

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНФОРМАТИКИ ТА КОМП'ЮТЕРНОЇ ТЕХНІКИ

Рівень вищої освіти	Перший (бакалаврський)
Спеціальність	Інформаційні системи та технології
Освітня програма	Інформаційні системи та технології
Група	6.04.126.010.18.1

ДИПЛОМНИЙ ПРОЕКТ

на тему: «Розроблення front-end частини сайту інформаційної системи Нововодолазької центральної районної лікарні»

Виконав: студент Артем КОНДРАТЕНКО

Керівник: к.т.н., доцент Ольга ТЮТЮНИК

Рецензент: к.т.н., доцент Віталій МАРТОВИЦЬКИЙ

Харків – 2022 рік

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

Факультет	<u>Інформаційних технологій</u>
Кафедра	<u>Інформатики та комп'ютерної техніки</u>
Освітній ступень	<u>Бакалавр</u>
Спеціальність	<u>126 «Інформаційні системи та технології»</u>

Завідувач кафедри
інформатики та комп'ютерної техніки
_____ проф. Сергій УДОВЕНКО
"01" лютого 2022р.

**З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТУ**

Кондратенко Артему

1. Тема проекту: «Розроблення front-end частини сайту інформаційної системи Нововодолазької центральної районної лікарні»

керівник проекту: Тютюнник Ольга, к.т.н., доцент, доцент кафедри ІКТ
затверджені наказом ректора від "01" лютого 2022 р. №178-С

2. Строк подання студентом проекту: «08» червня 2022 року

3. Вихідні дані до проекту: нормативно-правові, наукові джерела з питань програмного забезпечення, інформація про організацію роботи медичного закладу.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

Розділ 1. Аналіз предметної області та постановка задач дослідження

Розділ 2. Специфікація вимог

Розділ 3. Проектні рішення розроблення front-end частини сайту інформаційної системи Нововодолазької центральної районної лікарні

Розділ 4. Технічні рішення розроблення front-end частини сайту інформаційної системи Нововодолазької центральної районної лікарні

5. Перелік графічного матеріалу:

IDEF0 діаграма, екранні форми аналогів медичної системи, DFD діаграма, блок-схема алгоритму роботи ресурсу, структурна схема ресурсу, UML-діаграма варіантів використання; логічна модель бази даних, фізична

модель бази даних, екранні форми розробленого веб-сайту, що запущений за допомогою веб-серверу OpenServer.

6. Консультація розділів дипломного проекту:

Розділ	Прізвище, ім'я та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання: "01" лютого 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту	Термін виконання етапів проекту	Примітка
1	Розроблення плану дипломного проекту, ознайомлення з літературними джерелами за темою	01.02.2022	
2	Постановка завдань дослідження та проведення змістовного аналізу предметної області	10.03.2022	
3	Розроблення специфікації вимог, алгоритму роботи сайту медичного закладу	30.03.2022	
4	Розроблення проектних рішень	20.04.2022	
5	Розроблення технічних рішень	15.05.2022	
6	Перевірка чернетки дипломного проекту та внесення зміни до неї керівником	20.05.2022	
7	Перевірка якості дипломного проекту на плагіат	25.05.2022	
8	Оформлення дипломного проекту	01.06.2022	
9	Подання Голові Екзаменаційної комісії щодо захисту дипломного проекту	08.06.2022	

Студент
Керівник проекту

Артем КОНДРАТЕНКО
Ольга ТЮТЮНИК

РЕФЕРАТ

Пояснювальна записка до дипломного проекту: 82 сторінки, 33 рисунок, 3 таблиці, 43 джерела.

Об'єктом дослідження є сайт інформаційної системи Нововодолазької центральної районної лікарні.

Метою роботи є розроблення front-end частини сайту інформаційної системи Нововодолазької центральної районної лікарні.

Методами розроблення обрано метод ручного написання сайтів за допомогою мови розмітки гіпертексту HTML, мови стилю сторінок CSS, мови програмування JavaScript та системи управління сайтами WordPress. Для розробки сайту обрано редактор вихідного коду VisualStudioCode.

У результаті виконання роботи обґрунтовано актуальність обраної теми, проведено змістовний аналіз предметної області, розроблені відповідні проектні та технічні рішення та розроблено front-end частину сайту для онлайн запису на прийом до лікаря.

Результати дипломного проекту можуть бути впроваджені до інформаційної системи Нововодолазької центральної районної лікарні.

Теоретичні результати дипломного проекту рекомендовано для використання у навчальному процесі кафедри інформатики та комп'ютерної техніки Харківського національного економічного університету імені Семена Кузнеця щодо підготовки здобувачів вищої освіти першого (бакалаврського) рівня за освітньо-професійною програмою «Інформаційні системи та технології» під час вивчення освітньої компоненти "Об'єктно-орієнтоване програмування".

ІНФОРМАЦІЙНА СИСТЕМА, МЕДИЧНИЙ ЗАКЛАД, ЛІКАРНЯ,
ДИЗАЙН, CMS, FRONT-END, ПРОЕКТУВАННЯ, ТЕСТУВАННЯ

ABSTRACT

The bachelor's thesis report: 82 pages, 33 figures, 3 table, 43 sources.

The object of research is the front-end part of the site of Central District Hospital of Nova Vodolaha.

The purpose of the work is to develop the front-end part of the site of the information system of Central District Hospital of Nova Vodolaha.

The methods of development were the method of manual writing of sites using HTML hypertext markup language, CSS page style language, JavaScript programming language and WordPress site management system. The VisualStudioCode source code editor was chosen to develop the site.

As a result of the work the relevance of the chosen topic was substantiated, a meaningful analysis of the subject area was conducted, appropriate design and technical solutions were developed and the front-end part of the site for online registration for a doctor's appointment was developed.

The results of the diploma project can be implemented in the information system of Central District Hospital of Nova Vodolaha.

Theoretical results of the diploma project are recommended for use in the educational process of the Department of Informatics and Computer Engineering of Simon Kuznets Kharkiv National University of Economics to prepare applicants for higher education of the first (bachelor) level in the educational program "Information Systems and Technologies". "Object Oriented Programming".

ЗМІСТ

ПЕРЕЛІК ВИКОРИСТАНИХ СКОРОЧЕНЬ.....	8
ВСТУП.....	9
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ.....	11
1.1 Змістовний опис і аналіз предметної області.....	11
1.2 Поняття веб-сайту, види розробок	14
1.3 Огляд програмного забезпечення лікарських закладів	20
1.4 Розроблення діаграми бізнес-процесів системи «Інформаційна система Нововодолазької центральної районної лікарні»за стандартом IDEF0.....	26
1.5 Розроблення діаграми потоків даних системи «Інформаційна система Нововодолазької центральної районної лікарні»за стандартом DFD.....	31
1.6 Постановка задачі дослідження.....	33
РОЗДІЛ 2 СПЕЦИФІКАЦІЇ ВИМОГ.....	37
2.1 Специфікація варіантів використання.....	37
2.2 Розроблення алгоритму та дерева цілей розроблення веб-сайту медичного закладу.....	41
2.3 Вибір мови програмування.....	45
РОЗДІЛ 3 ПРОЕКТНІ РІШЕННЯ РОЗРОБЛЕННЯ FRONT-END ЧАСТИНИ САЙТУ НОВОВОДОЛАЗЬКОЇ ЦЕНТРАЛЬНОЇ РАЙОННОЇ ЛІКАРНІ.....	47
3.1 Проектування структури бази даних.....	48
3.2 Проектування логічної моделі даних.....	51
3.3 Проектування фізичної моделі даних.....	52
РОЗДІЛ 4. ТЕХНІЧНІ РІШЕННЯ РОЗРОБЛЕННЯ FRONT-END ЧАСТИНИ САЙТУ НОВОВОДОЛАЗЬКОЇ ЦЕНТРАЛЬНОЇ РАЙОННОЇ ЛІКАРНІ.....	54

4.1 Проектування front-end частини сайту Нововодолазької центральної районної лікарні.....	54
4.2 Тестування розробленого програмного додатку.....	56
ВИСНОВКИ.....	65
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	67
ДОДАТОК А.....	72

ПЕРЕЛІК ВИКОРИСТАНИХ СКОРОЧЕНЬ

БД	база даних
ВОЯ	внутрішня оцінка якості
ГОСТ	галузевий стандарт
ЗОЯ	зовнішня оцінка якості
ЗУЯ	загальне управління якістю
КДЛ	клініко-лабораторне дослідження
КК	контрольна картка
КЯ	контроль якості
ЛІС	лабораторно-інформаційна система
ЛПЗ	лікувально-профілактичний заклад
МОЗ	Міністерство охорони здоров'я
ОС	операційна система
ПДЗ	гранично допустимі значення
ПК	персональних комп'ютер
ППП	періодична перевірка підконтрольності
AMD	Advanced Micro Devices (процесор)
HDD	hard disk drive (жорсткий диск)
RAM	Random Access Memory (оперативна пам'ять)

ВСТУП

Стрімкий розвиток коронавірусної інфекції по всьому світу призвело до суттєвих змін у всіх сферах життя людини починаючи з 2020 року і далі. Реагуючи на розповсюдження цієї пандемії, в Україні проводиться низка комплексних заходів у відповідності до Постанови Кабінету Міністрів України від 9 грудня 2020 р. № 1236 «Про встановлення карантину та запровадження обмежувальних протиепідемічних заходів з метою запобігання поширенню на території України гострої респіраторної хвороби COVID-19, спричиненої коронавірусом SARS-CoV-2» [1], що спрямовані на запобігання та локалізацію небезпеки медико-біологічного характеру [2].

Уряд країни Постановою Кабінету Міністрів від 22 вересня 2021 р. № 981 «Про внесення змін до деяких актів Кабінету Міністрів України» внесено зміни до постанови Кабінету Міністрів України від 9 грудня 2020 року № 1236 продовживши дію карантину до 31 грудня 2021 року. При цьому з 23 вересня 2021 року на всій території країни встановлюється “жовтий” рівень епідемічної небезпеки [2].

У зв’язку із складною епідемічною ситуацією режим надзвичайної ситуації на всій території України, запроваджений відповідно до розпорядження Кабінету Міністрів України від 25.03.2020 № 338-р було подовжено до 31 серпня 2021 року [2].

20 вересня 2021 року Міністерство охорони здоров’я оновило єдиний державний протокол лікування хворих на COVID-19. Крім того, головним державним санітарним лікарем України розробляються та затверджуються протиепідемічні заходи, які визначають особливості провадження діяльності суб’єктами господарювання на період карантину [2-3].

Основні рекомендації дотримання протиепідемічної безпеки – дотримання дистанції, носіння маски та користування санітаїзерами. В період пандемії на території України людям похилого віку та мешканцям країни Міністерство охорони здоров’я рекомендовало менше часу перебувати в

громадських будинках і спорудах, громадському транспорті, медичних закладах та інших місцях з масовим перебуванням людей.

Через велику кількість хворих на COVID-19 всі медичні заклади країни працюють у посиленому режимі. Оскільки, згідно рекомендації МОЗ, необхідно проводити як можна менше часу у місцях з масовим перебуванням людей, зокрема лікарнях, мешканці міст, районів областей намагаються знаходити всю необхідну для них інформацію на сайтах медичних закладів.

Сайт медичного закладу – це сучасний та ефективний спосіб спілкування з аудиторією. Сайт медичного закладу повинен бути зручним та зрозумілим, реалізована можливість перемикання на версію для людей з обмеженими можливостями [2].

Саме тому при розробленні сайту велика увага приділяється front-end частини сайту медичного закладу, що і визначило актуальність та перспективність обраної теми дипломного проекту.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Змістовний опис і аналіз предметної області

Для ефективного функціонування підприємств, закладів та організацій країни необхідно удосконалювати методи управління. Впровадження в роботу підприємств, закладів та організацій інформаційних систем є одним з інструментів покращення систем управління підприємством або установою.

Лікарні відносяться до установ сфери охорони здоров'я України. Ці установи є багатофакторними об'єктами з великою кількістю вхідних та вихідних параметрів, а також внутрішніми прихованими (латентними) факторами.

Лікарня – це медичний заклад та лікувально-профілактична установа, яка надає стаціонарну кваліфіковану і спеціалізовану медичну допомогу. В Україні залежно від адміністративно-територіального розподілу розрізняють обласні, міські, центральні районні та сільські лікарні [4].

На території Харківської області функціонують 123 лікарських заклади. Наразі вони працюють за сьома госпітальними кругами [5]:

- 1-й округ - Богодухівський, Краснокутський, Коломацький, Валківський, Золочівський і Дергачівський райони (загальна чисельність населення - 227 тис. 747 осіб), 6 центрів первинної медико-санітарної допомоги;

- 2-й округ - місто Люботин і Харківський район (203 тис. 556 осіб), 6 центрів первинної медико-санітарної допомоги;

- 3-й округ – Зміївський, Нововодолазький, Чугуївський та Печенізький райони (194 тис. 361 особа), 6 центрів первинної медико-санітарної допомоги;

- 4-й округ – Великобурлуцький, Вовчанський, Шевченківський, Куп'янський та Дворічанський райони (188 тис. 522 особи), 5 центрів первинної медико-санітарної допомоги;

- 5-й округ – Сахновщинський, Красноградський, Лозівський, Близнюківський, Кегичівський та Зачепилівський райони (217 тис. 183 особи), 7 центрів первинної медико-санітарної допомоги;

- 6-й округ – Балаклійський, Ізюмський, Барвінківський, Борівський та Первомайський райони (234 тис. 714 осіб), 6 центрів первинної медико-санітарної допомоги;

- 7-й округ – місто Харків.

Особливістю цієї будови є те, що в кожному з округів функціонуватиме декілька лікарень першого рівня інтенсивного лікування та декілька лікарень другого рівня, а також 6 -7 центрів первинної медико-санітарної допомоги та не менше 2 потужних підрозділів служби екстреної медичної допомоги [5].

Відповідно до постанови КМУ №1074 від 27.10.2019 «Деякі питання створення госпітальних округів» на території країни наразі також функціонують опорні заклади охорони здоров'я.

Опорні заклади охорони здоров'я - це багатoproфільні заклади охорони здоров'я державної та комунальної форми власності, які відповідають або можуть бути підвищені до рівня багатoproфільної лікарні інтенсивного лікування першого або другого рівня [6].

Проте, в будь-якій лікарській уставові України повинна діяти ефективна інформаційна системи для вдалої роботи установи.

Питанням автоматизації роботи лікарських установ займаються багато розробників. Наразі існує напрям у штучному інтелекті – експертні системи в медицині, які здебільшого є складними програмними комплексами, що акумулюють знання фахівців в конкретних предметних галузях і тиражують цей емпіричний досвід для консультацій менш кваліфікованих користувачів [7].

На теперішній час класифікація інформаційних систем в охороні здоров'я передбачає [8]: медико-технологічні інформаційні системи, інформаційно-довідкові інформаційні системи, статистичні інформаційні системи, науково-дослідні та системи, які навчають. Так за допомогою

впровадження даних інформаційних систем до медичних закладів з'явилися нові принципи зберігання інформації та полегшилася робота з ними.

Лікарня – це установа, яка надає медичні послуги для громади. Є юридичною особою, користується правом оперативного управління відносно закріпленого майна, має штамп, печатку, статут, який затверджуються згідно з чинним законодавством, поточні та інші рахунки у фінансових установах, здійснює бухгалтерський і первинний статистичний облік. Головний лікар, який управляє лікарнею, призначається на умовах контракту, в якому прописано: строк найму, права, обов'язки, відповідальність, умови його матеріального забезпечення, тощо [4].

Лікарня розташовується у спеціально збудованих чи пристосованих будівлях, які за переліком і площею службових приміщень відповідають діючим санітарно-гігієнічним, протиепідемічним, будівельним нормам та правилам, технічнобезпечовим і протипожежним вимогам.

Загальна лікарня надає послуги більшості людей та хвороб. Спеціалізована лікарня лікує певних людей або певні хвороби чи групу хвороб. Наприклад, дитячі лікарні лікують лише дітей, реабілітаційні лікарні надають послуги, щоб допомогти людям ефективно жити після тяжких хвороб.

Більшість лікарень належать місцевій громаді. Некомерційна добровільна лікарня належить благодійним організаціям чи релігійним групам. Приватна лікарня працює як компанія, щоб отримати прибуток для власників. Державна лікарня належить державі або місцевій владі. Медичний персонал, що складається з лікарів, лікує хворих і надає поради всім іншим членам команди лікарів. Більшість лікарень мають у своєму складі різні лікувальні відділення. Кількість відділень залежить від розміру та типу лікарні. Лікарня інтенсивного лікування 1-го рівня має адміністративну частину / управління лікарнею. Включає керівництво лікарні (головний лікар, його заступники), відділ кадрів, фінансово-економічний відділ та бухгалтерію.

1.2 Поняття веб-сайту, види розробок

WEB-розробка (web development) – це процес професійного програмування сайту, який регулює всі наступні етапи, пов'язані з формуванням HTML-коду, додаванням різних функціональних компонентів та скриптів, що впливають на показники юзабіліті та технічної стабільності.

Завданням розробника є створення інфраструктури для реалізації оптимального функціоналу сайту, на який у майбутньому кріпляться візуальні компоненти, що забезпечують інтерактивні можливості.

Таким чином, WEB-розробка – це процедура створення WEB-програми або WEB-сайту.

Існують такі основні етапи розробки WEB-сайту:

- проектування WEB-дodatка або самого сайту, тобто збирання та подальший аналіз усіх вимог, розробка технічного завдання, складання проекту інтерфейсів;
- розробка концепції сайту з урахуванням креативу;
- розробка дизайнерської концепції інтернет-ресурсу;
- розробка макетів сторінок сайту;
- створення та виконання FLASH-елементів та мультимедіа;
- верстання шаблонів та сторінок;
- роботи з програмного забезпечення, як-то створення функціональних інструментів, або ж інтеграція у вже існуючу систему управління вмістом, т.зв. CMS;
- розміщення на сайті та оптимізація його текстових матеріалів;
- тестування сайту та внесення, за необхідності, коригування;
- запуск створеного проекту на майданчику в мережі Інтернет;
- роботи з обслуговування вже чинного порталу або його програмної частини.

Однак, залежно від необхідного завдання, якісь із вищевказаних етапів у процесі WEB-розробки можуть і не використовуватися, або ж бути тісно взаємопов'язані один з одним.

Слід зазначити, що всі етапи проекту WEB-сайту досить сильно залежні від безлічі різних факторів, як, наприклад, величина обсягу інтернет-порталу, його функціональність, а також завдання, для яких призначений створюваний інтернет-ресурс та багато іншого. Але, тим не менш, є і кілька етапів, які обов'язково присутні при плануванні будь-якого майбутнього проекту [9].

Будь-яка робота по інтернет-сайту починається зі створення його дизайну, використовуючи для цього графічний редактор. WEB-дизайнер створює, як правило, кілька таких варіантів, але в суворій відповідності до ТЗ.

При цьому окремо розробляється дизайн «Головної» сторінки сайту. А далі – дизайн інших типових сторінок (новини, статті, про нас, каталог). Власне, сам «дизайн» являє собою графічний файл, як листковий малюнок, що включає більш дрібні картинки у вигляді шарів у загальній картинці.

При цьому фахівець обов'язково враховує всі обмеження стандарту HTML, тобто не виробляє дизайн, який неможливо буде згодом реалізувати стандартними HTML-засобами. Винятком є тільки Flash-дизайн.

Кількість самих ескізів та порядок їх пред'явлення замовнику заздалегідь обумовлюються менеджерами всього проекту, який виконує контроль запланованих термінів. Ще, також у великих WEB-фірмах у процесі бере участь і Арт-директор, який контролює якість виконання графіки. Цей етап так само як і попередній, закінчується його твердженням у замовника.

Затверджений клієнтом дизайн далі передається фахівцеві-верстальнику, який «нарізає» графічне зображення на окремі картинки, з яких пізніше буде складена HTML-сторінка. Під час такої роботи створюється програмний код, який можна вже дивитися за допомогою будь-якого браузера (інтернет-браузера). Ну, а такі типові сторінки, згодом, будуть застосовуватися, як HTML-шаблони.

Після проведених вище згаданих заходів готові файли у форматі HTML передаються в роботи WEB-програмісту. Розробка програмного забезпечення інтернет-сайту цілком може виконуватися як «з самого нуля», так і на основі системи CMS, часто так званого «CMS-движка».

У разі застосування системи управління сайтом слід зазначити, що вона сама, в якомусь сенсі слова, вже готовий сайт, що включає блоки, що замінюються. Ну, а самого програміста, в такому разі, називають «CMS-фахівцем». Він повинен замінити існуючий стандартний шаблон на новий оригінальний, розроблений на базі початкового WEB-дизайну, з урахуванням індивідуальних побажань замовника.

При розробці програмного забезпечення інтернет-сайту спеціалісту CMS також встановлюються контрольні терміни проведення робіт. Сам такий процес цілком може містити в собі різні види перевірок, як-то, наприклад: зовнішній вигляд сторінки сайту зі збільшеними шрифтами, при різних розмірах браузерного вікна, або через відсутність Flash-плеєра, і багато іншого. Також використовується і тестування користувача, так зване – юзабіліті.

Виявлені помилки в роботі сайту відправляються для їх виправлення до тих пір, поки виконавець їх не усуне. У цьому випадку терміни роботи контролює той самий проектний менеджер. Хоча на етапі тестування ще залучають до роботи і самого дизайнера, щоб він здійснював авторський нагляд. Файли розробленого WEB-сайту розміщують на сервері, наприклад, провайдера, де здійснюють необхідні налаштування. Слід зазначити, що на цьому етапі інтернет-сайт ще закритий для широкого кола користувачів.

Новий сайт наповнюють контентом, тобто різними текстами, зображеннями, файлами тощо. Іноді самі тексти складаються копірайт-фахівцем фірми-розробника сайту, іноді ними займається сам клієнт, наймаючи автора збоку. Таке питання вирішується ще стадії складання самого ТЗ, оскільки, якщо контент розробляється копірайтером розробника, це обов'язково узгоджується із замовником одночасно з іншими етапами

реалізованого проекту. Внутрішня оптимізація пов'язана з певними змінами, що вносяться до самого інтернет-сайту.

