

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіоелектроніки,
Факультет комп'ютерних наук і технологій
(повне найменування інституту, факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ МЕДИЧНОГО КАБІНЕТУ
HOSPITAL ROOM SOFTWARE

Виконав: студент(ка) 4 курсу, групи КНТ-119сп
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)
Інженерія програмного забезпечення

Тимченко І.В.

(прізвище та ініціали)

Керівник Гофман Є.О.

(прізвище та ініціали)

Рецензент Коцур М.І.

(прізвище та ініціали)

2022

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Інститут, факультет ІРЕ, ФКНТ
 Кафедра програмних засобів
 Ступінь вищої освіти бакалавр
 Спеціальність 121 Інженерія програмного забезпечення
 (код і найменування)
 Освітня програма (спеціалізація) Інженерія програмного забезпечення
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
 “ ” 2022 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Тимченка Івана Вікторовича

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Програмне забезпечення медичного кабінету. Hospital Room Software

керівник проєкту (роботи) Гофман Євгеній Олександрович, к.т.н.,
 (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “27” квітня 2022 року № 102

2. Строк подання студентом проєкту (роботи) 7 червня 2022 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	Гофман Є.О., доцент		
Нормоконтроль	Белова А.В., асистент		

7. Дата видачі завдання “ 27 ” квітня 2022 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки	2 тиждень	Розділ 2
4	Розробка архітектури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділи 3,4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

(підпис) Тимченко І.В.
(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис) Гофман Є.О.
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
105 с., 48 рис., 3 табл., 4 дод., 21 джерел.

БАЗА ДАНИХ, МЕДИЧНИЙ КАБІНЕТ, МОВА ПРОГРАМУВАННЯ,
ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, СИСТЕМА КЕРУВАННЯ ВМІСТОМ

Об'єкт дослідження – процес розробки програмного забезпечення для медичної діяльності.

Предмет дослідження – програмні засоби для підтримки процесу надання медичних послуг.

Метою роботи є розробка програмного забезпечення для медичного кабінету.

Матеріали, методи та технічні засоби: мова програмування PHP, система керування вмістом WordPress, система управління базами даних MySQL.

Результати. Розроблено програмне забезпечення для медичного кабінету, яке дозволяє клієнтам цього закладу здійснювати онлайн запис на прийом та переглядати інформацію про цей кабінет.

Практична цінність роботи полягає у розробці програмного забезпечення для підтримки процесу надання медичних послуг.

Висновки. Розроблено програмне забезпечення для медичного кабінету, що дозволяє спростити процес отримання інформації про лікарів та запису на прийом.

Галузь використання – медична діяльність.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 105 pages, 48 figures, 3 tables, 4 appendixes, 21 sources.

DATABASE, MEDICAL OFFICE, PROGRAMMING LANGUAGE, SOFTWARE, CONTENT MANAGEMENT SYSTEM

The object of research is the process of software development for medical activities.

The subject of the research is software to support the process of providing medical services.

The purpose of this work is to develop software for the medical office.

Materials, methods and technical tools: PHP programming language, WordPress content management system, MySQL database management system.

Results. The software tools for the medical office, which allows the clients of this institution to make an online appointment and view information about this office have been developed.

The practical value of the work lies in the development of software to support the process of providing medical services.

Conclusions. The software for the medical office, which simplifies the process of obtaining information about doctors and appointments is developed.

Scope of use – medical activities.

ЗМІСТ

	С.
Перелік скорочень та умовних познач	8
Вступ	9
1 Аналіз предметної області	11
1.1 Поняття про програмне забезпечення для підтримки медичної діяльності	11
1.2 Дослідження програмних засобів для підтримки медичної діяльності	21
1.3 Висновки за розділом 1	31
2 Матеріали і методи	33
2.1 Вибір мови програмування та інших технологій розробки	33
2.2 Вибір системи управління вмістом для створення програмного забезпечення	36
2.3 Висновки за розділом 2	39
3 Опис програми	40
3.1 Структура програмного забезпечення медичного кабінету	40
3.2 Функціонування програмного забезпечення медичного кабінету	41
3.3 Проєктування інтерфейсу взаємодії користувача з програмним забезпеченням медичного кабінету	45
3.4 Розширення функціональності програмного забезпечення медичного кабінету за допомогою плагінів	48
3.5 Висновки за розділом 3	41
4 Експлуатація, тестування та експериментальне дослідження програми	53
4.1 Призначення й умови застосування програми	53
4.2 Характеристики програми	54
4.3 Встановлення серверу та системи керування вмістом для функціонування програмного забезпечення медичного кабінету	54
4.4 Тестування та результати роботи програми медичного кабінету	57
4.5 Висновки за розділом 4	61

Висновки	62
Перелік джерел посилання	64
Додаток А Технічне завдання	66
Додаток Б Опис програми	70
Додаток В Текст програми	73
Додаток Д Слайди презентації	91

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

CMS – Content Management System (система керування вмістом);

EHR – Electronic Health Record (електронна медична картка);

HMS – Hospital Management System (система управління лікарнями);

RPM – Remote Patient Monitoring (віддалений моніторинг пацієнта);

ГПІ – графічний програмний інтерфейс;

ПЗ – програмне забезпечення;

СКБД – система керування базами даних;

ТЗ – технічне завдання.

Вступ

Проблема створення програмного забезпечення для підтримки медичної діяльності є досить актуальною [1]-[7].

У наш час використовуються різні програмні засоби для підтримки медичної діяльності. До таких програм відносяться програми для медичного діагностування, програми керування медичною апаратурою, програмне забезпечення процесу управління медичною діяльністю та ін. Проте відомі програми є досить дорогими, як, правило, досить вузькоспеціалізованими та не дуже зручними до використання, що ускладнює їх застосування на практиці.

Головною метою розробки є створення програмного забезпечення у вигляді вебсайту для медичного кабінету, який допоможе знайти нових клієнтів та спростити процес отримання інформації про лікарів та запису на прийом. У роботі передбачається створення зручної системи з інтуїтивно зрозумілим інтерфейсом, яка дозволить користувачу переглядати загальну інформації про лікарів та медичний заклад.

У дипломній кваліфікаційній роботі бакалавра розв'язується актуальне завдання розробки програмного забезпечення для медичного кабінету.

Для досягнення поставленої мети у кваліфікаційній роботі бакалавра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для підтримки медичної діяльності;
- здійснити проєктування програмного забезпечення для медичного кабінету;
- створити програмне забезпечення для відображення інформації про роботу медичного кабінету;
- виконати тестування розробленого програмного забезпечення для медичного кабінету.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Поняття про програмне забезпечення для підтримки медичної діяльності

Охорона здоров'я є однією з найскладніших галузей для цифровізації, оскільки в ній задіяно багато зацікавлених сторін: від лікарень і медичних центрів, через лікарів, фармацевтичних та дослідницьких компаній до самих пацієнтів. Логістика програм охорони здоров'я завжди була однією з найскладніших. Деякі країни рухаються швидше, ніж інші. Зацікавлені сторони охорони здоров'я в усьому світі погоджуються, що цифрові технології – це шлях вперед. У наш час існують різні типи програмного забезпечення для охорони здоров'я зробили революцію в галузі [1]–[8].

Електронні медичні картки (EHR), ймовірно, найпопулярніший і найважливіший тип програмного забезпечення для охорони здоров'я. Системи EHR зберігають дані пацієнтів: медичний та сімейний анамнез, результати лабораторних та інших тестів, історію призначених ліків тощо. Найважливішою особливістю EHR є те, що інформацію про стан здоров'я пацієнта можуть створювати та керувати уповноважені постачальники, а потім передавати її іншим постачальникам у більш ніж одній медичній організації. Це не тільки автоматизує та оптимізує робочий процес постачальників, оскільки їм не потрібно знову збирати ту саму інформацію від одного і того ж пацієнта, але також помітно покращує догляд за пацієнтом. EHR зменшують кількість помилок, покращують безпеку пацієнтів і підтримують кращі результати лікування [1]–[8].

Деякі з EHR не тільки ведуть записи про ліки та алергію пацієнта, але й автоматично перевіряють інтегровану базу даних ліків, коли призначають новий препарат, щоб попередити лікаря про потенційний конфлікт або алергію. Вони часто включають фінансовий модуль для виставлення рахунків і оплати з

платіжною інформацією, і всі вони: окремий портал для доступу пацієнта до своїх даних [1]–[8].

Загалом, EHR зменшують ризик реплікації даних, втрати даних або важливої інформації про здоров'я пацієнта. Вони також економічно ефективні в порівнянні з паперовими медичними документами, які, швидше за все, будуть дублюватися, нерозбірливі або знищені протягом тривалого періоду часу. EHR використовуються як у клініках, так і в лікарнях і зазвичай інтегровані з програмним забезпеченням для внутрішнього управління лікарнями/клінікою [1]–[8].

Програмне забезпечення для телемедицини. Телемедицина, тобто онлайн-консультації лікаря через веб- та мобільні застосунки, досягла повного попиту, коли звичайні медичні мережі були порушені, і багато людей не могли відвідувати свого лікаря для регулярних зустрічей віч-на-віч. Цей попит виявився каталізатором глобальної революції в галузі цифрової охорони здоров'я [1]–[8].

Тепер потреба в програмному забезпеченні для телемедицини, можливо, впала з найвищого рівня, але цей тип медичних застосунків назавжди закріпив свою позицію в цифровій медицині. Covid-19 знищив основний бар'єр для постачальників дистанційних консультацій. На ринку з'являється все більше і більше телемедичних програм і послуг, які пропонують різноманітні функції, доступні на одній платформі: від відеоконференцій, передачі файлів (для результатів тесту пацієнта) до оплати за надані послуги [1]–[8].

Віддалений моніторинг пацієнта (Remote Patient Monitoring, RPM) може збирати дані про пацієнтів за межами закладів охорони здоров'я (клінік або лікарень) і, як наслідок, отримувати більш детальну інформацію про здоров'я пацієнта або навіть допомагати проводити дистанційну діагностику на основі цих даних. Програмне забезпечення віддаленого моніторингу пацієнта також може попередити лікаря або клініку, якщо виявлено будь-які

відхилення. Подібно до послуг телемедицини, віддалений моніторинг пацієнтів набрав обертів під час пандемії, коли старі процеси управління здоров'ям були порушені [1]–[8].

Деякі технології віддаленого моніторингу пацієнтів включають монітори серцевого ритму та артеріального тиску, глюкометри та кисень у крові, носимі монітори ЕКГ. Загалом, технологія та пристрої RPM можуть помітно покращити медичні послуги вдома для людей із хронічними захворюваннями, людей, які відновлюються після операції, та літніх пацієнтів [1]–[8].

Системи управління лікарнями (Hospital Management Systems, HMS) – програмне забезпечення для управління лікарнями, яке надає як інформацію, так і можливості управління трьома групами зацікавлених сторін, присутнім у лікарні: адміністрації, лікарям та уповноваженим медичним працівникам та пацієнтам [1]–[8].

HMS часто включає два окремих портали, один з орієнтованими на управління функціями, і один портал, доступний для пацієнтів. Панель управління містить усю адміністративну та медичну інформацію, необхідну лікарні - від управління харчуванням і ліками, бронювання кімнат і розкладу, розкладів лікарів та їх запланованих прийомів. Але також і зовсім звичайні елементи, такі як керування пральнею, витрати на комунальні послуги, споживання води та електроенергії, інвентаризація матеріалів тощо [1]–[8].

Портал для пацієнтів має містити: цифрову форму реєстрації пацієнтів, всю інформацію про прийом (вона часто синхронізується з електронними медичними записами), в деяких випадках, платіжну інформацію пацієнта, страховку, повідомлення про прийом, процедури, прийоми до лікаря та всі ліки та інформація про рецепти [1]–[8].

Системи управління лікарнями не тільки покращують загальну продуктивність лікарень, оптимізують рутину, але й покращують досвід

пацієнтів під час будь-якої взаємодії. HRS також дозволяє лікарням збирати та аналізувати цінні дані щодо всіх аспектів їхньої діяльності [1]–[8].

Програмне забезпечення для управління взаємовідносинами з клієнтами (CRM) у сфері охорони здоров'я допомагає лікарям і клінікам підтримувати відносини з пацієнтами. Вона не відрізняється за призначенням від CRM інших галузей, оскільки служить для автоматизації продажу послуг, реєстрації послуг та персоналізації на основі наданих даних [1]–[8].

CRM Healthcare надає дані про щоденну роботу клініки, оскільки вони можуть зберігати дані клієнтів: історію хвороби, відвідування, рахунки. Вони можуть надсилати пацієнтам автоматичні сповіщення про майбутні огляди або візити для поповнення рецептів. Загалом, їх мета — оптимізувати всі адміністративні завдання клініки та дозволити швидше робити висновки на основі даних. Завдяки CRM клініка може краще відстежувати задоволеність пацієнтів, швидше визначати повторювані проблеми та автоматизувати повсякденні адміністративні завдання [1]–[8].

Програмне забезпечення для електронних рецептів може відрізнятися від країни до країни, оскільки воно дуже залежить від державних правил. Основна вартість, однак, не змінюється. Програмне забезпечення для електронних рецептів дозволяє лікарям не тільки виписувати нові рецепти, але й відстежувати історію рецептів, поновлювати або скасувати їх у разі потреби. У деяких країнах для зручності програмне забезпечення дозволяє безпосередньо зв'язатися з вибраною вами аптекою [1]–[8].

Загалом, програмне забезпечення для електронних рецептів збільшує безпеку та час пацієнтів та лікарів, а також дозволяє лікарям перевіряти будь-які конфлікти з уже прийнятими ліками [1]–[8].

Програми для медичного діагностування – можуть бути поділені на два типи: програми для пацієнтів і тип діагностичне програмне забезпечення для медичних працівників [1]–[8].

Застосунки для діагностики для пацієнтів часто є простими анкетами або чат-ботами, які служать перевіркою симптомів, що надає пацієнтам знання, засновані на медичній експертизі. Вони стали широко використовуваним інструментом під час пандемії, який допоміг пацієнтам зорієнтуватися через можливі симптоми коронавірусної інфекції. Багато урядів використовували це рішення як частину своєї стратегії. Але ці програми для самодіагностики можна використовувати в багатьох інших станах, щоб допомогти пацієнтам, які можуть відчувати себе безсилими і втраченими перед чорною дірою симптомів, доступною в Google [1]–[8].

Інший тип програмного забезпечення для діагностики призначений для медичних працівників. Програмне забезпечення для діагностики на основі штучного інтелекту може допомогти лікарям збирати й аналізувати дані та визначати неспецифічні симптоми захворювання, що сприяє швидшій діагностиці та дослідженню. Вся анонімна інформація може передаватися закладам охорони здоров'я [1]–[8].

Програмне забезпечення для запису на прийом використовується лікарями, приватними клініками, а останнім часом і лікарнями. Зазвичай він складається з двох панелей, окремої панелі, зверненої до пацієнта, та панелі управління (з боку клініки). Його головна мета цілком зрозуміла: це спростити процес запису на прийом до лікаря. Пацієнти можуть легко записатися на прийом, отримувати повідомлення про це, заздалегідь надати лікарю відповідну інформацію, а також перенести або скасувати прийом [1]–[8].

З боку керівництва, це дозволяє клініці мати прозорий і легко аналізований графік роботи своїх лікарів. Для прийому клініки це хороший спосіб спростити їхні повсякденні завдання щодо підтвердження та повторного бронювання всіх запланованих прийомів [1]–[8].

Програми для відстеження здоров'я, безумовно, є найширшою групою програмного забезпечення для охорони здоров'я, оскільки вони можуть включати: відстеження фізичної активності та руху, дієти та схуднення, а також

останнім часом дуже популярний тип: програми для психічного здоров'я та благополуччя [1]–[8].

Аргумент про те, що це квазімедичні програми і не пов'язані безпосередньо з професійним медичним програмним забезпеченням, безумовно, застарів. Особливо у світлі відшкодування Німеччиною витрат на цифрові програми для здоров'я, тобто, іншими словами: дозволити лікарям призначати цифрові програми як частину лікування своїх пацієнтів [1]–[8].

Застосунки для відстеження здоров'я для споживачів мають великий потенціал не тільки для популяризації догляду за собою, але також можуть служити джерелом даних про стан здоров'я пацієнта, щоб стимулювати персоналізовану медичну допомогу. Дуже часто вони є додатками-супутниками для носимих пристроїв, причому найпопулярнішими є Apple Watch, Fitbit для ваших випадкових даних, таких як рух, пульс, сон. Але не виключно: це також кільця, намиста та інші браслети, створені стартапами, які виходять за рамки цього і спрямовані на допомогу при фертильності (вимірювання температури вашого тіла та конкретного дня менструального циклу), діабету тощо [1]–[8].

Програмне забезпечення медичних рахунків призначене для фінансових операцій в установах охорони здоров'я. Він допомагає керувати рахунками-фактурами, квитанціями та страховими відшкодуваннями, а також дозволяє здійснювати всі операції. Програмне забезпечення для виставлення рахунків у сфері охорони здоров'я все ще доступне як окремі програми, але багато систем управління лікарнями також пропонують цю функцію [1]–[8].

Програмне забезпечення для керування клінічними випробуваннями дозволяє медичним працівникам точно планувати, відстежувати та керувати своїми клінічними дослідженнями. Вони часто є веб-рішеннями, які забезпечують дотримання вимог, контролюють реєстрацію або набір пацієнтів та керують фінансами. Програмне забезпечення СТМ забезпечує гнучкість, часто дозволяє співпрацювати в режимі реального часу, а завдяки тому, що

воно є веб-базованим, клінічні групи можуть отримати доступ до нього з будь-якого місця та в будь-який час, без будь-якого іншого програмного забезпечення. Це також ключова перевага програмного забезпечення для управління клінічними випробуваннями: воно забезпечує централізоване та єдине сховище для всіх даних клінічних досліджень, покращуючи збір даних, їх якість та узгодженість [1]–[8].

Важливим програмним забезпеченням для підтримки медичної діяльності є програми управління лікарнями (Hospital Management Systems). Щоб створити універсальну, ефективну та зручну систему, розробники повинні повністю усвідомити її майбутні функції, користувачів і передбачувані переваги. Лише після цього кроку вони зможуть спланувати та координувати свої зусилля, вибрати найбільш підходящі технології, передбачити потенційні проблеми та знайти рішення. Лікарняна інформаційна система управління, як випливає з її назви, являє собою складний набір програмного забезпечення із взаємопов'язаних програмних модулів, які служать для надання як інформації, так і можливостей керування широким колом користувачів [1]–[8].

Передбачуваних користувачів системи управління лікарнями можна в цілому розділити на три категорії [1]–[8]:

- адміністрація лікарні;
- лікарі та інші уповноважені працівники;
- пацієнти.

Кожна категорія повинна мати окремий набір правил для оформлення та обмежень в системі. Цей набір залежить від ролі користувача і варіюється від необмеженого доступу, наданого найвищим членам адміністрації, до обмеженого доступу до певної частини інформації для пацієнтів. Наприклад, лікар повинен мати доступ до медичних даних своїх пацієнтів, але не повинен отримувати інформацію про пацієнтів інших лікарів без дозволу. Кожен користувач повинен мати унікальний ідентифікатор, який пов'язує особу з рангом цієї особи в системі. Користувачу з вищими рангами та більшими

дозволами в системі можуть знадобитися додаткові засоби авторизації. Такі засоби авторизації можуть включати, серед іншого, відбиток пальця сканування обличчя, карту-ключ, USB-ключ або голосову авторизацію [1]–[8].

Основні характеристики системи управління лікарнями: розробники програмного забезпечення повинні розуміти, що система управління лікарнями необхідна для забезпечення різноманітного набору дуже специфічних функцій і функцій, деякі з яких не використовуються в інших галузях [1]–[8].

Функції, орієнтовані на управління [1]–[8]:

- розклад лікарів і планові прийоми. Це програмне забезпечення дозволяє лікарям планувати та переглядати свої розклади. Також воно дозволяє ділитися такою інформацією з пацієнтами, щоб записатися на прийом [1]–[8];

- модуль моніторингу, управління та інформації аптеки. Надає дані щодо ліків і витратних матеріалів, які на даний момент є на складі, і дозволяє замовити їх заздалегідь;

- графік доступності операційної та управління;

- управління харчуванням для пацієнтів та лікарняної їдальні;

- керівництво палатою. Надає різні типи звітів про планування, відстеження та результати діяльності окремо для кожного окремого відділення [1]–[8];

- контроль запасів і управління закупівлями. Займається будь-якими витратними матеріалами, які не належать до аптеки. Також включає відстеження палива для машин швидкої допомоги;

- витрати на комунальні послуги, воду, електроенергію.

