

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

ОБЛІКУ ПАЦІЄНТІВ У МЕДИЧНОМУ ЗАКЛАДІ

SOFTWARE FOR PATIENT

REGISTRATION IN A MEDICAL FACILITY

Виконав(ла): студент(ка) 4 курсу, групи КНТ-121

Спеціальності 121 Інженерія програмного

(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

СЕРГЕВНІН І.В.

(ПРИЗВИЩЕ та ініціали)

Керівник СКРУПСЬКИЙ С.Ю.

(ПРИЗВИЩЕ та ініціали)

Рецензент ІЛЛЯШЕНКО М.Б.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
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.
Сергій СУББОТІН
“ ” 2025 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

СЕРГЕВНИНА Івана Владиславовича

(ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Програмне забезпечення обліку пацієнтів у медичному закладі. Software for Patient Registration in a Medical Facility

керівник проєкту (роботи) к.т.н., доцент, СКРУПСЬКИЙ Степан Юрійович,
(науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 28 ” квітня 2025 року № 209

2. Строк подання студентом проєкту (роботи) 03 червня 2025 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, кількість слайдів, плакатів)

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	СКРУПСЬКИЙ С.Ю., доцент		
Нормоконтроль	КАМІНСЬКА Ж.К., асистент		

7. Дата видачі завдання “28” квітня 2025 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 структури 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) Іван СЕРГЕВНІН
(Імя ПРІЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Степан СКРУПСЬКИЙ
(Імя ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
91 с., 3 табл., 32 рис., 3 дод., 17 джерел.

LARAVEL, PHP, ОБЛІК ПАЦІЄНТІВ, ПРОГРАМНИЙ ЗАСТОСУНОК,
МОВА ПРОГРАМУВАННЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

Об'єкт дослідження – процес розробки програмного забезпечення для медичних закладів.

Предмет дослідження – програмні засоби для медичних закладів.

Мета роботи – розробка програмного забезпечення для підтримки процесу обліку пацієнтів у медичному закладі.

Матеріали, методи та технічні засоби: мова програмування PHP, фреймворк Laravel.

Результати. Розроблено проєктні рішення для створення програмного забезпечення для підтримки процесу обліку пацієнтів у медичному закладі. Створено програмне забезпечення для обліку пацієнтів у медичному закладі. Проведено тестування розробленої програмної системи для обліку пацієнтів у медичному закладі.

Висновки. Мету роботи досягнуто. Розроблено програмне забезпечення для обліку пацієнтів у медичному закладі за допомогою мови програмування PHP та фреймворку Laravel.

Галузь використання – медичні заклади.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 91 pages, 3 tables, 32 figures, 3 appendixes, 17 sources.

LARAVEL, PHP, PATIENT ACCOUNTING, SOFTWARE APPLICATION, PROGRAMMING LANGUAGE, SOFTWARE.

The object of research is the process of developing software for medical institutions.

The subject of the research is software for medical institutions.

The purpose of this work is to develop the software to support the process of patient accounting in a medical institution.

Materials, methods and technical tools: PHP programming language, Laravel framework.

Results. Project solutions for the creation of software for process of patient accounting in a medical institution has been designed. The software to support the process of patient accounting in a medical institution has been developed. The developed software system for patient accounting in a medical institution has been tested.

Conclusions. The goal of the work has been achieved. The software tools for patient accounting in a medical institution using the PHP programming language and the Laravel framework have been developed.

Scope of use – medical institutions.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ.....	9
1 Аналіз предметної області.....	11
1.1 Програмне забезпечення для підтримки роботи медичних закладів	11
1.2 Аналіз програмного забезпечення для обліку пацієнтів у медичному закладі	21
1.3 Висновки за розділом 1	30
2 Матеріали і методи	32
2.1 Вибір мови програмування	32
2.2 Вибір фреймворку для створення програмного забезпечення для обліку пацієнтів у медичному закладі	35
2.3 Висновки за розділом 2	38
3 Опис програми	39
3.1 Структура програмного забезпечення для обліку пацієнтів у медичному закладі	39
3.2 Функціонування програмного забезпечення для обліку пацієнтів у медичному закладі	40
3.3 Розробка бази даних програмного забезпечення для обліку пацієнтів у медичному закладі	42
3.4 Проєктування інтерфейсу програмного забезпечення для обліку пацієнтів у медичному закладі	45
3.5 Висновки за розділом 3	48
4 Експлуатація, тестування та експериментальне дослідження програми.....	49
4.1 Призначення й умови застосування програми	49
4.2 Характеристики програми для обліку пацієнтів у медичному закладі...	50
4.3 Інструкція по експлуатації програми.....	51
4.3.1 Звернення до програми.....	51
4.3.2 Вхідні й вихідні дані	51

4.3.3 Повідомлення.....	52
4.4 Виконання програмного забезпечення для обліку пацієнтів у медичному закладі	52
4.5 Тестування програмного забезпечення для обліку пацієнтів у медичному закладі	56
4.6 Висновки за розділом 4	56
Висновки	57
Перелік джерел посилань	60
Додаток А Технічне завдання	62
Додаток Б Фрагмент тексту програми	67
Додаток В Слайди презентації	84

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

БД – база даних;

МЗ – медичний заклад;

ОП – облік пацієнтів;

ПЗ – програмне забезпечення.

ВСТУП

У сучасних умовах розвитку медичної галузі постійно зростають вимоги до якості надання медичних послуг, оперативності обслуговування пацієнтів і точності ведення медичної документації. Традиційні паперові методи обліку є застарілими, трудомісткими, схильними до помилок і потребують значних людських ресурсів для обробки інформації. Тому впровадження програмного забезпечення для автоматизації обліку пацієнтів стає особливо актуальним [1]-[3].

Використання спеціалізованих програмних систем сприяє зниженню кількості медичних помилок завдяки точності та повноті збережених даних, оптимізує роботу персоналу шляхом автоматизації рутинних процесів, а також підвищує рівень захисту персональних даних пацієнтів. Крім того, такі програмні системи дозволяють оперативно формувати звітність для внутрішнього використання та для органів охорони здоров'я [4], [5].

Додатково актуальність питання розробки подібних програмних засобів підсилюється глобальними тенденціями цифровізації медицини, серед яких важливу роль відіграють електронна медична картка, розвиток телемедицини та використання аналітики великих обсягів медичних даних для підтримки клінічних рішень. Сучасна медицина вимагає наявності ефективних, безпечних і зручних систем обліку пацієнтів [6], [7].

Проте деякі програми для обліку пацієнтів у медичному закладі мають високу вартість придбання, впровадження та обслуговування, що може бути непосильним для невеликих медичних закладів. Крім того, складність налаштування та інтеграції складного медичного програмного забезпечення може призводити до затримок у роботі та вимагати додаткових ресурсів. Крім того, надмірна стандартизація процесів у деяких програмних системах для обліку роботи медичних закладів може обмежувати індивідуальний підхід до пацієнтів і створювати додаткові адміністративні навантаження на лікарів.

Тому актуальною є розробка програмного забезпечення для обліку пацієнтів у медичному закладі. У дипломній кваліфікаційній роботі бакалавра розв'язується актуальне завдання розробки програмного забезпечення для обліку пацієнтів у медичному закладі.

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

- виконати аналіз предметної області та програмних засобів для підтримки роботи медичних закладів;
- здійснити проєктування програмного забезпечення для обліку пацієнтів у медичному закладі;
- створити програмне забезпечення для обліку пацієнтів у медичному закладі;
- виконати тестування розробленого програмного забезпечення для обліку пацієнтів у медичному закладі.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Програмне забезпечення для підтримки роботи медичних закладів

У сфері медичних послуг програмні рішення, що застосовуються для обліку пацієнтів, виступають важливим елементом управління клінічними процесами. Використання таких систем сприяє впорядкуванню роботи, усуненню зайвих адміністративних кроків та підвищенню якості обслуговування пацієнтів. Програмні продукти для обліку дозволяють здійснювати автоматичний запис даних, координувати роботу з медичними документами та оптимізувати взаємодію між співробітниками лікувальних закладів [1].

Такі програмні рішення допомагають медичним установам підтримувати відповідність нормативним вимогам і знижують ризик неточностей, що можуть виникати під час ручного введення даних. Їхнє використання створює умови для більшої прозорості в управлінні медичною інформацією, покращуючи доступ до актуальної інформації про пацієнтів. У контексті цифрової трансформації охорони здоров'я впровадження спеціалізованих програм розглядається як крок до підвищення ефективності роботи медичних закладів [1].

Програмне забезпечення, призначене для реєстрації пацієнтів, виконує функцію централізованого сховища даних, дозволяючи швидко створювати нові записи, оновлювати інформацію, відстежувати історію лікування та формувати звіти. Такий підхід полегшує процес призначення прийомів, поліпшує управління електронною медичною документацією та забезпечує більш точне ведення обліку. Автоматизація цих процесів дає змогу знизити навантаження на персонал, зменшити кількість помилок і сприяти підвищенню рівня медичного обслуговування [1].

Інтеграція програмного забезпечення для обліку пацієнтів стає стратегічною необхідністю для закладів, що прагнуть адаптуватися до

цифрових викликів сучасної медицини. Партнерство з технологічними постачальниками, які спеціалізуються на розробці таких рішень, дозволяє установам отримати доступ до гнучких, масштабованих систем, здатних підтримувати довгостроковий розвиток та ефективно реагувати на зміни потреб у сфері охорони здоров'я. Впровадження подібних систем відкриває нові можливості для підвищення якості управління, скорочення витрат і покращення комунікації між усіма учасниками медичного процесу [1].

Програмне забезпечення, що використовується для обліку пацієнтів у медичних закладах, виконує важливу роль у забезпеченні ефективного управління інформацією про пацієнтів. Його застосування сприяє створенню єдиного простору для збереження, оновлення та доступу до даних, що дозволяє зменшити потребу у дублюванні записів та знижує ризик помилок під час обробки інформації. У процесі взаємодії медичних працівників із такими системами забезпечується можливість швидкого внесення змін до інформаційних карток, що допомагає оперативно реагувати на потреби пацієнтів [1], [2].

Функціональність програмного забезпечення дозволяє налагодити обмін інформацією між різними підрозділами медичного закладу, забезпечуючи безперервність обслуговування та узгодженість даних. Кожна зміна, внесена до електронних записів, відображається у спільній базі, що унеможливорює втрату важливої інформації. Водночас система підтримує різні форми взаємодії, що дає змогу пацієнтам обирати найбільш зручний спосіб реєстрації та зв'язку з закладом через онлайн сервіси, мобільні застосунки чи спеціальні термінали [1], [2].

Завдяки можливості інтеграції з іншими інформаційними системами медичного закладу, програмне забезпечення забезпечує доступ до актуальних даних лікарями, адміністраторами та іншим персоналом, що спрощує координацію роботи та сприяє прийняттю обґрунтованих рішень. Аналітичні інструменти, що включені до системи, допомагають контролювати ключові

показники діяльності та дозволяють своєчасно виявляти проблемні моменти для подальшого вдосконалення процесів [1], [2].

Окремої уваги заслуговує захист даних, оскільки програмне забезпечення містить вбудовані механізми безпеки для збереження конфіденційності інформації про пацієнтів. Застосування таких рішень гарантує дотримання нормативних вимог та мінімізує ризики, пов'язані з несанкціонованим доступом або витоком даних. Окрім цього, системи такого типу розроблені з урахуванням можливості подальшого масштабування та налаштування відповідно до специфіки роботи конкретного медичного закладу, що забезпечує гнучкість у використанні та адаптацію до зростаючих потреб установи [1], [2].

Програмне забезпечення для реєстрації пацієнтів характеризується здатністю об'єднувати різноманітні канали взаємодії в єдину цифрову платформу, що забезпечує зручний доступ до послуг реєстрації. Пацієнти отримують можливість оновлювати власні дані та записуватися на прийом через Інтернет ресурси чи термінали в медичних закладах, що сприяє швидшому обслуговуванню та підвищенню рівня задоволеності. Системи автоматично надсилають нагадування про візити, зменшуючи ймовірність пропущених прийомів, а також підтримують різні пристрої, що дає змогу кожному користуватися зручним для нього способом доступу [1], [2].

Такі рішення передбачають легке поєднання з іншими цифровими інструментами, що використовуються в медичних установах, забезпечуючи єдину інформаційну базу та синхронізацію даних у режимі реального часу. Це дозволяє уникати дублювання записів і забезпечує своєчасний доступ медичних працівників до необхідної інформації. Обмін даними між різними системами, як-от фінансові, клінічні чи лабораторні модулі, відбувається без затримок, що спрощує робочі процеси й підвищує точність інформації [1], [2].

Програмні продукти також оснащені аналітичними інструментами, що надають можливість стежити за показниками діяльності через інтерактивні панелі та формувати звіти для управління процесами. Візуалізація даних

допомагає виявляти слабкі місця й оцінювати ефективність роботи, сприяючи обґрунтованому прийняттю рішень. Аналіз ключових показників дозволяє підтримувати належний рівень обслуговування й оперативно реагувати на проблеми [1], [2].

Крім того, рішення розробляються з урахуванням потреб різних медичних установ, що дозволяє адаптувати їх під специфіку конкретної організації. Програмне забезпечення здатне розширюватися відповідно до зростаючих обсягів роботи, додаючи нові функції та можливості без порушення основної діяльності. Така гнучкість сприяє ефективному використанню ресурсів і підтримує розвиток медичних закладів із мінімальними додатковими витратами [1], [2].

Програмне забезпечення, що використовується для реєстрації пацієнтів, стає основою для модернізації системи охорони здоров'я. Завдяки впровадженню цифрових рішень процес взаємодії з пацієнтами набуває нових рис: підвищується комфорт, скорочується час очікування, а самі пацієнти отримують позитивні враження від обслуговування. Такий підхід допомагає формувати довіру до медичних установ і створює відчуття турботи та уваги з боку медичних працівників [2], [3].

Відмова від паперових процедур дозволяє уникнути численних помилок, властивих традиційним методам ведення документації, і значно зменшує витрати, пов'язані з адміністративною діяльністю. Автоматизація реєстраційних операцій звільняє персонал від рутинних завдань, дозволяючи зосередити увагу на безпосередньому догляді за пацієнтами, а також забезпечує більш точне ведення облікових даних, що мінімізує ризики фінансових та юридичних непорозумінь [2], [3].

Цифровий формат обробки даних підвищує рівень їхньої достовірності та забезпечує їхнє оновлення у реальному часі, що позитивно позначається на прийнятті клінічних рішень і оптимізації розподілу ресурсів. Безперервний обмін інформацією між різними компонентами системи охорони здоров'я стає можливим завдяки інтеграції реєстраційного програмного забезпечення, що

дозволяє оперативно отримувати важливі відомості та своєчасно реагувати на потреби пацієнтів [2], [3].

Захист даних і відповідність нормативним вимогам залишаються пріоритетними завданнями, особливо в умовах підвищеної уваги до інформаційної безпеки. Використання спеціалізованого програмного забезпечення передбачає дотримання сучасних стандартів безпеки та правових норм, що забезпечує надійність обробки конфіденційної інформації та сприяє підтримці довіри з боку як пацієнтів, так і контролюючих органів [2], [3].

Постійний розвиток і впровадження інновацій у програмне забезпечення для підтримки медичної діяльності, зокрема, й програмні модулі для обліку пацієнтів, підтримуються завдяки можливостям аналітики та збору відгуків у реальному часі. Це дозволяє медичним організаціям регулярно переглядати та удосконалювати внутрішні процеси, залишаючись гнучкими та відкритими до змін. Такий підхід сприяє збереженню конкурентних позицій та підвищує якість медичних послуг [2], [3].

Програмні засоби для реєстрації пацієнтів продовжують розвиватися, адаптуючись до сучасних потреб медичних закладів. У майбутньому очікується, що такі системи все більше використовуватимуть можливості штучного інтелекту та алгоритмів машинного навчання, завдяки чому процеси реєстрації набуватимуть ще більшої швидкості та точності. Ці технології надаватимуть змогу прогнозувати завантаженість, індивідуалізувати взаємодію з пацієнтами й автоматично обробляти інформацію, що позитивно впливатиме на якість обслуговування та мінімізуватиме ризик помилок [2], [3].

Поширення хмарних сервісів визначає ще один важливий напрямок розвитку. Використання таких рішень забезпечує гнучкий доступ до даних, можливість віддаленої роботи та обробки інформації у реальному часі. Завдяки мобільним функціям пацієнтам надається можливість зручно керувати записами та реєструватися через мобільні телефони чи інші пристрої, що підвищує доступність медичних послуг і рівень залученості [2], [3].

Подальша еволюція програмного забезпечення також передбачає тісніше поєднання з іншими цифровими системами у сфері охорони здоров'я. Завдяки цьому дані передаватимуться безперешкодно між різними платформами, що сприятиме комплексному підходу до обслуговування пацієнтів, підвищенню ефективності та якості медичних послуг [2], [3].

Крім того, на перший план виходить орієнтація на потреби користувачів і впровадження аналітичних можливостей, здатних надавати оперативну інформацію. Застосування інтерактивних панелей управління та гнучких інструментів для аналізу допомагатиме медикам ухвалювати обґрунтовані рішення, що сприятиме безперервному вдосконаленню процесів і стимулюватиме інновації в галузі охорони здоров'я [2], [3].

