

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

**на тему: «Інформаційна система підтримки діяльності реєстратури
поліклініки»**

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології проектування»

Виконавець роботи: студентка групи ІТ-61 Бірченко Альона Володимирівна

**Кваліфікаційна робота бакалавра
захищена на засіданні ЕК
з оцінкою**
2020 р.

_____ «___»

Науковий керівник

(підпис)

к. т. н., доц. Шендрик В.В.

(науковий ступінь, вчене звання, прізвище та ініціали)

Голова комісії

(підпис)

Шифрін Д. М.

(науковий ступінь, вчене звання, прізвище та ініціали)

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Суми-2020

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук
Секція інформаційних технологій проектування
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології
проектування»

ЗАТВЕРДЖУЮ

Зав. секцією ІТП

_____ В. В. Шендрик
«__» _____ 2020 р.

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

Бирченко Альона Володимирівна

1 Тема роботи Інформаційна система підтримки діяльності реєстратури
поліклініки

керівник роботи Шендрик Віра Вікторівна, к.т.н., доцент,

затверджені наказом по університету від «14» травня 2020 р. №0576-III

2 Строк подання студентом роботи «1» червня 2020 р.

3 Вхідні дані до роботи технічне завдання на розробку інформаційної
системи, методичні вказівки і вимоги до оформлення дипломної роботи

**4 Зміст розрахунково-пояснювальної записки (перелік питань, які
потрібно розробити)** аналіз предметної області, проектування інформаційної
системи, розробка інформаційної системи.

**5 Перелік графічного матеріалу (з точним зазначенням обов'язкових
креслень)** мета та задачі ІС, аналіз продуктів-аналогів, функціональні
вимоги до інформаційної системи, макет головної сторінки, контекстна
діаграма, діаграма декомпозиції, діаграма варіантів використання, схема бази
даних, засоби реалізації, архітектура інформаційної системи, демонстрація
роботи, висновки, оприлюднення результатів роботи.

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 01.10.2019

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області	2	
2	Аналіз продуктів-аналогів	2	
3	Постановка задач та визначення вимог до ІС	2	
4	Складання Технічного завдання	3	
5	Розробка дизайну	3	
6	Розробка схеми бази даних	2	
7	Верстка та програмування	14	
8	Проведення тестування	2	
9	Виправлення знайдених дефектів	3	
10	Оформлення пояснювальної записки	2	

Студент _____
(підпис)

Бирченко А.В.

Керівник роботи _____
(підпис)

к.т.н., доц. Шендрик В.В.

РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Інформаційна система підтримки діяльності реєстратури поліклініки».

Пояснювальна записка складається зі вступу, 3 розділів, висновків, списку використаних джерел із 20 найменувань, додатків. Загальний обсяг роботи – 82 сторінок, у тому числі 56 сторінок основного тексту, 2 сторінки списку використаних джерел, 17 сторінок додатків.

У першому розділі було проведено аналіз продуктів-аналогів, поставлено мету та виділено задачі роботи.

У другому розділі було здійснено моделювання інформаційної системи, а саме було побудовано діаграму варіантів використання, контекстну діаграму та діаграму декомпозиції у нотації IDEF0, модель бази даних.

У третьому розділі було описано реалізацію інформаційної системи.

Результатом кваліфікаційної роботи бакалавра є розроблена інформаційна система підтримки діяльності реєстратури поліклініки

Ключові слова: інформаційна система, моделювання, ERD-діаграма, база даних, Wordpress, HTML, CSS, PHP, JS.

ЗМІСТ

Вступ.....	6
1. Аналіз предметної області.....	8
1.1 Загальна характеристика предметної області.....	8
1.3 Постановка задачі.....	13
1.4 Вибір засобів реалізації.....	14
2. Проектування інформаційної системи.....	16
2.1 Діаграми у нотаціях IDEF0.....	16
3. Розробка інформаційної системи.....	31
3.1 Архітектура інформаційної системи.....	31
3.2 Програмна реалізація.....	33
3.3 Використання інформаційної системи.....	42
Висновки.....	56
Список використаних джерел.....	57
Додаток А. Технічне завдання.....	59
Додаток Б. Планування робіт.....	71
Додаток Г. Лістинг програмного коду.....	78

ВСТУП

Характерною рисою сьогодення є інформаційні технології, адже вони застосовуються у всіх сферах життя людини та значно полегшують його. Майже кожна компанія, яка займається наданням послуг на ринку, не може обійтися без впровадження інформаційної системи в її роботу. Від функціонування та організаційної структури інформаційної системи залежать не лише внутрішні процеси роботи закладу, а також і його популярність серед населення.

Предметом особливої уваги сьогодні є впровадження інформаційних систем в заклади охорони здоров'я.

Велика кількість медичних організацій мають власну інформаційну систему, так як сучасні медичні установи зберігають та обробляють великий обсяг інформації, який до того ж постійно зростає. Від того наскільки ефективно і раціонально інформація буде використовуватися лікарями, керівниками або керуючими органами, залежить рівень медичної допомоги, загальний рівень життя населення та загальний рівень розвитку країни. Тому використання великих обсягів інформації при вирішенні різних завдань в сфері медицини, зумовлює в даний час використання інформаційних систем в медичних організаціях. Крім цього впровадження та використання інформаційних систем у діяльності медичних закладів налагоджує взаємодію з потенційними пацієнтами, дозволяє орієнтуватися на їх побажання, переконати їх у високій якості послуг, які надає медичний заклад та кваліфікованості лікарів.

Отже, актуальністю даної роботи є те, що впровадження інформаційної системи підтримки діяльності реєстратури поліклініки дозволить максимально ефективно та якісно організувати роботу лікарів та працівників реєстратури.

Метою даної кваліфікаційної роботи є створення інформаційної системи підтримки діяльності реєстратури поліклініки для того, щоб автоматизувати основні процеси.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- проаналізувати програмні продукти-аналоги;
- визначити функціональні та нефункціональні вимоги до інформаційної системи;
- розробити технічне завдання;
- здійснити моделювання інформаційної системи;
- розробити дизайн інформаційної системи;
- реалізувати інформаційну систему;
- розробити супровідну документацію (пояснювальна записка).

Практичне значення – використання розробленої інформаційної системи допоможе більш якісно організувати роботу поліклініки, значно основні спростить процеси, що виконуються лікарями, працівниками реєстратури пацієнтами поліклініки.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Загальна характеристика предметної області

Сьогодні для стабільного та успішного функціонування будь-якої установи, закладу або організації є необхідним удосконалення методів управління. Одним з інструментів покращення системи управління є впровадження інформаційних систем, що дозволить організувати ефективне управління діяльністю установ. Існує багато досліджень, присвячених питанню створення систем автоматизації медичних закладів, де виділяється їх класифікація та призначення, переваги та недоліки впровадження інформаційних систем, а також проводиться аналіз подальших перспектив їх розвитку у закладах медичного обслуговування.

На сьогодні в класифікації інформаційних систем в охороні здоров'я передбачається такі основні класи [5]: 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.

Поліклініка виконує важливу організаційно-методичну та координуючу роль у вдосконаленні якості медичної допомоги населенню. Необхідність автоматизації процесів роботи поліклініки пов'язана з витратами великої кількості часу на обробку інформації людськими ресурсами, а саме працівниками реєстратури та лікарями поліклініки.

В даній роботі мова йде про поліклініку, яка займається наданням медичних послуг, що включають в себе:

- реєстрація пацієнта;
- призначення прийому до лікаря;
- прийом пацієнтів та консультації фахівців;
- проведення обстежень різного ступеня складності;
- призначення курсу лікування.

Клієнти – люди різного віку і соціальних категорій, як приватні, так і корпоративні. Користувачами інформаційної системи є пацієнти і люди, що зацікавлені у співпраці.

Цілями розробки інформаційної системи є:

1. створення сприятливого враження про поліклініку;
2. залучення нових пацієнтів;
3. розміщення інформації про послуги та їх вартість;
4. спрощення процесу запису на прийом;
5. спрощення процесу надання призначень.

1.2 Аналіз програмних продуктів-аналогів

На даний час в Україні нараховується приблизно 23000 медичних закладів. Більшість з них є державними медичними закладами, але кількість упроваджень медичних інформаційних систем вища у приватних медичних установах.

У ході дослідження було виділено такі аналоги:

- web-сайт мережі медичних центрів «ОН Клінік»;

Використовуючи даний web-сайт, користувач має змогу отримати інформацію про лікарів, послуги та ціни медичного центру, а також обрати лікаря та записатися на прийом. На рисунку 1.1 зображено інтерфейс web-сайту.

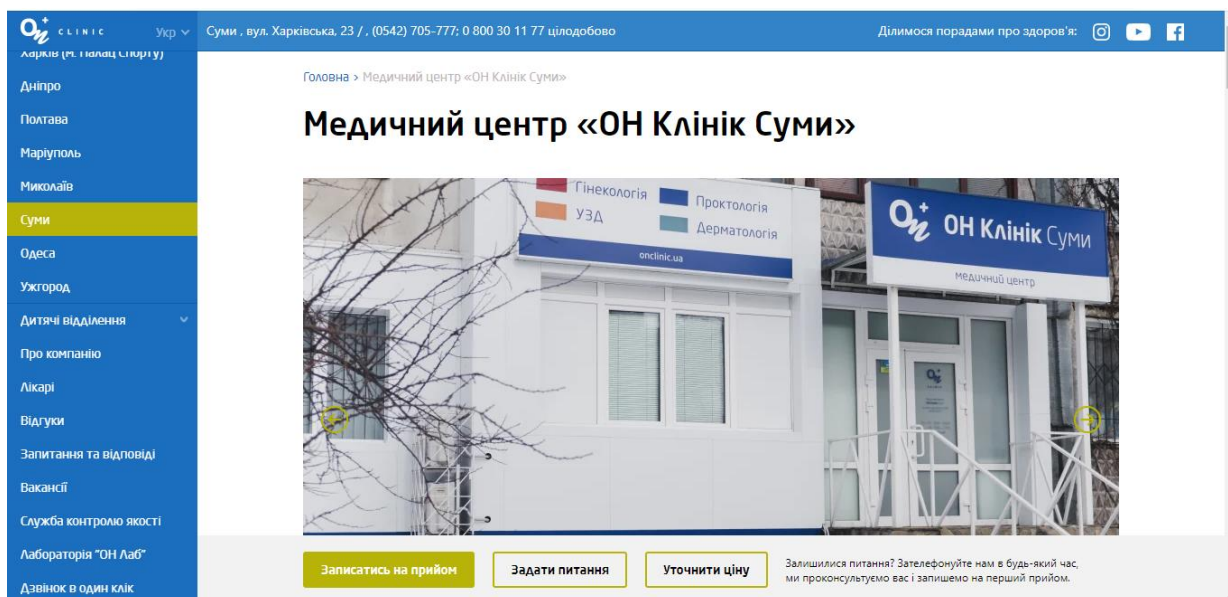


Рисунок 1.1 – Приклад інтерфейсу web-сайту ОН Клінік

– web-сайт клініки ТОВ «Верба Медікал»

Даний web-сайт надає можливість запису на прийом та перегляду послуг, цін та можливість задавання питання лікарям онлайн. На рисунку 1.2 зображено інтерфейс web-сайту.

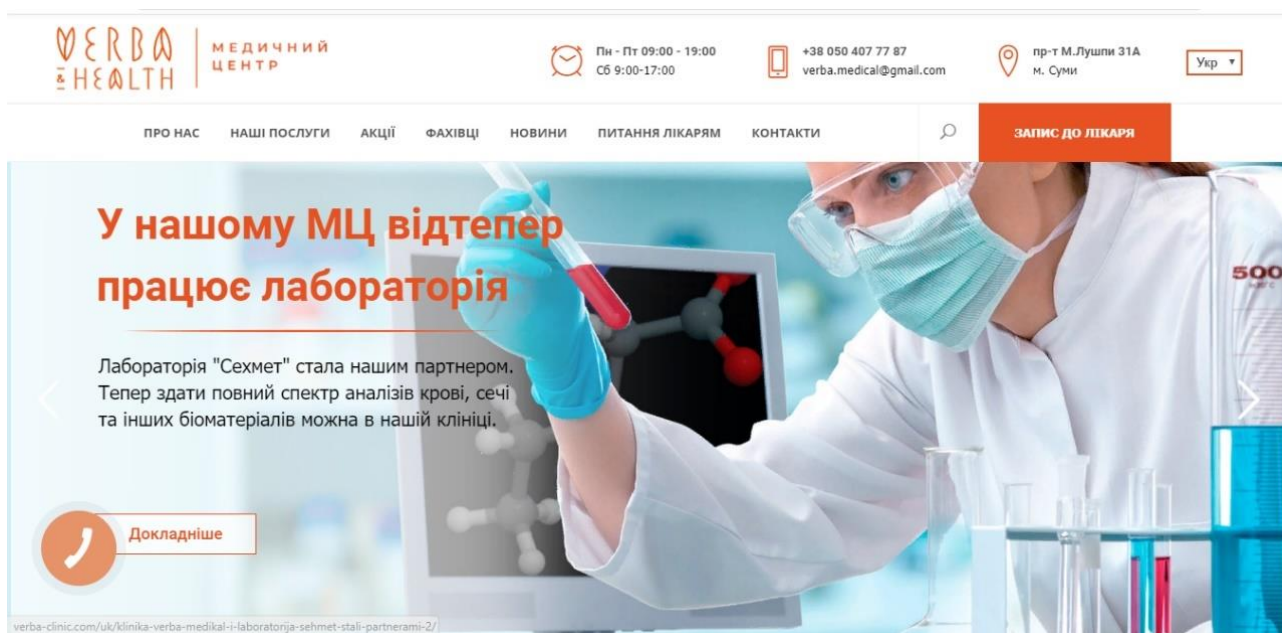


Рисунок 1.2 – Приклад інтерфейсу клініки ТОВ «Верба Медікал»

– Інформаційна система HELSI.ME

HELSI.ME є електронною медичною системою, яка розроблена для лікарів, пацієнтів, державних та приватних медичних закладів.

Дана інформаційна система надає можливості автоматизації реєстратури та роботи лікаря, управління розкладом лікаря, ведення електронних медичних карток та доступу пацієнтів до них. На рисунку 1.3 зображено інтерфейс інформаційної системи.

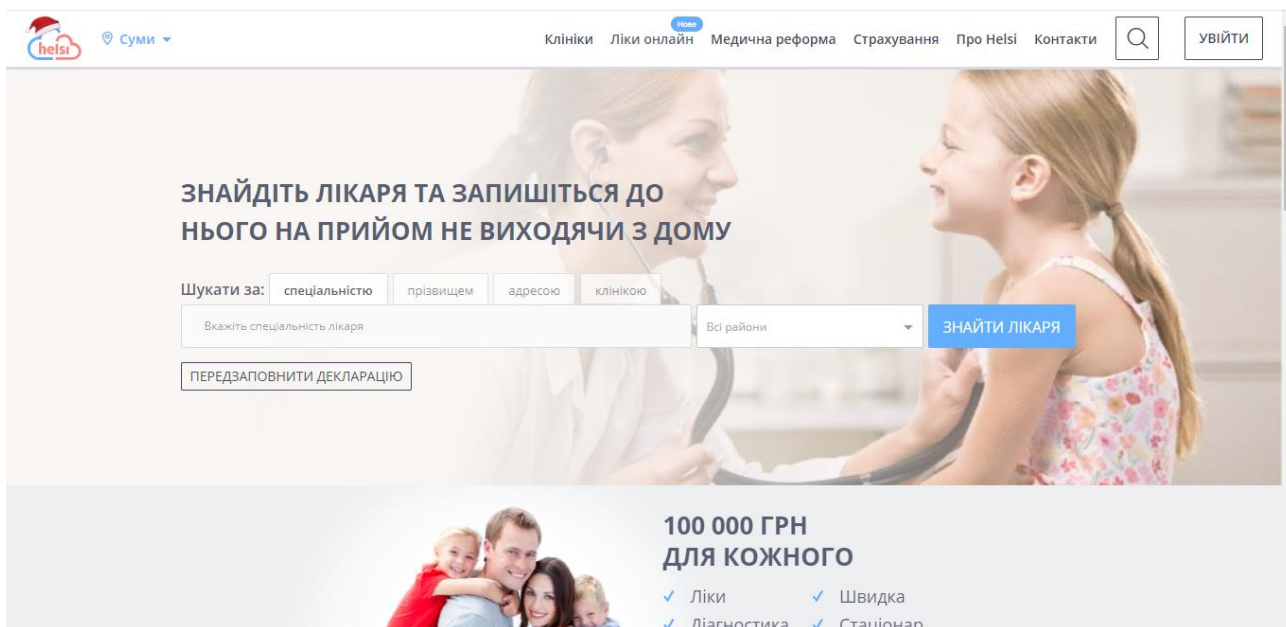


Рисунок 1.3 – Приклад інтерфейсу ІС HELSI.ME

Порівняльну характеристику розглянутих медичних інформаційних систем зображено в таблиці 1.1.

Таблиця 1.1 – Порівняльна характеристика медичних інформаційних систем

Критерій	ОН Клінік	Верба Медікал	HELSI.MI
Запис пацієнтів адміністратором	+	+	+
Запис пацієнтів лікарем	-	-	-
Перегляд послуг	+	-	-
Попередній запис на прийом	+	+	+
Редагування власного графіку прийомів лікарем	-	-	-
Наявність особистого кабінету пацієнта	-	-	+
Наявність особистого кабінету лікаря	-	-	+
Попередній запис на прийом	+	+	+
Редагування власного графіку прийомів лікарем	-	-	-

Отже, виділивши переваги та недоліки кожного з продуктів-аналогів було визначено, що для конкурентоспроможності необхідно розробити інформаційну систему підтримки реєстратури поліклініки, що матиме такі критерії: самостійний запис пацієнтів на прийом, а також запис пацієнтів лікарем або працівником реєстратури, надання призначень пацієнту, робота з розкладом, наявність особистого кабінету пацієнта, наявність особистого кабінету лікаря.

1.3 Постановка задачі

Головною метою проекту є розробка інформаційної системи підтримки діяльності реєстратури поліклініки для автоматизації роботи реєстратури та лікаря, управління розкладом лікаря, доступу пацієнта до його особистих даних (призначень лікаря та плану лікування).

Були визначені такі задачі для досягнення мети:

- визначити та проаналізувати функціональні та нефункціональні вимоги до інформаційної системи та розробити технічне завдання;
- обрати методи та засоби реалізації;
- визначити структуру інформаційної системи та розробити її дизайн;
- розробити модель бази даних;
- здійснити верстку та програмування;
- провести тестування інформаційної системи;
- розробити супровідну документацію.

