

Івано-Франківський Національний Технічний Університет Нафти і Газу
Інститут Інформаційних Технологій
кафедра комп'ютерних систем і мереж

Юрчило Роман Миколайовчи

УДК _____ 004.77

(індекс)

БАКАЛАВРСЬКА РОБОТА

**Розробка мобільного додатку системи комунікації користувачів
фармацевтичної компанії засобами Java**

Комп'ютерна інженерія

(назва освітньої програми)

123 - Комп'ютерна інженерія

(шифр і назва спеціальності)

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

Здобувач освітнього ступеня _____ Юрчило Р. М.

(підпис, ініціали та прізвище здобувача)

Науковий керівник _____ Заячук Ярослав Іванович, к. т. н., доцент

(підпис, прізвище, ім'я, по батькові, науковий ступінь, вчене звання керівника)

Допущено до захисту

д. т. н., професор

(посада)

(підпис)

(дата)

Мельничук С. І.

(прізвище та ініціали)

м. Івано-Франківськ – 2025

Івано-Франківський Національний Технічний Університет Нафти і Газу

Інститут Інформаційних Технологій

Кафедра Комп'ютерних систем і мереж

Освітній рівень Бакалавр

Спеціальність 123 Комп'ютерна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри _____

Мельничук Степан Іванович

" ____ " _____ 2025 року

З А В Д А Н Н Я

НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТОВІ

Юрчилу Роману Миколайовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Розробка мобільного додатку системи комунікації користувачів фармацевтичної компанії засобами Java

керівник роботи Заячук Ярослав Іванович, к. т. н., доцент

затверджені наказом закладу вищої освіти від " 05 " травня 2025 року № 275/7

2. Термін подання студентом роботи 12 червня 2025 року

3. Вихідні дані до роботи Методичні вказівки, технічна література

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

1. Аналіз існуючих рішень та технологій, Вибір інструментів та технологій розробки, аналіз предметної області, огляд існуючих програмних рішень, вибір інструментів та технологій розробки, постановка завдання, 2. Проектування інформаційної системи, цілі, завдання та вимоги до мобільного додатку, архітектура системи, проектування бази даних, 3. Реалізація мобільного додатку, функціональні можливості мобільного додатку, технічна реалізація програмного рішення, тестування додатку та аналіз результатів,

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

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

Дата видачі завдання 29 січня 2025 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз існуючих рішень та технологій, теоретична підготовка до виконання бакалаврської роботи	<i>Лютий 2025р</i>	виконав
2	Написання коду	<i>Березень 2025р</i>	виконав
3	Реалізація мобільного додатку	<i>Квітень 2025р</i>	виконав
4	Оформлення пояснювальної записки	<i>Травень 2025р</i>	виконав

Студент _____
(підпис)

Юрчило Р. М.
(прізвище та ініціали)

Керівник роботи _____
(підпис)

Заячук Я. І.
(прізвище та ініціали)

АНОТАЦІЯ

Дипломна робота присвячена розробці мобільного додатку для автоматизації обліку та управління логістичними процесами у фармацевтичній компанії. Метою дослідження є створення програмного продукту, який забезпечує ефективне ведення обліку лікарських засобів, контроль партій, управління залишками та переміщеннями між складами.

У першому розділі проведено аналіз бізнес-процесів фармацевтичного підприємства, визначено потреби в автоматизації та обґрунтовано вибір технологій розробки.

У другому розділі спроектовано архітектуру клієнт-серверної системи, розроблено логічну та фізичну моделі бази даних, визначено основні функціональні та нефункціональні вимоги до додатку.

Третій розділ присвячено реалізації мобільного додатку засобами Java з використанням архітектурного шаблону MVVM, бібліотеки Room для роботи з локальною базою даних SQLite та корутин для забезпечення асинхронної обробки даних.

Результатом роботи є функціональний мобільний додаток інформаційної системи підприємства, що забезпечує зручний інтерфейс користувача відповідно підтримує офлайн-режим роботи з можливою подальшою синхронізацією даних.

МОБІЛЬНИЙ ДОДАТОК, JAVA, ФАРМАЦЕВТИЧНА КОМПАНІЯ, ІНФОРМАЦІЙНА СИСТЕМА, ОБЛІК ЛІКАРСЬКИХ ЗАСОБІВ, MVVM, ROOM, SQLITE, REST API.

Summary

The bachelor's thesis is devoted to the development of a mobile application aimed at automating inventory management and logistics processes within a pharmaceutical company. The goal of the research is to create a software product that provides efficient accounting of medicinal products, batch tracking, stock level management, and inter-warehouse transfers.

The first chapter analyzes the business processes of a pharmaceutical enterprise, identifies the need for automation, and substantiates the choice of development technologies.

The second chapter presents the design of a client-server system architecture, the development of logical and physical database models, and the definition of the main functional and non-functional requirements for the application.

The third chapter is dedicated to the implementation of the mobile application using Java. The development is based on the MVVM architectural pattern, the Room library for working with a local SQLite database, and Kotlin coroutines to enable asynchronous data processing.

As a result, a functional mobile component of the company's information system has been created. It features a user-friendly interface based on Material Design principles and supports offline mode with the possibility of future data synchronization.

MOBILE APPLICATION, JAVA, PHARMACEUTICAL COMPANY, INFORMATION SYSTEM, INVENTORY MANAGEMENT, MVVM, ROOM, SQLITE, REST API.

ЗМІСТ

ВСТУП.....	4
1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ТЕХНОЛОГІЙ	7
1.1 Аналіз предметної області.....	7
1.2 Огляд існуючих програмних рішень	15
1.3 Вибір інструментів та технологій розробки	21
1.3.1 Обґрунтування вибору мови програмування.....	21
1.3.2 Обґрунтування вибору IDE.	22
1.3.3 Інструменти та бібліотеки	23
1.4 Постановка завдання.....	24
2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	26
2.1 Цілі, завдання та вимоги до мобільного додатку.....	26
2.1.1 Мета та призначення системи.....	26
2.1.2 Основні завдання, що вирішуються мобільним додатком.....	28
2.1.3 Функціональні вимоги до мобільного додатку	29
2.1.4 Нефункціональні вимоги до мобільного додатку	33
2.2 Архітектура системи	34
2.3 Проєктування бази даних	37
3 РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ	49
3.1 Функціональні можливості мобільного додатку	49
3.2 Технічна реалізація програмного рішення	53
3.2.1 Створення та налаштування проєкту	53
3.2.2 Структура та архітектура мобільного додатку.....	56
3.2.3 Використані бібліотеки та технології	57

					БР.КІ-23.00.00.000 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дат</i>	Розробка мобільного додатку системи комунікації користувачів фармацевтичної компанії засобами Java	<i>Лім.</i>	<i>Арк.</i>	<i>Акрушіє</i>
Розроб.		Юрчило Р. М.				3	75	
Перевір.		Заячук Я. І.				ІФНТУНГ, КІ-21-1		
Рецензент		Бабчук С. М.						
Н. Контр.		Лазорів А. М.						
Затверд.		Мельничук С. І.						

3.2.4	Опис основних класів та їхнього призначення	58
3.2.5	Опис фрагментів коду.....	60
3.2.6	Опис використаних шаблонів проєктування.....	63
3.3	Тестування додатку та аналіз результатів	65
ВИСНОВКИ.....		68
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ		70
ДОДАТКИ.....		71

4

					БР.КІ-23.00.00.000 ПЗ	Арк
						4
Змн.	Арк.	№ докум.	Підпис	Дат		

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

API Application Programming Interface — інтерфейс прикладного програмування

БД - База даних

CRUD – Create, Read, Update, Delete – базові операції над даними

DAO – Data Access Object – об’єкт доступу до даних

ER – Entity-Relationship — сутність–зв’язок (модель даних)

GUI – Graphical User Interface – графічний інтерфейс користувача

JSON – JavaScript Object Notation – формат обміну даними

MVVM – Model-View-ViewModel – архітектурний шаблон

ORM – Object-Relational Mapping – об’єктно-реляційне відображення

REST – Representational State Transfer – архітектурний стиль побудови веб-сервісів

SQLite - Легка вбудована реляційна база даних

UML - Unified Modeling Language – уніфікована мова моделювання

UI – User Interface – інтерфейс користувача

UX – User Experience – досвід користувача

Room – Android ORM-бібліотека для роботи з SQLite

ViewModel – Компонент архітектури Android Jetpack для збереження стану UI

CRUD - Create, Read, Update, Delete – базові операції над даними

IDE – Integrated Development Environment – інтегроване середовище розробки

WMS – Warehouse Management System – система управління складом

ERP – Enterprise Resource Planning – система управління підприємством

ВСТУП

Сучасний розвиток інформаційних технологій суттєво впливає на ефективність функціонування підприємств у різних галузях, зокрема у фармацевтичній сфері. Автоматизація бізнес-процесів дозволяє оптимізувати діяльність компаній, підвищити продуктивність та зменшити витрати. Згідно з актуальними дослідженнями впровадження сучасних інформаційних технологій сприяє розвитку фармацевтичного бізнесу в Україні, покращуючи якість обслуговування та управління ресурсами [1].

Використання мобільних додатків у фармацевтичній галузі є перспективним напрямом, оскільки вони забезпечують доступність, мобільність та зручність у використанні, оскільки розробка подібних додатків сприяє покращенню взаємодії між компанією та клієнтами, оптимізації процесів замовлення та доставки продукції, а також підвищенню конкурентоспроможності підприємства на ринку. Зважаючи на швидкий розвиток технологій та зростаючі потреби споживачів, створення ефективних мобільних рішень для фармацевтичних компаній є актуальним завданням [2].

Об'єктом дослідження є процеси автоматизації діяльності фармацевтичної компанії засобами сучасних інформаційних технологій.

Предмет дослідження – мобільний додаток, розроблений засобами мови програмування Java для інформаційної системи фармацевтичної компанії.

Метою роботи є розробка мобільного додатку для інформаційної системи фармацевтичної компанії.

Для досягнення мети роботи необхідно вирішити такі завдання:

- проаналізувати особливості бізнес-процесів фармацевтичної компанії;
- провести огляд існуючих програмних рішень для автоматизації процесів у фармацевтичній галузі;
- обґрунтувати вибір технологій та засобів розробки мобільних додатків;
- спроектувати архітектуру мобільного додатку та модель бази даних;

					БР.КІ-23.00.00.000 ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дат		

- реалізувати базовий функціонал мобільного додатку засобами java;
- провести тестування створеного додатку.

Застосовані методи дослідження включають аналіз літературних та інформаційних джерел, порівняння існуючих програмних рішень, методи моделювання (структурне, об'єктно-орієнтоване моделювання), програмування та тестування програмного забезпечення [3]. Зокрема, використовуються теоретичні методи, такі як аналіз та емпіричні методи, включаючи спостереження та експеримент [4].

Практичне значення отриманих результатів полягає у можливості використання розробленого мобільного додатку як демонстраційного проєкту.

					БР.КІ-23.00.00.000 ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дат		

1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ТЕХНОЛОГІЙ

1.1 Аналіз предметної області

Фармацевтичний сектор – одна з ключових ланок системи охорони здоров'я, адже саме він забезпечує населення безпечними, ефективними та якісними лікарськими засобами. Відповідно до ВООЗ, доступність «essential medicines» є необхідною передумовою універсального охоплення медичними послугами та зниження показників захворюваності й смертності [5].

Окрім медичного ефекту, галузь суттєво впливає на економіку, оскільки в Україні вона входить до переліку пріоритетних секторів, формуючи значну частку ВВП та створюючи високотехнологічні робочі місця [6]. У таблиці 1.1 систематизовано ключових учасників фармацевтичного ринку України, їхні основні функції та роль у ланцюгу створення вартості лікарських засобів.

Таблиця 1.1 - Ключові учасники ринку

Сегмент	Опис
Виробники	На ринку присутні великі міжнародні корпорації (Sanofi, Pfizer, Novartis тощо) та понад 100 вітчизняних заводів, що спеціалізуються переважно на генериках і безрецептурних препаратах.
Дистриб'ютори	Оптовий сегмент концентрується навколо кількох лідерів («БадМ», «Оптіма-Фарм»), які забезпечують логістику та зберігання з дотриманням GDP-стандартів і холодового ланцюга.
Аптечні мережі	Близько 70 % роздрібною обігу припадає на мережі («Аптека низьких цін», «Подорожник»,), тоді як частка незалежних аптек скорочується через вимоги до автоматизації та контролю залишків.
Регулятори	Державну політику формує Міністерство охорони здоров'я, а контроль якості й ліцензування здійснює Державна служба України. Перехід до європейської моделі регулювання передбачає запровадження e-CTD-досьє [7].
Професійна спільнота	Фармацевти виконують функції клінічної фармації, фармаконагляду й пацієнтської освіти, що підвищує раціональне використання ліків [8].

					БР.КІ-23.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		7

Таким чином, фармацевтична галузь інтегрує виробничі, логістичні та клінічні процеси, а її ефективність безпосередньо визначає доступність медичної допомоги та фінансову стійкість системи охорони здоров'я. Ці особливості слід враховувати під час розробки інформаційних систем, зокрема мобільних додатків, орієнтованих на управління запасами, відстеження серій, моніторинг температури та взаємодію з регуляторними базами даних.

Нормативно-правове регулювання у фармацевтичній галузі покликане забезпечити ефективність, якість і безпеку лікарських засобів на всіх етапах їх життєвого циклу [9]. Правові акти встановлюють стандарти та процедури для ліцензування виробництва й оптової/роздрібною торгівлі, регламентують умови зберігання й транспортування, а також визначають вимоги до контролю серій, відстежуваності, фармаконагляду та захисту персональних даних.

Умовою легальної діяльності фармацевтичних компаній є отримання ліцензії від Державної служби України з лікарських засобів та контролю за наркотиками (Держлікслужба). Наявність ліцензії засвідчує, що підприємство відповідає вимогам належної виробничої (GMP) та належної дистрибуторської практики (GDP).

