Издательский центр "Академия", 2012. - 208 с.
20. The Increment/Decrement Operators [Електронний ресурс] Режим доступу: <https://riptutorial.com/java/example/775/the-increment-decrement-operators--plusplus---->
21. What is the Difference Between i++ and ++i in Java? [Електронний ресурс]

Режим доступа: <https://www.geeksforgeeks.org/what-is-the-difference-between-i-and-i-in-java/>

22. Давыдов В.Г. Программирование и основы алгоритмизации: Учеб. пособие. / В.Г. Давыдов. М.: Высш. шк., 2003. 447 с
23. Java Database Connectivity with 5 Steps [Электронный ресурс] Режим доступа: <https://www.javatpoint.com/steps-to-connect-to-the-database-in-java>
24. Downloading and Installing Java DB [Электронный ресурс] Режим доступа: <http://www.herongyang.com/JDBC/Java-DB-Derby-Installation.html>
25. JavaFX Tutorial [Электронный ресурс] Режим доступа: <http://tutorials.jenkov.com/javafx/index.html#why-javafx>
26. Шилдт Г.6. Java: руководство для начинающих. М.: Вильямс, 2008. 720 с.
27. Руководство по JavaFX [Электронный ресурс] Режим доступа: <https://meta-nit.com/java/javafx/>
28. Introduction to JavaFx [Электронный ресурс] Режим доступа: <https://www.baeldung.com/javafx>
29. Scene Builder [Электронный ресурс] Режим доступа: <https://gluonhq.com/products/scene-builder/>
30. Scene Builder Installation [Электронный ресурс] Режим доступа: <https://riptutorial.com/javafx/topic/5445/scene-builder>
31. Васильев А.Н. Самоучитель Java с примерами и программами. СПб.: Наука и Техника, 2011. 352 с.
32. JavaFX Scene Builder: Getting Started with JavaFX Scene Builder [Электронный ресурс] Режим доступа: <https://docs.oracle.com/javase/8/scene-builder-2/get-started-tutorial/index.html>
33. Object-oriented programming [Электронный ресурс] Режим доступа: <https://searcharchitecture.techtarget.com/definition/object-oriented-programming-OOP>

34. Getting Started with AWS [Електронний ресурс] Режим доступу:
https://aws.amazon.com/getting-started/?sc_icontent=awssm-evergreen-getting_started&sc_iplace
35. Java Programming Language [Електронний ресурс] Режим доступу:
<https://www.sciencedirect.com/topics/computer-science/java-programming-language>
36. Алгоритми і структура даних: Навчальний посібник / В.М.Ткачук. – Івано-Франківськ : Видавництво Прикарпатського національного університету імені Василя Стефаника, 2016. 286 с.
37. About: Java [Електронний ресурс] Режим доступу: [https://dbpedia.org/page/Java_\(programming_language\)](https://dbpedia.org/page/Java_(programming_language))
38. Use Java 8 language features and APIs [Електронний ресурс] Режим доступу: <https://developer.android.com/studio/write/java8-support>
39. Java – Overview [Електронний ресурс] Режим доступу: https://www.tutorialspoint.com/java/java_overview.htm
40. Эккель Б. Философия Java. 4-е изд. СПб. : Питер, 2011. 640 с.
41. Do great things with Windows [Електронний ресурс] Режим доступу:
<https://www.microsoft.com/en-us/windows/features?activetab=NewPopular>
42. Создание и удаление базы данных [Електронний ресурс] Режим доступу:
<https://metanit.com/sql/sqlserver/3.1.php>
43. Процедура отключения и подключения БД [Електронний ресурс] Режим доступу: <https://shels.com.ua/blog.htm?b=74>
44. Томас Кормен, Чарльз Лейзерсон, Рональд Ривест і Кліффорд Штайн. Вступ в алгоритми, 2-ге видання. MIT Press and McGraw-Hill, 2001. Розділ
45. Пример базы данных SQL Server для обучения SQL [Електронний ресурс] Режим доступу: <https://betacode.net/10241/sample-sql-server-database-for-learning-sql>

46. Что такое схема базы данных? [Электронный ресурс] Режим доступа: <https://www.lucidchart.com/pages/ru/%D1%81%D1%85%D0%B5%D0%BC%D1%8B%D0>
47. Создание базы данных на SQL-сервере [Электронный ресурс] Режим доступа: <https://support.kaspersky.ru/KS4Sharepoint/9.3/ru-RU/36770.htm>
48. Виды переменных [Электронный ресурс] Режим доступа: <https://www.examclouds.com/ru/java/java-core-russian/integer-types>
49. Convert String to int or Integer in Java [Электронный ресурс] Режим доступа: <https://www.baeldung.com/java-convert-string-to-int-or-integer>
50. Основы работы с MSSQL [Электронный ресурс] Режим доступа: https://manifold.net/info/real_sql.shtml?gclid=Cj0KCQjwwNWKBhDAARIsAJ8
51. Java int keyword [Электронный ресурс] Режим доступа: <https://www.javatpoint.com/int-keyword-in-java>
52. M. Y. Appiah, M. Sasikath, R. Makrickaite, M. Gusaite, Robust Voice Activity Detection and Noise Reduction Mechanism — Institute of Electronics Systems, Aalborg University, 2015. — 81 с.
53. Стандарт SQL [Электронный ресурс] Режим доступа: https://www.sql.ru/docs/sql/u_sql/ch1.shtml
54. Алгоритми та структури даних. Навчальний посібник / Т. О. Коротєєва. Львів : Видавництво Львівської політехніки, 2014. - 280 с.
55. JavaFX Fonts [Электронный ресурс] Режим доступа: <http://tutorials.jenkov.com/javafx/fonts.html>
56. Types of Java Applications [Электронный ресурс] Режим доступа: <https://www.javatpoint.com/java-tutorial>
57. Блинов И.Н., Романчик В. С. Java. Методы программирования : уч.-мет. пособие / И. Н. Блинов, В. С. Романчик. Минск : издательство "Четыре

четверти", 2013. 896 с.