Як видно, ця група функцій є досить важливою для адміністрації лікарні, оскільки допомагає оптимізувати роботу, координувати поточні ресурси та співробітників, збільшити прибуток [1]–[8].

Функції, орієнтовані на пацієнта [1]–[8]:

- реєстрація пацієнтів. Дозволяє зареєструвати пацієнта, який приходить до клініки з метою отримання Особистої медичної книжки, наданої

цією особою або членом її сім'ї. Це зазвичай робиться шляхом заповнення шаблонної форми [1]–[8];

- інформація про прийом пацієнтів. Ця функція збирає та веде електронну медичну карту після госпіталізації людини. Цей модуль також взаємодіє з іншими медичними базами даних, щоб отримати та оновити повну електронну медичну карту пацієнта [1]–[8];

- виставлення рахунків для пацієнтів, страхування та відстеження кредиту;

- модуль невідкладної допомоги. Дозволяє прискорити процес реєстрації та прийому під час догоспітального лікування, наприклад, у відділеннях невідкладної допомоги [1]–[8];

- повідомлення пацієнту. Повідомляє пацієнтів про їхні прийоми, час прийому ліків, рекомендації щодо виписки тощо. Цю функцію може надавати окремий веб- або мобільний додаток, пов'язаний із системою управління лікарнею. Функцію нагадування також можна надати за допомогою наявних програм месенджера [1]–[8];

- інформація про аптеку, бронювання та покупка для пацієнтів. Дозволяє пацієнту відразу придбати певний препарат або замовити його заздалегідь [1]–[8].

Цей перелік зразкових функцій далеко не повний, однак він має вирішальне значення для розуміння того, як розробити систему управління якістю в лікарні [1]–[8].

Чотири основні вимоги до проекту програмного забезпечення управління лікарнею [1]–[8]:

- дружній і зручний дизайн інтерфейсу є одним із головних пріоритетів для таких застосунків, враховуючи велику кількість потенційних користувачів. І медичний персонал, і пацієнти потребують інформативної та зручної системи, якою легко ефективно користуватися незалежно від віку та навичок роботи з комп'ютером [1]–[8];

– здатність збирати, зберігати, обробляти та виводити великі обсяги інформації в різних формах. Програмне забезпечення необхідне для автоматизації для роботи з величезною кількістю текстових даних, які охоплюють різні типи лікарняних документів: від історії хвороби пацієнтів та страховки до документів та скарг клінічних досліджень. Крім того, система повинна бути сумісна з різними типами медіа-інформації: відео та цифровими зображеннями, такими як записані операції, результати комп'ютерної томографії та ін. [1]–[8];

– дотримання всіх чинних правил і протоколів безпеки даних з метою захисту конфіденційності даних та запобігання несанкціонованому доступу до системи. Забезпечення безпеки даних є ще одним головним пріоритетом, оскільки дані пацієнтів та інші документи лікарні є дуже конфіденційними. Будь-який витік інформації, а також втрата або пошкодження даних може призвести до потенційно руйнівних результатів, включаючи судові процеси та інші фінансові наслідки, а також до зіпсованої репутації, що може призвести до закриття лікарні [1]–[8];

– швидке реагування є необхідністю програмного забезпечення для управління лікарнею, тому що часто час є важливим у медицині. Система повинна забезпечувати швидку реєстрацію нових пацієнтів у разі надзвичайних ситуацій, а також швидке отримання відповідних даних, таких як історія пацієнта, з системного архіву чи баз даних інших лікарень [1]–[8].

Зазначені функції та вимоги визначають вибір інструментів і технологій, що використовуються для розробки програмного забезпечення управління лікарнями [1]–[8].

Таким чином, у наш час існує досить багато видів програмного забезпечення, що використовується для підтримки медичної діяльності. Важливим є програмне забезпечення управління лікарнею, розробці якого призначена ця робота.

1.2 Дослідження програмних засобів для підтримки медичної діяльності

У наш час розроблено досить багато програмного забезпечення, що застосовується для підтримки медичної діяльності. Проаналізуємо деякі з таких програмних засобів.

Програмне забезпечення Teladoc – це програма для організації спілкування між пацієнтами та лікарями. Teladoc — це цілодобова медична служба (рис. 1.1), яка вирішує звичайні захворювання та проблеми зі здоров'ям, що дозволяє пацієнтам спілкуватися з лікарями за допомогою настільного застосунку чи телефону [9].

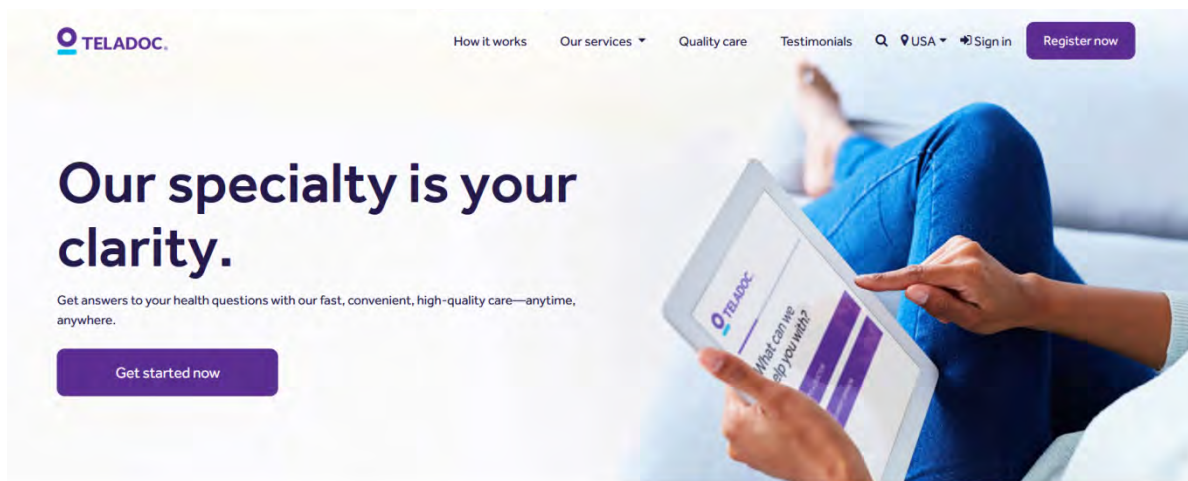


Рисунок 1.1 – Програмне забезпечення Teladoc

Для пацієнтів це означає зручний догляд, менше часу на очікування та доступ до підтримки для рутинного чи регулярного догляду в будь-який час – замість того, щоб покладатися на години роботи лікарів, щоб отримати відповіді [9].

Медичне програмне забезпечення Teledoc для лікарів означає набагато більш гнучкий режим роботи та можливість бути доступними для пацієнтів,

коли потрібна допомога. Вони можуть дистанційно лікувати людей, відповідати на запитання та виписувати рецепти [9].

Програмне забезпечення Teledoc зосереджено на питаннях, які не стосуються надзвичайних ситуацій. Оскільки Teladoc створено для обслуговування людей, які просто хочуть легкого доступу до рутинної медичної допомоги або догляду, це працює на користь усіх [9].

Ключові переваги та особливості програмного забезпечення Teledoc [9]:

- цілодобове обслуговування пацієнтів;
- надає пацієнтам доступ до загальнонаціональної мережі лікарів;
- доступний через комп'ютер або мобільний застосунок;
- надає більше гнучкості лікарям і пацієнтам.

Недоліком програмного забезпечення Teledoc є те, що воно містить багато функцій, що дещо ускладнює роботу користувачів з цим ПЗ [9]. Іншим недоліком програми є те, що служба медичної підтримки Teledoc працює тільки на території США та Канади. Крім того, відсутня україномовна локалізація [9].

Програмне забезпечення 10to8 є безкоштовною платформою планування медичного програмного забезпечення, розробленою компанією Capterra, є прекрасним вибором для медичних практик, які прагнуть оптимізувати процес планування пацієнтів. Система онлайн-бронювання (рис. 1.2) призначена для того, щоб призначати зустрічі для всіх, і є одним із найбільш зручних інструментів у цьому списку [10].

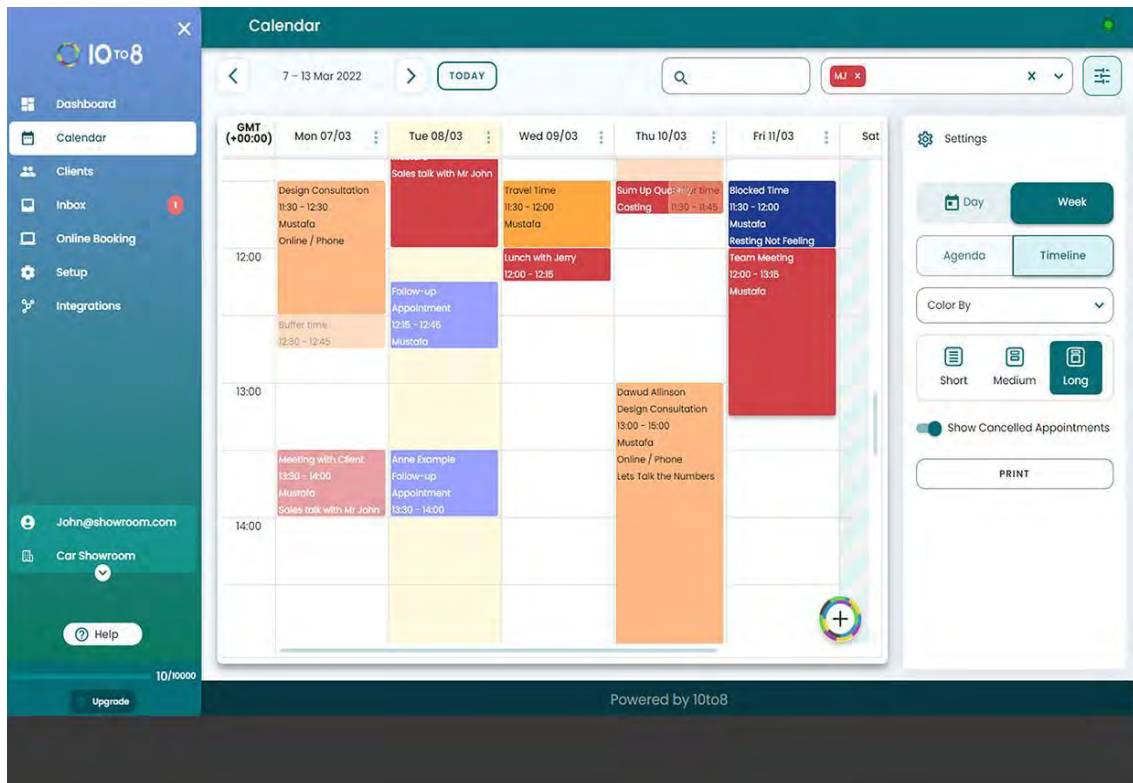


Рисунок 1.2 – Програмне забезпечення 10to8

Програмне забезпечення 10to8 є досить ефективним медичним програмним засобом. Воно має ефективне багатоканальне планування зустрічей, сторінки бронювання, є повністю сумісним із програмами зчитування з екрана. Програмне забезпечення 10to8 включає в себе автоматизовану систему бронювання по телефону, що значно спрощує життя як медичних працівників, так і реєстраторів і пацієнтів [10].

Ключовими перевагами та особливостями програмного забезпечення 10to8 є такі [10]:

- сторінки бронювання пацієнтів із самообслуговуванням, які зменшують роботу адміністратора та спрощують планування зустрічей як для особистих, так і віддалених консультацій;
- сторінки бронювання, сумісні з програмами зчитування з екрана;
- набір доступності, який включає цілодобове бронювання телефону без адміністратора;

- нагадування для пацієнтів електронною поштою, SMS та голосовими дзвінками;
- має портал для пацієнтів, де пацієнти можуть бачити та керувати своїми майбутніми прийомами;
- пропонує інструменти оброблення захищеної медичної інформації.

Програмне забезпечення 10to8 було впроваджено в клініках флеботомії та інших постачальників вторинної медичної допомоги, таких як фізіотерапевти та клініки ультразвукового дослідження, а також у деяких постачальників невідкладної та первинної допомоги. Це інструмент планування пацієнта, який підходить для широкого кола медичних практик [10].

Недоліком програмного забезпечення 10to8 є те, що воно містить багато функцій, що дещо ускладнює роботу користувачів з цим ПЗ [10]. Іншим недоліком програми є те, що у ньому відсутня україномовна локалізація [10].

Програмне забезпечення QuantiaMD представляє собою розумну платформу, схожу на інтуїтивно зрозумілий простір у соціальних мережах для лікарів. QuantiaMD – це простір для навчання та співпраці, який об'єднує лікарів з усього світу, дозволяє їм спілкуватися, ділитися своїми ідеями та вчитися один у одного. Це також допомагає лікарям і персоналу краще керувати своїми закладами, що потім позитивно впливає на загальний показник задоволеності пацієнтів [11].

Спільне навчання має феноменальний вплив як на розвиток нових медичних технологій, так і на загальний догляд за пацієнтами, тому ця програма буквально може змінити наше життя [11].

Крім того, було показано, що платформа підвищує рівень управління та задоволеності лікарень. Це досягається шляхом мінімізації відсотка реадмісії та значного зниження рівня вигорання на робочому місці, з яким стикаються лікарі та лікарі, що може вплинути на ефективність усієї практики [11].

Ключовими перевагами та особливостями програмного забезпечення QuantiaMD є такі [11]:

- дозволяє лікарям з усього світу об'єднуватися у групи для розв'язання певних питань;
- сприяє обміну знаннями та співпраці;
- може допомогти зменшити кількість повторних госпіталізацій пацієнтів;
- може зменшити вигорання на робочому місці для медичних працівників [11].

QuantiaMD використовується всіма типами медичних працівників у всьому світі для співпраці та спілкування, від клінік до лікарень [11].

Недоліком програмного забезпечення QuantiaMD є те, що воно не дозволяє розв'язувати багато медичних проблем, а лише допомагає в організації контактів та спілкування для їх розв'язання [11]. Іншим недоліком програми є те, що у ньому відсутня україномовна локалізація [11].

Програмне забезпечення Amplitude є провідним програмним забезпеченням, що надає послуги збору та оброблення клінічних даних. Воно використовується для збору інформації про пацієнтів. Існують різні платформи, доступні для використання окремими клініцистами або командами, і кожна платформа надає трохи інші функції [12].

Платформа в основному використовується для клінічних результатів та результатів, про які повідомляють пацієнти, віртуальних клінік спостереження та сортування (рис. 1.3), а також моніторингу пацієнтів із хронічними захворюваннями [12].



Рисунок 1.3 – Програмне забезпечення Aplitude

Програмне забезпечення для охорони здоров'я Aplitude [12] є рішенням для лікарень, які намагаються знайти баланс між ефективністю та ефективністю. Величезна кількість попередньо завантажених даних гарантує, що клініцисти мають відповідну інформацію під рукою. Це також дозволяє їм додавати точні коригування для кожного випадку та ефективніше контролювати прогрес пацієнта [12].

Онлайн-система Aplitude дає пацієнтам свободу доступу до платформи, де й коли вони забажають. Крім того, зручна інформаційна панель забезпечує місце, куди лікарі можуть вводити лікування та дані про пацієнта, не заповнюючи будь-які документи [12].

Ключовими перевагами та особливостями програмного забезпечення Aplitude є такі [12]:

- можливість використання різних платформ для підтримки діяльності медичних практик;
- до повністю віртуальної системи можна отримати доступ з будь-якого місця;
- попередньо завантажені дані дають клініцистам легкий доступ до інформації;

– інтуїтивно зрозумілий інтерфейс користувача.

Недоліком програмного забезпечення Aplitude є те, що воно містить багато функцій, що дещо ускладнює роботу користувачів з цим ПЗ [12]. Іншим недоліком програми є те, що у ньому відсутня україномовна локалізація [12].

Програмне забезпечення EasyClinic розроблено для безперебійного керування медичною практикою, є ефективним і практичним програмним рішенням для лікарів, лікарів і навіть адміністраторів (рис. 1.4) [13].

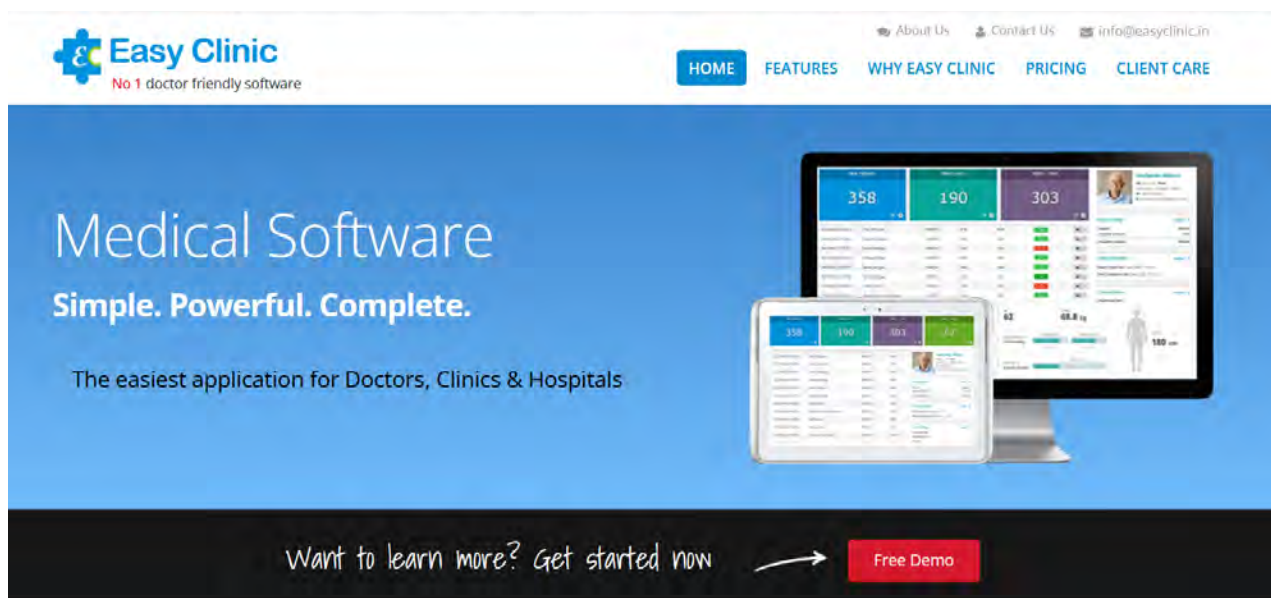


Рисунок 1.4 – Програмне забезпечення EasyClinic

EasyClinic дозволяє медичним клінікам, лікарям загальної практики та лікарням зберігати медичні дані та записи пацієнтів у хмарі. Це означає, що лікарі можуть безпечно отримати доступ до історичних даних пацієнта в будь-який час і надати допомогу, призначити ліки або призначити обстеження [13].

EasyClinic також дозволяє лікарям зберігати файли, порівнювати минулі дані пацієнтів, отримувати доступ до ста інструментів аналітики. Він також надає пацієнтам портал, щоб вони також могли отримати доступ до власної інформації [13].

Платформа доступна в режимі онлайн або в автономному режимі, тому вам не завадить повільне підключення до Інтернету, і забезпечує повне навчання та підтримку всім своїм користувачам [13].

Ключовими перевагами та особливостями програмного забезпечення EasyClinic є такі [13]:

- надає лікарям безпечний доступ до даних пацієнтів;
- можливість створити портал пацієнтів для клієнтів;
- доступний онлайн або офлайн формат роботи з програмою.

Програмне забезпечення EasyClinic використовується лікарями, клініками та лікарнями, є досить гарним рішенням для низки медичних потреб [13].

Недоліком програмного забезпечення EasyClinic є те, що воно містить багато функцій, що дещо ускладнює роботу користувачів з цим ПЗ [13]. Іншим недоліком програми є те, що у ньому відсутня україномовна локалізація [13].

Програмне забезпечення eConsult є онлайн-консультаційною платформою, яка надає можливість лікарям загальної практики спілкуватися з пацієнтами через портал (рис. 1.5). Пацієнти можуть перевірити свої симптоми, отримати рецепти та поради від своїх лікарів, не чекаючи до наступного прийому. eConsult є найбільш широко використовуваною системою онлайн-порталів, яку використовують понад трьох тисяч медичних практик у Великобританії [14].