GMP (Good Manufacturing Practice) встановлює стандарти для виробництва, контролю якості, валідації процесів і документації. Підприємство повинно дотримуватися принципів чистих приміщень, кваліфікації обладнання та навчання персоналу [10].

GDP (Good Distribution Practice) регулює умови транспортування, зберігання й обліку препаратів. Зокрема, враховуються температурний режим, захист від псування, система управління поверненнями й утилізацією.

Комплаєнс із GMP/GDP визначає можливість виведення продукції на ринок і є обов'язковою передумовою для міжнародних партнерств та експортних операцій [11].

Фармацевтична продукція випускається визначеними серіями, кожна з яких має унікальний номер і дату виготовлення. Відповідно до чинних вимог,

					БР.КІ-23.00.00.000 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дат		

компанії зобов'язані вести детальний облік кожної серії та стежити за її рухом ланцюгом поставок [12], що надає змогу для:

- швидкого виявлення й відкликання дефектної або простроченої продукції;
- забезпечення прозорості і убезпечення від фальсифікатів завдяки системам track & trace (маркування унікальними кодами);
- контролю термінів придатності на складах і в аптеках, щоб не допустити реалізації неякісних ліків.

Розвиток технологій, зокрема впровадження QR-кодів і RFID-міток, дає змогу автоматизувати процеси відстежуваності й спростити обмін даними між дистриб'юторами, аптеками та регуляторними органами.

Умови обробки та зберігання персональних і медичних даних пацієнтів/споживачів регламентуються Законом України «Про захист персональних даних» та низкою підзаконних актів. У контексті міжнародної співпраці дедалі більшого значення набувають норми GDPR (General Data Protection Regulation), що зобов'язують компанії:

- отримувати інформовану згоду від суб'єктів даних;
- забезпечувати належний рівень безпеки баз даних (шифрування, контроль доступу);
- здійснювати обмежене зберігання та своєчасне видалення медичних/персональних відомостей.

Для фармацевтичної діяльності, яка охоплює збір, аналіз і передачу даних про пацієнтів, ліки та рецептурні призначення, захист інформації набуває критичного значення. Порушення може спричинити не лише адміністративну чи кримінальну відповідальність, а й репутаційні втрати й судові позови [13].

Таким чином, у межах нормативно-правового поля фармацевтична галузь поєднує вимоги до виробництва, дистрибуції та контролю якості лікарських засобів із суворими приписами щодо відстежуваності серій і збереження конфіденційності даних. Усе це створює комплексну регуляторну базу, яку

					БР.КІ-23.00.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дат		

необхідно враховувати при розробці інформаційних та мобільних рішень для фармацевтичних компаній.

Ефективність фармацевтичного підприємства визначається злагодженістю трьох ключових процесів: безперервного забезпечення якісною сировиною, дотримання високих виробничих стандартів і стратегічного управління продуктами на всьому життєвому циклі. На рисунку 1.1 подано узагальнену схему, що демонструє логічний ланцюг цих процесів та їх взаємозв'язки.

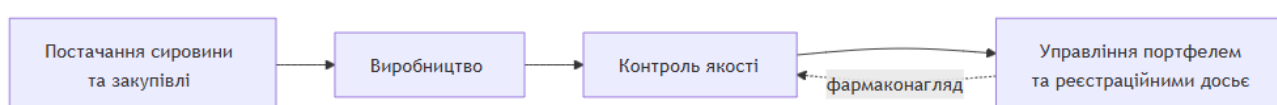


Рисунок 1.1 – Основні бізнес-процеси фармацевтичної компанії

Постачання сировини та закупівлі.

Компанія укладає довгострокові угоди з перевіреними постачальниками активних фармацевтичних інгредієнтів (API) та допоміжних речовин. До підписання контракту кожен постачальник проходить аудит на відповідність GMP-стандартам, а вже на етапі приймання сировина підлягає вхідному лабораторному контролю, що мінімізує ризики дефіциту, фальсифікації та порушення безпеки ланцюга постачання.

Виробництво й контроль якості.

Технологічний цикл охоплює підготовку сировини, змішування, формування дозованих форм (таблетування, капсулювання, асептичний розлив), пакування та маркування. Усі операції валідуються, а обладнання й середовище проходять регулярну кваліфікацію. Паралельно функціонує система QA/QC – зразки кожної серії аналізуються на відповідність специфікаціям, а результати фіксуються у виробничих протоколах для подальшого фармаконагляду.

					БР.КІ-23.00.00.000 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дат		

Управління портфелем препаратів та реєстраційними досьє

Після успішного контролю якості продукція переходить у фазу комерційного життєвого циклу:

– команда регуляторних справ підтримує електронні досьє (e-CTD), своєчасно подаючи оновлення щодо безпеки та змін у виробництві до регуляторних органів;

– маркетингові та наукові підрозділи аналізують попит, патентний статус і прибутковість, приймаючи рішення про розширення, модифікацію чи виведення препаратів. Дані фармаконагляду повертаються у відділ якості, замикаючи цикл безперервного удосконалення.

У фармацевтичному бізнесі фінансові операції тісно пов'язані з вимогами регуляторів і високими ризиками, що робить облік більш деталізованим, ніж у більшості галузей. Розглянемо ключові аспекти, які необхідно врахувати під час побудови облікової політики та вибору програмних рішень:

– облік партій і серій препаратів. Кожна серія має унікальний номер, дату виробництва й термін придатності, що фіксуються в обліковій системі. Це забезпечує точне відстеження руху товарів, швидке відкликання у разі виявлення браку та формування регуляторної звітності.

– калькулювання собівартості та облік повернень/утилізації. Собівартість розраховується на рівні серії з урахуванням вартості АРІ, допоміжних речовин, контролю якості та пакування. Повернення й утилізація прострочених або бракованих препаратів відображаються окремими центрами витрат, що дозволяє коректно оцінювати фінансові результати й податкові зобов'язання.

– інтеграція з системами ERP та CRM. Облікові дані синхронізуються з ERP-модулями (WMS, MRP, фінанси) та CRM-системою, формуючи єдине джерело правди щодо запасів, продажів і взаємодії з клієнтами. Це підвищує прозорість бізнес-процесів і підтримує оперативне управлінське рішення.

					БР.КІ-23.00.00.000 ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дат		

Оскільки логістика відіграє важливу роль у забезпеченні своєчасної доставки лікарських засобів із дотриманням необхідних стандартів якості, то важливо розглянути основні складовими цього процесу:

1. Холодовий ланцюг і температурний моніторинг. Препарати потребують суворого контролю температури (зазвичай 2–8 °С або –20 °С). Використовуються датчики й дистанційні системи моніторингу, що відстежують температуру та фіксують відхилення.

2. Транспортування контрольованих речовин. Перевезення препаратів із наркотичними або психотропними речовинами вимагає спеціальних ліцензій, GPS-контролю, пломбування вантажу і, за потреби, супроводу охорони.

3. Оптимізація маршрутів і управління поставками. Використання систем WMS/TMS дозволяє оптимізувати маршрути з урахуванням попиту, термінів придатності та завантаженості доріг, скорочуючи витрати і підвищуючи якість обслуговування.

Наступним після логістики є етап продажу лікарських засобів, що здійснюється через оптові компанії, аптечні мережі та канали електронної комерції (e-commerce). Ціноутворення формується з урахуванням регуляторних обмежень, а також застосовуються знижки й бонусні програми для стимулювання збуту. Взаємодія з лікарями контролюється законодавством і супроводжується маркетинговими обмеженнями щодо реклами рецептурних препаратів.

Зберігання лікарських препаратів здійснюється на спеціалізованих складах, які відповідають стандартам GDP (належної дистриб'юторської практики), зокрема оснащених холодowymi камерами для температурно чутливих ліків. Для оптимізації використовується адресна система зберігання із застосуванням принципів FIFO (перший прийшов – перший пішов) та FEFO [14] (перший закінчується термін придатності – перший пішов). Процеси інвентаризації та контролю термінів придатності автоматизовані, що мінімізує ризики реалізації прострочених чи неякісних препаратів.

					БР.КІ-23.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		12

Фармацевтичні компанії активно використовують інформаційні технології (ІТ) для автоматизації бізнес-процесів, підвищення прозорості та забезпечення відповідності нормативним вимогам. Основними складовими ІТ-інфраструктури є програмні модулі для управління складом, логістикою, лабораторними дослідженнями та документацією виробництва., де особлива увага приділяється безпеці даних, резервному копіюванню та здатності до масштабування систем. Типові програмні модулі програмного забезпечення перераховані в таблиці 1.2.

Таблиця 1.2 – Типові програмні модулі

Назва модуля	Призначення	Приклади задач
WMS	Управління складськими запасами	Прийом, адресне зберігання, інвентаризація
TMS	Управління транспортом і доставками	Планування маршрутів, моніторинг транспорту
LIMS	Управління лабораторними дослідженнями	Облік аналізів, контроль якості препаратів
e-Batch Record	Управління електронними записами виробничих серій	Ведення документації серій, трекінг виробництва

Окрім основних систем, компанії впроваджують мобільні додатки, які дозволяють працівникам складу, логістики та продажів оперативно виконувати завдання, використовуючи смартфони або планшети. До ІТ-інфраструктури фармацевтичних компаній ставляться особливі вимоги по трьом основним аспектам:

1. Безпека – обов’язкове шифрування даних, контроль доступу, захист від несанкціонованого втручання.
2. Резервне копіювання – автоматичне створення резервних копій зберігає цілісність інформації у разі аварійних ситуацій.
3. Масштабованість – здатність швидко розширювати чи модифікувати систему залежно від змін бізнес-процесів і потреб компанії.

Аналіз предметної області фармацевтичної компанії дозволив визначити основні бізнес-процеси, нормативні вимоги та інформаційні потреби, які слід

враховувати при розробці мобільного додатку. Такий додаток має не лише забезпечувати зручність роботи персоналу, але й відповідати стандартам безпеки, якості та галузевої специфіки.

Ключові функції, що мають бути реалізовані в мобільному додатку:

- перегляд залишків на складі, статусу поставок та термінів придатності;
- оформлення заявок на переміщення товарів і фіксація приймання/відвантаження;
- перегляд історії логістичних операцій;
- авторизація з обмеженням прав доступу відповідно до ролі користувача (опційно).

Обмеження та ризики, пов'язані з галузевою специфікою:

- необхідність дотримання стандартів GMP/GDP і законодавчих вимог щодо відстежуваності та обліку препаратів;
- підвищені вимоги до інформаційної безпеки, шифрування даних та автентифікації;
- обмеження у використанні мобільних пристроїв у зонах з контрольованим середовищем;
- можливі регуляторні перевірки та необхідність формування електронної звітності.

Можливості інтеграції з наявними системами:

- взаємодія з ERP, WMS та CRM-системами для обміну інформацією в реальному часі;
- підтримка API або middleware-рішень для синхронізації даних про запаси, переміщення та серії;
- використання єдиного облікового запису (SSO) та централізованого управління доступом.

Таким чином, мобільний додаток має стати ефективним інструментом для автоматизації операцій, що виконуються «в полі», підвищити точність обліку та забезпечити відповідність фармацевтичним стандартам.

					БР.КІ-23.00.00.000 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дат		

1.2 Огляд існуючих програмних рішень

Інформаційні технології у фармацевтичному бізнесі відіграють ключову роль у забезпеченні прозорості, якості та відповідності регуляторним вимогам. Існуючі програмні рішення умовно поділяються на кілька категорій залежно від їх функціонального призначення та масштабу впровадження:

1. ERP-системи (Enterprise Resource Planning) – комплексні рішення для управління ресурсами підприємства, що охоплюють виробництво, фінанси, закупівлі, логістику та облік. Часто мають модулі, адаптовані під фармацевтичну галузь.

2. WMS/TMS-системи (Warehouse / Transport Management Systems) – програмні платформи для управління складськими запасами та транспортуванням, із підтримкою холодового ланцюга, серійного обліку, FIFO/FEFO.

3. LIMS (Laboratory Information Management Systems) – системи для автоматизації лабораторного контролю якості лікарських засобів, зберігання результатів аналізів та відстеження тестів.

4. CRM-системи (Customer Relationship Management) – рішення для управління відносинами з клієнтами, лікарями, аптечними мережами, включаючи планування візитів, маркетинг і зворотний зв'язок.

5. Мобільні додатки – спеціалізовані інструменти для працівників складу, логістики або представників продажів, що працюють «у полі». Забезпечують доступ до облікових систем, сканування товарів, формування звітів у реальному часі.

6. Хмарні сервіси (SaaS, PaaS) – платформи, які надають доступ до функціональності через інтернет. Вони забезпечують гнучкість, швидке розгортання та зниження витрат на інфраструктуру, але потребують особливої уваги до захисту персональних і медичних даних.

					БР.КІ-23.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		15

Мобільні додатки у фармацевтичній логістиці дозволяють забезпечити оперативний доступ до даних, автоматизувати складські операції та підвищити точність обліку. Вони особливо актуальні для співробітників складу, логістики та доставки, які працюють поза межами стаціонарних робочих місць. Такі додатки зазвичай інтегруються з ERP/WMS-системами та підтримують функції сканування штрих-кодів, облік партій і серій, контроль температури та формування звітів у реальному часі. В таблиці 1.3 наведено приклади популярних мобільних рішень, які використовуються у фармацевтичній галузі.