Процес оптимізації починається зі складання так званого семантичного ядра, для якого підбираються такі ключові слова, які заволодіють найбільшою увагою користувачів порталу, і з якими обійти конкурентів простіше [10]. Згодом такі «ключовики» вносяться на сам інтернет-портал і всі гіперпосилання, тексти та метатеги адаптуються таким чином, щоб пошукові інтернет-системи змогли їх швидко виявляти за складеними «ключовиками». Зовнішнє SEO, як правило, зводиться до побудови певної структури для вхідних посилань на новий сайт. В принципі, це і є сама розкрутка нового порталу і до створення сайту зовнішня оптимізація не має жодного відношення. Сама ж SEOоптимізація поділяється на так звану: «білу» та «чорну», після проведення першої, інтернет-портал потрапляє до ТОПу, а після проведення другої – до «бану» пошукових систем.

Слід зазначити, що «біла» оптимізація – це досить тривалий і трудомісткий процес, при якому вартість його може перевищити матеріальні витрати на WEB-розробку самого сайту.

Замовник ознайомлюється з повністю готовим замовленням і у разі, коли його все повністю влаштовує, сторони підписують відповідні документи, а саме Акт виконаних робіт, про здачу всього проекту. Крім того, на такому етапі роботи здійснюється ще й навчання клієнта чи його представника необхідним навичкам адміністрування нового сайту. В якості примітки слід зазначити, що WEB-дизайн сайту, що розробляється, неодмінно зобов'язаний привабливо виглядати при використанні користувачами різних браузерів, особливо це стосується таких браузерів, як Chrome, Internet Explorer, Firefox і Opera.

Front-end розробка стосується управління клієнтським аспектом або зовнішнім інтерфейсом веб-сайтів. Це охоплює все, що бачить користувач, від тексту та зображень до меню та панелей навігації. За це відповідають фронтенд-розробники.

Інтерфейсний розробник читає файл дизайну і складає план перетворення цього дизайну на допустимий код HTML, CSS і JavaScript. Потім веб-браузер відображає цей код, коли користувач здійснює перехід на веб-сайт. HTML, CSS та JavaScript – це три основні мови програмування, які використовуються для створення зовнішнього інтерфейсу веб-сайту.

Основний обов'язок фронтенд-розробника – створити функціональний інтерфейс користувача. Користувачі повинні мати можливість легко переміщатися сайтом і отримувати відповіді, які вони шукали до того, як клацнути на сайт. Користувачі не повинні стикатися з помилками чи невідповідностями у процесі.

Існують різні області веб-розробки, і розробники, що працюють у кожній із них, мають унікальні обов'язки.

Веб-розробників серверної частини більше цікавить частина веб-сайту, яку користувач не бачить. Коди, які вони пишуть, можуть бути використані для обробки платежів на сайті або визначення контенту, який користувач бачить при відкритті сторінки. Для цього вони використовують мови програмування, такі як Python, Ruby та PHP, для створення програми.

Розробники повного стеку – мають досвід і знання з обох сторін веб-сайту.

Веб-розробники повного стеку мають вирішальне значення для будь-якого веб-проекту. Вони допомагають подолати розрив між клієнтською та серверною частиною веб-сайту. Це тягне за собою забезпечення функціональності та естетичності сайту. Звичайно, вони володіють як клієнтськими інструментами, такими як JavaScript, і серверними технологіями, такими як PHP і Ruby.

Терміни «інтерфейсний інженер» та «інтерфейсний розробник» часто використовуються як синоніми. Хоча їхні обов'язки схожі, між цими кодувальниками є кілька тонких відмінностей.

Інтерфейсні інженери більше залучені до створення зовнішнього інтерфейсу сайту. Більшість часу вони проводять за аналізом архітектури

сайту. За допомогою інших розробників та дизайнерів вони з'ясовують, як продати сайт.

З іншого боку, розробники зовнішнього інтерфейсу відповідають за написання коду зовнішнього інтерфейсу. Їхній код повинен бути підтримуваним і добре працювати в масштабі.

Хоча розуміння принципів дизайну, що лежать в основі сайту, є частиною обов'язків фронтенд-розробників, вони не витрачають багато часу на їх аналіз. Натомість вони приділяють більше уваги перетворенню макетів дизайну на код.

Розробникам зовнішнього інтерфейсу необов'язково бути експертами з веб-дизайну. Веб-дизайн – це окрема область. З урахуванням сказаного ключовим моментом є глибоке знання веб-дизайну. Дві основні частини веб-дизайну – це дизайн інтерфейсу користувача (UI) і (UX) "користувацький досвід". Тобто, це те, як користувач взаємодіє з інтерфейсом і наскільки сайт або додаток для нього зрозумілі і зручні.

Дизайн інтерфейсу користувача відноситься до створення зовнішнього вигляду веб-сторінки. Дизайнер інтерфейсу користувача вирішить, де на веб-сайті будуть відображатися такі елементи, як кнопки, текст і зображення.

UX-дизайн стосується іншого досвіду користувача на веб-сайті. Дизайнери інтерфейсу користувача аналізують і досліджують продукт, щоб визначити будь-які моменти, які можуть збентежити користувачів. Потім вони з'ясовують, як вирішити ці проблеми, щоб користувачі могли без проблем використовувати веб-сайт.

Люди використовують мобільні телефони, планшети та монітори з екранами різного розміру. Саме тут на допомогу приходить адаптивний дизайн. Адаптивний дизайн пов'язаний із створенням веб-сайту, який ефективно відображається на різних пристроях.

Це включає розуміння різних типів пристроїв, на яких користувач може переглядати сайт. Розробники зовнішнього інтерфейсу повинні мати можливість розробляти унікальні можливості для користувачів цих пристроїв.

Інтерфейси прикладного програмування (API) використовуються у різних контекстах веб-розробки. Часто серверні розробники створюють API-інтерфейси, які підтримують логіку сайту, таку як автентифікація користувачів та платежі. Потім розробники зовнішнього інтерфейсу повинні запитати ці API, щоб вони могли взаємодіяти з ними.

API також дозволяють взаємодіяти із зовнішніми службами. Наприклад, Google Sheets API дозволяє отримувати дані з Google Sheets, які можна використовувати на своєму веб-сайті. Це означає, що є можливість створювати інтеграцію поверх існуючого веб-сайту.

Тим часом робота з дизайнерами має вирішальне значення для розуміння того, як має виглядати сторінка. Не кажучи вже про всіх інших людей, з якими потрібно взаємодіяти, від менеджерів проектів та інженерів із забезпечення якості до клієнтів.

1.3 Огляд програмного забезпечення лікарських закладів

В даному дипломному проекті, для визначення недоліків роботи існуючих програмних додатків для медичних закладів, проаналізовано популярні програмні додадки, що використовуються у сучасних медичних закладах країни.

Медична система «Електронна медична карта» наразі є лідером ринку у приватному сегменті серед медичних інформаційних систем. У програмі реалізована велика кількість функцій, у тому числі і ведення графіків лікарів та запис на прийом у рамках блоку «Графік прийому лікарів».

Цей блок дозволяє складати графіки прийомів лікарів та записувати пацієнтів до лікарів. Програма відслідковує зайнятість кабінетів та попереджає помилковий запис двох різних пацієнтів в один кабінет на один і той самий час.

Рисунок 1.1 – Медична система «Електронна медична карта»

Також є можливість подивитися вільні інтервали декількох лікарів одночасно, а також внести пацієнта у лист очікування, якщо у момент запису вільних інтервалів немає. Також є можливість зателефонувати пацієнту (наявні його контактні дані) а також є можливість проводити SMS-розсилку для нагадування про візит. Програма надає звіти по завантаженню кабінетів в різний час дня, порівнює доходність різних кабінетів протягом заданого періоду часу, і оцінює ефективність роботи адміністраторів при попередньому записі пацієнтів.

«Електронна медична карта» дозволяє працювати в локальній мережі за технологією «клієнт-сервер», або із використанням віддаленого доступу через мережу Інтернет, а також можливість ведення єдиної бази даних для всіх територіально віддалених підрозділів.

До недоліків системи «Електронна медична карта» можна віднести: застаріла технологія для інтерфейсної частини (програма розроблена на Microsoft Access 2003 + VBA), відсутність у ресурсній моделі аналізу ресурсних потреб послуг та варіабельного інтервалу прийому – таким чином, система гарантує тільки незайнятість часу, а всі інші аспекти планування повинна пам'ятати людина, яка записує пацієнтів на прийом.

Робота у розподіленому середовищі із нестабільним інтернет-зв'язком в системі «Електронна медична карта» не передбачена (передбачено роботу або в локальній мережі, або через Інтернет-з'єднання із єдиною центральною базою даних). Відповідно, якщо в підрозділі медичного закладу, у якому встановлена система «Електронна медична карта», пропаде з'єднання із центральною базою даних, то будь-яка робота із планування ресурсів буде зупинена, що в свою чергу стане причиною незручностей в обслуговуванні пацієнтів.

Система Medcard24 – програма призначена для ведення бази даних пацієнтів, фіксації наданих послуг, обстежень, висновків.

Наявність журналу прийому, генерація офісних документів по шаблонах, потужний експорт і імпорт даних, зберігання картинок в БД, посилення на файли будь-яких форматів. Пошук, фільтри, сортування, групування по будь-якому полю. Можливість гнучко змінювати конфігурацію бази даних, додаючи поля і таблиці настроювання правил кольоровиділення, настройка нагадувань, створення довільних звітів.

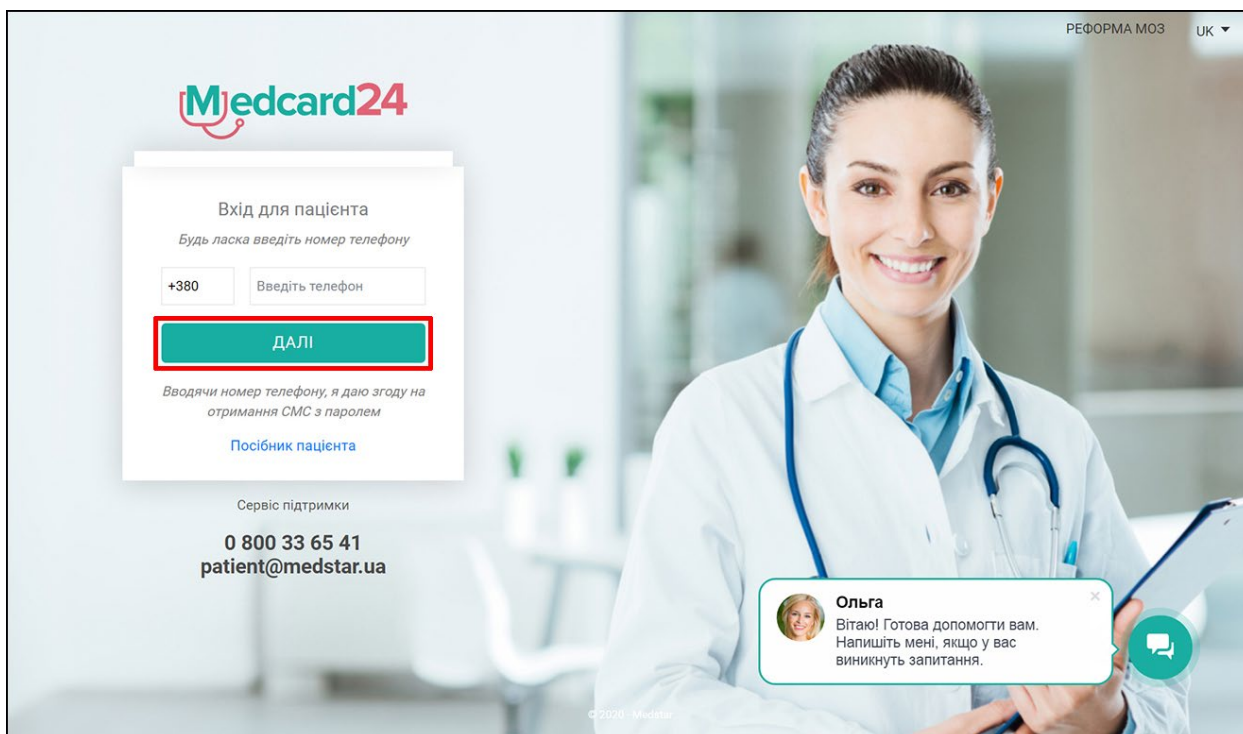


Рисунок 1.2 – Медична система «Medcard24»

Мережевий і багатокористувацький режими, гнучка настройка прав доступу кожному користувачеві і багато інші зручності.

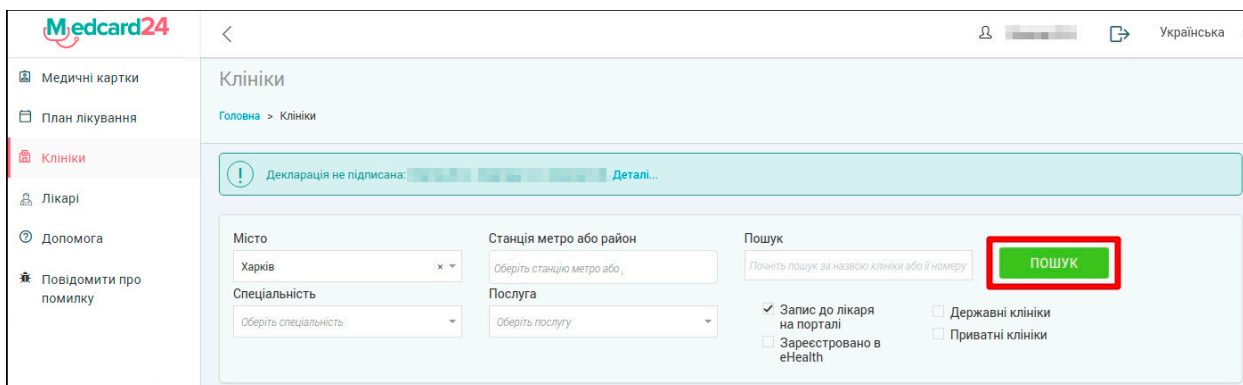


Рисунок 1.3 – Медична система «Medcard24»

Головним недоліком медичної системи «Medcard24» є те, що система не безкоштовна, середня вартість програми становить 20000 грн.

ІС Доктор Елекс – програма призначена для лікарів поліклініки та медичного персоналу. Дозволяє спростити роботу із заповненням всіх необхідних документів.

Програма дозволяє виконувати наступні функції:

- ведення журналу прийому пацієнтів;
- ведення і роздруківка щоденників амбулаторної карти
- планування поліклінічного прийому;
- можливість роботи з диспансерними хворими – планування відвідувань, прийом, швидка постановка на облік, зняття з обліку
- можливість роздруківки всіляких довідок, як-то: статистичний талон, довідка звернення, направлення на видачу листка непрацездатності, талон на закінчений випадок захворювання, направлення на фізіолікування, напрямок до медичного закладу, направлення на ВК, довідка для отримання путівки, санаторно-курортна карта тощо;
- використання шаблонів для швидкої підготовки довідок та напрямків;
- аналіз і уявлення в графічному вигляді результатів діяльності лікаря поліклініки за надання лікарської допомоги в рамках обліково-звітної форми 39 (лікарська відомість);
- генерацію звітної форми по диспансерним хворим;
- можливість контролю виконання плану по випадках захворювання, виходячи з їх вартості в залежності від алгоритму оплати.

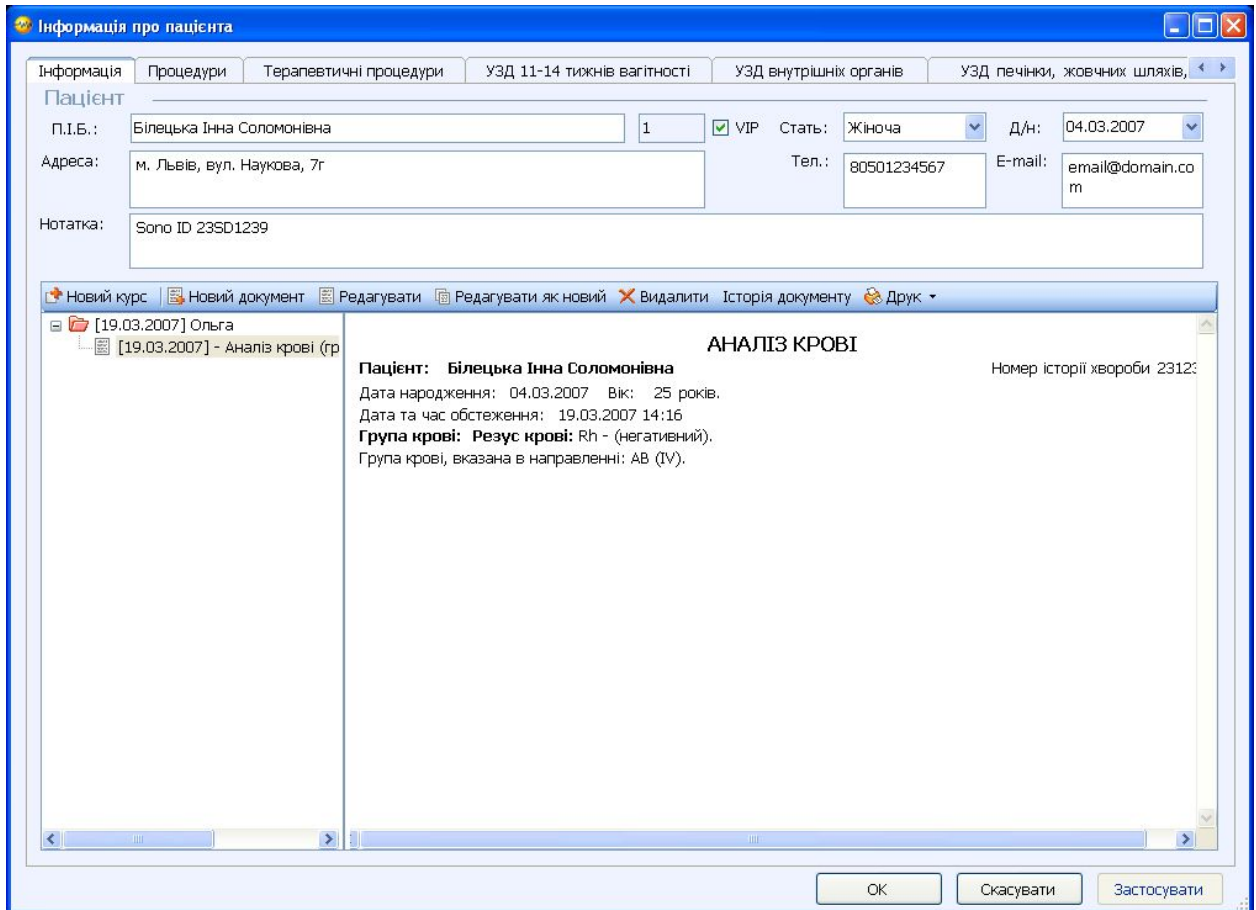


Рисунок 1.4 – Медична система ІС Доктор Елекс

Головним недоліком даного програмного продукту є вартість, складність налаштування та відсутність підказок при роботі з системою.

Також до недоліків варто віднести обмеженість функцій та відсутність розподіленого доступу.

На основі вищевикладеного аналізу аналогів розробки здійснимо порівняльний аналіз та результати занесемо до таблиці 1.1.

Таблиця 1.1

Порівняльний аналіз аналогів

№	Параметр	«Електронна медична карта»	Медична система «Medcard24»	ІС Доктор Елекс
1	2	3	4	5

1	Журнал прийому пацієнтів	+	+	+
2	Журнал зайнятості лікарів	+	-	+
3	Планування поліклінічного прийому	+	-	+
4	Математична обробка даних	+	-	-
5	Функція нагадування	-	+	-
6	Завантаження кабінетів	+	-	-
7	Ведення амбулаторної карти	-	+	+
8	Використання обліково-звітної форми	+	-	+
9	Графічний вивід результатів	+	-	+
10	Наявність Інтернет зв'язку	+	-	-
11	Об'єм займаної пам'яті, кб	990	689	560
12	Вартість, грн	4500	20000	5000

1.4 Розроблення діаграми бізнес-процесів системи «Інформаційна система Нововодолазької центральної районної лікарні» за стандартом IDEF0

Для аналізу бізнес-процесів роботи інформаційної системи Нововодолазької центральної районної лікарні, в дипломному проекті проведено проектування функціональної моделі за допомогою CASE-засобу Ramus Education.

На рис.1.5 наведено контекстну діаграму функціональної моделі «Інформаційна система Нововодолазької центральної районної лікарні».

В якості вхідних параметрів до процесу «Інформаційна система Нововодолазької центральної районної лікарні» виступають «Запис на прийом», «Інформація про пацієнта» та «Графік прийому лікаря».

Механізмами процесу «Інформаційна система Нововодолазької центральної районної лікарні» є «БД лікарні», «Лікарі», «Персонал лікарні».

Управління для процесу «Інформаційна система Нововодолазької центральної районної лікарні»: «Нормативні документи», «Статут Нововодолазької центральної районної лікарні».