У сучасних медичних закладах все частіше використовуються програмні інструменти для організації процесу прийому пацієнтів. Такі програмні рішення дає змогу автоматизувати збір інформації, включаючи особисті дані, відомості про страхування та історію хвороби, що призводить до підвищення точності обробки даних і загальної ефективності роботи установи. Застосування електронних форм і систем попереднього запису дозволяє зменшити кількість паперових документів і скоротити час, який витрачається пацієнтами на реєстрацію, водночас спрощуючи завдання для персоналу [3], [4].

Програмні рішення в цій сфері сприяють швидшому обслуговуванню та позитивно впливають на загальний досвід відвідування медичного закладу. Наявність таких систем допомагає забезпечити надійність і точність даних, мінімізуючи можливість помилок при введенні інформації вручну. Крім того, автоматизація рутинних завдань дає можливість співробітникам більше уваги приділяти догляду за пацієнтами, знижуючи навантаження на адміністративний персонал [3], [4].

Використання подібного програмного забезпечення позитивно відбивається не лише на роботі закладу, а й на якості медичної допомоги в цілому. Завдяки скороченню часу очікування та зменшенню обсягу паперових

процедур створюються умови для більш комфортного та організованого процесу обслуговування. Точне збирання даних сприяє підвищенню стандартів догляду та оптимізації управління медичними послугами. У комплексі це рішення впливає на різні етапи взаємодії з пацієнтами, формуючи більш ефективну та безпечну систему охорони здоров'я [3], [4].

Вибір відповідного програмного забезпечення для реєстрації пацієнтів є важливим кроком у вдосконаленні медичної практики. Цей процес вимагає врахування різноманітних аспектів, які впливають на ефективність роботи та якість обслуговування пацієнтів [3], [4].

Одним із ключових факторів є зручність використання програмної системи. Інтуїтивно зрозумілий інтерфейс сприяє швидкому освоєнню програмного забезпечення персоналом та пацієнтами, що зменшує кількість помилок і підвищує загальну продуктивність [3], [4]. Гнучкість налаштувань дозволяє адаптувати програмну систему медичного призначення до специфічних потреб медичного закладу. Можливість модифікації форм, робочих процесів та інших елементів забезпечує відповідність програмного забезпечення конкретним вимогам практики медичного закладу [3], [4].

Інтеграція з існуючими програмними системами, такими як електронні медичні картки та системи виставлення рахунків, забезпечує безперервний потік інформації та зменшує необхідність дублювання медичних даних. Це сприяє підвищенню точності та ефективності обробки інформації [3], [4].

Наявність підтримки та навчальних ресурсів сприяє швидкому вирішенню проблем та ефективному використанню програмної системи медичним персоналом. Це включає доступ до технічної підтримки, навчальних матеріалів та оновлень програмного забезпечення [3], [4].

Економічна ефективність програмного забезпечення підтримки медичної діяльності оцінюється не лише за початковими витратами, але й за довгостроковою вигодою, яку воно приносить у вигляді підвищення ефективності роботи та задоволеності пацієнтів [3], [4].

Урахування цих аспектів при виборі програмного забезпечення для реєстрації пацієнтів сприяє оптимізації процесів у медичному закладі та покращенню якості обслуговування пацієнтів [3], [4].

Завдяки застосуванню інтелектуальних технологій, автоматичних процесів, систем самостійного планування та захищених каналів комунікації вдається підвищити окупність інвестицій і знизити кількість відмов під час фінансових транзакцій. Перехід на цифрові способи реєстрації забезпечує більш швидкий прийом пацієнтів і допомагає краще розподіляти навантаження між працівниками [3], [4].

Використання програмних інструментів реєстрації сприяє підвищенню точності даних. Практика показує, що застосування таких систем дозволяє скоротити час очікування завдяки автоматичній перевірці форм, що мінімізує ризик помилок і забезпечує своєчасну передачу важливої інформації лікарям. Високий рівень достовірності даних важливий як для налагодження зв'язку з пацієнтами, так і для підтримки ефективних маркетингових кампаній у сфері медичних послуг [4], [5].

Автоматизовані рішення для реєстрації допомагають уникати неточностей, пов'язаних із ручним введенням, та гарантують отримання лікарями повного обсягу необхідної інформації. Відмова від паперових носіїв дозволяє швидше обробляти дані та полегшує доступ пацієнтів до власних записів [4], [5].

Численні труднощі під час введення або перевірки даних, що нерідко призводять до затримок, можна звести нанівець завдяки впровадженню інструментів автоматизації. Такі рішення, як платформи для автоматичного заповнення форм, зменшують кількість помилок, виключаючи потребу повторюваного введення одних і тих самих відомостей [4], [5].

Коректно зібрана інформація про пацієнтів сприяє кращому контролю за лікуванням і перевіркою страхових документів, що, своєю чергою, дозволяє зменшити кількість відмов під час виплат. Вивільнення часу адміністративного персоналу шляхом усунення рутинних завдань дозволяє

медичним працівникам приділити більше уваги безпосередньому догляду, що позитивно впливає на досвід пацієнтів [4], [5].

Пацієнти віддають перевагу сервісам, які забезпечують простоту та швидкість обслуговування. Наявність можливостей самостійної реєстрації сприяє та полегшує процес підготовки до візиту. Це покращує організацію роботи приймальні, скорочує час очікування та підвищує рівень задоволеності послугами [4], [5].

Сучасний підхід до реєстрації передбачає відмову від паперових форм на користь цифрових рішень, що дозволяє пацієнтам заздалегідь передавати свої дані, зменшуючи навантаження під час відвідування закладу. Це підвищує довіру до сервісу та формує позитивне ставлення до клініки [4], [5].

Впровадження програмних систем самостійного введення даних скорочує час оформлення формулярів і стимулює активність пацієнтів, створюючи умови для більш ефективного обслуговування. Цифрові рішення усувають потребу в громіздких документах, пропонуючи більш зручні електронні форми, автоматичне заповнення та прості засоби планування записів [4], [5].

Програмні рішення, що дозволяють заздалегідь надавати пацієнтам необхідні форми через електронні повідомлення, покращують їхній досвід і підвищують імовірність повернення до клініки. Щоб забезпечити успішну інтеграцію автоматизованих систем, важливо вибирати такі платформи, які легко впроваджуються персоналом і не створюють зайвих труднощів під час роботи. Зручні для користувача інтерфейси дають змогу сфокусувати увагу на підвищенні якості обслуговування [4], [5].

Серед функцій, що підвищують ефективність цифрових рішень, можна відзначити автоматичне надсилання нагадувань, відповідність стандартам захисту даних, багатомовну підтримку, а також можливість оновлення інформації в реальному часі для медичних працівників [4], [5].

Інтеграція таких систем з іншими платформами дозволяє уникати дублювання даних і зменшує кількість ручної роботи, підвищуючи точність і

знижуючи час, витрачений на адміністративні завдання [5], [6]. Додаткові можливості, як-от налаштування швидких команд для введення стандартних фраз та обмін ними між співробітниками, покращують комунікацію між різними підрозділами. Безпечна передача даних і дотримання нормативних вимог гарантують збереження конфіденційності пацієнтів [5], [6].

Використання мобільних каналів зв'язку, зокрема текстових повідомлень, відкриває додаткові можливості для спрощення реєстрації. Запрошення до заповнення форм, відправлені завчасно, а також безпечне завантаження документів значно прискорюють підготовку до прийому, скорочуючи час, витрачений на паперові процедури [5], [6].

У процесі впровадження програмних рішень для реєстрації пацієнтів важливо зосередитися на оптимізації ресурсів та підвищенні ефективності. Замість витрат на повторювані завдання, варто спрямувати зусилля на вдосконалення обслуговування та зменшення адміністративного навантаження [5], [6].

Забезпечення конфіденційності та безпеки даних пацієнтів є ключовим аспектом при виборі програмного забезпечення. Використання платформ, які відповідають вимогам, гарантує захист особистої інформації та зміцнює довіру пацієнтів [5], [6].

Визначено, що програмне забезпечення для обліку пацієнтів виконує функцію централізованого сховища даних, дозволяючи швидко створювати нові записи, оновлювати інформацію, відстежувати історію лікування та формувати звіти. Такий підхід полегшує процес призначення прийомів, поліпшує управління електронною медичною документацією та забезпечує більш точне ведення обліку. Такі програмні рішення допомагають медичним установам підтримувати відповідність нормативним вимогам і знижують ризик неточностей, що можуть виникати під час ручного введення даних. Їхнє використання створює умови для більшої прозорості в управлінні медичною інформацією, покращуючи доступ до актуальної інформації про пацієнтів.

1.2 Аналіз програмного забезпечення для обліку пацієнтів у медичному закладі

Проаналізуємо програмне забезпечення для обліку пацієнтів у медичному закладі [7]-[13].

Програмне забезпечення Luma Health (рис. 1.1) для обліку пацієнтів у медичних закладах являє собою сучасну платформу, створену для підвищення ефективності взаємодії між медичними працівниками та пацієнтами. Ця система покликана полегшити доступ до медичної допомоги, зробити процеси управління більш зрозумілими та зручними, а також забезпечити безперервну комунікацію на кожному етапі обслуговування. Платформа, що здобула довіру сотень медичних установ, сприяє покращенню якості медичних послуг завдяки впровадженню інноваційних рішень [7], [8], [11].

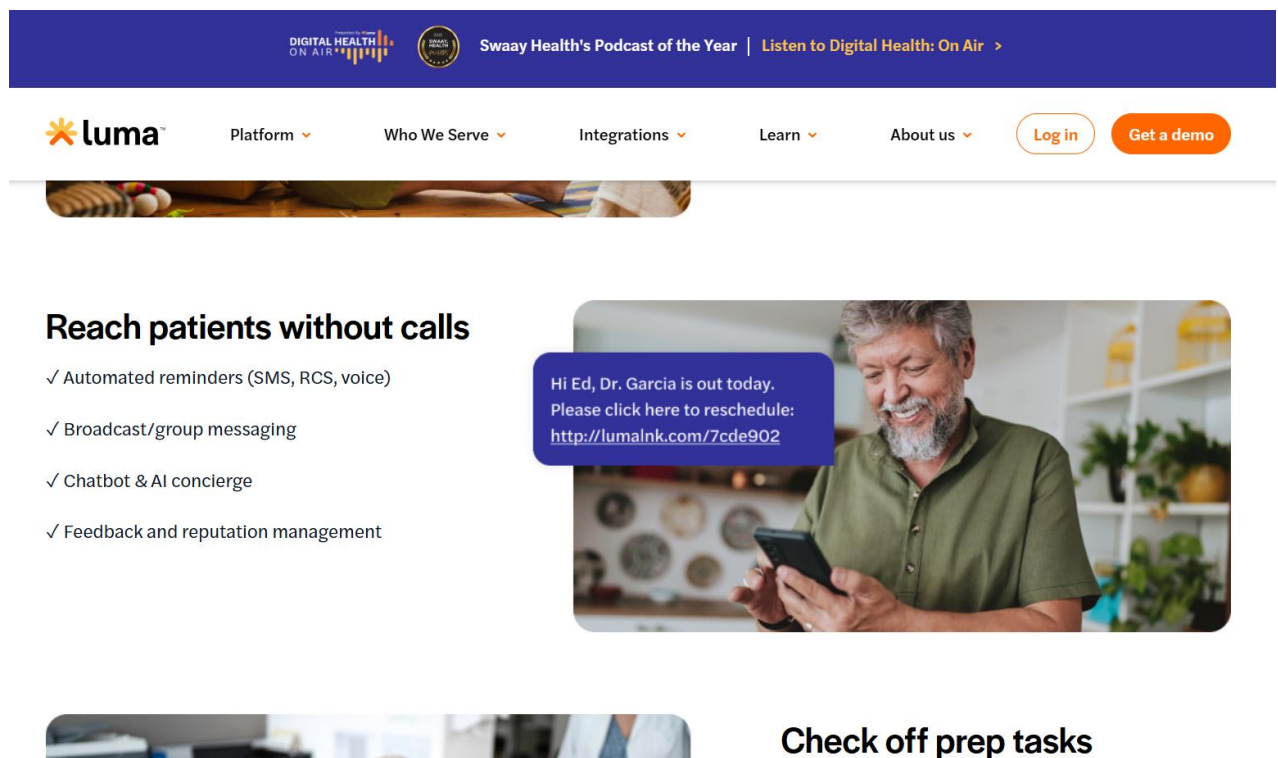


Рисунок 1.1 – Програмне забезпечення Luma Health [11]

Програмне забезпечення Luma Health охоплює всі ключові процеси взаємодії, починаючи від первинного збору даних і закінчуючи підтримкою

після надання медичних послуг. Програмне забезпечення дозволяє пацієнтам самостійно записуватися на прийом, що значно спрощує доступ до послуг лікарів. Завдяки автоматичним нагадуванням про заплановані візити зменшується ймовірність пропущених прийомів, а отже, підвищується ефективність роботи закладу. Функціонал управління направленнями забезпечує оперативну обробку запитів та пришвидшує процес направлення пацієнтів до відповідних спеціалістів [7], [8], [11].

Додатково програмне забезпечення Luma Health підтримує реалізацію інформаційних кампаній, що допомагає залучати пацієнтів до заходів профілактики та підвищувати їхню обізнаність. Інструменти збору відгуків дозволяють медичним установам отримувати важливу інформацію для вдосконалення якості послуг. Інтеграція можливостей телемедицини відкриває доступ до дистанційних консультацій, що особливо актуально для пацієнтів із обмеженим доступом до традиційних медичних послуг. Гнучкість системи дозволяє налаштовувати окремі модулі відповідно до специфічних потреб конкретного закладу [7], [8], [11].

Ціноутворення платформи Luma Health визначається індивідуально для кожної організації, враховуючи її потреби та масштаби діяльності. Для отримання детальної інформації щодо вартості послуг рекомендується безпосередньо звертатися до постачальника [7], [8], [11].

Використання Luma Health забезпечує комплексний підхід до управління взаємодією з пацієнтами, дозволяючи адаптувати систему до конкретних завдань і інтегрувати її з іншими медичними системами. Разом із широким функціоналом і перевіреною ефективністю платформи окремі етапи впровадження можуть потребувати детальнішого аналізу процесів, що важливо враховувати під час інтеграції у роботу закладу [7], [8], [11].

Основними характеристиками програмного забезпечення Luma Health є такі [7], [8], [11]:

- комплексність рішення: програмне забезпечення Luma Health охоплює всі етапи взаємодії з пацієнтами, від запису на прийом до збору відгуків [7], [8], [11];

- автоматизація процесів: програмне забезпечення Luma Health надає автоматичні нагадування, управління направленнями та організацію інформаційних кампаній [7], [8], [11];

- інтеграція з іншими системами: програмне забезпечення Luma Health сумісне з електронними медичними записами та іншими платформами для закладів охорони здоров'я [7], [8], [11];

- можливість налаштування: програмне забезпечення Luma Health дозволяє адаптувати функціонал під специфічні потреби кожної медичної організації [7], [8], [11];

- підтримка телемедицини: програмне забезпечення Luma Health підтримує інтеграцію функцій віртуальної допомоги для зручного дистанційного обслуговування пацієнтів [7], [8], [11];

- орієнтація на пацієнта: програмне забезпечення Luma Health спрямоване на покращення доступу до медичних послуг, підвищення комунікації та задоволеності пацієнтів [7], [8], [11];

- індивідуальне ціноутворення: вартість програмного забезпечення Luma Health визначається окремо для кожної організації відповідно до її потреб [7], [8], [11].

Серед недоліків програмного забезпечення Luma Health можна відзначити потребу в ретельному налаштуванні під конкретні робочі процеси медичного закладу, що може вимагати додаткового часу та ресурсів на етапі впровадження. Також певні функції програмної платформи можуть потребувати навчання персоналу для ефективного використання, а адаптація до існуючих систем обліку та електронних медичних записів інколи супроводжується технічними труднощами, особливо у випадках з нестандартною інфраструктурою закладу [7], [8], [11].

Програмне забезпечення Phreesia (рис. 1.2) представляє собою інтегровану платформу, що призначена для вдосконалення процесу реєстрації та супроводу пацієнтів у медичних закладах. Ця система забезпечує автоматизацію основних процедур, починаючи від моменту запису на прийом і закінчуючи отриманням відгуків після відвідування, що сприяє підвищенню ефективності обслуговування. Особливістю цього рішення є акцент на активній участі пацієнтів у власному лікуванні, завдяки чому пацієнти отримують необхідні інструменти для більш усвідомленого управління своїм здоров'ям [7], [8], [10], [12].