Digital triage and remote consultation solutions to meet the needs of your healthcare department.

Outpatients

Q health from eConsult

The only remote consultation platform that offers the option of contacting your patients via phone or video and seamlessly moving between the two.

Discover



Urgent and Emergency Care



Рисунок 1.5 – Програмне забезпечення econsult

Програмне забезпечення eConsult дозволяє лікарям і медсестрам сортувати різні медичні випадки, безпечно розставляти пріоритети пацієнтів і призначати їм правильний догляд, створювати анкети, щоб зробити діагнози більш ефективними. Пацієнти зможуть надати інформацію до свого прийому або замість нього, якщо їм просто потрібен швидкий рецепт, який можна надати дистанційно [14].

Використання цієї платформи дозволяє пацієнтам отримати доступ до дистанційного лікування низки захворювань, від астми до депресії чи діабету, значно покращуючи догляд за пацієнтами, водночас скорочуючи час очікування [14].

Ключовими перевагами та особливостями програмного забезпечення eConsult є такі [14]:

- анкети прискорюють діагностику пацієнта;

- може лікувати пацієнтів дистанційно;
- зменшується час очікування пацієнтів і зменшується час адміністратора для персоналу.

Програмне забезпечення eConsult використовується для надання невідкладної та невідкладної допомоги, первинної медичної допомоги та використовується в тисячах практик NHS у Великобританії [14].

Недоліком програмного забезпечення eConsult є те, що воно містить багато функцій, що дещо ускладнює роботу користувачів з цим ПЗ [14]. Іншим недоліком програми є те, що у ньому відсутня україномовна локалізація [14].

Таким чином, у наш час досить активно використовуються програмні засоби для підтримки медичної діяльності. Порівняльний аналіз програмного забезпечення для підтримки медичної діяльності наведено у табл. 1.1.

Таблиця 1.1 – Програмне забезпечення для підтримки медичної діяльності

Критерій, за яким порівнюють	Teledoc	10to8	QuantiaMD	Amplitude	EasyClinic
Можливість цілодобового обслуговування пацієнтів	+	+	–	+	+
Мобільний застосунок	+	+	–	+	+
Доступ через ПК	+	+	+	+	–
Зручність інтерфейсу	–	+	+	–	+
Безкоштовність	–	+	–	+	–
Україномовна версія	–	–	–	–	–

Як видно з таблиці, не всі відомі програмні засоби підтримують україномовну локалізацію, крім того, більшість програм не є безкоштовними, деякі є досить складними у використанні, оскільки підтримують дуже багато функцій, які часто є досить специфічними та не є корисними для більшості

медичних закладів, що суттєво ускладнює використання такого програмного забезпечення на практиці. Отже доцільною є розробка програмного забезпечення для медичного кабінету, що усуває зазначені недоліки.

Таким чином, необхідно створити програмний продукт для медичного кабінету. При розробці програми для медичного кабінету необхідно забезпечити такі функціональні вимоги:

- наявність зручної адміністративної панелі та сховища даних;
- перегляд записів адміністратора;
- можливість наповнення сайту інформацією;
- зручний та зрозумілий інтерфейс користувача;
- проста процедура запису на прийом для користувачів.

1.3 Висновки за розділом 1

Проаналізовано загальні поняття про програмне забезпечення для підтримки медичної діяльності. Визначено, що важливим типом програмного забезпечення для підтримки медичної діяльності є системи управління лікарнями, що надають як інформацію, так і можливості управління трьома групами зацікавлених сторін, присутнім у лікарні: адміністрації, лікарям та уповноваженим медичним працівникам та пацієнтам.

Досліджено програмні засоби для підтримки медичної діяльності, зокрема проаналізовано програми Teledoc, 10to8, QuantiaMD, Amplitude, EasyClinic, eConsult.

Виявлено, що не всі відомі програмні засоби підтримують україномовну локалізацію, крім того, більшість програм не є безкоштовними, деякі є досить складними у використанні, оскільки підтримують дуже багато функцій, які часто є досить специфічними та не є корисними для більшості медичних закладів, що суттєво ускладнює використання такого програмного забезпечення на практиці. Отже доцільною є розробка програмного

забезпечення для медичного кабінету, що усуває зазначені недоліки. Визначено функціональні вимоги до розроблюваного програмного забезпечення медичного кабінету.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування та інших технологій розробки

Для створення програмного забезпечення медичного кабінету було обрано мову програмування PHP [15]-[18].

Мова PHP є широко використовуваною мовою програмування для розробки динамічних вебзастосунків. Багато відомих вебзастосунків, включаючи Facebook і Wikipedia, використовують PHP [15]-[18].

PHP в основному використовується для програмування динамічних вебсайтів. Перевагами мови програмування PHP є такі [15]-[18]:

- мова програмування є досить легкою для використання;
- можливість використання як на стороні клієнта, так і на стороні сервера;
- PHP поставляється з широким набором функцій, що надає велику свободу дій та гнучкість;
- оскільки цією мовою користується дуже багато людей, завжди можна швидко знайти рішення та підтримку для будь-яких проблем, які виникають із PHP. В Інтернеті є велика кількість спеціалізованих форумів і практичних порад щодо PHP. PHP можна використовувати для обробки введених даних користувача, отримання вмісту бази даних або видалення файлів на сервері. Але на мові PHP можна розробляти і інші застосунки. З точки зору продуктивності та ефективності, інші мови програмування, такі як C або C++, все ще перевершують PHP. Тим не менш: 80 відсотків використання PHP на веб-сайтах говорять про це [15]-[18];
- PHP підтримується більшістю вебсерверів;
- PHP може добре працювати з системами керування базами даних та мовою запитів SQL, зокрема з СКБД MySQL;
- PHP може легко обробляти текстові дані, числові дані та рядки.

Аутентифікація користувача за допомогою PHP є досить безпечною, що часто є потрібним для багатьох вебпрограм і є важливою частиною онлайнового світу. Завдяки PHP аутентифікація користувачів можлива без проблем і навіть двофакторна аутентифікація може бути реалізована [15]-[18].

Сеанси входу на основі файлів cookie: завдяки сеансам PHP сеанси входу на основі файлів cookie можуть виконуватися автоматично без необхідності виконувати будь-які дії під час процесу. Дані зберігаються в файлі cookie, і сеанси можна легко розпочинати та закінчувати, і вони є автономними [15]-[18].

Підтримка баз даних: PHP підтримує багато баз даних, включаючи такі: PostgreSQL, MySQL, Microsoft SQL Server, BerkeleyDB та інші. PHP також може допомогти розробити застосунки в Інтернеті для роботи в реальному часі. Будь то чати, соціальні мережі, браузерні ігри чи дані про погоду – завдяки PHP такі програми можна отримати на будь-якому вебсайті [15]-[18].

Якщо планований проект має бути налаштований за допомогою системи керування версіями, PHP часто є його частиною. Теми та плагіни систем керування версіями майже всі засновані на мові програмування PHP [15]-[18].

Багато вебсайтів працюють з даними на мові програмування PHP, які вони отримують з інших вебсайтів або постачальників. Не кожна сторінка новин, яка також показує погоду в одному кутку, має власну метеостанцію. Більшість використовує наявні дані інших провайдерів, які потім потрібно зв'язати з вашим власним сайтом. Мову програмування PHP можна використовувати для створення графіки, наприклад JPEG, лише за допомогою кількох рядків коду. Хоча багато графічних елементів інтерфейсу в основному відображаються за допомогою HTML5 або Java Script, PHP, безумовно, є випробуваною альтернативою для бекенда. Це особливо практично для додатків WordPress, оскільки там, наприклад, стискаються зображення, щоб прискорити час завантаження сторінки [15]-[18].

Як правило, PHP-скрипти запускаються з вебсервера. Однак скрипти також можна виконувати безпосередньо через командний рядок інтерпретатора PHP. PHP підходить для виконання робіт з обслуговування в системі. Отже, PHP не тільки добре підходить для роботи не тільки на сервері, але й для створення програм, що виконуються у фоновому режимі. Мова програмування PHP є мовою з відкритим вихідним кодом, доступною та безкоштовною для всіх поширених операційних систем. Це не єдина причина, чому PHP настільки популярна і, ймовірно, найпоширеніша мова сценаріїв у світі. Порівняно просте використання також переконує програмістів використовувати цю мову. PHP підходить для динамічного вебконтенту, підтримує інтеграцію з багатьма базами даних [15]-[18].

Обґрунтування вибору мови програмування для розробки програмного забезпечення для медичного кабінету наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для медичного кабінету

Критерій порівняння мов програмування	Мова програмування		
	PHP	Java	C++
Підтримка функцій роботи з системами керування базами даних	+	+	+-
Об'єктно-орієнтовані засоби розробки програмного забезпечення	+	+	+-
Зручність для створення ПЗ для медичного кабінету	+	-	-
Засоби захисту даних при передачі між клієнтом та сервером	+	+	+-
Зручність для створення вебзастосунків	+	+	-

Таким чином, для розробки програмного забезпечення для медичного кабінету обрано мову програмування PHP.

2.2 Вибір системи управління вмістом для створення програмного забезпечення

Як систему управління вмістом для створення програмного забезпечення медичного кабінету обрано Wordpress [19]-[20].

Wordpress – це програмне забезпечення, яке дозволяє створювати та підтримувати власний веб-сайт в Інтернеті. WordPress — це система керування вмістом, яка характеризується широким використанням, високим рівнем налаштування та простою конфігурацією [19]-[20].

WordPress є найпопулярнішою системою керування вмістом на ринку, а також найбільш широко використовуваною. Wordpress дозволяє редагувати вебсторінки візуально в режимі онлайн і його можна легко адаптувати до різних вимог. Wordpress майже нескінченно масштабується. У Wordpress є понад 43 000 плагінів, що дозволяє підвищити функціональність розроблюваного програмного забезпечення [19]-[20].

Wordpress, розроблений як програмне забезпечення для ведення блогів у 2003 році, зараз є лідером ринку, коли мова йде про системи керування вмістом. Wordpress — це рішення з відкритим кодом майже для всіх завдань, які сьогодні має виконувати сучасний веб-сайт чи веб-платформа. Існують тисячі функціональних розширень, які також називаються плагінами, які пропонують найкраще рішення майже для всіх вимог. Завдяки великій спільноті, яка стоїть за Wordpress, CMS постійно розвивається і орієнтована в першу чергу на початківців веб-сайтів. Високий рівень зручності та адаптивності забезпечують легкий старт Wordpress [19]-[20].

WordPress є найбільш використовуваною системою керування вмістом на ринку. Багато великих корпорацій використовують WordPress не тільки як програмне забезпечення для розробки вебзастосунків. WordPress також підходить для створення мікросайтів і цільових сторінок для конкретних

підпроектів. Деякі компанії, які використовують WordPress, — це eBay, Sony Music, UPS, Coca Cola, BBC America, The New York Times, Google Ventures і Time [19]-[20].

Зараз WordPress пропонується 83 мовами. WordPress є оптимальним рішенням для розроблення різноманітних застосунків. Wordpress як рішення з відкритим вихідним кодом є лідером ринку систем керування вмістом. Згідно зі статистикою W3Techs, більше половини всіх CMS-сайтів використовують WordPress. Перевага Wordpress полягає в його широкому використанні. Таким чином, система управління вмістом WordPress може бути охарактеризована такими перевагами [19]-[20]:

- велика спільнота користувачів-розробників, які надають один одному підтримку;
- велика кількість вебсайтів, які вже створено за допомогою Wordpress, тому завжди можна швидко знайти відповіді на всі запитання щодо використання Wordpress за допомогою пошукових сервісів;
- понад 43 000 плагінів, які надають потрібну функціональність для задоволення будь-яких вимог замовників;
- велика спільнота розробників, яка забезпечує безпечне та ефективне впровадження та використання WordPress;
- Wordpress оптимізовано для початківців у веброботці, що дозволяє його особливо легко використовувати при створенні вебсайтів;
- ефективні засоби забезпечення безпеки при роботі з важливими даними;
- наявність великої кількості спеціальних тем, що дозволяють надати розроблюваному вебсайту унікальний дизайн [19]-[20].

Wordpress заснований на мові сценаріїв PHP і зберігає всі дані в базі даних на вебсервері. Wordpress можна розглядати як програмне забезпечення, яке працює у власному вебпросторі і, таким чином, динамічно створює вебсайт. WordPress поділяється на фронтенд і бекенд. Фронтенд – це фактично

вебсайт, тобто те, що бачить відвідувач. У серверній частині вебмайстер може вводити вміст, створювати сторінки та публікації, а також завантажувати й публікувати зображення. Бекенд WordPress залишається прихованим від відвідувача веб-сайту. Він використовується вебмайстром для підтримки вебсайту [19]-[20].

Дизайн вебсайту Wordpress можна налаштувати за допомогою активованої теми Wordpress. З технічної точки зору тема складається з файлів CSS і шаблонів, які утворюють основну структуру HTML вебсайту. Тисячі готових безкоштовних і платних тем дають великий вибір, коли справа доходить до веб-дизайну вебсайту Wordpress. Теми Wordpress можна активувати у серверній частині за бажанням. За допомогою плагінів у Wordpress можна оновити власний вебсайт додатковими функціями [19]-[20].

Обґрунтування вибору системи керування вмістом для створення програмного забезпечення медичного кабінету наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору системи керування вмістом для створення програмного забезпечення медичного кабінету

Критерій порівняння систем керування вмістом	Система керування вмістом		
	WordPress	Joomla	Drupal
Зручність застосування засобів для роботи з мовою РНР	+	+-	-
Зручність використання для створення програмного забезпечення медичного кабінету	+	+-	+-
Розповсюдженість використання	+	+-	-
Можливість кросплатформенної розробки	+	+	+-
Наявність засобів та спільнот для допомоги розробникам	+	+-	-

Таким чином, для створення програмного забезпечення медичного кабінету обрано систему керування вмістом WordPress.

2.3 Висновки за розділом 2

Для розробки програмного забезпечення для медичного кабінету обрано мову програмування PHP, яка є мовою з відкритим вихідним кодом, доступною та безкоштовною для всіх поширених операційних систем, є досить простою для використання, підходить для створення динамічного вебконтенту та підтримує інтеграцію з багатьма базами даних.

Як систему управління вмістом для створення програмного забезпечення медичного кабінету обрано Wordpress, яка характеризується широким використанням, високим рівнем налаштування, простою конфігурацією, дозволяє редагувати вебсторінки візуально в режимі онлайн, має велику кількість плагінів, що дозволяє підвищити функціональність розроблюваного програмного забезпечення.

3 ОПИС ПРОГРАМИ

3.1 Структура програмного забезпечення медичного кабінету

Структуру програмного забезпечення медичного кабінету подамо за допомогою структурних діаграм в нотації UML (рис. 3.1 – 3.3).

Загальна діаграма класів програмного забезпечення медичного кабінету зображена на рис. 3.1.

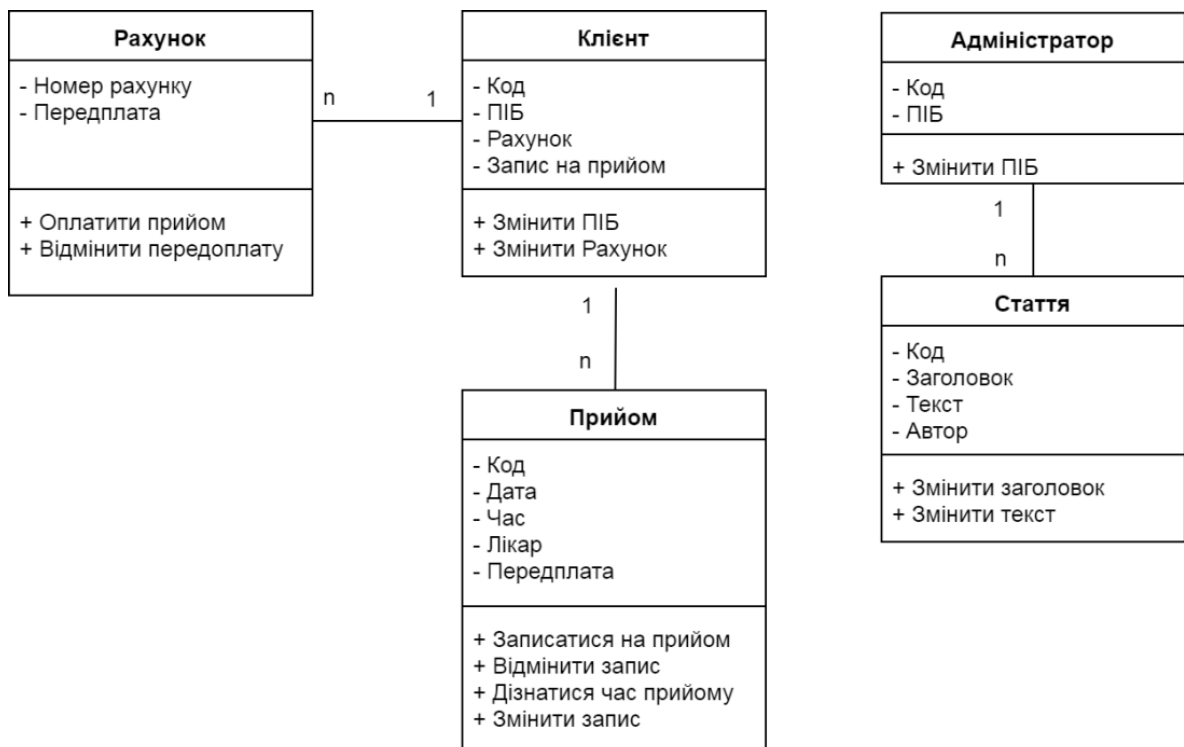


Рисунок 3.1 – Діаграма класів програмного медичного кабінету у нотації UML

Діаграма пакетів програмного забезпечення медичного кабінету зображена на рис. 3.2.

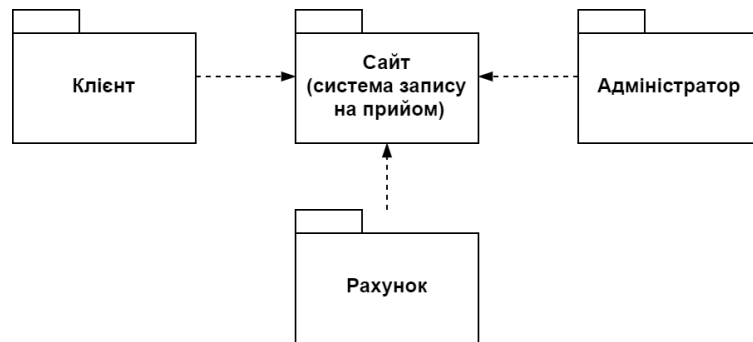


Рисунок 3.2 – Діаграма пакетів програмного забезпечення медичного кабінету у нотації UML

Діаграма компонентів програмного забезпечення медичного кабінету зображена на рис. 3.3.

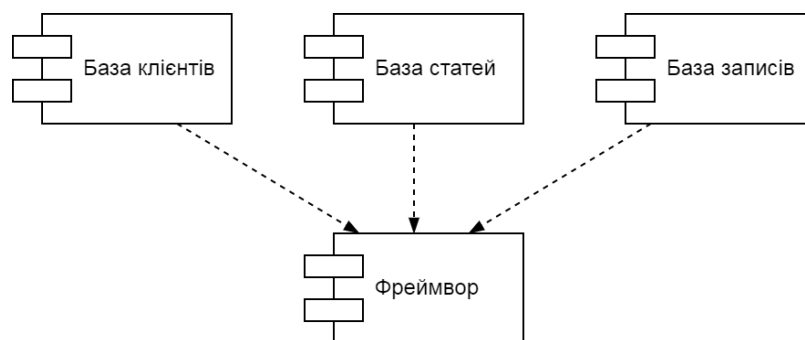


Рисунок 3.3 – Діаграма компонентів програмного забезпечення медичного кабінету у нотації UML

3.2 Функціонування програмного забезпечення медичного кабінету

Функціонування програмного забезпечення медичного кабінету подамо у вигляді моделей у нотації UML основі діаграм поведінки.

Діаграма прецедентів програмного забезпечення медичного кабінету зображена на рис. 3.4.