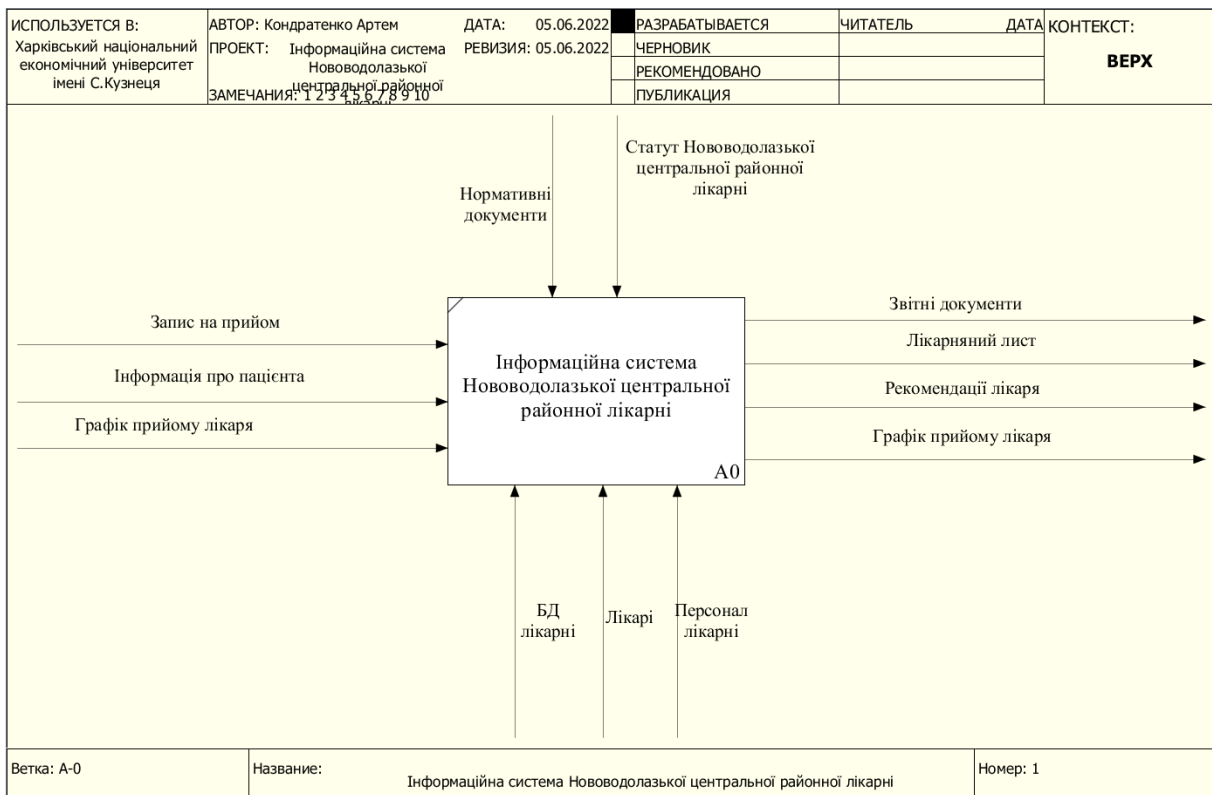


Рисунок 1.5 – Контексна діаграма функціональної моделі

В якості вихідних параметрів із процесу «Інформаційна система Нововодолазької центральної районної лікарні» виступають «Звітні документи», «Рекомендації лікаря», «Лікарняний лист» та «Графік прийому лікаря».

Декомпозиція контекстної діаграми відбувається на 6 бізнес-процесів: «Робота приймального відділення лікарні», «Прийом пацієнта терапевтом (сімейним лікарем)», «Проведення діагностики стану здоров'я пацієнта», «Встановлення попереднього/остаточного діагнозу терапевтом», «Консультація спеціаліста призначеного терапевтом», «Встановлення остаточного діагнозу та рекомендації лікаря». Декомпозиція першого рівня наведена на рис. 1.6.

До бізнес-процесу «Робота приймального відділення лікарні» надходять вхідні параметри «Запис на прийом», «Інформація про пацієнта» «Графік прийому лікаря» та «Інформація про години прийому» від «Прийом пацієнта терапевтом (сімейним лікарем)». Вихідними параметрами «Робота приймального відділення лікарні» є «Амбулаторна картка» та «Перелік пацієнтів записаних на прийом». Механізмами для цього процесу є «БД лікарні» та «Персонал лікарні». Для бізнес-процесу «Прийом пацієнта терапевтом (сімейним лікарем)» вхідними параметрами є «Амбулаторна картка», «Перелік пацієнтів записаних на прийом», «Уточнений діагноз від призначеного спеціаліста» та «Рекомендації щодо лікування» від бізнес-процесу «Встановлення остаточного діагнозу та рекомендації лікаря». Вихідними параметрами є «Перелік призначених аналізів та процедур», «Графік прийому лікаря», «Лікарняний лист» та «Рекомендації лікаря». Механізмами для процесу є «Лікарі» та «БД лікарні».

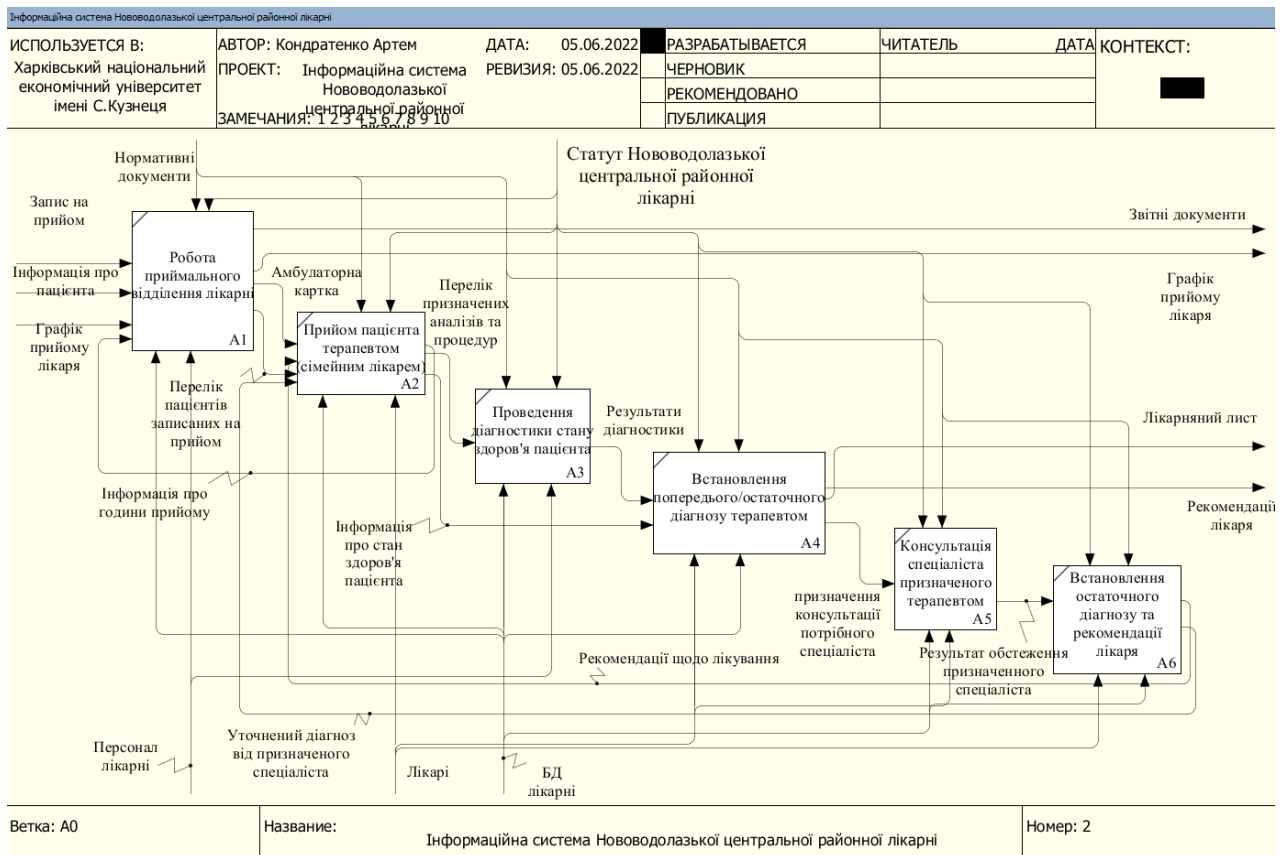


Рисунок 1.6 – Декомпозиція першого рівня

До бізнес-процесу «Проведення діагностики стану здоров'я пацієнта» надходить «Перелік призначених аналізів та процедур», на виході – «Результати діагностики». Механізмами для цього процесу є «БД лікарні» та «Персонал лікарні».

Для бізнес-процесу «Встановлення попереднього/остаточного діагнозу терапевтом» вхідними даними будуть «Результати діагностики» від «Проведення діагностики стану здоров'я пацієнта» та «Інформація про стан здоров'я пацієнта» від «Прийом пацієнта терапевтом (сімейним лікарем)». Вихідними даними будуть «Призначення консультації потрібного спеціаліста» у випадку якщо для встановлення остаточного діагнозу потрібна консультація стороннього спеціаліста (кардіолог, хирург тощо) та «Лікарняний лист» й «Рекомендації лікаря», якщо терапевт встановив остаточний діагноз. Механізмами для процесу є «Лікарі» та «БД лікарні».

Бізнес-процес «Консультація спеціаліста призначеного терапевтом» в якості вхідного параметра має «Призначення консультації потрібного спеціаліста», вихідним параметром є «Результат обстеження призначеного спеціаліста». Механізмами для процесу є «Лікарі» та «БД лікарні».

Вхідним параметром для процесу «Встановлення остаточного діагнозу та рекомендації лікаря» є «Результат обстеження призначеного спеціаліста», вихідними параметрами є «Уточнений діагноз від призначеного спеціаліста» та «Рекомендації щодо лікування», які передаються до бізнес-процесу «Прийом пацієнта терапевтом (сімейним лікарем)». Механізмами для процесу є «Лікарі» та «БД лікарні».

Управлінням для всіх процесів є «Нормативні документи» та «Статут Нововодолазької центральної районної лікарні».

Декомпозиція другого рівня бізнес-процесу «Робота приймального відділення лікарні» наведена на рис. 1.7.

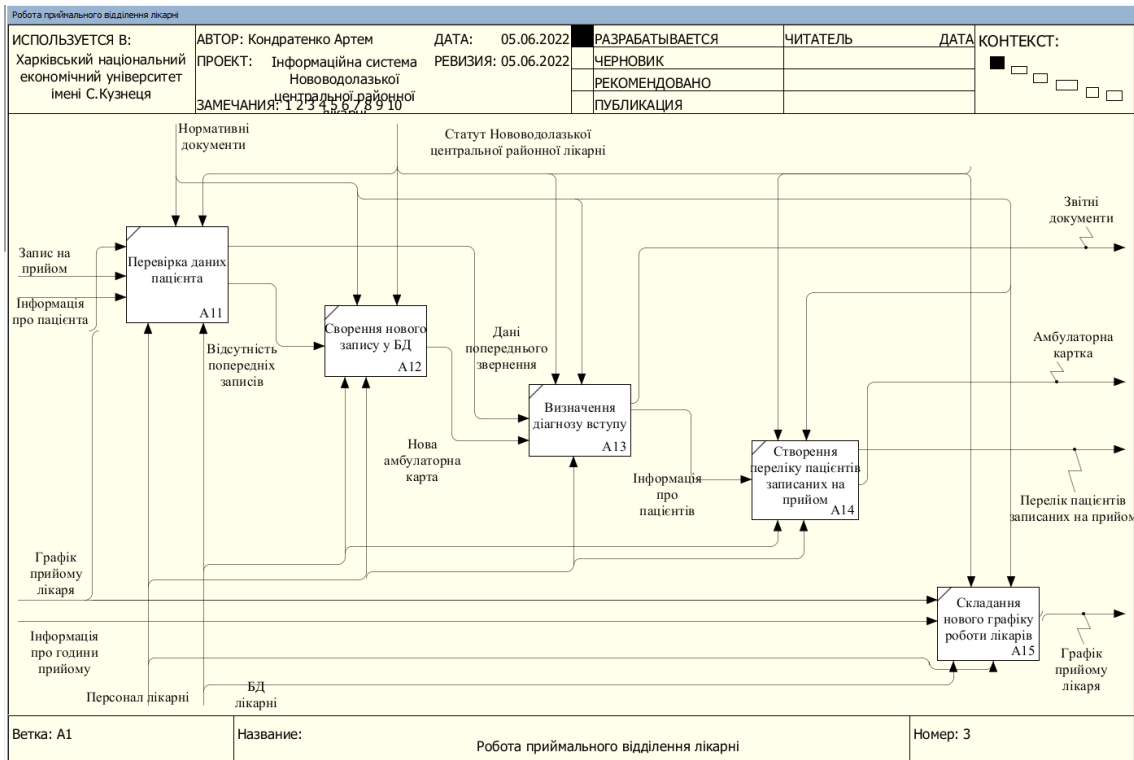


Рисунок 1.7 – Декомпозиція другого рівня бізнес-процесу «Робота приймального відділення лікарні»

1.5 Розроблення діаграми потоків даних системи «Інформаційна система Нововодолазької центральної районної лікарні» за стандартом DFD

Для аналізу потоків даних інформаційної системи Нововодолазької центральної районної лікарні, в дипломному проекті проведено проектування моделі потоків даних за допомогою CASE-засобу Ramus Education.

На рис.1.8 наведено контекстну діаграму потоків даних «Інформаційна система Нововодолазької центральної районної лікарні».

З системою «Інформаційна система Нововодолазької центральної районної лікарні» взаємодіють такі зовнішні сутності «Пацієнт», «Лікарі» та «Міністерство охорони здоров'я України». Зовнішні сутності «Пацієнт» та «Лікарі» надають таку вхідну інформацію, як «Графік прийому лікаря», «Інформація про пацієнта» та «Запис на прийом». Зовнішня сутність «Міністерство охорони здоров'я України» отримує вихідну інформацію

«Звітна інформація». Зовнішня сутність «Пацієнт» отримує інформацію «Лікарняний лист», «Рекомендації лікаря», «Графік прийому лікаря».

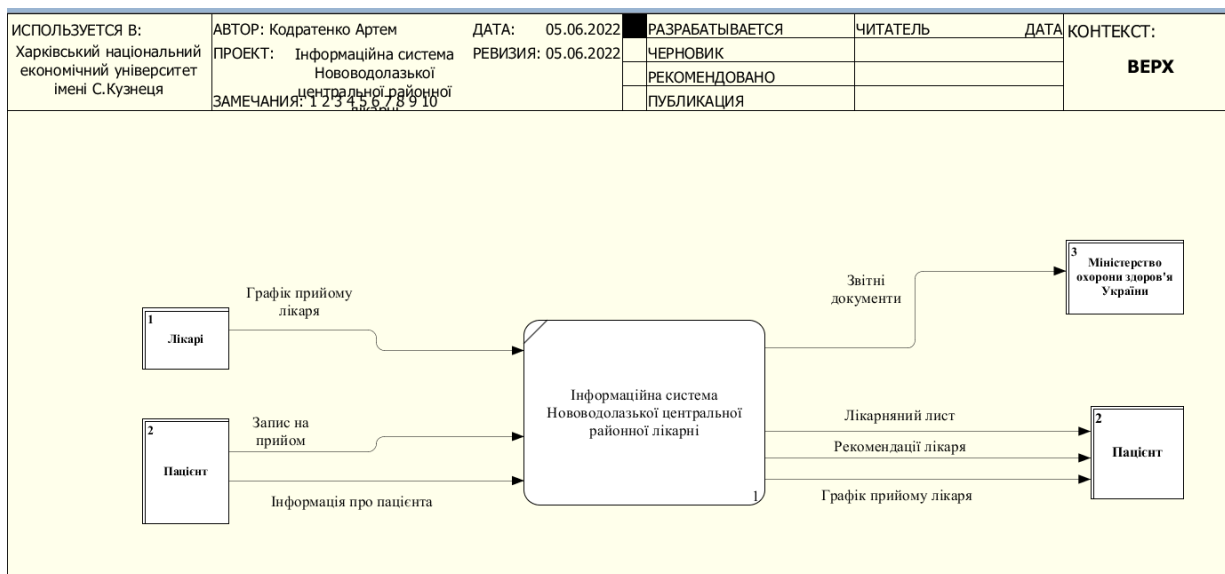


Рисунок 1.8 – Контекстна діаграма діаграми потоків даних

Декомпозиція контекстної діаграми потоків даних наведено на рис.1.9.

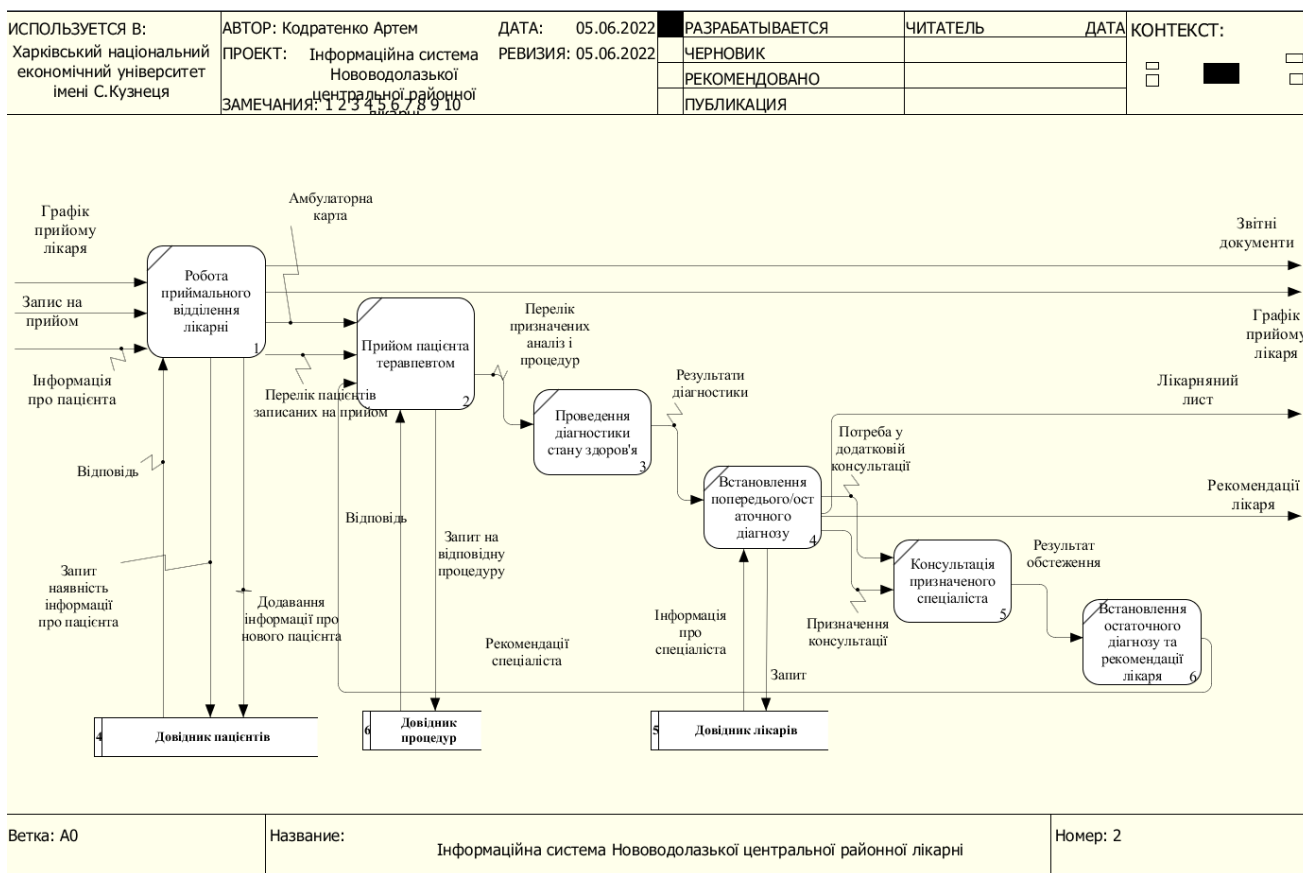


Рисунок 1.9 – Декомпозиція першого рівня

1.6 Постановка задачі дослідження

У державній медицині історично склалося так, що основним способом організації розподілення ресурсів є «жива черга», тобто по суті планування виробничих ресурсів (робочий персонал, кабінети) зводиться до затвердження графіку прийомів лікарів у певних кабінетах, а управління будь-якими іншими аспектами не відбувається. Крім того, час, який виділено на прийом, формувався за допомогою командно-адміністративних методів та не брав до уваги витрати часу пацієнтів та особливості конкретного закладу. Такий спосіб організації знижує видатки медичного закладу на планування, але збільшує нееконімічні видатки пацієнтів (спричиняє очікуванню у черзі у випадку перевантаження лікаря) та не дає можливості управляти тривалістю прийому, часом очікування пацієнтів, аналізувати інформацію по затримкам прийомів, утилізації ресурсів та іншим важливим показникам. Ведення документації, а саме карток пацієнтів відбувається виключно вручну медичним персоналом. Причиною існування такої системи дотепер є відсутність у медичних закладів потреб відслідковувати такі показники (через відсутність таких вимог з боку системи охорони здоров'я), а також відсутність конкуренції між державними медичними закладами.

Певним удосконаленням системи живої черги стала система талонів на прийом, які видавалися із зазначенням на них певного часу, а отже вирішували дві проблеми: неспроможність лікаря прийняти всіх бажаючих пацієнтів (талонів видавалося кожен день обмежена кількість), коли одночасно приходять багато пацієнтів і так чи інакше комусь із пацієнтів доводиться чекати довше за інших (талони видаються із зазначеним на них часом, таким чином більш-менш рівномірно розподіляючи пацієнтів протягом часу прийому). Іншою варіацією талонної системи став запис пацієнта (по прізвищу) у заздалегідь розплановану часову сітку (так зване «простирадло»). Варто сказати, що талонна система перекладала на пацієнта організаційну відповідальність (він повинен був мати із собою талон, як право на прийом), тоді як запис у часову сітку більше навантажував медичний заклад (оскільки

інформація про запис мала передаватися лікарю перед прийомом). Такий різновид запису вже був принципово новим рівнем планування у медичних закладах, але також не знімав проблеми черг, оскільки за талонами до «рідких» лікарів все рівно збиралася жива черга, так само як і на запис до таких лікарів. Також такий спосіб не давав змоги варіювати тривалість прийому, закладаючи стандартний часовий інтервал, а також не давав можливості аналізувати завантаження лікарів.

Закордонні джерела, які були проаналізовані під час підготовки дипломного проекту, описують рішення наступних проблем систем ведення обліку пацієнтів:

1. Пацієнти, що потрапляють у заклад без попереднього запису на прийом (walk-ins). З одного боку, такі пацієнти створюють можливість для медичного закладу надати послугу та, окрім отриманої виручки, задовольнити клієнта, який потім повернеться знов. З іншого боку, якщо тримати в наявності вільні ресурси, які могли б використовуватися для обслуговування інших клієнтів, це може суттєво зменшити ефективність використання ресурсів та фінансовий результат.

Найбільш поширений спосіб забезпечити обслуговування таких пацієнтів – резервування частини інтервалів, з неможливістю планування прийомів, яке знімається за декілька годин до початку зарезервованого часу.

В умовах українського приватного медичного закладу, суттєва кількість таких пацієнтів не характерна, та резервування звичайно не проводиться, бо знижує фінансові результати.