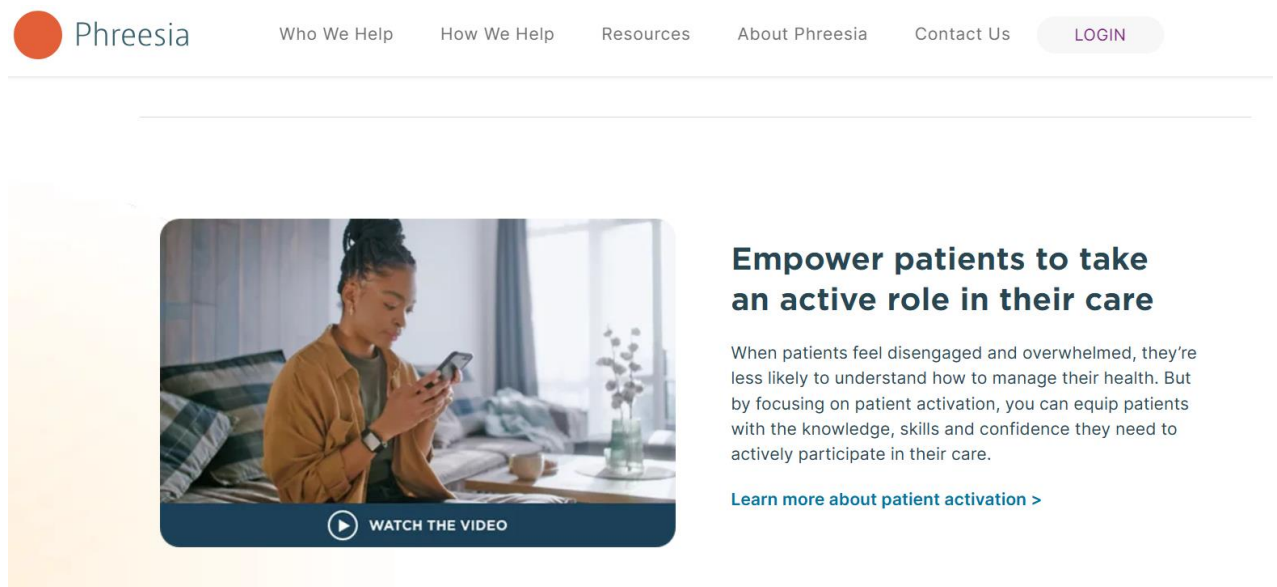


Рисунок 1.2 – Програмне забезпечення Phreesia [12]

Програмне забезпечення Phreesia дозволяє спростити процедуру реєстрації шляхом заповнення індивідуальних анкет до візиту, що значно полегшує прийом. Phreesia підтримує автоматичне планування прийомів і надсилення нагадувань, допомагаючи зменшити кількість пропущених візитів. Крім того, програмне забезпечення виконує перевірку страхових полісів у автоматичному режимі, що підвищує точність даних і скорочує витрати часу на адміністративні завдання [7], [8], [10], [12].

Програмне забезпечення Phreesia також інтегрує можливості для збору платежів під час надання послуг, що оптимізує фінансові процеси й сприяє

кращому контролю доходів медичної установи. Програмне забезпечення Phreesia передбачає збирання відгуків пацієнтів після їхніх відвідувань, що дозволяє медичному закладу отримувати цінну інформацію для підвищення якості обслуговування [7], [8], [10], [12].

Окремою перевагою програмного забезпечення Phreesia є можливість адаптації робочих процесів під потреби різних спеціалізацій, що забезпечує гнучкість використання у таких сферах, як кардіологія, дерматологія, гастроентерологія, неврологія, педіатрія та інших напрямках медицини. Платформа також підтримує інтеграцію з електронними медичними картками та системами управління пацієнтами, що забезпечує безперервний обмін даними й знижує потребу у ручному введенні [7], [8], [10], [12].

Вартість Програмне забезпечення Phreesia визначається індивідуально для кожної медичної установи залежно від її потреб і масштабів діяльності, а детальна інформація про цінову політику надається за запитом [7], [8], [12].

Програмне забезпечення для обліку пацієнтів Phreesia має такі особливості [7], [8], [10], [12]:

- автоматизований прийом пацієнтів: програмне забезпечення Phreesia підтримує швидкий та індивідуалізований процес реєстрації за допомогою онлайн анкет до візиту [7], [8], [10], [12];

- планування та нагадування: програмне забезпечення Phreesia дозволяє пацієнтам легко записуватись на прийом і отримувати автоматичні нагадування для зменшення кількості неявок [7], [8], [10], [12];

- перевірка страхових даних: програмне забезпечення Phreesia автоматизує процес перевірки страхового покриття, підвищуючи точність та скорочуючи час обробки [7], [8], [10], [12];

- обробка платежів: програмне забезпечення Phreesia надає можливість оплачувати послуги під час надання допомоги, що покращує фінансовий цикл організації [7], [8], [10], [12];

- збір відгуків пацієнтів: програмне забезпечення Phreesia збирає інформацію про досвід пацієнтів після візиту для підвищення якості обслуговування [7], [8], [10], [12];

- налаштовувані робочі процеси: програмне забезпечення Phreesia адаптує функціонал платформи під потреби різних медичних спеціальностей, враховуючи специфіку кожної галузі [7], [8], [10], [12];

- інтеграція з медичними системами: програмне забезпечення Phreesia забезпечує безперервний обмін даними з електронними медичними записами та системами управління пацієнтами [7], [8], [10], [12].

Серед недоліків програмного забезпечення Phreesia можна відзначити обмежену гнучкість у надсиланні індивідуальних повідомлень пацієнтам, які не заплановані на конкретний день, що створює труднощі у вибірковій комунікації та потребує додаткових рішень для ефективного інформування окремих груп пацієнтів [7], [8], [10], [12].

Програмне забезпечення Doctible (рис. 1.3) призначене для оптимізації взаємодії між медичними закладами та пацієнтами, забезпечуючи ефективне управління прийомами, комунікацією та маркетингом [7], [13].

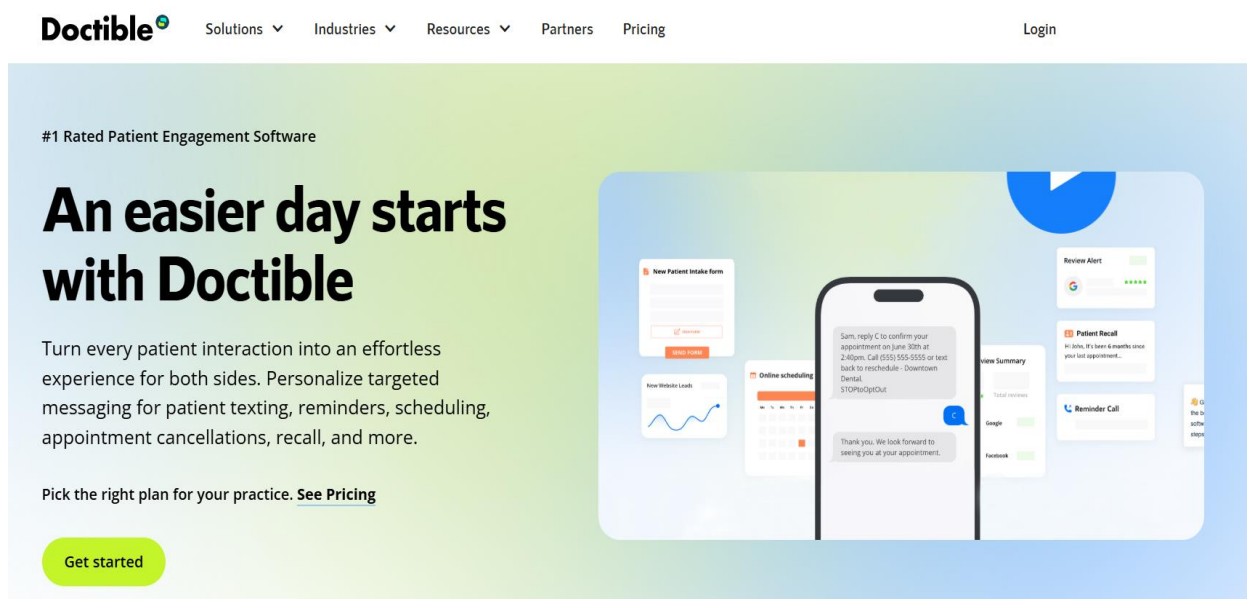


Рисунок 1.3 – Програмне забезпечення Doctible [13]

Програмне забезпечення Doctible підтримує медичні, стоматологічні, оптометричні та інші практики, надаючи інструменти для автоматизації процесів та покращення досвіду пацієнтів [7], [13].

Серед основних можливостей програмного забезпечення Doctible можна відзначити автоматизовані нагадування про прийоми через текстові повідомлення, голосові дзвінки та електронну пошту, що зменшує кількість пропущених візитів. Функція Smart Messaging дозволяє надсилати персоналізовані повідомлення на основі даних електронної медичної картки, включаючи інформацію про лікування, процедури чи кабінети, що підвищує ефективність комунікації [7], [13].

Програмне забезпечення Doctible також включає інструменти для онлайн запису на прийом, цифрові форми для збору інформації та згоди пацієнтів, а також систему EasyFill для автоматичного заповнення скасованих зустрічей за допомогою цифрового списку очікування. Інтеграція з електронними медичними картками та програмними системами управління практикою забезпечує безперервний потік даних та зменшує потребу в ручному введенні інформації [7], [13].

Для покращення взаємодії з пацієнтами програмне забезпечення Doctible пропонує інструменти для управління репутацією, включаючи автоматичний збір відгуків та аналіз задоволеності пацієнтів, що дозволяє швидко виявляти області для покращення. Крім того, платформа підтримує двосторонню текстову комунікацію з пацієнтами через стаціонарний номер телефону, що спрощує процес спілкування та забезпечує відповідність стандартам. Завдяки широкому спектру функцій та можливостей налаштування, програмне забезпечення Doctible дозволяє медичним закладам підвищити ефективність роботи, покращити досвід пацієнтів та сприяти зростанню практики [7], [13].

Програмне забезпечення Doctible має такі особливості [7], [13]:

- цифрові форми: забезпечують можливість заповнення анкет та документів онлайн, спрощуючи процес реєстрації [7], [13];

- розумний планувальник: програмне забезпечення Doctible надає пацієнтам зручний спосіб самостійно записатися на прийом через вебсайт клініки [7], [13];
- повторне записування пацієнта: програмне забезпечення Doctible автоматично нагадує про необхідність повторного візиту, що сприяє підвищенню рівня утримання пацієнтів [7], [13];
- easyFill: програмне забезпечення Doctible дозволяє заповнювати скасовані записи на прийом завдяки цифровому списку очікування, що активується автоматично [7], [13];
- інтеграція з електронною медичною картою: програмне забезпечення Doctible підтримує взаємодію з існуючими програмними системами для оптимізації внутрішніх процесів [7], [13];
- маркетингові інструменти: програмне забезпечення Doctible надає інструменти, які допомагають покращити онлайн присутність медичного закладу та залучити нових пацієнтів завдяки управлінню репутацією та email-кампаніям [7], [13];
- гнучкі тарифні плани: пропонуються різні варіанти підписки на програмне забезпечення Doctible відповідно до потреб медичного закладу, від базових функцій до розширених можливостей з персоналізованими програмними рішеннями [7], [13].

Серед недоліків програмного забезпечення Doctible можна відзначити надмірну кількість автоматичних повідомлень, що іноді викликає незадоволення пацієнтів через перевантаження інформацією, а також затримки в оновленні даних на вебсайті, що може призводити до надсилання застарілих нагадувань, якщо зміни в розкладі роботи закладу відбуваються в останню хвилину [7], [13].

Порівняльну характеристику програмного забезпечення для обліку пацієнтів у медичних закладах наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння програмного забезпечення для обліку пацієнтів у медичних закладах

Критерій порівняння	Luma Health [11]	Phreesia [12]	Doctible [13]
Простота використання	+	+—	+
Маркетингові інструменти	+—	—	+
Зручність автоматизованих нагадувань	+—	+	+—
Інтеграція з електронною медичною картою	+	+—	+
Налаштовування під спеціалізацію	+—	+	+
Підтримка пацієнтів через кілька каналів	+	+—	+—

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних засобів для обліку пацієнтів у медичних закладах. Проте деякі програмні засоби для обліку пацієнтів у медичних закладах мають високу вартість придбання, впровадження та обслуговування, що може бути непосильним для невеликих медичних закладів. Крім того, складність налаштування та інтеграції складного медичного програмного забезпечення може призводити до затримок у роботі та вимагати додаткових ресурсів. Крім того, надмірна стандартизація процесів у деяких програмних системах для обліку роботи медичних закладів може обмежувати індивідуальний підхід до пацієнтів і створювати додаткові адміністративні навантаження на лікарів. Тому актуальною є розробка програмного забезпечення для обліку пацієнтів у медичному закладі.

При розробці програмного забезпечення для обліку пацієнтів у медичному закладі необхідно забезпечити такі функціональні вимоги:

- підтримка можливості пошуку лікаря за прізвищем та спеціалізацією роботи у медичному закладі;
- можливість реєстрації та авторизації пацієнтів медичного закладу;
- можливість відображення інформації стосовно візитів пацієнта;
- підтримка можливості додавання інформації про дітей пацієнтів медичного закладу;
- підтримка можливості ознайомлення з розкладом роботи лікаря та запису на вільний час;
- підтримка можливості роботи з профілем користувача.

1.3 Висновки за розділом 1

Визначено, що програмне забезпечення для обліку пацієнтів виконує функцію централізованого сховища даних, дозволяючи швидко створювати нові записи, оновлювати інформацію, відстежувати історію лікування та формувати звіти. Такий підхід полегшує процес призначення прийомів, поліпшує управління електронною медичною документацією та забезпечує більш точне ведення обліку. Такі програмні рішення допомагають медичним установам підтримувати відповідність нормативним вимогам і знижують ризик неточностей, що можуть виникати під час ручного введення даних. Їхнє використання створює умови для більшої прозорості в управлінні медичною інформацією, покращуючи доступ до актуальної інформації про пацієнтів.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для підтримки роботи медичних закладів. Проте деякі програмні засоби для обліку пацієнтів у медичних закладах мають високу вартість придбання, впровадження та обслуговування, що може бути непосильним для невеликих медичних закладів. Крім того, складність налаштування та інтеграції складного медичного програмного забезпечення може призводити до затримок у роботі та вимагати додаткових ресурсів. Крім того, надмірна стандартизація процесів у деяких програмних

системах для обліку роботи медичних закладів може обмежувати індивідуальний підхід до пацієнтів і створювати додаткові адміністративні навантаження на лікарів. Тому актуальною є розробка програмного забезпечення для обліку пацієнтів у медичному закладі.

Сформульовано функціональні вимоги до програмного забезпечення для обліку пацієнтів у медичному закладі.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці програмного забезпечення для обліку пацієнтів у медичному закладі було використано мову програмування PHP [14], [15].

Мова програмування PHP використовується для створення серверних програмних рішень, здатних обробляти запити користувачів, управляти базами даних та формувати відповіді для вебінтерфейсів. Завдяки своїй простоті, гнучкості й високій сумісності з різними платформами, вона добре підходить для розробки програмного забезпечення для обліку пацієнтів у медичних закладах. Підтримка великої кількості операційних систем і серверів забезпечує зручність впровадження та масштабування рішень, що базуються на мові PHP [14], [15].

PHP дозволяє організувати ефективну роботу з базами даних різних типів, що є ключовим фактором для систем збереження медичних даних. Об'єктно-орієнтовані можливості мови програмування PHP сприяють створенню структурованих, гнучких і підтримуваних рішень, що відповідають сучасним вимогам до програмних продуктів. Оптимізація обробки даних і пам'яті у нових версіях мови програмування PHP підвищує її продуктивність і надійність [14], [15].

Розробка на мові програмування PHP активно підтримується широким вибором фреймворків, серед яких виділяються Laravel, Symfony, Yii та CodeIgniter. Завдяки цим фреймворкам процес побудови складних інформаційних систем, у тому числі медичних, стає більш структурованим, швидким та безпечним. Інтеграція мови програмування PHP із зовнішніми сервісами через інтерфейси прикладного програмування дає змогу легко з'єднувати програмне забезпечення обліку пацієнтів з іншими медичними або адміністративними системами [14], [15].

Мова програмування PHP залишається актуальним інструментом для розробки завдяки своїй універсальності, активній підтримці спільноти та

здатності відповідати вимогам різних галузей, включно з медициною. Його використання забезпечує стабільну основу для побудови надійних, безпечних та функціональних програмних продуктів для обліку пацієнтів у лікувальних установах [14], [15].

Для розробки програмного забезпечення обліку пацієнтів у медичних установах ефективним рішенням стає використання мови PHP. Ця мова програмування надає широкі можливості для створення веборієнтованих систем, що ідеально підходить для середовища, де важливо забезпечити доступ як з локальної мережі, так і через Інтернет. Системи реєстрації пацієнтів, побудовані на PHP, можуть легко працювати з популярними базами даних, такими як MySQL, що дає змогу організувати надійне зберігання медичних записів [14], [15].

Потенціал мови програмування PHP полягає у спрощенні процесу розробки завдяки наявності великої кількості готових рішень, бібліотек і фреймворків, що підтримують високу гнучкість налаштувань. Використання таких інструментів, як Laravel чи Symfony, сприяє швидкій інтеграції додаткових функцій, що особливо важливо для систем, де безпека персональних даних та стабільність доступу до інформації є першочерговими вимогами [14], [15].

Активна спільнота розробників, що підтримує мову PHP, сприяє постійному вдосконаленню знань та оперативному вирішенню можливих труднощів у процесі створення програмного забезпечення. Гнучкість цієї мови дозволяє інтегрувати рішення обліку пацієнтів із мобільними додатками, сервісами електронних платежів та іншими елементами сучасної інфраструктури охорони здоров'я [14], [15].

