

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

Магістр

(назва освітнього ступеня)

на тему: Розробка платформи медичних послуг та CRM системи
для керування нею із використанням Next.js

Виконав(ла): студент(ка) 6 курсу, групи СПм-61
спеціальності 121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

	<u>Гурак М.П.</u> (прізвище та ініціали)
Керівник	<u>Михалик Д.М.</u> (прізвище та ініціали)
Нормоконтроль	<u>Стоянов Ю.М.</u> (прізвище та ініціали)
Завідувач кафедри	<u>Петрик М.Р.</u> (прізвище та ініціали)
Рецензент	<u></u> (прізвище та ініціали)

РЕФЕРАТ

Кваліфікаційна робота на тему «Розробка платформи медичних послуг та CRM системи для керування нею із використанням Next.js» написана Гураком Максимом Петровичем студентом Тернопільського національного технічного університету імені Івана Пулюя, Факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра програмної інженерії, група СПМ-61.

Відомості про обсяг: сторінок – __, рисунків – 36, таблиць – 10, частин – 3, додатків – 3, посилань – 11.

У першому розділі проаналізовано існуючі аналоги медичних платформ, складено специфікацію вимог до системи та описано поведінку системи за допомогою діаграм послідовності та варіантів використання.

У другому розділі вибрано програмне середовище для розроблення медичної платформи та CRM системи для керування нею із використанням Next.js, розроблено архітектуру системи, спроектовано та створено реляційну базу даних для збереження інформації та вибрано інфраструктуру для розгортання системи.

У третьому розділі розроблено структуру проектів медичної платформи та структуру веб-вузла CRM системи. Реалізовано програмне забезпечення для медичної платформи та CRM системи для керування нею із використанням Next.js.

Ключові слова: лікар, лікарня, відділення, сертифікат, галерея, програма лікування, медичний туризм, медична діагностика, керування лікарнею, керування лікарем, керування відділенням, керування програмою лікування, медична бібліотека, захворювання, медичний блог.

ABSTRACT

The qualification work on the topic "Development of a medical services platform and a CRM system for its management using Next.js" was written by Gurak Maksym Petrovych, a student of Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Software Engineering, group SPm-61.

Information about the volume: pages – __, figures – 36, tables – 10, parts – 3, appendices – 3, references – 11.

In the first section analyzes existing analogs of medical platforms, system requirements specification is drawn up, and system behavior is described with the help of sequence diagrams and use cases.

In the second section, a software environment for developing a medical platform and a CRM system for managing it using Next.js was selected, the system architecture was developed, a relational database was designed and created to store information, and an infrastructure was selected for system deployment.

In the third section, the structure of medical platform projects and the structure of the web node of the CRM system are developed. Implemented software for medical platform and CRM system to manage it using Next.js.

Keywords: doctor, hospital, department, certificate, gallery, treatment program, medical tourism, medical diagnosis, hospital management, doctor management, department management, treatment program management, medical library, disease, medical blog.

ЗМІСТ

ПЕРЕЛІК ОСНОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ.....	7
ВСТУП.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА СПЕЦИФІКАЦІЇ ВИМОГ	
1.1 Коротка характеристика об’єкту управління.....	10
1.2 Аналіз існуючих аналогів.....	11
1.3 Специфікація вимог для веб-застосунку	21
Висновки до розділу.	27
2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	
2.1 Вибір програмного середовища розробки.....	28
2.2 Розроблення архітектури програмної системи	32
2.3 Розробка бази даних системи	37
2.4 Інфраструктури системи	40
Висновки до розділу	46
3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	
3.1 Структурна схема додатку	47
3.2 Реалізація архітектури програмного забезпечення	54
Висновки до розділу	68
ВИСНОВКИ.....	69
ПЕРЕЛІК ПОСИЛАНЬ НА ДЖЕРЕЛА	70
ДОДАТКИ.....	71
Додаток А – Публікація у науковому виданні	71
Додаток Б – ER-діаграма основної частини бази даних.....	71
Додаток В – Компоненти React.js.....	73

ПЕРЕЛІК ОСНОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

AJAX	Asynchronous JavaScript And XML
DOM	Document Object Model
HTML	HyperText Markup Language
HTTP	Hyper Text Transfer Protocol
JSON	Javascript Object Notation
MVC	Model–view–controller
MySQL	Open-source relational database management system
SQL	Structured Query Language
SRS	Software Requirements Specification
ORM	Object-Relational Mapping
ACID	Atomicity, consistency, isolation, durability
ACE	Command line framework embedded into the core of AdonisJS

ВСТУП

Актуальність дослідження. Сучасна медична галузь стрімко розвивається, використовуючи передові технології для покращення якості та ефективності надання медичних послуг. У цьому контексті особливо важливим є розробка та впровадження інноваційних платформ медичних послуг та систем управління клієнтами (CRM), які відповідають сучасним вимогам та потребам медичного співтовариства. Платформа визначається своєю високотехнологічністю та інтегрованістю, надаючи широкий спектр медичних послуг та використовуючи інструменти управління клієнтами для оптимізації роботи медичних закладів.

Розробка платформи медичних послуг та CRM системи з використанням Next.js стає актуальною на тлі успішного впровадження подібних технологій у сфері медицини. Інтеграція технічно продуманої платформи дозволить не лише покращити доступ до медичних послуг, але й забезпечити ефективне управління клієнтською базою для медичних закладів.

Нижче розглянемо ключові аспекти актуальності використання платформи:

- технологічна інновація - платформа представляє собою технологічний прорив у сфері медичних платформ, інтегруючи передові технології та інтелектуальний аналіз даних та системи управління клієнтами. Такий комплекс технічних рішень має потенціал трансформувати спосіб, яким медичні послуги надаються та управляються;

- глобальний контекст здоров'я - у світлі глобальних викликів, таких як пандемія COVID-19, потреба в інноваційних рішеннях для вдосконалення медичної допомоги та роботи медичних закладів надзвичайно актуальна. Платформа може виступити як ефективний інструмент для підтримки та полегшення навантаження на систему охорони здоров'я в умовах кризи.

Мета і завдання роботи: Метою магістерської роботи є дослідження, розробка та вдосконалення програмного забезпечення, спрямованого на оптимізацію медичних послуг та управління клієнтською базою медичних

закладів.

Основними завданнями роботи є: провести аналіз функціоналу програмного забезпечення, зокрема його здатності до управління медичною інформацією та інтегрованого клієнтського обліку, вивчити архітектурні рішення, використані у програмі, зокрема звертаючи увагу на технологічні аспекти, що підтримують високу продуктивність та безпеку, розгляд і впровадження нових інструментів та можливостей управління клієнтами для медичних закладів, забезпечення ефективної комунікації та взаємодії із пацієнтами та медичним персоналом, розробка та впровадження нових функцій у систему управління клієнтами для медичних закладів.

Об'єкт дослідження: платформи для медичного туризму.

Предмет дослідження: розроблення медичної платформи та CRM систем для управління клієнтами за допомогою передових технологій, зокрема фреймворку Next.js.

Методи дослідження: для досягнення мети дослідження використовувалися такі загальнонаукові методи: аналітичний, порівняння, узагальнення.

Теоретичні та практичні результати роботи.

Теоретичне значення роботи полягає в тому, що на основі аналізу різних джерел проведено дослідження медичних платформ, проаналізовано існуючі медичні платформи та окреслено їхні особливості.

Практичне значення роботи полягає у розробленні високопродуктивної медичної платформи, яка відповідає вимогам сьогодення із використанням Next.js, а також CRM системи для управління нею.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА СПЕЦИФІКАЦІЇ ВИМОГ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Коротка характеристика об'єкту управління

В сучасному світі, де технології активно переплітаються з усіма сферами життя, особливе значення набуває впровадження інноваційних підходів у сфері медичних послуг. Розробка та впровадження медичних платформ, які інтегрують у себе функції надання медичних послуг, є важливим напрямком для поліпшення доступності та якості медичної допомоги.

Основна привабливість медичних платформ полягає у їх здатності забезпечувати швидкий доступ до необхідної медичної інформації. Також, вони сприяють ефективному обміну даними та інтеграції з різними медичними системами, що значно полегшує роботу медичних установ.

Для користувачів такі платформи відкривають можливості для кращого розуміння власного здоров'я, доступу до рекомендованих фахівців, вибору лікарів та лікарень згідно відгуків інших користувачів, а також для зручного планування лікування та контролю за його перебігом.

В цілому, розвиток медичних платформ відображає загальну тенденцію до цифровізації медичної галузі, відкриваючи нові можливості для покращення якості життя людей та ефективності медичного обслуговування

1.2 Аналіз та огляд існуючих аналогів

1.2.1 Веб-застосунок «Booking Health»

Booking Health [1] – портал прямого бронювання лікування в передових лікарнях світу. На порталі представлено понад 250 лікарень з найвищою акредитацією по всьому світу.

Архітектура – веб-додаток. Мова реалізації серверної частини – PHP. На клієнтській частині використовуються такі бібліотеки: jQuery, jQuery UI, core-js.

У додатку доступні:

- пошук лікарень;
- перегляд відділень лікарень;
- перегляд програм лікувань;
- можливість залишити запит на лікування;
- детальна інформація про сертифікації лікарень;
- розділ із інформацією про різні захворювання;
- блог;
- багатомовність;
- окрема сторінка із відгуками.

Недоліки даної системи:

- пошук в системі працює тільки по лікарнях;
- відсутня можливість детальної фільтрації лікарень;
- немає можливості залишити відгук про лікарню;
- немає можливості залишити відгук про лікаря;
- відсутня можливість детальної фільтрації лікарів;
- обмежений вибір медичних закладів.

При пошуку лікарні можна вибрати наступні фільтри: вік, захворювання, країна, місто. Також є можливість відсортувати дані за наступними опціями: ціна (від найнижчої), топ-лікарні, відгуки користувачів. Секція пошуку показано на рис. 1.1.

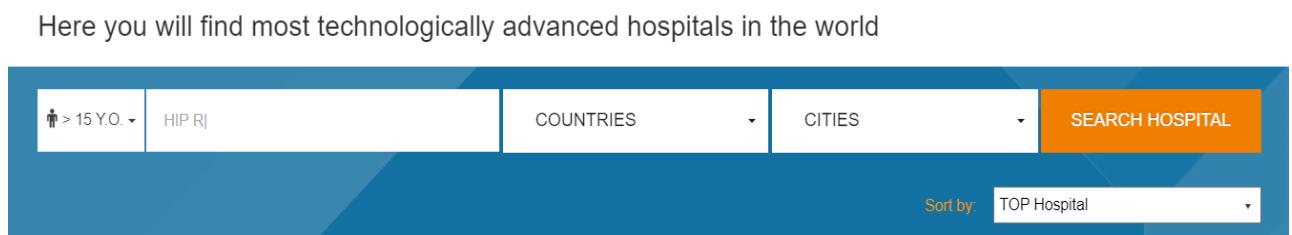


Рисунок 1.1 – Секція пошуку лікарень

Сторінка перегляду лікарні містить детальний огляд лікарні та її послуг.

Ключові особливості:

- загальна інформація;
- форма для створення запиту на лікування;
- відділення: тут перераховані різні спеціалізовані відділення, такі як загальна хірургія, урологія, ортопедія, кардіологія та багато інших;
- проживання та зручності: надається детальна інформація про кімнати для пацієнтів, харчування, меню та доступні послуги для супроводжуючих осіб.
- медичні послуги: викладено інформацію про спектр діагностичних та лікувальних методів, доступних у лікарні, включаючи лабораторні дослідження, різноманітні сканування та сучасні терапевтичні методики.
- розташування: є інтерактивна картка та розділ про місто, у якому знаходиться лікарня.

На сторінці зображено рейтинг лікарні (рис 1.2), проте немає відгуків від реальних користувачів.

Charite University Hospital Berlin

 Berlin, Germany  9.9/10 from 95 Votes

ABOUT THE HOSPITAL

DEPARTMENTS

PRICES

Рисунок 1.2 – Рейтинг лікарні

Wikimed (рис 1.3) – ресурс для людей, яким потрібна інформація про різні медичні стани та захворювання. На цій сторінці представлено перелік різних захворювань та станів здоров'я, кожне з яких пов'язане з додатковою детальною інформацією. Мета сторінки – надати користувачам зручний і доступний спосіб ознайомитися з інформацією про різні медичні проблеми, їх симптоми, варіанти лікування та найкращі лікарні для лікування. Wikimed включає в себе 104 захворювання.

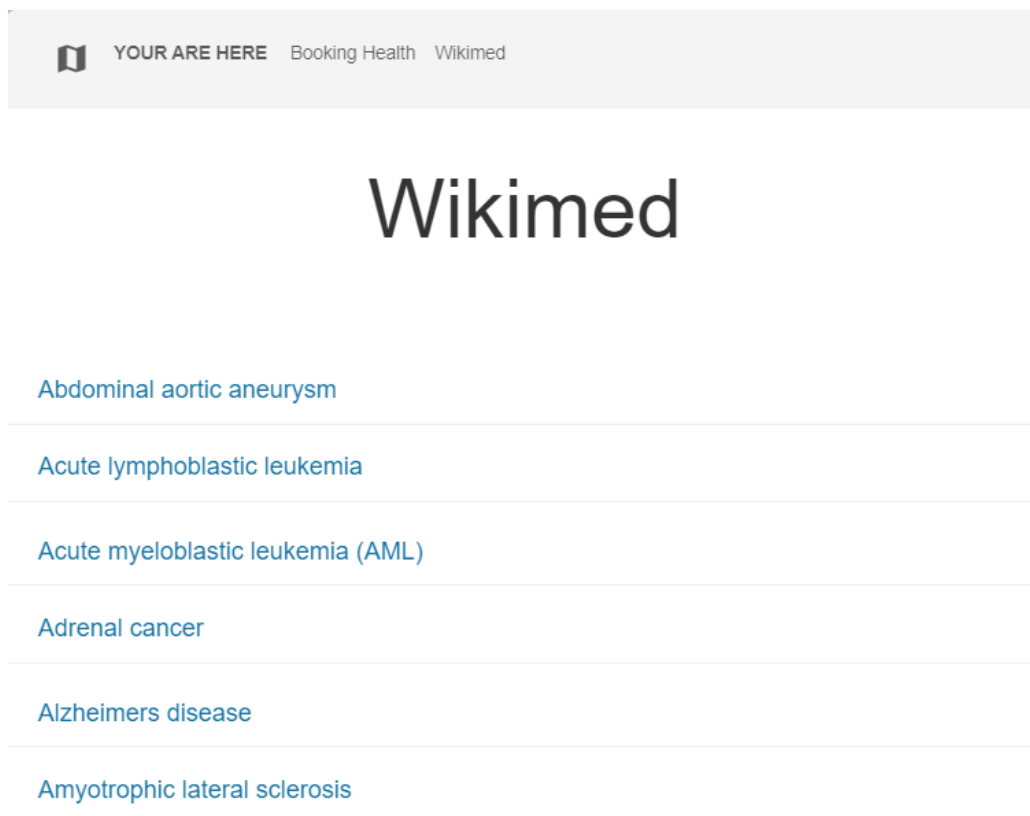


Рисунок 1.3 – Сторінка «Wikimed»

1.2.2 Веб-застосунок «PRIMO MEDICO»

PRIMO MEDICO [2] - це медична платформа, розроблена для забезпечення доступу до висококваліфікованих медичних спеціалістів та закладів у Німеччині та інших країнах. Платформа використовує архітектуру веб-застосунку. Деталі щодо мови програмування не вказані. Вебсервер – nginx. На клієнтській стороні використовуються такі бібліотеки як jQuery та Bootstrap.

У додатку доступні такі можливості:

- пошук медичних спеціалістів: пацієнти мають можливість використовувати розширений пошук для знаходження лікарів за такими критеріями як спеціалізація та географічне розташування (рис 1.4);

- подробиці про лікарні та медичні установи: надається детальний огляд лікарень, послуги та лікувальні методики;

- запити на лікування: пацієнти можуть залишати запити на лікування через вбудовану форму;
- медичні послуги та процедури: інформація про різноманітні діагностичні та терапевтичні методи, доступні в лікарнях, надається для ознайомлення пацієнтам;
- блог: розділ із статтями та інформаційними матеріалами, призначений для пацієнтів, які шукають корисні поради та новини в галузі медицини.

Exclusive List for Excellent Medical Experts in German-Speaking Countries

Start » Doctors and Centres

Select specialist field ▼

Select city ▼

Рисунок 1.4 – Секція пошуку лікарів

На рисунку 1.5 наведено сторінку із захворюваннями де користувач може знайти інформацію про симптоми, причини хворіб і методи лікування.

Frequently searched diseases and treatments

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

- | | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> ▶ Aneurysm surgery ▶ Aortic Aneurysm Surgery ▶ Atherosclerosis | <ul style="list-style-type: none"> ▶ Ankle Prosthesis ▶ Artificial fertilisation |
| <ul style="list-style-type: none"> ▶ Bladder Cancer ▶ Brachytherapy ▶ Breast Augmentation ▶ Burnout therapy | <ul style="list-style-type: none"> ▶ Bowel Cancer ▶ Brain Tumor ▶ Breast Cancer |
| <ul style="list-style-type: none"> ▶ Cardiac Arrhythmia ▶ Cervical cancer ▶ Cruciate Ligament Surgery | <ul style="list-style-type: none"> ▶ Carotid Stenosis ▶ Check-up ▶ Cystic fibrosis (CF) |
| <ul style="list-style-type: none"> ▶ Dental implant | <ul style="list-style-type: none"> ▶ Diabetic foot |

Рисунок 1.5 – Сторінка «Захворювання та лікування»

Кількість захворювань, що презентовані на сторінці, обмежена всього лише 65 найменуваннями. Це може бути недостатньо для всебічного охоплення медичних захворювань та лікувань, з якими може зіткнутися середньостатистичний користувач. Враховуючи різноманіття та складність медичних умов, такий обмежений вибір не забезпечує повного та всебічного інформаційного ресурсу.

Представлений перелік може не відображати реальну частоту пошуку захворювань. Це створює враження, що сайт не використовує достатньо даних або зворотній зв'язок для адаптації своїх ресурсів до актуальних потреб відвідувачів.

Недоліки системи:

- відсутність пошуку лікарень: користувачам недоступна можливість шукати лікарні за конкретними критеріями, що може ускладнити процес вибору;
- обмежена фільтрація лікарів: відсутність розширених фільтрів при виборі лікарів може обмежити точність вибору для пацієнтів;
- відсутність можливості залишення відгуків: користувачі не мають можливості поділитися своїми враженнями та відгуками про медичні заклади чи лікаря, що обмежує обмін досвідом;
- не велика кількість медичних закладів: існує потреба у розширенні вибору доступних медичних закладів для задоволення різноманітних потреб пацієнтів.

