

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіоелектроніки
Факультет комп'ютерних наук та технологій

(повне найменування інституту, факультету)

Кафедра комп'ютерних систем та мереж

(повне найменування кафедри)

ПОЯСНЮВАЛЬНА ЗАПИСКА
до дипломного проекту (роботи)
бакалавра

(ступінь вищої освіти (освітній ступінь))

на тему
РОЗРОБКА ВЕБ-ОРІЄНТОВАНОЇ СИСТЕМИ ОБЛІКУ
ЛІКАРСЬКИХ ПРИЗНАЧЕНЬ

Виконав: студент 4 курсу, групи КНТз-511сп

Спеціальності 123 «Комп'ютерна інженерія»
(код і найменування спеціальності)

Освітня програма (спеціалізація) _____
«Комп'ютерна інженерія»

ЗАНЕМОНЦЕВ А. Є.
(прізвище та ініціали)

Керівник ТЯГУНОВА М.Ю.
(прізвище та ініціали)

Рецензент СТЕПАНЕНКО О.О.
(прізвище та ініціали)

2024 рік

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет _____ Комп'ютерних наук та технологій
 Кафедра _____ Комп'ютерних систем та мереж
 Ступінь вищої освіти _____ бакалаврський
 Спеціальність _____ 123 Комп'ютерна інженерія
 (код і найменування)
 Освітня програма (спеціалізація) _____ Комп'ютерна інженерія
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Кудерметов Р.К.

« _____ » _____ 2024 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

_____ ЗАНЕМОНИЦЕВУ Антону Євгеновичу

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) _____ Розробка веб-орієнтованої системи обліку лікарських
 призначень

керівник проєкту (роботи) _____ к.т.н доцент ТЯГУНОВА Марія Юріївна,
 (науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від « 01 » квітня 2024 року №105

2. Строк подання студентом проєкту (роботи) _____ 01.06. 2024 року

3. Вихідні дані до проєкту (роботи) лікарські призначення між закладом охорони
 здоров'я, пацієнтом та аптекою

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

1. Аналіз предметної області;

2. Проєктування веб-орієнтованої системи;

3. Реалізація прототипу веб-орієнтованої системи.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень,
 кількість слайдів, плакатів)

Прототипи веб-сторінок, Прототип головної сторінки для лікаря/фармацевта,
 Прототип сторінки «Довідка», Прототип сторінки «Особистий кабінет» для пацієнта,
 Прототип сторінки «Особистий кабінет» для лікаря/фармацевта, Прототип сторінки
 «Аптеки поруч» для пацієнта

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-3	ТЯГУНОВА М.Ю., к.т.н. доцент		
Нормоконтроль	ПОЛЬСЬКА О.В., ст. викладач		

7. Дата видачі завдання 01.04.2024 р.

КАЛЕНДАРНИЙ ПЛАН

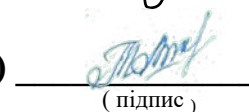
№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Дослідження стану автоматизації бізнес-процесів	05.03.2024	
2	Формування вимог до веб-орієнтованої системи	05.04.2024	
3	Проектування веб-орієнтованої системи	20.04.2024	
4	Реалізація прототипу веб-орієнтованої системи	26.04.2024	
5	Оформлення графічного матеріалу	12.05.2024	
6	Оформлення пояснювальної записки	29.05.2024	

Студент(ка)


 (підпис)
АНТОН ЗАНЕМОНЦЕВ

(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)


 (підпис)
Марія ТЯГУНОВА

(Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

ПЗ: 46 стор., 10 рис., 1 додаток

MYSQL, PHP, WEB, АВТОМАТИЗАЦІЯ, АВТОМАТИЗАЦІЯ ПРОЦЕСІВ,
АПТЕКА, СУБД, ФАРМАЦЕВТИЧНЕ ПІДПРИЄМСТВО, КЕРУВАННЯ.

Метою даної кваліфікаційної роботи є створення системи, що автоматизує процес документообігу лікарських призначень і дозволить зберігати дані про призначення на вебсервері з можливістю доступу до них у будь-який час.

Об'єктом розроблення є вебсистема для керування лікарськими призначеннями, що містить серверну та клієнтську частини, а також механізм інтеграції з наявними медичними системами.

Предметом дослідження є технології розроблення вебдодатків, методи керування лікарськими призначеннями та підходи до забезпечення безпеки даних у медичних інформаційних системах.

Для досягнення поставленої мети було проведено всебічний аналіз технічної літератури та наявних технологій розробки вебдодатків. Було вивчено методи управління лікарськими призначеннями, а також сучасні підходи до забезпечення безпеки даних. На основі цього аналізу було обрано найбільш підходящі технології та методи для створення ефективної вебсистеми.

Результатом роботи є створення вебсистеми для управління лікарськими призначеннями. Система включає інтерфейс для медичного та фармацевтичного персоналу, серверну частину для зберігання та обробки даних, а також механізм інтеграції з наявними медичними системами. Система може бути впроваджена у медичних установах і аптеках, що дасть змогу підвищити точність і ефективність призначення та відпуску лікарських засобів, а також забезпечити надійне зберігання та доступ до медичних даних.

ЗМІСТ

Вступ.....	6
1 Аналіз предметної області	8
1.1 Специфіка та функції аптечних закладів	8
1.2 Поточний рівень автоматизації бізнес-процесів.....	9
1.3 Визначення ключових вимог для веб-орієнтованої системи	11
2 Проєктування вебдодатку	14
2.1 Проєктування бізнес-процесів.....	14
2.1.1 Data Flow Diagram	15
2.1.2 Unified Modeling Language	15
2.1.3 Language Integrated Definition for Function Modeling (IDEF0, IDEF3)	16
2.1.4 Business Process Modeling Notation.....	16
2.2 Архітектурна модель і методи реалізації системи	18
3 Реалізація прототипу вебсистеми.....	21
3.1 Етапи створення вебдодатка	21
3.2 Реалізація вебсистеми	23
3.3 Розробка керівництва з експлуатації.....	27
3.4 Економічна доцільність розробки вебдодатка.....	29
Висновки.....	33
Перелік джерел посилання	35
Додаток А	40

ВСТУП

У сучасному світі цифрових технологій та інформатизації охорона здоров'я також не залишається осторонь, активно впроваджуючи нові розробки для підвищення ефективності медичного обслуговування. Однією з таких важливих галузей є управління лікарськими призначеннями, що особливо актуально в умовах зростаючого навантаження на медичні заклади та збільшення обсягу інформації. Технології управління лікарськими призначеннями дають змогу оптимізувати процес призначення та відпуску лікарських засобів, мінімізувати помилки, пов'язані з людським фактором, і забезпечувати високий ступінь безпеки та конфіденційності даних пацієнтів [1, 2].

Основою для створення ефективної системи управління лікарськими призначеннями служить інтеграція сучасних технологій веб-розробки, баз даних та інформаційної безпеки. Впровадження таких систем істотно полегшує роботу фармацевтичного персоналу, надаючи їм зручні інструменти для управління та моніторингу призначень ліків.

На сьогоднішній день медичні установи прагнуть до автоматизації багатьох процесів, включно з управлінням лікарськими призначеннями. Це зумовлено необхідністю підвищення точності та оперативності в наданні медичних послуг, а також поліпшенням взаємодії між лікарями та аптеками. Сучасні вебсистеми дають змогу зберігати й обробляти великі обсяги даних, забезпечувати доступ до них у режимі реального часу та захищати конфіденційну інформацію пацієнтів [3].

Розробка та впровадження вебсистеми управління лікарськими призначеннями є важливим кроком на шляху до цифрової трансформації охорони здоров'я. Така система надає можливість впорядкувати процес виписування та обліку ліків й інтегрувати його з іншими медичними інформаційними системами, забезпечуючи комплексний підхід до лікування пацієнтів [4].

У межах цієї дипломної роботи було поставлено за мету розробити

вебсистему для управління лікарськими призначеннями, яка забезпечувала б зберігання даних на сервері та доступ до них медичного та фармацевтичного персоналу в будь-який час.

Актуальність створення вебсистеми управління лікарськими призначеннями зумовлена такими факторами:

Метою даної кваліфікаційної роботи є створення системи, що автоматизує процес документообігу лікарських призначень і дозволить зберігати дані про призначення на вебсервері з можливістю доступу до них у будь-який час. Передбачається створення реляційної бази даних, взаємодія користувача з якою відбуватиметься через вебдодаток.

Дослідження фокусується на вивченні циркуляції документів, пов'язаних із призначенням ліків, між клінікою, пацієнтом і аптекою.

Предметом дослідження виступають моделі та методи, програмні засоби та методи створення веб-орієнтованих інформаційних систем для обліку лікарських призначень.

У центрі уваги перебувають різні підходи, технології та методики створення інформаційних систем на основі веб-технологій для реєстрації та відстеження призначених медикаментів.

Ключові цілі кваліфікаційної роботи включають:

- аналіз процесу обміну документацією щодо лікарських призначень;
- визначення основних і додаткових вимог до розроблюваної системи обліку медикаментів;
- проектування структури інформаційної системи для ведення обліку;
- створення дослідного прототипу облікової інформаційної системи;
- прогнозування потенційних результатів від застосування вебплатформи для управління призначеннями лікарських засобів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Специфіка та функції аптечних закладів

Аптека являє собою спеціалізовану установу системи охорони здоров'я, що відіграє ключову роль у забезпеченні населення необхідними медикаментами та наданні медичних послуг. Основна діяльність аптеки пов'язана з виготовленням, фасуванням, аналізом і реалізацією лікарських засобів. Аптечні заклади виконують важливі завдання з виготовлення ліків за рецептами лікарів, контролю їхньої якості та реалізації як рецептурних, так і безрецептурних препаратів.

У межах функції закладу охорони здоров'я аптеки часто надають першу медичну допомогу, проводять санітарно-просвітницьку роботу та здійснюють фармацевтичну опіку. Фармацевти консультують пацієнтів з питань застосування ліків, можливих побічних ефектів і взаємодій з іншими препаратами. Крім того, аптеки слугують джерелом актуальної інформації про нові ліки для медичних фахівців.

Виробнича функція аптек включає виготовлення індивідуальних рецептурних ліків і серійне виробництво деяких видів медикаментів. В аптеках здійснюється суворий контроль якості, правильне маркування та зберігання лікарських засобів. Торгова функція передбачає закупівлю медикаментів у постачальників, проведення вхідного контролю якості, а також забезпечення правильного зберігання і реалізації ліків.

Аптечні пункти і кіоски, будучи меншими за масштабом, також виконують важливі функції з реалізації ліків, надання інформації про їхнє правильне використання і зберігання, а в деяких випадках можуть надавати першу медичну допомогу.

В Україні існують різні форми власності аптек: приватні, колективні, комунальні та державні. Кожна з цих форм має свої особливості управління та фінансування, але всі вони мають відповідати єдиним стандартам якості обслуговування та безпеки.