Таблиця 1.3 - Огляд популярних мобільних додатків в світі

Назва додатку	Розробник	Основні функції	Особливості використання
SAP Warehouse Operator	SAP	Сканування штрих-кодів, облік залишків, робота з серіями	Глибока інтеграція з SAP S/4HANA
Oracle Mobile SCM	Oracle	Управління поставками, контроль партій, перегляд замовлень	Підтримує SCM Cloud; потребує стабільного інтернету
Mobile WMS	Tasklet Factory	Складські операції, offline-режим, підтримка сканерів	Сумісність із Microsoft Dynamics
LogiPharma App	LogiPharma Solutions	Трекінг доставки, моніторинг температури, звітність	Орієнтований на cold chain логістику

Але не дивлячись на широкий функціонал, більшість з перелічених додатків мають низку обмежень у контексті українського фармацевтичного ринку, наприклад:

- відсутність україномовного інтерфейсу та локалізованої документації;
- складна інтеграція з наявними обліковими системами українських підприємств;
- залежність від платформи (в основному працюють лише в екосистемах SAP, Oracle тощо);
- висока вартість ліцензування та технічної підтримки.

Ці фактори створюють передумови для розробки власного адаптованого мобільного рішення з урахуванням локальних вимог, мовної підтримки та інтеграції з уже існуючими внутрішніми системами фармацевтичної компанії.

Також фармацевтичні підприємства активно впроваджують ERP- та CRM-системи, адаптовані під галузеві вимоги. Такі рішення забезпечують централізоване управління виробництвом, дистрибуцією, якістю та взаємодією з клієнтами, прикладами яких є:

- SAP S/4HANA Life Sciences – містить модулі для управління серіями, e-Batch Record, GMP-звітністю;
- Microsoft Dynamics 365 for Pharma – підтримує контроль якості, облік залишків, CRM для медичних представників;
- PharmaSuite (Rockwell Automation) – орієнтована на виробничі процеси з детальним контролем партій та відповідністю GDP.

Усі перелічені системи підтримують інтеграцію з мобільними застосунками та мають інструменти для traceability, електронного документообігу, а також відповідають міжнародним вимогам до зберігання та обробки даних.

Хмарні сервіси (SaaS/Cloud-based) дедалі частіше використовуються у фармацевтичній галузі для автоматизації логістики, обліку, фармаконагляду та взаємодії з партнерами, так як вони не потребують локальної інфраструктури, швидко впроваджуються та легко масштабуються:

1. TrackTraceRx – платформа для моніторингу ланцюга постачання та контролю за серіями (traceability) [15];
2. Veeva Systems – комплексне хмарне рішення для управління документами, клінічними випробуваннями та взаємодією з медичними працівниками [16];
3. TraceLink – спеціалізується на серіалізації, track & trace та дотриманні вимог регуляторів різних країн [17].

Отже, серед переваг - це гнучкість, доступність з будь-якого пристрою, регулярні оновлення, масштабованість, а серед недоліків - залежність від інтернет-з'єднання, питання інформаційної безпеки, ризику зберігання даних за кордоном.

					БР.КІ-23.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		17

В Україні активно розвиваються мобільні додатки, які спрощують пошук, бронювання та замовлення лікарських засобів. Ці сервіси інтегрують дані з численних аптек, надаючи користувачам актуальну інформацію про наявність та ціни препаратів. В таблиці 1.4 наведено порівняльні характеристики популярних мобільних додатків для пошуку та замовлення ліків в Україні.

Ці додатки значно полегшують процес пошуку та замовлення лікарських засобів для українських користувачів, забезпечуючи зручний доступ до актуальної інформації про наявність та ціни препаратів у найближчих аптеках.

Таблиця 1.4 - Огляд популярних мобільних додатків в Україні

Додаток	Основні функції	Особливості
Tabletki.ua	Пошук і бронювання ліків, порівняння цін, довідник препаратів	Понад 12 000 аптек, підтримка пошуку за діючою речовиною, безкоштовний доступ без реєстрації
Apteki.ua	Бронювання ліків, пошук аналогів, інструкції до препаратів	Понад 7 000 аптек, оновлення цін кожні 15 хвилин, можливість бронювання частини упаковки
Ліки Контроль	Пошук офіційних інструкцій, підбір аналогів, контроль термінів придатності	База всіх зареєстрованих в Україні ліків, щоденне оновлення, відсутність реклами
Аптека АНЦ	Замовлення ліків, програма лояльності, історія покупок	Пошук аптек поруч, нагадування про терміни придатності, бонусна система
Аптека D.S.	Пошук товарів з фільтрами, електронна картка лояльності, історія покупок	Вибір аптеки для самовивозу, фільтрація за формою випуску, дозуванням та виробником
Подорожник	Інтернет-магазин, доставка замовлень, категоризація товарів	Доставка протягом 15 хвилин або 2–3 днів, чотири категорії товарів: ліки, вітаміни, гігієна, інше
Мед-Сервіс	Віртуальна картка лояльності, перегляд акцій та новин	Обмежена функціональність, часткова українізація, реєстрація російською мовою

Наприклад, Tabletki.ua – один із найпопулярніших в Україні сервісів для пошуку, порівняння цін та бронювання лікарських засобів. Додаток об'єднує дані понад 12 000 аптек по всій країні та забезпечує швидкий доступ до актуальної інформації про наявність препаратів, їх вартість, інструкції та аналоги (рис 1.2).

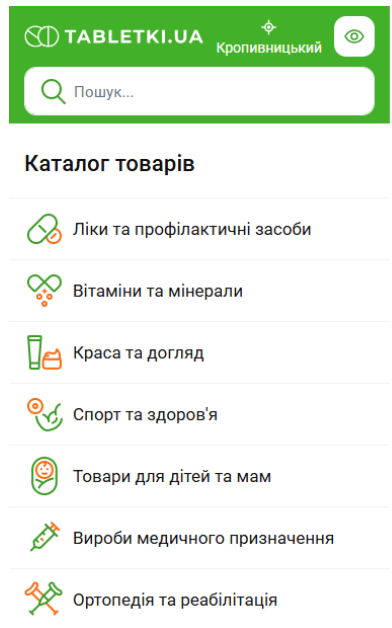


Рисунок 1.2 - Інтерфейс додатку Tabletki.ua

Таблиця 1.5 - Основні характеристики мобільного додатку Tabletki.ua

Категорія	Опис
Основні функції	Пошук ліків за назвою або діючою речовиною Порівняння цін у різних аптеках Бронювання з можливістю самовивозу Перегляд офіційних інструкцій до препаратів Пошук аптек поблизу та перегляд на карті
Переваги	Зручний інтерфейс українською мовою Висока точність і регулярне оновлення інформації Не потребує реєстрації Підтримка додатків для Android та iOS
Обмеження	Відсутність функцій обліку партій і серій Немає інтеграції з ERP/WMS-системами Орієнтація лише на кінцевого споживача

Таким чином, Tabletki.ua є прикладом ефективного B2C-рішення для пацієнтів, однак не покриває потреб підприємств фармацевтичної галузі в автоматизації обліку, логістики чи контролю якості. Це підкреслює необхідність розробки спеціалізованих мобільних додатків, орієнтованих на внутрішні операційні процеси компаній.

Аналіз існуючих мобільних додатків, ERP/CRM-систем та хмарних сервісів, що використовуються у фармацевтичній галузі, дозволив виявити

низку обмежень, які знижують їх ефективність у контексті потреб українських підприємств, де переважна більшість доступних рішень:

- орієнтована на глобальні стандарти без урахування локального законодавства України (зокрема, щодо обігу контрольованих речовин, вимог до серіалізації та звітності перед МОЗ і Держлікслужбою);
- не містить україномовного інтерфейсу або документації, що ускладнює навчання персоналу;
- не адаптована до специфічних внутрішніх процесів підприємств, таких як облік за серіями, швидке формування звітів, контроль залишків у режимі реального часу;
- має високу вартість ліцензування та інтеграції, особливо для малих і середніх фармкомпаній.

Крім того, більшість популярних додатків (наприклад, Tabletki.ua, Artekі.ua) орієнтовані виключно на кінцевого споживача (B2C), тоді як для корпоративного сегменту (B2B) відсутні гнучкі, мобільні та інтегровані рішення.

З огляду на зазначене, розробка власного мобільного додатку є доцільною та стратегічно виправданою, що дозволить:

- реалізувати тільки ті функції, які відповідають реальним потребам підприємства (наприклад, облік партій, температурний контроль, автоматизація приймання/відвантаження);
- забезпечити повну сумісність із внутрішніми ERP/WMS-системами;
- знизити залежність від іноземних сервісів та адаптувати інтерфейс під українські стандарти та користувачів.

Таким чином, власний мобільний додаток для інформаційної системи фармацевтичної компанії стане ефективним інструментом для цифровізації процесів обліку, логістики та контролю якості в умовах сучасних викликів і регуляторних вимог.

					БР.КІ-23.00.00.000 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дат		

1.3 Вибір інструментів та технологій розробки

1.3.1 Обґрунтування вибору мови програмування

Під час вибору технологій для розробки мобільного додатку було враховано як загальні вимоги до сучасного програмного забезпечення, так і специфічні особливості фармацевтичної галузі, а отже основними критеріями стали:

1. Сумісність із Android - оскільки цільовою платформою є Android, технології мали забезпечувати повноцінну підтримку створення нативних застосунків із доступом до системних ресурсів, камер, сканерів штрих-кодів, геолокації тощо. Важливим також є забезпечення сумісності з різними версіями Android та широким спектром пристроїв.

2. Надійність і підтримка Java - мова Java обрана як одна з найстабільніших і найпоширеніших мов для Android-розробки. Вона має потужну екосистему, широку спільноту розробників, велику кількість бібліотек і рішень з відкритим кодом. Крім того, Java відзначається хорошою продуктивністю, а її суворі типізація знижує кількість помилок у коді.

3. Можливість інтеграції з корпоративними системами - розроблюване рішення має підтримувати взаємодію з ERP, CRM та WMS-системами, що вже використовуються у фармацевтичному підприємстві. Технології повинні легко взаємодіяти з REST API, базами даних, форматами JSON і XML, що є типовими для корпоративного середовища.

4. Безпека та масштабованість - фармацевтична галузь передбачає роботу з критично важливими й персональними даними. Тому вибрані технології повинні підтримувати шифрування, контроль доступу, а також можливість розширення функціоналу в майбутньому без необхідності повного переписування архітектури. Масштабованість також важлива для розширення застосунку відповідно до потреб підприємства.

					БР.КІ-23.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		21

Таким чином, зазначені критерії сформували основу для подальшого обґрунтованого вибору технологічного стеку, який забезпечить стабільну роботу та відповідність вимогам фармацевтичної компанії.

Основною мовою програмування для реалізації мобільного додатку обрано Java, вибір якої зумовлений офіційною підтримкою Java для розробки під платформу Android, а також її стабільністю та поширеністю в корпоративному середовищі. Java забезпечує повну сумісність з Android SDK, дозволяє реалізувати нативний інтерфейс, роботу з локальними базами даних, мережевими запитами та сторонніми бібліотеками. Серед основних переваг мови:

- підтримка об'єктно-орієнтованого програмування;
- висока стабільність і безпека виконання;
- сумісність із більшістю корпоративних форматів обміну даними (JSON, XML, REST API);
- наявність розвинутої екосистеми бібліотек для обробки даних, сканування штрих-кодів тощо.

Таким чином, Java є доцільним вибором для створення надійного та масштабованого мобільного рішення для фармацевтичного підприємства.

1.3.2 Обґрунтування вибору IDE

Для розробки мобільного додатку обране середовище IntelliJ IDEA 2024.2.3, яке забезпечує повноцінну підтримку Java-проектів та має розширені можливості інтеграції з інструментами Android-розробки.

IntelliJ IDEA має високу продуктивність, підтримує всі основні мови JVM-платформи, містить вбудовані засоби роботи з Git, базами даних, а також має зручні засоби автодоповнення, рефакторингу та налагодження коду. Версія 2024.2.3 підтримує оновлені інструменти для Android та роботу з Gradle.

Для реалізації Android-проекту у середовищі необхідно встановити: – плагін Android із JetBrains Marketplace;

					БР.КІ-23.00.00.000 ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дат		

- Android SDK та емулятор (AVD Manager);
- налаштування Gradle-проєкту відповідно до вимог Android Studio.

Після налаштування IntelliJ IDEA дозволяє створювати Android-застосунки з повноцінною підтримкою XML-розмітки, ресурсів, AndroidManifest.xml, логування та тестування.

Також IDE забезпечує зручний вбудований інтерфейс для підключення до баз даних (SQLite, PostgreSQL, MySQL тощо), перегляду таблиць, виконання SQL-запитів та аналізу структури даних. Це полегшує розробку логіки взаємодії з локальними та зовнішніми БД у межах мобільного застосунку.

Таким чином, використання IntelliJ IDEA забезпечує ефективне середовище розробки, що відповідає вимогам до створення функціонального, безпечного та підтримуваного Android-додатку.

1.3.3 Інструменти та бібліотеки

Для реалізації функціональності мобільного додатку використовуються сучасні бібліотеки, які забезпечують стабільну взаємодію з API, зберігання даних, обробку зображень, побудову інтерфейсу та підтримку архітектурних шаблонів. В таблиці 1.6 наведено перелік ключових інструментів із коротким описом їхнього призначення.

Таблиця 1.6 - Використані бібліотеки та інструменти

Бібліотека / інструмент	Призначення
Retrofit	Клієнт REST API, що забезпечує обмін даними у форматі JSON/XML.
Room	ORM-бібліотека для роботи з локальною базою даних SQLite.
ZXing / ML Kit	Сканування штрих-кодів і QR-кодів за допомогою камери пристрою.
ViewModel, LiveData	Компоненти архітектури MVVM, що відповідають за зберігання стану і реактивність інтерфейсу.
Material Components	Набір елементів інтерфейсу відповідно до стандартів Material Design.

Використання зазначених бібліотек дозволяє спростити реалізацію функціоналу, підвищити надійність та забезпечити підтримку сучасних вимог до Android-додатків.

