

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА
Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці**

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»
галузь знань 12 «Інформаційні технології»
спеціальність 122 «Комп'ютерні науки»

Форма навчання: денна

КВАЛІФІКАЦІЙНИЙ БАКАЛАВРСЬКИЙ ПРОЕКТ

на тему

**РОЗРОБКА МЕДИЧНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ
АВТОМАТИЗАЦІЇ РОБОТИ РЕЄСТРАТУРИ**

здобувача Чуланової Мілени Олегівни _____

Науковий керівник:

ст. викладач

_____ Сквордякова О. О.

**Проект допущений до захисту
перед екзаменаційною комісією з
атестації здобувачів вищої освіти**

завідувач кафедри:

к.е.н., доцент

_____ Тішков Б.О.

Київ 2024

Міністерство освіти і науки України

**Київський національний економічний університет імені Вадима Гетьмана
Навчально-науковий інститут «Інститут інформаційних технологій в економіці»**

Кафедра інформаційних систем в економіці

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»

галузь знань 12 «Інформаційні технології»

спеціальність 122 «Комп'ютерні науки»

ПОГОДЖЕНО:

Керівник проектної групи(гарант)
освітньо-професійної програми

_____ Іванченко Г.Ф.
“ _____ ” _____ 2024 р.

ЗАТВЕРДЖУЮ:

Завідувач кафедри
_____ Тішков Б.О.

“ _____ ” _____ 2024 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувача вищої освіти ***Чуланової Мілени Олегівни***

очної (денної) форми навчання

**на підготовку кваліфікаційного бакалаврського проекту
на тему: «РОЗРОБКА МЕДИЧНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ
АВТОМАТИЗАЦІЇ РОБОТИ РЕЄСТРАТУРИ»**

Тему затверджено наказом ректора Університету від «11» _03_ 2024 р. № 529-ст.

Кваліфікаційний кваліфікаційний бакалаврський проект виконується на матеріалах, отриманих з мережі Інтернет, а також зібраного фактичного матеріалу з обраної теми дослідження під час практики

План кваліфікаційного бакалаврського проекту

Розділ I. Характеристика та аналіз предметної галузі

Розділ II. Розробка вимог і моделювання інформаційної системи

Розділ III. Проектування та реалізація компонентів системи

Об'єкт дослідження - процес обслуговування пацієнтів у реєстратурах медичних закладів.

Предмет дослідження - автоматизація процесів роботи поліклініки.

Мета кваліфікаційного бакалаврського проекту - підвищення ефективності роботи реєстратур та приймальних відділень лікувально-профілактичних

установ, у тому числі – для скорочення часу очікування пацієнтів запису на прийом до лікаря.

Конкретні завдання, які студент повинен виконати для досягнення поставленої мети:

У розділі I. Охарактеризувати предметну галузь та об’єкт дослідження, проаналізувати літературні джерела та практичний досвід використання технологій в предметній галузі.

У розділі II. Розробити специфікацію вимог на основі результатів характеристики предметної області та аналізу прикладів. Описати постановку задачі та алгоритми функціонування системи. Здійснити моделювання системи.

У розділі III. Розробити інформаційне, технічне та програмне забезпечення для компонентів медичної інформаційної системи для автоматизації роботи реєстратури. Представити результати реалізації системи.

**Завдання підготував
науковий керівник** _____

Сквордякова О. О.

“__15__” __03__ 2024 р.

**Завдання одержав
студент** _____

Чуланова Мілена Олегівна

“__15__” __03__ 2024 р.

Відгук
про кваліфікаційний бакалаврський проект
здобувача навчально-наукового інституту
«Інститут інформаційних технологій в економіці»
освітньо-професійної програми
«Комп'ютерні науки»

Чуланової Мілени Олегівни

на тему
**«Розробка медичної інформаційної системи для автоматизації роботи
реєстратури»**

1. **Актуальність теми:** На даний момент у закладах охорони здоров'я виникає необхідність у впровадженні інформаційних систем, які можуть автоматизувати та прискорити діяльність персоналу, забезпечуючи швидкий доступ до даних про лікарів, пацієнтів та їх медичні записи, а також ефективно управління електронними медичними документами. Ця потреба обумовлена зростаючим обсягом інформації, що вимагає обробки, для ефективного виконання різних завдань у медичній сфері. Тому робота здобувача відповідає актуальному запиту на створення ефективних медичних інформаційних систем.
2. **Позитивні риси кваліфікаційного бакалаврського проекту:** здобувачем було здійснено глибокий аналіз предметної сфери, застосовано актуальні методи синтезу інформації, а також програмного забезпечення для розроблення медичної інформаційної системи.
3. **Наявність самостійних розробок автора:** здобувачем було спроектовано та розроблено медичну інформаційну систему для реєстратури, запропоновано своє бачення оптимізації роботи медичних закладів.
4. **Цінність теоретичних висновків та практичних рекомендацій:** отримані результати можуть слугувати фундаментом для створення повноцінної медичної інформаційної системи та подальших досліджень в галузі розробки медичних інформаційних систем.
5. **Наявність недоліків:** розроблена система потребує подальшого вдосконалення за рахунок розширення функціональних можливостей.
6. **Загальна оцінка кваліфікаційного бакалаврського проекту та його допущення до захисту перед ЕК:** кваліфікаційний бакалаврський проект здобувача Чуланової М.О. відповідає встановленим вимогам методичних вказівок, рекомендується до захисту і заслуговує високої позитивної оцінки.

Науковий керівник

ст. викладач Сквордякова О.О.

A small, square, grayscale image of a handwritten signature in cursive script, appearing to read 'Сквордякова'.

“07” _____ червня _____ 2024 р.

АНОТАЦІЯ

Кваліфікаційного бакалаврського проекту студентки 4 курсу
Навчально-наукового інституту «Інститут інформаційних технологій в
економіці»

Чуланової Мілени Олегівни, виконаної на тему:
«Розробка медичної інформаційної системи для автоматизації роботи
реєстратури»

Київ: кафедра інформаційних технологій в економіці, 2024 р.

Кваліфікаційний бакалаврський проект присвячений розробці медичної інформаційної системи, спрямованої на автоматизацію процесів роботи реєстратури поліклініки.

Кваліфікаційний бакалаврський проект складається з трьох розділів, логічно пов'язаних між собою.

У першому розділі роботи проведений аналіз предметної галузі, її функцій та організаційної структури медичних закладів. Також аналізується література, що стосується досліджуваної предметної області, та надається порівняльна характеристика існуючих медичних програмних систем, оцінюються їх переваги та недоліки.

Другий розділ присвячений розробці вимог і моделюванню інформаційної системи. У цьому розділі описуються бізнес-вимоги, функціональні та нефункціональні вимоги. Проведено аналіз вхідної та вихідної інформації, основних формул для оцінки ефективності роботи системи, а також моделювання функціональності та поведінки системи.

Третій розділ роботи містить практичну частину дослідження, в якій розробляються компоненти системи, що включає розробку основних складових системи, таких як інформаційне, технічне, програмне та

організаційне забезпечення. Детально розглядається структура бази даних, а також способи зберігання та обробки даних, обґрунтовуються вибір СКБД, мережевого обладнання та його розміщення, середовища розробки інформаційної системи та мови програмування.

У висновках підбиваються підсумки роботи та пропонуються рекомендації щодо практичного застосування розробки системи в медичних установах.

РЕФЕРАТ

Кваліфікаційний бакалаврський проект містить 127 сторінок, 17 таблиць, 65 рисунків, список літератури з 44 найменувань, 8 додатків.

РОЗРОБКА МЕДИЧНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ АВТОМАТИЗАЦІЇ РОБОТИ РЕЄСТРАТУРИ

Перелік ключових слів: автоматизація, проектування, інформаційна система, реєстратура, медична інформаційна система, поліклініка, вимоги, розробка, тестування, прототипування, верифікація, діаграми, база даних, забезпечення, розробка.

Предметом дослідження є автоматизація процесів роботи поліклініки.

Об'єктом дослідження виступає процес обслуговування пацієнтів у реєстратурах медичних закладів.

Метою кваліфікаційного бакалаврського проекту є підвищення ефективності роботи реєстратур та приймальних відділень лікувально-профілактичних установ, у тому числі – для скорочення часу очікування пацієнтів запису на прийом до лікаря.

Завданнями кваліфікаційного бакалаврського проекту є розробка інформаційної системи, яка дозволить скоротити час очікування пацієнтів запису на прийом до лікаря шляхом дослідження сучасних та популярних підходів і технологій розробки, проектування та створення інформаційних систем.

Апаратні та програмні засоби, що використовувались при проектуванні: Visual Studio 2022, Visual Studio Code, база даних MySQL, MySQL Workbench, HTML, CSS, Enterprise Architect, AllFusion Process Modeler BPwin.

Результати, досягнуті в процесі роботи, включають глибокий аналіз предметної галузі, розробку детальної концепції та архітектури, моделі даних та прототипів інтерфейсу користувача інформаційної системи.

Отримані результати можуть слугувати фундаментом для створення повноцінної медичної інформаційної системи та подальших досліджень в галузі розробки медичних інформаційних систем.

Рік виконання випускного бакалаврського проекту 2024.

Рік захисту випускного бакалаврського проекту 2024.

ЗМІСТ

ЗМІСТ	2
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, ВЕЛИЧИН І ТЕРМІНІВ.....	4
ВСТУП	5
РОЗДІЛ 1. ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ.....	7
1.1. Характеристика предметної галузі та об'єкта дослідження.....	7
1.2 Аналіз літературних джерел	10
1.3 Практичний досвід використання ІС	18
1.4 Порівняльна характеристика існуючих програмних систем	24
РОЗДІЛ 2. РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	30
2.1. Аналіз і специфікація вимог до інформаційної системи	30
2.1.1 Бізнес вимоги	30
2.1.2 Функціональні вимоги.....	32
2.1.3 Нефункціональні вимоги	37
2.2. Постановка та математичний опис задачі	40
2.2.1. Постановка задачі.	40
2.2.2 Вихідна інформація	42
2.2.3 Вхідна інформація.....	43
2.2.4 Математичний опис	47
2.2.4 Алгоритм розв'язання задачі	48
2.3. Моделювання інформаційної системи.....	52
2.3.1 Моделювання та аналіз бізнес-процесів.....	52
2.3.2 Моделювання поведінки системи	54
2.3.3 Моделювання структури системи	57
2.3.4 Розподіл вимог за компонентами системи	58

РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ	
ПІДСИСТЕМИ	63
3.1. Інформаційне забезпечення	63
3.2. Технічне забезпечення	94
3.3. Програмне забезпечення	98
3.3.1 Побудова структури програмного забезпечення	98
3.3.2 Програмне забезпечення	100
3.4 Результати реалізації інформаційної системи	104
ВИСНОВОК	117
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	119
ДОДАТКИ	124

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, ВЕЛИЧИН І ТЕРМІНІВ

IC	Інформаційна система
MIC	Медична інформаційна система
СКБД	Система керування базами даних
EP	Електронна реєстратура
SQL	Structured Query Language
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
IDEF	Integration Definition for Function Modeling
BPMN	Business Process Model and Notation
SysML	Systems Modeling Language
GUI	Graphical User Interface
ЖМД	Жорсткий магнітний диск
HTTP	Hypertext Transfer Protocol
PHP	Hypertext Preprocessor
SMTP	Simple Mail Transfer Protocol
IMAP	Internet Message Access Protocol
POP3	Post Office Protocol version 3
VPN	Virtual Private Network
TLS/SSL	Transport Layer Security / Secure Sockets Layer
SSD	Solid-State Drive
DDR4	Double Data Rate 4th generation
UTP	Unshielded Twisted Pair
ADSL	Asymmetric Digital Subscriber Line

ВСТУП

На даний момент у закладах охорони здоров'я виникає необхідність у впровадженні інформаційних систем, які можуть автоматизувати та прискорити діяльність персоналу, забезпечуючи швидкий доступ до даних про лікарів, пацієнтів та їх медичні записи, а також ефективне управління електронними медичними документами. Ця потреба обумовлена зростаючим обсягом інформації, що вимагає обробки ефективного виконання різних завдань у медичній сфері. Впровадження інформаційних систем дозволяє вирішити проблеми управління інформаційними потоками, оптимізації часу обробки даних та складнощів, пов'язаних із їх зберіганням. З цих міркувань мною і була обрана тема бакалаврського проекту «Розробка медичної інформаційної системи для автоматизації роботи реєстратури».

Донедавна система охорони здоров'я України мало використовувала автоматизацію, що призводило до ручного виконання всіх процесів і документообігу. Це ускладнювало роботу медичних закладів. На сьогоднішній день автоматизація роботи медично-лікувальних закладів стала актуальною у зв'язку із прагненням покращити якість медичних послуг, оптимізувати процеси та забезпечити повну доступність інформації для персоналу та пацієнтів.

Мета даної бакалаврської роботи полягає у створенні інформаційної системи реєстратури поліклініки, яка міститиме дані про пацієнтів, їх звернення, лікарів, графік роботи медичного персоналу та основну інформацію поліклініки.

Дана мета досягається виконанням наступних задач:

- аналіз поточного стану роботи реєстратур медичних закладів;

- вивчення літературних джерел та практичного досвіду використання інформаційних систем у сфері охорони здоров'я;
- аналіз існуючих медичних інформаційних систем;
- визначення вимог до функціоналу та структури інформаційної системи реєстратури;
- розробка прототипу інтерфейсу медичної інформаційної системи реєстратури;
- проектування компонентів підсистеми, включаючи інформаційне, технічне, програмне та організаційне забезпечення;
- реалізація розроблених компонентів системи;
- оцінка результатів та формування висновків.

Створення такої системи дозволить підвищити ефективність роботи медичного закладу за рахунок скорочення часу та трудовитрат, а також підвищення якості обслуговування пацієнтів.

Предметом дослідження є автоматизація процесів роботи поліклініки. Даний процес пов'язаний з впровадженням комплексів та методів, які допоможуть персоналу опрацьовувати медичну документацію автоматично.

Об'єктом дослідження є процес обслуговування пацієнтів у реєстратурах медичних закладів, який завдяки автоматизації забезпечує комфортне та якісне обслуговування пацієнтів, які звертаються до поліклініки.

Таким чином, розробка інформаційної системи реєстратури поліклініки є актуальним і значущим завданням, яке має на меті поліпшення якості медичних послуг, що надаються, та оптимізації роботи медичних закладів.

РОЗДІЛ 1. ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1. Характеристика предметної галузі та об'єкта дослідження

Поліклініка є ключовим елементом у системі охорони здоров'я, що забезпечує оперативну та якісну діагностику, планування та проведення лікування відповідно до сучасних міжнародних стандартів.

Основними функціями поліклініки є [35]:

Прийняття пацієнтів як за попереднім записом, так і при гострій необхідності;

Надання первинної та спеціалізованої медичної допомоги, у тому числі в невідкладному порядку;

Проведення діагностики та лікування різних захворювань та станів;

Встановлення медичних показань та направлення пацієнтів до спеціалізованих медичних закладів за потреби;

Проведення медичних оглядів усіх видів;

Надання лікарських консультацій;

Дотримання правил ведення медичної документації та подання відповідної звітності.

Організаційна структура поліклініки наведена на рисунку 1.1.

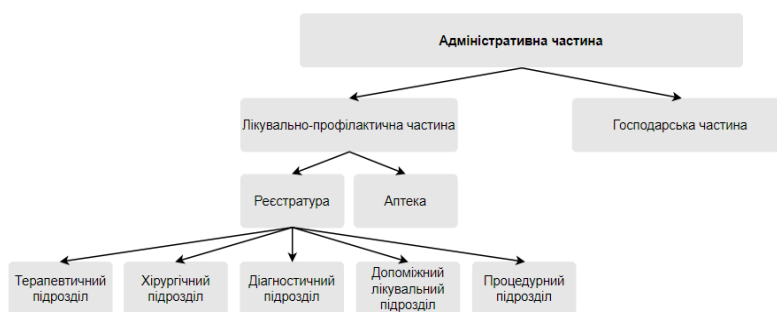


Рисунок 1.1 - Орієнтовна організаційна структура міської поліклініки

Координаційну роботу при зверненні пацієнтів в лікувально-профілактичні заклади забезпечує реєстратура, яка виконує функції основного структурного підрозділу поліклініки щодо організації прийому хворих у медичному закладі.

Отже, реєстратура забезпечує ефективний розподіл медичних ресурсів, організовуючи запис на прийоми та видачу пацієнтам направлень, надаючи пацієнтам професійне та якісне обслуговування.

Загальне керування роботою даного підрозділу виконується завідувачем реєстратури, який координує роботу персоналу, а також регулює взаємозв'язок з іншими підрозділами поліклініки.

Медичні реєстратори є основним персоналом реєстратури, який є відповідальним за організацію взаємодії між пацієнтами та медичними спеціалістами, а також консультує пацієнтів з питань медичного обслуговування.

Середній час очікування пацієнта на запис до лікаря може коливатися від декількох хвилин до декількох тижнів, в залежності від різних факторів: спеціалізація лікаря, місцезнаходження та регіон, наявна завантаженість, специфіка пацієнта.

Раціональна організація прийому та наявність автоматизованих систем запису можуть суттєво скорочувати час очікування, завдяки можливості запису заздалегідь та керування утвореними записами.

Комплексна інформаційна автоматизована система "Реєстратура поліклініки" створена для полегшення процесу запису пацієнтів на прийом до медичних закладів та скорочення часу очікування хворих на прийом до лікарів з метою покращення якості медичного обслуговування населення.

Основними функціями автоматизованої ІС реєстратури є:

Забезпечення оптимального використання ресурсів поліклініки та підвищення ефективності операційних процесів: зберігання та упорядкування документації, організація швидкої реєстрації пацієнтів на прийом до лікаря, звітування про наявність лікарів та їх спеціальностей, години прийому та місце прийому, облік медичного персоналу та осіб, які звертаються до медичного закладу для отримання медичної допомоги, збереження та обробка медичних даних пацієнтів.

Оптимізація використання медичного персоналу, обладнання та ресурсів для підвищення рівня обслуговування пацієнтів: ефективний розподіл ресурсів медичного закладу, включаючи персонал та обладнання, контроль над потоком пацієнтів з метою рівномірного розподілу навантаження на медичний персонал та ефективності надання медичних послуг.

Створення інтегрованої інформаційної системи для управлінського аналізу та інтеграції із зовнішніми інформаційними системами у сфері охорони здоров'я: створення звітів та аналітичних даних для адміністративного та управлінського використання, можливість інтеграції з іншими інформаційними системами, такими як лабораторні інформаційні системи або системи електронної медичної історії.

Дана система оброблятиме дані про поліклініку, реєстраторів, лікарів, персональні дані пацієнтів, записи з розкладом прийому лікарів та записи відвідування пацієнтів. Процеси включатимуть реєстрацію пацієнтів та їх прийомів, облік прийомів у лікарів, і передачу даних між реєстратурою і лікарями.

Таким чином, автоматизована система має полегшувати керування медичними даними, забезпечувати виконання основних функціональних вимог та задовольняти потребам предметної області.

1.2. Аналіз літературних джерел

Аналіз літературних джерел, що стосуються теми оптимізації операцій медичних установ через автоматизацію робочих процесів, стає невід'ємною частиною наукового та практичного вивчення сучасних тенденцій та проблем даної сфери.

Метою даного аналізу є проведення широкого огляду актуальних наукових праць, що супроводжується вивченням сучасних теорій, методів та результатів досліджень у галузі МІС. Шляхом огляду різних джерел підкреслюються основні тренди, проблеми та перспективи розвитку даної галузі, забезпечуючи глибше розуміння аспектів застосування медичних інформаційних систем, а також питань та викликів, пов'язаних із впровадженням автоматизації у сфері охорони здоров'я.

Як вже було зазначено вище, основною тенденцією розвитку сучасної галузі охорони здоров'я є перехід до ефективних засобів автоматизації роботи з медичними даними. Дана трансформація відбувається в декілька етапів: відмова від старих моделей, цифровізація та трансформація.

Автори дослідницького журналу «The Journal of Strategic Information Systems» у своїй доповіді «Understanding digital transformation: A review and a research agenda» [1] запропонували таке визначення цифрової трансформації: цифрова трансформація відноситься до «різноманітних соціотехнічних явищ та процесів прийняття та використання [цифрових] технологій у більш широкому індивідуальному, організаційному та суспільному контекстах».

Основні аспекти цифрової трансформації, що охоплює не лише технологічні аспекти, а й зміни у відносинах з клієнтами, процесами та корпоративною культурою, називає Джордж Вестерман [2]: "Цифрова трансформація — це руйнівна еволюція до абсолютно нового способу роботи та мислення. І цей процес може вимагати повної трансформації корпоративних частин тіла для встановлення нових способів функціонування."

Європейська комісія називає цифровою трансформацією застосування інноваційних розробок на основі інформаційних та телекомунікаційних технологій для вирішення різноманітних завдань. [3]

Звідси витікає, що головною метою цифрової трансформації галузі охорони здоров'я є підвищення ефективності, доступності та оперативності надання медичних послуг, шляхом автоматизації документообігу та рутинних процесів управління медичною інформацією в цілому.

Конвергенція технологій, політики та нагальність змін створює так званий «ідеальний шторм для інновацій» [4]. Такі моменти є ключовими для просування нових парадигм охорони здоров'я, як це було під час пандемії COVID-19, яка прискорила впровадження телемедицини та інших цифрових рішень для охорони здоров'я.

За останнє десятиліття сектор охорони здоров'я зазнав істотних змін у контексті впровадження медичних інформаційних технологій, що свідчить

про зростання визнання трансформаційного потенціалу автоматизації у сфері медичного обслуговування.

У своїй основоположній роботі «The Impact of Health Information Technology on the Quality of Medical and Health Care: A Systematic Review» [5] Чаудрі Б., Ван Дж., Ву С. провели комплексний аналіз впливу медичних інформаційних технологій на якість проведення медичного обслуговування пацієнтів. Їхнє дослідження ґрунтується на ретельному аналізі, спрямованому на з'ясування потенційних переваг і проблем, пов'язаних із впровадженням МІС у медичних установах. У статті підкреслено важливість автоматизації процесів охорони здоров'я шляхом впровадження медичних інформаційних систем. Завдяки спрощенню рутинних адміністративних завдань, полегшенню обміну даними та покращенню зв'язку між постачальниками медичних послуг МІС визначає потенціал для оптимізації різних аспектів надання медичної допомоги. "Of 23 studies included in the current review, 17 assessed the impact of HIT/HIS on healthcare practitioners' performance. A positive improvement, ... , was seen in 14 studies. Studies that included an assessment of patient outcomes, however, showed insufficient evidence of either clinically or statistically important improvements". Проте, попри оптимістичний настрій, автори також виділяють постійні проблеми, що перешкоджають широкому прийняттю та впровадженню МІС. Такі проблеми, як обмеження у процесі впровадження та сумісності, є серйозними бар'єрами, які потрібно подолати для повної реалізації трансформаційного потенціалу медичній сфері. Автори у своїй роботі наголошують: "However, challenges such as implementation barriers and interoperability issues need to be addressed to maximize the benefits of health information technology."

Шляхом застосування методології систематичного огляду можна виявити наслідки впровадження МІС для часу, який медичні працівники

витрачають на виконання своїх обов'язків, а отже тим самим і вивчити вплив електронних медичних записів на ефективність використання робочого часу медичних працівників.

Проводячи синтез результатів різних досліджень, можливо виявити закономірності та тенденції впливу використання МІС на робочі процеси та розподіл часу виконання рутинних завдань. Метод систематичного огляду досліджень допомагає уважно проаналізувати наявні дані та виділити важливі аспекти, такі як проектування системи та оптимальний час її впровадження, оцінюючи вплив систем електронних медичних записів. Однак, використання даного методу не вирішує питання втрати останніх технологічних досягнень в галузі МІС, які могли б допомогти вирішити деякі з виявлених проблем у сфері оптимізації часу впровадження системи. Крім того, конкретні програмні системи, використані в аналізованих дослідженнях, не завжди можуть бути застосовні у різних ситуаціях, що також послаблює якість виявлених результатів. У роботі авторів [33] виявлено декілька ключових проблем, включаючи збільшення часу на документування через використання деяких автоматизованих систем, а також складність навчання персоналу та недоліки в проектуванні системи, що можуть призвести до погіршення ефективності робочих процесів. Крім того, стаття наголошує на ризики втрати інших потенційних переваг, якщо впровадження системи спрямоване виключно на скорочення часу на оформлення документації. Дана стаття містить цінні відомості про труднощі впровадження систем електронних медичних систем та їх вплив на оптимізацію робочого часу. Вона наголошує, що для повного розуміння ситуації необхідно подолати поверхове оцінювання та більш глибоко вивчити проблему. Особливу увагу приділено процесу проектування системи, оскільки це може суттєво покращити роботу медичних працівників завдяки оптимізації робочих процесів.

Як зазначають автори, " The impact of electronic health records on time efficiency of physicians and nurses is complex and multifaceted, with both positive and negative effects observed across different healthcare settings and specialties".

Галузь охорони здоров'я потребує технологічної перебудови з метою підвищення ефективності та якості медичної допомоги. Книга "Healthcare Information Systems: A Practical Approach for Healthcare Management" [6] описує переваги впровадження інформаційних технологій в сфері охорони здоров'я, керування даними та інформацією, а також проблеми, які виникають під час впровадження автоматизації. Дослідники у своїй роботі визначили наступні найбільш поширені проблеми: вибір та придбання найбільш підходящої системи для конкретних потреб організації, забезпечення безпеки та захисту конфіденційності медичних даних, інтеграція системи з поточними робочими процесами та вже впровадженими системами, навчання персоналу нових технологій, оцінка рентабельності інвестицій у систему. Проте, все ж таки, незважаючи на ретельний огляд обраної теми в даній книзі, слід визнати її певні обмеження. По-перше, вона значною мірою спирається на літературу, опубліковану в період з 2005 по 2012 рік, що може знизити актуальність наданої інформації для аудиторії, яка цікавиться останніми досягненнями в галузі інформаційних систем сфери охорони здоров'я. По-друге, автори визнають недостатність повної звітності та інформації про процеси впровадження та можливості інформаційних технологій у сфері охорони здоров'я, що ускладнює повну оцінку ефективності та практичності використання таких систем.

Багато існуючих досліджень обговорюють складності впровадження електронних медичних реєстрів у клінічну практику, наголошуючи на важливість питання захищеності та точності даних, а також оптимізації робочого процесу [7]. Однак Джулія Адлер-Мільштейн, Кіртін Вуді Скотт та

Ашіш К Джа. описують свою точку зору на дану проблему у своєму дослідженні "Leveraging EHRs to improve hospital performance: the role of management" [8]. Хоча багато хто вважав, що впровадження електронних медичних систем реєстратури покращить ефективність та якість медичної обслуговування, дослідження виявило між цими параметрами досить неоднорідний зв'язок. Основний висновок дослідження полягає в тому, що саме ефективне управління медичним закладом відіграє ключову роль у розкритті потенційних переваг впровадження ЕР. Лікарні з ефективними побудованими практиками управління помітили значне покращення у фінансових витратах, рівні якості медичної допомоги та безпеки пацієнтів після впровадження ЕР, на відміну від закладів, де управління не таке ефективне, і де впровадження ЕР не виявило такого явного позитивного впливу на зазначені аспекти. Отже, дослідження наголошує, що керівництво та корпоративна культура відіграють важливу роль у підвищенні позитивного впливу на ефективність роботи лікарні. Інтеграція з поточними робочими процесами розглядається як одна із основних стратегій [9]. Проведено дослідження, як лікарні можуть покращити аналіз великих даних за допомогою МІС, опитавши 580 медичних працівників у великій китайській лікарні в 2019 році. Проаналізувавши дані за допомогою моделювання структурних рівнянь, можна побачити, чи розуміють робітники, як використовувати МІС, як зберігати та використовувати їх дані та генерувати цінну інформацію з даних систем. За допомогою автоматизації існуючих робочих процесів зменшується навантаження медичного персоналу, що, в свою чергу, призводить до скорочення часу на обробку даних покращення точності їх збору. Ефективна реалізація впроваджених систем також вимагає навчання персоналу [10]. Вирішення цього питання акцентує увагу на таких стратегіях, як побудова зручного інтерфейсу, оптимізація процесів введення

даних та введення спеціалізованих програм навчання для медичного персоналу. Нарешті, захист та конфіденційність даних залишаються в центрі уваги, як зазначається у рекомендаціях щодо безпеки від Американської асоціації медичної інформатики (AMIA) [11].

У книзі Кетлін А. МакКормік «Healthcare Information Technology Exam Guide for CompTIA Healthcare IT Technician and HIT Pro Certifications» [12] наведено вичерпний огляд впровадження медичних інформаційних систем, електронних медичних реєстратур, автоматизованого обміну медичною інформацією та встановлення нормативних вимог. Хоча вона і пропонує практичні вказівки щодо впровадження та управління автоматизованими МІС, у ній може бракувати поглибленого обговорення передових тем або нових технологій даної сфери.

Стаття «Opportunities for the use of health information technology to support research» [13] аналізує проблеми та можливості, пов'язані з використанням медичних інформаційних технологій у здійсненні медичної обслуговування. Вона пропонує критичний огляд поточної практики впровадження МІС та визначає поточні ключові проблеми, такі як інтеграція, безпека даних і зручність використання. Однак вона в повній мірі не пропонує конкретних рішень чи практичних рекомендацій щодо вирішення цих проблем.

Багато існуючих досліджень у сфері МІС спрямовані на проведення економічного аналізу впровадження медичних інформаційних систем, огляд існуючої літератури для визначення факторів, що впливають на витрати, вигоди та рентабельність інвестицій, а також для оцінки економічних моделей і методів, які використовуються в процесі розробки і впровадження МІС [24].

Використовуючи підхід до систематичного огляду, проводяться дослідження, пов'язані з економікою впровадження МІС із таких баз даних, як PubMed, Scopus і Web of Science. Статті, відібрані для рецензування, проходять

якісний аналіз, щоб отримати інформацію про фактори вартості, економічні моделі та методи оцінки. Фактори, що впливають на вартість впровадження ІС, включають складність системи, розмір і готовність організації, вибір постачальника, відповідність нормативним вимогам і складність інтеграції даних. Визначені економічні моделі та методи включають аналіз витрат і переваг, аналіз ефективності витрат і аналіз повернення інвестицій.

У статті “Public management and public administration of hospital information system as the main tool of information communication technology in Ukrainian healthcare” [14] українського автора Дмитра Самофалова, досліджується стан інформаційних систем лікарень в Україні, а також аналізується правова база, яка оточує електронні медичні записи та інформаційні системи лікарень. Хоча конкретне дослідження не надає конкретних висновків, воно аналізує поточний стан інформаційних систем лікарень в Україні за допомогою аналізу правової бази, публічних установ керування лікарнями, результатів впровадження електронних медичних систем та потенційних ризиків, пов'язаних з безпекою даних та проблемами конфіденційності.

Таким чином, після ґрунтовного проведення аналізу значних літературних джерел можна дійти висновку, що хоча за останні роки був досягнений значний прогрес у впровадженні медичних інформаційних систем, на сьогоднішній день все ще існує ряд суттєвих питань, вирішення яких потребує уваги. Серед них найчастіше включають проблеми інтеграції, питання безпеки та захисту даних, адаптація працівників до нововведень та фінансові обмеження. Крім того, існує потреба у проведенні додаткових досліджень, спрямованих на вироблення практичних рішень та виявлення найкращих практик для урегулювання вищезазначених питань.

З проведеного аналізу виділено такі ключові питання:

Як оптимізувати інтеграцію між різними медичними інформаційними системами для забезпечення ефективного обміну даними?

Які стратегії можуть бути розроблені для адаптації медичного персоналу до змін під час впровадження медичних інформаційних систем?

Як організації охорони здоров'я можуть подолати фінансові перешкоди для інвестування та підтримки впровадження медичних інформаційних систем?

В цілому, вирішення цих проблем потребує багато дисциплінарного підходу, який передбачає активну взаємодію між закладами охорони здоров'я, фахівцями в галузі інформаційних технологій, політиками та дослідниками. Крім того, безперервні дослідження та оцінка стратегій впровадження є ключовими для забезпечення ефективного використання медичних інформаційних систем з метою покращення надання медичної допомоги.

1.3. Практичний досвід використання ІС

На сьогоднішній день можна виділити значну кількість робіт, спрямованих на аналіз конкретних прикладів упровадження та застосування таких медичних інформаційних систем у реальних умовах. Мета проведення цих робіт полягає в аналізі практичного досвіду, виявленні викликів, з якими стикаються організації при впровадженні та у процесі використання медичних інформаційних систем.

Автори дослідження розглянули як успішні, так і невдалі випадки впровадження медичних інформаційних систем, виявивши ключові фактори, що впливають на результати їх використання. Провівши аналіз різноманітних випадків, дослідження виявляє тенденції, найкращі практики та стратегії, які

сприяють ефективному впровадженню та експлуатації інформаційних систем у сфері охорони здоров'я.

The Healthcare Information and Management Systems Society (HIMSS) Analytics Case Studies проводили тематичні дослідження [15], які демонстрували реальні випадки впровадження медичних інформаційних систем у різних закладах охорони здоров'я. HIMSS співпрацює з іншими організаціями, урядовими установами та стейкхолдерами галузі для просування досліджень, розробки стандартів та політичних ініціатив, пов'язаних із впровадженням медичних інформаційних систем. Працюючи разом, ці організації можуть використати свій колективний досвід і ресурси для досягнення прогресу в IT-сфері охорони здоров'я. У своїй практиці HIMSS Analytics за допомогою досліджень, опитувань, вебінарів, оцінок і порівняльних досліджень оцінює впровадження та вплив електронних медичних систем. Результати цих досліджень підтверджують значний позитивний вплив, що виявляється в оптимізації управління ресурсами поліклініки, зменшенні кількості помилок під час обробки інформації та підвищенні ефективності роботи медичного персоналу. Це надає розробникам можливість отримати чітке уявлення про взаємодію між технологічними рішеннями, яка в кінцевому підсумку формує успішне розгортання МІС.

Epic Systems [16] є провідним постачальником програмного забезпечення для МІС, яке використовується в медичних організаціях по всьому світу. Його платформа пропонує повний набір функцій, призначених для оптимізації робочих процесів лікувальних закладів. Завдяки потужній технічній підтримці та постійним оновленням Epic Systems надає організаціям інструменти та ресурси, необхідні для успішного впровадження та обслуговування МІС. Mayo Clinic, одна з найбільших і найвідоміших систем охорони здоров'я в усьому світі, використовує програмне забезпечення Epic

Systems у своїй мережі клінік. Програмне забезпечення допомагає оптимізувати обслуговування пацієнтів, покращити робочі процеси та співпрацю між постачальниками медичних послуг. Kaiser Permanente, велика інтегрована система надання медичних послуг, базується на ПЗ Epic Systems. Програмне забезпечення дозволяє комплексно документувати дані пацієнтів, сприяє безпечному обміну цими даними і підтримує їх аналіз і звітність. Johns Hopkins Medicine, провідний академічний медичний центр, також використовує дане програмне забезпечення для стандартизації клінічних процесів, покращення координації обслуговування та підвищення безпеки даних. Функції інтеграції програмного забезпечення дозволяють безперебійно обмінюватися інформацією про пацієнта із зовнішніми постачальниками медичних послуг.

SoftServe [17] – українська цифрова консалтингова компанія, яка надає різноманітні IT-рішення, зокрема інформаційні системи охорони здоров'я. Вони пропонують послуги з розробки спеціального програмного забезпечення, адаптованого до потреб організацій охорони здоров'я, включаючи системи електронних медичних реєстратур, телемедичні платформи та рішення для аналітики охорони здоров'я.

Системи EP від SoftServe розроблені для оцифрування та оптимізації записів про стан здоров'я пацієнтів, що дозволяє постачальникам медичних послуг отримувати безпечний доступ до повної інформації про пацієнтів. Реальні випадки використання включають впровадження систем EP у лікарнях і клініках для підвищення точності та захисту даних, оптимізації робочого процесу, покращення координації обслуговування, а також інтеграція систем EP з іншими IT-рішеннями лікувальних закладів.

Проте, як зазначає сама компанія, медичні організації стикаються з різними проблемами впровадження, зокрема, проблемами конфіденційності

даних, проблемами сумісності з іншими з системами, перешкодами, які виникають у користувачів при впровадженні, та обмеженнями ресурсів, такими як обмеження бюджету, людських ресурсів або технічних можливостей.

Практичний кейс " Advancing health information technology roadmaps in long term care" [32], оснований на квалітативному дослідженні, аналізує успішну реалізацію інформаційних технологій у сфері охорони. Дане дослідження базується на аналізі стратегій і змісту досвіду впровадження МІС від спільних компаній у Сполучених Штатах, Австралії, Великобританії та Новій Зеландії. На основі даного аналізу вивчаються фактори, які сприяли успішній імплементації, а також перешкоди та виклики, які доводиться долати під час цього процесу. Наступним етапом команда провела систематичний огляд наявної літератури з 2000 по 2018 рік. Цей огляд був розроблений, щоб підтвердити припущення, щодо впровадження МІС вищезгаданими компаніями. Основні висновки дослідження включають в себе: визначення факторів, які сприяли успішній імплементації, таких як ефективне планування, підтримка керівництва, залучення ключових стейкхолдерів; виявлення перешкод, з якими зіткнулися під час процесу впровадження, таких як технічні проблеми, недостатня підтримка персоналу та фінансові обмеження. На основі отриманих даних надано рекомендації для майбутніх проектів з впровадження інформаційних технологій у сфері охорони здоров'я, зокрема, щодо стратегій залучення персоналу, планування та управління ризиками. Дане дослідження підкреслює важливість збалансованого підходу до впровадження інформаційних технологій у медичній сфері та розуміння перспектив різних зацікавлених сторін.

В рамках механізації роботи медичних установ в Україні було впроваджено різні системи, включаючи електронну чергу, електронну

медичну документацію, фінансову та управлінську автоматизацію. Наприклад, діє система електронних медичних записів «E-Health Ukraine», яка забезпечує цифрове зберігання та управління медичною інформацією, підвищуючи якість та безпеку медичних послуг. Ця система підключена до централізованої бази даних та забезпечує доступ до медичної документації у реальному часі. Механізація медичних закладів, включаючи поліклініки, продовжується та розвивається.