Для забезпечення належного функціонування аптеки повинні відповідати низці організаційних вимог. Це включає наявність необхідних приміщень та обладнання, кваліфікований персонал, дотримання умов зберігання лікарських засобів і наявність плану дій щодо вилучення неякісних ліків.

Організаційна структура аптеки зазвичай включає керівника, завідувача та його заступників, бухгалтера, фармацевтів і провізорів, а також обслуговуючий персонал. Кожен співробітник виконує свої специфічні функції, забезпечуючи ефективну роботу аптеки та якісне обслуговування клієнтів.

1.2 Поточний рівень автоматизації бізнес-процесів

В основі ефективного функціонування аптеки лежить використання персоналом низки найважливіших інформаційних ресурсів. Центральне місце серед них посідає всеукраїнська база даних постачальників фармацевтичних препаратів і супутньої продукції, інтегрована в інформаційно-маркетингову платформу LIKIS. Для внутрішнього обліку лікарських засобів і медичних виробів застосовується власна база даних аптеки. Доступ до інформації про поточні складські запаси здійснюється через програмний комплекс «1MASTER:Бухгалтерія», що гарантує актуальність і достовірність відомостей про наявність медикаментів.

Для опрацювання документації аптекою використовується пакет прикладних програм MS Office, включно з MS Excel і MS Word. Важливо зазначити, що в аптеці відсутня клієнтська база, оскільки відпуск товарів відбувається у відділенні аптеки наживо, без збору даних про відвідувачів. Це спрощує роботу з клієнтами, але також створює певні складнощі у відстеженні купівельних переваг та історії замовлень.

Однією з ключових проблем у процесі відпуску медикаментів є рукописні призначення лікарів. Пацієнти приходять в аптеку з паперовими рецептами,

написаними від руки, які часто бувають нерозбірливими. Це вимагає від фармацевтів значних зусиль і часу для розшифровки тексту, що уповільнює процес обслуговування.

Для розв'язання цих проблем було вирішено спроектувати та розробити веб-орієнтовану інформаційну систему обліку лікарських призначень. Дана система сприятиме зменшенню часу обслуговування одного клієнта на касі, що збільшить пропускну спроможність усіх кас і дасть змогу за той самий час обслуговувати більше клієнтів. У результаті, це призведе до збільшення потоку клієнтів до аптек, що використовують дану систему, завдяки зручності взаємодії між клієнтами і фармацевтами та підвищенню довіри до таких аптек.

Система полегшить процес отримання, зберігання і використання лікарських призначень для пацієнтів, роблячи його більш зручним і доступним. Лікарі зможуть виписувати призначення комфортніше, вводячи текст за допомогою клавіатури, що дасть змогу одразу виправляти помилки та змінювати препарати до внесення даних у базу. Також можлива виписка призначень без безпосередньої присутності пацієнта, за умови попереднього огляду.

Важливим аспектом є надійність зберігання призначень. Усі дані про призначення пацієнтів зберігатимуться в базі даних на захищеному вебсервері, що виключає ризик втрати паперових рецептів. Зручність доступу до призначень також значно зростає, оскільки на сьогоднішній день смартфон є обов'язковим пристроєм для більшості населення. Доступ до призначень можна отримати через смартфон, комп'ютер, ноутбук або планшет, за наявності підключення до інтернету. У разі відсутності цифрових пристроїв, призначення можна роздрукувати на папері або знайти на касі в аптеці за ПІБ клієнта.

Створення веб-орієнтованої системи обліку лікарських призначень має безліч переваг. Друковані літери завжди будуть зрозумілими як для пацієнта, так і для фармацевта, що виключає можливість помилок під час розшифровки почерку лікаря. Користувачі охочіше звертатимуться до аптек, де встановлено систему, адже фармацевти зможуть швидко знаходити призначення та обробляти їх, що збільшить пропускну спроможність каси, а отже, кількість клієнтів і рівень

доходів аптеки.

Використання сучасних веб-технологій забезпечить високу швидкість роботи і зрозумілість інтерфейсу системи. У перспективі можна додавати нові функції, як-от інтеграція системи обліку лікарських призначень із локальною мережею аптеки, що дасть змогу бачити наявність або відсутність препаратів у реальному часі.

Таким чином, створення веб-орієнтованої системи обліку лікарських призначень є обґрунтованим і значним кроком для покращення роботи аптек та зручності клієнтів. Переваги її створення та використання очевидні та значні, що робить цей проєкт перспективним і затребуваним.

1.3 Визначення ключових вимог для веб-орієнтованої системи

Процес створення веб-орієнтованої системи обліку лікарських призначень починатиметься з чіткого визначення ключових вимог, які визначатимуть як функціональні, так і нефункціональні аспекти системи. Ці вимоги формуються на основі аналізу наявних проблем в аптечних бізнес-процесах і потреб кінцевих користувачів системи.

Веб-орієнтована облікова система буде розроблена у формі вебдодатка, що дасть змогу користувачам взаємодіяти з нею через браузер. Таке рішення підходить для виконання різноманітних завдань і може використовуватися як великими компаніями, так і окремими особами [5, 6]. Однією з основних переваг вебдодатків є можливість одночасного доступу безлічі користувачів. Користувачам не потрібно встановлювати додаткове програмне забезпечення на свої пристрої, що зробить систему доступною на різних платформах, включно з комп'ютерами, ноутбуками, мобільними пристроями та різними операційними системами. Крім того, система підтримуватиме роботу через різні браузери і забезпечуватиме можливість використання декількох акаунтів одночасно [7].

Під час розроблення системи враховуватимуть законодавчу базу України, документацію та регламенти підприємств, а також базу даних зареєстрованих користувачів і введені ними дані. Ключовими користувачами системи стануть медичні фахівці (лікарі), аптечні працівники (фармацевти) і пацієнти. Медичні фахівці використовуватимуть систему для створення та управління рецептами, аптечні працівники - для перегляду та обробки рецептів, а пацієнти - для отримання та зберігання своїх електронних рецептів [8].

Система буде доступна тільки для зареєстрованих користувачів, причому функціонал буде обмежений залежно від ролі користувача. Це забезпечить безпеку даних і точний розподіл завдань серед різних категорій користувачів. Для медичних фахівців система запропонує функції пошуку пацієнтів і створення рецептів. Аптечні працівники зможуть шукати пацієнтів і переглядати рецепти, а пацієнти - переглядати свої рецепти.

Вебдодаток складатиметься з кількох ключових компонентів: форма реєстрації з введенням персональних даних, форма авторизації з введенням облікових даних, особистий кабінет з можливістю редагування персональних даних, форма для створення рецепта з автозаповненням даних лікаря та пацієнта, форми для перегляду рецептів пацієнтів та онлайн-мапа для пошуку найближчих закладів, де встановлена система. Ці компоненти забезпечать зручний та інтуїтивно зрозумілий інтерфейс для всіх категорій користувачів, даючи змогу легко та ефективно керувати лікарськими призначеннями.

Визначення та деталізація цих вимог дасть змогу розробити веб-орієнтовану систему, яка відповідатиме потребам усіх користувачів, забезпечуючи високу продуктивність, надійність і безпеку в роботі з лікарськими призначеннями.

Головна сторінка вебдодатку міститиме інформацію про переваги використання системи для різних категорій користувачів. Лікарі зможуть прискорити процес призначення препаратів, вести електронні записи та отримувати доступ до даних про пацієнтів. Для фармацевтів система спростить процес перевірки призначень і зменшить імовірність помилок під час відпуску ліків. Пацієнти отримають доступ до списку своїх призначень, зможуть шукати

найближчі аптеки та використовувати електронні рецепти.

Головна сторінка міститиме інфографіку, що показуватиме переваги системи, скріншоти інтерфейсу для різних типів користувачів і схеми взаємодії між лікарями, фармацевтами та пацієнтами.

Сторінка «Довідк» надасть детальну інформацію про умови та правила використання системи. Тут будуть описані процеси реєстрації та авторизації, інструкції з використання особистого кабінету, а також правила та умови використання системи. Також буде розміщено інформацію про тарифні плани для лікарів і аптек, з можливістю безкоштовного використання для пацієнтів.

Футер сайту міститиме контактну інформацію, включно з адресою розробника, телефоном та email для зв'язку, посиланнями на соціальні мережі та месенджери для підтримки. У цьому розділі також буде вказана інформація про компанію-розробника, рік випуску та копірайт.

Хедер сайту забезпечить перемикання мови інтерфейсу (українська, англійська, російська) і зміну зовнішнього вигляду (світла і темна теми).

Аналіз аналогічних систем, наприклад таких як «Доступні ліки» та «Tabletki.ua», показав, що перша система пропонує лікарям можливості для виписування рецептів, але обмежує доступ пацієнтів до електронних рецептів, тоді як друга система дає змогу пацієнтам створювати списки препаратів і замовляти ліки, але обмежує можливості лікарів у виписуванні призначень [9,10]. Веб-орієнтована облікова система поєднуватиме функціональність обох систем, пропонуючи мінімалістичний і зручний інтерфейс без великої кількості реклами та другорядних функцій [11].

Функціональні вимоги визначають поведінку системи, включно з користувацькими історіями користувачів (user story) і сценаріями використання (use cases). Нефункціональні вимоги включають властивості продукту, що забезпечують зручність використання системи таких як безпека, масштабованість, кросбраузерність, надійність, доступність, зручність використання, продуктивність, швидкість відновлення і локалізація.

Основні ролі користувачів включають пацієнта, лікаря і фармацевта.

Таблиця функціональних вимог включатиме можливості реєстрації та авторизації користувачів, підтвердження реєстрації через email і доступ до особистого кабінету з можливістю редагування даних [12, 13].

Для лікарів і фармацевтів функціонал включатиме наскрізний пошук пацієнтів, введення призначень у спеціальну форму та перегляд останніх призначень пацієнта. Пацієнти зможуть переглядати найближчі аптеки на онлайн-карті, редагувати список необхідних препаратів і масштабувати карту для точного визначення місця розташування аптек [14].

Кросфункціональні вимоги забезпечуватимуть безпеку даних користувачів, масштабованість і продуктивність системи, кросбраузерну сумісність і надійність, зручність використання і швидкість відновлення, а також можливість вибору мови інтерфейсу.

2 КОНЦЕПТУАЛЬНЕ ПРОЄКТУВАННЯ ВЕБДОДАТКУ

2.1 Проєктування бізнес-процесів

Проєктування бізнес-процесів у створенні вебдодатка являє собою створення моделі, де графічно відображаються інформаційні потоки і пов'язані з ними підпроцеси. Графічне представлення бізнес-процесів дає змогу візуалізувати й аналізувати складні робочі процеси, що сприяє їхній оптимізації та автоматизації.

Таке подання охоплює події та активності, які відбуваються в рамках робочого процесу. Воно відображає виконавців та ініціаторів цих подій, показуючи, хто відповідає за виконання конкретних завдань. Також графічне представлення включає точки прийняття рішень і різні шляхи, якими можуть слідувати робочі процеси залежно від цих рішень [15].