Використання розробленої інформаційної системи спрямовано на автоматизацію виконуваних бізнес-процесів при роботі поліклініки.

Для детального опису цілей та задач було сформоване та затверджене технічне завдання, яке знаходиться у додатку А.

1.4 Вибір засобів реалізації

Перед початком реалізації інформаційної системи потрібно обрати засоби реалізації, адже існує багато мов програмування, систем управління базами даних (СУБД), web-серверів та фреймворків за допомогою яких можна створити будь-яку інформаційну систему. На основі поставлених мети та задач, описаних вище, оптимальними засобами реалізації є:

1. HTML (HyperText Markup Language) – стандартна мова розмітки гіпертексту в Інтернеті. Її основним призначенням є створення web-сторінок та забезпечення нормального розташування в документі заголовків, списків, картинок, таблиць та інших матеріалів. Крім того, HTML-документи можуть містити додаткові метадані, такі як, автор документу, мова тексту, стислий підсумок.[2]

2. CSS (Cascading Style Sheets) – це формальна мова, що служить для опису оформлення зовнішнього вигляду документа, створеного з використанням мови розмітки (HTML, XHTML, XML). [1]

3. PHP – це серверна мова створення сценаріїв, що призначена для генерування HTML-сторінок на веб-сервері та роботи з базами даних. PHP має багато переваг, серед яких: простота у використанні, висока продуктивність та функціональність. [8]

4. JavaScript (JS) – це повноцінна динамічна мова програмування, яка застосовується до HTML документу, і може забезпечити динамічну

інтерактивність на web-сайтах. JavaScript дозволяє створювати інтерактивні елементи на web-сторінці, обробляти події з цими елементами, створювати автоматизовані програми і т.д. [4] Однією з особливостей JavaScript є об'єктна орієнтованість. Розробнику доступні численні об'єкти, такі, як документи, гіперпосилання, форми, фрейми і т.д.

5. MySQL – це реляційна система управління базами даних з відкритим вихідним кодом. В даний час ця СУБД одна з найбільш популярних в веб-додатках. З переваг СУБД MySQL варто відзначити простоту використання, гнучкість, низьку вартість володіння (щодо платних СУБД), а також масштабованість і продуктивність. Працювати з Mysql можна як в текстовому режимі, так і в графічному. Для цього існує візуальний інтерфейс для роботи з цією СУБД – PhpMyAdmin, що дозволяє значно спростити роботу з базами даних в Mysql.[3]

6. CMS WordPress – це сучасний проект для організації власного інформаційного порталу. Ця система реалізована на мові програмування PHP, а в якості бази даних використовується MySQL. Код WordPress є відкритим, а модулі дозволяють збільшити можливості, закладені в систему. Відкритість платформи дає можливість додавати або змінювати вихідний код, дозволяючи допрацьовувати продукт під вимоги конкретного проекту. [11]

2. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Діаграми у нотаціях IDEF0

Перед тим як почати роботу над розробкою інформаційної системи необхідно правильно та зрозуміло побудувати структурно-функціональну модель процесу, що виконує інформаційна система підтримки діяльності реєстратури поліклініки.

IDEF0 являє собою сукупність методів, правил і процедур, призначених для побудови функціональної моделі системи. [7] Метою даної методології є побудова ієрархічної системи. Спочатку потрібно провести загальний опис системи та описати її взаємодію з навколишнім середовищем (контекстна діаграма) [18].

Для відображення структурно-функціональної моделі процесів було побудовано кілька контекстних діаграм, що відображають процеси забезпечення діяльності лікаря, забезпечення діяльності пацієнта та забезпечення діяльності працівника реєстратури. Для їх побудови було виділено було виділено: вхідні дані, вихідні дані, механізми та керуючі елементи.

Вхідними даними до функції «Забезпечення діяльності лікаря» є запит лікаря.

Вихідними даними є:

- записи на прийом;
- змінений графік лікаря;
- призначення лікаря.

Керуючими елементами процесу в даному випадку є:

- нормативні документи;

- інструкції користувача;

Механізмами є:

- пацієнт;
- база даних;
- працівник реєстратури;
- інформаційна система.

Контекстну діаграму, на якій зображено процес забезпечення діяльності лікаря, зображено на рисунку 2.1.

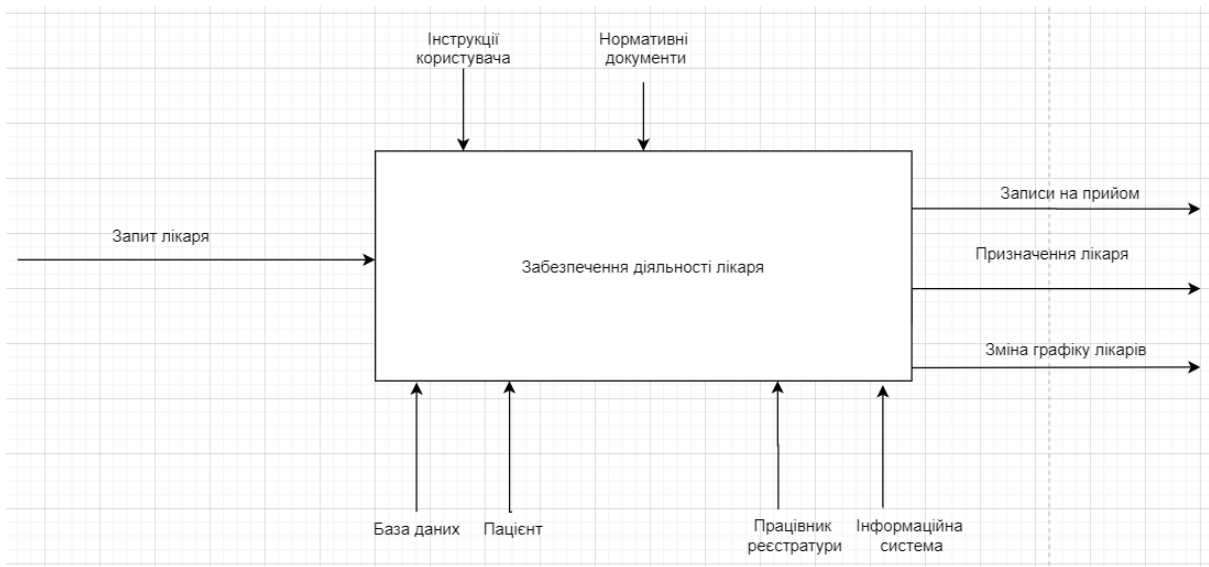


Рисунок 2.1 – Контекстна діаграма процесу забезпечення діяльності лікаря

Вхідними даними до функції «Забезпечення діяльності пацієнта» є запит пацієнта.

Вихідними даними є:

- інформація про успішний запис на прийом;
- призначення лікаря.

Керуючими елементами процесу в даному випадку є:

- нормативні документи;
- інструкції користувача.

Механізмами є:

- база даних;
- інформаційна система;
- лікар.

Контекстну діаграму, на якій зображено процес забезпечення діяльності пацієнта, зображено на рисунку 2.2.

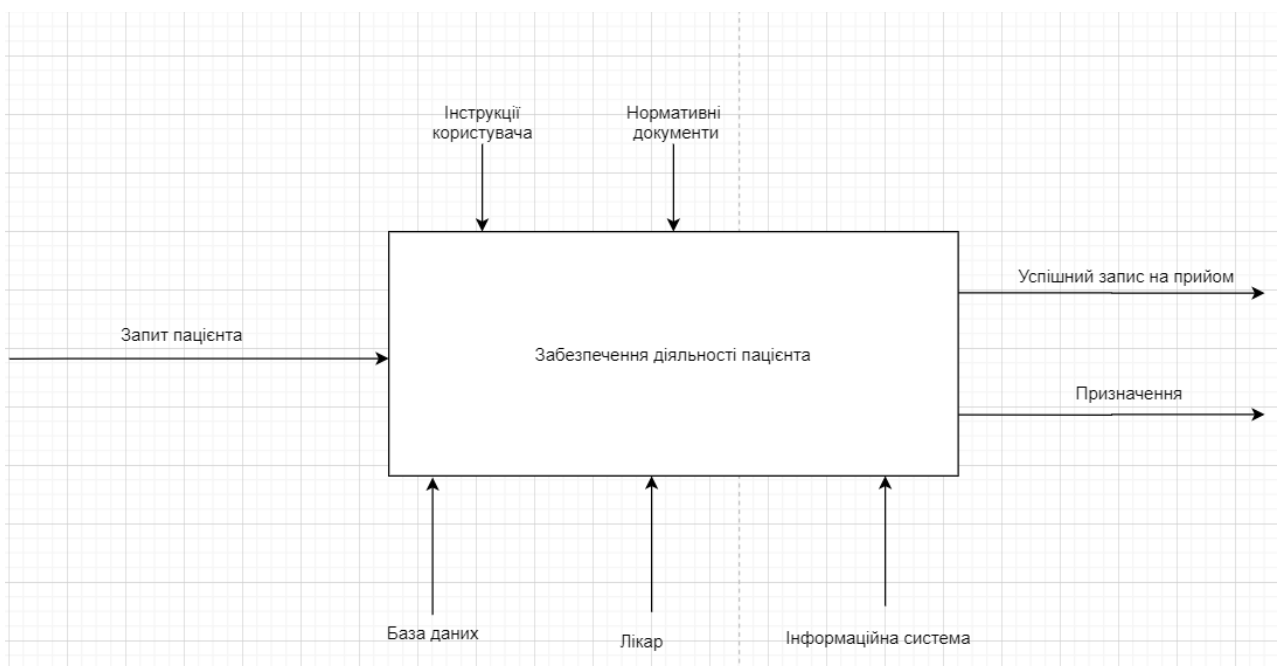


Рисунок 2.2 – Контекстна діаграма процесу забезпечення діяльності пацієнта

Вхідними даними до функції «Забезпечення діяльності працівника реєстратури» є запит працівника реєстратури.

Вихідними даними є:

- записи на прийоми;
- журнал обліку пацієнтів;

- змінений розклад прийомів;

Керуючими елементами процесу в даному випадку є:

- нормативні документи;
- інструкції користувача;

Механізмами є:

- пацієнт;
- лікар;
- база даних;
- інформаційна система.

Контекстну діаграму, на якій зображено процес забезпечення діяльності працівника реєстратури, зображено на рисунку 2.4.



Рисунок 2.3 - Контекстна діаграма процесу забезпечення діяльності працівника реєстратури

Після побудови контекстних діаграм необхідно провести функціональну декомпозицію. Для цього систему потрібно розбити на підсистеми і описати

кожну з них окремо (діаграма декомпозиції). У результаті аналізу бізнес-процесів поліклініки кожну з наведених вище діаграм було декомпововано.

Після декомпозиції процесу забезпечення діяльності лікаря було виділено такі блоки:

- Авторизація;
- Перегляд даних в особистому кабінеті;
- Запис на прийом;
- Надання призначень пацієнту.

Діаграму декомпозиції процесу забезпечення діяльності лікаря зображено на рисунку 2.4.

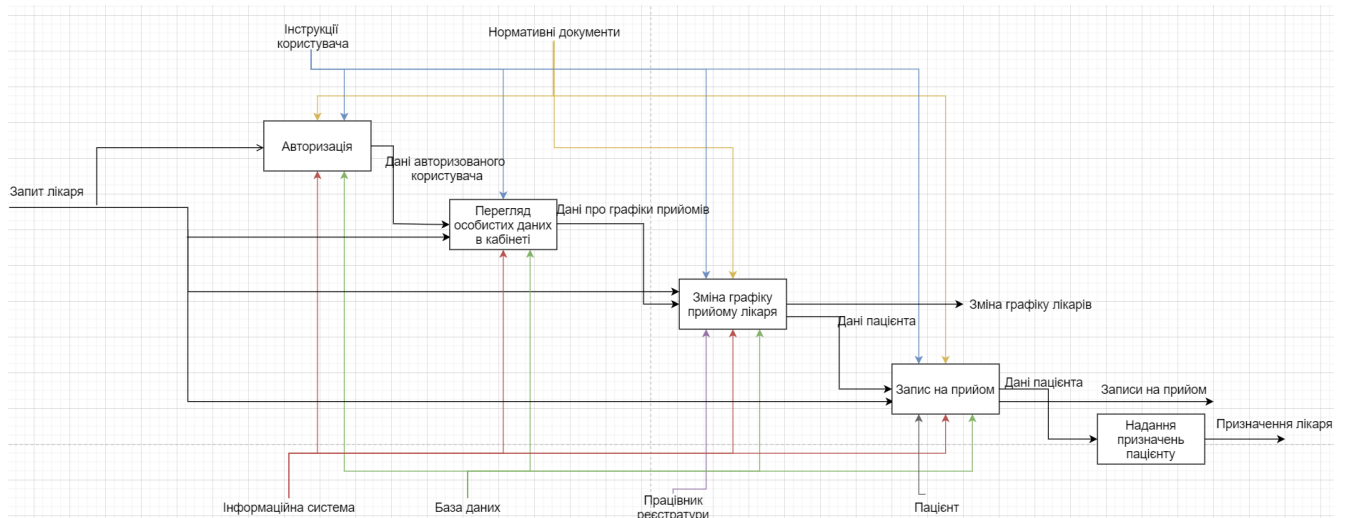


Рисунок 2.4 – Діаграма декомпозиції процесу забезпечення діяльності лікаря

Після декомпозиції процесу забезпечення діяльності пацієнта було виділено такі блоки:

- Авторизація;
- Перегляд даних про послуги та графік прийомів;
- Запис на прийом;

- Перегляд особистих даних та призначення.

Діаграму декомпозиції процесу забезпечення діяльності пацієнта зображено на рисунку 2.5.

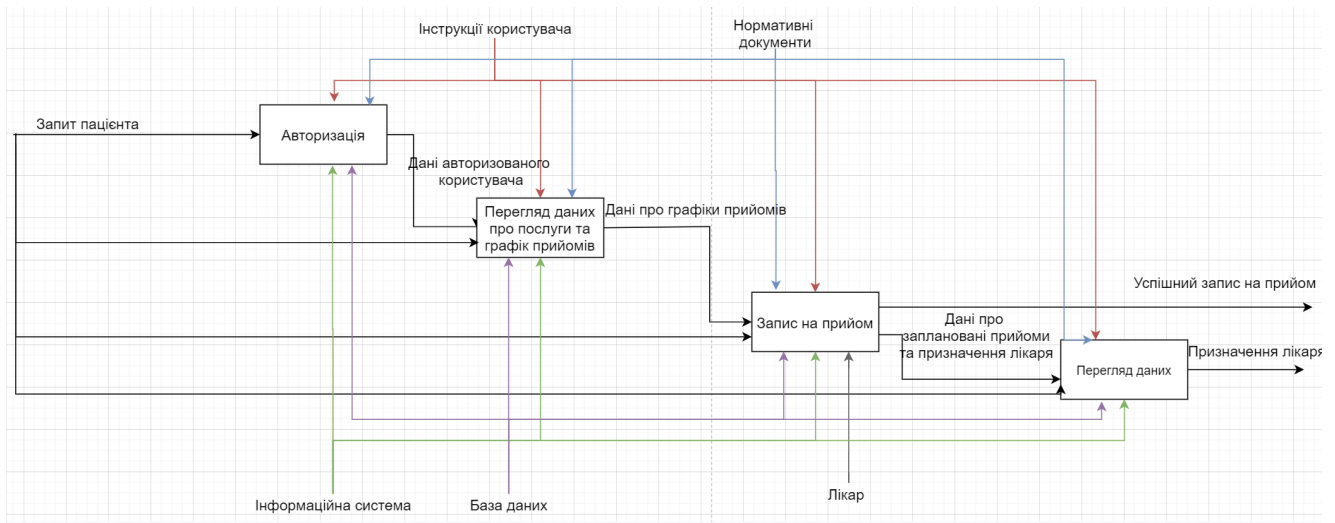


Рисунок 2.5 – Діаграму декомпозиції процесу забезпечення діяльності пацієнта

Після декомпозиції процесу забезпечення діяльності працівника реєстратури було виділено такі блоки:

- Авторизація;
- Управління особистими даними користувачів;
- Зміна графіку прийому лікаря;
- Запис на прийом;

Діаграму декомпозиції процесу забезпечення діяльності працівника реєстратури зображено на рисунку 2.6.

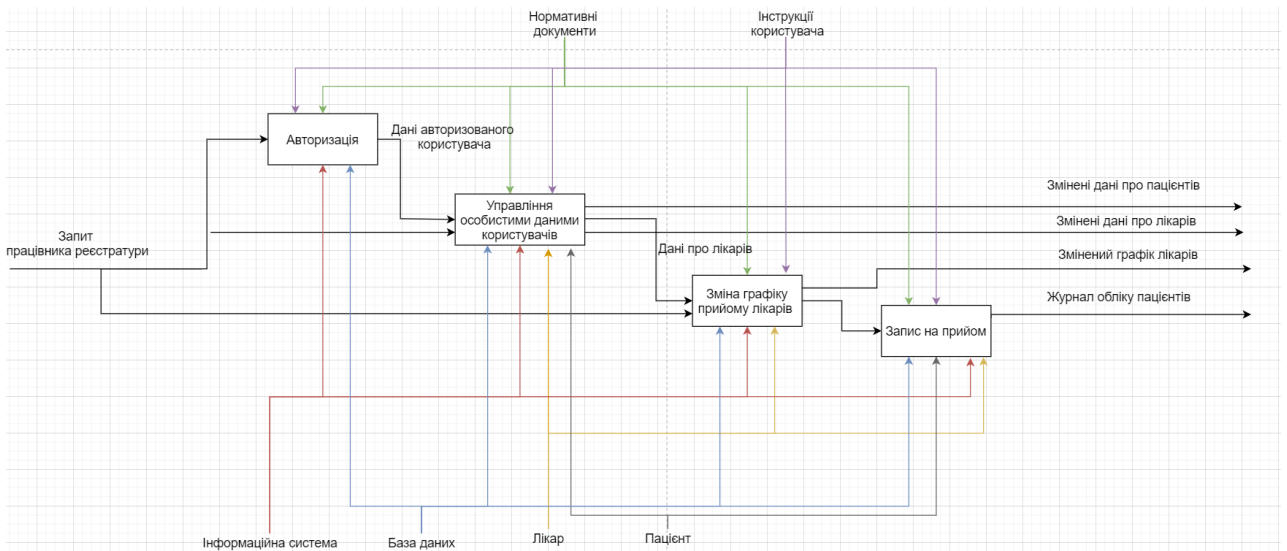


Рисунок 2.6 – Діаграма декомпозиції процесу забезпечення діяльності працівника реєстратури

Далі було проведено декомпозицію процесу запису пацієнта на прийом до лікаря, де було виділено такі блоки:

- Вибір лікаря;
- Заповнення даних для запису;
- Обробка даних ІС;
- Занесення даних до БД;
- Відображення даних про запис на прийом в особистому кабінеті.