2. Пацієнти, що не приходять на прийом (no-shows). Це дуже суттєва проблема, якщо про факт того, що пацієнт не прийде, становиться відомим вже коли час для прийому минув. Клініка у такому разі витрачає ресурси марно та не має можливості скомпенсувати свої витрати. При одночасній наявності декількох лікарів, які можуть замінювати один іншого (одна спеціальність), відносно невеликої тривалості прийому (до 30 хв.) та суттєвої кількості пацієнтів, які не з'являються, рішенням проблеми може бути надлишковий

запис (overbooking), тобто на визначений інтервал створюється більше записів на прийом, ніж клініка може обслужити (пропорціонально відсотку пацієнтів; що не з'явилися, наприклад, якщо 3 лікаря приймають 9 пацієнтів на годину та відомо, що відсоток пацієнтів, які не з'явилися, становить 15%, клініка може записати на цю годину не 9, а 10 пацієнтів – у середньому обслуговування не буде суттєво страждати, а коефіцієнт утилізації збільшиться.

3. Обробка отриманих чисельних значень за всіма показниками математичного аналізу, для складання статистичних відомостей та встановлення достовірного діагнозу.

В умовах державної медицини України ця проблема звичайно не є надто серйозною, обробка даних у багатьох клініках відбувається за допомогою прикладних програм, таких як Microsoft Excel тощо.

У Нововодолазькій громаді мешкає 35 тисяч осіб, майже кожна людина зареєстрована у Нововодолазькій центральній районній лікарні. Кожна особа, яка має реєстрацію в лікарні підписала декларацію з лікарем. На період 2022 року в Нововодолазькій центральній районній лікарні працюють 46 спеціалістів. Кожен бажаючий може отримати допомогу у лікарів, які мають такі спеціальності: невропатолог, дерматовенеролог, нарколог, лікар з ультразвукової діагностики, кардіолог, терапевт, анестезіолог, хірург тощо.

В умовах стрімкого розвитку коронавірусної інфекції по всьому світу підвищилась увага до сайтів медичних закладів, оскільки згідно рекомендації МОЗ, хворим необхідно проводити як можна менше часу у місцях з масовим перебуванням людей, тому вони намагаються знаходити всю необхідну для них інформацію на сайтах медичних закладів [2].

На сайту медичного закладу повинна бути розміщена наступна інформація: адреса медичного закладу, контактні телефони реєстратури, головного лікаря, графік прийому спеціалістів, перелік відділень, лабораторій та кабінетів (рентгенологічний, функціональної діагностики тощо), посилання для попереднього запису на прийом онлайн та іншу інформацію [2].

Сайт медичного закладу повинен бути зручним та зрозумілим, реалізована можливість перемикання на версію для людей з обмеженими можливостями.

При розробленні сайту велика увага приділяється front-end частини сайту медичного закладу, тому тема дипломного проекту «Розроблення front-end частини сайту інформаційної системи Нововодолазької центральної районної лікарні» є актуальною.

Таким чином, в даному дипломному проекті необхідно розробити сайт Нововодолазької центральної районної лікарні, який повинен виконувати наступні функції:

1. Надати необхідні для користувача функції;
2. Забезпечити зручний інтерфейс користувача;
3. Забезпечити невелику ресурсоємність;
4. Забезпечити роботу в різних режимах.

2 СПЕЦИФІКАЦІЇ ВИМОГ

2.1 Специфікація варіантів використання

Візуальне моделювання в UML можна представити як певний процес по рівневого спуску від найбільш загальної і абстрактної концептуальної моделі вихідної системи до логічної, а потім і до фізичної моделі відповідної програмної системи [13]. Для досягнення цих цілей спочатку будується модель у формі так званої діаграми варіантів використання (use case diagram), яка описує функціональне призначення системи або, іншими словами, те, що система буде робити в процесі свого функціонування. Діаграма варіантів використання є вихідним концептуальним уявленням або концептуальною моделлю системи в процесі її проектування і розробки [14].

Розробка діаграми варіантів використання переслідує наступні цілі [14]:

Визначити загальні межі і контекст модельованої предметної області на початкових етапах проектування системи.

Сформулювати загальні вимоги до функціональної поведінки проектованої системи сайту Нововодолазької центральної районної лікарні.

Розробити вихідну концептуальну модель системи для її подальшої деталізації у формі логічних і фізичних моделей.

Підготувати вихідну документацію для взаємодії розробників системи з її замовниками і користувачами.

Суть даної діаграми полягає в наступному: проектована система сайту Нововодолазької центральної районної лікарні представляється у вигляді безлічі сутностей або акторів, що взаємодіють з системою за допомогою так званих варіантів використання. При цьому актором (actor) або дійовою особою називається будь-яка сутність, що взаємодіє з системою ззовні. Це може бути людина, технічний пристрій, програма або будь-яка інша система, яка може служити джерелом впливу на модельовану систему так, як визначить сам розробник. У свою чергу, варіант використання (use case) служить для опису сервісів, які система надає акторові. Іншими словами, кожен варіант

використання визначає деякий набір дій, який чинять системою при діалозі з актором.

В даному розділі спроектовано діаграму варіантів використання сайту Нововодолазької центральної районної лікарні (рис. 2.1), що розробляється в даному дипломному проекті.

Сайт Нововодолазької центральної районної лікарні є основою для взаємодії пацієнта/персоналу лікарні із лікарнею, тобто пацієнт чи персонал центру ПМД будуть використовувати цей програмний засіб за для автоматизації роботи Нововодолазької центральної районної лікарні, ведення оглядів та бази пацієнтів, запис на прийом, тощо.

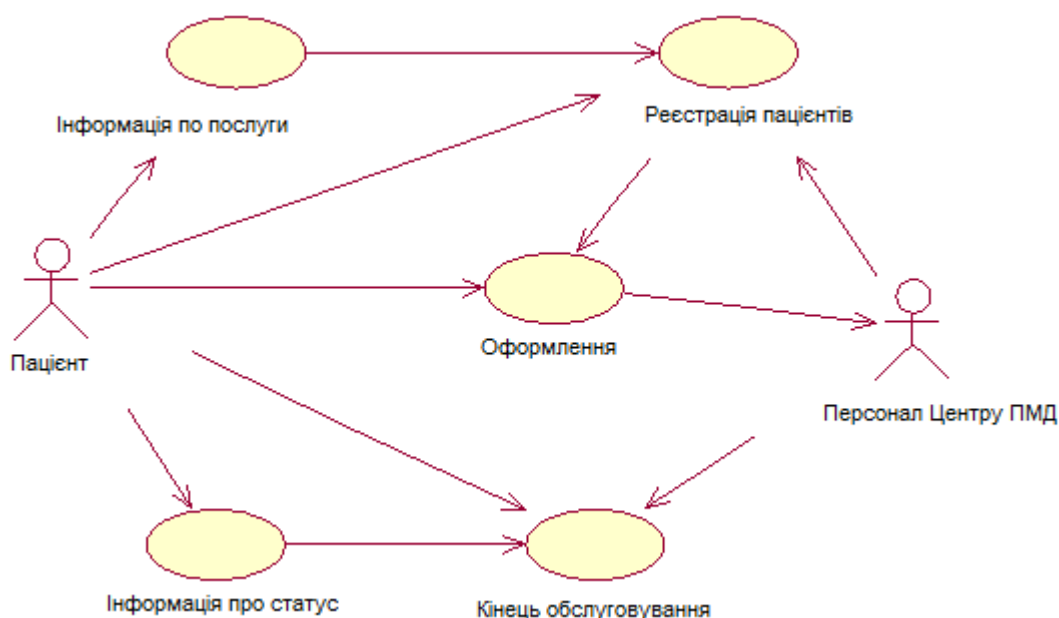


Рисунок 2.1 – Діаграма варіантів використання сайту Нововодолазької центральної районної лікарні

На діаграмі розташовані два актори – «Пацієнт» та «Персонал Центру первинної медичної допомоги». Кожен актор має доступ до окремого функціоналу системи сайту Нововодолазької центральної районної лікарні.

Всі актори можуть запустити систему та авторизуватися в ній.

Актор «Пацієнт» має можливість записатися на прийом до обраного лікаря, переглядати інформаційні послуги, які надає лікарня, обирати дату та час зустрічі, має доступ до особистого кабінету та до інформації щодо своїх записів та коментарів від лікаря.

Персонал Центру ПМД, користувач, який має можливість редагувати розклад прийомів до лікаря, управляти даними про пацієнтів та лікарів, записувати, видаляти, а також змінювати інформацію про статус онлайн запису пацієнтів.

Специфікація варіантів використання

Варіант використання «Інформаційні послуги»

Короткий опис	Пацієнт переглядає та обирає послугу, яку надає лікарня.
Дійові особи	Пацієнт
Предумова	Пацієнт знаходиться на сайті.
Основний потік подій (сценарій)	<ol style="list-style-type: none"> 1. Пацієнт заходить в розділ «Про лікарню» 2. Пацієнт обирає пункт «Більше» 3. На сторінці зображуються послуги, які надає лікарня, та оглядає перелік лікарів за спеціальністю.
Альтернативний потік подій	–
Постумова	Успішне завершення

Варіант використання «Реєстрація пацієнтів»

Короткий опис	<p>Пацієнт реєструється та отримує обліковий запис, щоб мати можливість записатися на прийом до лікаря.</p> <p>Персонал центру ПМД може зареєструвати пацієнта з власного особистого кабінету.</p>
Дійові особи	Пацієнт, Персонал центру ПМД.
Предумова	Дійова особа знаходиться на сайті.
Основний потік подій (сценарій)	<ol style="list-style-type: none"> 1.1 Дійова особа обирає розділ «Головна сторінка» 1.2 Обирає пункт «Пацієнт»

	1.3 Вводить свої прізвище, ім'я, номер телефону, електронну пошту та створює пароль. 1.4 Дані надходять до БД. 1.5 Дійова особа отримує обліковий запис.
Альтернативний потік подій	Пацієнт зареєстрований в системі та може записуватися на прийом.
Постумова	Успішне завершення

Варіант використання «Оформлення»

Короткий опис	1. Пацієнт обирає потрібного лікаря та записується на прийом. 2. Пацієнт переглядає свої записи, історію призначення та рецепти. 3. Персонал центру ПМД переглядає записаних пацієнтів та редагує інформацію про записи.
Дійові особи	Пацієнт, персонал центру ПМД.
Предумова	Дійова особа авторизована в системі.
Основний потік подій (сценарій)	1.1 Пацієнт обирає розділ «Головна сторінка» 2.1 Пацієнт обирає пункт «Панель управління» 2.2 В пункті «Панель управління» пацієнт може зробити запит, щоб Працівник центру ПМД зареєстрував його через свій обліковий запис. 2.3 В пункті «Панель управління» пацієнт може переглянути «Мої зустрічі». 2.4 В пункті «Панель управління» пацієнт має можливість переглянути «Рецепти». 3.1 Пацієнт обирає пункт «Запис на прийом» 3.2 Пацієнт обирає спеціалізацію, лікаря, дату призначення, час призначення. 3.3 Пацієнт натискає «Створити новий запис». 4.1 Пацієнт обирає пункт «Історія призначення». 4.2 Пацієнт має можливість переглянути інформацію про свій запис. 5.1 Пацієнт обирає пункт «Рецепти». 5.2 Пацієнт має можливість подивитися свою історію рецептів.
Альтернативний потік подій	—
Постумова	Успішне завершення

Варіант використання «Інформація про статус»

Короткий опис	Пацієнт записується на прийом до лікаря та отримує результати, щодо цього запису.
Дійові особи	Пацієнт
Предумова	Пацієнт авторизований в системі
Основний потік подій (сценарій)	<ol style="list-style-type: none"> 1. Пацієнт заходить до свого особового запису. 2. Пацієнт перевіряє інформацію про запис. 3. У пацієнта можуть бути статуси «Зареєстрований на запис», «Запис неможливий», «Термін запису закінчений», «Запис завершений».
Альтернативний потік подій	–
Постумова	Успішне завершення

Варіант використання «Кінець обслуговування»

Короткий опис	Пацієнт отримує інформацію про завершення обслуговування. Персонал центру ПМД вносить інформацію про кінець процедури.
Дійові особи	Пацієнт, персонал центру ПМД
Предумова	Дійова особа авторизована в системі
Основний потік подій (сценарій)	<ol style="list-style-type: none"> 1. Дата та час запису закінчуються в БД. 2. Персонал центру ПМД вносить інформацію про рецепти видані пацієнту, про завершення періоду запису.
Альтернативний потік подій	–
Постумова	Успішно завершена процедура.

2.2 Розроблення алгоритму та дерева цілей розроблення веб-сайту медичного закладу

Для вирішення поставлених завдань було розроблено схему алгоритму [10], показану на рис. 2.2.

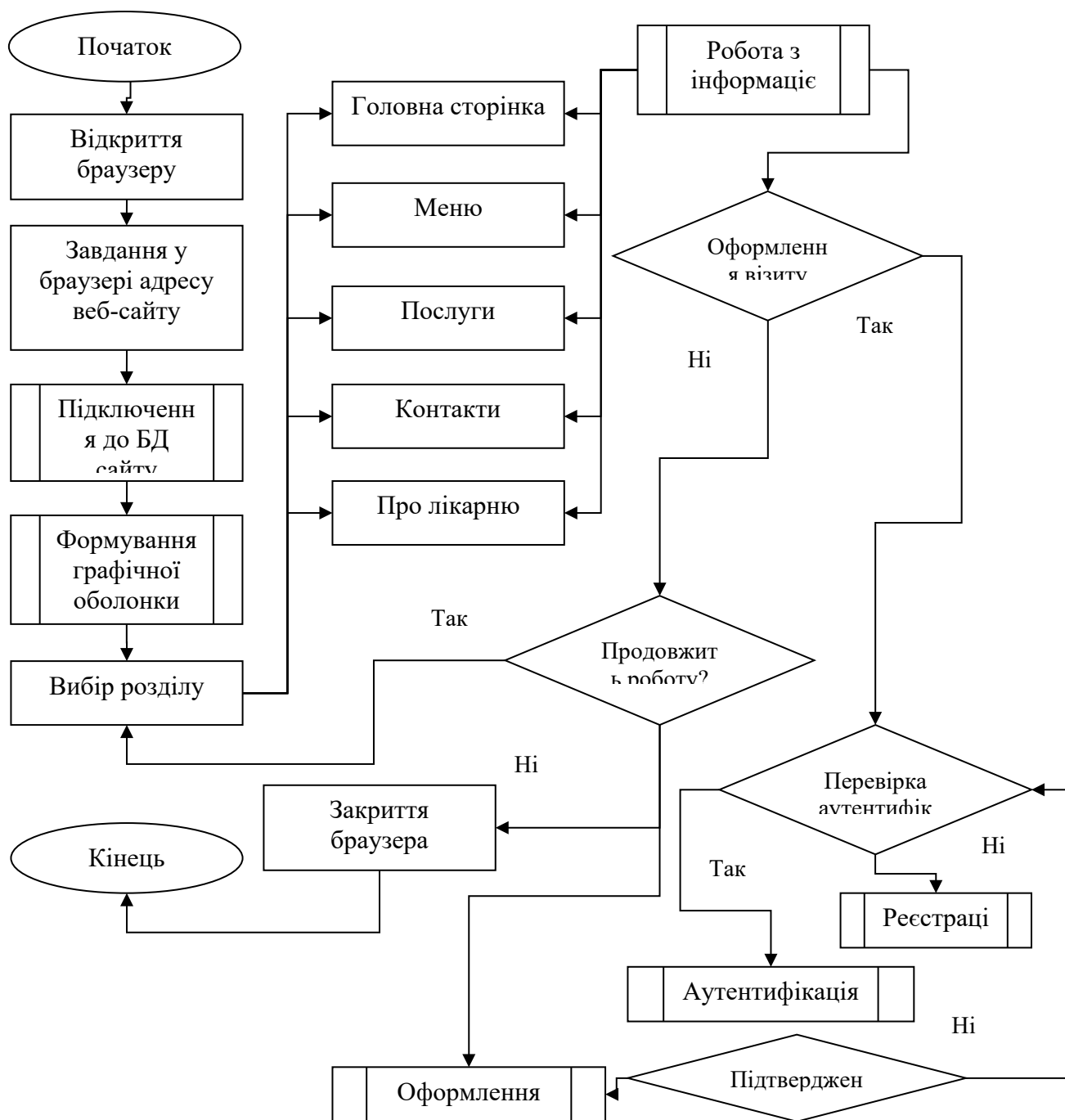


Рисунок 2.2 – Алгоритм роботи веб-сайту Нововодолазької центральної районної лікарні

Метою розробки програмного продукту є реалізація процесу автоматизації сайту Нововодолазької центральної районної лікарні. На виході роботи програмного продукту користувач отримує дієвий веб-сайт Нововодолазької центральної районної лікарні.

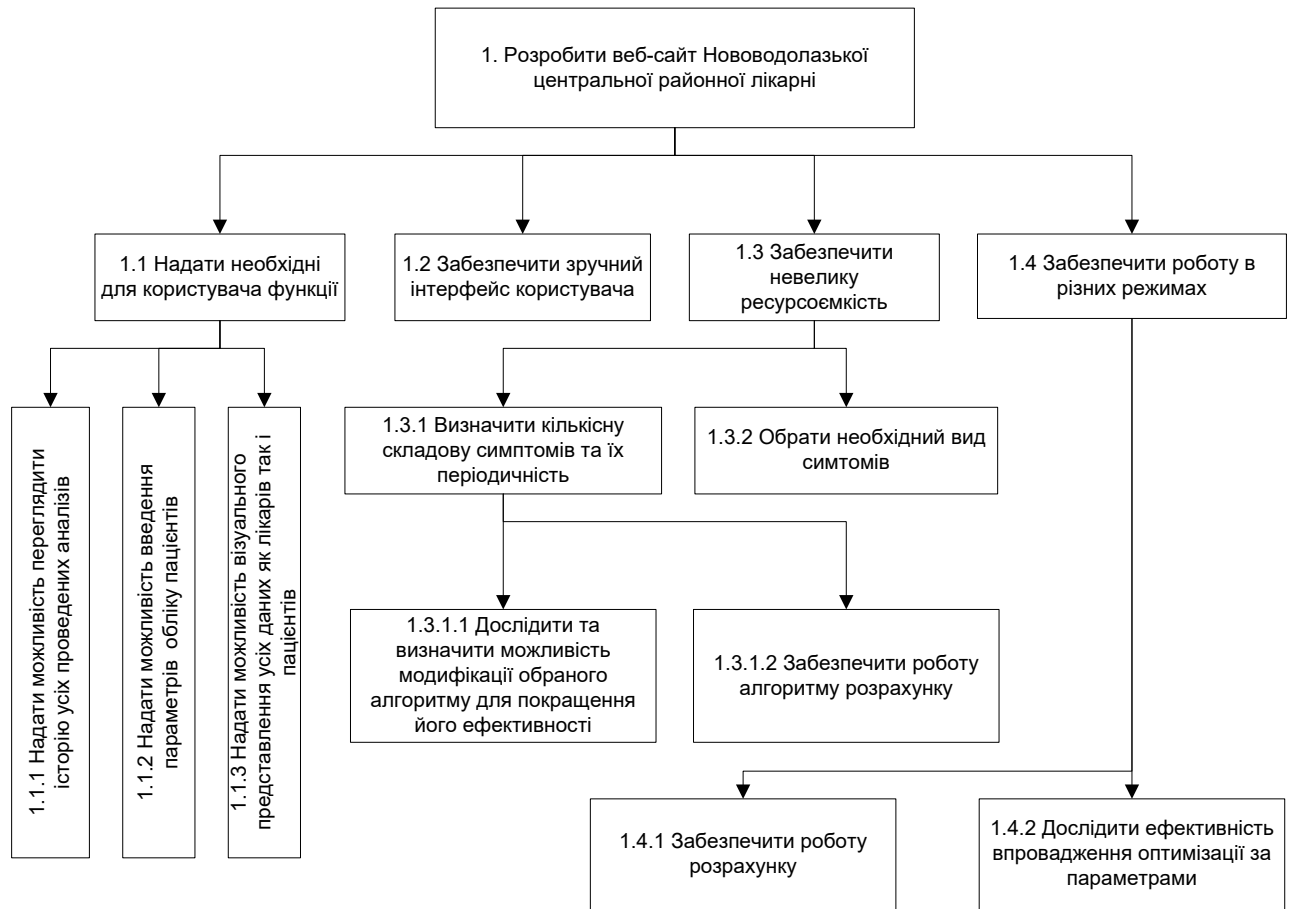


Рисунок 2.3 – Декомпозиція цілей цілей розробки сайту
Нововодолазької центральної районної лікарні

По результатах проведеного аналізу у розділі 1 та беручи до уваги вищезазначені функції, було розроблено дерево цілей розробки сайту Нововодолазької центральної районної лікарні, що зображене на рис. 2.3.

Програмою передбачено:

- формування вихідних даних на розрахунок;
- визначення попереднього діагнозу;
- формування результатів.

Загальна структура сайту Нововодолазької центральної районної лікарні зображена на рис. 2.4 та складається з дев'яти модулів, на які вона поділяється за функціональними ознаками:

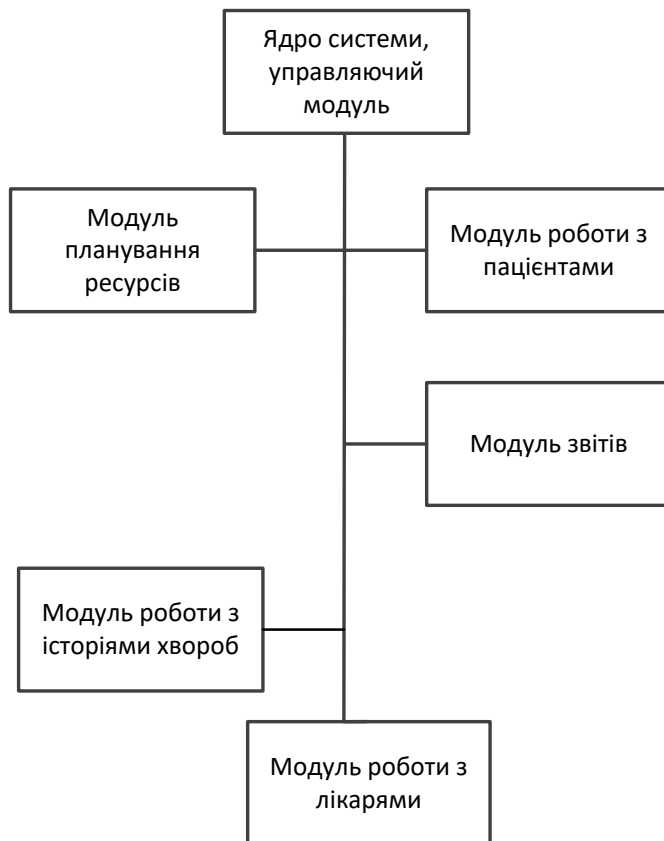


Рисунок 2.4 – Структурна схема сайту Нововодолазької центральної районної лікарні

Управляючий модуль – складає ядро системи та забезпечує взаємодію всіх її частин:

- 1) Модуль планування ресурсів – забезпечує управління ресурсними потребами послуг, а також планування та перепланування медичних прийомів
- 2) Модуль роботи з пацієнтами – забезпечує роботу з паспортними картками пацієнтів
- 3) Модуль звітів – відповідає за роботу з управлінськими, оперативними, фінансовими, маркетинговими так іншими звітами.
- 4) Модуль роботи з історіями хвороб – дозволяє вести електронну медичну документацію
- 5) Модуль роботи з лікарями – забезпечує роботу лікарів.