Як вже зазначалося раніше, системи ЕР трансформують адміністративні процеси в медичних установах. По-перше, ЕР мінімізує використання паперової документації, автоматизує планування прийомів, реєстрацію пацієнтів і перевірку медичних даних. Це скорочує час очікування на обслуговування пацієнтів, черги та спрощує роботу персоналу завдяки порядкуванню інформації, а отже і підвищує якість послуг. Системи електронної реєстратури можуть скоротити час очікування на обслуговування до 30%, що, в свою чергу, сприяє підвищенню задоволеності пацієнтів і продуктивності роботи персоналу [18].

Електронні системи забезпечують точність даних, усуваючи помилки, пов'язані з їх ручним введенням. Це полегшує обмін інформацією між медичним персоналом, покращує координацію медичної допомоги та зменшує ризик утворення помилок.

Крім того, зменшується кількість витрат на документообіг завдяки економії ресурсів на друк і зберігання інформації. Ефективне розподілення потоку пацієнтів сприяє збільшенню показника пропускної здатності осіб, що завертаються до медичного закладу, потенційно збільшуючи при цьому доходи клінік. Дослідження “The potential of electronic health records for reducing healthcare costs.”[31] спрямоване на аналіз потенційних можливостей зменшення витрат у сфері охорони здоров'я, пов'язаних з електронними

медичними записами, які охоплюють ЕР. Результати дослідження підтвердили, що впровадження систем електронної реєстрації може скоротити адміністративні витрати на 20%. Проте, хоча автоматизація в медичних закладах має численні переваги, існують проблеми, пов'язані з її реалізацією.

Медична інформаційна система, об'єднуючи проведення багатьох робочих процесів одночасно, забезпечує появу багатьох переваг. Серед них зручний доступ до медичних записів, зменшення витрат, а також впровадження політик та втручань у галузі охорони здоров'я, заснованих на доказах [19,20,21,22,23]. Кілька авторів показали, що переваги МІС мають значний вплив у сфері охорони здоров'я [23]. Багато медичних організацій запроваджують МІС через наявність цих численних переваг [25]. Крім того, інвестиції в МІС уможливили ефективне прийняття рішень, надання повної медичної інформації в режимі реального часу для, ефективну політику в сфері охорони здоров'я, розширений моніторинг та оцінку, ефективний розподіл ресурсів, покращене управління та звітність [21,26,27,28]. В ідеалі МІС підходить для обміну даними, інформацією та широкими знаннями в середовищі охорони здоров'я. Критичні функції МІС тепер цінуються завдяки їх об'єднанню з різними технологіями [29,30].

Виходячи з реальних практичних випадків впровадження і використання МІС, можна у висновку виділити основні ряди досить складних питань, вирішення яких і на сьогоднішній день потребує уваги спеціалістів. Впровадження нової системи вимагає початкових інвестицій в розробку, обладнання та інтеграцію системи з існуючою інфраструктурою. Ці інвестиції можуть бути значними, особливо для менших установ. Перехід на електронну реєстрацію потребує ресурсів навчання персоналу для забезпечення ефективного використання системи.

Безпечне зберігання даних, надійні засоби контролю доступу та постійне навчання персоналу протоколам інформаційної безпеки також мають вирішальне значення у впровадженні нової ІС. До того ж, інтеграція з існуючими ІТ-системами закладу охорони здоров'я може бути складною. Забезпечення безперебійного обміну даними між різними платформами вимагає ретельного планування та технічного досвіду спеціалістів.

1.4. Порівняльна характеристика існуючих програмних систем

У контексті української електронної системи охорони здоров'я eHealth, використання МІС є невід'ємною частиною. Ці системи сприяють автоматизації роботи медичних установ, завдяки створенню зв'язку із Центральною базою даних (ЦБД). Відповідно до постанови Кабінету Міністрів України від 25 квітня 2018 р. №411 про «Деякі питання електронної системи охорони здоров'я», взаємодія з ЦБД та передача даних можливі лише за використанням МІС [34].

Деякі приклади найбільш популярних медичних інформаційних систем в Україні, які підтримують роботу електронної реєстратури медичного закладу включають:

Хелсі (Helsi) [36]: Helsi – це найбільша в Україні медична інформаційна система, призначена для пацієнтів та лікарів медичних закладів різного типу. Нині понад 1300 лікувальних установ (72% від загальної кількості в Україні) користуються цією системою. Основна мета Helsi – спростити процес звернення до лікаря. Пацієнти можуть скористатися системою для вибору сімейного лікаря, пошуку зручної клініки, запису себе або членів сім'ї на прийом до лікаря, перегляду своєї електронної медичної історії, що містить інформацію про вакцинацію, візити до лікаря, напрямки, медичні документи

та рецепти, а також для перегляду результатів аналізів та діагностичних процедур, призначень лікаря, отримання онлайн-консультацій лікаря та записи на вакцинацію.

Серед переваг можна виділити зручний доступ до медичних даних для пацієнтів, проте є й недоліки: система може бути несумісною з іншими медичними інформаційними системами, не завжди доступною для всіх лікарів та медичних закладів, а також має обмежені можливості роботи в офлайн-режимі. Інтерфейс даної МІС показано на рисунку 1.2.

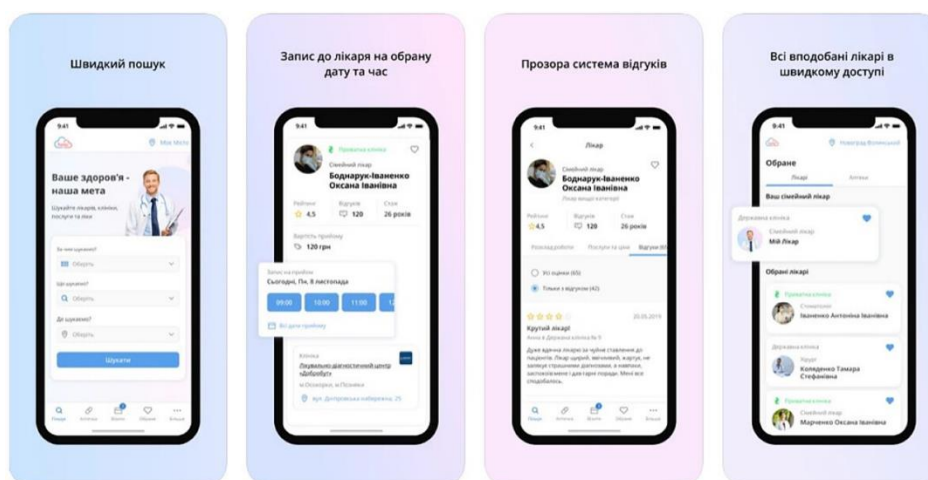


Рисунок 1.2 - Інтерфейс медичної ІС Helsi

МедКард24 (MedCard24) [37]: Medcard24 - є електронним документом, що містить медичну історію пацієнта, до якої користувач може отримати доступ у будь-який необхідний йому момент (рис.1.3). Тут зберігається вся інформація про візити до лікарів, включаючи дані про спеціаліста, поставлений діагноз, призначене лікування, дозування препаратів та рецепти. Ці відомості доступні як самим пацієнтам, так і медичному персоналу. Хоча система проста у використанні, її необхідно інтегрувати з іншими системами для обміну інформацією та враховувати обмеження роботи в офлайн-режимі.

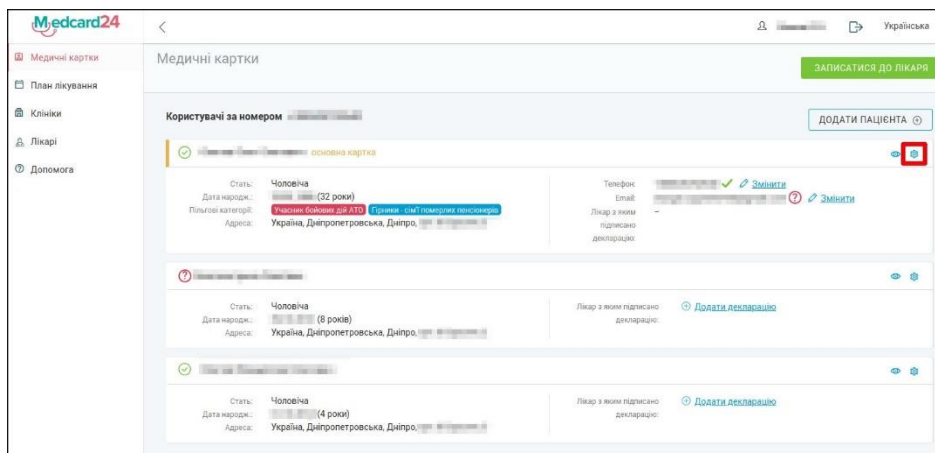


Рисунок 1.3 - Інтерфейс медичної IC MedCard 24

Медікс (Medics) [38]: Medics - це інформаційна система для лікарів та медичних працівників, яка допомагає вести облік та обробку медичних даних пацієнтів, лабораторних результатів, а також створювати медичну документацію (рис. 1.4). Дана IC є однією із перших в Україні, яка долучилась до першого етапу медичної реформи та розробки центрального компонента eHealth. Medics пропонує низку значних переваг, включаючи онлайн-запис, вбудовану інтеграцію з телефонією UIS та іншими телефонними системами через API, підтримку маркетингових інструментів та інтеграцію з програмами для організації обліку та зберігання довідкових даних в електронному вигляді. Технічна та клієнтська підтримка включена у вартість. Платформа має зручний та інтуїтивно зрозумілий інтерфейс. Однак у системі відсутня підтримка багатфакторної аутентифікації та існують

обмежені можливості вбудованої інтеграції з іншими медичними системами.

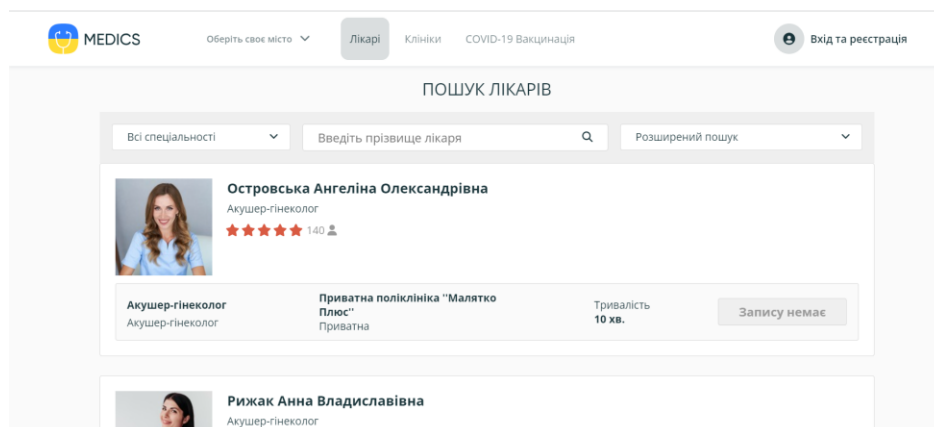


Рисунок 1.4 - Інтерфейс медичної IC Medics

Емсімед (EMCiMED) [39]: EMCiMED - інноваційна медична інформаційна платформа, розроблена для використання українськими медичними закладами (рис. 1.5). Ця система тісно інтегрована з eHealth України та забезпечує функціонування широкого спектру гнучких модулів, які дозволяють адаптувати можливості системи до потреб кожного медичного закладу. Такі модулі, як реєстратура, управління персоналом та управління організацією, становлять основу цієї платформи, але можливості її розширення додатковими модулями, включаючи облік послуг та архів медичних зображень PACS, роблять її ще більш потужним інструментом. EMCiMED також забезпечує надійний захист даних за допомогою шифрування інформації, а також підтримує інтеграцію з іншими продуктами, що забезпечує її універсальність та гнучкість. Рекомендація МОЗ України підтверджує високий рівень якості та надійності цієї системи, при цьому можливі недоліки не суттєво впливають на її функціональність.

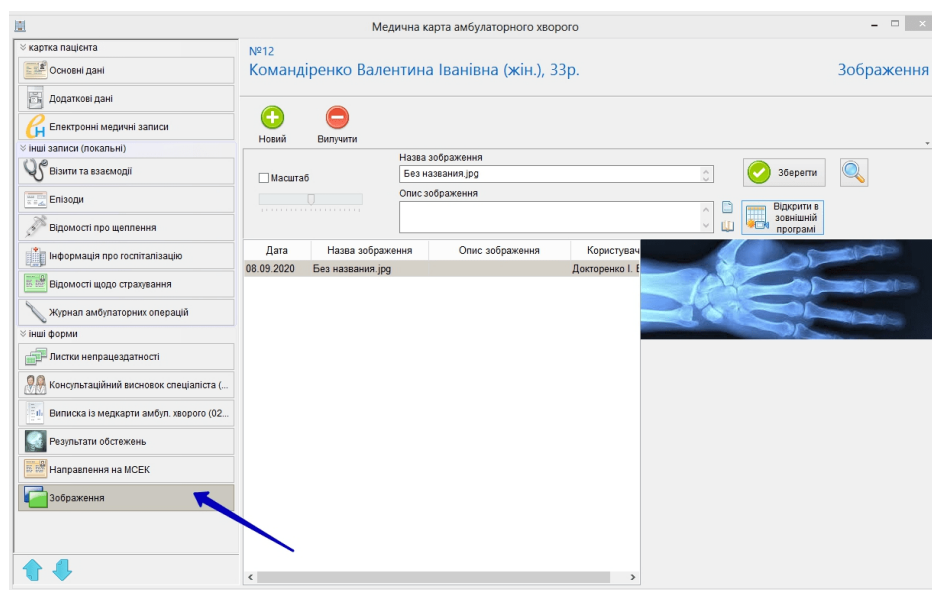


Рисунок 1.5 - Інтерфейс медичної ІС EMCiMED

Нижче наведений порівняльний аналіз згаданих медичних інформаційних систем (табл. 1.1).

Таблиця 1.1 - Порівняльний аналіз програмних продуктів

Вимога	Medcart24	Helsi	Medics	EMCiMED
Вимоги до функціональності системи				
розмежування доступу	+	+	+	-
виведення додаткової графічної інформації	+	+	+	+
висновок на друк	+	+	+	+
відстеження здійснених операцій та транзакцій	+	+	+	+
робота з текстовими даними	+	+	+	+
робота з біометричними даними	-	-	-	-
Вимоги щодо зручності використання				
естетичний та логічний інтерфейс	+	+	+	+
експлуатаційна документація	+	+	-	+
можливість відновлення системи після збоїв	+	+	+	+

Продовження таблиці 1.1

можливість відновлення зв'язку із сервером після збоїв	+	+	-	-
Вимога до продуктивності				
швидкість запуску	+	+	+	+
час відгуку системи	+	+	+	+
пропускна спроможність системи	+	+	-	+
Вимоги до підтримки				
можливість розширення системи	+	+	-	-
безкоштовне розповсюдження	-	-	-	-
Підсумок:	13	13	9	10

Таким чином, в результаті аналітичного огляду та оцінки систем у таблиці було виявлено та підраховано остаточні результати, що відображають рівень відповідності цих систем вимогам предметної області. З аналізу таблиці видно, що кожен із програмних продуктів, які розглядаються, має широкий функціонал роботи, а отже задовольняє більшості встановлених вимог. Однак слід зазначити, що дані системи мають помітні недоліки, у тому числі відсутність можливості безкоштовного поширення, відсутність підтримки біометричних даних, обмеження у можливості розширення системи, а також недостатня експлуатаційна документація, що є важливими критерієм.

Після вивчення та аналізу зазначених вище аналогів розроблюваної інформаційної системи можна виявити і коректно сформулювати ряд функціональних і не функціональних вимог до системи, що розробляється.

РОЗДІЛ 2. РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Аналіз і специфікація вимог до інформаційної системи

При розробці вимог надзвичайно важливим є врахування потреб користувачів та особливостей роботи медичного закладу для подальшого створення ефективної та зручної у використанні системи.

2.1.1. Бізнес вимоги

Бізнес-вимоги відіграють не менш важливу роль у розробці ІС, оскільки вони допомагають зрозуміти, що необхідно створити, як це зробити та що для цього потрібно. Вони також визначають, які задачі має вирішувати система у процесі своєї роботи, а також встановлюють контекст і дозволяють вимірювати переваги, які бізнес сподівається отримати від реалізації проекту. Організації не повинні розпочинати будь-який проект без чіткого розуміння цінності, яку він додасть для бізнесу. Вимоги являють собою конкретні та вимірні характеристики, що впливають із потреб поліклініки в організації ефективного управління записами пацієнтів, оптимізації робочого процесу реєстраторів та забезпечення якісного обслуговування пацієнтів.

Ці вимоги впливають із наступних *бізнес-цілей*:

Створення інформаційної системи реєстратури поліклініки, що забезпечує ефективне та безпечне управління медичною інформацією, процесами запису на прийом, покращує якість обслуговування пацієнтів та

підвищує ефективність роботи персоналу, при цьому доводячи рентабельність інвестицій у розробку системи;

Максимізація доступності, якості медичних послуг, рівня задоволеності громадян та забезпечення безперервності медичної допомоги;

Підвищення продуктивності управління медичною установою, ефективності роботи персоналу, мінімізація можливих помилок та витрат на виконання рутинних завдань, полегшення аналізу роботи поліклініки, покращення статистики відвідуваності;

Підвищення ефективності прийняття управлінських рішень;

Підвищення ключових показників ефективності (KPI) персоналу та досягнення відповідності нормам і стандартам охорони здоров'я;

Мінімізація часу очікування на прийом та обслуговування пацієнтів;

Підвищення інноваційності і конкурентоспроможності клініки;

Оптимізація фінансових потоків;

Зниження операційних витрат на 5%.

Для досягнення поставлених цілей необхідно виконати такі *вимоги*:

Автоматизація звітування та аналізу стану здоров'я пацієнтів;

Автоматизація процесів запису на прийом до лікаря та формування черги;

Забезпечення повної автоматизації обліку пацієнтів;

Впровадження електронного документообігу;

Інтеграція модулів звітності та статистики;

Впровадження інструментів аналізу даних;

Впровадження автоматизованих процесів запису на прийом;

Дотримання законодавчих та нормативних вимог;

Інтеграція з національною системою охорони здоров'я;

Реалізація аналізу даних і прогнозної аналітики;

Забезпечення відповідності національним та міжнародним стандартам охорони здоров'я;

Автоматизація рутинних завдань реєстратури;

Розробка звітних та аналітичних інструментів.

Усі вище наведені вимоги представлені у вигляді діаграми вимог SysML (рис. 2.1).

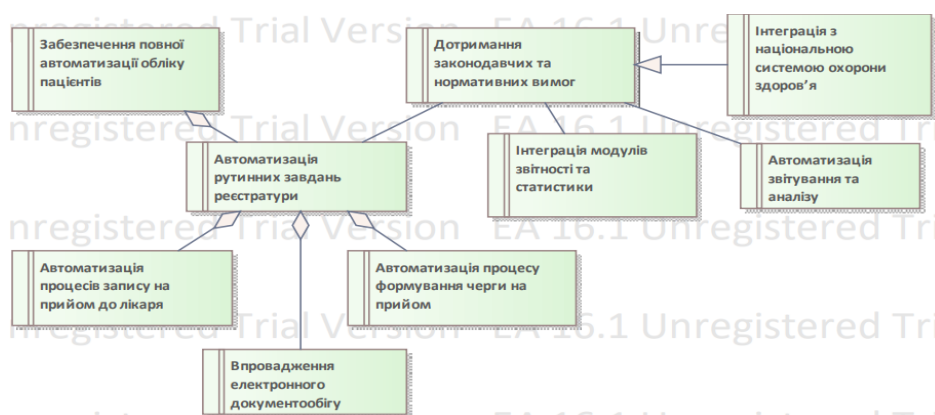


Рис. 2.1 - Діаграма бізнес-вимог

2.1.2. Функціональні вимоги

Функціональні вимоги описують те, що розробники повинні реалізувати, щоб дозволити користувачам виконувати свої завдання (вимоги користувача), таким чином задовольняючи бізнес-вимоги. Часто функціональні вимоги формулюються через твердження про необхідність певних дій чи можливостей. Наприклад, "Користувач повинен мати можливість..." або "Система повинна автоматично...".

На основі бізнес-вимог були виділені функціональні вимоги до системи. Для їхньої візуалізації використовується діаграма функціональних вимог SysML (рис. 2.2).

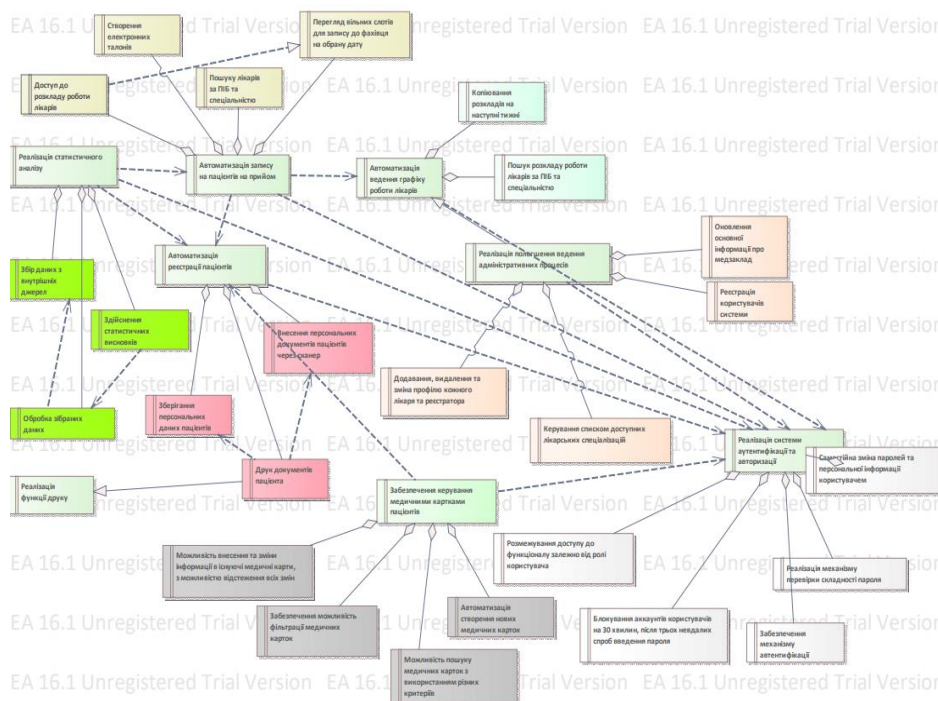


Рис. 2.2 - Діаграма функціональних вимог

Функціональні вимоги виступають основою для розробки та тестування ІС. Вони виявляють, які саме функції має містити система, щоб якомога точніше забезпечити відтворення вимоги користувачів та відповідати описаним бізнес цілям. Точний опис функціональних вимог дає змогу правильно побудувати архітектуру системи, з попередньою можливістю виключення ризиків та скорочення термінів впровадження системи.

Нижче наведені основні функціональні вимоги до системи.

Забезпечення можливості проведення статистичного аналізу.

Система має здійснювати збір даних з внутрішніх джерел;

Необхідно забезпечити обробку зібраних даних для їхнього перетворення у вигляді, придатному для аналізу та візуалізації;

Система має здійснювати аналіз зібраних даних для виявлення тенденцій, патернів та кореляцій між різними параметрами;

Система має створювати статистичні звіти використання робочих ресурсів медичного закладу.

Забезпечення реєстрації пацієнтів.

Система має забезпечувати можливість зберігання персональних даних пацієнтів;

Система повинна мати можливість внесення персональних документів пацієнтів через сканер;

Система повинна мати можливість створення та зберігання медичної карти пацієнта, талонів запису на прийом та друку цих документів;

Система має реалізовувати можливість пошуку пацієнтів з використанням різних критеріїв, таких як ПІБ пацієнта та ПІН;

Функціонал системи повинен надавати можливість внесення та зміни інформації в існуючі записи пацієнтів;

Система повинна автоматизувати додавання нових пацієнтів.

Організація системи запису на прийом до медичних фахівців.

Система має гарантувати швидкий доступ до розкладу роботи лікарів;

Користувач повинен мати можливість пошуку лікарів по ПІБ;

Користувач повинен мати можливість виконувати пошук лікарів за спеціальністю;

Користувач повинен мати можливість переглядати вільні слоти для запису до фахівця на обрану дату;

Система має автоматизувати процес запису пацієнтів на прийом до лікаря за допомогою електронних талонів;

Система має реалізовувати перехід з поля встановлення знаходження пацієнта та необхідного лікаря до відображення розкладу для запису на прийом;

Впровадження вбудованої системи обліку навантаження лікарів.

Забезпечення ведення графіку роботи медичного персоналу.

Система має автоматизувати процес копіювання розкладів на наступні тижні;

Система має автоматизувати процес ведення розкладу роботи лікарів;

Користувач повинен мати можливість виконувати пошук розкладу роботи лікарів за ПІБ та спеціальністю.

Підтримка функціоналу для ефективного управління адміністративними процесами поліклініки.

Можливість створення опису медичного закладу та оновлення основної інформації про нього;

Система має забезпечувати реалізація функціоналу призначення та реєстрації користувачів системи реєстратури медичної установи;

Функціонал системи повинен включати можливість додавання, видалення та зміни інформації кожного лікаря та реєстратора, включаючи персональні дані, спеціалізацію, контактну інформацію та розклад прийому лікарів;

Реалізація функціоналу для керування списком доступних лікарських спеціалізацій.

Реалізація функції друку.

Система повинна забезпечувати:

Можливість друку розкладу лікаря на певну дату;

Можливість друку карти пацієнта;

Можливість друку талону на запис до лікаря;

Можливість друку документів пацієнта;

Можливість друку факсимільє та логотипу медичного закладу;

Друк інформованої згоди на обробку персональних даних.

Забезпечення безпеки та контролю доступу користувачів до системи, включаючи аутентифікацію та авторизацію.

Реалізація системи керування правами доступу для розмежування доступу до функціоналу залежно від ролі користувача;

Система має забезпечувати механізм автентифікації для усіх користувачів з можливістю використання унікальних паролей;

Система повинна забезпечувати зберігання паролей у відповідності до стандартів безпеки та конфіденційності;

Користувач повинен мати можливість керувати своїми даними, що передбачає зміну паролей та персональної інформації;

Система повинна реалізовувати механізм перевірки складності пароля для його подальшого захисту.

Забезпечення керування медичними картками пацієнтів.

Система повинна автоматизувати створення нових медичних карток для пацієнтів;

Функціонал системи повинен надавати можливість внесення та зміни інформації в існуючі медичні карти, з можливістю відстеження та аудиту всіх змін, що здійснені для забезпечення прозорості та надійності даних;

Система має реалізовувати можливість пошуку медичних карток з використанням різних критеріїв, таких як ПІБ пацієнта, ідентифікатор картки, дата народження;

Функціонал системи має забезпечити можливість фільтрації медичних карток за їх поточним статусом, включаючи активні та архівні картки.

Додаткові функціональні вимоги.

Реалізація цілодобового, швидкого та безперервного протікання інформаційних процесів;

Система має забезпечувати роботу 24 години на добу, 7 днів на тиждень, 365 днів на рік;

Система повинна демонструвати рівень доступності, за якого допустимий час простою:

за годину - не більше 5 хвилин;

на день - не більше 1 година;

на місяць - не більше 10 годин.

2.1.3. Нефункціональні вимоги

Нефункціональні вимоги визначають характеристики системи, не пов'язані безпосередньо з її функціональністю, але що впливають на її працездатність, надійність, безпеку та зручність використання. Вони описують характеристики продукту в різних вимірах, які важливі як для користувачів, так і для розробників і супроводжувачів, наприклад, продуктивність, безпека, доступність і портативність. Інші класи нефункціональних вимог описують зовнішні інтерфейси між системою та зовнішнім світом. До них належать з'єднання з іншими програмними системами, апаратними компонентами та користувачами, а також комунікаційні інтерфейси.

Інтерфейс користувача [39].

Система має забезпечувати візуальний графічний інтерфейс (GUI) для взаємодії користувачів з інформаційною системою;

Інтерфейс системи має бути простим, інтуїтивно зрозумілим та заснований на уніфікованих формах введення інформації;

Система повинна дотримуватись загальних правил оформлення екранів, включаючи шрифти, колірну палітру, розташування вікон та елементів керування, а також правила використання клавіатури та миші.

Уніфікація інтерфейсу.

Основні елементи інтерфейсу мають бути стандартизовані та оформлені в одному стилі, необхідно дотримуватися приблизного розташування елементів та форм на кожній сторінці системи.

Поведінка інтерфейсу.

Поведінка аналогічних елементів інтерфейсу повинна бути однаковою для всіх ідентичних елементів, а отже усі аналогічні групи інтерфейсів повинні однаково реагувати на взаємодію користувача з ними.

Для побудови основного інтерфейсу системи слід використовувати стандартний графічний інтерфейс користувача операційної системи MS Windows.

Підтримка мови та локалізації.

Допомога та інформаційні повідомлення мають бути написані українською мовою;

Інтерфейс повинен підтримувати можливість введення та виведення з клавіатури та за допомогою миші для забезпечення зручності користувача.

Безпека.

Система повинна працювати з захищеними базами даних, забезпечуючи при цьому цілісність та конфіденційність даних;

Система має використовувати брандмауери з метою захисту мережевої взаємодії системи;

Система повинна автоматично блокувати акаунти користувачів на 30 хвилин після трьох невдалих спроб введення пароля;

Акаунти повинні бути наділені правами доступу, відповідними до певних посад користувачів (розмежовані права доступу до інформаційної системи з урахуванням принципу мінімальних привілеїв).

Швидкість.

Система повинна забезпечувати відповіді на користувацькі запити впродовж 2-3 секунд;

Система повинна завантажувати сторінку не більше 2 секунд;

Локалізація.

Інтерфейс системи має бути локалізований українською та англійською мовами;

Формати дати та часу мають відповідати формату часової зони поліклініки.

Надійність.

Система має забезпечувати надійність роботи не менше 99% для мінімізації системних збоїв;

Система повинна мати здатність автоматично відновлюватися протягом 10 хвилин без втрати даних.

Навколишнє середовище.

Системна інфраструктура повинна бути розміщена в сухих, прохолодних приміщеннях, та приміщеннях, що підлягають цілодобовому контролю.

Діаграма нефункціональних вимог представлена рисунку 2.3.

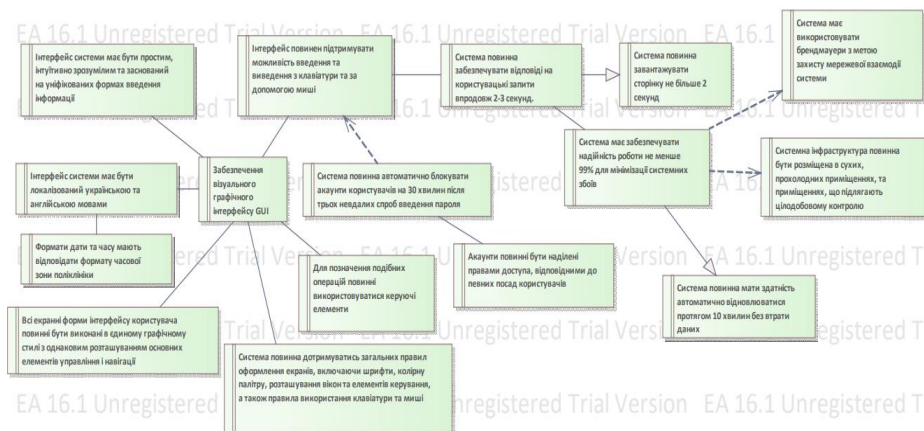


Рис. 2.3 - Діаграми нефункціональних вимог

Для кращої організації моделі використовуємо діаграму пакетів. Діаграма пакетів дозволяє групувати елементи моделі до логічних блоків для спрощення управління та розуміння структури системи. На рисунку 2.4 представлена діаграма пакетів, що відображає різні аспекти вимог до системи.

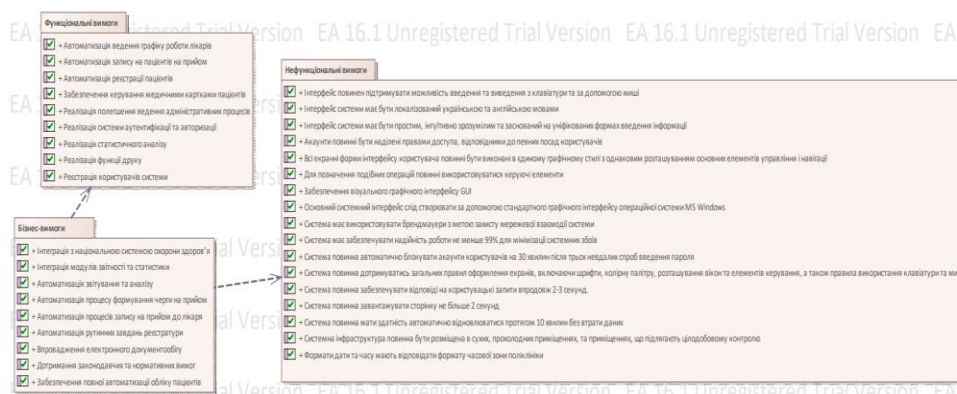


Рис. 2.4 - Діаграма пакетів

Специфікація надає загальне уявлення про вимоги до системи, включаючи їх тип, статус, складність, пріоритетність, ризикованість та примітки. Надалі ці вимоги використовуватимуться для розробки та тестування системи.

Специфікації вимог, підготовлені з використанням менеджера специфікацій Enterprise Architect, представлені у Додатку А.

2.2. Постановка та математичний опис задачі

2.2.1. Постановка задачі

Розробка інформаційної системи реєстрації поліклініки є комплексним завданням, спрямованим на оптимізацію процесів управління прийомом пацієнтів.

Основна задача інформаційної системи реєстратури – запис пацієнта на прийом. Дана задача вирішується щодня з урахуванням динаміки надходження записів на прийом, забезпечуючи цілодобову ефективну організацію робочого процесу та оптимальне розподілення ресурсів медичного закладу.

Техніко-економічна сутність завдання полягає у скороченні часу, що витрачається на управління записами, оптимізації робочого часу медичного персоналу та підвищенні рівня обслуговування пацієнтів.

При розв’язанні задачі здійснюється керування наступними об’єктами: пацієнтами (їх даними), медичними картками, записами на прийом, зареєстрованими та незареєстрованими користувачами (їх даними), графіком роботи лікарів.

Розв’язок задачі припиняється у випадках несправності апаратної або програмної частини інформаційної системи, відсутності доступу до мережі Інтернет, втрати зв’язку між клієнтом та сервером системи, завершенні роботи реєстратури за встановленим розкладом або при досягненні максимальної кількості запису пацієнтів на прийом до певного лікаря.

На рисунку 2.5 показано інформаційну модель задачі.

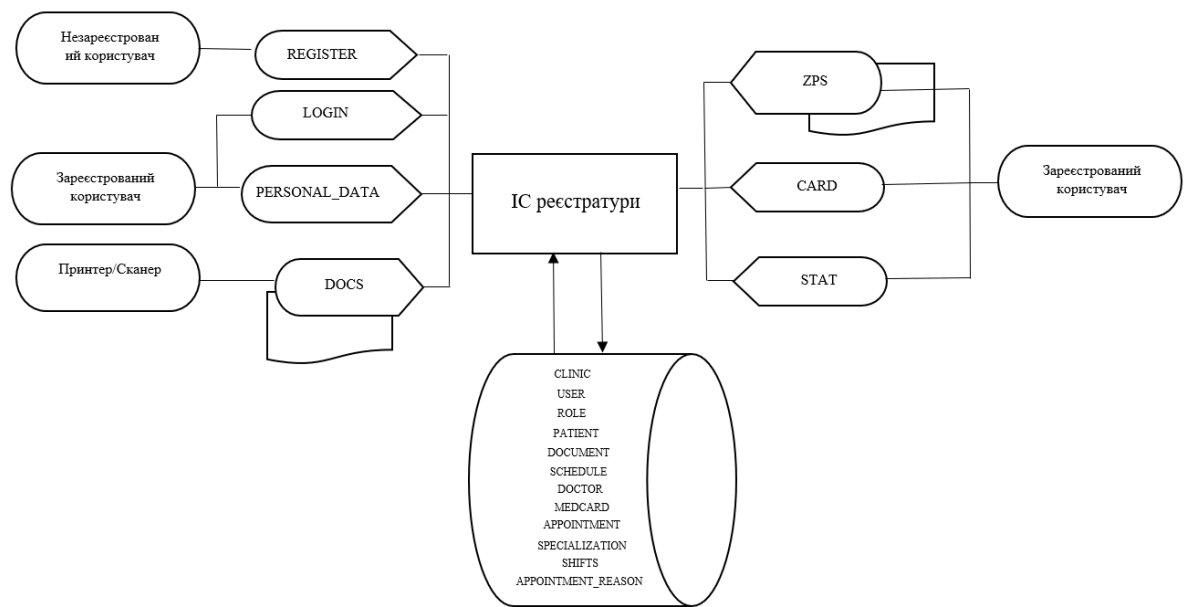


Рисунок 2.5 - Інформаційна модель розв’язання задачі запису пацієнта на прийом

2.2.2. Вихідна інформація

Вихідною інформацією є запис пацієнта на прийом, статистика роботи медичного закладу та медична карта пацієнта. Вихідна інформація необхідна для подання результатів обробки даних користувачам, іншим системам або зовнішнім пристроям. Перелік і опис вихідних повідомлень наведено у табл. 2.1.

Таблица 2.1 – Перелік та опис вихідних повідомлень

№	Назва вихідного повідомлення	Ідентифікатор	Форма подання	Періодичність видання	Термін видання і допустимий час затримки	Користувач інформації
1	Запис на прийом	ZPS	Паперовий документ, екранна форма	При записі пацієнта	Видання в реальному часі при реєстрації; затримка не більше 5 секунд	Зареєстрований користувач

Продовження таблиці 2.1

2	Статистика роботи закладу	STAT	Таблична форма, екранна форма	Щодня, тижня, щомісячно	Після закінчення відповідного періоду (день, тиждень, місяць); затримка не більше 1 хвилини	Зареєстрований користувач
3	Медична карта пацієнта	CARD	Екранна форма	При створенні нової карти та її оновленні	Формування та оновлення в реальному часі при зверненні пацієнта; затримка не більше 5 секунд	Зареєстрований користувач

Запис на прийом (ZPS) містить інформацію про час та дату прийому, ПІБ пацієнта, обраного ним лікаря, реєстратора, що створював запис та причину запису.