Діаграму декомпозиції процесу запису на прийом з точки зору пацієнта зображено на рисунку 2.7.

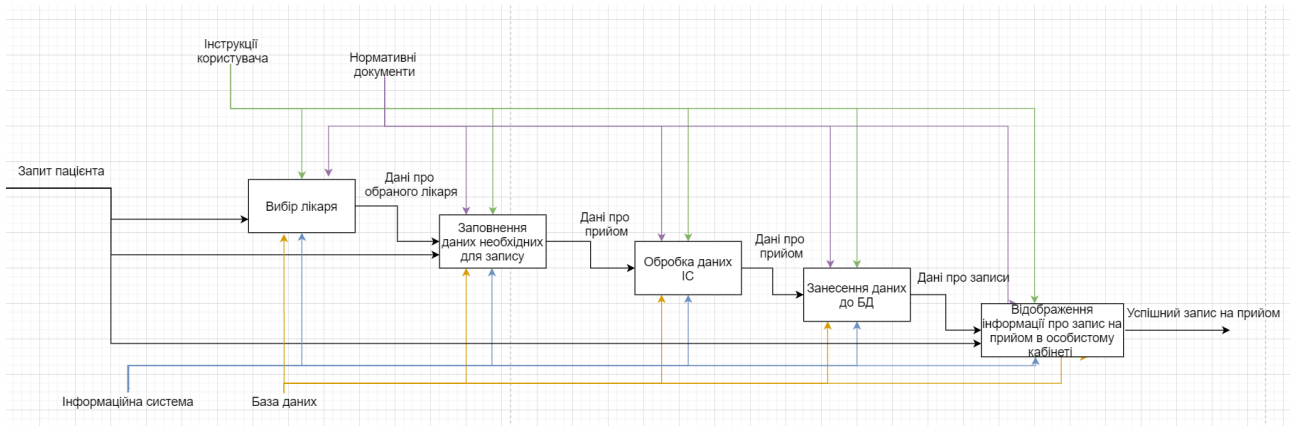


Рисунок 2.7 – Діаграма декомпозиції процесу запису на прийом з точки зору пацієнта

2.2 Моделювання варіантів використання інформаційної системи

Для описання дій, які виконує інформаційна система було створено діаграму варіантів використання. Дана діаграма показує функціональність системи та її взаємодію з користувачами. Діаграма варіантів використання складається з акторів, елементів зовнішнього зв'язку з системою, самих варіантів використання та зв'язків між всіма частинами діаграми. [9] Діаграму варіантів використання зображено на рисунку 2.8.

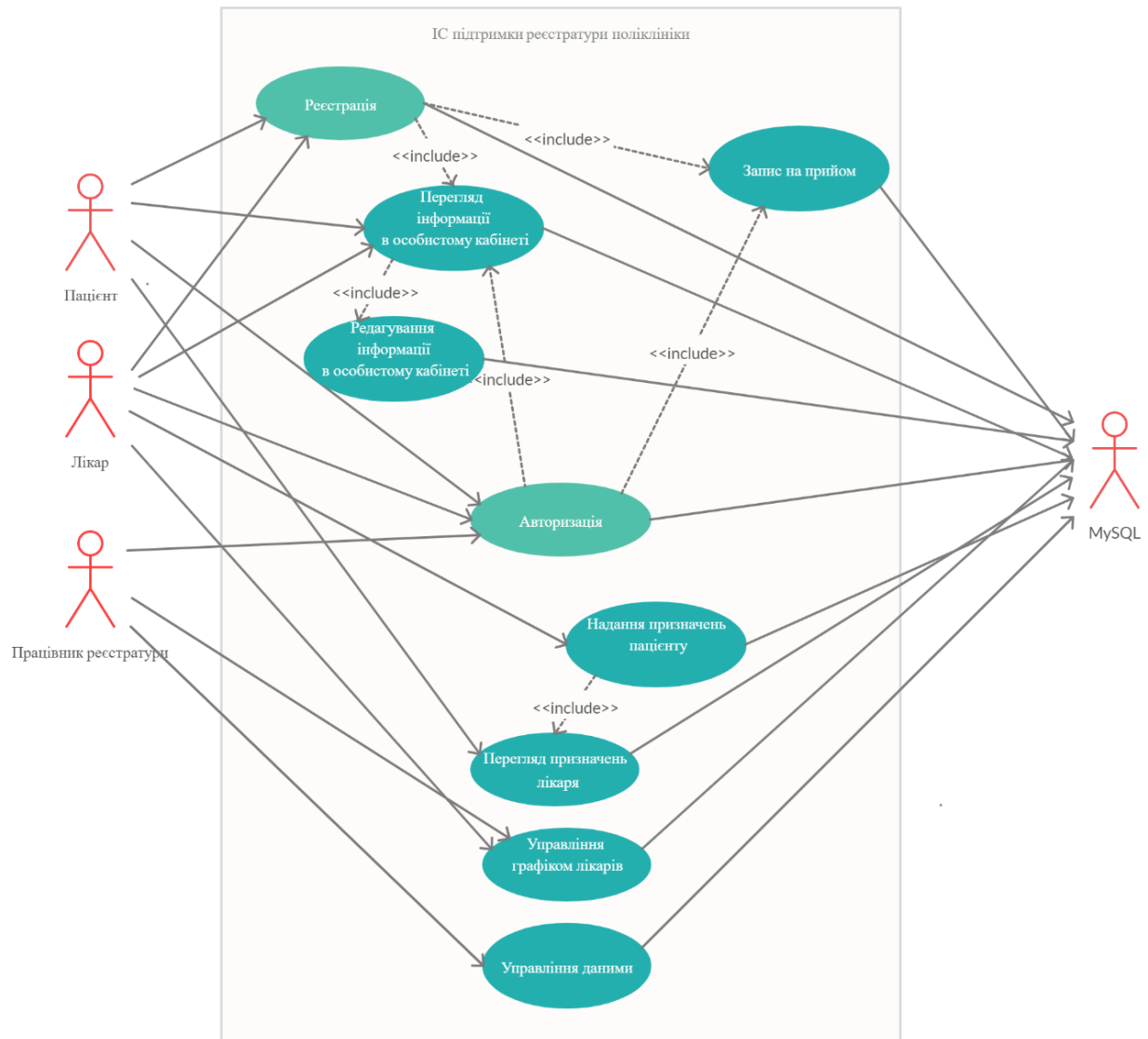


Рисунок 2.8 – Діаграма варіантів використання

Було виділено таких акторів:

- лікар – користувач, який авторизується в системі та має можливість змінювати розклад прийомів та надавати призначення пацієнту;
- пацієнт – користувач, який авторизується в системі, має можливість записуватися на прийом до обраного лікаря на обраний час та має доступ до особистого кабінету де може переглядати свої записи на прийом а також має доступ до призначень, наданих йому лікарем;

– працівник реєстрації – користувач, який авторизується в системі та має можливість змінювати розклад прийомів, записувати пацієнта на прийом, управляти даними про пацієнтів, лікарів;

– Mysql – сховище даних, зберігає дані, що використовуються або що додаються впродовж роботи з інформаційною системою.

Варіанти використання:

– ВВ реєстрація – надає можливість користувачам зареєструватися в системі для подальшої роботи з нею;

– ВВ авторизація – надає можливість користувачам працювати з системою відповідно до наданих їм прав доступу;

– ВВ редагування інформації в особистому кабінеті – надає можливість користувачам редагувати їх власні дані, для входу в систему, а саме пароль, ім'я користувача та адресу електронної пошти;

– ВВ запис на прийом – надає можливість пацієнту записатися на прийом або здійснити запис пацієнта працівнику реєстрації;

– ВВ управління графіком лікарів – надає можливість працівнику реєстрації та лікарю змінювати графік роботи лікарів ;

– ВВ надання призначень пацієнту – надає можливість лікарю описувати хворобу та метод лікування пацієнта;

– ВВ надання перегляд призначень лікаря – надає можливість пацієнту переглядати призначення, діагнози та методи лікування;

– ВВ перегляд інформації в особистому кабінеті – надає можливість пацієнту та лікарю переглядати особисту інформацію. Для пацієнта інформація про заплановані прийоми. Для лікаря та працівника реєстрації графіки прийомів;

– ВВ управління даними – надає можливість працівнику реєстратури додавати, видаляти та редагувати інформацію про пацієнтів, лікарів та контент, що містить інформаційна система;

2.3 Проектування моделі бази даних

Важливим компонентом будь-якої інформаційної системи є база даних. Моделлю бази даних називають графічне або текстове представлення бази даних, де виділяються сутності, атрибути та зв'язки між ними. [20]

Згідно з побудованою моделлю варіантів використання дана інформаційна система повинна зберігати всю інформацію в базі даних (БД). Оскільки інформаційна система розроблялася за допомогою CMS Wordpress, то для її функціонування використовується дві бази даних: база даних Wordpress та власно створена база даних.

База даних стандартної установки Wordpress складається з 11 таблиць, необхідних для функціонування розроблюваного програмного продукту [13]. Модель даної бази даних відображено на рисунку 2.9.

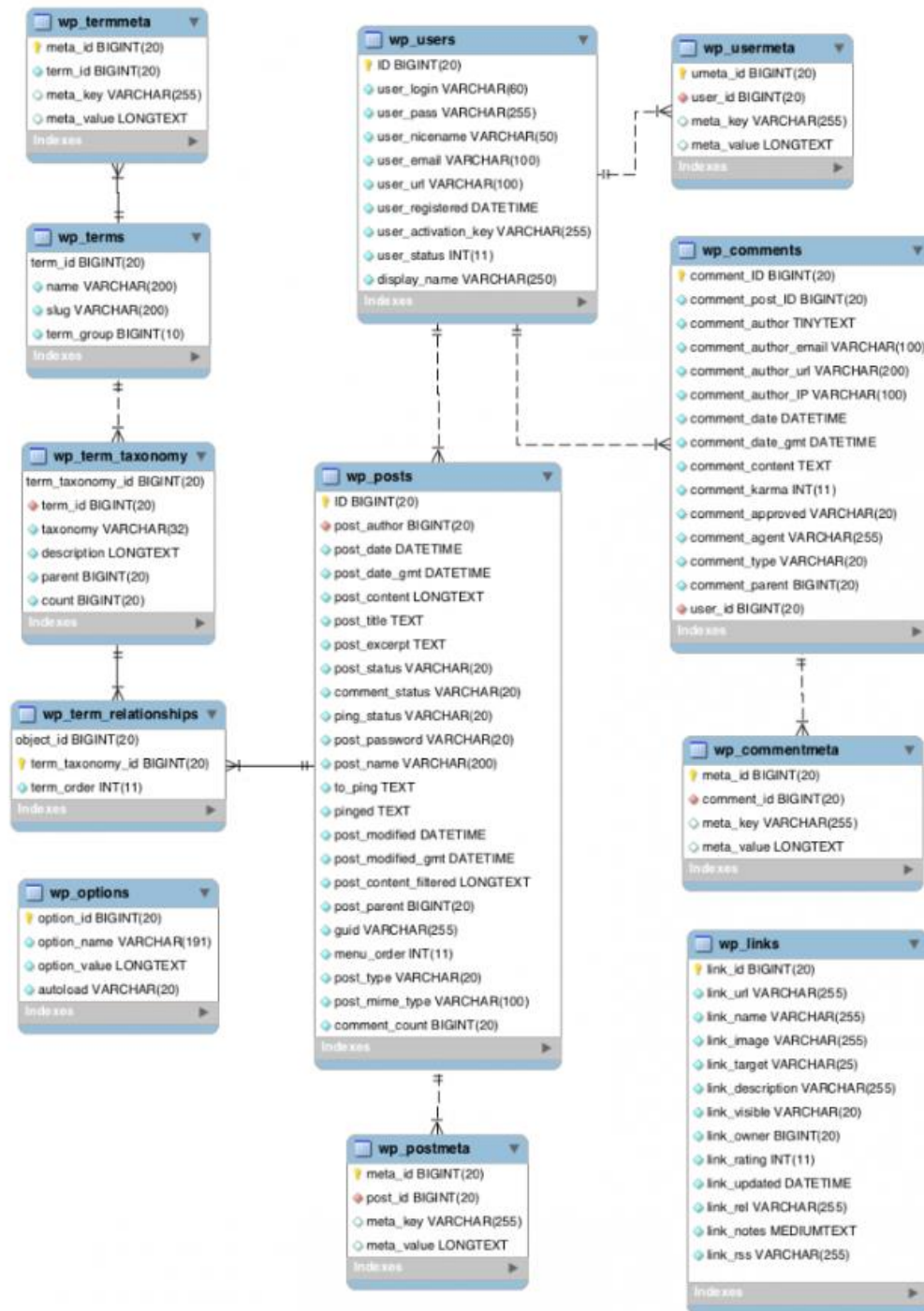


Рисунок 2.10 – Стандартна база даних CMS Wordpress

wp_commentmeta – дана таблиця зберігає метадані всіх коментарів на сторінках і в записах.

`wp_comments` – дана таблиця зберігає всі коментарі, включаючи опубліковані, схвалені, спам та ті, що очікують перевірки.

`wp_links` – дана таблиця зберігає інформацію про посилання в менеджері посилань.

`wp_options` – дана таблиця зберігає налаштування. Також дана таблиця зберігає дані з встановлених тем та плагінів.

`wp_postmeta` – дана таблиця зберігає метадані постів та сторінок.

`wp_posts` – дана таблиця зберігає контент постів, сторінок, також дані меню і навігації.

`wp_terms` – дана таблиця зберігає категорії постів, посилань та тегів.

`wp_term_relationships` – дана таблиця зберігає дані пов'язані з категоріями і тегами в таблиці `wp_terms`.

`wp_term_taxonomy` – дана таблиця зберігає таксономію категорій, посилань і тегів для записів в таблиці `wp_terms`.

`wp_usermeta` – дана таблиця зберігає метадані всіх користувачів з таблиці `wp_users`.

`wp_users` – дана таблиця зберігає дані про всіх користувачів. [12]

Для додавання власного функціоналу до інформаційної системи було також побудовано базу даних. [15] Модель розробленої бази даних відображено на рисунку 2.11.

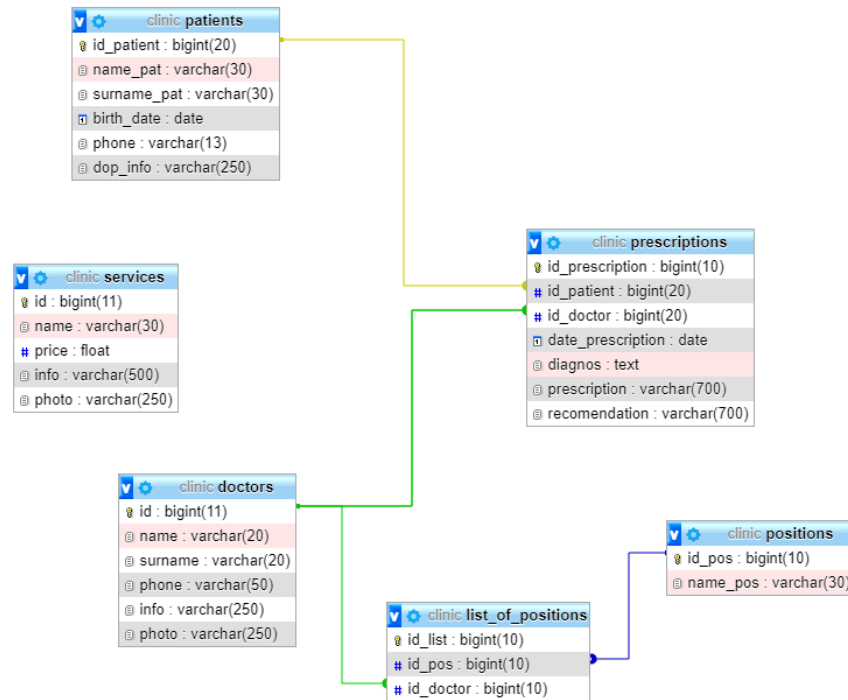


Рисунок 2.11 – Модель бази даних

Дана база даних містить 5 таблиць, що мають певні атрибути:

- **Services** – таблиця, що зберігає інформацію про послуги, які надає поліклініка. Таблиця містить такі атрибути: `id` (ідентифікатор послуги), `name` (назва послуги), `price` (ціна за послугу), `info` (додаткова інформація про послугу) та `photo` (фото, що відображається на відповідних сторінках);
- **Doctors** – таблиця, що зберігає інформацію про лікарів поліклініки. Таблиця містить такі атрибути: `id` (ідентифікатор лікаря), `name` (ім'я лікаря), `surname` (прізвище лікаря), `phone` (номер телефону лікаря), `info` (додаткова інформація лікаря), `photo` (фото лікаря);
- **Positions** – таблиця, яка зберігає спеціальності, що присутні в поліклініці. Таблиця містить такі атрибути: `id_pos` (ідентифікатор спеціальності), `name_pos` (назва спеціальності);

– List_of_positions – таблиця, яка зберігає інформацію про спеціальність кожного з лікарів. Таблиця містить такі атрибути: id_list (ідентифікатор спеціальності лікаря), id_pos (ідентифікатор спеціальності), id_doctor (ідентифікатор лікаря);

– Patients – таблиця, яка зберігає інформацію про пацієнтів. Таблиця містить такі атрибути: id_patient (ідентифікатор пацієнта), name_pat (ім'я пацієнта), surname_pat (прізвище пацієнта), birth_date (дата народження пацієнта), phone (номер телефону пацієнта), dor_info (додаткова інформація про пацієнта);

– Prescriptions – таблиця, яка зберігає інформацію про призначення пацієнтів. Таблиця містить такі атрибути: id_prescription (ідентифікатор призначення), id_doctor (ідентифікатор лікаря, що надав призначення), id_patient (ідентифікатор пацієнта), date_prescription (дата призначення), diagnos (діагноз), recommendation (рекомендація, щодо лікування), prescription (призначення для лікування).

3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Архітектура інформаційної системи

Архітектура інформаційної системи являє собою концепцію, що визначає структуру і взаємозв'язки між компонентами інформаційної системи. [16]

Для реалізації інформаційної системи було обрано CMS Wordpress. Вона містить такі компоненти: плагіни, шаблони, теми, сторінки та базу даних. [17]

Архітектура інформаційної системи відображена на рисунку 3.1.