Задачі програмного модуля управління виробничими ресурсами є наступними:

Забезпечення процесу зручного планування та перепланування медичних прийомів пацієнтів. Для вирішення цієї задачі розроблено інтерфейс користувача, який прозоро та інтуїтивно дозволяє вести облік пацієнтів.

2.3 Вибір мови програмування

Ключові веб-технології, що використовувались в реалізації задачі дипломного проекту: HTML, CSS, PHP, MySQL та JavaScript.

Мова гіпертекстової розмітки (HTML). Використовується для опису структури веб-сторінки. HTML дозволяє вказати, де на веб-сайті розміщуються текст, зображення та інші форми вмісту.

Каскадні таблиці стилів або CSS описують, як елементи мають відображатися на веб-сторінці. CSS дозволяє визначати такі стилі, як колір елементів на сторінці, їх розташування та спосіб відображення тексту на веб-сайті.

JavaScript. Ця мова сценаріїв гарантує більш динамічний веб-сайт. Будь-яка інтерактивна веб-сторінка, яку бачить користувач, ймовірно, певною мірою працює на JavaScript. Наприклад, контент, який оновлюється автоматично, та веб-сайти з картами використовують JavaScript для візуалізації контенту у зовнішньому інтерфейсі.

Бібліотека JavaScript, jQuery

Бібліотека JavaScript – це розширення мови JavaScript. Всі ці бібліотеки мають певний набір функцій, які мають допомогти розробникам створювати веб-сайти більш ефективно.

Візьмемо для прикладу jQuery. JQuery було розроблено, щоб спростити реалізацію JavaScript на веб-сайтах. JQuery абстрагує багато функцій, які можуть зустрітися в JavaScript, на користь більш простого синтаксису.

Використання jQuery дозволяє ефективно виконувати різноманітні завдання. Існує просте керування CSS на веб-сторінці та зміна елементів

HTML. Також є додавання ефектів та анімації на сайт і потокова передача даних на веб-сторінку з використанням AJAX.

Системи контролю версій (VCS) використовуються для керування змінами у програмному проєкті. Приклади цих систем включають Git та Mercurial.

VCS є важливими, тому що вони дозволяють розробникам бачити, як розвивається проєкт. Вони відстежують кожну зміну, внесену до кожного файлу в проєкті. Поряд із кожним записом про зміну файлу міститься інформація про те, хто і коли його змінив.

Наявність цього запису означає, що легко побачити, як проєкт з'явився у певний момент історії. Це також спрощує повернення проєкту до попередньої версії, якщо розробник припустився помилки.

Фреймворк переднього плану – це каркас, який постачається із попередньо написаним кодом, за допомогою якого є можливість створювати програму. До поширених фреймворків JavaScript відносяться React та Next.js.

Наприклад, React спрощує створення інтерактивної веб-програми. Він має функції, які дають змогу відображати потрібні частини веб-сторінки при зміні даних на веб-сайті. React також дозволяє розділити проєкт на компоненти, щоб користувач мав змогу зменшити повторення в кодовій базі.

Використання React може скоротити час роботи над проєктом, тому що він готовий використовуватись прямо з коробки. Він також надає ряд функцій, які можна використовувати в процесі розробки програми.

Використання мови PHP при web-розробці дозволяє створити простий, ефективний, багатофункціональний, безпечний і найголовніше динамічний сайт.

Вся текстова інформація, надана на сайті, легко керована, оновлювана та видалена. Для забезпечення даних умов контент зберігається в базі даних, в даному випадку використовується база даних MySQL.

Дизайн сайту легко реалізується за допомогою каскадних таблиць стилів – CSS, а поділ сторінок сайту на блоки у багато разів спрощує і економить час на його зміну.

3 ПРОЕКТНІ РІШЕННЯ РОЗРОБЛЕННЯ FRONT-END ЧАСТИНИ САЙТУ НОВОВОДОЛАЗЬКОЇ ЦЕНТРАЛЬНОЇ РАЙОННОЇ ЛІКАРНІ

3.1 Проектування структури бази даних

3.1.1 Концептуальне інфологічне проектування

В підрозділі 3.1.1 виконано побудову моделі даних системи, незалежної від СУБД, яка охоплює створення словника даних, що відображено у таблиці 3.1.

Таблиця 3.1

Словник даних

№ п/п	Найменування елемента	Ідентифікатор	Тип довжина	Призначення елемента
1	2	3	4	5
1	Код відділення	idВідділення	INTEGER	Зберігає ID відділення. Фактичний
2	Назва відділення	Найменування	TEXT	Зберігає назву відділення. Фактичний
3	Кількість місць у відділенні	Кількість місць	INTEGER	Зберігає кількість місць. Фактичний
4	Завідувач відділення	Зав відділенням	TEXT	Зберігає ПІБ завідувача відділенням. Фактичний
5	Код категорії діагнозу	idКатегорія діагнозу	INTEGER	Зберігає код категорії діагнозу. Фактичний
6	Найменування категорії	Найменування категорії	TEXT	Зберігає найменування категорії. Фактичний
7	Код діагнозу	idДіагноз	INTEGER	Зберігає код діагнозу. Фактичний
8	Найменування діагнозу	Найменування діагнозу	TEXT	Зберігає найменування діагнозу. Фактичний
9	Код пацієнта	idПацієнт	INTEGER	Зберігає код пацієнта. Фактичний
10	Прізвище пацієнта	Прізвище	TEXT	Зберігає прізвище пацієнта. Фактичний
11	Ім'я пацієнта	Ім'я	TEXT	Зберігає ім'я пацієнта. Фактичний
12	По батькові пацієнта	По батькові	TEXT	Зберігає по батькові пацієнта. Фактичний
13	Дата народження пацієнта	Дата народження	DATE	Зберігає дату народження пацієнта. Фактичний
14	Адреса проживання пацієнта	Адреса	TEXT	Зберігає адресу проживання пацієнта. Фактичний

15	Контактний телефон пацієнта	Контактний телефон	CHAR (15)	Контактний телефон пацієнта. Фактичний
16	Контактна особа	Контактна особа	TEXT	Зберігає ПІБ контактної особи, з якою можна зв'язатися. Фактичний
17	Паспортні дані пацієнта	Паспортні дані	TEXT	Зберігає паспортні дані пацієнта. Фактичний
18	Ідентифікаційний номер пацієнта	Ідентифікаційний номер	INTEGER	Зберігає ідентифікаційний номер пацієнта. Фактичний
19	Код лікаря	idЛікар	INTEGER	Зберігає код лікаря. Фактичний
20	Прізвище лікаря	Прізвище	TEXT	Зберігає прізвище лікаря. Фактичний
21	Ім'я лікаря	Ім'я	TEXT	Зберігає ім'я лікаря. Фактичний
22	По батькові лікаря	По батькові	TEXT	Зберігає по батькові лікаря. Фактичний
23	Контактний телефон лікаря	Контактний телефон	CHAR (15)	Зберігає контактний телефон лікаря. Фактичний
24	Стаж роботи лікаря	Стаж роботи	FLOAT	Зберігає стаж роботи лікаря. Фактичний
25	Код обліку пацієнтів	idОблік пацієнтів	INTEGER	Зберігає код обліку пацієнтів. Фактичний
26	Дата надходження пацієнта	Дата надходження пацієнта	DATE	Зберігає дату надходження пацієнта. Фактичний
27	Дата виписки пацієнта	Дата виписки	DATE	Зберігає дату виписки пацієнта. Фактичний
28	Код графіку прийому лікаря	idГрафік прийому лікаря	INTEGER	Зберігає код графіку прийому лікаря. Фактичний
29	День тижня	День тижня	TEXT	Зберігає день тижня прийому. Фактичний
30	Час початку прийому	Час початку прийому	TIME	Зберігає час початку прийому. Фактичний
31	Час завершення прийому	Час завершення прийому	TIME	Зберігає час завершення прийому. Фактичний
32	Код спеціалізації	idСпеціалізації	INTEGER	Зберігає код спеціалізації. Фактичний
33	Найменування спеціалізації	Найменування	TEXT	Зберігає найменування спеціалізації. Фактичний

Обмеження атрибутів сутностей БД інформаційної системи Нововодолазької центральної районної лікарні, що були спроектовані у процесі концептуального інфологічного проектування наведені у таблиці 3.2.

Таблиця 3.2

Обмеження атрибутів сутностей

№ п/п	Ім'я атрибуту або агрегату	Межі / допустимі значення	Умова	Структура/Формат	Значення за замовчуванням
1	2	3	4	5	6
1	Код відділення	Цілі числа від 0 до 10000	Унікальне, NOT NULL	–	–
2	Назва відділення	–	NOT NULL	–	–
3	Кількість місць у відділенні	Цілі числа від 0 до 10000	NOT NULL	–	–
4	Завідувач відділення	–	NOT NULL	–	–
5	Код категорії діагнозу	Цілі числа від 0 до 10000	Унікальне, NOT NULL	–	–
6	Найменування категорії	–	NOT NULL	–	–
7	Код діагнозу	Цілі числа від 0 до 10000	Унікальне, NOT NULL	–	–
8	Найменування діагнозу	–	NOT NULL	–	–
9	Код пацієнта	Цілі числа від 0 до 10000	Унікальне, NOT NULL	–	–
10	Прізвище пацієнта	–	NOT NULL	–	–
11	Ім'я пацієнта	–	NOT NULL	–	–
12	По батькові пацієнта	–	NOT NULL	–	–
13	Дата народження пацієнта	–	NOT NULL	ДД.ММ.РРРР	–
14	Адреса проживання пацієнта	–	–	–	Пацієнт може бути безхатним
15	Контактний телефон пацієнта	Цифри від 0 до 9 включно	NOT NULL	+380(XX)XX XXXXX	–
16	Контактна особа	–	–	–	Особа може не бути
17	Паспортні дані пацієнта	–	NOT NULL	–	–
18	Ідентифікаційний номер пацієнта	–	NOT NULL	–	–

19	Код лікаря	Цілі числа від 0 до 10000	Унікальне, NOT NULL	–	–
20	Прізвище лікаря	–	NOT NULL	–	–
21	Ім'я лікаря	–	NOT NULL	–	–
22	По батькові лікаря	–	NOT NULL	–	–
23	Контактний телефон лікаря	Цифри від 0 до 9 включно	NOT NULL	+380(XX)XX XXXXX	–
24	Стаж роботи лікаря	Невід'ємні числа	–	<ціла частина>.XX	Стаж може бути відсутнім
25	Код обліку пацієнтів	Цілі числа від 0 до 10000	Унікальне, NOT NULL	–	–
26	Дата надходження пацієнта	–	NOT NULL	ДД.ММ.РРРР	–
27	Дата виписки пацієнта	–	NOT NULL	ДД.ММ.РРРР	–
28	Код графіку прийому лікаря	Цілі числа від 0 до 10000	Унікальне, NOT NULL	–	–
29	День тижня	–	NOT NULL	ДД.ММ.РРРР	–
30	Час початку прийому	–	NOT NULL	ГГ-ХВ	–
31	Час завершення прийому	–	NOT NULL	ГГ-ХВ	–
32	Код спеціалізації	Цілі числа від 0 до 10000	Унікальне, NOT NULL	–	–
33	Найменування спеціалізації	–	NOT NULL	–	–

3.2 Проектування логічної моделі даних

У підрозділі 3.2. спроектовано логічну модель даних у вигляді ERD (в нотації IDEF1X). Логічна модель бази даних проектувалась за допомогою CASE-засобу DBDesigner Fork.

На рис. 3.1 наведено графічне подання логічної моделі бази даних. Логічна модель даних – спосіб представлення даних, відповідно до якого дані відповідають своєму відображенню у реальному світі, і можуть називатися так, як вони називаються у реальному світі.

У логічній моделі даних виділяють основні об'єкти БД та визначають зв'язки між цими об'єктами. Створена логічна модель даних не прив'язана до конкретної СУБД. При побудові логічної моделі даних було використано тип зв'язку «один до багатьох».

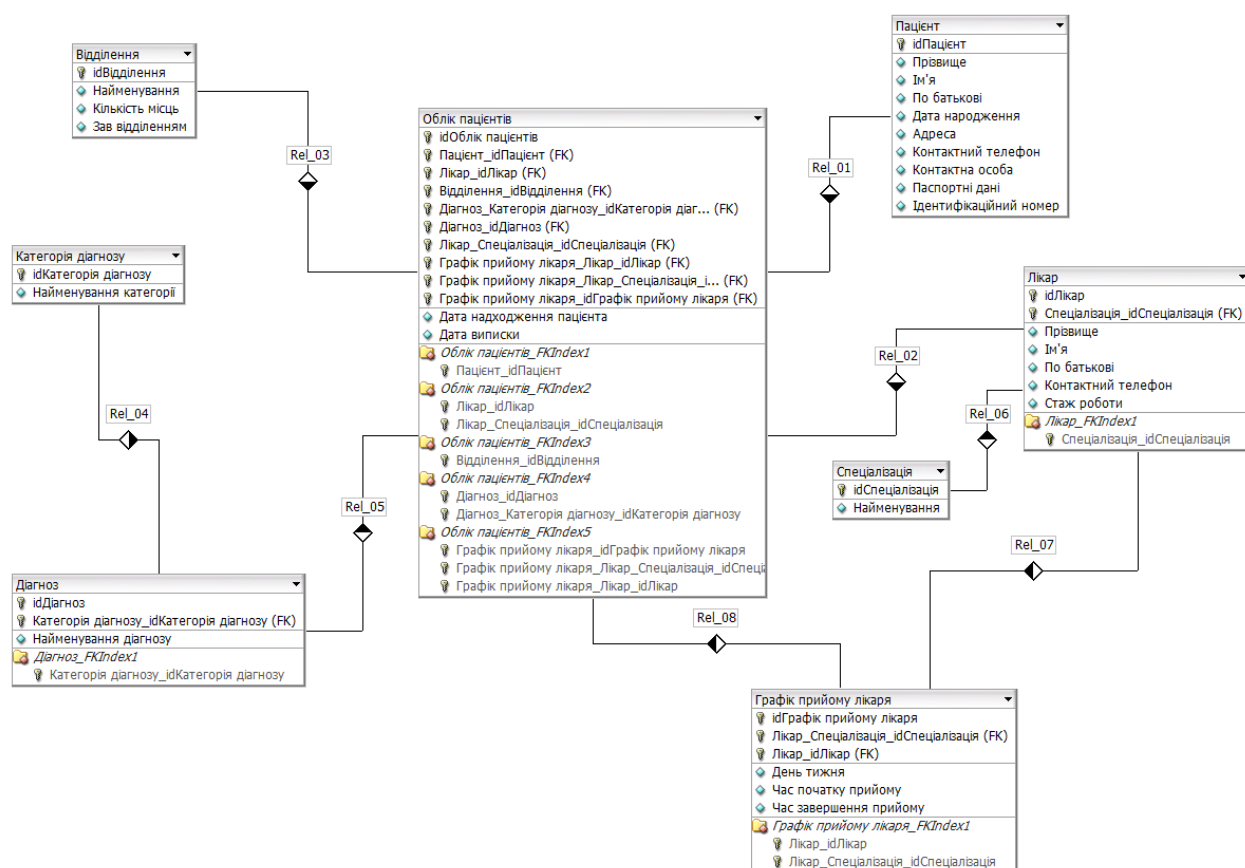


Рисунок 3.1 – Логічна модель даних ІС Нововодолазької центральної районної лікарні

В розробленій логічній моделі даних ІС Нововодолазької центральної районної лікарні представлені такі сутності: «Пацієнт», «Лікар», «Спеціалізація», «Діагноз», «Категорія діагнозу», «Відділення», «Графік прийому лікаря» та «Облік пацієнтів». Всі сутності мають первинні ключі, частина сутностей, таких як «Лікар», «Діагноз», «Графік прийому лікаря» та «Облік пацієнтів» мають зовнішні ключі.

Найменування елементів та їх призначення створено відповідно до концептуального інфологічного проектування, наведеного у пункті 3.1.1.

3.3 Проектування фізичної моделі даних

На основі створеної логічної моделі даних ІС Нововодолазької центральної районної лікарні було побудовано фізичну модель даних, що визначає, яким чином представляються дані та містить усі деталі, необхідні СУБД для створення бази даних. Фізична модель даних містить типи даних та довжину всіх атрибутів сутностей та первинні ключі кожної таблиці, а також показує взаємозв'язок між таблицями за допомогою зовнішніх ключів.

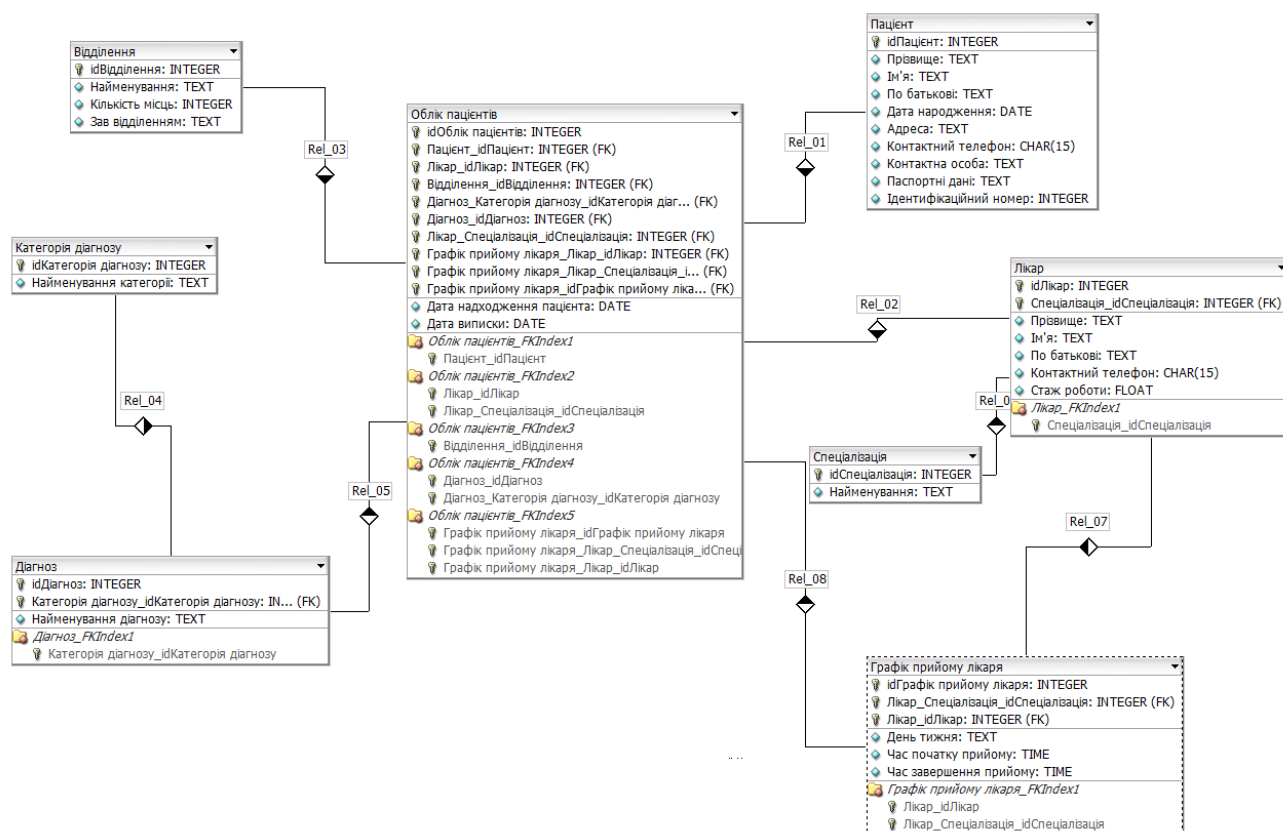


Рисунок 3.2 – Фізична модель даних ІС Нововодолазької центральної районної лікарні

На рис. 3.2. наведена побудована фізична модель бази даних ІС Нововодолазької центральної районної лікарні. Тип і довжина елементів відповідають елементам словника даних, створеного у ході концептуального інфологічного проектування, наведеного у пункті 3.1.1.

4. ТЕХНІЧНІ РІШЕННЯ РОЗРОБЛЕННЯ FRONT-END ЧАСТИНИ САЙТУ НОВОВОДОЛАЗЬКОЇ ЦЕНТРАЛЬНОЇ РАЙОННОЇ ЛІКАРНІ

У якості об'єкту дослідження в дипломному проекті обрано Нововодолазьку центральну районну лікарню, яка розташована за адресою Україна, 63202, Харківська обл., Харківський р-н, селище міського типу Нова Водолага, вул.Пушкіна, будинок 16.

Як показав аналіз проведений у розділі 1, Нововодолазька центральна районна лікарня – це лікувально-профілактичний заклад, який забезпечує кваліфіковану та спеціалізовану поліклінічну та стаціонарну (у тому числі консультативну) допомогу населенню адміністративного району.

Нововодолазька центральна районна лікарня є сучасним багатопрофільним медичним закладом, який надає амбулаторно-поліклінічну, консультативну, діагностичну, стаціонарну та лікувально-профілактичну допомогу як дорослому, так і дитячому населенню Харківського району.

Тому розроблення сайту Нововодолазької центральної районної лікарні, як інструмента взаємодії пацієнтів лікарні із персоналом, є актуальною задачею.

4.1 Проектування front-end частини сайту Нововодолазької центральної районної лікарні

Архітектура ідентифікує основні компоненти системи та методи їх організації та взаємодії. Також важливою характеристикою будь-якої архітектури є вибір таких методів, які будуть незмінними у майбутньому і становитимуть базис системи.