Статистика роботи закладу (STAT) включає загальну статистику та динаміку роботи клініки. Показники, такі як загальна кількість звернень, потребують точного підрахунку та аналізу. Ці дані джерело інформації черпає з електронних записів про прийоми та звернення пацієнтів у базі даних клініки.

Медична карта пацієнта (CARD) має загальний опис стану здоров'я пацієнтів (група крові, резус фактор, групу інвалідності), дату створення картки, ПІБ пацієнта, якому дана карта належить, та ПІБ сімейного лікаря пацієнта.

Всі ці повідомлення є невід'ємною частиною роботи медичної інформаційної системи, забезпечуючи точне та своєчасне надання даних для ефективного управління та надання медичного обслуговування.

2.2.3. Вхідна інформація

Вхідна інформація необхідна для функціонування системи, оскільки вона забезпечує систему даними, які вона може обробляти, перетворювати та генерувати на їх основі нову інформацію.

Вхідна інформація ІС наведена нижче в таблиці 2.2.

Таблиця 2.2 – Перелік та опис вхідних повідомлень

Назва вхідного повідомлення	Ідентифікатор	Форма подання	Частота надходження	Джерело	Вимоги щодо повноти
Персональні дані пацієнта	PERSONAL_DATA	Екранна форма	Щоденно	Реєстратор	Повністю заповнена, введена негайно після надходження та має бути точною для забезпечення якісної реєстрації пацієнта в системі
Документи пацієнта	DOCS	Скановані копії документів	Щоденно	Пацієнт/Сканер	Забезпечення повного сканування та завантаження всіх необхідних документів

Продовження таблиці 2.2

Дані авторизації користувачів	LOGIN	Екранна форма	При авторизації	Зареєстрований користувач	Забезпечення повного введення та верифікації імені користувача, пароля, рівня доступу.
Дані реєстрації користувачів	REGISTER	Екранна форма	При реєстрації	Незареєстрований користувач	Забезпечення повного введення та верифікації імені користувача, пароля, імейлу та імейлу.
Масив користувачів	USER	Файл	Під час функціонування системи	База даних	Наявність усієї необхідної інформації без пропусків або відсутності даних.
Масив поліклінік	CLINIC	Файл	Під час функціонування системи	База даних	Наявність усієї необхідної інформації без пропусків або відсутності даних.
Масив ролей користувачів	ROLE	Файл	Під час функціонування системи	База даних	Наявність усієї необхідної інформації без пропусків або відсутності даних.
Масив пацієнтів	PATIENT	Файл	Під час функціонування системи	База даних	Наявність усієї необхідної інформації без пропусків або відсутності даних.
Масив документів пацієнтів	DOCUMENT	Файл	Під час функціонування системи	База даних	Наявність усієї необхідної інформації без пропусків або відсутності даних.
Масив графіків роботи	SCHEDULE	Файл	Під час функціонування системи	База даних	Наявність усієї необхідної інформації без пропусків або відсутності даних.

Продовження таблиці 2.2

Масив лікарів	DOCTOR	Файл	Під час функціонування системи	База даних	Наявність усієї необхідної інформації без пропусків або відсутності даних.
Масив медичних карток	MEDCARD	Файл	Під час функціонування системи	База даних	Наявність усієї необхідної інформації без пропусків або відсутності даних.
Масив записів на прийом	APPOINTMENT	Файл	Під час функціонування системи	База даних	Наявність усієї необхідної інформації без пропусків або відсутності даних.
Масив спеціалізацій	SPECIALIZATION	Файл	Під час функціонування системи	База даних	Наявність усієї необхідної інформації без пропусків або відсутності даних.
Масив робочих змін	SHIFTS	Файл	Під час функціонування системи	База даних	Наявність усієї необхідної інформації без пропусків або відсутності даних.
Масив причин записів на прийом	APPOINTMENT_REASON	Файл	Під час функціонування системи	База даних	Наявність усієї необхідної інформації без пропусків або відсутності даних.

Персональні дані пацієнта включають інформацію про особистість пацієнта, таку як ПІБ, дата народження, адреса, контактна інформація, ПІН та імейл. Ці дані вимагають високого ступеня захисту та точності, оскільки є основою обліку та надання медичних послуг.

Документи пацієнта є різноманітними документами, що підтверджують особу пацієнта та надходять у систему у вигляді сканованих документів.

Документ у системі має назву, що співпадає з ПІБ пацієнта, якому належить цей документ.

Дані авторизації користувачів включають облікові дані, необхідні для доступу до медичної інформаційної системи. Безпека та конфіденційність цих даних є критично важливими для запобігання несанкціонованому доступу.

Дані реєстрації користувачів містять інформацію про користувачів системи, таку як ПІБ, посада, контактні дані, пароль, імейл. Ці дані використовуються для організації роботи працівників у рамках медичної установи та забезпечення безпечного доступу до інформаційних ресурсів.

Для своєї роботи система використовує інформацію, що зберігається в масивах.

Кожен з цих елементів вхідної інформації відіграє ключову роль у роботі медичної інформаційної системи, забезпечуючи ефективне управління медичними даними та ресурсами.

2.2.4. Математичний опис

У рамках розробки інформаційної системи реєстратури поліклініки проводиться математичне моделювання процесу реєстрації нових та прийому повторних пацієнтів. Використовуючи методи теорії ймовірностей та статистики, формуються моделі для розподілу навантаження на реєстратуру у різні часові інтервали. Ці моделі надають основу для оптимального розподілу робочого часу працівників реєстратури та зменшення часу очікування пацієнтів.

1. Модель роботи реєстратури

$$T_{ab} = \sum (T_{on} \times P_n), \text{ де}$$

T_{ab} – загальний час створення а-того запису b-того пацієнта (1)

T_{on} - час обслуговування одного пацієнта n-ого типу

P_n – кількість пацієнтів n-ого типу

2. Завантаженість реєстратури

$D_{ab} = R_{nc} / H_c$, де

D_{nc} – середня кількість зареєстрованих пацієнтів n-ого типу c-ого періоду;

R_{nc} – кількість зареєстрованих пацієнтів n-ого типу за c-ий період; (2)

H_c – кількість відпрацьованих годин за c-ий період;

3. Моделювання потоку пацієнтів

$N_{pzt}(t+1) = \alpha N_n(t) + (1-\alpha)N_{n_fct}(t)$, де

$N_n(t)$ – кількість пацієнтів n-ого типу у момент часу t (3)

$N_{n_fct}(t)$ – фактична кількість пацієнтів n-ого типу, що зареєструвалися

α - коефіцієнт згладжування

4. Моделювання розкладу лікарів

$\min(TC_{nk} + TB_{nk})$, за умови

$N_k * TB_{nk} \geq R_{nc}$, де

TC_{nk} – час очікування пацієнта n-ого ти на прийом k-того спеціаліста

TB_{nk} – час обслуговування пацієнта n-ого типу k-того спеціаліста (4)

N_k – кількість лікарів k-того типу

2.2.4. Алгоритм розв'язання задачі

Розуміння алгоритму розв'язання задачі за допомогою автоматизованої системи реєстратури відіграє ключову роль в оптимізації процесів керування пацієнтами та медичними даними. Далі описано основну послідовність дій, які виконує інформаційна система з моменту отримання вхідних даних до надання певного результату (див. рис. 2.6 – 2.11).

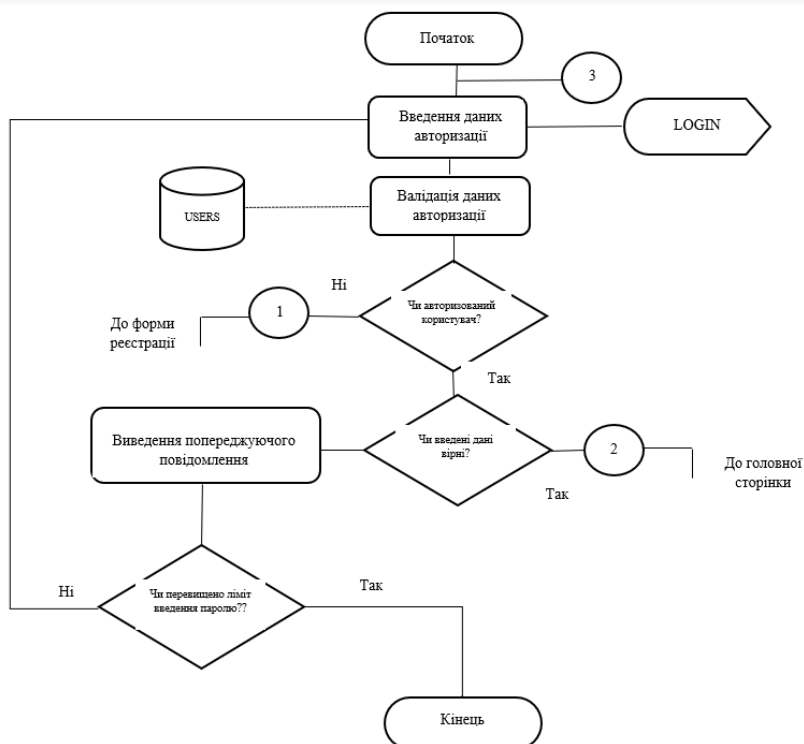


Рисунок 2.6 - Алгоритм розв'язання задачі. Лист 1

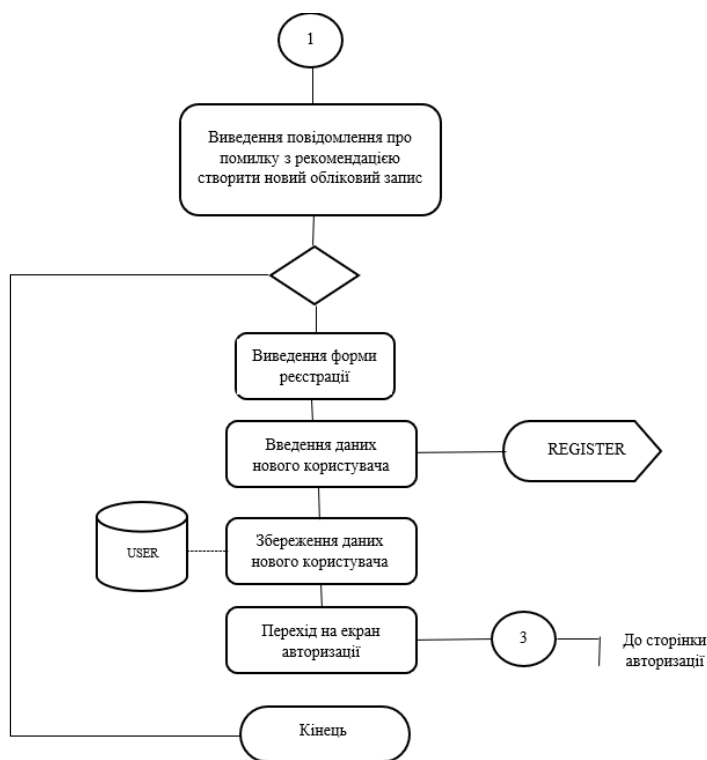


Рисунок 2.7 - Алгоритм розв'язання задачі. Лист 2

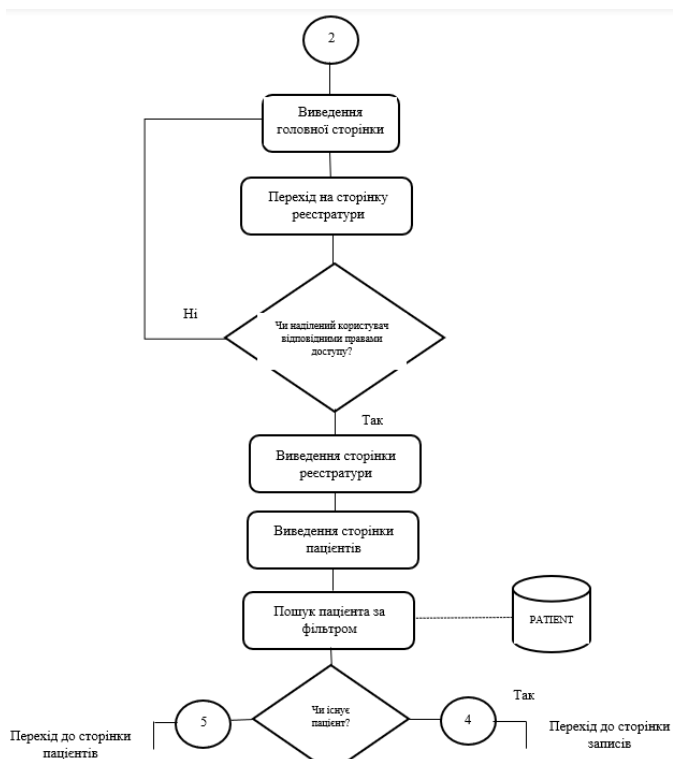


Рисунок 2.8 - Алгоритм розв'язання задачі. Лист 3

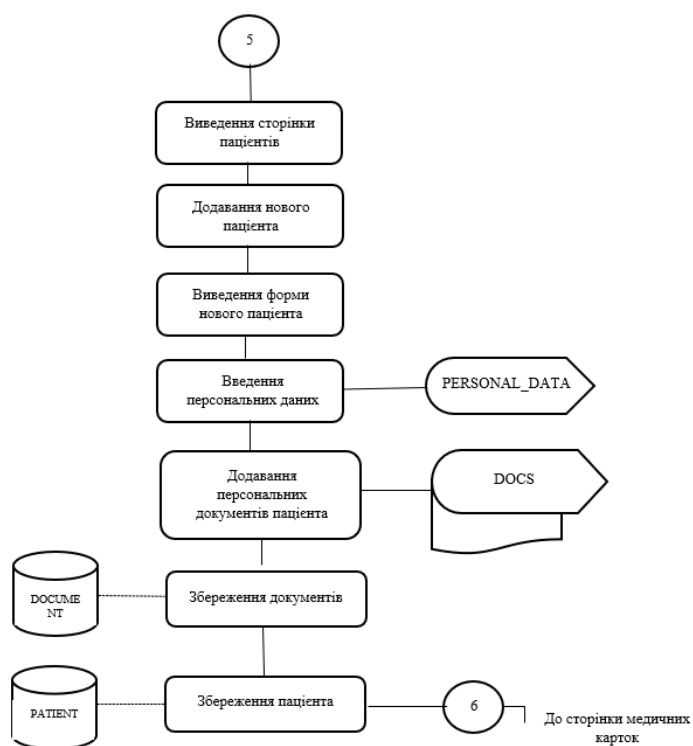


Рисунок 2.9 - Алгоритм розв'язання задачі. Лист 4

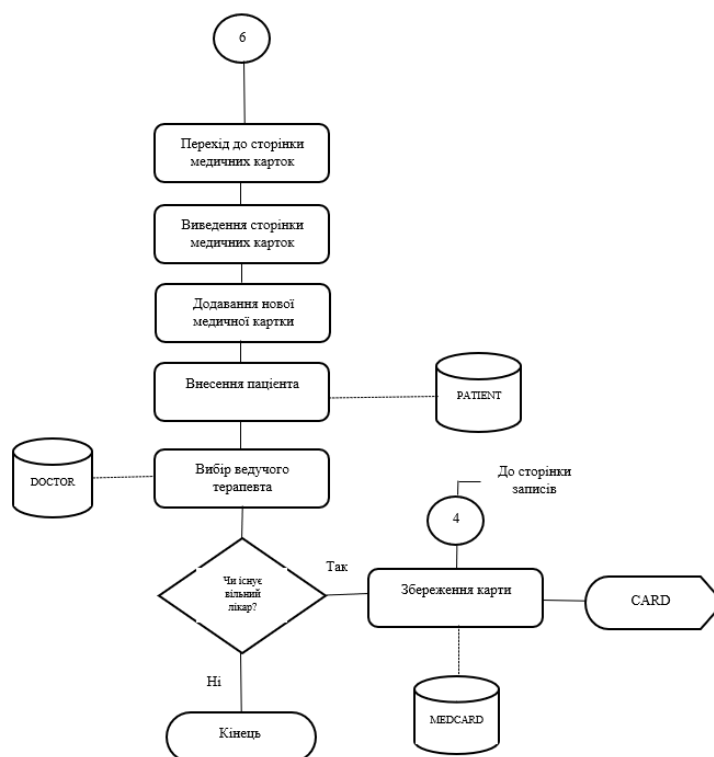


Рисунок 2.10 - Алгоритм розв'язання задачі. Лист 5

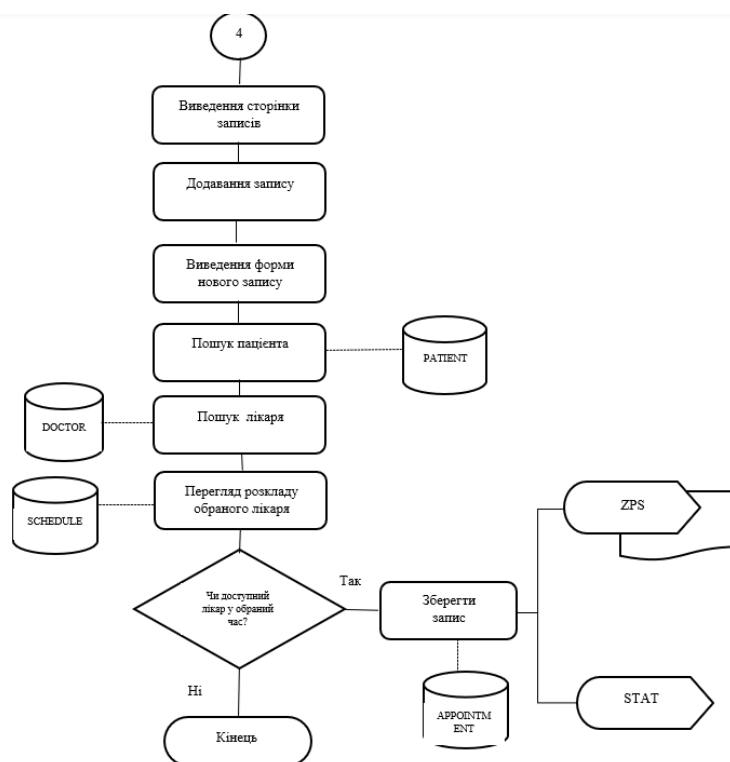


Рисунок 2.11 - Алгоритм розв'язання задачі. Лист 6

2.3. Моделювання інформаційної системи

Основою проектування ІС є моделювання її предметної області. Моделювання інформаційної системи поліклініки спрямоване на створення абстрактних моделей, які відбивають її структуру, функціональність і взаємодію з зовнішніми компонентами. Даний процес не тільки надає повне розуміння коректного функціонування майбутньої ІС, але й допомагає виявити вузькі місця та проблеми у роботі системи, що дозволяє оптимізувати її процеси та підвищити ефективність. Це допомагає зменшити та структурувати час, необхідний для проектування, та отримати більш якісний результат розробки проекту. Упущення етапу моделювання предметної області може згодом призвести до створення помилок під час прийняття важливих стратегічних рішень, фінансових збитків та необхідності подальшої дорозробки системи з додатковими витратами.

Далі наведено огляд різних методів та підходів до моделювання, які широко застосовуються в сучасній практиці розробки інформаційних систем.

2.3.1. Моделювання та аналіз бізнес-процесів

Перш ніж розпочати розробку майбутньої інформаційної системи, необхідно провести моделювання та аналіз бізнес-процесів, які будуть підтримуватися цією системою. Бізнес-процеси являють собою сукупність взаємопов'язаних ресурсів та послідовності дій, що перетворюють вхідні елементи у вихідні та використовуються для досягнення конкретної мети. Моделювання бізнес-процесів дозволяє розробникам та зацікавленим

сторонам зрозуміти поточні процеси, виявити вузькі місця та ефективно впровадити зміни, спрямовані на покращення роботи системи.

В рамках моделювання використовуються різні нотації та методи, такі як BPMN (Business Process Model and Notation) та IDEF (Integrated Definition), які дозволяють формалізувати процеси та візуалізувати їх для більш глибокого аналізу.

BPMN є стандартною мовою моделювання бізнес-процесів, яка служить сполучною ланкою між формалізацією і візуалізацією бізнес-процесів, іншими словами вона дозволяє детально описати основні кроки та операції у рамках бізнес-процесів.

BPMN представляє набір графічних елементів, які використовуються для створення схеми потоку виконання бізнес-процесу, забезпечуючи зрозуміле і наочне уявлення його структури та логіки.

Нижче на рисунку 2.12 описується процес роботи реєстратури.

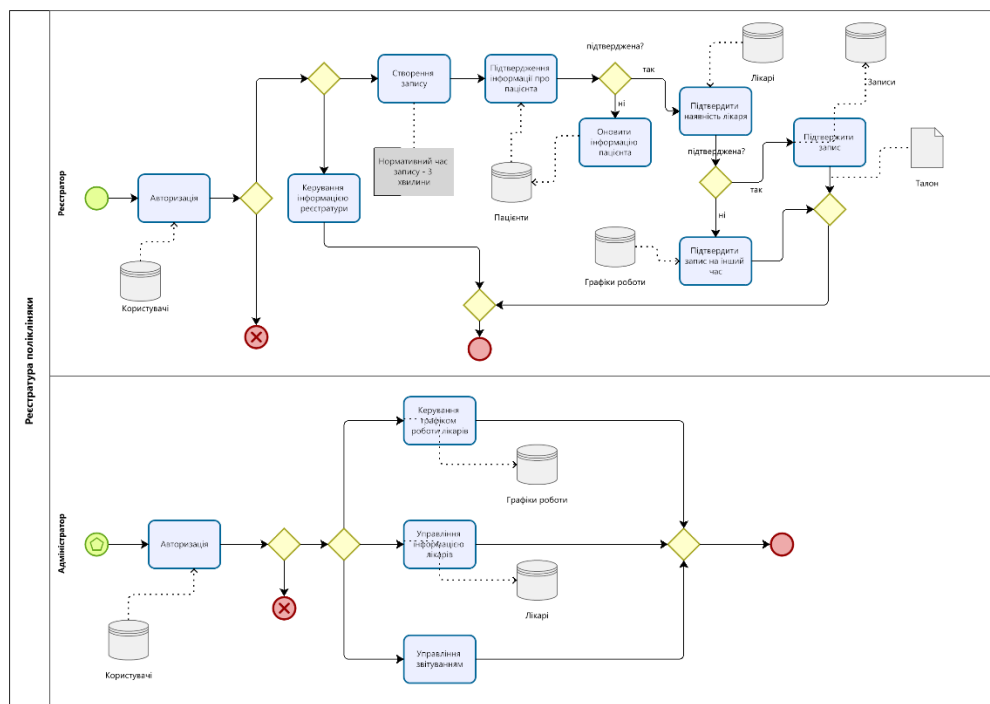


Рисунок 2.12 - BPMN-діаграма бізнес-процесу медичної інформаційної системи реєстратури

Таким чином, застосування BPMN діаграм у контексті моделювання бізнес-процесів інформаційної системи поліклініки є важливим інструментом для більш глибокого розуміння роботи інформаційної системи, оскільки вони надають ясне уявлення про потоки інформації, етапи виконання робіт та взаємозв'язки між різними процесами у системі.

2.3.2. Моделювання поведінки системи

Для наочного представлення функціональних можливостей системи та їх взаємозв'язків використовується діаграма прецедентів UML. Вона описує статичну поведінку системи в процесі, як система взаємодіятиме з навколишнім середовищем, які дії виконуватиме та які результати повинні очікуватись від цих дій.

Базовими елементами діаграми прецедентів є актори та варіанти використання. Актори – це певні ролі (користувачі системи), які взаємодіють із варіантами використання. Варіанти використання, або прецеденти – функціональні можливості системи, які можуть виконуватися у процесі взаємодії з акторами.

Діаграма прецедентів для системи реєстратури поліклініки представлена на рисунку 2.16.

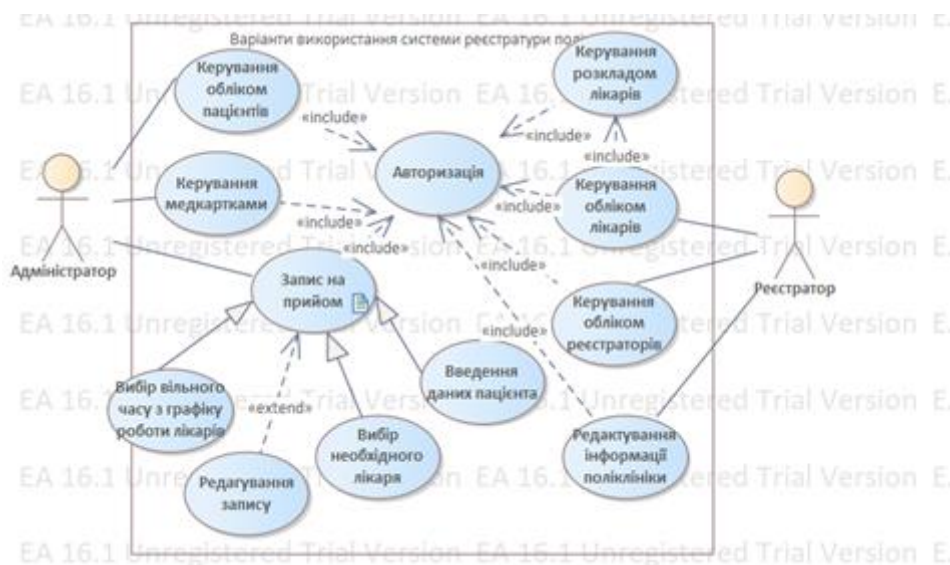


Рисунок 2.16 - Діаграма варіантів використання програмної системи

Актори, представлені на діаграмі, – «Реєстратор», «Адміністратор».

Прецеденти «Керування обліком лікарів», «Керування обліком реєстраторів» та «Редагування інформації поліклініки» доступні для використання актором «Адміністратор». Дані прецеденти передбачають можливість вносити зміни у дані про лікарів, їх спеціалізації, кабінети та графік роботи; керувати даними про реєстраторів, їх облікові записи, а також керувати загальною інформацією про поліклініку, таку як контактна інформація, адреса, години роботи та інше.

Прецеденти «Керування обліком пацієнтів», «Керування медкартками» та «Запис на прийом» забезпечують ефективне управління інформацією про пацієнтів, актуальною та повною медичною історією кожного пацієнта та дозволяє реєстратурі швидко організувати запис пацієнта на прийом.

Для динамічного моделювання системи існує діаграма послідовності, що описує послідовність прийому та відправки повідомлень між об'єктами системи. В якості прикладу взаємодії всередині системи розглянемо прецедент «Запис пацієнта на прийом» (рис. 2.17).

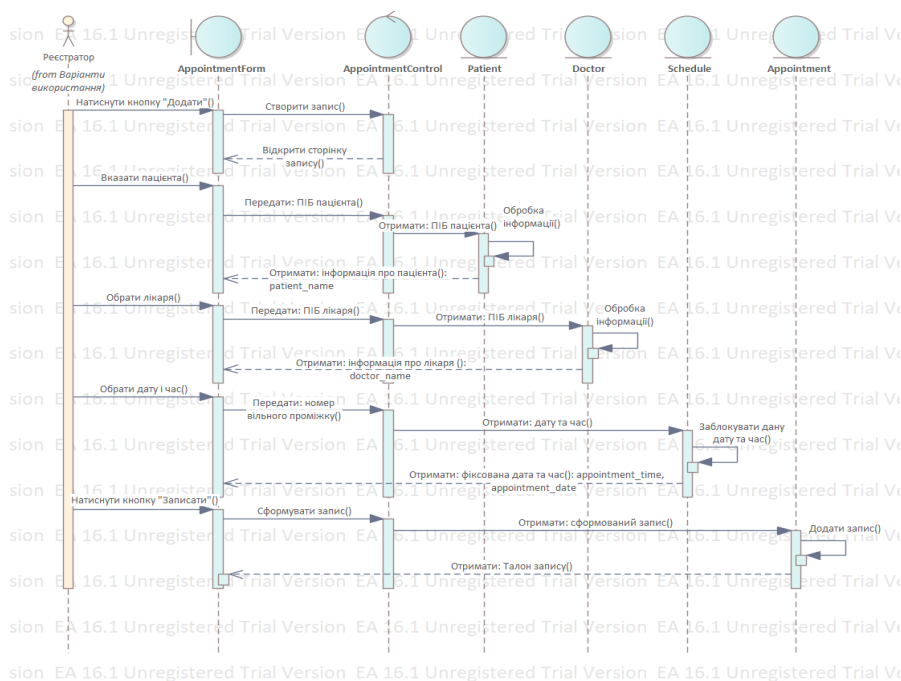


Рисунок 2.17 - Діаграма послідовності запису пацієнта на прийом

Із рисунка видно, що контролери забезпечують обробку запитів користувачів та здійснюють доступ до даних, що зберігаються в базі даних, а також відіграють роль посередника між користувачем інформаційної системи й її базою даних, забезпечуючи коректне виконання запитів користувача та управління даними у системі. Наступні діаграми послідовності наведені у додатку Є.

2.3.3. Моделювання структури системи

Модель структури системи – це та складова проектування, яка допомагає визначити основні компоненти системи та їх взаємозв'язок. Дана модель включає основні класи системи, що розроблюється, їх атрибути та зв'язки між класами. Даний процес проектування допомагає полегшити розуміння архітектури майбутньої системи.

Діаграми прецедентів та послідовності допомогли визначити функціональність системи, уточнити вимоги та визначити структуру та взаємозв'язки між класами для їх реалізації. На основі побудованих вище діаграм прецедентів та послідовності було розроблено діаграму класів, наведену на рисунку 2.18, які відображає статичний внутрішній пристрій програми.

Рисунок наочно демонструє основні компоненти та взаємозв'язки між ними всередині системи. Аналіз класів допомагає зрозуміти, із чого складається система та які взаємозв'язки між її елементами, що полегшує розуміння функціональних властивостей системи та потенційних областей її оптимізації.

Тестування інформаційної системи – це процес, який використовується для визначення правильності, повноти, цілісності даних, безпеки та якості

розробленого проекту. Цей процес призначений для виявлення інформації, пов'язаної з якістю продукту, шляхом виявлення помилок.

Верифікація системи включає тестування апаратного та програмного забезпечення в реальних умовах, забезпечення повноти документації та впровадження заходів безпеки.

Трасування вимог значно полегшує процес верифікації системи, оскільки воно дозволяє встановити прямий зв'язок між встановленими раніше вимогами та певними складовими її архітектури, забезпечуючи прозорість взаємозв'язків між цими елементами, сприяючи успішній реалізації проекту.

Процес трасування вимог є важливою частиною процесу розробки програмного забезпечення, оскільки він дозволяє виявити можливі існуючі прогалини у розробленій системі, а також перевіряє узгодженість вимог одна з одною та їх підтримку попередньо-визначених бізнес-цілей. Отже, сам процес трасування можна визначити, як спеціалізований інструмент управління, який допомагає інженерам та розробникам відстежувати тестування та переконатися, що розроблений продукт відповідає вимог. Додатково, відстеження надає можливість описати зв'язки між вимогами, прецедентами, варіантами тестування, елементами інтерфейсу користувача та проектними компонентами інформаційної системи реєстратури поліклініки. Для забезпечення верифікації проектних рішень використовуються діаграми трасування та матриці взаємозв'язків.

Діаграма трасування являє собою графічне зображення взаємозв'язків між елементами системи та забезпечує розуміння того, як кожна частина системи відповідає встановленим вимогам.

Діаграма трасування представлена на рисунку 2.18.

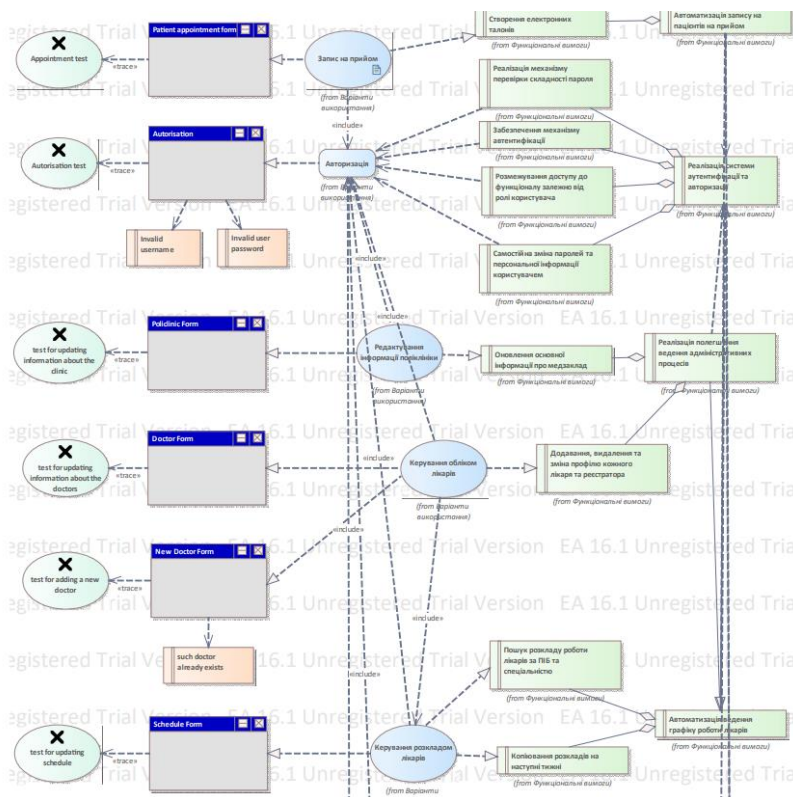
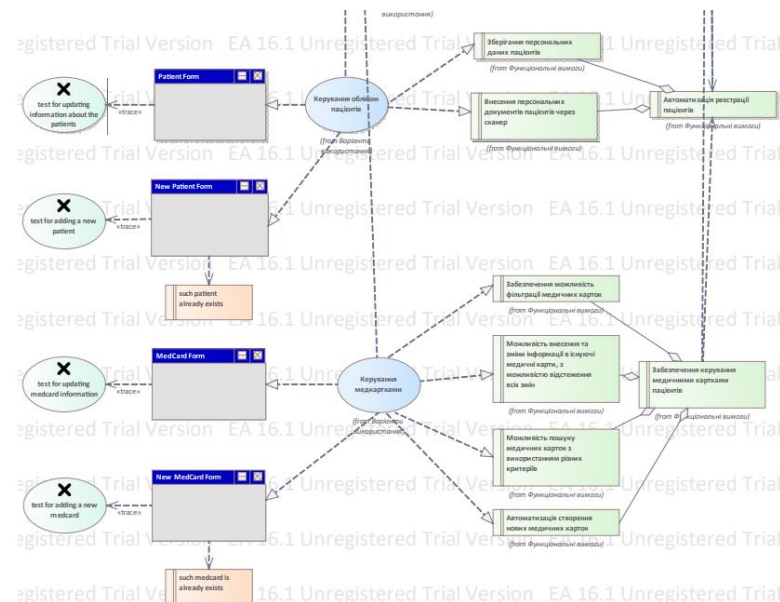


Рисунок 2.18 - Діаграма трасувань вимог



Продовження рисунку 2.18

На діаграмі визначені такі елементи:

Вимоги: перелік функціональних вимог до системи.

Кожен елемент матриці свідчить про наявність зв'язку між відповідними елементами трасування.

Таким чином, діаграма трасування та матриці взаємозв'язків забезпечують зручний інструмент для верифікації проектних рішень шляхом відстеження відповідності між вимогами системи та її компонентами.

РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ ПІДСИСТЕМИ

3.1. Інформаційне забезпечення

Серед найважливіших підсистем, що гарантують успішну роботу інформаційної системи реєстратури поліклініки, є інформаційне забезпечення. В загальному розумінні інформаційне забезпечення трактується як об'єднання єдиної системи класифікації, документації та інформаційної бази. Даний елемент дозволяє однозначно представляти інформацію, ефективно проводити її аналіз та обробку й оптимізувати взаємодію системи з користувачем.

Інформаційне забезпечення має відповідати певним вимогам:

Достатність автоматизованих функцій;

Використання класифікаторів, що були затверджені замовником;

Сумісність з вже існуючими забезпеченнями системи;

Підтримка контролю даних та їх захист від несанкціонованого доступу;

Відповідність корпоративним стандартам ведення документації.

Організація збору та передачі первинної інформації в інформаційній системі реєстратури поліклініки повинна відповідати таким вимогам:

Своєчасність подання інформації: Інформація має надаватися на обробку в інформаційну систему безперервно та у встановлені терміни для забезпечення оперативності системи;

Точність та достовірність даних: Інформація, що надходить до системи, має бути повною, точною та достовірною, щоб уникнути помилок під час формування медичних записів;

Безпека та конфіденційність: Забезпечення захисту конфіденційних даних від несанкціонованого доступу та використання;

Періодичність надання інформації: Інформація повинна надаватися в інформаційну систему згідно з встановленим графіком роботи та потребами системи;

Способи та носії надання інформації: Інформація може надаватися у вигляді електронних документів, через мережеві з'єднання або за допомогою спеціалізованих протоколів передачі даних;

Схема інформаційного забезпечення представлена рисунку 3.1, який у загальному вигляді описує позамашинну та внутрішньомашинну складові ІЗ.



Рисунок 3.1 - Схема інформаційного забезпечення

Під час розробки інформаційного забезпечення важливим аспектом є побудова системи класифікації та кодування даних. Даний процес відіграє значну роль у забезпеченні структурованого зберігання інформації та сумісності з міжнародними стандартами обробки медичних даних.

Перш ніж розпочати розробку системи кодування, необхідно визначити основні принципи, яких необхідно дотримуватись під час її побудови, щоб у

результаті роботи забезпечити повне структуроване зберігання інформації, зручний пошук та аналіз даних.