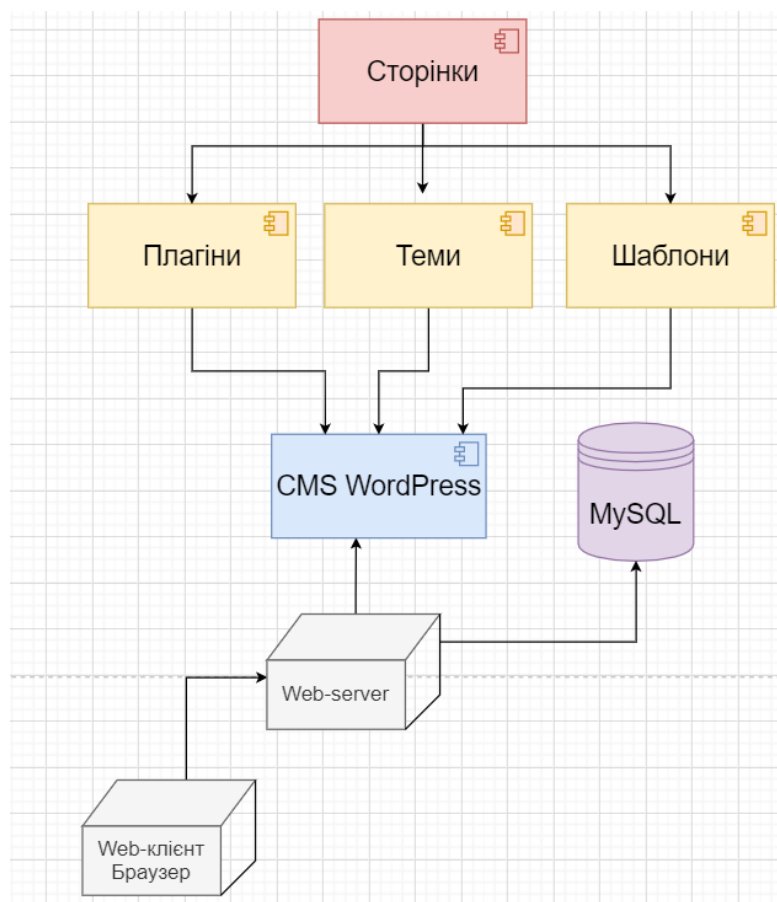


Рисунок 3.1 – Архітектура інформаційної системи

Шаблони та теми представляють собою набір файлів, які відповідають за зовнішній вигляд розроблювального проекту. Вони зберігаються в папці `wp-content/themes`

Плагіни є доповненнями, які розширюють функціональність і допомагають у розробці повноцінного проекту. Плагіни зберігаються у папці `wp-content/plugins`. [19]

В базі даних зберігається вся інформація, що представлена в інформаційній системі, а саме інформація про користувачі, різні налаштування, інформація про послуги, записи на прийом, призначення.

Сторінки є статичними елементами на яких розміщується вся інформація для відображення користувачам.

Відображення вмісту файлів проекту зображено на рисунку 3.2

Имя	Дата изменения	Тип	Размер
wp-admin	07.05.2020 17:58	Папка с файлами	
wp-content	28.05.2020 11:43	Папка с файлами	
wp-includes	07.05.2020 17:58	Папка с файлами	
.htaccess	07.05.2020 18:25	Файл "HTACCESS"	1 КБ
index.php	06.02.2020 08:33	Файл "PHP"	1 КБ
license.txt	29.04.2020 23:19	Текстовый докум...	20 КБ
readme.html	29.04.2020 23:19	Chrome HTML Do...	8 КБ
wp-activate.php	06.02.2020 08:33	Файл "PHP"	7 КБ
wp-blog-header.php	06.02.2020 08:33	Файл "PHP"	1 КБ
wp-comments-post.php	06.02.2020 08:33	Файл "PHP"	3 КБ
wp-config.php	15.05.2020 11:11	Файл "PHP"	4 КБ
wp-config-sample.php	01.04.2020 00:06	Файл "PHP"	3 КБ
wp-cron.php	06.02.2020 08:33	Файл "PHP"	4 КБ
wp-links-opml.php	06.02.2020 08:33	Файл "PHP"	3 КБ
wp-load.php	06.02.2020 08:33	Файл "PHP"	4 КБ
wp-login.php	10.02.2020 05:50	Файл "PHP"	47 КБ
wp-mail.php	29.04.2020 23:19	Файл "PHP"	9 КБ
wp-settings.php	29.04.2020 23:19	Файл "PHP"	19 КБ
wp-signup.php	06.02.2020 08:33	Файл "PHP"	31 КБ
wp-trackback.php	06.02.2020 08:33	Файл "PHP"	5 КБ
xmlrpc.php	06.02.2020 08:33	Файл "PHP"	4 КБ

Рисунок 3.2 – Вміст файлів проекту

3.2 Програмна реалізація

Обрана CMS Wordpress є зручною у використанні, так як містить не лише велику кількість вже реалізованих плагінів, тем та шаблонів, але й дозволяє створювати власні.

Оскільки Wordpress є системою керування вмістом з відкритим кодом, то було створено власну базу даних та написано реалізацію для таких компонентів як перегляд інформації про послуги та лікарів на відповідних сторінках, надання призначень пацієнту, їх редагування, видалення та перегляд пацієнтами, управління інформацією про послуги, лікарів та їх спеціальностями, пацієнтів. Крім цього було використано вже реалізовані плагіни Booked, Jetpack, theme-my-login. Також було додано код для додавання полів до форми реєстрації користувачів.

Розробку інформаційної системи підтримки діяльності реєстратури поліклініки було розпочато з розробки її дизайну. Після цього було створено власну базу даних (clinic). Розроблена база даних зберігає інформацію про послуги, лікарів, спеціальності лікарів та призначення пацієнтів. Приклад коду створення та заповнення даними таблиці з спеціальностями відображено на рисунку 3.3.

```

--
CREATE TABLE `positions` (
  `id_pos` bigint(10) NOT NULL,
  `name_pos` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Dumping data for table `positions`
--

INSERT INTO `positions` (`id_pos`, `name_pos`) VALUES
(1, 'Травматолог'),
(3, 'Дерматолог'),
(5, 'Офтальмолог');

--
ALTER TABLE `positions`
  ADD PRIMARY KEY (`id_pos`);

ALTER TABLE `positions`
  MODIFY `id_pos` bigint(10) NOT NULL AUTO_INCREMENT;
COMMIT;

```

Рисунок 3.3 – Приклад коду створення таблиці бази даних

Після написання коду для створення всіх таблиць було написано код для підключення до бази даних (dbcontroller.php). Код підключення до бази даних зображено на рисунку 3.4.

```

private $host = "localhost";
private $user = "root";
private $password = "";
private $database = "clinic";
private $conn;

function __construct() {
    $this->conn = $this->connectDB();
}

```

Рисунок 3.4 – Фрагмент коду для підключення до бази даних

Далі за допомогою HTML, CSS, PHP та JS було написано код для головної сторінки (mainpage.php). Фрагмент коду головної сторінки, що відповідає за відображення блоку про послуги, зображено на рисунку 3.5.

```

<body class="landing">
<h1 style = " padding: 10px;"></h1>

    <section id="one" class="wrapper style1">
        <div class="container1">
            <h1 style = " padding: 10px;"><span>Головна сторінка</span></a></h1>

                <div class="slide">
                    
                </div>
            <h1></h1>
        </div>

<center><h2><span style = " padding: 30px; ">Послуги</span></a></h2></center>
<!-- Two -->
<?php
require_once("perpage.php");
require_once("dbcontroller.php");
$db_handle = new DBController();
?> <hr>
<div id="gridview">
    <div class="header"></div>
<?php
$query = $db_handle->runQuery("SELECT * FROM services ORDER BY rand() limit 6 ");
if (! empty($query)) {
    foreach ($query as $key => $value) {
        <div class="image" width = "100%">
            " />
            <?php echo $query[$key]["name"];?>
        </div>
    }
}
?>
...

```

Рисунок 3.5 – Фрагмент коду головної сторінки

На наступному кроці було створено сторінки, що відображають перелік всіх послуг (allservices.php) та лікарів (all_doctors_php) поліклініки, а також сторінки для детальної інформації про лікаря (doctor.php) чи послугу (service.php) . На даних сторінках крім виведення всіх лікарів або послуг було реалізовано функцію пошуку та пагінацію. Фрагмент коду сторінки про лікарів зображено на рисунку 3.6.

```

}
$orderby = " ORDER BY id desc";
$sql = "SELECT * FROM list_of_positions l
join doctors d
on d.id = l.id_doctor
join positions p
on p.id_pos = l.id_pos" . $queryCondition;
$href = 'doctor_page.php';

$perPage = 9;
$page = 1;
if(isset($_POST['page'])){
    $page = $_POST['page'];
}
$start = ($page-1)*$perPage;
if($start < 0) $start = 0;

$query = $sql . $orderby . " limit " . $start . ", " . $perPage;
$result = $db_handle->runQuery($query);

if(!empty($result)) {
    $result["perpage"] = showperpage($sql, $perPage, $href);
}
}
}

<form style = " padding: 10px; "name="frmSearch" method="post" action="<?php echo get_stylesheet_directory_uri() ?>/allservices.php">
<p><input type="text" placeholder="Назва" name="search[surname]" class="demoInputBox" value="<?php echo $surname; ?>" /><input type="text" placeholder="Спеціаліст"
<?php
if(!empty($result)) {
    foreach($result as $k=>$v) {
        if(is_numeric($k)) {
            <div style = " padding: 10px;" class="image">
<?php echo $result[$k]["name_pos"];?>

```

Рисунок 3.6 – Фрагмент коду сторінки лікарів

Після цього було розроблено функціонал, що дозволяє надавати призначення пацієнтам. Приклад відображення сторінки з інформацією про призначення пацієнтам та додаткової інформації про обраного пацієнта (pres_view.php) зображено на рисунку 3.7.

Головна Про нас Лікарі Перегляд послуг Онлайн-запис Особистий кабінет Призначення Вийти Q

Призначення

Додати нове

ПІБ	Дата призначення	Лікар	Призначення	Діагноз	Рекомендації	Дія
Аліна Семенова	2020-05-26	Юрій Пашко	Напівжорсткий ортез на попереково-крижовий відділ хребта	Остеохондроз поперекового відділу хребта	спостереження у травматолога	Редагувати Видалити

Додаткова інформація про обраного пацієнта

Дата народження: 1998-05-10

Додаткова інформація: Місцева алергія на анальгін

Номер телефону: 380958264984

Рисунок 3.7 – Перегляд призначень пацієнта

Було реалізовано функції, що дозволяють додавати нові призначення, видаляти та редагувати власні призначення. Фрагмент коду, що відповідає за додавання нових призначень зображено на рисунку 3.8.

```

require_once("dbcontroller.php");
$db_handle = new DBController();
if(!empty($_POST["submit"])) {
    $cur_user_id = get_current_user_id();

    $query = "INSERT INTO prescriptions (id_doctor, date_prescription, id_patient, prescription, diagnos, recommendation) VALUES ('$cur_user_id', sysdate(), '".$_POST["id_patient"]."'";
    $result = $db_handle->executeQuery($query);
    if(!$result){
        $message="Дані не додано";
    } else {
        require("presc_view.php");
        exit;
    }
}

```

Рисунок 3.8 – Фрагмент коду для додавання нового призначення до бази даних

Для надання можливості працівнику реєстратури управління даними було розроблено функції для редагування, видалення та додавання інформації про послуги, лікарів, спеціальності та пацієнтів. Фрагмент коду, що відповідає за редагування інформації про пацієнтів (edit_doctor.php) зображено на рисунку 3.9.

```

require_once("dbcontroller.php");
$db_handle = new DBController();
if(!empty($_POST["submit"])) {
    $query = "UPDATE patients set surname_pat = '".$_POST["surname"]."', name_pat = '".$_POST["name"]."',
    phone = '".$_POST["phone"]."', dop_info = '".$_POST["info"]."', birth_date= '".$_POST["photo"]."'
    WHERE id_patient='".$_GET["id_patient"]";
    $result = $db_handle->executeQuery($query);
    if(!$result){
        $message = "Проблеми з редагуванням!";
    } else {
        require("patients_view.php");
        exit;
    }
}
$result = $db_handle->runQuery("SELECT * FROM patients WHERE id_patient = '" . $_GET["id_patient"] . "'");
?>

```

Рисунок 3.8 – Фрагмент коду для оновлення інформації про пацієнтів

Фрагмент коду, що відповідає за видалення інформації про спеціальності лікарів (delete_doctorpro.php) зображено на рисунку 3.9.

```

1 <?php
2     require_once("dbcontroller.php");
3     $db_handle = new DBController();
4     if(!empty($_GET["id_list"])) {
5         $query = "DELETE FROM list_of_positions WHERE id_list=".$_GET["id_list"];
6         $result = $db_handle->executeQuery($query);
7         if(!empty($result)){
8             require("doctorpos_view.php");
9             exit;
10        }
11    }
12 >?

```

Рисунок 3.9 – Фрагмент коду видалення інформації про спеціальності лікарів

Для пацієнтів було розроблено сторінку на якій вони можуть переглядати інформацію про всі призначення (user_prescription.php), наданих лікарем, переглядати та змінювати додаткову інформацію про себе (editforpatient.php). Фрагмент коду, що відповідає за виведення інформації про призначення пацієнта з бази даних зображено на рисунку 3.10.

```

echo "<h1>Медична карта: $cur_user_name</h1>";
$sql = mysqli_query($link, "select *
from prescriptions
join doctors
on prescriptions.id_doctor = doctors.id
join patients
on prescriptions.id_patient = patients.id_patient and patients.id_patient = '$cur_user_id'");
?>
<?php
while ($result = mysqli_fetch_array($sql)) {
    echo
        "<tr> .
        "<td>{$result['date_prescription']}</td>" .
        "<td>{$result['diagnos']}</td>" .
        "<td>{$result['name']} {$result['surname']}</td>" .
        "<td>{$result['prescription']}</td>" .
        "<td>{$result['recomendation']}</td>" .
        "</tr>";
}
?>
</table>

```

Рисунок 3.10 – Фрагмент коду виведення інформації про призначення пацієнта з бази даних

Для авторизації та реєстрації користувачів у системі було використано плагін theme-my-login. Вміст файлів плагіну зображено на рисунках 3.11-3.12.

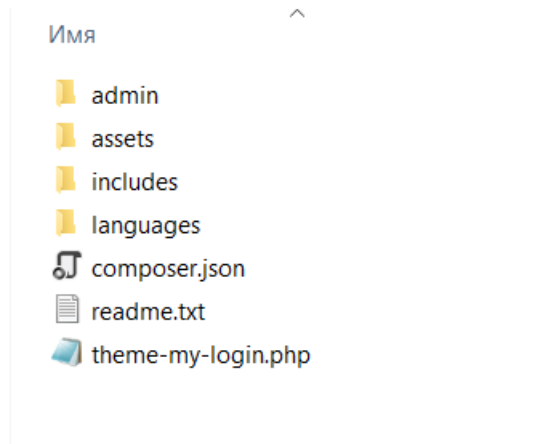


Рисунок 3.11 – Вміст файлу плагіну

Також для додавання полів до форми реєстрації було додано код. Фрагмент коду для додавання поля номер телефону зображено на рисунку 3.12.

```

<?php
add_action('register_form','show_fields');
add_action('register_post','check_fields',10,3);
add_action('user_register','register_fields');

function show_fields() {
    ?>
<p>
    <label>Номер телефону<br/>
    <input id="mobile" class="input" size = "150%" type="text" value="<?php echo $_POST['mobile']; ?>" name="mobile" /></label>
</p>
<?php }

function check_fields ( $login, $email, $errors ) {
    global $mobile;

    if ($_POST['mobile'] == ''){
        $errors->add( 'empty_realname', "Помилка: помилка у введенні номеру телефону" );
    } else {
        $mobile = $_POST['mobile'];
    }
    return $errors;
}

function register_fields($user_id,$password= "",$meta=array()){
    update_user_meta( $user_id, 'booked_phone', $_POST['mobile'] );
}
?>

```

Рисунок 3.12 – Фрагмент коду додавання нового поля до форми реєстрації

Фрагмент коду, що відповідає за відображення полів на формі авторизації зображено на рисунку 3.13.

```

        <p class="login-info">
            <?php _e( 'Введіть Вашу email-адресу', 'personalize-login' ); ?>
        </p>
    <?php endif; ?>

    <?php if ( $attributes['password_updated'] ) : ?>
        <p class="login-info">
            <?php _e( 'Ваш пароль успішно змінено', 'personalize-login' ); ?>
        </p>
    <?php endif; ?>

    <?php
        wp_login_form(
            array(
                'label_username' => __( 'Email', 'personalize-login' ),
                'label_log_in' => __( 'Вхід', 'personalize-login' ),
                'redirect' => $attributes['redirect'],
            )
        );
    ?>

    <a class="forgot-password" href="<?php echo wp_lostpassword_url(); ?>">
        <?php _e( '', 'personalize-login' ); ?>
    </a>

</div>
<?php else : ?>
    <div class="login-form-container">
        <form method="post" action="<?php echo wp_login_url(); ?>">
            <p class="login-username">
                <label for="user_login"><?php _e( 'Email', 'personalize-login' ); ?></label>
                <input type="text" name="log" id="user_login">
            </p>
            <p class="login-password">
                <label for="user_pass"><?php _e( 'Password', 'personalize-login' ); ?></label>
                <input type="password" name="pwd" id="user_pass">
            </p>
            <p class="login-submit">
                <input type="submit" value="<?php _e( 'Sign In', 'personalize-login' ); ?>">
            </p>
        </form>
    </div>
<?php endif; ?>

```

Рисунок 3.13 – Фрагмент коду форми авторизації

Для реалізації функціоналу запису на прийом та управління розкладом лікарів було використано плагін Booked та його додаток Booked Add-On: Front-End Agents, які для зручності використання було перекладено на українську мову. Вміст файлів даного плагіну зображено на рисунку 3.14.