58. Блинов, И.Н. Java. Промышленное программирование : практ. пособие / И.Н. Блинов, В.С. Романчик. - Минск : УниверсалПресс, 2007. - 704 с.
59. Tutorial: Java-based Migrations [Электронный ресурс] Режим доступа: <https://flywaydb.org/documentation/tutorials/java>
60. How to Use Windows 10 [Электронный ресурс] Режим доступа: <https://www.laptopmag.com/articles/how-to-use-windows-10>

ДОДАТКИ

Додаток А
(обов'язковий)
ВНТУ

ЗАТВЕРДЖУЮ

Завідувач кафедри КСУ

д.т.н., проф. В.М. Дубовой

“ 30 ” 09 2021 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи

«Автоматизована система управління стоматологічною клінікою. Частина 2.

Розробка модуля ведення пацієнтів»

08-01.МКР.006.00.000 ТЗ

Виконав: студент 2-го курсу, групи
2АКІТ-20м

спеціальності 151 – Автоматизація та
комп'ютерно-інтегровані технології

(шифр і назва спеціальності)

Сергій Серета _____

(ім'я та прізвище)

(підпис)

Керівник: к.т.н., доцент каф. КСУ

Марія Юхимчук _____

(ім'я та прізвище)

(підпис)

Вінниця 2021

1. Назва та галузь застосування

1.1. Назва – Автоматизована система управління стоматологічною клінікою.
Частина 2. Розробка модуля ведення пацієнтів.

1.2. Галузь застосування – Комп’ютеризовані системи ведення пацієнтів в стоматологічній клініці.

2. Підстава для проведення розробки.

Тема магістерської кваліфікаційної роботи затверджена наказом по ВНТУ №277 від “24 “ 09 2021 р.

3. Мета та призначення розробки.

Метою магістерської кваліфікаційної роботи є підвищення ефективності роботи й автоматизація систем ведення пацієнтів, що дозволить зручно заносити пацієнтів в базу даних.

4. Джерела розробки.

Магістерська кваліфікаційна робота виконується вперше. В ході проведення розробки повинні використовуватись такі документи:

1. Медична інформатика : підруч. для студентів мед. ВНЗ /за ред. В. Г. Кнігавка. – Харків : ХНМУ, 2015. – 243 с.
2. Трегуб В. Проектування систем автоматизації. – М.: Видавництво «Ліра-К», 2014. - 162 с.
3. Джонсон Р. Патерни об’єктно-орієнтованого програмування / Гамма Э., Хелм Р., Влссидес Д.. - М.: Издательство «ЛЮРИ», 2020. - 159 с.
4. Осипов Д.Л. Технології проектування баз даних / М.: Издательство «ДМК Пресс», 2007. - 284 с.

5. Вимоги до розробки.

5.1. Перелік головних функцій:

- реєстрація та авторизація в системі;
- використання бази даних пацієнтів;
- призначення дати й часу прийому
- запис пацієнта на прийом.
- перевірка на наявність вільного часу для прийому;
- призначення методу лікування.
- запис скарг пацієнта
- показ інформації зареєстрованого пацієнту
- перегляд історії прийому пацієнта.

5.2. Основні технічні вимоги до розробки.

5.2.1. Вимоги до програмної платформи:

- WINDOWS 7\8\10.

5.2.2. Умови експлуатації системи:

- робота на стандартних ПЕОМ в приміщеннях зі стандартними умовами;
- можливість цілодобового функціонування системи;
- текст програмного забезпечення системи є цілком закритим.

6. Стадії та етапи розробки.

6.1 Пояснювальна записка:

- Дослідження актуальності поставленої задачі 04.10.2021р.
- Аналіз функцій автоматизованих систем
Ведення пацієнтів стоматологічної клініки 02.10.2021р.
- Розробка структури програмного забезпечення
системи 13.10.2021р.
- Апробація результатів дослідження 03.11.2021р.
- Оформлення пояснювальної записки,
графічного матеріалу і презентації 08.12.2021р.

7. Порядок контролю і приймання.

- 7.1. Хід виконання роботи контролюється керівником роботи. Рубіжний контроль провести до «7» грудня 2021 р.
- 7.2. Атестація проекту здійснюється на попередньому захисті. Попередній захист магістерської кваліфікаційної роботи провести до «10» грудня 2021 р.
- 7.3. Підсумкове рішення щодо оцінки якості виконання роботи приймається на засіданні ЕК.
- 7.4. Захист магістерської кваліфікаційної роботи провести до «29» грудня 2021 р.

Додаток Б

Код програми

Клас HelloApplication

```
package com.example.patientcontrol;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.stage.Stage;
import java.io.IOException;

public class HelloApplication extends Application {

    @Override

    public void start(Stage stage) throws IOException {

        FXMLLoader fxmlLoader = new
FXMLLoader(HelloApplication.class.getResource("hello-view.fxml"));

        Scene scene = new Scene(fxmlLoader.load(), 700, 400);

        stage.setTitle("Hello, dantist!");

        stage.setScene(scene);

        stage.show();

    }
}
```

Клас Const

```
package com.example.patientcontrol;

public class Const {

    public static final String USER_TABLE = "users";

    public static final String USERS_ID = "idusers";

    public static final String USERS_FIRSTNAME = "firstname";

    public static final String USERS_LASTNAME = "lastname";

    public static final String USERS_PHONENUMBER = "phoneNumber";

    public static final String USERS_EMAIL = "email";

    public static final String USERS_TIMEOFRECEIPT = "timeOfReceipt";

    public static final String USERS_DATEOFRECEIPT = "dateOfReceipt";

}
```