Додатково для забезпечення гнучкості та можливості масштабування є можливість застосування інструментів інтеграції та налаштування проєкту:

- Firebase (опційно) використовується для аналітики, моніторингу стабільності (Crashlytics) та push-сповіщень.
- REST API забезпечує інтеграцію мобільного додатку з корпоративною ERP-системою для обміну даними (наприклад, облік товарів, залишків, замовлень).
- Gradle використовується як система збірки для управління бібліотеками, конфігурацією залежностей та середовищем виконання.

Ці інструменти підвищують адаптивність розробки та дозволяють швидко додавати нову функціональність.

Обраний стек технологій, зокрема мова Java у поєднанні з середовищем IntelliJ IDEA, забезпечує стабільну та перевірену екосистему для розробки Android-додатку. Такий підхід дає повний контроль над архітектурою, спрощує підтримку, дозволяє реалізувати безпечне та масштабоване рішення, адаптоване до специфіки фармацевтичного підприємства.

1.4 Постановка завдання

Після проведеного аналізу бізнес-процесів фармацевтичного підприємства та огляду існуючих мобільних рішень для контролю запасів, сформовано чітке бачення функціоналу і цілей майбутнього додатку. Дослідження показало, що актуальними є питання обліку серій, моніторингу термінів придатності та оперативного оновлення даних у «польових» умовах, що й лягло в основу постановки завдання.

Тому мета роботи полягає у розробці стабільного й інтуїтивно зрозумілого Android-додатку на Java, який автоматизує облік фармацевтичних

					БР.КІ-23.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		24

партій, забезпечує контроль термінів придатності і відстеження переміщень між складами в реальному часі за допомогою сканування штрих- та QR-кодів.

Для реалізації зазначеної мети необхідно виконати такі завдання:

1. Провести детальний аналіз існуючих процедур обліку й логістики фармацевтичного підприємства на основі інтерв'ю з користувачами та вивчення документації.

2. Сформулювати функціональні та нефункціональні вимоги до мобільного додатку.

3. Розробити архітектуру клієнт-серверної системи та логічну модель бази даних.

4. Створити інтерфейс користувача за стандартами Material Design і інтегрувати модуль сканування штрих- та QR-кодів.

					БР.КІ-23.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		25

2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Цілі, завдання та вимоги до мобільного додатку

2.1.1 Мета та призначення системи

Мобільний додаток створюється з метою підвищення ефективності роботи фармацевтичної компанії шляхом автоматизації операцій на етапах обліку, складування та логістики лікарських засобів. Основна мета полягає у забезпеченні оперативного доступу до даних про запаси, переміщення товарів та їхню якість без прив'язки до стаціонарного робочого місця. Додаток має надати користувачам інструмент для швидкого та безпечного виконання типових бізнес-завдань у мобільному середовищі.

Розроблюване рішення автоматизує такі ключові бізнес-процеси підприємства:

- облік залишків лікарських засобів на складах;
- контроль термінів придатності;
- облік партій і серій;
- оформлення заявок на переміщення продукції між складами або аптечними точками;
- реєстрацію фактів приймання та відвантаження товару.

Крім того, мобільний додаток сприятиме зниженню кількості помилок за рахунок автоматичного сканування штрихкодів і уніфікованого заповнення даних.

У загальній архітектурі інформаційної системи фармацевтичної компанії мобільний додаток виконує роль клієнтського інтерфейсу для оперативної роботи співробітників «у полі». Він забезпечує інтеграцію з основною ERP-системою підприємства через REST API, синхронізуючи інформацію про залишки, партії та рух товару. Таким чином, додаток розширює можливості основної системи, дозволяючи підтримувати актуальність даних у реальному

					БР.КІ-23.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		26

часі та забезпечуючи мобільність персоналу. На рисунку 2.1 подано місце мобільного додатку в загальній архітектурі інформаційної системи фармацевтичної компанії.



Рисунок 2.1 - Загальне місце мобільного застосунку в інформаційній системі

Як показано на рисунку, мобільний додаток виступає клієнтським інтерфейсом для користувачів, забезпечуючи обробку запитів і інтеграцію з ERP-системою підприємства через REST API. Основні бізнес-дані (залишки на складі, облік партій, логістичні операції) зберігаються у централізованій базі даних, до якої мобільний додаток має доступ через серверну частину системи. Такий підхід забезпечує актуальність інформації, високу швидкість обміну даними та можливість розширення функціональності у майбутньому.

2.1.2 Основні завдання, що вирішуються мобільним додатком

Мобільний додаток розроблений для вирішення основних ключових завдань, що спрямовані на оптимізацію внутрішніх бізнес-процесів фармацевтичної компанії. Визначимо основні задачі, які реалізуються в межах функціоналу додатку:

1. Облік залишків лікарських засобів на складах – забезпечення оперативного доступу до інформації про наявні запаси продукції, її кількість та місце розташування в системі складського обліку. Щоб правильно реалізувати цей функціонал у мобільному додатку, потрібно детально описати відповідний перелік ключових функцій та можливостей:

- формат і структура даних про залишки (атрибути лікарських засобів потрібно відображати, додаткова інформація);
- джерело даних і синхронізація (ERP-система, окрема база даних, API);
- частота оновлення даних (в реальному часі, за запитом користувача, періодично);
- фільтрація і пошук (пошук за назвою препарату, за серією, за складом, за терміном придатності тощо);
- швидкий пошук за ключовими словами або скануванням штрихкоду;
- відображення інформації в інтерфейсі (таблиця, картки, списки);
- безпека доступу до даних.

2. Контроль термінів придатності препаратів – моніторинг дати виготовлення і закінчення терміну придатності для забезпечення вчасного оновлення запасів та недопущення реалізації прострочених лікарських засобів.

3. Оформлення заявок на переміщення товарів – створення та реєстрація внутрішніх заявок на переміщення продукції між складами або до аптечних мереж із фіксацією ключових параметрів товару (кількість, серія, термін придатності).

					БР.КІ-23.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		28

4. Реєстрація операцій приймання і відвантаження – фіксація фактів отримання продукції на склад або її відвантаження зі складу з автоматичним оновленням даних в обліковій системі.

5. Облік партій і серій лікарських засобів – ведення обліку унікальних ідентифікаторів серій препаратів для забезпечення відстежуваності та відповідності вимогам GMP/GDP.

6. Сканування штрихкодів і QR-кодів – автоматизація введення даних шляхом сканування кодів товарів за допомогою камери мобільного пристрою, що підвищує точність і зменшує час обробки.

7. Реалізація авторизації користувачів та обмеження доступу – захист даних додатку через реалізацію механізмів входу в систему та диференціацію прав доступу залежно від ролі користувача.

Отже, мобільний додаток є інструментом для підвищення прозорості, оперативності й точності складських і логістичних операцій на фармацевтичному підприємстві, сприяючи автоматизації процесів та зниженню ризиків людських помилок.

2.1.3 Функціональні вимоги до мобільного додатку

Функціональні вимоги до мобільних додатків у фармацевтичній галузі регламентуються низкою нормативних документів, які встановлюють стандарти якості, безпеки та ефективності інформаційних систем. Зокрема, в Україні діють такі нормативні акти:

1. ДСТУ ISO/IEC 25010:2016 «Системи та програмне забезпечення. Вимоги до якості та оцінювання якості» — визначає модель якості програмного забезпечення, включаючи функціональну придатність, надійність, зручність використання, ефективність, підтримуваність та портативність [22].

2. СТ-Н МОЗУ 42-4.3:2011 «Лікарські засоби. Фармацевтична система якості (ICH Q10)» — встановлює вимоги до систем управління якістю у

					БР.КІ-23.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		29

фармацевтичній промисловості, включаючи розробку, виробництво та контроль якості лікарських засобів [23].

3. ДСТУ EN 301 549:2022 «Інформаційні технології. Вимоги щодо доступності продуктів та послуг ІКТ» — визначає вимоги до доступності інформаційно-комунікаційних технологій, включаючи мобільні додатки, для осіб з інвалідністю [24].

4. Загальні технічні вимоги до медичних та фармацевтичних (аптечних) інформаційних систем для підключення до інформаційної системи електронної охорони здоров'я (ЕСОЗ) — встановлюють технічні та функціональні вимоги до інформаційних систем, що взаємодіють з ЕСОЗ, включаючи вимоги до безпеки, сумісності та обміну даними [25].

Таким чином, при розробці мобільного додатку для фармацевтичної компанії необхідно враховувати вимоги зазначених нормативних документів, щоб забезпечити відповідність стандартам якості, безпеки та ефективності у сфері охорони здоров'я.

Також функціональні вимоги до мобільного додатку сформовано з урахуванням особливостей бізнес-процесів фармацевтичної компанії та завдань, що мають бути автоматизовані. Перелік основних функцій додатку розділено на групи:

- складування,
- логістика,
- облік партій,
- користувацькі права доступу.

Це дозволяє структурувати функціонал відповідно до реальних потреб підприємства та забезпечити зручність використання мобільного рішення. Основні функціональні можливості наведено в таблиці 2.1.

					БР.КІ-23.00.00.000 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дат		

Таблиця 2.1 – Основні функції мобільного додатку

Група функцій	Основні функції	Опис
Складування	Перегляд залишків лікарських засобів	Оперативний доступ до інформації про запаси продукції із деталізацією за серіями та партіями.
	Пошук препаратів за назвою, серією, штрихкодом	Швидкий пошук за ключовими атрибутами лікарських засобів.
	Фіксація операцій приймання товару	Реєстрація приймання нових партій товару на склад.
	Проведення інвентаризації	Сканування штрихкодів для перевірки фактичної наявності товарів.
Логістика	Оформлення внутрішніх заявок на переміщення товару	Створення та відправка заявок на міжскладські переміщення.
	Реєстрація операцій відвантаження і приймання	Фіксація фактів переміщення товару зі складу чи на склад.
	Відстеження статусу замовлень і переміщень	Моніторинг поточного стану доставки товарів у системі.
Облік партій	Ідентифікація серій лікарських засобів	Ведення обліку партій і серій для забезпечення відстежуваності.
	Моніторинг термінів придатності	Автоматичне повідомлення про наближення закінчення термінів зберігання продукції.
	Формування переліків продукції для вилучення або утилізації	Підготовка списків товарів із простроченим або близьким до завершення терміном придатності.
Користувацькі права	Автентифікація та авторизація користувачів	Забезпечення входу до системи із перевіркою облікових даних.
	Розмежування прав доступу	Надання доступу до функцій відповідно до ролі користувача (комірник, логіст, адміністратор).

Інтерфейс мобільного додатку повинен забезпечувати високу зручність користування, відповідати вимогам специфіки фармацевтичної галузі та враховувати обмеження мобільних пристроїв. Правильно організований інтерфейс є ключовим фактором для підвищення ефективності роботи користувачів та зниження ризику виникнення помилок під час виконання складських і логістичних операцій. Основні вимоги до інтерфейсу подано в таблиці 2.2.

Таблиця 2.2 – Основні вимоги до інтерфейсу мобільного додатку

Вимога	Опис вимоги
Інтуїтивність	Інтерфейс має бути простим у використанні, мінімізуючи кількість дій для виконання операцій.
Повна локалізація українською мовою	Усі елементи інтерфейсу повинні бути українською мовою з використанням правильної термінології.
Адаптивний дизайн	Коректне відображення на пристроях із різними розмірами екранів (від 5 до 10 дюймів).
Використання візуальних індикаторів	Підсвічування важливої інформації (наприклад, товарів із термінами придатності, що спливають).
Підтримка сканування штрихкодів і QR-кодів	Забезпечення можливості сканування за допомогою камери пристрою для швидкого введення даних.

Мобільний додаток повинен відповідати підвищеним вимогам до безпеки, оскільки обробляє дані про лікарські засоби, партії, а також може взаємодіяти з персональними даними користувачів. Забезпечення конфіденційності, цілісності та доступності інформації є критичним аспектом роботи системи у фармацевтичній галузі. Основні вимоги до безпеки мобільного додатку подано в таблиці 2.3.

Таблиця 2.3 – Основні вимоги до безпеки мобільного додатку

Вимога	Опис вимоги
Захищена передача даних	Усі дані передаються через захищені канали зв'язку (наприклад, HTTPS із використанням SSL/TLS).
Локальне шифрування даних	Збережені дані в локальній базі на пристрої повинні бути зашифровані для запобігання несанкціонованому доступу.
Автентифікація користувачів	Обов'язковий вхід до додатку через логін і пароль із можливістю інтеграції механізму єдиного входу (SSO).
Розмежування прав доступу	Надавати користувачам доступ до функціоналу відповідно до їхніх ролей та повноважень.
Автоматичне блокування сесії	Автоматичне завершення сесії при тривалій бездіяльності для запобігання доступу третіх осіб.
Можливість віддаленого очищення даних	Реалізація механізму очищення локальних даних у разі втрати або крадіжки пристрою.

Таким чином, визначені функціональні, інтерфейсні та безпекові вимоги до мобільного додатку забезпечують відповідність розроблюваної системи особливостям фармацевтичної галузі, сприяють підвищенню ефективності

внутрішніх процесів підприємства та гарантують надійний захист оброблюваної інформації. З урахуванням сформульованих вимог у наступному підрозділі буде розглянуто загальну архітектуру системи та особливості взаємодії її компонентів.

2.1.4 Нефункціональні вимоги до мобільного додатку

Окрім функціональних характеристик, мобільний додаток повинен відповідати ряду нефункціональних вимог, що забезпечують стабільність, продуктивність, сумісність та можливість подальшого розвитку системи. Вимоги сформульовані з урахуванням особливостей експлуатації мобільних рішень у фармацевтичній галузі та представлені в таблиці 2.4.