ГОТ КОМПЬЮТЕР > Локальный диск (C:) > хampp > htdocs > site > wp-content > plugins > bc

Имя	Дата изменения	Тип
assets	25.02.2020 22:28	Папка с файлами
includes	25.02.2020 22:28	Папка с файлами
languages	25.02.2020 22:28	Папка с файлами
post-types	25.02.2020 22:28	Папка с файлами
templates	25.02.2020 22:28	Папка с файлами
booked.php	16.05.2020 20:56	Файл "PHP"
class-tgm-plugin-activation.php	05.01.2018 22:38	Файл "PHP"
readme.txt	10.05.2020 12:23	Текстовый докум...
wpml-config.xml	26.12.2017 23:20	Документ XML

Рисунок 3.14 – Вміст файлів плагіну Booked

Також було використано плагін Jetpack, що елементом безпеки та який захищає розроблений продукт від атак методом перебору і несанкціонованого входу. Вміст файлів даного плагіну зображено на рисунку 3.15.

Этот компьютер > Локальный диск (C:) > хampp > htdocs > site > wp-content > plugins > jetpack >

<ul style="list-style-type: none"> assets includes languages post-types templates booked.php class-tgm-plugin-activation.php readme.txt wpml-config.xml 	<ul style="list-style-type: none"> 3rd-party extensions languages src .svnignore class.jetpack.php class.jetpack-autoupdate.php class.jetpack-client-server.php class.jetpack-debugger.php class.jetpack-heartbeat.php class.jetpack-modules-list-table.php class.jetpack-plan.php class.jetpack-user-agent.php class.json-api-endpoints.php composer.json functions.gallery.php functions.photon.php json-api-config.php locales.php uninstall.php 	<ul style="list-style-type: none"> bin images modules vendor changelog.txt class.jetpack-admin.php class.jetpack-bbpress-json-api.compat.php class.jetpack-connection-banner.php class.jetpack-error.php class.jetpack-idc.php class.jetpack-network.php class.jetpack-post-images.php class.jetpack-xmlrpc-server.php class.photon.php functions.compat.php functions.global.php jest.config.js json-endpoints.php readme.txt wpml-config.xml
--	---	--

Рисунок 3.15 – Вміст файлів плагіну Jetpack

3.3 Використання інформаційної системи

При завантаженні інформаційної системи перед неавторизованим користувачем з'являється головна сторінка, фрагмент якої зображено на рисунку 3.16

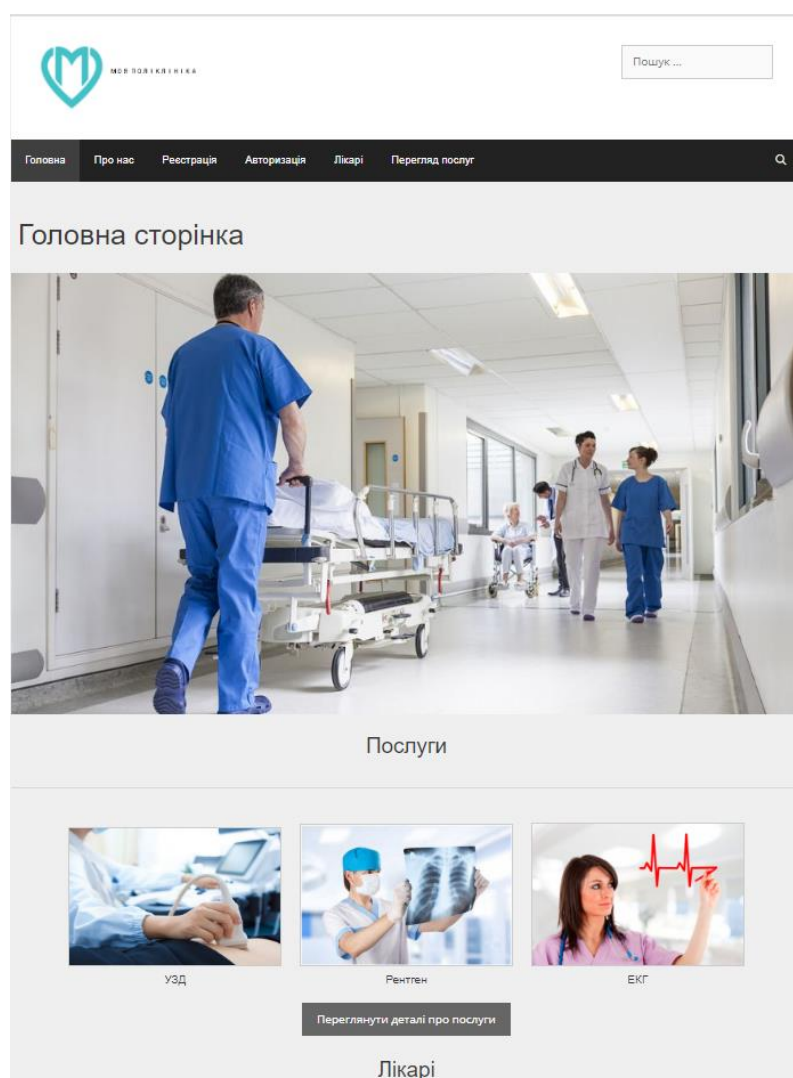


Рисунок 3.16 – Головна сторінка

Неавторизований користувач має змогу перегляду послуг та лікарів поліклініки та інформацію з контактами поліклініки. Для цього необхідно

перейти натиснувши на пункт меню «Лікарі» або «Перегляд послуг» та «Про нас». Для зручності користувача, на даних сторінках реалізовано функцію пошуку. Сторінку з переліком всіх лікарів відображено на рисунку 3.17.

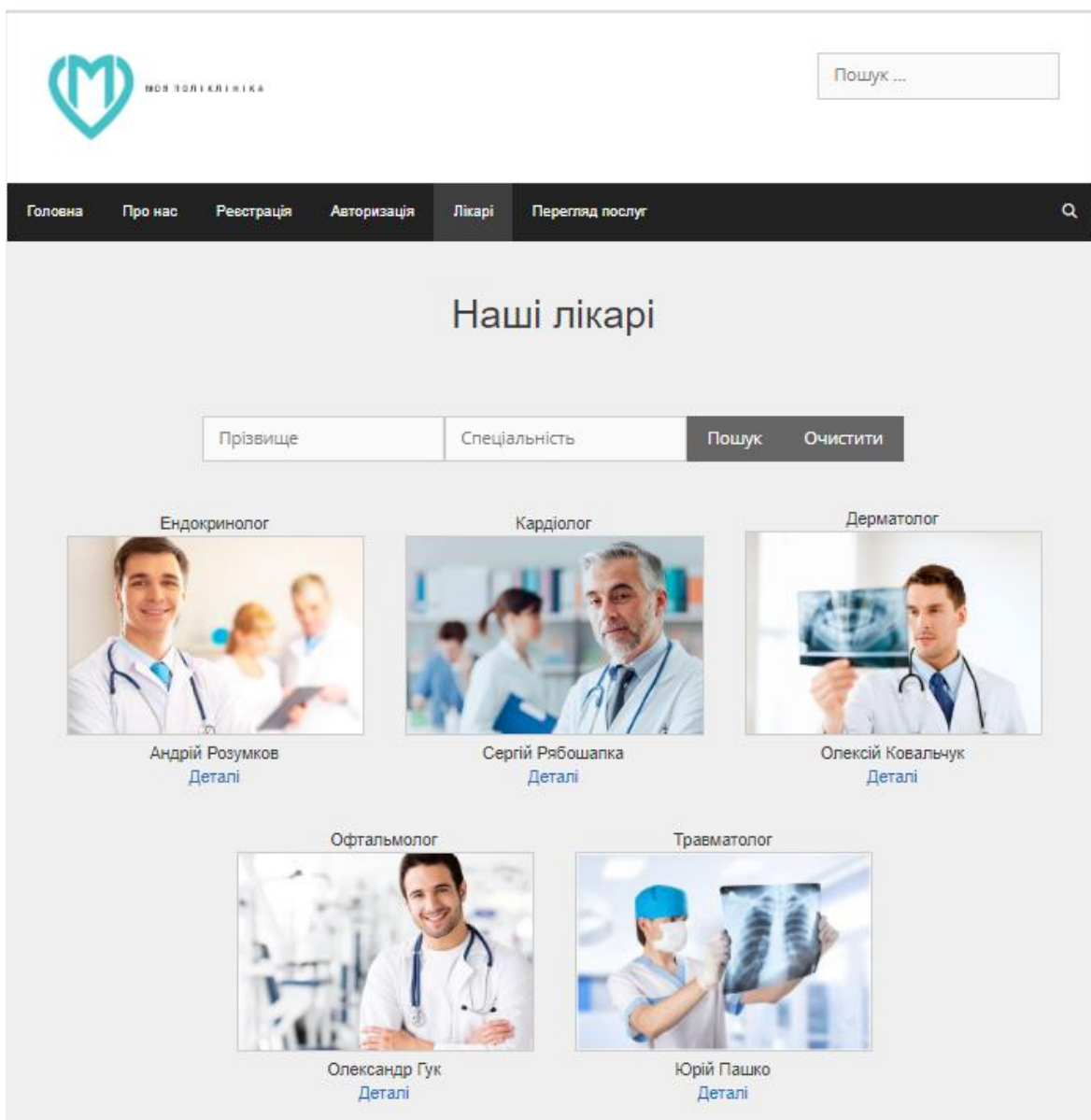


Рисунок 3.17 – Відображення переліку всіх лікарів

Сторінку перегляду всіх послуг, які надаються поліклінікою, відображено на рисунку 3.18

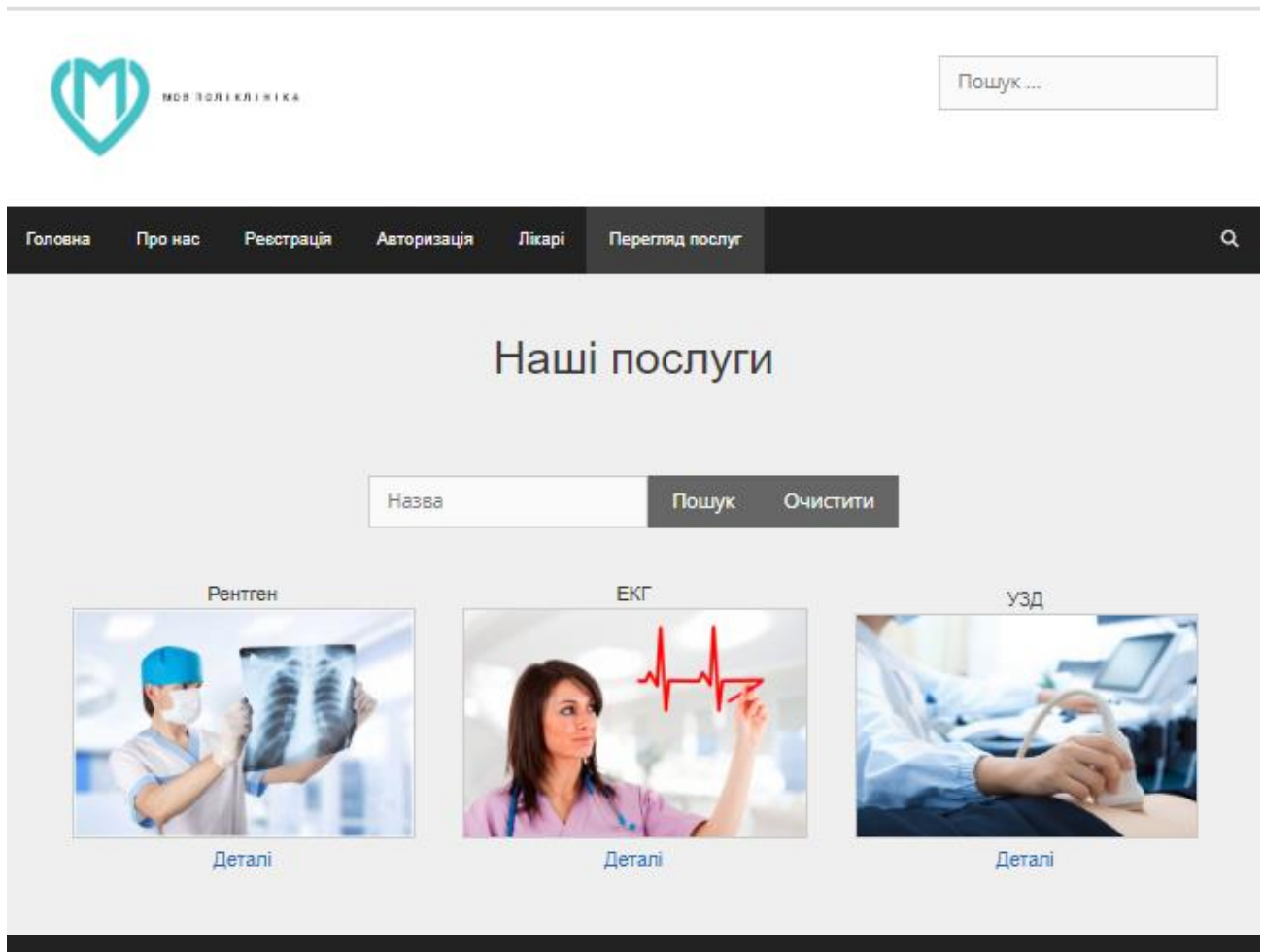


Рисунок 3.18 – Сторінка з переліком послуг

Для додаткових функцій користувачі повинні авторизуватися або реєструватися. Для авторизації в системі користувач повинен заповнити поля «Email» та «Пароль». Форма авторизації зображена на рисунку 3.19.

Авторизація

Ім'я користувача або Email адреса

Пароль

Пам'ятати мене

Увійти

Рисунок 3.18 – Форма авторизації користувачів

Авторизація в системі може відбуватися під трьома ролями, а саме пацієнт, лікар та пацієнт реєстратури. Кожен з цих користувачів має доступ до різних функцій. При успішній авторизації в системі перед користувачем відображується головна сторінка та з'являються нові пункти меню, в залежності від прав доступу. На рисунку 3.19 відображено меню для користувача під роллю «Пацієнт».

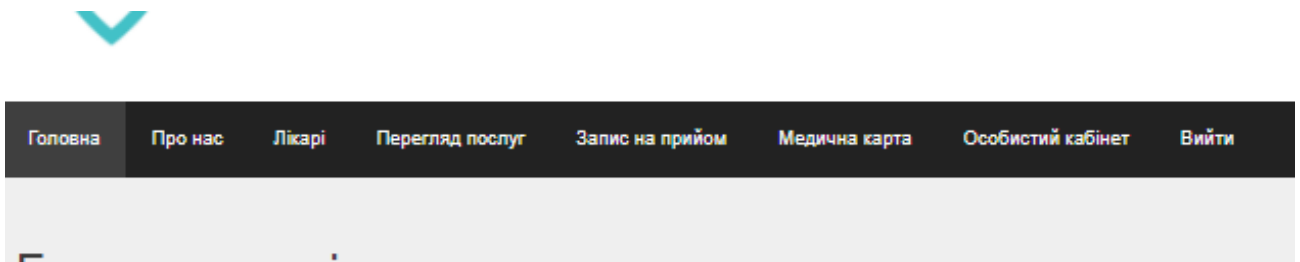


Рисунок 3.19 – Меню для пацієнта

Додатковими функціями, що може користуватися пацієнт є:

- Запис на прийом;
- Перегляд призначень лікаря (Медична карта);
- Особистий кабінет.

При переході на сторінку запис на прийом. Пацієнт має можливість обрати лікаря та час прийому. Приклад запису на прийом зображено на рисунках 3.20 - 3.21.

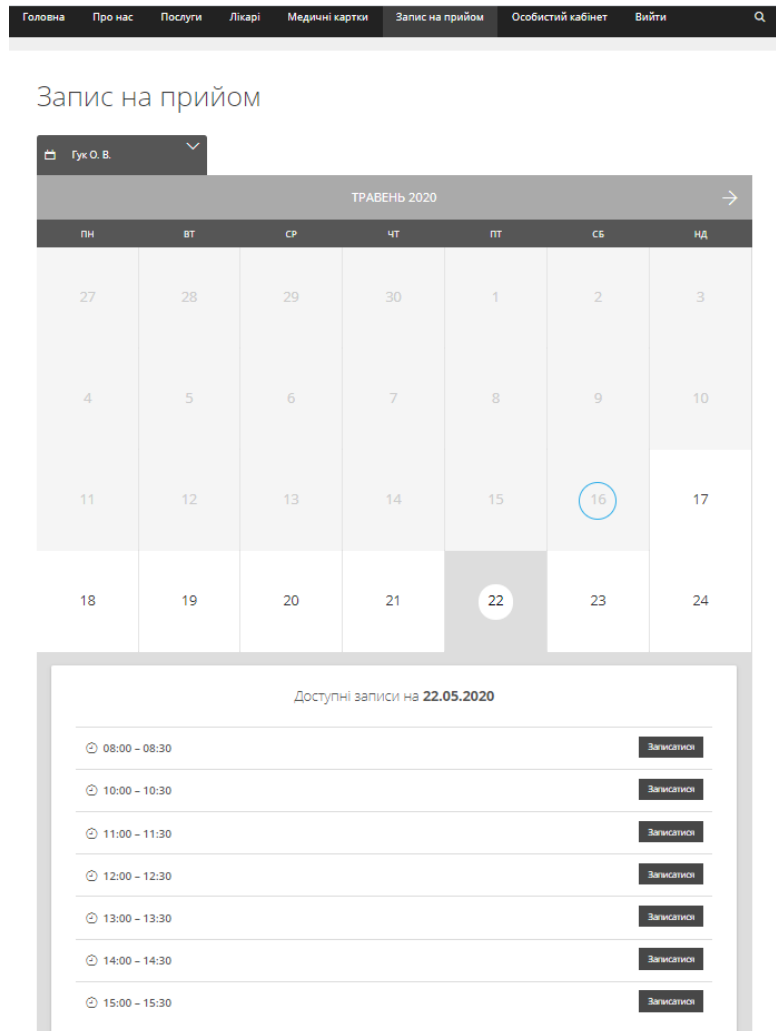


Рисунок 3.20 – Приклад запису на прийом пацієнта

Після натиснення кнопки записатися з'являється вікно для підтвердження запису. Вікно підтвердження запису на прийом відображено на рисунку 3.21.

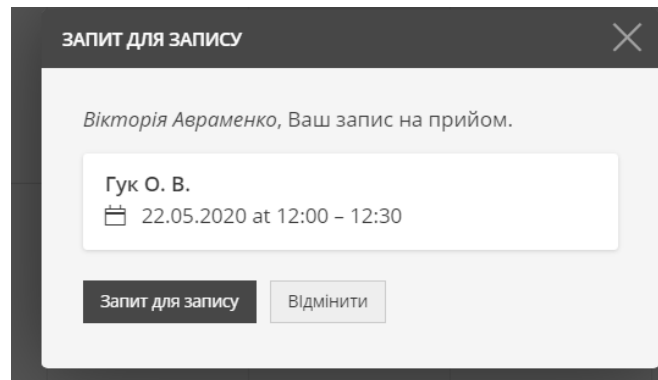


Рисунок 3.21 – Підтвердження запису на прийом

Для перегляду статусу запису на прийом його відміни або перегляду історії всіх прийомів, пацієнт переходить на сторінку «Особистий кабінет». Записи на прийом можуть бути в статусах очікується, підтверджено або відмінено, що визначається працівником реєстратури або лікарем. Особистий кабінет пацієнта відображено на рисунку 3.22.

