

БАКАЛАВРСЬКА РОБОТА

БР. ІІ - 04.00.00.000 ІІЗ

Група ІІ-21-4

Гоблянська Владислава

2025

Івано-Франківський національний технічний університет нафти і газу

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Гоблянська Владислава Володимирівна

(прізвище, ім'я, по батькові)

УДК 004.4
(індекс)

БАКАЛАВРСЬКА РОБОТА

Побудова Андроїд-аплікації адміністрування роботи медичного закладу

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Здобувач освітнього рівня Гоблянська В.В.
(підпис, ініціали та прізвище здобувача)

Науковий керівник Юрчишин Володимир Миколайович, д.т.н., проф.
(підпис, прізвище, ім'я, по батькові, науковий ступінь, вчене звання керівника)

Допущено до захисту
Завідувач кафедри

доц. Бандура В.В.
(посада) (підпис) (дата) (ініціали та прізвище)

Івано-Франківський національний технічний університет нафти і газу

Інститут, факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти бакалавр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедрою ІІЗ

доц.

В.В. Бандура

“ ” 2025 р.

ЗАВДАННЯ

НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТОВІ

Гоблянській Владиславі Володимирівні

(прізвище, ім'я, по-батькові)

1. Тема проекту (роботи) “ Побудова Андройд-аплікації адміністрування роботи медичного закладу ”

керівник проекту (роботи) Юрчишин В.М., професор, д.т.н.

затверджені наказом закладу вищої освіти від “ 28 ” квітня 2025 р. № 264/7

2. Строк подання студентом проекту (роботи) 10 червня 2025 р.

3. Вихідні дані до проекту (роботи) Результати і матеріали отримані під час проходження переддипломної практики

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

1. Аналіз предметної області використання ІТ управління медичними закладами

2. Програмні засоби, орієнтовані на інноватизацію процесів управління в медичних закладах

3. Алгоритмічна реалізація Android аплікації та вибір засобів розробки

4. Реалізація Android аплікації для медичного закладу

5. Розробка інтерфейсу користувача

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

1. Основні аспекти протоколу HL7 (рис. 1.1)

2. Основні типи повідомлень ADT (рис. 1.2)

3. Фрагмент повідомлення HL7 ADT (рис. 1.3)

4. Реляційна схема автоматизованої системи управління об'єктами (рис. 1.4)

5. Скріншот системи управління медичним закладом (рис. 1.5)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 28 квітня 2025 р.

Керівник

_____ (підпис)

Завдання прийняв до виконання

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту	Примітка
1	Аналіз предметної області використання ІТ управління медичними закладами	03.05.2025	виконано
2	Програмні засоби, орієнтовані на інноватизацію процесів управління в медичних закладах	14.05.2025	виконано
3	Алгоритмічна реалізація Android аплікації та вибір засобів розробки	22.05.2025	виконано
4	Реалізація Android аплікації для медичного закладу	28.05.2025	виконано
5	Розробка інтерфейсу користувача	04.06.2025	виконано
6	Оформлення пояснювальної записки дипломної роботи завідувачем кафедри	10.06.2025	виконано

Студент – дипломник

_____ (підпис)

Керівник роботи

_____ (підпис)

АНОТАЦІЯ

Бакалаврська робота містить 79 сторінок, 28 рисунків, список використаних джерел із 39 найменуваннями, 1 додаток.

Мета роботи - розробити функціональну Android-аплікацію, яка дозволяє ефективно адмініструвати діяльність медичного закладу для обміну даними та сучасних методів розробки мобільного програмного забезпечення.

Об'єкт дослідження - інформаційні системи управління медичними закладами.

Предмет дослідження - процеси розробки та впровадження мобільних додатків для адміністрування роботи медичних установ.

В першому розділі здійснено всебічний аналіз предметної області використання інформаційних систем для управління медичними закладами, що дозволяє окреслити ключові передумови для розробки Android-додатку

В другому розділі розглянуто ключові технічні аспекти створення Android-додатку для адміністрування роботи медичного закладу, починаючи зі структури Android-середовища й закінчуючи реалізацією програмних модулів

В третьому розділі представлено повну програмну реалізацію Android-додатку, призначеного для адміністрування роботи медичного закладу

Висновок: створено структуру Android-додатку, розроблено та описано ключові програмні компоненти, включно з класами, інтерфейсами та методами. Здійснено побудову користувацького інтерфейсу з урахуванням специфіки цільової аудиторії, що забезпечує зручність використання та високу інформативність.

КЛЮЧОВІ СЛОВА: ІНФОРМАЦІЙНІ СИСТЕМИ, МЕДИЧНИЙ ЗАКЛАД, ANDROID-ДОДАТОК, АДМІНІСТРУВАННЯ, HL7, REST, JSON, МОБІЛЬНА РОЗРОБКА.

ANNOTATION

The bachelor's thesis contains 79 pages, 28 figures, a list of used sources with 39 names, 1 appendix.

The purpose of the work is to develop a functional Android application that allows you to effectively administer the activities of a medical institution for data exchange and modern methods of developing mobile software.

The object of research is information systems for managing medical institutions.

The subject of research is the processes of developing and implementing mobile applications for administering the work of medical institutions.

The first section provides a comprehensive analysis of the subject area of using information systems for managing medical institutions, which allows us to outline the key prerequisites for developing an Android application

The second section examines the key technical aspects of creating an Android application for administering the work of a medical institution, starting with the structure of the Android environment and ending with the implementation of software modules

The third section presents a complete software implementation of an Android application designed for administering the work of a medical institution

Conclusion: the structure of the Android application has been created, key software components have been developed and described, including classes, interfaces and methods. The user interface has been built taking into account the specifics of the target audience, which ensures ease of use and high informativeness.

KEYWORDS: INFORMATION SYSTEMS, MEDICAL FACILITY, ANDROID APPLICATION, ADMINISTRATION, HL7, REST, JSON, MOBILE DEVELOPMENT.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	8
ВСТУП	9

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ УПРАВЛІННЯ МЕДИЧНИМИ ЗАКЛАДАМИ.....	12
1.1. Передумови розробки Android аплікації для медичного закладу. Протокол HL7.....	12
1.2. Мета проекту. Вимоги до мобільної аплікації.....	15
1.2.1. Перелік показників ефективності лікарні, що пропонується в мобільному додатку	17
1.3. Особливості ADT повідомлень в контексті розробки мобільного додатку	19
1.3.1. Типи повідомлень ADT	20
1.3.2. Структура повідомлення HL7 ADT	23
1.4. Використання інформаційних систем управління об'єктами для медичних закладів.....	24
1.5. Програмні засоби, орієнтовані на інноватизацію процесів управління в медичних закладах.....	36
1.5.1. Онлайн платформа Doctolib	38
1.5.2. Онлайн платформа Doctena.....	40
Висновки до розділу	43

РОЗДІЛ 2. АЛГОРИТМІЧНА РЕАЛІЗАЦІЯ ANDROID АПЛІКАЦІЇ ТА ВИБІР ЗАСОБІВ РОЗРОБКИ.....	44
---	----

					БР.ІП – 04.00.00.000 ПЗ								
Змн.	Арк.	№ докум.	Підпис	Дата	Побудова Андроїд-аплікації адміністрування роботи медичного закладу Пояснювальна записка				Літ.	Арк.	Акрушів		
Розроб.		Гоблянська В.										6	
Перевір.		Юрчишин В.М.											
Реценз.													
Н. Контр.		Піх М.М.											
Затверд.		Бандура В.В.											
					ІФНТУНГ ІП-21-4								

2.1. Опис структури Android.....	44
2.2. RESTful вебсервіс. Формат JSON	46
2.3. Визначення пакетів, використаних у проєкті	50
2.4. Реалізація Android аплікації для медичного закладу	51
2.4.1. Декодування тестових повідомлень HL7 ADT	53
2.4.2. Реалізація PHP для отримання даних.....	54
Висновки до розділу	56

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ANDROID-АПЛІКАЦІЇ

АДМІНІСТРУВАННЯ РОБОТИ МЕДИЧНОГО ЗАКЛАДУ	57
3.1. Реалізація структури Android-аплікації.....	57
3.1.1. Опис розроблених класів.....	58
3.1.2. Опис основних інтерфейсів та методів	59
3.2. Розробка інтерфейсу користувача.....	62
3.3. Опис процесу тестування мобільної аплікації.....	70
Висновки до розділу	73

ВИСНОВКИ.....	74
---------------	----

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	76
---------------------------------------	----

ДОДАТКИ

БІБЛІОГРАФІЧНА ДОВІДКА

					БР.ІП – 04.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

HITECH - Health Information Technology for Economic and Clinical Health

PPACA - Patient Protection and Affordable Care Act

EHR - Electronic Health Records

HL7 - Health Level 7

ORMs - Order Messages

ADT - Admission Discharge Transfer

HIPAA - Health Insurance Portability and Accountability Act

REST - Representational State Transfer

URIs - Uniform Resource Identifiers

JDBC - Java Database Connectivity

CAFM - Computer Aided Facility Management

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Інформаційні технології відіграють все вагомішу роль в організації ефективного управління у сфері охорони здоров'я. Забезпечення оперативного доступу до даних про діяльність медичних закладів, автоматизація адміністративних процесів, інтеграція з національними та міжнародними стандартами обміну медичними даними є ключовими чинниками підвищення якості медичного обслуговування.

У зв'язку з цим особливої актуальності набуває розробка спеціалізованих мобільних програмних засобів, що дозволяють адміністративному персоналу оперативно контролювати стан основних показників діяльності установи, приймати обґрунтовані управлінські рішення та взаємодіяти з інформаційними системами охорони здоров'я. Одним із ефективних рішень у цьому контексті є Android-додатки, які завдяки своїй гнучкості, доступності та функціональності є придатними для впровадження у медичній галузі.

У роботі здійснено аналіз передумов і вимог до створення мобільного застосунку для адміністрування медичного закладу, проведено проектування та реалізацію Android-додатку із застосуванням протоколу HL7 та REST-архітектури. Також проведено тестування функціональності розробленого рішення та оцінено його відповідність сучасним вимогам до цифровізації процесів управління в охороні здоров'я.

Актуальність роботи

Сучасні виклики в сфері охорони здоров'я вимагають оперативної цифрової трансформації процесів управління медичними установами. Зростаючі обсяги адміністративної інформації, потреба в інтеграції з електронними медичними системами та вимоги до підвищення ефективності управлінських рішень потребують розробки інноваційних ІТ-рішень. Зокрема, мобільні застосунки для платформ Android забезпечують зручний

					БР.ІП – 04.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

інтерфейс, швидкий доступ до ключових показників діяльності закладу та можливість інтеграції з галузевими інформаційними стандартами, такими як HL7. Отже, дослідження, пов'язані з розробкою таких додатків, є надзвичайно актуальними в контексті модернізації медичної сфери.

У сучасних умовах цифрової трансформації системи охорони здоров'я дедалі більшої ваги набуває впровадження інформаційних технологій в управління медичними закладами. Ефективне адміністрування ресурсів, моніторинг ключових показників діяльності лікарень та взаємодія між медичним персоналом і адміністрацією потребують інтегрованих цифрових рішень. Створення мобільної Android-аплікації, яка використовує протоколи обміну медичними даними (зокрема HL7 ADT), сприяє автоматизації управлінських процесів, забезпечує оперативний доступ до даних та покращує якість прийняття управлінських рішень. Актуальність підсилюється потребою у гнучких програмних засобах, адаптованих до національної системи охорони здоров'я та сумісних із міжнародними стандартами.

Мета роботи - розробити функціональну Android-аплікацію, яка дозволяє ефективно адмініструвати діяльність медичного закладу для обміну даними та сучасних методів розробки мобільного програмного забезпечення.

Завдання дослідження

1. Проаналізувати специфіку використання інформаційних технологій в управлінні медичними установами.
2. Вивчити протокол HL7, зокрема повідомлення типу ADT, та визначити їхню роль у передачі медичних даних.
3. Обґрунтувати вибір інструментів для створення Android-додатка.
4. Реалізувати серверну та клієнтську частини додатку з підтримкою RESTful-інтерфейсу та обробкою HL7-повідомлень.
5. Розробити інтерфейс користувача та протестувати працездатність мобільного додатку.

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

Об’єкт дослідження - інформаційні системи управління медичними закладами.

Предмет дослідження - процеси розробки та впровадження мобільних додатків для адміністрування роботи медичних установ.

Методи дослідження

- Аналіз наявних IT-рішень в охороні здоров’я (Doctolib, Doctena).
- Моделювання структури мобільного додатку та інформаційного обміну.
- Програмна реалізація з використанням Android SDK, Java, PHP та RESTful API.
- Тестування функціональності, стабільності та відповідності вимогам.
- Емпіричне дослідження результатів роботи мобільного застосунку у тестовому середовищі.

Наукова новизна

У роботі запропоновано комплексне рішення, що поєднує протокол HL7 ADT та Android-технології для створення мобільного додатку з функціональністю адміністрування медичного закладу, що раніше обмежувалося лише стаціонарними інформаційними системами.

Практичне застосування

Результати роботи можуть бути впроваджені в державних та приватних медичних установах для підвищення ефективності управління, контролю за процесами госпіталізації, виписки, переміщення пацієнтів, а також моніторингу ключових показників діяльності закладу через мобільний інтерфейс.

Бакалаврська робота містить 79 сторінок, 28 рисунків, 3 розділи список використаних джерел із 38 найменуваннями, 1 додаток.

					БР.ІП – 04.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ УПРАВЛІННЯ МЕДИЧНИМИ ЗАКЛАДАМИ

1.1. Передумови розробки Android аплікації для медичного закладу.

Протокол HL7

Керівництву лікарень потрібно щодня відстежувати показники ефективності. До таких показників відносяться заповненість лікарні (щоденний перепис), графік операцій на день, потреби в персоналі та показники догляду за пацієнтами, такі як тривалість перебування кожного пацієнта, деталі діагнозу, демографічний розподіл пацієнтів тощо. Керівництво лікарень переглядає ці показники за допомогою звітів, які генеруються щодня, або на папері, або на їхніх робочих станціях через пропрієтарне програмне забезпечення.

У результаті впровадження інформаційних технологій в охороні здоров'я та клінічній медицині та про захист пацієнтів та доступну медичну допомогу усі медичні заклади, включаючи лікарні, впровадили електронні медичні записи. Крім того, ці норми вимагають відповідно до стандартів "значущого використання" взаємосумісності та електронного обміну медичних даних через стандартні інтерфейси.

Інтерфейс для обміну медичних даних використовує набір даних Health Level 7 (HL7). HL7 — це стандарт розробки інформації про здоров'я [7].

Health Level Seven (HL7) — це набір міжнародних стандартів для обміну, інтеграції, спільного використання та отримання електронної медичної інформації. У простішому розумінні, це як "мова", якою можуть "спілкуватися" різні інформаційні системи в сфері охорони здоров'я, незалежно від їхнього постачальника або призначення.

					БР.ІП – 04.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

Організація Health Level Seven International розробляє ці стандарти, щоб забезпечити сумісність між різними медичними інформаційними системами. Це дозволяє, наприклад:

- Передавати дані про пацієнтів між лікарнями, лабораторіями та іншими медичними установами.
- Обмінюватися результатами аналізів, призначеннями ліків, історією хвороби та іншою важливою клінічною інформацією.
- Автоматизувати робочі процеси та зменшити кількість паперової документації.
- Покращити якість та безпеку надання медичної допомоги завдяки швидкому та точному обміну даними.
- Забезпечити відповідність нормативним вимогам щодо обміну медичною інформацією.

Назва "Health Level Seven" походить від сьомого рівня моделі OSI (Open Systems Interconnection), який є рівнем застосунків. Це означає, що HL7 стандарти зосереджені на змісті та структурі даних, які передаються між системами, а не на технічних деталях їхньої передачі.

Існують різні версії та стандарти HL7, серед яких найбільш поширеними є HL7 v2, HL7 v3 та FHIR (Fast Healthcare Interoperability Resources). FHIR є новітнім стандартом, який вважається більш гнучким та легким у впровадженні, особливо для мобільних та веб-додатків.

Таким чином, HL7 є критично важливим для забезпечення ефективної та безпечної взаємодії між різними інформаційними системами в охороні здоров'я, що в кінцевому підсумку сприяє покращенню якості медичної допомоги

Стандарт HL7 допомагає покращити адміністрування лікарні, передачу та збереження записів, а також забезпечує обмін інформацією між медичними програмними застосунками. Тому інтерфейси HL7, які містять більшість вищезгаданих показників ефективності шпиталю, є універсально

					БР.ІП – 04.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

доступними у всіх EMR лікарень. Інтерфейси HL7 містять різні типи повідомлень, такі як повідомлення про замовлення (ORMs), повідомлення про прийом, випис та передачу (ADT), детальні фінансові транзакції (DFTs) тощо.

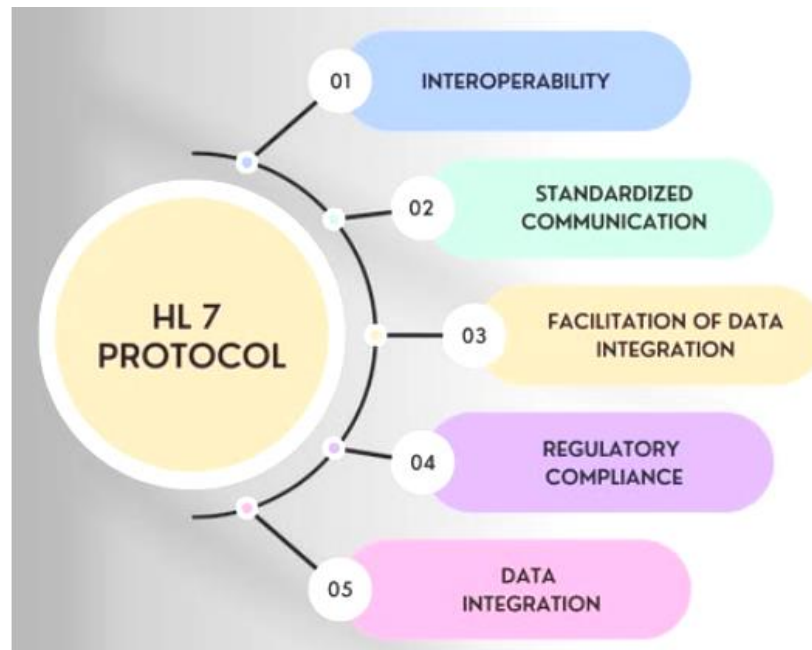


Рисунок 1.1 – Основні аспекти протоколу HL7

На рисунку 1.1 зображена діаграма, яка ілюструє п'ять ключових аспектів HL7 протоколу. У центрі діаграми розташоване коло з написом "HL 7 PROTOCOL". Від цього центрального елемента розходяться п'ять кривих ліній, кожна з яких з'єднується з окремим овальним блоком, що містить опис однієї з переваг або характеристик HL7.

Наведемо опис кожного з цих блоків:

- 01 INTEROPERABILITY (Сумісність). Цей блок підкреслює здатність HL7 забезпечувати взаємодію між різними медичними інформаційними системами.