Таким чином, мова програмування PHP відкриває широкі можливості розробникам для створення стабільних, масштабованих і захищених програмних систем обліку пацієнтів, які легко адаптуються до змінних вимог медичних закладів [14], [15].

Обґрунтування вибору мови програмування для розробки програмного забезпечення для обліку пацієнтів у медичному закладі наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки програмного забезпечення для обліку пацієнтів у медичному закладі

Критерій порівняння мов програмування	Мова програмування		
	PHP	Python	Java
Простота розробки вебзастосунків	+	+—	+—
Швидкість розробки	+	+—	—
Готові фреймворки для веброзробки	+	+	+—
Зручність для створення ПЗ для обліку пацієнтів у медичному закладі	+	—	—
Сумісність з більшістю серверів	+	+—	+—
Невисокі вимоги до ресурсів сервера	+	+—	+—

Отже, для реалізації програмного забезпечення для обліку пацієнтів у медичному закладі обрано мову програмування PHP, яка є однією з найпоширеніших мов для створення веборієнтованих систем. Крім того, завдяки своїй високій сумісності з різноманітними системами управління базами даних мова PHP дозволяє ефективно організовувати зберігання і обробку великих обсягів медичної інформації. Іншою важливою перевагою мови програмування PHP є її доступність і відкритість, що суттєво знижує вартість розробки програмної системи порівняно з використанням комерційних технологій.

2.2 Вибір фреймворку для створення програмного забезпечення для обліку пацієнтів у медичному закладі

В якості фреймворку для створення програмного забезпечення для обліку пацієнтів у медичному закладі було обрано Laravel [16], [17].

Фреймворк Laravel вирізняється серед інструментів PHP завдяки своїй гнучкості та потужності, що робить його особливо придатним для створення систем обліку пацієнтів у медичних установах. Його архітектурні рішення сприяють чіткій організації коду, завдяки чому спрощується керування великими проектами та підтримка їх у подальшому. Використання підходу MVC забезпечує ясну розподіленість обов'язків у коді, що значно полегшує розробку складних систем [16], [17].

Функціональні можливості Laravel дозволяють створювати динамічні вебсервіси з високим ступенем інтеграції. Засоби роботи з базами даних через ORM Eloquent дають змогу ефективно оперувати записами пацієнтів, мінімізуючи обсяг коду та підвищуючи надійність обробки даних. Інструмент шаблонізації Blade полегшує побудову інтерфейсів, що забезпечує швидке створення зручних та доступних користувацьких рішень [16], [17].

Особливу увагу в Laravel приділено підтримці безпеки, що має вирішальне значення при обробці конфіденційної інформації, такої як особисті дані пацієнтів. Передбачено готові механізми для автентифікації та авторизації, що підвищує рівень захищеності додатків. Вбудовані засоби для побудови API відкривають можливості інтеграції з іншими системами або мобільними застосунками, що є важливим аспектом у розвитку цифрових медичних сервісів [16], [17].

Системи тестування та керування залежностями, що підтримуються у Laravel, допомагають забезпечувати стабільність і якість продукту на кожному етапі створення. Робоче середовище, яке може бути налаштоване за допомогою інструментів типу Sail, спрощує розгортання та обслуговування застосунків у різних середовищах. Laravel залишається одним із найбільш

адаптивних фреймворків для сучасної розробки, підтримуючи високі вимоги до якості, безпеки та ефективності медичних інформаційних систем [16], [17].

Laravel можна ефективно використовувати для створення програмного забезпечення, призначеного для обліку пацієнтів у медичних закладах. Завдяки своїм характеристикам цей фреймворк забезпечує зручне середовище для розробки вебрішень, де особливу увагу приділено швидкості створення, простоті підтримки та подальшій масштабованості застосунків [16], [17].

Структурована архітектура на основі концепції Model-View-Controller сприяє логічному впорядкуванню коду, що особливо важливо у сфері охорони здоров'я, де часто доводиться працювати з великими обсягами змінних даних пацієнтів. Чіткість побудови систем дозволяє оперативно реагувати на нові вимоги, забезпечуючи безперервну доступність інформації [16], [17].

Особливе місце у фреймворку Laravel займають механізми безпеки, які вбудовано для протидії типовим загрозам вебзастосунків. Інструменти для захисту даних користувачів дають змогу гарантувати конфіденційність облікових записів пацієнтів. Системи автентифікації та авторизації у Laravel впроваджуються без додаткових ускладнень, що дозволяє будувати середовища із контрольованим доступом до чутливої інформації [16], [17].

Гнучкі можливості інтеграції з зовнішніми сервісами, обробка подій у фоновому режимі, системи кешування та планування завдань значно полегшують обробку великих обсягів даних і забезпечують високу продуктивність навіть у складних сценаріях використання. Організоване управління потоками інформації відіграє ключову роль у підтримці стабільної роботи медичних систем [16], [17].

Крім того, Laravel вирізняється доступністю супровідної документації та широкими можливостями для навчання, що пришвидшує процес розробки та підтримки системи обліку. Інструменти для автоматизації процесів, побудови API, відправки повідомлень або взаємодії з платіжними платформами роблять його універсальним вибором для розширення функціоналу [16], [17].

Завдяки своїй продуманості, сучасним рішенням і сильній підтримці спільноти Laravel забезпечує надійний фундамент для створення інноваційного та безпечного програмного забезпечення у сфері обліку пацієнтів [16], [17].

Обґрунтування вибору фреймворка для створення програмного забезпечення для обліку пацієнтів у медичному закладі наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору фреймворка для створення програмного забезпечення для обліку пацієнтів у медичному закладі

Критерій порівняння фреймворків	Фреймворки		
	Laravel	Symfony	CodeIgniter
Легкість інтеграції зовнішніх сервісів	+	+—	+—
Активність і підтримка спільноти	+	+	+—
Розвинена документація	+	+	+—
Простота та швидкість розробки	+	+—	+
Готові рішення для аутентифікації та авторизації	+	+—	—

Таким чином, для створення програмного забезпечення для обліку пацієнтів у медичному закладі обрано фреймворк Laravel завдяки його потужним можливостям, які дозволяють швидко створювати надійні, масштабовані та безпечні вебзастосунки. Крім того, Laravel також має високу продуктивність і масштабованість завдяки можливості легкої інтеграції з кешуванням, чергами задач і асинхронною обробкою подій, що дозволяє забезпечити стабільну роботу навіть при великій кількості користувачів і

запитів. Наявність розвинутого механізму міграцій баз даних значно полегшує керування структурою даних, що особливо актуально для проєктів, де інформація постійно оновлюється та доповнюється.

2.3 Висновки за розділом 2

Для реалізації програмного забезпечення для обліку пацієнтів у медичному закладі обрано мову програмування PHP, яка є однією з найпоширеніших мов для створення веборієнтованих систем. Крім того, завдяки своїй високій сумісності з різноманітними системами управління базами даних мова PHP дозволяє ефективно організовувати зберігання і обробку великих обсягів медичної інформації. Іншою важливою перевагою мови програмування PHP є її доступність і відкритість, що суттєво знижує вартість розробки програмної системи порівняно з використанням комерційних технологій.

Для створення програмного забезпечення для обліку пацієнтів у медичному закладі обрано фреймворк Laravel завдяки його потужним можливостям, які дозволяють швидко створювати надійні, масштабовані та безпечні вебзастосунки. Крім того, Laravel також має високу продуктивність і масштабованість завдяки можливості легкої інтеграції з кешуванням, чергами задач і асинхронною обробкою подій, що дозволяє забезпечити стабільну роботу навіть при великій кількості користувачів і запитів.

3 ОПИС ПРОГРАМИ

3.1 Структура програмного забезпечення для обліку пацієнтів у медичному закладі

Структуру програмного забезпечення для обліку пацієнтів у медичному закладі наведено на рис. 3.1.

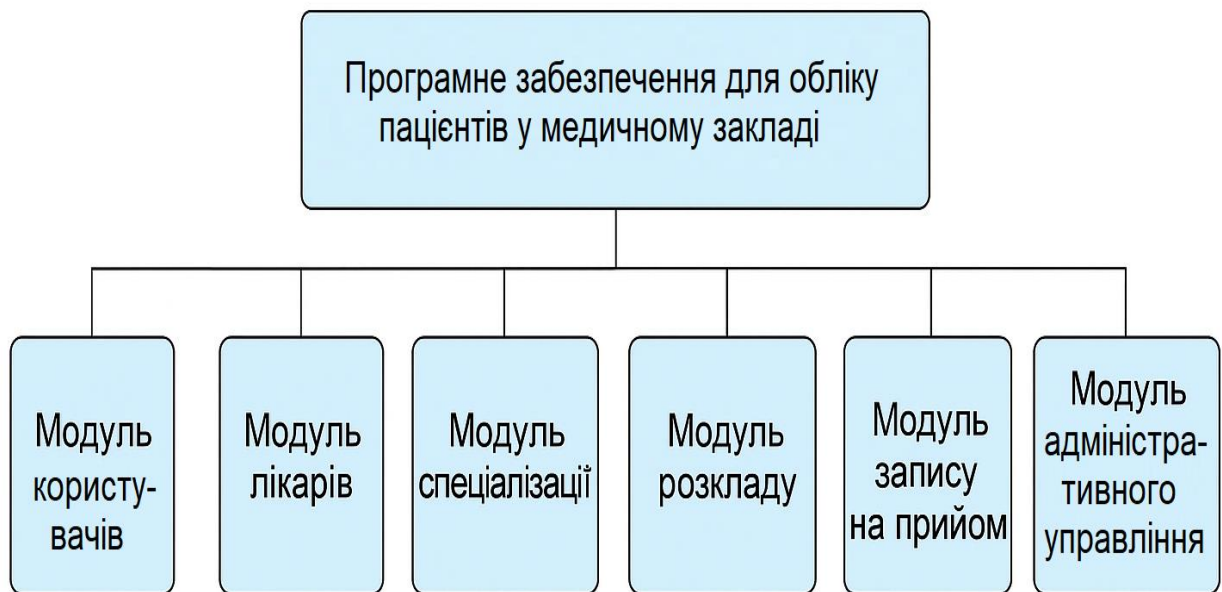


Рисунок 3.1 – Структура програмного забезпечення для обліку пацієнтів у медичному закладі

Структура програмного забезпечення для обліку пацієнтів у медичному закладі передбачає наявність шести взаємопов'язаних модулів, кожен з яких відповідає за окремий аспект функціонування системи.

Модуль користувачів виконує роль основного інтерфейсу для взаємодії з пацієнтами. Він забезпечує реєстрацію, авторизацію, зберігання персональних даних, а також управління обліковими записами як дорослих користувачів, так і їхніх дітей. У межах цього модуля обробляються запити щодо оновлення інформації та взаємодії з іншими функціональними частинами системи.

Модуль лікарів призначений для обліку медичних працівників. У ньому підтримується інформація про персонал, включаючи повне ім'я, спеціалізацію, контактні дані та інші професійні характеристики. Цей модуль також тісно пов'язаний із модулями розкладу й спеціалізацій.

Модуль спеціалізацій відповідає за класифікацію медичних напрямків. Він надає змогу пов'язати кожного лікаря з відповідною спеціальністю, забезпечуючи структурованість при формуванні розкладів і записів на прийом. Це дозволяє здійснювати швидкий пошук фахівців за певним профілем.

Модуль розкладу визначає графік роботи медичних працівників. Його функціональність полягає у формуванні часу прийомів, узгодженні з доступністю лікарів та керуванні часовими проміжками. У межах цього модуля створюється база доступних слотів для пацієнтів, що унеможливорює конфлікти при записі.

Модуль запису на прийом реалізує механізм взаємодії між пацієнтами та лікарями. Він дозволяє обирати зручний час, переглядати вільні місця в розкладі й підтверджувати запис на прийом. Цей компонент забезпечує збереження зв'язку між пацієнтом, лікарем та відповідним часовим інтервалом.

Модуль адміністративного управління виконує контрольні та аналітичні функції. Він дозволяє адміністратору системи керувати доступом, перевіряти активність користувачів, отримувати звітність, а також налаштовувати параметри безпеки. Завдяки цьому забезпечується ефективне функціонування всієї системи в межах заданих політик медичного закладу.

3.2 Функціонування програмного забезпечення для обліку пацієнтів у медичному закладі

Функціонування програмного забезпечення для обліку пацієнтів у медичному закладі подамо за допомогою схеми, зображеної на рис. 3.2.

Зазначимо, що на схемі наведено один з варіантів використання програмного забезпечення для обліку пацієнтів у медичному закладі.

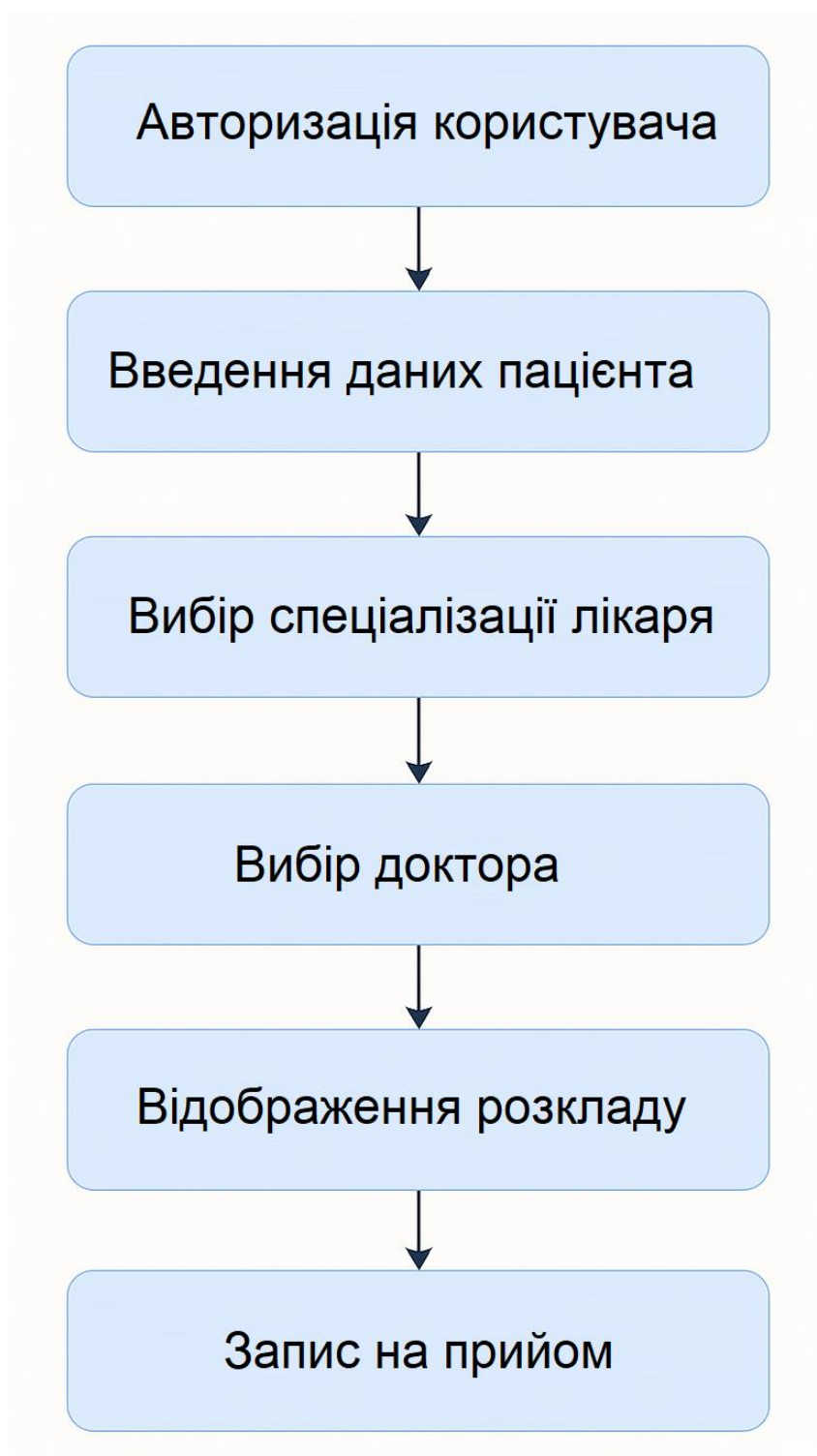


Рисунок 3.2 – Схема функціонування програмного забезпечення для обліку пацієнтів у медичному закладі

Функціонування програмного забезпечення обліку пацієнтів у медичному закладі можна представити як послідовність узгоджених кроків, де кожен етап логічно пов'язаний з наступним. Процес починається з авторизації користувача ПЗ для обліку пацієнтів у медичному закладі, під час якої здійснюється перевірка облікових даних пацієнта або адміністратора. Після успішного входу користувач ПЗ для обліку пацієнтів у медичному закладі має можливість ввести або оновити свої персональні дані, включно з контактною інформацією та, за потреби, відомостями про дітей.

Далі відбувається вибір медичної спеціалізації, за якою система відображає перелік доступних лікарів. Обравши лікаря, користувач отримує доступ до розкладу прийому цього спеціаліста. Після перегляду доступних часових проміжків пацієнт здійснює запис на прийом, що зберігається в базі даних як окрема подія з прив'язкою до конкретного часу та лікаря.