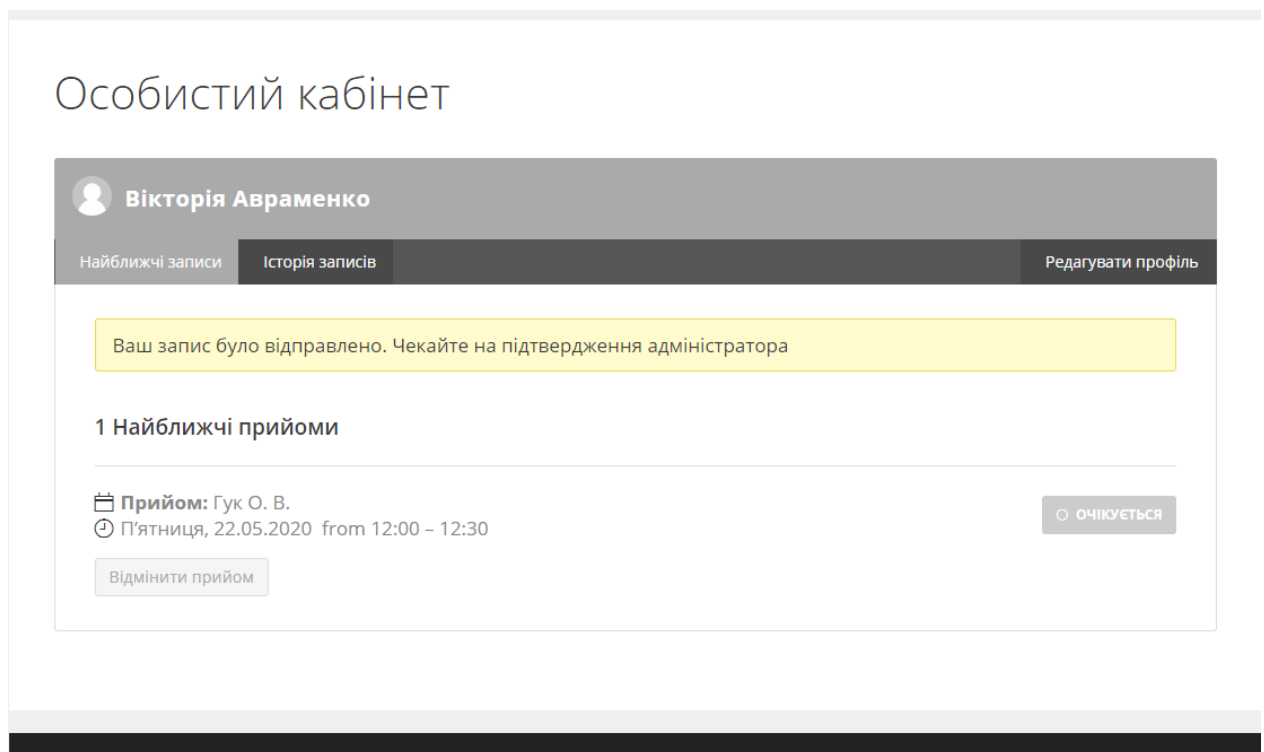
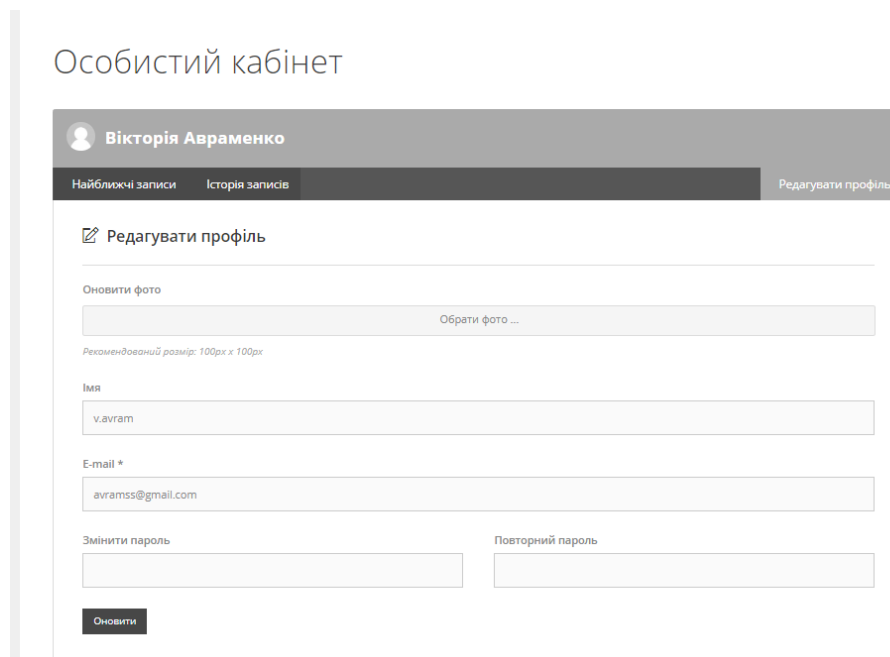


Рисунок 3.22 – Особистий кабінет пацієнта

Крім того пацієнт має змогу змінити дані для авторизації в системі, перейшовши на вкладку редагувати профіль та заповнивши всі необхідні дані, а саме ім'я користувача, пароль, фото профілю та пароль. Приклад редагування профілю відображено на рисунку 3.23.



The screenshot shows a web interface for editing a patient's profile. At the top, it says "Особистий кабінет" (Personal Cabinet) and identifies the user as "Вікторія Авраменко". Below this, there are navigation tabs: "Найближчі записи" (Upcoming appointments), "Історія записів" (Appointment history), and "Редагувати профіль" (Edit profile). The "Редагувати профіль" tab is active, showing a form with the following fields:

- Оновити фото** (Update photo): A button labeled "Обрати фото ..." (Choose photo ...) with a note below: "Рекомендований розмір: 100px x 100px" (Recommended size: 100px x 100px).
- Ім'я** (Name): A text input field containing "v.avram" and a clear button (X).
- Е-mail *** (Email): A text input field containing "avram55@gmail.com" and a clear button (X).
- Змінити пароль** (Change password): A text input field.
- Повторний пароль** (Repeat password): A text input field.
- Оновити** (Update): A dark button at the bottom left of the form.

Рисунок 3.23 – Редагування профілю

Для перегляду призначень лікарів пацієнт повинен перейти на сторінку «Медична карта», де йому відображається його призначення для лікування, а також додаткова інформація про нього, яку він має змогу змінювати натиснувши на кнопку «Редагувати». Сторінку з призначеннями пацієнта відображено на рисунку 3.24.

Дата призначення	Діагноз	Лікар	Призначення	Рекомендація
2020-05-26	Остеохондроз поперекового відділу хребта	Юрій Пашко	Напівжорсткий ортез на попереково-крижовий відділ хребта	спостереження у травматолога

Додаткова інформація

Дата народження: 1998-05-10

Додаткова інформація: Місцева алергія на анальгін

Номер телефону: 380958264984

[Редагувати](#)

Рисунок 3.24 – Перегляд призначення пацієнта

При авторизації від імені лікаря, крім загальнодоступних, з’являються такі пункти меню:

- Онлайн-запис;
- Особистий кабінет;
- Запис на прийом;

На рисунку 3.25 відображено меню для користувача під роллю «Лікар».

Рисунок 3.25 – Меню для лікаря

Крім цього на панелі адміністрування Wordpress, з’являється пункт «Записи». При переході на дану сторінку лікар має можливість налаштувати власний графік прийомів або записати пацієнта на свій прийом. Приклад налаштування графіку прийомів відображено на рисунку 3.26.

Редагування графіку прийомів:

Пашко Ю. Н.

ПОНЕДІЛОК	ВІВТОРОК	СЕРЕДА	ЧЕТВЕР
<div style="display: flex; justify-content: space-between;"> ОДИН ГРУПОВИЙ </div> <div style="margin-top: 5px;"> <input type="text" value="Час початку ..."/> </div> <div style="margin-top: 5px;"> <input type="text" value="Час закінчення ..."/> </div> <div style="margin-top: 5px;"> <input type="text" value="1 Доступні місця"/> </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> <input type="button" value="Закрити"/> <input type="button" value="Додати"/> </div>	<input type="button" value="Запис"/> <input type="button" value="Очистка"/>	<input type="button" value="Запис"/> <input type="button" value="Очистка"/>	<input type="button" value="Запис"/> <input type="button" value="Очистка"/>

Рисунок 3.26 – Налаштування графіку прийому

При переході на сторінку особистий кабінет лікар має змогу переглянути його найближчі прийоми та всю історію його прийомів, а також підтвердити або відмінити прийом пацієнта. Крім цього лікар має змогу редагування даних профілю для авторизації в системі.

При переході на вкладку призначення лікар має можливість перегляду та пошуку призначень за номером телефону або ПІБ пацієнта, а також може додавати, редагувати або видаляти призначення. Приклад відображення призначень знайденого пацієнта відображено на рисунку 3.27

ПІБ	Дата призначення	Лікар	Призначення	Діагноз	Рекомендації	Дія
Аліна Семенова	2020-05-26	Юрій Пашко	Напівжорсткий ортез на попереково-крижовий відділ хребта	Остеохондроз поперекового відділу хребта	спостереження у травматолога	Редагувати Видалити

Додаткова інформація про обраного пацієнта

Дата народження: 1998-05-10

Додаткова інформація: Місцева алергія на анальгін

Номер телефону: 380958264984

Рисунок 3.27 – Відображення призначень знайденого пацієнта

Приклад додавання нових призначень відображено на рисунку 3.28

ПІБ

380958264

26

Аліна Семенова (380958264984)

380958264984

Рекомендація

Додати

Рисунок 3.28 – Приклад додавання нового призначення

Лікар також має можливість редагувати власні призначення та видалити призначення. При спробі натиснення кнопки редагування призначення наданого іншим лікарем з'являється повідомлення, що зображено на рисунку 3.29

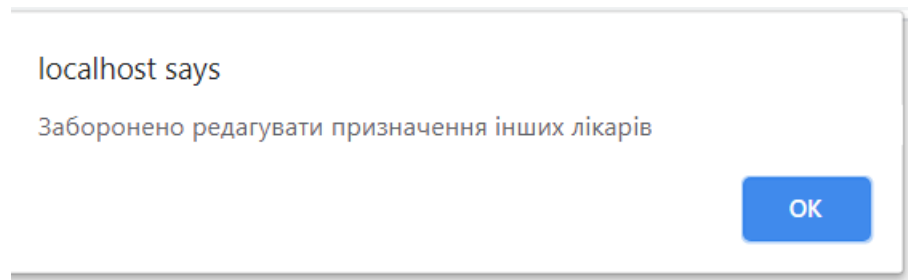


Рисунок 3.30 – Повідомлення при спробі редагування не власного призначення

Приклад редагування призначення відображено на рисунку 3.31

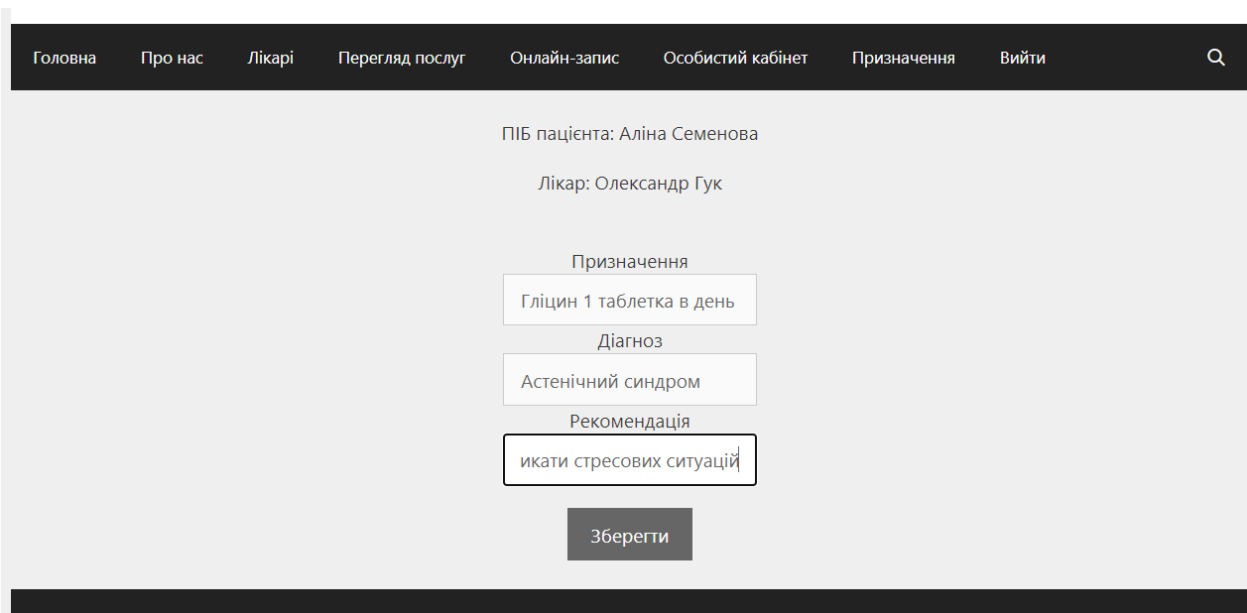


Рисунок 3.31 – Редагування призначень пацієнта

При авторизації від імені працівника реєстратури з'являться такі пункти меню як

- Онлайн-запис;
- Особистий кабінет;
- Редагування даних.

На рисунку 3.32 відображено меню для користувача під роллю «Працівник реєстратури».

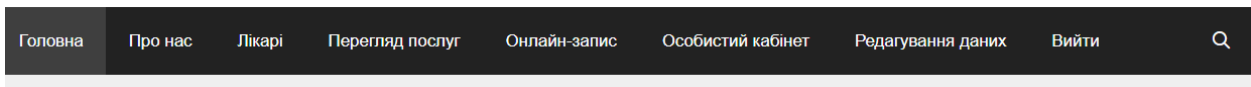


Рисунок 3.32 – Меню для працівника реєстратури

Крім цього буде доступна панель адміністрування Wordpress, де працівник реєстратури матиме можливість призначати ролі користувачам, створювати нові графіки прийомів, редагувати графіки прийомів. Редагування графіків прийомів здійснюється за аналогією редагування графіку прийомів лікарем, але працівник реєстратури має доступ до всіх прийомів.

Приклад створення нового графіку зображено на рисунку 3.33

A form for creating a new custom calendar. It has four main sections: 1. 'Назва' (Name) with a text input field and a note: 'Назва, як вона відображається на вашому сайті.' 2. 'Частина посилання' (Link part) with a text input field and a note: '“Частина посилання” це версія назви для URL. Зазвичай це маленькі букви, цифри, та дефіси.' 3. 'Опис' (Description) with a large text area and a note: 'Опис схований за замовчуванням, але деякі теми можуть його показувати.' 4. 'Assign this calendar to:' with a dropdown menu showing 'Рябошанка Сергій (riabosh@gmail.com)' and a note: 'This will use your setting from the Booked Settings panel by default.' At the bottom is a blue button labeled 'Add New Custom Calendar'.

Рисунок 3.33 – Створення нового графіку

При створенні нового графіку та призначення, його певному лікарю, він автоматично буде відображений у лікаря, за яким цей графік закріплено.

При переході на сторінку особистий кабінет працівник реєстратури має змогу переглянути всі найближчі прийоми та всю історію прийомів, а також підтвердити або відмінити прийоми пацієнта. Крім цього працівник має змогу редагування даних профілю для авторизації в системі.

Працівник реєстратури має змогу управляти даними, а саме видаляти, редагувати та додавати. Ці дії він може виконувати над даними пацієнтів, послуг, лікарів, спеціальностей та може призначати певну спеціальність конкретному лікарю. Приклади деяких з цих приведено на рисунках 3.34 -3.36. Приклад сторінки відображення спеціальностей відображено на рисунку 3.34

Управління спеціальностями

[Додати нову спеціальність](#)
[Редагувати послуги](#)
[Редагувати лікарів](#)
[Редагувати пацієнтів](#)

Назва Пошук

Назва	Дія
Невропатолог	Редагувати Видалити
Кардіолог	Редагувати Видалити
Ендокринолог	Редагувати Видалити
Офтальмолог	Редагувати Видалити
Дерматолог	Редагувати Видалити

1 2

Рисунок 3.34 – Управління спеціальностями

Приклад призначення спеціальностей лікарю зображено на рисунку 3.35.

Рисунок 3.35 – Призначення спеціальностей лікарю

Приклад додавання нової послуги зображено на рисунку 3.36

Рисунок 3.36 – Додавання нової послуги

Отже, в результаті роботи над даним розділом було реалізовано інформаційну систему підтримки діяльності реєстратури поліклініки. Також було продемонстровано роботу її основних функцій. Розроблена інформаційна система допоможе поліклініці та її працівникам спростити виконання їх робочих обов'язків. Крім цього, дана інформаційна система спростить для пацієнта виконання таких функцій як запис на прийом та перегляд призначень лікарів.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи бакалавра, було реалізовано інформаційну систему підтримки діяльності реєстратури поліклініки. Дану інформаційну систему було розроблено з цілями створення сприятливого враження про поліклініку, залучення нових пацієнтів, розміщення інформації про послуги та їх вартість, спрощення процесу запису на прийом, спрощення процесу надання призначень.

Для досягнення поставленої мети було виконано такі задачі: проведено аналіз програмних продуктів-аналогів, визначено функціональні вимоги та розроблено технічне завдання, здійснено моделювання системи, розроблено дизайн інформаційної системи, реалізовано інформаційну систему.

У ході роботи над першою частиною для подальшої роботи над реалізацією інформаційної системи проведено, було обгрунтовано підставу на основі якої необхідно розробити інформаційну систему, було здійснено деталізацію мети та задач проекту та обрано і обгрунтовано вибір засобів та технологій для розробки початку розробки. А також розроблено технічне завдання проекту головні проблеми та вимоги, для продовження роботи над об'єктом автоматизації.

Під час виконання другої частини роботи було розроблено діаграми, що є необхідними для моделювання інформаційної системи, які показують процеси роботи інформаційної системи, що далі були необхідними при її реалізації.