1.2.3 Веб-застосунок «MediGlobus»

MediGlobus [3] – це медична платформа, яка забезпечує користувачам доступ до світового рівня медичних послуг та лікарень через медичний туризм. Платформа пропонує пошуковий інструмент для вибору лікарень відповідно до індивідуальних потреб. Кожна лікарня має детальну сторінку з описом

особливості лікарні та додатковою.

У веб-додатку доступні такі можливості:

- сторінка із списком лікарень;
- сторінка детального огляду лікарні;
- можливість побачити сертифікати лікарні;
- блог;
- багатомовність;
- сторінка із вибором напрямку лікування;
- можливість залишити заявку.

Недоліки даної системи:

- відсутня сторінка лікаря;
- відсутня можливість детальної фільтрації лікарень;
- немає можливості залишити відгук про лікарню;
- обмежений вибір медичних закладів;
- немає можливості ознайомитись із програмами лікувань;
- картка лікарні є велика та не інформативною.

Платформа надає два варіанти фільтрування лікарень: можна обрати потрібне захворювання або бажаний напрямок (рис 1.6).



Рисунок 1.6 – Секція пошуку лікарень

На рис 1.7 зображено картку лікарні яка містить такі дані:

- назва лікарні;
- рейтинг;
- місце розташування;
- короткий опис;
- обмежена кількість лікарів - немає можливості перейти на сторінку конкретного лікаря та ознайомитись із інформацією про нього більш детально.

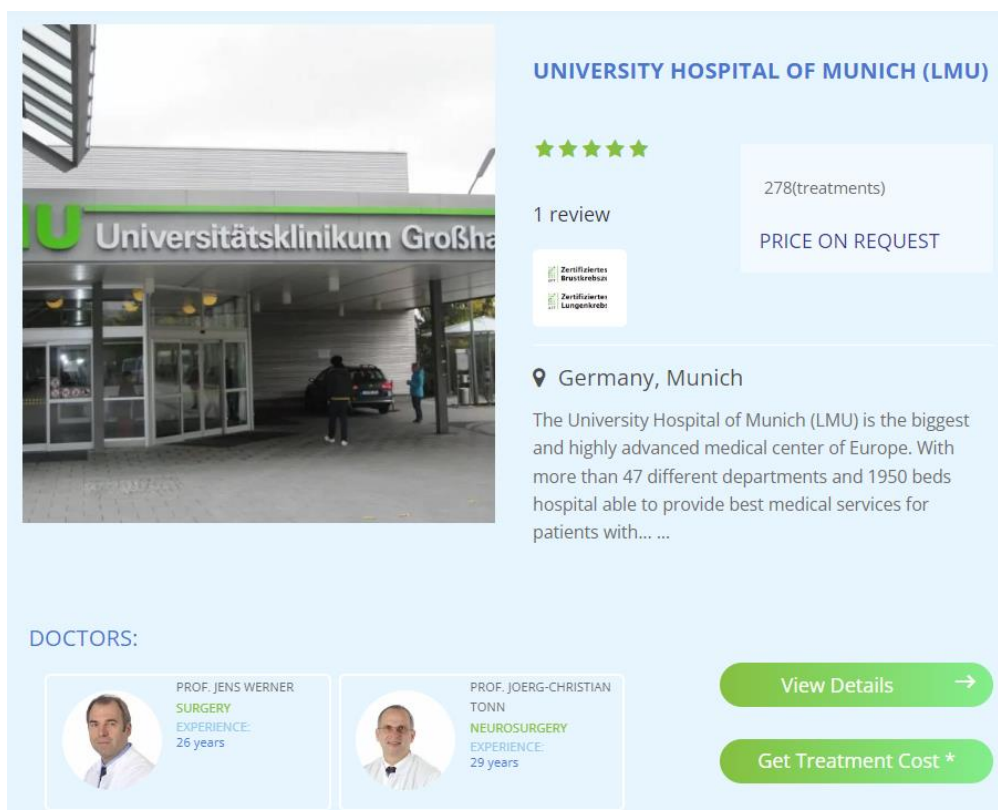


Рисунок 1.7 – «Картка лікарні»

1.2.4 Веб-застосунок «Qunomedical»

Qunomedical [4] – медична платформа, на якій користувачі можуть переглядати список із понад 1000 лікарів у понад 25-ти країнах. Медичний портал має власну систему оцінки, яка називається Qunoscore та враховує різні дані, такі як: відгуки пацієнтів, досвід лікарів, співвідношення ціни та якості та інші.

Преваги платформи:

- зручний пошук який відкривається на повний екран та є змога знайти лікування, лікаря, лікарню та статтю (рис 1.8);
- сторінка перегляду лікаря;
- сторінка перегляду лікарні (рис 1.9);
- блог.



Рисунок 1.8 – Пошук у веб-додатку

Helios Klinikum Berlin-Buch

Berlin, Germany

Get more info

ABOUT THE CLINIC

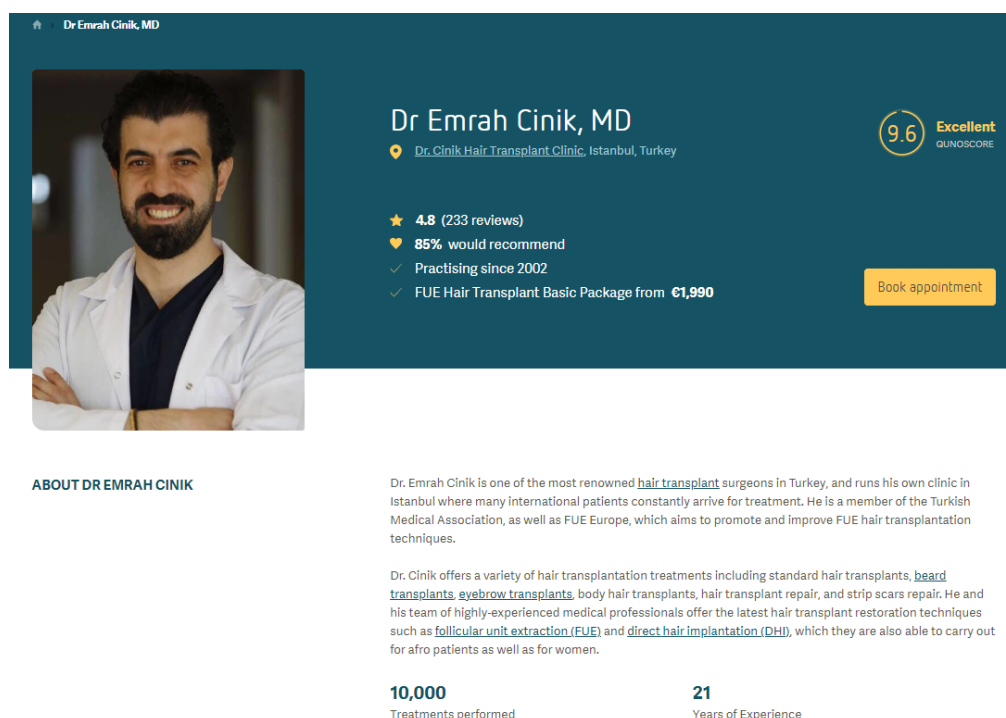
Opened in 2007, the HELIOS Clinic Berlin-Buch is one of the newest hospitals in the HELIOS network of medical centers that combines high-level care with maximum comfort. This large, state-of-the-art facility has over 1,000 beds, 20 operating rooms, 25 specialty departments and six research institutes. The centre is known for its interdisciplinary care, most notably in the Breast Centre, the Bowel Centre, the Perinatal Centre, the Stroke Unit and the Chest Pain Unit, treating the whole patient across a variety of medical specialties. The German Cancer Society has recognized HELIOS Clinic Berlin-Buch as a certified oncological treatment because of its complex treatments for prostate, bone and other cancers. Comprehensive diagnosis and treatment is also available at the Specialized Centre for Diabetes Care, which is certified by the German Diabetes Association. A helicopter landing pad allows badly injured patients from all over the region to receive care at the clinic's extensive emergency care centre. The Level 1 Perinatal Centre provides comprehensive care for pre-term infants. The large campus also provides treatment on an outpatient basis.

Location



Рисунок 1.9 – Сторінка лікарні

На рисунку 1.10 представлено сторінку лікаря, де користувач має змогу ознайомитись із інформацією про лікаря, відгуками та результатами роботи (фото).



Dr Emrah Cinik, MD
 Dr. Cinik Hair Transplant Clinic, Istanbul, Turkey

9.6 Excellent QUNOSCORE

★ **4.8** (233 reviews)
 ♥ **85%** would recommend
 ✓ Practising since 2002
 ✓ FUE Hair Transplant Basic Package from **€1,990**

[Book appointment](#)

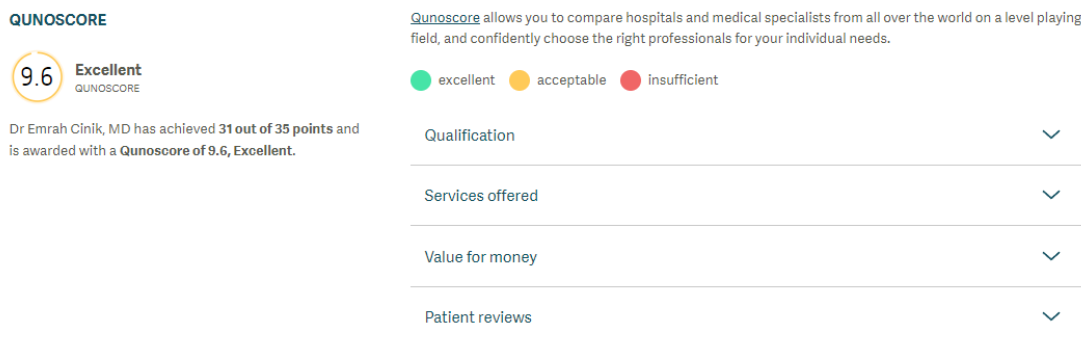
ABOUT DR EMRAH CINIK

Dr. Emrah Cinik is one of the most renowned [hair transplant](#) surgeons in Turkey, and runs his own clinic in Istanbul where many international patients constantly arrive for treatment. He is a member of the Turkish Medical Association, as well as FUE Europe, which aims to promote and improve FUE hair transplantation techniques.

Dr. Cinik offers a variety of hair transplantation treatments including standard hair transplants, [beard transplants](#), [eyebrow transplants](#), body hair transplants, hair transplant repair, and strip scars repair. He and his team of highly-experienced medical professionals offer the latest hair transplant restoration techniques such as [follicular unit extraction \(FUE\)](#) and [direct hair implantation \(DHI\)](#), which they are also able to carry out for afro patients as well as for women.

10,000 Treatments performed
21 Years of Experience

Рисунок 1.10 – Сторінка лікаря



QUNOSCORE

9.6 Excellent QUNOSCORE

Dr Emrah Cinik, MD has achieved 31 out of 35 points and is awarded with a Quoscore of 9.6, Excellent.

Quoscore allows you to compare hospitals and medical specialists from all over the world on a level playing field, and confidently choose the right professionals for your individual needs.

● excellent ● acceptable ● insufficient

Qualification ▾

Services offered ▾

Value for money ▾

Patient reviews ▾

Рисунок 1.11 – Секція «Quoscore»

Недоліки платформи:

- немає достатньо інформації про лікарні, які є на платформі;
- немає можливості залишити відгук про конкретну лікарню чи лікаря;
- немає сторінки із списком лікарів, відповідно немає можливості здійснити фільтрацію лікарів по певних параметрах;
- немає сторінки із списком лікарень, а отже, користувач не може

підібрати лікарню під свої потреби.

1.3 Специфікація вимог для медичної платформи (веб-застосунок)

Мета створення специфікації вимог – опис поведінки системи. Цей опис включає множину прецедентів (функціональні вимоги), що описують взаємодії, користувачів із системою. Крім прецедентів, SRS також включає нефункціональні вимоги. Опис глосарію проекту відображений у табл. 1.1.

Таблиця 1.1 – Глосарій проекту

Термін	Опис терміну
1	2
1. Основи поняття предметної області проекту	
Лікарня	Медичний заклад, що надає широкий спектр медичних послуг, включаючи діагностику, лікування та реабілітацію пацієнтів.
Відділення	Спеціалізований підрозділ в лікарні, який займається певною областю медичної практики або типом лікування. Кожне відділення має свої особливості та забезпечує специфічні медичні послуги та може включати спеціалізованих лікарів.
Лікар	Медичний фахівець, який діагностує та лікує. Лікарі можуть спеціалізуватися в різних медичних галузях та надавати широкий спектр послуг.
Програма лікування	Комплекс медичних послуг, які пропонуються пацієнтам для лікування різних захворювань та станів. Ці програми можуть включати різні методи лікування, а також інноваційні та альтернативні терапії. Забезпечують цілісний підхід до лікування, враховуючи індивідуальні потреби кожного .
Медична бібліотека	Розділ, де надається обширна інформація про різні аспекти здоров'я та лікування. Цей ресурс включає статті, дослідження, посібники та інші освітні матеріали, що допомагають користувачам зрозуміти свої медичні стани, доступні варіанти лікування та загальні питання здоров'я.

Продовження табл. 1.1

1	2
Каталог лікарень	Організований перелік медичних закладів, який надає користувачам інформацію про лікарні, їх спеціалізації та розташування. Каталог дозволяє користувачам зручно шукати та порівнювати лікарні, сприяючи інформованому вибору медичного закладу для відповідного лікування.
Каталог програм лікування	Каталог представляє собою перелік різноманітних медичних програм та пакетів лікування, які пропонуються користувачам. Цей каталог є важливим ресурсом для пацієнтів, які шукають специфічні види лікування, від традиційної медицини до інноваційних та альтернативних терапій.
Каталог лікарів	Організований перелік лікарів різних спеціальностей, що доступні для консультацій та лікування. Він містить профілі лікарів із інформацією про їхню кваліфікацію, досвід, спеціалізацію та відгуки пацієнтів. Цей каталог дозволяє пацієнтам легко знаходити та обирати лікарів відповідно до їхніх медичних потреб і переваг.
Каталог напрямків лікувань	Каталог включає перелік країн та міст, де доступні різні медичні послуги та лікування. Цей каталог допомагає користувачам ознайомитися з географічним розподілом медичних закладів та вибрати найбільш підходяще місце для лікування залежно від їхніх переваг, потреб та фінансових можливостей.
Блог	Розділ де публікуються освітні та інформаційні статті на теми здоров'я, медицини, новини в галузі охорони здоров'я, а також поради та рекомендації від фахівців. Цей розділ може слугувати ресурсом для користувачів, які шукають додаткові знання та хочуть бути в курсі останніх тенденцій та інновацій у медицині.
2. Користувачі системи	
Користувач	Особа, яка використовує платформу для отримання медичних послуг. Це може бути пацієнт який шукає інформацію про лікування, консультації із лікарями, або варіанти лікування у різних країнах. Користувачі можуть також використовувати ресурси платформи, такі як блоги та медичну бібліотеку, для отримання інформації про здоров'я та лікування.
Вендор	Постачальник медичних послуг, таких як лікарні, медичні центри, а також лікарів та спеціалістів, які пропонують різноманітні медичні послуги через платформу.

1	2
Адміністратор	Користувач, який має технічну можливість управління контентом та структурою платформи. Адміністратор має повноваження додавати, редагувати, видаляти, блокувати будь-які сутності в системі.

Проаналізувавши наведені вище джерела, приймаємо наступну специфіку вимог до медичної платформи. Вимоги будемо специфікувати за допомогою діаграми варіантів використання.

Актори даної системи: адміністратор, вендор та користувач. На рис. 1.12 наведено діаграму варіантів використання проектованої системи для користувача.

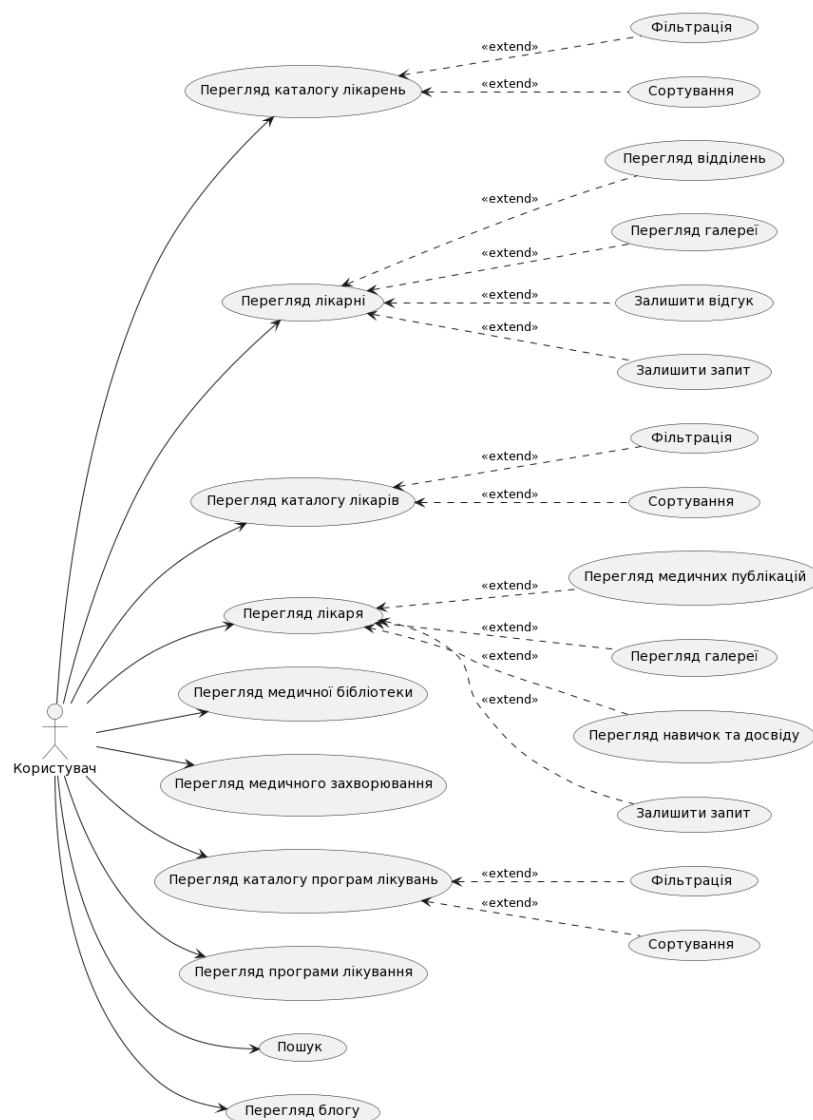


Рисунок 1.12 – Діаграма варіантів використання для користувача

На рисунку 1.13 наведено діаграму варіантів використання проекрованої системи для адміністратора. Також, подібні варіанти використання має актор «Вендор».