Такими принципами є:

Однозначність: Кожна одиниця інформації повинна мати однозначне позначення, щоб унеможливити двозначність;

Ієрархічність: Класифікація повинна бути побудована на принципі ієрархії, де кожна одиниця може включати підкатегорії і додаткові атрибути для більш детального опису;

Доступність: Класифікація повинна бути доступною та зрозумілою для всіх користувачів системи.

Дотримуючись попередньо встановлених правил, була побудована наступна система кодування, яка підтримує точну обробку та облік транзакцій в ІС реєстратури поліклініки:

Код поліклініки

метод кодування - послідовий;

метод класифікації - ієрархічний;

довжина коду - 11 символи;

структура коду: XXX-XXX-XXX - порядковий номер поліклініки.

Код користувача

метод кодування - порядковий;

довжина коду - 2 символи;

структура коду: XX - порядковий номер користувача.

Код ролі користувача

метод кодування - порядковий;

довжина коду – 2 символи;

структура коду: XX - порядковий номер ролі користувача.

Код пацієнта

метод кодування - порядковий;

довжина коду - 4 символи;

структура коду: XXXX- порядковий номер пацієнта.

Код лікаря

метод кодування – серійно-порядковий;

довжина коду - 7 символів;

структура коду: XXX-XXX - порядковий номер лікаря.

Код спеціалізації

метод кодування - паралельний;

метод класифікації - фасетний;

довжина коду - 3 символи;

структура коду: XXX - порядковий номер спеціалізації.

Код талону

метод кодування - порядковий;

довжина коду – 4 символів;

структура коду: XXXX – порядковий номер талону.

Код медичної карти

метод кодування – послідовний;

метод класифікації - ієрархічний;

довжина коду – 11 символів;

структура коду: XXXXXXXXX-XX – порядковий номер медичної карти.

Код документу

метод кодування – послідовний;

метод класифікації - ієрархічний;

довжина коду – 8 символів;

структура коду: XXXX-XXX – порядковий номер документу.

Код розкладу роботи

метод кодування - порядковий;

довжина коду –2 символи;

структура коду: XX – порядковий номер розкладу роботи.

Код робочої зміни

метод кодування - порядковий;

довжина коду –2 символи;

структура коду: XX – порядковий номер робочої зміни.

Код причини звернення

метод кодування - порядковий;

довжина коду –2 символи;

структура коду: XX – порядковий номер причини звернення.

Класифікатори ІС для системи реєстратури поліклініки подані в таблиці

3.1.

Таблиця 3.1 - Класифікатор ІС для системи реєстратури поліклініки

Пацієнт	Пацієнт 1	0001
	Пацієнт 100	0100
	Пацієнт 9999	9999
Поліклініка	Поліклініка 1 Дніпровського району Києва	KYV-DNP-001

Продовження таблиці 3.1

Користувач	Користувач 1	01
	Користувач 99	99
Роль користувача	Роль 1	01
	Роль 99	99
Лікар	Терапевт 1	001-TRP
Робоча зміна	Зміна 1	01
	Зміна 99	99
Розклад роботи	Розклад 1	01
	Розклад 99	99
Документ	Документ 1	KON-001
	Документ 999	KON-999
Медична карта	Медична карта 1	08022000-01
	Медична карта 99	08022000-99
Талон	Талон 1	0001
	Талон 100	0100
	Талон 9999	9999
Спеціалізація	Кардіолог	KRD
	Терапевт	TRP
	Хірург	HRG
	Невролог	NVR

Продовження таблиці 3.1

Причина звернення	Причина 1	01
	Причина 99	99

Побудована система класифікації та кодування забезпечує ефективність використання ЕР, дозволяючи структурувати та швидко отримувати доступ до інформації реєстратури.

Щоб забезпечити результативне керування та використання даних, необхідно встановити структуру інформаційних масивів. Кожен масив являє собою організоване сховище даних, необхідних для повноцінної роботи системи. До кожного масиву наводиться опис його структури у вигляді таблиці.

Опис атрибутів масиву «Поліклініка» наведено у таблиці 3.2:

Опис масиву “Поліклініка”

Найменування масиву – Поліклініка ;

Ідентифікатор масиву – Policlinica ;

Найменування носію інформації – ЖМД;

Максимальний обсяг масиву - 1000 записів;

Довжина запису - 16384 символів;

Метод організації – послідовний;

Ключ упорядкування - код поліклініки.

Таблиця 3.2 – Масив «Поліклініка»

Найменування	Ідентифікатор	Формат	Бізнес-правила			
			Первинний (втор.) ключ	Умова на значення	Обов'язкове поле	Індексне поле
Код поліклініки	Id_policlinica	Varchar(11)	ПК	NotNull	Так	Так
Повна назва	policlinica_lname	Varchar(100)	—	NotNull	Так	—
Скорочена назва	policlinica_sname	Varchar(45)	—	NotNull	Так	—
Контактний номер	policlinica_phone	Varchar(20)	—	NotNull	Так	—
Адреса	policlinica_address	Varchar(100)	—	NotNull	Так	—
Поштова адреса	policlinica_postadress	Varchar(100)	—	NotNull	Так	—
Імейл	policlinica_email	Varchar(45)	—	NotNull	Так	—
РНОКПП	policlinica_rnokpp	Char(10)	—	NotNull	Так	—
Тип поліклініки	policlinica_type	Varchar(45)	—	NotNull	Так	—
Дата утворення	establishment_date	date	—	NotNull	Так	—

Опис атрибутів масиву «Користувач» наведено у таблиці 3.3:

Опис масиву “Користувач”

Найменування масиву – Користувач ;

Ідентифікатор масиву – User ;

Найменування носію інформації – ЖМД;

Максимальний обсяг масиву - 1000 записів;

Довжина запису - 8192 символів;

Метод організації – послідовний;

Ключ упорядкування - код користувача.

Таблиця 3.3 – Масив «Користувач»

Найменування	Ідентифікатор	Формат	Бізнес-правила			
			Первинний (втор.) ключ	Умова на значення	Обов'язкове поле	Індексне поле
Код користувача	Id_user	Char(2)	PK	NotNull	Так	Так
Ім'я	user_name	Varchar(45)	—	NotNull	Так	—
Імейл	user_email	Varchar(45)	—	NotNull	Так	—
Номер телефону	user_phone	Varchar(20)	—	NotNull	Так	—
Пароль	user_password	Varchar(20)	—	NotNull	Так	Так
Код поліклініки	policlinica_id_policlinica	Varchar(11)	FK	NotNull	Так	Так
Код ролі користувача	role_id_role	Char(2)	FK	NotNull	Так	Так

Опис атрибутів масиву «Роль користувача» представлено у таблиці 3.4:

Опис масиву “Роль користувача”

Найменування масиву – Роль користувача ;

Ідентифікатор масиву – Role ;

Найменування носію інформації – ЖМД;

Максимальний обсяг масиву - 1000 записів;

Довжина запису - 8192 символів;

Метод організації – послідовний;

Ключ упорядкування - код ролі користувача.

Таблиця 3.4 – Масив «Роль користувача»

Найменування	Ідентифікатор	Формат	Бізнес-правила			
			Первинний (втор.) ключ	Умова на значення	Обов'язкове поле	Індексне поле
Код ролі	Id_role	Char(2)	PK	NotNull	Так	Так
Назва ролі	role_name	Varchar(45)	—	NotNull	Так	—

Опис атрибутів масиву «Лікар» представлено у таблиці 3.5:

Опис масиву “Лікар”

Найменування масиву – Лікар ;

Ідентифікатор масиву – Doctor ;

Найменування носію інформації – ЖМД;

Максимальний обсяг масиву - 1000 записів;

Довжина запису - 2730 символів;

Метод організації – послідовний;

Ключ упорядкування - код лікаря.

Таблиця 3.5 – Масив «Лікар»

Найменування	Ідентифікатор	Формат	Бізнес-правила			
			Первинний (втор.) ключ	Умова на значення	Обов'язкове поле	Індексне поле
Код лікаря	Id_doctor	Char(7)	PK	NotNull	Так	Так
Ім'я	doctor_name	Varchar(45)	—	NotNull	Так	—
Кабінет	doctorc_office	int	—	NotNull	Так	—

Продовження таблиці 3.5

Код поліклініки	policlinica_id_policlinica	Varchar(11)	FK	NotNull	Так	Так
Код спеціальності	specialization_id_specialization	Char(3)	FK	NotNull	Так	Так

Опис атрибутів масиву «Пацієнт» наведено у таблиці 3.6:

Опис масиву “Пацієнт”

Найменування масиву – Пацієнт ;

Ідентифікатор масиву – Patient ;

Найменування носію інформації – ЖМД;

Максимальний обсяг масиву - 1000 записів;

Довжина запису - 5461 символів;

Метод організації – послідовний;

Ключ упорядкування - код пацієнта.

Таблиця 3.6 – Масив «Пацієнт»

Найменування	Ідентифікатор	Формат	Бізнес-правила			
			Первинний (втор.) ключ	Умова на значення	Обов'язкове поле	Індексне поле
Код пацієнта	Id_patient	Char(4)	PK	NotNull	Так	Так
Ім'я	patient_name	Varchar(45)	—	NotNull	Так	—
Адреса	patient_address	Varchar(100)	—	—	Hi	—
Номер телефону	patient_phone	Varchar(20)		NotNull	Так	-
Імейл	patient_email	Varchar(45)		—	Hi	-

Продовження таблиці 3.6

ПІН	patient_ipn	Char(10)		NotNull	Так	
Дата народження	patient_birthdate	date		NotNull	Так	
Стать	patient_sex	Varchar(10)		—	НІ	

Опис атрибутів масиву «Спеціалізація» наведено у таблиці 3.7:

Опис масиву “Спеціалізація”

Найменування масиву – Спеціалізація ;

Ідентифікатор масиву – Specialization;

Найменування носію інформації – ЖМД;

Максимальний обсяг масиву - 1000 записів;

Довжина запису - 4096 символів;

Метод організації – послідовний;

Ключ упорядкування - код спеціалізації.

Таблиця 3.7 – Масив «Спеціалізація»

Найменування	Ідентифікатор	Формат	Бізнес-правила			
			Первинний (втор.) ключ	Умова на значення	Обов'язкове поле	Індексне поле
Код спеціалізації	Id_specialization	Char(3)	РК	NotNull	Так	Так
Назва спеціалізації	specialization_name	Varchar(45)	—	NotNull	Так	—

Опис атрибутів масиву «Робоча зміна» наведено у таблиці 3.8:

Опис масиву “Робоча зміна”

Найменування масиву – Робоча зміна ;

Ідентифікатор масиву – Shifts ;

Найменування носію інформації – ЖМД;

Максимальний обсяг масиву - 1000 записів;

Довжина запису - 8192 символів;

Метод організації – послідовний;

Ключ упорядкування - код зміни.

Таблиця 3.8 – Масив «Робоча зміна»

Найменування	Ідентифікатор	Формат	Бізнес-правила			
			Первинний (втор.) ключ	Умова на значення	Обов'язкове поле	Індексне поле
Код зміни	Id_shifts	Char(2)	РК	NotNull	Так	Так
Початок зміни	hours_from	time	—	NotNull	Так	—
Кінець зміни	hours_to	time	—	NotNull	Так	—

Опис атрибутів масиву «Розклад роботи» приведено у таблиці 3.9:

Опис масиву “Розклад роботи”

Найменування масиву – Розклад роботи ;

Ідентифікатор масиву – Schedule;

Найменування носію інформації – ЖМД;

Максимальний обсяг масиву - 1000 записів;

Довжина запису - 1365 символів;

Метод організації – послідовний;

Ключ упорядкування - код розкладу.

Таблиця 3.9 – Масив «Розклад роботи»

Найменування	Ідентифікатор	Формат	Бізнес-правила			
			Первинний (втор.) ключ	Умова на значення	Обов'язкове поле	Індексне поле
Код розкладу	Id_schedule	Char(2)	PK	NotNull	Так	Так
Робочі дні тижня	working_days	Varchar(50)	—	NotNull	Так	—
Код лікаря	working_days	Char(7)	FK	NotNull	Так	Так
Код зміни	shifts_id_shifts	Char(2)	FK	NotNull	Так	Так

Опис атрибутів масиву «Медична карта» приведено у таблиці 3.10:

Опис масиву “Медична карта”

Найменування масиву – Медична карта ;

Ідентифікатор масиву – Medcard ;

Найменування носію інформації – ЖМД;

Максимальний обсяг масиву - 1000 записів;

Довжина запису - 4096 символів;

Метод організації – послідовний;

Ключ упорядкування - код медкарти.

Таблиця 3.10 – Масив «Медична карта»

Найменування	Ідентифікатор	Формат	Бізнес-правила			
			Первинний (втор.) ключ	Умова на значення	Обов'язкове поле	Індексне поле
Код медкарти	Id_patient	Char(13)	PK	NotNull	Так	Так

Продовження таблиці 3.10

Дата створення	creation_date	date	—	NotNull	Так	—
Резус фактор	resus_phactor	Char(1)	—	—	Hi	—
Група крові	blood_group	int	—	—	Hi	—
Код лікаря	doctor_id_doctor	Char(7)	FK	NotNull	Так	Так
Код пацієнта	patient_id_patient	Char(4)	FK	NotNull	Так	Так

Опис атрибутів масиву «Документ пацієнта» приведено у таблиці 3.11:

Опис масиву “Документ пацієнта”

Найменування масиву – Документ пацієнта ;

Ідентифікатор масиву – Document ;

Найменування носію інформації – ЖМД;

Максимальний обсяг масиву - 1000 записів;

Довжина запису - 8192 символів;

Метод організації – послідовний;

Ключ упорядкування - код документа.

Таблиця 3.11 – Масив «Документ пацієнта»

Найменування	Ідентифікатор	Формат	Бізнес-правила			
			Первинний (втор.) ключ	Умова на значення	Обов'язкове поле	Індексне поле
Код документа	Id_document	Char(10)	PK	NotNull	Так	Так
Назва	document_name	Varchar(45)	—	NotNull	Так	—
Код пацієнта	patient_id_patient	Char(4)	FK	NotNull	Так	Так

Опис атрибутів масиву «Талон» показано у таблиці 3.12:

Опис масиву “Талон”

Найменування масиву – Талон ;

Ідентифікатор масиву – Appointment ;

Найменування носію інформації – ЖМД;

Максимальний обсяг масиву - 1000 записів;

Довжина запису - 4096 символів;

Метод організації – послідовний;

Ключ упорядкування - код талону.

Таблиця 3.12 – Масив «Талон»

Найменування	Ідентифікатор	Формат	Бізнес-правила			
			Первинний (втор.) ключ	Умова на значення	Обов'язкове поле	Індексне поле
Код талону	id_appointment	Char(4)	PK	NotNull	Так	Так
Дата запису	appointment_date	date	—	NotNull	Так	—
Час запису	appointment_time	time	—	NotNull	Так	—
Дата створення талону	creation_date	Date	—	NotNull	Так	—
Код лікаря	doctor_id_doctor	Char(7)	FK	NotNull	Так	Так
Код пацієнта	patient_id_patient	Char(4)	FK	NotNull	Так	Так
Код користувача	user_id_user	Char(2)	FK	NotNull	Так	Так

Опис атрибутів масиву «Причина запису» наведено у таблиці 3.13:

Опис масиву “Причина запису”

Найменування масиву – Причина запису ;

Ідентифікатор масиву – Appointment_reason ;

Найменування носію інформації – ЖМД;

Максимальний обсяг масиву - 1000 записів;

Довжина запису - 3276 символів;

Метод організації – послідовний;

Ключ упорядкування - код причини запису.

Таблиця 3.13 – Масив «Причина запису»

Найменування	Ідентифікатор	Формат	Бізнес-правила			
			Первинний (втор.) ключ	Умова на значення	Обов'язкове поле	Індексне поле
Код причини	id_appointment_reason	Char(2)	РК	NotNull	Так	Так
Назва причини	appointment_reason_name	Varchar(100)	—	NotNull	Так	—

Кожен масив має власну унікальну структуру даних, що відбиває необхідну інформацію для ефективної роботи системи. Ці масиви забезпечують зберігання та організацію даних про пацієнтів, лікарів та записи на прийом, що дозволяє системі ефективно керувати інформацією та забезпечувати якісне обслуговування пацієнтів.

Вибір системи керування базами даних (СКБД) є одним із ключових етапів при проектуванні інформаційної системи. Функціональні

характеристики обраної СКБД істотно впливатимуть на продуктивність роботи системи.

Перш ніж розпочати вибір конкретної системи управління БД, необхідно провести аналіз вимог, які висуваються до бази даних ЕР. Отже, основні вимоги включають:

Обсяг і масштабування інформації, що зберігається:

Система повинна забезпечувати ефективне зберігання та обробку медичної інформації, включаючи дані про медичний заклад, пацієнтів, персонал та записи про прийоми.

Прогнозований розмір даних: від 5 до 10 тисяч пацієнтів із можливістю розширення до 50 тисяч пацієнтів строком у 5 років.

Збільшення обсягу даних у системі має здійснюватися без суттєвого впливу на її продуктивність та доступність.

Управління робочим навантаженням та паралелізмом запитів:

Система повинна підтримувати паралельні обробки запитів при одночасному опрацюванні запитів користувачів.

Мережа та безпека:

Система повинна забезпечувати захист даних та дотримання вимог медичної конфіденційності відповідно до відповідних законодавчих норм, таких як HIPAA (Health Insurance Portability and Accountability Act) або GDPR (General Data Protection Regulation).

Необхідна підтримка шифрування даних, автентифікації користувачів та контролю доступу на основі ролей.

Забезпечує захищений доступ до бази даних за допомогою засобів безпечної передачі даних.

Масштабованість та доступність:

Система повинна забезпечувати горизонтальну масштабованість для можливості розширення ресурсів зі збільшенням навантаження чи обсягу даних.

Забезпечення високої доступності бази даних із мінімальним часом простою (downtime) задля забезпечення безперервності роботи реєстратури.

Забезпечення наявності механізмів резервування даних та автоматичного відновлення після збоїв чи аварійних ситуацій.

Фінансові аспекти та обслуговування:

Система має бути розроблена та впроваджена в рамках визначеного бюджету, враховуючи витрати на ліцензії, обладнання та технічне обслуговування.

Забезпечення оптимізації ресурсів з метою мінімізації витрат на обладнання та максимізації ефективності функціонування системи.

Підтримка та обслуговування СКБД мають бути доступними та гарантувати оперативне реагування на запити користувачів та вирішення будь-яких проблемних ситуацій.

Після визначення вимог до бази даних необхідно проаналізувати різні системи управління базами даних та оцінити їхню відповідність встановленим критеріям (таб. 3.14)

Таблиця 3.14 – Порівняльний аналіз популярних СКБД

Критерій	MySQL	Microsoft SQL Server	Oracle	PostgreSQL	MongoDB
Робоче навантаження	-	+	+	-	+
Кросплатформність	+	-	+	+	-
Безпека	+	+	+	+	+
Масштабування інформації	+	+	-	-	+
Фінансові аспекти та обслуговування	+	+	-	-	+
Швидкість навчання	+	-	-	-	+

Продовження таблиці 3.14

Тип бази даних	SQL	T-SQL	SQL, Multi-model	SQL, Object-relational	NoSQL, document-oriented
Усього	5	4	3	2	5

За результатами таблиці можна дійти висновку, що MySQL і MongoDB отримали однакову кількість балів і найкраще відповідають нашим цілям. Однак, на основі мого власного досвіду роботи з системами управління базами даних, широкою поширеністю, надійністю та добре розробленою документацією системи керування, вибір був зроблений на користь *сервера MySQL*.

Вирішальну роль у формуванні ефективного інформаційного забезпечення становить моделювання БД.

Для розробки ефективної бази даних використовуються різні методики та інструменти, серед яких важливе місце займає розробка інфологічної моделі бази даних із застосуванням різних CASE-засобів.

Інфологічна модель БД надає концептуальний опис елементів бази даних, їхніх атрибутів і зв'язків один з одним.

Якісно розроблена інфологічна модель дозволяє послідовно структурувати елементи архітектури БД, що зменшує ймовірність виникнення помилок і невідповідностей у наступних етапах розробки та забезпечує більш ефективне управління даними в інформаційній системі реєстратури поліклініки.

AllFusion Erwin Data Modeler є CASE-засобом, спеціалізованим програмним забезпеченням, призначеним для проектування баз даних. Цей інструмент володіє широким набором функціональних можливостей, що дозволяють аналізувати, моделювати та візуалізувати структуру даних. За

допомогою AllFusion Erwin Data Modeler розробники можуть створювати інфологічні моделі, визначати сутності та їх атрибути, встановлювати зв'язки між таблицями, а також генерувати скрипти для створення фізичної бази даних на різних платформах.

Для автоматизованої інформаційної системи реєстратури поліклініки виділено такі сутності:

- Appointment;
- Appointment_reason;
- Doctor;
- Document;
- Medcard;
- Patient;
- Polyclinica;
- Role;
- Schedule;
- Shifts;
- Specialization;
- User.

На підставі наведених вище сутностей побудовано інфологічну модель БД за допомогою середовища Erwin Data Modeler, яка представлена на рисунку 3.2.

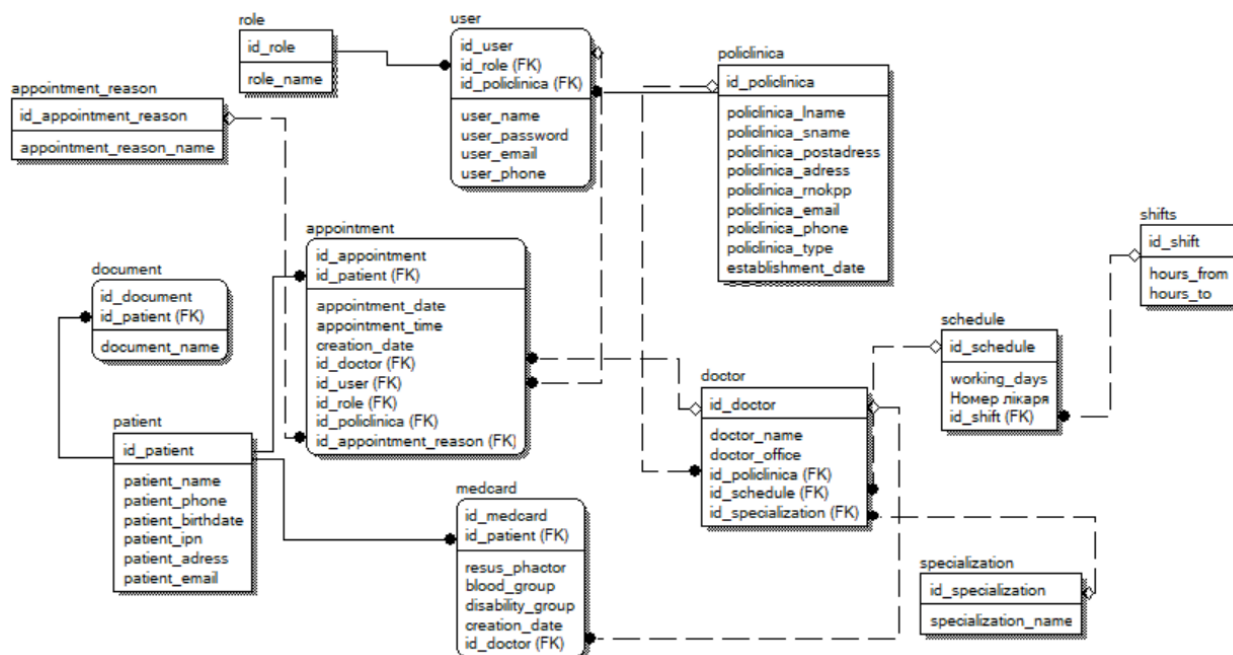


Рисунок 3.2 - Інфологічна модель системи

Зв'язок працює шляхом зіставлення даних у ключових стовпцях; зазвичай це стовпці з тим самим ім'ям в обох таблицях [40].

Відповідно до попередньої схеми бази даних, встановлений зв'язок типу "Один до багатьох" між наступними таблицями:

«policlinic» та «user» - поліклініка має декілька користувачів, і кожний з цих користувачів зареєстрований в одній поліклініці;

«user» та «role» - одна роль відводиться декільком працівникам, проте кожен працівник виконує одну роль;

«doctor» та «schedule» - кожен співробітник має динамічний графік роботи, але кожний конкретний час і місце роботи ставитиметься лише до одного співробітника;

«schedule» та «shifts» - один розклад лікаря може містити кілька інтервалів роботи, але кожен інтервал роботи належить лише одному розкладу лікаря;

«appointment» та «doctor» - кожен лікар може мати більше одного запису на прийом, але певний запис на прийом належить тільки одному лікарю;

«specialization» та «doctor» - кожен доктор має одну спеціалізацію, але одна спеціалізація може бути притаманна декільком лікарям;

«patient» та «appointment» - кожен пацієнт може мати більше одного запису, але кожен запис належить лише одному пацієнту;

«user» та «appointment» - кожен співробітник може створити більше одного запису на прийом, але запис на прийом належить тільки одному співробітнику;

«doctor» та «policlinic» - поліклініка містить багато лікарів, ці лікарі працюють в одній поліклініці;

«doctor» та «medcard» - одну медичну карту веде лише один сімейний лікар, але лікар може вести карти багатьох своїх пацієнтів;

«appointment» та «appointment_reason» - кожен запис має одну причину звернення.

Зв'язок «Один до одного» представлений таблицями:

«patient» та «medcard» - кожен пацієнт має лише одну медкарту, і кожна мед.карта може відноситися тільки до одного пацієнта;

«patient» та «document» - кожен пацієнт має копію одного документу, і кожна копія належить одному пацієнту.

Крім того, у додатку Б представленні екранні форми, які ілюструють етапи розробки інфологічної моделі бази даних для полегшення розуміння методології проектування бази даних.

Попередньо проаналізувавши предметну область, вдалося виявити та виділити такі сутності та їх атрибути:

Сутність «policlinic» містить усі необхідні дані про медичний заклад.

Сутність «appointment» зберігає інформацію про відвідування пацієнтом фахівця.

Сутність «doctor» має у собі відомості про медичних фахівців.

Сутність «specialization» визначає галузі роботи медичних фахівців.

Сутність «patient» зберігає атрибути для ідентифікації пацієнтів.

Сутність «user» надає дані про користувачів системи.

Сутність «role» визначає права доступу користувачів.

Сутність «schedule» містить інформацію про графік прийому лікарів.

Сутність «shifts» вказує на робочі зміни лікарів.

Сутність «medcard» містить попередню медичну інформацію пацієнта.

Сутність «document» зберігає скановані копії документів пацієнтів.

Сутність «appointment_reason» зберігає причини, з якими пацієнти звертаються на прийом до лікаря.

Даталогічна модель даних розширює логічну за допомогою опису конкретних деталей, а саме: імен таблиць і стовпців, індексів, типів даних та механізмів зберігання. Це допоможе наочно представити, як дані будуть зберігатися та витягуватися у системі. Даталогічна модель бази даних реєстратури поліклініки наведена нижче на рисунку 3.3.

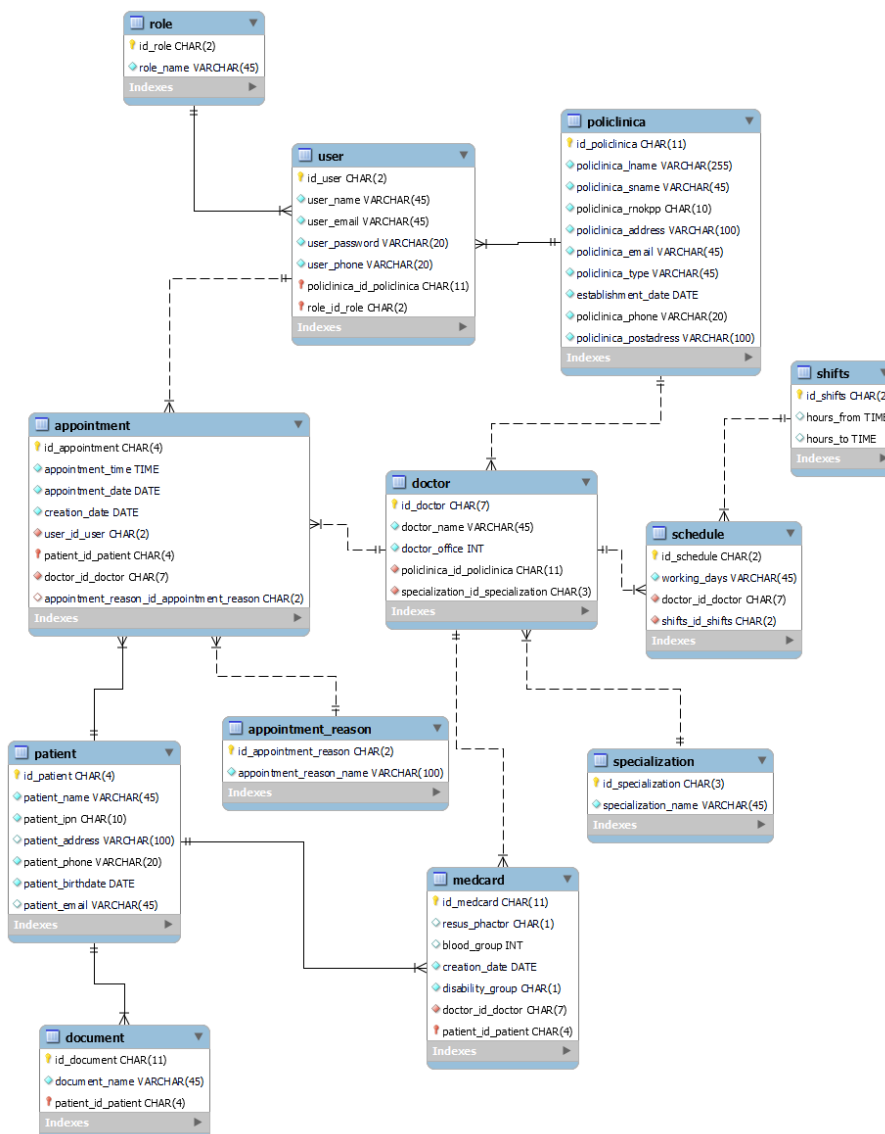


Рисунок 3.3 - Даталогічна модель бази даних

Ці сутності та їх атрибути становлять основу створення таблиць у базі даних, які будуть використовуватися для зберігання інформації. Зміна даних в одній таблиці відобразиться на іншій, що гарантує цілісність даних. Для створення таблиць було визначено всього 12 сутностей, що у нашому випадку є достатнім для забезпечення функціональності системи, що розробляється.

Таблиця "patient" (рисунок 3.4) використовується для зберігання даних про пацієнтів.

Містить такі поля:

id_patient – номер пацієнта;
 patient_name – ПІБ;
 patient_ipn – ІПН;
 patient_address – адреса проживання;
 patient_phone – мобільний телефон;
 patient_birthdate – дата народження;
 patient_email – емейл.

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
id_patient	char(4)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
patient_address	varchar(100)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references
patient_birthdate	date		NO			select,insert,update,references
patient_email	varchar(45)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references
patient_ipn	char(10)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
patient_name	varchar(45)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
patient_phone	varchar(20)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references

Рисунок 3.4 - Таблиця "patient"

Таблиця «medcard» (рисунок 3.5) – це таблиця, у якій зазначені попередні медичні характеристики пацієнта.

Містить такі поля:

id_medcard – номер медичної картки;
 resus_phactor – резус-фактор;
 blood_group – група крові;
 creation_date – дата створення картки;
 disability_group – група інвалідності;
 patient_id_patient – зв'язок із таблицею «patient»;
 doctor_id_doctor – зв'язок із таблицею «doctor».

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
◇ blood_group	int		YES			select,insert,update,references
◇ creation_date	date		NO			select,insert,update,references
◇ disability_group	char(1)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
◇ doctor_id_dodtor	char(7)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
◇ id_medcard	char(11)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
◇ patient_id_patient	char(4)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
◇ resus_phactor	char(1)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references

Рисунок 3.5 - Таблица "medcard"

Таблиця «appointment» (рисунок 3.6) – це таблиця, у якій зберігаються записи на прийом до лікаря.

Містить такі поля:

id_appointment – номер запису на прийом;

appointment_time – час прийому;

appointment_date – дата прийому;

creation_date – дата створення запису;

patient_id_patient - зв'язок із таблицею «patient»;

doctor_id_doctor - зв'язок із таблицею «doctor»;

user_id_user - зв'язок із таблицею «user»;

appointment_reason_id_appointment_reason – зв'язок з таблицею «appointment_reason».

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
◇ appointment_date	date		NO			select,insert,update,references
◇ appointment_reason...	char(2)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references
◇ appointment_time	time		NO			select,insert,update,references
◇ creation_date	date		NO			select,insert,update,references
◇ doctor_id_dodtor	char(7)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
◇ id_appointment	char(4)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
◇ patient_id_patient	char(4)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
◇ user_id_user	char(2)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references

Рисунок 3.6 - Таблица "appointment"

Таблиця «user» (рисунок 3.7) - це таблиця, в якій зберігається інформація про користувачів системи.

Містить такі поля:

id_user – номер користувача;

user_name – ім'я користувача;

user_email – імейл користувача;

user_password – пароль користувача;

user_phone – номер телефону користувача;

policlínica_id_policlinica - зв'язок із таблицею «policlínica»;

role_id_role - зв'язок із таблицею «role».

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
id_user	char(2)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
policlínica_id_polidini...	char(11)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
role_id_role	char(2)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
user_email	varchar(45)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
user_name	varchar(45)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
user_password	varchar(20)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
user_phone	varchar(20)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references

Рисунок 3.7 - Таблиця "user"

Таблиця "specialization" (рисунок 3.8) - таблиця, в якій є дані про спеціалізацію лікарів поліклініки.

id_specialization – код спеціалізації;

specialization_name – назва спеціалізації.

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
id_specialization	char(3)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
specialization_name	varchar(45)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references

Рисунок 3.8 - Таблиця "specialization"

Таблиця "role" (рисунок 3.9) - таблиця, в якій зберігаються дані про права доступу співробітників поліклініки.

Містить такі поля:

id_role – код ролі;

role_name – назва ролі.

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
id_role	char(2)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
role_name	varchar(45)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references

Рисунок 3.9 - Таблиця "role"

Таблиця "schedule" (рисунок 3.10) - це таблиця, в якій зберігається інформація про графік роботи співробітників поліклініки.

Містить такі поля:

id_schedule – номер розкладу;

working_days – робочі дні тижня;

doctor_id_doctor – номер лікаря;

shifts_id_shifts – робочі зміни.

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
doctor_id_doctor	char(7)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
id_schedule	char(2)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
shifts_id_shifts	char(2)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
working_days	varchar(45)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references

Рисунок 3.10 - Таблиця "schedule"

Таблиця "shifts" (рисунок 3.11) відображає робочі зміни у графіку роботи лікарів.

id_shifts – номер зміни;

hours_from – початок зміни;

hours_to – кінець зміни.

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
id_shifts	char(2)		NO	utf8mb3	utf8mb3_genera...	select,insert,update,references
hours_from	time		NO			select,insert,update,references
hours_to	time		NO			select,insert,update,references

Рисунок 3.11 - Таблиця "shifts"

Таблиця «policlinica» (рисунок 3.12) відображає основну інформацію медичного закладу.

id_policlinica – код поліклініки;

policlinica_lname – повна назва;

policlinica_sname – скорочена назва;

policlinica_rnokpp – РНОКПП;

policlinica_address – юридична адреса;

policlinica_email – емейл;

policlínica_type – тип медичного закладу;

establishment_date – дата заснування;

policlínica_phone – номер телефону;

policlínica_postadress – поштова адреса.

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
◇ establishment_date	date		NO			select,insert,update,references
◇ id_policlinica	char(11)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
◇ policlinica_address	varchar(100)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
◇ policlinica_email	varchar(45)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
◇ policlinica_lname	varchar(255)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
◇ policlinica_phone	varchar(20)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
◇ policlinica_postadress	varchar(100)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
◇ policlinica_rnokpp	char(10)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
◇ policlinica_sname	varchar(45)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
◇ policlinica_type	varchar(45)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references

Рисунок 3.12 - Таблиця "policlínica"

Таблиця "document" (рисунок 3.13) - це таблиця, в якій зберігаються дані сканованих копій документів пацієнтів.

Містить такі поля:

id_document – код документа;

document_name – назва документа;

patient_id_patient - зв'язок із таблицею «patient».

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
◇ id_document	char(10)		NO	utf8mb3	utf8mb3_genera...	select,insert,update,references
◇ document_name	varchar(45)		NO	utf8mb3	utf8mb3_genera...	select,insert,update,references
◇ patient_id_patient	char(4)		NO	utf8mb3	utf8mb3_genera...	select,insert,update,references

Рисунок 3.13 - Таблиця "document"

Таблиця "doctor" (рисунок 3.14) - це таблиця, в якій зберігається інформація про лікарів поліклініки.

Містить такі поля:

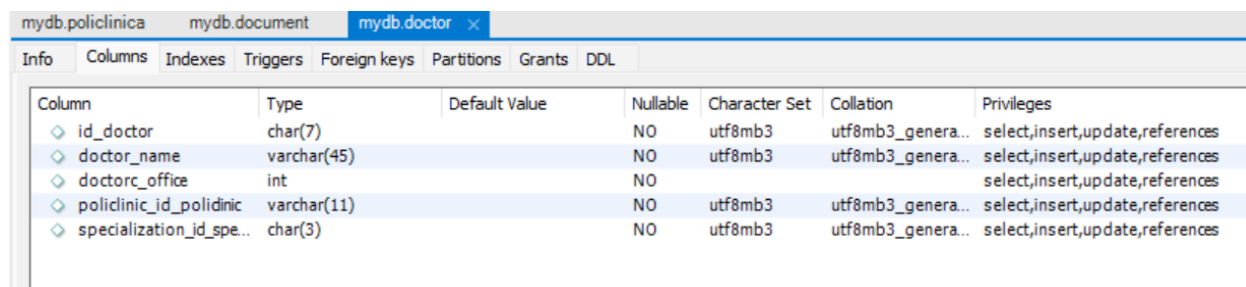
id_doctor – номер лікаря;

doctor_name – ім'я лікаря;

doctor_office – номер кабінету;

policlinic_id_policlinic - зв'язок із таблицею «policlinica»;

specialization_id_specialization - зв'язок із таблицею «specialization».



Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
id_doctor	char(7)		NO	utf8mb3	utf8mb3_genera...	select,insert,update,references
doctor_name	varchar(45)		NO	utf8mb3	utf8mb3_genera...	select,insert,update,references
doctorc_office	int		NO			select,insert,update,references
policlinic_id_polidinic	varchar(11)		NO	utf8mb3	utf8mb3_genera...	select,insert,update,references
specialization_id_spe...	char(3)		NO	utf8mb3	utf8mb3_genera...	select,insert,update,references

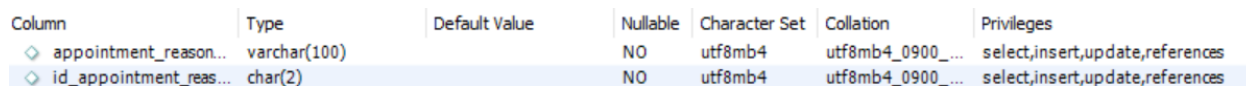
Рисунок 3.14 - Таблиця "doctor"

Таблиця "appointment_reason" (рисунок 3.15) - це таблиця, в якій зберігаються можливі причини запиту на прийом до лікаря.

Містить такі поля:

id_appointment_reason – код причини;

appointment_reason_name – назва причини;



Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
appointment_reason...	varchar(100)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
id_appointment_reas...	char(2)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references

Рисунок 3.15 - Таблиця "appointment_reason"

MySQL [41] - це потужна та широко використовувана система управління базами даних (СУБД), яка підтримує різні операційні системи, включаючи різні дистрибутиви UNIX (такі як Linux та FreeBSD), а також платформу Microsoft Windows. Вона відрізняється високою продуктивністю, надійністю та розширюваністю, що робить її популярним вибором для розробників та адміністраторів баз даних.

Код SQL, розроблений для роботи з базою даних на сервері MySQL, представлений у додатку В.

3.2. Технічне забезпечення

Технічні засоби впливають на ефективність процесу створення інформаційних систем. Їх потужність визначає, які завдання можна вирішувати у певній галузі.

Сучасне технічне забезпечення інформаційних систем включає різноманітну техніку: комп'ютери, принтери, сканери, офісне обладнання, засоби зв'язку, мережеве обладнання, мультимедіа і так далі. Даний комплекс має бути розроблений з акцентом на надійність, масштабованість та взаємодію, тому кожен модуль системи має бути оснащений спеціальними автоматизованими модулями обробки для забезпечення продуктивності та цілісності даних. Для покращення розуміння системи, планування фізичної архітектури, відповідного апаратного та мережевого обладнання та оптимізації конфігурації системи на рисунку 3.16 представлена діаграма розгортання.

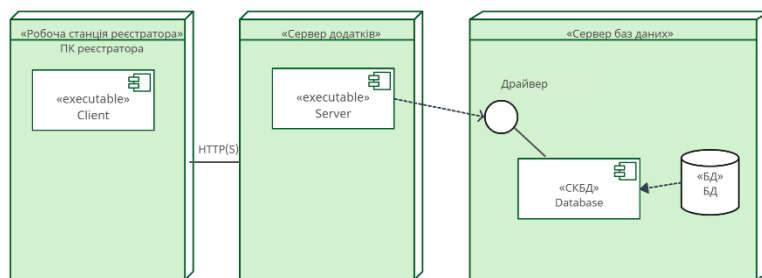


Рисунок 3.16 - Діаграма розгортання

Усі сервери обладнані інформаційною системою Windows для забезпечення стабільності, безпечності та ефективного керування ресурсами. Для обробки HTTP-запитів і надання інтерфейсу користувача використовується веб-сервер, сервер додатку Apache Tomcat забезпечує виконання Java та PHP коду та управління бізнес-логікою системи, сервер БД працює на платформі з СКБД MySQL.

Серверна інфраструктура має бути встановлена та розміщена згідно з усіма правилами безпеки у спеціально обладнаному для цього приміщенні. Для повноцінної та цілісної роботи інформаційної системи в реєстратурі медичного закладу необхідно поєднання спеціалізованих серверів для ефективного виконання різних завдань. Сервер друку та сервер сканера керують інтеграцією цих пристроїв з інформаційною системою реєстратури, полегшуючи централізоване керування завданнями скану/друку. Поштовий сервер необхідний для керування поштовими скриньками користувачів, маршрутизацією електронної пошти, фільтрацією спаму. Він підтримує протоколи доступу до поштових повідомлень та відправки електронної пошти SMTP, IMAP і POP3. Сервер додатків забезпечує роботи між різними модулями системи, оскільки там розміщені основні додатки, необхідні для роботи ІС. На сервері бази даних розміщується уся центральна БД.

Забезпечення мережної взаємодії між серверами та клієнтською частиною забезпечується за допомогою маршрутизатора та комутаторів. Для підключення до Інтернету бажано використовувати високошвидкісне широкосмугове з'єднання, наприклад, Fiber-to-the-Home, FTTHxxx-SM-02, так як він забезпечує швидке та стабільне з'єднання. Щодо модему, рекомендовано використовувати пристрої з широкосмуговою технологією, наприклад GPON-модем. Для підключення до локальної мережі обрані кабелі UTP-8. Також бажано використовувати ретранслятори, сумісні з кабелями UTP-8. Мережеве обладнання має бути розміщене в захищених корпусах, а канали зв'язку мають бути зашифровані за допомогою стандартних протоколів. До того ж, мережеве обладнання має витримувати електромагнітні перешкоди. Інформаційна система працює з персональними даними осіб, тому необхідно забезпечити високий рівень безпеки. Для цього використовуються засоби захисту мережі, такі як брандмауери та системи виявлення вторгнень.

Усі передані та збережені медичні дані шифруються з використанням стандартних криптографічних алгоритмів для запобігання несанкціонованому доступу.

Забезпечення надійного зберігання даних медичної системи забезпечується високопродуктивною системою зберігання, при цьому проводиться регулярне копіювання даних із можливістю їх подальшого відновлення у разі виникнення аварійних та непередбачуваних ситуацій.

Додатково для захисту даних необхідно реалізувати систему однозначної ідентифікації користувачів для забезпечення доступу до відповідних даних залежно від ролі.

Вибір комп'ютерної та серверної інфраструктури у великій мірі базується на сформульованих нефункціональних вимогах. Таким чином, *системні вимоги встановлюються наступним чином:*

Процесор: Intel Core i5 або еквівалент, що забезпечує плавну багатозадачність і швидкість реагування.

Оперативна пам'ять: мінімум 8 ГБ оперативної пам'яті DDR4 для ефективної обробки даних і багатозадачності без сповільнень.

Жорсткий диск: твердотільний накопичувач (SSD) з мінімальною ємністю 256 ГБ для швидкого завантаження та доступу до даних, що підвищує продуктивність.

Операційна система: Windows 10 Professional або вище, що забезпечує стабільність, безпеку та сумісність із програмним забезпеченням.

Для забезпечення стабільної та безперешкодної взаємодії користувача з ІС використовуються наступні *периферійні технічні засоби:*

Монітори: світлодіодні монітори Full HD із мінімальним розміром 21 дюйм, що забезпечує чітке та детальне відображення записів пацієнтів та інформації про планування.

Ергономічні клавіатури та миші для зручного та точного введення даних.

Сканери документів та лазерні принтери для оцифрування паперових записів і двостороннього друку різноманітних документів і звітів.

Шумопоглинаючі гарнітури з мікрофонами для чіткого та безперешкодного спілкування під час телефонних дзвінків.

ІС, що розробляється, підключається до мережі Ethernet. Також реалізовується застосування сучасних стандартів та протоколів для захисту персональних медичних даних, таких як TLS/SSL для шифрування з'єднань та віртуальні приватні мережі (VPN) для безпечного доступу до системи ззовні.

Припустимо, що підсистема поліклініки – реєстратура, - середнього розміру потребує наступного *мережевого обладнання*:

Комутатори Ethernet: 2 блоки для підключення системи і периферійних пристроїв.

Маршрутизатор: 1 блок для керування мережевим трафіком і надання доступу до Інтернету.

Гіпервізор: 1 блок для керування ресурсами серверів, забезпечення ефективного розгортання, масштабування та керування віртуальними машинами.

VPN-шлюз: 1 блок для забезпечення безпечної та захищеної передачі даних між віддаленими пристроями та мережами через загальнодоступні або ненадійні мережі, забезпечуючи шифрування та автентифікацію для захисту конфіденційності та цілісності даних.

Мережеві кабелі: достатня кількість для підключення периферійних пристроїв і мережевого обладнання.

Стабільна робота ІС та ефективна взаємодія користувачів із системою забезпечується завдяки впровадженню системи технічної підтримки, забезпечує механізм швидкого реагування та вирішення технічних проблем,

що виникають, та використанню технічної документації та посібників користувача.

Нижче на рисунку 3.17 приведена загальна схема мережі передачі даних.

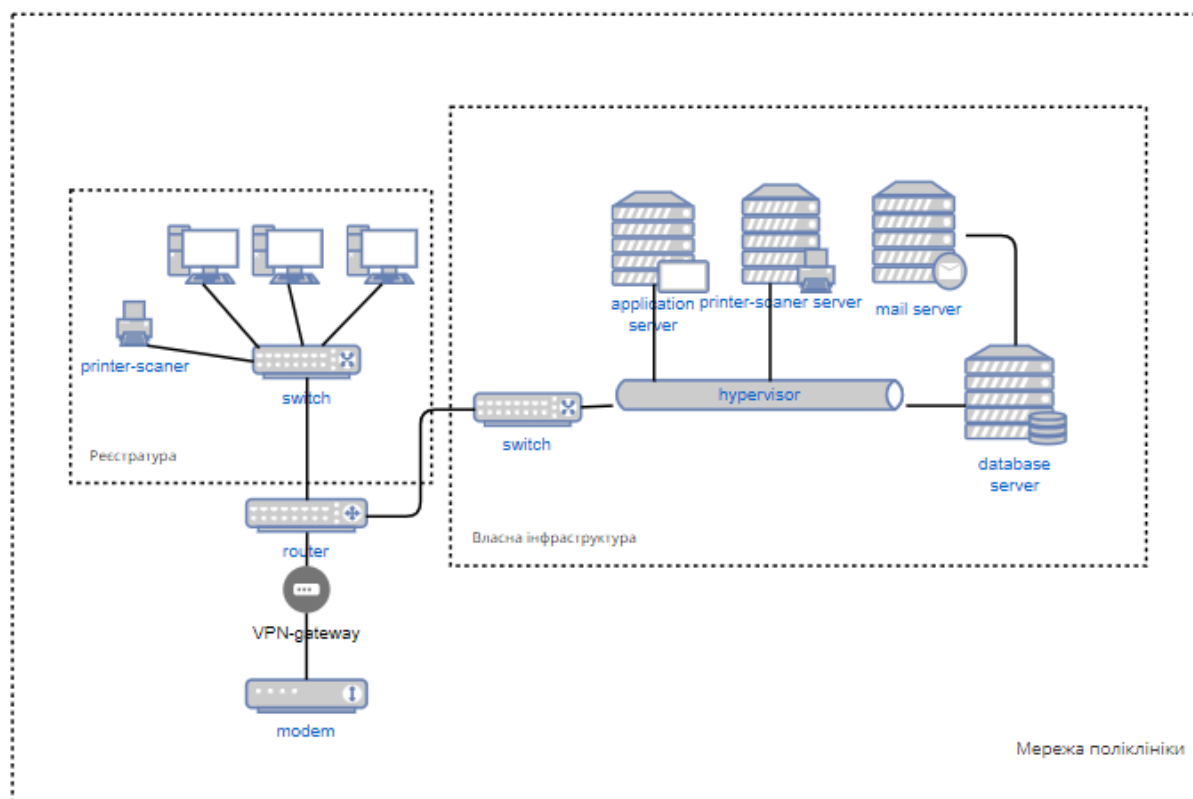


Рисунок 3.17 - Загальна схема мережі передачі даних

3.3. Програмне забезпечення

3.3.1. Побудова структури програмного забезпечення

Структурна схема відображає, з яких частин складається програмне забезпечення, оскільки саме ці частини взаємодіють одна з одною.

Проектована система поділяється на дві підсистеми:

Підсистема автентифікації: включає функціонал перевірки прав доступу користувачів до ЕР та містить такі елементи, як введення логіна та пароля, аутентифікація користувача; обробка виняткових ситуацій за помилки аутентифікації.

Підсистема роботи з довідниками та документами: відповідає за формування та обробку інформації, що зберігається у довідниках та документах.

Підсистема роботи з довідниками та документами у свою чергу включає ще дві підсистеми:

Підсистема пошуку: дозволяє здійснювати пошук інформації у довідниках та документах за заданими критеріями.

Підсистема аналітики та звітування: формує звіти на основі даних із довідників та документів.

Підсистема обробки виняткових ситуацій: перевіряє коректність дій користувача та виводить повідомлення про помилки у разі їх виникнення.

Надалі дані підсистеми розкладаються на інші:

Відображення довідників та документів: забезпечує виведення інформації із довідників та документів на екран.

Введення даних: здійснює введення інформації, яка потрібна для роботи підсистем, та містить такі елементи, як введення даних для автентифікації, введення даних для пошуку, введення даних для роботи з довідниками та документами.

Збереження даних: відповідає за збереження введеної інформації до бази даних.

Відображення результатів пошуку: виводить на екран результати пошуку у довідниках та документах.

Налаштування звіту: дозволяє налаштовувати відображення інформації у відповідно до вимог користувача.

Збереження звіту: забезпечує можливість збереження звітів у вибраному каталозі.

Ці елементи утворюють основу структури програмного забезпечення системи реєстратури поліклініки, забезпечуючи функціонал для управління інформацією, обробки запитів користувачів та забезпечення безпеки доступу до системи.

Загальна структурна схема проектного ПЗ представлена на рисунку 3.18



Рисунок 3.18 - Структурна схема проектного ПЗ

3.3.2. Програмне забезпечення

Розробка МІС реєстратури потребує ретельного вибору прикладних програмних засобів для забезпечення ефективного та безперебійного

функціонування системи. З цією метою необхідно розглянути сукупність програмних засобів різних типів, що відповідають вимогам функціональності, інтеграції та швидкості розробки, а також інструменти для роботи з глобальними мережами та організації обчислювального процесу.

На сьогоднішній день ринок програмного забезпечення пропонує широкий асортимент інструментів, здатних оптимізувати та прискорити процес розробки інформаційних систем. Користувачами таких продуктів й інструментів є прикладні та системні програмісти.

Операційна система Windows 10 була обрана як базова платформа для розробки інформаційної системи. Вона забезпечує зручне середовище розробки, широкий вибір інструментів та підтримку різних мов програмування.

Для управління базою даних вибрано систему управління базами даних MySQL. MySQL надає надійне та ефективне сховище даних для ІС, а також великий набір функцій для роботи з даними.

Як мову програмування для розробки програми доцільно обрати Java. Java SE 8 надає потужні засоби розробки, включаючи багатий набір бібліотек та API, необхідних для створення високопродуктивних та масштабованих додатків.

Для забезпечення ефективного управління різними ролями користувачів та їх функціональністю вибрано патерн "Модель-представлення-контролер" (MVC). Даний патерн має на меті розділити інтерфейс користувача і керуючу логіку на три частини таким чином, що модифікація кожної з частин здійснюється окремо (рис. 3.19).

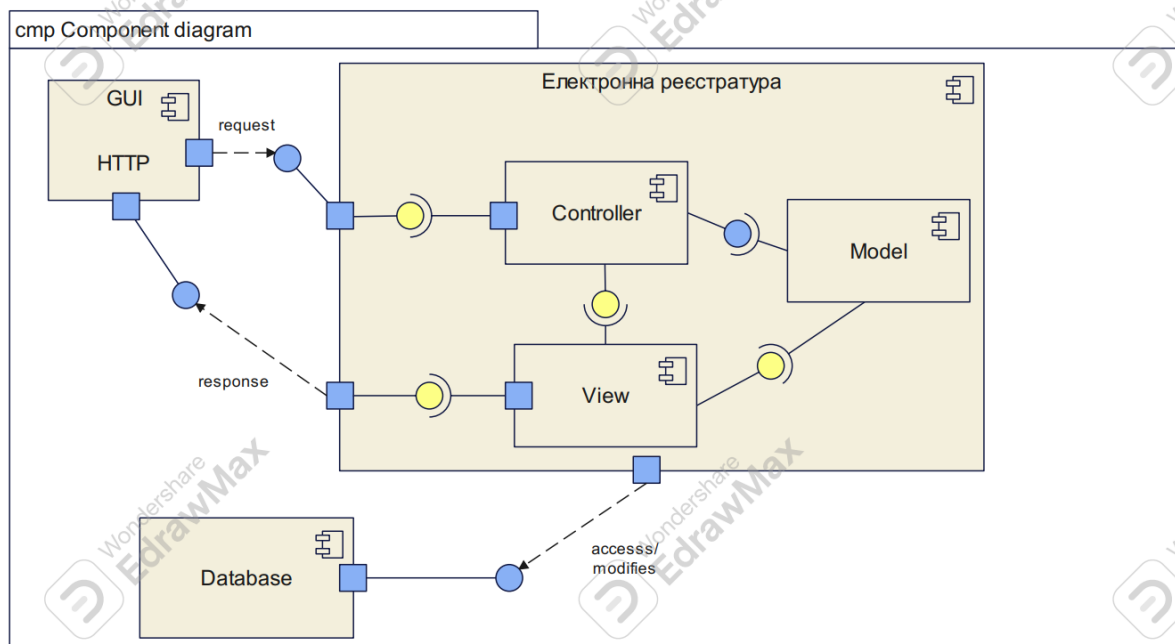


Рисунок 3.19 - Компонентна діаграма

Yii фреймворк надає потужний інструментарій для роботи з MVC. Моделі Yii можуть бути створені автоматично або вручну, визначенням класів на основі таблиць бази даних. Подання Yii обробляють дані, отримані з контролера, і генерують відповідний HTML. Контролери в Yii відповідають за обробку запитів, вибір та виконання відповідних дій, а також передачу даних між моделлю та поданням.

Взаємодія між компонентами у цьому фреймворку відбувається так:

Користувач системи здійснює взаємодію з нею через інтерфейс, тим самим надсилає запит серверу.

Фреймворк отримує запит (перегляд інформації про пацієнта або додавання пацієнта) і визначає, який контролер повинен обробити запит.

Контролер отримує запит, перетворює його на зрозумілий для моделей формат та команди, і викликає відповідний метод, який взаємодіє з моделлю.

Модель обробляє запит, виконує необхідні операції з даними (наприклад, Doctor.php може містити методи для отримання списку лікарів, їх редагування та інші операції) та повертає результат контролеру.

Контролер отримує результат від моделі і передає їх у представлення.

Представлення відображає результат взаємодії користувача із системою у необхідному представленні через відповідний інтерфейс.

Для реалізації MVC у Yii2 достатньо створити відповідні класи моделі, представлення та контролера, а потім зв'язати їх у відповідних місцях.

При створенні ІС також необхідно передбачити, які саме технології будуть використовуватися для побудови системи. На сьогоднішній день ринок програмного забезпечення пропонує безліч корисних засобів та інструментів, здатних прискорити та покращити процес розробки у багато разів. Дані продукти підтримують всі технологічні етапи проектування, будь то кодування, налагодження чи тестування. Користувачами таких продуктів є прикладні та системні програмісти. До основних критеріїв вибору технологій розробки відноситься також відносно низька цінова категорія.

В якості операційної системи, на якій буде розроблятися ІС реєстратури медичного закладу, була обрана Windows 10. Для розробки веб-інтерфейсу доцільно використати JavaScript, HTML та CSS, а для створення модулів ІС – мову програмування Python.

Для взаємодії з базою даних рекомендовано використати інструмент JDBC. JDBC забезпечує доступ до бази даних із програми Java, дозволяючи виконувати запити та оновлення даних.

Для керування версіями коду та спільної розробки переважно використовувати систему контролю версій Git. Для відстеження завдань та управління проектом рекомендується скористатися Jira. Для автоматизації

розгортання та масштабування програми пропонується інтегрувати Docker та Kubernetes.

Для обробки HTTP-запитів і надання інтерфейсу користувача доцільно використати веб-сервер, сервер додатку Apache Tomcat забезпечує виконання Java-коду та управління бізнес-логікою системи. Apache HTTP-сервер є вільним веб-сервером, інакше кажучи, це вільне програмне забезпечення для розміщення веб-сервера. За статистикою на травень 2019 року використовується в 46% всіх веб-серверів [42], що становить приблизно 1/3 сайтів у глобальній мережі Інтернет і це робить його найпопулярнішим веб-сервером. Сервер має велику кількість модулів для роботи з багатьма серверними технологіями. Сервер БД працює на платформі з СКБД MySQL.

Для створення графічного інтерфейсу користувача (GUI) можна використати платформу JavaFX. JavaFX надає багаті можливості для створення сучасних та інтуїтивно зрозумілих інтерфейсів користувача за допомогою xml-подібної мови розмітки FXML.

3.4. Результати реалізації інформаційної системи

У ході реалізації системи, спираючись на попередні етапи моделювання та проектування, проведемо створення прототипу за допомогою мов розмітки даних та описань зовнішнього вигляду сторінки – HTML і CSS.

Це допоможе налаштувати візуальну логіку та користувацький інтерфейс програми, що є важливим етапом перед реальним впровадженням системи. Так ми реалізуємо те, яким чином користувач буде взаємодіяти із системою, та як реагуватимуть на дії користувача її окремі компоненти.

На рисунку 3.20 представлена структура електронної реєстратури.

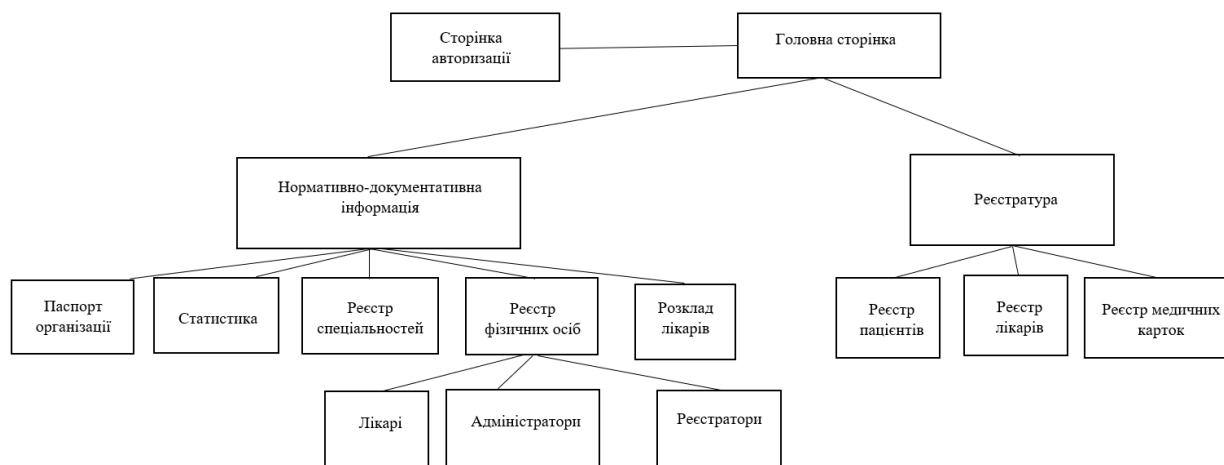


Рисунок 3.20 - Структура електронної реєстратури

Головна сторінка містить у собі усі розділи, бокове меню-навігацію та профіль користувача.

Сторінка авторизації дозволяє робітнику увійти у систему або зареєструватися в ній.

Сторінка реєстратури є основною сторінкою, де зберігається інформація, що постійно оновлюється (дані пацієнтів, записи на прийом, медичні картки), а також саме на цій сторінці здійснюється регулювання записами пацієнтів на прийоми до лікарів.

Сторінка нормативно-довідникової інформації забезпечує відображення стабільних даних: даних про адміністраторів, реєстраторів, лікарів, а також їх графік роботи.

Прототип був розроблений згідно з вище встановлених вимог та створеної раніше структури прототипу у онлайн-сервісі Figma. Структури прототипу наведені у додатку В.

При запуску користувачу необхідно провести авторизацію у системі, форма авторизації зображена на рисунку 3.21.

Поліклініка
КНП КДЦ №2 Дніпровського району

Логін:

Пароль:

+

OK Вийти

Рисунок 3.21 - Форма авторизації

У випадку, якщо користувач не зареєстрований, він натискає на знак додавання та відкриває форму реєстрації нового користувача (рис. 3.22).

Поліклініка
КНП КДЦ №2 Дніпровського району
Новий користувач

Логін:

Пароль:

Номер працівника:

Телефон:

Email:

Записати та закрити Закрити

Рисунок 3.22 - Форма додавання нового користувача

Після авторизації у системі користувач отримує доступ до головної сторінки. Там він матиме змогу перейти на будь-які сторінки системи або

вийти зі свого профілю, або зайти у свій профіль користувача (обліковий запис). Головна сторінка відображена на рисунку 3.23.

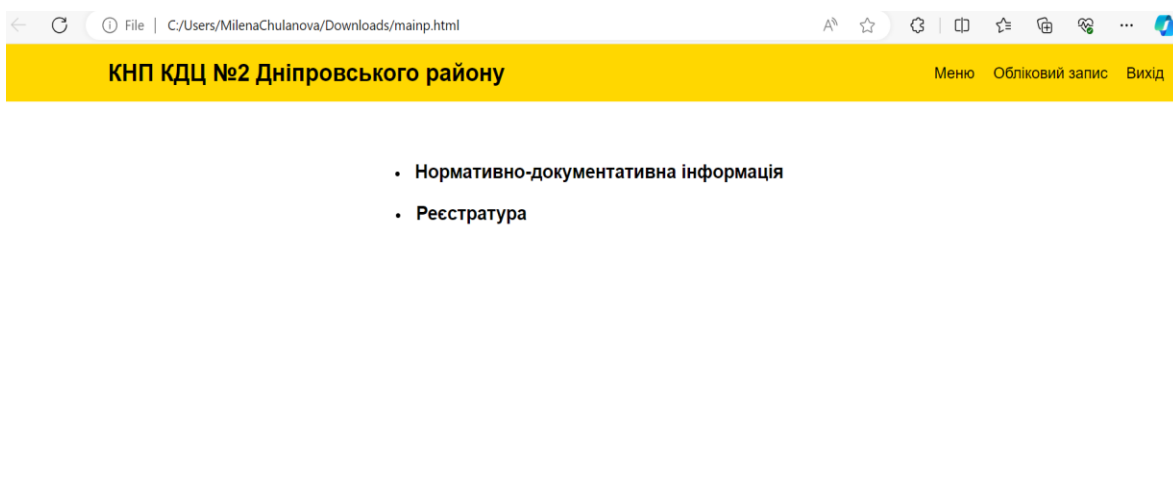


Рисунок 3.23 - Головна сторінка

Далі на рисунку 3.24 представлена сторінка нормативно-документативної інформації медичного закладу.

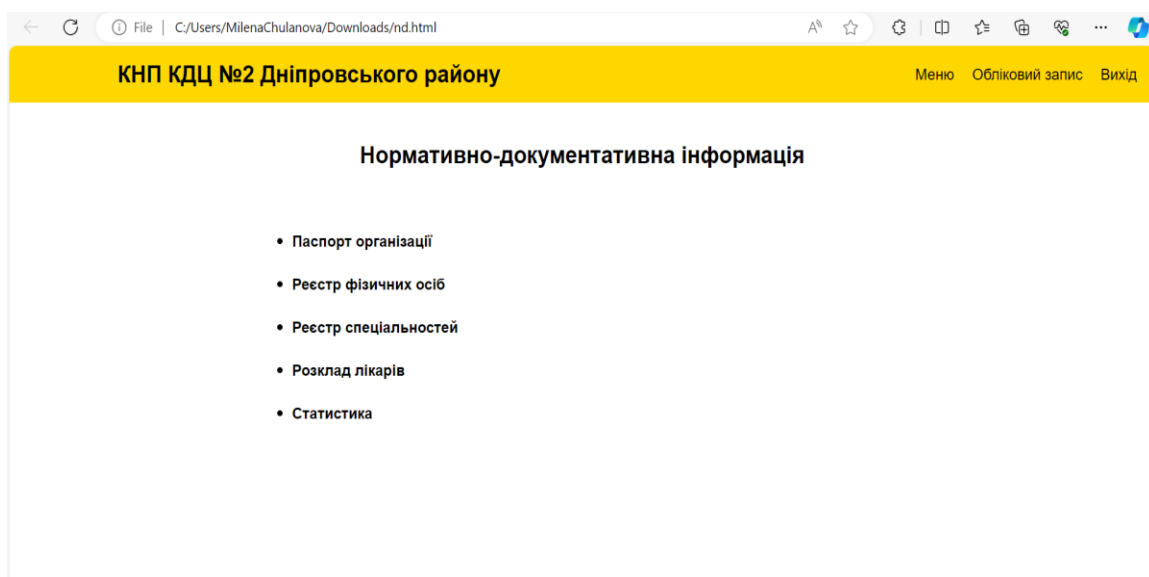


Рисунок 3.24 - Сторінка нормативно-документативної інформації

Для редагування та оновлення даних медичного закладу користувач переходить на сторінку паспорту організації, де може використовувати три

форми: загальна інформація, адреса та телефони, налаштування печатки (рис. 3.25).

Рисунок 3.25 - Сторінка паспорту організації

Користувач має можливість виконати пошук, внесення та редагування даних реєстраторів, адміністраторів та лікарів на відповідних сторінках (рис. 3.26 – 3.29).

Рисунок 3.26 - Сторінка реєстру фізичних осіб

КНП КДЦ №2 Дніпровського району Меню Обліковий запис Вихід

Лікарі

Додати

Спеціальність Введіть прізвище

Номер	Прізвище	Ім'я	По-батькові	Кабінет	Спеціальність
23424	Бірук	Наталія	Семенівна	32	Терапевт

Рисунок 3.27 - Сторінка реєстру лікарів

КНП КДЦ №2 Дніпровського району Меню Обліковий запис Вихід

Лікарі

Новий лікар

Номер: Введіть номер

Прізвище: Введіть прізвище

Ім'я: Введіть ім'я

По-батькові: Введіть ім'я по-батькові

Кабінет: Введіть кабінет

Спеціальність: Спеціальність

Додати документ Додати та закрити Закрити

Рисунок 3.28 - Форма додавання нового лікаря у систему

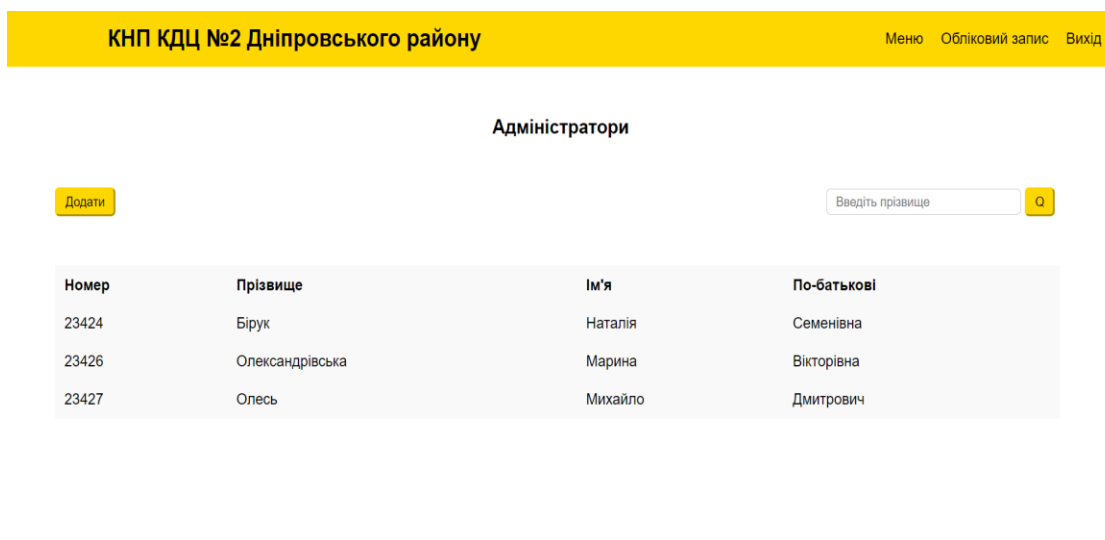


Рисунок 3.29 - Сторінка реєстру адміністраторів

На сторінці спеціальностей користувач може додати нову спеціальність, доступну у медичній структурі, здійснити пошук необхідної спеціальності, а натиснувши на неї, користувач може перейти на сторінку відображення лікарів за обраною спеціальністю (рис.3.30).

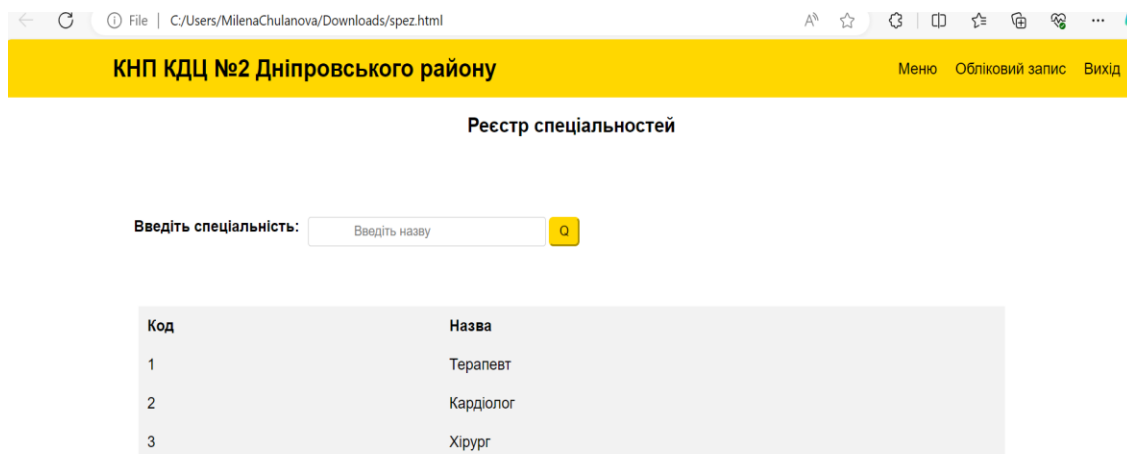


Рисунок 3.30 - Сторінка реєстру спеціальностей

Сторінка статистики надає ключові показники роботи медичного закладу через графіки та діаграми, дозволяючи користувачам аналізувати дані

та відстежувати тенденції. Функції фільтрації, сортування та експорту даних допомагають полегшувати налаштування та використання інформації, поданої у звітах, що автоматично генеруються системою (рис. 3.31 – 3.32).

Рисунок 3.31 - Сторінка подання внутрішньої статистики роботи закладу

Рисунок 3.32 - Сторінка формування звітів

Сторінка розкладу роботи лікарів надає користувачам інформацію про доступні тимчасові інтервали для запису на прийом до лікарів. Дану інформацію можна фільтрувати за спеціальністю та прізвищами медичних працівників. Прототип даної сторінки наведено на рисунку 3.33.

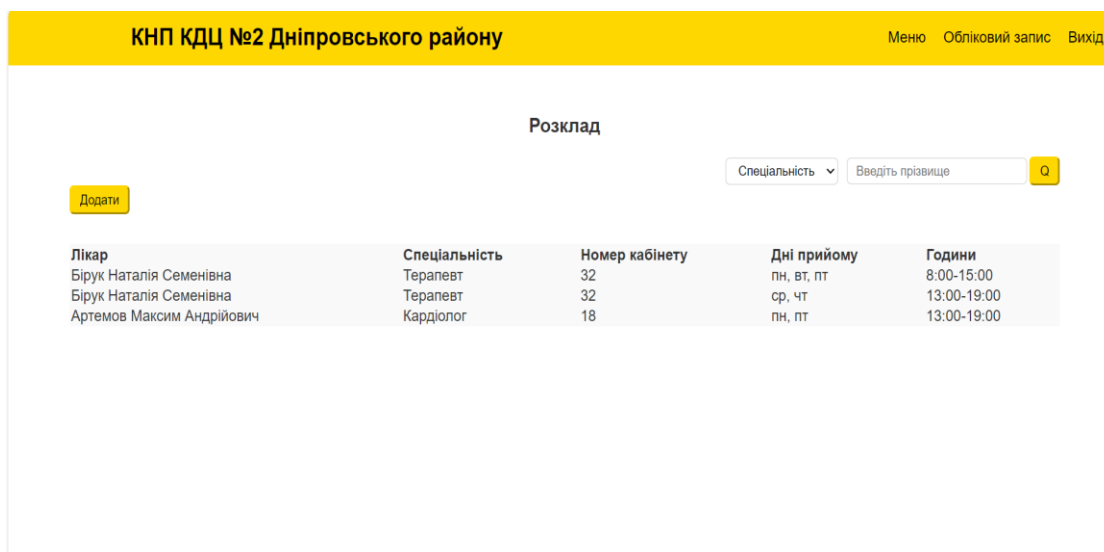


Рисунок 3.33 Сторінка розкладу роботи лікарів

Далі користувач має можливість перейти на основну сторінку роботи із системою, де можна перейти на інші сторінки роботи з медичними картками, даними пацієнтів і записами на прийом. Прототип даної сторінки відображений на рисунку 3.34.