- 02 STANDARDIZED COMMUNICATION (Стандартизована комунікація). Цей блок вказує на те, що HL7 визначає єдині стандарти для

обміну медичними даними, що полегшує їхнє розуміння та обробку різними системами.

- 03 FACILITATION OF DATA INTEGRATION (Сприяння інтеграції даних). Цей блок наголошує на тому, що HL7 спрощує процес об'єднання даних з різних джерел в єдину систему.

- 04 REGULATORY COMPLIANCE (Відповідність нормативним вимогам). Цей блок показує, що використання HL7 допомагає медичним установам відповідати різним нормативним вимогам щодо обміну та зберігання медичної інформації.

- 05 DATA INTEGRATION (Інтеграція даних). Цей блок, схоже, повторює пункт 03, підкреслюючи важливість HL7 для об'єднання даних. Можливо, це зроблено для більшого акценту на цьому аспекті.

Таким чином, схема візуально представляє HL7 протокол як основу, що забезпечує сумісність, стандартизовану комунікацію, сприяє інтеграції даних та допомагає у дотриманні нормативних вимог у сфері охорони здоров'я

Пропонований мобільний додаток використовує тестовий вхід ADT для отримання операційних даних лікарні та відображає їх на платформі Android за допомогою вебсервісів REST. Основне значення проєкту полягає в мобільному додатку, який використовує тестові дані HL7 ADT як вхідні дані для цього додатку.

1.2. Мета проєкту. Вимоги до мобільної аплікації

Основна мета цього проєкту — розробити мобільний додаток Android, який допоможе керівництву лікарень відстежувати та переглядати показники ефективності лікарні на мобільній платформі, а не використовувати звіти, які генеруються щодня, або на папері, або на їхніх робочих станціях через пропрієтарне програмне забезпечення. Мобільний додаток в основному орієнтований на керівництво лікарні і може бути використаний усіма

					БР.ІП – 04.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

співробітниками управління лікарні, включаючи координаторів та директорів з питань персоналу. Керівництво лікарні може швидко та легко отримати доступ до інформації, що заощадить час.

Мобільний додаток надасть керівництву лікарень можливість оперативно відстежувати, візуалізувати та аналізувати ключові показники ефективності (KPI) лікарні безпосередньо на їхніх мобільних пристроях. Замість щоденного використання статичних звітів у паперовій або цифровій формі, додаток забезпечить доступ до актуальних даних у реальному часі, дозволяючи приймати обґрунтовані управлінські рішення на основі оперативної інформації.

Оскільки всі медичні заклади, включаючи лікарні, використовують EMR, медичні заклади, які впроваджують сертифіковані технології EMR [6], можуть отримати додаткову вигоду. Отже, цей мобільний додаток допоможе в електронному зберіганні медичних записів у портативному режимі за допомогою мобільної платформи Android.

Мобільний додаток дозволяє керівництву медичного закладу переглядати показники ефективності шпиталю з легкістю доступу. Мобільний додаток міститиме операційні дані лікарні, включаючи щоденний перепис, середню тривалість перебування в лікарні, кількість пацієнтів, які поступили та були виписані за певний день, деталі страхування та діагнозу пацієнтів, а також демографічний розподіл пацієнтів.

Для використання цього мобільного додатку користувачеві необхідно увійти, використовуючи дійсні облікові дані. Неправильна або недійсна автентифікація призводить до повідомлення про помилку. Успішний вхід відкриває головну панель, що складається з набору іконок для кожного з вищезазначених показників. Користувач може легко переглянути необхідну інформацію, просто натиснувши на іконку метрики. Щоб переглянути інший показник, користувач може повернутися на екран метрик, використовуючи опцію "назад". Цей мобільний додаток містить опцію оновлення для

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

перегляду оновленої інформації. Після завершення процесу перегляду керівник лікарні може вийти з програми, використовуючи опцію "вийти", і йому потрібно буде знову увійти кожного разу, коли він захоче скористатися програмою.

1.2.1. Перелік показників ефективності лікарні, що пропонується в мобільному додатку

Розглянемо основний перелік показників ефективності лікарні, що буде передбачено в розробці:

- **Прийняття**

Надає інформацію про кількість пацієнтів, прийнятих за певний день.

- **Виписка**

Показник "Виписка" надає детальну інформацію про кількість виписаних пацієнтів за певний день.

- **Кількість пацієнтів**

Щоденна кількість пацієнтів у лікарні є комбінацією кількості пацієнтів, прийнятих та зареєстрованих за певний день, і цю інформацію можна отримати за допомогою цього показника.

- **Страхування**

Містить деталі страхування, такі як назва страхової компанії та кількість пацієнтів, які користуються відповідним видом страхування.

- **Демографія**

Цей показник відображає демографічний розподіл пацієнтів у вигляді стовпчастої діаграми, яка представляє відсоток пацієнтів з кожного населеного пункту, району, області.

- **Тривалість перебування**

Середня тривалість перебування пацієнта в лікарні в днях, тобто скільки часу пацієнт перебував у лікарні до виписки.

- **Діагноз**

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

Містить інформацію про деталі діагнозу пацієнта, такі як найменш та найпоширеніші діагностовані захворювання.

Аналіз інформації про вищезазначені показники допомагає керівникам лікарень приймати ефективні рішення. Наприклад, розглянемо організацію, яка має мережу лікарень у певній області. Аналізуючи демографію пацієнтів для будь-якої географічної області, адміністратори лікарень мають інформацію для належного прогнозування подій та змін навколишнього середовища, щоб скоригувати послуги закладів. Ці зміни дозволяють підтримувати додаткових пацієнтів, лікарняні ліжка, спеціалізації в галузі охорони здоров'я та наймати кваліфікованих клініцистів, орієнтованих на задоволення потреб пацієнтів.

Мобільний додаток частково звільнить персонал медичного закладу від необхідності щоденних паперових звітів та необхідності входу в пропрієтарне програмне забезпечення, яке доступне лише на їхніх індивідуальних робочих станціях.

Розглянемо мінімальні вимоги до системи.

1. Програмні вимоги

- Операційна система Android Mobile: Версія 5.2 або вище.
- Інструменти IDE: Android Studio та Eclipse.
- SQL Server Management Studio.
- Клієнт FTP FileZilla.
- Інтерфейс користувача: XML.
- Код: JAVA, XML, PHP, SQL та JSON.
- Тестовий вхід HL7 ADT.
- Портал Microsoft Azure.
- Інтернет: Так.

2. Апаратні вимоги

- Оперативна пам'ять: 512 МБ або вище.
- Пристрій повинен підтримувати відладку через USB.

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

- Процесор: 1 ГГц або вище.

1.3. Особливості ADT повідомлень в контексті розробки мобільного додатку

Повідомлення про прийом, випис та передачу (ADT) є найбільш поширеними повідомленнями в стандарті HL7. Система інформації медичного закладу (HIS) ініціює ці повідомлення для спілкування з іншими системами щодо подій тригерів, таких як інформація, пов'язана з прийомом, випискою та передачею пацієнта. Ці повідомлення також використовуються для збереження деталей діагнозу, деталей страхування та демографічних даних пацієнта, таких як ім'я пацієнта та адреса. Воно також допомагає дізнатися інформацію про відвідування пацієнтів, таку як місцезнаходження пацієнта, основний заклад, лікар, який веде тощо.

У контексті охорони здоров'я, ADT розшифровується як Admission, Discharge, and Transfer. Це один із фундаментальних типів повідомлень, що використовуються в стандартах HL7 (Health Level Seven) для обміну інформацією про рух пацієнтів у медичному закладі.

ADT-повідомлення фіксують і передають інформацію про такі події, як:

- Admission (Прийняття): Коли пацієнт поступає до лікарні або іншого медичного закладу.
- Discharge (Виписка): Коли пацієнт покидає лікарню або інший медичний заклад.
- Transfer (Переведення): Коли пацієнта переводять з одного відділення лікарні в інше, або між різними медичними закладами в межах однієї системи.

Кожне ADT-повідомлення містить важливу інформацію про пацієнта та подію, включаючи:

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

- Ідентифікаційні дані пацієнта (ім'я, прізвище, ідентифікатор пацієнта тощо).

- Інформацію про подію (тип події - прийняття, виписки, переведення, час події, місцезнаходження).

- Додаткову інформацію, залежно від типу події (наприклад, причина прийняття, діагноз при виписці, відділення призначення при переведенні).

ADT-повідомлення є критично важливими для підтримки точної та актуальної інформації про статус та місцезнаходження пацієнтів у медичній інформаційній системі. Вони використовуються різними системами для:

- Оновлення реєстру пацієнтів.

- Ініціювання робочих процесів (наприклад, призначення палати, формування медичної карти).

- Інформування зацікавлених сторін (наприклад, інші відділення, лабораторії).

- Забезпечення точності білінгу та фінансової звітності.

- Підтримки статистичного обліку.

Таким чином, ADT-повідомлення є основою для багатьох інтеграційних процесів у сфері охорони здоров'я, забезпечуючи безперебійний обмін інформацією про пацієнтів між різними системами та сприяючи більш ефективній та скоординованій медичній допомозі.

1.3.1. Типи повідомлень ADT

Існує 51 тип повідомлень ADT, які використовуються для представлення різних подій тригерів. Цей мобільний додаток використовує 3 типи повідомлень ADT. Це повідомлення ADT^A01 про прийом, повідомлення ADT^A03 про виписку та повідомлення ADT^A04 про реєстрацію.

Повідомлення ADT^A01 про прийом - ця подія тригера генерується, коли пацієнта приймають до медичного закладу.

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

Повідомлення ADT^A03 про виписку - ця подія тригера ініціюється, коли пацієнта виписують з медичного закладу.

Повідомлення ADT^A04 про реєстрацію. Система медичного закладу ініціює повідомлення A04, коли пацієнт зареєструвався як амбулаторний пацієнт у системі або зареєструвався на конкретний візит.

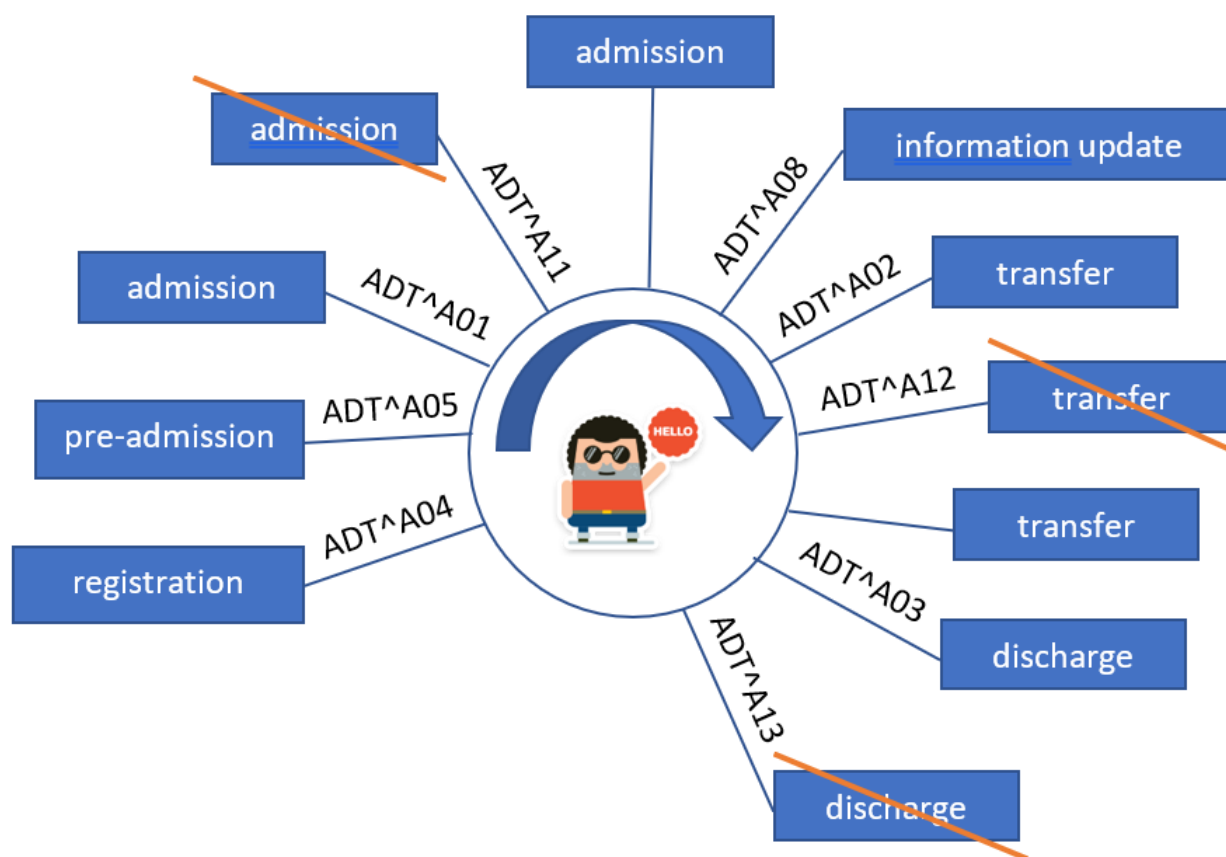


Рисунок 1.2 – Основні типи повідомлень ADT

На рисунку 1.2 зображена діаграма, яка ілюструє різні типи ADT (Admission, Discharge, Transfer) повідомлень у контексті HL7 (Health Level Seven) протоколу, що обертаються навколо центральної фігури, яка, ймовірно, символізує пацієнта або медичну інформаційну систему.

Від центрального елемента розходяться стрілки, позначені різними кодами ADT-повідомлень, до блоків, що описують відповідну подію або дію.

Блоки та відповідні ADT-повідомлення:

- admission (прийняття): З'єднано зі стрілкою, позначеною ADT^A01. Це повідомлення фіксує інформацію про первинне прийняття пацієнта до медичного закладу.

- admission (прийняття): З'єднано зі стрілкою, позначеною ADT^A11. Це повідомлення використовується для скасування попереднього повідомлення про прийняття.

- admission (прийняття): З'єднано зі стрілкою, позначеною ADT^A08. Це повідомлення використовується для оновлення інформації про прийняття пацієнта.

- pre-admission (попереднє прийняття): З'єднано зі стрілкою, позначеною ADT^A05. Це повідомлення використовується для реєстрації інформації про пацієнта до його фактичного надходження.

- registration (реєстрація): З'єднано зі стрілкою, позначеною ADT^A04. Це повідомлення використовується для реєстрації пацієнта в системі.

- discharge (виписка): З'єднано зі стрілкою, позначеною ADT^A03. Це повідомлення фіксує інформацію про виписку пацієнта з медичного закладу.

- discharge (виписка): З'єднано зі стрілкою, позначеною ADT^A13. Це повідомлення використовується для скасування попереднього повідомлення про виписку.

- transfer (переведення): З'єднано зі стрілкою, позначеною ADT^A02. Це повідомлення фіксує інформацію про переведення пацієнта з одного відділення в інше в межах одного закладу.

- transfer (переведення): З'єднано зі стрілкою, позначеною ADT^A12. Це повідомлення використовується для скасування попереднього повідомлення про переведення.

- transfer (переведення): З'єднано зі стрілкою, позначеною ADT^A06. (Цей код не відображений на схемі, але логічно впливає з контексту інших повідомлень про переведення).

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

- information update (оновлення інформації): З'єднано зі стрілкою, позначеною ADT^A02. (Схоже на помилку на схемі, оскільки ADT^A02 зазвичай позначає переведення. Можливо, малося на увазі інший код для оновлення загальної інформації про пацієнта).

Стрілки показують, що різні події ADT пов'язані з пацієнтом (центральною елементом) і відображають етапи його перебування в медичному закладі або дії, пов'язані з його обліком. Кругова стрілка в центрі може символізувати цикл життя пацієнта в системі або постійний обмін інформацією.

Отже, схема ілюструє основні типи ADT-повідомлень, які використовуються в HL7 для відстеження руху пацієнтів та пов'язаних з цим дій у медичній інформаційній системі. Вона показує, як різні події, починаючи від попередньої реєстрації та прийняття, через переведення між відділеннями, і закінчуючи випискою, фіксуються та передаються за допомогою стандартизованих повідомлень ADT з відповідними кодами. На схемі присутня невелика неточність у позначенні одного з повідомлень (ADT^A02 для "information update"), але загалом вона чітко відображає концепцію ADT в HL7.

1.3.2. Структура повідомлення HL7 ADT

Нижче наведено приклад повідомлення ADT у форматі HL7.

```
MSH|^~&|EPIC|WAMC|IGUANA||20160314092821|EDRN|ADT^A01|423||2.4
EVN|A01|20160314092821||ED_AFTER_ARRIVAL|EDRN^EMERGENCY^NUR SE^^^^^PHSOC^^^^^WAMC
PID|1||800000812^^^EPI^MR||YOUNG^ASAP||19701015|M||^A^US^L|||||7700000037504|555-12-4652||
DG1|1|^brain injury | brain injury ||^10151;EPT
IN1|1||MEDICARE||||||20061115|20170316|| YOUNG^ASAP |Self||555 MAIN ST^^ONTARIO^CA^91761^US
```

Рисунок 1.3 – Фрагмент повідомлення HL7 ADT

Повідомлення HL7 ADT, показане вище, містить інформацію про показники ефективності шпиталю. Дуже важко читати ці повідомлення та

					БР.ІП – 04.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

отримати необхідну інформацію про показники. У цьому додатку ці повідомлення декодуються в зрозумілий формат, такий як таблиці в базі даних, за допомогою мови програмування Java.

1.4. Використання інформаційних систем управління об'єктами для медичних закладів

Сучасні медичні заклади повинні працювати з складними інформаційними потоками, пов'язаними з інформацією з різних джерел і сфер. Крім того, шпиталі повинні відповідати суворим гігієнічним, якісним та організаційним стандартним вимогам, встановленим національними та міжнародними інститутами для виконання клінічних та медичних обов'язків. Служби інженерії клініки та технічні відділи повинні знаходити рішення для повного задоволення всіх технологічних, структурних та організаційних потреб такого складного об'єкта, як лікарня (клініка).

Для цих потреб є розроблено кілька технічних інструментів для моніторингу так само, як і будь-якого іншого складного об'єкта, вимірюючи кількісні, архітектурні, технологічні та пов'язані з людьми параметри.

Еволюцією простих СКБД є географічні інформаційні системи (ГІС), які є інструментами, призначеними для захоплення, збереження, маніпулювання, аналізу, управління та представлення всіх типів географічно орієнтованих даних. Це технологія, яка поєднує картографію, статистичний аналіз і дані. Існує два специфічні типи даних, пов'язаних з картою, а саме:

- просторові дані, які описують місце та форму географічних об'єктів та їхні просторові відносини з іншими об'єктами;
- описові дані, що стосуються географічних об'єктів (атрибути).

Просторові дані відображаються за допомогою графічних функцій: точок, векторів і полігонів. Атрибути представляють якісну або кількісну сторону відображуваного явища на цифровій карті. Таким чином, для

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

географічного об'єкта таблиця атрибутів може асоціювати різні числові та текстові параметри. Програмне забезпечення, пов'язане з ГІС, зазвичай виконує типові вимірювання, такі як аналіз шляхів і пошук людей, але зазвичай не надає кількісних порівняльних даних [16, 17].