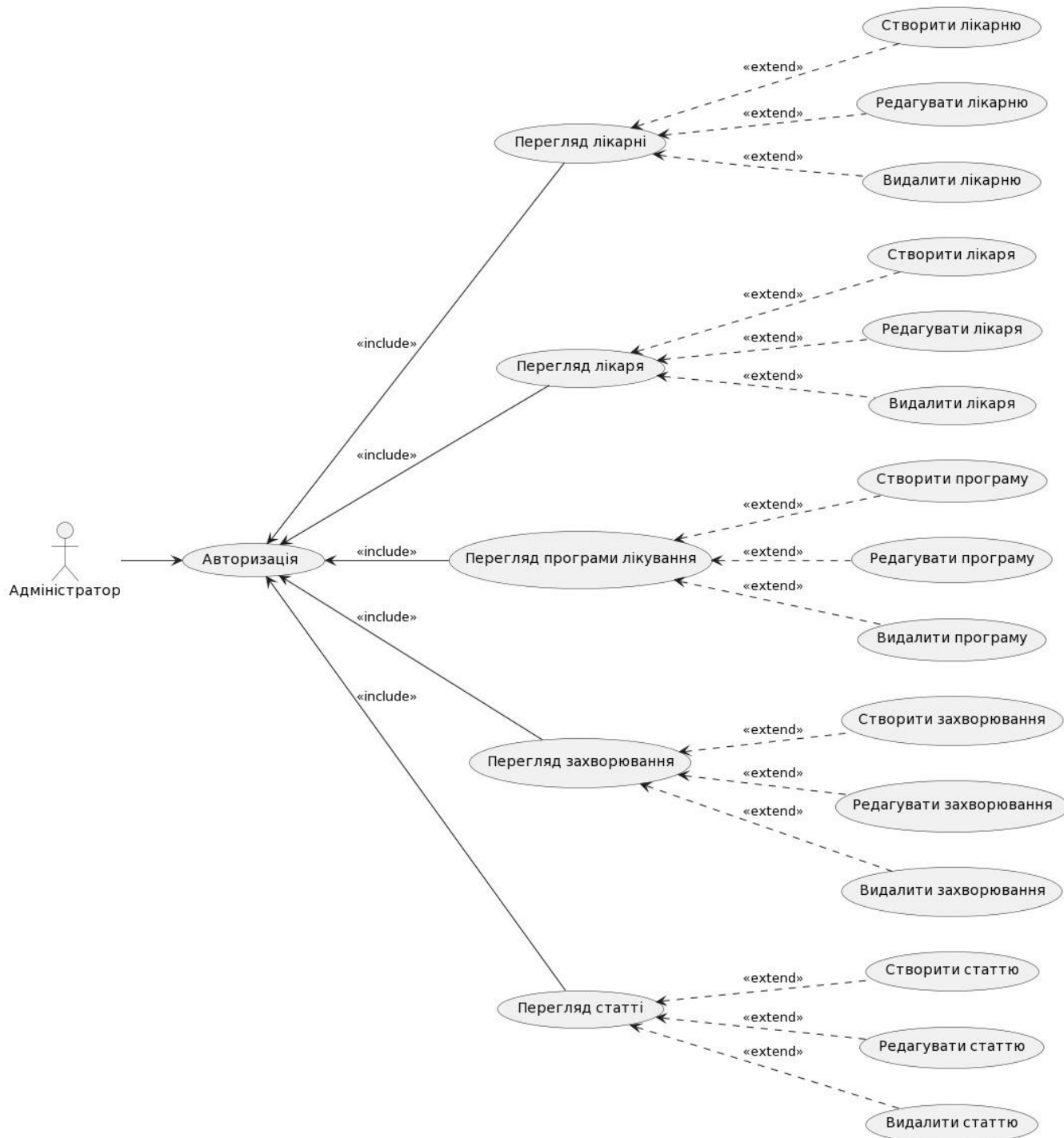


Рисунок 1.13 – Діаграма варіантів використання для адміністратора

Варіант використання авторизації в системі представлений у вигляді діаграми послідовності на рис. 1.14. При ініціалізації діалогу для авторизації (1) відображається вікно з формою для авторизації (2). Користувач вводить свої дані (3) та відправляє їх (4). Система проводить валідацію вхідних даних і авторизацію користувача (5) та повертає результат (6).

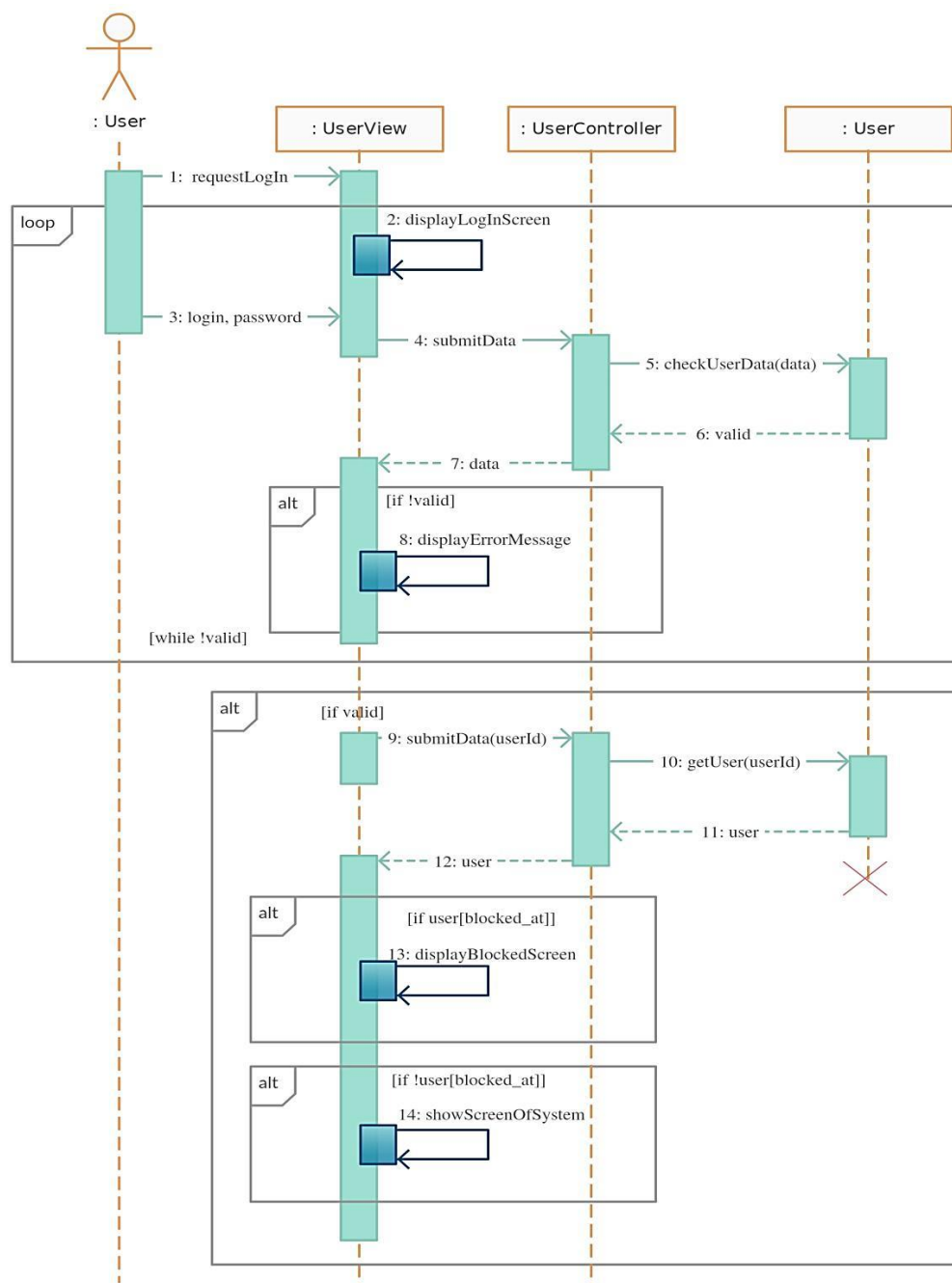


Рисунок 1.14 – Авторизація адміністратора в CRM - системі

Проаналізувавши предметну область та існуючі аналоги медичних платформ, можна умовно виокремити ті, які доцільно мати в проєктованій системі

для користувача, зокрема:

- каталог лікарень;
- детальний перегляд інформації про конкретну лікарню;
- каталог лікарів;
- детальний перегляд інформації про лікаря;
- каталог програм лікування;
- детальний перегляд програми лікування;
- медична бібліотека;
- детальний перегляд інформації про захворювання;
- зручна фільтрація для каталогів лікарень, лікарів та програм лікування;
- блог.

У результаті отримуємо специфікацію функціональних вимог для користувача представлену в таблиці 1.2.

Таблиця 1.2 – Специфікація функціональних вимог для користувача

№	Назва вимоги	Пріоритет	Складність
1	2	3	4
1	Каталог лікарень	Обов'язкова	Середня
2	Фільтрація лікарень в каталозі по різних параметрах	Обов'язкова	Висока
3	Сортування лікарень в каталозі за різними параметрами	Обов'язкова	Середня
4	Детальний перегляд лікарні	Обов'язкова	Висока
5	Перегляд відділень для конкретної лікарні	Обов'язкова	Середня
6	Галерея для лікарні	Обов'язкова	Висока
7	Залишити відгук про лікарню	Обов'язкова	Середня
8	Залишити запит на лікування в лікарні	Обов'язкова	Середня
9	Каталог лікарів	Обов'язкова	Висока
10	Фільтрація лікарів в каталозі по різних параметрах	Обов'язкова	Висока

Продовження табл. 1.2

1	2	3	4
11	Сортування лікарів в каталозі за різними параметрами	Обов'язкова	Середня
12	Детальний перегляд лікаря	Обов'язкова	Висока
13	Перегляд медичних публікацій для конкретного лікаря	Рекомендована	Висока
14	Галерея для лікаря	Обов'язкова	Висока
15	Перегляд навичок та досвіду для лікаря	Рекомендована	Висока
16	Залишити запит на лікування в конкретного лікаря	Обов'язкова	Середня
17	Медична бібліотека	Обов'язкова	Висока
18	Детальний перегляд інформації про захворювання	Обов'язкова	Висока
19	Каталог програм лікувань	Обов'язкова	Висока
20	Фільтрація програм лікувань в каталозі за різними параметрами	Обов'язкова	Висока
21	Детальний перегляд програми лікування	Обов'язкова	Висока
22	Пошук в системі	Обов'язкова	Висока
23	Блог	Рекомендована	Висока

Для адміністратора отримуємо специфікацію функціональних вимог представлену в таблиці 1.3.

Таблиця 1.3 – Специфікація функціональних вимог для адміністратора

№	Назва вимоги	Пріоритет	Складність
1	2	3	4
1	Створити лікарню	Обов'язкова	Середня
2	Створення сертифіката для лікарні	Обов'язкова	Середня
3	Створити галерею для лікарні	Обов'язкова	Висока
4	Створити лікаря	Обов'язкова	Середня

Продовження табл. 1.3

1	2	3	4
5	Створити спеціалізацію для лікаря	Обов'язкова	Середня
6	Створити галарею для лікаря	Обов'язкова	Висока
6	Створити програму лікування	Обов'язкова	Висока
7	Створити метод лікування	Обов'язкова	Середня
8	Створити тип лікування	Обов'язкова	Середня
9	Створити захворювання (для медичної бібліотеки)	Обов'язкова	Висока
10	Створити напрямок лікування	Обов'язкова	Середня
11	Створити статтю (блог)	Обов'язкова	Висока

Для вендора отримуємо специфікацію функціональних вимог представлену в таблиці 1.4.

Таблиця 1.4 – Специфікація функціональних вимог для вендора

№	Назва вимоги	Пріоритет	Складність
1	Створити лікарню	Рекомендована	Середня
2	Вибір сертифіката для лікарні	Рекомендована	Середня
3	Створити галарею для лікарні	Рекомендована	Висока
4	Створити лікаря	Рекомендована	Середня
5	Створити спеціалізацію для лікаря	Рекомендована	Середня
6	Створити галарею для лікаря	Рекомендована	Висока

Висновки до розділу

У цьому розділі проведено аналіз об'єкта управління, який охоплює медичну платформу для туризму. Досліджено ринок існуючих аналогів та оцінено їх функціональність, переваги та недоліки. Було визначено ключові потреби

користувачів. Все це дало змогу сформуванати чіткі, деталізовані та обґрунтовані вимоги для розробки веб-застосунку, який би відповідав поточним тенденціям ринку та вимогам користувачів.

Аналізуючи аналоги було ідентифіковано ключові функції, які повинні бути інтегровані в медичну платформу, а також нововведення, які дозволять нам відрізнитися від конкурентів. Специфікація вимог була складена з урахуванням усіх аспектів роботи медичної платформи, включаючи безпеку даних та масштабованість.

2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Вибір програмного середовища розроблення

2.1.1 Node.js. AdonisJS фреймворк

Для розроблення серверної частини для платформи медичних послуг було обрано мову програмування Node.js та фреймворк AdonisJS 5.4. Node.js — це міжплатформне середовище виконання JavaScript із відкритим кодом, яке виконує код JavaScript поза веб-браузером [5]. Він відомий своєю ефективною обробкою асинхронних операцій і неблокуючою моделлю вводу-виводу, що робить його особливо придатним для інтенсивних даних програм, які вимагають операцій у реальному часі. Головна перевага Node.js є надання розробникам можливості швидкого створення динамічних web-сторінок, а також:

- подійно-орієнтована архітектура та асинхронність: Node.js працює в одному потоці, використовуючи неблокуючі виклики вводу-виводу, що дозволяє йому підтримувати десятки тисяч одночасних з'єднань, що є вкрай важливим для високопродуктивних програм;
- універсальність JavaScript: використання JavaScript як для клієнтської сторони, так і на сервері спрощує процеси розробки;
- багата екосистема: NPM, екосистема пакетів Node.js, є найбільшою екосистемою бібліотек з відкритим вихідним кодом, яка надає велику кількість інструментів і модулів, які підвищують продуктивність;
- масштабованість: Node.js має високу масштабованість завдяки своїй подійно-орієнтованій архітектурі, що робить його кращим вибором для архітектури мікросервісів.

AdonisJS – сучасний і комплексний фреймворк Node.js, виділяється своїм підходом, зручним для розробників, що забезпечує простоту використання, зручність обслуговування та підвищену продуктивність [6]. Ця структура ретельно розроблена, щоб задовольнити постійні потреби розробки сучасних веб-додатків. Він інтегрує архітектуру MVC, яка організовує логіку програми та

інтерфейс користувача, таким чином спрощуючи розробку програм на стороні сервера.

Філософія фреймворку зосереджена навколо забезпечення структурованого, але гнучкого середовища, що дозволяє розробникам використовувати весь потенціал Node.js без складнощів, які часто виникають у програмуванні на стороні сервера. AdonisJS оснащений набором вбудованих функцій, які значно скорочують шаблонний код, дозволяючи розробникам зосередитися на створенні багатофункціональних і масштабованих веб-додатків. Ключові аспекти AdonisJS:

- архітектура MVC: дотримання AdonisJS принципів MVC допомагає структурувати програму на логічні компоненти, полегшуючи обслуговування та масштабованість;

- надійна система маршрутизації: фреймворк пропонує потужну систему маршрутизації з підтримкою груп маршрутів, проміжного програмного забезпечення та ресурсних маршрутів, що забезпечує чітке та виразне керування URL-адресами;

- інтегрована ORM (Lucid): AdonisJS поставляється з власною ORM Lucid, яка спрощує взаємодію з базою даних і підтримує шаблони Active Record і Data Mapper для різноманітних операцій з базою даних;

- автентифікація та безпека: AdonisJS забезпечує готові механізми автентифікації та авторизації користувачів у поєднанні з надійними функціями безпеки, такими як захист CSRF, шифрування та хешування, гарантуючи безпеку програм за замовчуванням;

- підтримка розширюваного проміжного програмного забезпечення: фреймворк дозволяє впроваджувати користувальницьке проміжне програмне забезпечення, покращуючи здатність ефективно контролювати та керувати циклом HTTP-запит-відповідь;

- вбудований інструмент CLI (Ace): AdonisJS постачається з інструментом командного рядка Ace, який надає набір корисних команд для прискорення завдань розробки, таких як міграція, заповнення та створення каркасів;

- управління середовищем і конфігурацією: AdonisJS пропонує

спрощений підхід до керування змінними середовища та конфігураціями додатків, допомагаючи підтримувати різні етапи розробки та розгортання;

– спільнота та екосистема: незважаючи на те, що AdonisJS відносно новий в екосистемі, він має зростаючу спільноту та велику кількість розширень, що робить його перспективним вибором для розробки сучасних веб-додатків.

Структура AdonisJS адаптована для створення RESTful API, важливого компонента сучасної веб-розробки. REST API, що означає Representational State Transfer Application Programming Interface, є популярним підходом для створення мережеских програм. Він забезпечує взаємодію між системами, які використовують HTTP для виконання таких операцій, як створення, читання, оновлення та видалення (CRUD), перетворюючи саму мережу в джерело даних.

AdonisJS спрощує процес розробки RESTful API, надаючи простий та інтуїтивно зрозумілий механізм маршрутизації, легку обробку запитів та інтегровану підтримку ORM. Ці функції дозволяють розробникам ефективно створювати та керувати API, які взаємодіють із різноманітними клієнтами, такими як веб-програми, мобільні програми та інші веб-служби. Використовуючи AdonisJS для розробки REST API, програмісти можуть забезпечити оптимізовану, придатну для обслуговування та масштабовану серверну архітектуру.

2.1.2 Система керування базами даних MySQL

Для даної системи було обрано систему керування базами даних MySQL.

MySQL – вільна реляційна система керування базами даних, яка вирізняється своєю універсальністю та надійністю та працює як сервер для забезпечення багатокористувацького доступу до великої кількості баз даних [7].

MySQL – одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

MySQL написаний під десятки видів операційних систем. Це і FreeBSD, OpenBSD, MacOS, SunOS, Linux. Сьогодні MySQL особливо поширена на платформах Linux та Windows.

MySQL, як і будь-яка інша СКБД, являє собою програму-сервер, яка знаходиться в пам'яті комп'ютера і обслуговує TCP порт. У випадку з MySQL, номером порту буде число 3306.

MySQL має розвинену систему доступу до баз даних. Користувачеві бази даних може бути наданий доступ до всієї бази даних, до окремих таблиць і окремих стовпців таблиць. Є обмеження на дії, які користувач може накладати самостійно. Для організації такої структури доступу використовується декілька таблиць у спеціальній базі даних. На основі цих таблиць встановлюється політика надання доступу.