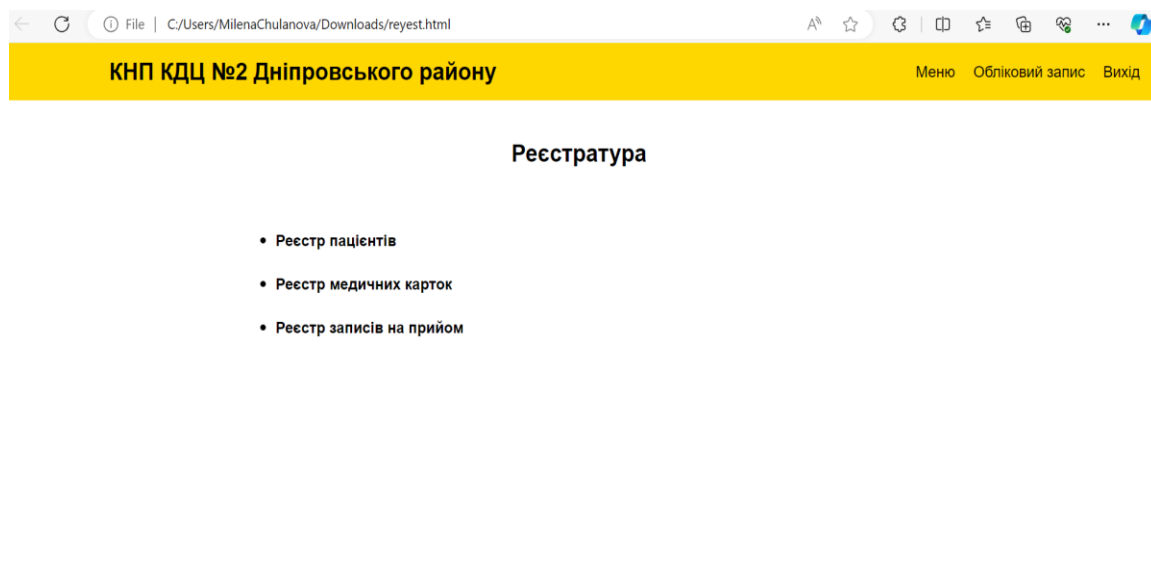


Рисунок 3.34 - Сторінка реєстратури

На сторінці "Реєстр медичних карток" представлена інформація про медичні картки пацієнтів. Співробітники поліклініки можуть переглядати,

шукати медичні картки пацієнтів за допомогою фільтрування та додавати нові (рис. 3.35 – 3.36).

КНП КДЦ №2 Дніпровського району Меню Обліковий запис Вихід

Реєстр медичних карток

Додати

Введіть прізвище лікаря Введіть прізвище пацієнта Введіть ІПН пацієнта Q

Номер	Пацієнт	Лікар	Група крові	Резус-фактор
71235	Олексів Максим Петрович	Бірюк Наталія Семенівна	2	+

Рисунок 3.35 - Сторінка реєстру медичних карток пацієнтів

КНП КДЦ №2 Дніпровського району Меню Обліковий запис Вихід

Медична карта пацієнта №752

ПІБ Пацієнта:

Лікарі:

Група крові:

Резус-фактор:

Група інвалідності:

Дата реєстрації:

Відобразити результати Очистити

June 2024

Su Mo Tu We Th Fr Sa

26 27 28 29 30 31 1

2 3 4 5 6 7 8

9 10 11 12 13 14 15

16 17 18 19 20 21 22

23 24 25 26 27 28 29

30 1 2 3 4 5 6

Clear Today

Рисунок 3.36 - Форма додавання нової медичної картки

Сторінка реєстру пацієнтів, дозволяє користувачам шукати, переглядати та додавати інформацію про пацієнтів.

Користувачі можуть шукати пацієнтів за номером телефону, ІПН або прізвища, а також переглядати їх медичні картки, дату народження та іншу

інформацію. При натисканні кнопки «Додати» відкривається форма створення нового пацієнта (рис. 3.37 – 3.39).

ІПН	Код	Мед. карта	Прізвище	Ім'я	По-батькові	Дата народження
3566308257	1	71235	Олексю	Максим	Петрович	30.11.2002
4532886790	2	16129	Антоненко	Олена	Дмитрівна	06.07.2000

Рисунок 3.37 - Сторінка реєстру пацієнтів

Новий пацієнт

ПІБ Пацієнта:

ІПН:

Телефон:

Адреса:

E-mail:

Дата народження:

Рисунок 3.38 - Форма додавання нового пацієнта

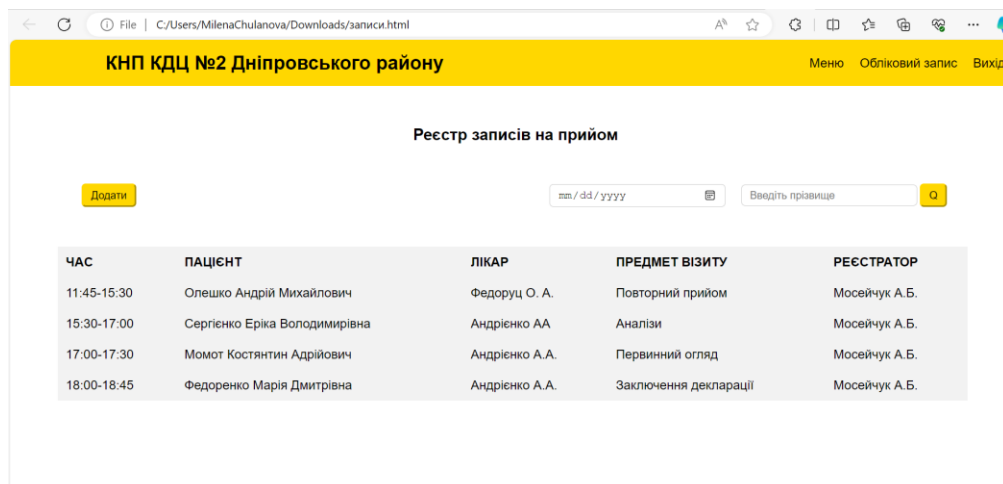


Рисунок 3.39 - Сторінка реєстру записів на прийом

Користувачі можуть знайти інформацію про раніше створені записи, додати новий запис, а також відфільтрувати записи по прізвищу пацієнта або даті візиту. Додатково сторінка надає можливість створення нового запису на прийом. Сторінку реєстру записів та форму створення нового запису показано на рисунках 3.40 – 3.421.

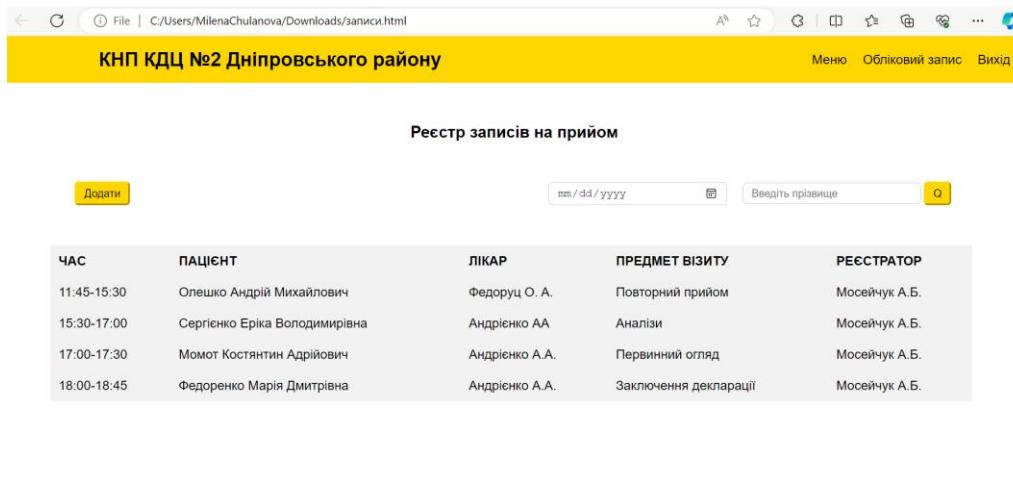


Рисунок 3.40 - Сторінка реєстрів записів на прийом

КНП КДЦ №2 Дніпровського району

Меню Обліковий запис Вихід

Реєстр записів на прийом

Дата візиту:

Час візиту:

Пацієнт:

Лікар:

Додатково

Відповідальний:

Предмет:

Рисунок 3.41 - Форма створення нового запису на прийом до лікаря

HTML та CSS код розробки даного прототипу системи наведено у додатку Д.

ВИСНОВОК

В рамках випускної кваліфікаційної роботи було здійснено проектування та розробку медичної інформаційної системи реєстратур медичних установ, яка має на меті покращити якість обслуговування пацієнтів та скорочення часу очікування прийому до лікаря.

У процесі виконання роботи було зроблено:

- Детальний аналіз предметної області, що включав визначення основних процесів, які відбуваються у медичній організації; аналіз практик впровадження МІС; огляд існуючих МІС реєстратури поліклініки та їх порівняння;
- Постановку вимог до системи, що розроблюється, включаючи бізнес-вимоги, функціональні та нефункціональні вимоги, що дозволило створити комплексну модель системи, яка може ефективно виконувати свої функції та задовольняти потреби користувачів;
- Функціональне та об'єктно-орієнтоване моделювання процесів для реалізації системи, на основі методології IDEF0 та уніфікованого моделювання UML;
- Реалізацію основних компонентів системи, включаючи інформаційне, технічне та програмне забезпечення;
- Розробку бази даних, вибір СКБД та мережевого обладнання, а також середовище розробки та мови програмування для реалізації системи;
- Створено прототипи інтерфейсів з використанням мови розмітки та мови стилів HTML та CSS.

Результатом виконання випускної кваліфікаційної роботи є розроблений прототип системи, розробка якого може бути продовжена, а отже отримані результати можуть слугувати фундаментом для створення повноцінної

медичної інформаційної системи та подальших досліджень у галузі розробки медичних інформаційних систем.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дмитро Плеханов, Хенрік Франке, Торбйорн Х. Нетланд (2022). Digital transformation: A review and research agenda, - 3 с. - ResearchGate. [Електронний ресурс]. URL: https://www.researchgate.net/publication/363798326_Digital_transformation_A_review_and_research_agenda
2. Том Сібель (2019) Digital Transformation: Survive and Thrive in the Era of Mass Extinction. – Boston: Harvard Business Review Press. - Розділ 2. “Digital transformation”.
3. Балдіні Джанмарко, Барбоні Марчелло, Боно Флавіо та ін. (2019) Digital Transformation of European Industry: A Policy Perspective. – Luxembourg: Publications Office of the European Union, – 148 с. – URL: <https://publications.jrc.ec.europa.eu/repository/handle/JRC116179>
4. Тополь, Є. (2020). Джон Халамка: How Mayo Clinic is Transforming Healthcare with A.I. Ground Truths - [Електронний ресурс]. URL: <https://erictopol.substack.com/p/john-halamka-how-mayo-clinic-is-transforming>
5. Азіз Джамал, Кірстен Вальмуур, Мікеле Кларк (2009). The Impact of Health Information Technology on the Quality of Medical and Health Care: A Systematic Review. – 26 с. - [Електронний ресурс]. URL: https://www.researchgate.net/publication/38054315_The_Impact_of_Health_Information_Technology_on_the_Quality_of_Medical_and_Health_Care_A_Systematic_Review
6. Карен А. Вейгер, Френсіс В. Лі, Джон П. Глейзер (2021). Health Care Information Systems: A Practical Approach for Health Care Management. Perlego. – 259, 23-24, 95, 342, 296-298 URL: <https://www.perlego.com/book/3118785/health-care-information-systems-a-practical-approach-for-health-care-management-pdf>
7. Девід Блюменталь, Мерилін Тавеннер (2010). The "meaningful use" regulation for electronic health records. PubMed. URL: <https://pubmed.ncbi.nlm.nih.gov/20647183/>
8. Джулія Адлер-Мільштейн, Кірстін Вуді Скотт, Ашіш К. Джа (2014). Leveraging EHRs to improve hospital performance: the role of management. PubMed. URL: <https://pubmed.ncbi.nlm.nih.gov/25811825/>

9. Caifeng Zhang, Rui Ma, Yujie Li, Yichuan Wang. (2019). Optimizing the Electronic Health Records Through Big Data Analytics: A Knowledge-Based View. [Електронний ресурс]. – 4-6 с.- URL: https://www.researchgate.net/publication/335633872_Optimizing_the_Electronic_Health_Records_Through_Big_Data_Analytics_A_Knowledge-Based_View
- 10.Марія Кучиньєлло, Ірвін Лепслі, Грета Насі, Клаудія Пальяррі (2015). Understanding key factors affecting electronic medical record implementation: a sociotechnical approach. – 9, 17-19 с. - [Електронний ресурс]. URL: <https://bmchealthservres.biomedcentral.com/articles/10.1186/s12913-015-0928-7>
- 11.AMIA 2024 Annual Symposium. (2024). American Medical Informatics Association. URL: <https://amia.org/education-events/amia-2024-annual-symposium>
- 12.Браян Гугерті та Кетлін Енн Маккормік (2013). Healthcare Information Technology Exam Guide for CompTIA Healthcare IT Technician and HIT Pro Certifications.
- 13.Тереза Заяс-Кабан, Джонатан С. Уолд (2020). Opportunities for the use of health information technology to support research. JAMIA Open, 3, - 321–327 с. – [Електронний ресурс]. URL: Opportunities for the use of health information technology to support research | JAMIA Open | Oxford Academic (oup.com)
- 14.Дмитро Самофалов (2021). Public management and public administration of hospital information system as the main tool of information communication technology in Ukrainian healthcare. – 50-51 с. - [Електронний ресурс]. URL: https://www.researchgate.net/publication/353797001_PUBLIC_MANAGEMENT_AND_PUBLIC_ADMINISTRATION_OF_HOSPITAL_INFORMATION_SYSTEM_AS_THE_MAIN_TOOL_OF_INFORMATION_COMMUNICATION_TECHNOLOGY_IN_UKRAINIAN_HEALTHCARE
- 15.Healthcare Data Management: Three Case Studies. HIMSS. [Електронний ресурс]. URL: <https://www.himss.org/resources/healthcare-data-management-three-case-studies>
- 16.Epic Systems Corporation. [Електронний ресурс]. URL: <https://www.epic.com/>

17. SoftServe Inc. [Електронний ресурс]. URL:
<https://www.softserveinc.com/en-us>
18. Азіз Джамал, Кірстен Вальмуур, Мікеле Кларк (2009). The Impact of Health Information Technology on the Quality of Medical and Health Care: A Systematic Review. – 30-33 с. - [Електронний ресурс]. URL:
https://www.researchgate.net/publication/38054315_The_Impact_of_Health_Information_Technology_on_the_Quality_of_Medical_and_Health_Care_A_Systematic_Review
19. Хауле К.Д., Муханга М.; Нгові Е. (2022). The what, why, and how of health information systems: A systematic review. – 37-43 с. - [Електронний ресурс]. URL:
<http://41.73.194.142/bitstream/handle/123456789/4398/Paper%205.pdf?sequence=1&isAllowed=y>
20. Tocci T. (2014). Major challenges and constraint of integrating health information systems in african countries: A Namibian experience. – 273-279 с. - [Електронний ресурс]. URL:
https://www.researchgate.net/profile/Titus-Tossy-2/publication/272163842_Major_Challenges_and_Constraint_of_Integrating_Health_Information_Systems_in_African_Countries_A_Namibian_Experience/links/54dca52b0cf28a3d93f8233d/Major-Challenges-and-Constraint-of-Integrating-Health-Information-Systems-in-African-Countries-A-Namibian-Experience.pdf
21. Лілі Кпобі, Леслі Шварц, Анжела Л. Офопі-Атта (2018). Challenges in the use of the mental health information system in a resource-limited setting: Lessons from Ghana. – 98 с. - [Електронний ресурс]. URL: Challenges in the use of the mental health information system in a resource-limited setting: lessons from Ghana - PubMed (nih.gov)
22. Фетейра-Сантос Р.; Камарінья С.; Нобре М.Д.; Еліас К.; Коста А.С.; Furtado С.; Ногейра П.Ж. (2022). Improving morbidity information in Portugal: Evidence from data linkage of COVID-19 cases surveillance and mortality systems. – 163 с. - [Електронний ресурс]. URL: Improving morbidity information in Portugal: Evidence from data linkage of COVID-19 cases surveillance and mortality systems - ScienceDirect
23. Кер Я.І.; Ванг Я.С.; Хайлі Н. (2018). Examining the impact of health information systems on healthcare service improvement: The case of reducing in patient-flow delays in a US hospital. – 127, 188-198 с. - [Електронний ресурс]. URL: Examining the impact of health information

- systems on healthcare service improvement: The case of reducing in patient-flow delays in a U.S. hospital - ScienceDirect
24. Мохаммад Ямін, Шейкер Алхарті (2019). Measuring impact of healthcare information systems in administration and operational management. - 5, 9-10 с. - [Електронний ресурс]. URL: (PDF) Measuring impact of healthcare information systems in administration and operational management (researchgate.net)
 25. Айман Аль-Ахмар, Муханнад Аль-Муса, Рачід Бенламер (2022). Automated clinical pathway standardization using SNOMED CT- based semantic relatedness. – 1-17 с. - [Електронний ресурс]. URL: Automated clinical pathway standardization using SNOMED CT- based semantic relatedness - PubMed (nih.gov)
 26. Красуська М.; Вільямс Р.; Шейх А.; Франклін Б.; Гіндер С.; Лейн В.; Мозаффар Х.; Мейсон К.; Ісон С.; та ін. (2021). Driving digital health transformation in hospitals: A formative qualitative evaluation of the English Global Digital Exemplar programme. – 28 с. - [Електронний ресурс]. URL: Driving digital health transformation in hospitals: a formative qualitative evaluation of the English Global Digital Exemplar programme - PubMed (nih.gov)
 27. Томас Джеймс Данн, Аннабель Браун, Стівен Хаворт, Фатіма Вурі, Інес Кампос-Матос (2021). Service Evaluation of the English Refugee Health Information System: Considerations and Recommendations for Effective Resettlement. – 18 с. - [Електронний ресурс]. URL: Service Evaluation of the English Refugee Health Information System: Considerations and Recommendations for Effective Resettlement - PubMed (nih.gov)
 28. Белло А.К.; Левін А. ; Лунні М.; Осман М.А.; Ашунтанг Дж.И., Беллорін-Фонт Е.; Бенганем Гарбі М.; Девісон С.; та ін. (2022). Availability, coverage, and scope of health information systems for kidney care across world countries and regions. – 159-167 с. - [Електронний ресурс]. URL: Availability, coverage, and scope of health information systems for kidney care across world countries and regions | Nephrology Dialysis Transplantation | Oxford Academic (oup.com)
 29. Р.Б. Панерай (2014). Health Information Systems; Global Perspective of Health; Department of Medical Physics, University of Leicester: Leicester, UK, - 1-6 с. - [Електронний ресурс]. URL: Health Information Systems (eolss.net)

- 30.Джоб Ньянгена, Рохіні Раджгопал, Елізабет Адхіамбо Омбеч, Енок Олоо та ін. (2021). Maturity assessment of Kenya's health information system interoperability readiness. - [Електронний ресурс]. URL: Maturity assessment of Kenya's health information system interoperability readiness - PMC (nih.gov)
- 31.Джордж Хріпцак, Лян Дж. (2013). The potential of electronic health records for reducing healthcare costs. Journal of the American Medical Informatics Association. – 326-330 с.
- 32.Грегорі Л. Александер, Кевін Дауті, Ендрю Джорджіу та ін. (2021). Advancing health information technology roadmaps in long term care – 3-7 с. - [Електронний ресурс]. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7340178/pdf/nihms-1594548.pdf>
- 33.Пуассан Л., Перейра Дж., Тамблін Р. (2005). The Impact of Electronic Health Records on Time Efficiency of Physicians and Nurses: A Systematic Review – 507-508, 514 с. - [Електронний ресурс]. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1205599/pdf/505.pdf>
- 34.Міністрів. Про затвердження Положення про Міністерство охорони здоров'я України: Постанова від 28 березня 2018 р. № 411. - [Електронний ресурс]. URL: <https://zakon.rada.gov.ua/laws/show/411-2018-%D0%BF>
- 35.Головний військовий клінічний госпіталь МВС України. Стаціонарні відділення. - [Електронний ресурс]. URL: <https://kr-hospital.mvs.gov.ua/departments/show/2>
- 36.Великий гід по ХЕЛСІ: як зареєструватися і чим може допомогти [Електронний ресурс]. - [Електронний ресурс]. URL: <https://bf.diia.gov.ua/articles/velikij-gid-po-helsi-yak-zareyestrivatis-i-chim-mozhe-dopomogti>
- 37.Всі переваги електронної медичної карти пацієнта. БукІнфо. - [Електронний ресурс]. URL: <https://bukinfo.com.ua/zdorov-ya/vsi-perevagy-elektronnoji-medychnoji-karty-paciyenta>
- 38.Медичні інформаційні системи: як вони працюють і чому їх важливо впроваджувати [Електронний ресурс]. - URL: <https://evergreens.com.ua/ua/articles/medical-information-systems.html>
- 39.РНАК. Технічне завдання. - [Електронний ресурс]. URL: <https://pnak.ru/uk/tehnicheskaya-dokumentaciya-tehnicheskoe-zadanie-na/>

40. QA Group. "SQL шпаргалка на співбесіду." - [Електронний ресурс].
URL: <https://qagroup.com.ua/publications/sql-info/>
41. Freehost. "FAQ "Apache: що це? " - [Електронний ресурс]. URL:
<https://freehost.com.ua/ukr/faq/wiki/apache-cto-eto/>
42. Фріденталь, С., Мур, А., і Штайнер, Р. (2014). A Practical Guide to SysML: The Systems Modeling Language. – Розділ 5. «Viewing SysML Models with Diagrams». [Електронний ресурс]. - URL: A Practical Guide to SysML, 3rd Edition [Book] (oreilly.com)
43. Карл Е. Вігерс, Джой Бітті (2013). Software Requirements (3rd Edition). [Електронний ресурс]. – С. 77-78, 9-11 - URL:
https://www.booksfree.org/wpcontent/uploads/2022/03/Software_Requirements_3rd_Edition_compressed.pdf
44. Гершенсон С. (ред. 2019). Handbook of Research on the Object-Oriented Paradigm in Information Systems. - С. 31-35.

ДОДАТКИ

ДОДАТОК А

Специфікації вимог

☒ Автоматизація звітування та аналізу		Proposed	High	High
☒ Автоматизація процесів запису на прийом до лікаря		Proposed	Medium	High
☒ Автоматизація процесу формування черги на прийом		Proposed	Medium	High
☒ Автоматизація рутинних завдань реєстратури		Proposed	Low	High
☒ Впровадження електронного документообігу	business re...	Proposed	Low	Medium
☒ Дотримання законодавчих та нормативних вимог		Proposed	Medium	Medium
☒ Забезпечення повної автоматизації обліку пацієнтів		Proposed	Medium	Medium
☒ Інтеграція з національною системою охорони здоров'я	FunctionalL...	Proposed	Medium	Medium
☒ Інтеграція модулів звітності та статистики		Proposed	Medium	Medium

Рисунок А.1 - Специфікація бізнес-вимог

Specification Manager				
Specification Manager: Package: "Функціональні вимоги" [Requirement]				
+ - Model Diagrams Traceability Starter Package Diagram Функціональні вимоги				
Item	Difficulty	Status	Type	Priority
☒ Автоматизація ведення графіку роботи лікарів	Low	Proposed	Requir...	High
☒ Копіювання розкладів на наступні тижні	Low	Proposed	Requir...	High
☒ Пошук розкладу роботи лікарів за ПІБ та спеціальністю	Low	Proposed	Requir...	High
☒ Автоматизація запису на пацієнтів на прийом	Medium	Proposed	Requir...	High
☒ Доступ до розкладу роботи лікарів	Medium	Proposed	Requir...	High
☒ Перегляд вільних слотів для запису до фахівця на обрану дату	Medium	Proposed	Requir...	High
☒ Пошуку лікарів за ПІБ та спеціальністю	Low	Proposed	Requir...	High
☒ Створення електронних талонів	Medium	Proposed	Requir...	High
☒ Автоматизація реєстрації пацієнтів	Medium	Proposed	Requir...	High
☒ Внесення персональних документів пацієнтів через сканер	Medium	Proposed	Requir...	High

Рисунок А.2 - Специфікація функціональних вимог

Specification Manager

Specification Manager:: Package: "Функціональні вимоги" [Requirement]

Model > Diagrams > Traceability > Starter Package Diagram > Функціональні вимоги

Item	Difficulty	Status	Type	Priority
☒ Друк документів пацієнта	Medium	Proposed	Requir...	High
☒ Зберігання персональних даних пацієнтів	Medium	Proposed	Requir...	High
☒ Забезпечення керування медичними картками пацієнтів	Medium	Proposed	Requir...	High
☒ Автоматизація створення нових медичних карток	Medium	Proposed	Requir...	High
☒ Забезпечення можливості фільтрації медичних карток	Low	Proposed	Requir...	High
☒ Можливість внесення та зміни інформації в існуючі медичні карти, з можливістю відстеження всіх змін	Medium	Proposed	Requir...	High
☒ Можливість пошуку медичних карток з використанням різних критеріїв	Low	Proposed	Requir...	High
☒ Реалізація полегшення ведення адміністративних процесів	Medium	Proposed	Requir...	High
☒ Додавання, видалення та зміна профілю кожного лікаря та реєстратора	Medium	Proposed	Requir...	High

Продовження рисунку А.2

Specification Manager

Specification Manager:: Package: "Функціональні вимоги" [Requirement]

Model > Diagrams > Traceability > Starter Package Diagram > Функціональні вимоги

Item	Difficulty	Status	Type	Priority
☒ Керування списком доступних лікарських спеціалізацій	Medium	Proposed	Requir...	High
☒ Оновлення основної інформації про медзаклад	Medium	Proposed	Requir...	High
☒ Реалізація системи аутентифікації та авторизації	Medium	Proposed	Requir...	High
☒ Забезпечення механізму аутентифікації	High	Proposed	Requir...	High
☒ Реалізація механізму перевірки складності пароля	Medium	Proposed	Requir...	High
☒ Розмежування доступу до функціоналу залежно від ролі користувача	Medium	Proposed	Requir...	High
☒ Реалізація статистичного аналізу	High	Proposed	Requir...	Medium
☒ Збір даних з внутрішніх джерел	Medium	Proposed	Requir...	Medium

Продовження рисунку А.2

Specification Manager

Specification Manager:: Package: "Функціональні вимоги" [Requirement]

Model > Diagrams > Traceability > Starter Package Diagram > Функціональні вимоги

Item	Difficulty	Status	Type	Priority
☒ Збір даних з внутрішніх джерел	Medium	Proposed	Requir...	Medium
☒ Здійснення статистичних висновків	Medium	Proposed	Functi...	Medium
☒ Обробка зібраних даних	High	Proposed	Functi...	Medium
☒ Реалізація функції друку	Low	Proposed	Requir...	Medium
☒ Реєстрація користувачів системи	High	Proposed	Requir...	High

Add New

Продовження рисунку А.2

Specification Manager

Specification Manager:: Package: "Нефункціональні вимоги" [Requirement]

Model > Diagrams > Traceability > Starter Package Diagram > Нefункціональні вимоги

Item	Difficulty	Status	Type	Priority
☒ Акаунти повинні бути наділені правами доступу, відповідними до певних посад користувачів	High	Proposed	Requir...	High
☒ Всі екранні форми інтерфейсу користувача повинні бути виконані в єдиному графічному стилі з однаковим розташуванням основних елементів управління і навігації	Medium	Proposed	Requir...	High
☒ Для позначення подібних операцій повинні використовуватися керуючі елементи	Low	Proposed	Requir...	Medium
☒ Забезпечення візуального графічного інтерфейсу GUI	Low	Proposed	Requir...	High
☒ Інтерфейс повинен підтримувати можливість введення та виведення з клавіатури та за допомогою миші	Low	Proposed	Requir...	High
☒ Інтерфейс системи має бути локалізований українською та англійською мовами	Low	Proposed	Requir...	High
☒ Інтерфейс системи має бути простим, інтуїтивно зрозумілим та заснований на уніфікованих формах введення інформації	Low	Proposed	Requir...	High

Рисунок А.3 - Специфікація нефункціональних вимог

Specification Manager

Specification Manager:: Package: "Нефункціональні вимоги" [Requirement]

Model > Diagrams > Traceability > Starter Package Diagram > Нефункціональні вимоги

Item	Difficulty	Status	Type	Priority
☒ Основний системний інтерфейс слід створювати за допомогою стандартного графічного інтерфейсу операційної системи MS Windows	Low	Proposed	Requir...	Medium
☒ Система має використовувати брендмауери з метою захисту мережевої взаємодії системи	Medium	Proposed	Requir...	High
☒ Система має забезпечувати надійність роботи не менше 99% для мінімізації системних збоїв	High	Proposed	Requir...	High
☒ Система повинна автоматично блокувати акаунти користувачів на 30 хвилин після трьох невдалих спроб введення пароля	Medium	Proposed	Requir...	High
☒ Система повинна дотримуватись загальних правил оформлення екранів, включаючи шрифти, колірну палітру, розташування вікон та елементів керування, а також правила використання клавіатури та миші	Low	Proposed	Requir...	High
☒ Система повинна забезпечувати відповіді на користувацькі запити впродовж 2-3 секунд.	High	Proposed	Requir...	High

Продовження рисунку А.3

Specification Manager

Specification Manager:: Package: "Нефункціональні вимоги" [Requirement]

Model > Diagrams > Traceability > Starter Package Diagram > Нефункціональні вимоги

Item	Difficulty	Status	Type	Priority
☒ Система повинна завантажувати сторінку не більше 2 секунд	High	Proposed	Requir...	High
☒ Система повинна мати здатність автоматично відновлюватися протягом 10 хвилин без втрати даних	High	Proposed	Requir...	High
☒ Системна інфраструктура повинна бути розміщена в сухих, прохолодних приміщеннях, та приміщеннях, що підлягають цілодобовому контролю	Medium	Proposed	Requir...	High
☒ Формати дати та часу мають відповідати формату часової зони поліклініки	Medium	Proposed	Requir...	High

Add New

Продовження рисунку А.3

ДОДАТОК Б

Етапи розробки інфологічної моделі бази даних

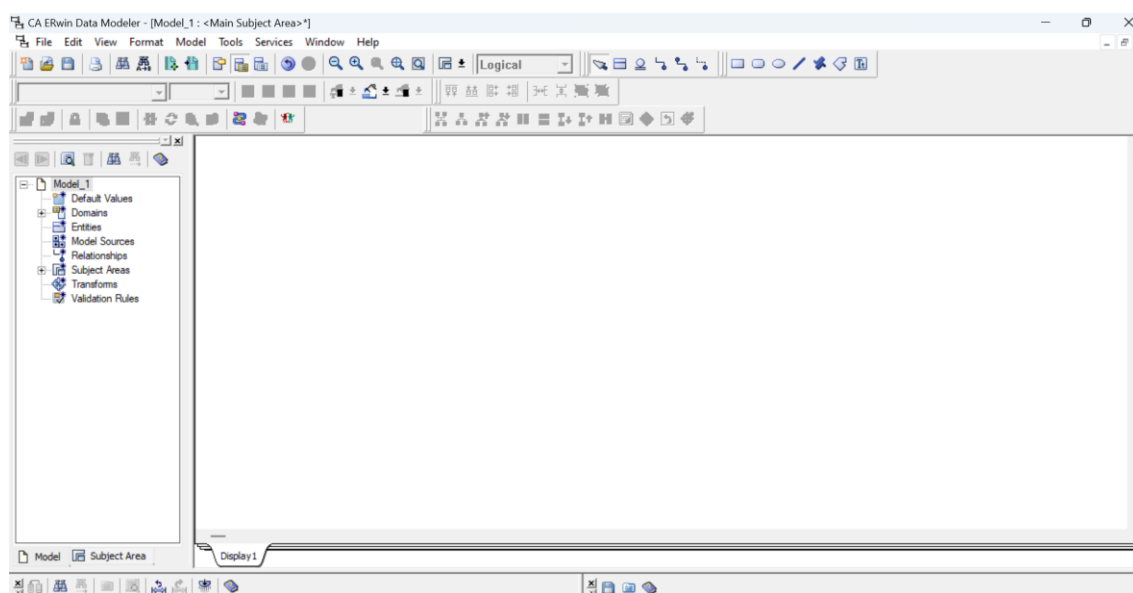


Рисунок Б.1 – Середовище Erwin Data Modeler

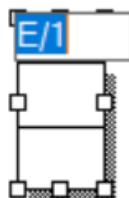


Рисунок Б.2 - Зміна назви таблиці

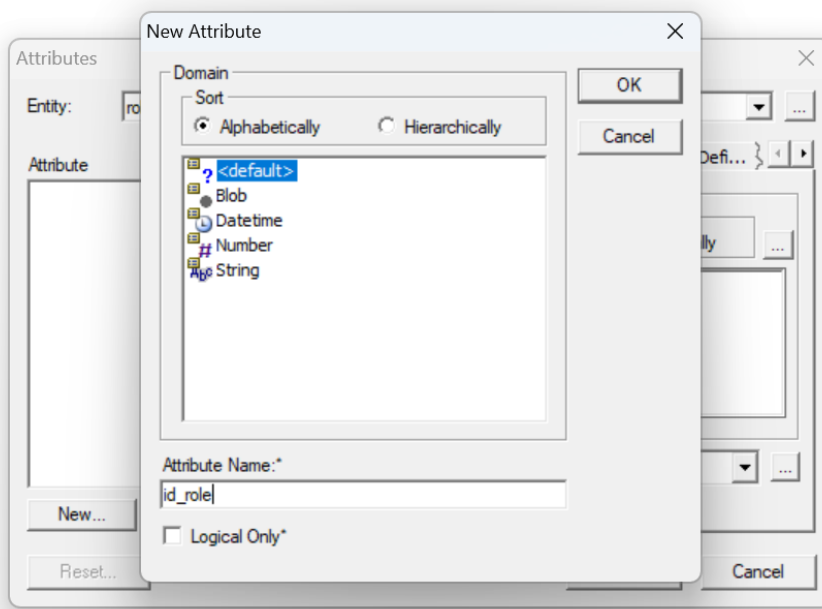


Рисунок Б.3 - Додавання нових атрибутів

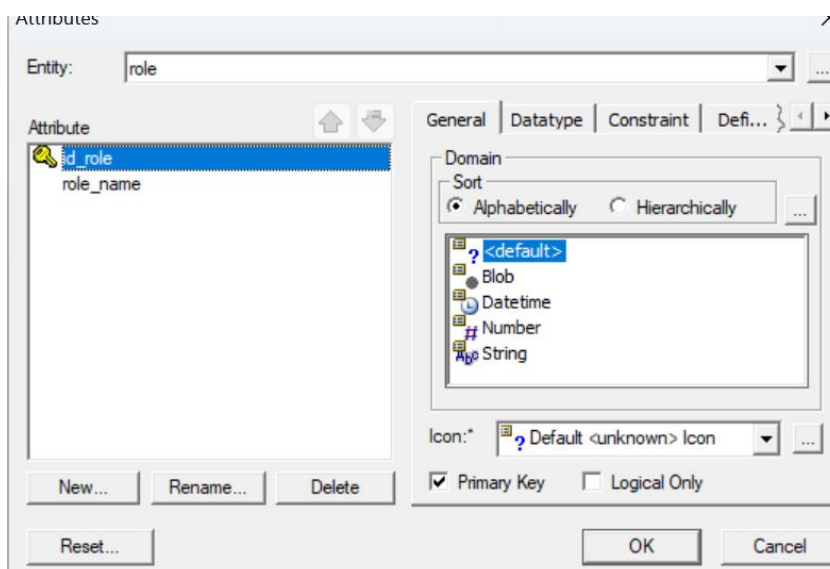


Рисунок Б.4 - Визначення атрибута



Рисунок Б.5 - Встановлення зв'язку між таблицями

ДОДАТОК В

SQL код

```
CREATE SCHEMA `mydbs`;
```

```
USE `mydbs`;
```

```
CREATE TABLE `appointment_reason` (  
  `id_appointment_reason` CHAR(2) NOT NULL,  
  `appointment_reason_name` VARCHAR(100) NOT NULL,  
  PRIMARY KEY (`id_appointment_reason`)  
);
```

```
CREATE TABLE `policlinica` (  
  `id_policlinica` CHAR(11) NOT NULL,  
  `policlinica_lname` VARCHAR(255) NOT NULL,  
  `policlinica_sname` VARCHAR(45) NOT NULL,  
  `policlinica_rnokpp` CHAR(10) NOT NULL,
```

```

`policlinica_address` VARCHAR(100) NOT NULL,
`policlinica_email` VARCHAR(45) NOT NULL,
`policlinica_type` VARCHAR(45) NOT NULL,
`establishment_date` DATE NOT NULL,
`policlinica_phone` VARCHAR(20) NOT NULL,
`policlinica_postadress` VARCHAR(100) NOT NULL,
PRIMARY KEY (`id_policlinica`)
);

```

```

CREATE TABLE `specialization` (
  `id_specialization` CHAR(3) NOT NULL,
  `specialization_name` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`id_specialization`)
);

```

```

CREATE TABLE `doctor` (
  `id_doctor` CHAR(7) NOT NULL,
  `doctor_name` VARCHAR(45) NOT NULL,
  `doctor_office` INT NOT NULL,
  `policlinica_id_policlinica` CHAR(11) NOT NULL,
  `specialization_id_specialization` CHAR(3) NOT NULL,
  PRIMARY KEY (`id_doctor`),
  CONSTRAINT `fk_doctor_policlinica1`
    FOREIGN KEY (`policlinica_id_policlinica`)
    REFERENCES `policlinica` (`id_policlinica`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_doctor_specialization1`
    FOREIGN KEY (`specialization_id_specialization`)
    REFERENCES `specialization` (`id_specialization`)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