Крім того, моделі бізнес-процесів відображають пристрої, засоби та документи, що беруть участь у процесах. Хронометраж процесу і його окремих

етапів дає змогу оцінити часові витрати на виконання завдань. Показники успіхів і невдач процесу допомагають виявити сильні та слабкі сторони й визначити сфери, що потребують поліпшення.

Під час розроблення вебсистем ключову роль відіграє вивчення та візуалізація бізнес-процесів. Важливо точно виявити, які елементи процесів - від вхідних даних до кінцевих результатів - піддаються автоматизації та оптимізації. Для цієї мети існує безліч методик, що поєднують текстові описи та графічні інструменти [16]. У сучасній практиці найбільш популярними способами представлення бізнес-процесів вважаються:

- Data Flow Diagram (DFD);
- Unified Modeling Language (UML);
- Integrated Definition for Function Modeling (IDEF0, IDEF3);
- Business Process Modeling Notation (BPMN).

2.1.1 Data Flow Diagram

Data Flow Diagram (DFD) являє собою графічний інструмент, який використовується для відображення потоку даних через систему. DFD показує, як дані входять у систему, як вони обробляються і як виходять із системи. Основними елементами DFD є процеси, потоки даних, сховища даних і зовнішні сутності. Серед переваг можна визначити:

- простота і зрозумілість;
- DFD дає змогу детально розглянути, як дані переміщуються і трансформуються в системі.

Недоліки:

- обмежені можливості для моделювання складних бізнес-логік: DFD фокусуються на потоках даних і не завжди підходять для моделювання складних логічних процесів;
- відсутність стандартів для детального моделювання.

2.1.2 Unified Modeling Language

Unified Modeling Language (UML) є стандартною мовою для специфікації, візуалізації та документування компонентів програмних систем. UML включає

безліч діаграм, таких як діаграми класів, об'єктів, станів, послідовностей і взаємодій.

Переваги:

- UML надає різні типи діаграм для моделювання різних аспектів системи;
- UML є визнаним стандартом, що полегшує спілкування між розробниками та іншими учасниками проєкту.

2.1.3 Language Integrated Definition for Function Modeling (IDEF0, IDEF3)

Integrated Definition for Function Modeling (IDEF0, IDEF3) - це методологія моделювання, що використовується для представлення функцій системи та їхніх взаємодій. IDEF0 фокусується на моделюванні функцій та їхніх взаємозв'язків, а IDEF3 використовується для моделювання процесів та їхньої часової послідовності.

Переваги:

- IDEF0 дає змогу детально аналізувати функції та їхні взаємозв'язки;
- дає змогу моделювати часові послідовності та залежності процесів.

Недоліки: методи можуть бути менш гнучкими для адаптації до різних типів бізнес-процесів.

2.1.4 Business Process Modeling Notation

Business Process Modeling Notation (BPMN) - це стандарт для графічного представлення бізнес-процесів у формі діаграм. BPMN надає багатий набір символів і правил для моделювання складних процесів [17].

Переваги:

- BPMN дає змогу детально моделювати всі аспекти бізнес-процесів;
- BPMN широко використовується і підтримується багатьма інструментами для моделювання.

На рисунку 1.1 наведено діаграму очікуваного процесу функціонування веб-орієнтованої системи обліку лікарських призначень відповідно до BPMN методології. Ця діаграма наочно показує загальний процес функціонування системи, починаючи з моменту створення призначення лікарем до моменту його перегляду пацієнтом або фармацевтом. Використання BPMN дає змогу

стандартизувати символи, що полегшує розуміння карт процесів для керівництва, співробітників і консультантів.

BPMN був обраний з суттєвої переваги яка полягає у використанні стандартизованих символів, що робить діаграми більш зрозумілими для всіх учасників процесу. Це спрощує аналіз і проєктування процесів, а також покращує комунікацію між різними підрозділами та зацікавленими сторонами. Порівняно з діаграмами потоків даних або діаграмами активності UML, BPMN забезпечує більш просте представлення інформації та краще підходить для проєктування процесів [17].

Для моделювання потоків даних часто використовують метод DFD (Data Flow Diagrams). DFD дає змогу графічно уявити процеси передавання даних в інформаційній системі, описуючи, як дані переміщуються від вхідних джерел через процеси та сховища даних до формування звітів [18]. Приклади декомпозиції процесів включають такі завдання, як «Створення призначення для пацієнта» і «Перегляд призначень пацієнта». На рисунках 1.2 і 1.3 представлено відповідні DFD діаграми, які деталізують ці процеси.

Важливість аналізу потоків даних і їх коректного моделювання не може бути переоцінена. Правильно побудовані діаграми DFD і BPMN забезпечують повне розуміння системи, що є основою для успішного проєктування і впровадження веб-орієнтованої системи обліку лікарських призначень.

2.2 Архітектурна модель і методи реалізації системи

Розроблена вебсистема автоматизації процесу документообігу лікарських призначень включає інформаційну та програмну архітектуру. В основі системи лежить пірамідальна структура, де на вершині розташована домашня сторінка. Ця структура забезпечує основний доступ до всіх розділів сайту, спрощуючи навігацію і покращуючи користувацький досвід.

Для реалізації системи використовуються сучасні та надійні технології. Серверна частина побудована на PHP, а для зберігання даних обрано базу даних MySQL. Клієнтська частина розроблена з використанням HTML, CSS і JavaScript. Ці технології були обрані завдяки їхній популярності, надійності та можливості інтеграції з наявними медичними системами [19, 20, 21].

Процес розробки структури сайту починається з ретельної оцінки контенту і функціональних блоків. Для візуального представлення сторінок використовуються wireframes, які створюють у застосунку Figma. Wireframes допомагають спланувати розташування елементів на сторінці, забезпечуючи інтуїтивність і зручність використання.

Головна сторінка сайту призначена для привернення уваги користувачів і спонукання їх до реєстрації. Вона містить візуальні елементи, такі як зображення та іконки, а також коротку інформацію про можливості сайту. Після авторизації користувача, навігаційне меню головної сторінки оновлюється, надаючи доступ до особистих даних і функціональності, що відповідає ролі користувача.

Ключовими сторінками системи є профіль користувача, створення призначення для лікарів, перегляд призначень для фармацевтів і пацієнтів, а також пошук аптек. Профіль користувача надає персональну інформацію та налаштування, створення призначення дає змогу лікарям вводити та редагувати лікарські призначення, перегляд призначень забезпечує фармацевтам і пацієнтам доступ до історії призначень, а пошук аптек допомагає пацієнтам знайти найближчі аптеки та перевірити наявність ліків.

Особливості дизайну сайту спрямовані на забезпечення зручності використання та пошукової оптимізації. Хедер містить навігацію, логотип, кнопки входу/виходу та налаштування профілю, а футер містить інформацію про розробників і контакти. Дизайн орієнтований на інтуїтивну взаємодію користувача з системою, що підвищує його задоволеність і ефективність використання.

Система передбачає кілька користувацьких ролей, кожна з яких має свої права та можливості. Лікарі мають доступ до функцій створення та редагування лікарських призначень, фармацевти можуть переглядати призначення та відмічати виконання рецептів, а пацієнти мають доступ до своїх призначень та інформації про ліки.

Розробку системи вели з урахуванням зручності використання та пошукової оптимізації. Це передбачає використання адаптивного дизайну, забезпечення швидкого завантаження сторінок і оптимізацію контенту для пошукових систем. Інтеграція з наявними медичними системами здійснюється через API, що дає змогу використовувати вже наявні дані та розширювати функціональність системи без необхідності кардинальних змін в наявній інфраструктурі.

Розробку системи вели з урахуванням зручності використання та пошукової оптимізації. Це охоплює використання адаптивного дизайну, забезпечення швидкого завантаження сторінок і оптимізацію контенту для пошукових систем. Інтеграція з наявними медичними системами здійснюється через API, що дає змогу використовувати вже наявні дані та розширювати функціональність системи без необхідності кардинальних змін в наявній інфраструктурі [22, 23].

Використання CMS для створення оригінального вебдодатку було відхилено з кількох причин. По-перше, типові шаблони обмежують можливості для створення унікального користувацького інтерфейсу. По-друге, функціональність таких систем часто буває обмеженою, що перешкоджає реалізації специфічних вимог проєкту. Також CMS можуть містити надлишкові модулі, які не використовуються, але займають ресурси і знижують продуктивність системи. До того ж, CMS нерідко виявляються вразливими до

різних атак, що ставить під загрозу безпеку даних користувачів [24].

Для розробки та тестування вебдодатку було обрано кілька популярних браузерів: Google Chrome, Opera, Edge, Apple Safari та Mozilla Firefox. Це забезпечує сумісність і коректне відображення вебдодатка на різних пристроях і платформах.

Технології, використані для розробки фронтенду, включають HTML5, CSS3, Bootstrap 4 (для адаптивної верстки) і JavaScript із фреймворком jQuery. Ці технології дають змогу створювати сучасний, динамічний і зручний інтерфейс, що адаптується до різних розмірів екранів і забезпечує зручну взаємодію користувача з системою [25, 26].

Для розробки бекенда використовувався PHP, який обробляє серверні запити і взаємодіє з базою даних MySQL. PHP-скрипти відправляють запити до бази даних, обробляють результати і передають дані на фронтенд [27]. Це дає змогу реалізувати складну логіку застосунку та забезпечувати надійне зберігання даних.

Структурно-функціональна схема системи включає кілька ключових компонентів. Запити до бази даних здійснюються через PHP-скрипти, які обробляють запити і відправляють відповіді на фронтенд. Дані передаються між сервером і фронтендом, забезпечуючи взаємодію користувача з системою в реальному часі.

Ключовими компонентами системи є вебсервер, протоколи HTTP, SMTP і FTP, а також поштовий сервер для підтвердження реєстрації та надсилання призначень. Вебсервер обробляє HTTP-запити і надсилає відповіді користувачам, поштовий сервер використовується для надсилання повідомлень і підтверджень електронною поштою, забезпечуючи надійну комунікацію між системою і користувачами [28, 29].

Поштові протоколи SMTP використовуються для надсилання повідомлень, а протоколи POP3 і IMAP для отримання повідомлень [31]. Це дає змогу системі ефективно керувати електронними повідомленнями та забезпечувати оперативне сповіщення користувачів про призначення та інші важливі події [32].

Таким чином, архітектурна модель і методи реалізації розробленої вебсистеми дають змогу ефективно автоматизувати процес документообігу лікарських призначень, забезпечуючи надійне зберігання даних і доступ до них у будь-який час.

3 РЕАЛІЗАЦІЯ ПРОТОТИПУ ВЕБСИСТЕМИ

3.1 Етапи створення вебдодатка

Якісна розробка вебдодатка вимагає чіткого алгоритму дій, що забезпечує послідовне і структуроване виконання всіх етапів. Процес створення вебдодатка включає кілька ключових етапів, кожен з яких відіграє важливу роль у досягненні кінцевого результату. Важливим кроком є збір даних і формулювання вимог до системи [37, 36]. На цьому етапі проводиться аналіз потреб користувачів і функціональних можливостей, які має надати система. Результатом є створення Технічного завдання (ТЗ), ключового документа, що описує мету і параметри проєкту. ТЗ слугує основою для всіх наступних етапів розроблення, забезпечуючи чітке розуміння завдань і очікувань усіх учасників процесу.