Системи комп'ютерної підтримки управління об'єктами (CAFM) — це інструменти підтримки прийняття рішень на основі інтегрованих моделей управління медичними закладами (INFMM), які надають KPI для процесів, які можуть вплинути на ефективність медичної структури. Ці інструменти можуть бути дуже корисними для прийняття рішень щодо оцінки ефективності та ризиків, управління бізнесом та розвитку.

Системи управління робочими місцями (WMS) — це рішення, призначені для управління нерухомістю, які дозволяють користувачам оцінювати, аналізувати та реорганізовувати активи компанії, щоб зберегти їхню вартість. Вони також дуже корисні для покращення ефективності та задоволення різноманітних потреб. Вони надають доступ до збереженої інформації незалежно від робочого місця: дані та плани можуть бути отримані через веб-сервіси за допомогою звичайного браузера через Інтернет або інтранет.

Система, що аналізується є набором програмного забезпечення, що складається з основного програмного модуля під назвою SPOT та додаткових інструментів, які посилаються на ту саму внутрішню базу даних, пов'язану з інформаційною системою шпиталю (HIS) (рис. 1.4). Вона складається з автономного основного виконуваного додатку, який відстежує стан будівель щодо ліжок, квадратних метрів, призначення використання, функціональних зон та багатьох інших характеристик для кожної кімнати. Основний модуль призначений лише для техніків, тоді як агреговані та вже оцінені дані показуються персоналу шпиталю та керівництву за допомогою RESTful (REpresentational State Transfer) HTML5-сумісного рендерингового двигуна та макета, доступного через веб-браузер під назвою SPOTWEB. Його також

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

можна описати як центральну панель управління: насправді основна база даних пов'язана з іншими сховищами даних шпиталю, що дозволяє агрегувати гетерогенну та спочатку непов'язану інформацію.

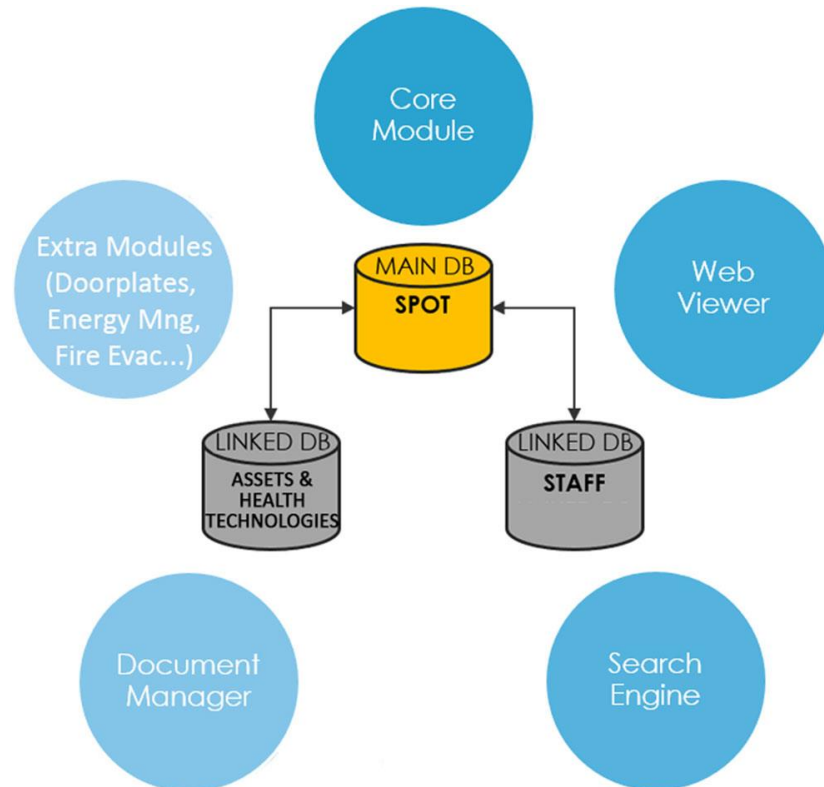


Рисунок 1.4 - Реляційна схема автоматизованої системи управління об'єктами

Крім того, SPOT FINDER — це веб-пошукова система, яка дозволяє користувачам виконувати вільні текстові запити до даних, збережених у базі даних. Система також оснащена двигуном реального часу для створення звітів. Крім того, DOCUMENT MANAGER — це додатковий модуль, який використовується для посилення на зовнішні цифрові вкладення до кімнат або функціональних груп, незалежно від форматів і джерел.

Ідея цієї роботи полягає в тому, щоб максимально розширити обмін знаннями, дозволяючи персоналу знати якомога більше про свій медичний заклад. Незважаючи на те, що це система CAFM, застосунок не безпосередньо керує програмним забезпеченням CAD (наприклад, AutoCAD, ArchiCAD тощо), а використовує внутрішні бібліотеки для конвертації

					БР.ІП – 04.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

цифрових планів DXF у SVG, щоб їх можна було відображати та керувати без будь-якого іншого програмного забезпечення, крім звичайних браузерів, сумісних з HTML5 (Google Chrome, Mozilla Firefox, Microsoft Edge, Apple Safari тощо). Єдиною вимогою для успішної конвертації є наявність присвяченого шару на карті DXF (ім'я якого можна встановити всередині системи) з замкненими полілініями, кожна з яких обрисовує окрему кімнату. Координати зберігаються безпосередньо в центральній базі даних, у якій зберігаються всі дані. Після завершення конвертації початковий DXF більше не потрібен. Крім того, вся графічна інформація про стіни, вікна, двері та будь-які інші архітектурні елементи також конвертується в SVG, щоб кінцевий користувач міг бачити план поверху в його цілісності.

Після завершення імпорту користувач може ввести інформацію про різні перелічені дані для кожної кімнати (див. таблицю 1.1).

Дані можуть бути зібрані через опитування на місці та інтерв'ю з персоналом, потім простори класифікуються за їх використанням та очікуваннями клієнтів щодо комфорту середовища. Інформація з опитування вводиться в систему відповідно до таблиці 1.1. Система автоматично призначає кольори кімнатам відповідно до призначення використання, відділу, оперативної одиниці або класу прибирання. Колірний код повністю налаштовується в розділі налаштувань програмного забезпечення. Це дозволяє користувачам одразу ідентифікувати основну інформацію про кімнату, просто подивившись на план поверху.

Зібрана інформація не лише дозволяє оцінювати просторові показники (наприклад, доступна поверхня для ліжка, кількість ліфтів для пацієнтів, які не можуть ходити тощо), але й дозволяє керівникам керувати загальними річними витратами на кожне приміщення (наприклад, знаючи його конкретний клас прибирання та пов'язані витрати за квадратний метр). Інформація про фактичну доступність приміщень також може бути керована,

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

знаючи, наприклад, періоди часу, коли кімнати не працювали (наприклад, для будівельних робіт).

Таблиця 1.1 – Дані для системи

Назва даних	Опис даних
Код кімнати	Унікальний алфацифровий код для ідентифікації будівлі, поверху та номера певної кімнати відповідно до правила PREMISE_BUILDING_LEVEL_ROOM, де PREMISE — це одна літера, яка ідентифікує приміщення, BUILDING — це унікальний чотиризначний код для будівлі, LEVEL — це двозначний номер поверху (опціонально з мінусом для підвалів), а ROOM — це чотиризначний алфацифровий код. Об'єднання цих 4 кодів дає унікальний "говорячий код", який ідентифікує кожну окрему кімнату всередині шпиталю, дозволяючи користувачеві одразу дізнатися про її розташування.
Призначення використання (DU)	Набір з 36 категорій, до яких може бути віднесена кожна кімната, щоб одразу ідентифікувати виконувану діяльність.
Клас	Підмножина для кожного призначення використання, використовувана для уточнення інформації про діяльність (наприклад, категорія операційної DU групує різноманітні зони, які повинні бути оброблені по-різному як з архітектурної, так і з інженерної точки зору. Класи допомагають розрізняти "Зона фільтрації пацієнтів", "Зона фільтрації операторів" тощо).
Відділ	Медичний або технічний відділ, до якого віднесена кімната.
Оперативна одиниця	Оперативна одиниця, призначена для кімнати.
Клас прибирання	Набір з 6 класів, пов'язаних із складністю обов'язків з прибирання (Дуже високий ризик, Високий ризик, Середній ризик, Низький ризик 1, Низький ризик 2, Незастосовано). Чим складніше прибирання, тим вищими є витрати на нього.
Кількість ліжок	Кількість ліжок всередині кімнати (де застосовно).
Площа	Площа кімнати автоматично оцінюється за її координатами за допомогою алгоритму Гауса (кв. м).
Висота	Висота кімнати (м).
Об'єм	Об'єм кімнати оцінюється за її площею та висотою (куб. м).
Статус кімнати	Поточний статус кімнати відповідно до її фактичного використання.
Індекс обслуговування	Числовий індекс, який підкреслює складність обслуговування для цієї конкретної кімнати. Він пов'язаний з призначенням використання, але може бути перевизначений користувачами з адміністративними привілеями.
Персонал	Інформація про людей, які працюють у цьому приміщенні, ідентифікованих за їхнім персональним ідентифікатором.

Карти відображаються за допомогою автоматично згенерованих полігонів SVG на основі координат кімнати, збережених у центральній базі даних. Формат SVG (сумісний з W3C HTML5) можна переглядати в будь-якому веб-браузері без необхідності сторонніх плагінів.

Додаток дозволяє користувачам переглядати останню оновлену карту запитаного поверху будівлі в реальному часі за допомогою AJAX і JavaScript. Кожне натискання мишею викликає асинхронний доступ до веб-сервера через веб-сервіси. Потім контролери обробляють запит до бази даних і відповідають запитаною інформацією. Використовуються стандартні формати, такі як JSON і XML.

Двигун також надає базові функції двигуна CAD, такі як багатовибір, прокручування, масштабування, розміщення тексту та масштабування (рис. 1.5).



Рисунок 1.5 - Скріншот системи управління медичним закладом. У прикладі виділено кольором усі коридори поверху

Крім того, користувач може отримати доступ до інформації, збереженої в основній базі даних або в інших пов'язаних базах даних, відповідно до привілеїв, пов'язаних з їхнім ім'ям користувача та вибором кімнат. Це

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

включає загальні та організаційні деталі (рис. 1.6) або конкретні деталі про доступні або встановлені активи та медичні технології (рис. 1.7).

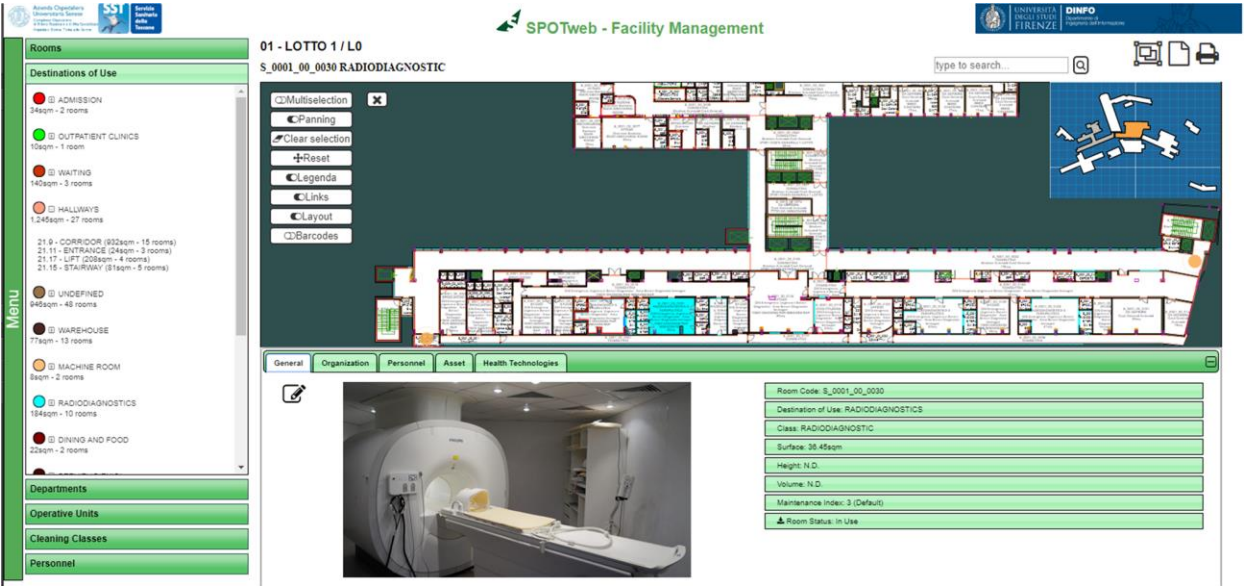


Рисунок 1.6 - Загальні відомості відображаються в нижньому вікні. Зліва розташований елемент керування, який показує функціональні групи доступних "Призначень використання" для обраного поверху

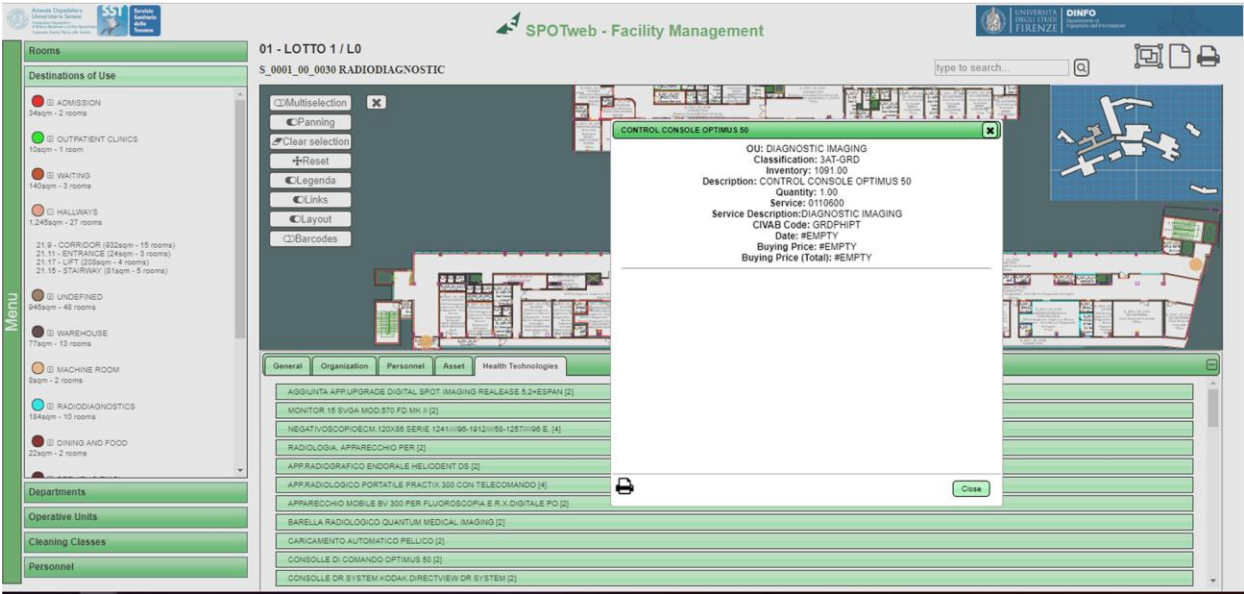


Рисунок 1.7 - Детальна інформація про консоль керування рентгенодіагностикою для обраного приміщення (Код приміщення: S_0001_00_0030)

Зв'язки між базами даних здійснюються за допомогою унікального коду кімнати як ідентифікаційного ключа, що дозволяє агрегувати гетерогенну інформацію за допомогою простих зв'язків "один до багатьох". Цей підхід робить SPOT зручним інструментом для доступу до всіх видів інформації, доступної через кілька джерел даних, з єдиним центральним графічним веб-інтерфейсом користувача.

Ще одна важлива функція, яку повинен забезпечувати глядач, — це можливість навігації по поверхах шпиталю в горизонтальному напрямку. Це майже обов'язкова вимога через невідповідність між поверхами різних павільйонів, описаних вище. Насправді користувач повинен легко помітити, чи є такий рівень в іншій будівлі, який має горизонтальне з'єднання з поточно відображеним поверхом, і, можливо, перейти до нього (Рис. 5).



Рисунок 1.8 - Червона стрілка показує існуюче посилання (пульсуюче помаранчеве коло) в кінці коридору на інший рівень в іншому сусідньому павільйоні (також виділеному помаранчевим кольором у верхньому правому навігаторі)

Структурні дані (поверхні, висоти та ліжка), організаційні дані (імена та контакти персоналу, код кімнати, гетерогенні організаційні підгрупи, такі

як відділи, оперативні одиниці та класи прибирання), технічні дані (медичні пристрої та активи), а також дані про установки та архітектурні дані (установки обробки повітря, рами, світильники) доступні для кінцевого користувача.

Через велику кількість даних, збережених у базі даних CAFM, потрібна веб-пошукова система для виконання вільних текстових запитів відповідно до непередбачуваних потреб користувача. Пошукова система інтегрована в веб-переглядач через спеціальний веб-сервіс, але також має власний графічний інтерфейс користувача (GUI) для виконання більш детальних пошуків, які можуть бути ще більше уточнені за допомогою інструменту розширеного пошуку для обмеження виводу (рис. 1.9).

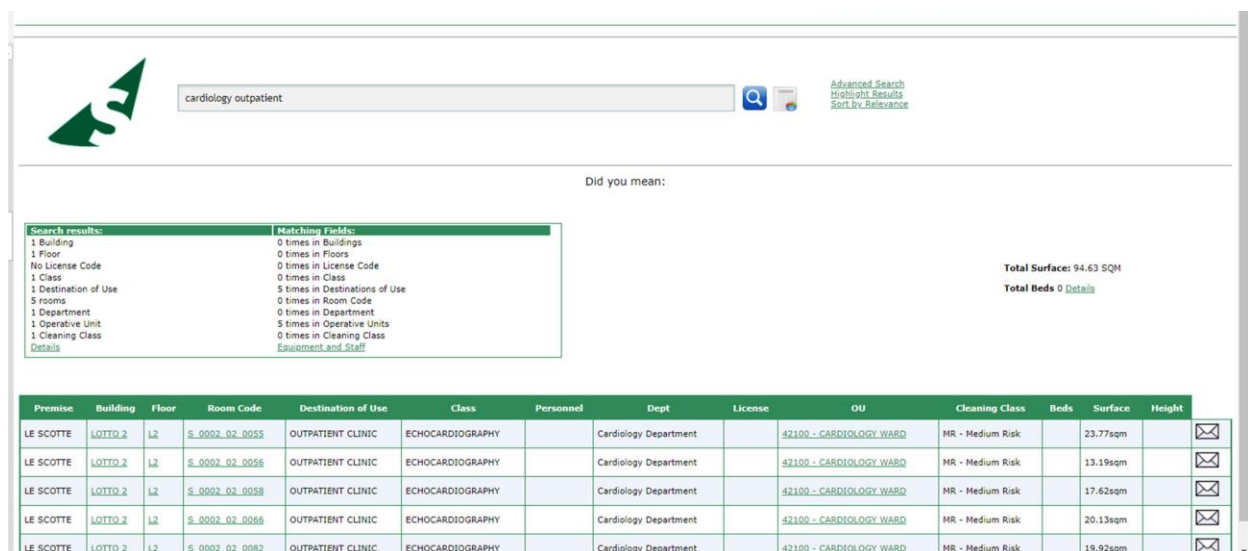


Рисунок 1.9 - Скріншот пошукової системи. У прикладі відображено результати за ключовими словами "кардіологічна амбулаторія"

Алгоритм пошуку дозволяє користувачам виконувати запити за допомогою звичайного синтаксису Google та символів ASCII (включати, виключати, точний пошук). Також реалізовані розмиті словники та API Google для надання функції "Ви бачили?".