Таблиця 2.4 – Нефункціональні вимоги до мобільного додатку

Категорія вимог	Опис вимог
Продуктивність	Забезпечення швидкого завантаження інтерфейсу (менше 2 секунд) та обробки основних операцій без затримок на середньостатистичних Android-пристроях (2–4 ГБ оперативної пам'яті).
Сумісність	Підтримка Android-пристроїв із версією операційної системи від Android 8.0 (Oreo) та вище. Коректна робота на смартфонах і планшетах із різними діагоналями екранів.
Надійність	Реалізація офлайн-режиму для основних операцій: перегляд залишків, створення заявок на переміщення з подальшою синхронізацією даних після відновлення інтернет-з'єднання.
Масштабованість	Забезпечення архітектури додатку, що дозволяє легко додавати нові функціональні модулі без необхідності повної переробки існуючої системи. Використання модульного підходу в розробці.

Таким чином, у межах постановки задачі були визначені основні цілі, функціональні та нефункціональні вимоги до мобільного додатку, розробка якого здійснюється з урахуванням особливостей фармацевтичної галузі. При проектуванні системи обов'язково враховуються галузеві обмеження, зокрема вимоги належної виробничої (GMP) та дистриб'юторської практики (GDP), а також положення щодо захисту персональних і медичних даних. Для забезпечення ефективної роботи додатку передбачена інтеграція з існуючими

корпоративними системами підприємства, такими як ERP та WMS, з використанням відкритих стандартів обміну даними, зокрема REST API та форматів JSON. Визначені вимоги слугують основою для подальшого проектування архітектури мобільного рішення, що буде розглянуто у наступному підпункті.

2.2 Архітектура системи

Для забезпечення ефективної роботи мобільного додатку та його взаємодії з корпоративними ресурсами підприємства обрано клієнт-серверну модель архітектури. Такий підхід дозволяє розподілити обробку даних між мобільним клієнтом та серверною частиною, оптимізувати навантаження на пристрій користувача та забезпечити централізоване управління бізнес-логікою і даними. Основні компоненти архітектури та їхнє призначення подано в таблиці 2.5.

Таблиця 2.5 – Основні компоненти архітектури системи

Компонент	Опис	Функціональні особливості
Мобільний клієнт	Android-додаток, розроблений засобами Java	Забезпечення користувацького інтерфейсу, обробка запитів, сканування штрихкодів і QR-кодів
Серверна частина	Корпоративна ERP-система підприємства	Обробка запитів мобільного клієнта, виконання бізнес-логіки, керування обліковими процесами
База даних	Централізоване сховище даних	Зберігання інформації про залишки, партії, серії, замовлення, логістичні операції
Інтерфейс обміну даними	REST API, формат JSON	Забезпечення обміну інформацією між мобільним клієнтом і сервером через захищене з'єднання (HTTPS)

Система побудована за клієнт-серверною архітектурою, де мобільний додаток (клієнт) взаємодіє з серверною частиною через REST API. Це забезпечує ефективний обмін даними, централізоване управління бізнес-логікою та можливість масштабування системи.

Отже, взаємодія компонентів системи виглядає наступним чином:

					БР.КІ-23.00.00.000 ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дат		

1. Користувач ініціює дії через мобільний додаток.

2. Мобільний додаток:

- інтерфейс користувача - забезпечує взаємодію користувача з додатком;
- менеджер запитів - формує та надсилає запити до серверної частини.
- сканер штрихкодів/QR-кодів - зчитує інформацію з упаковок лікарських засобів.

3. Серверна частина:

- REST API сервер - приймає запити від мобільного додатку та передає їх до бізнес-логіки;
- модуль бізнес-логіки - обробляє запити відповідно до бізнес-процесів компанії;
- база даних - зберігає інформацію про лікарські засоби, партії, запаси тощо.

Послідовність взаємодії виглядає наступним чином:

1. Користувач виконує дію в мобільному додатку (наприклад, сканує штрихкод).
2. Менеджер запитів формує відповідний запит у форматі JSON.
3. Запит надсилається через REST API до серверної частини.
4. REST API сервер передає запит до модуля бізнес-логіки.
5. Модуль бізнес-логіки обробляє запит та взаємодіє з базою даних.
6. Результат обробки повертається через REST API до мобільного додатку.
7. Інтерфейс користувача відображає отриману інформацію користувачу.

Для наочного подання структури інформаційної системи та зв'язків між її основними компонентами побудовано UML-діаграму компонентів. Діаграма відображає ключові модулі мобільного додатку та серверної частини, їхню взаємодію та потоки даних, що забезпечують функціонування системи відповідно до поставлених вимог (рис. 2.2).

					БР.КІ-23.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		35

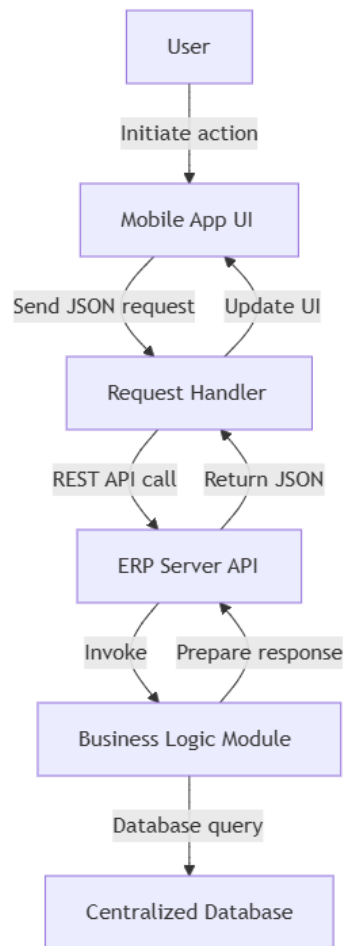


Рисунок 2.2 – UML-діаграма компонентів мобільного додатку та серверної частини системи

Таким чином, побудована архітектура базується на клієнт-серверній моделі з чітким розмежуванням функціональності між мобільним додатком і серверною частиною. Це забезпечує гнучкість, масштабованість та надійність системи, а також дозволяє легко інтегрувати нові функціональні можливості у майбутньому. Визначена структура підтримує вимоги фармацевтичної галузі щодо безпеки даних, відстежуваності партій продукції та відповідності міжнародним стандартам.

У наступному підпункті буде розглянуто проектування бази даних, яка є важливою складовою всієї системи.

2.3 Проектування бази даних

База даних (БД) — центральний елемент інформаційної системи, на який покладається зберігання критичних даних про лікарські засоби, партії, переміщення та користувачів. У клієнт-серверній архітектурі (рис. 2.2) саме БД забезпечує єдине джерело достовірної інформації, а отже підтримує відстежуваність (traceability) продукції, формування регуляторної звітності та оперативну синхронізацію з мобільним додатком і ERP/WMS-модулями. Ключові нефункціональні вимоги до БД наведено в таблиці 2.6.

Таблиця 2.6 – Ключові нефункціональні вимоги до БД

Категорія	Вимога	Пояснення / джерела
Обсяг	≥ 1 млн записів про партії з 5-річним історичним зберіганням	відповідає циклу зберігання фарм-документації, установленому GMP/GDP [26]
Продуктивність	Час відповіді < 300 мс для типових CRUD-операцій; «повільні» запити ≤ 1 с	гарантує безперебійну роботу мобільного клієнта на середніх Android-пристроях
Безпека	Шифрування даних at-rest (AES-256) і in-transit (TLS 1.3); повний audit trail	вимога PIC/S «Guidance on Data Integrity» [27] і FDA CGMP Data Integrity [28]
Регуляторні обмеження	Підтримка атрибутів серії, температурного ланцюга (GMP) та маршрутів поставок (GDP)	визначено моделлю Pharmaceutical Quality System ICH Q10 [26]
Масштабованість	Модульна схема, шардінг таблиць Batch і Movement; версіонування Liquibase	забезпечує безперервний розвиток системи

Отже, роль БД у загальній архітектурі системи наступна:

- зберігання серій і термінів придатності для виконання вимог track & trace та FEFO/FIFO;
- ведення повного audit trail, що забезпечує data integrity згідно з рекомендаціями PIC/S [27] та FDA [28];
- централізований доступ до даних для бізнес-логіки ERP, мінімізація дублювання інформації та можливість масштабування функцій.

Врахування вимог GMP/GDP [26], а також деталізованих настанов з цілісності даних PIC/S [27] та FDA [28] дозволяє спроектувати БД, що забезпечує цілісність, доступність і надійність інформації та відповідає міжнародним стандартам фармацевтичної галузі.

З огляду на сформульовану мету та нефункціональні вимоги до бази даних, наступним кроком є побудова концептуальної моделі, що відображає реальні бізнес-об'єкти фармацевтичної компанії та їх взаємозв'язки. На цьому рівні визначаємо ключові сутності, необхідні для підтримки процесів виробництва, обліку, логістики й контролю якості, а також мінімальний набір атрибутів кожної сутності, достатній для подальшої логічної нормалізації та фізичної реалізації (табл. 2.7).

Таблиця 2.7 – Концептуальна (об'єктна) модель БД

Сутність	Коротка функціональна роль	Основні атрибути (ключові)
Drug	Базовий довідник лікарських засобів	DrugID (PK), TradeName, INN (діюча речовина), DosageForm, Strength, Manufacturer
Batch	Конкретна виробнича серія препарату	BatchID (PK), BatchNumber, ProductionDate, ExpiryDate, DrugID (FK), Status
Warehouse	Локація зберігання продукції	WarehouseID (PK), Name, Address, TemperatureRange, ResponsiblePerson
Inventory	Поточний залишок певної серії на конкретному складі	InventoryID (PK), BatchID (FK), WarehouseID (FK), Quantity, LastUpdate
Movement	Операція переміщення товару між складами / у відвантаження	MovementID (PK), BatchID (FK), FromWarehouseID, ToWarehouseID, Quantity, MoveDate, PerformedBy
User	Авторизований користувач системи	UserID (PK), Login, PasswordHash, FullName, RoleID (FK), IsActive
Role	Набір прав доступу	RoleID (PK), RoleName, Description
AuditLog	Журнал змін і доступів (data integrity)	LogID (PK), UserID (FK), Action, EntityName, EntityID, Timestamp

В таблиці PK – це первинний ключ, FK – зовнішній ключ.

Цей перелік сутностей охоплює:

- довідковий рівень (Drug, Warehouse, Role),
- операційний рівень (Batch, Inventory, Movement),

– користувацький та безпековий рівень (User, AuditLog).

У таблиці 2.8 наведено детальний склад кожної таблиці, типи даних SQL, первинні (PK) і зовнішні (FK) ключі, а також важливі обмеження й індекси. Схеми нормалізовані щонайменше до 3-ї нормальної форми (3НФ), тобто всі непервинні атрибути залежать лише від ключа, відсутні транзитивні залежності, а повторювані групи винесено в окремі таблиці [30].

Таблиця 2.8 – Логічна структура бази даних

Таблиця	Атрибути (тип → обмеження)	Ключі та індекси	Коментар щодо нормалізації
Drug	DrugID INT → PK, AI TradeName VARCHAR(120) → NOT NULL, UNIQUE INN VARCHAR(120) DosageForm VARCHAR(60) Strength VARCHAR(40) Manufacturer VARCHAR(120)	PK (DrugID) UX (TradeName)	3НФ – ознаки виробу описують тільки сам препарат
Batch	BatchID INT → PK, AI BatchNumber VARCHAR(50) → NOT NULL, UNIQUE DrugID INT → FK → Drug(DRUGID) ProductionDate DATE ExpiryDate DATE Status ENUM('Released', 'Quarantine', 'Rejected') DEFAULT 'Released'	PK (BatchID) UX (BatchNumber) IX (DrugID)	3НФ – серії відокремлені від складів і рухів
Warehouse	WarehouseID INT → PK, AI Name VARCHAR(80) → NOT NULL, UNIQUE Address VARCHAR(150) TemperatureRange VARCHAR(20)	PK (WarehouseID) UX (Name)	3НФ – окрема довідкова сутність
Inventory	InventoryID INT → PK, AI BatchID INT → FK → Batch WarehouseID INT → FK → Warehouse Quantity INT → CHECK (Quantity ≥ 0) LastUpdate DATETIME DEFAULT CURRENT_TIMESTAMP	PK (InventoryID) UX (BatchID, WarehouseID) IX (WarehouseID)	3НФ – атрибути залежать лише від «Batch–Warehouse»
Movement	MovementID INT → PK, AI BatchID INT FK → Batch FromWarehouseID INT FK → Warehouse ToWarehouseID INT FK → Warehouse Quantity INT → CHECK (Quantity > 0) MoveDate DATETIME DEFAULT CURRENT_TIMESTAMP	PK (MovementID) IX (BatchID) IX (FromWarehouseID, ToWarehouseID)	3НФ – перелік полів мінімальний, без дублювання назв складів

Таблиця	Атрибути (тип → обмеження)	Ключі та індекси	Коментар щодо нормалізації
Role	RoleID INT → PK, AI RoleName VARCHAR(40) → UNIQUE, NOT NULL Description VARCHAR(120)	PK (RoleID)	3НФ
User	UserID INT → PK, AI Login VARCHAR(60) → UNIQUE, NOT NULL PasswordHash CHAR(60) → NOT NULL FullName VARCHAR(120) RoleID INT FK→Role IsActive BOOLEAN DEFAULT TRUE	PK (UserID) UX (Login) IX (RoleID)	3НФ – права доступу винесено у Role
AuditLog	LogID BIGINT → PK, AI UserID INT FK→User Action VARCHAR(40) EntityName VARCHAR(40) EntityID INT Timestamp DATETIME DEFAULT CURRENT_TIMESTAMP	PK (LogID) IX (UserID)	3НФ – забезпечує вимоги data-integrity GMP / GDP

Також всі таблиці мають додаткові `created_at` / `updated_at` полі (DATETIME), що зберігають службові часові мітки.

Рівні нормалізації описані наступним чином:

- 1НФ — усі атрибути атомарні (наприклад, `Manufacturer` зберігається як одна колонка, а не структурований об'єкт).
- 2НФ — у складених ключах (`Inventory: BatchID, WarehouseID`) не-ключові атрибути (`Quantity`) залежать від усього ключа, а не від частини.
- 3НФ — атрибути не залежать один від одного транзитивно (наприклад, назва складу не дублюється у `Movement`, а витягується через FK).