Клас DAtabaseHandler

```
package com.example.patientcontrol;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;
```

```
public class DatabaseHandler extends Configs {  
    Connection dbConnection;  
  
    public Connection getDbConnection() throws ClassNotFoundException,  
SQLException{  
        String connectionString = "jdbc:mysql://" + dbHost + ":" + dbPort + "/" +  
dbName;  
  
        Class.forName("com.mysql.jdbc.Driver");  
  
        dbConnection = DriverManager.getConnection(connectionString, dbUser,  
dbPass);  
  
        return dbConnection;  
    }  
}
```

app.fxml

```
<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.Button?>  
<?import javafx.scene.control.Label?>  
<?import javafx.scene.control.TextField?>  
<?import javafx.scene.layout.AnchorPane?>  
<?import javafx.scene.text.Font?>
```

```
<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="508.0" prefWidth="899.0" style="-fx-
background-color: #2E3348;" xmlns="http://javafx.com/javafx/16"
xmlns:fx="http://javafx.com/fxml/1"
fx:controller="com.example.patientcontrol.HomeController">
```

```
<children>
```

```
<AnchorPane layoutX="-5.0" layoutY="95.0" prefHeight="404.0"
prefWidth="897.0" style="-fx-background-color: #FAFAFA;">
```

```
<children>
```

```
<TextField fx:id="userName" layoutX="261.0" layoutY="67.0"
opacity="0.69" prefHeight="40.0" prefWidth="154.0" promptText="User name"
style="-fx-background-color: #FAFAFA; -fx-border-color: silver; -fx-border-radius:
5;">
```

```
<font>
```

```
<Font size="17.0" />
```

```
</font>
```

```
</TextField>
```

```
<Label layoutX="84.0" layoutY="59.0" prefHeight="56.0"
prefWidth="113.0" text="Прізвище" textFill="#171717">
```

```
<font>
```

```
<Font name="MV Boli" size="21.0" />
```

```
</font>
```

```
</Label>
```

```
<Label layoutX="84.0" layoutY="121.0" prefHeight="56.0"
prefWidth="41.0" text="Ім'я" textFill="#171717">

    <font>

        <Font name="MV Boli" size="21.0" />

    </font>

</Label>

<TextField fx:id="userName1" layoutX="261.0" layoutY="129.0"
opacity="0.69" prefHeight="40.0" prefWidth="154.0" promptText="User name"
style="-fx-background-color: #FAFAFA; -fx-border-color: silver; -fx-border-radius:
5;">

    <font>

        <Font size="17.0" />

    </font>

</TextField>

<Label layoutX="84.0" layoutY="183.0" prefHeight="56.0"
prefWidth="172.0" text="Номер телефону" textFill="#171717">

    <font>

        <Font name="MV Boli" size="21.0" />

    </font>

</Label>

<TextField fx:id="userName11" layoutX="261.0" layoutY="191.0"
opacity="0.69" prefHeight="40.0" prefWidth="154.0" promptText="User name"
```



```
style="-fx-background-color: #FAFAFA; -fx-border-color: silver; -fx-border-radius:
5;">
```

```
<font>
```

```
<Font size="17.0" />
```

```
</font>
```

```
</TextField>
```

```
<TextField fx:id="userName2" layoutX="659.0" layoutY="67.0"
opacity="0.69" prefHeight="40.0" prefWidth="154.0" promptText="User name"
style="-fx-background-color: #FAFAFA; -fx-border-color: silver; -fx-border-radius:
5;">
```

```
<font>
```

```
<Font size="17.0" />
```

```
</font>
```

```
</TextField>
```

```
<Label layoutX="482.0" layoutY="59.0" prefHeight="56.0"
prefWidth="61.0" text="Email" textFill="#171717">
```

```
<font>
```

```
<Font name="MV Boli" size="21.0" />
```

```
</font>
```

```
</Label>
```

```
<TextField fx:id="userName3" layoutX="659.0" layoutY="129.0"
opacity="0.69" prefHeight="40.0" prefWidth="154.0" promptText="User name"
```

```
style="-fx-background-color: #FAFAFA; -fx-border-color: silver; -fx-border-radius:
5;">
```

```
<font>
```

```
<Font size="17.0" />
```

```
</font>
```

```
</TextField>
```

```
<Label layoutX="482.0" layoutY="121.0" prefHeight="56.0"
prefWidth="134.0" text="Час прийому" textFill="#171717">
```

```
<font>
```

```
<Font name="MV Boli" size="21.0" />
```

```
</font>
```

```
</Label>
```

```
<TextField fx:id="userName4" layoutX="659.0" layoutY="191.0"
opacity="0.69" prefHeight="40.0" prefWidth="154.0" promptText="User name"
style="-fx-background-color: #FAFAFA; -fx-border-color: silver; -fx-border-radius:
5;">
```

```
<font>
```

```
<Font size="17.0" />
```

```
</font>
```

```
</TextField>
```

```
<Label layoutX="482.0" layoutY="183.0" prefHeight="56.0"
prefWidth="147.0" text="Дата прийому" textFill="#171717">
```

```
<font>
```

```

    <Font name="MV Boli" size="21.0" />

</font>

</Label>

    <Button fx:id="registerButton" layoutX="349.0" layoutY="266.0"
mnemonicParsing="false" prefHeight="39.0" prefWidth="216.0" style="-fx-
background-color: #F39C63;" text="Зареєструвати клієнта">

    <font>

        <Font size="18.0" />

    </font>

</Button>

</children>

</AnchorPane>

<Label layoutX="253.0" layoutY="14.0" prefHeight="52.0"
prefWidth="409.0" text="Patient control" textFill="WHITE">

    <font>

        <Font name="MV Boli" size="50.0" />

    </font>

</Label>

</children>

</AnchorPane>

```

Клас ControllerSignUp

```
public class ControllerSignUp {  
  
    @FXML  
    private ResourceBundle resources;  
  
    @FXML  
    private URL location;  
  
    @FXML  
    private Button SignUpButton;  
  
    @FXML  
    private TextField SignUpNameField;  
  
    @FXML  
    private TextField loginField;  
  
    @FXML  
    private PasswordField passwordField;  
  
    @FXML  
    private TextField signUpLastNameField;  
  
    @FXML  
    void initialize() {
```

```
assert SignUpButton != null : "fx:id=\"SignUpButton\" was not injected: check  
your FXML file 'signUp.fxml.'";
```

```
assert SignUpNameField != null : "fx:id=\"SignUpNameField\" was not  
injected: check your FXML file 'signUp.fxml.'";
```

```
assert loginField != null : "fx:id=\"loginField\" was not injected: check your  
FXML file 'signUp.fxml.'";
```

```
assert passwordField != null : "fx:id=\"passwordField\" was not injected: check  
your FXML file 'signUp.fxml.'";
```

```
assert signUpLastNameField != null : "fx:id=\"signUpLastNameField\" was not  
injected: check your FXML file 'signUp.fxml.'";
```

Файл signUp.fxml

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<?import javafx.scene.control.Button?>
```