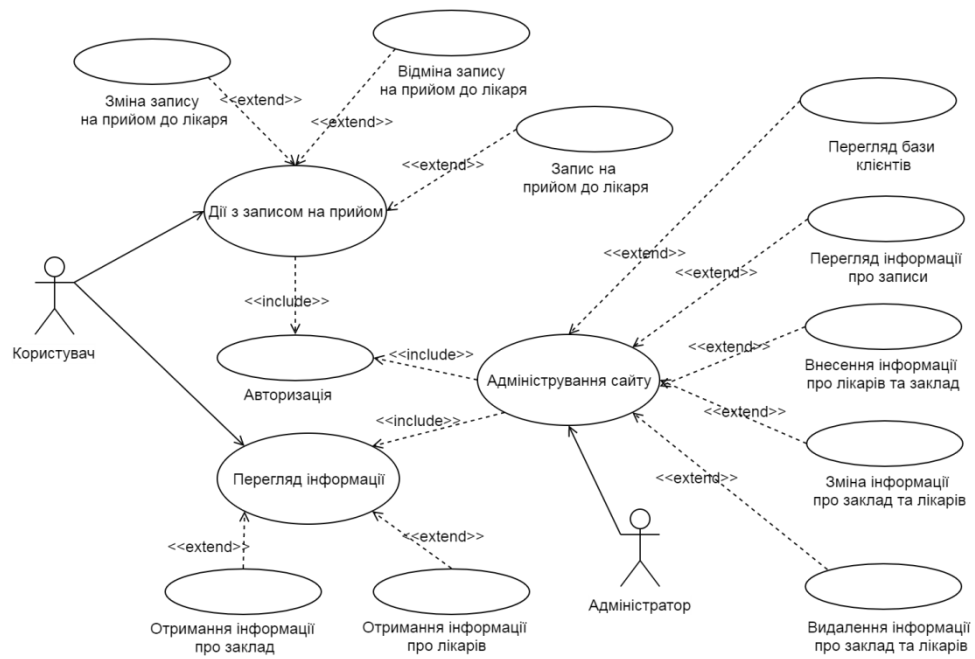


Рисунок 3.4 – Діаграма прецедентів програмного забезпечення медичного кабінету у нотації UML

Діаграма станів програмного забезпечення медичного кабінету подана на рис. 3.5.

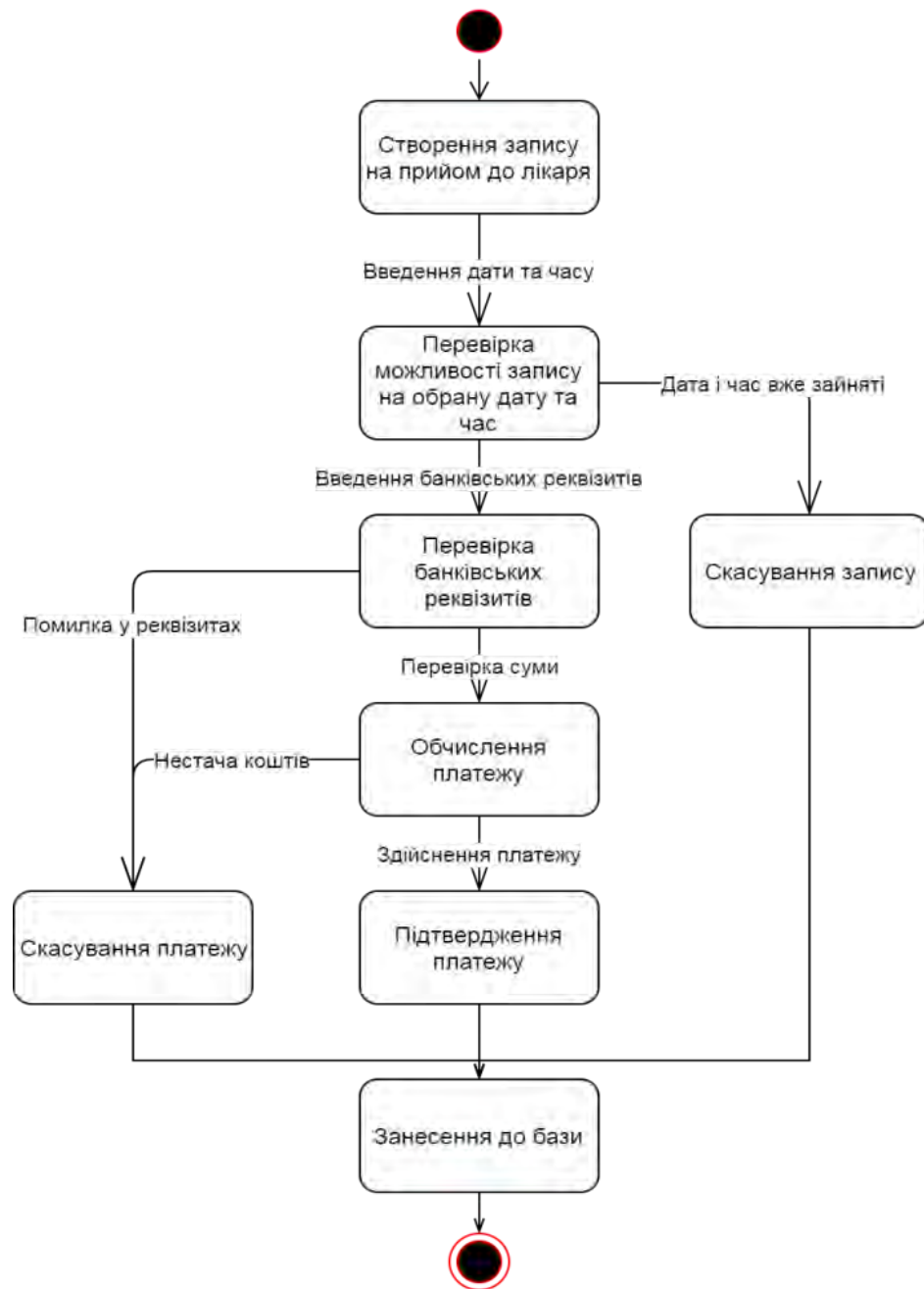


Рисунок 3.5 – Діаграма станів програмного забезпечення медичного кабінету у нотації UML

Діаграма діяльності програмного забезпечення медичного кабінету зображена на рис. 3.6.

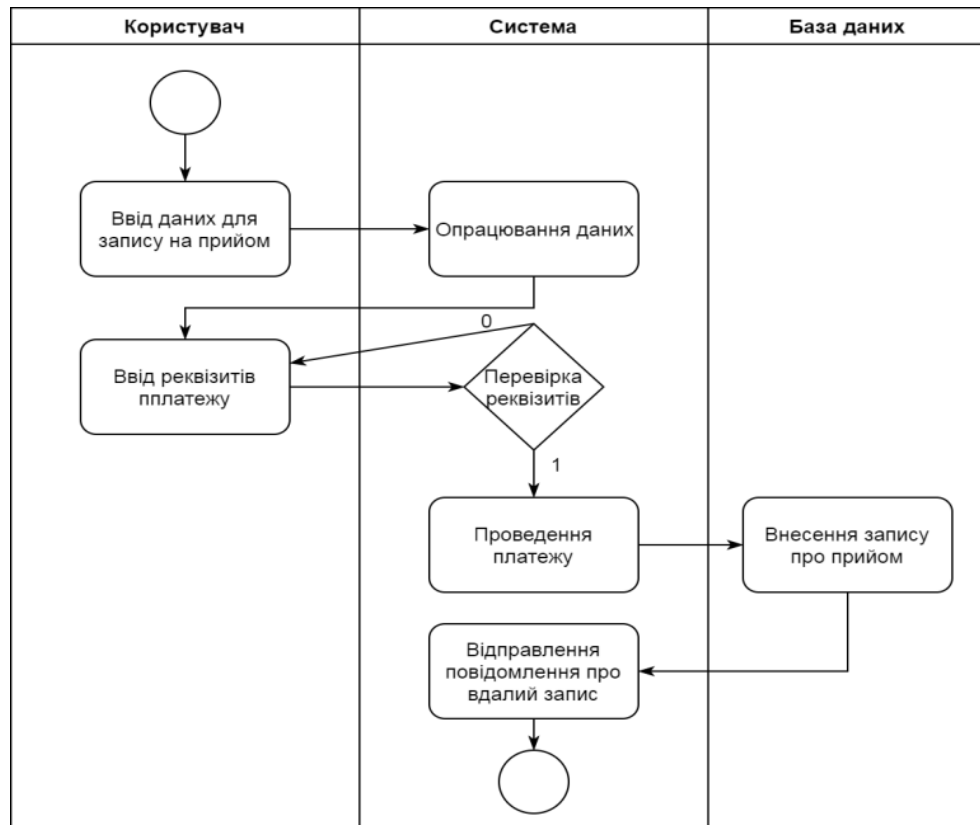


Рисунок 3.6 – Діаграма діяльності програмного забезпечення медичного кабінету у нотації UML

Також в розроблено діаграми кооперації та послідовності у нотації UML.

Діаграма послідовності програмного забезпечення медичного кабінету у нотації UML зображена на рис. 3.7.

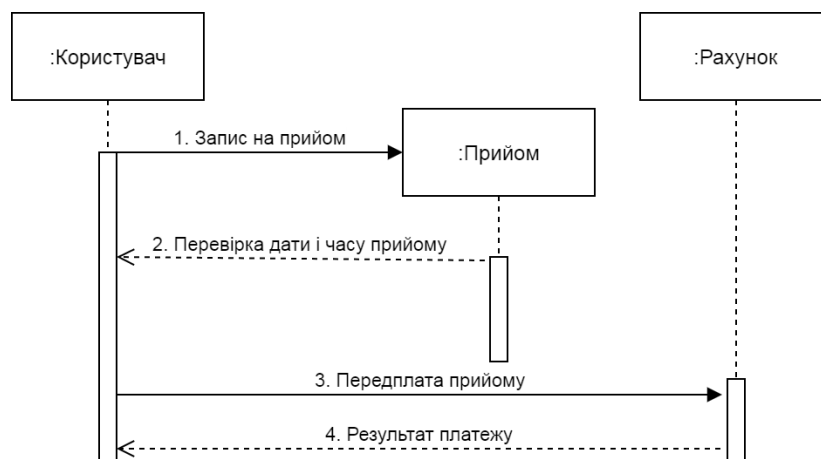


Рисунок 3.7 – Діаграма послідовностей програмного забезпечення медичного кабінету у нотації UML

Діаграма кооперацій програмного забезпечення медичного кабінету у нотації UML зображена на рис. 3.8.

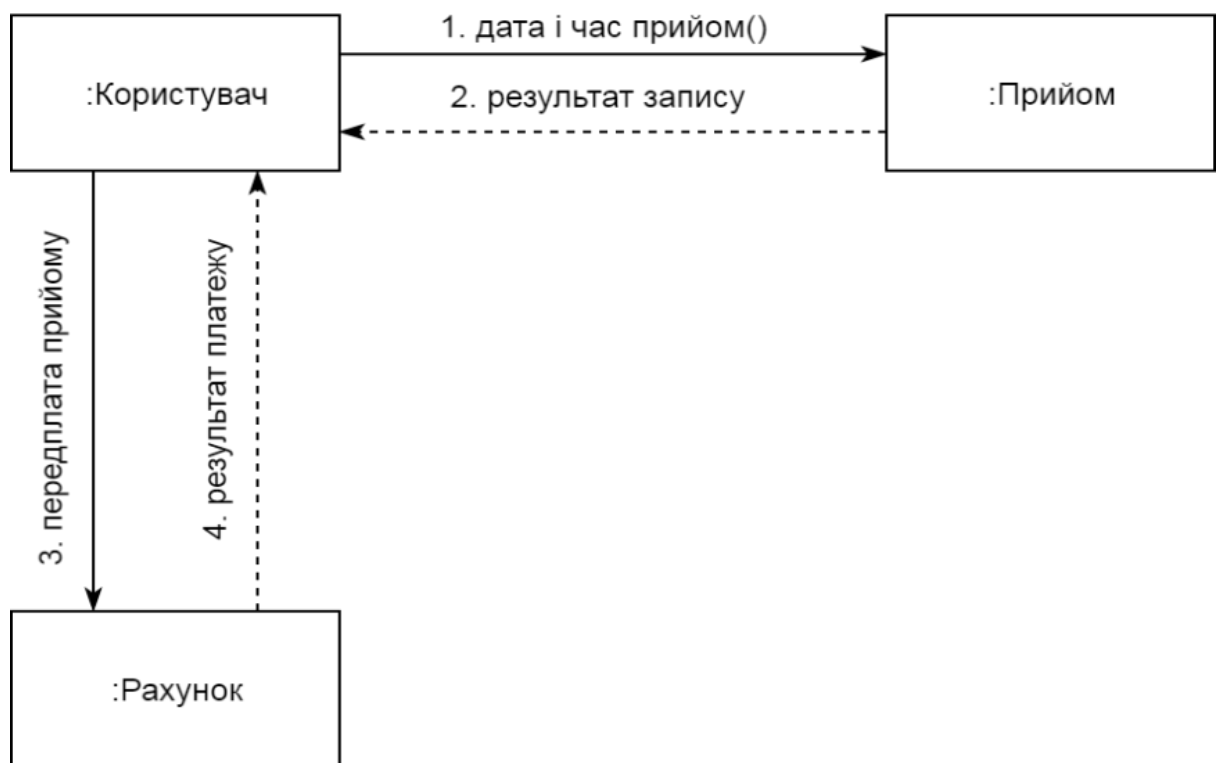


Рисунок 3.8 – Діаграма кооперацій програмного забезпечення медичного кабінету у нотації UML

3.3 Проєктування інтерфейсу взаємодії користувача з програмним забезпеченням медичного кабінету

Проєктування інтерфейсу взаємодії користувача з програмним забезпеченням медичного кабінету виконано з урахуванням вимог щодо інтерфейсу з використанням кольорових наборів, зручних для користувача.

На рис. 3.9 зображено макет головної сторінки користувача програмного забезпечення у вигляді вебсайту медичного кабінету.



Рисунок 3.9 – Проєктування головної сторінки користувача

На рис. 3.10 зображений макет сторінки авторизації користувача.



Рисунок 3.10 – Проєктування інтерфейсу форми авторизації користувача

На рис. 3.11 зображений макет сторінки, яка містить інформацію про надавані послуги.



Рисунок 3.11 – Проєктування інтерфейсу сторінки з інформацією про надавані послуги

На рис. 3.12 зображений макет сторінки запису на прийом до лікаря.

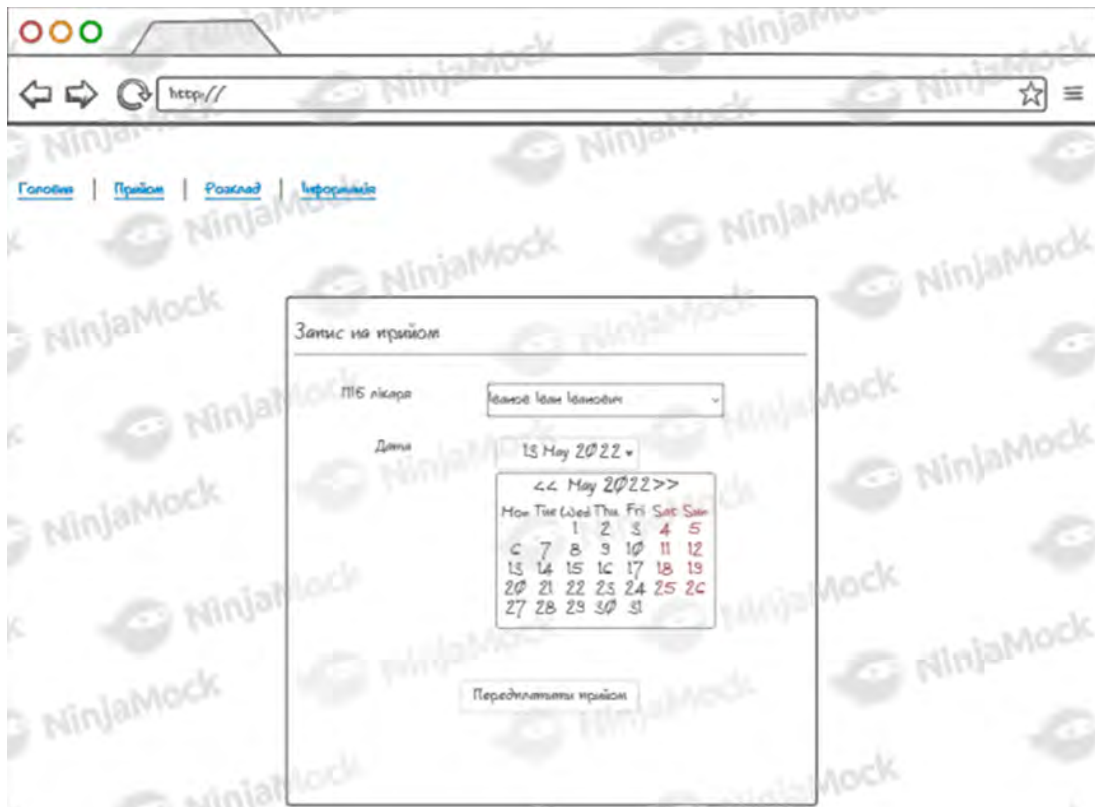


Рисунок 3.12 – Проектування інтерфейсу сторінки запису на прийом до лікаря.

3.4 Розширення функціональності програмного забезпечення медичного кабінету за допомогою плагінів

З метою додавання можливості відображення розташування медичного кабінету за допомогою Google-мап було обрано плагін WP Google Maps.

Для того, щоб додати модуль для розширення функціональності програмного забезпечення медичного кабінету у CMS WordPress було здійснено вхід до адміністративної частини (рис. 3.13), після чого на вкладці Плагіни (рис. 3.14) було завантажено плагін WP Google Maps.

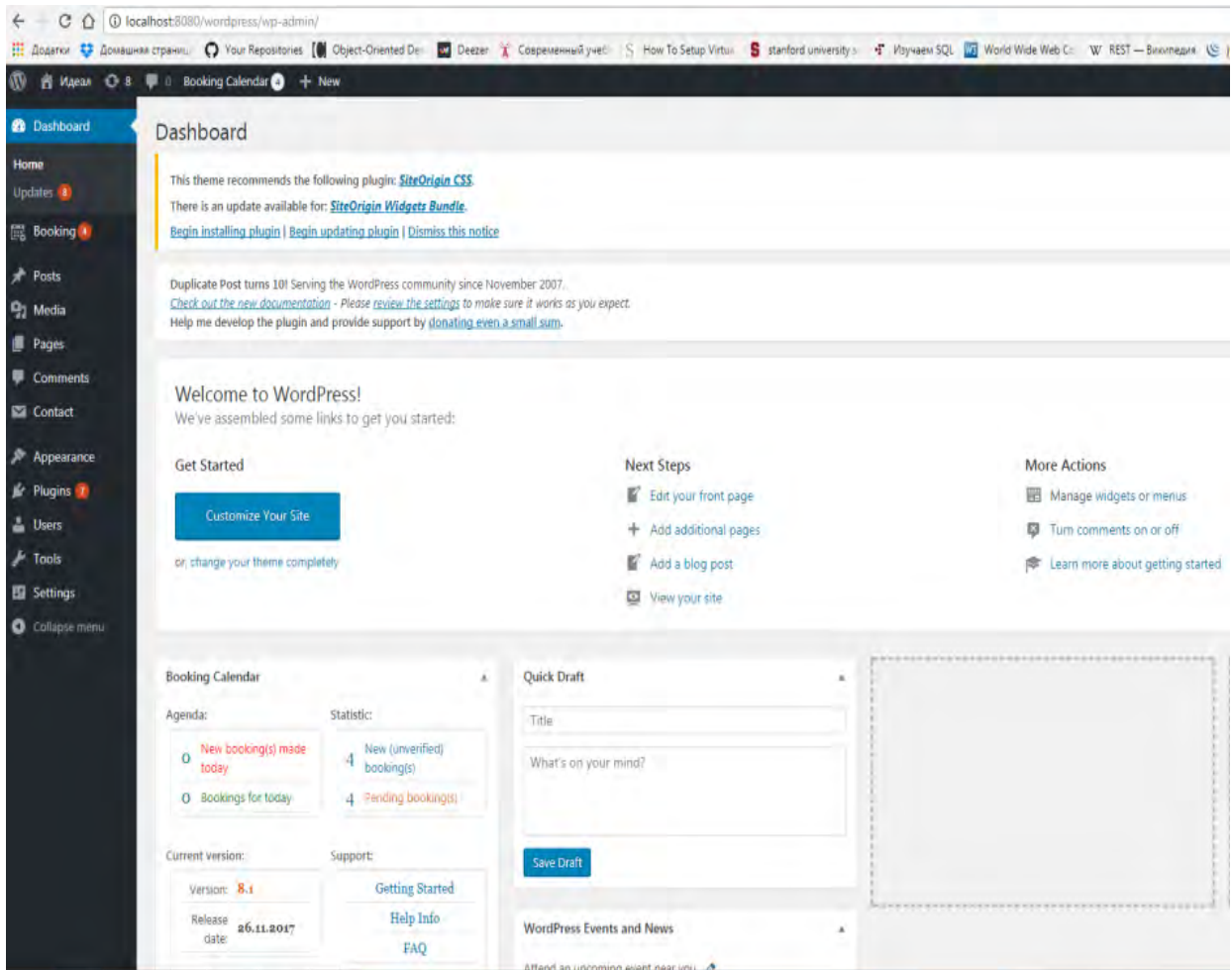


Рисунок 3.13 – Адміністративна сторінка CMS WordPress

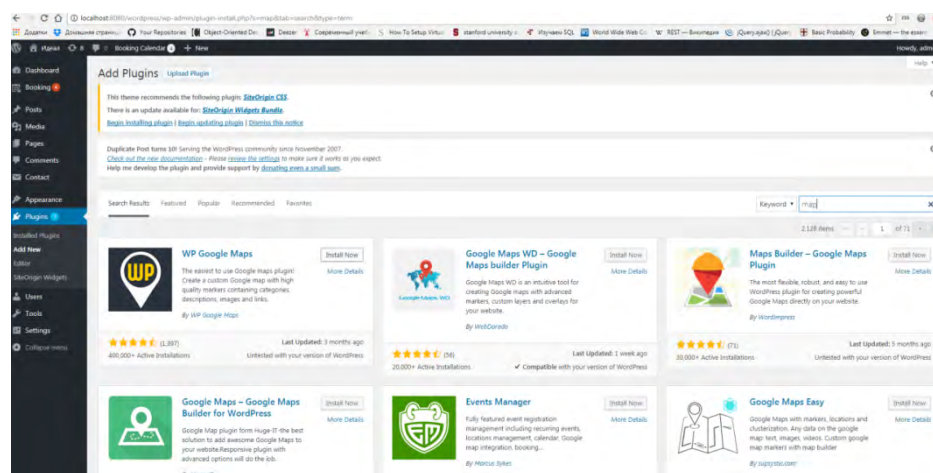


Рисунок 3.14 – Перелік плагінів WordPress

Після цього було обрано та додано до програмного проєкту плагін WP Google Maps, що додає віджет, який відображає мапу. Після встановлення

плагіну WP Google Maps було налаштовано відповідний віджет та створено нову мапу (рис. 3.15).