Такий логічний дизайн мінімізує дублювання даних, спрощує підтримку регуляторних вимог GMP/GDP та забезпечує масштабованість для майбутнього розширення функціоналу.

В логічній моделі фармацевтичної БД використовуються всі базові типи зв'язків — «один-до-одного», «один-до-багатьох» і «багато-до-багатьох». Вони реалізовані первинними / зовнішніми ключами, унікальними індексами й, за потреби, каскадними правилами ON UPDATE / ON DELETE, що спрощують

										Арк.
										40
Змн.	Арк.	№ докум.	Підпис	Дат						

підтримку цілісності даних та відповідають вимогам GMP / GDP до data-integrity (табл. 2.9).

Таблиця 2.9 – Типи й правила зв’язків

Тип	Визначення і джерело	Приклади у нашій схемі
1:1	Кожному рядку батьківської таблиці відповідає рівно один рядок дочірньої; часто реалізують PK = FK + UNIQUE [turn0search1]	User ↔ (UserProfile — за потреби розширення)
1:N	Один запис батька пов’язаний із багатьма записами нащадка [turn0search3]	Drug → Batch; Batch → Inventory; Warehouse → Inventory
M:N	Реалізується проміжною (junction) таблицею, що містить два FK та складений PK [turn0search2]	необхідна для майбутнього модуля «Batch ↔ Document» (серія може належати кільком документам контролю якості)

ON UPDATE CASCADE дозволяє змінювати ключі (рідко в фармацевтичних БД, але корисно для складів з тимчасовими кодами).

ON DELETE CASCADE безпечно застосовується лише до технологічних таблиць, де видалення не порушує регуляторного audit trail; для критичних сутностей (Batch) зазвичай використовують ON DELETE RESTRICT.

Зв’язки в поточній моделі представлено в таблиці 2.10.

Таблиця 2.10 - Зв’язки між об’єктами в розроблюваній моделі

Батьківська таблиця → Дочірня	Кард.	Каскад	Призначення
Drug → Batch	1 : N	RESTRICT / NO ACTION	Один препарат має багато серій
Batch → Inventory	1 : N	CASCADE (UPDATE), RESTRICT (DELETE)	Контроль залишків конкретної серії
Warehouse → Inventory	1 : N	CASCADE (UPDATE), RESTRICT (DELETE)	Зв’язування серії з конкретним складом
Batch → Movement	1 : N	RESTRICT	Історія переміщень серії
Warehouse → Movement	1 : N	RESTRICT	Звідки / куди переміщують товар
Role → User	1 : N	RESTRICT	Ролі задають права доступу
User → AuditLog	1 : N	CASCADE (UPDATE), NO ACTION (DELETE)	Audit-trail для вимог PIC/S [turn0search0]

В додатку А наведено приклади SQL-констрейнтів.

У додатку Б подано повну ER-діаграму логічної моделі БД та її ключові зв'язки, що інтерпретуються наступним чином:

Drug ↔ Batch (1 : N). Один лікарський препарат може мати багато виробничих серій.

Batch ↔ Inventory (1 : N). Кожна серія обліковується у залишках різних складів.

Warehouse ↔ Inventory (1 : N). Кожен склад містить записи про кількість кожної серії, що там зберігається.

Batch ↔ Movement (1 : N). Серія може брати участь у багатьох операціях переміщення товару.

Warehouse ↔ Movement (1 : N). Для кожного переміщення фіксуються склад-джерело та склад-призначення.

Role ↔ User (1 : N). Користувачі успадковують права доступу від призначених ролей.

User ↔ AuditLog (1 : N). Усі дії в системі фіксуються у журналі, що забезпечує вимоги GMP/GDP щодо audit trail.

Діаграма демонструє повну відстежуваність шляху серії лікарського засобу — від довідника препарату через складські запаси та переміщення — і одночасно охоплює контур безпеки (ролі, користувачі, логування), що відповідає нормам ICH Q10 та керівництвом PIC/S щодо цілісності даних.

Для центральної серверної БД обґрунтовано обрати PostgreSQL, а для локального оф-лайн-кешу мобільного клієнта — SQLite. Така гібридна пара дає поєднання промислової масштабованості й багатих можливостей PostgreSQL із легкою, вбудованою й оф-лайн-стійкою природою SQLite [31]

Розглянемо практичне фізичне проектування:

- параметри індексів,
- секціонування,
- бекапи,
- реплікації,

					БР.КІ-23.00.00.000 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дат		

- контроль транзакцій з огляду на GMP/GDP,
- вимоги цілісності даних FDA / PIC/S.

У процесі фізичного проектування бази даних було прийнято рішення використовувати гібридний підхід до вибору систем управління базами даних (СУБД): для серверної частини — PostgreSQL, для мобільного клієнта — SQLite. Такий вибір обумовлений потребами фармацевтичної галузі щодо цілісності даних, підтримки офлайн-режиму та відповідності регуляторним вимогам GMP/GDP. PostgreSQL забезпечує масштабованість, секціонування таблиць, журналювання транзакцій (WAL), логічну реплікацію та гнучке управління транзакційною ізоляцією. Натомість SQLite оптимально підходить для мобільного середовища завдяки легкості, швидкодії та можливості локального шифрування даних (зокрема, із використанням SQLCipher). Комбіноване застосування обох СУБД дозволяє забезпечити як надійність та стабільність серверного зберігання, так і офлайн-функціональність мобільного додатку, що критично важливо для складів без постійного підключення до мережі.

Таблиця 2.11 – Порівняння СУБД у контексті мобільного рішення

Характеристика	PostgreSQL	SQLite
Призначення	Серверна частина (центральна БД)	Локальна БД мобільного додатку
Архітектура	Клієнт-сервер	Вбудована, файлова
Підтримка транзакцій	Повна (ACID, рівень SERIALIZABLE)	Повна, але без багатокористувацької роботи
Реплікація	Логічна, потокова, read-only вузли	Відсутня (можлива ручна синхронізація)
Бекапи та відновлення	PITR, WAL-архів, pg_basebackup	VACUUM INTO, backup API, копія файлу
Секціонування / партиціонування	Так (вбудоване з PG10+)	Ні
Шифрування	pgcrypto, Transparent Data Encryption (TDE)	SQLCipher або обгортки з AES-256
Розмір і продуктивність	Оптимізований для великих систем	Легкий, ідеальний для мобільного клієнта
Відповідність GMP/GDP	Висока (журнали, контроль доступу, аудит)	Може застосовуватись як тимчасове сховище з шифруванням

Ефективність виконання запитів у фармацевтичній інформаційній системі, особливо при зростанні обсягів даних (серії, переміщення, залишки), значною мірою залежить від грамотно реалізованої стратегії індексування та секціонування таблиць. З урахуванням вимог до продуктивності, частоти доступу до окремих таблиць, а також необхідності зберігання історичних даних відповідно до норм GMP/GDP, для основних таблиць бази даних було запроєктовано різні типи індексів і секцій (табл. 2.12).

Таблиця 2.12 – Індксація та секціонування ключових таблиць PostgreSQL

Таблиця	Індекси	Тип секціонування	Призначення / обґрунтування
batch	UNIQUE (batch_number) INDEX (drug_id)	Немає	Пошук за номером серії; фільтрація за типом препарату
inventory	UNIQUE (batch_id, warehouse_id)	Немає	Контроль залишків у складі по серії
movement	INDEX (batch_id) INDEX (move_date)	RANGE по move_date	Високочастотні запити, агрегація переміщень за період
auditlog	INDEX (user_id) BRIN (timestamp)	RANGE по timestamp	Стиснена індксація для великих журналів; щомісячне секціонування
user	UNIQUE (login)	Немає	Забезпечення унікальності облікових записів

Ключові аспекти реалізації заключаються в наступному:

1. Для таблиць з великою кількістю записів (movement, auditlog) використано секціонування типу RANGE за датою (move_date, timestamp) із розбиттям на щомісячні або щотижневі партиції. Це спрощує обслуговування даних (VACUUM, архівація) та прискорює виконання звітних запитів.

2. Таблиця auditlog, як правило, має значні обсяги історичних записів. Для неї використано BRIN-індекс, який суттєво зменшує розмір структури індексу при збереженні ефективного доступу за часовими діапазонами.

3. Індекси створено лише там, де запити регулярно використовують фільтрацію або сортування за певним полем, щоби уникнути надмірного навантаження на INSERT/UPDATE-операції.

Додаткові заходи, які необхідні при розробці БД:

1. Валідація бекапів - щомісячне тестове відновлення з копії (регламентоване внутрішнім SOP), що відповідає вимогам FDA Data Integrity [28].

2. Контроль доступу - доступ до файлів WAL, журналів аудиту та бекапів мають лише адміністратори з відповідними ролями, відповідно до принципу мінімального доступу (least privilege).

3. Інтеграція з журналом AuditLog: транзакції з критичними діями (UPDATE, DELETE) завжди супроводжуються записом в таблицю auditlog, що забезпечує відповідність ALCOA+ принципам (Attributable, Legible, Contemporaneous, Original, Accurate, Complete) [26-27].

Таким чином, реалізовані стратегії бекапів, реплікації та контролю транзакцій забезпечують відмовостійкість, цілісність і трасованість даних у системі, що є критичними для відповідності фармацевтичній регуляції на міжнародному рівні (GMP, GDP, ICH Q10, PIC/S, FDA).

Для дотримання вимог регуляторних документів (GMP, GDP, Закон України «Про захист персональних даних» [13], ICH Q10 [26]), у базі даних впроваджено низку технічних механізмів захисту та контролю цілісності даних:

- контроль цілісності: усі зовнішні ключі мають правила ON DELETE RESTRICT, щоб запобігти випадковому видаленню критичних записів (наприклад, серій). Встановлені обмеження CHECK для кількісних полів (наприклад, quantity >= 0);

- автоматичний аудит: зміни в системі фіксуються в таблиці auditlog із записом користувача, сутності, дії та часу. Це забезпечує відповідність принципам ALCOA+ [27];

- рольовий контроль доступу: доступ до даних обмежено на рівні ролей користувачів (admin, warehouse_staff, viewer), що відповідає принципу мінімального привілею;

					БР.КІ-23.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		46

– шифрування: локальні БД (SQLite) шифруються з використанням AES-256 (через SQLCipher), а передача даних між клієнтом і сервером здійснюється через HTTPS.

Перерахувагі заходи забезпечують надійний захист персональних і критичних виробничих даних, а також гарантують достовірність усіх транзакцій у системі.

Архітектура бази даних проєктувалась із врахуванням майбутнього зростання обсягу даних, інтеграції нових підсистем та модулів, а також еволюції бізнес-вимог. Забезпечення розширюваності означає, що структура БД має бути гнучкою до змін без необхідності повного рефакторингу або втрати сумісності з існуючими даними та логікою.

1. Додавання нових сутностей

Проєктована модель дозволяє безпечно додавати нові сутності з мінімальним втручанням у вже реалізовану структуру.

Наприклад, TemperatureLog – таблиця, яка може зберігати значення температури з датчиків для кожного складу або навіть партії товару. Її можна реалізувати як:

```
Table temperature_log {
  id int [pk, increment]
  warehouse_id int [ref: > warehouse.warehouse_id]
  batch_id int [ref: > batch.batch_id, null]
  value_celsius decimal(5,2)
  recorded_at datetime
}
```

Така реалізація не потребує змін у таблицях batch або warehouse, а зв'язки встановлюються через зовнішні ключі.

2. Версіонування схеми БД

Щоб уникнути конфліктів між версіями застосунку та бази даних під час оновлень, рекомендовано використовувати інструменти керування версіями БД, зокрема:

					БР.КІ-23.00.00.000 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дат		

– Liquibase (платно) – дозволяє зберігати всі зміни схеми (створення таблиць, індексів, модифікації) як послідовні changelog-файли (у форматі XML, YAML або SQL), що виконуються поетапно.

– Flyway (безкоштовно) – спрощений інструмент з підтримкою скриптів міграції, які запускаються автоматично при оновленні проєкту.

3. Уніфікація доступу та розширення API

Для кожної нової сутності (наприклад, логування температури, контроль вологості, моніторинг доступу) можливо легко реалізувати відповідні REST-ендпоінти в API без зміни базової логіки. Завдяки REST-орієнтованій архітектурі, додаткові модулі не потребують змін у існуючому функціоналі мобільного застосунку, що відповідає принципам modularity та low coupling.

Таким чином, гнучка модель БД, підтримка зовнішніх ключів, інструменти версіонування та модульна архітектура забезпечують системі високу адаптивність до змін, що є критично важливим у динамічних умовах фармацевтичного обліку та зберігання продукції.

У результаті проєктування бази даних було сформовано логічну та фізичну структуру, яка повністю відповідає функціональним та нефункціональним вимогам інформаційної системи фармацевтичної компанії. Запропонована модель враховує особливості обліку серій лікарських засобів, контролю залишків, операцій переміщення, управління користувачами та забезпечення аудиту відповідно до вимог GMP, GDP та ICH Q10.

База даних проєктована у 3-й нормальній формі, що гарантує відсутність надлишковості та підвищує цілісність даних. Враховано аспекти безпеки, розмежування доступу, резервного копіювання, реплікації та контролю транзакцій, що забезпечує відповідність стандартам цілісності даних (ALCOA+). Реалізована структура є масштабованою та підтримує версіонування змін, що дозволяє ефективно розширювати її у разі появи нових бізнес-завдань.