Архітектура – це організація системи, втілена в її компонентах, їх відносинах між собою та з оточенням.

Структуру Frontend, як і в принципі структуру майже будь-якого ПЗ, умовно можна розділити на певну кількість великих блоків. Кожен із цих блоків має якусь свою реалізацію, яка дозволяє зв'язуватися з іншими блоками.

Усередині цих блоків можна назвати модулі, у яких є методи. І кожен вузол цього ланцюжка вкладеності має якийсь свій контракт спілкування, так звані вхідні та вихідні параметри. Можна сказати, що кожна ланка має свою архітектуру. І тому архітектура, в принципі, має властивість ієрархічності.

Сучасна архітектура Frontend базується на трьох принципах:

1. Потік даних займає центральне місце у архітектурі
2. Компонентний підхід
 - 1) Особлива увага потоків даних.
 - 2) Постійно ускладнюючі View (Уявлення).
 - 3) Структура компонентів. Будь-яка компонента повинна складатися з 4 елементів: Графічна структура (HTML-View), Стилізація (CSS-View), Поведінка (JavaScriptComponent), бізнес-логіка (JavaScript-Component/Model).

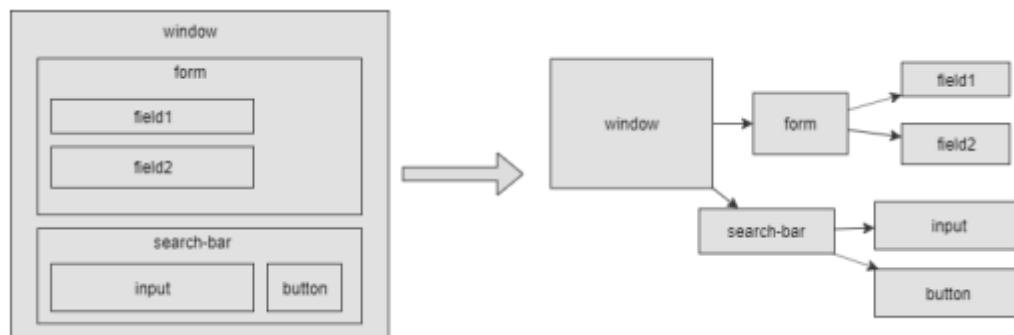


Рисунок 4.1 – Декомпозиції View

3. Автоматичне вибіркве оновлення DOM

- 1) Data-binding між Model та View.
- 2) Оповіщення про зміну даних (Change detection).
- 3) Оновлення DOM.

Сучасні принципи побудови архітектури Frontend, як і в принципі будь-якої іншої галузі розробки ПЗ, ґрунтуються головним чином на компонентному та модульному підході, який дозволяє легко та швидко масштабувати будь-який продукт, займаючись здебільшого не проблемами реалізації та впровадження у поточну архітектуру, а бізнес-вимогами до ПЗ.

Також варто зазначити, що важливим критерієм є швидкість ефективності того чи іншого рішення.

4.2 Тестування розробленого програмного додатку

Головна сторінка сайту Нововодолазької центральної районної лікарні наведена на рис. 4.2.

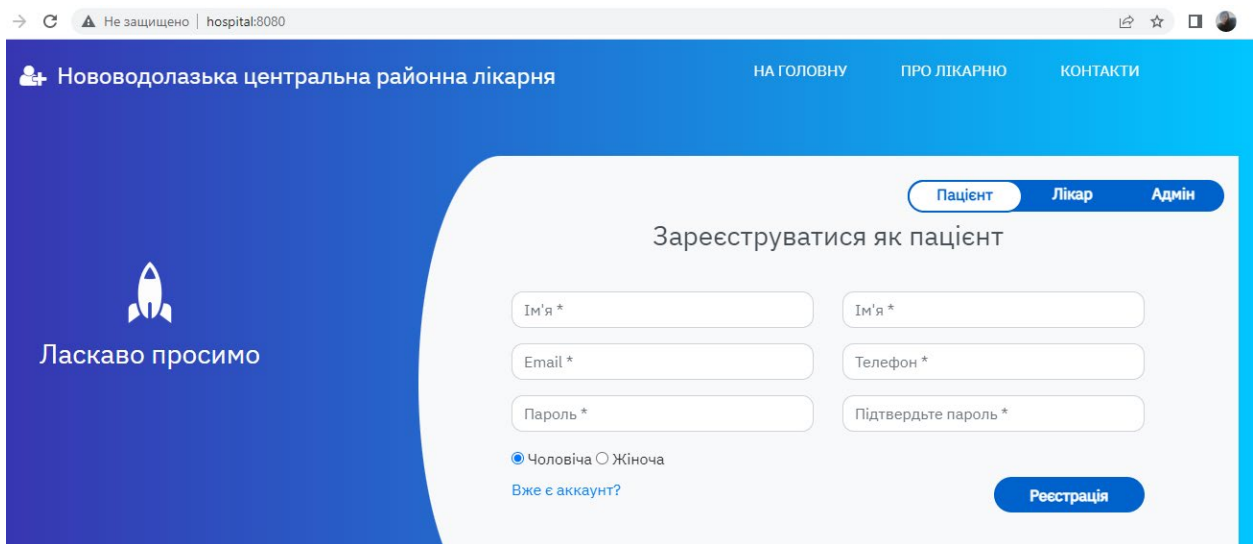


Рисунок 4.2 – Головна сторінка веб-сайту Нововодолазької центральної районної лікарні

Головна сторінка складається з певних блоків кожен з яких відповідає за функціональну складову сайту, так у верхній частині знаходиться навігація яка є провідником по головним сторінкам:

- на головну (рис. 4.2);
- про лікарню (рис. 4.3);
- контакти (рис. 4.4).

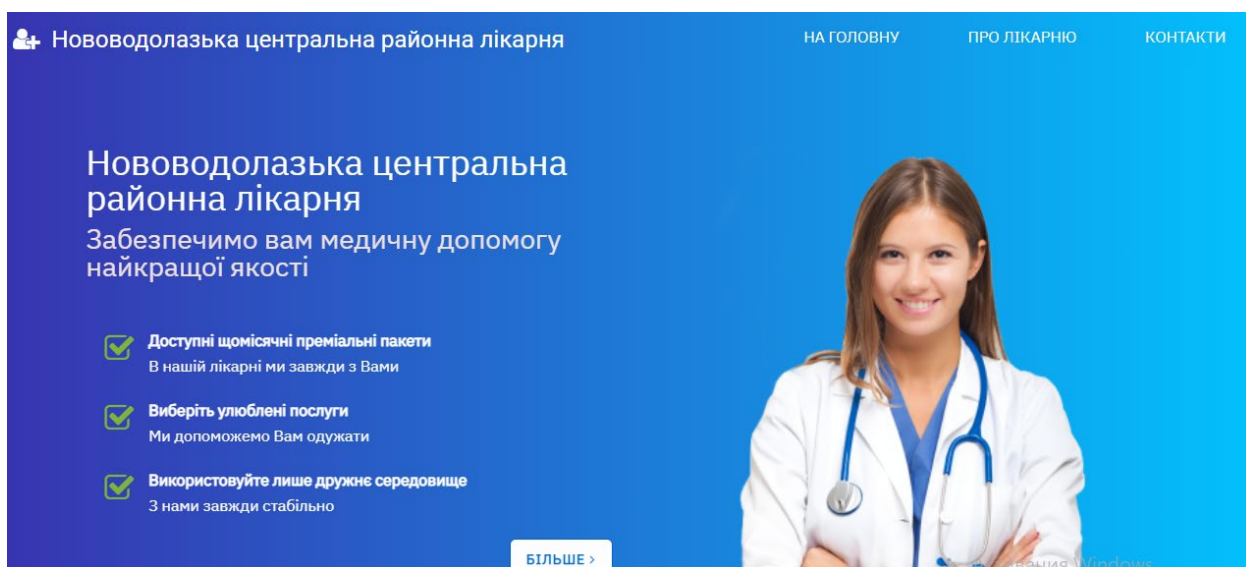


Рисунок 4.3 – Сторінка «Про лікарню» веб-сайту Нововодолазької центральної районної лікарні

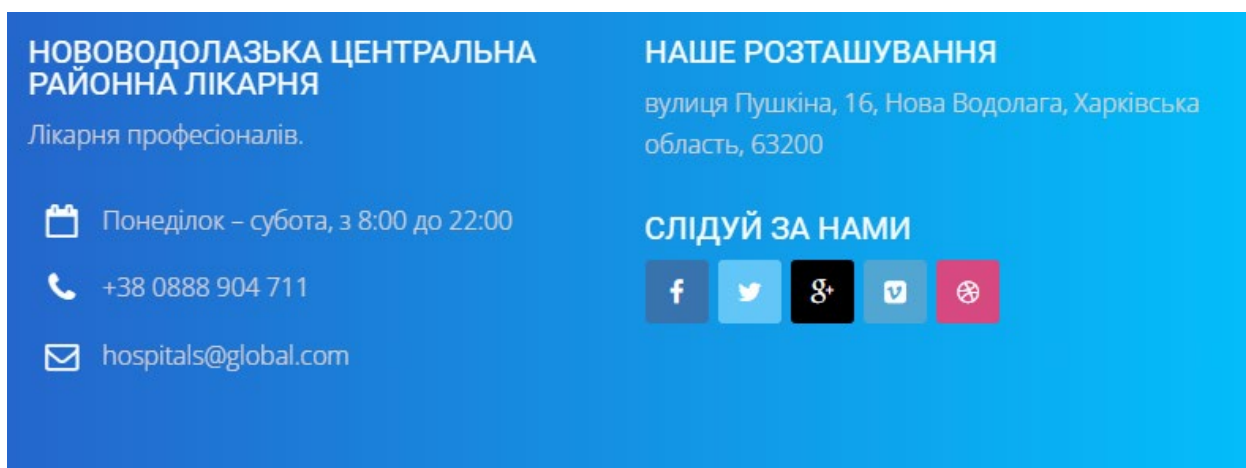


Рисунок 4.4. – Сторінка «Контакти» веб-сайту Нововодолазької центральної районної лікарні

Також на головній сторінці є можливість ознайомитись з головними послугами, що надає лікарня, та переглянути перелік лікарів за кваліфікацією роботи (рис. 4.5).

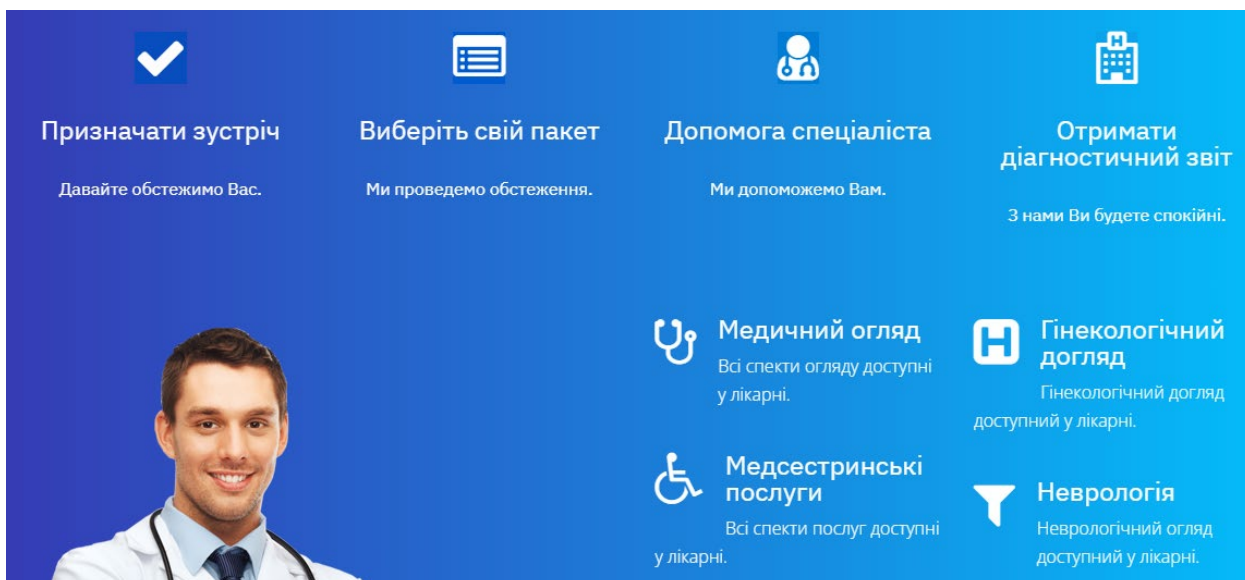


Рисунок 4.5 – Головна сторінка веб-сайту Нововодолазької центральної районної лікарні

На головній сторінці є можливість зареєструватися (якщо немає акаунту) та пройти авторизацію, якщо користувач вже зареєстрований.

У випадку авторизації як адміністратора (рис. 4.6)

- логін – admin
- пароль – admin123

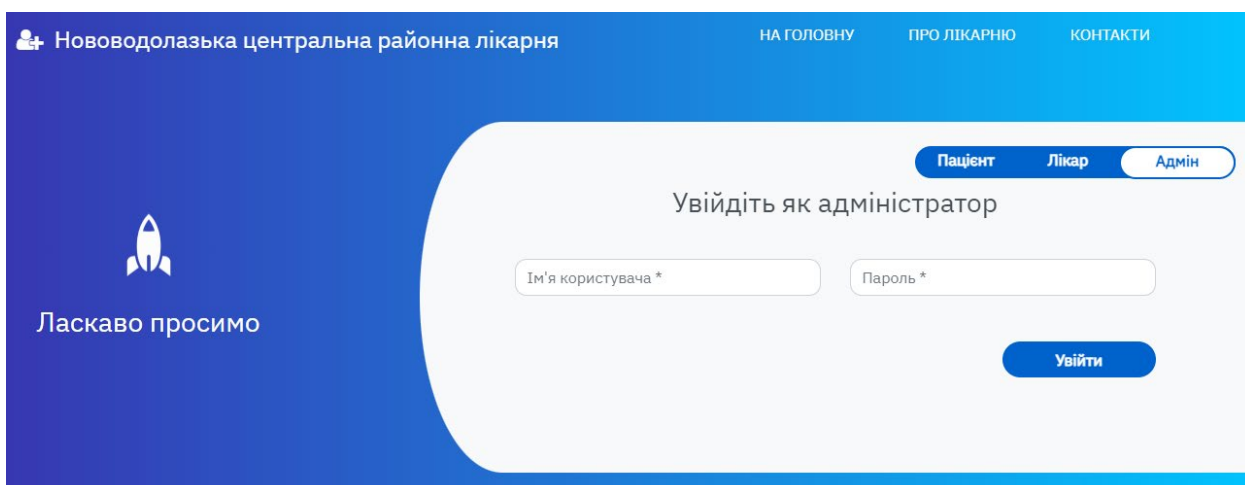


Рисунок 4.6 – Сторінка реєстрації адміну веб-сайту Нововодолазької центральної районної лікарні

Користувач після успішного введення логіну та пароля потрапляє на сторінку адміністратора та у рамках своїх повноважень має змогу переглядати всю наявну інформацію на сайті, додавати та видаляти лікарів, вносити корективи (рис. 4.7).

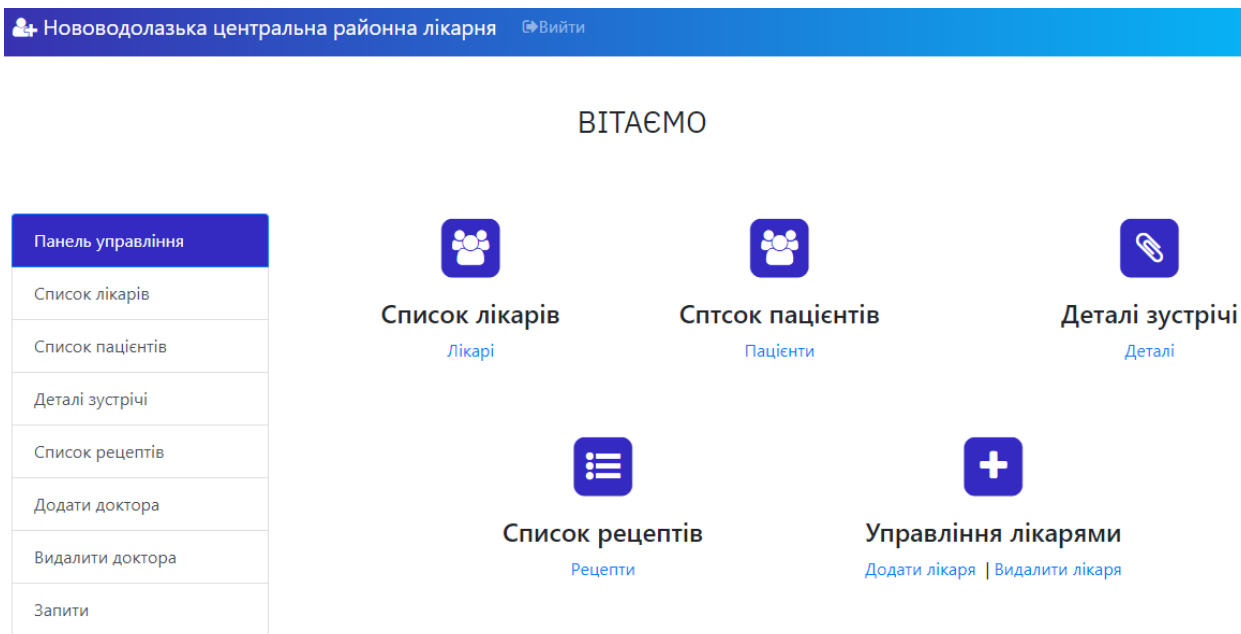


Рисунок 4.7 – Сторінка адміну веб-сайту Нововодолазької центральної районної лікарні

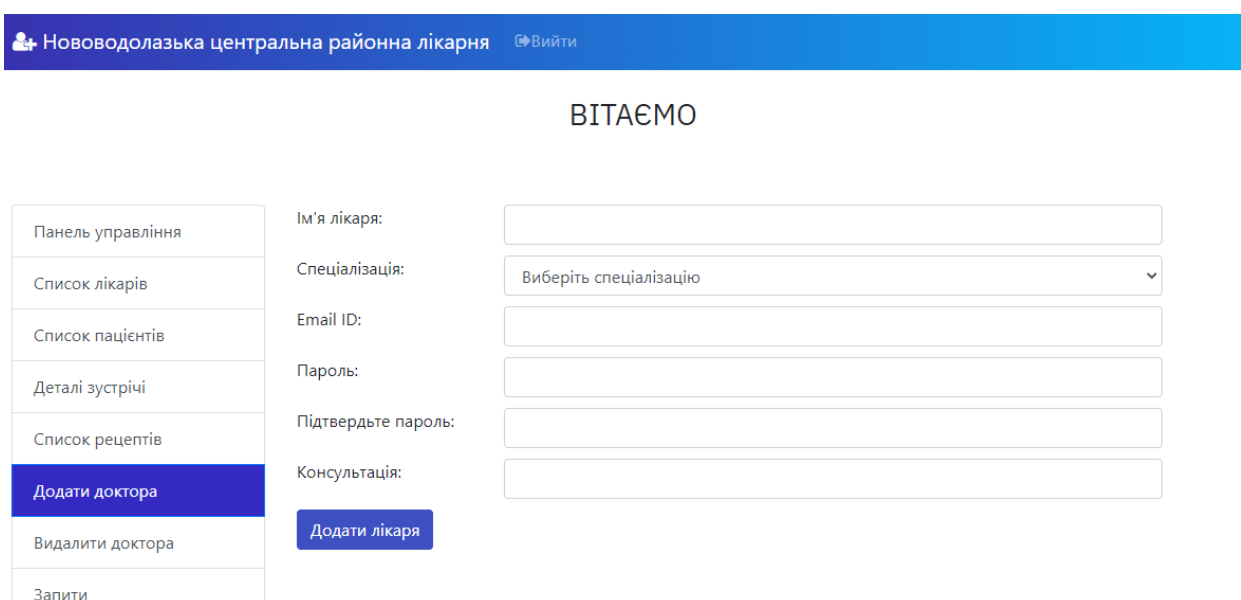


Рисунок 4.8 – Сторінка адміну веб-сайту Нововодолазької центральної районної лікарні – «Додати лікаря»

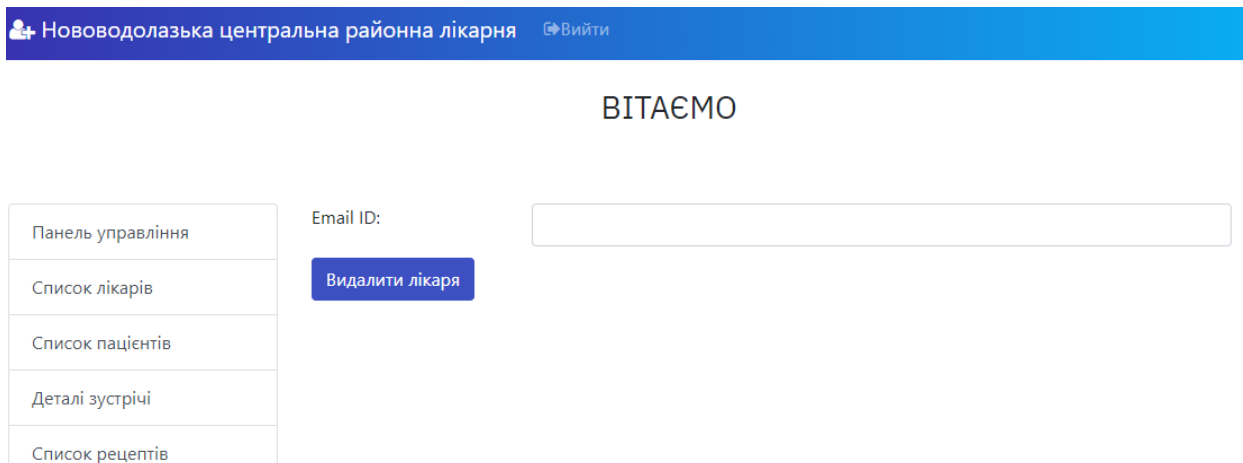


Рисунок 4.9 – Сторінка адміну веб-сайту Нововодолазької центральної районної лікарні - «Видалити лікаря»

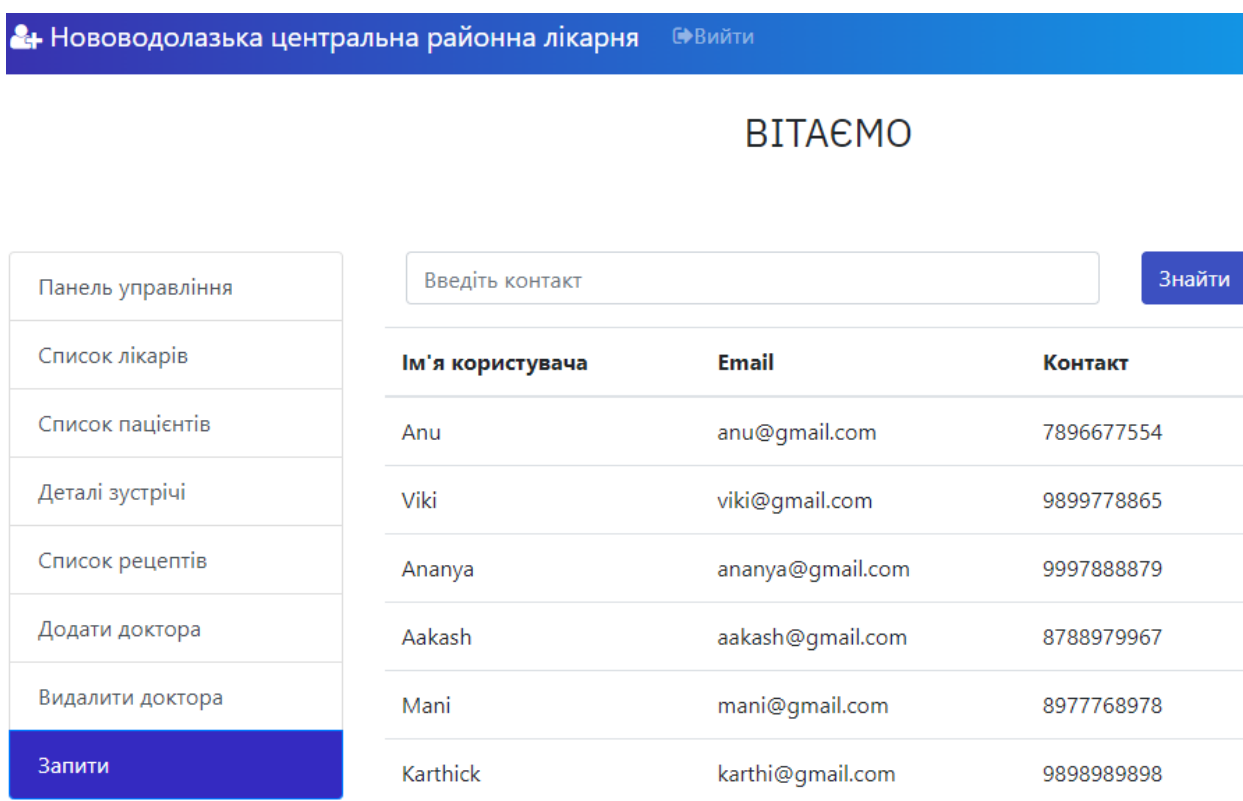


Рисунок 4.10 – Сторінка адміну веб-сайту Нововодолазької центральної районної лікарні – «Запити»

Проведемо тестування front-end частини сайту Нововодолазької центральної районної лікарні, результати тестування наведемо на рис. 4.11-4.16.