Можливості сервера MySQL:

- простота у встановленні та використанні;
- підтримання необмеженої кількості користувачів, що одночасно працюють із базами даних;
- висока швидкість виконання команд;
- висока продуктивність та надійність: MySQL ефективно опрацьовує великі обсяги даних, забезпечуючи швидке виконання запитів;
- масштабованість: система дозволяє легко масштабуватися відповідно до потреб проекту, забезпечуючи ефективну роботу як з малими, так і з великими базами даних;
- підтримка ACID: гарантує цілісність даних завдяки дотриманню стандартів ACID.
- безпека та реплікація даних: має вбудовані функції для захисту даних та їх реплікації.

MySQL також відома своєю високою сумісністю з хмарними сервісами, особливо з Amazon Web Services (AWS) RDS (Relational Database Service). AWS RDS надає гнучкі та масштабовані рішення для розгортання баз даних MySQL, забезпечуючи автоматизоване управління, бекапи, відновлення та патчі безпеки.

Це дозволяє командам розробників зосередитися на основних аспектах проекту, замість того, щоб витрачати час на управління інфраструктурою бази даних. Використання MySQL з AWS RDS гарантує високу доступність, масштабованість та безпеку, які є ключовими для ефективної роботи сучасних веб-додатків і сервісів.

2.1.3 Redis

Для кешування даних обрано високопродуктивна база даних Redis [8]. Цей вибір обумовлений кількома ключовими характеристиками, які ідеально відповідають потребам сучасних веб-додатків.

Перш за все, Redis славиться своєю швидкістю. Виконуючи операції зчитування та запису безпосередньо в оперативній пам'яті, Redis забезпечує миттєвий доступ до даних, що є важливим для функціонування високошвидкісних додатків. Це робить його ідеальним рішенням для кешування, оскільки доступ до часто запитуваних даних може бути значно прискорений.

Крім того, Redis підтримує різноманітні типи даних, такі як рядки, списки, множини, впорядковані множини тощо, що надає гнучкість у використанні. Це дозволяє ефективно вирішувати різні задачі кешування, наприклад, зберігання часто використовуваної інформації.

Redis також відзначається високим рівнем підтримки відповідності та надійності. Він пропонує різні механізми для забезпечення стійкості даних, включаючи періодичні знімки пам'яті та журналювання транзакцій. Це означає, що дані можуть бути відновлені у випадку системних збоїв.

Використання Redis не лише підвищує загальну продуктивність застосунку, але й забезпечує ефективне управління ресурсами, оскільки дозволяє знизити навантаження на основну базу даних. Таким чином, Redis є важливим компонентом у архітектурі проекту, сприяючи швидкій та ефективній обробці

даних.

2.1.4 Next.js

Для розроблення клієнтської частини для платформи медичних послуг було обрано фреймворк Next.js [9]. Next.js – це фреймворк для веб-розробки який пропонує ряд технічних можливостей, які роблять його ідеальним вибором для створення сучасних веб-додатків. Основні переваги Next.js включають:

- маршрутизація: маршрутизатор на основі файлової системи, створений на основі серверних компонентів, який підтримує макети, вкладену маршрутизацію, стани завантаження, обробку помилок та інші функції;
- візуалізація на стороні клієнта та на стороні сервера за допомогою компонентів клієнта та сервера. Додатково оптимізовано за допомогою статичного та динамічного рендерингу на сервері з Next.js;
- отримання даних: спрощене отримання даних за допомогою `async/await` у серверних компонентах та розширений API отримання для запам'ятовування запитів, кешування даних і повторної перевірки;
- оптимізація зображень, шрифтів і сценаріїв для покращення основних веб-показників вашої програми та взаємодії з користувачем;
- статичний експорт та генерація статичних сторінок: підтримка статичного експорту та Incremental Static Regeneration забезпечує високу швидкість завантаження та гнучкість управління контентом;
- модульна архітектура: сприяє гнучкості у розробці, дозволяючи легко інтегрувати різні плагіни та бібліотеки.

Основні переваги серверних компонентів:

- отримання даних: серверні компоненти дозволяють перемістити вибірку даних на сервер ближче до джерела даних. Це може підвищити продуктивність за рахунок скорочення часу, необхідного для отримання даних, необхідних для

візуалізації, і кількості запитів, які клієнт повинен зробити;

- безпека: серверні компоненти дозволяють зберігати на сервері конфіденційні дані та логіку, наприклад маркери та ключі API, без ризику розкриття їх клієнту;

- розміри пакетів: серверні компоненти дозволяють зберігати великі залежності, які раніше впливали на розмір клієнтського пакета JavaScript на сервері. Це корисно для користувачів із повільнішим Інтернетом або менш потужними пристроями, оскільки клієнту не потрібно завантажувати, аналізувати та виконувати будь-який JavaScript для серверних компонентів;

- початкове завантаження сторінки та перше малювання вмісту (FCP): на сервері можемо генерувати HTML, щоб дозволити користувачам переглядати сторінку негайно, не чекаючи, поки клієнт завантажить, розбере та виконає JavaScript, необхідний для відтворення сторінки;

- оптимізація для пошукових систем та можливість спільного використання в соціальних мережах: візуалізований HTML може використовуватися роботами пошукової системи для індексування сторінок і ботами соціальних мереж для створення попереднього перегляду карток соціальних мереж.

2.1.5 React.js. Асинхронна передача даних – AJAX

Для розроблення CRM – системи медичної платформи обрано JavaScript бібліотеку ReactJS [10]. React.js дозволяє створювати великі веб-застосунки, які використовують дані, що змінюються з часом, без перезавантаження сторінки. Його мета полягає в тому, щоб бути швидким, простим, масштабованим. React.js обробляє тільки користувацький інтерфейс у додатках. Широка спільнота та екосистема забезпечує велику кількість бібліотек та інструментів які можуть спростити процес розробки CRM – системи медичної платформи. Це відповідає

видові у шаблоні модель-вид-контролер (MVC) і може бути використане у поєднанні з іншими JavaScript бібліотеками або у великих фреймворках MVC.

При розробленні проекту використовуватиметься технологія AJAX. AJAX – це не самостійна технологія, а ідея, яка базується на двох основних принципах:

- динамічна зміна змісту сторінки за допомогою DHTML;
- звернення до серверу «на льоту» через об'єкт XMLHttpRequest.

Такий підхід дозволяє при створенні Web-додатків реалізувати спілкування з сервером без перезавантаження сторінки.

У якості менеджера станів обрано мінімалістичну бібліотеку Zustand.

Zustand – це бібліотека стану для React.js, яка пропонує простоту та гнучкість [11]. Вона дозволяє легко створювати стани без потреби в обгортанні компонентів у вищих порядкових компонентах. Завдяки підходу хуків, Zustand уникає пасток з перерендерингом, які можуть виникнути у деяких інших менеджерах стану. Це робить Zustand ідеальним вибором для швидких та ефективних додатків React.js.

2.2 Розроблення архітектури програмної системи

Для реалізації серверної частини системи вибираємо об'єктно-орієнтований підхід до проектування та програмування. Логічна структура програмованої системи представлена у вигляді діаграм класів.

Основними елементами діаграми класів є класи та відношення між ними. Кожен клас представлений прямокутником, який містить секції: ім'я класу, перелік атрибутів, перелік операцій. Для кожного атрибуту заданий тип даних, для кожної операції – тип даних для значення, що повертається, та перелік параметрів.

На діаграмі класів, що на рис. 2.1, зображено основні класи, які використовуються в системі для розділу лікарів. Зокрема використано відношення

узагальнення (відношення між загальним та частковим). Це відношення відображено між класом BaseModel та іншими класами, такими як User, Doctor, Skill, Specialization та Publication. Успадкування – це ключова концепція об’єктно-орієнтовного програмування, де підклас успадковує (наслідує) властивості та методи свого батьківського класу. В цьому випадку, всі названі класи є підкласами BaseModel і тому успадковують його методи та властивості.

Також використано відношення асоціації, яке зображено між класами Doctor та класами Skill, Specialization та Publication.

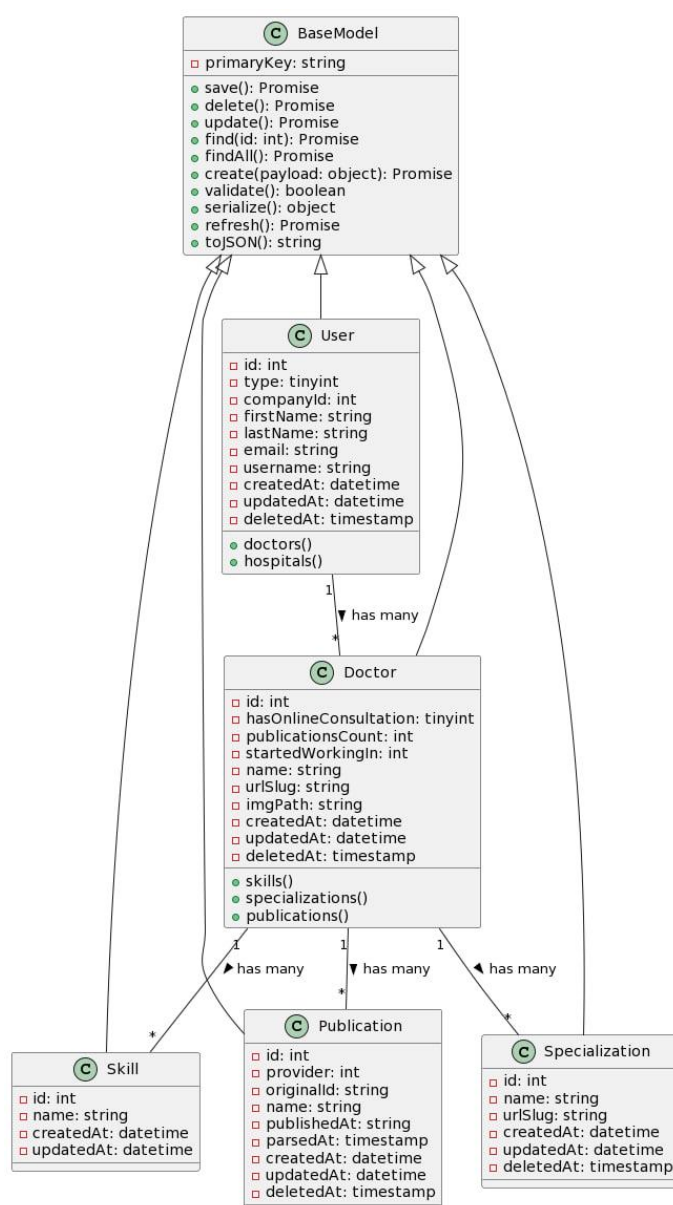


Рисунок 2.1 – Діаграма класів для розділу лікарів

Клас `BaseModel` – загальний клас для моделей в ORM, `Lucid`, і він надає ряд базових функцій для взаємодії з таблицями бази даних. Ось детальний опис загальнодоступних методів і властивостей, які надає клас:

- методи CRUD (створення, читання, оновлення, видалення): `BaseModel` містить стандартні методи CRUD, які дозволяють легко виконувати операції з даними в базі. Це включає методи для створення (`create`), пошуку (`find`, `findOrFail`), оновлення (`update`) та видалення (`delete`) записів;

- дозволяє інтегрувати валідацію даних безпосередньо в модель, забезпечуючи централізований контроль над даними, які надходять у вашу систему;

- надає гнучку систему запитів з використанням ланцюгових виразів, дозволяючи створювати складні запити з легкістю;

- підтримує визначення зв'язків між різними моделями, такими як `hasOne`, `hasMany`, `belongsTo`, `belongsToMany`, та інші;

- підтримка різних подій життєвого циклу моделі, таких як `beforeSave`, `afterSave`, дозволяє виконувати дії автоматично в певні моменти життєвого циклу запису;

- кешування запитів: використання кешування для покращення продуктивності застосунку, зокрема при повторному виконанні однакових запитів до бази даних;

- можливість міграцій в базі даних, дозволяючи легко змінювати схему бази даних у відповідності з потребами застосунку;

- м'яке видалення (`Soft Delete`): записи не видаляються назавжди, а замість цього позначаються як видалені, що дозволяє їх відновити.

На рисунку 2.2 зображено основні класи, які використовуються в системі для розділу лікарень. Зокрема використано такі відношення: узагальнення (відображено між класом `BaseModel` та іншими класами, такими як `Hospital`, `Department`, `Service Gallery` та `Certificate`), асоціація (відображено між класом `Doctor` та класами `Department`, `Service`, `Gallery`, `Certificate`).

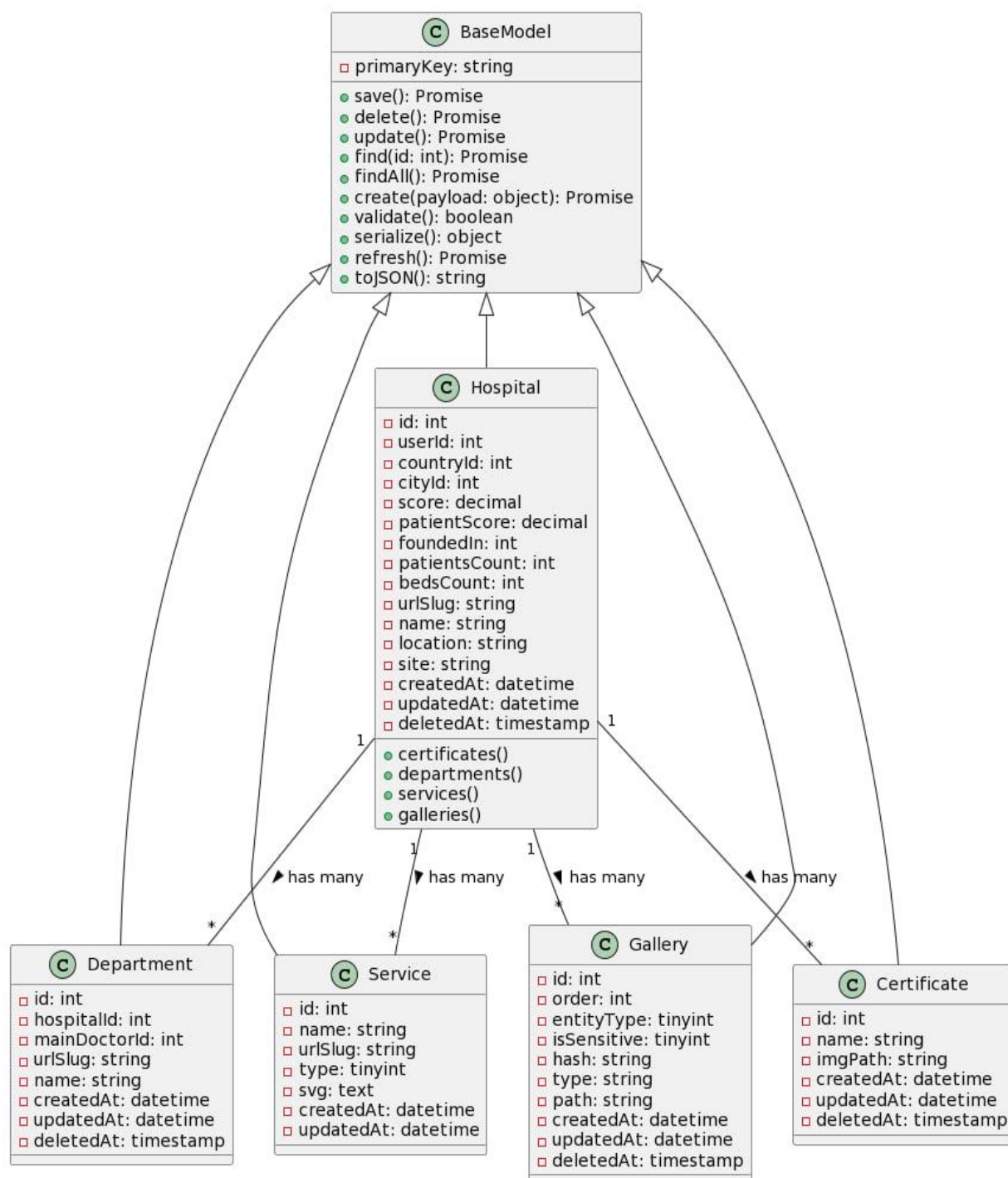


Рисунок 2.2 – Діаграма класів для розділу лікарень

На рисунку 2.3 зображено основні класи, які використовуються в системі для розділу програм лікування. Використано такі відношення: узагальнення (відображено між класом BaseModel та іншими класами, такими як: TreatmentProgram, TreatmentProgramPrice, TreatmentMethod, TreatmentTypeOfCare), асоціація (відображено між класом TreatmentProgram та

класами TreatmentProgramPrice, TreatmentMethod, TreatmentTypeOfCare).