```
<?import javafx.scene.control.Label?>
```

```
<?import javafx.scene.control.PasswordField?>
```

```
<?import javafx.scene.control.TextField?>
```

```
<?import javafx.scene.layout.AnchorPane?>
```

```
<?import javafx.scene.text.Font?>
```

```
<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"  
minWidth="-Infinity" prefHeight="630.0" prefWidth="700.0" style="-fx-
```

```

background-color: #2e3348;"          xmlns="http://javafx.com/javafx/17"
xmlns:fx="http://javafx.com/fxml/1"
fx:controller="com.example.demo.ControllerSignUp">

    <children>

        <AnchorPane layoutY="98.0" prefHeight="530.0" prefWidth="700.0" style="-
fx-background-color: #E0FFFF;">

            <children>

                <Label      layoutX="79.0"      layoutY="37.0"      prefHeight="476.0"
prefWidth="545.0" style="-fx-background-color: FFFFFFFF;" />

                <TextField  fx:id="loginField"  layoutX="156.0"   layoutY="288.0"
prefHeight="47.0"  prefWidth="420.0"  promptText="Логін"  style="-fx-border-
radius: 10;">

                    <font>

                        <Font name="Agency FB" size="25.0" />

                    </font>

                </TextField>

                <Label      layoutX="241.0"      layoutY="56.0"      prefHeight="64.0"
prefWidth="250.0" text="Реєстрація" textFill="#727272">

                    <font>

                        <Font name="Agency FB" size="44.0" />

                    </font>

                </Label>

                <Button  fx:id="SignUpButton"  layoutX="231.0"   layoutY="436.0"

```

```
mnemonicParsing="false" prefHeight="51.0" prefWidth="271.0" style="-fx-  
background-color: #f39c63;" text="Зареєструватися" textFill="LIGHTCYAN">
```

```
<font>
```

```
<Font name="Agency FB" size="27.0" />
```

```
</font>
```

```
</Button>
```

```
<PasswordField fx:id="passwordField" layoutX="156.0" layoutY="358.0"  
prefHeight="47.0" prefWidth="420.0" promptText="Пароль">
```

```
<font>
```

```
<Font name="Agency FB" size="25.0" />
```

```
</font>
```

```
</PasswordField>
```

```
<TextField fx:id="SignUpNameField" layoutX="156.0" layoutY="143.0"  
prefHeight="47.0" prefWidth="420.0" promptText="Ім'я " style="-fx-border-radius:  
10;">
```

```
<font>
```

```
<Font name="Agency FB" size="25.0" />
```

```
</font>
```

```
</TextField>
```

```
<TextField fx:id="signUpLastNameField" layoutX="156.0" layoutY="218.0"  
prefHeight="47.0" prefWidth="420.0" promptText="Прізвище" style="-fx-border-  
radius: 10;">
```

```
<font>
```

```
<Font name="Agency FB" size="25.0" />
</font>
</TextField>
</children>
</AnchorPane>
<Label layoutX="114.0" layoutY="24.0" prefHeight="59.0" prefWidth="472.0"
text="Dental control" textFill="LIGHTCYAN">
  <font>
    <Font name="Copperplate Gothic Bold" size="52.0" />
  </font>
</Label>
</children>
</AnchorPane>
Файл visitsOfPatient
<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.DatePicker?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
```



```
<?import javafx.scene.image.ImageView?>
```

```
<?import javafx.scene.layout.AnchorPane?>
```

```
<?import javafx.scene.text.Font?>
```

```
<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity"    prefHeight="794.0"    prefWidth="700.0"    style="-fx-
background-color:      #2e3348;"    xmlns="http://javafx.com/javafx/17"
xmlns:fx="http://javafx.com/fxml/1"
fx:controller="com.example.demo.ControllerHome">
```

```
<children>
```

```
<AnchorPane layoutY="98.0" prefHeight="769.0" prefWidth="700.0" style="-
fx-background-color: #E0FFFF;">
```

```
<children>
```

```
<Label    layoutX="80.0"    layoutY="32.0"    prefHeight="719.0"
prefWidth="545.0" style="-fx-background-color: FFFFFFFF;" />
```

```
<Label    fx:id="exitButton"    layoutX="172.0"    layoutY="668.0"
prefHeight="64.0"    prefWidth="448.0"    text="Вийти в головне меню"
textFill="#727272">
```

```
<font>
```

```
<Font name="Agency FB" size="24.0" />
```

```
</font>
```

```
</Label>
```

```
<ImageView    fx:id="exitImage"    fitHeight="48.0"    fitWidth="51.0"
```

```

layoutX="115.0" layoutY="676.0" pickOnBounds="true" preserveRatio="true">
    <image>
        <Image url="@../../../../java/assets/exit.png" />
    </image>
</ImageView>
<Label    layoutX="170.0"    layoutY="48.0"    prefHeight="71.0"
prefWidth="377.0" text="Візити пацієнта">
    <font>
        <Font size="49.0" />
    </font>
</Label>
<DatePicker    layoutX="127.0"    layoutY="336.0"    prefHeight="48.0"
prefWidth="273.0" promptText="Обрати дату" />
<Label    layoutX="127.0"    layoutY="283.0"    prefHeight="42.0"
prefWidth="280.0" text="Записати на прийом">
    <font>
        <Font size="28.0" />
    </font>
</Label>
<Label    layoutX="199.0"    layoutY="172.0"    prefHeight="42.0"
prefWidth="319.0" text="Список візитів пустий" textFill="#00000088">
    <font>

```

```

    <Font size="28.0" />

</font>

</Label>

<TextField    layoutX="127.0"    layoutY="435.0"    prefHeight="31.0"
prefWidth="273.0" promptText="Місце прийому" />

<Label    layoutX="127.0"    layoutY="393.0"    prefHeight="42.0"
prefWidth="280.0" text="Обрати місце прийому">

    <font>

        <Font size="18.0" />

    </font>

</Label>

<Label    layoutX="127.0"    layoutY="489.0"    prefHeight="42.0"
prefWidth="280.0" text="Обрати час прийому">

    <font>

        <Font size="18.0" />

    </font>

</Label>

<TextField    layoutX="127.0"    layoutY="531.0"    prefHeight="31.0"
prefWidth="273.0" promptText="Час прийому" />