Нововодолазька центральна районна лікарня

НА ГОЛОВНУ ПРО ЛІКАРНЮ КОНТАКТИ

Пацієнт Лікар Адмін

Зареєструватися як пацієнт

Ласкаво просимо

Марун Tina

gfddddd@com 2922789

.....

Matched

Чоловіча Жіноча

[Вже є акаунт?](#)

Реєстрація

Рисунок 4.11 – Тестування веб-сайту Нововодолазької центральної районної лікарні

Нововодолазька центральна районна лікарня [Вийти](#)

Ласкаво просимо Марун Tina

Панель управління

- Запис на прийом
- Історія призначення
- Рецепти

Забронюйте мій прийом
Запис на прийом

Мої зустрічі
Переглянути історію зустрічей

Рецепти
Переглянути список рецептів

Рисунок 4.12 – Тестування веб-сайту Нововодолазької центральної районної лікарні

Нововодолазька центральна районна лікарня [Вийти](#)

Ласкаво просимо Марун Тіна

Панель управління

Запис на прийом

Історія призначення

Рецепти

Призначте зустріч

Спеціалізація:

Лікарі:

Консультація

Дата призначення:

Час призначення:

[Створити новий запис](#)

Рисунок 4.13 – Тестування веб-сайту Нововодолазької центральної районної лікарні

Нововодолазька центральна районна лікарня [Вийти](#)

Ласкаво просимо Марун Тіна

Панель управління

Запис на прийом

Історія призначення

Рецепти

Ім'я лікаря	Консультація	Дата призначення	Час призначення	Поточний статус	Дія

Рисунок 4.14 – Тестування веб-сайту Нововодолазької центральної районної лікарні

Нововодолазька центральна районна лікарня [Вийти](#)

Ласкаво просимо Марун Тіна

Панель управління

Запис на прийом

Історія призначення

Рецепти

Ім'я лікаря	Призначення ID	Дата призначення	Час призначення	Хвороби	Алергія	Рецепти	Оплата рахунків

Рисунок 4.15 – Тестування веб-сайту Нововодолазької центральної районної лікарні

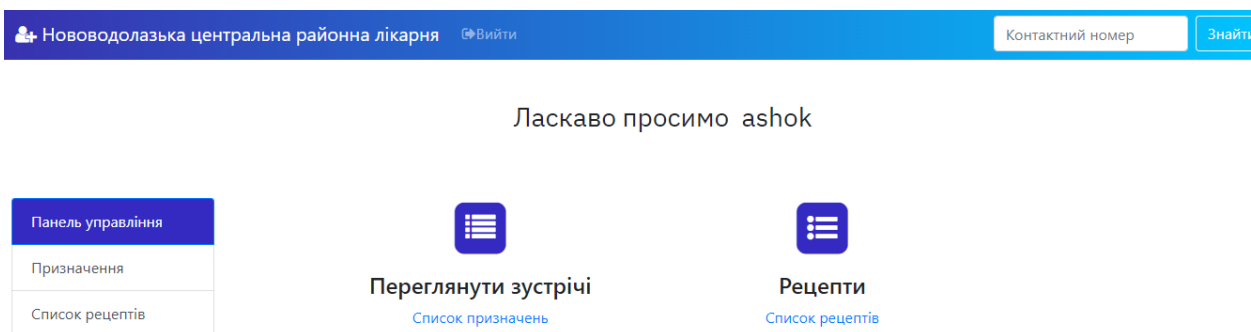


Рисунок 4.16 – Тестування веб-сайту Нововодолазької центральної районної лікарні, сторінка лікаря

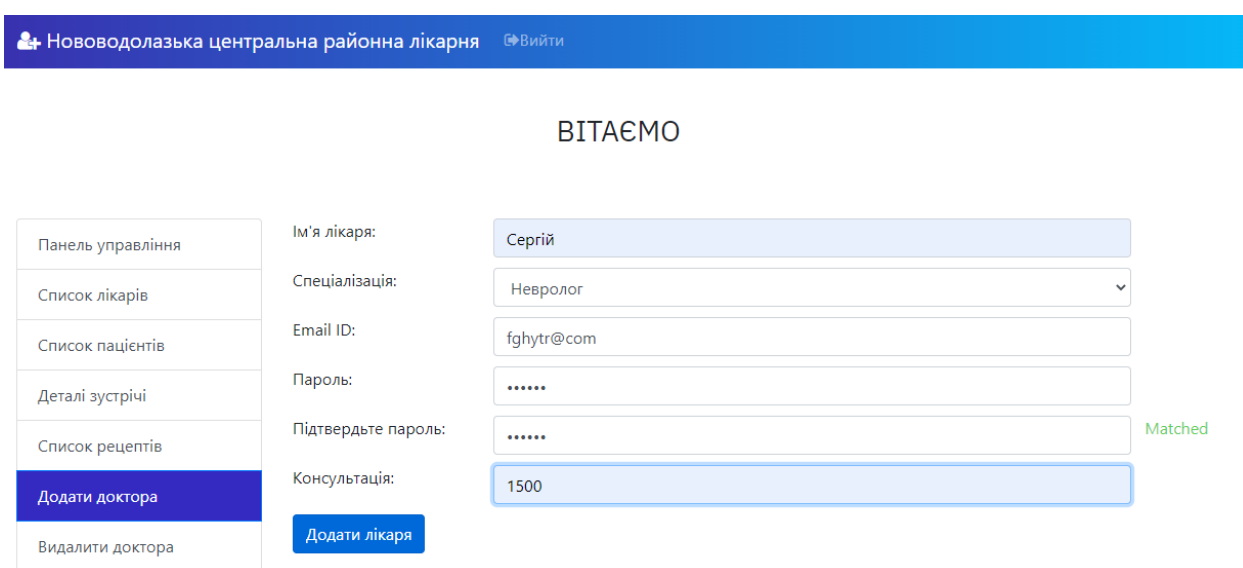


Рисунок 4.17 – Тестування веб-сайту Нововодолазької центральної районної лікарні

Призначення ID	Пацієнт ID	Ім'я	Прізвище	Стать	Email	Контакт	Ім'я лікаря	Консультація	Дата призначення	Час призначення	Статус призначення
1	4	Петро	Петренко	чоловік	kishansmart0@gmail.com	8838489464	Нестеренко Віталій Павлович	550	2022-02-14	10:00:00	Скасовано лікарем
2	4	Сергій	Григоренко	Чоловік	kishansmart0@gmail.com	8838489464	Сухий Перто Петрович	700	2022-05-28	10:00:00	Скасовано пацієнтом
3	4	Віталій	Сич	Чоловік	kishansmart0@gmail.com	8838489464	Семенов Григорій	1000	2022-05-19	03:00:00	Скасовано пацієнтом

Рисунок 4.18 – Тестування веб-сайту Нововодолазької центральної районної лікарні

Під час проведення тестування front-end частини сайту Нововодолазької центральної районної лікарні суттєвих збоїв та недоліків у роботі сайту не виявлено.

ВИСНОВКИ

В ході виконання дипломного проекту на тему "Розроблення front-end частини сайту інформаційної системи Нововодолазької центральної районної лікарні" проведено змістовний аналіз предметної області та обґрунтовано актуальність розробки front-end частини сайту інформаційної системи Нововодолазької центральної районної лікарні.

В першому розділі дипломного проекту проведено аналіз особливостей роботи медичних закладів України, розглянуті поняття веб-сайту, види розробок, проведено огляд програмного забезпечення лікарських закладів. В результаті проведеного аналізу у кожній з цих систем були виявлені недоліки, що доставляють певні незручності під час їх використання.

З метою усунення виявлених недоліків в дипломному проекті розроблено front-end частину сайту інформаційної системи Нововодолазької центральної районної лікарні.

Для дослідження бізнес-процесів інформаційної системи Нововодолазької центральної районної лікарні, в дипломному проекті було розроблено функціональну модель бізнес-процесів системи та діаграму потоків даних.

З метою зображення того, яким чином буде реалізована взаємодія користувача із інформаційною системою Нововодолазької центральної районної лікарні, була розроблена діаграма варіантів використання. Для детального опису функціональних можливостей інформаційною системою Нововодолазької центральної районної лікарні створена специфікація варіантів використання.

Також в дипломному проекті розроблено дерево цілей для реалізації поставленої задачі та алгоритм роботи сайту Нововодолазької центральної районної лікарні.

В третьому розділі даного дипломного проекту були розроблені проєктні рішення щодо реалізації роботи сайту Нововодолазької центральної районної лікарні. Для цього в проєкті було проведено концептуальне інфологічне проєктування, а саме створено таблиці «Словник даних» та «Обмеження атрибутів сутностей». Здійснено проєктування логічної моделі даних для виділення основних сутностей та зв'язків між ними. На основі логічної моделі даних була спроектована фізична модель даних.

В четвертому розділі представлені технічні рішення щодо роботи сайту Нововодолазької центральної районної лікарні.

Результати реалізації дипломного проекту «Розроблення front-end частини сайту Нововодолазької центральної районної лікарні» були апробовані на Міжнародній науково-практичній конференції молодих учених, аспірантів та студентів «Інформаційні технології в сучасному світі: дослідження молодих вчених» [2].

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Постанова Кабінету Міністрів України від 9 грудня 2020 р. № 1236 «Про встановлення карантину та запровадження обмежувальних протиепідемічних заходів з метою запобігання поширенню на території України гострої респіраторної хвороби COVID-19, спричиненої коронавірусом SARS-CoV-2». [Електронний ресурс]. Режим доступу: <https://www.kmu.gov.ua/npas/pro-vstanovlennya-karantinu-ta-zaprovadzhennya-obmezhuvalnih-protiepidemichnih-zahodiv-1236-091220>
2. Кондратенко А. Особливості розроблення front-end частини сайту медичного закладу. Міжнародна науково-практична конференція молодих учених, аспірантів та студентів Інформаційні технології в сучасному світі: дослідження молодих вчених: тези доповідей, (м. Харків, 17 – 18 лютого 2022 року) / Харківський національний економічний університет імені Семена Кузнеця. – Харків: ХНЕУ імені Семена Кузнеця, 2022. – С.78. [Електронний ресурс]. Режим доступу: <https://it.hneu.edu.ua/wp-content/uploads/2022/02/sbornyk-stud.-konf.2022.pdf>
3. Оперативна інформація про поширення та профілактику COVID-19 [Електронний ресурс]. Режим доступу: <https://moz.gov.ua/article/news/operativna-informacija-pro-poshirennja-koronavirusnoi-infekcii-2019-cov19>
4. Лікарня. [Електронний ресурс]. Режим доступу: <https://uk.wikipedia.org/wiki/%D0%9B%D1%96%D0%BA%D0%B0%D1%80%D0%BD%D1%8F>
5. У Харківській області визначили сім госпітальних округів. [Електронний ресурс]. Режим доступу: <https://kharkivoda.gov.ua/news/87904>
6. Постанова Кабінету Міністрів України №1074 від 27.11.2019 р. «Деякі питання створення госпітальних округів». [Електронний ресурс]. Режим доступу: <https://zakon.rada.gov.ua/laws/show/1074-2019-%D0%BF#Text>

7. Любченко К.М. Експертні системи в практичній медицині. [Електронний ресурс]. Режим доступу: http://it-visnyk.kpi.ua/wp-content/uploads/2011/03/49_01.pdf
8. Інформаційні технології в медицині [Електронний ресурс]. Режим доступу: <https://medytsyna.com/informatsijni-tehnologiyi-v-meditsini-1-docx/>
9. Гузенко, З & Шикула, О. (2020). Розробка веб-сервісу-агренатора лікарів та медичних закладів. *Vodnij transport*. 104-111. 10.33298/2226-8553/2020.2.30.12.
10. Ісакович Н. Н. (2015) Концепція розвитку служби клінічної лабораторної діагностики [Текст] / Н. Н. Ісакович. – Ейск: ПАО ЕЦРБ, 2015. – 18 с.
11. Грабовський, Є.М. (2021). Методика створення сайту Event-агентства. Збірник наукових праць Харківського національного університету Повітряних Сил. 70-76. 10.30748/zhups.2021.70.10.
12. Болгович М. Ю. (2016) Проблема формализации текстовых данных в универсальных медицинских информационных системах [Текст] / М. Ю. Болгович, Д. А. Мурта // Украинский журнал телемедицины и медицинской телематики. – 2016. – Т. 4, № 2. – С. 171-176.
13. Пархоменко, С. В. Мобільний ігровий додаток "Space Invasion": робота на здобуття кваліфікаційного рівня магістр: спец. 122 - комп'ютерні науки / наук. кер. Н. А. Федотова. Суми : СумДУ, 2020. 117 с. [Електронний ресурс]. Режим доступу: https://essuir.sumdu.edu.ua/bitstream-download/123456789/81427/1/master_thesis_Parkhomenko.pdf;jsessionid=A2F7AEE9A87FA7407347165DFA3EE3E0
14. Власюк В.В., Методи і засоби аналізу вимог та оцінювання якості патернів архітектур програмного забезпечення: дипломна робота магістра за спеціальністю "8.05010201 Комп'ютерні системи і мережі" / Власюк Володимир Володимирович; Тернопільський національний технічний університет імені Івана Пулюя. – Тернопіль: ТНТУ, 2013. – 135 с.

[Электронный ресурс]. Режим доступа:
<http://elartu.tntu.edu.ua/bitstream/123456789/2684/1/VlasiukVVDiplom.pdf>

15. Uzayr, Sufyan. (2022). Optimization Techniques for Front End, Content, and Back End. 10.1201/9781003203735-2.

16. Carvalho, Matheus (2013-08-01). Integration of Analytical Instruments with Computer Scripting [Electronic resource]. – Available at:\www/URL:[https://slas-technology.org/article/S2472-6303\(22\)01626-0/fulltext](https://slas-technology.org/article/S2472-6303(22)01626-0/fulltext).

17. Pearce, Joshua (2014-01-01). Indicators for the use of robotic labs in basic biomedical research: a literature analysis [Electronic resource]. – Available at:\www/URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5681851/>.

18. Bika 3 User manual [Electronic resource] // Bika Lab Systems. – Available at: \www/URL: <http://demo.bikalabs.com/knowledge-centre/manual/bika-3-user-manual> (Станом на 25.05.2022)

19. ISO/IEC 17025. General requirements for the competence of testing and calibration laboratories [Electronic resource]. – publ. 25.05.2022. – Available at:\www/URL: <http://dx.doi.org/10.3403/02033502u>

20. Nurpalah, A & Pasha, M & Rhamdhan, D & Maulana, Hanhan & Rafdhi, Agis. (2021). Effect of UI/UX Designer on Front End. International Journal of Research and Applied Technology. 1. 335-341. 10.34010/injuratech.v1i2.6759.

21. Ozdikililer, Egnar. (2021). Data Binding in Front End for Web Applications. The Journal of CIEES. 1. 31-34. 10.48149/jciees.2021.1.2.6.

22. Sloby, C. (2020). front end process. 10.1002/9783527809080.catanz07151.

23. Awasthi, Arnav & More, Shubham & Viegas, Warren. (2022). Research and Analysis of the Front-end Frameworks and Libraries in Web Development. International Journal for Research in Applied Science and Engineering Technology. 10. 411-416. 10.22214/ijraset.2022.41254.

24. Lopez, Daniel. (2021). Vue.js as a Front-End Layer. 10.1007/978-1-4842-6342-6_2.

25. Ferreira, Fabio & Borges, Hudson & Valente, Marco. (2021). On the (un-) adoption of JavaScript front-end frameworks. *Software: Practice and Experience*. 10.1002/spe.3044.
26. Takahashi, Sérgio & Takahashi, Vania. (2022). Analysis of front end dynamic in the value co-creation with multiple stakeholders. *International Journal of Managing Projects in Business*. 10.1108/IJMPB-11-2021-0301.
27. García, Moisés. (2020). A Minimal Front End with React. 10.1007/978-1-4842-6131-6_4.
28. Єфімов О.М. (2011). Інформаційні системи в економіці [Текст] / О.М. Єфімов, С.М. Патрушина, Л.Ф. Панфьорова, Л.І. Хашиєв. – Львів: ІКЦ "МарТ", 2011. – 352 с.
29. Житкова О. (2012). Проектування в СУБД Access [Текст] / О. Житкова, М. Журина, Є. Кудрявцева. – К.: "Інтелект-Центр", 2012. – 64 с.
30. Захаров, В.П. (2012). Інформаційні системи: навч. посібник / В.П. Захаров. – Х., 2012. – 187 с.
31. Зірка М.О., Бухтоярова В., Коробкіна Є., Кручинін Р. (2011). Бази даних. Розрахунково – графічні роботи з інформаційних систем [Текст] / За заг. ред. Зірка М.О. Навчально – методичний посібник для студентів вищих навчальних закладів (Спеціальність 080801 "Прикладна інформатика в економіці"), 2011. – 69 с.
32. Глушаков, С. В. (2006). Практикум по С++: учебное пособие [Текст] / С.В. Глушаков, А.В. Коваль, С.В. Смирнов. – Х.: Фолио, 2006. – 525 с.
33. Kavitha, M & Akhila, S & Kannan, Anand. (2021). Flexible Analog Front End Architecture for Biomedical Applications. 1-2. 10.1109/PhDEDITS53295.2021.9649471.
34. Batygin, Yuri & Gorelov, D & Kurennoy, S & Lewellen, J & Moody, N & Rybarcyk, L. (2021). ADVANCEMENT OF LANSCE FRONT END ACCELERATOR FACILITY. 10.18429/JACoW-IPAC2021-TUPAB205.

35. Cathelain, Guillaume & Rivet, Bertrand & Achard, Sophie & Bergounioux, Jean & Jouen, François. (2020). Smart ballistocardiography front-end. 10.1109/I2MTC43012.2020.9128774.

36. Al-Sabbagh, Taha. (2020). Front End to Back End Speech Scrambler. International Journal of Computing and Network Technology. 7. 65-70. 10.12785/ijcnt/070204

37. Han, Shuqin. (2021). Research on Web Front-end Performance Optimization Based on XML. 700-704. 10.1145/3510858.3511366.

38. Uzayr, Sufyan. (2022). Optimization Techniques for Front End, Content, and Back End. 10.1201/9781003203735-2.

39. Дипломний проект: методичні рекомендації для студентів спеціальності 126 "Інформаційні системи та технології" першого (бакалаврського) рівня [Електронний ресурс] / уклад. С. Г. Удовенко, О. О. Тютюник, В. А. Затхей та ін. – Харків: ХНЕУ ім. С. Кузнеця, 2022. – 52 с.

40. Проектування: [Електронне видання] методичні рекомендації до виконання курсового проекту для студентів спеціальності 126 "Інформаційні системи та технології" першого (бакалаврського) рівня / уклад. О. О. Тютюник, Н. О. Бринза, І. О. Ушакова. – Харків: ХНЕУ ім. С. Кузнеця, 2022. – 32 с.

41. Робоча програма навчальної дисципліни «Системний аналіз в ІТ» для студентів спеціальності 126 "Інформаційні системи та технології" першого (бакалаврського) рівня [Електронний ресурс] / уклад. С. Г. Удовенко, Н. О. Бринза. – Харків: ХНЕУ ім. С. Кузнеця, 2022. – 11 с.