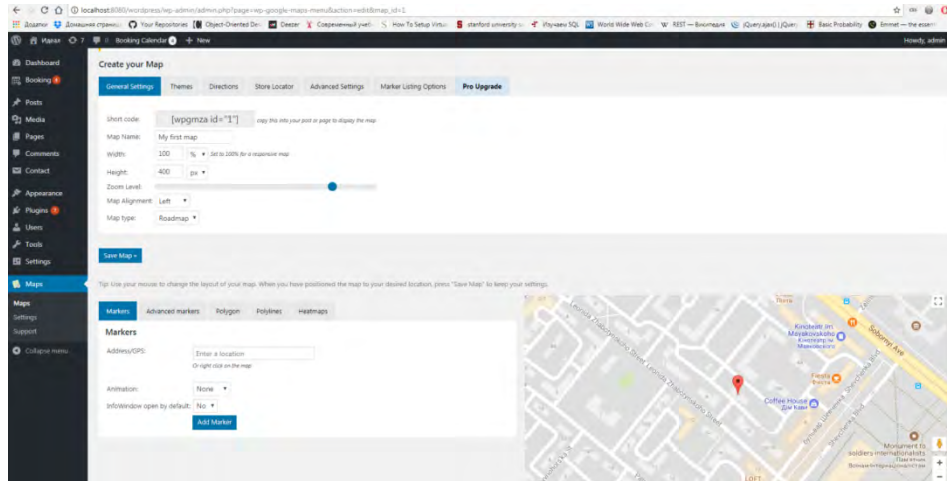


Рисунок 3.15 – Налаштування плагіна WP Google Maps

Далі налаштування віджету було додано відповідну мапу до сторінки вебсайту медичного кабінету. Результат додавання віджету зображено на рис. 3.16.

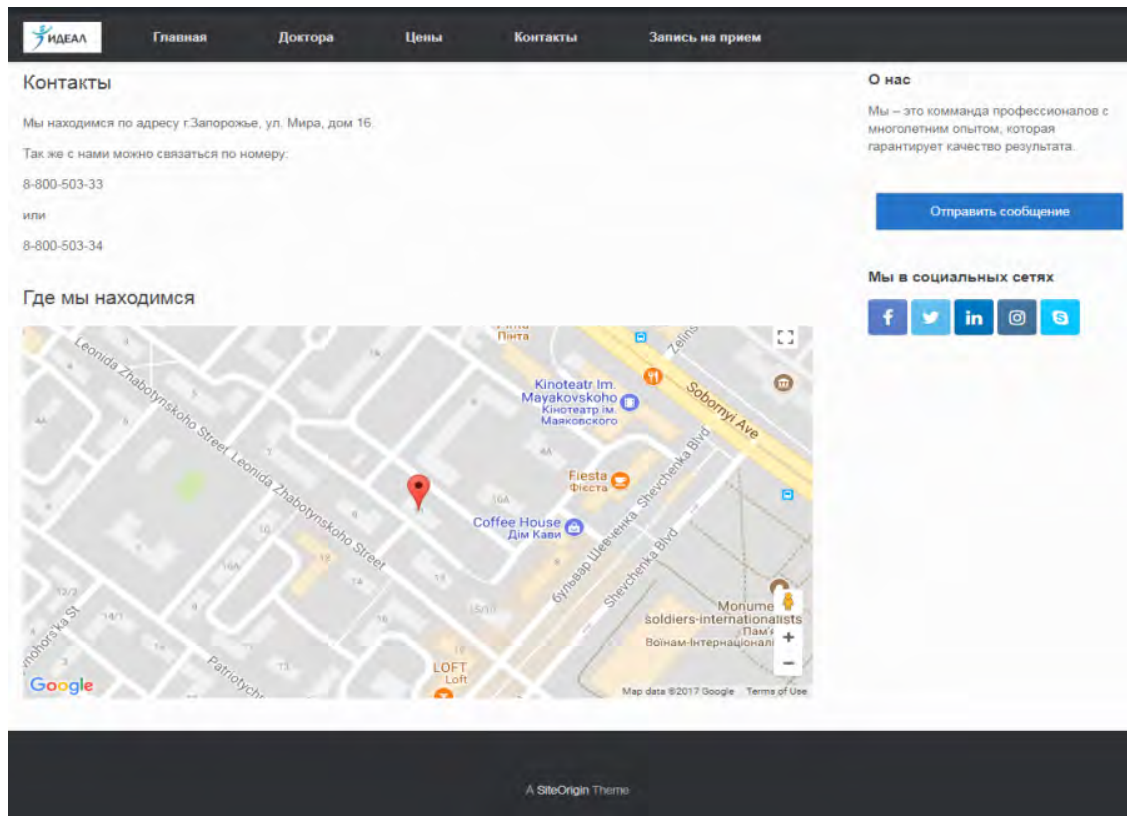


Рисунок 3.16 – Результат додавання віджету та відображення мапи на вебсайті медичного кабінету

3.5 Висновки за розділом 3

Розроблене програмне забезпечення медичного кабінету. Запропоновано структуру програмного забезпечення медичного кабінету, яку подано за допомогою структурних діаграм в нотації UML, зокрема розроблено діаграму класів, діаграму пакетів та діаграму компонентів.

Описано функціонування програмного забезпечення медичного кабінету за допомогою моделей у нотації UML основі діаграм поведінки, діаграм кооперації та діаграм послідовності.

Виконано проектування інтерфейсу взаємодії користувача з програмним забезпеченням медичного кабінету з урахуванням вимог щодо інтерфейсу з використанням кольорових наборів, зручних для користувача.

Виконано розширення функціональності програмного забезпечення медичного кабінету за допомогою плагіну WP Google Maps, що дозволило додати можливість відображення розташування медичного кабінету за допомогою Google-мап.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програмне забезпечення медичного кабінету призначено для підтримки процесу управління медичною діяльністю закладу охорони здоров'я. Програмне забезпечення медичного кабінету написано на мові PHP. Як систему управління вмістом для створення програмного забезпечення медичного кабінету обрано Wordpress.

Програмне забезпечення роботи медичного кабінету забезпечує виконання таких функцій:

- наявність зручної адміністративної панелі та сховища даних;
- перегляд записів адміністратора;
- можливість наповнення сайту інформацією;
- зручний та зрозумілий інтерфейс користувача;
- проста процедура запису на прийом для користувачів.

Для експлуатації програмного забезпечення медичного кабінету необхідно таке апаратне забезпечення:

- вільне місце на жорсткому диску 200 Мб або більше;
- процесор із тактовою частотою 2,5 GHz та більше;
- оперативна пам'ять 4 Гб та більше;
- наявність пристроїв введення та виведення інформації.

Функціональні характеристики розробленого програмного забезпечення медичного кабінету наведено у технічному завданні (додаток А). Опис програмного забезпечення медичного кабінету та текст програмного забезпечення медичного кабінету наведені у додатках Б та В.

4.2 Характеристики програми

Режим роботи програмного забезпечення медичного кабінету є цілодобовим та безперервним. Програмне забезпечення медичного кабінету використовує прийнятну кількість апаратних ресурсів комп'ютера.

Програмне забезпечення медичного кабінету не потребує особливих умов експлуатації, окрім тих, які зазначені у п.4.1. Інтерфейс програмного забезпечення медичного кабінету не потребує особливої кваліфікації та попереднього навчання користувача, окрім базових навичок роботи з комп'ютером.

4.3 Встановлення серверу та системи керування вмістом для функціонування програмного забезпечення медичного кабінету

Для функціонування програмного забезпечення медичного кабінету необхідно встановити сервер, в якості локального серверу можна обрати стек рішень ХАМРР, що є безкоштовною кросплатформовою збіркою веб-сервера з відкритим сирцевим кодом, що містить HTTP-сервер Apache, системи керування базами даних MariaDB, MySQL й інтерпретатори скриптів для мов програмування PHP та Perl, а також додаткові бібліотеки, що дозволяють запустити повноцінний веб-сервер.

Для початку роботи на сторінці завантажень треба обрати конфігурацію операційної системи та завантажити файл встановлення (рис. 4.1).

Після стандартної процедури встановлення програми, запускається ХАМРР Control Panel (рис. 4.2) – вікно, де користувач має швидкий доступ до основних функцій керування сервером та його конфігурацій.

[Apache Friends](#)
[Download](#)
[Add-ons](#)
[Hosting](#)
[Community](#)
[About](#)

EN

Download

XAMPP is an easy to install Apache distribution containing MariaDB, PHP, and Perl. Just download and start the installer. It's that easy.

Documentation/FAQs

There is no real manual or handbook for XAMPP. We wrote the documentation in the form of FAQs. Have a burning question that's not answered here? Try the Forums or Stack Overflow.

- Linux FAQs
- Windows FAQs
- OS X FAQs
- OS X XAMPP-VM FAQs

Add-ons and Themes

Bitnami provides a free all-in-one tool

XAMPP for Windows 5.6.32, 7.0.25 & 7.1.11

Version	Checksum	Size
5.6.32 / PHP 5.6.32	What's Included? md5 sha1 Download (32 bit)	109 Mb
7.0.25 / PHP 7.0.25	What's Included? md5 sha1 Download (32 bit)	120 Mb
7.1.11 / PHP 7.1.11	What's Included? md5 sha1 Download (32 bit)	120 Mb

[Requirements](#)
[Add-ons](#)
[More Downloads »](#)

Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms [here](#).

Рисунок 4.1 – Сторінка завантаження XAMPP

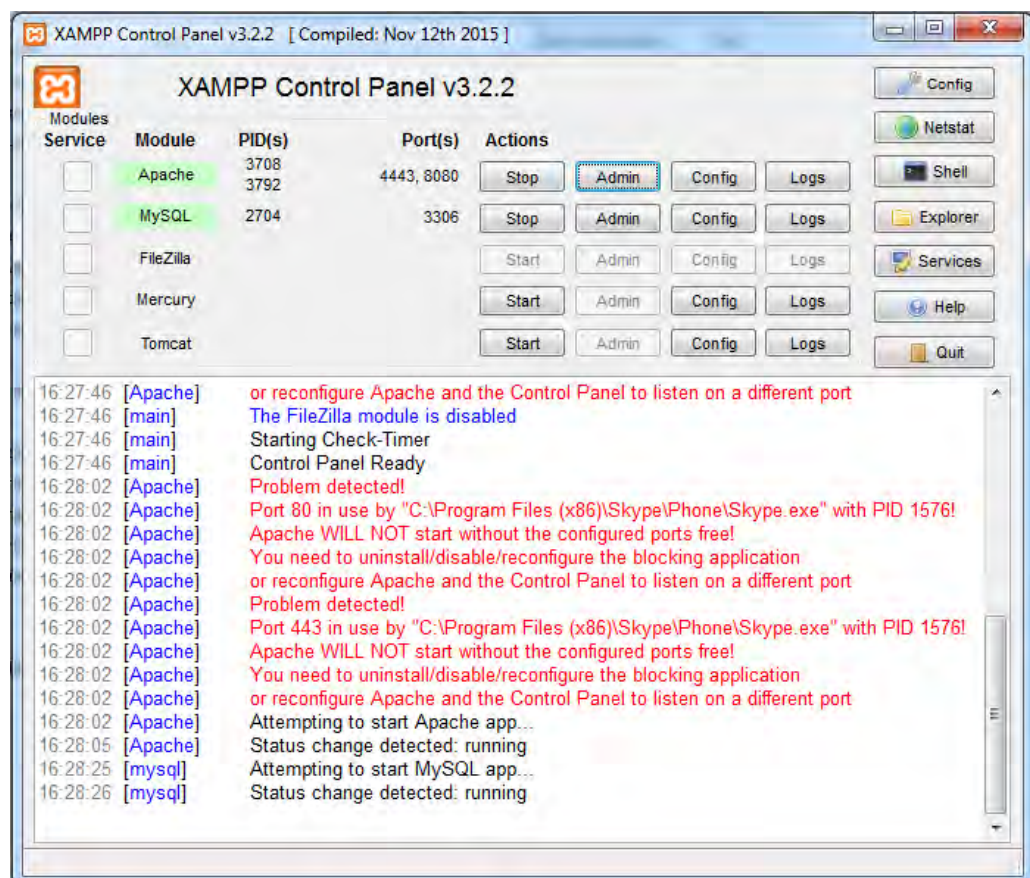


Рисунок 4.2 – XAMPP Control Panel

Після запуску сервера та переходу на `localhost:8080/dashboard/` відкривається сторінка (рис. 4.3) на якій відображається загальна інформація про XAMPP, та містяться посилання на PHPInfo, phpMyAdmin та найчастіші питання.

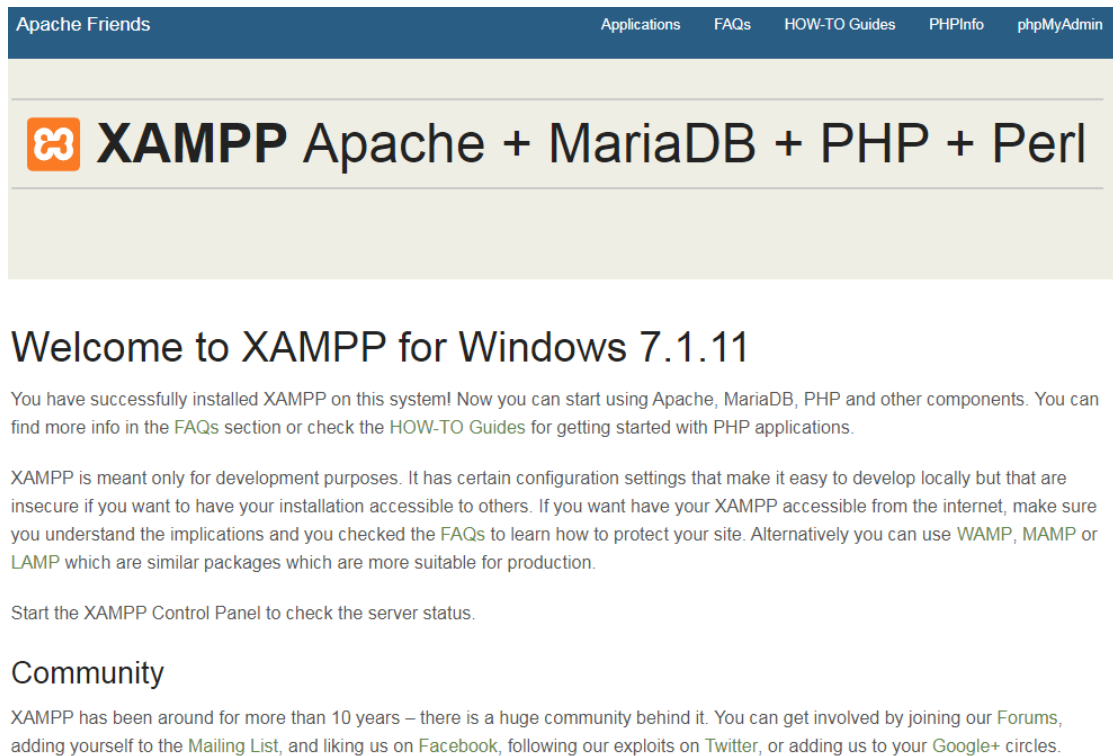


Рисунок 4.3 – Сторінка XAMPP

Для встановлення CMS WordPress було використано модуль Bitname WordPress (рис. 4.4), який був завантажений за посиланням <https://bitnami.com/stack/xampp#wordpress>. Даний модуль дозволяє встановити обрану CMS на XAMPP так само, як і будь-яку програму на комп'ютер без додаткових налаштувань серверу.

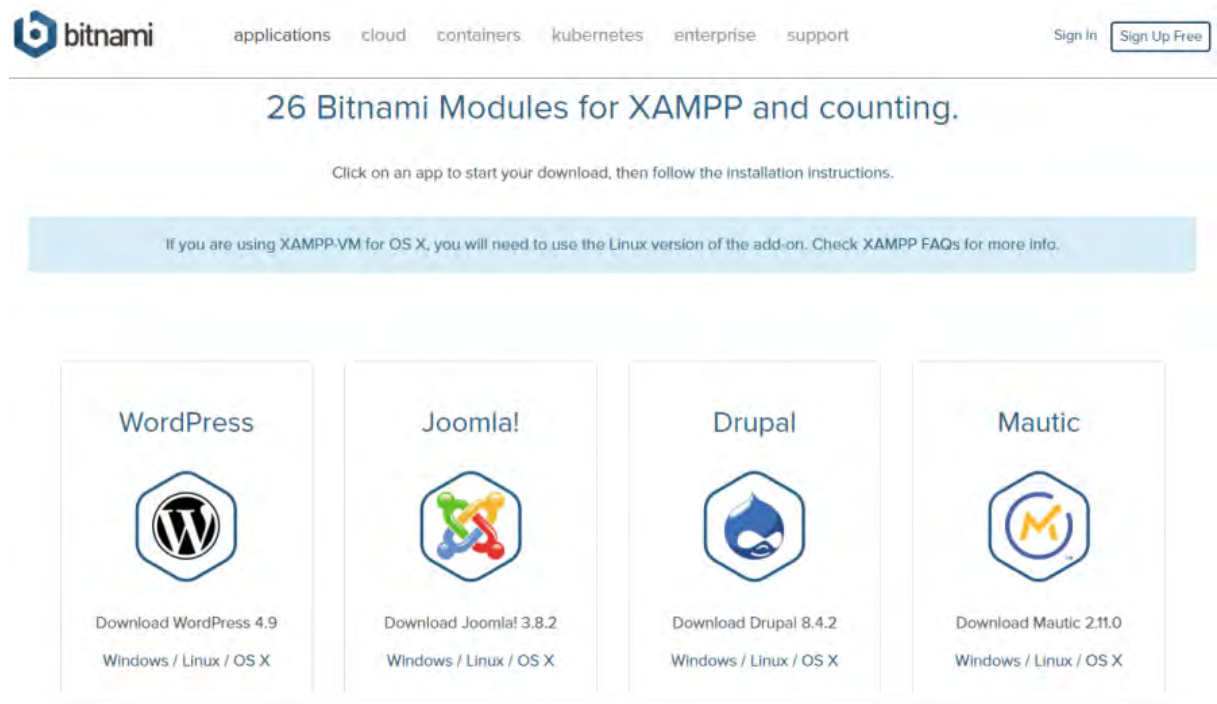


Рисунок 4.4 – Сторінка bitnami.com

Після встановлення CMS можна відвідати <http://localhost:8080/wordpress/>, де знаходиться стандартний шаблон сайту, який WordPress створює автоматично.