При виконанні третьої частини роботи було реалізовано інформаційну систему та продемонстровано її роботу. Отже впровадження даної інформаційної системи дозволить удосконалити процеси роботи поліклініки, лікарів. А також допоможе пацієнту у забезпеченні для нього основних функцій, що надають лікарі та працівники реєстратури.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Руководство разработчика CSS. [Электронный ресурс]. – Точка доступа: URL: https://developer.mozilla.org/ru/docs/Web/Guide/CSS/Getting_started/What_is_CSS
2. Довідник HTML [Электронный ресурс]. – Точка доступа: URL: <http://htmlbook.ru/html>
3. Mysql [Электронный ресурс]. – Точка доступа: URL: http://www.znannya.org/?view=PHP_SUBD_mysql
4. Основы JavaScript [Электронный ресурс]. – Точка доступа: URL: https://developer.mozilla.org/ru/docs/Web/Guide/CSS/Getting_started/What_is_CSS
<https://learn.javascript.ru/intro>
5. Классификация медицинских информационных систем [Электронный ресурс]. – Точка доступа: URL: <https://medic.studio/tehnologii-meditsine-informatsionnyie/klassifikatsiya-meditsinskih-informatsionnyih-58978.html>
6. Пилипенко Т. О., Зайцева Т. А. The actual medical information system model on the example "Information card of the patient of anesthesiology department" // Регіональна науково-практична конференція молодих учених та студентів. 2017. С. 56–58.
7. Інформаційні технології в медицині [Электронный ресурс]. – Точка доступа: URL: <https://medytsyna.com/informatsijni-tehnologiyi-v-meditsini-1-docx/>
8. МЕТОДОЛОГИЯ IDEF. [Электронный ресурс]. – Точка доступа: URL: https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2
9. Что такое PHP? [Электронный ресурс]. – Точка доступа: URL: <https://www.php.net/manual/ru/intro-what-is.php>

10. Розробка uml діаграми варіантів використання [Електронний ресурс]. – Точка доступу: URL: <https://studfile.net/preview/5200239/page:6/>
11. Создание Web-сайта на базе WordPress CMS [Електронний ресурс]. – Точка доступу: URL: <https://www.ibm.com/developerworks/ru/library/os-wordpress/index.html>
12. База данных Вордпресс [Електронний ресурс]. – Точка доступу: URL: <https://techbear.ru/kak-rabotaet-baza-dannyh/>
13. What is Database? What is SQL? [Електронний ресурс]. – Точка доступу: <https://www.guru99.com/introduction-to-database-sql.html>
14. Database Description [Електронний ресурс]. – Точка доступу: https://codex.wordpress.org/Database_Description
15. Элементы модели "сущность-связь" [Електронний ресурс]. – Точка доступу: <http://citforum.ru/database/dblearn/dblearn08.shtml>
16. Архитектура информационных систем [Електронний ресурс]. – Точка доступу: <https://www.intuit.ru/studies/courses/611/467/lecture/28784>
17. Из чего состоит сайт на WordPress: основные составляющие интернет-проекта [Електронний ресурс]. – Точка доступу: <https://wpuroki.ru/novichkam/iz-chego-sostoit-sajt.html>
18. Функціональне моделювання [Електронний ресурс]. – Точка доступу: <http://moodle.ipk.kpi.ua/moodle/mod/resource/view.php?id=37025>
19. Знакомство с wordpress [Електронний ресурс]. – Точка доступу: <https://wp-kama.ru/handbook/wordpress/papka-wp-content>
20. Создание ER-Диаграмм [Електронний ресурс]. – Точка доступу: <http://inf-teh-lotos.ru/sozdanie-er-diagramm>

Додаток А.
Технічне завдання

ТЕХНІЧНЕ ЗАВДАННЯ
на розробку інформаційної системи підтримки діяльності реєстратури
поліклініки

Суми 2020

1 ПРИЗНАЧЕННЯ Й МЕТА СТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Призначення інформаційної системи

Інформаційна система призначена для підвищення ефективності роботи поліклініки, в тому числі - для скорочення часу очікування пацієнтів запису на прийом до лікаря.

Мета створення інформаційної системи

Основною метою є використання можливостей інформаційної системи в повсякденній роботі медичного та управлінського персоналу і, як наслідок, підвищення ефективності роботи поліклініки за рахунок:

- довготривалого зберігання медичних даних в електронному вигляді;
- звільнення співробітників від рутинної паперової роботи і необхідності дублювання інформації в різних журналах;
- забезпечення конфіденційності інформації.

Цільова аудиторія

У цільовій аудиторії інформаційної системи можна виділити наступні групи:

1. Працівники реєстратури лікарні.
2. Пацієнти.

3. Лікарі.
4. Інші зацікавлені відвідувачі.

2 ВИМОГИ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Вимоги до інформаційної системи в цілому

Вимоги до структури й функціонування інформаційної системи

Інформаційна система повинна бути реалізована у вигляді web-додатку, що має бути доступним в мережі Інтернет. Web-додаток повинен складатися із взаємозалежних розділів із чітко розділеними функціями.

Вимоги до персоналу

Для підтримки ІС й експлуатації інтерфейсу системи керування сайтом від персоналу не повинно вимагатися спеціальних технічних навичок, знання технологій або програмних продуктів, за винятком загальних навичок роботи з персональним комп'ютером і стандартним веб-браузером (наприклад, Google Chrome).

Вимоги до збереженні інформації

Вся основна інформація повинна зберігатися в базі даних.

Вимоги до розмежування доступу

Інформація, розташовувана у web-додатку, є загальнодоступною.

Користувачів сайту можна розділити на такі групи відповідно до прав доступу:

- Відвідувачі
- Лікарі
- Пацієнти
- Адміністратор
- Працівники реєстратури

Відвідувачі мають доступ тільки до загальнодоступної частини сайту.

Доступ до адміністративної частини мають користувачі із правами адміністратора та працівники реєстратури.

Лікарі мають доступ до загальнодоступної частини сайту та до особистого кабінету лікаря. Можуть змінювати розклад прийомів та вести призначення пацієнта.

Пацієнти мають доступ до загальнодоступної частини сайту, призначень лікаря та до особистого кабінету пацієнта. Можуть записуватися на прийом до обраного лікаря на обраний час.

Працівники реєстратури можуть реєструвати пацієнтів та лікарів в системі, записувати пацієнтів на прийом, змінювати розклад лікаря, змінювати матеріали розділів, змінювати інформацію про пацієнтів,

Доступ повинен здійснюватися з використанням унікального логіна й пароля.

Вимоги до функцій, виконуваних сайтом

Основні вимоги

Структура сайту

Сайт повинен складатися з наступних розділів:

Головна – містить загальну інформацію про поліклініку, послуги та лікарів.

Послуги – містить повну інформацію про відділення, послуги та їх вартість.

Лікарі – містить повну інформацію про всіх лікарів поліклініки, їх спеціальності, досвід роботи і т.д.

Запис на прийом – слугує для організації самостійного запису пацієнтів на прийом до лікаря через мережу Інтернет.

Онлайн-запис (для працівників реєстратури) – слугує для автоматизації реєстрації пацієнтів, що звертаються в поліклініку, ведення розкладу прийому лікарів і запису пацієнтів на прийом.

Особистий кабінет пацієнта – слугує для відображення його прийомів.

Особистий кабінет лікаря/працівника реєстратури – слугує для відображення всіх прийомів лікаря з можливістю підтвердження або відміни прийомів.

Призначення – слугує для відображення призначень (медичної карти) пацієнта. Пацієнт має можливість редагувати додаткові дані. Лікар має можливість додавати, редагувати (власні призначення) та видаляти призначення.

Навігація

Інтерфейс повинен бути інтуїтивно зрозумілий і мати досить простий функціонал, інформація повинна бути структурована і легкодоступна в пошуку.

Система повинна забезпечувати навігацію по всіх доступних користувачеві розділам і відображати відповідну інформацію. Навігаційні елементи повинні забезпечувати простоту розуміння. Посилання на сторінки повинні мати заголовок, умовні позначки відповідати загальноприйнятим.

Система повинна забезпечувати навігацію по всіх доступних користувачеві ресурсам і відображати відповідну інформацію. Для навігації повинна використовуватися система контент-меню. Меню повинне являти собою текстовий блок (список гіперпосилань) у лівій колонці або у верхній частині сторінки (залежно від затвердженого дизайну).

Наповнення сайту (контент)

Наповнення всіх сторінок повинне формуватися програмним шляхом на підставі інформації з бази даних на сервері.

Редагування вмісту повинно здійснюватися за допомогою адміністраторського веб-інтерфейсу, який повинен передбачати можливість редагування інформаційного вмісту сторінок сайту без застосування спеціальних навичок програмування. Наповнення інформацією повинне проводитися з використанням шаблонів сторінок сайту.

Система навігації (карта сайту)

Взаємозв'язок між розділами й підрозділами сайту (карта сайту) представлено на рисунку **Ошибка! Источник ссылки не найден..**



Рисунок А.1– Карта сайту

Вимоги до функціональних можливостей

Система керування контентом (адміністративна частина) повинна надавати можливість додавання, редагування й видалення вмісту сторінок. Також повинна бути передбачена можливість додавання інформації без відображення на сайті.

Функціональні можливості розділів

На головній сторінці будуть представлені наступні елементи:

- Пошук лікарів;
- Вибір з пунктів меню розділів;
- Авторизація користувачів;
- Перегляд інформації про поліклініку;
- Перегляд послуг;
- Перегляд лікарів;
- Можливість переглянути контактні дані поліклініки.

Загальні вимоги

Стиль web-додатку можна описати як сучасний, діловий, що має лаконічний і зручний вигляд.

Оформлення не повинне заземляти інформативність: хоча на сайті не повинно бути досить багато графіки, він повинен бути зручний користувачам у плані навігації й цікавий для багаторазового відвідування.

Розташування елементів на головній сторінці сайту схематично показано на рис. А.2-А.3.



Рисунок А.2 – Типова сторінки

Типові навігаційні й інформаційні елементи

- Шапка сайту
- Основне меню
- Основне поле контенту

Шапка сайту

Шапка сайту повинна містити логотип та назву. Логотип є посиланням на головну сторінку.

Основне меню

Основне меню повинне розташовуватися у верхній частині вікна (під шапкою) і містити посилання на всі розділи.

Основне поле контенту

Основне поле контенту повинне розташовуватися в центрі сторінки. В даному полі відображається інформація відносно певного розділу. Стильове оформлення матеріалів і їх елементів (посилань, заголовків, основного тексту, зображень, форм, таблиць і т.п.) повинне бути єдиним для всього веб-додатку.

2.3 Вимоги до видів забезпечення

Вимоги до інформаційного забезпечення

Реалізація відбувається за допомогою, CMS Wordpress, PHP, HTML, JS, Mysql, CSS.

Вимоги до лінгвістичного забезпечення

Web-додаток повинен бути виконаний українською мовою.

Вимоги до програмного забезпечення

Програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам:

- Наявний доступ до Інтернету
- Операційна система: Microsoft Windows 2000(і вище)
- Веб-браузер: Firefox 3.1 (і вище), Google Chrome, Internet Explorer 7.0 (і вище).

Вимоги до апаратного забезпечення

Апаратне забезпечення клієнтської частини повинне забезпечувати підтримку програмного забезпечення клієнтської частини, зазначеного в п. 2.2.3

3 СКЛАД І ЗМІСТ РОБІТ

Докладний опис етапів роботи розробки ІС наведено на рисунку А.4 у вигляді діаграми Ганта.

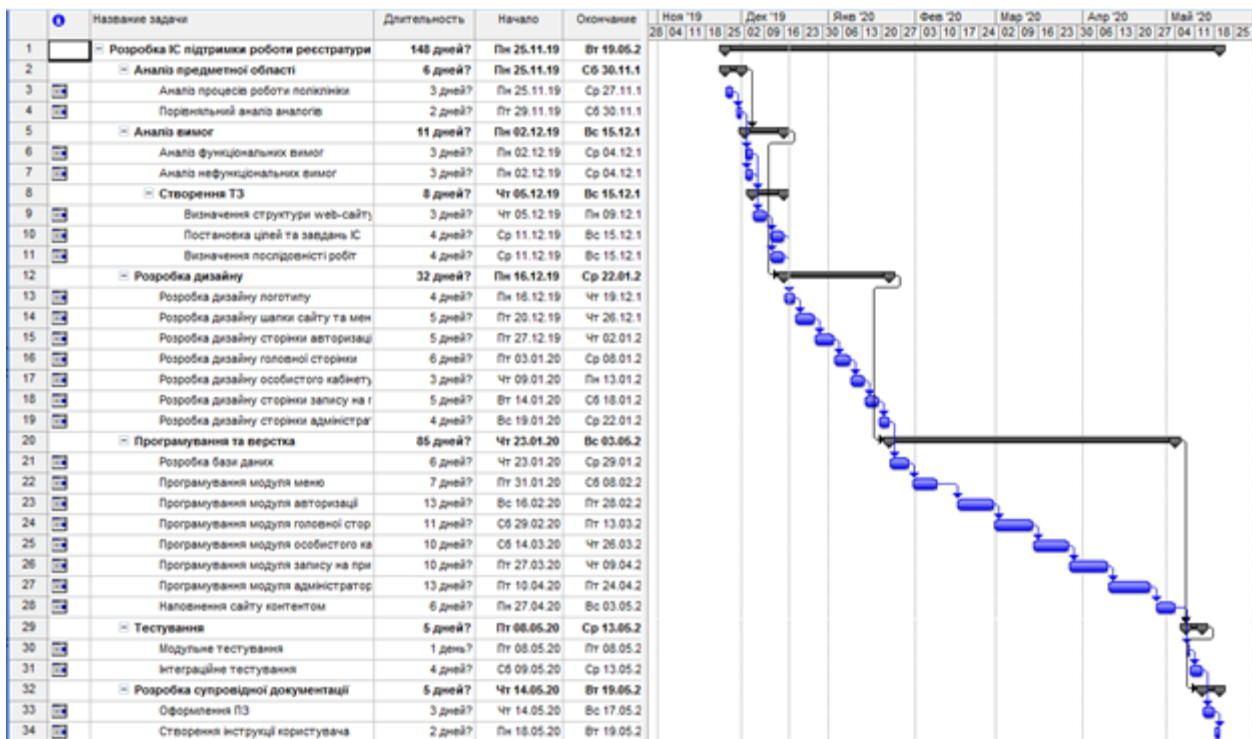


Рисунок 4 – діаграма Ганта

Вимоги до складу й змісту робіт із введення сайту в експлуатацію

Для створення умов функціонування, при яких гарантується відповідність створюваного програмного продукту вимогам сьогодення, ТЗ і можливість його ефективної роботи, в організації Замовника повинен бути проведений певний комплекс заходів.

Додаток Б.

Планування робіт

Деталізація мети інформаційної системи методом SMART. Продуктом дипломного проекту є інформаційна система підтримки діяльності реєстратури поліклініки деталізації методом SMART розміщені у табл. Б.1.

Таблиця Б.1 – Деталізація мети інформаційної системи методом SMART

Specific (конкретна)	Розробити інформаційна системпідтримки діяльності реєстратури поліклініки.
Measurable (вимірювана)	Результатом роботи є оцінка проекту замовником.
Achievable (досяжна)	Мету реально досягнути, так як розробка інформаційної системи відбувається за допомогою web-програмування і вимагає мінімум зусиль.
Relevant (реалістична)	Всі необхідні технічні та програмні засоби є у наявності. Достатньо кваліфіковані розробники для виконання поставлених задач
Time-framed (обмежена у часі)	Обмеженість в часі зумовлена рішенням замовника, і інформаційна система повинна бути розроблена в оговорені терміни. Проект необхідно виконати згідно з календарним планом.

Планування змісту структури робіт. Основним інструментом для планування змісту структури робіт служить WBS-діаграма служить основним інструментом для планування змісту структури робіт. Вона зображує

інформаційну систему у вигляді ієрархічної структури робіт та досягається за допомогою послідовної декомпозиції. WBS-діаграму зображено на рисунку Б.2.

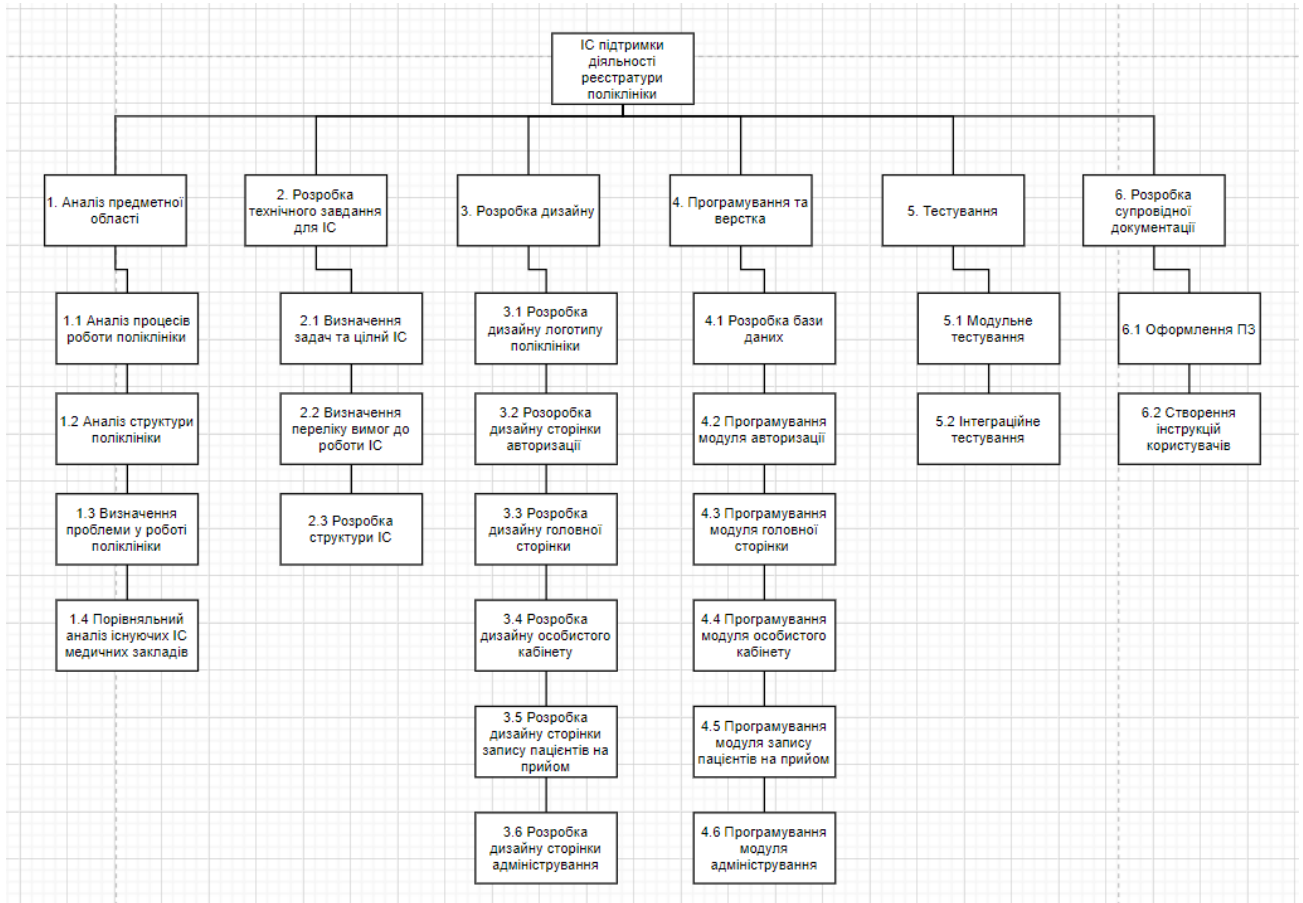


Рисунок Б.2 – WBS-діаграма

Планування структури організації, для впровадження готового проекту (OBS). Після побудови WBS-діаграми було розроблено організаційну структуру виконавців (OBS), що за своєю структурою відповідає WBS-діаграмі, але її елементами є виконавці робіт. OBS-діаграма являє собою ієрархічну структурою управління проектом і показує відносини між учасниками проекту. OBS-діаграму зображено на рис. Б.3.