Після завершення запису лікар отримує інформацію про запланований прийом через свій інтерфейс. По завершенні консультації до програмної системи для обліку пацієнтів у медичному закладі може вноситися додаткова інформація про пацієнта або результати обстеження. Адміністративна частина програмного забезпечення для обліку пацієнтів у медичному закладі забезпечує моніторинг усіх подій, створення статистичних звітів, контроль доступу та підтримку користувачів.

3.3 Розробка бази даних програмного забезпечення для обліку пацієнтів у медичному закладі

База даних програмного забезпечення обліку пацієнтів у медичному закладі слугує основою для централізованого зберігання, організації та доступу до інформації, що стосується пацієнтів, лікарів, медичних спеціальностей та записів на прийом. Її призначення полягає у забезпеченні надійного, структурованого та швидкого обміну даними між усіма

компонентами системи, що дозволяє автоматизувати щоденні процеси медичної установи.

Особливістю такої бази даних є її чітка логічна модель, що охоплює основні сутності: користувачів, їхніх дітей (якщо йдеться про педіатричну практику), лікарів, спеціалізації, часові проміжки прийому, розклад і записи. Забезпечується зв'язність усіх елементів, що дозволяє уникнути дублювання інформації та забезпечити цілісність даних.

Схему бази даних програмного забезпечення для обліку пацієнтів у медичному закладі наведено на рис. 3.3.

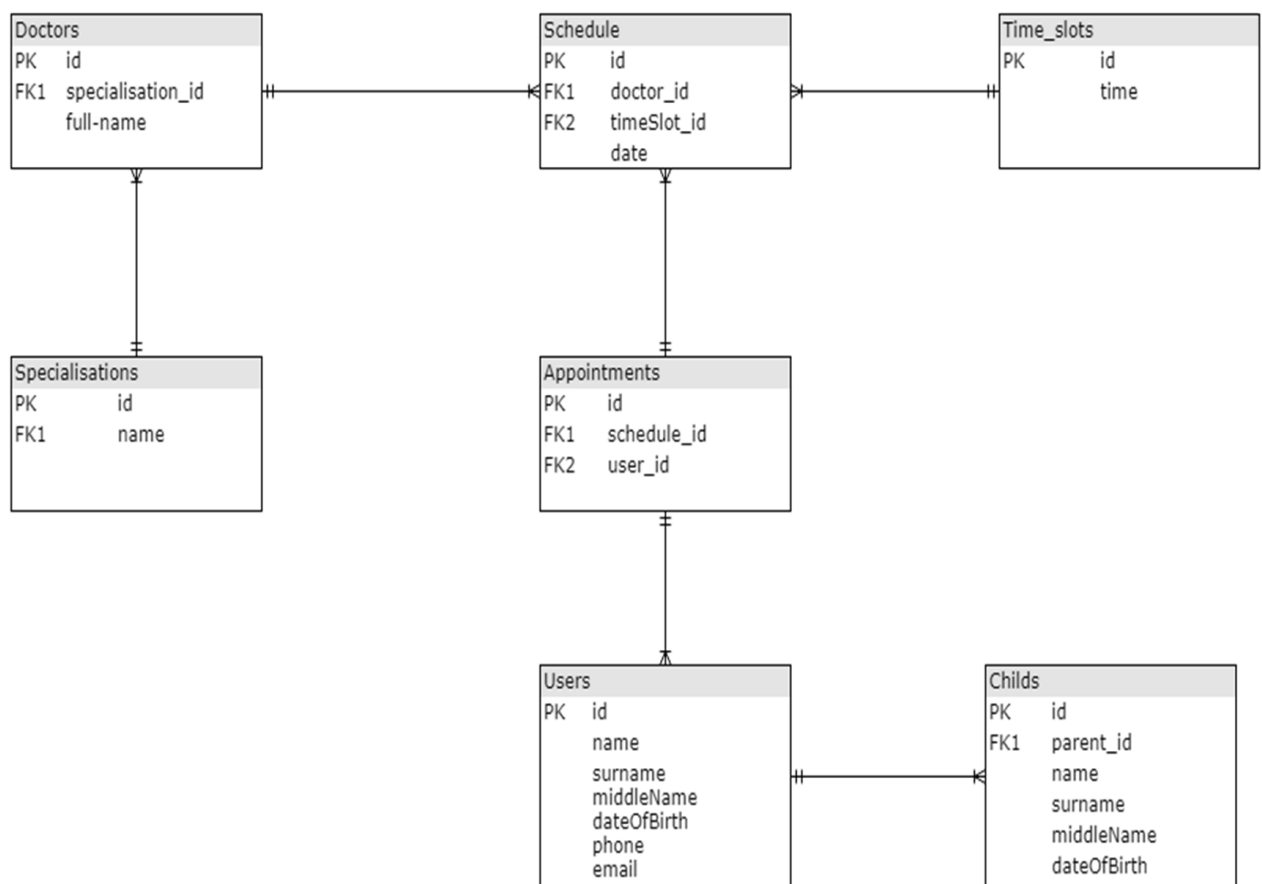


Рисунок 3.3 – Схема бази даних програмного забезпечення для обліку пацієнтів у медичному закладі

У структурі бази даних програмного забезпечення для обліку пацієнтів у медичних установах передбачено взаємозв'язану систему зберігання

відомостей, що забезпечує повноцінне функціонування електронного запису до лікаря та ведення персональних облікових даних.

Передбачається існування окремого сховища для інформації про лікарські спеціалізації, що дозволяє системі групувати медичних працівників за професійним профілем. Для кожного лікаря фіксується повне ім'я та вказується належність до певного напрямку медицини. Відомості про доступний час прийому структуруються у вигляді окремих часових інтервалів, що формують основу для планування графіку.

Дані пацієнтів організовано у вигляді набору полів, які охоплюють основну персональну інформацію, включно з іменем, прізвищем, по батькові, датою народження, контактними номерами та електронною адресою. За потреби система дозволяє вести облік неповнолітніх осіб, які перебувають на піклуванні зареєстрованих користувачів, з відповідною фіксацією імені, прізвища, по батькові та дати народження таких осіб.

Формування розкладу лікарів реалізується через вказівку часових діапазонів, прив'язаних до конкретного спеціаліста. Інформація про кожний прийом містить посилання на особу пацієнта та відповідний розклад, що дає змогу точно зіставляти дані про консультації та забезпечувати ефективне управління записами.

Завдяки логічно організованій структурі всі елементи бази даних взаємодіють між собою таким чином, щоб забезпечити безперебійне функціонування системи медичного обліку, зберігаючи повноту, точність і актуальність кожного внесеного запису.

Крім того, така структура створена з урахуванням медичної специфіки: передбачена можливість обліку кількох категорій пацієнтів, гнучкість у формуванні розкладів лікарів та наявність детальної контактної інформації для зворотного зв'язку. Завдяки цьому база даних слугує не лише сховищем інформації, а й інструментом для управління взаємодією між пацієнтами та персоналом медичного закладу.

3.4 Проектування інтерфейсу програмного забезпечення для обліку пацієнтів у медичному закладі

Проектування інтерфейсу взаємодії користувача з програмним забезпеченням для обліку пацієнтів у медичному закладі виконано з урахуванням вимог технічного завдання. При розробленні програмного забезпечення для обліку пацієнтів у медичному закладі враховані функціональні вимоги до програми:

- підтримка можливості пошуку лікаря за прізвищем та спеціалізацією роботи у медичному закладі;
- можливість реєстрації та авторизації пацієнтів медичного закладу;
- можливість відображення інформації стосовно візитів пацієнта;
- підтримка можливості додавання інформації про дітей пацієнтів медичного закладу;
- підтримка можливості ознайомлення з розкладом роботи лікаря та запису на вільний час;
- підтримка можливості роботи з профілем користувача.

Інтерфейс головної форми програмного забезпечення для обліку пацієнтів у медичному закладі наведено на рис. 3.4.

The screenshot displays a web interface titled "Знайти лікаря" (Find Doctor). At the top right, there are links for "Пошук" (Search), "Увійти" (Login), and "Зареєструватися" (Register). The main content area contains two search forms. The first form, labeled "Прізвище" (Surname), has a text input field with the placeholder "Шукати за прізвищем" (Search by surname) and a blue "Знайти" (Find) button below it. The second form, labeled "Спеціальність" (Specialty), has a text input field with the placeholder "Шукати за спеціальністю" (Search by specialty) and a blue "Знайти" (Find) button below it.

Рисунок 3.4 – Інтерфейс головної форми програми

Інтерфейс форми реєстрації наведено на рис. 3.5, інтерфейс форми авторизації наведено на рис. 3.6.

Пошук Увійти Зареєструватися

РЕЄСТРАЦІЯ

Особиста інформація:

Прізвище *

Ім'я *

По батькові

Дата народження *

Контактна інформація:

Контактний телефон*

Електронна пошта *

Пароль

Підтвердження паролю

[Зареєструватися](#)

Рисунок 3.5 – Форма реєстрації

Пошук Увійти Зареєструватися

УВІЙТИ

Електронна пошта

Пароль

☐ Запам'ятати мене

[Увійти](#) [Забули пароль?](#)

Рисунок 3.6 – Форма авторизації

Вкладку Візити програмного забезпечення для обліку пацієнтів у медичному закладі наведено на рис. 3.7.

Іван Сергєвнїн	Візити Запис для Сергєвнїн Іван Владиславович Лікар: Петренко Володимир Григорович Спеціалізація лікаря: сімейний лікар Дата: 08.05.2025 Час: 08:30
 Візити  Діти  Профіль	

Рисунок 3.7 – Вкладка Візити

Вкладку Профілю користувача програмного забезпечення для обліку пацієнтів у медичному закладі наведено на рис. 3.8.

Іван Сергєвнїн	Профіль Особиста інформація: Прізвище Сергєвнїн Ім'я Іван По батькові Владиславович Дата народження 18.06.2004 Контактна інформація: Контактний телефон +380500778901 Електронна пошта sergevnivld@gmail.com Зберегти зміни
 Візити  Діти  Профіль	

Рисунок 3.8 – Сторінка профілю користувача

Сторінку з розкладом роботи лікаря наведено на рис. 3.9.

Розклад лікаря Петренко Володимир Григорович				
Понеділок	Вівторок	Середа	Четвер	П'ятниця
08:00	08:00	08:00	08:00	08:00
08:15	08:15	08:15	08:15	08:15
08:30	08:30	08:30	08:30	08:30
08:45	08:45	08:45	08:45	08:45
09:00	09:00	09:00	09:00	09:00
09:15	09:15	09:15	09:15	09:15

[Записатися](#)

Рисунок 3.9 – Сторінка з розкладом роботи лікаря

3.5 Висновки за розділом 3

Розроблено програмне забезпечення для обліку пацієнтів у медичному закладі.

Запропоновано структуру програмного забезпечення для обліку пацієнтів у медичному закладі, що складається з шести взаємопов'язаних модулів, кожен з яких відповідає за окремий аспект функціонування системи.

Наведено схему функціонування програмного забезпечення обліку пацієнтів у медичному закладі, на які роботу програми подано як послідовність узгоджених кроків, де кожен етап логічно пов'язаний з наступним.

Розроблено базу даних програмного забезпечення для обліку пацієнтів у медичному закладі, яка слугує основою для централізованого зберігання, організації та доступу до інформації, що стосується пацієнтів, лікарів, медичних спеціальностей та записів на прийом. Її призначення полягає у забезпеченні надійного, структурованого та швидкого обміну даними між усіма компонентами системи, що дозволяє автоматизувати щоденні процеси медичної установи.

Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для обліку пацієнтів у медичному закладі.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програма призначена для обліку пацієнтів у медичному закладі. Програма написана на мові PHP, яка є однією з найпоширеніших мов для створення веборієнтованих систем, крім того, завдяки своїй високій сумісності з різноманітними системами управління базами даних мова PHP дозволяє ефективно організовувати зберігання і обробку великих обсягів медичної інформації. В якості фреймворку для створення програмного забезпечення для обліку пацієнтів у медичному закладі обрано Laravel завдяки його потужним можливостям, які дозволяють швидко створювати надійні, масштабовані та безпечні вебзастосунки.

Програмне забезпечення для обліку пацієнтів у медичному закладі забезпечує виконання таких функцій:

- підтримка можливості пошуку лікаря за прізвищем та спеціалізацією роботи у медичному закладі;
- можливість реєстрації та авторизації пацієнтів медичного закладу;
- можливість відображення інформації стосовно візитів пацієнта;
- підтримка можливості додавання інформації про дітей пацієнтів медичного закладу;
- підтримка можливості ознайомлення з розкладом роботи лікаря та запису на вільний час;
- підтримка можливості роботи з профілем користувача.

Для роботи програмного забезпечення для обліку пацієнтів у медичному закладі потрібні такі технічні та програмні ресурси: комп'ютерна система повинна бути оснащена процесором з тактовою частотою не нижче середнього рівня сучасних багатоядерних процесорів, що забезпечує належну швидкість обробки запитів та стабільну роботу програми навіть при високому навантаженні. Обсяг оперативної пам'яті має становити не менше восьми

гігабайт, що дозволить комфортно працювати з базами даних, веб-інтерфейсами та сервісами фонового виконання завдань. Обсяг вільного місця на жорсткому диску повинен забезпечувати збереження великої кількості інформації про пацієнтів, розклади, записи на прийом та інші дані, тож рекомендується наявність не менше ста гігабайт дискового простору, з можливістю масштабування у разі зростання обсягу даних.

Серед необхідного програмного забезпечення варто зазначити встановлення сучасної операційної системи, сумісної з вебсерверами, базами даних і фреймворком Laravel, який використовується для реалізації самого застосунку. Для коректної роботи програми потрібен встановлений вебсервер, наприклад Apache або Nginx, підтримка PHP актуальної версії, а також система управління базами даних.

Функціональні характеристики розробленого програмного забезпечення для обліку пацієнтів у медичному закладі наведено у технічному завданні (додаток А). Фрагмент тексту програмного забезпечення для обліку пацієнтів у медичному закладі наведено у додатку Б.

4.2 Характеристики програми для обліку пацієнтів у медичному закладі

Програмне забезпечення для обліку пацієнтів у медичному закладі складається з шести взаємопов'язаних модулів, кожен з яких відповідає за окремий аспект функціонування системи.

Програмне забезпечення для обліку пацієнтів у медичному закладі характеризується зручністю у використанні, адаптивністю до потреб різних категорій користувачів та високим рівнем надійності. Його інтерфейс реалізується таким чином, щоб забезпечити швидкий доступ до основних функцій без потреби у спеціальній підготовці персоналу. Система дозволяє автоматизувати облік, зберігання та обробку персональних і медичних даних, з урахуванням вимог до захисту конфіденційної інформації.

Серед ключових особливостей програмного забезпечення для обліку пацієнтів у медичному закладі варто відзначити модульну структуру, яка дозволяє змінювати або доповнювати функціонал без порушення роботи основної системи. База даних програмного забезпечення для обліку пацієнтів у медичному закладі слугує основою для централізованого зберігання, організації та доступу до інформації, що стосується пацієнтів, лікарів, медичних спеціальностей та записів на прийом. База даних забезпечує надійний, структурований та швидкий обміну даними між усіма компонентами системи, що дозволяє автоматизувати щоденні процеси медичної установи.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для запуску програмного забезпечення для обліку пацієнтів у медичному закладі на серверній частині необхідно запустити відповідний файл. Для звертання до клієнтської сторони необхідно перейти на відповідну вебсторінку.

4.3.2 Вхідні й вихідні дані

Вхідними даними до програмного забезпечення для обліку пацієнтів у медичному закладі є інформація про пацієнтів, зокрема особисті дані, контактна інформація, дата народження, а також записи про дітей. Додатково вводяться дані про лікарів, спеціалізації, графік роботи.

Вихідними даними програмного забезпечення для обліку пацієнтів у медичному закладі є підтвердження реєстрації, сформовані розклади прийомів, та інша інформація.

4.3.3 Повідомлення

Користувач програмного забезпечення для обліку пацієнтів у медичному закладі може отримати такі повідомлення: повідомлення про успішну реєстрацію, повідомлення про незаповнені поля, повідомлення про некоректно введені дані.

4.4 Виконання програмного забезпечення для обліку пацієнтів у медичному закладі

Після запуску програмного забезпечення для обліку пацієнтів у медичному закладі користувачу відображається головна сторінка (рис. 4.1), де надається можливість пошуку лікарів за прізвищем та спеціалізацією.