42. Робоча програма навчальної дисципліни «Інформаційні системи та технології» для студентів спеціальності 126 "Інформаційні системи та технології" першого (бакалаврського) рівня [Електронний ресурс] / уклад. О. О. Тютюник, С. Г. Удовенко, Г. О. Плеханова. – Харків: ХНЕУ ім. С. Кузнеця, 2022. – 11 с.

43. Тютюнник В.В. Теорія систем та системний аналіз: курс лекцій / В.В. Тютюнник, О.О. Писклакова. – Харків: Національний університет цивільного захисту України, 2020. – 108 с.

ДОДАТОК А

Лістинг програмного коду

```

<!DOCTYPE html>
<?php
include('func.php');
include('newfunc.php');
$con=mysqli_connect("localhost","root","","myhmsdb");

$pid = $_SESSION['pid'];
$username = $_SESSION['username'];
$email = $_SESSION['email'];
$fname = $_SESSION['fname'];
$gender = $_SESSION['gender'];
$lname = $_SESSION['lname'];
$contact = $_SESSION['contact'];

if(isset($_POST['app-submit']))
{
    $pid = $_SESSION['pid'];
    $username = $_SESSION['username'];
    $email = $_SESSION['email'];
    $fname = $_SESSION['fname'];
    $lname = $_SESSION['lname'];
    $gender = $_SESSION['gender'];
    $contact = $_SESSION['contact'];
    $doctor=$_POST['doctor'];
    $email=$_SESSION['email'];
    # $fees=$_POST['fees'];
    $docFees=$_POST['docFees'];

    $appdate=$_POST['appdate'];
    $apptime=$_POST['apptime'];
    $cur_date = date("Y-m-d");
    date_default_timezone_set('Asia/Kolkata');
    $cur_time = date("H:i:s");
    $apptime1 = strtotime($apptime);
    $appdate1 = strtotime($appdate);

    if(date("Y-m-d",$appdate1)>=$cur_date){
        if((date("Y-m-d",$appdate1)==$cur_date and date("H:i:s",$apptime1)>$cur_time) or date("Y-m-d",
$appdate1)>$cur_date) {
            $check_query = mysqli_query($con,"select apptime from appointmenttb where doctor='$doctor' and
$appdate='$appdate' and apptime='$apptime'");

            if(mysqli_num_rows($check_query)==0){
                $query=mysqli_query($con,"insert into
appointmenttb(pid,fname,lname,gender,email,contact,doctor,docFees,appdate,apptime,userStatus,doctorStatus)
values($pid,$fname,$lname,$gender,$email,$contact,$doctor,$docFees,$appdate,$apptime,'1','1')");

                if($query)
                {
                    echo "<script>alert('Вашу зустріч успішно заброньовано');</script>";
                }
            }
        }
    }
}

```



```

        echo "<script>alert('Не вдається обробити ваш запит. Будь ласка спробуйте ще раз!');</script>";
    }
}
else{
    echo "<script>alert('З вибаченням повідомляємо, що лікар не доступний у цей час чи дату. Будь ласка, виберіть інший час або дату!');</script>";
}
}
else{
    echo "<script>alert('Виберіть час або дату в майбутньому!');</script>";
}
}
else{
    echo "<script>alert('Виберіть час або дату в майбутньому!');</script>";
}
}

if(isset($_GET['cancel']))
{
    $query=mysqli_query($con,"update appointmenttb set userStatus='0' where ID = '".$_GET['ID']."'");
    if($query)
    {
        echo "<script>alert('Вашу зустріч успішно скасовано');</script>";
    }
}

function generate_bill()
{
    $con=mysqli_connect("localhost","root","","myhmsdb");
    $pid = $_SESSION['pid'];
    $output="";
    $query=mysqli_query($con,"select
p.pid,p.ID,p.fname,p.lname,p.doctor,p.appdate,p.apptime,p.disease,p.allergy,p.prescription,a.docFees from prestb p
inner join appointmenttb a on p.ID=a.ID and p.pid = '$pid' and p.ID = '".$_GET['ID']."'");
    while($row = mysqli_fetch_array($query)){
        $output .= '
<label> Patient ID : </label>'. $row["pid"].'<br/><br/>
<label> Appointment ID : </label>'. $row["ID"].'<br/><br/>
<label> Patient Name : </label>'. $row["fname"].' '. $row["lname"].'<br/><br/>
<label> Doctor Name : </label>'. $row["doctor"].'<br/><br/>
<label> Appointment Date : </label>'. $row["appdate"].'<br/><br/>
<label> Appointment Time : </label>'. $row["apptime"].'<br/><br/>
<label> Disease : </label>'. $row["disease"].'<br/><br/>
<label> Allergies : </label>'. $row["allergy"].'<br/><br/>
<label> Prescription : </label>'. $row["prescription"].'<br/><br/>
<label> Fees Paid : </label>'. $row["docFees"].'<br/>

';
    }

    return $output;
}

if(isset($_GET["generate_bill"])){
    require_once("TCPDF/tcpdf.php");
    $obj_pdf = new TCPDF('P',PDF_UNIT,PDF_PAGE_FORMAT,true,'UTF-8',false);
}

```

```

$objj_pdf -> SetCreator(PDF_CREATOR);
$objj_pdf -> SetTitle("Generate Bill");
$objj_pdf -> SetHeaderData("",PDF_HEADER_TITLE,PDF_HEADER_STRING);
$objj_pdf -> SetHeaderFont(Array(PDF_FONT_NAME_MAIN,"PDF_FONT_SIZE_MAIN));
$objj_pdf -> SetFooterFont(Array(PDF_FONT_NAME_MAIN,"PDF_FONT_SIZE_MAIN));
$objj_pdf -> SetDefaultMonospacedFont('helvetica');
$objj_pdf -> SetFooterMargin(PDF_MARGIN_FOOTER);
$objj_pdf -> SetMargins(PDF_MARGIN_LEFT,'5',PDF_MARGIN_RIGHT);
$objj_pdf -> SetPrintHeader(false);
$objj_pdf -> SetPrintFooter(false);
$objj_pdf -> SetAutoPageBreak(TRUE, 10);
$objj_pdf -> SetFont('helvetica',"",12);
$objj_pdf -> AddPage();

$content = "";

$content .= '
<br/>
<h2 align="center"> Нововодолазька центральна районна лікарня</h2></br>
<h3 align="center"> Bill</h3>

';

$content .= generate_bill();
$objj_pdf -> writeHTML($content);
ob_end_clean();
$objj_pdf -> Output("bill.pdf",'I');
}

function get_specs(){
$con=mysqli_connect("localhost","root","","myhmsdb");
$query=mysqli_query($con,"select username,spec from doctb");
$docarray = array();
while($row =mysqli_fetch_assoc($query))
{
    $docarray[] = $row;
}
return json_encode($docarray);
}

?>
<html lang="en">
<head>

<!-- Required meta tags -->
<meta charset="utf-8">
<link rel="shortcut icon" type="image/x-icon" href="images/favicon.png" />
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<link rel="stylesheet" type="text/css" href="font-awesome-4.7.0/css/font-awesome.min.css">
<link rel="stylesheet" href="style.css">
<!-- Bootstrap CSS -->

<link rel="stylesheet" href="vendor/fontawesome/css/font-awesome.min.css">

```

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta/css/bootstrap.min.css"
integrity="sha384-Y6pD6FV/Vv2HJnA6t+vsIU6fwYXjCFtcEpHbNJ0lyAFsXTsjBbfaDjzALeQsN6M"
crossorigin="anonymous">
```

```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
```

```
<link href="https://fonts.googleapis.com/css?family=IBM+Plex+Sans&display=swap" rel="stylesheet">
<nav class="navbar navbar-expand-lg navbar-dark bg-primary fixed-top">
<a class="navbar-brand" href="#"><i class="fa fa-user-plus" aria-hidden="true"></i> Нововодолазка централна
районна ликарня </a>
<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>
```

```
<style >
.bg-primary {
background: -webkit-linear-gradient(left, #3931af, #00c6ff);
}
.list-group-item.active {
z-index: 2;
color: #fff;
background-color: #342ac1;
border-color: #007bff;
}
.text-primary {
color: #342ac1 !important;
}

.btn-primary {
background-color: #3c50c1;
border-color: #3c50c1;
}
</style>
```

```
<div class="collapse navbar-collapse" id="navbarSupportedContent">
<ul class="navbar-nav mr-auto">
<li class="nav-item">
<a class="nav-link" href="logout.php"><i class="fa fa-sign-out" aria-hidden="true"></i> Вийти</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#"></a>
</li>
</ul>
</div>
</nav>
</head>
```

```
<style type="text/css">
button:hover {cursor:pointer;}
#inputbtn:hover {cursor:pointer;}
</style>
<body style="padding-top:50px;">
```

```
<div class="container-fluid" style="margin-top:50px;">
<h3 style="margin-left: 40%; padding-bottom: 20px; font-family: 'IBM Plex Sans', sans-serif;"> Ласкаво
просимо &nbsp;<?php echo $username ?>
</h3>
<div class="row">
```

```

<div class="col-md-4" style="max-width:25%; margin-top: 3%">
<div class="list-group" id="list-tab" role="tablist">
<a class="list-group-item list-group-item-action active" id="list-dash-list" data-toggle="list" href="#list-dash"
role="tab" aria-controls="home">Панель управління</a>
<a class="list-group-item list-group-item-action" id="list-home-list" data-toggle="list" href="#list-home" role="tab"
aria-controls="home">Запис на прийом</a>
<a class="list-group-item list-group-item-action" href="#app-hist" id="list-pat-list" role="tab" data-toggle="list"
aria-controls="home">Історія призначення</a>
<a class="list-group-item list-group-item-action" href="#list-pres" id="list-pres-list" role="tab" data-toggle="list"
aria-controls="home">Рецепти</a>

</div><br>
</div>
<div class="col-md-8" style="margin-top: 3%;">
<div class="tab-content" id="nav-tabContent" style="width: 950px;">

<div class="tab-pane fade show active" id="list-dash" role="tabpanel" aria-labelledby="list-dash-list">
<div class="container-fluid container-fullw bg-white" >
<div class="row">
<div class="col-sm-4" style="left: 5%">
<div class="panel panel-white no-radius text-center">
<div class="panel-body">
<span class="fa-stack fa-2x"><i class="fa fa-square fa-stack-2x text-primary"></i><i class="fa fa-terminal fa-stack-
1x fa-inverse"></i></span>
<h4 class="StepTitle" style="margin-top: 5%;"> Забронюйте мій прийом</h4>
<script>
function clickDiv(id) {
document.querySelector(id).click();
}
</script>
<p class="links cl-effect-1">
<a href="#list-home" onclick="clickDiv('#list-home-list')">
Запис на прийом
</a>
</p>
</div>
</div>
</div>
</div>

<div class="col-sm-4" style="left: 10%">
<div class="panel panel-white no-radius text-center">
<div class="panel-body" >
<span class="fa-stack fa-2x"><i class="fa fa-square fa-stack-2x text-primary"></i><i class="fa fa-paperclip fa-
stack-1x fa-inverse"></i></span>
<h4 class="StepTitle" style="margin-top: 5%;">Мої зустрічі</h2>

<p class="cl-effect-1">
<a href="#app-hist" onclick="clickDiv('#list-pat-list')">
Переглянути історію зустрічей
</a>
</p>
</div>
</div>
</div>
</div>

<div class="col-sm-4" style="left: 20%;margin-top:5%">
<div class="panel panel-white no-radius text-center">
<div class="panel-body" >
<span class="fa-stack fa-2x"><i class="fa fa-square fa-stack-2x text-primary"></i><i class="fa fa-list-ul fa-stack-1x
fa-inverse"></i></span>

```

```

<h4 class="StepTitle" style="margin-top: 5%;">Рецепти</h2>

<p class="cl-effect-1">
<a href="#list-pres" onclick="clickDiv('#list-pres-list')">
    Переглянути список рецептів
</a>
</p>
</div>
</div>
</div>

</div>
</div>

<div class="tab-pane fade" id="list-home" role="tabpanel" aria-labelledby="list-home-list">
<div class="container-fluid">
<div class="card">
<div class="card-body">
<center><h4>Призначте зустріч</h4></center><br>
<form class="form-group" method="post" action="admin-panel.php">
<div class="row">

<!-- <?php
    $con=mysqli_connect("localhost","root","","myhmsdb");
    $query=mysqli_query($con,"select username,spec from doctb");
    $docarray = array();
    while($row =mysqli_fetch_assoc($query))
    {
        $docarray[] = $row;
    }
    echo json_encode($docarray);

?> -->

<div class="col-md-4">
<label for="spec">Спеціалізація:</label>
</div>
<div class="col-md-8">
<select name="spec" class="form-control" id="spec">
<option value="" disabled selected>Виберіть спеціалізацію</option>
<?php
    display_specs();
?>

</select>
</div>

<br><br>

<script>
document.getElementById('spec').onchange = function foo() {
    let spec = this.value;
    console.log(spec)
    let docs = [...document.getElementById('doctor').options];

    docs.forEach((el, ind, arr)=>{

```

```

        arr[ind].setAttribute("style","");
        if (el.getAttribute("data-spec") != spec ) {
            arr[ind].setAttribute("style","display: none");
        }
    });
};

</script>

<div class="col-md-4"><label for="doctor">Лікарі:</label></div>
<div class="col-md-8">
<select name="doctor" class="form-control" id="doctor" required="required">
<option value="" disabled selected>Виберіть лікаря</option>

<?php display_docs(); ?>
</select>
</div><br/><br/>

<script>
document.getElementById('doctor').onchange = function updateFees(e) {
    var selection = document.querySelector('[value=${this.value}]').getAttribute('data-value');
    document.getElementById('docFees').value = selection;
};
</script>

<!-- <div class="col-md-4"><label for="doctor">Doctors:</label></div>
<div class="col-md-8">
<select name="doctor" class="form-control" id="doctor1" required="required">
<option value="" disabled selected>Select Doctor</option>

</select>
</div>
<br><br> -->

<!-- <script>
document.getElementById("spec").onchange = function updateSpecs(event) {
    var selected = document.querySelector('[data-value=${this.value}]').getAttribute("value");
    console.log(selected);

    var options = document.getElementById("doctor1").querySelectorAll("option");

    for (i = 0; i < options.length; i++) {
        var currentOption = options[i];
        var category = options[i].getAttribute("data-spec");

        if (category == selected) {
            currentOption.style.display = "block";
        } else {
            currentOption.style.display = "none";
        }
    }
}
</script> -->

<!-- <script>
let data =

```

```

    document.getElementById('spec').onchange = function updateSpecs(e) {
        let values = data.filter(obj => obj.spec === this.value).map(o => o.username);
        document.getElementById('doctor1').value =
document.querySelector(`[value=${values}]`).getAttribute('data-value');
    };
</script> -->

```

```

<div class="col-md-4"><label for="consultancyfees">
    Консультація

```

```

</label></div>

```

```

<div class="col-md-8">

```

```

<!-- <div id="docFees">Select a doctor</div> -->

```

```

<input class="form-control" type="text" name="docFees" id="docFees" readonly="readonly"/>

```

```

</div><br><br>

```

```

<div class="col-md-4"><label>Дата призначення</label></div>

```

```

<div class="col-md-8"><input type="date" class="form-control datepicker" name="appdate"></div><br><br>

```

```

<div class="col-md-4"><label>Час призначення</label></div>

```

```

<div class="col-md-8">

```

```

<!-- <input type="time" class="form-control" name="apptime"> -->

```

```

<select name="apptime" class="form-control" id="apptime" required="required">

```

```

<option value="" disabled selected>Оберіть час</option>

```

```

<option value="08:00:00">8:00</option>

```

```

<option value="10:00:00">10:00</option>

```

```

<option value="12:00:00">12:00</option>

```

```

<option value="14:00:00">14:00</option>

```

```

<option value="16:00:00">16:00</option>

```

```

</select>

```

```

</div><br><br>

```

```

<div class="col-md-4">

```

```

<input type="submit" name="app-submit" value="Створити новий запис" class="btn btn-primary" id="inputbtn">

```

```

</div>

```

```

<div class="col-md-8"></div>

```

```

</div>

```

```

</form>

```

```

</div>

```

```

</div>

```

```

</div><br>

```

```

</div>

```

```

<div class="tab-pane fade" id="app-hist" role="tabpanel" aria-labelledby="list-pat-list">

```

```

<table class="table table-hover">

```

```

<thead>

```

```

<tr>

```

```

<th scope="col">Ім'я лікаря</th>

```

```

<th scope="col">Консультація</th>

```

```

<th scope="col">Дата призначення</th>

```

```

<th scope="col">Час призначення</th>

```

```

<th scope="col">Поточний статус</th>

```

```

<th scope="col">Дія</th>

```

```

</tr>

```

```

</thead>

```

```

<tbody>

```

```

<?php

```

```

$con=mysqli_connect("localhost","root","","myhmsdb");
global $con;

$query = "select ID,doctor,docFees,appdate,apptime,userStatus,doctorStatus from appointmentb where
fname ='$fname' and lname='$lname';";
$result = mysqli_query($con,$query);
while ($row = mysqli_fetch_array($result)){

    #$fname = $row['fname'];
    #$lname = $row['lname'];
    #$email = $row['email'];
    #$contact = $row['contact'];
    ?>
<tr>
<td><?php echo $row['doctor'];?></td>
<td><?php echo $row['docFees'];?></td>
<td><?php echo $row['appdate'];?></td>
<td><?php echo $row['apptime'];?></td>

<td>
<?php if(($row['userStatus']==1) && ($row['doctorStatus']==1))
    {
        echo "Активний";
    }
    if(($row['userStatus']==0) && ($row['doctorStatus']==1))
    {
        echo "Скасовано вами";
    }

    if(($row['userStatus']==1) && ($row['doctorStatus']==0))
    {
        echo "Скасовано лікарем";
    }
    ?></td>

<td>
<?php if(($row['userStatus']==1) && ($row['doctorStatus']==1))
    { ?>

        <a href="admin-panel.php?ID=<?php echo $row['ID'];?>&cancel=update"
            onClick="return confirm('Ви впевнені, що хочете скасувати цю зустріч? ?)'"
            title="Cancel Appointment" tooltip-placement="top" tooltip="Remove"><button class="btn btn-
danger">Відмінити</button></a>
        <?php } else {

            echo "Скасовано";
        } ?>

</td>
</tr>
<?php } ?>
</tbody>
</table>
<br>
</div>

<div class="tab-pane fade" id="list-pres" role="tabpanel" aria-labelledby="list-pres-list">

```



```

<table class="table table-hover">
<thead>
<tr>

<th scope="col">Ім'я лікаря</th>
<th scope="col">Призначення ID</th>
<th scope="col">Дата призначення</th>
<th scope="col">Час призначення</th>
<th scope="col">Хвороби</th>
<th scope="col">Алергія</th>
<th scope="col">Рецепти</th>
<th scope="col">Оплата рахунків</th>
</tr>
</thead>
<tbody>
<?php

    $con=mysqli_connect("localhost","root","","myhmsdb");
    global $con;

    $query = "select doctor,ID,appdate,apptime,disease,allergy,prescription from prestb where pid='$pid'";

    $result = mysqli_query($con,$query);
    if(!$result){
        echo mysqli_error($con);
    }

    while ($row = mysqli_fetch_array($result)){
?>
<tr>
<td><?php echo $row['doctor'];?></td>
<td><?php echo $row['ID'];?></td>
<td><?php echo $row['appdate'];?></td>
<td><?php echo $row['apptime'];?></td>
<td><?php echo $row['disease'];?></td>
<td><?php echo $row['allergy'];?></td>
<td><?php echo $row['prescription'];?></td>
<td>
<form method="get">
<!-- <a href="admin-panel.php?ID="
        onClick=""
            title="Pay Bill" tooltip-placement="top" tooltip="Remove"><button class="btn btn-
success">Pay</button>
</a></td> -->

<a href="admin-panel.php?ID=<?php echo $row['ID'];?>">
<input type="hidden" name="ID" value="<?php echo $row['ID'];?>" />
<input type="submit" onclick="alert('Рахунок успішно оплачено');" name="generate_bill" class="btn btn-
success" value="Оплатити рахунок" />
</a>
</td>
</form>

</tr>
<?php }
?>
</tbody>
</table>
<br>
</div>

```

```

<div class="tab-pane fade" id="list-messages" role="tabpanel" aria-labelledby="list-messages-list">...</div>
<div class="tab-pane fade" id="list-settings" role="tabpanel" aria-labelledby="list-settings-list">
<form class="form-group" method="post" action="func.php">
<label>Прізвища лікарів: </label>
<input type="text" name="name" placeholder="Введіть прізвище лікаря" class="form-control">
<br>
<input type="submit" name="doc_sub" value="Додати лікаря" class="btn btn-primary">
</form>
</div>
<div class="tab-pane fade" id="list-attend" role="tabpanel" aria-labelledby="list-attend-list">...</div>
</div>
</div>
</div>
</div>
<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DKtIkVYIK3UENzmM7KCKRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.11.0/umd/popper.min.js" integrity="sha384-
b/U6ypiBEHpOf/4+1nzFpr53nxSS+GLCkfwBdFNTxtclqenISfwAzpKaMNFNmj4"
crossorigin="anonymous"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta/js/bootstrap.min.js" integrity="sha384-
h0AbiXch4ZDo7tp9hKZ4TsHbi047NrKGL03SEJAg45jXxnGIfYzk4Si90RDIqNm1"
crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/limonte-sweetalert2/6.10.1/sweetalert2.all.min.js">
</script>

</body>
</html>

```

ЗВІТ З ПЕРЕВІРКИ НА ПЛАГІАТ

ЦЕЙ ЗВІТ ЗАСВІДЧУЄ, ЩО ПРИКРІПЛЕНА РОБОТА

Кондратенко А.В.6.04.126.010.18.1.docx

БУЛА ПЕРЕВІРЕНА СЕРВІСОМ ДЛЯ ЗАПОБІГАННЯ ПЛАГІАТУ

MY.PLAGRAMME.COM І МАЄ:

СХОЖІСТЬ

14%

РИЗИК ПЛАГІАТУ

100%

ПЕРЕФРАЗУВАННЯ

1%

НЕПРАВИЛЬНІ ЦИТУВАННЯ

0%

Назва файлу: Кондратенко А.В.6.04.126.010.18.1.docx

Файл перевірено: 2022-06-12

Звіт створено: 2022-06-12