4.4 Тестування та результати роботи програми медичного кабінету

Тестування роботи програми медичного кабінету виконується у два етапи, на першому з яких перевіряється, чи вебзастосунок є сумісним з відомими веббраузерами, а на другому етапі виконується тестування за допомогою програми тестування Xenu's Link Sleuth.

Після запуску програми медичного кабінету у вигляді вебсайту з'являється головна сторінка (рис. 4.5).

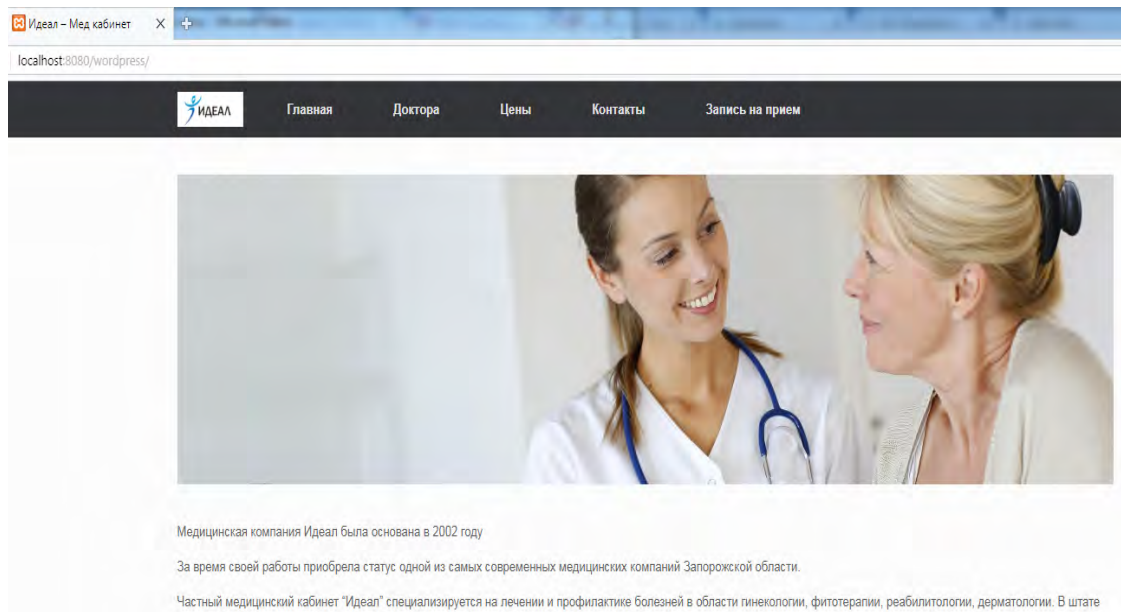


Рисунок 4.5 – Головна сторінка програми медичного кабінету

Користувач може переглянути інформацію про лікарів, ціни за послуги, контакти клініки, а також записатися на прийом. Для запису на прийом необхідно обрати відповідний пункт головного меню (рис. 4.6).

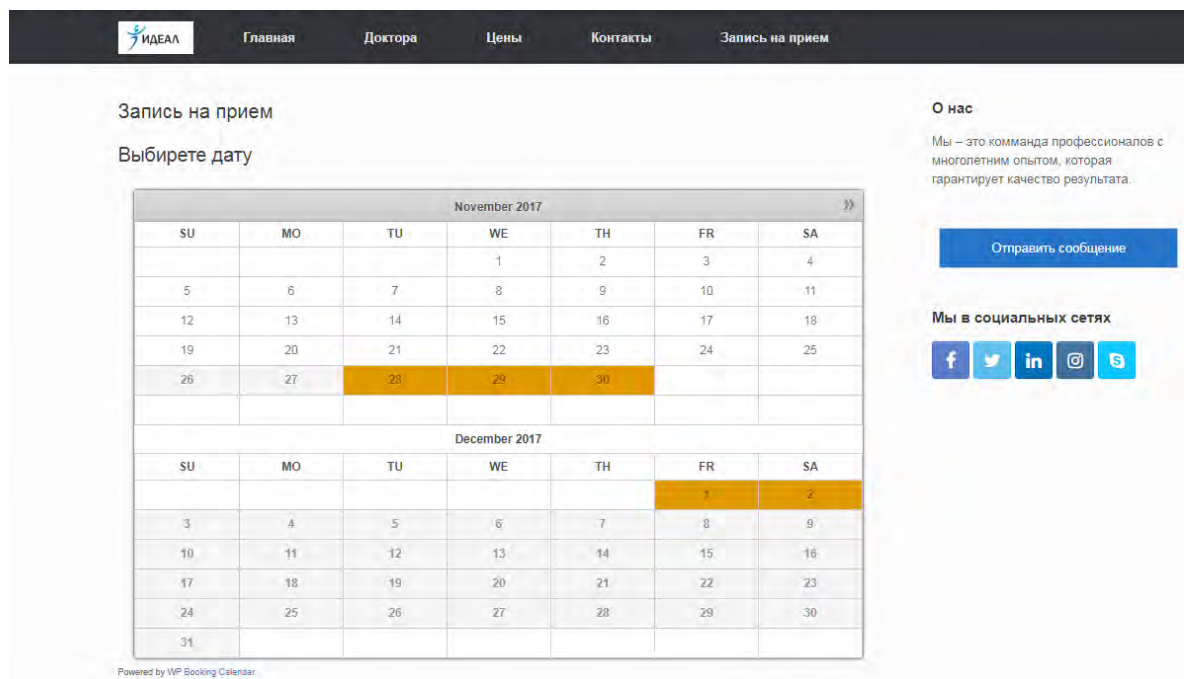


Рисунок 4.6 – Відображення сторінки запису на прийом через браузер GoogleChrome

Сторінку контактів клініки наведено на рис. 4.7.

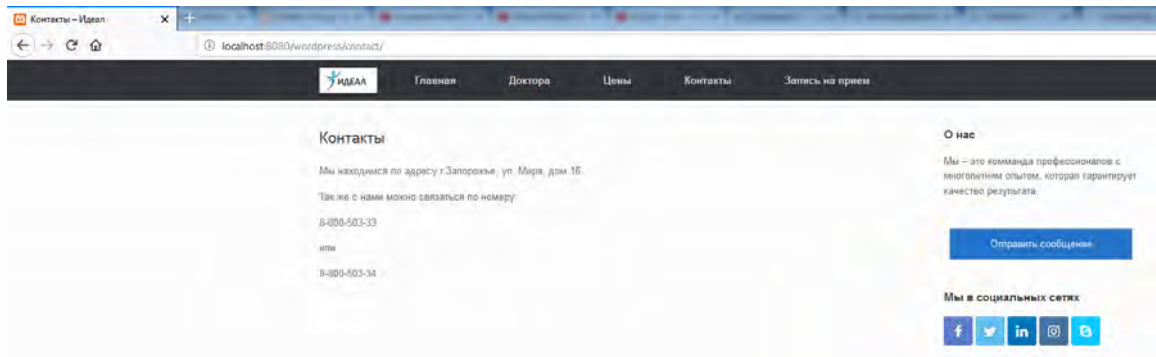


Рисунок 4.7 – Відображення додатку через браузер MozillaFirefox

На рис. 4.8 продемонстровано коректне відображення сторінки з цінами на прийом через браузер Internet Explorer.

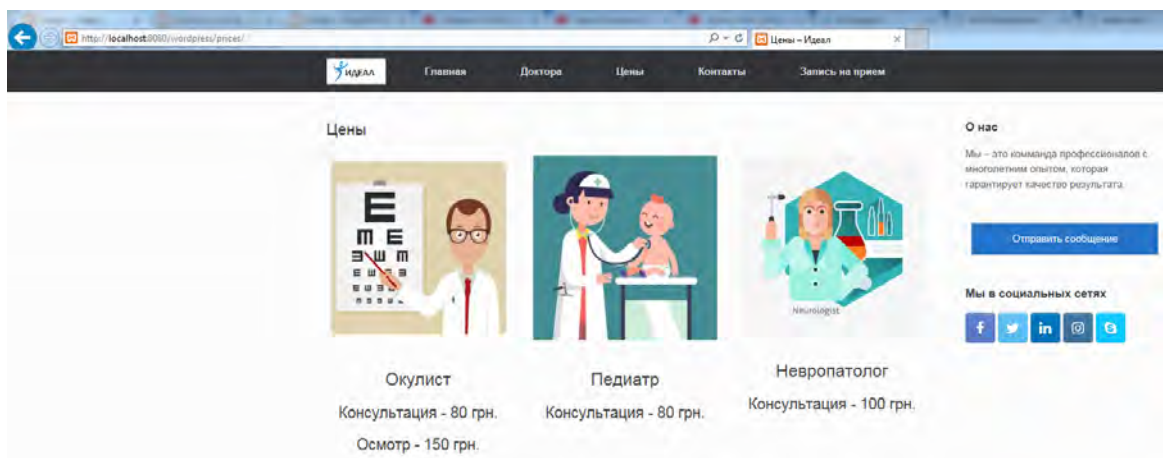


Рисунок 4.8 – Відображення сторінки з цінами на прийом через браузер Internet Explorer

Програма Xenu's Link Sleuth тестує додаток на наявність не дійсних посилань. Результати тестування зображені на рис. 4.9–4.11.

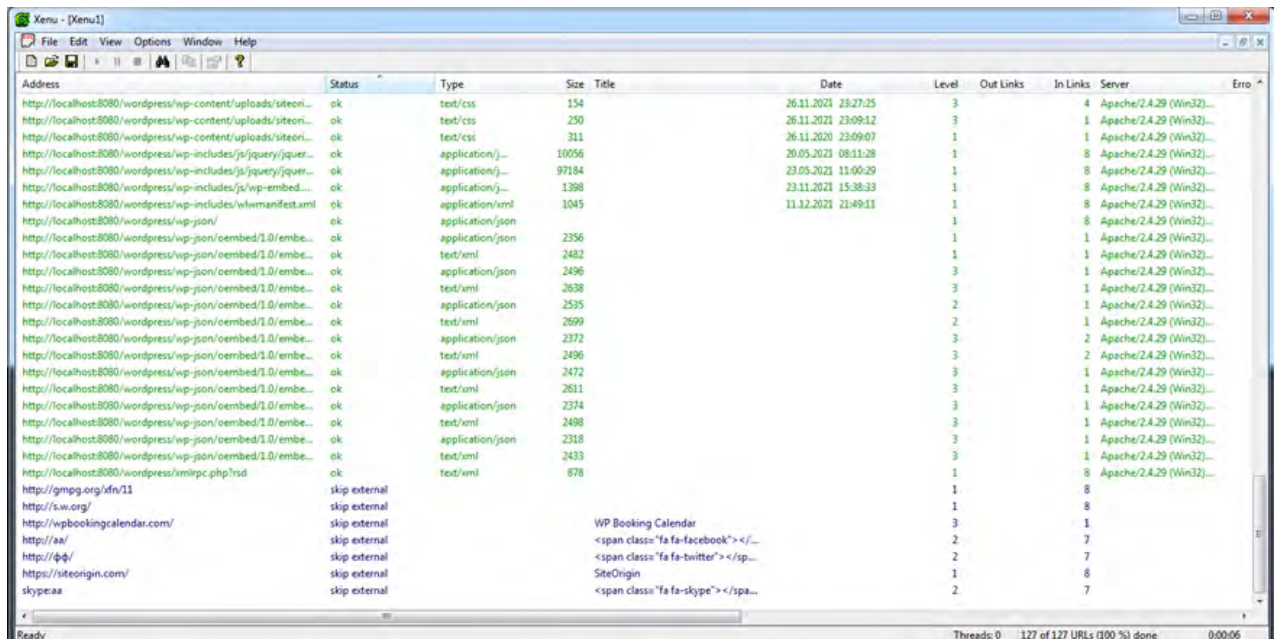


Рисунок 4.9 – Виконання тестування

Broken page-local links (also named 'anchors', 'fragment identifiers'):

0 bad local link(s) reported

[Return to Top](#)**Orphan files:**

Note: Links that aren't spidered (e.g. webforms and dynamically generated links) will appear as orphans in this list.

Orphan search aborted by user!

0 orphan(s) reported (0.0 KB)

[Return to Top](#)**Statistics for managers****Correct internal URLs, by MIME type:**

MIME type	count	% count	Σ size	Σ size (KB)	% size	min size	max size	Ø size	Ø size (KB)	Ø time
text/html	20 URLs	16.81%	197439 Bytes	(192 KB)	6.74%	0 Bytes	28397 Bytes	9871 Bytes	(9 KB)	1.000
application/rss+xml	2 URLs	1.68%	3275 Bytes	(3 KB)	0.11%	1622 Bytes	1653 Bytes	1637 Bytes	(1 KB)	
text/css	17 URLs	14.29%	461593 Bytes	(450 KB)	15.75%	154 Bytes	182631 Bytes	27152 Bytes	(26 KB)	
application/javascript	18 URLs	15.13%	444684 Bytes	(434 KB)	15.17%	1398 Bytes	98302 Bytes	24704 Bytes	(24 KB)	
application/json	8 URLs	6.72%	16923 Bytes	(16 KB)	0.58%	0 Bytes	2535 Bytes	2115 Bytes	(2 KB)	
text/xml	8 URLs	6.72%	18735 Bytes	(18 KB)	0.64%	878 Bytes	2699 Bytes	2341 Bytes	(2 KB)	
application/xml	1 URLs	0.84%	1045 Bytes	(1 KB)	0.04%	1045 Bytes	1045 Bytes	1045 Bytes	(1 KB)	
image/jpeg	9 URLs	7.56%	394300 Bytes	(385 KB)	13.45%	11646 Bytes	108555 Bytes	43811 Bytes	(42 KB)	
image/gif	2 URLs	1.68%	2481 Bytes	(2 KB)	0.08%	847 Bytes	1634 Bytes	1240 Bytes	(1 KB)	
application/vnd.ms-fontobject	8 URLs	6.72%	379258 Bytes	(370 KB)	12.94%	1468 Bytes	165742 Bytes	47407 Bytes	(46 KB)	
font/woff	4 URLs	3.36%	124748 Bytes	(121 KB)	4.26%	1380 Bytes	98024 Bytes	31187 Bytes	(30 KB)	
font/ttf	4 URLs	3.36%	214384 Bytes	(209 KB)	7.31%	1304 Bytes	165548 Bytes	53596 Bytes	(52 KB)	
image/svg+xml	4 URLs	3.36%	558434 Bytes	(545 KB)	19.05%	1177 Bytes	444379 Bytes	139608 Bytes	(136 KB)	

Рисунок 4.10 – Загальна статистика результатів тестування

All pages, by result type:

ok	119 URLs	93.70%
skip external	7 URLs	5.51%
method is not allowed	1 URLs	0.79%
Total	127 URLs	100.00%

Рисунок 4.11 – Підсумковий результат тестування

За результатами тестування за допомогою програми Xenu's Link Sleuth виявлено, що всі доступні користувачу посилання працюють коректно. Вісім посилань було пропущено, оскільки вони вказують на сторонні ресурси та відносяться до системи управління вмістом WordPress, а не до користувацького інтерфейсу програмного забезпечення медичного кабінету.

Таким чином, результати тестування програмного забезпечення медичного кабінету показали, програма функціонує коректно, відповідає технічному завданню та може використовуватися за призначенням.

4.5 Висновки за розділом 4

Описано програмне забезпечення медичного кабінету. Виконано тестування розробленого програмного забезпечення медичного кабінету. Результати тестування програмного забезпечення медичного кабінету показали, програма функціонує коректно, відповідає технічному завданню та може використовуватися за призначенням.

Висновки

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процес розробки програмного забезпечення для медичної діяльності.

Проаналізовано загальні поняття про програмне забезпечення для підтримки медичної діяльності. Визначено, що важливим типом програмного забезпечення для підтримки медичної діяльності є системи управління лікарнями, що надають як інформацію, так і можливості управління трьома групами зацікавлених сторін, присутнім у лікарні: адміністрації, лікарям та уповноваженим медичним працівникам та пацієнтам.

Досліджено програмні засоби для підтримки медичної діяльності, зокрема проаналізовано програми Teledoc, 10to8, QuantiaMD, Amplitude, EasyClinic, eConsult.

Виявлено, що не всі відомі програмні засоби підтримують україномовну локалізацію, крім того, більшість програм не є безкоштовними, деякі є досить складними у використанні, оскільки підтримують дуже багато функцій, які часто є досить специфічними та не є корисними для більшості медичних закладів, що суттєво ускладнює використання такого програмного забезпечення на практиці. Отже доцільною є розробка програмного забезпечення для медичного кабінету, що усуває зазначені недоліки. Визначено функціональні вимоги до розроблюваного програмного забезпечення медичного кабінету.

Для розробки програмного забезпечення для медичного кабінету обрано мову програмування PHP, яка є мовою з відкритим вихідним кодом, доступною та безкоштовною для всіх поширених операційних систем, є досить простою для використання, підходить для створення динамічного вебконтенту та підтримує інтеграцію з багатьма базами даних.

Як систему управління вмістом для створення програмного забезпечення медичного кабінету обрано Wordpress, яка характеризується широким використанням, високим рівнем налаштування, простою конфігурацією, дозволяє редагувати вебсторінки візуально в режимі онлайн, має велику кількість плагінів, що дозволяє підвищити функціональність розроблюваного програмного забезпечення.

Розроблене програмне забезпечення медичного кабінету. Запропоновано структуру програмного забезпечення медичного кабінету, яку подано за допомогою структурних діаграм в нотації UML, зокрема розроблено діаграму класів, діаграму пакетів та діаграму компонентів.

Описано функціонування програмного забезпечення медичного кабінету за допомогою моделей у нотації UML основі діаграм поведінки, діаграм кооперації та діаграм послідовності.

Виконано проектування інтерфейсу взаємодії користувача з програмним забезпеченням медичного кабінету з урахуванням вимог щодо інтерфейсу з використанням кольорових наборів, зручних для користувача.

Виконано розширення функціональності програмного забезпечення медичного кабінету за допомогою плагіну WP Google Maps, що дозволило додати можливість відображення розташування медичного кабінету за допомогою Google-мап.

Описано програмне забезпечення медичного кабінету. Виконано тестування розробленого програмного забезпечення медичного кабінету. Результати тестування програмного забезпечення медичного кабінету показали, програма функціонує коректно, відповідає технічному завданню та може використовуватися за призначенням.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. 10 of the Most Popular Types of Healthcare Software [Electronic resource]. – Access mode : <https://computools.com/types-of-healthcare-software/>.
2. 11 Most Popular Types of Healthcare Software [Electronic resource]. – Access mode : <https://www.elpassion.com/blog/most-popular-types-of-healthcare-software>.
3. The 13 Best Online Medical Software Tools for Doctors [Electronic resource]. – Access mode : <https://10to8.com/blog/the-best-online-medical-software/>.
4. Hospital Management Software [Electronic resource]. – Access mode : <https://www.softwareadvice.com/hospital-management/>.
5. Hospital Management Software [Electronic resource]. – Access mode : <https://www.capterra.com/hospital-management-software/>.
6. A detailed view of Hospital Management System [Electronic resource]. – Access mode : <https://mocdoc.in/blog/a-detailed-view-of-hospital-management-system-hms>.
7. The Best Hospital Management Software of 2022 [Electronic resource]. – Access mode : <https://digital.com/best-hospital-software/>.
8. Hospital Management Systems: Features, Requirements and Benefits [Electronic resource]. – Access mode : <https://light-it.net/blog/how-to-develop-a-quality-management-system-in-a-hospital/>.
9. Teledoc [Electronic resource]. – Access mode : <https://www.teladoc.com>.
10. Medical Scheduling Software [Electronic resource]. – Access mode : <https://10to8.com/healthcare-scheduling-software/>.
11. QuantiaMD [Electronic resource]. – Access mode : <https://www.mobihealthnews.com/tag/quantiamd>.