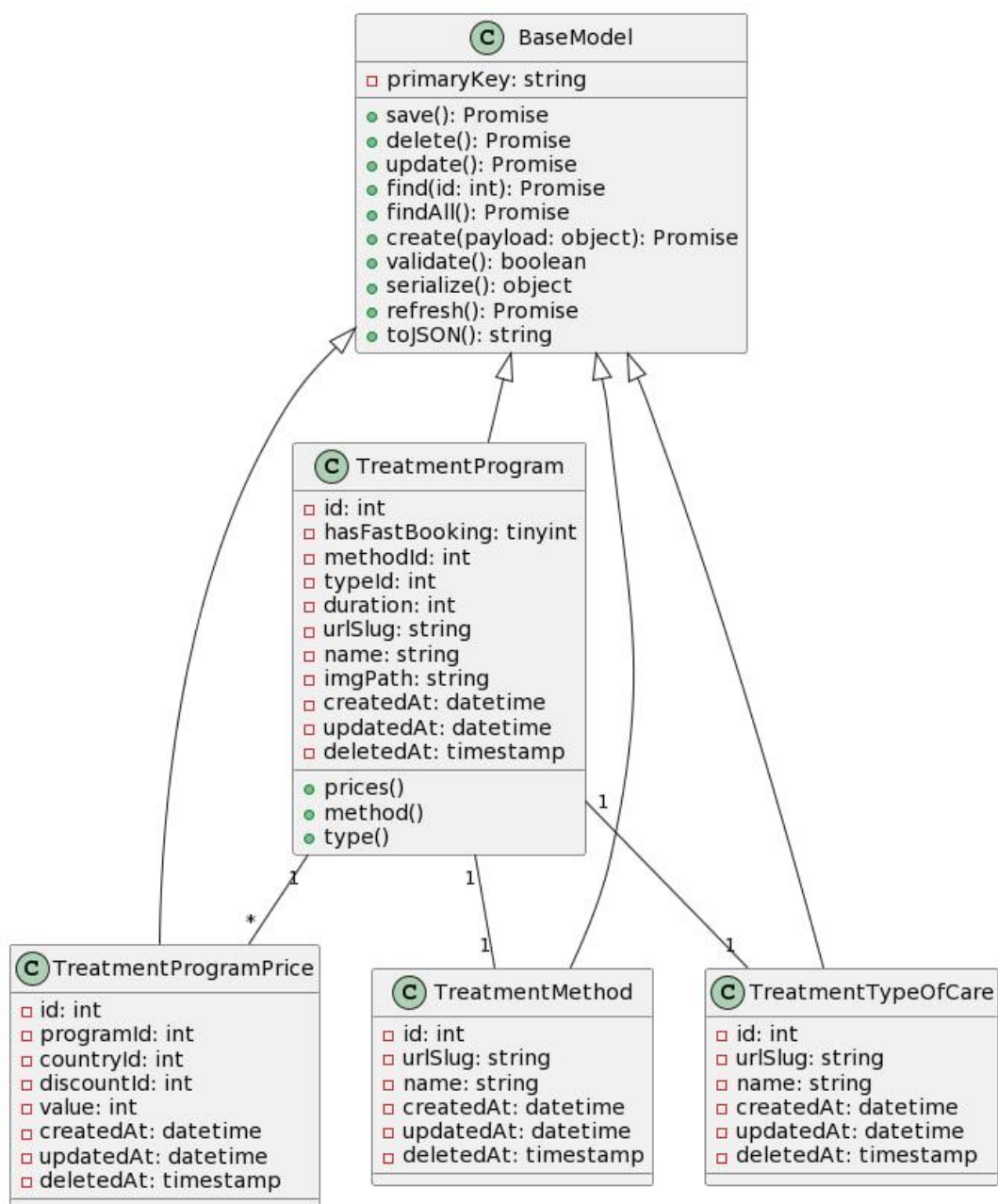


Рисунок 2.3 – Діаграма класів для розділу програм лікувань

2.3 Розроблення бази даних системи

Структура бази даних впливає на подальшу організацію роботи системи. Концептуальні схеми предметної області представлені у вигляді ER-моделі у

додатку А. ER-модель – це мета-модель даних, тобто засіб опису моделей даних. Дані представлені у вигляді компонентів, які пов'язані між собою певними зв'язками та виражають залежності та вимоги між ними.

Основними поняттями ER-моделі даних є сутність (клас однотипних об'єктів та має назву виражену іменником), атрибут (іменована характеристика, яка є деякою властивістю сутності) та зв'язок (відношення однієї сутності до іншої чи до самої себе).

Розглянемо детальніше призначення деяких із таблиць, а також проаналізуємо їх структуру. Таблиці будемо розглядати по групах, критерій групування – інформація, що зберігається в таблицях.

Таблиця users призначена для зберігання користувачів в системі, структура наведена в таблиці 2.1.

Таблиця 2.1 – Структура таблиці users

Поле	Тип	Атрибути
1	2	3
id	int unsigned	PK, auto_increment
type	tinyint	
company_id	int unsigned	
first_name	varchar(254)	
last_name	varchar(254)	
email	varchar(254)	
username	varchar(254)	
deleted_at	timestamp	
created_at	timestamp	
updated_at	timestamp	

Поле id – призначено для зберігання ідентифікаційного номеру запису і має числовий тип даних з атрибутами: PK – первинний ключ, який гарантує унікальність значень у цьому полі, auto_increment – MySQL автоматично збільшить значення цього поля при додаванні нового користувача.

Поле `type` містить цілочисельне значення яке представляє тип користувача.

Поле `company_id` містить цілочисельне значення та використовується для зв'язку із таблицею `companies`.

Поле `first_name` містить ім'я користувача.

Поле `last_name` містить прізвище користувача.

Поле `email` містить електронну пошту користувача. Кожен користувач повинен мати унікальну `email` адресу.

Поле `created_at` призначене для зберігання дати та часу створення користувача.

Поле `updated_at` призначене для зберігання дати та часу коли інформація про користувача була оновлена.

Поле `deleted_at` призначене для зберігання дати та часу видалення користувача. Необхідне для реалізації «м'якого видалення».

Таблиця `companies` призначена для зберігання компаній в системі, структура наведена в таблиці 2.2.

Таблиця 2.2 – Структура таблиці `companies`

Поле	Тип	Атрибути та властивості
<code>id</code>	<code>int unsigned</code>	PK, <code>auto_increment</code>
<code>owner_id</code>	<code>int unsigned</code>	
<code>name</code>	<code>varchar(254)</code>	
<code>org_name</code>	<code>varchar(254)</code>	
<code>created_at</code>	<code>timestamp</code>	
<code>updated_at</code>	<code>timestamp</code>	
<code>deleted_at</code>	<code>timestamp</code>	

Поле `id` – призначено для зберігання ідентифікаційного номеру.

Поле `owner_id` використовується для зберігання ідентифікатора власника компанії з таблиці користувачів.

Поле `name` містить назву компанії.

Поле `org_name` містить унікальну назву компанії.

Поле `created_at` призначене для зберігання дати та часу створення компанії.

Поле `updated_at` призначене для зберігання дати та часу коли інформація про компанію була оновлена.

Поле `deleted_at` призначене для зберігання дати та часу видалення компанії. Необхідне для реалізації «м'якого видалення».

Таблиця `hospitals` призначена для зберігання лікарень в системі, структура якої наведена в таблиці 2.3.

Таблиця 2.3 – Структура таблиці `hospitals`

Поле	Тип	Атрибути
<code>id</code>	<code>int unsigned</code>	PK, <code>auto_increment</code>
<code>user_id</code>	<code>int unsigned</code>	
<code>country_id</code>	<code>int unsigned</code>	
<code>city_id</code>	<code>int unsigned</code>	
<code>score</code>	<code>decimal(8,2)</code>	
<code>patient_score</code>	<code>decimal(8,2)</code>	
<code>founded_in</code>	<code>int unsigned</code>	
<code>patients_count</code>	<code>int unsigned</code>	
<code>beds_count</code>	<code>int unsigned</code>	
<code>url_slug</code>	<code>varchar(128)</code>	
<code>site</code>	<code>varchar(254)</code>	
<code>deleted_at</code>	<code>timestamp</code>	
<code>created_at</code>	<code>timestamp</code>	
<code>updated_at</code>	<code>timestamp</code>	

Поле `id` – призначено для зберігання ідентифікаційного номеру.

Поле `user_id` виступає зовнішнім ключем для зв'язку цієї таблиці із таблицею `users`, у якій зберігаються користувачі системи.

Поле `country_id` виступає зовнішнім ключем для зв'язку цієї таблиці із таблицею `countries`, у якій зберігаються країни.

Поле `city_id` виступає зовнішнім ключем для зв'язку цієї таблиці із таблицею `cities`, у якій зберігається міста.

Поля score містить загальний рейтинг лікарні який визначає система.

Поле patient_score містить рейтинг лікарні який визначають пацієнти.

Поле founded_in містить рік заснування лікарні.

Поле patients_count містить інформацію про кількість пацієнтів за рік.

Поле beds_count містить інформацію про кількість ліжок в лікарні.

Поле url_slug містить унікальний URL-псевдонім лікарні в системі.

Поле site містить інформацію про вебсайт лікарні.

Поле created_at призначене для зберігання дати та часу створення лікарні.

Поле updated_at призначене для зберігання дати та часу коли інформація про лікарню була оновлена.

Поле deleted_at призначене для зберігання дати та часу видалення лікарні. Необхідне для реалізації «м'якого видалення».

2.4 Інфраструктура системи

Інфраструктура для медичної платформи повинна бути з урахуванням ключових характеристик: масштабованість, надійність, безпека та висока доступності. Масштабованість гарантує, що платформа зможе витримувати зростаючі навантаження, а надійність забезпечує безперебійну роботу критично важливих служб. Безпека є пріоритетом, оскільки медична платформа буде зберігати конфіденційні дані пацієнтів. Висока доступність забезпечує безперервний доступ до медичних послуг та інформації.

Відповідно до вимог системи обрано AWS (Amazon Web Services) та трьохрівневу архітектуру, оскільки AWS надає надійну, масштабовану та ефективну хмарну платформу, що підтримує широкий спектр сервісів та технологій.

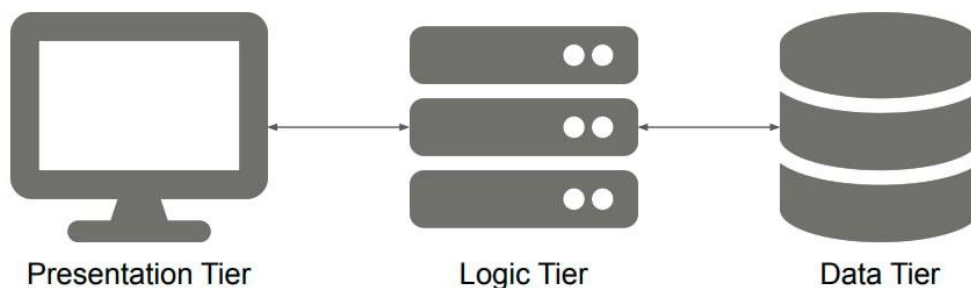


Рисунок 2.4 – Трьохрівнева архітектура

Трьохрівнева архітектура включає такі рівні як: презентаційний, логічний та рівень даних. Також вона сприяє модульності, легкості управління та масштабування. Така архітектура дозволяє ефективно розподілити ресурси, оптимізувати продуктивність та забезпечити високу доступність медичної платформи.

Презентаційний рівень відповідає за відображення інформації користувачу та запити від користувача. Також може бути веб-інтерфейсом, мобільним додатком чи іншим інтерфейсом.

На логічному рівні розташована бізнес-логіка та обробка даних. Цей рівень отримує дані від презентаційного рівня та виконує логіку додатку.

Рівень даних відповідає за зберігання та доступ до даних. Цей рівень включає базу даних або інші системи зберігання даних.

Висновки до розділу

У цьому розділі розроблено архітектуру медичної платформи, виділено основні об'єкти та побудовано ER-діаграму базу даних для медичної платформи. Проектування логічної структури здійснювалось з використанням об'єктно-орієнтованого підходу за допомогою діаграм класів (головний тип діаграм UML).

3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Розроблення структури системи

Проект є програмним комплексом, що включає декілька підпроектів, кожен з яких відіграє свою роль у загальній функціональності та взаємодіє з іншими елементами системи. Основні складові включають:

- Front-end (fe) – Next.js. Для розробки клієнтської частини веб-додатку було обрано Next.js, що дозволяє створювати оптимізовані та швидкісні веб-сторінки. Ця технологія підтримує серверний рендеринг, що забезпечує кращу SEO оптимізацію та більш швидке завантаження сторінок;

- CRM (crm) – React.js. Використання React.js для розробки CRM системи дозволяє ефективно управляти станом інтерфейсу та реалізувати більш динамічну взаємодію з користувачем. Фреймворк забезпечує швидку відмалювання компонентів і підтримує модульність та перевикористання;

- Back-end (be) – Node.js, AdonisJS. Для серверної частини було обрано Node.js з використанням фреймворку AdonisJS, що забезпечує структурованість та чіткість коду. AdonisJS пропонує потужну систему ORM, зручну роботу з базами даних, а також вбудовані засоби аутентифікації та авторизації;

- Структура інфраструктури в AWS (stg-iac, prod-iac). Використання AWS CDK для налаштування staging та production середовищ дозволяє контролювати інфраструктуру як код, що забезпечує більшу гнучкість, зниження помилок та автоматизацію процесів розгортання.

У проекті fe для керування залежностями використовується npm. Усі необхідні дані для роботи компонентів беруться по REST API у JSON форматі із проекту be.

Для збірки проекту використовується Webpack, що дозволяє оптимізувати вихідний код та ресурси.

Для керування станом використовується бібліотека Zustand.

Структура проекту fe представлена в табл. 3.1.

Таблиця 3.1 – Структура проекту fe

Директорія	Призначення
1	2
<ROOT>	Файли проекту fe
<ROOT>/storybook	Каталог, який використовується для налаштувань та конфігурації Storybook – інструменту для розробки та тестування UI компонентів у ізольованому середовищі.
<ROOT>/node_modules	Каталог який містить залежні та підзалежні пакети, що використовуються поточним додатком
<ROOT>/public	Каталог, який використовується для статичних файлів, таких як зображення
<ROOT>/src	Основна директорія для розміщення коду проекту
<ROOT>/src/api	Каталог для збереження API методів, які виконують HTTP запити до be проекту
<ROOT>/src/app	Каталог, який містить основну бізнес-логіку додатку
<ROOT>/src/components	Каталог, який містить React компоненти
<ROOT>/src/layouts	Каталог, який містить шаблони, що використовуються в даному проекті
<ROOT>/src/helpers	Каталог, який містить допоміжні функції та утиліти, які спрощують деякі загальні завдання чи операції, такі як форматування даних, обробка помилок, або логічні операції, які повторюються в різних частинах додатку
<ROOT>/src/hooks	Використовується для зберігання хуків, які можуть бути використані в різних компонентах
<ROOT>/src/store	Каталог, який використовується для управління глобальним станом додатку
<ROOT>/src/types	Каталог, який призначений для зберігання типів TypeScript. Це допомагає підтримувати типізацію у всьому додатку, підвищуючи його надійність та читабельність

Продовження табл. 3.1

1	2
<ROOT>/env	Файл, у якому зберігаються основні налаштування для проекту. Цей файл доданий до .gitignore
<ROOT>/env.example	Файл-приклад, у якому зберігаються основні налаштування для проекту
<ROOT>/gitignore	Це стандартний файл, який використовує git, щоб визначити, які файли та директорії ігнорувати
<ROOT>/package.json	Файл, який використовується менеджером пакетів npm та зберігає список пакетів, необхідних для проекту з потрібними версіями

Для збірки CRM – системи медичної платформи використовується Webpack, що дозволяє оптимізувати вихідний код та ресурси. У проекті CRM застосовується Yarn для керування залежностями, тоді як у fe проекті - npm, забезпечуючи ефективне управління пакетами.

Усі необхідні дані для роботи компонентів беруться по REST API у JSON форматі із проекту be. Структура проекту ctm представлена в табл. 3.2.

Таблиця 3.2 – Структура проекту ctm

Директорія	Призначення
1	2
<ROOT>	Файли проекту ctm
<ROOT>/config	Каталог, який містить файли налаштувань для webpack та для проекту
<ROOT>/node_modules	Каталог який містить залежні та підзалежні пакети, що використовуються поточним додатком React, як це вказано в package.json. Цей каталог доданий до .gitignore
<ROOT>/public	Каталог, який містить файли, які працюють без додаткової обробки. Файл index.html забезпечує точку входу для веб-додатку

1	2
<ROOT>/scripts	Каталог, який містить скрипти, які використовує webpack
<ROOT>/src	Тут міститься JavaScript, який буде оброблений webpack і є ядром програми React. У цьому каталозі міститься файл index.ts який запускає додаток.
<ROOT>/src/components	Каталог, який містить React компоненти
<ROOT>/src/layouts	Каталог, який містить шаблони, що використовуються в даному проекті
<ROOT>/src/index.ts	Файл, у якому відбувається ініціалізація додатку.
<ROOT>/src/Routes.ts	Файл, у якому зберігаються маршрути, що є необхідними для роботи додатку
<ROOT>/env	Файл, у якому зберігаються основні налаштування для проекту. Цей файл доданий до .gitignore
<ROOT>/env.example	Файл-приклад, у якому зберігаються основні налаштування для проекту
<ROOT>/.gitignore	Це стандартний файл, який використовує git, щоб визначити, які файли та директорії ігнорувати
<ROOT>/package.json	Файл, який використовується менеджером пакетів yarn та зберігає список пакетів, необхідних для проекту з потрібними версіями

Проект be надає API для роботи проектів admin.area та customer.area. Структура проекту представлена в табл. 3.3.

Таблиця 3.3 – Структура проекту be

Директорія	Призначення
1	2
<ROOT>	Файли проекту be
<ROOT>/app	Каталог, який містить бізнес логіку додатку
<ROOT>/app/Controllers	Каталог, який містить контролери, які обробляють вхідні HTTP запити

Продовження табл. 3.3

1	2
<ROOT>/app/Exceptions	Каталог, який містить користувацькі винятки, які можуть бути використані у додатку
<ROOT>/app/Helpers	Каталог, який містить класи-помічники для спільного використання в усьому додатку
<ROOT>/app/Middleware	Каталог, який містить посередники, які виконуються перед контролерами
<ROOT>/app/Models	Каталог, який містить моделі, які представляють таблиці бази даних
<ROOT>/app/Policies	Каталог, який містить файли, для визначення політик доступу
<ROOT>/app/Validators	Каталог, який містить файли для валідації вхідних даних
<ROOT>/commands	Каталог, який містить користувацькі CLI команди
<ROOT>/config	Каталог, який містить конфігураційні файли додатку
<ROOT>/contracts	Каталог, який містить інтерфейси та типи додатку
<ROOT>/database	Каталог, який містить міграції та фабрики
<ROOT>/docs	Каталог, який використовується для зберігання документації проекту
<ROOT>/resources	Каталог, який містить необхідні ресурси для додатку, такі як шаблони представлень (email)
<ROOT>/node_modules	Каталог, який містить всі пакети та бібліотеки, встановлені через npm чи yarn
<ROOT>/public	Каталог, який містить статичні файли, які доступні для відкритого перегляду
<ROOT>/env	Файл, у якому зберігаються основні налаштування для проекту. Цей файл доданий до .gitignore
<ROOT>/env.example	Файл-приклад, у якому зберігаються основні налаштування для проекту
<ROOT>/package.json	Файл, який використовується менеджером пакетів yarn та зберігає список пакетів, необхідних для проекту з потрібними версіями

Схема веб-вузла CRM-системи із зв'язками показана на рис. 3.1. Для того, щоб переглядати сторінки необхідно здійснити вхід в систему. На всіх сторінках системи є сайдбар – головне меню системи. З нього можна перейти на основні розділи системи такі як: список лікарів, список спеціалізацій, список лікарень, список сертифікатів, список програм лікування, список типів лікування, блог, медична бібліотека.

Адміністратор може виконувати CRUD (створення (Create), читання (Read), оновлення (Update), та видалення (Delete)) операції для управління вмістом основних сторінок. Це дозволяє адміністратору ефективно керувати інформацією на сайті.