					БР.КІ-23.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		48

3 РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ

У цьому розділі розглядається процес практичної реалізації мобільного додатку для фармацевтичної компанії відповідно до розробленої в попередньому етапі архітектури та структури бази даних. Особливу увагу приділено вибору інструментів, налаштуванню середовища розробки, розгортанню необхідної інфраструктури та поетапній реалізації ключового функціоналу. Весь перелік програмних засобів, бібліотек і технологій, які використовуються для реалізації застосунку, наведено в додатку В у вигляді структурованого документа requirements.md, що забезпечує відтворюваність середовища розробки та його відповідність сучасним вимогам Android-програмування й регуляторним стандартам фармацевтичної галузі.

3.1 Функціональні можливості мобільного додатку

Розроблений мобільний додаток є частиною інформаційної системи фармацевтичної компанії, призначеної для автоматизації обліку лікарських засобів, управління залишками, контролю партій і логістичних операцій. У цьому підрозділі наведено огляд реалізованих функціональних можливостей мобільного застосунку та опис основних екранів, навігації й обробки подій.

Інтерфейс мобільного додатку побудований у відповідності до принципів Material Design, забезпечуючи інтуїтивність та зручність взаємодії для кінцевого користувача. Основні екрани реалізовані за допомогою Activity та RecyclerView, із підтримкою асинхронного оновлення даних за допомогою LiveData та ViewModel, що відповідає архітектурному шаблону MVVM.

Основні функціональні екрани:

1. Головний екран (рис. 3.1).

Відображає список наявних препаратів на складах. Для кожної позиції виводиться:

					БР.КІ-23.00.00.000 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дат		

- назва препарату та виробник;
- номер партії та термін придатності;
- кількість одиниць у наявності;
- назва складу зберігання.

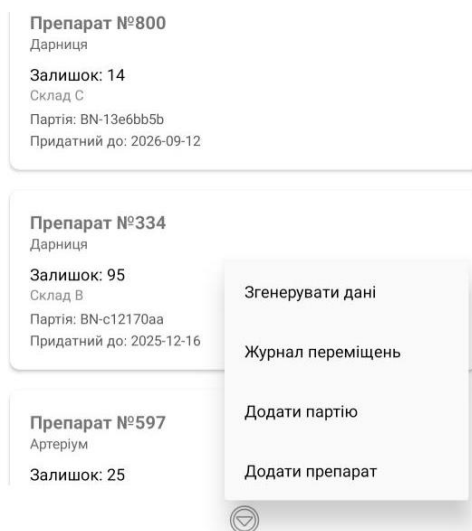


Рисунок 3.1 - Головне вікно додатку

Список реалізовано за допомогою InventoryAdapter, який використовує модель InventoryWithDrugAndBatch, що дозволяє виводити зведену інформацію з кількох таблиць бази даних.

2. Екран додавання препарату

Користувач має можливість додати новий препарат, вказавши назву, виробника, дозування та форму випуску. Перевірка коректності введення виконується перед записом у локальну БД (рис. 3.2).

The screenshot shows the 'MyPharmApp' interface for adding a medicine. It features a purple header with the app name, two input fields for 'Назва препарату' and 'Виробник', and a grey button labeled 'ЗБЕРЕГТИ ПРЕПАРАТ'.

Рисунок 3.2 - Екран додавання препарату

					БР.КІ-23.00.00.000 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дат		

3. Екран додавання партії (рис. 3.3)

Для вибраного препарату можна створити нову партію із зазначенням:

- номера партії;
- дати виробництва;
- терміну придатності;
- складу та початкової кількості.

У полі вибору препарату (Spinner) реалізовано підвантаження наявних препаратів із бази даних, що виключає помилки користувача та спрощує вибір.

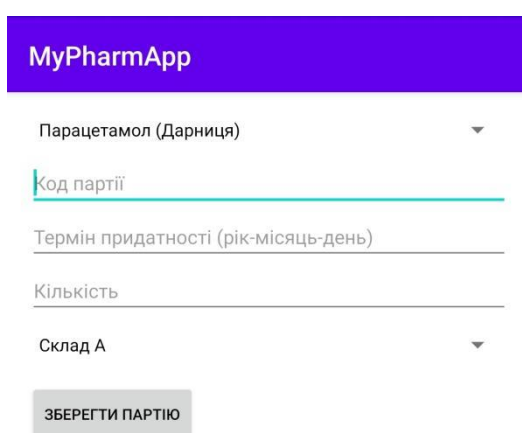


Рисунок 3.3 - Екран додавання партії препаратів

4. Діалог переміщення препарату (рис. 3.4)

Відкривається при натисканні на будь-який елемент у списку на головному екрані. Користувач вказує:

- кількість для переміщення;
- склад призначення (обирається зі списку Spinner).

Після підтвердження переміщення:

- зменшується кількість на поточному складі;
- оновлюється або створюється запис у складі-призначенні;
- записується операція в журнал MovementEntity.

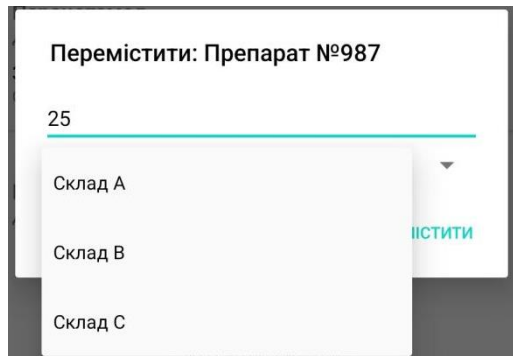


Рисунок 3.4 - Екран меню переміщення препарату

5. Журнал переміщень (рис. 3.5)

Окремий екран дозволяє переглянути історію логістичних операцій: дату, препарат, склад відправлення та призначення, кількість. Реалізовано через MovementAdapter із відображенням усіх ключових полів.

Журнал переміщень

Препарат ID: 4
 Кількість: 35
 З: Склад В → ДО: Склад Б
 2025-05-02 02:38

Рисунок 3.5 - Екран журналу переміщень препаратів

З метою покращення зручності інтерфейсу кнопки «Додати препарат», «Додати партію», «Перегляд журналу» та «Згенерувати тестові дані» винесено у випадаюче меню або нижню панель навігації.

6. Обробка подій

Уся обробка подій у додатку реалізована через `setOnClickListener` у відповідних активностях, а для переміщень та додавання — через `AlertDialog` або окремі форми. Для тривалих операцій (наприклад, оновлення залишків або запис у базу) використовується `CoroutineScope` з контекстом `Dispatchers.IO`, що гарантує некритичність до ANR (`Application Not Responding`).

Також реалізовано:

- LiveData-оновлення інтерфейсу при зміні вмісту бази даних;
- валидацію користувацьких даних;

					БР.КІ-23.00.00.000 ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дат		

– вивід повідомлень через Toast.makeText.

Можливості розширення функціоналу

Запропонована архітектура додатку побудована за принципами модульності, що дозволяє без проблем реалізувати додатковий функціонал у разі необхідності. Використання патерну MVVM та бібліотеки Android Jetpack (Room, LiveData, ViewModel) гарантує зручне внесення змін та розширення функцій без суттєвих змін у структурі додатку.

Для демонстрації в рамках дипломної роботи реалізованого функціоналу цілком достатньо, оскільки він повністю відповідає завданням автоматизації обліку лікарських засобів та контролю переміщень. Подальший розвиток може включати, наприклад, інтеграцію зі сторонніми API, додаткові аналітичні звіти, підтримку авторизації користувачів з різними ролями, або розширення аналітичних можливостей системи.

3.2 Технічна реалізація програмного рішення

У цьому підпункті розглянуто технічні аспекти створення мобільного додатку, що реалізує функціональність інформаційної системи фармацевтичної компанії. Проєкт було реалізовано з використанням середовища розробки IntelliJ IDEA та мови програмування Kotlin з підтримкою бібліотек Android Jetpack. Під час розробки застосовано сучасні архітектурні підходи, зокрема шаблон MVVM, що забезпечує поділ відповідальностей, гнучкість та легкість розширення функціоналу. У підпункті докладно описано структуру проєкту, призначення ключових компонентів, підключені бібліотеки та основні принципи реалізації взаємодії з базою даних і графічним інтерфейсом.

3.2.1 Створення та налаштування проєкту

Розробку мобільного додатку було розпочато зі створення нового Android-проєкту у середовищі IntelliJ IDEA, яке надає повноцінну підтримку

					БР.КІ-23.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		53

інструментів для створення Android-додатків. При створенні проєкту було обрано шаблон Empty Activity, що дозволяє вручну визначити компоненти інтерфейсу та логіку роботи застосунку, без нав'язування заздалегідь підготовленої структури.

Як мову програмування обрано Kotlin, оскільки вона є офіційно рекомендованою для Android-розробки, має лаконічний синтаксис, підтримує функціональні підходи, а також повністю сумісна з Java. Для керування залежностями та конфігурацією проєкту використовується Gradle, що забезпечує гнучке підключення зовнішніх бібліотек і модулів.

У конфігураційному файлі build.gradle (Module: app) було підключено такі основні бібліотеки (додаток Г):

- androidx.room:room-runtime — для роботи з базою даних SQLite через абстракцію Room;
- androidx.lifecycle:lifecycle-viewmodel-ktx — для реалізації архітектурного шаблону MVVM;
- androidx.lifecycle:lifecycle-livedata-ktx — для реактивного оновлення інтерфейсу;
- org.jetbrains.kotlinx:kotlinx-coroutines-android — для асинхронної обробки операцій з використанням корутин;
- androidx.recyclerview:recyclerview — для побудови списків у вигляді карток;
- com.google.android.material:material — для візуальних компонентів, відповідних до Material Design.

Також було встановлено мінімальну підтримувану версію Android SDK — API 24 (Android 7.0), що дозволяє підтримувати широке коло сучасних пристроїв.

Після генерації проєкту IntelliJ IDEA автоматично створює базову структуру директорій, яку в подальшому було розширено відповідно до потреб

					БР.КІ-23.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		54

архітектури додатку (таблиця 3.1). В додатку Д наведено загальний вигляд структури проєкту.

Таблиця 3.1 - Базова структура директорій

Файл / Директорія	Призначення
AddBatchActivity.kt	Додавання нової партії препарату до бази даних
AddDrugActivity.kt	Додавання нового препарату до системи
MainActivity.kt	Головна активність: перегляд залишків і переміщення
MovementLogActivity.kt	Перегляд журналу переміщень препаратів
adapter/InventoryAdapter.kt	Адаптер для відображення залишків у RecyclerView
adapter/MovementAdapter.kt	Адаптер для журналу переміщень
data/dao/BatchDao.kt	DAO для доступу до таблиці партій
data/dao/DrugDao.kt	DAO для доступу до таблиці препаратів
data/dao/InventoryDao.kt	DAO для доступу до залишків
data/dao/MovementDao.kt	DAO для доступу до журналу переміщень
database/AppDatabase.kt	Оголошення Room-бази даних та зв'язок з DAO
database/DatabaseProvider.kt	Ініціалізація бази даних (Singleton)
model/BatchEntity.kt	Сутність партії препарату
model/DrugEntity.kt	Сутність препарату
model/InventoryEntity.kt	Сутність залишку на складі
model/InventoryWithDrugAndBatch.kt	Зв'язана модель: залишок + препарат + партія
model/MovementEntity.kt	Сутність переміщення між складами
ui/theme/Color.kt	Автоматично згенерований файл (не використовується)
ui/theme/Theme.kt	Автоматично згенерований файл (не використовується)
ui/theme/Type.kt	Автоматично згенерований файл (не використовується)
utils/FakeDataGenerator.kt	Генерація тестових даних
viewmodel/DrugViewModel.kt	ViewModel для роботи з препаратами
viewmodel/InventoryViewModel.kt	ViewModel для роботи з залишками
viewmodel/MovementViewModel.kt	ViewModel для журналу переміщень

Таким чином, на етапі створення та первинного налаштування мобільного додатку було обрано ефективне середовище розробки (IntelliJ IDEA), мову програмування Kotlin, а також сучасні інструменти Android

Jetpack, що дозволили забезпечити стабільну основу для реалізації функціональності системи. Завдяки використанню шаблону Empty Activity, мінімальної структури та гнучкої конфігурації Gradle, проєкт отримав чітку архітектуру, яку легко підтримувати й розширювати. У подальших підпунктах розглянуто структуру проєкту, використані шаблони проєктування та реалізовані компоненти.

3.2.2 Структура та архітектура мобільного додатку

Після створення основи проєкту, його структура була логічно організована відповідно до вимог архітектурного шаблону MVVM (Model-View-ViewModel) та принципів чистої архітектури. Такий підхід дозволив розділити відповідальність між окремими компонентами додатку, що значно спрощує його підтримку, тестування та масштабування.

Загальна структура проєкту наведена у таблиці 3.1. Вона включає окремі пакети для моделей (model), доступу до бази даних (database і dao), адаптерів для відображення списків (adapter), представлень (Activity), логіки взаємодії з даними (viewmodel) і допоміжних утиліт (utils). Коротко опишемо призначення основних пакетів:

adapter — містить адаптери для RecyclerView, що відповідають за відображення залишків (InventoryAdapter) та записів переміщення (MovementAdapter).

data/dao — включає DAO-інтерфейси, які забезпечують доступ до таблиць бази даних: DrugDao, BatchDao, InventoryDao, MovementDao.

database — включає AppDatabase, який описує конфігурацію бази даних Room, та DatabaseProvider, що реалізує патерн Singleton для ініціалізації БД.

model — містить сутності (Entity) для таблиць, а також моделі з анотаціями @Relation (InventoryWithDrugAndBatch), що забезпечують агреговане відображення даних;

viewmodel — містить ViewModel-класи, які зв'язують базу даних з UI-компонентами через LiveData: DrugViewModel, InventoryViewModel, MovementViewModel;

utils — містить допоміжні функції, зокрема FakeDataGenerator для генерації тестових записів у базу даних.

В основній директорії розміщено ключові активності (MainActivity, AddBatchActivity, AddDrugActivity, MovementLogActivity), які реалізують екранну взаємодію з користувачем. Навігація між ними здійснюється через Intent.