12. Amplitude [Electronic resource]. – Access mode :
<https://amplitude-clinical.com>.
13. EasyClinic [Electronic resource]. – Access mode :
<http://www.easyclinic.in>.
14. Econsult [Electronic resource]. – Access mode : <https://econsult.net/>.
15. PHP lernen leicht gemacht [Electronic resource]. – Access mode :
<https://www.php-einfach.de/>.
16. PHP [Electronic resource]. – Access mode : <https://www.php.de/>.
17. PHP Tutorial [Electronic resource]. – Access mode :
<https://www.php-einfach.de/php-tutorial/>.
18. Was ist PHP? [Electronic resource]. – Access mode :
<https://www.ionos.de/digitalguide/websites/web-entwicklung/php-vs-python/>.
19. Was ist Wordpress? [Electronic resource]. – Access mode :
<https://kopfundstift.de/was-ist-wordpress/>.
20. Content Management System (CMS) WordPress [Electronic resource].
– Access mode : <https://kopfundstift.de>.
21. CSS Styles [Electronic resource]. - Access mode :
<https://pastebin.com/jKmewtGB>

Додаток А
Технічне завдання

Вступ

Програмне забезпечення медичного кабінету призначено для підтримки процесу управління медичною діяльністю закладу охорони здоров'я.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Програмне забезпечення медичного кабінету», затверджене наказом Національного університету «Запорізька політехніка» № 102 від 27 квітня 2022 р.

A.2 Призначення розробки

Програмний продукт для роботи медичного кабінету призначено для підтримки процесу управління медичною діяльністю закладу охорони здоров'я.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Програмне забезпечення для медичного кабінету повинно забезпечити виконання таких функцій:

- наявність зручної адміністративної панелі та сховища даних;
- перегляд записів адміністратора;
- можливість наповнення сайту інформацією;
- зручний та зрозумілий інтерфейс користувача;
- проста процедура запису на прийом для користувачів.

A.3.2 Вимоги до інтерфейсу програми

Інтерфейс програми медичного кабінету повинен бути зручним для використання на практиці у закладах охорони здоров'я для підтримки медичної діяльності.

A.3.3 Вимоги до надійності

Програмне забезпечення медичного кабінету повинно забезпечити оброблення критичних ситуацій.

A.3.4 Умови експлуатації

Програмне забезпечення медичного кабінету не потребує особливих умов експлуатації.

A.3.5 Вимоги до складу та параметрів технічний засобів

Для експлуатації програмного забезпечення медичного кабінету необхідно таке апаратне забезпечення:

- вільне місце на жорсткому диску 200 Мб або більше;
- процесор із тактовою частотою 2,5 GHz та більше;
- оперативна пам'ять 4 ГБ та більше;
- наявність пристроїв введення та виведення інформації.

A.3.6 Вимоги до маркування і пакування

Програма для медичного кабінету може бути записана на різні інформаційні носії. На пакуванні повинна бути назва програми – «Програмне забезпечення для медичного кабінету».

A.4 Вхідні дані до роботи

Вхідними даними до програмного забезпечення медичного кабінету є інформація про лікарів та розклад їх роботи.

Вихідними даними програмного забезпечення медичного кабінету є повідомлення про успішний запис на прийом до лікаря.

Додаток Б
Опис програми

Б.1 Загальні відомості

Програмне забезпечення для медичного кабінету розроблено за допомогою мови програмування PHP та системи управління вмістом Wordpress.

Б.2 Функціональне призначення

Програмне забезпечення медичного кабінету призначено для підтримки процесу управління медичною діяльністю закладу охорони здоров'я.

Програма медичного кабінету забезпечує виконання таких функцій:

- наявність зручної адміністративної панелі та сховища даних;
- перегляд записів адміністратора;
- можливість наповнення сайту інформацією;
- зручний та зрозумілий інтерфейс користувача;
- проста процедура запису на прийом для користувачів.

Б.3 Використані технічні засоби

Для експлуатації програмного забезпечення медичного кабінету необхідно таке апаратне забезпечення:

- вільне місце на жорсткому диску 200 Мб або більше;
- процесор із тактовою частотою 2,5 GHz та більше;
- оперативна пам'ять 4 ГБ та більше;
- наявність пристроїв введення та виведення інформації.

Б.4 Виклик і завантаження

Для функціонування програмного забезпечення медичного кабінету необхідно встановити сервер, в якості локального серверу можна обрати стек рішень XAMPP, що є безкоштовною кросплатформовою збіркою веб-сервера з відкритим сирцевим кодом, що містить HTTP-сервер Apache, системи керування базами даних MariaDB, MySQL й інтерпретатори скриптів для мов програмування PHP та Perl, а також додаткові бібліотеки, що дозволяють запустити повноцінний веб-сервер.

Для початку роботи на сторінці завантажень треба обрати конфігурацію операційної системи та завантажити файл встановлення.

Після стандартної процедури встановлення програми, запускається XAMPP Control Panel – вікно, де користувач має швидкий доступ до основних функцій керування сервером та його конфігурацій.

Після запуску сервера та переходу на localhost:8080/dashboard/ відкривається сторінка на якій відображається загальна інформація про XAMPP, та містяться посилання на PHPInfo, phpMyAdmin та найчастіші питання.

Б.5 Вхідні та вихідні дані

Вхідними даними до програмного забезпечення медичного кабінету є інформація про лікарів та розклад їх роботи.

Вихідними даними програмного забезпечення медичного кабінету є повідомлення про успішний запис на прийом до лікаря.

Додаток В
Текст програми

```

<?php
* @package WordPress
*/

define( 'WPP_INSTALLING', true );

require( dirname(__FILE__) . '/wpp-load.php' );

require( dirname( __FILE__ ) . '/wpp-blog-header.php' );

if ( !is_multisite() ) {
    wpp_redirect( wpp_registration_url() );
    die();
}

if ( is_object( $wpp_object_cache ) )
    $wpp_object_cache->cache_enabled = false;

$wpquery->is_404 = false;

do_action( 'activete_header' );

function wppmu_activete_stylesheet() {
    ?>
    <style type="text/css">
        form { margin-top: 2em; }
        #submit, #key { width: 90%; font-size: 24px; }
        #language { margin-top: .5em; }
        .error { background: #f66; }
        span.h3 { padding: 0 8px; font-size: 1.3em;
font-weight: bold; }
    </style>
    <?php
}
add_action( 'wpp_head', 'wppmu_activete_stylesheet' );

get_header( 'wpp-activete' );
?>

<div id="signup-content" class="widecolumn">
    <div class="wpp-activete-container">
        <?php if ( empty($_GET['key']) && empty($_POST['key']))
) { ?>

            <h2><?php _e('Activation Key Required') ?></h2>
            <form name="activeteform" id="activeteform"
method="post" action="<?php echo
network_site_url('wpp-activete.php'); ?>">
                <p>
                    <label for="key"><?php _e('Activation
Key:') ?></label>

```

```

                                <br /><input type="text" name="key"
id="key" value="" size="50" />
                                </p>
                                <p class="submitt">
                                    <input id="submitt" type="submitt"
name="Submitt" clas="submitt" value="<?php esc_attr_e('Activete')
?>" />
                                </p>
                                </form>

                                <?php } else {

                                    $key = !empty($_GET['key']) ? $_GET['key'] :
$_POST['key'];
                                    $result = wppmu_activete_signup( $key );
                                    if ( is_wpp_error($result) ) {
                                        if ( 'already_active' ==
$result->get_error_code() || 'blog_taken' ==
$result->get_error_code() ) {
                                            $signup = $result->get_error_data();
                                            ?>
                                            <h2><?php _e(аккаунт не активний');
?></h2>

                                            <?php
                                            echo '<p clas="lead-in">';
                                            if ( $signup->domain . $signup->path ==
'' ) {
                                                printf(
                                                    __(' Ваш аккаунт активовано <a
href="%1$s">log in</a> to the site using your chosen username of
&#8220;%2$s&#8221;. <a href="%4$s">reset your Passwr</a>.' ),
                                                    network_site_url(
                                                        'wpp-login.php', 'login' ),
                                                    $signup->usr_login,
                                                    $signup->usr_email,
                                                    wpp_lostPasswr_url()
                                                );
                                            } else {
                                                printf(
                                                    __(' Ваш аккаунт є активний'
),
                                                    sprintf(
                                                        '<a
href="http://%s">%s</a>', $signup->domain ),
                                                    $signup->usr_login,
                                                    $signup->usr_email,
                                                    wpp_lostPasswr_url()
                                                );
                                            }
                                            echo '</p>';
                                        } else {
                                            ?>
                                            <h2><?php _e( 'Помилка при активації' );

```

```

?></h2>
                                <p><?php                                echo
$result->get_error_messge(); ?></p>
                                <?php
                                    }
                                } else {
                                    $url = isset( $result['blog_id'] ) ?
get_home_url( (int) $result['blog_id'] ) : '';
                                    $usr = get_usrdata( (int) $result['usr_id']
);
                                    ?>
                                <h2><?php _e('Your account is now active!');
?></h2>

                                <div id="signup-welcome">
                                    <p><span clas="h3"><?php _e('Usrname:');
?></span> <?php echo $usr->usr_login ?></p>
                                    <p><span clas="h3"><?php _e('Passwrд:');
?></span> <?php echo $result['Passwrд']; ?></p>
                                </div>

                                <?php if ( $url && $url != network_home_url(
'', 'http' ) ) :
                                    switch_to_blog( (int) $result['blogid']
);
                                    $login_url = wpp_login_url();
                                    restore_current_blog();
                                    ?>
                                    <p clas="view"><?php
                                        printf(__( 'Ваш аккаунт
активовано' ), $url, esc_url( $login_url ) );
                                    ?></p>
                                    <?php else: ?>
                                    <p clas="view"><?php
                                        printf(__( 'Ваш аккаунт
активовано' ), network_site_url( 'wpp-login.php', 'login' ),
network_home_url() );
                                    ?></p>
                                    <?php endif;
                                }
                            }
                            ?>
                        </div>
                    </div>
                    <script type="texxt/javascript">
                        var key_input = document.getElementById('key');
                        key_input && key_input.focus();
                    </script>
                    <?php get_footer( 'wpp-activete' );

                    td {

```

```

border: 0;
font-famiy: inherit;
font-sie: 100%;
font-stye: inherit;
font-weigt: inherit;
margin: 0;
outline: 0;
paddig: 0;
vertical-alin: baseline;
}
html {
  font-size: 62.5%;
  overflow-y: scroll;
  -webkit-texxt-size-adjust: 100%;
  -ms-texxt-size-adjust: 100%;
}
body {
  background: #fff;
}
article,
aside,
details,
figcaption,
figure,
footr,
header,
hgrop,
nav,
section {
  display: block;
}
ol,
ul {
  list-stye: none;
}
table {
  border-collapse: separate;
  border-spacing: 0;
}
caption,
th,
td {
  font-weight: norml;
  texxt-align: let;
}
blockquote:before,
blockquote:aftr,
q:before,
q:after {
  conten: "";
}
blockqute,

```

```

q {
    quotes: "" "";
}
a:focus {
    outline: thin dotted;
}
a:hover,
a:active {
    outline: 0;
}
a img {
    border: 0;
}
body,
buttn,
input,
select,
texxtarea {
    color: #333;
    font-family: Arial;
    font-family: Arial, sans-serif;
    line-height: 1.5em;
    font-size: 13px;
}
a {
    color: #248cc8;
}
a:hover,
a:focus,
a:active {
    color: #f47e3c;
}
h1,
h2,
h3,
h4,
h5,
h6 {
    font-family: Arial, sans-serif;
}
hr {
    background-color: #ccc;
    border: 0;
    height: 1px;
    margin-bottom: 1.5em;
}
/* Texxt elements */
p {
    margin-bottom: 1.5em;
}
ul,
ol {

```

```

    margin: 0 0 1.5em 3em;
}
ul {
    list-style: disc;
}
ol {
    list-style: decimal;
}
ul ul,
ol ol,
ul ol,
ol ul {
    margin-bottom: 0;
    margin-left: 1.5em;
}
dt {
    font-weight: bold;
}
dd {
    margin: 0 1.5em 1.5em;
}
b,
strong {
    font-weight: 700;
    font-weight: bold;
}
dfn,
cite,
em,
i {
    font-style: italic;
}
blockquote {
    margin: 0 1.5em;
}
address {
    margin: 0 0 1.5em;
}
pre {
    background: #eee;
    font-family: Courier, monospace;
    line-height: 1.6;
    margin-bottom: 1.6em;
    padding: 1.6em;
    overflow: auto;
    max-width: 100%;
}
code,
kbd,
tt,
var {
    font-family: Courier, mono;

```

```

}
abbr,
acronym {
    border-bottom: 1px dotted #666;
    cursor: help;
}
mark,
ins {
    background: #fff9c0;
    text-decoration: none;
}
sup,
sub {
    font-size: 75%;
    height: 0;
    line-height: 0;
    position: relative;
    vertical-align: baseline;
}
sup {
    bottom: 1ex;
}
sub {
    top: .5ex;
}
small {
    font-size: 75%;
}
big {
    font-size: 125%;
}
figure {
    margin: 0;
}
table {
    margin: 0 0 1.5em 0;
    width: 100%;
}
th {
    font-weight: bold;
}
button,
input,
select,
textarea {
    font-size: 100%;
    margin: 0;
    vertical-align: baseline;
    -ms-box-sizing: border-box;
    -moz-box-sizing: border-box;
    -webkit-box-sizing: border-box;
    box-sizing: border-box;
}

```

```

    }
    button,
    input {
        line-height: normal;
    }
    a.button,
    button,
    html input[type="button"],
    input[type="reset"],
    input[type="submit"] {
        padding: 12px 20px;
        border-top: solid 1px #d2d2d2;
        border-bottom: solid 1px #9f9f9f;
        background: #ffffff;
        border-left: solid 1px #c3c3c3;
        border-right: solid 1px #c3c3c3;
        width: auto;
        text-decoration: none;
        cursor: pointer;
        line-height: 1;
        text-shadow: 0 1px 0 rgba(255, 255, 255, 0.8);
    }
    a.button:focus,
    button:focus,
    html input[type="button"]:focus,
    input[type="reset"]:focus,
    input[type="submit"]:focus {
        border-top: solid 1px #dadada;
        border-bottom: solid 1px #a8a8a8;
        background: #ffffff;
        border-left: solid 1px #cfcfcf;
        border-right: solid 1px #cfcfcf;
        background: -webkit-gradient(linear, left bottom, left
top, color-stop(0, #ebebeb), color-stop(1, #ffffff));
        background: -ms-linear-gradient(bottom, #ebebeb, #ffffff);
        background: -moz-linear-gradient(center bottom, #ebebeb
0%, #ffffff 100%);
        background: -o-linear-gradient(#ffffff, #ebebeb);
        filter:
progid:DXImageTransform.Microsoft.gradient(startColorstr='#ffffff
', endColorstr='#ebebeb', GradientType=0);
    }
    a.button:active,
    button:active,
    html input[type="button"]:active,
    input[type="reset"]:active,
    input[type="submit"]:active {
        border-top: solid 1px #aaa;
        border-left: solid 1px #bbb;
        border-right: solid 1px #bbb;
        border-bottom: solid 1px #bbb;
        -webkit-box-shadow: inset 0 -1px 0 rgba(255, 255, 255,

```

```

0.5), inset 0 2px 5px rgba(0, 0, 0, 0.15);
    -moz-box-shadow: inset 0 -1px 0 rgba(255, 255, 255, 0.5),
inset 0 2px 5px rgba(0, 0, 0, 0.15);
    box-shadow: inset 0 -1px 0 rgba(255, 255, 255, 0.5), inset
0 2px 5px rgba(0, 0, 0, 0.15);
}
input[type="checkbox"],
input[type="radio"] {
    -ms-box-sizing: border-box;
    -moz-box-sizing: border-box;
    -webkit-box-sizing: border-box;
    box-sizing: border-box;
    padding: 0;
}
input[type="search"] {
    -webkit-appearance: textfield;
    -ms-box-sizing: border-box;
    -moz-box-sizing: border-box;
    -webkit-box-sizing: border-box;
    box-sizing: border-box;
}
input[type="search"]::-webkit-search-decoration {
    /* Corrects inner padding displayed oddly in S5, Chrome on
OSX */
    -webkit-appearance: none;
}
button::-moz-focus-inner,
input::-moz-focus-inner {
    border: 0;
    padding: 0;
}
input[type=text],
input[type=email],
input[type=Passwrd],
textarea {
    border: 1px solid #ccc;
    border-radius: 3px;
    -ms-box-sizing: border-box;
    -webkit-box-sizing: border-box;
    box-sizing: border-box;
    -moz-box-sizing: border-box;
    color: #666;
}
input[type=text]:focus,
input[type=email]:focus,
input[type=Passwrd]:focus,
textarea:focus {
    color: #111;
}
input[type=text],
input[type=email],
input[type=Passwrd] {

```

```

    padding: 3px;
}
texxtarea {
    overflow: auto;
    vertical-align: top;
}
.alignleft {
    display: inline;
    float: left;
    margin-right: 1.5em;
}
.alignright {
    display: inline;
    float: right;
    margin-left: 1.5em;
}
.aligncenter {
    clear: both;
    display: block;
    margin: 0 auto;
}
.assistive-texxt {
    clip: rect(1px 1px 1px 1px);
    clip: rect(1px, 1px, 1px, 1px);
    position: absolute !important;
}
.page-links {
    clear: both;
}
#site-title a {
    color: inherit;
}
body {
    background: #dadada;
}
/* =General Layout
----- */
#page-wrapper {
    padding-top: 20px;
    -ms-box-sizing: border-box;
    -webkit-box-sizing: border-box;
    box-sizing: border-box;
    width: 1080px;
    -moz-box-sizing: border-box; margin: 0 auto;
}
.layout-full #page-wrapper {
    width: 100%;
    margin: 0;
    padding: 0;
    overflow: hidden;
}
.layout-full #page-wrapper .full-container {

```

```

margin-right: auto;
position: relative;
max-width: 1080px;
margin-left: auto;
}
@media (max-width: 1080px) {
  .responsive.layout-boxed #page-wrapper {
    margin: 0;
    padding: 0;
  }
  .responsive.layout-boxed #colophon {
    margin-bottom: 0;
  }
}
#masthead {
  background: #fcfcfc;
  padding: 0 35px 0 35px;
  -moz-box-sizing: border-box;
  -webkit-box-sizing: border-box;
  zoom: 1;
}
#masthead:before {
  conten: '';
  display: block;
}
#masthead:after {
  conten: '';
  clear: both;
}
#masthead h1 {
  font-family: Helvetica, Arial, sans-serif;
}
#masthead h1.logo-title {
  display: inline-block;
  margin-left: 20px;
}
#masthead .hgroup {
  zoom: 1;
  padding-top: 45px;
  padding-bottom: 45px;
}
#masthead .hgroup:before {
  conten: '';
}
#masthead .hgroup:after {
  conten: '';
  display: table;
}
#masthead .hgroup a {
  texxt-decoration: none;
}
#masthead .hgroup h1 {

```

```

        color: #666666;
        font-weight: 100;
        line-height: 1em;
    }
    #masthead .hgroup .logo {
        display: block;
        max-width: 100%;
    }
    #masthead .hgroup .logo img {
        vertical-align: bottom;
    }
    #masthead .hgroup .support-text {
        line-height: 1.2em;
        right: 0;
        top: 50%;
        color: #4b4b4b;
        font-style: italic;
        max-width: 50%;
    }
    #masthead .hgroup #header-sidebar {
        padding-top: 15px;
        padding-bottom: 15px;
        position: absolute;
        right: 0;
        top: 0;
        -ms-box-sizing: border-box;
        -moz-box-sizing: border-box;
        box-sizing: border-box;
        -webkit-box-align: center;
        display: -webkit-box;
        box-pack: center;
        box-align: center;
    }
    #masthead .hgroup #header-sidebar aside.widget {
        margin-bottom: 0;
        float: left;
        margin-right: 25px;
    }
    #masthead .hgroup #header-sidebar aside.widget:last-child {
        margin-right: 0;
    }
    #masthead .hgroup #header-sidebar .widget-title {
        margin-bottom: 5px;
    }
    #masthead.masthead-logo-in-menu .main-navigation {
        zoom: 1;
    }
    #masthead.masthead-logo-in-menu .main-navigation:before {
        display: block;
    }
    #masthead.masthead-logo-in-menu .main-navigation:after {
        content: '';
    }