Розробка wireframes і макетів допомагає узгодити бачення замовника і розробника, надаючи візуальне уявлення структури і функціональності майбутнього вебдодатка. Ці елементи дають змогу визначити основні компоненти користувацького інтерфейсу та їхні взаємозв'язки. Front-end розробка фокусується на створенні користувацького інтерфейсу, забезпечуючи інтерактивність і зручність використання системи. На цьому етапі використовуються сучасні веб-технології, як-от HTML, CSS і JavaScript, для створення всіх візуальних елементів, як-от кнопки, форми, меню та інші елементи інтерфейсу [38].

Back-end розробка забезпечує роботу серверної частини застосунку та зв'язок із базою даних. Цей етап охоплює створення логіки застосунку, управління даними, обробку запитів користувачів і забезпечення безпеки системи [39,40]. Важливо також приділити увагу інтеграції системи з реляційною базою даних, яка

зберігає дані про користувачів, установи, призначення та інші ключові елементи системи. Вебдодаток обліку лікарських призначень використовує бази даних, такі як таблиці Cities, Roles, InstitutionTypes, Institutions, Users і Recipes. Кожна таблиця має свою структуру і зв'язки, що забезпечують цілісність даних. Наприклад, таблиця Users містить інформацію про зареєстрованих користувачів, включно з їхніми іменами, контактними даними та хешованими паролями для підвищення безпеки [40].

Тестування проводиться для перевірки відповідності системи вимогам, описаним у ТЗ. Тестування включає функціональні тести, які перевіряють працездатність усіх функцій системи, і нефункціональні тести, які оцінюють продуктивність, безпеку та зручність використання програми. Реєстрація доменного імені є необхідним кроком для ідентифікації сайту в інтернеті. Доменне ім'я має бути унікальним і легко запам'ятовуваним, що дає змогу користувачам швидко знаходити та ідентифікувати сайт. Веб-хостинг забезпечує доступність вебдодатка в інтернеті, надаючи серверні ресурси для зберігання та управління всіма файлами сайту.

Перенесення сайту на хостинг робить його доступним користувачам цілодобово. Вибір надійного хостинг-провайдера відіграє ключову роль у забезпеченні стабільної роботи та доступності вебдодатка. Підтримка й оновлення вебдодатка можуть знадобитися після запуску системи. Цей етап охоплює регулярне оновлення контенту, виправлення помилок, поліпшення функціональності та забезпечення безпеки системи, що дає змогу підтримувати високу якість і актуальність вебдодатка в довгостроковій перспективі [41].

Послідовне виконання всіх етапів, починаючи від збору даних і закінчуючи підтримкою та оновленням системи, забезпечує створення якісного продукту. Кінцевим результатом є веб-орієнтована система обліку лікарських призначень, що автоматизує процес документообігу, забезпечує надійне зберігання і доступ до даних та інтегрується з наявними медичними системами, підвищуючи точність і ефективність призначення та відпуску лікарських засобів [42].

Алгоритм створення вебдодатку включає такі етапи: збір даних і

формулювання вимог, створення ТЗ, розробка wireframes і макетів сайту, front-end розробка, back-end розробка, тестування функціоналу, реєстрація доменного імені, перенесення сайту на хостинг, підтримка і супровід. Технічне завдання є ключовим документом, що визначає завдання і вимоги до вебсайту. Front-end розробка фокусується на призначеному для користувача інтерфейсі та взаємодії з користувачем, тоді як back-end розробка забезпечує працездатність системи та зв'язок із базою даних [43]. Тестування проводиться для перевірки відповідності системи вимогам, а реєстрація доменного імені необхідна для ідентифікації сайту в інтернеті. Веб-хостинг забезпечує доступність вебдодатка в інтернеті, а підтримка та супровід вебдодатка може тривати тривалий час після запуску.

3.2 Реалізація вебсистеми

Розробка прототипу вебсистеми починається з проєктування її інформаційного забезпечення. Важливою частиною цього процесу є визначення типів вебсторінок, які будуть використовуватися в системі. Вебсторінки поділяються на статичні та динамічні. Статичні сторінки містять фіксовану інформацію, яка рідко змінюється. Вони простіші в розробці та потребують менших обчислювальних ресурсів. Динамічні сторінки, навпаки, генеруються на сервері в реальному часі на основі запитів користувачів і містять актуальну інформацію, яка може часто оновлюватися. Динамічні сторінки є основою інтерактивних вебдодатків, тому що вони дають змогу користувачам взаємодіяти з системою в режимі реального часу (рисунок 3.1).

Статичні сторінки	Динамічні сторінки
Вміст сторінки не змінюється і є однаковим щоразу після завантаження сторінки	Контент генерується за певною логікою і регулярно змінюється
Для розробки використовується лише HTML код	В додачу до HTML використовуються також серверні мови програмування
Незалежно від запиту користувача сторінка надсилає однакову відповідь	Може генерувати інший HTML код залежно від запиту
Вміст оновлюється лише коли хтось оновлює та публікує оновлений файл сторінки	Серверний код дозволяє генерувати унікальний вміст під час завантаження сторінки
Гнучкість є основною перевагою статичного веб-сайту	Основною перевагою динамічного веб-сайту є система управління контентом (CMS)

Рисунок 3.1 – Порівняльний аналіз статичних і динамічних вебсторінок

Для генерації динамічних сторінок необхідно використовувати реляційну базу даних (БД). Реляційна БД організовує дані у вигляді таблиць, пов'язаних між собою ключами, що дає змогу ефективно зберігати та керувати інформацією. У рамках розробленого вебдодатка для обліку лікарських призначень було створено базу даних, що складається з кількох таблиць: Cities, Roles, InstitutionTypes, Institutions, Users і Recipes. Кожна таблиця має первинний ключ (зазвичай "id"), який забезпечує унікальну ідентифікацію записів, а також може мати зовнішні ключі для зв'язку з іншими таблицями [33].

Таблиця Cities містить назви населених пунктів України. Ці дані необхідні для точного зазначення місця розташування медичних установ і користувачів системи (рисунок 3.2).

Номер атрибуту	Назва атрибуту	Тип даних	Обмеження		Призначення атрибуту
			Порожнє значення	Значення за замовчуванням	
1	City_id	int(10)	NOTNULL	–	Первинний ключ
2	CityName	varchar(100)	NOTNULL	–	Назва населеного пункту

Рисунок 3.2 – Модель даних таблиці Cities

Таблиця Roles зберігає інформацію про ролі користувачів системи, таких як лікар, пацієнт або фармацевт. Це дає змогу розмежовувати доступ до різних функціональних можливостей системи залежно від ролі користувача [34] (рисунок 3.3).

Номер атрибуту	Назва атрибуту	Тип даних	Обмеження		Призначення атрибуту
			Порожнє значення	Значення за замовчуванням	
1	Role_id	int(10)	NOTNULL	–	Первинний ключ
2	RoleName	varchar(50)	NOTNULL	–	Назва ролі

Рисунок 3.3 – Модель даних таблиці Roles

Таблиця InstitutionTypes включає типи установ, що використовують систему, такі як лікарні, клініки та аптеки. Це допомагає структурувати інформацію про різні типи медичних установ та їхні специфічні функції [35] (рисунок 3.4).

Номер атрибуту	Назва атрибуту	Тип даних	Обмеження		Призначення атрибуту
			Порожнє значення	Значення за замовчуванням	
1	InstType_id	int(10)	NOTNULL	–	Первинний ключ
2	TypeName	Varchar(50)	NOTNULL	–	Назва типу закладу

Рисунок 3.4 – Модель даних таблиці InstitutionTypes

Таблиця Institutions зберігає повну інформацію про медичні заклади, включно з їхньою назвою, типом, місцем розташування та контактними даними. Це забезпечує централізоване управління даними про установи, що беруть участь у системі [36] (рисунок 3.5).

Номер атрибуту	Назва атрибуту	Тип даних	Обмеження		Призначення атрибуту
			Порожнє значення	Значення за замовчуванням	
1	Inst_id	int(10)	NOTNULL	–	Первинний ключ
2	InstType	int(10)	NOTNULL	–	Зовнішній ключ
3	InstName	varchar(100)	NOTNULL	–	Повна назва закладу
4	City	int(10)	NOTNULL	–	Зовнішній ключ
5	Address	varchar(200)	NOTNULL	–	Адреса закладу

Рисунок 3.5 – Модель даних таблиці Institutions

Таблиця Users містить дані зареєстрованих користувачів, включно з їхніми

іменами, контактними даними та хешованими паролями. Хешування паролів використовується для підвищення безпеки та захисту даних користувачів від несанкціонованого доступу (рисунок 3.6).

Таблиця Recipes зберігає інформацію про виписані лікарські призначення, включно з текстом призначення, датою створення і даними про пацієнта та лікаря, який виписав рецепт. Ці дані критично важливі для забезпечення точного обліку та відстеження призначень (рисунок 3.7).

Номер атрибуту	Назва атрибуту	Тип даних	Обмеження		Призначення атрибуту
			Порожнє значення	Значення за замовчуванням	
1	User_id	int(10)	NOTNULL	–	Первинний ключ
2	FullName	varchar(100)	NOTNULL	–	ПІБ користувача
3	Email	varchar(70)	NOTNULL	–	Email користувача
4	Age	int(3)	NULL	NULL	Рік народження користувача
5	City	int(10)	NOTNULL	–	Зовнішній ключ
6	Password	varchar(100)	NOTNULL	–	Пароль
7	Role	int(10)	NOTNULL	–	Зовнішній ключ
8	Institution	int(10)	NOTNULL	–	Зовнішній ключ

Рисунок 3.6 – Модель даних таблиці Users

Номер атрибуту	Назва атрибуту	Тип даних	Обмеження		Призначення атрибуту
			Порожнє значення	Значення за замовчуванням	
1	Recipe_id	int(10)	NOTNULL	–	Первинний ключ
2	RecipeText	varchar(500)	NOTNULL	–	Текст призначення
3	Doctor	int(10)	NOTNULL	–	Зовнішній ключ
4	Patient	int(10)	NOTNULL	–	Зовнішній ключ
5	InstType	int(10)	NOTNULL	–	Зовнішній ключ
6	Institution	int(10)	NOTNULL	–	Зовнішній ключ
7	RecipeDate	date	NOTNULL	–	Дата створення призначення

Рисунок 3.7 – Модель даних таблиці Recipes

Усі таблиці бази даних пов'язані між собою через зовнішні ключі, що забезпечує цілісність і узгодженість даних. Наприклад, таблиця Recipes пов'язана з таблицями Users і Institutions через зовнішні ключі, що дає змогу відстежувати, хто виписав рецепт і в якій установі.

Структура бази даних дає змогу ефективно зберігати та використовувати інформацію для роботи веб-орієнтованої системи обліку лікарських призначень.