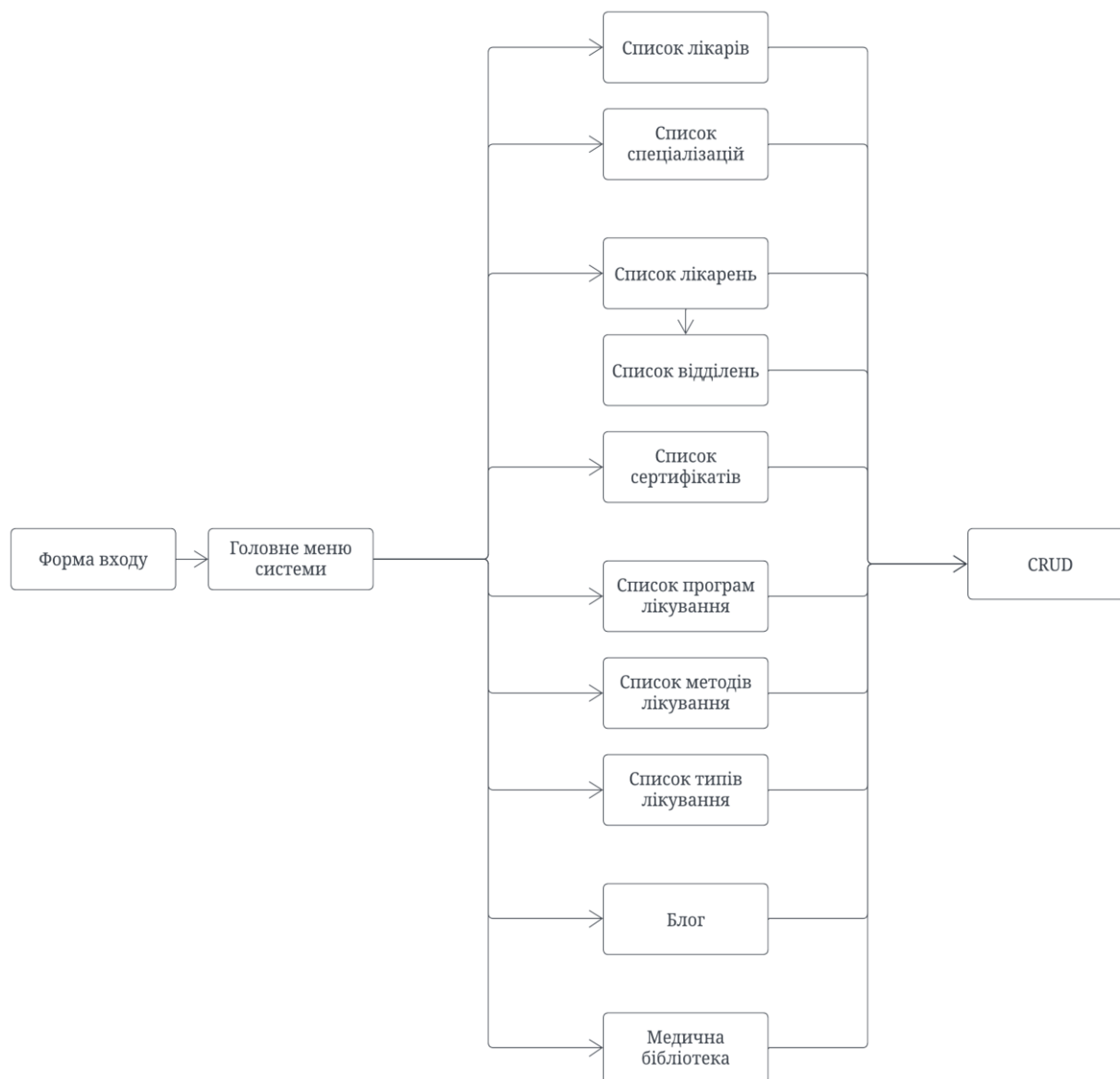


Рисунок 3.1 – Структура веб-вузла CRM-системи

3.2 Реалізація архітектури медичної платформи

3.2.1 Клієнтська частина веб-додатку

Система володіє можливістю локалізації, проте на даний момент доступна тільки англійська мова. Відповідно до розробленої структури, реалізуємо відповідні сторінки та форми.

Шапка веб-додатку включає меню навігації, пошук та контактну інформацію. Меню навігації містить посилання на основні розділи сайту, такі як захворювання, програми лікування, лікарні, лікарі, блог та інші важливі розділи. Також присутні кнопки для зв'язку. Дизайн шапки сучасний і виконаний у легкому та зрозумілому стилі.

Користувач має можливість скористатись пошуком по лікарнях, лікарях, захворюваннях, програмах лікувань із будь якої сторінки додатку. Головна сторінка із статистичною інформацією про додаток та основним пошук на платформі наведено на рис 3.2.

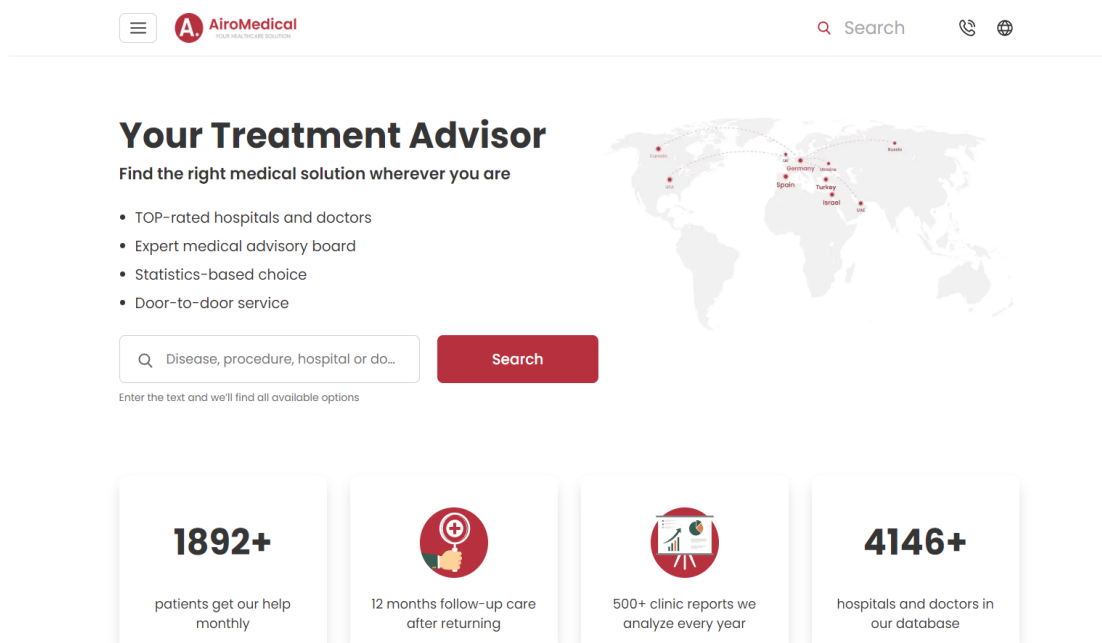


Рисунок 3.2 – Головна сторінка

Також на головній сторінці веб-додатку користувач може побачити секцію із найбільш популярними напрямками для медичного туризму та детально з ним

ознайомитись. Ця секція наведена на рис 3.3.

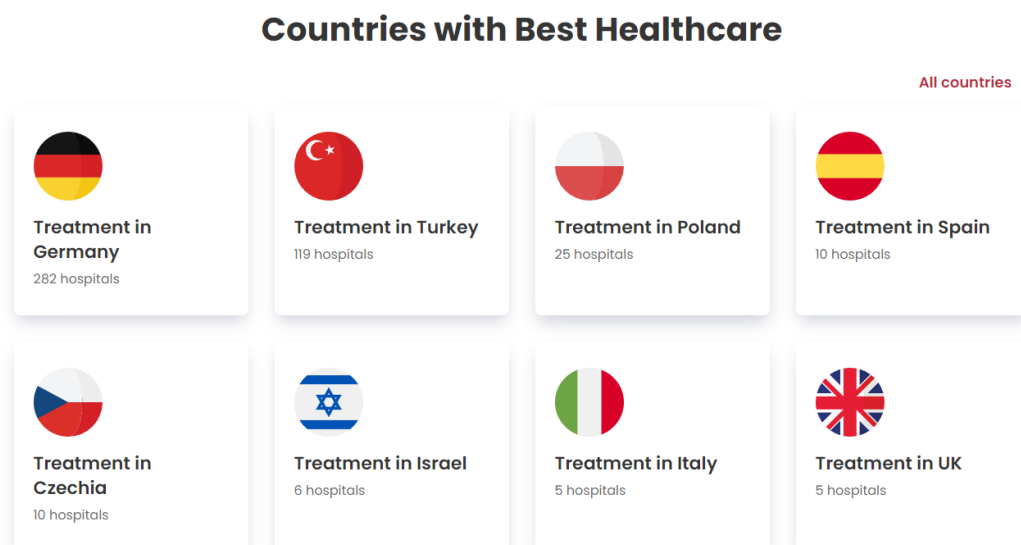


Рисунок 3.3 – Секція «Країни з найкращою охороною здоров'я»

У свою чергу користувач може ознайомитись із ключовими перевагами та особливостями додатку. Сюди входить інформація про якість обслуговування, доступність різноманітних медичних процедур, співпрацю з передовими лікарями та лікарнями, а також про зручність бронювання та організацію подорожей.

Перейшовши на сторінку захворювання (рис 3.4), користувач може знайти інформацію про лікування цього захворювання в різних країнах, включаючи деталі про передових лікарів та лікарні, які спеціалізуються на цьому захворюванню.

Neurology **Cancer Treatment** Ear, Nose & Throat (ENT) Neurosurgery Orthopedics & Joints Health Check-up [C](#) [>](#) [Location](#)

Oncology (Cancer Treatment) Worldwide: Best Hospitals, Doctors, Options, & Cost

135 clinics, 1211 doctors, 394 packages, 930 prices, and 40 blogs are found for cancer treatment worldwide. Check the best of them in AiroMedical ratings. Require our support? Our medical team is ready to help 24/7.

Nasopharyngeal cancer Oral cavity cancer Parotid gland cancer Salivary gland cancer [>](#)

A **cancer diagnosis** is a considerable challenge and can change anything. Leaving home and looking for cancer treatments abroad can be an additional adversity.

We discuss looking for the 'right' medical center and an **oncologist**, comparing prices, shorting waiting times, and choosing the best available cancer treatment option for you. At the same time, we are concerned about the additional difficulties that international patients face and help to avoid them.

Looking for the right **hospital abroad** on our platform is easy and convenient. You can apply for treatment at multiple clinics at once. Careful planning and coordination with our support system make looking for a hospital for cancer treatments abroad more accessible and comfortable.

[Approved by AiroMedical Board](#) [Update: Dec 19, 2023](#) [Fact Checked](#)

Content

- Overview
- Best hospitals
- TOP doctors
- Diagnosis
- Treatments & Packages
- Advanced treatment
- Costs
- Benefits
- About AiroMedical

Рисунок 3.4 – Сторінка захворювання та навігація

Також тут представлені описи методів лікування, можливості для діагностики, а також поради щодо вибору відповідного лікування та місця. Ця сторінка надає корисну інформацію для тих, хто шукає опції лікування якогось захворювання за кордоном. Зручна навігація дозволяє користувачу легко переходити між різними розділами та знаходити потрібну інформацію.

Варто зазначити що до кожного захворювання додається секція із релевантними статтями які написані медичними письменниками які перевіряються медичними фахівцями. Ця секція наведена на рис 3.5.

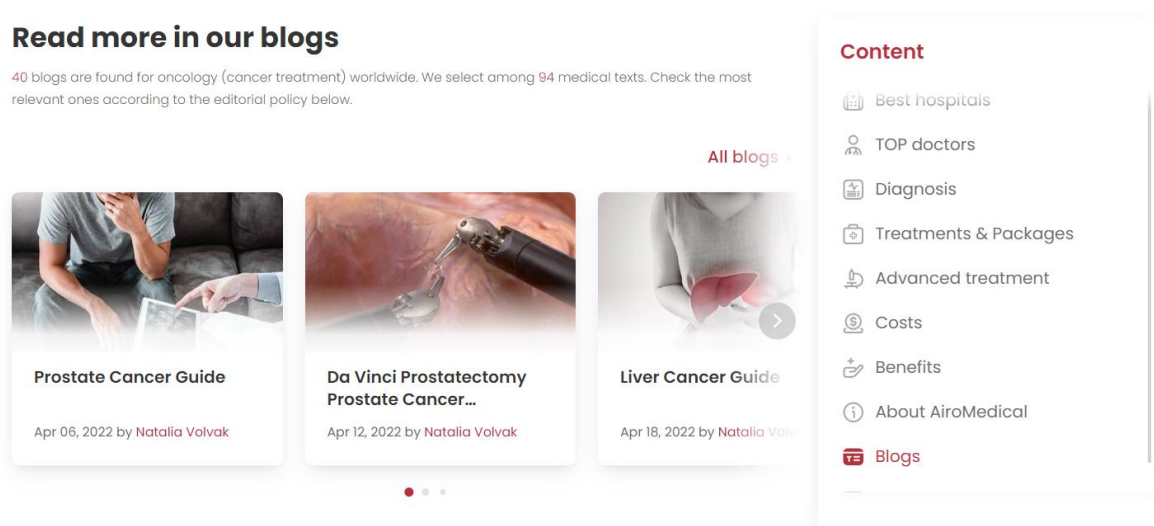


Рисунок 3.5 – Секція із релевантними статтями

Додатково, для більш розширеного пошуку, користувач може перейти на наступні сторінки: список лікарень, список лікарів, список програм лікування.

Розглянемо більш детально сторінку із списком лікарень. Ця сторінка має сучасний та зрозумілий дизайн та зручну навігацію (рис 3.7), можливість сортування та фільтрації лікарень (рис 3.6) за різними критеріями такими як: рейтинг, цінова категорія, вікова група (діти, дорослі), тип лікування (стаціонар, амбулаторно), розмір лікарні (маленький, середній, великий), фокус лікарні (загальний, спеціалізований), право власності (приватна, державна), співвідношення ціни та якості, наявність університетської лікарні, академічного

центру, сертифікатів, можливості онлайн консультації, участі у клінічних випробуваннях, наявності відгуків в додатку чи на інших ресурсах таких як Google. Для більшої зручності користувач відфільтрувати список лікарень по додаткових послугах, які можуть бути надані лікарнею. Ці фільтри включають опції, такі як цілодобова медична допомога, трансфер з аеропорту, наявність кафетерію, можливість госпіталізації супроводжуючої особи, спеціальні пропозиції на авіа-перельоти, фінансування лікування, домашній догляд, наявність підходів для людей з обмеженими можливостями, банкомати, координація клініки, цифрові медичні записи, переклад медичних документів, медичне страхування та харчування в лікарні та багато інших.



Рисунок 3.6 – Секція для фільтрації списку лікарень

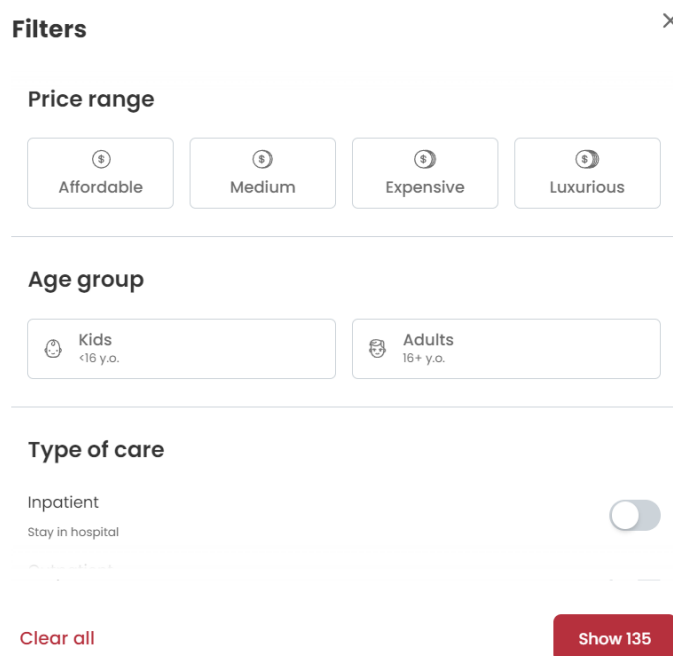


Рисунок 3.7 – Модальне вікно «Фільтрація лікарень»

На рисунку 3.8 показано секцію із списком лікарень, що спеціалізуються на лікуванні конкретного захворювання. На картці лікарні користувач може

побачити рейтинг лікарні в додатку, загальну кількість відгуків користувачів, індикатор цінового діапазону, кількість сертифікатів для лікарні, кількість відділень які є в лікарні, кількість файлів в галереї та інформацію про місцезнаходження. Користувачі можуть сортувати лікарні за рейтингом, ціною, кількістю відгуків та сертифікатами. Зверху сторінки є інформаційна панель, яка зазначає загальну кількість лікарень, доступних для пошуку.

Find the best cancer treatment hospitals

135 hospitals for oncology (cancer treatment) worldwide. We rank among 487 clinics. Check the best ones according to the clinic rating below. Require our support? Our medical team is ready to help 24/7.

The screenshot displays a search interface for cancer treatment hospitals. At the top, there are filter buttons: 'rating' (set to 10.00), 'price range', 'reviews', 'certificates', and 'foundation'. Below the filters, a grid of hospital cards is shown. Each card includes a star rating, a photo of the hospital building, the hospital name, a UserScore percentage, the number of reviews, a price range indicator (e.g., \$\$\$\$), and the location. The hospitals listed are:

- University Hospital Charite Berlin:** 10.00 rating, 99% UserScore, 809 reviews, \$\$\$\$ price range, Berlin, Germany.
- University Hospital Rechts der Isar...:** 9.90 rating, \$\$\$\$ price range, 28 certificates, 28 departments, Munich, Germany.
- University Hospital Frankfurt am M...:** 9.90 rating, departments, 24 media files, since 19..., Frankfurt am Main, Germany.
- Beta Clinic Bonn:** 9.90 rating, 98% UserScore, 119 reviews, \$\$\$\$ price range, Bonn, Germany.
- Helios Hospital Berlin-Buch:** 9.90 rating, 98% UserScore, 902 reviews, \$\$\$ price range, Berlin, Germany.
- University Pediatric Hospital Sant...:** 9.90 rating, 97% UserScore, 1,552 reviews, \$\$\$\$ price range, Barcelona, Spain.
- Anadolu Medical Centre:** 9.90 rating, 98% UserScore, 1,879 reviews, \$\$\$\$ price range, Gebze, Turkey.
- Teknon Medical Centre Barcelona:** 9.90 rating, 99% UserScore, 2,016 reviews, \$\$\$\$ price range, Barcelona, Spain.