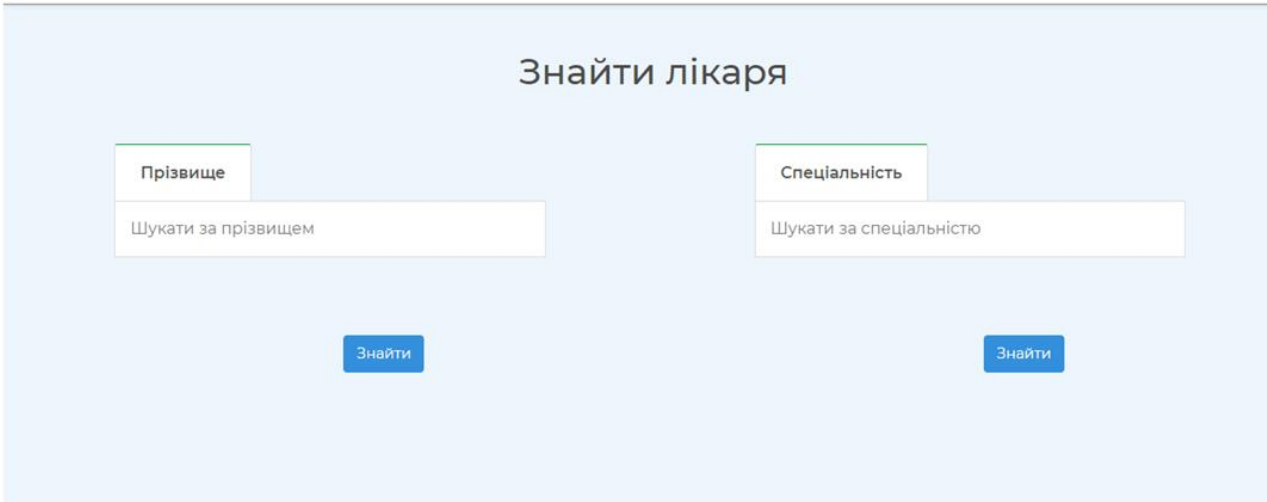


Рисунок 4.1 – Інтерфейс головної форми програмного забезпечення для обліку пацієнтів у медичному закладі

При виборі посилання Зареєструватися відбувається перехід до реєстраційної форми (рис. 4.2), де необхідно ввести особисті дані, зокрема, контактну інформацію (телефон та електронну пошту).

Пошук Увійти Зареєструватися

РЕЄСТРАЦІЯ

Особиста інформація:

Прізвище *

Ім'я *

По батькові

Дата народження *

Контактна інформація:

Контактний телефон *

Електронна пошта *

Пароль

Підтвердження паролю

Рисунок 4.2 – Форма реєстрації у програмному забезпеченні для обліку пацієнтів у медичному закладі

Зареєстровані користувачі для входу у програмне забезпечення для обліку пацієнтів у медичному закладі проходять процедуру авторизації (рис. 4.3) шляхом введення своєї електронної пошти та пароля.

Пошук Увійти Зареєструватися

УВІЙТИ

Електронна пошта

Пароль

☐ Запам'ятати мене

[Забули пароль?](#)

Рисунок 4.3 – Авторизація користувача програми для обліку пацієнтів у медичному закладі

Авторизованому користувачу відкривається можливість переходу до вкладки Візити (рис. 4.4), де відображається інформація про відвідування пацієнта медичного закладу.

Іван Сергєвнїн	Візити
Візити Діти Профіль	Запис для Сергєвнїн Іван Владиславович Лікар: Петренко Володимир Григорович Спеціалізація лікаря: сімейний лікар Дата: 08.05.2025 Час: 08:30

Рисунок 4.4 – Вкладка Візити програмного забезпечення для обліку пацієнтів у медичному закладі

При виборі пункту меню Діти відбувається перехід до відповідної форми (рис. 4.5), де надається можливість введення інформації про дитину користувача.

Іван Сергєвнїн	Діти
Візити Діти Профіль	Відсутні записи про дітей Додати дитину: Провище * Ім'я * По батькові Дата народження * дд.мм.гггг Додати

Рисунок 4.5 – Форма додавання даних про дітей

При виборі пункту меню Профіль відбувається перехід до профілю користувача (рис. 4.6), де надається можливість редагування інформації про себе, зокрема, особистих даних та контактної інформації.

Профіль

Іван Сергєвнн

Особиста інформація:

Прізвище: Сергєвнн

Ім'я: Іван

По батькові: Владиславович

Дата народження: 18.06.2004

Контактна інформація:

Контактний телефон: +380500778901

Електронна пошта: sergevnivld@gmail.com

[Зберегти зміни](#)

Рисунок 4.6 – Сторінка профілю користувача

Після знаходження потрібного лікаря користувачу відкривається форма з розкладом роботи лікаря (рис. 4.7), де шляхом натискання на відповідний час відбувається запис на прийом.

Розклад лікаря Петренко Володимир Григорович

Понеділок	Вівторок	Середа	Четвер	П'ятниця
08:00	08:00	08:00	08:00	08:00
08:15	08:15	08:15	08:15	08:15
08:30	08:30	08:30	08:30	08:30
08:45	08:45	08:45	08:45	08:45
09:00	09:00	09:00	09:00	09:00
09:15	09:15	09:15	09:15	09:15

[Записатися](#)

Рисунок 4.7 – Сторінка з розкладом роботи лікаря

4.5 Тестування програмного забезпечення для обліку пацієнтів у медичному закладі

Виконано тестування програмного забезпечення для обліку пацієнтів у медичному закладі.

Виявлено, що програма для обліку пацієнтів у медичному закладі функціонує правильно та злагоджено. Вона реалізує всі функціональні вимоги та успішно виконує свою основну задачу обліку пацієнтів у медичному закладі.

Розроблене програмне забезпечення дозволяє забезпечувати автоматизацію процесів, пов'язаних з підтримкою процесу обліку пацієнтів.

4.6 Висновки за розділом 4

Описано програмне забезпечення для обліку пацієнтів у медичному закладі. Виконано тестування розробленого програмного забезпечення для обліку пацієнтів у медичному закладі. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою процесу обліку пацієнтів.

ВИСНОВКИ

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

Визначено, що програмне забезпечення для обліку пацієнтів виконує функцію централізованого сховища даних, дозволяючи швидко створювати нові записи, оновлювати інформацію, відстежувати історію лікування та формувати звіти. Такий підхід полегшує процес призначення прийомів, поліпшує управління електронною медичною документацією та забезпечує більш точне ведення обліку. Такі програмні рішення допомагають медичним установам підтримувати відповідність нормативним вимогам і знижують ризик неточностей, що можуть виникати під час ручного введення даних. Їхнє використання створює умови для більшої прозорості в управлінні медичною інформацією, покращуючи доступ до актуальної інформації про пацієнтів.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для підтримки роботи медичних закладів. Проте деякі програмні засоби для обліку пацієнтів у медичних закладах мають високу вартість придбання, впровадження та обслуговування, що може бути непосильним для невеликих медичних закладів. Крім того, складність налаштування та інтеграції складного медичного програмного забезпечення може призводити до затримок у роботі та вимагати додаткових ресурсів. Крім того, надмірна стандартизація процесів у деяких програмних системах для обліку роботи медичних закладів може обмежувати індивідуальний підхід до пацієнтів і створювати додаткові адміністративні навантаження на лікарів. Тому актуальною є розробка програмного забезпечення для обліку пацієнтів у медичному закладі.

Сформульовано функціональні вимоги до програмного забезпечення для обліку пацієнтів у медичному закладі.

Для реалізації програмного забезпечення для обліку пацієнтів у медичному закладі обрано мову програмування PHP, яка є однією з найпоширеніших мов для створення веборієнтованих систем. Крім того, завдяки своїй високій сумісності з різноманітними системами управління базами даних мова PHP дозволяє ефективно організовувати зберігання і обробку великих обсягів медичної інформації. Іншою важливою перевагою мови програмування PHP є її доступність і відкритість, що суттєво знижує вартість розробки програмної системи порівняно з використанням комерційних технологій.

Для створення програмного забезпечення для обліку пацієнтів у медичному закладі обрано фреймворк Laravel завдяки його потужним можливостям, які дозволяють швидко створювати надійні, масштабовані та безпечні вебзастосунки. Крім того, Laravel також має високу продуктивність і масштабованість завдяки можливості легкої інтеграції з кешуванням, чергами задач і асинхронною обробкою подій, що дозволяє забезпечити стабільну роботу навіть при великій кількості користувачів і запитів.

Розроблено програмне забезпечення для обліку пацієнтів у медичному закладі.

Запропоновано структуру програмного забезпечення для обліку пацієнтів у медичному закладі, що складається з шести взаємопов'язаних модулів, кожен з яких відповідає за окремий аспект функціонування системи.

Наведено схему функціонування програмного забезпечення обліку пацієнтів у медичному закладі, на які роботу програми подано як послідовність узгоджених кроків, де кожен етап логічно пов'язаний з наступним.

Розроблено базу даних програмного забезпечення для обліку пацієнтів у медичному закладі, яка слугує основою для централізованого зберігання, організації та доступу до інформації, що стосується пацієнтів, лікарів, медичних спеціальностей та записів на прийом. Її призначення полягає у забезпеченні надійного, структурованого та швидкого обміну даними між

усіма компонентами системи, що дозволяє автоматизувати щоденні процеси медичної установи.

Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для обліку пацієнтів у медичному закладі.

Виконано тестування розробленого програмного забезпечення для обліку пацієнтів у медичному закладі. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою процесу обліку пацієнтів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Patient Registration Software: Transforming Healthcare Efficiency and Patient Experience [Electronic resource]. – Access mode: <https://savvycomsoftware.com/blog/patient-registration-software/>.
2. How Patient Registration Automation Tools Enhance Efficiency [Electronic resource]. – Access mode: <https://www.getmagical.com/blog/patient-registration-automation-tools>.
3. Efficient Patient Registration [Electronic resource]. – Access mode: <https://www.ehr-medical-software.com/patient-management-patient-registration.htm>.
4. The Essential Guide to Streamlining Your Patient Registration Process [Electronic resource]. – Access mode: <https://healthray.com/blog/hospital-management-software/essential-guide-streamlining-patient-registration-process/>.
5. Best Practices to Optimize Patient Registration Process in Hospitals [Electronic resource]. – Access mode: <https://www.invensis.net/blog/patient-registration-process-best-practices>.
6. Patient Registration Process [Electronic resource]. – Access mode: <https://www.carepatron.com/templates/patient-registration-process-flowchart>.
7. Best Patient Registration Software for Seamless Practice Management [Electronic resource]. – Access mode: <https://docresponse.com/blog/patient-registration-software/>.
8. Top 10 Digital Patient Intake Software Solutions [Electronic resource]. – Access mode: <https://kyruushealth.com/top-digital-patient-intake-software-solutions/>.
9. Top Patient Registry Platforms [Electronic resource]. – Access mode: <https://www.mahalo.health/insights/top-patient-registry-platforms>.
10. Best Patient Registration Software [Electronic resource]. – Access mode: <https://technologycounter.com/patient-registration-software>.

11. Luma Health [Electronic resource]. – Access mode: <https://www.lumahealth.io/>.
12. Phreesia [Electronic resource]. – Access mode: <https://www.phreesia.com/>.
13. Doctible [Electronic resource]. – Access mode: <https://www.doctible.com/>.
14. PHP 8.4.6 Released [Electronic resource]. – Access mode: <https://www.php.net/archive/2025.php#2025-04-10-3>.
15. PHP Form Handling [Electronic resource]. – Access mode: https://www.w3schools.com/php/php_forms.asp.
16. Laravel news [Electronic resource]. – Access mode: <https://laravel-news.com/>.
17. Laravel Request Lifecycle [Electronic resource]. – Access mode: <https://laravel.com/docs/11.x/lifecycle>.

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення може використовуватися для обліку пацієнтів у медичному закладі.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Програмне забезпечення обліку пацієнтів у медичному закладі», затверджене наказом Національного університету «Запорізька політехніка» № 209 від 28 квітня 2025 р.

A.2 Призначення розробки

Програмний продукт призначений для обліку пацієнтів у медичному закладі та дозволяє зареєстрованим користувачам ознайомлюватися з інформацією про роботу лікаря за виконувати запис на прийом.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Програмне забезпечення для обліку пацієнтів у медичному закладі забезпечує виконання таких функцій:

- підтримка можливості пошуку лікаря за прізвищем та спеціалізацією роботи у медичному закладі;
- можливість реєстрації та авторизації пацієнтів медичного закладу;
- можливість відображення інформації стосовно візитів пацієнта;
- підтримка можливості додавання інформації про дітей пацієнтів медичного закладу;

- підтримка можливості ознайомлення з розкладом роботи лікаря та запису на вільний час;
- підтримка можливості роботи з профілем користувача.

А.3.2 Вимоги до інтерфейсу програми

Інтерфейс програмного забезпечення для обліку пацієнтів у медичному закладі повинен бути зручним для користувачів. Повинна бути забезпечена можливість роботи в візуальному режимі.

А.3.3 Вимоги до надійності

Програмне забезпечення для обліку пацієнтів у медичному закладі повинно забезпечити надійне функціонування.

А.3.4 Умови експлуатації

Для експлуатації програмного забезпечення для обліку пацієнтів у медичному закладі необхідна наявність персонального комп'ютера.

А.3.5 Вимоги до складу та параметрів технічних засобів

Для роботи програмного забезпечення для обліку пацієнтів у медичному закладі потрібні такі технічні та програмні ресурси: комп'ютерна система повинна бути оснащена процесором з тактовою частотою не нижче середнього рівня сучасних багатоядерних процесорів, що забезпечує належну швидкість обробки запитів та стабільну роботу програми навіть при високому навантаженні. Обсяг оперативної пам'яті має становити не менше восьми

гігабайт, що дозволить комфортно працювати з базами даних, веб-інтерфейсами та сервісами фонового виконання завдань. Обсяг вільного місця на жорсткому диску повинен забезпечувати збереження великої кількості інформації про пацієнтів, розклади, записи на прийом та інші дані, тож рекомендується наявність не менше ста гігабайт дискового простору, з можливістю масштабування у разі зростання обсягу даних.

Серед необхідного програмного забезпечення варто зазначити встановлення сучасної операційної системи, сумісної з вебсерверами, базами даних і фреймворком Laravel, який використовується для реалізації самого застосунку. Для коректної роботи програми потрібен встановлений вебсервер, наприклад Apache або Nginx, підтримка PHP актуальної версії, а також система управління базами даних.

A.3.6 Вимоги до маркування і пакування

Програма для обліку пацієнтів у медичному закладі може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва програми – «Програмне забезпечення для обліку пацієнтів у медичному закладі».

A.4 Вхідні дані до роботи

Вхідними даними до програмного забезпечення для обліку пацієнтів у медичному закладі є інформація про пацієнтів, зокрема особисті дані, контактна інформація, дата народження, а також записи про дітей. Додатково вводяться дані про лікарів, спеціалізації, графік роботи.

Вихідними даними програмного забезпечення для обліку пацієнтів у медичному закладі є підтвердження реєстрації, сформовані розклади прийомів, та інша інформація.

ДОДАТОК Б
Фрагмент тексту програми

```

class PtnMng
{
    private $ptns = [];
    private $flnm = 'ptns_db.json';

    public function __cnstrct()
    {
        $this->ld();
    }

    private function ld()
    {
        if (flxsts($this->flnm)) {
            $cntnt = flgtscnts($this->flnm);
            $this->ptns = json_dcd($cntnt, tr);
        }
    }

    private function sv()
    {
        $cntnt = json_encd($this->ptns, JSON_PRTTY_PRNT);
        flptcnts($this->flnm, $cntnt);
    }

    public function dd_ptn($nm, $gndr, $dbrth, $phn, $ml)
    {
        $id = $this->gnrt_id();
        $this->ptns[$id] = [
            'nm' => $nm,
            'gndr' => $gndr,
            'dbrth' => $dbrth,
            'phn' => $phn,
            'ml' => $ml
        ];
        $this->sv();
        rtn $id;
    }

    private function gnrt_id()
    {
        return uniqid('ptn_');
    }

    public function gt_ptn($id)
    {
        if (isset($this->ptns[$id])) {
            return $this->ptns[$id];
        }
    }
}

```

```

    }
    return null;
}

public function pdt_ptn($id, $nm, $gndr, $dbrth, $phn, $ml)
{
    if (isset($this->ptns[$id])) {
        $this->ptns[$id]['nm'] = $nm;
        $this->ptns[$id]['gndr'] = $gndr;
        $this->ptns[$id]['dbrth'] = $dbrth;
        $this->ptns[$id]['phn'] = $phn;
        $this->ptns[$id]['ml'] = $ml;
        $this->sv();
        return true;
    }
    return false;
}

public function dl_ptn($id)
{
    if (isset($this->ptns[$id])) {
        unset($this->ptns[$id]);
        $this->sv();
        return true;
    }
    return false;
}

public function gt_ll_ptns()
{
    return $this->ptns;
}

public function fnd_ptn_nm($nm)
{
    $rslts = [];
    foreach ($this->ptns as $id => $ptn) {
        if (strcmp(strtlwr($ptn['nm']), strtlwr($nm)) == 0)
        {
            $rslts[$id] = $ptn;
        }
    }
    return $rslts;
}
}

$mng = new PtnMng();

```

```

function shw_mnu()
{
    echo "\n==== PTN MNG SYS ==== \n";
    echo "1. DD PTN \n";
    echo "2. VW LL PTNS \n";
    echo "3. GT PTN BY ID \n";
    echo "4. PDT PTN \n";
    echo "5. DL PTN \n";
    echo "6. FND PTN BY NM \n";
    echo "0. X \n";
    echo "CHC: ";
}

while (true) {
    shw_mnu();
    $chc = trim(fgets(STDIN));
    if ($chc == '0') {
        echo "BYE! \n";
        brk;
    }

    switch ($chc) {
        case '1':
            echo "NM: ";
            $nm = trim(fgets(STDIN));
            echo "GNDR (M/F): ";
            $gndr = trim(fgets(STDIN));
            echo "DBRTH (YYYY-MM-DD): ";
            $dbrth = trim(fgets(STDIN));
            echo "PHN: ";
            $phn = trim(fgets(STDIN));
            echo "ML: ";
            $ml = trim(fgets(STDIN));
            $id = $mng->dd_ptn($nm, $gndr, $dbrth, $phn, $ml);
            echo "PTN DD WTH ID: $id \n";
            brk;

        case '2':
            $ptns = $mng->gt_ll_ptns();
            foreach ($ptns as $id => $ptn) {
                echo "ID: $id | NM: {$ptn['nm']} | GNDR:
{$ptn['gndr']} | DBRTH: {$ptn['dbrth']} | PHN: {$ptn['phn']} |
ML: {$ptn['ml']} \n";
            }
            brk;

        case '3':
            echo "ID: ";