Вебсистема забезпечує інтерфейс для медичного та фармацевтичного персоналу, серверну частину для зберігання та обробки даних, а також механізм інтеграції з наявними медичними системами. Система може бути впроваджена в медичних установах і аптеках, що дасть змогу підвищити точність і ефективність призначення та відпуску лікарських засобів, а також забезпечити надійне зберігання і доступ до медичних даних.

У результаті проведеного аналізу технічної літератури та сучасних технологій розроблення вебдодатків було обрано найбільш підходящі методи та інструменти для створення цієї системи. Це включає використання фреймворків для розроблення веб-інтерфейсів, систем управління базами даних і засобів забезпечення безпеки даних. На основі обраних технологій і методів було розроблено архітектуру системи, яка забезпечує її масштабованість, надійність і безпеку.

3.3 Розробка керівництва з експлуатації

У процесі розробки системи особливу увагу приділено створенню докладної інструкції користувача, яка допомагає новим користувачам швидко освоїти функціонал системи. Інструкція містить точні вказівки щодо використання всіх можливостей вебсистеми, що забезпечує її легке та ефективне впровадження в робочий процес. Для зручності користувачів у системі створено спеціальну сторінку «Довідка», на якій детально описано мету і функції системи, а також доступні тарифні плани. Ця сторінка слугує важливим ресурсом для користувачів, даючи їм змогу швидко знаходити відповіді на запитання, що виникають, і отримувати необхідні роз'яснення.

Інтерфейс системи розроблено з урахуванням потреб різних категорій користувачів. До нього включено підказки (tooltips) і заповнювачі (placeholders), які допомагають користувачам правильно заповнювати форми й ефективно

використовувати можливості системи. Ці елементи інтерфейсу сприяють інтуїтивно зрозумілій взаємодії з системою, знижуючи ймовірність помилок і підвищуючи загальну задоволеність користувачів. Головна сторінка вебсистеми надає загальну інформацію про її призначення та переваги, що дає змогу новим користувачам швидко ознайомитися з ключовими функціями та зрозуміти, як система може допомогти в їхній роботі.

Навігаційне меню системи адаптовано під різні ролі користувачів, забезпечуючи зручний доступ до необхідних розділів і функцій. Система передбачає авторизацію та реєстрацію користувачів, що дає змогу захистити дані та забезпечити доступ до системи лише авторизованим особам. Під час реєстрації користувачі обирають свою роль (пацієнт, лікар або фармацевт) і заповнюють необхідні дані, що дає змогу системі адаптувати інтерфейс і функціонал під конкретні потреби користувача. Це робить використання системи більш цілеспрямованим і ефективним [45].

Пацієнти, які зареєструвалися в системі, можуть переглядати свої призначення та знаходити найближчі аптеки, що значно спрощує процес отримання необхідних лікарських засобів. Усі користувачі мають доступ до особистого кабінету, де вони можуть керувати своїми даними та налаштовувати систему під свої індивідуальні потреби. Лікарі та фармацевти можуть шукати пацієнтів через рядок пошуку, що спрощує процес роботи з медичними даними та дає змогу швидко знаходити потрібну інформацію.

Фармацевти можуть переглядати та роздруковувати призначення пацієнтів, що полегшує процес відпуску лікарських засобів і знижує ймовірність помилок. Лікарі, своєю чергою, мають можливість виписувати нові призначення обраним пацієнтам через спеціальну форму, яка автоматично заповнює деякі поля, що прискорює процес і знижує навантаження на медичний персонал.

Система управління лікарськими призначеннями побудована з урахуванням різних ролей користувачів, що робить її гнучкою та універсальною. Пацієнти, лікарі та фармацевти можуть використовувати її для виконання своїх професійних завдань, при цьому кожен користувач отримує доступ тільки до тих функцій, які

необхідні для виконання його обов'язків. У системі передбачено різні рівні доступу та захисту даних, що забезпечує надійність і безпеку використання [46].

Для всіх основних функцій системи надано програмний код у Додатку А, що дає змогу за потреби вносити зміни та адаптувати систему під специфічні вимоги медичного закладу або аптеки. Таким чином, розроблена система є потужним інструментом для автоматизації процесу управління лікарськими призначеннями, поліпшення якості медичних послуг і підвищення ефективності роботи медичного та фармацевтичного персоналу.

3.4 Економічна доцільність розробки вебдодатка

Оцінка економічних ефектів від впровадження вебсистеми управління лікарськими призначеннями передбачає аналіз як непрямих, так і прямих ефектів. Непрямі ефекти полягають у збільшенні клієнтського потоку, підвищенні довіри до медичного закладу, зростанні продуктивності та зручності для всіх учасників процесу. Це виражається в прискоренні обслуговування пацієнтів, поліпшенні пошуку інформації та підвищенні комфорту для співробітників і клієнтів. Усе це сприяє створенню більш ефективної та привабливої системи охорони здоров'я [47].

Прямі економічні ефекти оцінюються за допомогою різних економічних показників, таких як річний економічний ефект, коефіцієнт економічної ефективності капітальних витрат і період окупності. Для розрахунку річного економічного ефекту використовують спеціальну формулу з аналізу джерела, вигляд формули приведений на рисунку 3.8. Формула враховує різницю між сукупним доходом від використання системи та витратами на її впровадження й експлуатацію. Представлені показники: E_y – річний економічний ефект, S – годовая економія (річний приріст прибутку), C – капітальні вкладення, r_n – нормативний коефіцієнт окупності капітальних вкладень (для ІТ-сфери в межах 0,3-0,35) [49].

$$E_y = S - C \cdot r_n$$

Рисунок 3.8 – Формула для розрахунку річного економічного ефекту

Капітальні витрати на впровадження системи включають купівлю необхідного обладнання, налаштування та навчання персоналу. Як приклад, для аптеки капітальні витрати можуть включати закупівлю комп'ютерів для касових апаратів, налаштування цих комп'ютерів і навчання персоналу. На основі цих даних було проведено конкретні розрахунки.

Результати показали, що річний економічний ефект становитиме 94842,67 грн, коефіцієнт економічної ефективності - 2,54, а період окупності - близько 5 місяців. Ці дані вказують на економічну доцільність впровадження системи, оскільки вона окупиться менш ніж за півроку.

Для приватних медичних центрів впровадження системи управління лікарськими призначеннями також принесе значні переваги. Система дасть змогу скоротити час прийому пацієнтів, що призведе до збільшення кількості обслуговуваних пацієнтів і, відповідно, зростання доходів. Це сприяє поліпшенню якості медичних послуг і підвищенню конкурентоспроможності медичних центрів.

Державні установи, хоча й не отримують прямої економічної вигоди від впровадження системи, можуть розраховувати на значні непрямі ефекти. Підвищення комфорту пацієнтів і зростання задоволеності медичним обслуговуванням зміцнюють позиції Міністерства охорони здоров'я та підвищують рейтинги чинної влади. Впровадження такої системи демонструє прихильність держави до поліпшення якості медичних послуг і турботи про добробут громадян.

Таким чином, впровадження вебсистеми управління лікарськими призначеннями має широкий спектр позитивних ефектів як для приватних, так і для державних медичних закладів. Прямі економічні вигоди, такі як висока

економічна ефективність і швидкий період окупності, роблять систему привабливою для інвестицій. Непрямі вигоди, такі як поліпшення якості обслуговування та підвищення задоволеності пацієнтів, сприяють створенню більш ефективної та стійкої системи охорони здоров'я.

На рисунку 3.9 представлено інформація витрат на впровадження системи. На рисунк 3.10 представлено вхідні дані у вигляді таблиці для розрахунку і результати, що підтверджують економічну доцільність впровадження системи. Капітальні витрати на впровадження системи, що автоматизує процес документообігу лікарських призначень і дає змогу зберігати дані про призначення на вебсервері з можливістю доступу до них у будь-який час, для аптеки охоплюють закупівлю обладнання, налаштування комп'ютерів і навчання персоналу.

Найменування	Кількість(шт)	Вартість(грн)	Загальна вартість (грн)
Комп'ютер Artline Home H44V09 (AMD Ryzen 3 3200G (3.6 - 4.0 ГГц) / RAM 8 ГБ / SSD 240 ГБ / AMD Radeon Vega 8 / LAN / без ОС)	3	9681	29043
Інсталяція та налаштування ОС	3	500	1500
Навчання персоналу роботі із АІС	5	400	2000
Річна підписка на веборієнтовану систему обліку лікарських призначень	1	9900	9900
Разом	42443		

Рисунок 3.9 – Капітальні витрати на впровадження системи, що автоматизує процес документообігу лікарських призначень

Показник	Значення
Капітальні витрати (C)	42443 грн
Річний прибуток аптеки до впровадження системи	720000 грн
Очікуваний річний прибуток аптеки після впровадження системи	828000 грн
Сума річного приросту прибутку (S)	108000 грн
Нормативний коефіцієнт окупності капітальних вкладень (r_n)	0,31

Рисунок 3.10 – Вхідні дані для розрахунку узагальнюючих показників економічної ефективності

На закінчення можна зазначити, що впровадження вебсистеми управління лікарськими призначеннями є значним кроком вперед у модернізації процесів документообігу в медицині. Воно сприяє підвищенню ефективності роботи медичних установ, поліпшенню якості обслуговування пацієнтів і зміцненню позицій охорони здоров'я в цілому.

ВИСНОВКИ

У процесі виконання кваліфікаційної бакалаврської роботи було проведено детальний аналіз предметної області, що стосується документообігу лікарських призначень. Дослідження охоплювало всі аспекти взаємодії між лікарем, пацієнтом і фармацевтом, що дало змогу виявити слабкі сторони наявного процесу. Увагу було приділено як внутрішньому, так і зовнішньому середовищам аптек, а також їхнім бізнес-процесам, що допомогло отримати цілісне уявлення про поточний стан системи.

Під час аналізу було ухвалено рішення про створення системи, що автоматизує процес документообігу лікарських призначень і дає змогу зберігати дані про призначення на вебсервері з можливістю доступу до них у будь-який час. Було визначено функціональні та нефункціональні вимоги до системи, встановлено ролі користувачів і відповідні їм функціональні блоки, що стало фундаментом для подальшого розроблення. Діаграми функціонування системи, виконані в нотаціях BPMN і DFD, забезпечили візуалізацію і розуміння ключових процесів і взаємодій у системі.

Розробка архітектури вебдодатку та навігаційної схеми сайту дала змогу чітко визначити структуру майбутньої системи та забезпечити зручність її використання для всіх категорій користувачів. Створення wireframes – «каркасів» сторінок з основними елементами інтерфейсу, дало змогу попередньо оцінити зручність і функціональність користувацького інтерфейсу. Вибір технологій для розробки враховував сучасні тренди та рекомендації в галузі веб-розробки, що забезпечило використання найефективніших і найнадійніших рішень.

Уважне вивчення переваг і можливостей кожної обраної технології, їхньої сполучуваності, надійності та простоти застосування, дало змогу створити систему, що не тільки відповідає сучасним стандартам, а й забезпечує високу продуктивність і безпеку.