Двигун також надає функцію динамічного створення звітів, яка дозволяє користувачам виводити спеціальні звіти на основі результатів пошуку з різними рівнями агрегації.

Однією з проблем, з якою часто стикається керівний склад при управлінні медичним закладом, є те, як об'єднати гетерогенну інформацію не лише за змістом, але й за форматами та джерелами. Наприклад, деякі текстові документи (наприклад, протоколи зустрічей або звіти про перевірки на місці) можуть бути у форматі .pdf і стосуватися певної організаційної структури; звіти можуть бути файлами .csv з таблицями та графіками про поточні витрати на обслуговування КТ, розташованого в конкретному відділенні; асбуйлд цифрові плани поверху можуть бути у форматі .dwg або будь-якому іншому форматі CAD. Це лише кілька прикладів того, як різномісна та складна може бути організація інформації.

DOCUMENT MANAGER — це додатковий модуль, спеціально розроблений для пом'якшення цієї проблеми, який допомагає офісам та секретаріатам у організації цифрових документів, економлячи час на їх пошук керівництвом або будь-ким, хто в них потребує.

Файли будь-якого типу та формату можуть бути завантажені в систему за допомогою вищеприписаного інтерфейсу веб-переглядача, і їх можна пов'язати з окремою кімнатою або функціональною підгрупою, такою як відділи, оперативні одиниці або класи прибирання для певної будівлі та поверху.

Після цього користувачі входять у додаток за допомогою іншого спеціального GUI, щоб побачити структуру папок для завантажених вкладень, угрупованих за павільйоном, рівнем, організаційною зоною та кімнатою, тоді як числові значки поруч з іменами папок легко ідентифікують кількість завантажених документів (рис. 1.10). У верхній частині інтерфейсу доступна панель пошуку для виконання текстових пошуків у дереві вкладень.

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

Коли файл вибрано, користувач, який увійшов у систему, може завантажити його або, якщо він має високі привілеї, також видалити.



Рисунок 1.10 - Графічний інтерфейс користувача (GUI) менеджера документів з правами адміністративного входу

Кімнати з прикріпленими документами або такі, що входять до функціональної підгрупи з прикріпленими документами, також виділяються в веб-переглядачі за допомогою значка скріпки поруч з ім'ям.

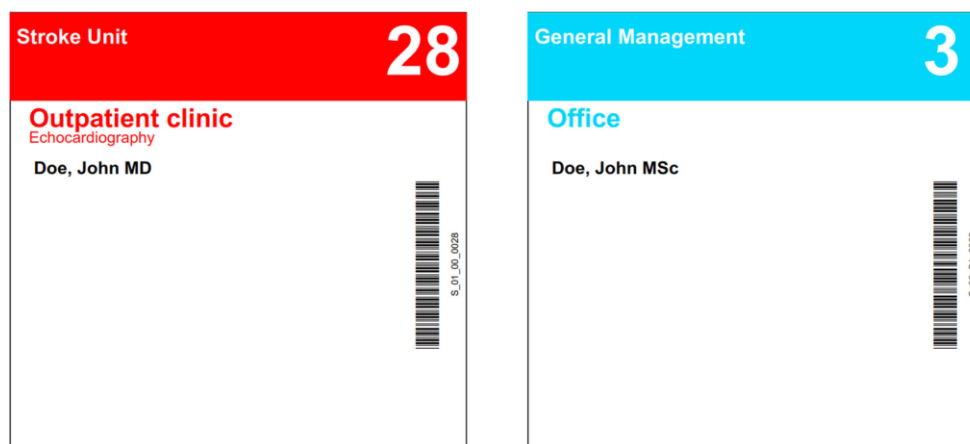


Рисунок 1.11 - Доступні варіанти оформлення табличок

Останній доступний модуль набору — це модуль створення табличок. За допомогою основного додатку користувач призначає персонал для кожної кімнати в закладі (як показано в таблиці 1.1). Основна база даних пов'язана з базою даних персоналу шпиталю, де люди пов'язані з оперативною одиницею. Співробітників спочатку фільтрують, вибираючи оперативну одиницю для аналізованої кімнати, а потім пов'язують її з самою кімнатою.

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

Після створення зв'язку система може автоматично генерувати таблички для вибраних кімнат з попередньо спроектованими макетами відповідно до призначення використання та будівлі (рис. 1.11).

Система дозволяє запити на кімнати з виводом як числових, так і графічних звітів (рис. 1.12), тим самим стаючи інструментом підтримки для планування охорони здоров'я.

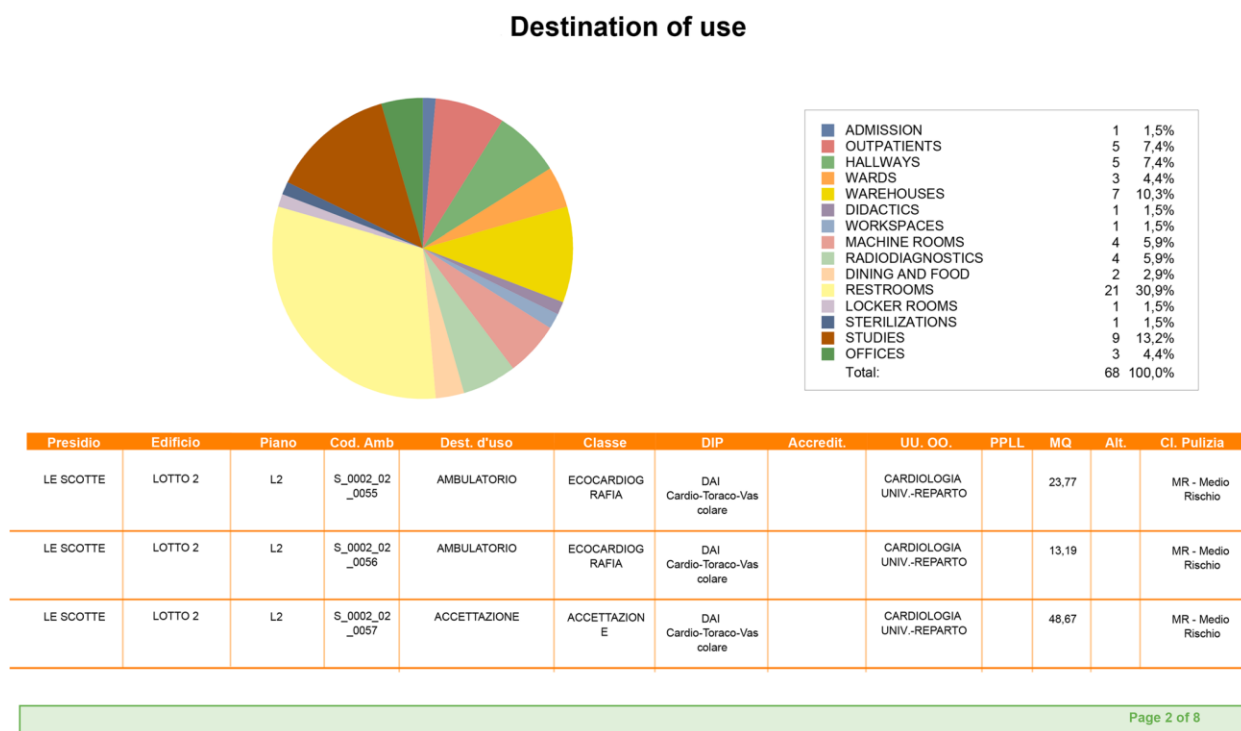


Рисунок 1.12 – Звіт результатів пошуку

Нижче наведено кілька прикладів, угрупованих за користувачами:

- директори та координатори медсестер використовують SPOTWEB разом із SPOT FINDER, щоб дізнатися про просторове розташування одиниць/відділів під їхнім контролем, наявність медичного обладнання та активів. Вони також запитують систему для отримання інвентарного номера пристрою для планових або корегуючих цілей обслуговування.

- служба охорони праці та безпеки запитує SPOT (зокрема модуль евакуації при пожежі) для дізнання про шляхи втечі по будівлях, де знаходяться пожежні виходи та пожежні сходи, і для тестування часу втечі

для різних відділень і відділів відповідно до різних симуляцій пожежі. Крім того, система надає точні дані про те, які місця є більш чутливими, отже, потребують більше уваги (кімнати з горючими або окислювальними речовинами, такі як відділення або відділення інтенсивної терапії, супермагнітні кімнати відділень магнітно-резонансної томографії тощо) [37].

- менеджер з енергетики часто використовує систему управління енергією для дізнання про відношення між кондиціонером і його відповідною установкою обробки повітря або для оцінки відповідності між теоретичними та реальними значеннями для певної кімнати з метою планування втручань та обслуговування.

- технічний персонал використовує SPOT майже щодня для отримання параметрів, які використовуються для управління, та для розрахунку показників ефективності якості обслуговування.

- будь-хто в лікарні може запитати SPOT FINDER, щоб дізнатися коди кімнат та оперативні одиниці будь-якої медичної або немедичної діяльності. Цей інструмент також надає інформацію про персонал, тому він також є корисним інструментом для пошуку людей.

Дана система управління використовується в багатьох сценаріях і для різних цілей, таких як управління перевезеннями, оцінка вимог до акредитації, управління медичними технологіями та активами та загальне проектування та ремоделювання. Система також використовується в управлінні та управлінні медичним закладом, таких як перевірка призначень використання та аналіз витрат/простору оперативних одиниць.

1.5. Програмні засоби, орієнтовані на інноватизацію процесів управління в медичних закладах

На рисунку 1.13 представлено структуровану класифікацію програмних засобів, орієнтованих на інноватизацію процесів управління в медичних

					БР.ІП – 04.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

зкладах. Класифікаційна схема виконана у форматі блок-діаграми, де центральним елементом є вертикально орієнтований блок, що ідентифікує загальну категорію досліджуваних засобів: "Програмні засоби для інноватизації управління медичними закладами".

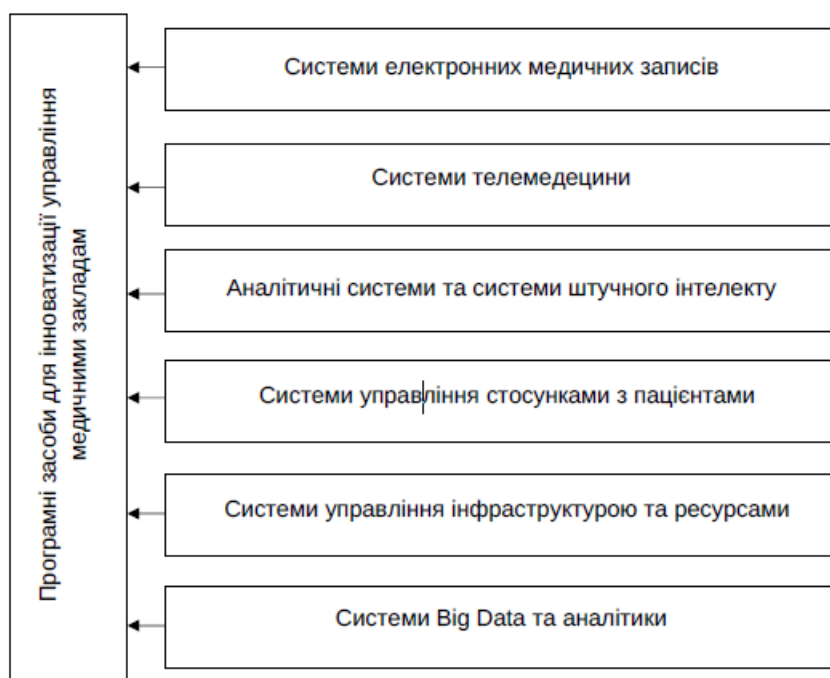


Рисунок 1.13 – Класифікація програмних засобів для медичних закладів

Від зазначеного центрального блоку по горизонталі відходять сполучні елементи (стрілки) до шести термінальних блоків прямокутної форми, кожен з яких репрезентує окрему, чітко визначену категорію програмного забезпечення. Зазначені категорії є наступними:

- Системи електронних медичних записів. Програмні комплекси, призначені для створення, зберігання, обробки та управління інформацією про стан здоров'я пацієнтів в електронному форматі, забезпечуючи інтероперабельність та доступність клінічних даних.
- Системи телемедицини. Технологічні рішення, що забезпечують надання медичних послуг на відстані за допомогою інформаційно-комунікаційних технологій, включаючи консультації, діагностику та моніторинг стану пацієнтів.

- Аналітичні системи та системи штучного інтелекту. Програмні засоби, що використовують методи статистичного аналізу, машинного навчання та інших технологій штучного інтелекту для виявлення закономірностей, прогнозування та підтримки прийняття рішень у сфері управління медичним закладом та клінічної практики.

- Системи управління стосунками з пацієнтами (CRM). Програмні платформи, спрямовані на оптимізацію взаємодії медичного закладу з пацієнтами на всіх етапах, від первинного звернення до післялікувального супроводу, з метою підвищення лояльності та якості обслуговування.

- Системи управління інфраструктурою та ресурсами. Програмні комплекси, призначені для ефективного планування, обліку та розподілу матеріально-технічних ресурсів медичного закладу, включаючи обладнання, приміщення, медичні матеріали та персонал.

- Системи Big Data та аналітики. Програмні рішення для обробки та аналізу великих обсягів різнорідних даних, що генеруються в медичному закладі, з метою виявлення значущих тенденцій, оптимізації процесів та покращення якості медичної допомоги.

Представлена класифікація демонструє комплексний підхід до інноватизації управління медичними закладами шляхом впровадження спеціалізованих програмних засобів, що охоплюють ключові функціональні області діяльності. Структура схеми підкреслює ієрархічну залежність між загальною метою інноватизації та конкретними типами програмного забезпечення, що використовуються для її досягнення.

Розглянемо декілька існуючих систем управління в медичних закладах.

1.5.1. Онлайн платформа Doctolib

Doctolib — це провідна європейська онлайн-платформа для запису на прийом до лікарів та інших медичних працівників. Її основна мета —

					БР.ІП – 04.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

спростити та оптимізувати процес пошуку, вибору та запису на медичні консультації як для пацієнтів, так і для медичних працівників.

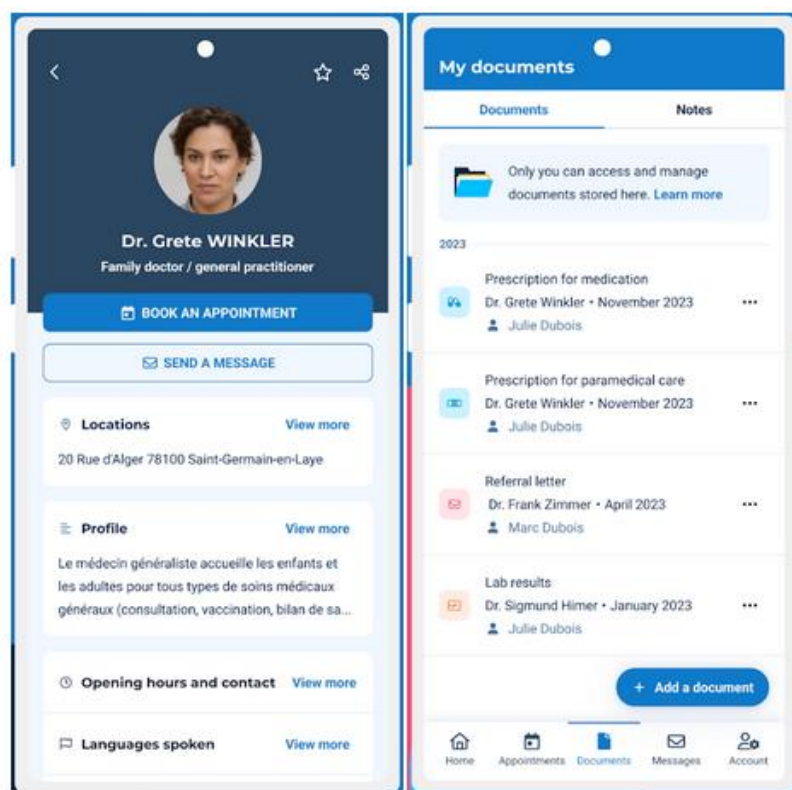


Рисунок 1.14 – Інтерфейс додатку Doctolib

Ось ключові характеристики та функціональні можливості системи Doctolib:

- Doctolib надає зручний інтерфейс для пошуку лікарів, стоматологів, фізіотерапевтів та інших медичних спеціалістів за спеціалізацією, місцем розташування, доступністю та іншими критеріями.

- Кожен медичний працівник має свій профіль з інформацією про його спеціалізацію, досвід, освіту, доступні години прийому, вартість консультацій (за наявності), мови спілкування та відгуки інших пацієнтів.

- Пацієнти можуть у режимі реального часу переглядати доступні слоти для запису та бронювати прийом онлайн, без необхідності телефонувати до реєстратури.

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

- Платформа дозволяє пацієнтам переглядати свої майбутні та минулі записи, скасовувати або переносити їх (згідно з політикою лікаря).

- Doctolib надсилає автоматичні нагадування пацієнтам перед запланованим візитом через SMS та/або електронну пошту, що допомагає зменшити кількість неявок.

Телеконсультації (відеоконсультації): Doctolib також підтримує можливість проведення онлайн-консультацій (відеодзвінків) з лікарями, що є особливо актуальним для повторних візитів, консультацій з питань загального здоров'я або для пацієнтів, які не можуть особисто відвідати лікаря.

- Doctolib надає лікарям та їхнім адміністраторам інструменти для ефективного управління розкладом прийомів, блокування часу, налаштування доступності та інтеграції з іншими календарями.

Doctolib може інтегруватися з існуючими системами управління медичною практикою (PMS) та електронними медичними записами (EMR), спрощуючи обмін даними. Doctolib зазвичай працює за моделлю платної підписки для медичних працівників та закладів, надаючи їм доступ до повного спектру функцій платформи. Для пацієнтів користування платформою є безкоштовним.

Отже, Doctolib є комплексною та зручною платформою, яка полегшує взаємодію між пацієнтами та медичними працівниками, оптимізує процес запису на прийом та надає ряд корисних інструментів для обох сторін.

1.5.2. Онлайн платформа Doctena

Doctena — це онлайн-платформа для пошуку та запису на прийом до лікарів та інших медичних працівників, подібна до Doctolib. Вона функціонує як посередник між пацієнтами та медичними закладами, спрощуючи процес запису на консультації та підвищуючи доступність медичних послуг.