Рисунок 3.8 – Список лікарень

Обравши лікарню яка цікавить користувача, він може перейти на сторінку детального огляду лікарні (рис 3.9). На цій сторінці доступні наступні секції: загальний опис, перелік отриманих сертифікатів, особливості та факти про клініку, основні медичні напрямки, відділення та лікарів, галерею зображень та відео, відгуки пацієнтів, перелік додаткових послуг, розташування на карті та FAQ секцію з частими запитаннями. Це дозволяє користувачам отримати детальне уявлення про лікарню та її послуги.

На рисунку 3.10 зображена галерея лікарні. Користувач має змогу обрати змешаний режим показу (фотографії та відео), тільки фотографії та тільки

відео. Також доступний режим повноекранного перегляду фотографій чи відео.

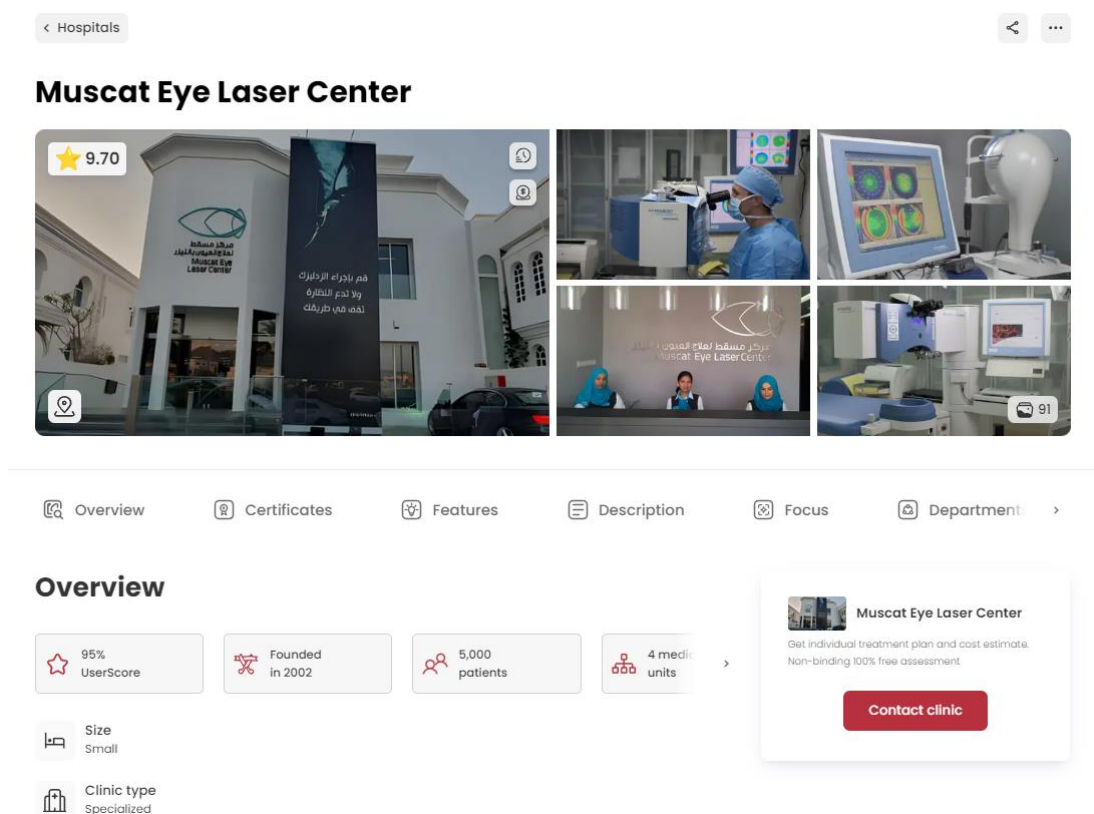
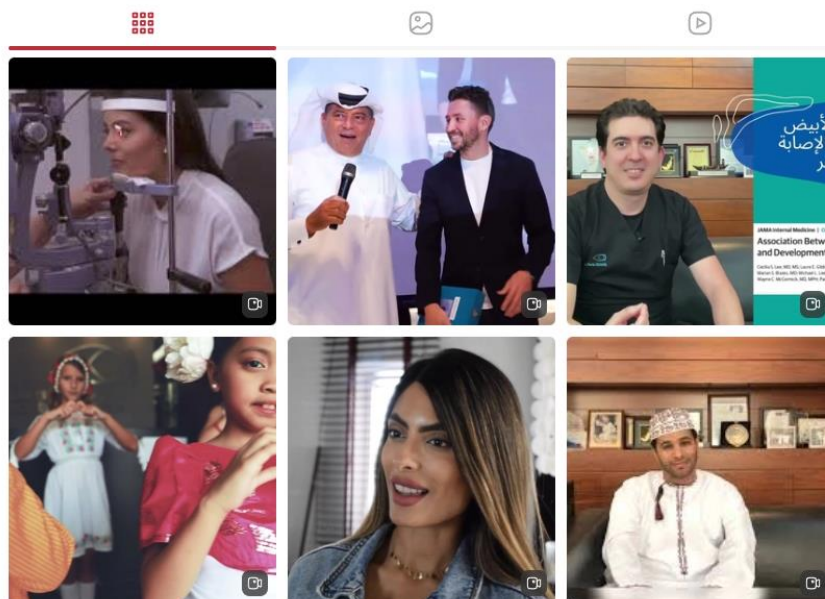


Рисунок 3.9 – Сторінка лікарні

Gallery

51 photos & 40 videos from Muscat Eye Laser Center. Check the most relevant ones according to the media policy below.



Show more

Рисунок 3.10 – Галерея лікарні

На рисунку 3.11 зображена секція яка показує ключові спеціалізації та фокус лікарні. Використовуючи інструменти штучного інтелекту для аналізу, система генерує теги, які відображають основні напрямки діяльності клініки. Теги дозволяють користувачам швидко зрозуміти, які медичні послуги та області експертизи пропонує медичний заклад, та визначити, чи відповідає лікарня їхнім потребам.

Primary focus

24 expert tags for Muscat Eye Laser Center. Our tools analyze the strengths of the clinic using artificial intelligence. Check the data selected according to the AI policy below.

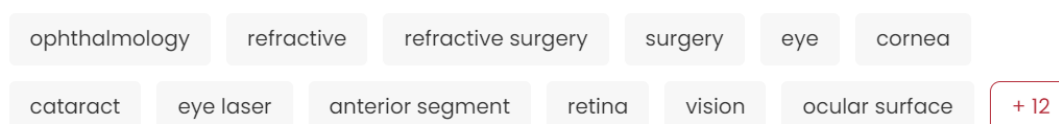


Рисунок 3.11 – Секція «Основні напрямки»

На рисунку 3.12 зображена секція яка показує список відділень які є у лікарні та лікарів надаючи користувачам інформацію про спеціалізації лікарів у кожному відділенні. Також у користувача є можливість переглянути всіх лікарів які є у відділенні натиснувши на відповідну кнопку.

Departments & Doctors

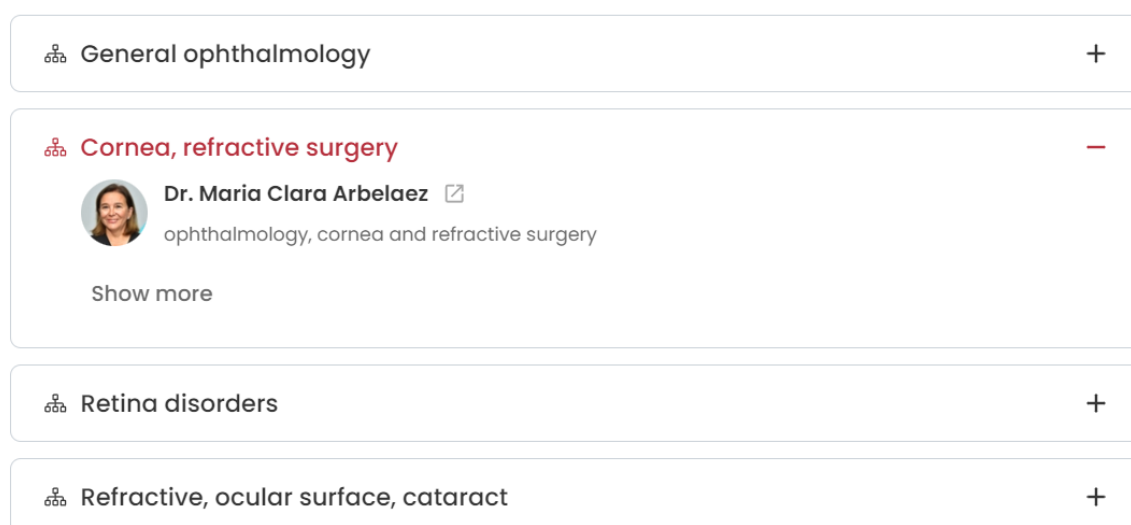


Рисунок 3.12 – Секція «Відділення та лікарі»

На рисунку 3.13 зображена секція яка надає інформацію про відзначені досягнення та особливості медичного закладу. Вона включає статистику обслуговування пацієнтів, отримані нагороди, а також особливості та напрями клінічної практики.

Features & Facts

<p>🔦 Since 2002, an internationally recognized eye center</p> <p>The only ACHSI-accredited eye clinic in Oman has served over 100,000 patients and won awards like "Oman's Most Trusted Brand," Best Eye Doctor in Oman, and the International Society of Refractive Surgery award.</p>	—
<p>🔦 Oman's pioneer in advanced vision correction technologies</p>	+
<p>🔦 Specialized diagnostic capabilities</p>	+
<p>🔦 World-class treatments in cataracts</p>	+
<p>🔦 Dedicated patient-centric approach</p>	+

Рисунок 3.13 – Секція «Особливості та факти»

Секція «Сертифікати» яка наведена на рисунку 3.14 демонструє нагороди, сертифікати та визнання. Наявність таких сертифікатів, як ISO та відомчих відзнак, інформує користувачів про високі стандарти клініки, засновані на міжнародних критеріях та найкращих практиках. Це забезпечує довіру та впевненість у виборі цього закладу для медичних послуг

Certificates

8 awards, certificates, and recognitions for Muscat Eye Laser Center. Check the data collected according to the [editorial policy](#) below.



Рисунок 3.14 – Секція «Сертифікати»

Розглянемо більш детально сторінку «каталог лікарів», яка зображена на рисунку 3.15. Кожен лікар представлений з фотографією, оцінкою, сферою

експертизи та досвідом роботи. Користувач може відфільтрувати лікарів за спеціалізацією, локацією, захворюванням, рейтингом, досвідом, кількістю медичних публікацій, та багатьма іншими фільтрами. На рисунку 3.16 представлено модальне вікно для встановлення необхідних фільтрів.

The screenshot shows a user interface for finding doctors. At the top, there are navigation icons for various medical specialties: Internal medicine, Pediatrics, General surgery, Orthopedics, Cardiology, and Oncology. To the right, there are buttons for 'Condition', 'Location', and a menu icon. Below this is the heading 'Find the best doctors' and a sub-heading '3,659 doctors worldwide. Check the best ones according to the doctor's rating below. Require our support? Our medical team is ready to help 24/7.'

Below the heading, there are three filter buttons: 'rating' (selected), 'experience', and 'reviews'. The main content is a grid of eight doctor profiles, each featuring a profile picture, a circular 'Excellent' rating badge with a score of 10.00, and a list of medical specialties. Below each profile, there are statistics: '99% UserScore', 'years of experience', 'cost level' (represented by dollar signs), and 'reviews'.

Doctor Name	Specialties	UserScore	Experience (years)	Cost Level	Reviews
Prof. Dr. med. Frank Tacke	internal medicine, gastroenterology, hep...	99%	15	\$\$\$\$	1,081
Prof. Dr. med. Britta Siegmund	internal medicine, gastroenterology, Infe...	99%	25	\$\$\$\$	486
Prof. Dr. med. Christoph Buhner	pediatric and neonatology	99%	37	\$\$\$\$	707
Prof. Dr. med. Wolfgang Henrich	gynecology and obstetrics, fetal medicine	99%	34	\$\$\$\$	505
Prof. Dr. med. Kai-Uwe Eckardt	internal medicine, nephrology	99%	39	\$\$\$\$	722
Prof. Dr. med. Ulrich-Wilhelm Tho...	pediatric neurosurgery, neurotraumatolo...	99%	28	\$\$\$\$	265
Prof. Dr. med. Lars Bullinger	haematology, oncology, leukaemia rese...	99%	26	\$\$\$\$	665
Prof. Dr. med. dent. Paul-Georg Jo...	pediatric dentistry, orthodontics, dentofa...	99%	38	\$\$\$\$	2

Рисунок 3.15 – Сторінка «Каталог лікарів»

The screenshot shows a modal window titled 'Filters' with a close button (X) in the top right corner. It contains several sections for filtering doctors:

- Age Group:** Two buttons: 'Kids <16 y.o.' and 'Adults 16+ y.o.'.
- Research activity:** A sub-heading 'Research activity' followed by the text 'The number of scientific publications'. Below are three buttons: 'Basic', 'Moderate', and 'Extensive'.
- Features:** A sub-heading 'Features' followed by four checkboxes: 'CV (resume)', 'Fast appointment', 'Online consultation', and 'Media gallery'.
- Actions:** At the bottom left, a 'Clear all' link. At the bottom right, a red button labeled 'Show 3 659'.

Рисунок 3.16 – Модальне вікно «Фільтрація лікарів»

Обравши лікаря який цікавить користувача, він може перейти на сторінку детального огляду лікаря (рис 3.17). На цій сторінці доступні наступні секції: статистична інформація про лікаря, коротка інформація про лікаря, теги із навичками та досвідом роботи (рис 3.18), галерея зображень та відео, резюме (CV), відгуки пацієнтів, медичні публікації (рис 3.19), розташування на карті та FAQ секцію з частими запитаннями. Це дозволяє користувачам отримати детальне уявлення про лікаря.

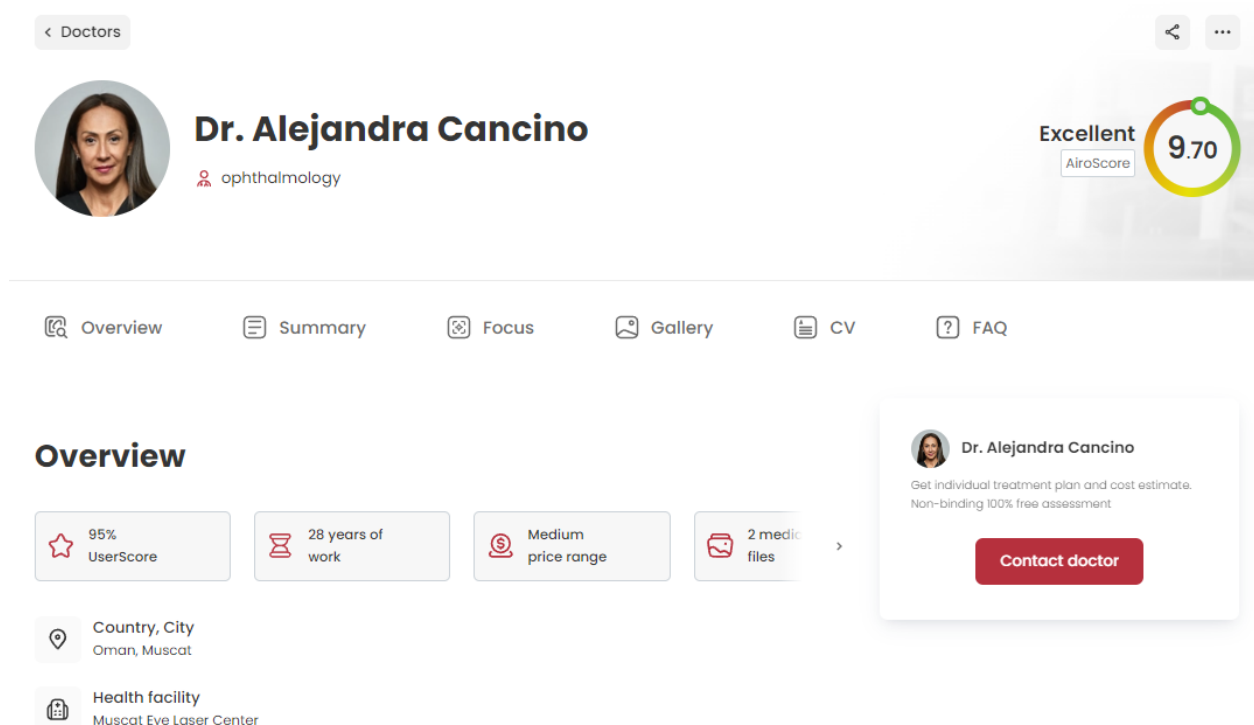


Рисунок 3.17 – Сторінка «Профіль лікаря»

Секція «Навички та досвід» зображена на рисунку 3.18 та демонструє ключові області знань та навичок, якими володіє лікар.

Skills & Expertise

50 expert tags for Prof. Dr. med. Wolfgang Henrich. Our tools analyze the strengths of the doctor using artificial intelligence. Check the data selected according to the AI policy below.

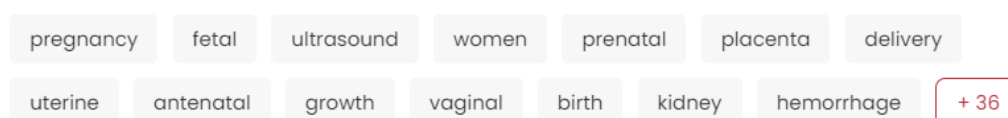


Рисунок 3.17 – Секція «Навички та досвід»

На рисунку 3.18 представлена секція «Медичні публікації» яка відображає перелік наукових робіт лікаря.

Publications

505 scientific publications from Prof. Dr. med. Wolfgang Henrich. Check the most relevant ones according to the [editorial policy](#) below.

Human papillomavirus prevalence in pregnant women living with human immunodeficiency virus infection: a scoping review of the literature.

7 August, 2023

Spectrum of congenital anomalies of the kidney and urinary tract (CAKUT) including renal parenchymal malformations during fetal life and the implementation of prenatal...

August, 2023

Comment: pregnancy after bariatric surgery – achievements and open questions

August, 2023

[Show more >](#)



Рисунок 3.18 – Секція «Медичні публікації»

На рисунку 3.19 представлена секція «Розташування» яка відображає місце де знаходить лікарня у якій працює лікар. Також у користувача є можливість прочитати детальну інформацію про це місто та країну.

Location



Address

Charitépl., 1, 10117
Berlin, Germany

About this place

Berlin is a place with a rich and amusing history. It is well-known a...

[Read more >](#)

Рисунок 3.18 – Секція «Розташування»

Сторінка «Каталог програм лікування» (рис 3.19) представляє різні медичні пакети, доступні для бронювання через медичну платформу. Користувачі можуть фільтрувати варіанти за ціною, тривалістю лікування, типом догляду (амбулаторний, стаціонарний), а також швидкістю бронювання. Кожен пакет включає детальний опис послуг, що надаються, та їх вартість, що спрощує для пацієнтів процес вибору програми лікування за їх потребами.