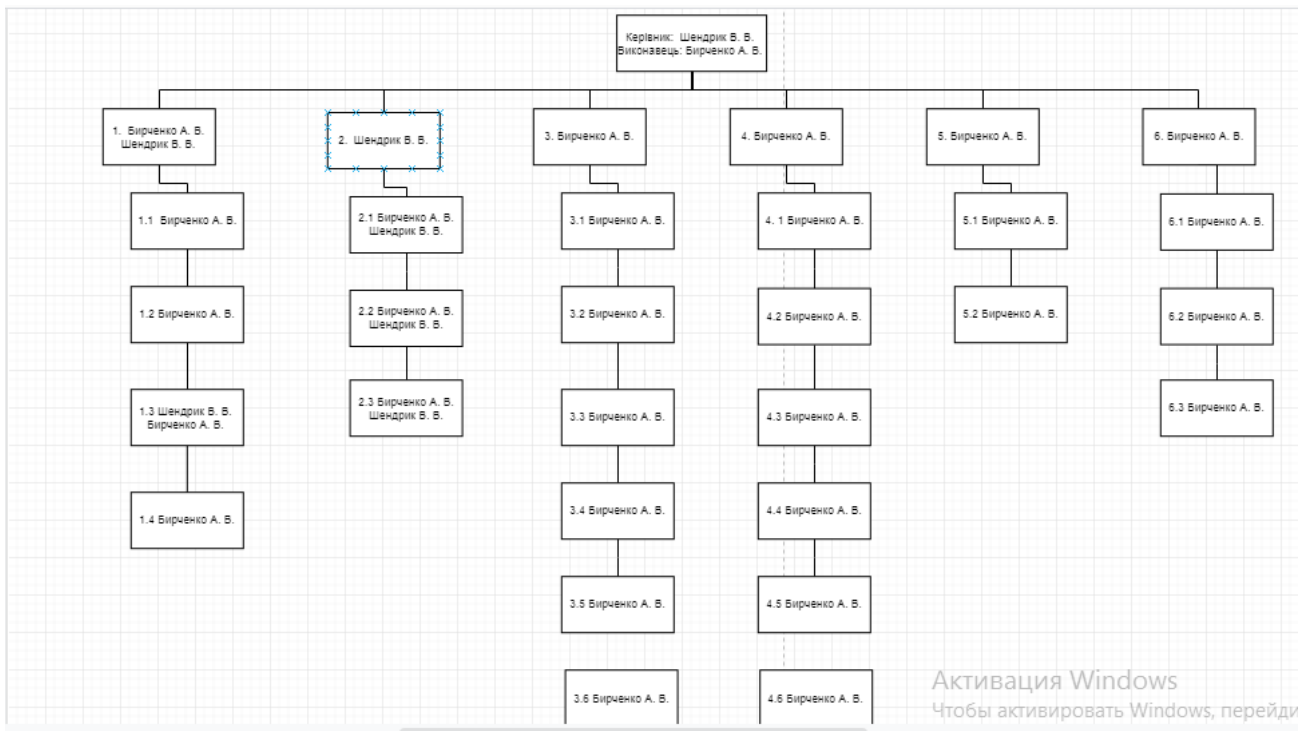


Рисунок Б.3 – OBS-діаграма

Діаграма Ганта. Далі було розроблено календарний план виконання дипломного проекту. Найпоширенішим форматом календарного плану є діаграма Ганта. Діаграма Ганта та список робіт діаграми Ганта зображені на рисунках Б.3-Б.6.

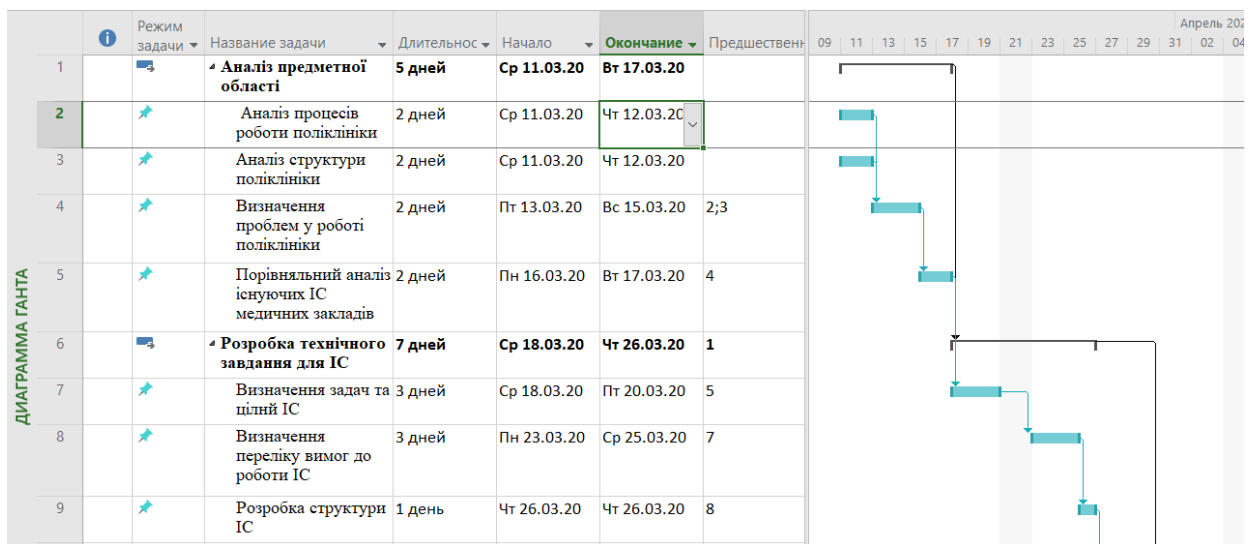


Рисунок Б.4 – Діаграма Ганта

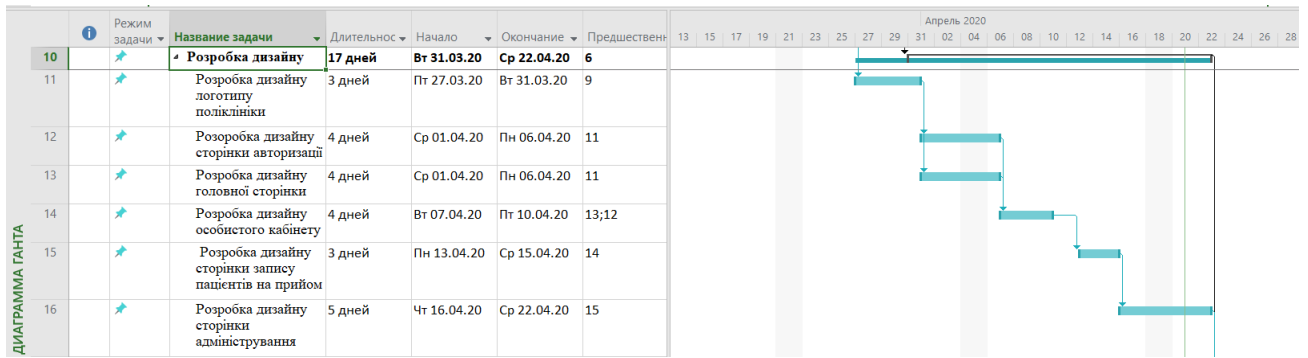


Рисунок Б.5 – Продовження діаграми Ганта



Рисунок Б.5 – Продовження діаграми Ганта

Аналіз ризиків. Далі було проведено аналіз ризиків. Основною стратегією управління ризиками вважається їх мінімізація. Управління ризиками містить у собі планування ризиків, реалізацію дій по реагуванню, контроль, роботу з новими ризиками, або ризиками, що залишилися.

В таблицях Б.2 та Б.4 зображено реєстрацію ризиків.

Таблиця Б.2 – Risk Register

	Risk	Impact	Probability	Mitigation	RV	How to solve
1	Часте внесення змін до ТЗ	3	3		Виправданий	Погодити всі питання з замовником та внести необхідні зміни і поправки.
2	Неоптимальний розподіл часу	5	3		Недопустимий	Змінити порядок пріоритетів робіт. Знайти способи оптимізації із уже існуючою розташовкою. Обговорити варіанти внесення змін до плану з замовником.

Продовження таблиці Б.2 – Risk Register

3	Проблеми в комунікації з замовником	1	2		Прийнятний	Вияснити, що саме стало причиною непорозуміння та обговорити її
4	Не оптимальне планування бюджету	5	3		Недопустимий	Компенсувати витрати за рахунок інших етапів проекту
5	Проблеми з доступом до серверу	5	2		Виправданий	Необхідне втручання розробника, для налагодження роботи сервера
6	Зміна вимог замовника в процесі розробки	4	2		Виправданий	Переоцінка проекту

Відповідно до таблиці ризику поділяються на:

- зелений колір – прийнятні ризики;
- жовтий колір – виправданні ризики;
- червоний колір – недопустимі ризики.

Оцінка ризиків здійснюється за допомогою двох критеріїв: ступінь впливу (impact) та ймовірність виникнення (probability).

Таблиця Б.3 – Шкала оцінювання ймовірності виникнення та впливу ризику на виконання проекту

Оцінка	Ймовірність виникнення	Ступінь впливу
1	very low	negligible
2	low	minor
3	medium	moderate
4	high	serious
5	very high	critical

Таблиця Б.4 – Probability/Impact matrix

Probability	Probability/Impact matrix				
5 very high					
4 high					
3 medium			1		2, 4
2 low	3			6	5
1 very low					
Impact	1 negligible	2 minor	3 moderate	4 serious	5 critical

Додаток Г.

Лістинг програмного коду

Лістинг створення таблиць бази даних

```

CREATE TABLE `doctors` (
  `id` bigint(11) NOT NULL,
  `name` varchar(20) DEFAULT NULL,
  `surname` varchar(20) DEFAULT NULL,
  `phone` varchar(50) DEFAULT NULL,
  `info` varchar(250) DEFAULT NULL,
  `photo` varchar(250) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO `doctors` (`id`, `name`, `surname`, `phone`, `info`, `photo`) VALUES
(15, 'Юрій', 'Пашко', '0', '', 'gallery/90.jpg'),
(22, 'Олександр', 'Гук', '0', '', 'gallery/9.jpg'),
(23, 'Олексій', 'Ковальчук', '0', '', 'gallery/6.jpg'),
(24, 'Сергій', 'Рябошапка', '0', '', 'gallery/5.jpg'),
(25, 'Андрій', 'Розумков', '0', '', 'gallery/4.jpg');

CREATE TABLE `list_of_positions` (
  `id_list` bigint(10) NOT NULL,
  `id_pos` bigint(10) NOT NULL,
  `id_doctor` bigint(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO `list_of_positions` (`id_list`, `id_pos`, `id_doctor`) VALUES
(60, 1, 15),
(62, 3, 23),
(63, 7, 25),

```

(64, 8, 24),

(65, 9, 22);

```
CREATE TABLE `patients` (
  `id_patient` bigint(20) NOT NULL,
  `name_pat` varchar(30) DEFAULT NULL,
  `surname_pat` varchar(30) DEFAULT NULL,
  `birth_date` date NOT NULL,
  `phone` varchar(13) DEFAULT NULL,
  `dop_info` varchar(250) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
INSERT INTO `patients` (`id_patient`, `name_pat`, `surname_pat`, `birth_date`, `phone`, `dop_info`)
VALUES
```

(26, 'Аліна', 'Семенова', '1998-05-10', '380958264984', 'Місцева алергія на анальгін'),

(42, 'Вікторія', 'Авраменко', '0000-00-00', '380992345123', ''),

(43, 'Андрій', 'Куриленко', '1999-10-10', '380957612123', 'fff')

```
CREATE TABLE `positions` (
  `id_pos` bigint(10) NOT NULL,
  `name_pos` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
INSERT INTO `positions` (`id_pos`, `name_pos`) VALUES
```

(1, 'Травматолог'),

(3, 'Дерматолог'),

(5, 'Офтальмолог'),

(7, 'Ендокринолог'),

(8, 'Кардіолог'),

(9, 'Невропатолог');

```
CREATE TABLE `prescriptions` (
  `id_prescription` bigint(10) NOT NULL,
  `id_patient` bigint(20) NOT NULL,
  `id_doctor` bigint(20) DEFAULT NULL,
  `date_prescription` date DEFAULT NULL,
```

```

`diagnos` text NOT NULL,
`prescription` varchar(700) DEFAULT NULL,
`recomendation` varchar(700) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

INSERT INTO `prescriptions` (`id_prescription`, `id_patient`, `id_doctor`, `date_prescription`,
`diagnos`, `prescription`, `recomendation`) VALUES

```

```

(51, 26, 15, '2020-05-26', 'Остеохондроз поперекового відділу хребта', 'Напівжорсткий ортез на
попереково-крижовий відділ хребта', 'спостереження у травматолога'),

```

```

(59, 26, 22, '2020-05-28', 'Астенічний синдром', 'Гліцин 1 таблетка в день', 'Уникати стресових
ситуацій');

```

```

CREATE TABLE `services` (
`id` bigint(11) NOT NULL,
`name` varchar(30) DEFAULT NULL,
`price` float DEFAULT NULL,
`info` varchar(500) DEFAULT NULL,
`photo` varchar(250) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

INSERT INTO `services` (`id`, `name`, `price`, `info`, `photo`) VALUES

```

```

(3, 'УЗД', 400, 'Ультразвукове дослідження (УЗД діагностика) – найважливіше обстеження
внутрішніх органів людини і тих процесів, які протікають в них, за допомогою ультразвукових хвиль.
Ультразвукова діагностика (ультразвукове дослідження, УЗД) зайняла одне з провідних місць у
сучасній клінічній медицині.', 'gallery/узд.jpg'),

```

```

(4, 'ЕКГ', 400, 'Електрокардіографія (скорочено ЕКГ) — метод графічної реєстрації електричних
явищ, які виникають у серцевому м'язі під час його діяльності, з поверхні тіла. Криву, яка відображає
електричну активність серця, називають електрокардіограмою (ЕКГ). Таким чином, ЕКГ — це запис
коливань різниці потенціалів, які виникають у серці під час його збудження.', 'gallery/3.jpg'),

```

```

(5, 'Рентген', 300, 'Рентгенографія — дослідження внутрішньої структури об'єктів, які
відображаються за допомогою рентгенівських променів. Найчастіше термін використовують у
медичному контексті. Методика Rö (рентгенівського) дослідження ґрунтується на утворенні тіньового
забарвлення на рентгенівській плівці під дією Rö-променів.', 'gallery/90.jpg');

```

```

ALTER TABLE `doctors`

```



```
ADD PRIMARY KEY (`id`);

ALTER TABLE `list_of_positions`
  ADD PRIMARY KEY (`id_list`),
  ADD KEY `id_doctor` (`id_doctor`),
  ADD KEY `id_pos` (`id_pos`);

ALTER TABLE `patients`
  ADD PRIMARY KEY (`id_patient`);

ALTER TABLE `positions`
  ADD PRIMARY KEY (`id_pos`);

ALTER TABLE `prescriptions`
  ADD PRIMARY KEY (`id_prescription`),
  ADD KEY `id_doctor` (`id_doctor`),
  ADD KEY `id_patient` (`id_patient`);

ALTER TABLE `services`
  ADD PRIMARY KEY (`id`);

ALTER TABLE `doctors`
  MODIFY `id` bigint(11) NOT NULL AUTO_INCREMENT;

ALTER TABLE `list_of_positions`
  MODIFY `id_list` bigint(10) NOT NULL AUTO_INCREMENT;

ALTER TABLE `patients`
  MODIFY `id_patient` bigint(20) NOT NULL AUTO_INCREMENT;

ALTER TABLE `positions`
  MODIFY `id_pos` bigint(10) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=10;

ALTER TABLE `prescriptions`
```

```
MODIFY `id_prescription` bigint(10) NOT NULL AUTO_INCREMENT,  
AUTO_INCREMENT=60;
```

```
ALTER TABLE `services`  
MODIFY `id` bigint(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;
```

```
ALTER TABLE `list_of_positions`  
ADD CONSTRAINT `list_of_positions_ibfk_1` FOREIGN KEY (`id_doctor`) REFERENCES  
`doctors` (`id`) ON UPDATE CASCADE,  
ADD CONSTRAINT `list_of_positions_ibfk_2` FOREIGN KEY (`id_pos`) REFERENCES  
`positions` (`id_pos`) ON UPDATE CASCADE;
```

```
ALTER TABLE `prescriptions`  
ADD CONSTRAINT `prescriptions_ibfk_1` FOREIGN KEY (`id_doctor`) REFERENCES `doctors`  
(`id`) ON UPDATE CASCADE,  
ADD CONSTRAINT `prescriptions_ibfk_3` FOREIGN KEY (`id_doctor`) REFERENCES `doctors`  
(`id`),  
ADD CONSTRAINT `prescriptions_ibfk_4` FOREIGN KEY (`id_patient`) REFERENCES  
`patients` (`id_patient`) ON UPDATE CASCADE;
```