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

Наведемо основні характеристики та функціональні можливості системи Doctena:

- Doctena дозволяє пацієнтам шукати лікарів за спеціалізацією, місцем розташування, мовою спілкування, доступністю та типом страхування (в залежності від країни).

- Кожен медичний працівник має детальний профіль, що містить інформацію про його спеціалізацію, досвід, контактні дані, години прийому, вартість консультацій (за наявності) та, часто, фотографію.

- Пацієнти можуть переглядати доступні слоти в календарі лікаря та записуватися на прийом у будь-який час доби, без необхідності телефонувати.

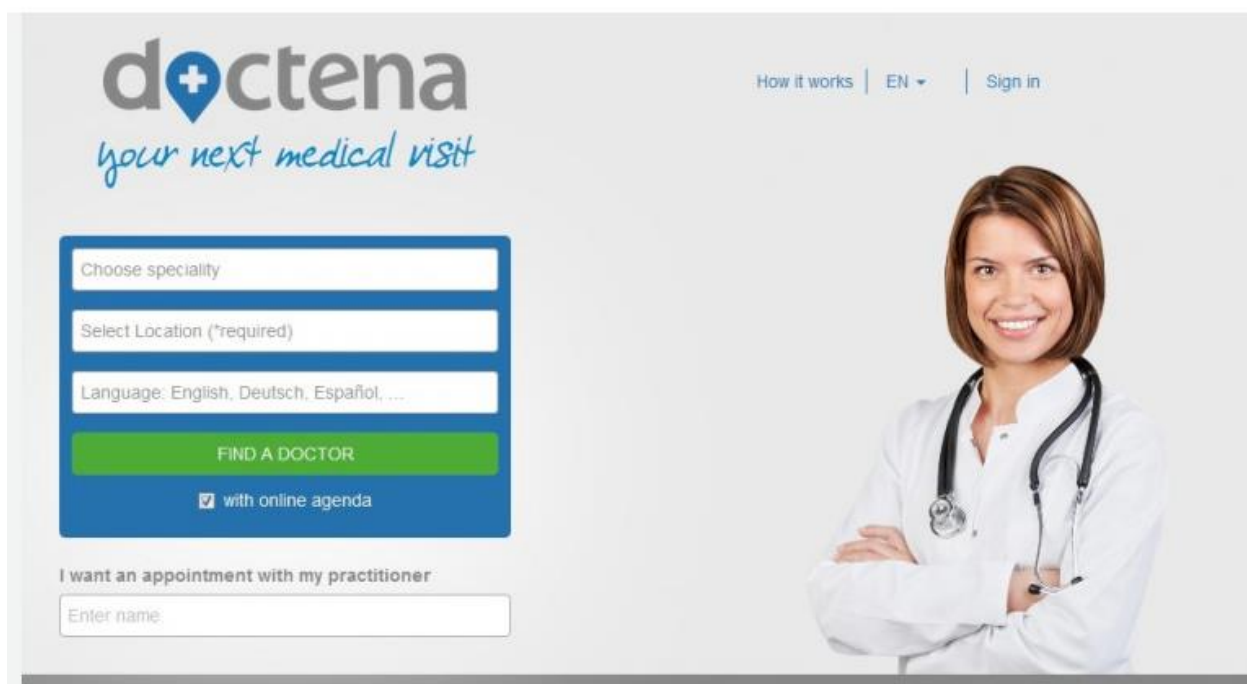


Рисунок 1.15 – Інтерфейс Doctena

- Після успішного запису пацієнт отримує підтвердження на електронну пошту та SMS-нагадування перед запланованим візитом, що допомагає уникнути неявок.

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

- Пацієнти можуть переглядати свої майбутні та минулі записи, а також скасовувати або переносити їх відповідно до політики лікаря.

- Doctena надає лікарям онлайн-календар для ефективного управління своїм розкладом, визначення доступних годин прийому та уникнення подвійних бронювань.

- Система автоматично обробляє онлайн-записи пацієнтів, зменшуючи адміністративне навантаження на персонал реєстратури.

- Присутність на платформі Doctena підвищує видимість медичного працівника для нових пацієнтів, які шукають спеціаліста в їхньому регіоні.

- Платформа може надавати інструменти для обміну повідомленнями з пацієнтами з питань, пов'язаних з їхніми записами.

- Лікарі можуть отримувати доступ до статистики щодо записів та інших ключових показників своєї практики.

Doctena є європейською платформою і представлена в кількох країнах, включаючи Бельгію, Люксембург, Нідерланди, Швейцарію, Німеччину та Австрію.

Doctena, як правило, використовує модель платної підписки для медичних працівників та закладів, які отримують доступ до повного набору функцій платформи. Для пацієнтів користування платформою є безкоштовним.

Doctena та Doctolib є схожими платформами за своєю основною функціональністю, спрямованою на спрощення запису до лікарів. Обидві надають зручний інтерфейс для пацієнтів та інструменти для оптимізації роботи медичних працівників. Основні відмінності можуть полягати в географічному охопленні, специфічних функціях, інтеграціях з іншими системами та деталях користувацького інтерфейсу. Doctolib має ширше поширення в деяких країнах (наприклад, Франція), тоді як Doctena має сильну присутність в інших (наприклад, Бельгія та Люксембург).

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

У підсумку, Doctena є ефективною онлайн-платформою, яка полегшує процес пошуку та запису на прийом до лікарів, приносячи користь як пацієнтам, так і медичним працівникам шляхом підвищення доступності, зручності та ефективності медичних послуг.

Висновки до розділу

В даному розділі, було виявлено актуальну потребу у створенні мобільних рішень, які забезпечують доступ до управлінської інформації в режимі реального часу, враховуючи сучасні виклики у сфері охорони здоров'я. Аналіз протоколу HL7, з особливим акцентом на повідомлення ADT, дав змогу визначити ключові інформаційні потоки, що можуть бути інтегровані у функціонал мобільного додатку, включаючи структуру та типи повідомлень, які підтримують обмін адміністративними даними між системами.

Також було визначено мету проєкту – створення зручної та функціональної Android-аплікації для моніторингу ефективності роботи медичного закладу, а також сформульовано вимоги до її основних компонентів. Особливу увагу приділено показникам ефективності лікарні, які можуть бути відображені в інтерфейсі додатку, що сприятиме прийняттю обґрунтованих управлінських рішень. Окремо розглянуто існуючі інформаційні системи управління та інструменти інноватизації процесів у медичних установах, включно з популярними платформами Doctolib та Doctena. Їх аналіз дозволив виокремити функціональні можливості, які можуть бути адаптовані або вдосконалені у власному рішенні. Таким чином, на основі проведеного аналізу, сформовано чітке технічне та концептуальне підґрунтя для проєктування та реалізації Android-додатку, що відповідатиме сучасним вимогам цифровізації процесів у сфері медичного менеджменту.

					БР.ІП – 04.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

РОЗДІЛ 2. АЛГОРИТМІЧНА РЕАЛІЗАЦІЯ ANDROID АПЛІКАЦІЇ ТА ВИБІР ЗАСОБІВ РОЗРОБКИ

2.1. Опис структури Android

Android був розроблений Річем Майнером, Ніком Сірсом, Енді Рубіном та Крісом Вайтом у Пало-Альто, Каліфорнія, у жовтні 2003 року. 17 серпня 2005 року Google придбала Android і створила мобільний пристрій, який працює на базі ядра Linux, групою осіб під назвою команда Рубін. Google гарантувала надати адаптовану, оновлювану та безкоштовну систему платформи для користувачів. Це саме тоді Android був оновлений, і зараз Android — це операційна система, розроблена для смартфонів, звичайних ПК, планшетів та інших електронних пристроїв [8].

Зараз Android — це найпопулярніша операційна система на ринку. Згідно з продажами 2024 року, програмне забезпечення Android має найбільшу кількість продажів порівняно з iOS, Microsoft Windows та Mac OS. Більше трьох мільйонів додатків Android було випущено Google Play Store, і близько двісті п'ятдесяти мільярдів додатків було завантажено станом на липень 2024 року. Згідно з опитуванням, близько 71 відсотків мобільних розробників використовують платформу Android для створення своїх додатків [8].

Android є популярною архітектурою програмного забезпечення з відкритим вихідним кодом, розробленою Open Handset Alliance [8], яка наразі орієнтована на мобільні пристрої, такі як смартфони та планшетні комп'ютери. Архітектура програмного забезпечення Android (рис. 2.1) поділяється на п'ять рівнів: Програми, Фреймворк програм, Бібліотеки, Середовище виконання Android та Ядро Linux [9]. Верхній рівень, Програми, надає доступ до набору основних програм. Рівень Фреймворк програм реалізує програмний фреймворк, який перебирає функції, що

					БР.ІП – 04.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

використовуються існуючими програмами. Усі доступні бібліотеки написані на C/C++ і викликаються через інтерфейс Java. Середовище виконання Android складається з набору основних бібліотек та віртуальної машини Dalvik. Найнижчий рівень — це ядро Linux, яке забезпечує взаємодію між верхніми рівнями за допомогою драйверів пристроїв [9, 10].

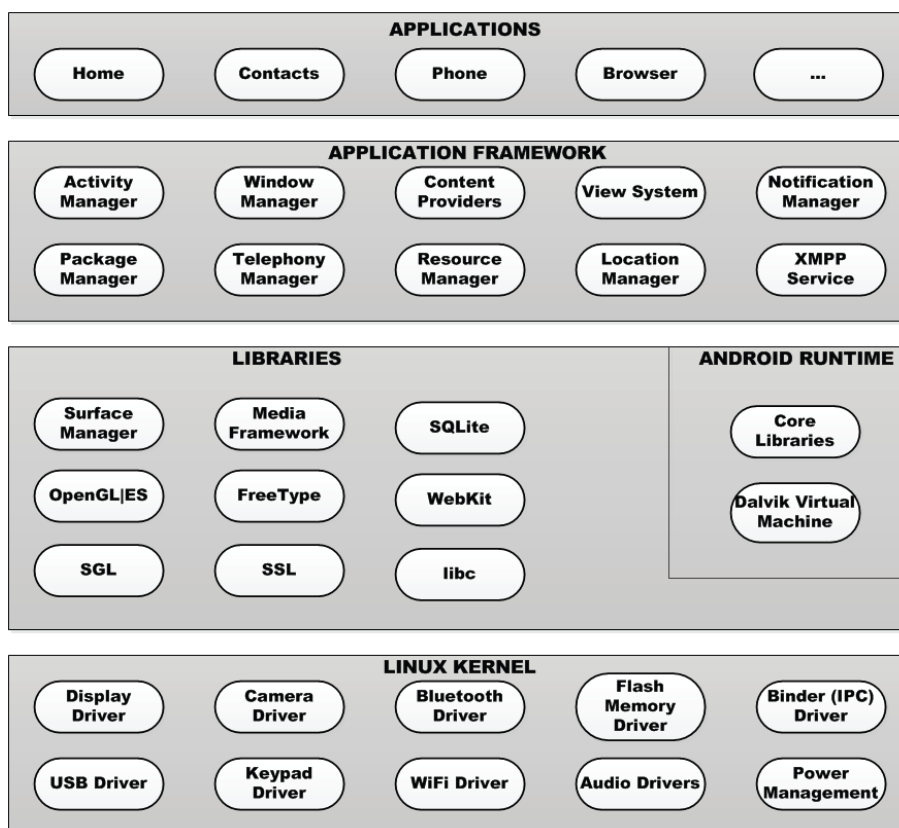


Рисунок 2.1 - Архітектура операційних систем Android

Інструмент програмного забезпечення, такий як Android Studio, необхідний для розробки додатків Android, є безкоштовним інструментом з відкритим вихідним кодом і доступним для завантаження широкому загалу. Це зіграло ключову роль у популяризації та домінуванні Android серед своїх конкурентів.

Android Studio — це офіційне інтегроване середовище розробки (IDE) для розробки додатків під операційну систему Android. Воно базується на

потужній платформі IntelliJ IDEA від JetBrains і спеціально розроблене для потреб Android-розробників.

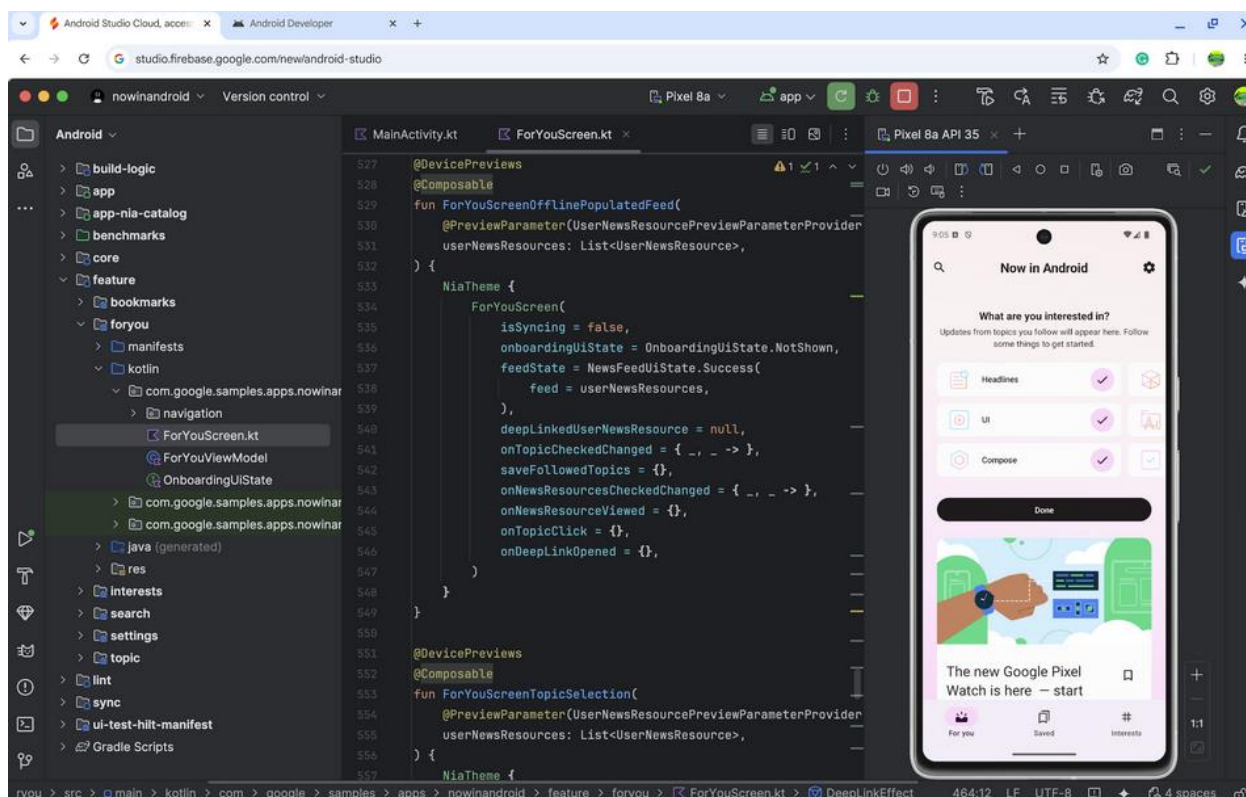


Рисунок 2.2 - Android Studio Cloud

Android Studio є комплексним та потужним інструментом, який надає розробникам все необхідне для створення якісних та функціональних Android-додатків. Його інтегроване середовище, розширені можливості редагування коду, потужні інструменти налагодження та тестування, а також інтеграція з іншими інструментами роблять процес розробки більш ефективним та зручним.

2.2. RESTful вебсервіс. Формат JSON

Вебсервіс визначається як послуга, надавана електронним пристроєм для спілкування з іншим електронним пристроєм через Всесвітню павутину. Він включає веб-технології, такі як протокол передачі гіпертексту (HTTP), і

					БР.ІП – 04.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

призначений в основному для забезпечення спілкування між машинами, такими як передача конкретних форматів файлів, таких як JSON та XML. Загалом, вебсервіс зазвичай надає об'єктно-орієнтований веб-інтерфейс до бази даних сервера, який, наприклад, використовується іншим вебсервером або мобільним додатком, який надає користувацький інтерфейс кінцевому користувачеві [5].

RESTful вебсервіс — це вебсервіс, розроблений за принципами архітектурного стилю REST (Representational State Transfer). REST — це не протокол, а набір обмежень та рекомендацій для створення масштабованих, гнучких та простих вебсервісів.

Основна ідея REST полягає в тому, щоб розглядати кожен елемент даних (наприклад, користувача, товар, замовлення) як ресурс, який можна ідентифікувати за допомогою унікального URI (Uniform Resource Identifier), найчастіше URL-адреси. Клієнтські програми взаємодіють з цими ресурсами, використовуючи стандартні HTTP-методи для виконання різних операцій.

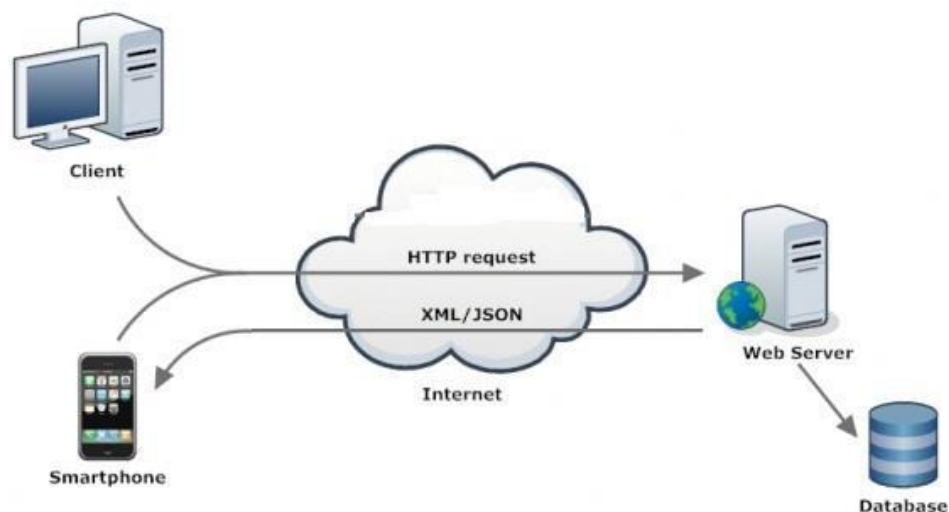


Рисунок 2.3 - RESTful вебсервіс

RESTful вебсервіси є широко поширеним підходом для створення API (Application Programming Interface) для веб-додатків, мобільних додатків та

інших розподілених систем. Вони забезпечують ефективний та стандартизований спосіб взаємодії між різними сервісами.

Вебсервіси, які використовують архітектуру Representational State Transfer (REST), називаються RESTful вебсервісами. На практиці все є ресурсом у архітектурі REST, і воно легке, масштабоване та підтримуване. Уніфіковані ідентифікатори ресурсів (URIs) використовуються для доступу до ресурсів, таких як записи бази даних у REST вебсервісах. Він використовує HTTP для спілкування з клієнтами, такими як телефони Android тощо. Для представлення ресурсів в різних форматах, таких як XML, простий текст, JSON тощо, ресурси зазвичай ізольовані від їхнього початкового представлення. У цьому додатку використовується формат JSON.