<Button    fx:id="authorizationSignUpButton"    layoutX="273.0"
layoutY="598.0" mnemonicParsing="false" prefHeight="42.0" prefWidth="171.0"
style="-fx-background-color: #f39c63;" text="Зберегти" textFill="LIGHTCYAN">

    <font>

```

```
        <Font name="Agency FB" size="17.0" />
    </font>
</Button>
</children>
</AnchorPane>
<Label layoutX="114.0" layoutY="24.0" prefHeight="59.0" prefWidth="472.0"
text="Dental control" textFill="LIGHTCYAN">
    <font>
        <Font name="Copperplate Gothic Bold" size="52.0" />
    </font>
</Label>
</children>
</AnchorPane>
```

Файл signUp.fxml

```
<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.PasswordField?>
```

```
<?import javafx.scene.control.TextField?>
```

```
<?import javafx.scene.layout.AnchorPane?>
```

```
<?import javafx.scene.text.Font?>
```

```
<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="630.0" prefWidth="700.0" style="-fx-
background-color: #2e3348;" xmlns="http://javafx.com/javafx/17"
xmlns:fx="http://javafx.com/fxml/1"
fx:controller="com.example.demo.ControllerSignUp">
```

```
<children>
```

```
<AnchorPane layoutY="98.0" prefHeight="530.0" prefWidth="700.0" style="-
fx-background-color: #E0FFFF;">
```

```
<children>
```

```
<Label layoutX="79.0" layoutY="37.0" prefHeight="476.0"
prefWidth="545.0" style="-fx-background-color: FFFFFFFF;" />
```

```
<TextField fx:id="loginField" layoutX="156.0" layoutY="288.0"
prefHeight="47.0" prefWidth="420.0" promptText="Логин" style="-fx-border-
radius: 10;">
```

```
<font>
```

```
<Font name="Agency FB" size="25.0" />
```

```
</font>
```

```
</TextField>
```

```
<Label layoutX="241.0" layoutY="56.0" prefHeight="64.0"
```

```
prefWidth="250.0" text="Рєєстрація" textFill="#727272">
```

```
<font>
```

```
<Font name="Agency FB" size="44.0" />
```

```
</font>
```

```
</Label>
```

```
<Button fx:id="SignUpButton" layoutX="231.0" layoutY="436.0"
mnemonicParsing="false" prefHeight="51.0" prefWidth="271.0" style="-fx-
background-color: #f39c63;" text="Зареєструватися" textFill="LIGHTCYAN">
```

```
<font>
```

```
<Font name="Agency FB" size="27.0" />
```

```
</font>
```

```
</Button>
```

```
<PasswordField fx:id="passwordField" layoutX="156.0" layoutY="358.0"
prefHeight="47.0" prefWidth="420.0" promptText="Пароль">
```

```
<font>
```

```
<Font name="Agency FB" size="25.0" />
```

```
</font>
```

```
</PasswordField>
```

```
<TextField fx:id="SignUpNameField" layoutX="156.0" layoutY="143.0"
prefHeight="47.0" prefWidth="420.0" promptText="Ім'я " style="-fx-border-radius:
10;">
```

```
<font>
```

```

        <Font name="Agency FB" size="25.0" />
    </font>
</TextField>

<TextField fx:id="signUpLastNameField" layoutX="156.0" layoutY="218.0"
prefHeight="47.0" prefWidth="420.0" promptText="Прізвище" style="-fx-border-
radius: 10;">

    <font>

        <Font name="Agency FB" size="25.0" />

    </font>

</TextField>

</children>

</AnchorPane>

<Label layoutX="114.0" layoutY="24.0" prefHeight="59.0" prefWidth="472.0"
text="Dental control" textFill="LIGHTCYAN">

    <font>

        <Font name="Copperplate Gothic Bold" size="52.0" />

    </font>

</Label>

</children>

</AnchorPane>

Файл makeAnAppointment.fxml

<?xml version="1.0" encoding="UTF-8"?>

```

```

<?import javafx.scene.control.Label?>

<?import javafx.scene.control.TextField?>

<?import javafx.scene.image.Image?>

<?import javafx.scene.image.ImageView?>

<?import javafx.scene.layout.AnchorPane?>

<?import javafx.scene.text.Font?>

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity"    prefHeight="630.0"    prefWidth="700.0"    style="-fx-
background-color:      #2e3348;"          xmlns="http://javafx.com/javafx/17"
xmlns:fx="http://javafx.com/fxml/1"
fx:controller="com.example.demo.ControllerHome">

    <children>

        <AnchorPane layoutY="98.0" prefHeight="589.0" prefWidth="700.0" style="-
fx-background-color: #E0FFFF;">

            <children>

                <Label    layoutX="79.0"    layoutY="37.0"    prefHeight="527.0"
prefWidth="545.0" style="-fx-background-color: FFFFFFFF;" />

                <Label    fx:id="exitButton"    layoutX="183.0"    layoutY="483.0"
prefHeight="64.0"    prefWidth="448.0"    text="Вийти в головне меню"
textFill="#727272">

                    <font>

```



```

        <Font name="Agency FB" size="28.0" />
    </font>
</Label>
    <ImageView fx:id="exitImage" fitHeight="64.0" fitWidth="57.0"
layoutX="108.0" layoutY="486.0" pickOnBounds="true" preserveRatio="true">
    <image>
        <Image url="@../../../../java/assets/exit.png" />
    </image>
</ImageView>
    <Label layoutX="203.0" layoutY="37.0" prefHeight="71.0"
prefWidth="281.0" text="Записати на прийом">
    <font>
        <Font size="28.0" />
    </font>
</Label>
<Label layoutX="117.0" layoutY="212.0" text="Місце прийому:">
    <font>
        <Font size="28.0" />
    </font>
</Label>
    <TextField layoutX="331.0" layoutY="217.0" prefHeight="30.0"
prefWidth="222.0">

```