Таким чином, результатом роботи стала розробка комплексної системи, яка

автоматизує процес документообігу лікарських призначень, забезпечуючи зручний доступ до даних для всіх учасників процесу. Система включає в себе інтерфейс для медичного та фармацевтичного персоналу, серверну частину для зберігання та обробки даних, а також механізм інтеграції з існуючими медичними системами. Її впровадження в медичних установах і аптеках сприяє підвищенню точності та ефективності призначення та відпуску лікарських засобів, а також забезпечує надійне зберігання і доступ до медичних даних, що, своєю чергою, покращує якість обслуговування пацієнтів і знижує навантаження на медичний персонал.

На завершальному етапі роботи було проведено тестування та оцінка системи в реальних умовах, що підтвердило її відповідність поставленим цілям і вимогам. Розроблена система довела свою ефективність і зручність у використанні, отримавши позитивні відгуки від користувачів. Надалі можливе розширення функціоналу системи та її адаптація під специфічні потреби різних медичних установ, що дасть змогу ще більш ефективно використовувати її можливості та покращувати якість медичного обслуговування.

Беручи до уваги наведені результати, можна зробити висновок, що спочатку поставлену мету цього кваліфікаційного дослідження було успішно реалізовано, а всі сформульовані на початку роботи завдання було вирішено.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Global, regional, and national incidence, prevalence, and years lived with disability for 328 diseases and injuries for 195 countries, 1990–2016: a systematic analysis for the Global Burden of Disease Study 2016. The Lancet. 2017. Vol. 390, no. 10100. P. 1211–1259. URL: [https://doi.org/10.1016/S0140-6736\(17\)32154-2](https://doi.org/10.1016/S0140-6736(17)32154-2) (date of access: 12.04.2024).

2. В якій аптеці можна придбати ліки за електронним рецептом: у МОЗ дали роз'яснення. MedPlatforma. URL: <https://medplatforma.com.ua/news/62648-v-yakiy-apteci-mozhna-pridbati-liki-za-elektronnim-retseptom-u-moz-dali-roz'iasnennya> (дата звернення: 11.04.2024).

3. Доступні ліки 2024 р.: затверджено нові переліки граничних оптово-відпускних цін. Держлікслужба. URL: https://www.dls.gov.ua/for_subject/доступні-ліки-2024-р-затверджено-нові-пер/ (дата звернення: 11.05.2024).

4. Як виписувати електронний рецепт на «Доступні ліки»: інструкція для лікарів. URL: <https://moz.gov.ua/article/for-medical-staff/jak-vipisuvati-elektronnij-recept-na-dostupni-liki-instrukcija-dlja-likariv>

5. Luchaninov Y. Web application architecture: choosing the right type in 2024. MobiDev. URL: <https://mobidev.biz/blog/web-application-architecture-types#:~:text=Web%20application%20architecture%20is%20a,on%20the%20software%20development%20market.> (date of access: 15.05.2024).

6. Everything You Need to Know About Web Application Architecture. URL: <https://topdevs.org/blog/web-app-architecture-guide>

7. Як скористатися електронним рецептом програми «Доступні ліки»: алгоритм дій в аптеці. URL: <https://uozkmr.gov.ua/aktualno/uryadova-programa-dostupn-lki/1226-yak-skoristatisya-elektronnim-receptom-programi-dostupn-lki-algorithm-dy-v-aptec.html>

8. Tabletki.ua: Про компанію. URL: <https://tabletki.ua/uk/about/company/>

9. A Guide to Create an Exceptional Web Design and Usability. Medium. URL:

<https://medium.com/@indglobalindia25/a-guide-to-create-an-exceptional-web-design-and-usability-d760c7ef3177> (date of access: 21.05.2024).

10. Smack. How to write A website brief in 2022 (12 step guide). SMACK. URL: <https://smackagency.co.uk/blog/ux-design/how-to-write-a-website-brief/> (date of access: 10.05.2024).

11. The Use Case Technique -- An Overview. URL: https://portland.iiba.org/sites/portland/files/the_use_case_technique_-_an_overview.pdf

12. The Business Analysis Body Of Knowledge (BABOK) – In 3 Minutes. URL: <https://www.vanharen.net/blog/business-analysis-body-knowledge-babok-3-minutes/>

13. Ziemek M. Ocumenting non-functional requirements using FURPS+. Marcin Ziemek Blog. URL: https://marcinziemek.com/blog/content/articles/8/article_en.html (date of access: 19.05.2024).

14. Beaton C. What are functional and non-functional requirements?. Reuiment. URL: <https://www.reuiment.com/what-are-functional-and-non-functional-requirements/> (date of access: 20.05.2024).

15. What is Business Process Modeling and How to Implement it? URL: <https://www.cflowapps.com/business-process-modeling/>

16. Business Process Modeling: Definition, Benefits and How to. URL: <https://kissflow.com/workflow/bpm/business-process-modeling/>

17. What is Business Process Model and Notation (BPMN): Benefits and Examples. URL: <https://kissflow.com/workflow/bpm/what-is-bpmn/>

18. What is Data Flow Diagram? URL: <https://www.visual-paradigm.com/guide/data-flow-diagram/what-is-data-flow-diagram/>

19. Modern Web Application Architecture Explained: Components, Best Practices and More. URL: <https://litslink.com/blog/web-application-architecture>

20. What is Website structure? Why structure is important? URL: <https://www.marketing91.com/website-structure/>

21. What Exactly Is Wireframing? A Comprehensive Guide. URL: <https://careerfoundry.com/en/blog/ux-design/what-is-a-wireframe-guide/#what-is-a-wireframe>

22. What is Figma? (And How to Use Figma for Beginners). URL: <https://www.theme-junkie.com/what-is-figma/>
23. Головна сторінка. URL: <https://igroup.com.ua/seo-articles/holovna-storinka/>
24. Сайт на CMS, фреймворке или собственная разработка – что лучше? URL: <https://www.uplab.ru/blog/how-to-make-a-website/>
25. What Is a Web Browser? URL: <https://www.avast.com/c-what-is-a-web-browser#topic-2>
26. Frontend vs Backend. URL: <https://www.geeksforgeeks.org/frontend-vs-backend/>
27. What is PHP? The PHP Programming Language Meaning Explained. URL: <https://www.freecodecamp.org/news/what-is-php-the-php-programming-language-meaning-explained/>
28. DBMS. URL: <https://www.omnisci.com/technical-glossary/dbms>
29. What is MySQL? Everything You Need to Know. URL: <https://www.talend.com/resources/what-is-mysql/>
30. Definition of 'Web Server'. URL: <https://economictimes.indiatimes.com/definition/web-server>
31. Website. URL: <https://www.javatpoint.com/website-static-vs-dynamic>
32. What is a Relational Database (RDBMS)? URL: <https://www.oracle.com/database/what-is-a-relational-database/>
33. Понятие ER-модели. Понятие сущности (entity). Атрибуты. Виды атрибутов. URL: <https://www.bestprog.net/ru/2019/01/24/the-concept-of-er-model-the-concept-of-essence-and-communication-attributes-attribute-types-ru/>
34. What are the different types of relationships in DBMS? URL: <https://afteracademy.com/blog/what-are-the-different-types-of-relationships-in-dbms>
35. Реляционные базы данных. Понятие ключа. Виды ключей. Отношения. Главная и подчиненная таблицы. URL: https://www.bestprog.net/ru/2017/10/02/relational-databases-the-concept-of-a-key-types-of-keys-relations-master-table-and-detail-table_ru/

36. What is an algorithm? URL: <https://www.bbc.co.uk/bitesize/topics/z3tbwmn/articles/z3whpv4>
37. How to write technical task. URL: <https://encomage.com/2020/04/27/how-to-write-technical-task/>
38. What is Frontend Development? URL: <https://www.techslang.com/definition/what-is-frontend-development/>
39. How does software testing work? URL: <https://www.ibm.com/topics/software-testing>
40. What is domain name registration? URL: https://www.siteground.com/kb/domain_name_registration/
41. What is web hosting and why do you need it? URL: <https://www.techradar.com/news/what-is-web-hosting-and-why-do-you-need-it>
42. What is Website Maintenance and Why is It so Important for Your Business. URL: <https://www.exai.com/blog/website-maintenance-services-costs>
43. This is How to Build the Best User Documentation. URL: <https://www.techsmith.com/blog/user-documentation/>
44. Tooltip. URL: <https://www.computerhope.com/jargon/t/tooltip.htm>
45. Placeholder text. URL: <https://www.computerhope.com/jargon/p/plactext.htm>
46. Оцінка економічної ефективності Інтернет-ресурсів. URL: http://dipplus.com.ua/metodicheskiye-ukazaniya-i-informatsiya/article_post/otsinka-yekonomichnoi-yefektivnosti-internet-resursiv
47. Оцінка ефективності проєктів. URL: <https://buklib.net/books/22886/>
48. Показники та методики визначення ефективності планових заходів щодо підвищення організаційно-технічного рівня. URL: https://pidru4niki.com/92408/finansi/pokazniki_metodiki_viznachennya_efektivnosti_planovih_zahodiv_pidvischennya_organizatsiyno-tehnichnogo_rivnya
49. Методика визначення ефективності капітальних вкладень. URL: https://pidru4niki.com/1541010436255/ekonomika/metodika_viznachennya_efektivnosti_kapitalnih_vkladen

50. Термін окупності: формула. Вкладення і прибуток. URL:
<https://uk.campwaltblog.com/3983846-payback-period-formula-investment-and-profit>

ДОДАТОК А

Програмний код

Лістинг А.1- код HTML

```
body {  
    background-color: #FFFFFF;  
    margin: 0;  
    font-family: Arial, sans-serif;  
}  
  
.container {  
    width: 775px;  
    margin: 0 auto;  
    padding: 20px;  
}  
  
.header {  
    display: flex;  
    justify-content: flex-end;  
    background-color: #CBCFD2;  
    padding: 10px;  
}  
  
.header .menu a {  
    margin: 0 10px;  
    text-decoration: none;  
    color: #000;  
}  
  
.title img {  
    width: 100%;  
    height: auto;  
}  
  
.row {  
    display: flex;  
    flex-wrap: wrap;  
    margin: 20px 0;  
}  
  
.card {  
    flex: 1;  
    margin: 10px;  
    padding: 20px;  
    background-color: #f4f4f4;  
    border: 1px solid #ccc;  
    border-radius: 8px;  
    text-align: center;  
    transition: transform 0.2s;  
}
```

```

.card img {
    width: 100%;
    height: auto;
}

.card-content h3 {
    margin: 10px 0;
    font-size: 1.2em;
}

.card-content p {
    font-size: 0.9em;
    color: #666;
}

.read-more {
    display: inline-block;
    margin-top: 10px;
    padding: 8px 16px;
    background-color: #007BFF;
    color: #fff;
    text-decoration: none;
    border-radius: 4px;
}

.read-more:hover {
    background-color: #0056b3;
}

.footer {
    text-align: center;
    margin-top: 20px;
    font-size: 0.8em;
    color: #666;
}

.footer a {
    color: #007BFF;
    text-decoration: none;
}

.footer a:hover {
    text-decoration: underline;
}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Medical Company</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <center>