JSON (JavaScript Object Notation) — це легкий, людиночитабельний формат обміну даними. Він базується на підмножині синтаксису JavaScript, але є незалежним від мови програмування і використовується багатьма мовами. JSON часто застосовується для передачі даних між вебсервером та вебзастосунком, а також для зберігання конфігураційних файлів та інших структурованих даних. Завдяки своїй простоті та універсальності, JSON став стандартом де-факто для обміну даними в багатьох вебтехнологіях та API. Він легко парситься та генерується більшістю сучасних мов програмування, що робить його зручним вибором для розробників.

Основні характеристики формату JSON:

- JSON має дуже простий синтаксис, що робить його легким для розуміння та парсингу (розбору).
- Завдяки своїй структурі, дані у форматі JSON легко читати та редагувати вручну.
- JSON легко обробляється комп'ютерами та парсерами в різних мовах програмування.

					БР.ІП – 04.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

- JSON не прив'язаний до конкретної мови програмування, що робить його ідеальним форматом для обміну даними між різними системами.

- JSON підтримує прості типи даних, такі як рядки, числа, булеві значення, а також структуровані типи даних: об'єкти та масиви.

Об'єкти (Objects) в JSON представляються у вигляді пари ключ-значення, укладеної у фігурні дужки {}. Ключ завжди є рядком у подвійних лапках ("). Значення може бути одним із підтримуваних типів даних (рядок, число, булеве значення, інший об'єкт, масив або null). Пари ключ-значення розділяються комами.

JavaScript Object Notation (JSON) — це формат обміну даними [4]. Він легкий і зручний для аналізу значень JSON додатками Android та відображення результатів на інтерфейсі користувача. Формат JSON представлений на рисунку 2.4.

```
{
  "dial": [
    {"diagnosis_description": "Allergies", "count_des": 1},
    {"diagnosis_description": "Amputation of leg below knee left traumatic subsequent",
    {"diagnosis_description": "appendicitis", "count_des": 5}
  ]
}
```

Рисунок 2.4 – Представлення даних в форматі JSON

Формат JSON вище містить об'єкт dial, який є масивом, що містить три об'єкти. Кожен об'єкт має два записи, такі як diagnosis_description і count_des.

PHP

Hypertext Preprocessor (PHP) — це мова скриптів, яка використовується для створення динамічних вебсторінок та інтернет-застосунків [12]. Файли з розширенням .php називаються файлами PHP і можуть містити текст і скрипти. Ці файли обробляються вебсервером, а потім повертаються необхідні значення на вебсторінку.

У цьому додатку файли PHP використовуються для генерування представлення JSON значень, які отримуються з бази даних за допомогою мови запитів Sequential Query Language (SQL).

2.3. Визначення пакетів, використаних у проєкті

Нижче наведені деякі з основних пакетів, які в основному використовуються для запуску цього додатку. Ці пакети також містять файли класів, які використовуються для реалізації цього додатку.

```
import android.os.AsyncTask
```

Допомагає виконувати операції в фоновому режимі та дозволяє легко використовувати інтерфейс користувача без необхідності маніпулювати потоками.

```
import org.json.JSONArray
```

Містить інформацію про масив JSON, який складається з щільної індексованої послідовності значень.

```
import org.json.JSONObject
```

Відповідає за збереження інформації про об'єкт JSON, який є модифікованим набором імен або значень карти, і потрібен для аналізу та відображення на інтерфейсі користувача.

```
import java.net.HttpURLConnection
```

Клас підключення URL дозволяє спілкуватися з URL через мережу. Він допомагає клієнту, наприклад, додатку Android, читати дані JSON з URL за допомогою з'єднання HTTP.

```
import android.widget.DatePicker
```

Надає віджет для вибору дати, який називається селектор дати. Дату можна вибрати за допомогою року, місяця та дня, присутніх у календарному вигляді селектора дати.

```
import android.content.Intent
```

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

Відповідає за зв'язування різних дій, присутніх у додатку, таких як зв'язування між екранами входу в систему та показників.

```
import java.io.InputStreamReader
```

Використовується для читання даних у байтах і декодування даних у кодуванні символів за замовчуванням або вказаному кодуванні символів. У цьому додатку дані JSON читаються та конвертуються у формат рядка для оновлення інтерфейсу користувача.

```
import android.widget.TextView
```

Забезпечує відображення тексту користувачеві за допомогою текстового поля та дозволяє користувачеві копіювати значення текстового поля. Він не дозволяє редагувати текст. Однак підклас редагування тексту можна використовувати для редагування тексту, наприклад, введення пароля тощо.

```
import android.widget.ImageView
```

Допомагає відображати довільне зображення, наприклад, іконку зображення. Цей клас також може завантажувати зображення з ресурсів або постачальників контенту тощо.

```
import android.widget.Toast
```

Містить клас toast, який надає повідомлення з коротким текстом для користувачів. Тривалість повідомлення можна налаштувати за допомогою функцій короткого та довгого терміну. У цьому додатку клас toast використовується для відображення повідомлень про помилку входу в систему.

2.4. Реалізація Android аплікації для медичного закладу

У цьому додатку потрібно декодувати повідомлення HL7 ADT у зрозумілий формат та зберегти їх у базу даних SQL за допомогою мови програмування Java та з'єднання JDBC через інструмент IDE Eclipse.

					БР.ІП – 04.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

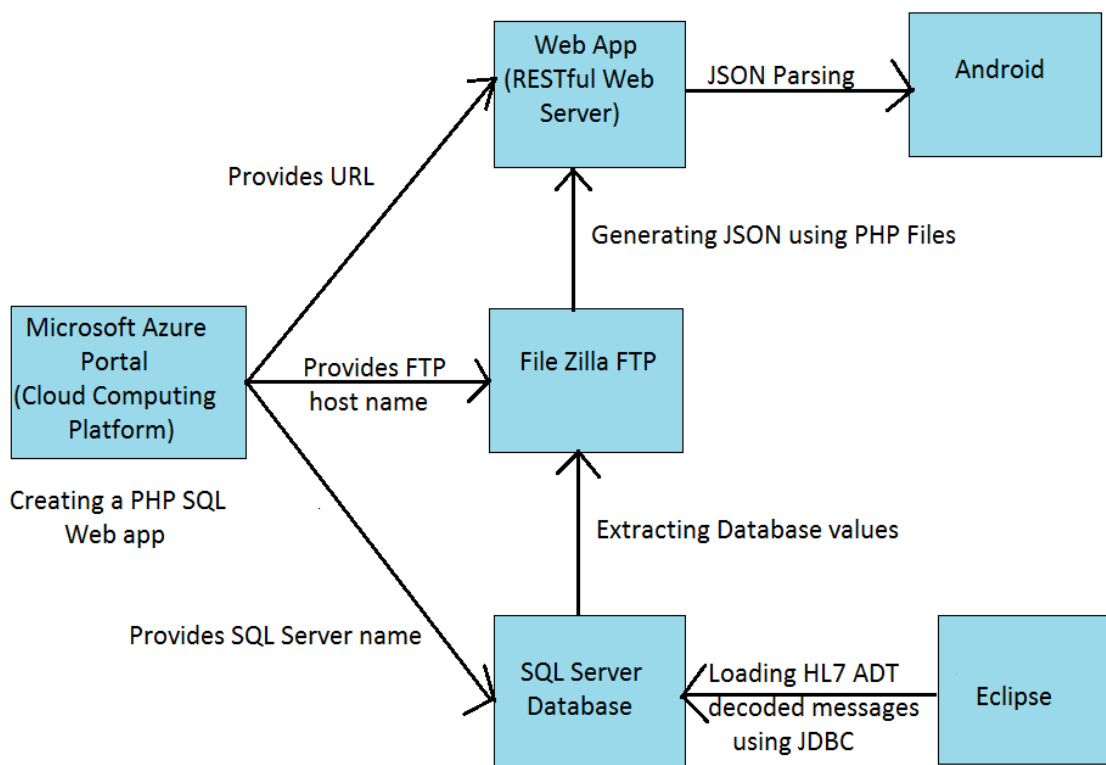


Рисунок 2.5 – Огляд реалізації

Додатки Android не можуть безпосередньо отримувати дані з бази даних. Тому для отримання даних потрібен додаток для підключення до зовнішніх REST вебсервісів. Цей додаток використовує портал Microsoft Azure, який є хмарною обчислювальною платформою, розробленою Microsoft для створення, публікації та управління різними додатками та послугами за допомогою глобальної мережі дата-центрів, керованих Microsoft [11]. Основна перевага використання хмарної обчислювальної платформи дозволяє користувачам отримувати доступ до даних дистанційно, а не через локальний хост. Цей додаток вимагає створення веб-додатку PHP SQL, який діє як REST вебсервіс. На практиці розробнику потрібно сплатити за створення вебдодатку за допомогою Microsoft Azure. Оскільки я розробив цей додаток спільно з навчальною ціллю, то мені надали реквізити для веб-додатку PHP SQL, який можна розгорнути за допомогою протоколу передачі файлів (FTP).

Ці дані включають ім'я сервера для SQL Server Management Studio, яке зберігає декодовані повідомлення HL7 ADT у таблицях, URL для вебдодатку, який містить записи з бази даних у форматі JSON, та ім'я хоста для FileZilla FTP, яке використовується для розгортання файлів PHP у вебдодаток для генерування масиву JSON з значень, отриманих з бази даних.

Наступним кроком є підключення додатку Android до вебдодатку для доступу до даних JSON та оновлення інтерфейсу користувача. Це можна зробити за допомогою асинхронної задачі (AsyncTask) в Android.

2.4.1. Декодування тестових повідомлень HL7 ADT

Нижче наведено код Java для декодування тестових повідомлень HL7 ADT та завантаження їх у базу даних SQL за допомогою інструменту IDE Eclipse.

Лістинг 2.1. Код для декодування тестових повідомлень HL7

```
public class HL7_decode {
    // Додавання реквізитів бази даних для з'єднання jdbc
    static String db = "назва бази даних, створена в sql server";
    static String url = "jdbc:sqlserver://servername;databaseName=" + db + ";";
    static String userid = "username";
    static String password = "password";
    static Connection con = null;

    public static void main(String args[]) throws SQLException, ClassNotFoundException, IOException {
        // Підключення до бази даних за допомогою jdbc
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
        con = DriverManager.getConnection(url, userid, password);

        // Оголошення необхідних полів повідомлення ADT, тут оголошено лише одне поле
        String diagnosis_description = "DG1-4";

        // Читання тестових повідомлень ADT
        BufferedReader br = new BufferedReader(new FileReader("path of ADT test messages.txt"));
        String line = "";
        int i = 0, j = 0, new_line = 0, msg_count = 0;

        // Логіка для вставки значень у таблиці бази даних після читання тестового вхідного потіку
        while ((line = br.readLine()) != null) {
            j++;
            String[] fields = line.split("\\|");
            if (fields[0].equals("MSH"))
```

					БР.ІП – 04.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

```

        if (msg_count > 0 || (fields[0].equals("MSH") && fields[8].substring(4, 7).equa
            switch (fields[0]) {
                case "MSH":
                    i++;
                    msg_count++;
                    if (msg_count != 1) {
                        String inst = "INSERT INTO [dbo].[table name]([DIAGNOSIS_DESCRIP
                            "VALUES(" + diagnosis_description + " )";
                        Statement s = con.createStatement();
                        s.executeUpdate(inst);
                    }
                    new_line = 1;
                    break;
                case "DG1":
                    i++;
                    diagnosis_description = fields[4];
                    break;
                default:
            }
        }
    }
}

```

2.4.2. Реалізація PHP для отримання даних

Файли PHP містять запити SQL для отримання необхідних даних з бази даних. Вони використовують функцію `json_encode` для повернення представлення значень JSON, отриманих з бази даних за допомогою запиту SQL. Зазвичай для виконання файлів PHP потрібен вебсервер, такий як Tomcat або Apache. Вебдодаток, створений за допомогою хмарної обчислювальної платформи, діє як вебсервер за замовчуванням. Тому цей додаток вимагає лише FileZilla FTP, який використовує протокол передачі файлів, для розгортання файлів PHP. FileZilla — це вільний, кросплатформний FTP (File Transfer Protocol), SFTP (SSH File Transfer Protocol) та FTPS (FTP over SSL/TLS) клієнт. Тобто, це програма, яка дозволяє підключатися до віддалених серверів в Інтернеті або локальній мережі для передачі файлів між вашим комп'ютером і цими серверами. Кожен показник цього додатку вимагає окремого файлу PHP, і після розгортання до нього можна отримати доступ за допомогою URL.

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

Наприклад, розглянемо показник діагнозу. Код PHP для цього показника наведено нижче.

Лістинг 2.2. Код PHP для показника діагнозу

```
<?php
// Повертає масив під назвою dia1 на вебсторінку
$return_arr = array();
$return_arr['dia1'] = array();
try {
    // Додавання реквізитів бази даних
    $serverName = "tcp:server name";
    $connectionOptions = array("Database" => "назва бази даних", "Uid" => "username", "PWD" =>
    // Підключення до бази даних sql server
    $conn = sqlsrv_connect($serverName, $connectionOptions);
    // sql запит для отримання значень з бази даних тут деталі показника діагнозу
    $tsql = "select count(DIAGNOSIS_DES) as 'COUNT_DES', DIAGNOSIS_DES from dbo.PATIENT1 GROUP BY DIAGNOSIS_DES";
    $getProducts = sqlsrv_query($conn, $tsql);
    while ($row = sqlsrv_fetch_array($getProducts, SQLSRV_FETCH_ASSOC)) {
        $row_array['COUNT_DES'] = $row['COUNT_DES'];
        $row_array['DIAGNOSIS_DES'] = $row['DIAGNOSIS_DES'];
        array_push($return_arr['dia1'], $row_array);
    }
    // Повертає представлення JSON значень бази даних у масиві під назвою dia
    echo json_encode($return_arr);
    sqlsrv_free_stmt($getProducts);
    sqlsrv_close($conn);
}
?>
```

Файли PHP потрібно розгорнути в вебдодаток для генерування масиву JSON за допомогою FileZilla FTP. Потім додатки Android можуть отримувати доступ до даних JSON, аналізуючи та відображаючи значення на інтерфейсі користувача. Після завершення розгортання буде згенеровано масив JSON.

Подібним чином потрібно згенерувати значення JSON для інших показників за допомогою файлів PHP. Тому файли PHP кожного показника розгортаються окремо за допомогою FileZilla FTP. Розробник може переглянути згенерований JSON на вебдодатку, просто ввівши "URL/filename.php". Наприклад, введення URL/admit.php у веббраузері дозволяє розробнику переглянути відповідну вебсторінку, яка містить інформацію про прийом у форматі JSON.

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновки до розділу

В розділі описано архітектуру Android-додатку, включно з основними компонентами (Activity, Service, BroadcastReceiver, Content Provider), що забезпечують належну модульність та масштабованість системи. Вибір RESTful вебсервісу та використання формату обміну даними JSON обґрунтовано як оптимальне рішення для взаємодії з віддаленими медичними серверами, зважаючи на простоту інтеграції, легкість обробки та ефективність.

У підрозділі, присвяченому структуруванню проєкту, представлено деталізацію пакетів і логіку їх використання, що дозволяє забезпечити прозорість, повторне використання коду та зручність супроводу додатку. Це стало фундаментом для побудови надійної та масштабованої архітектури мобільного рішення. Особливу увагу приділено реалізації функціоналу обробки HL7 ADT повідомлень – ключового джерела адміністративної інформації про пацієнтів. Було реалізовано модуль декодування тестових повідомлень та адаптовано серверну частину на PHP для отримання необхідних даних, що забезпечує злагоджену взаємодію між клієнтською мобільною частиною та сервером медичного закладу.

Загалом, у межах цього розділу здійснено техніко-технологічне обґрунтування рішень, що лягли в основу проєктування й реалізації Android-додатку. Це створює передумови для розробки надійного, масштабованого й адаптивного інструменту цифрового управління в медичній сфері.

					БР.ІП – 04.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ANDROID-АПЛІКАЦІЇ АДМІНІСТРУВАННЯ РОБОТИ МЕДИЧНОГО ЗАКЛАДУ

3.1. Реалізація структури Android-аплікації

Основна папка цього додатку містить наступні підпапки:

- .idea
- Модуль проєкту (app)
- Gradle
- Зовнішні бібліотеки

Папка .idea містить файли конфігурації (.xml) проєкту. Кожен файл .xml містить лише частину конфігураційних даних, пов'язаних з конкретною функціональною областю, яка визначається назвою файлу, таким як compiler.xml, modules.xml тощо. Ці файли містять інформацію про назви та розташування компонентів модулів проєкту, параметри компіляції тощо. Більшість файлів у .idea містять інформацію про сам проєкт.

Папка Gradle містить jar-обгортку системи побудови Gradle, і цей jar використовується для зв'язку Android Studio з Gradle, встановленим у операційній системі. Операційна система Windows використовується для цього додатку.

Зовнішні бібліотеки не є папкою, але містять інформацію про посилавні бібліотеки та цільову платформу SDK у Android Studio. У моєму додатку Android API 23 використовується як цільова платформа.

Папка app — це фактична папка проєкту, яка містить код додатку в вигляді файлів java. Папка додатку містить підпапки build, libs та src. Папка build зберігає повний вихід процесу створення, такий як classes.dex, скомпільовані класи та ресурси тощо. Папка libs містить бібліотеки та файли .jar. Папка src містить код додатку Android у підпапці main та скрипти

					БР.ІП – 04.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57

тестування одиниць у підпапці androidTest. Папка main містить дві підпапки: java та res.

Папка res містить підпапки, такі як drawable, layout, mipmap тощо. Папка layout містить файли xml, які в основному використовуються для створення інтерфейсу користувача. Кожен макет має відповідний клас java. Папка mipmap містить іконки додатку, які можна побачити на панелі інструментів додатку. Папка java відповідає за всі коди Java, включаючи методи, розроблені розробником додатку, і ці класи Java діють як бекенд-код для зв'язку макетів інтерфейсу користувача, присутніх у res, та для отримання значень з RESTful вебсервісів.

3.1.1. Опис розроблених класів

Файли java в моєму додатку наведені нижче:

MainActivity.java

Цей клас використовується для розробки вітального екрану цього додатку. Натискання на вітальний екран перенаправляє користувача на екран входу в систему.

Login.java

Цей клас використовується для розробки функцій, натискання на кнопку входу в систему призводить до переходу на екран показників, якщо введені правильні реквізити, а неправильна аутентифікація викликає повідомлення про помилку.

Metrics.java

Натискання на певну іконку показника на екрані показників перенаправляє користувача на відповідний екран. Ця функція реалізована за допомогою цього класу java.

Admit.java

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

Інтерфейс користувача цього показника містить календар для вибору дати, а цей клас java допомагає розробнику отримати кількість пацієнтів, прийнятих за певну дату.

Discharge.java

Цей клас java використовується для відображення кількості пацієнтів, виписаних за певну дату, після вибору дати в календарі календаря.

Census.java

Загальна кількість пацієнтів на день відображається за допомогою цього класу java.

Insurance.java

Список назв страхових компаній та кількість пацієнтів можна реалізувати за допомогою цього класу java.