```

```

        $id = trim(fgets(STDIN));
        $ptn = $mng->gt_ptn($id);
        if ($ptn) {
            echo "ID: $id | NM: {$ptn['nm']} | GNDR:
{$ptn['gndr']} | DBRTH: {$ptn['dbrth']} | PHN: {$ptn['phn']} |
ML: {$ptn['ml']}\n";
        } else {
            echo "NT FND\n";
        }
        brk;

    case '4':
        echo "ID: ";
        $id = trim(fgets(STDIN));
        echo "NM: ";
        $nm = trim(fgets(STDIN));
        echo "GNDR (M/F): ";
        $gndr = trim(fgets(STDIN));
        echo "DBRTH (YYYY-MM-DD): ";
        $dbrth = trim(fgets(STDIN));
        echo "PHN: ";
        $phn = trim(fgets(STDIN));
        echo "ML: ";
        $ml = trim(fgets(STDIN));
        $rslt = $mng->pdt_ptn($id, $nm, $gndr, $dbrth, $phn,
$ml);

        if ($rslt) {
            echo "PTN PDT\n";
        } else {
            echo "PTN NT FND\n";
        }
        brk;

    case '5':
        echo "ID: ";
        $id = trim(fgets(STDIN));
        $rslt = $mng->dl_ptn($id);
        if ($rslt) {
            echo "PTN DLT\n";
        } else {
            echo "PTN NT FND\n";
        }
        brk;

    case '6':
        echo "NM: ";
        $nm = trim(fgets(STDIN));
        $rslts = $mng->fnd_ptn_nm($nm);

```

```

        if (count($rslts) > 0) {
            foreach ($rslts as $id => $ptn) {
                echo "ID: $id | NM: {$ptn['nm']} | GNDR:
{$ptn['gndr']} | DBRTH: {$ptn['dbrth']} | PHN: {$ptn['phn']} |
ML: {$ptn['ml']}\n";
            }
        } else {
            echo "NT FND\n";
        }
        brk;

        default:
            echo "NVD CHC\n";
    }
}

```

```

class Lgr
{
    private $flnm = 'lg.txt';

    public function wrt($msg)
    {
        $dt = dt('Y-m-d H:i:s');
        $ln = "[$dt] $msg\n";
        flppnd($this->flnm, $ln);
    }
}

```

```

class PtnMngXtd extends PtnMng
{
    private $lgr;

    public function __cnstrct()
    {
        prnt "LDG...\n";
        $this->lgr = new Lgr();
        prnt "LG INIT\n";
        prnt parent::__cnstrct();
    }

    public function dd_ptn($nm, $gndr, $dbrth, $phn, $ml)
    {
        $id = parent::dd_ptn($nm, $gndr, $dbrth, $phn, $ml);
        $this->lgr->wrt("DD: $id");
        rtrn $id;
    }
}

```

```

public function dl_ptn($id)
{
    $rslt = parent::dl_ptn($id);
    if ($rslt) {
        $this->lgr->wrt("DL: $id");
    }
    rtn $rslt;
}

public function pdt_ptn($id, $nm, $gndr, $dbrth, $phn, $ml)
{
    $rslt = parent::pdt_ptn($id, $nm, $gndr, $dbrth, $phn,
$ml);
    if ($rslt) {
        $this->lgr->wrt("PDT: $id");
    }
    rtn $rslt;
}

public function xpvt_csv($fn)
{
    $hdl = fopen($fn, 'w');
    fputcsv($hdl, ['ID', 'NM', 'GNDR', 'DBRTH', 'PHN', 'ML']);
    foreach ($this->gt_ll_ptns() as $id => $ptn) {
        fputcsv($hdl, [$id, $ptn['nm'], $ptn['gndr'],
$ptn['dbrth'], $ptn['phn'], $ptn['ml']]);
    }
    fclose($hdl);
    $this->lgr->wrt("XPRT: $fn");
}

public function fltr_g($g)
{
    $rslts = [];
    foreach ($this->gt_ll_ptns() as $id => $ptn) {
        if (strtlwr($ptn['gndr']) == strtlwr($g)) {
            $rslts[$id] = $ptn;
        }
    }
    return $rslts;
}

public function fltr_age_gt($age)
{
    $rslts = [];
    $now = dt('Y-m-d');
    foreach ($this->gt_ll_ptns() as $id => $ptn) {
        $y = dt_diff(dt_crt($ptn['dbrth']), dt_crt($now))->y;

```

```

        if ($y >= $age) {
            $rslts[$id] = $ptn;
        }
    }
    return $rslts;
}

public function stt()
{
    $cnt = count($this->gt_ll_ptns());
    $m = count($this->fltr_g('M'));
    $f = count($this->fltr_g('F'));
    echo "TTL: $cnt\n";
    echo "M: $m\n";
    echo "F: $f\n";
}
}

function dt($fmt)
{
    return dt_frmt(new dtmt(), $fmt);
}

function dt_crt($str)
{
    return dtmt::crt_frmt('Y-m-d', $str);
}

function dt_diff($d1, $d2)
{
    return $d1->dff($d2);
}

function flxsts($fn)
{
    return fl_exsts($fn);
}

function flgtscnts($fn)
{
    return fl_get_cnts($fn);
}

function flptcnts($fn, $cnt)
{
    return fl_pt_cnts($fn, $cnt);
}

```

```

function flppnd($fn, $ln)
{
    $hdl = fopen($fn, 'a');
    fwrite($hdl, $ln);
    fclose($hdl);
}

function json_dcd($str, $as_arr)
{
    return json_decode($str, $as_arr);
}

function json_encd($arr, $opt)
{
    return json_encode($arr, $opt);
}

$mng = new PtnMngXtd();

function shw_mnu_x()
{
    echo "\n==== PTN MNG SYS XTD++ ==== \n";
    echo "1. DD PTN \n";
    echo "2. VW LL PTNS \n";
    echo "3. GT PTN BY ID \n";
    echo "4. PDT PTN \n";
    echo "5. DL PTN \n";
    echo "6. FND PTN BY NM \n";
    echo "7. XPRT TO CSV \n";
    echo "8. FLTR BY GNDR \n";
    echo "9. FLTR BY AGE >= N \n";
    echo "10. STT \n";
    echo "11. IMPRT CSV \n";
    echo "12. SRT BY NM \n";
    echo "13. SRT BY DBRTH \n";
    echo "14. SRT BY ID \n";
    echo "15. CLR ALL \n";
    echo "16. BCKP \n";
    echo "17. GEN RNG \n";
    echo "18. FND BY PHN \n";
    echo "19. FND BY ML \n";
    echo "20. FLTR BY YR \n";
    echo "21. XPRT JSON \n";
    echo "22. AG AVG \n";
    echo "23. AG MIN \n";
    echo "24. AG MAX \n";
    echo "0. X \n";
    echo "CHC: "; }

```

```

while (true) {
    shw_mnu_x();
    $chc = trim(fgets(STDIN));
    if ($chc == '0') {
        echo "BYE!\n";
        break;
    }

    switch ($chc) {
        case '1':
            echo "NM: ";
            $nm = trim(fgets(STDIN));
            echo "GNDR (M/F): ";
            $gndr = trim(fgets(STDIN));
            echo "DBRTH (YYYY-MM-DD): ";
            $dbrth = trim(fgets(STDIN));
            echo "PHN: ";
            $phn = trim(fgets(STDIN));
            echo "ML: ";
            $ml = trim(fgets(STDIN));
            $id = $mng->dd_ptn($nm, $gndr, $dbrth, $phn, $ml);
            echo "PTN DD WTH ID: $id\n";
            break;

        case '2':
            $ptns = $mng->gt_ll_ptns();
            foreach ($ptns as $id => $ptn) {
                echo "ID: $id | NM: {$ptn['nm']} | GNDR:
{$ptn['gndr']} | DBRTH: {$ptn['dbrth']} | PHN: {$ptn['phn']} |
ML: {$ptn['ml']}\n";
            }
            break;

        case '3':
            echo "ID: ";
            $id = trim(fgets(STDIN));
            $ptn = $mng->gt_ptn($id);
            if ($ptn) {
                echo "ID: $id | NM: {$ptn['nm']} | GNDR:
{$ptn['gndr']} | DBRTH: {$ptn['dbrth']} | PHN: {$ptn['phn']} |
ML: {$ptn['ml']}\n";
            } else {
                echo "NT FND\n";
            }
            break;

        case '4':

```

```

echo "ID: ";
$id = trim(fgets(STDIN));
echo "NM: ";
$nm = trim(fgets(STDIN));
echo "GNDR (M/F): ";
$gndr = trim(fgets(STDIN));
echo "DBRTH (YYYY-MM-DD): ";
$dbrth = trim(fgets(STDIN));
echo "PHN: ";
$phn = trim(fgets(STDIN));
echo "ML: ";
$m1 = trim(fgets(STDIN));
$rslt = $mng->pdt_ptn($id, $nm, $gndr, $dbrth, $phn,
$m1);

if ($rslt) {
    echo "PTN PDT\n";
} else {
    echo "PTN NT FND\n";
}
break;

case '5':
echo "ID: ";
$id = trim(fgets(STDIN));
$rslt = $mng->dl_ptn($id);
if ($rslt) {
    echo "PTN DLT\n";
} else {
    echo "PTN NT FND\n";
}
break;

case '6':
echo "NM: ";
$nm = trim(fgets(STDIN));
$rslts = $mng->fnd_ptn_nm($nm);
if (count($rslts) > 0) {
    foreach ($rslts as $id => $ptn) {
        echo "ID: $id | NM: {$ptn['nm']} | GNDR:
{$ptn['gndr']} | DBRTH: {$ptn['dbrth']} | PHN: {$ptn['phn']} |
ML: {$ptn['ml']}\n";
    }
} else {
    echo "NT FND\n";
}
break;

case '7':

```

```

echo "CSV FN: ";
$fn = trim(fgets(STDIN));
$mng->xprt_csv($fn);
echo "XPRT DNE\n";
break;

case '8':
echo "GNDR (M/F): ";
$g = trim(fgets(STDIN));
$rslts = $mng->fltr_g($g);
if (count($rslts) > 0) {
    foreach ($rslts as $id => $ptn) {
        echo "ID: $id | NM: {$ptn['nm']} | GNDR:
{$ptn['gndr']} | DBRTH: {$ptn['dbrth']} | PHN: {$ptn['phn']} |
ML: {$ptn['ml']}\n";
    }
} else {
    echo "NT FND\n";
}
break;

case '9':
echo "AGE >=: ";
$age = trim(fgets(STDIN));
$rslts = $mng->fltr_age_gt($age);
if (count($rslts) > 0) {
    foreach ($rslts as $id => $ptn) {
        echo "ID: $id | NM: {$ptn['nm']} | GNDR:
{$ptn['gndr']} | DBRTH: {$ptn['dbrth']} | PHN: {$ptn['phn']} |
ML: {$ptn['ml']}\n";
    }
} else {
    echo "NT FND\n";
}
break;

case '10':
$mng->stt();
break;

case '11':
echo "CSV FN: ";
$fn = trim(fgets(STDIN));
$mng->imprt_csv($fn);
echo "IMPRT DNE\n";
break;

case '12':

```

```

        $rslts = $mng->srt_nm();
        foreach ($rslts as $id => $ptn) {
            echo "ID: $id | NM: {$ptn['nm']} | GNDR:
{$ptn['gndr']} | DBRTH: {$ptn['dbrth']} | PHN: {$ptn['phn']} |
ML: {$ptn['ml']}\n";
        }
        break;

    case '13':
        $rslts = $mng->srt_dbrth();
        foreach ($rslts as $id => $ptn) {
            echo "ID: $id | NM: {$ptn['nm']} | GNDR:
{$ptn['gndr']} | DBRTH: {$ptn['dbrth']} | PHN: {$ptn['phn']} |
ML: {$ptn['ml']}\n";
        }
        break;

    case '14':
        $rslts = $mng->srt_id();
        foreach ($rslts as $id => $ptn) {
            echo "ID: $id | NM: {$ptn['nm']} | GNDR:
{$ptn['gndr']} | DBRTH: {$ptn['dbrth']} | PHN: {$ptn['phn']} |
ML: {$ptn['ml']}\n";
        }
        break;

    case '15':
        $mng->clr_all();
        echo "ALL CLR\n";
        break;

    case '16':
        echo "BCKP FN: ";
        $fn = trim(fgets(STDIN));
        $mng->bckp($fn);
        echo "BCKP DNE\n";
        break;

    case '17':
        echo "CNT: ";
        $cnt = trim(fgets(STDIN));
        $mng->gen_rng($cnt);
        echo "GEN $cnt DONE\n";
        break;

    case '18':
        echo "PHN SSTR: ";
        $phn = trim(fgets(STDIN));

```

```

        $rsalts = $mng->fnd_phn($phn);
        foreach ($rsalts as $id=>$ptn) {
            echo "ID: $id | NM: {$ptn['nm']} | PHN:
{$ptn['phn']}\n";
        }
        break;

    case '19':
        echo "ML SSTR: ";
        $ml = trim(fgets(STDIN));
        $rsalts = $mng->fnd_ml($ml);
        foreach ($rsalts as $id=>$ptn) {
            echo "ID: $id | NM: {$ptn['nm']} | ML:
{$ptn['ml']}\n";
        }
        break;

    case '20':
        echo "YR: ";
        $yr = trim(fgets(STDIN));
        $rsalts = $mng->fltr_yr($yr);
        foreach ($rsalts as $id=>$ptn) {
            echo "ID: $id | NM: {$ptn['nm']} | DBRTH:
{$ptn['dbrth']}\n";
        }
        break;

    case '21':
        echo "JSON FN: ";
        $fn = trim(fgets(STDIN));
        $mng->xprt_json($fn);
        echo "XPRT JSON DNE\n";
        break;

    case '22':
        $avg = $mng->ag_avg();
        echo "AVG AGE: $avg\n";
        break;

    case '23':
        $min = $mng->ag_min();
        echo "MIN AGE: $min\n";
        break;

    case '24':
        $max = $mng->ag_max();
        echo "MAX AGE: $max\n";
        break;

```

```

        default:
            echo "NVD CHC\n";
    }
}

public function imprt_csv($fn)
{
    if (!flxsts($fn)) {
        echo "FN NT FND\n";
        return;
    }
    $hdl = fopen($fn, 'r');
    fgetcsv($hdl); // skip hdr
    while (($row = fgetcsv($hdl)) !== false) {
        [$id, $nm, $gndr, $dbrth, $phn, $ml] = $row;
        $this->dd_ptn($nm, $gndr, $dbrth, $phn, $ml);
    }
    fclose($hdl);
    $this->lgr->wrt("IMPRT: $fn");
}

public function srt_nm()
{
    $ptns = $this->gt_ll_ptns();
    uasort($ptns, fn($a,$b)=>strcmp($a['nm'],$b['nm']));
    return $ptns;
}

public function srt_dbrth()
{
    $ptns = $this->gt_ll_ptns();
    uasort($ptns, fn($a,$b)=>strcmp($a['dbrth'],$b['dbrth']));
    return $ptns;
}

public function srt_id()
{
    $ptns = $this->gt_ll_ptns();
    ksort($ptns);
    return $ptns;
}

public function clr_all()
{
    $this->ptns = [];
    $this->sv();
    $this->lgr->wrt("CLR ALL");
}

```

```

}

public function bckp($fn)
{
    $json = json_encd($this->gt_ll_ptns(), 128);
    flptcnts($fn, $json);
    $this->lgr->wrt("BCKP: $fn");
}

public function gen_rng($cnt)
{
    $nms = ['ALX', 'MRX', 'JHN', 'ANN', 'BRT', 'KRL', 'DNL', 'VCT'];
    $gndrs = ['M', 'F'];
    for ($i=0; $i<$cnt; $i++) {
        $nm = $nms[rand(0, count($nms)-1)] . rand(1, 999);
        $gndr = $gndrs[rand(0, 1)];
        $yr = rand(1950, 2010);
        $mth = str_pad(rand(1, 12), 2, '0', STR_PAD_LEFT);
        $dy = str_pad(rand(1, 28), 2, '0', STR_PAD_LEFT);
        $dbrth = "$yr-$mth-$dy";
        $phn = '+380' . rand(100000000, 999999999);
        $ml = strtolower($nm) . '@mail.com';
        $this->dd_ptn($nm, $gndr, $dbrth, $phn, $ml);
    }
    $this->lgr->wrt("GEN RNG $cnt");
}

public function fnd_phn($phn)
{
    $res = [];
    foreach ($this->ptns as $id => $ptn) {
        if (strstr($ptn['phn'], $phn)) {
            $res[$id] = $ptn;
        }
    }
    return $res;
}

public function fnd_ml($ml)
{
    $res = [];
    foreach ($this->ptns as $id => $ptn) {
        if (strstr($ptn['ml'], $ml)) {
            $res[$id] = $ptn;
        }
    }
    return $res;
}

```

```

public function fltr_yr($yr)
{
    $res = [];
    foreach ($this->ptns as $id => $ptn) {
        if (substr($ptn['dbrth'],0,4) == $yr) {
            $res[$id] = $ptn;
        }
    }
    return $res;
}

public function xpirt_json($fn)
{
    $json = json_encd($this->gt_ll_ptns(), JSON_PRTTY_PRINT);
    flptcnts($fn, $json);
    $this->lgr->wrt("XPRT JSON: $fn");
}

public function ag_avg()
{
    $ages = [];
    foreach ($this->ptns as $ptn) {
        $byr = substr($ptn['dbrth'],0,4);
        $age = date('Y') - $byr;
        $ages[] = $age;
    }
    if (cnt($ages)==0) return 0;
    return array_sum($ages) / cnt($ages);
}

```

ДОДАТОК В
Слайди презентації

Міністерство освіти і науки України
Національний університет «Запорізька політехніка»
Кафедра програмних засобів

Програмне забезпечення обліку пацієнтів у медичному закладі

Виконав: студент групи КНТ-121
Керівник: к.т.н., доцент

Іван СЕРГЕВНІН
Степан СКРУПСЬКИЙ

Рисунок В.1 – Слайд 1

ОБ'ЄКТ, ПРЕДМЕТ ТА МЕТА РОБОТИ

- **Об'єкт дослідження** – процес розробки програмного забезпечення для медичних закладів.
- **Предмет дослідження** – програмні засоби для медичних закладів.
- **Метою роботи є** розробка програмного забезпечення для підтримки процесу обліку пацієнтів у медичному закладі.