```

```

<div class="container">
  <div class="header">
    <div class="menu">
      <a href="#" class="b">Home</a> |
      <a href="#" class="b">Site Map</a> |
      <a href="#" class="b">Contacts</a>
    </div>
  </div>
  <div class="title">
    
  </div>
  <div class="row">
    <div class="card">
      
      <div class="card-content">
        <h3>Home</h3>
        <p>Welcome to our medical company.</p>
        <a href="#" class="read-more">read more</a>
      </div>
    </div>
    <div class="card">
      
      <div class="card-content">
        <h3>Company</h3>
        <p>Learn more about our company.</p>
        <a href="#" class="read-more">read more</a>
      </div>
    </div>
    <div class="card">
      
      <div class="card-content">
        <h3>Services</h3>
        <p>Discover our services.</p>
        <a href="#" class="read-more">read more</a>
      </div>
    </div>
    <div class="card">
      
      <div class="card-content">
        <h3>Products</h3>
        <p>Check out our products.</p>
        <a href="#" class="read-more">read more</a>
      </div>
    </div>
  </div>
  <div class="footer">
    <p>Copyright © Your Company Name</p>
    <p>Design by <a href="http://www.templatesbox.com/"
target="_blank" class="adv">Templates Box</a>. Free <a
href="http://www.wix.com/start/matrix"
class="adv">Website Builder</a>.</p>
  </div>
</div>
</center>

```

```
</body>
</html>
```

Лістинг А.2- код HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Medical Company</title>
  <link rel="stylesheet" href="styles.css">
  <script type="text/javascript">
    function newImage(arg) {
      if (document.images) {
        rslt = new Image();
        rslt.src = arg;
        return rslt;
      }
    }

    function changeImages() {
      if (document.images && (preloadFlag == true)) {
        for (var i = 0; i <
changeImages.arguments.length; i += 2) {
          document[changeImages.arguments[i]].src =
changeImages.arguments[i + 1];
        }
      }
    }

    var preloadFlag = false;
    function preloadImages() {
      if (document.images) {
        home_over = newImage("images/home-over.gif")
        company_over = newImage("images/company-
```

```

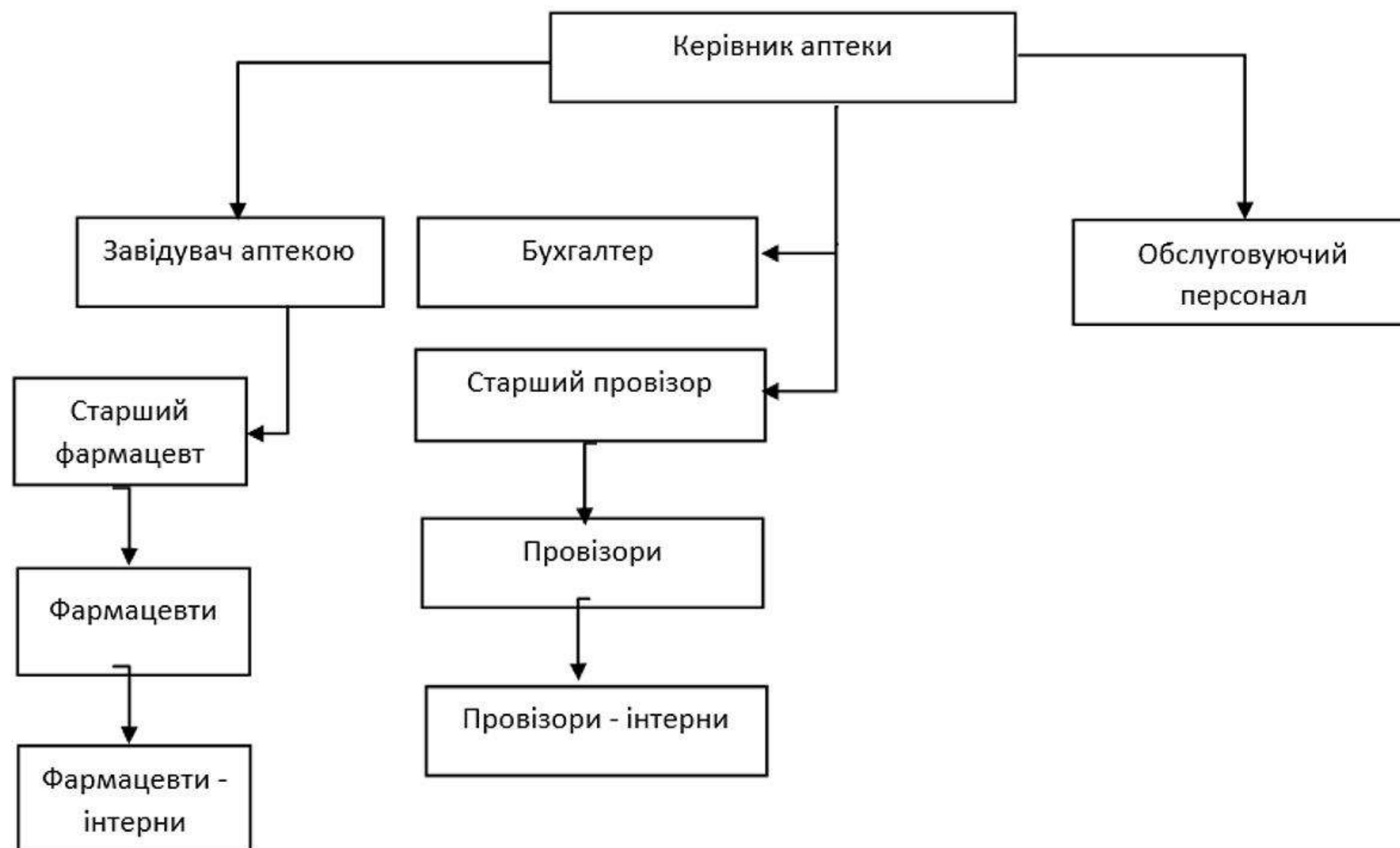
over.gif");
        services_over = newImage("images/services-
over.gif");
        products_over = newImage("images/products-
over.gif");
        preloadFlag = true;
    }
}
</script>
</head>
<body bgcolor="#FFFFFF" leftmargin="0" topmargin="0"
marginwidth="0" marginheight="0" onload="preloadImages();">
    <center>
        <!-- ImageReady Slices (index.psd) -->
        <table width="775" border="0" cellpadding="0"
cellspacing="0" class="frame">
            <tr>
                <td rowspan="12"></td>
                <td colspan="11"></td>
                <td colspan="4" align="center"
valign="middle" bgcolor="#CBCFD2" class="text5">
                    <a href="#" class="b">Home</a> | <a
href="#" class="b">Site Map</a> | <a href="#"
class="b">Contacts</a>
                </td>
                <td rowspan="12"></td>
                <td></td>
            </tr>
            <tr>
                <td colspan="7"></td>
                <td colspan="3">
                    <a href="#"
onmouseover="window.status='home'; changeImages('home',
'images/home-over.gif');"
                    return true;"
onmouseout="window.status=''; changeImages('home',
'images/home.gif'); return true;">
                    

```

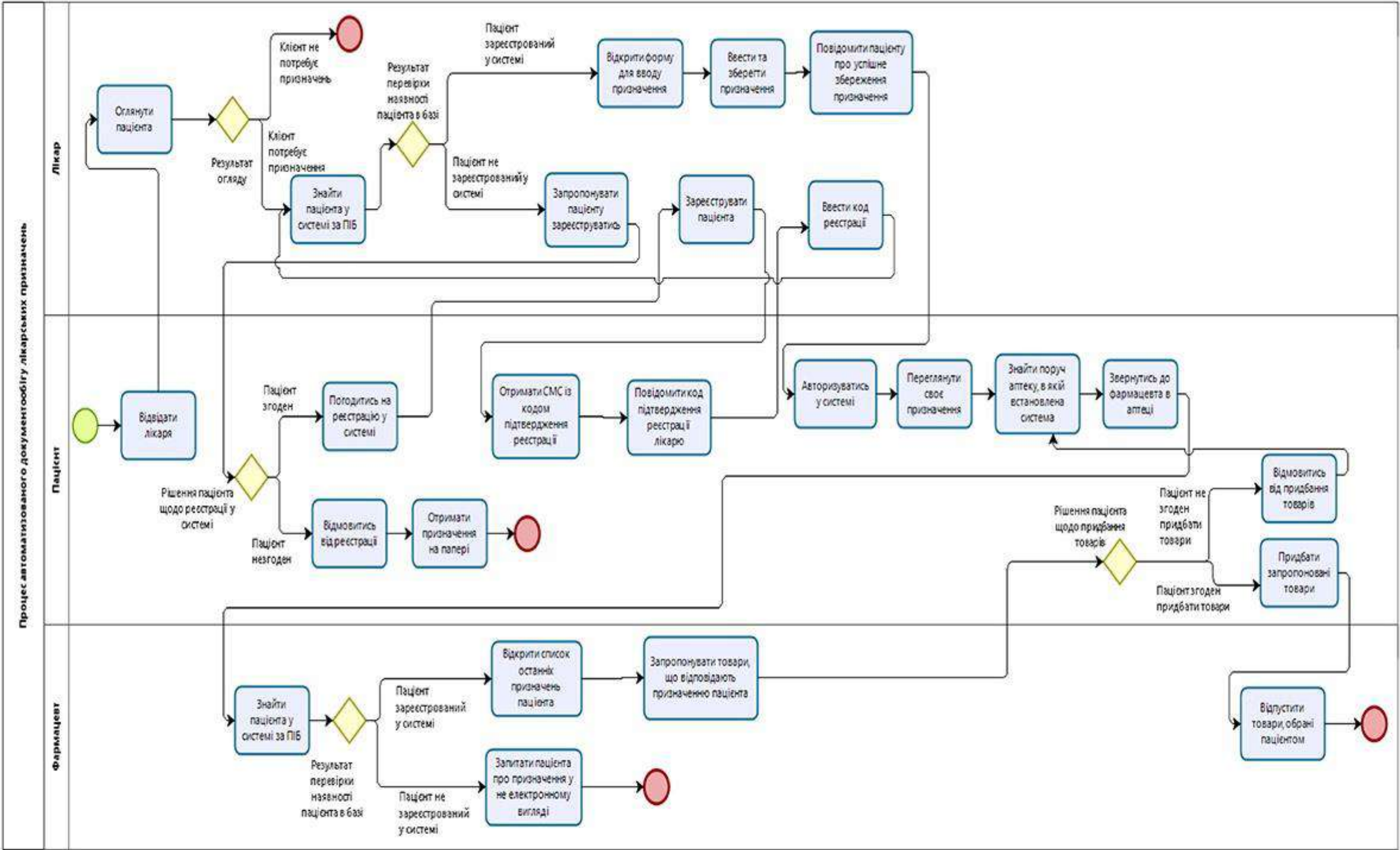
```