```

);

```
CREATE TABLE `patient` (
  `id_patient` CHAR(4) NOT NULL,
  `patient_name` VARCHAR(45) NOT NULL,
  `patient_ipn` CHAR(10) NOT NULL,
  `patient_address` VARCHAR(100) NULL DEFAULT NULL,
  `patient_phone` VARCHAR(20) NOT NULL,
  `patient_birthdate` DATE NOT NULL,
  `patient_email` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`id_patient`)
);
```

```
CREATE TABLE `role` (
  `id_role` CHAR(2) NOT NULL,
  `role_name` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`id_role`)
);
```

```
CREATE TABLE `user` (
  `id_user` CHAR(2) NOT NULL,
  `user_name` VARCHAR(45) NOT NULL,
  `user_email` VARCHAR(45) NOT NULL,
  `user_password` VARCHAR(20) NOT NULL,
  `user_phone` VARCHAR(20) NOT NULL,
  `policlinica_id_policlinica` CHAR(11) NOT NULL,
  `role_id_role` CHAR(2) NOT NULL,
  PRIMARY KEY (`id_user`, `policlinica_id_policlinica`, `role_id_role`),
  CONSTRAINT `fk_user_policlinica`
  FOREIGN KEY (`policlinica_id_policlinica`)
```

```

REFERENCES `policlinica` (`id_policlinica`)
ON DELETE CASCADE
ON UPDATE CASCADE,
CONSTRAINT `fk_user_role1`
FOREIGN KEY (`role_id_role`)
REFERENCES `role` (`id_role`)
ON DELETE CASCADE
ON UPDATE CASCADE
);
CREATE TABLE IF NOT EXISTS `appointment` (
`id_appointment` CHAR(4) NOT NULL,
`appointment_time` TIME NOT NULL,
`appointment_date` DATE NOT NULL,
`creation_date` DATE NOT NULL,
`user_id_user` CHAR(2) NOT NULL,
`patient_id_patient` CHAR(4) NOT NULL,
`doctor_id_doctor` CHAR(7) NOT NULL,
`appointment_reason_id_appointment_reason` CHAR(2) NULL DEFAULT NULL,
PRIMARY KEY (`id_appointment`, `patient_id_patient`),
CONSTRAINT `fk_appointment_appointment_reason1`
FOREIGN KEY (`appointment_reason_id_appointment_reason`)
REFERENCES `appointment_reason` (`id_appointment_reason`)
ON DELETE CASCADE
ON UPDATE CASCADE,
CONSTRAINT `fk_appointment_doctor1`
FOREIGN KEY (`doctor_id_doctor`)
REFERENCES `doctor` (`id_doctor`)
ON DELETE CASCADE
ON UPDATE CASCADE,
CONSTRAINT `fk_appointment_patient1`
FOREIGN KEY (`patient_id_patient`)
REFERENCES `patient` (`id_patient`)

```

```

ON DELETE CASCADE
ON UPDATE CASCADE,
CONSTRAINT `fk_appointment_user1`
FOREIGN KEY (`user_id_user`)
REFERENCES `user` (`id_user`)
ON DELETE CASCADE
ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS `document` (
  `id_document` CHAR(11) NOT NULL,
  `document_name` VARCHAR(45) NOT NULL,
  `patient_id_patient` CHAR(4) NOT NULL,
  PRIMARY KEY (`id_document`, `patient_id_patient`),
  CONSTRAINT `fk_document_patient1`
FOREIGN KEY (`patient_id_patient`)
REFERENCES `patient` (`id_patient`)
ON DELETE CASCADE
ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS `medcard` (
  `id_medcard` CHAR(11) NOT NULL,
  `resus_phactor` CHAR(1) NULL DEFAULT NULL,
  `blood_group` INT NULL DEFAULT NULL,
  `creation_date` DATE NOT NULL,
  `disability_group` CHAR(1) NOT NULL,
  `doctor_id_doctor` CHAR(7) NOT NULL,
  `patient_id_patient` CHAR(4) NOT NULL,
  PRIMARY KEY (`id_medcard`, `patient_id_patient`),
  CONSTRAINT `fk_medcard_doctor1`

```

```

FOREIGN KEY (`doctor_id_doctor`)
REFERENCES `doctor` (`id_doctor`)
ON DELETE CASCADE
ON UPDATE CASCADE,
CONSTRAINT `fk_medcard_patient1`
FOREIGN KEY (`patient_id_patient`)
REFERENCES `patient` (`id_patient`)
ON DELETE CASCADE
ON UPDATE CASCADE
);

CREATE TABLE IF NOT EXISTS `shifts` (
  `id_shifts` CHAR(2) NOT NULL,
  `hours_from` TIME NULL DEFAULT NULL,
  `hours_to` TIME NULL DEFAULT NULL,
  PRIMARY KEY (`id_shifts`)
);

CREATE TABLE IF NOT EXISTS `schedule` (
  `id_schedule` CHAR(2) NOT NULL,
  `working_days` VARCHAR(45) NOT NULL,
  `doctor_id_doctor` CHAR(7) NOT NULL,
  `shifts_id_shifts` CHAR(2) NOT NULL,
  PRIMARY KEY (`id_schedule`),
  CONSTRAINT `fk_schedule_doctor1`
FOREIGN KEY (`doctor_id_doctor`)
REFERENCES `doctor` (`id_doctor`)
ON DELETE CASCADE
ON UPDATE CASCADE,
CONSTRAINT `fk_schedule_shifts1`
FOREIGN KEY (`shifts_id_shifts`)
REFERENCES `shifts` (`id_shifts`)

```

ON DELETE CASCADE

ON UPDATE CASCADE

);

INSERT INTO appointment (id_appointment, appointment_time, appointment_date, creation_date, user_id_user, patient_id_patient, doctor_id_doctor, appointment_reason_id_appointment_reason)

VALUES ('0001', '08:30:00', '2024-06-12', '2024-05-31', '02', '0001', '001-PED', '01');

INSERT INTO appointment (id_appointment, appointment_time, appointment_date, creation_date, user_id_user, patient_id_patient, doctor_id_doctor, appointment_reason_id_appointment_reason)

VALUES ('0002', '17:00:00', '2024-06-01', '2024-05-31', '03', '0002', '001-NEU', '01');

INSERT INTO appointment (id_appointment, appointment_time, appointment_date, creation_date, user_id_user, patient_id_patient, doctor_id_doctor, appointment_reason_id_appointment_reason)

VALUES ('0003', '19:30:00', '2024-06-01', '2024-06-05', '02', '0003', '001-CAR', '02');

INSERT INTO specialization (id_specialization, specialization_name)

VALUES ('CAR', 'Кардіолог');

INSERT INTO specialization (id_specialization, specialization_name)

VALUES ('NEU', 'Невропатолог');

INSERT INTO specialization (id_specialization, specialization_name)

VALUES ('PED', 'Педіатр');

INSERT INTO specialization (id_specialization, specialization_name)

VALUES ('TER', 'Терапевт');

INSERT INTO appointment_reason (id_appointment_reason, appointment_reason_name)

VALUES ('01', 'Первинна консультація');

INSERT INTO appointment_reason (id_appointment_reason, appointment_reason_name)

VALUES ('02', 'Повторна консультація');

INSERT INTO appointment_reason (id_appointment_reason, appointment_reason_name)

VALUES ('03', 'Лікування гострого стану');

INSERT INTO appointment_reason (id_appointment_reason, appointment_reason_name)

VALUES ('04', 'Діагностичні тести');

INSERT INTO appointment_reason (id_appointment_reason, appointment_reason_name)

VALUES ('05', 'Профілактичний огляд');

INSERT INTO policlinica (id_policlinica, policlinica_lname, policlinica_sname, policlinica_rnokpp, policlinica_address, policlinica_email, policlinica_type, establishment_date, policlinica_phone, policlinica_postaddress)

VALUES ('POD-KYV-001', 'КОМУНАЛЬНЕ НЕКОМЕРЦІЙНЕ ПІДПРИЄМСТВО «ЦЕНТР ПЕРВИННОЇ МЕДИКО-САНІТАРНОЇ ДОПОМОГИ № 1» ПОДІЛЬСЬКОГО РАЙОНУ М. КИЄВА', 'КНП «ЦПМСД № 1» ПОДІЛЬСЬКОГО РАЙОНУ М. КИЄВА', '3894626890', 'Україна, місто Київ, вулиця Волоська, будинок 47', 'podpol3@ukr.net', 'Державна', '2022-10-20', '+380 (44) 482-44-02', '04070');

INSERT INTO doctor (id_doctor, doctor_name, doctor_office, policlinica_id_policlinica, specialization_id_specialization)

VALUES ('001-CAR', 'Бондаренко Андрій Петрович', 12, 'POD-KYV-001', 'CAR');

INSERT INTO doctor (id_doctor, doctor_name, doctor_office, policlinica_id_policlinica, specialization_id_specialization)

VALUES ('001-NEU', 'Максимчук Олексій Вікторович', 15, 'POD-KYV-001', 'NEU');

INSERT INTO doctor (id_doctor, doctor_name, doctor_office, policlinica_id_policlinica, specialization_id_specialization)

VALUES ('001-PED', 'Назар Катерина Олександрівна', 5, 'POD-KYV-001', 'PED');

INSERT INTO doctor (id_doctor, doctor_name, doctor_office, policlinica_id_policlinica, specialization_id_specialization)

VALUES ('002-CAR', 'Лисенко Тетяна Валентинівна', 12, 'POD-KYV-001', 'CAR');

INSERT INTO doctor (id_doctor, doctor_name, doctor_office, policlinica_id_policlinica, specialization_id_specialization)

VALUES ('003-TER', 'Грищенко Ірина Сергіївна', 6, 'POD-KYV-001', 'TER');

INSERT INTO patient (id_patient, patient_name, patient_ipn, patient_phone, patient_birthdate)

```
VALUES ('0001', 'Бондаренко Іван Петрович', '1234567890', '+380987654321', '1970-01-01');
```

```
INSERT INTO patient (id_patient, patient_name, patient_ipn, patient_phone, patient_birthdate)
VALUES ('0002', 'Максимчук Петро Степанович', '0987654321', '+380671234567', '1980-05-12');
```

```
INSERT INTO patient (id_patient, patient_name, patient_ipn, patient_phone, patient_birthdate)
VALUES ('0003', 'Назарова Ольга Дмитрівна', '2345678901', '+380509876543', '1990-03-25');
```

```
INSERT INTO patient (id_patient, patient_name, patient_ipn, patient_phone, patient_birthdate)
VALUES ('0004', 'Лисиченко Юрій Миколайович', '3456789012', '+380984567890', '2000-11-10');
```

```
INSERT INTO patient (id_patient, patient_name, patient_ipn, patient_phone, patient_birthdate)
VALUES ('0005', 'Гриценко Данило Сергійович', '4567890123', '+380631234567', '2010-07-15');
```

```
INSERT INTO role (id_role, role_name)
VALUES ('01', 'Адміністратор');
```

```
INSERT INTO role (id_role, role_name)
VALUES ('02', 'Ресстратор');
```

```
INSERT INTO user (id_user, user_name, user_email, user_password, user_phone, policlinica_id_policlinica,
role_id_role)
VALUES ('01', 'Петренко Олена Ігорівна', 'olena.petrenko@gmail.com', '111111', '+380501234567', 'POD-KYV-001', '01');
```

```
INSERT INTO user (id_user, user_name, user_email, user_password, user_phone, policlinica_id_policlinica,
role_id_role)
VALUES ('02', 'Кондратенко Олександр Васильович', 'kondr@gmail.com', '000000', '+380671234567', 'POD-KYV-001', '02');
```

```
INSERT INTO user (id_user, user_name, user_email, user_password, user_phone, policlinica_id_policlinica,
role_id_role)
VALUES ('03', 'Іваненко Марія Павлівна', 'mariya.ivanenko@ukr.net', '222222', '+380911234567', 'POD-KYV-001', '02');
```

```
INSERT INTO document (id_document, document_name, patient_id_patient)
VALUES ('KON-001', 'Кондратенко К.К.', '0001');
```

```
INSERT INTO document (id_document, document_name, patient_id_patient)
VALUES ('MAKS-001', 'Максименко Л. Д.', '0002');
```

```
INSERT INTO document (id_document, document_name, patient_id_patient)
VALUES ('OLEK-001', 'Олександрівська О. М.', '0003');
```

```
INSERT INTO medcard (id_medcard, resus_phactor, blood_group, creation_date, disability_group,
doctor_id_doctor, patient_id_patient)
VALUES ('08022000-01', '+', 1, '2023-12-28', '0', '003-TER', '0001');
```

```
INSERT INTO medcard (id_medcard, resus_phactor, blood_group, creation_date, disability_group,
doctor_id_doctor, patient_id_patient)
VALUES ('12082003-01', '+', 2, '2022-05-05', '0', '003-TER', '0001');
```

```
INSERT INTO medcard (id_medcard, resus_phactor, blood_group, creation_date, disability_group,
doctor_id_doctor, patient_id_patient)
VALUES ('20011995-01', '-', 1, '2023-05-12', '0', '003-TER', '0002');
```

```
INSERT INTO shifts (id_shifts, hours_from, hours_to)
VALUES ('01', '15:00:00', '21:00:00');
```

```
INSERT INTO shifts (id_shifts, hours_from, hours_to)
VALUES ('02', '13:00:00', '19:00:00');
```

```
INSERT INTO shifts (id_shifts, hours_from, hours_to)
VALUES ('03', NULL, NULL);
```

```
INSERT INTO schedule (id_schedule, working_days, doctor_id_doctor, shifts_id_shifts)
VALUES ('01', 'ПН, ВТ, ЧТ', '001-CAR', '01');
```

```
INSERT INTO schedule (id_schedule, working_days, doctor_id_doctor, shifts_id_shifts)
VALUES ('02', ' ', '001-CAR', '03');
```

```
INSERT INTO schedule (id_schedule, working_days, doctor_id_doctor, shifts_id_shifts)
VALUES ('03', 'ПТ, сб, вс', '002-CAR', '02');
```

```
INSERT INTO schedule (id_schedule, working_days, doctor_id_doctor, shifts_id_shifts)
VALUES ('04', 'ПН, Вт, ПТ', '001-PED', '01');
```

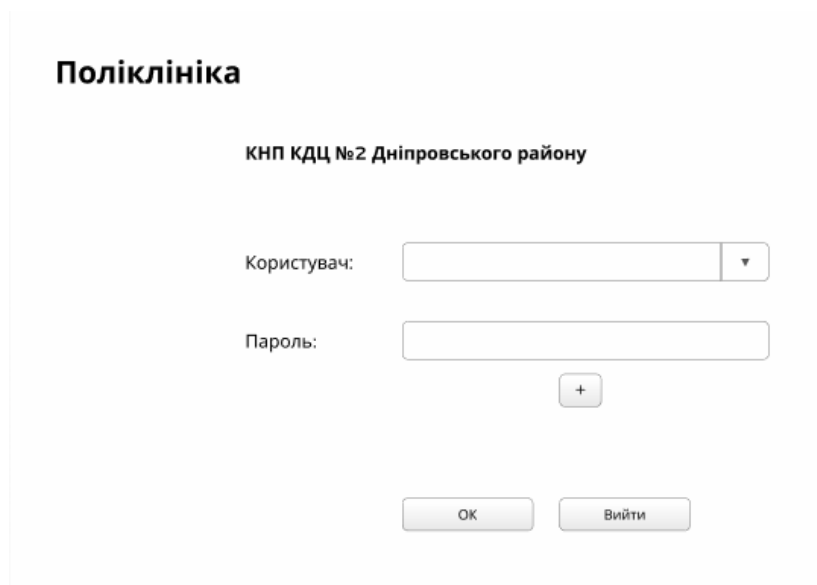
```
INSERT INTO schedule (id_schedule, working_days, doctor_id_doctor, shifts_id_shifts)
VALUES ('05', 'ср, сб', '001-NEU', '03');
```

```
INSERT INTO schedule (id_schedule, working_days, doctor_id_doctor, shifts_id_shifts)
VALUES ('06', 'Вт, чт, вс', '003-TER', '01');
```

```
INSERT INTO schedule (id_schedule, working_days, doctor_id_doctor, shifts_id_shifts)
VALUES ('07', 'ПН, ПТ', '003-TER', '02');
```

ДОДАТОК Г

Структури прототипу, розроблені у середовищі Figma



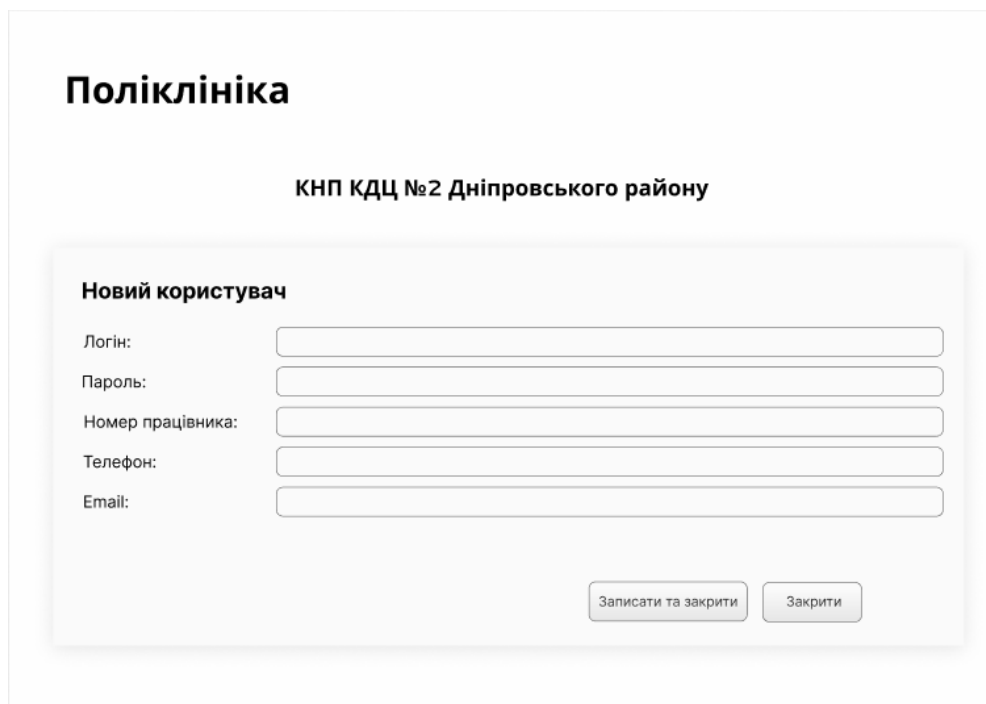
Поліклініка

КНП КДЦ №2 Дніпровського району

Користувач:

Пароль:

Рисунок Г.1- Сторінка авторизації



Поліклініка

КНП КДЦ №2 Дніпровського району

Новий користувач

Логін:

Пароль:

Номер працівника:

Телефон:

Email:

Рисунок Г.2 - Форма реєстрації нового користувача

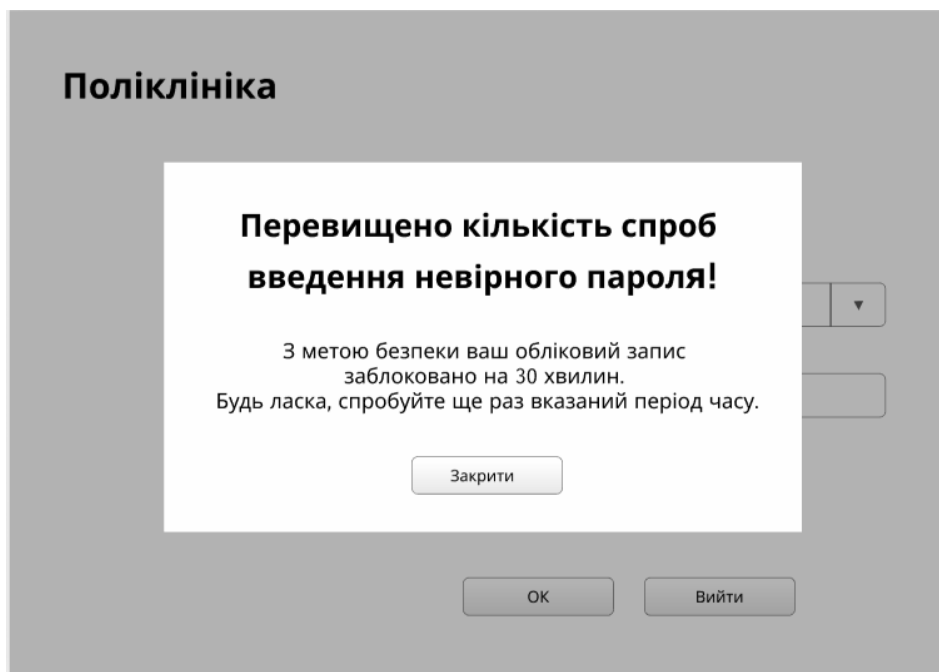


Рисунок Г.3 - Форма помилки перевищення некоректних спроб введення пароля користувача

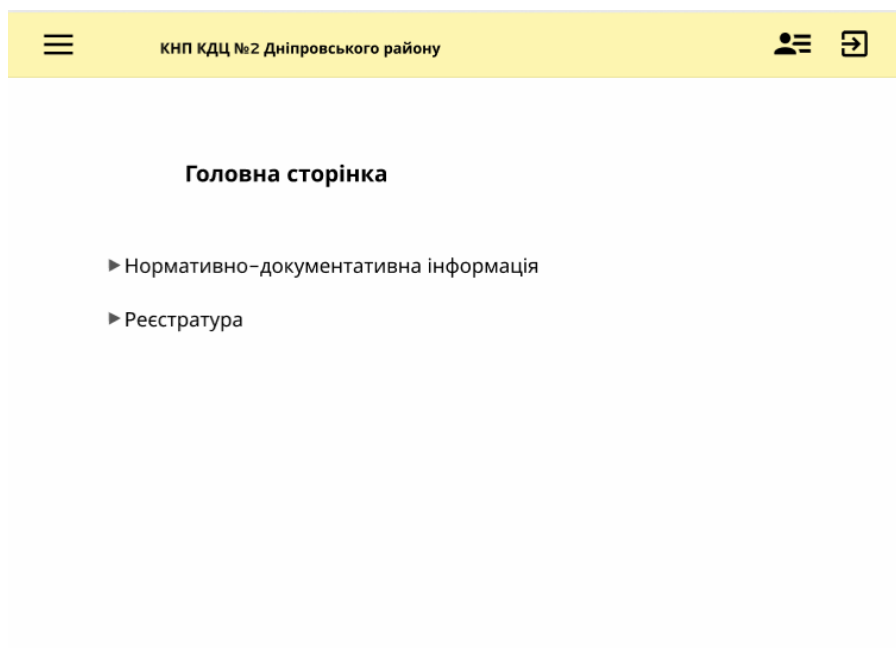


Рисунок Г.4 - Головна сторінка

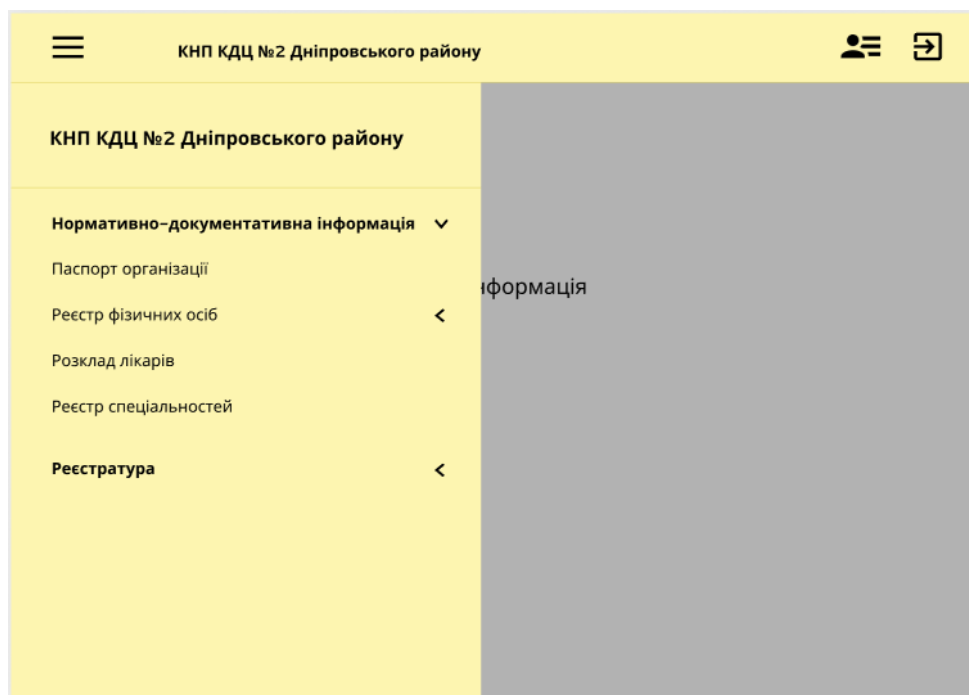


Рисунок Г.5 - Бокове меню

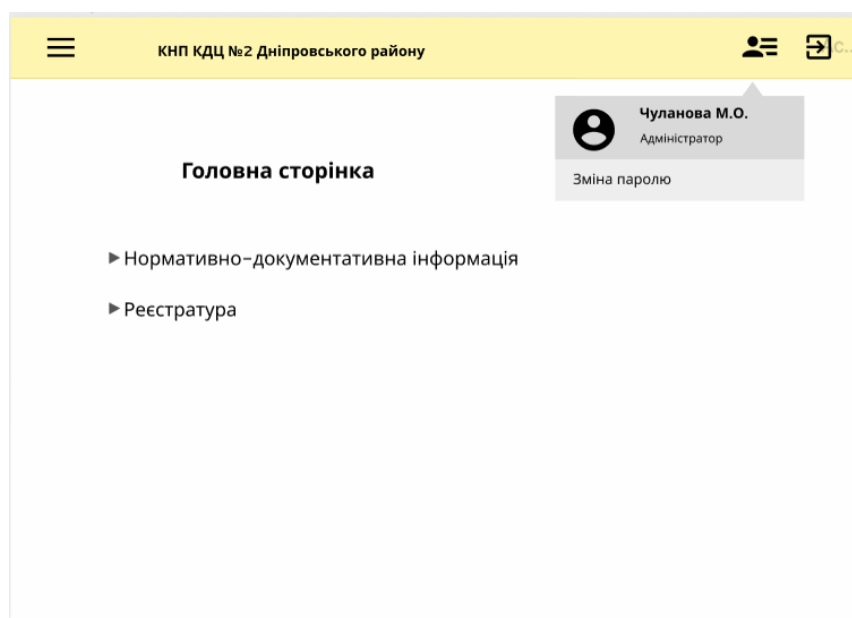
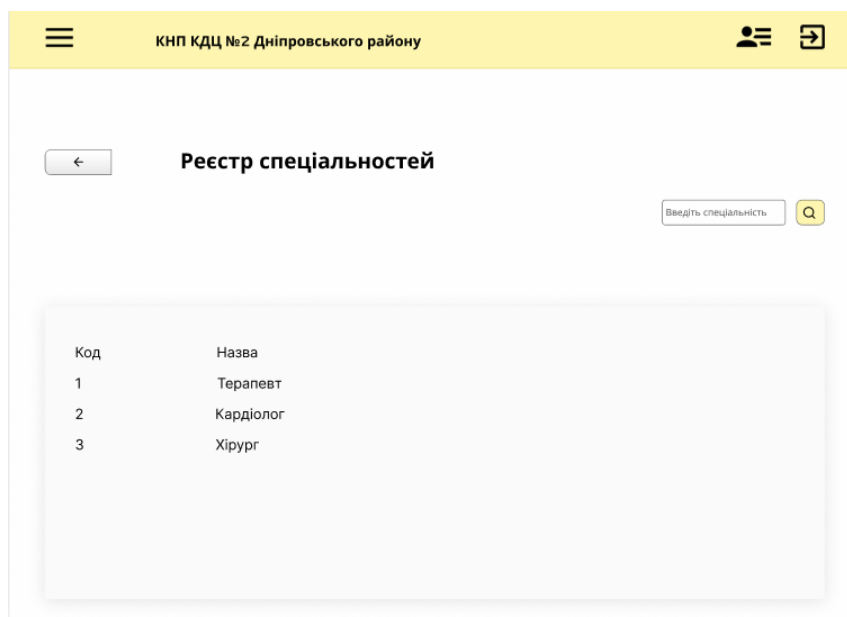
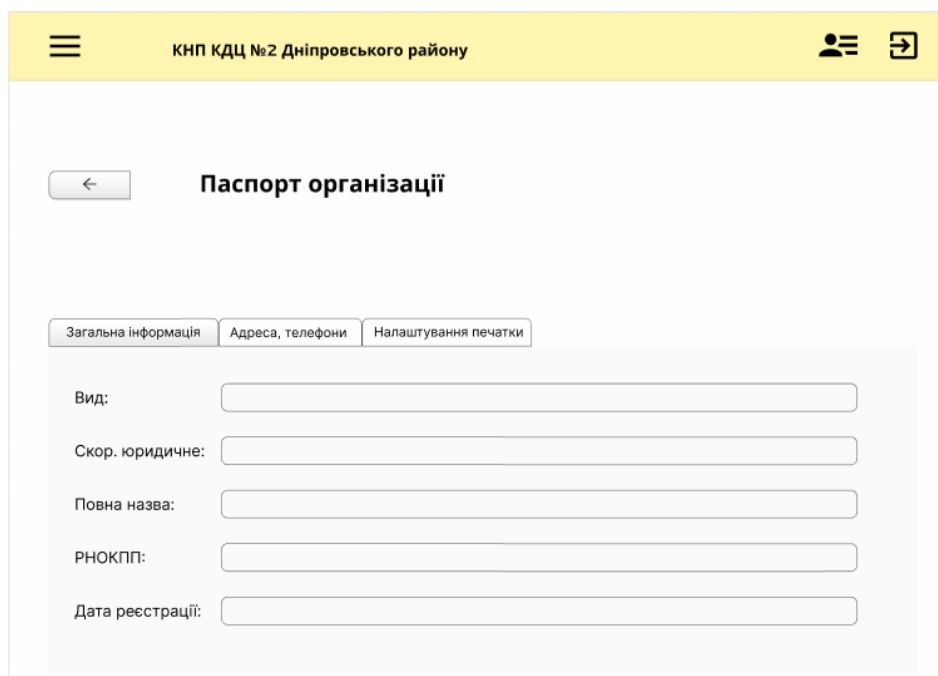


Рисунок Г.4 - Форма облікового запису користувача



Код	Назва
1	Терапевт
2	Кардіолог
3	Хірург

Рисунок Г.5 - Сторінка реєстру спеціальностей



Загальна інформація | Адреса, телефони | Налаштування печатки

Вид:

Скор. юридичне:

Повна назва:

РНОКПП:

Дата реєстрації:

Рисунок Г.6 - Сторінка паспорту організації

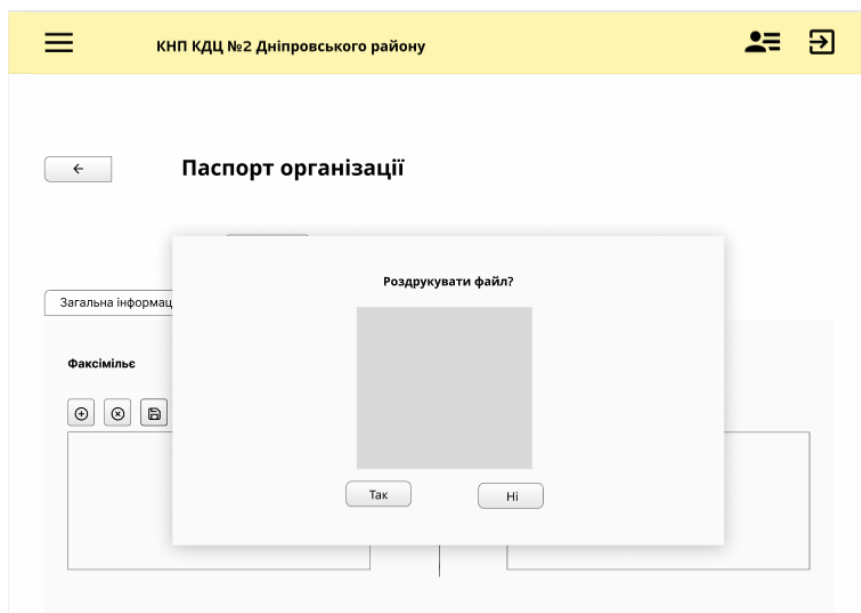


Рисунок Г.7 - Видача файлу на друк

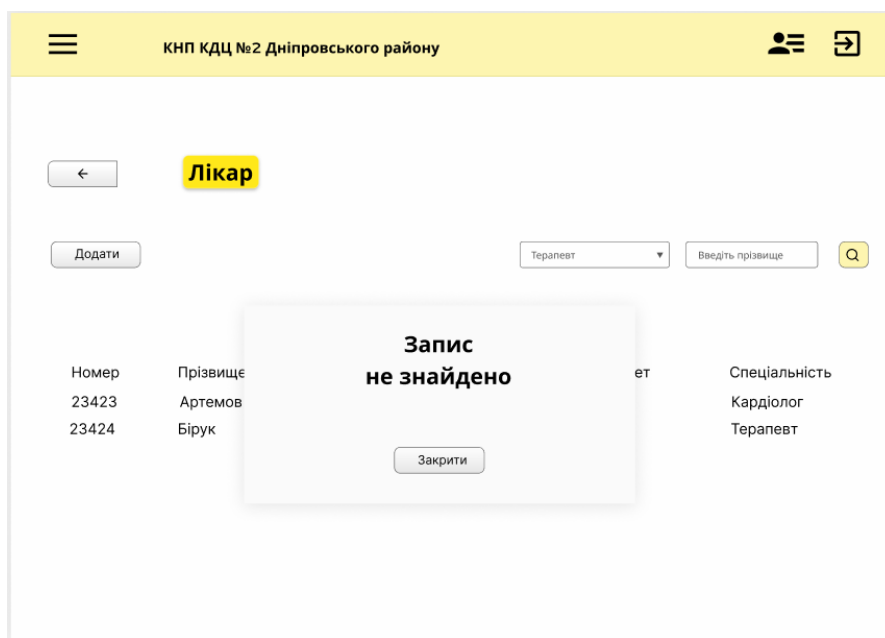


Рисунок Г.8 - Форма помилки відсутності даних, що запитуються

КНП КДЦ №2 Дніпровського району

← **Лікарі**

Новий лікар

Номер:

Прізвище:

Ім'я:

По-батькові:

Кабінет:

Спеціалізація:

Додати та закрити Закрити

Рисунок Г.9 - Форма створення нового лікаря

КНП КДЦ №2 Дніпровського району

← **Розклад**

Додати

Кардіолог x

Введіть прізвище

Лікар	Спеціальність	Номер кабінету	Дні прийому	Години
Артемов Максим Андрійович	Кардіолог	18	пн, пт	13:00

Рисунок Г.10 - Сторінка розкладу роботи лікарів

КНП КДЦ №2 Дніпровського району

Статистика

Внутрішня статистика | Формування звітів

Показники:

Відділення:

Період: 3 по

Результати запиту

Параметри не задані

Рисунок Г.11 - Сторінка внутрішньої статистики

КНП КДЦ №2 Дніпровського району

Реєстратура

- ▶ Реєстр пацієнтів
- ▶ Реєстр медичних карток
- ▶ Реєстр записів на прийом

Рисунок Г.12 - Сторінка реєстратури

КНП КДЦ №2 Дніпровського району

← **Реєстр записів на прийом**

Записати

Дата візиту: ДН:

Пацієнт:

Лікар:

Додатково

Відповідальний:

Предмет:

Талон успішно роздруковано!

ОК

Рисунок Г.13 Форма повідомлення про успішний друк талону запису на прийом

КНП КДЦ №2 Дніпровського району

← **Реєстр спеціальностей**

Терапевт

Номер	Прізвище	Ім'я	По-батькові	Кабінет
23424	Бірук	Наталія	Семенівна	32

Закрити

Рисунок Г.14 - Форма видачі лікарів за запитом певної спеціальності

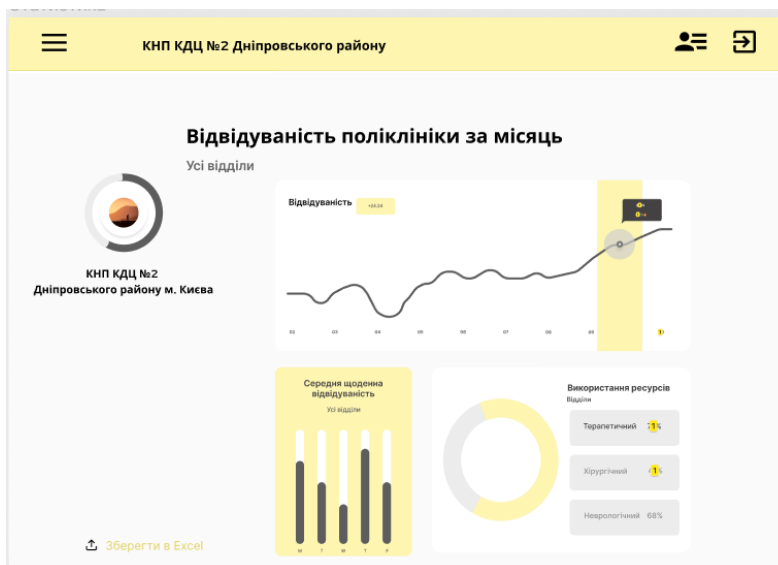


Рисунок Г.15 - Результати запиту відображення статистики роботи медичного закладу

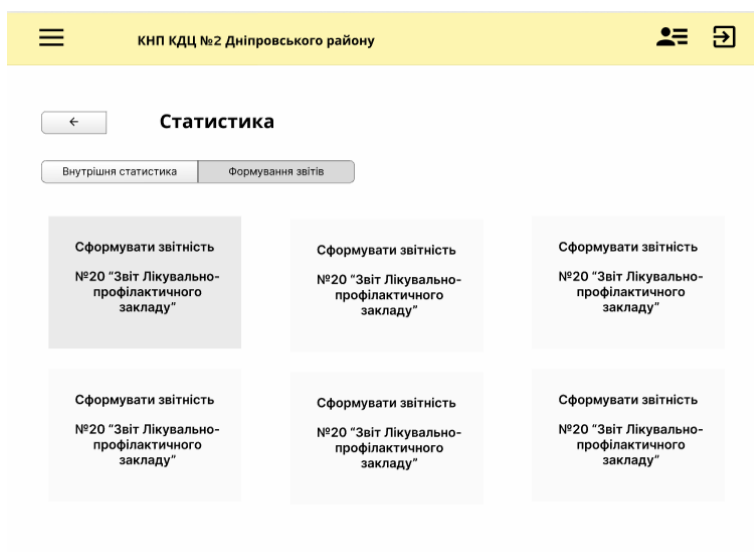


Рисунок Г.16 - Сторінка формування звітів

The screenshot shows a web application interface for a medical center. At the top, a yellow header bar contains a menu icon, the text 'КНП КДЦ №2 Дніпровського району', and user profile and navigation icons. Below the header, the main content area has a title 'Реєстр записів на прийом' with a back arrow. A 'Записати' (Record) button is positioned above the form fields. The form includes:

- 'Дата візиту:' (Visit date) with a date picker and 'Обрати дату:' (Select date) label.
- 'Час візиту:' (Visit time) with a time picker and 'Обрати час:' (Select time) label.
- 'Пацієнт:' (Patient) with a text input field.
- 'Лікар:' (Doctor) with a dropdown menu.
- A 'Додатково' (Additional) section containing:
 - 'Відповідальний:' (Responsible) with a dropdown menu.
 - 'Предмет:' (Subject) with a text input field.