2

Рисунок В.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 роботи медичних закладів;
- здійснити проектування програмного забезпечення для обліку пацієнтів у медичному закладі;
- створити програмне забезпечення для обліку пацієнтів у медичному закладі;
- виконати тестування розробленого програмного забезпечення для обліку пацієнтів у медичному закладі.

3

Рисунок В.3 – Слайд 3

АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

Критерій порівняння	Luma Health	Phreesia	Doctible
Простота використання	+	+-	+
Маркетингові інструменти	+-	-	+
Зручність автоматизованих нагадувань	+-	+	+-
Інтеграція з електронною медичною картою	+	+-	+
Налаштовування під спеціалізацію	+-	+	+
Підтримка пацієнтів через кілька каналів	+	+-	+-

4

Рисунок В.4 – Слайд 4

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

При розробці програмного забезпечення для обліку пацієнтів у медичному закладі необхідно забезпечити такі функціональні вимоги:

- підтримка можливості пошуку лікаря за прізвищем та спеціалізацією роботи у медичному закладі;
- можливість реєстрації та авторизації пацієнтів медичного закладу;
- можливість відображення інформації стосовно візитів пацієнта;
- підтримка можливості додавання інформації про дітей пацієнтів медичного закладу;
- підтримка можливості ознайомлення з розкладом роботи лікаря та запису на вільний час;
- підтримка можливості роботи з профілем користувача.

5

Рисунок В.5 – Слайд 5

ВИБІР МОВИ ПРОГРАМУВАННЯ

Критерій порівняння мов програмування	Мова програмування		
	PHP	Python	Java
Простота розробки вебзастосунків	+	+-	+-
Швидкість розробки	+	+-	-
Готові фреймворки для веброзробки	+	+	+-
Зручність для створення ПЗ для обліку пацієнтів у медичному закладі	+	-	-
Сумісність з більшістю серверів	+	+-	+-
Невисокі вимоги до ресурсів сервера	+	+-	+-

6

Рисунок В.6 – Слайд 6

ВИБІР ФРЕЙМВОРКУ

Критерій порівняння фреймворків	Фреймворки		
	Laravel	Symfony	CodeIgniter
Легкість інтеграції зовнішніх сервісів	+	+–	+–
Активність і підтримка спільноти	+	+	+–
Розвинена документація	+	+	+–
Простота та швидкість розробки	+	+–	+
Готові рішення для аутентифікації та авторизації	+	+–	–

Рисунок В.7 – Слайд 7

СТРУКТУРА ПЗ ДЛЯ ОБЛІКУ ПАЦІЄНТІВ У МЕДИЧНОМУ ЗАКЛАДІ

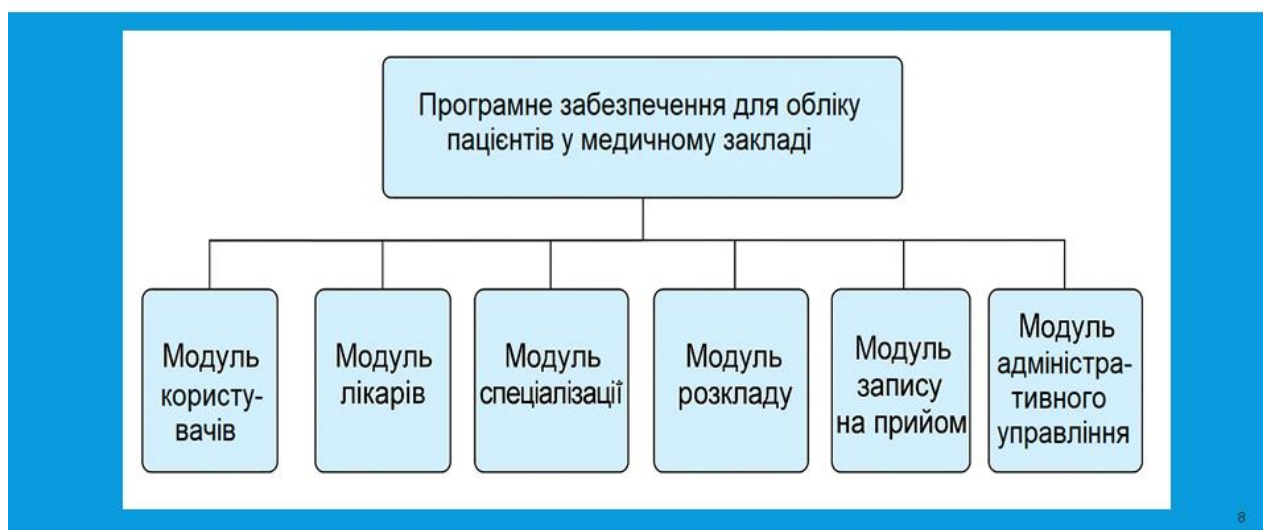


Рисунок В.8 – Слайд 8

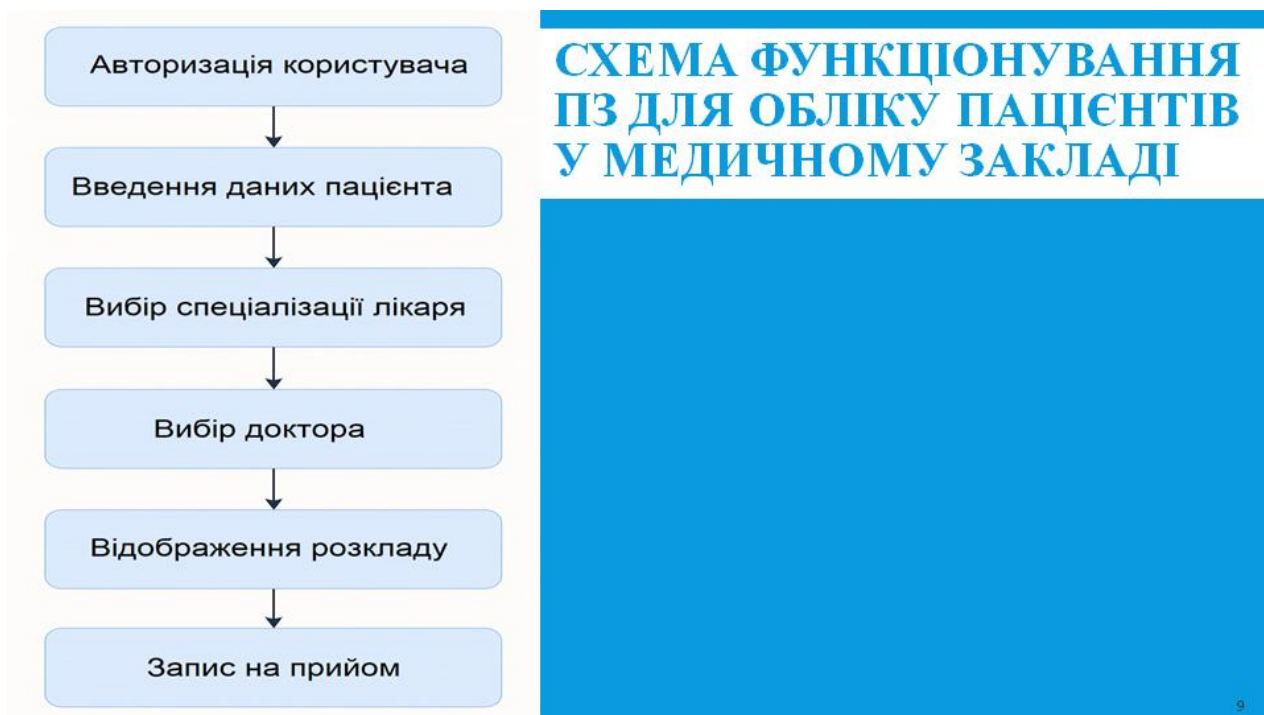


Рисунок В.9 – Слайд 9

СХЕМА БАЗИ ДАНИХ

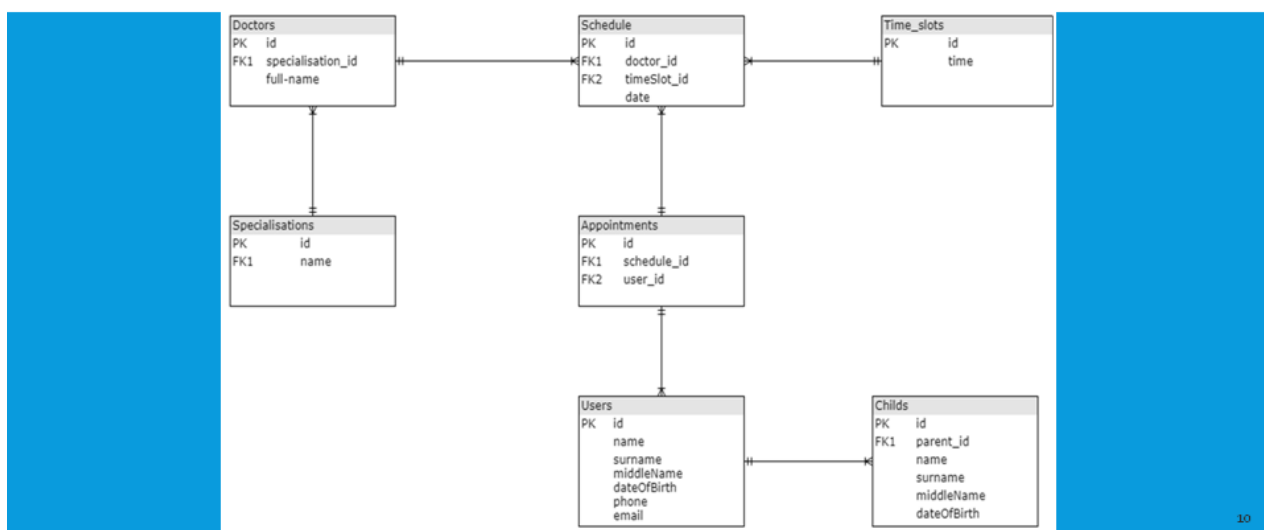


Рисунок В.10 – Слайд 10

РОБОТА З ПРОГРАМОЮ. ГОЛОВНА СТОРІНКА. РЕЄСТРАЦІЯ ТА АВТОРИЗАЦІЯ

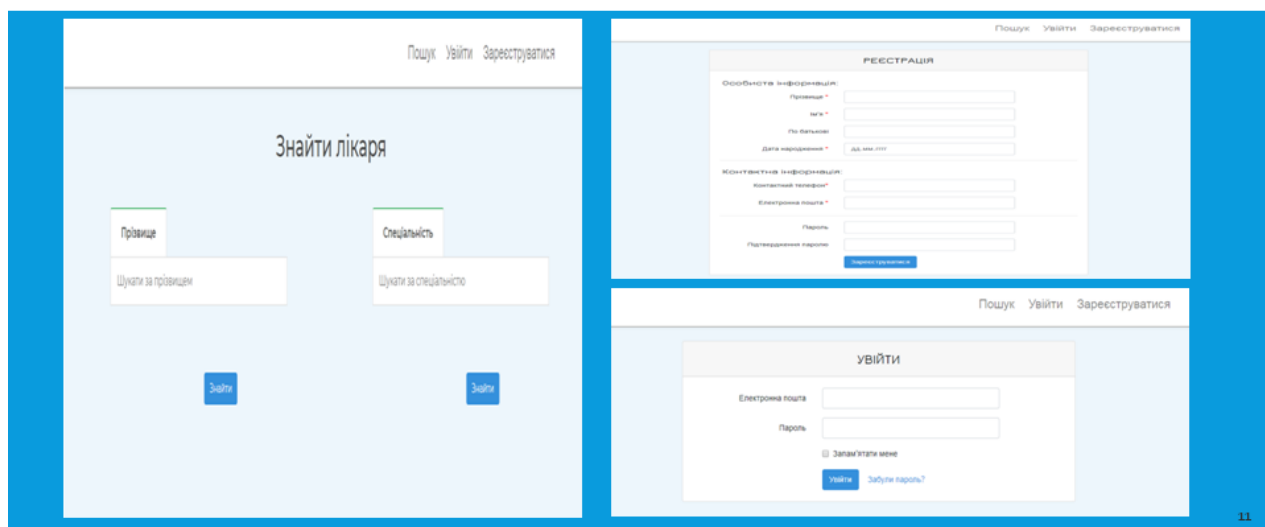


Рисунок В.11 – Слайд 11

РОБОТА З ПРОГРАМОЮ. ПРОФІЛЬ КОРИСТУВАЧА. ЗАПИС ДО ЛІКАРЯ

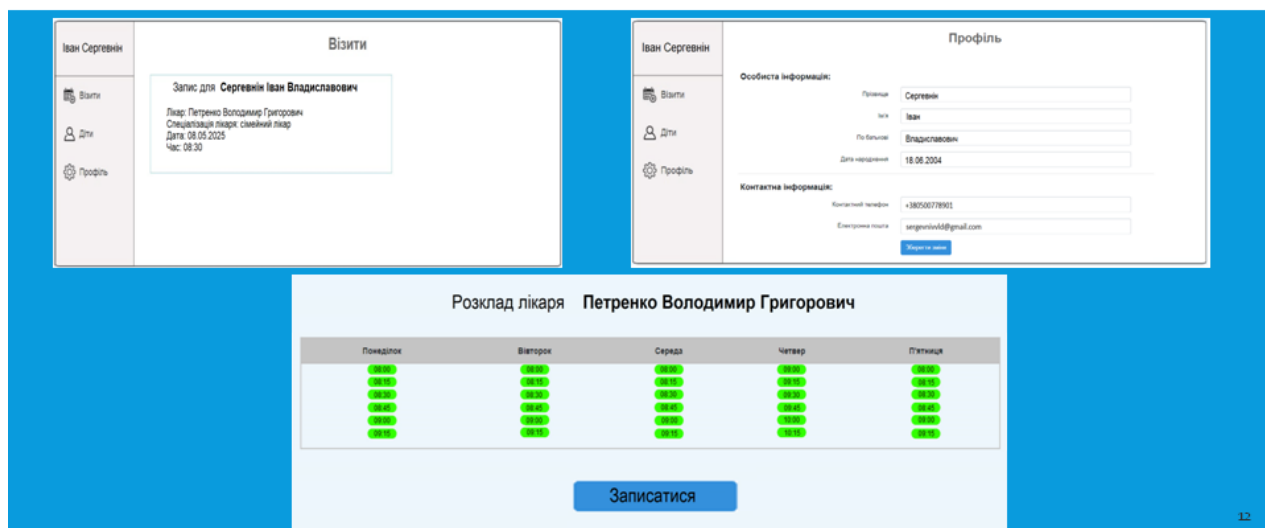


Рисунок В.12 – Слайд 12

ВИСНОВКИ

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

Сформульовано функціональні вимоги до програмного забезпечення для обліку пацієнтів у медичному закладі. Для реалізації програмного забезпечення для обліку пацієнтів у медичному закладі обрано мову програмування PHP та фреймворк Laravel.

Розроблено програмне забезпечення для обліку пацієнтів у медичному закладі. Розроблено базу даних програмного забезпечення для обліку пацієнтів у медичному закладі, яка слугує основою для централізованого зберігання, організації та доступу до інформації, що стосується пацієнтів, лікарів, медичних спеціальностей та записів на прийом.

Виконано тестування розробленого програмного забезпечення для обліку пацієнтів у медичному закладі. Результати тестування показали, що програма відповідає вимогам.

13

Рисунок В.13 – Слайд 13