Treatments & Packages in Germany

We found 798 packages in Germany. Our database has 809 treatment deals. Find and book the best option for you with AiroMedical. Require our support? Our medical team is ready to help 24/7.

The screenshot displays a search interface for medical treatments in Germany. On the left, there are filter panels for Price, Duration, Type of care, and Fast booking. The main content area shows two treatment cards. The first card is for a 'Health check-up for men' with a price starting at €6,889.40 (discounted from €7,252.00). The second card is for a 'Basic check-up' with a price starting at €2,728.61 (discounted from €2,813.00). Both cards include details about availability, insurance, and diagnostic services.

Price

Min price, €: 290 | Max price, €: 412,540

Duration

<input type="checkbox"/> less than 3 days	30
<input type="checkbox"/> 3 - 7 days	530
<input type="checkbox"/> 8 - 14 days	184
<input type="checkbox"/> more than 14 days	54

Type of care

<input type="checkbox"/> inpatient	269
<input type="checkbox"/> outpatient	520
<input type="checkbox"/> distant	9
<input type="checkbox"/> inpatient/outpatient	

Fast booking

Health check-up for men

Available in: +5 others

Men get to talk to cardiologists, gastroenterologists, urologists, and ophthalmologists as part of their health checkups. Each doctor looks at th...

3 days | fast booking | outpatient | insurance | diagnostic

from ~~€7,252.00~~ -5% **€6,889.40**

Check program

Basic check-up

Available in: +6 others

A primary health check-up generally includes a physical examination, lab tests, and instrumental studies: Regular check-ups help detect and...

3 days | fast booking | outpatient | insurance | diagnostic

from ~~€2,813.00~~ -3% **€2,728.61**

Check program

Рисунок 3.18 – Сторінка «Каталог програм лікування»

Висновки до розділу

У цьому розділі розроблено структуру проектів медичної платформи та структуру веб-вузла системи. Описано можливості веб-додатку та наведено результат реалізації програмного забезпечення.

ВИСНОВКИ

В даній кваліфікаційній роботі було створено веб-додаток для медичної платформи та CRM системи для керування нею із використанням Next.js, React.js, Node.js та AdonisJS. Додаток дозволяє користувачам легко знаходити лікарні, лікарів та медичні послуги онлайн, що робить медичну допомогу більш доступною. Медична платформа забезпечує користувачів детальною інформацією, допомагаючи їм робити обґрунтовані рішення про своє здоров'я.

Досліджено предметну область та основні підходи до розробки програмного продукту. Проаналізовано існуючі аналоги систем обліку витрат. Складено специфікацію вимог до системи та описано поведінку системи за допомогою діаграм послідовності та варіантів використання.

Вибрано програмне середовище розробки медичної платформи та CRM системи для керування нею, розроблено архітектуру системи, спроектовано та створено реляційну базу даних для збереження інформації та вибрану інфраструктуру яка може легко масштабуватись.

Розроблено структуру проектів медичної платформи та CRM системи для керування нею, а також структуру веб-вузла системи. Реалізовано програмне забезпечення для системи обліку витрат.

ПЕРЕЛІК ПОСИЛАНЬ НА ДЖЕРЕЛА

1. Медична платформа Booking Health. – [Електронний ресурс]. – Режим доступу: <https://bookinghealth.net>
2. Медична платформа PRIMO MEDICO. – [Електронний ресурс]. – Режим доступу: <https://www.primomedico.com/en>
3. Медична платформа MediGlobus. – [Електронний ресурс]. – Режим доступу: <https://mediglobus.com>
4. Медична платформа Qunomedical. – [Електронний ресурс]. – Режим доступу: <https://www.qunomedical.com/en>
5. Документація Node.js v18.19.0. – [Електронний ресурс]. – Режим доступу: <https://nodejs.org/docs/latest-v18.x/api/index.html>
6. Документація AdonisJS v5.4. – [Електронний ресурс]. – Режим доступу: <https://docs.adonisjs.com/guides/introduction>
7. Документація MySQL 8.0. – [Електронний ресурс]. – Режим доступу: <https://dev.mysql.com/doc/refman/8.0/en/introduction.html>
8. Документація Redis – [Електронний ресурс]. – Режим доступу: <https://redis.io/docs/get-started/data-store/>
9. Документація Next.js. – [Електронний ресурс]. – Режим доступу: <https://nextjs.org/docs>
10. Документація React.js. – [Електронний ресурс]. – Режим доступу: <https://react.dev/learn>
11. Документація Zustand. – [Електронний ресурс]. – Режим доступу: <https://github.com/pmndrs/zustand>

ДОДАТКИ

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ
УНІВЕРСИТЕТ ІМЕНІ ІВАНА ПУЛЮЯ**

МАТЕРІАЛИ

XI НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**«ІНФОРМАЦІЙНІ МОДЕЛІ,
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



13-14 грудня 2023 року

**ТЕРНОПІЛЬ
2023**

УДК 004.9

Максим Гурак; Дмитро Михалик, к.т.н., доцент

Тернопільський національний технічний університет імені Івана Пулюя

**РОЗРОБКА ПЛАТФОРМИ МЕДИЧНИХ ПОСЛУГ ТА CRM СИСТЕМИ ДЛЯ
КЕРУВАННЯ НЕЮ З ВИКОРИСТАННЯМ NEXT.JS, NODE.JS, ADONIS.JS**

Maksym Hurak; Dmytro Mykhalyk, Ph.D; Assoc. Prof.

**DEVELOPMENT OF A HEALTHCARE SERVICES PLATFORM AND CRM
SYSTEMS FOR ITS MANAGEMENT USING NEXT.JS, NODE.JS, ADONIS.JS**

У сучасному світі, де висока доступність медичних послуг та ефективне управління здоров'ям є надзвичайно важливими, використання фреймворку Next.js у поєднанні з інтеграцією системи управління клієнтами (CRM) стає визначальним фактором для створення швидкого та динамічного веб-інтерфейсу медичної платформи [1]. Аналіз даних у реальному часі виявляється ключовим елементом у покращенні точності діагностики та підвищенні ефективності лікування [2]. У цьому контексті, забезпечення високого рівня безпеки та конфіденційності медичної інформації не лише є пріоритетом, але і формує основу для довіри пацієнтів та відповідає найвищим стандартам у галузі медичної інформатики.

Платформа визначається своєю високотехнологічністю та інтегрованістю, надаючи широкий спектр медичних послуг та використовуючи інструменти управління клієнтами для оптимізації роботи медичних закладів. Головною метою цієї інновації є передові інструменти на основі штучного інтелекту. Платформа надає інформацію про фокус і сильні сторони кожного закладу, гарантуючи, що пацієнти знайдуть найбільш підходящого постачальника медичних послуг. Пацієнти можуть детально вивчати профілі клінік, переглядати віртуальні галереї, читати справжні відгуки попередніх пацієнтів та залишати власні.

Ефективне використання наступних технологій, таких як: React.js Next.js, Node.js, AdonisJS у розробці інтегрованої медичної платформи дозволяє забезпечити швидку та ефективну навігацію для пацієнтів [3, 4]. Дослідження базується на аналізі функціональності та ефективності платформи в реальних умовах використання. Залучення реальних медичних закладів та пацієнтів дозволяє отримати об'єктивні дані щодо впливу платформи на якість надання медичних послуг та підвищення ефективності управління. Це сприяє точнішим діагнозам та персоналізованому лікуванню, адаптованому до індивідуальних потреб кожного пацієнта.

Таким чином, зазначений підхід не лише оптимізує роботу медичних закладів, а й піднімає якість та ефективність надання медичних послуг.

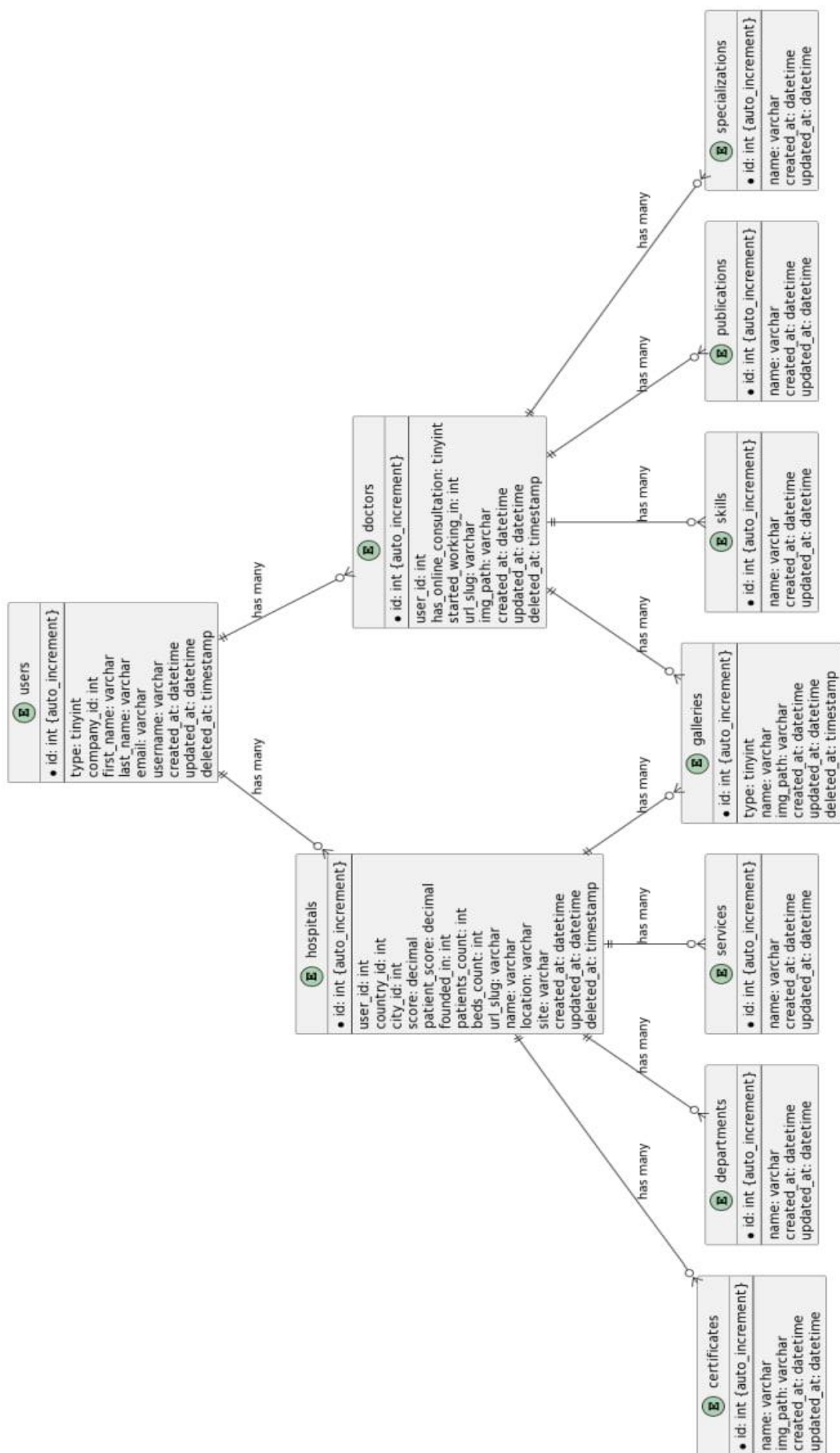
Література

1. Документація Next.js. – [Електронний ресурс]. – Режим доступу: <https://nextjs.org/docs>
2. Johnson, M. CRM Systems in Healthcare: Implementation and Benefits / M. Johnson // Journal of Healthcare Technology - 2019.
3. Документація React.js – [Електронний ресурс]. – Режим доступу: <https://react.dev/learn>
4. Документація Node.js v18.19.0. – [Електронний ресурс]. – Режим доступу: <https://nodejs.org/docs/latest-v18.x/api/index.html>

СЕКЦІЯ 4. ПРОГРАМНА ІНЖЕНЕРІЯ ТА МОДЕЛЮВАННЯ СКЛАДНИХ РОЗПОДІЛЕНИХ СИСТЕМ

Борніцький Р.І., Яворська Є.Б., Андрійчук Н.Є. ПАРАДИГМА ІНФОРМАТИЗАЦІЇ В МЕДИЦИНІ R. Bornitskyi, E. Yavorska, N. Andriychuk THE PARADIGM OF INFORMATIZATION IN MEDICINE	194
К. Вергелес, Т. Ємел'яненко ЗАСТОСУВАННЯ МОДЕЛІ GROUNDING DINO ДО РОЗВ'ЯЗАННЯ ЗАДАЧ КОМП'ЮТЕРНОГО ЗОРУ K. Verheles, T. Yemeljanenko APPLICATION OF THE GROUNDING DINO MODEL FOR SOLVING COMPUTER VISION TASKS	195
Генік М.І. РОЗРОБКА СИСТЕМИ БРОНЮВАННЯ КВИТКІВ НА БАЗІ ТЕХНОЛОГІЙ MYSQL ТА .NET Henyk M.I. DEVELOPMENT OF THE TICKET BOOKING SYSTEM BASED ON MYSQL TA .NET TECHNOLOGIES	197
Максим Гурак; Дмитро Михалук РОЗРОБКА ПЛАТФОРМИ МЕДИЧНИХ ПОСЛУГ ТА CRM СИСТЕМИ ДЛЯ КЕРУВАННЯ НЕЮ З ВИКОРИСТАННЯМ NEXT.JS, NODE.JS, ADONIS.JS Maksym Hurak; Dmytro Mykhaluk DEVELOPMENT OF A HEALTHCARE SERVICES PLATFORM AND CRM SYSTEMS FOR ITS MANAGEMENT USING NEXT.JS, NODE.JS, ADONIS.JS	198
Н.Я.Дарнобит, Г.Б.Цуприк РОЗРОБКА ANDROID-ЗАСТОСУНКУ ДЛЯ ПОШУКУ ТА ЗБОРУ ІНФОРМАЦІЇ МОВОЮ JAVA N.Ya.Darnobyt, H.B. Tsupryk DEVELOPMENT OF ANDROID APPLICATION FOR INFORMATION SEARCHING AND COLLECTION ON JAVA LANGUAGE	199
Долінський А.М. ВИКЛИКИ ЕТИЧНОГО ВИМІРУ ТА ІНТЕРПРЕТАЦІЇ У МОДЕЛЯХ МАШИННОГО НАВЧАННЯ Dolinskiy A. M. ETHICAL CHALLENGES AND INTERPRETATION IN MACHINE LEARNING MODELS	200
Дроздов В.Я., Яворська Є.Б., Андрійчук Н.Є. РОЗРОБКА ПРОГРАМНО-АПАРАТНИХ ЗАСОБІВ ВІДБОРУ ТА АНАЛІЗУ БІОСИГНАЛІВ V. Drozdov, E. Yavorska, N. Andriychuk DEVELOPMENT OF SOFTWARE AND HARDWARE TOOLS FOR SELECTING AND ANALYZING BIOSIGNALS	201
Зайченко М. І., Швирло К.Б. ДОСЛІДЖЕННЯ МОДЕЛЕЙ І МЕТОДІВ УПРАВЛІННЯ ІТ-ПРОЄКТОМ ДЛЯ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ПЛАНУВАННЯ ЧАСУ ТА УПРАВЛІННЯ ЗАДАЧАМИ Zaichenko Maryna, Shvyrlo Kostanyu RESEARCH OF MODELS AND METHODS OF IT PROJECT MANAGEMENT TO IMPROVE THE EFFICIENCY OF TIME PLANNING AND TASK MANAGEMENT	202
П.М. Костур ТЕХНОЛОГІЇ ПІДТРИМКИ ТУМАННИХ ОБЧИСЛЕНЬ P.M. Kostur FOG COMPUTING SUPPORT TECHNOLOGIES	204

Додаток Б – ER – діаграма основної частини системи



```

export default function AboutBlock({ aboutContent, aboutModalContent, title }: AboutBlockProps) {
  const contentRef = useRef<HTMLDivElement>(null);
  const [isShowMore, setIsShowMore] = useState(true);

  useEffect(() => {
    if (contentRef.current && contentRef.current.scrollHeight <= contentRef.current.clientHeight) {
      setIsShowMore(false);
    }
  }, []);

  return (
    <>
      <h2 className={doctorStyles.sectionTitle}>{title}</h2>
      <div
        className={cn(styles.message)}
        dangerouslySetInnerHTML={{ __html: aboutContent }}
        ref={contentRef}
      />
      {isShowMore
        && <AboutBlockModal
          message={aboutModalContent}
          buttonContent={
            <>
              Read more
            </>
          }
          title={title}
        />
      }
    </>
  );
};

```

```

24 const MAX_ITEM = 5;
25 const DescriptionListModal = dynamic(() => import('./DescriptionListModal'));
26
27 export default function DescriptionList({
28   collapsibleClassName,
29   headerClassName,
30   titleClassName,
31   contentClassName,
32   contentContainerClassName,
33   data,
34   iconHeader,
35   showModal = false,
36 }: DescriptionListProps) {
37   const [openIndex, setOpenIndex] = useState(0);
38   const [isModalOpen, setIsModalOpen] = useState(false);
39
40   return (
41     <>
42     <div className={cn(styles.descriptionWrapper, 'dl')}>
43       {data.map((item, index) => {
44         return (
45           <DescriptionItem
46             style={showModal && index > MAX_ITEM - 1 ? { display: 'none' } : undefined}
47             key={index}
48             collapsibleClassName={collapsibleClassName}
49             headerClassName={headerClassName}
50             titleClassName={titleClassName}
51             contentClassName={contentClassName}
52             contentContainerClassName={contentContainerClassName}
53             header={item.title}
54             description={item.description}
55             open={index === openIndex}
56             onOpen={() => setOpenIndex(index === openIndex ? -1 : index)}
57             iconHeader={iconHeader}
58           />
59         );
60       })}
61     </div>
62     {data.length > MAX_ITEM && showModal && (
63       <>
64         <Button className={styles.showMoreButton} variant={'outlined'} onClick={() => setIsModalOpen(true)}>
65           Show more
66         </Button>
67         <DescriptionListModal
68           data={data}
69           title={'Features & Facts'}
70           openModal={isModalOpen}
71           close={() => setIsModalOpen(false)}
72           collapsibleClassName={collapsibleClassName}
73           headerClassName={headerClassName}
74           titleClassName={titleClassName}
75           contentClassName={contentClassName}
76           contentContainerClassName={contentContainerClassName}
77           iconHeader={iconHeader}
78         />
79       </>
80     )}
81   </>
82 );
83 }

```