Рисунок Г.17 - Форма створення нового запису пацієнта на прийом

The screenshot shows the 'Медична карта пацієнта № 71235' (Patient Medical Card) form. The yellow header bar is identical to the previous form. The title 'Медична карта пацієнта № 71235' is followed by a dropdown menu. The form fields include:

- 'ПІБ Пацієнта:' (Patient's name) with a text input field.
- 'Лікар:' (Doctor) with a dropdown menu.
- 'Група крові:' (Blood group) with a dropdown menu.
- 'Резус-фактор:' (Rh factor) with a dropdown menu.
- 'Група інвалідності:' (Disability group) with a dropdown menu showing 'Rh+' and 'Rh-' options.
- 'Дата створення:' (Creation date) with a date picker.

 At the bottom, there are two buttons: 'Змінити' (Change) on the left and 'Закрити' (Close) on the right.

Рисунок Г.19 - Форма створення нової медичної карти пацієнта

КНП КДЦ №2 Дніпровського району

Реєстр пацієнтів

Новий пацієнт

ПІБ пацієнта:

ІПН: ДН:

Телефон:

Адреса:

E-mail:

Додати документ Додати та закрити Закрити

Рисунок Г.20 - Форма створення нового пацієнта

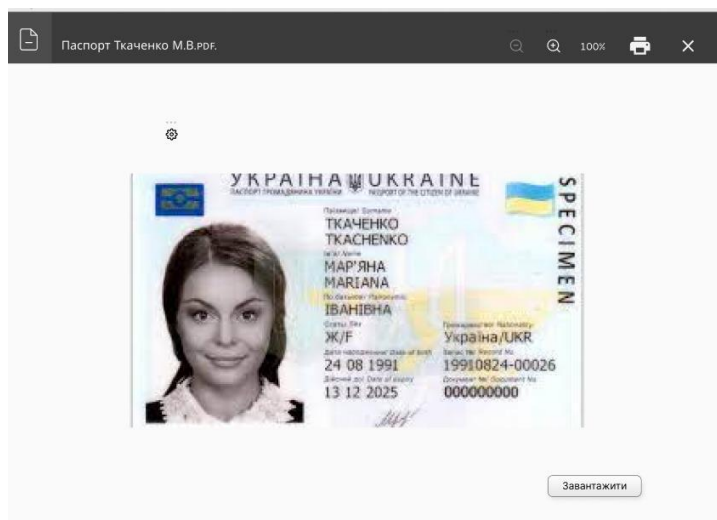


Рисунок Г.21 - Сторінка відображення переданих персональних документів пацієнтів

Номер	Пацієнт	Лікар	Група крові	Резус-фак
71235	Олексів Максим Петрович	Бірюк Наталія Семенівна	2	+

Рисунок Г.22 - Сторінка медичних карток

Номер	Прізвище
1	Чуланова

Рисунок Г.23 - Форма повідомлення про успішний друк талону запису на прийом

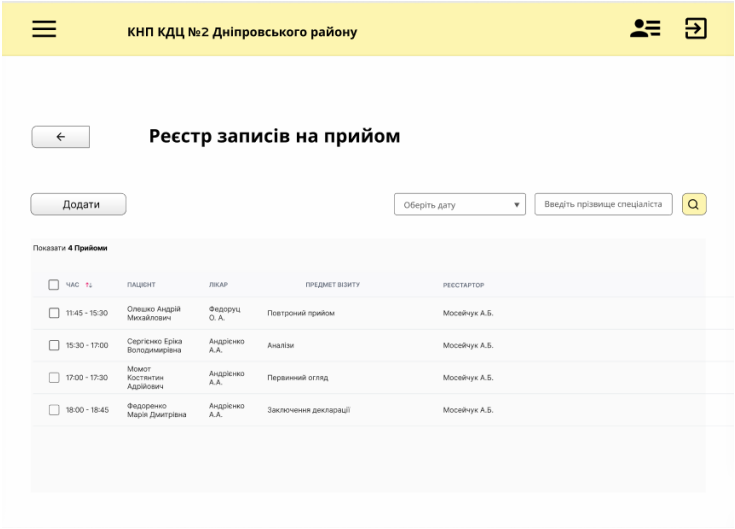


Рисунок Г.23 - Сторінка реєстру записів на прийом

ДОДАТОК І

Частина HTML та CSS коду

1.html

```

<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Поліклініка</title>
  <link rel="stylesheet" href="style1.css">
</head>
<body>
  <div class="container">
    <div class="header">
      <h1>Поліклініка</h1>
      <p>КНП КДЦ №2 Дніпровського району</p>
    </div>
    <div class="login">
      <label for="username">Логін:</label><input type="text" name="username" id="username" required>
      <label for="password">Пароль:</label>
      <input type="password" name="password" id="password">
      <button>+</button>
      <div class="buttons">
        <button>ОК</button>
        <button>Вийти</button>
      </div>
    </div>
  </div>
</body>
</html>

```

style1.css

```

body {
  font-family: sans-serif;
  margin: 0;
  padding: 0;
}

```

```
.container {
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  width: 100vw;
  height: 100vh;
}

.header {
  text-align: center;
  /* Increased margin-bottom for more space after header */
  margin-bottom: 30px;
}

.header p {
  font-size: 22px;
}

.login {
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: space-around;
  width: 500px;
  padding: 20px;
  border: 1px solid #ddd;
  border-radius: 4px;
  gap: 15px; /* Add spacing between child elements in .login */
}

.login label {
  display: block;
  margin-bottom: 10px; /* Increased margin-bottom for more space between label and input */
}
```

```
.login button {
  border: 1px solid #ddd;
  border-radius: 4px;
  cursor: pointer;
  padding: 7px 10px;
}

.login input,
.login select {
  width: 100%;
  padding: 10px;
  border: 1px solid #ddd;
  border-radius: 4px;
  margin-bottom: 15px; /* Increased margin-bottom for more space between inputs */
}

.buttons {
  display: flex;
  justify-content: space-between;
  margin: 20px 0; /* Increased margin top and bottom for more space around buttons */
  gap: 40px; /* Add spacing between buttons */
}

.buttons button {
  padding: 10px 20px;
  border: 1px solid #ddd;
  border-radius: 4px;
  cursor: pointer;
}

.buttons button:last-child { /* Target last button ("Вийти") */
  margin-left: 10px; /* Add margin to the left */
}

.buttons button:hover {
  background-color: #ddd;
}
```

```
}
```

новий працівник.html

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Поліклініка - Новий користувач</title>
  <link rel="stylesheet" href="newk.css">
</head>
<body>
  <div class="container">
    <h1>Поліклініка</h1>
    <h2>КНП КДЦ №2 Дніпровського району</h2>
    <h3>Новий користувач</h3>
    <form id="user-form">
      <div class="form-group">
        <label for="username">Логін:</label><input type="text" name="username" id="username" required>
      </div>
      <div class="form-group">
        <label for="password">Пароль:</label><input type="password" name="password" id="password"
required>
      </div>
      <div class="form-group">
        <label for="employee-number">Номер працівника:</label><input type="text" name="employee-
number" id="employee-number" required>
      </div>
      <div class="form-group">
        <label for="phone">Телефон:</label><input type="tel" name="phone" id="phone" required>
      </div>
      <div class="form-group">
        <label for="email">Email:</label><input type="email" name="email" id="email" required>
      </div>
      <div class="form-group">
        <button type="submit">Записати та закрити</button>
        <button type="button" onclick="clearForm()">Закрити</button>
      </div>
    </form>
  </div>
</body>
</html>
```

```
</div>
</form>
</div>
<script src="script.js"></script>
</body>
</html>
newk.css
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
}

.container {
    width: 400px;
    margin: 50px auto;
    padding: 30px;
    border: 1px solid #ccc;
    border-radius: 5px;
}

h1, h2, h3 {
    text-align: center;
}

h1 {
    font-size: 24px;
    margin-bottom: 10px;
}

h2 {
    font-size: 18px;
    margin-bottom: 15px;
}

h3 {
    font-size: 16px;
```

```
    margin-bottom: 20px;
}
```

```
.form-group {
    margin-bottom: 15px;
    display: flex;
    align-items: center;
}
```

```
label {
    flex: 0 0 100px;
    margin-right: 10px;
}
```

```
input, button {
    width: 100%;
    padding: 10px;
    border: 1px solid #ccc;
    border-radius: 3px;
    font-size: 16px;
}
```

```
button[type="submit"] {
    background-color: #ccc;
    color: black;
    margin-right: 10px;
}
```

```
button[type="submit"]:hover {
    background-color: #ddd;
}
```

```
button[type="button"] {
    background-color: #ccc;
    color: black;
}
```

```
button[type="button"]:hover {
    background-color: #ddd;
}
```

лікарі.html

```
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>КНП КДЦ №2 Дніпровського району</title>
    <link rel="stylesheet" href="лікарі.css">
</head>
<body>
    <header>
        <div class="logo">
            </div>
        <div class="header-text">
            <h1>КНП КДЦ №2 Дніпровського району</h1>
        </div>
        <ul class="account-buttons">
            <li><a href="#">Меню</a></li>
            <li><a href="#">Обліковий запис</a></li>
            <li><a href="#">Вихід</a></li>
        </ul>
    </header>
    <main>
        <h2 class="header-text">Лікарі</h2>
        <div class="search-bar">
            <select class="specialties">
                <option value="">Спеціальність</option>
                <option value="терапевт">Терапевт</option>
                <option value="хірург">Хірург</option>
            </select>
            <input type="text" placeholder="Введіть прізвище">
            <button type="button">Q</button>
        </div>
```

```

        <div class="add-doctor"><button type="button">Додати</button>
    </div>
</div>
<div class="table-container">
    <table>
        <thead>
            <tr>
                <th>Номер</th>
                <th>Прізвище</th>
                <th>Ім'я</th>
                <th>По-батькові</th>
                <th>Кабінет</th>
                <th>Спеціальність</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>23424</td>
                <td>Бірук</td>
                <td>Наталія</td>
                <td>Семенівна</td>
                <td>32</td>
                <td>Терапевт</td>
            </tr>
        </tbody>
    </table>
</div>
</main>
<script src="script.js"></script>
</body>
</html>

```

лікарі.css

```

body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background-color: #fff;}

```

```
.search-bar {  
  display: flex;  
  align-items: center;  
  margin-top: 50px;  
}
```

```
.search-bar button {  
  background-color: #ffd700;  
  color: black;  
  padding: 5px 10px;  
  border-radius: 5px;  
  border-color: #ffd700;  
  margin-left: 5px;  
}
```

```
.add-doctor button {  
  background-color: #ffd700;  
  color: black;  
  padding: 5px 10px;  
  border-radius: 5px;  
  border-color: #ffd700;  
  margin-left: 55px;  
}
```

```
.search-bar input {  
  padding: 5px 10px;  
  border: 1px solid #ccc;  
  border-radius: 5px;  
  width: 200px;  
}
```

```
.specialties {  
  padding: 5px 10px;  
  border: 1px solid #ccc;  
  border-radius: 5px;  
  margin-right: 10px;
```

```
margin-left: 800px;  
}
```

```
.admin {  
padding: 5px 10px;  
margin-right: 10px;  
margin-left: 650px;  
}
```

```
table {  
border-collapse: collapse;  
border: 1px solid #F9F9F9;  
background-color: #F9F9F9;  
width: 90%; /* Or a specific width value */  
margin-top: 50px;  
margin-left: 60px;  
}
```

```
th, td {  
padding: 10px; /* Adjust as needed */  
text-align: left; /* Or 'center' or 'right' */  
}
```

```
header {  
background-color: #ffd700;  
padding: 15px 20px;  
display: flex;  
justify-content: space-between;  
align-items: center;  
}
```

```
.logo {  
width: 100px;
```

```
    height: auto;
}
```

```
.header-text {
    flex: 1;
}
```

```
header h1 {
    margin: 0;
    font-size: 1.5rem;
    color: #000;
}
```

```
nav ul {
    margin: 0;
    padding: 0;
    list-style: none;
    display: flex;
}
```

```
nav li {
    margin-right: 20px;
}
```

```
nav a {
    text-decoration: none;
    color: #000;
    font-weight: bold;
}
```

```
.account-buttons {
    margin: 0;
    padding: 0;
    list-style: none;
    display: flex;
}
```

```
.account-buttons li {
    margin-left: 20px;
}
```

```
.account-buttons a {
    text-decoration: none;
    color: #000;
}
```

```
h2 {
    font-size: 20px;
    margin-bottom: 20px;
    text-align: center; /* Center the "Розклад" text horizontally */
    margin-top: 50px;
}
```

запис.html

```
<!DOCTYPE html>
```

```
<html lang="uk">
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
    <title>КНП КДЦ №2 Дніпровського району</title>
```

```
    <link rel="stylesheet" href="запис.css">
```

```
</head>
```

```
<body>
```

```
    <header>
```

```
        <div class="logo">
```

```
        </div>
```

```
        <div class="header-text">
```

```
            <h1>КНП КДЦ №2 Дніпровського району</h1>
```

```
        </div>
```

```
        <ul class="account-buttons">
```

```
            <li><a href="#">Меню</a></li>
```

```
            <li><a href="#">Обліковий запис</a></li>
```

```
            <li><a href="#">Вихід</a></li>
```

```
        </ul>
```

```

</header>
<main>
  <div class="container">
    <h2>Реєстр записів на прийом</h2>
    <form id="appointment-form">
      <div class="form-group">
        <label for="visit-date">Дата візиту:</label>
        <input type="date" name="data_reestr" id="data_reestr">
      </div>
      <div class="form-group">
        <label for="visit-time">Час візиту:</label>
        <select id="visit-time" name="visit-time">
          <option value="">Обрати час</option>
        </select>
      </div>
      <div class="form-group">
        <label for="patient">Пацієнт:</label>
        <input type="text" id="patient" name="patient" placeholder="Ім'я та прізвище пацієнта">
      </div>
      <div class="form-group">
        <label for="doctor">Лікар:</label>
        <select id="doctor" name="doctor">
          <option value="">Обрати лікаря</option>
        </select>
      </div>
      <div class="form-group1">
        <div class="form-group">
          <h3>Додатково</h3>
        </div>
        <div class="form-group">
          <label for="responsible">Відповідальний:</label>
          <select id="responsible" name="responsible">
            <option value="">Обрати відповідального</option>
          </select>
        </div>
        <div class="form-group">

```

```

        <label for="subject">Предмет:</label>
        <input type="text" id="subject" name="subject" placeholder="Тема консультації">
    </div>
</div>
    <button type="submit">Записати</button>
</form>
</div>
</main>
<script src="script.js"></script>
</body>
</html>

body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background-color: #fff;
}

.container {
    max-width: 1000px;
    margin: 0 auto;
    padding: 20px;
}

h2 {
    text-align: center;
    margin-bottom: 30px;
}

.form-group {
    margin-bottom: 15px;
    display: flex;
}

.form-group1 {
    background-color: #F9F9F9;
}

```

```
label {
  display: block;
  margin-bottom: 5px;
  flex: 0 0 20%; /* Set labels to use 20% of available space */
}

select, input[type="text"], input[type="date"] {
  flex: 1; /* Make the fields flexible and share remaining space */
  padding: 10px;
  border: 1px solid #ccc;
  border-radius: 3px;
}

button {
  background-color: #ffc107;
  color: #000;
  padding: 10px 20px;
  border: none;
  border-radius: 3px;
  cursor: pointer;
  margin-top: 10px;
}

h3 {
  margin-top: 20px;
}

header {
  background-color: #ffd700;
  padding: 15px 20px;
  display: flex;
  justify-content: space-between;
  align-items: center;
}
```

```
.logo {  
  width: 100px;  
  height: auto;  
}
```

```
.header-text {  
  flex: 1;  
}
```

```
header h1 {  
  margin: 0;  
  font-size: 1.5rem;  
  color: #000;  
}
```

```
nav ul {  
  margin: 0;  
  padding: 0;  
  list-style: none;  
  display: flex;  
}
```

```
nav li {  
  margin-right: 20px;  
}
```

```
nav a {  
  text-decoration: none;  
  color: #000;  
  font-weight: bold;  
}
```

```
.account-buttons {  
  margin: 0;  
  padding: 0;  
  list-style: none;  
  display: flex;
```

```
}
```

```
.account-buttons li {
  margin-left: 20px;
}
```

```
.account-buttons a {
  text-decoration: none;
  color: #000;
}
```

Stat.html

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>КНП КДЦ №2 Дніпровського району</title>
  <link rel="stylesheet" href="stat.css">
</head>
<body>
  <header>
    <div class="logo">
      </div>
    <div class="header-text">
      <h1>КНП КДЦ №2 Дніпровського району</h1>
    </div>
    <ul class="account-buttons">
      <li><a href="#">Меню</a></li>
      <li><a href="#">Обліковий запис</a></li>
      <li><a href="#">Вихід</a></li>
    </ul>
  </header>
  <main>
    <div class="header2-container">
      <h3 class="header-text">Статистика</h3>
    </div>
```

```

<main>
  <div class="stats-container">
    <button class="stats-btn">Внутрішня статистика</button>
    <button class="report-btn">Формування звітів</button>
  </div>
  <div class="inputs">
    <div class="input-group">
      <label for="indicator">Показники</label>
      <select name="indicator" id="indicator">
        <option value="">Оберіть дані</option>
      </select>
    </div>
    <div class="input-group">
      <label for="department">Відділення</label>
      <select name="department" id="department">
        <option value="">Оберіть відділення</option>
      </select>
    </div>
    <div class="date-range">
      <label for="start-date">Період</label>
      <input type="date" id="start-date">
      <span>по</span>
      <input type="date" id="end-date">
    </div>
  </div>
  <div class="button-container">
    <button class="display-btn">Відобразити результати</button>
    <button class="clear-btn">Очистити</button>
  </div>
  <div class="results">
    <div class="results-header">Результати запиту</div>
    <div class="warning">Параметри не задані</div>
  </div>
</main>
</body>
</html>
Stat.css

```

```
body {  
    margin: 0;  
    padding: 0;  
    font-family: Arial, sans-serif; /* Specify a professional font */  
    color: #333; /* Set a base text color for better readability */  
    text-align: justify;  
}  
  
h3 {  
    font-size: 20px;  
    margin-bottom: 20px;  
    text-align: center; /* Center the "Розклад" text horizontally */  
}  
  
.container {  
    max-width: 1500px; /* Increase max-width for wider content areas */  
    margin: 50px;  
    padding: 0 20px;  
}  
  
.main table {  
    width: 100%; /* Set the table width to 100% of its container */  
    margin-top: 30px;  
}  
  
h1 {  
    font-size: 24px;  
    margin: 0;  
    color: #fff; /* White text for better contrast on yellow background */  
}  
  
.logo {  
    width: 150px; /* Adjust logo size as needed */  
    margin-right: 20px;  
}
```

```
.search-bar {  
  display: flex;  
  align-items: center;  
}  
  
.search-bar select {  
  padding: 1px 15px; /* Adjust padding for a better fit */  
  border: 1px solid #ddd;  
  border-radius: 3px;  
  margin-right: 10px;  
  margin-left: 650px;  
  font-size: 16px; /* Adjust font size for consistency */  
}  
  
.logo {  
  width: 100px;  
  height: auto;  
}  
  
.header-text {  
  flex: 1;  
}  
  
header h1 {  
  margin: 0;  
  font-size: 1.5rem;  
  color: #000;  
}  
  
nav ul {  
  margin: 0;  
  padding: 0;  
  list-style: none;  
  display: flex;  
}
```

```
nav li {  
    margin-right: 20px;  
}
```

```
nav a {  
    text-decoration: none;  
    color: #000;  
    font-weight: bold;  
}
```

```
.account-buttons {  
    margin: 0;  
    padding: 0;  
    list-style: none;  
    display: flex;  
}
```

```
.account-buttons li {  
    margin-left: 20px;  
}
```

```
.account-buttons a {  
    text-decoration: none;  
    color: #000;  
}
```

```
header {  
    background-color: #ffd700;  
    padding: 15px 20px;  
    display: flex;  
    justify-content: space-between;  
    align-items: center;  
}
```

```
main {  
    background-color: #fff; /* Pure white */  
    padding: 20px;
```

```

}

.stats-container {
  display: flex;

  margin-bottom: 20px;
}

.stats-btn {
  background: #C8C8C8;
  border: 1px solid #ddd; /* Subtle border */
  margin-right: 5px;
  padding: 10px 20px;
  border-radius: 5px;
  cursor: pointer;
}

.report-btn {
  background-image: linear-gradient(to top, #b8b8b8, #bebdb8, #c3c3c4, #c9c8c9, #cfcecf, #d4d3d4, #d8d8d8,
#d4d3d4, #e2e2e2, #e7e7e7, #ececec, #f1f1f1);
  border: 1px solid #ddd; /* Subtle border */
  margin-right: 5px;
  padding: 10px 20px;
  border-radius: 5px;
  cursor: pointer;
}

.inputs {
  display: flex;
  flex-wrap: wrap;
  margin-bottom: 20px;
}

.input-group {
  margin-right: 20px;
  margin-bottom: 10px;
  width: 48%; /* Slightly wider input fields */

```

```

}

label {
  display: block;
  margin-bottom: 5px;
  font-weight: bold;
  margin-right: 10px;
}

select,
input[type="date"] {
  width: 100%;
  padding: 10px;
  border: 1px solid #ccc;
  border-radius: 5px;
}

.date-range {
  display: flex;
  align-items: center;
  width: 48%; /* Slightly wider date range */
}

.date-range span {
  margin: 0 10px;
}

.button-container {
  display: flex;
  justify-content: space-between;
  margin-bottom: 20px;
}

.display-btn,
.clear-btn {
  background-color: #ffd700; /* Grey for primary action */
  color: #0A0A0A; /* Black text for better contrast */
}

```

```
padding: 10px 20px;
border: none; /* Removed border for cleaner look */
border-radius: 5px;
cursor: pointer;
}

.results {
background-color: #f5f5f5; /* Light gray for results section */
padding: 15px;
border-radius: 5px;
height: 150px;
}

.results-header {
font-weight: bold;
margin-bottom: 10px;
font-size: 18px;
}

.warning {
color: #5A5A5A; /* Dark grey for warning message */
font-weight: bold;
margin-top: 50px;
margin-left: 460px;
}

button {
background-color: #FFD700;
color: #000;
padding: 10px 20px;
border: none;
cursor: pointer;
}
```

Stat2.html

```
<!DOCTYPE html>
<html lang="uk">
```

```

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>КНП КДЦ №2 Дніпровського району</title>
  <link rel="stylesheet" href="stat2.css">
</head>
<body>
  <header>
    <div class="logo">
      </div>
    <div class="header-text">
      <h1>КНП КДЦ №2 Дніпровського району</h1>
    </div>
    <ul class="account-buttons">
      <li><a href="#">Меню</a></li>
      <li><a href="#">Обліковий запис</a></li>
      <li><a href="#">Вихід</a></li>
    </ul>
  </header>
  <main>
    <div class="header2-container">
      <h3 class="header-text">Статистика</h3>
    </div>
    <main>
      <div class="stats-container">
        <button class="stats-btn">Внутрішня статистика</button>
        <button class="report-btn">Формування звітів</button>
      </div>
    <div class="reports-section">
      <div class="report">
        <h4>№20 "Звіт Лікувально-профілактичного закладу"</h4>
        <p style="text-align:center">
          <a href="#">Сформувати звітність</a>
        </p>
      </div>
      <div class="report">
        <h4>№20 "Звіт Лікувально-профілактичного закладу"</h4>

```

```

        <p style="text-align:center">
        <a href="#">Сформувати звітність</a>
        </p>
    </div>
    <div class="report">
        <h4>№20 "Звіт Лікувально-профілактичного закладу"</h4>
        <p style="text-align:center">
        <a href="#">Сформувати звітність</a>
        </p>
    </div>
    <div class="report">
        <h4>№20 "Звіт Лікувально-профілактичного закладу"</h4>
        <p style="text-align:center">
        <a href="#">Сформувати звітність</a>
        </p>
    </div>
</div>
</div>
</main>
</body>
</html>

.logo {
    width: 100px;
    height: auto;
}

Stat2.css

.header-text {
    flex: 1;
}

header h1 {
    margin: 0;
    font-size: 1.5rem;
    color: #000;
}

nav ul {

```

```
margin: 0;
padding: 0;
list-style: none;
display: flex;
}

nav li {
margin-right: 20px;
}

nav a {
text-decoration: none;
color: #000;
font-weight: bold;
}

.account-buttons {
margin: 0;
padding: 0;
list-style: none;
display: flex;
}

.account-buttons li {
margin-left: 20px;
}

.account-buttons a {
text-decoration: none;
color: #000;
}

header {
background-color: #ffd700;
padding: 15px 20px;
display: flex;
justify-content: space-between;
align-items: center;
```

```

}
body {
  margin: 0;
  padding: 0;
  font-family: Arial, sans-serif; /* Specify a professional font */
  color: #333; /* Set a base text color for better readability */
  text-align: justify;
}
.stats-container {
  display: flex;
  margin-left: 20px;
  margin-bottom: 20px;
}

.report-btn {
  background: #C8C8C8;
  border: 1px solid #ddd; /* Subtle border */
  margin-right: 5px;
  padding: 10px 20px;
  border-radius: 5px;
  cursor: pointer;
}

.stats-btn {
  background-image: linear-gradient(to top, #b8b8b8, #bebdb8, #c3c3c4, #c9c8c9, #cfcecf, #d4d3d4, #d8d8d8,
  #d4d3d4, #e2e2e2, #e7e7e7, #ecec, #f1f1f1);
  border: 1px solid #ddd; /* Subtle border */
  margin-right: 5px;
  padding: 10px 20px;
  border-radius: 5px;
  cursor: pointer;
}

.reports-section {
  display: flex;
  justify-content: space-between;

```

```
align-items: center;
margin-bottom: 20px;
margin-right: 20px;
}

.report {
  margin-bottom: 10px;
  margin-left: 20px;
  background-color: #ddd;
  padding: 10px;
  border-radius: 5px;
  height: 200px;
  align-items: center; /* Added property for vertical centering */
  justify-content: center;
}

.report h4 {
  margin: 0;
  text-align: center;
}

.report a {
  text-decoration: none;
  color: #000;
  margin-top: 5px;
}
```

ДОДАТОК Д

Тест-кейс

TESTING	Test Report
Basic Path Scenario test. Запис на прийом до недоступного лікаря Description Користувач намагається записатися на прийом до недоступного лікаря. Input Користувач обирає лікаря. Користувач обирає дату та час прийому. Користувач вписує ПІБ пацієнта, якого записує на прийом. Acceptance Criteria Користувач повинен отримати повідомлення про помилку, яке вказує на недоступність обраного лікаря. Results Система сповіщає користувача про те, що обраний лікар недоступний на обрану дату та час. [Last run Fail at Invalid DateTime by MilenaChulanova and checked by MilenaChulanova.]	Alternate Scenario test. Скасування запису на прийом Description Користувач скасовує попередню створену зустріч. Input Користувач ініціює процес скасування заброньованої зустрічі Acceptance Criteria Перш ніж продовжити, система повинна підтвердити дію скасування. Після скасування зустріч має бути видалено із системи, а всі пов'язані слоти мають бути знову доступними. Results Попередньо створений запис успішно скасовано, на екрані з'явиться повідомлення про підтвердження, а пов'язані слоти знову стануть доступними для запису. 4/7/2024 - Cancelled Run by: MilenaChulanova Checked by: MilenaChulanova 4/7/2024 - Cancelled Run by: MilenaChulanova Checked by: MilenaChulanova [Last run Cancelled at 4/7/2024 by MilenaChulanova and checked by MilenaChulanova.]
Basic Path Scenario test. Запис на прийом не має необхідної інформації Description Користувач намагається записатися на прийом без надання необхідної інформації. Input Користувач обирає лікаря. Користувач обирає дату та час прийому. Користувач вписує ПІБ пацієнта, якого записує на прийом. Acceptance Criteria Система має запропонувати користувачеві заповнити усю необхідну інформацію перед підтвердженням бронювання зустрічі. Користувач не зможе продовжити бронювання, доки не буде надано всю необхідну інформацію. Results Система застерігає користувача про нестачі необхідної інформації для створення запису. 4/7/2024 - Fail Run by: MilenaChulanova Checked by: MilenaChulanova 4/7/2024 - Fail Run by: MilenaChulanova Checked by: MilenaChulanova	Basic Path Scenario test. Успішний запис на прийом Description Пацієнта успішно записано на прийом до лікаря. Input Користувач обирає лікаря. Користувач обирає дату та час прийому. Користувач вписує ПІБ пацієнта, якого записує на прийом. Acceptance Criteria Система має відобразити доступні дати та час для обраного лікаря. Користувач повинен мати можливість підтвердити запис на зустріч без помилок. Results Запис успішно створений, на екрані відображається повідомлення про підтвердження створення запису.

Рисунок Д.1 - Тестовий випадок "Запис на прийом"

ДОДАТОК Е

ЕЛЕКТРОННА РЕЕСТРАТУРА

Ідентифікатор
призначення

5455766447709

Талон

на прийом до лікаря

Медичний заклад:

Назва медичного закладу

Адреса:

Адреса медичного закладу

Номер талону:

№ 45643

Пацієнт:

ПІБ пацієнта

Дата та час:

П'ятниця, 03.11.2023, 14:30

Спеціалізація:

Терапевт

Лікар:

Бірук Наталя Семенівна

Кабінет:

32

Рисунок Е.1 - Форма талону на прийом

ДОДАТОК Є

Діаграми послідовності

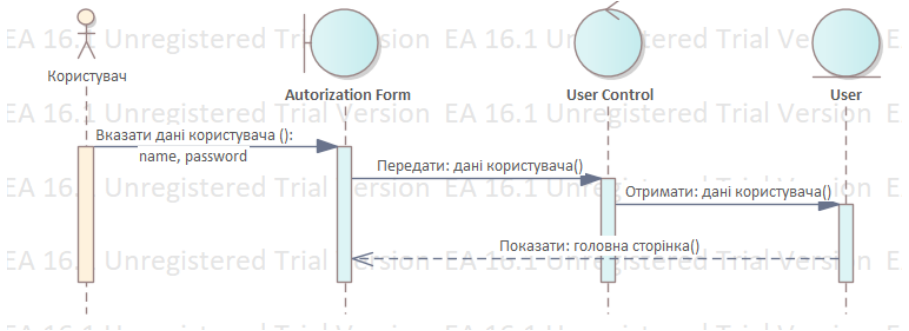


Рисунок Є.1 - Діаграма послідовності для процесу авторизації

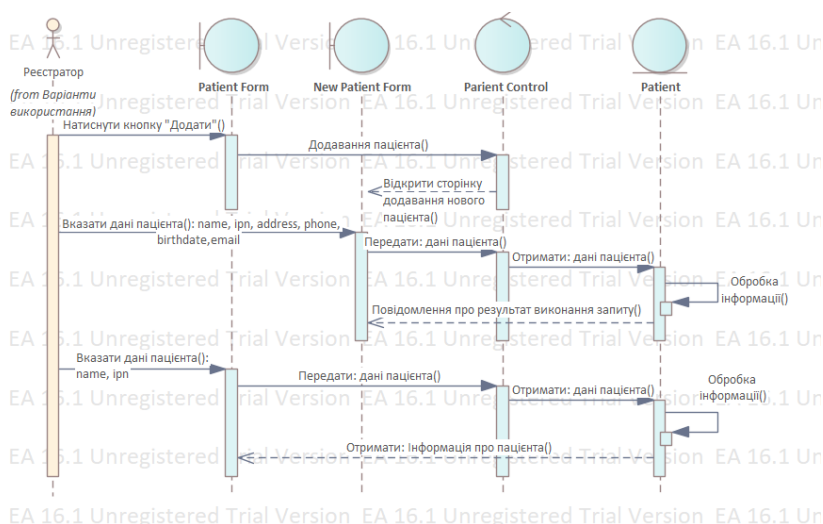


Рисунок Є.2 - Діаграма послідовності для процесу обліку пацієнтів

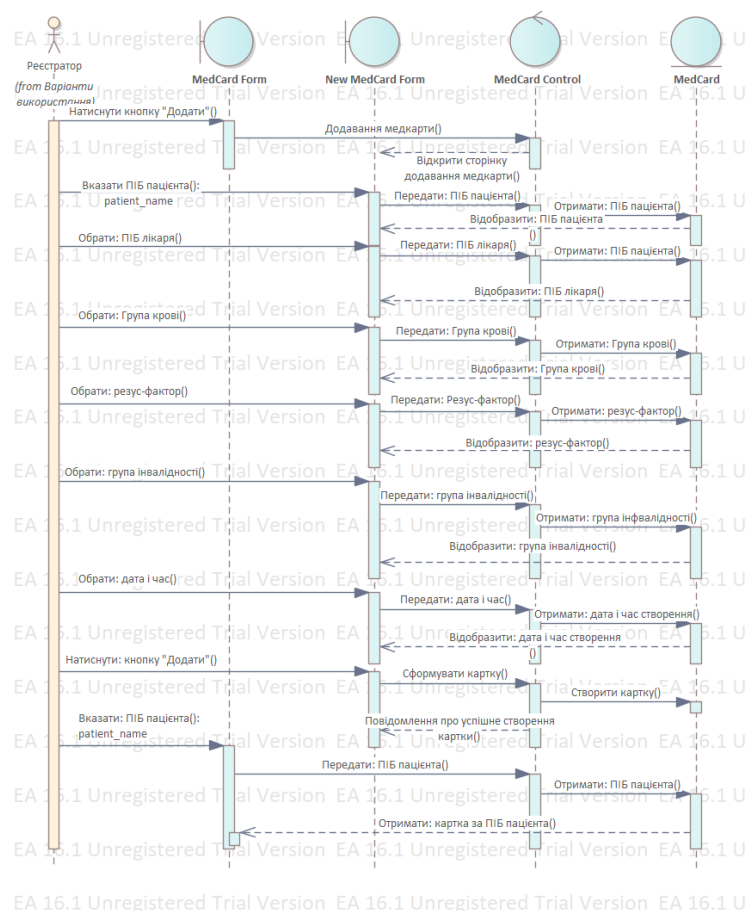


Рисунок Є.3 - Діаграма послідовності для процесу обліку медичних карток

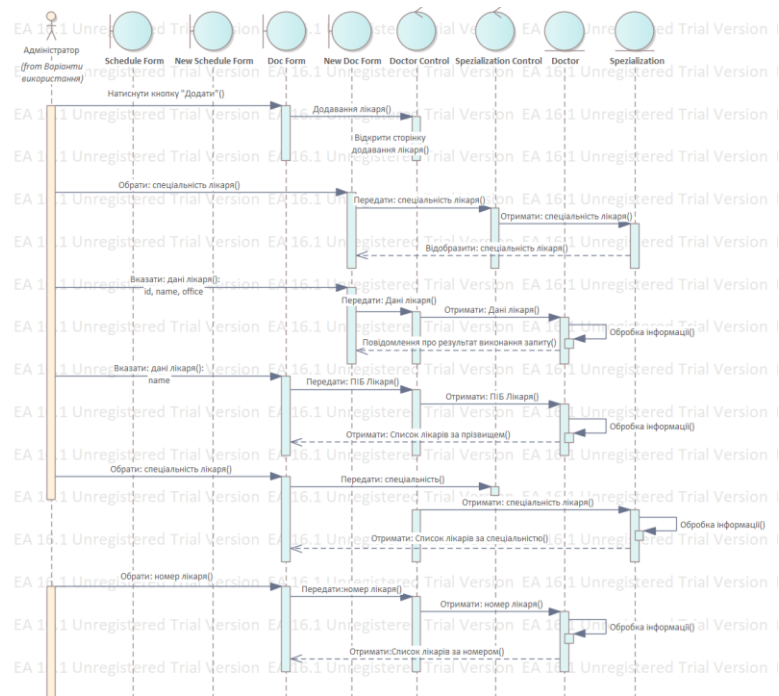


Рисунок Є.4 - Діаграма послідовності для процесу обліку лікарів



Ім'я користувача: Інформаційних систем в економіці Шкуратовська Те... ID перевірки: 1016341611
 Дата перевірки: 10.06.2024 22:24:57 EEST Тип перевірки: Doc vs Internet + Library
 Дата звіту: 10.06.2024 22:29:24 EEST ID користувача: 100005745

Назва документа: Чуланова_M_IH-402

Кількість сторінок: 125 Кількість слів: 16030 Кількість символів: 132836 Розмір файлу: 6.72 MB ID файлу: 1016142857

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

6.31%
Схожість

Найбільша схожість: 1.65% з джерелом з Бібліотеки (ID файлу: 1008404602)

3.88% Джерела з Інтернету 374 Сторінка 127

5.34% Джерела з Бібліотеки 442 Сторінка 130

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 69

Підозріле форматування 21 сторінка

Назва документа: Чуланова_M_IN-402

ID файлу:
1016142857

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА
Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»
галузь знань 12 «Інформаційні технології»
спеціальність 122 «Комп'ютерні науки»

Форма навчання: денна

КВАЛІФІКАЦІЙНИЙ БАКАЛАВРСЬКИЙ ПРОЕКТ

на тему

**РОЗРОБКА МЕДИЧНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ
АВТОМАТИЗАЦІЇ РОБОТИ РЕЄСТРАТУРИ**

здобувача Чуланової Мілени Олегівни

Науковий керівник:

ст. викладач

Сквордякова О. О.

**Проект допущений до захисту
перед екзаменаційною комісією з
атестації здобувачів вищої освіти**
завідувач кафедри:

к.е.н., доцент

Тішков Б.О.

Київ 2024