```
<font>
  <Font size="14.0" />
</font>
</TextField>
<Label layoutX="117.0" layoutY="280.0" text="Причина:">
  <font>
    <Font size="28.0" />
  </font>
</Label>
  <TextField layoutX="247.0" layoutY="285.0" prefHeight="30.0"
prefWidth="306.0">
  <font>
    <Font size="14.0" />
  </font>
</TextField>
<Label layoutX="117.0" layoutY="344.0" text="Час:">
  <font>
    <Font size="28.0" />
  </font>
</Label>
  <TextField layoutX="180.0" layoutY="349.0" prefHeight="30.0"
prefWidth="373.0">
```

```
<font>
  <Font size="14.0" />
</font>
</TextField>
<Label layoutX="117.0" layoutY="407.0" text="Дата:">
  <font>
    <Font size="28.0" />
  </font>
</Label>
  <TextField    layoutX="193.0"    layoutY="412.0"    prefHeight="30.0"
prefWidth="360.0">
  <font>
    <Font size="14.0" />
  </font>
</TextField>
  <Label    layoutX="167.0"    layoutY="91.0"    prefHeight="71.0"
prefWidth="373.0" text="Пацієнта «Прізвище» «Ім'я» ">
  <font>
    <Font size="28.0" />
  </font>
</Label>
</children>
```

```

</AnchorPane>

<Label layoutX="114.0" layoutY="24.0" prefHeight="59.0" prefWidth="472.0"
text="Dental control" textFill="LIGHTCYAN">

  <font>

    <Font name="Copperplate Gothic Bold" size="52.0" />

  </font>

</Label>

</children>

</AnchorPane>

Файл information.patient

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Label?>

<?import javafx.scene.control.TextArea?>

<?import javafx.scene.control.TextField?>

<?import javafx.scene.image.Image?>

<?import javafx.scene.image.ImageView?>

<?import javafx.scene.layout.AnchorPane?>

<?import javafx.scene.text.Font?>

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="981.0" prefWidth="700.0" style="-fx-

```

```

background-color: #2e3348;"          xmlns="http://javafx.com/javafx/17"
xmlns:fx="http://javafx.com/fxml/1"
fx:controller="com.example.demo.ControllerHome">

    <children>

        <AnchorPane layoutY="98.0" prefHeight="757.0" prefWidth="700.0" style="-
fx-background-color: #E0FFFF;">

            <children>

                <Label layoutX="79.0" layoutY="18.0" prefHeight="720.0"
prefWidth="545.0" style="-fx-background-color: FFFFFFFF;" />

                <Label fx:id="exitButton" layoutX="157.0" layoutY="638.0"
prefHeight="64.0" prefWidth="448.0" text="Вийти в головне меню"
textFill="#727272">

                    <font>

                        <Font name="Agency FB" size="28.0" />

                    </font>

                </Label>

                <ImageView fx:id="exitImage" fitHeight="51.0" fitWidth="49.0"
layoutX="101.0" layoutY="646.0" pickOnBounds="true" preserveRatio="true">

                    <image>

                        <Image url="@../../../../java/assets/exit.png" />

                    </image>

                </ImageView>

                <Label layoutX="182.0" layoutY="36.0" prefHeight="71.0"

```

```
prefWidth="390.0" text="    Серeda Сергій">
    <font>
        <Font size="28.0" />
    </font>
</Label>
<Label    layoutX="158.0"    layoutY="304.0"    prefHeight="64.0"
prefWidth="96.0" text="Статус:">
    <font>
        <Font name="Agency FB" size="28.0" />
    </font>
</Label>
<Label    layoutX="158.0"    layoutY="582.0"    text="Візити    пацієнта"
textFill="#727272">
    <font>
        <Font size="28.0" />
    </font>
</Label>
<Label    layoutX="182.0"    prefHeight="71.0"    prefWidth="390.0"
text="Медична картка пацієнта">
    <font>
        <Font size="28.0" />
    </font>
```

```
</Label>
```

```
<Label layoutX="156.0" layoutY="390.0" text="Скарги">
```

```
<font>
```

```
<Font size="28.0" />
```

```
</font>
```

```
</Label>
```

```
<Label layoutX="156.0" layoutY="447.0" text="Діагноз">
```

```
<font>
```

```
<Font size="28.0" />
```

```
</font>
```

```
</Label>
```

```
<TextField layoutX="257.0" layoutY="313.0" prefHeight="46.0"  
prefWidth="294.0" style="-fx-background-color: #7FFFD4;" text="Проходить курс  
лікування">
```

```
<font>
```

```
<Font size="22.0" />
```

```
</font>
```

```
</TextField>
```

```
<Label layoutX="156.0" layoutY="510.0" text="Метод лікування">
```

```
<font>
```

```
<Font size="28.0" />
```

```
</font>
```

```

</Label>

<TextArea layoutX="269.0" layoutY="391.0" prefHeight="38.0"
prefWidth="338.0" />

<TextArea layoutX="269.0" layoutY="448.0" prefHeight="38.0"
prefWidth="338.0" />

<TextArea layoutX="381.0" layoutY="511.0" prefHeight="37.0"
prefWidth="225.0" />

<ImageView fx:id="exitImage1" fitHeight="51.0" fitWidth="49.0"
layoutX="107.0" layoutY="573.0" pickOnBounds="true" preserveRatio="true">

    <image>

        <Image url="@../../../../java/assets/visit.png" />

    </image>

</ImageView>

<ImageView fitHeight="276.0" fitWidth="267.0" layoutX="227.0"
layoutY="60.0" pickOnBounds="true" preserveRatio="true">

    <image>

        <Image url="@../../../../java/assets/nopicture.png" />

    </image>

</ImageView>

</children>

</AnchorPane>

<Label layoutX="114.0" layoutY="24.0" prefHeight="59.0" prefWidth="472.0"
text="Dental control" textFill="LIGHTCYAN">

```



```

    <font>
        <Font name="Copperplate Gothic Bold" size="52.0" />
    </font>
</Label>
</children>
</AnchorPane>

```

Файл Registration.fxml

```

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.PasswordField?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity"    prefHeight="630.0"    prefWidth="700.0"    style="-fx-
background-color:      #2e3348;"    xmlns="http://javafx.com/javafx/17"
xmlns:fx="http://javafx.com/fxml/1"
fx:controller="com.example.demo.ControllerAuthorization">
    <children>

```

```
<AnchorPane layoutY="98.0" prefHeight="574.0" prefWidth="700.0" style="-fx-background-color: #E0FFFF;">
```