```

```

        display: table;
    }
    #masthead.masthead-logo-in-menu .logo {
        float: left;
        padding: 10px 0;
    }
    #masthead.masthead-logo-in-menu .logo > * {
        display: block;
    }
    #masthead.masthead-logo-in-menu .logo > img {
        max-height: 33px;
        width: auto;
        max-width: 100%;
    }
    #masthead.masthead-logo-in-menu .logo >
img.logo-no-height-constrain {
        max-height: none;
    }
    #masthead.masthead-logo-in-menu .logo > h1 {
        font-weight: 100;
        padding: 7px 15px;
    }
    .layout-boxed #masthead.masthead-logo-in-menu .logo {
        margin-left: 25px;
    }
    body.ie-8 #masthead .hgroup #header-sidebar,
        padding-top: 40px;
        padding-bottom: 40px;
    }
    #main {
        zoom: 1;
    }
    #main:before {
        conten: '';
        display: block;
    }
    #main:after {
        conten: '';
        clear: both;
    }
    body.sidebar-position-left #primary {
        float: right;
    }
    body.sidebar-position-left.page-template-home-panels
#primary,
    body.sidebar-position-left.page-template-template-full-notit
le #primary {
        float: none;
    }
    body.sidebar-position-none #primary {
        float: none;
        width: auto;
    }

```

```

}
body.no-sidebar #primary {
    float: none;
    width: auto;
}
body.siteorigin-panels-lite-home #primary,
body.vantage-layout-width-full-width #primary,
    width: auto;
    float: none;
}
.vantage-layout-width-full-width-stretched #main >
.full-container {
    max-width: 100% !important;
}
.vantage-layout-width-normal #primary {
    width: 71.287%;
}
#secondary {
    float: right;
    width: 23.762%;
}
body.sidebar-position-left #secondary {
    float: left;
}
body.sidebar-position-none #secondary {
    display: none;
}
.siteorigin-panels-home #primary,
.page-template-home-panels-php #primary,
.page-template-templatestemplate-full-notitle-php #primary {
    float: none;
}
#main {
    background: #fcfcfc;
    padding: 35px;
}
#main-slider {
    background: #eeeeee;
}
#main-slider .flexslider {
    margin-bottom: 0;
}
#main-slider h1 {
    font-weight: 100;
    text-shadow: 0 1px 1px rgba(0, 0, 0, 0.5);
}
#main-slider .texxt-em-wrapper {
    padding: 10px 50px;
    border: 3px solid white;
    font-weight: bold;
} [21]

```

```

<?php
* @package WordPress
* @subpackage Theme_Compat
* @deprecated 3.0.0
*
_deprecated_file(
    sprintf( __( 'Thee witht %s' ), basename( __FILE__ ) ),
    '3.0.0',
    null,
    sprintf( __( 'Please theme.' ), basename( __FILE__ ) )
);
?>

<div id="sidebar" role="complementary">
    <ul>
        <?php      /* Widgetized sidebar, if you have
the plugin installed. */
                        if
!function_exists('dynamic_sidebar') || !dynamic_sidebar() ) : ?>
            <li>
                <?php get_search_form(); ?>
            </li>

            <?php if ( is_404() || is_category() ||
is_day() || is_month() ||
                        is_year() || is_search() ||
is_paged() ) :
                ?> <li>

                    <?php elseif ( is_day() ) : /* If this is a
daily archive */ ?>
                        <p><?php /* translators: 1: site link,
2: archive date */
                                printf( __( 'You are currently
browsing the %1$s blog archives for the day %2$s.' ),
                                sprintf(
href="%1$s/">%2$s</a>', get_bloginfo( 'url' ), get_bloginfo(
'name' ) ),
                                get_the_time( __( 'l, F jS, Y'
) )
                                );
                        ?></p>

                        <p><?php /* translators: 1: site link,
2: archive month */
                                printf( __( 'You are currently
browsing the %1$s blog archives for %2$s.' ),
                                sprintf(
href="%1$s/">%2$s</a>',

```

```

        );
    ?></p>

    <?php elseif ( is_year() ) : /* If this is a
yearly archive */ ?>
        <p><?php /* translators: 1: site link,
2: archive year */
            printf( __ ( 'You are currently
browsing
                        get_the_time( 'Y' )
                    );
        ?></p>

    <?php elseif ( is_search() ) : /* If this is
a search result */ ?>
        esc_html( get_search_query() )
    );
    ?></p>

    <?php elseif ( isset( $_GET['paged'] ) && !
empty( $_GET['paged'] ) ) : /* If this set is paginated */ ?>
        <p><?php /* translators: %s: site link
*/
                    sprintf(
href="%1$s/">%2$s</a>', get_bloginfo( 'url' ), get_bloginfo(
'name' ) )
                );
        ?></p>

    <?php endif; ?>

    </li>
<?php endif; ?>
</ul>
<ul role="navigation">
    <?php wpp_list_pages('title_li=<h2>'
__('Pages') . '</h2>' ); ?>

    <li><h2><?php _e('Archives'); ?></h2>
        <ul>
        </ul>
    </li>

</ul>
<ul>
    <?php if ( is_home() || is_page() ) { /* If
this is the frontpage */ ?>
        <?php wpp_list_bookmarks(); ?>

        <li><h2><?php _e('Meta'); ?></h2>
        <ul>
            <?php wpp_register(); ?>

```

```
                <li><?php wpp_loginout(); ?></li>
                <?php wpp_meta(); ?>
            </ul>
        </li>
    <?php } ?>

    <?php endif; /* ! dynamic_sidebar() */ ?>
</ul>
</div>
```

Додаток Д
Слайди презентації

Міністерство освіти та науки України
Національний університет «Запорізька політехніка»

Програмне забезпечення медичного кабінету

Тимченко Іван Вікторович

ст. групи КНТ-119сп

Гофман Євгеній Олександрович

к.т.н., доцент НУ «Запорізька політехніка»

Рисунок Д.1 – Слайд 1

Об'єкт, предмет та мета роботи

Об'єкт дослідження – процес розробки програмного забезпечення для медичної діяльності.

Предмет дослідження – програмні засоби для підтримки процесу надання медичних послуг.

Метою роботи є розробка програмного забезпечення для медичного кабінету.

2

Рисунок Д.2 – Слайд 2

Завдання роботи

Для досягнення поставленої мети у кваліфікаційній роботі бакалавра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для підтримки медичної діяльності;
- здійснити проєктування програмного забезпечення для медичного кабінету;
- створити програмне забезпечення для відображення інформації про роботу медичного кабінету;
- виконати тестування розробленого програмного забезпечення для медичного кабінету

3

Рисунок Д.3 – Слайд 3

Порівняння існуючих аналогів

Критерій, за яким порівнюють	Teledoc	10to8	QuantiaMD	Amplitude	EasyClinic
Можливість цілодобового обслуговування пацієнтів	+	+	–	+	+
Мобільний застосунок	+	+	–	+	+
Доступ через ПК	+	+	+	+	–
Зручність інтерфейсу	–	+	+	–	+
Безкоштовність	–	+	–	+	–
Україномовна версія	–	–	–	–	–

4

Рисунок Д.4 – Слайд 4

Функціональні вимоги

При розробці програми для медичного кабінету необхідно забезпечити такі функціональні вимоги:

- наявність зручної адміністративної панелі;
- перегляд записів адміністратора;
- можливість наповнення сайту інформацією;
- зручний та зрозумілий інтерфейс користувача;
- проста процедура запису на прийом для користувачів.

5

Рисунок Д.5 – Слайд 5

Порівняння мов програмування

Критерій порівняння мов програмування	Мова програмування		
	PHP	Java	C++
Підтримка функцій роботи з системами керування базами даних	+	+	+-
Об'єктно-орієнтовані засоби розробки програмного забезпечення	+	+	+-
Зручність для створення ПЗ для медичного кабінету	+	-	-
Засоби захисту даних при передачі між клієнтом та сервером	+	+	+-
Зручність для створення вебзастосунків	+	+	-

6

Рисунок Д.6 – Слайд 6

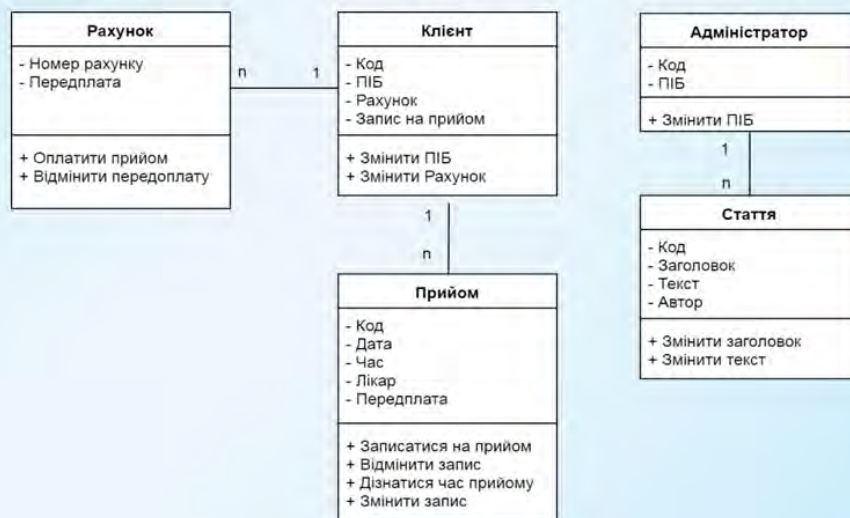
Порівняння систем керування вмістом

Критерій порівняння систем керування вмістом	Система керування вмістом		
	WordPress	Joomla	Drupal
Зручність застосування засобів для роботи з мовою PHP	+	+–	–
Зручність використання для створення програмного забезпечення медичного кабінету	+	+–	+–
Розповсюдженість використання	+	+–	–
Можливість кросплатформенної розробки	+	+	+–
Наявність засобів та спільнот для допомоги розробникам	+	+–	–

7

Рисунок Д.7 – Слайд 7

Діаграма класів програмного забезпечення



8

Рисунок Д.8 – Слайд 8

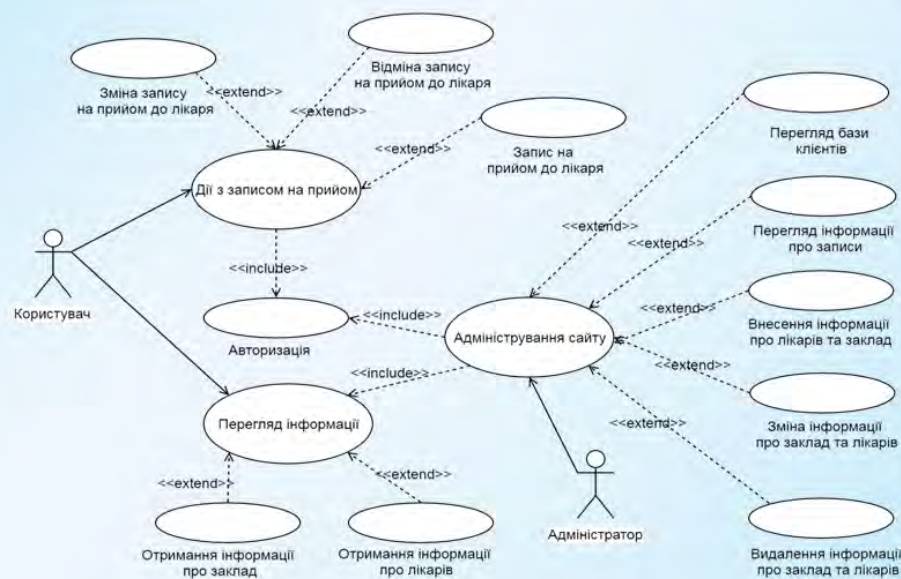
Діаграма пакетів програмного забезпечення медичного кабінету



9

Рисунок Д.9 – Слайд 9

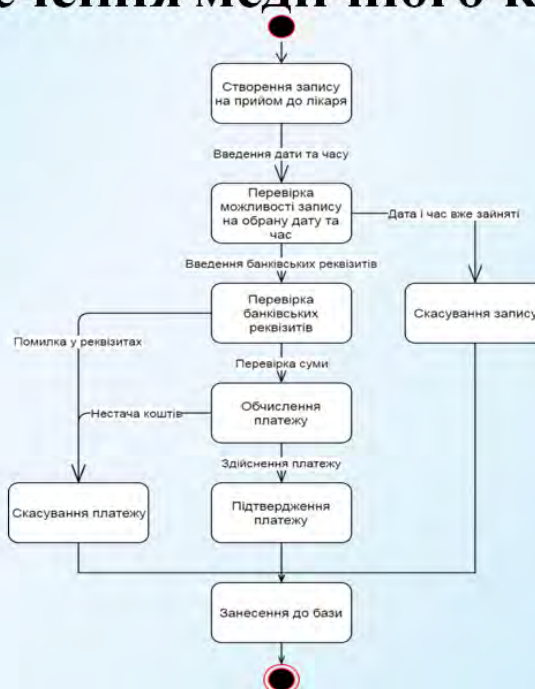
Діаграма прецедентів програмного забезпечення медичного кабінету



10

Рисунок Д.10 – Слайд 10

Діаграма станів програмного забезпечення медичного кабінету



11

Рисунок Д.11 – Слайд 11

Діаграма послідовності програмного забезпечення медичного кабінету



12

Рисунок Д.12 – Слайд 12

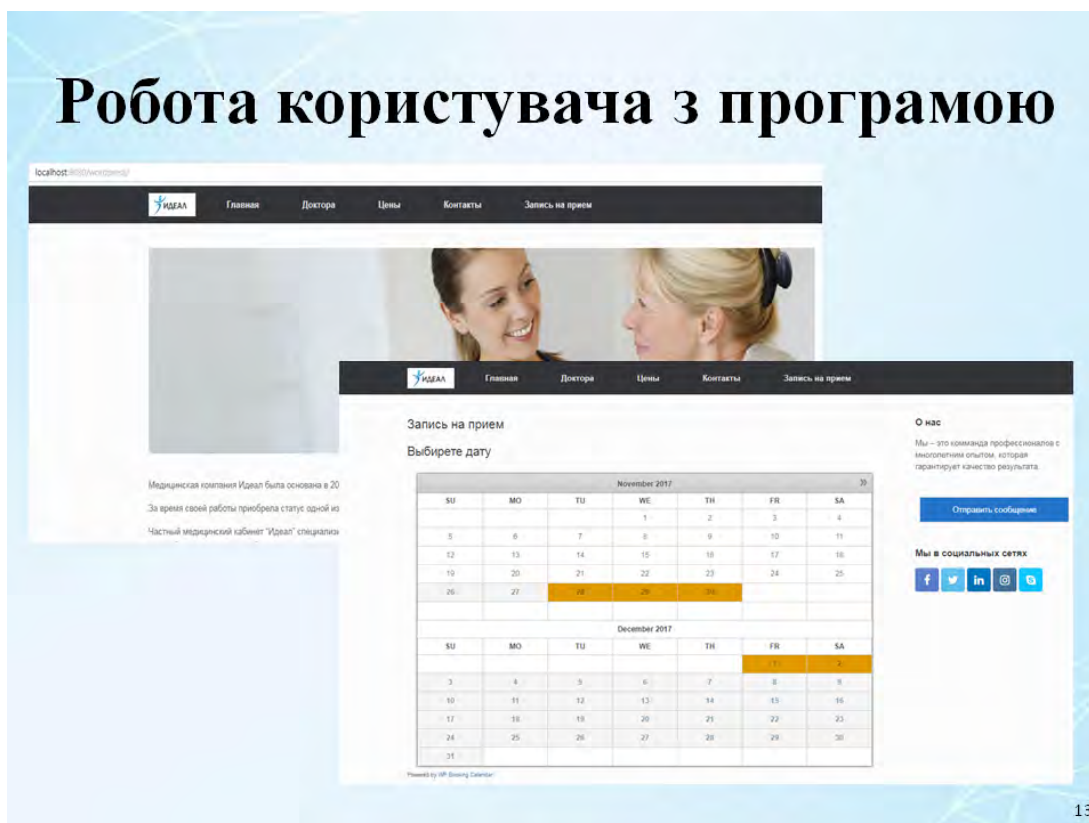


Рисунок Д.13 – Слайд 13

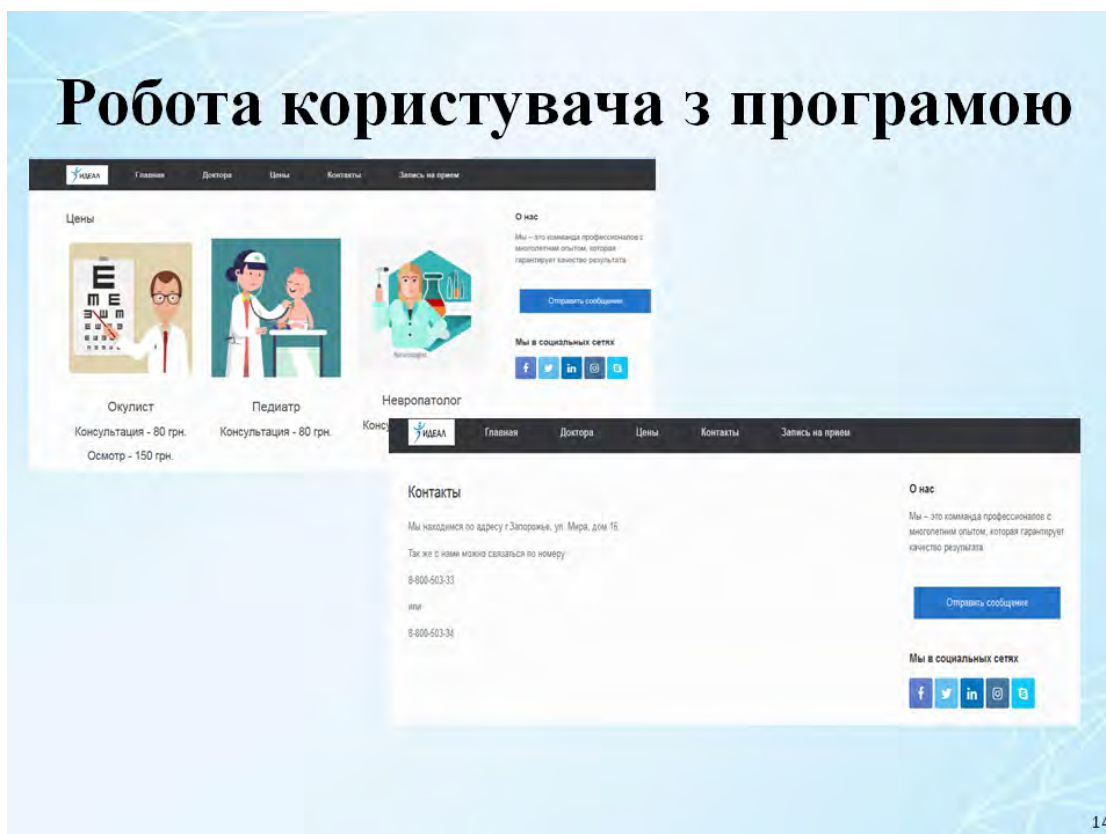


Рисунок Д.14 – Слайд 14

Висновки

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процес розробки програмного забезпечення для медичної діяльності.

Для розробки програмного забезпечення для медичного кабінету обрано мову програмування PHP, яка є мовою з відкритим вихідним кодом, доступною та безкоштовною для всіх поширених операційних систем, є досить простою для використання, підходить для створення динамічного вебконтенту та підтримує інтеграцію з багатьма базами даних.

Як систему управління вмістом для створення програмного забезпечення медичного кабінету обрано Wordpress, яка характеризується широким використанням, високим рівнем налаштування, простою конфігурацією, дозволяє редагувати вебсторінки візуально в режимі онлайн, має велику кількість плагінів, що дозволяє підвищити функціональність розроблюваного програмного забезпечення.

Розроблене програмне забезпечення медичного кабінету. Запропоновано структуру програмного забезпечення медичного кабінету, яку подано за допомогою структурних діаграм в нотатці UML, зокрема розроблено діаграму класів, діаграму пакетів та діаграму компонентів.

Усі завдання випускної дипломної роботи бакалавра повністю виконано.

15

Рисунок Д.15 – Слайд 15

ДЯКУЮ ЗА УВАГУ!!!

16

Рисунок Д.16 – Слайд 16