Length.java

Середня тривалість перебування пацієнта можна показати на текстовому полі за допомогою цього класу java.

Demographics.java

Цей клас використовується для малювання стовпчастої діаграми.

Diagnosis.java

Цей клас використовується для відображення назв захворювань на текстовому полі.

3.1.2. Опис основних інтерфейсів та методів

Основні інтерфейси та методи, присутні в згаданих вище класах java, описані нижче:

OnClickListener

OnClickListener — це інтерфейс, який визначає метод onclick, в якому натискання на певний текст, кнопку або зображення реалізовує визначений метод onclick.

Intent

					БР.ІП – 04.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59

Intent в Android — це абстрактне описання операції для виконання. Він використовується для зв'язку різних дій. У цьому додатку intent використовується як метод onclick, натискання на кнопку, зображення або текст реалізовує метод intent, перехід до іншої дії. Класи Java, такі як Admit, Discharge, Insurance, Census, Metrics, Length, Demographics та Diagnosis, містять OnClickListener, в якому натискання на зображення View, таке як Back, Refresh та Logout, перенаправляє до відповідного екрану. Екран входу в систему використовує кнопку, а вітальний екран використовує текстове поле. Метод onclicklistener разом з методом intent описаний нижче.

Лістинг 3.1. Методи onclicklistener і intent

```

ImageView iv1 = (ImageView) findViewById(R.id.image2);
iv1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View a) {
        Intent b = getIntent();
        finish();
        startActivity(b);
    }
});

```

Вищезгаданий код реалізовує зображення View, а потім призначає OnClickListener створеному View. Коли користувач натискає, OnClickListener викличе метод onclick (метод intent).

AsyncTask

Використовується для виконання операцій у фоновому режимі та публікації результатів на інтерфейсі користувача. Щоб отримати значення масиву JSON, додатки Android повинні підключатися до RESTful вебсервісів за допомогою з'єднань HTTP URL. AsyncTask допомагає отримувати значення з RESTful вебсервісів та оновлювати інтерфейс користувача. Нижче наведені кроки, включені в AsyncTask [2].

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

HTTP URL Connection. Підключення до RESTful вебсервісів за допомогою з'єднання HTTP URL в AsyncTask можна реалізувати за допомогою наведеного нижче коду.

Лістинг 3.2. HTTP URL Connection

```
public class Http extends AsyncTask<Void, String, Integer> {
    @Override
    protected Integer doInBackground(String... params) {
        InputStream input = null;
        HttpURLConnection urlCon = null;
        Integer result = 0;
        try {
            URL url = new URL(params[1]);
            urlCon = (HttpURLConnection) url.openConnection();
            urlCon.setRequestMethod("GET");
            int statusCode = urlCon.getResponseCode();
            if (statusCode == 200) {
                Json parsing(output);
                return result;
            }
        }
    }
}
```

JSON Parsing Цей метод використовується для аналізу даних JSON для отримання необхідних значень записів [4]. Код нижче використовується для аналізу масиву JSON.

Лістинг 3.3. Метод JSON Parsing

```
private void jsonParsing(String Result) {
    try {
        JSONObject output = new JSONObject(Result);
        JSONArray A = output.optJSONArray("name of the Json array");
        for (int j = 0; j < A.length(); j++) {
            JSONObject O = A.optJSONObject(j);
            String rec = O.optString("record name");
        }
    }
}
```

Метод Post Execute використовується, коли завантаження значень з вебсторінки завершено, і інтерфейс користувача потрібно оновити. Нижче наведено код, використаний для відображення значення JSON на текстовому полі.

Лістинг 3.4. Метод Post Execute

```
public void onPostExecute(Integer Result) {
    if (Result == 1) {
        TextView.setText(rec);
    } else {
        System.out.println("failed to fetch");
    }
}
```

3.2. Розробка інтерфейсу користувача

Нижче наведені скріншоти екрана, зроблені під час тестування цього додатка на різних рівнях, які виконувалися успішно без помилок.

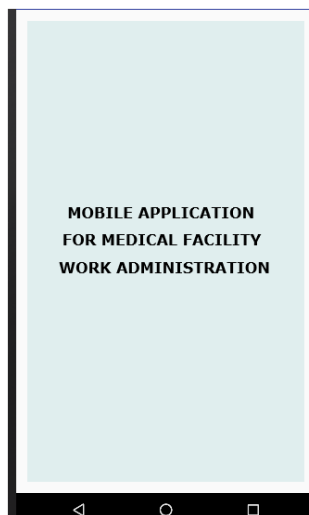


Рисунок 3.1 – Вікно початку роботи

На рисунку 3.1 подано завантажувальний екран мобільного додатку, призначеного для адміністрування роботи медичного закладу. Він містить назву додатку і стандартні елементи навігації Android.

					БР.ІП – 04.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

Дотик до екрану призведе до відкриття сторінки входу в систему, що показано на рисунку 3.2.

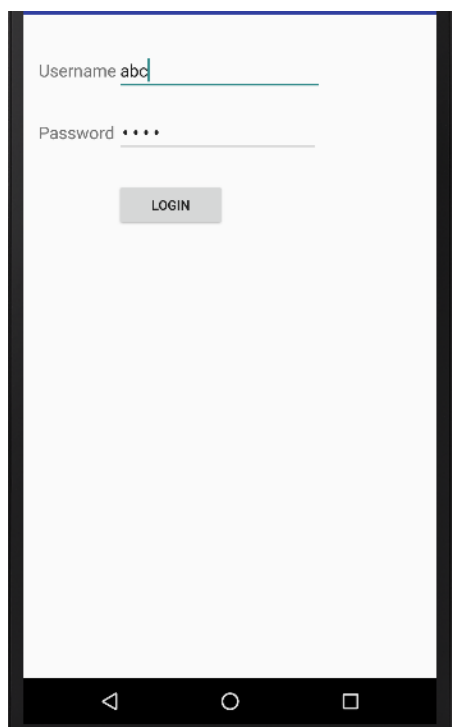


Рисунок 3.2 – Вікно авторизації

Наведений вище скріншот є сторінкою входу, де користувачеві необхідно ввести облікові дані для доступу до застосунку. Успішна автентифікація призведе до відкриття екрану з показниками. Повідомлення про помилку з'явиться, якщо користувач введе недійсні облікові дані, як показано на рисунку 3.3.

У нижній частині екрану відображається повідомлення у формі закругленого прямокутника з текстом "Invalid username or password!" (Неправильне ім'я користувача або пароль!). Це повідомлення вказує на те, що була спроба входу з невірними обліковими даними.

У самому низу екрану знаходиться стандартна панель навігації Android з трьома піктограмами: трикутником ліворуч ("Назад"), колом посередині ("Додому") та квадратом праворуч ("Останні програми").

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дата		

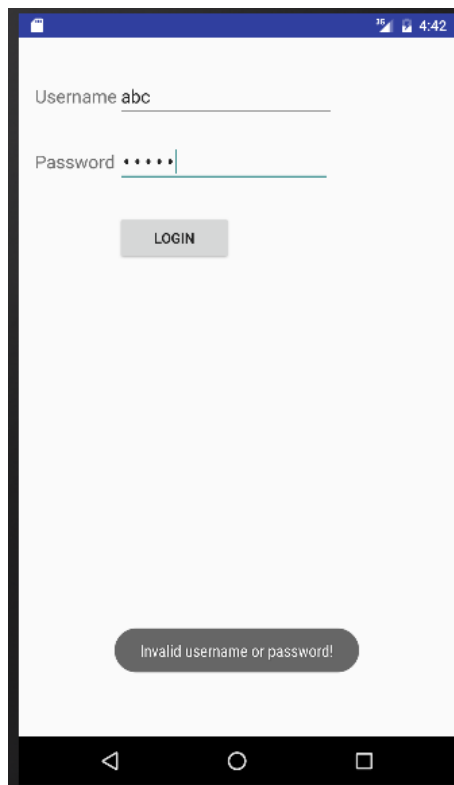


Рисунок 3.3 – Вікно невдалої спроби входу через введення неправильних імені користувача або пароля

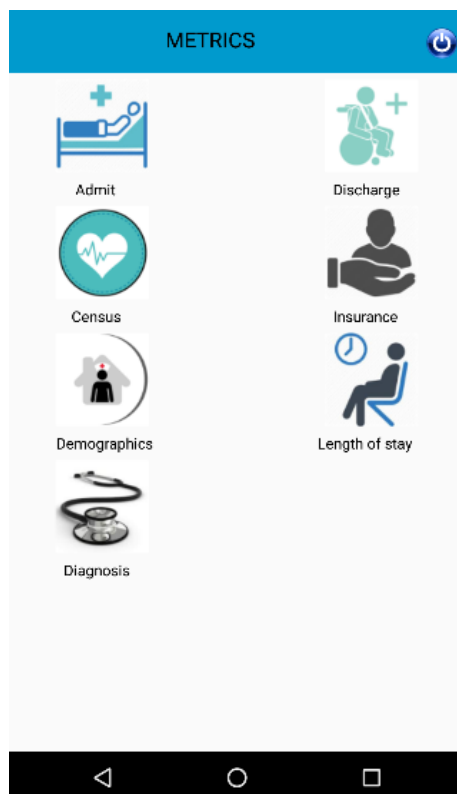


Рисунок 3.4 – Вікно з представленням метрик ефективності лікарні

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

Цей скріншот (рис. 3.4) показує сторінку з показниками. Тут ми бачимо метрики ефективності лікарні, натискання на конкретний показник відкриває відповідну сторінку.

Основна частина екрану має білий фон та містить шість великих іконок, розташованих попарно, з підписами, що представляють різні показники ефективності лікарні.

В першому ряді представлено:

- Зліва: Іконка ліжка з хрестом (+) та силуетом пацієнта на ньому. Під іконкою розміщено підпис "Admit" (Прийняття).

- Справа: Іконка силуету людини в інвалідному візку з хрестом (+). Під іконкою розміщено підпис "Discharge" (Виписка).

В другому ряді представлено:

- Зліва: Кругла іконка із зображенням кардіограми (пульсу) всередині. Під іконкою розміщено підпис "Census" (Кількість пацієнтів).

- Справа: Іконка силуету людини на долоні. Під іконкою розміщено підпис "Insurance" (Страхування).

В третьому ряді представлено:

- Зліва: Іконка будинку з силуетом людини всередині та графіком у вигляді стовпчиків позаду. Під іконкою розміщено підпис "Demographics" (Демографія).

- Справа: Іконка силуету людини, що сидить, з годинником біля неї. Під іконкою розміщено підпис "Length of stay" (Тривалість перебування).

На рисунку 3.5 ми бачимо екран "Прийняття", де вибір конкретної дати відображає кількість пацієнтів, прийнятих у цей день.

Цей скріншот відображає екран, де користувач може вибрати дату в календарі та побачити кількість пацієнтів, прийнятих у цей конкретний день. 7 квітня 2025 року було прийнято 15 пацієнтів на лікування у відповідний заклад.

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

Admit

2025
Thu, Apr 7

< April >

S	M	T	W	T	F	S
					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

Number of Patients Admitted

15

Рисунок 3.5 – Вікно із кількістю прийнятих пацієнтів в певний день

Discharge

2025
Thu, Apr 14

< April >

S	M	T	W	T	F	S
					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

Number of Patients Discharged

16

Рисунок 3.6 - Вікно із кількістю виписаних пацієнтів в певний день

На рисунку 3.6 показано екран "Виписка", де вибір дати відображає кількість пацієнтів, виписаних у цей день.

Рисунок 3.7 описує, як користувач може переглянути загальну кількість пацієнтів після вибору дати з календаря.

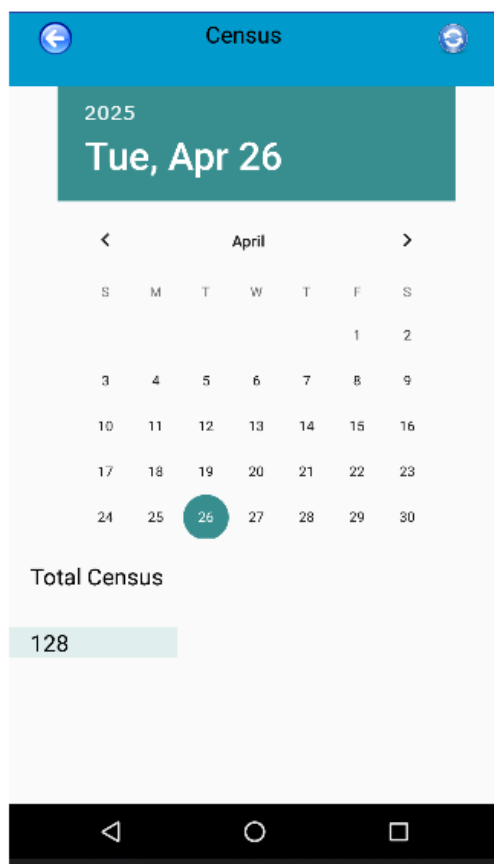


Рисунок 3.7 – Кількість пацієнтів на певну дату

Загалом, цей скріншот відображає екран, де користувач може вибрати дату в календарі та побачити загальну кількість пацієнтів у лікарні на цей конкретний день, в даному випадку загальна кількість пацієнтів становила 128.

На рисунку 3.8 показано деталі страхування, що включають назви страхових компаній та кількість пацієнтів. Під синьою смужкою розташований світло-блакитний заголовок "Insurance Company Name - Number of Patients" (Назва страхової компанії - Кількість пацієнтів).

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

Insurance	
Insurance Company Name - Number of Patients	
AARP	- 5
AETNA	- 200
BLUE CROSS	- 110
BLUE CROSS ANTHEM	- 16
BLUE CROSS CA	- 22
CHESTERFIELD RESOURCES	- 2
CIGNA	- 117
HUMANA MEDICARE HMO	- 40
KAISER	- 24

Рисунок 3.8 – Вікно із даними про стахування пацієнтів

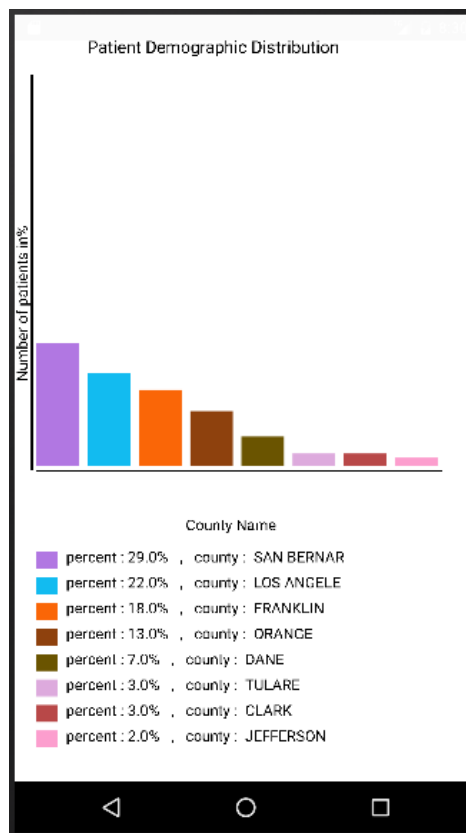


Рисунок 3.9 – Демографічний розподіл пацієнтів

На рисунку 3.9 показано демографічний розподіл пацієнтів у вигляді стовпчастої діаграми, що відображає назву області (країни) та відсоток пацієнтів.

На рисунку 3.10 показано середню тривалість перебування пацієнта в днях.

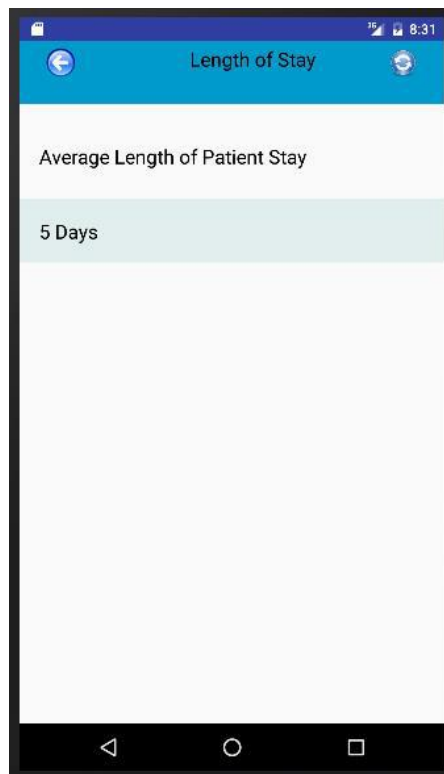


Рисунок 3.10 – Вікно, що показує середню тривалість перебування пацієнта в лікарні

На скріншоті (рис. 3.11) зображено екран "Diagnosis" (Діагноз) мобільного додатку. Основна частина екрану має білий фон та містить інформацію про найпоширеніші та найменш поширені діагностовані захворювання.

Спочатку йде заголовок "Most common diagnosed disease" (Найпоширеніше діагностоване захворювання). Під цим заголовком на світло-сірому прямокутнику вказано "PAIN" (БІЛЬ).

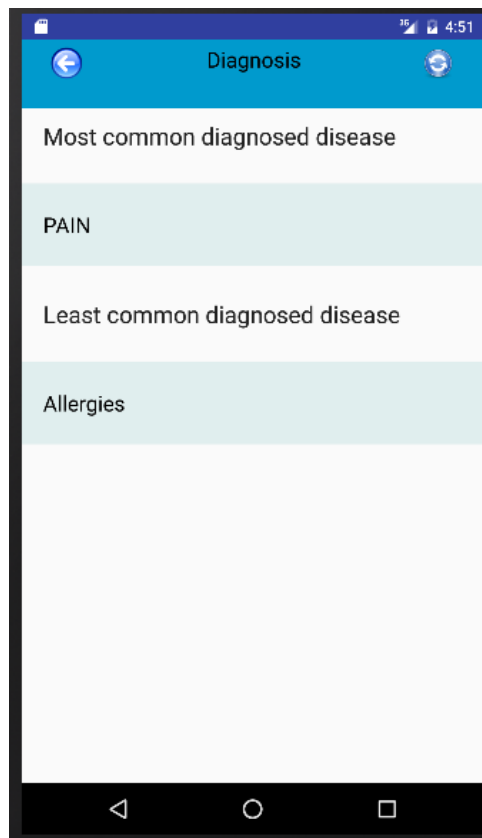


Рисунок 3.11 – Вікно показу захворювань

Далі йде заголовок "Least common diagnosed disease" (Найменш поширене діагностоване захворювання). Під цим заголовком великими літерами написано "Allergies" (АЛЕРГІЇ).

Загалом, цей скріншот відображає інформацію про найчастіше та найрідше зустрічаючіся діагнози серед пацієнтів медичного закладу. Найпоширенішим діагнозом є "БІЛЬ", а найменш поширеним - "АЛЕРГІЇ".

3.3. Опис процесу тестування мобільної аплікації

Тестування є невід'ємною частиною проєкту, яка використовується для забезпечення якості програмного забезпечення. Тестування — це процес виконання додатку для виявлення помилок програмного забезпечення. Тестування включає розробку різних методів тестових випадків, які

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

виконуються за допомогою послідовності кроків для успішної розробки програмного забезпечення.