```
<children>
```

```
<Label layoutX="79.0" layoutY="37.0" prefHeight="509.0"
prefWidth="545.0" style="-fx-background-color: FFFFFFFF;" />
```

```
<TextField fx:id="loginField" layoutX="142.0" layoutY="245.0"
prefHeight="47.0" prefWidth="420.0" promptText="Ім'я " style="-fx-border-radius:
10;">
```

```
<font>
```

```
<Font name="Agency FB" size="25.0" />
```

```
</font>
```

```
</TextField>
```

```
<Button fx:id="authorizationSignUpButton" layoutX="254.0"
layoutY="463.0" mnemonicParsing="false" prefHeight="59.0" prefWidth="192.0"
style="-fx-background-color: #f39c63;" text="Зареєструватися"
textFill="LIGHTCYAN">
```

```
<font>
```

```
<Font name="Agency FB" size="17.0" />
```

```
</font>
```

```
</Button>
```

```
<PasswordField fx:id="passwordField" layoutX="142.0" layoutY="383.0"
prefHeight="47.0" prefWidth="420.0" promptText="Пароль">
```

```
<font>
```

```
<Font name="Agency FB" size="25.0" />

</font>

</PasswordField>

<TextField fx:id="loginField1" layoutX="142.0" layoutY="182.0"
prefHeight="47.0" prefWidth="420.0" promptText="Прізвище" style="-fx-border-
radius: 10;">

<font>

<Font name="Agency FB" size="25.0" />

</font>

</TextField>

<Label layoutX="238.0" layoutY="84.0" prefHeight="64.0"
prefWidth="250.0" text="Реєстрація" textFill="#727272">

<font>

<Font name="Agency FB" size="44.0" />

</font>

</Label>

<PasswordField fx:id="passwordField1" layoutX="142.0" layoutY="313.0"
prefHeight="47.0" prefWidth="420.0" promptText="Логін">

<font>

<Font name="Agency FB" size="25.0" />

</font>

</PasswordField>
```

```
</children>

</AnchorPane>

<Label layoutX="114.0" layoutY="24.0" prefHeight="59.0" prefWidth="472.0"
text="Dental control" textFill="LIGHTCYAN">