        </a>
    </td>
    <td colspan="2">
        <a href="#"
onmouseover="window.status='company'; changeImages('company',
'images/company-over.gif');
        return true;"
onmouseout="window.status=''; changeImages('company',
'images/company.gif'); return true;">
            
        </a>
    </td>
    <td colspan="2">
        <a href="#"
onmouseover="window.status='services';
changeImages('services', 'images/services-over.gif'); return
true;
        " onmouseout="window.status='';
changeImages('services', 'images/services.gif'); return
true;">
            
        </a>
    </td>
    <td>
        <a href="#"
onmouseover="window.status='products';
changeImages('products', 'images/products-over.gif'); return
true;
        " onmouseout="window.status='';
changeImages('products', 'images/products.gif'); return
true;">
            
        </a>
    </td>
    <td></td>
</tr>
<!-- Other rows... -->
</table>
<!-- End ImageReady Slices -->
</center>
</body>
</html>

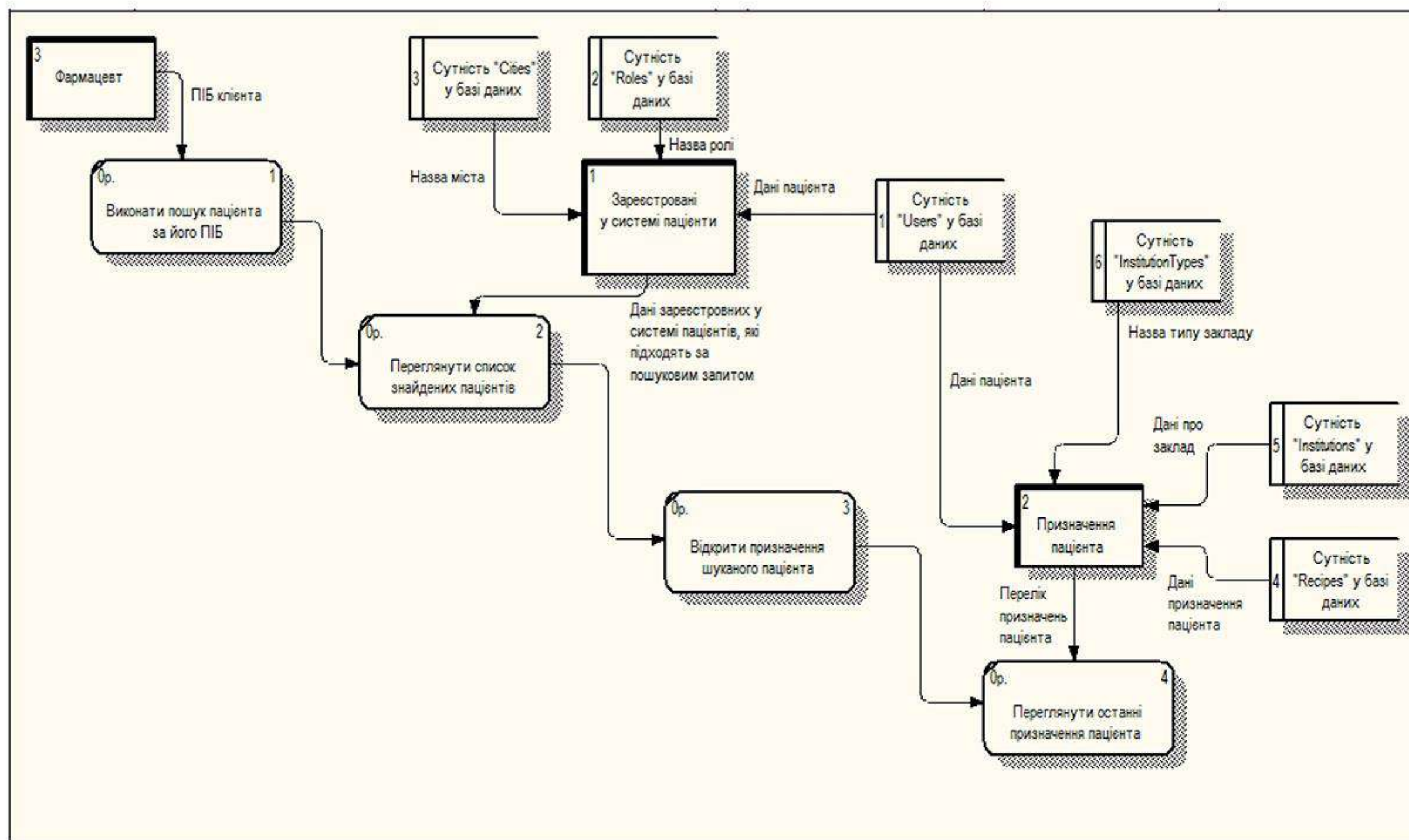
```



					13.02070849.00062 ПЛ 1						
					Розробка веб-орієнтованої системи обліку лікарських призначень Організаційна структура аптечного закладу	Літ.		Маса	Маштаб		
Зм.	Лист	№ докум.	Підп.	Дата							
Розроб.		Запемонцев А.С.	Засед								
Перевір.		Тягунова М.Ю.	Тягунова								
Т. контр.											
						Лист		Листів			
Н. контр.		Польська О.В.	Польська			НУ «Запорізька політехніка», гр. КНТЗ-511сн					
Затвердж.		Кудерметов Р.К.	Кудерметов								



					13.02070849.00062 ПЛ 2				
					Розробка веб-орієнтованої системи обліку лікарських призначень Діаграма очікуваного процесу функціонування системи	Лім.	Маса	Маштаб	
Зм.	Лист	№ докум.	Підп.	Дата					
Розроб.		Загемонцев А.С.	Зем						
Перевір.		Тягунова М.Ю.	Тяг						
Т. контр.									
						Лист	Листів		
Н. контр.		Польська О.В.	Польс			НУ «Запорізька політехніка», гр. КНТз-511сн			
Затвердж.		Кудерметов Р.К.	Куд						



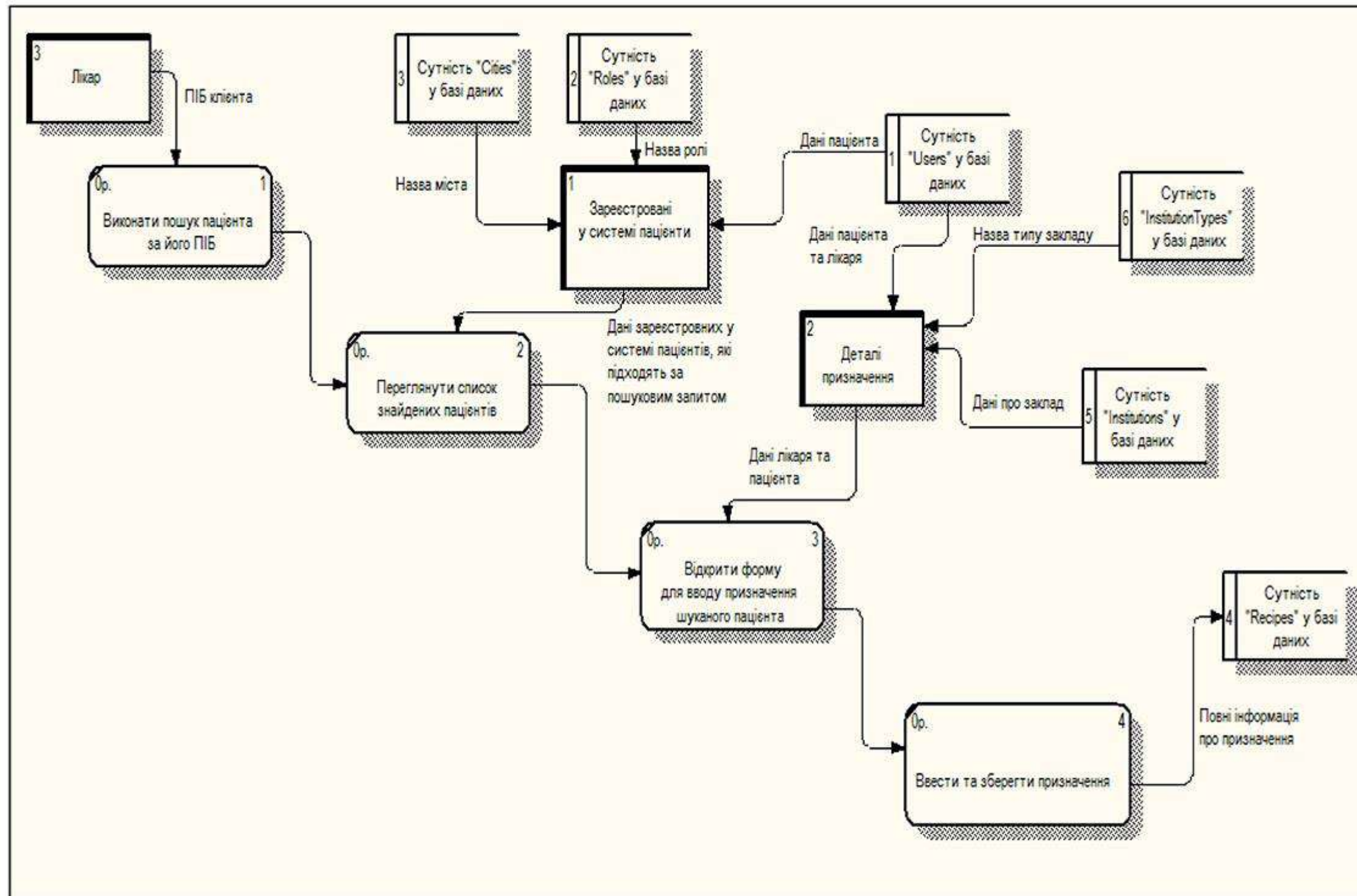
Зм.	Лист	№ докум.	Підп.	Дата
Розроб.	Запорожцев А.Є.	Заявка		
Перевір.	Тягунова М.Ю.			
Т. контр.				
Н. контр.	Польська О.В.			
Затвердж.	Кудеристов Р.К.			

13.02070849.00062 ПЛЗ

Розробка веб-орієнтованої
системи обліку лікарських
призначеньДекомпозиція процесу «Перегляд
призначення для пацієнта»

Літ.	Маса	Маштаб
Лист	Листів	

НУ «Запорізька
політехніка», гр. КНТЗ-511сн



					13.02070849.00062 ПЛ4				
					Розробка веб-орієнтованої системи обліку лікарських призначень Декомпозиція процесу «Створення призначення для пацієнта»	Лім.	Маса	Маштаб	
Зм.	Лист	№ докум.	Підп.	Дата					
Розроб.		Занемонцев А.С.	Занемонцев А.С.						
Перевір.		Тягунова М.Ю.	Тягунова М.Ю.						
Т. контр.						Лист	Листів		
Н. контр.		Польська О.В.	Польська О.В.			НУ «Запорізька політехніка», гр. КНТЗ-511сн			
Затвердж.		Кудерметов Р.К.	Кудерметов Р.К.						