Тестування — це процес перевірки та підтвердження всіх компонентів програмного забезпечення. Основна мета процесу тестування — переконатися, що програмне забезпечення працює правильно для заданого набору можливих вхідних даних. Тестування дозволяє розробнику виявляти помилки, присутні в додатку, і виправляти їх заздалегідь, щоб користувачі не мали проблем під час використання додатку.

Різні типи тестування потрібні для виконання на різних рівнях, щоб забезпечити якість програмного забезпечення.

Модульне тестування дозволяє тестувальнику тестувати кожний компонент, присутній у додатку/програмному забезпеченні. Наприклад, модульне тестування комп'ютера включає перевірку операційної системи, арифметичного та логічного блоку, клавіатури, миші тощо, щоб переконатися, що кожен компонент працює правильно.

Інтеграційне тестування дозволяє розробнику тестувати всю систему після інтеграції кожного модуля додатку як єдиного цілого. Цей рівень тестування дозволяє розробнику знайти рішення для всіх можливих помилок.

Системне тестування забезпечує, що додаток або програмне забезпечення відповідає системним вимогам після тестування з усіма очікуваними вимогами користувачів.

Тестування користувачами — це критично важливий рівень тестування в процесі тестування, оскільки він включає значну участь розробника з кінцевими користувачами, щоб переконатися, що програмне забезпечення/додаток задовольняє функціональні вимоги користувачів, і також переконатися, що кінцева розроблена програма забезпечує легкий доступ і контроль над додатком для користувачів.

Розроблений додаток було протестовано на різних рівнях, виконуючи різні тести, згадані вище, щоб перевірити, чи працює мій мобільний додаток

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						71
Змн.	Арк.	№ докум.	Підпис	Дата		

правильно. Спочатку було проведено модульне тестування, щоб визначити програмний потік додатку. Потім проведено тестування, яке включає перевірку функціональності кожної функції окремо, щоб дізнатися про її продуктивність, наприклад, у моєму додатку перевірка того, чи правильно отримує JSON дані кожен показник. Після завершення написання всього коду було перевірено чи відображають усі показники додатку ті самі дані, що і в базі даних.

Також додаток було протестовано кілька разів після оновлення бази даних, і він відображав лише оновлені значення. Під час тестування також було введено не вірні дані, додаток не прийняв їх і показав повідомлення про помилку.

В подальшому цей додаток можна покращити за допомогою інтерфейсних двигунів, які зберігають декодовані повідомлення HL7. Інтерфейсні двигуни можуть бути синхронізовані з базою даних. Додаток можна розширити для відображення нових показників, таких як графік операцій, потреби в персоналі тощо.

Отже, цей додаток є корисним для керівництва медичних закладів і допомагає їм щодня переглядати, моніторити та аналізувати показники ефективності лікарні, такі як щоденний перепис, середня тривалість перебування пацієнта в лікарні, кількість пацієнтів, прийнятих і виписаних за певну дату, деталі страхування пацієнтів, такі як назва страхової компанії та кількість пацієнтів, які використовують відповідне страхування, деталі діагнозу, такі як найменш і найбільш поширені діагностовані захворювання, та демографічний розподіл пацієнтів, такі як відсоток пацієнтів з кожної області, без використання паперових звітів.

Ця система може покращити роботу керівництва медичних закладів, надаючи операційні дані в портативному режимі з легкістю доступу, і допоможе їм приймати ефективні рішення, необхідні для покращення якості послуг догляду за пацієнтами.

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновки до розділу

В розділі було реалізовано структуру додатку відповідно до принципів чистої архітектури та сучасних підходів Android-розробки. Класи були структуровані за функціональним призначенням: для обробки даних, відображення інтерфейсу, комунікації з сервером тощо. Такий підхід забезпечує зручність супроводу, масштабованість та можливість подальшої модернізації додатку.

Опис інтерфейсів і методів демонструє реалізацію взаємодії між компонентами додатку, включаючи обробку подій, отримання й обробку даних, а також їх відображення у зручному для користувача форматі. Було впроваджено низку механізмів обробки помилок і перевірки коректності даних, що підвищує надійність роботи аплікації.

Особливу увагу приділено створенню інтерфейсу користувача — з урахуванням вимог до юзабіліті та специфіки цільової аудиторії (адміністративний персонал медичного закладу). Розроблений інтерфейс є інтуїтивно зрозумілим, оптимізованим для мобільних пристроїв і дозволяє оперативно отримувати доступ до ключових показників діяльності установи.

Завершальним етапом стала розробка та проведення тестування мобільного додатку. Було реалізовано модульне тестування окремих компонентів, а також інтеграційне тестування функціоналу в цілому. Отримані результати свідчать про стабільність роботи додатку та відповідність його функціональності заявленим вимогам.

					БР.ІП – 04.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		73

ВИСНОВКИ

В дипломній роботі здійснено всебічне дослідження проблематики застосування інформаційних технологій в управлінні медичними закладами та реалізовано практичну розробку Android-додатку, орієнтованого на адміністрування їхньої діяльності.

На основі аналізу предметної області було встановлено необхідність цифровізації управлінських процесів у сфері охорони здоров'я, зокрема через мобільні платформи. Розглянуто передумови розробки відповідного мобільного застосунку, зокрема нормативні (протокол HL7) та організаційні аспекти, а також визначено перелік ключових показників ефективності, релевантних для моніторингу в межах додатку.

У роботі приділено значну увагу стандартам обміну медичними даними, зокрема повідомленням типу ADT протоколу HL7, що стали основою для структурування інформаційних потоків між серверною та клієнтською частинами мобільного застосунку. На основі REST-архітектури було реалізовано взаємодію клієнт-сервер, використовуючи формат JSON як зручний спосіб передачі структурованих даних.

У ході розробки програмного забезпечення було створено структуру Android-додатку, розроблено та описано ключові програмні компоненти, включно з класами, інтерфейсами та методами. Здійснено побудову користувацького інтерфейсу з урахуванням специфіки цільової аудиторії, що забезпечує зручність використання та високу інформативність.

Проведено комплексне тестування мобільного додатку, яке засвідчило його працездатність, відповідність функціональним вимогам та готовність до впровадження в реальних умовах роботи медичних установ.

Узагальнюючи результати, можна стверджувати, що реалізований додаток має потенціал для подальшого масштабування та інтеграції в існуючі інформаційні системи охорони здоров'я, сприяючи підвищенню

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						74
Змн.	Арк.	№ докум.	Підпис	Дата		

ефективності управлінських процесів, прозорості та оперативності прийняття рішень у медичних закладах.

Таким чином, у межах даної роботи виконано практичну реалізацію повнофункціонального Android-рішення, що може бути інтегроване у процеси управління сучасного медичного закладу.

					БР.ІП – 04.00.00.000 ПЗ	Арк.
						75
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. HL7 Integration | Healthcluster. – <https://healthcluster.co/the-role-of-hl7-integration-in-healthcare-communication/>
2. Prolis - Health Level 7 (HL7) is a set of international... | Facebook. – <https://www.facebook.com/Prolislabs/posts/health-level-7-hl7-is-a-set-of-international-standards-for-the-exchange-integrat/862653722516190/>
3. Types of HL7 ADT message and an example of ADT^A04 | InterSystems Developer – <https://community.intersystems.com/post/types-hl7-adt-message-and-example-adta04>
4. Android Studio Cloud | Android Developers - <https://developer.android.com/studio/preview/android-studio-cloud>
5. Patient Protection and Affordable Care Act - Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Patient_Protection_and_Affordable_Care_Act.
6. Android – JSON Parser Tutorial. http://www.tutorialspoint.com/android/android_json_parser.html.
7. Web Service- Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Web_service.
8. P. Chinniah and S. Muttan, “HL7 Standard based hospital information System,” International Journal of Biosciences & Technology (IJBST), Vol. 5, pp. 32-39, Sep. 2012.
9. Android (Operating System) - Wikipedia, the free encyclopedia. [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system)).
10. D. Blumenthal and M. Tavenner, “The Meaningful Use Regulation for Electronic Health Records,” The New England Journal of Medicine, Vol. 363, no. 6, pp. 501–504, Aug. 2010.

					БР.ІІІ – 04.00.00.000 ІІЗ	Арк.
						76
Змн.	Арк.	№ докум.	Підпис	Дата		

11. Ahmed, M. U., et al. (2019). "Mobile Health Application for Hospital Management System." *Procedia Computer Science*, 154, 31–38. <https://doi.org/10.1016/j.procs.2019.06.005>
12. Kaur, G., & Rani, R. (2020). "Design and Development of a Hospital Management System Using Android." *International Journal of Scientific & Technology Research*, 9(3), 743–746.
13. Rathod, A., & Patel, S. (2018). "Smart Hospital Management System Using Android." *International Journal of Engineering Research and Applications*, 8(6), 21–25.
14. Adithya, S., et al. (2016). "Android Based Hospital Management System." *International Journal of Computer Applications*, 144(6), 18–21.
15. Lee, J. H. (2013). "Smart Healthcare System Using Android Mobile Device." *International Journal of Smart Home*, 7(5), 83–94.
16. Raj, A., & Tiwari, P. (2015). "Android Based Health Care Monitoring System." *International Journal of Advanced Research in Computer Science and Software Engineering*, 5(3), 178–181.
17. Supriya, N., et al. (2017). "Design and Implementation of Android Based Healthcare Application." *International Journal of Engineering and Technology*, 9(2), 1055–1059.
18. Panwar, A., & Dhiman, G. (2021). "Design and Implementation of E-Health Monitoring System on Android Platform." *Journal of Ambient Intelligence and Humanized Computing*, 12, 8125–8135. <https://doi.org/10.1007/s12652-020-02608-4>
19. Jebraeil, M., et al. (2016). "Developing an Android-Based Mobile Application for Hospital Appointment Scheduling." *Journal of Health Management & Informatics*, 3(4), 101–105.
20. Manogaran, G., & Lopez, D. (2017). "A Survey of Big Data Architectures and Machine Learning Algorithms in Healthcare." *Journal of King Saud*

- University - Computer and Information Sciences, 34(4), 384–396.
<https://doi.org/10.1016/j.jksuci.2018.09.014>
21. Patel, A., et al. (2016). "Mobile Application for Hospital Management System." International Journal of Computer Applications, 145(9), 22–25.
 22. Bashir, S., et al. (2018). "Design and Implementation of a Mobile-Based Health Management System." International Journal of Online and Biomedical Engineering (iJOE), 14(12), 121–134.
 23. Garbharran, H., & Hurkchand, H. (2014). "An Investigation into Android-Based mHealth Applications." South African Journal of Bioinformatics and Health Informatics, 5(2), 5–12.
 24. Kumar, A., & Rajput, A. (2017). "Designing an Android App for Healthcare Management." International Journal of Engineering Development and Research, 5(3), 664–667.
 25. Luxton, D. D., et al. (2011). "mHealth for Mental Health: Integrating Smartphone Technology in Behavioral Healthcare." Professional Psychology: Research and Practice, 42(6), 505–512.
<https://doi.org/10.1037/a0024485>
 26. Kannan, R., et al. (2020). "Mobile-Based Patient Management System in Health Care." Materials Today: Proceedings, 33, 2534–2537.
 27. Wac, K. (2012). "Smartphone-Based Health Monitoring: Challenges and Opportunities." IEEE 2012 International Conference on Bioinformatics and Biomedicine Workshops, 660–664.
 28. Asif, N., et al. (2021). "An Efficient Health Monitoring System Using Mobile Devices." International Journal of Information Management Data Insights, 1(2), 100030. <https://doi.org/10.1016/j.jjime.2021.100030>
 29. Chen, M., et al. (2017). "Smart Healthcare: Challenges and Solutions." Sensors, 17(6), 1306. <https://doi.org/10.3390/s17061306>

30. Misra, S., & Mukherjee, S. (2020). "An Android-Based Framework for Smart Healthcare Monitoring System." *Procedia Computer Science*, 167, 2414–2423.
31. Rana, R. K., & Kumar, A. (2019). "Design and Development of Mobile Based Hospital Management System." *International Journal of Recent Technology and Engineering*, 8(3), 146–150.
32. Arora, D., & Tiwari, A. (2020). "A Review of Android-Based Mobile Applications in Healthcare." *International Journal of Research and Analytical Reviews*, 7(1), 367–371.
33. Pradhan, A., et al. (2018). "How to Evaluate Health Apps: A Scoping Review of Guidelines and Quality Assessment Tools." *NPJ Digital Medicine*, 1(1), 11. <https://doi.org/10.1038/s41746-018-0011-y>
34. Srinivas, K., & Rao, P. S. (2016). "Cloud-Based Hospital Information System Using Android Platform." *International Journal of Advanced Research in Computer Engineering & Technology*, 5(3), 805–809.
35. Jalali, M. S., et al. (2019). "Information Technology and Hospital Performance: A Review of the Literature." *Applied Clinical Informatics*, 10(3), 489–501.
36. Free, C., et al. (2013). "The Effectiveness of Mobile-Health Technology-Based Health Behaviour Change or Disease Management Interventions." *PLoS Medicine*, 10(1), e1001362.
37. Rao, M., & Jaiswal, D. (2017). "Role of Mobile Applications in Health Care System." *International Journal of Engineering Research and Applications*, 7(6), 31–34.
38. Bakht H. mobile ad hoc networks Various problems and some solutions. School of Computing and Mathematical Sciences, Liverpool John Mores University. <http://www.oocities.org/humayunbakht/PMANET.pdf>.

ДОДАТКИ

Додаток А

Частина програмних кодів розробленої аплікації

```
import React from 'react';

const MetricCard = ({ icon: Icon, label }: { icon: React.ReactNode;
label: string }) => (
  <div className="flex flex-col items-center justify-center p-4">
    {Icon}
    <span className="text-sm mt-2">{label}</span>
  </div>
);

const MetricsPage = () => {
  const metrics = [
    {
      label: 'Admit',
      icon: (
        <div className="w-12 h-12 bg-blue-100 flex items-center
justify-center">
          <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"
fill="none" stroke="currentColor" strokeWidth="2" className="w-6 h-6
text-blue-600">
            <path d="M3 10h2.5M9 10h2.5M15 10h2.5M21 10h-1.993" />
            <path d="M12 2v2M4.2 4.2l1.4 1.4M19.4 4.2l1.4 1.4M4.2
19.6l1.4-1.4M19.4 19.6l1.4-1.4" />
          </svg>
          <span className="absolute top-1 right-1 text-xs font-
bold">+</span>
        </div>
      ),
    },
    {
      label: 'Discharge',
      icon: (
        <div className="w-12 h-12 bg-green-100 flex items-center
justify-center">
          <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"
fill="none" stroke="currentColor" strokeWidth="2" className="w-6 h-6
text-green-600">
            <path d="M12 2C6.5 2 2 6.5 2 12s4.5 10 10 10-4.5 10-
10S17.5 2 12 2z" />
            <path d="M9.5 10.5l2 2 4-4" />
          </svg>
          <span className="absolute top-1 right-1 text-xs font-
bold">+</span>
        </div>
      ),
    },
    {
      label: 'Census',
      icon: (
        <div className="w-12 h-12 bg-teal-100 rounded-full flex items-
center justify-center">
```

```

        <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"
fill="none" stroke="currentColor" strokeWidth="2" className="w-6 h-6
text-teal-600">
            <path d="M12 2C6.5 2 2 6.5 2 12s4.5 10 10 10 10-4.5 10-
10S17.5 2 12 2z" />
            <path d="M10 11l2 2 4-4" />
        </svg>
    </div>
    ),
},
{
    label: 'Insurance',
    icon: (
        <div className="w-12 h-12 bg-gray-100 flex items-center
justify-center">
            <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"
fill="none" stroke="currentColor" strokeWidth="2" className="w-6 h-6
text-gray-600">
                <path d="M12 2C6.5 2 2 6.5 2 12s4.5 10 10 10 10-4.5 10-
10S17.5 2 12 2z" />
                <path d="M10 10v4h4v-4h-4z" />
            </svg>
        </div>
    ),
},
{
    label: 'Demographics',
    icon: (
        <div className="w-12 h-12 bg-gray-100 flex items-center
justify-center">
            <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"
fill="none" stroke="currentColor" strokeWidth="2" className="w-6 h-6
text-gray-600">
                <path d="M17 21v-2a4 4 0 0 0-4-4H5a4 4 0 0 0-4 4v2" />
                <circle cx="9" cy="7" r="4" />
                <path d="M23 21v-2a4 4 0 0 0-3-3.87" />
                <path d="M16 3.13a4 4 0 0 1 0 7.75" />
            </svg>
        </div>
    ),
},
{
    label: 'Length of stay',
    icon: (
        <div className="w-12 h-12 bg-gray-100 flex items-center
justify-center">
            <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"
fill="none" stroke="currentColor" strokeWidth="2" className="w-6 h-6
text-gray-600">
                <circle cx="12" cy="12" r="10" />
                <path d="M12 6v6l4 2" />
            </svg>
        </div>
    ),
},
{
    label: 'Diagnosis',

```

```

        icon: (
          <div className="w-12 h-12 bg-gray-100 flex items-center
justify-center">
            <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"
fill="none" stroke="currentColor" strokeWidth="2" className="w-6 h-6
text-gray-600">
              <path d="M3 3h18M3 9h18M9 21V9" />
              <path d="M9 3v18M12 7h.01M12 12h.01M12 17h.01M15 7h.01M15
12h.01M15 17h.01M18 7h.01M18 12h.01M18 17h.01" />
            </svg>
          </div>
        ),
      },
    ],
  );

  return (
    <div className="h-screen flex flex-col">
      <header className="bg-blue-500 text-white p-4 flex justify-
between items-center">
        <h1 className="text-lg font-bold">METRICS</h1>
        <button className="bg-blue-700 hover:bg-blue-800 p-2 rounded-
full">
          <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"
fill="none" stroke="currentColor" strokeWidth="2" className="w-4 h-4">
            <path d="M12 6v6l4 2" />
            <circle cx="12" cy="12" r="10" />
          </svg>
        </button>
      </header>
      <main className="flex-1 grid grid-cols-2 gap-4 p-4 overflow-y-
auto">
        {metrics.map((metric, index) => (
          <MetricCard
            key={index}
            icon={metric.icon}
            label={metric.label} />
        ))}
      </main>
      <footer className="bg-black text-white p-4 flex justify-around
items-center">
        <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"
fill="none" stroke="currentColor" strokeWidth="2" className="w-6 h-6">
          <path d="M15 18l-6-6 6-6" />
        </svg>
        <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"
fill="none" stroke="currentColor" strokeWidth="2" className="w-6 h-6
rounded-full border border-white p-1">
          <circle cx="12" cy="12" r="10" />
        </svg>
        <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"
fill="none" stroke="currentColor" strokeWidth="2" className="w-6 h-6">
          <path d="M9 18l6-6 6-6" />
        </svg>
      </footer>
    </div>
  );
};

export default MetricsPage;

```

БІБЛІОГРАФІЧНА ДОВІДКА

Тема дипломної роботи: “Побудова Андроїд-аплікації адміністрування роботи медичного закладу ”

Обсяг пояснювальної записки: 79 аркушів.

Дата закінчення роботи: 10 червня 2025 р.

Підпис студента _____