  <font>

    <Font name="Copperplate Gothic Bold" size="52.0" />

  </font>

</Label>

</children>

</AnchorPane>
```

Додаток В
(обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

Автоматизована система управління стоматологічною клінікою. Частина 2.

Розробка модуля ведення пацієнтів

Перелік ілюстративних матеріалів:

1. UML-діаграма алгоритмічної структури програми;
2. UML-діаграма послідовності «Авторизація»;
3. UML-діаграма послідовності «Реєстрація нового користувача»;
4. UML-діаграма послідовності «Пошук пацієнта»;
5. UML-діаграма послідовності «Редагування профілю пацієнта»;
6. Сторінка авторизації користувача
7. Головне меню;
8. Екран пошуку пацієнта;
9. Вікно візитів пацієнта;
10. Медична картка пацієнта.

Виконав: студент 2-го курсу, групи
2АКІТ-20м
спеціальності 151 – Автоматизація та
комп'ютерно-інтегровані технології
(шифр і назва спеціальності)

Сергій Серета

(ім'я та прізвище)

Керівник: доц. кафедри КСУ

Марія Юхимчук

(ім'я та прізвище)

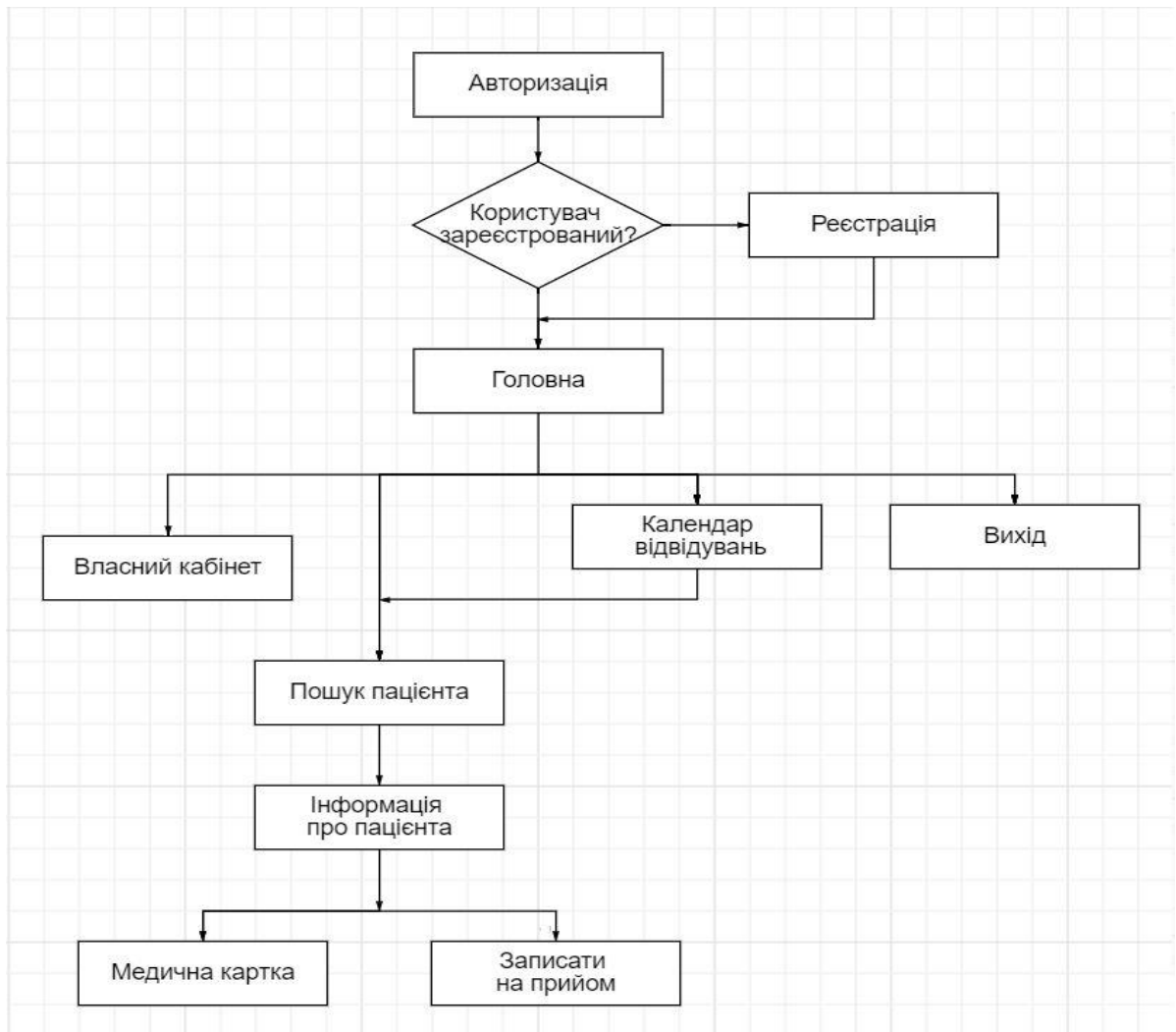
«15» _____ 12 _____ 2021 р.

Опонент: к.т.н., доц. каф. АІВТ

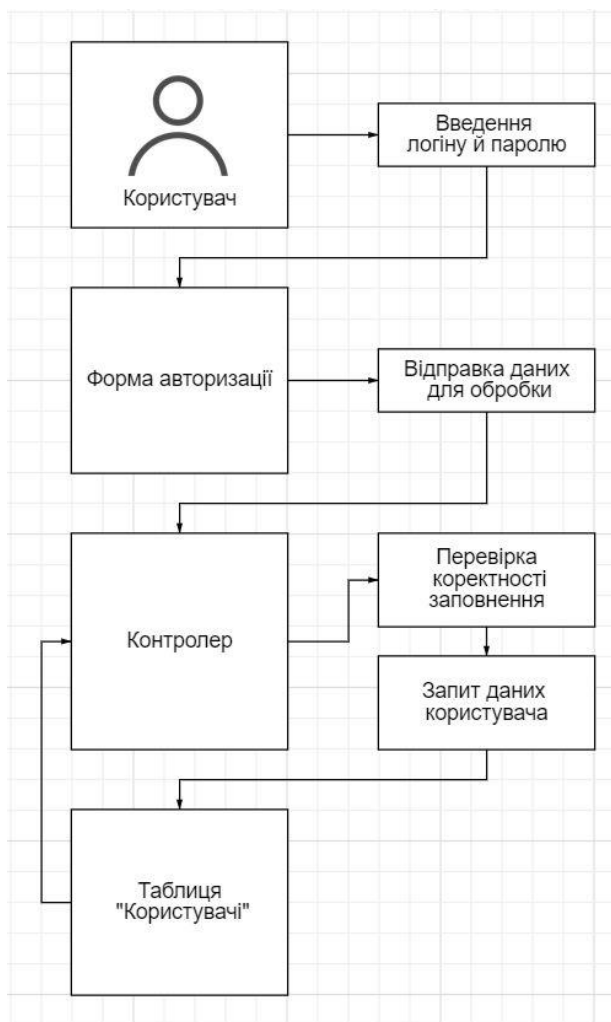
Марія Барабан

(ім'я та прізвище)

«16» _____ 12 _____ 2021 р.



UML-діаграма алгоритмічної структури програми



UML-діаграма послідовності «Авторизація»



UML-діаграма послідовності «Реєстрація нового користувача»



UML-діаграма послідовності «Пошук пацієнта»



UML-діаграма послідовності «Редагування профілю пацієнта»





DENTAL CONTROL

Авторизація

[Вхід](#)
або
[Зареєструватися](#)

Сторінка авторизації користувача

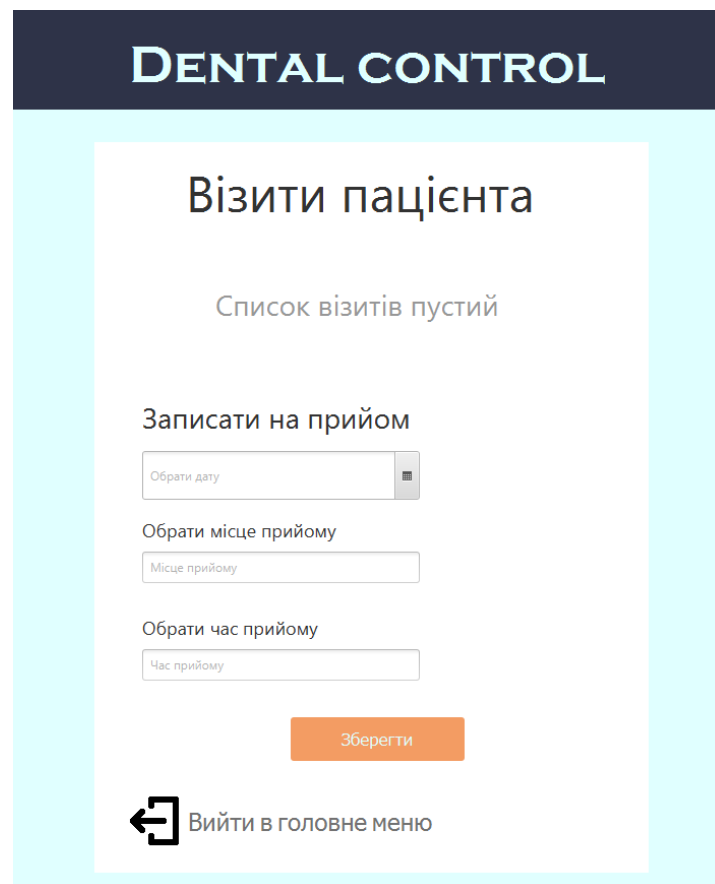
DENTAL CONTROL

-  Знайти пацієнта
-  Календар відвідувань
-  Мій кабінет
-  Вийти

Головне меню




Екран пошуку пацієнта



Вікно візитів пацієнта

DENTAL CONTROL

Медична картка пацієнта
Середа Сергій





Статус: Проходить курс лікування

Скарги

Діагноз

Метод лікування

 Візити пацієнта

 Вийти в головне меню

Медична картка пацієнта