Структура проєкту побудована відповідно до сучасних практик розробки Android-додатків і дозволяє легко масштабувати функціональність.

3.2.3 Використані бібліотеки та технології

У процесі розробки мобільного додатку було використано набір сучасних бібліотек та технологій, що входять до екосистеми Android Jetpack, а також стандартні інструменти платформи Kotlin. Це дозволило забезпечити стабільну, масштабовану та зручну у підтримці архітектуру проєкту. Кожен компонент виконує чітко визначену роль у взаємодії з даними та інтерфейсом користувача. Основні бібліотеки, що були використані в роботі зазначені в таблиці 3.2.

Таблиця 3.2 - Перелік використаних бібліотек

Бібліотека	Призначення
androidx.room:room-runtime	Абстракція над SQLite, спрощення роботи з базою даних
androidx.lifecycle:lifecycle-viewmodel-ktx	Реалізація ViewModel для збереження стану та логіки
androidx.lifecycle:lifecycle-livedata-ktx	Реактивна прив'язка UI до даних у базі
org.jetbrains.kotlin:kotlinx-coroutines-android	Асинхронна робота з базою даних без блокування UI
androidx.recyclerview:recyclerview	Виведення списків у вигляді динамічних карток
com.google.android.material:material	Компоненти інтерфейсу користувача відповідно до Material Design

Додатково використані технології:

- Kotlin — основна мова розробки, рекомендована Google для Android
- Gradle — система збирання проєкту і керування залежностями
- ViewModel + LiveData — архітектурний шаблон MVVM
- Room (DAO, Entity, Relation) — ORM-рівень для роботи з базою
- CoroutineScope / Dispatchers.IO — асинхронна обробка даних у фоновому потоці
- Spinner, AlertDialog, RecyclerView — компоненти для взаємодії з користувачем.

Усі бібліотеки інтегровані через систему Gradle, що забезпечує автоматичне завантаження останніх стабільних версій і зменшує складність підтримки проєкту.

3.2.4 Опис основних класів та їхнього призначення

Розроблений мобільний додаток має чітку структурну організацію, де кожен клас виконує окрему роль у взаємодії з користувачем, базою даних або відображенням даних. У цьому підпункті наведено короткий опис основних класів, реалізованих у межах проєкту.

Архітектура додатку ґрунтується на шаблоні MVVM, що дозволяє забезпечити поділ логіки, моделі даних та інтерфейсу. Для відображення структури додатку в цілому у Додатку Е наведено UML-діаграму класів, яка відображає основні сутності, їх поля, зв'язки між класами та спадковість.

Активності (Activity):

- MainActivity.kt — головна активність застосунку, що відповідає за відображення залишків лікарських засобів на складах. Тут також реалізовано логіку переміщення товарів між складами через спливаючий діалог.
- AddDrugActivity.kt — активність для додавання нового лікарського препарату до системи. Забезпечує введення назви, виробника, дозування.

					БР.КІ-23.00.00.000 ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дат		

- AddBatchActivity.kt — активність для створення нової партії із зазначенням препарату, дати виготовлення, терміну придатності, кількості та складу.

- MovementLogActivity.kt — окрема активність для перегляду історії переміщень товарів між складами.

Адаптери:

- InventoryAdapter.kt — адаптер для RecyclerView, що використовується в MainActivity для відображення залишків. Отримує на вхід модель InventoryWithDrugAndBatch.

- MovementAdapter.kt — адаптер для відображення списку операцій переміщення в MovementLogActivity.

DAO-класи (data/dao):

- DrugDao.kt — інтерфейс доступу до таблиці препаратів.

- BatchDao.kt — забезпечує операції над партіями (додавання, отримання по ID, тощо).

- InventoryDao.kt — дозволяє працювати з залишками на складах, в тому числі з @Relation.

- MovementDao.kt — обробка історії переміщень препаратів.

Моделі (model):

- DrugEntity.kt — опис структури таблиці препаратів.

- BatchEntity.kt — модель партії препарату з датою виготовлення та терміном придатності.

- InventoryEntity.kt — модель залишку, що містить кількість, ID препарату та партії, склад.

- MovementEntity.kt — модель запису операції переміщення між складами.

- InventoryWithDrugAndBatch.kt — агрегована модель з використанням @Relation, яка об'єднує інформацію про препарат, партію та залишок.

ViewModel (viewmodel):

					БР.КІ-23.00.00.000 ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дат		

– `DrugViewModel.kt` — забезпечує передачу даних між `AddDrugActivity` та базою даних.

– `InventoryViewModel.kt` — використовується в `MainActivity` для відображення залишків у реальному часі через `LiveData`.

– `MovementViewModel.kt` — `ViewModel` для журналу переміщень.

Інші допоміжні класи:

– `AppDatabase.kt` — абстрактний клас `Room`, який об'єднує всі DAO-інтерфейси.

– `DatabaseProvider.kt` — реалізація патерну `Singleton` для доступу до бази з будь-якої частини програми.

– `FakeDataGenerator.kt` — генератор тестових записів для демонстраційних цілей.

Кожен із вказаних класів взаємодіє з іншими згідно з принципами MVVM та відповідає за конкретну область відповідальності в архітектурі додатку.

3.2.5 Опис фрагментів коду

Для забезпечення роботи мобільного додатку було реалізовано низку ключових методів, які відповідають за взаємодію з базою даних, обробку введених користувачем даних, обчислення залишків та реєстрацію переміщень. Розглянемо найбільш суттєві фрагменти коду, які забезпечують функціональність системи.

1. Додавання препарату (`AddDrugActivity.kt`)

Користувач вводить назву та виробника препарату у відповідні поля. Після натискання кнопки «Зберегти» запускається корутина, яка викликає DAO-метод:

```
CoroutineScope(Dispatchers.IO).launch {  
    val drug = DrugEntity(name = name, manufacturer =  
manufacturer)
```

					БР.КІ-23.00.00.000 ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дат		

```

DatabaseProvider.getDatabase(applicationContext).drugDao().insert(
drug)

    runOnUiThread {
        Toast.makeText(this@AddDrugActivity, "Препарат
збережено", Toast.LENGTH_SHORT).show()
        finish()
    }
}

```

Таким чином, запис виконується асинхронно, що не блокує інтерфейс користувача.

2. Додавання партії (AddBatchActivity.kt)

Після вибору препарату зі Spinner і введення інших параметрів, створюється новий запис у таблиці Batch, а також відповідний запис у Inventory:

```

val batch = BatchEntity(
drugId = selectedDrug.id,
batchNumber = code,
manufactureDate = LocalDate.now().toString(),
expirationDate = expiration
)
val batchId = db.batchDao().insert(batch).toInt()
val inventory = InventoryEntity(
drugId = selectedDrug.id,
batchId = batchId,
quantity = quantity,
warehouse = warehouse
)
db.inventoryDao().insert(inventory)

```

3. Переміщення препарату між складами (MainActivity.kt)

При підтвердженні переміщення система:

1. Зменшує залишок на поточному складі.
2. Додає або оновлює запис на складі призначення.
3. Реєструє запис у журналі переміщень:

```

val currentInventory =
inventoryDao.findByDrugAndWarehouse(drugId, fromWarehouse)

```

					БР.КІ-23.00.00.000 ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дат		

```

        if (currentInventory != null && currentInventory.quantity >=
quantity) {
            inventoryDao.update(currentInventory.copy(quantity =
currentInventory.quantity - quantity))

            val destination =
inventoryDao.findByDrugAndWarehouse(drugId, toWarehouse)
            if (destination != null) {
                inventoryDao.update(destination.copy(quantity =
destination.quantity + quantity))
            } else {
                inventoryDao.insert(InventoryEntity(...))
            }

            movementDao.insert(MovementEntity(...))
        }

```

Цей механізм гарантує цілісність даних і логічну правильність переміщень.

4. DAO-запити

Усі запити до бази реалізовані через інтерфейси DAO з використанням анотацій Room:

- отримання всіх залишків з препаратом і партією:

```

@Transaction
@Query("SELECT * FROM inventory")
fun getAllWithDrugAndBatch():
Flow<List<InventoryWithDrugAndBatch>>
    - пошук залишку за складом і препаратом:

@Query("SELECT * FROM inventory WHERE drugId = :drugId AND
warehouse = :warehouse LIMIT 1")
suspend fun findByDrugAndWarehouse(drugId: Int, warehouse:
String): InventoryEntity?

```

Такі запити забезпечують швидку та зручну роботу з базою, мінімізуючи кількість коду, необхідного для обробки SQL.

5. Реактивне оновлення інтерфейсу (InventoryViewModel.kt)

Для автоматичного оновлення головного списку при зміні бази використовується LiveData, яке спостерігає за потоком Flow:

```

val inventoryList: LiveData<List<InventoryWithDrugAndBatch>> =
inventoryDao.getAllWithDrugAndBatch().asLiveData()

```

Це дозволяє UI автоматично реагувати на зміни даних без явного оновлення адаптера вручну.

					БР.КІ-23.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		62

Показані фрагменти демонструють узгоджену роботу між частинами проекту — від введення даних до їх відображення й збереження. Така реалізація відповідає вимогам до якісних Android-додатків і забезпечує високу масштабованість.

3.2.6 Опис використаних шаблонів проєктування

У процесі розробки мобільного додатку було використано кілька перевірених шаблонів проєктування (design patterns), що дозволяють організувати код зрозуміло, гнучко та масштабовано. Застосування таких шаблонів позитивно впливає на підтримуваність, тестованість і розширюваність програмного рішення.

1. MVVM (Model–View–ViewModel)

Це основний архітектурний шаблон, який використовується в Android-додатках. У рамках реалізації:

- Model — це сутності бази даних (DrugEntity, BatchEntity, InventoryEntity, тощо), а також DAO-інтерфейси, які відповідають за доступ до даних.
- View — активності (MainActivity, AddDrugActivity, AddBatchActivity), які відповідають за інтерфейс користувача.

ViewModel — InventoryViewModel, DrugViewModel,

MovementViewModel, які опосередковують взаємодію між View і Model та містять логіку обробки даних.

Завдяки MVVM було досягнуто чіткого поділу обов'язків, зменшення залежності між компонентами та спрощення тестування.

Шаблон Singleton застосовано для створення лише одного екземпляра бази даних у всьому застосунку. Це реалізовано у класі DatabaseProvider.kt, який гарантує, що база даних ініціалізується лише один раз:

```
object DatabaseProvider {
    private var INSTANCE: AppDatabase? = null

    fun getDatabase(context: Context): AppDatabase {
        return INSTANCE ?: synchronized(this) {
            val instance = Room.databaseBuilder(...)
        }
    }
}
```

									Арк.
									63
Змн.	Арк.	№ докум.	Підпис	Дат					

```

        INSTANCE = instance
        instance
    }
}
}

```

Такий підхід забезпечує централізований та безпечний доступ до бази.

3. Adapter

Шаблон Adapter реалізований у класах InventoryAdapter та MovementAdapter, які перетворюють список моделей у візуальні елементи RecyclerView. Завдяки адаптерам можна змінювати спосіб виводу інформації, не змінюючи самі моделі даних.

4. Repository (умовно)

Хоча окремий клас Repository не було створено, його функції виконують DAO-інтерфейси (DrugDao, InventoryDao, тощо). DAO відповідають за абстрагування джерела даних, і виступають посередником між базою Room та ViewModel.

5. Factory (непрямо)

ViewModel-и створюються автоматично за допомогою by viewModels(), що використовує внутрішню реалізацію фабричного патерну (ViewModelProvider.Factory) в Android Jetpack. Це забезпечує збереження стану ViewModel під час життєвого циклу активності.

6. Observer

LiveData реалізує шаблон Observer: ViewModel передає потік даних, а View (Activity) спостерігає за ним через observe(...). У результаті інтерфейс автоматично оновлюється при зміні даних у базі.

Таким чином, у додатку поєднано декілька важливих шаблонів, які в сукупності формують стабільну, підтримувану та легко розширювану архітектуру.

					БР.КІ-23.00.00.000 ПЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дат		

3.3 Тестування додатку та аналіз результатів

Після завершення реалізації функціональних модулів мобільного додатку було проведено тестування з метою перевірки коректності роботи основних сценаріїв використання, а також стабільності додатку під час взаємодії з базою даних. Тестування проводилося вручну на етапі розробки за допомогою запуску додатку у середовищі IntelliJ IDEA на вбудованому емуляторі Android, а також на фізичному пристрої з операційною системою Android 10.

1. Компіляція та запуск проєкту

Компіляція проєкту виконувалася засобами Gradle через IntelliJ IDEA. Для перевірки працездатності збірки використовувалася команда:

```
./gradlew clean build
```

Проєкт успішно компілюється без помилок (рис. 3.6), що свідчить про відсутність критичних синтаксичних або типових помилок. Основний застосунок було зібрано з мінімальним рівнем API 24 (Android 7.0), що дозволяє запускати його на більшості сучасних пристроїв.

```
PS D:\MyPharmApp> ./gradlew clean build

> Task :app:compileReleaseKotlin
w: file:///D:/MyPharmApp/app/src/main/java/com/example/mypharmapp/utills/FakeDataGenerator.kt:35:21 Variable 'expirationDate' is never used

> Task :app:compileDebugKotlin
w: file:///D:/MyPharmApp/app/src/main/java/com/example/mypharmapp/utills/FakeDataGenerator.kt:35:21 Variable 'expirationDate' is never used

> Task :app:lintReportDebug
Wrote HTML report to file:///D:/MyPharmApp/app/build/reports/lint-results-debug.html

Deprecated Gradle features were used in this build, making it incompatible with Gradle 9.0.

You can use '--warning-mode all' to show the individual deprecation warnings and determine if they come from your own scripts or plugins.

For more on this, please refer to https://docs.gradle.org/8.7/userguide/command\_line\_interface.html#sec:command\_line\_warnings in the Gradle documentation.

BUILD SUCCESSFUL in 27s
106 actionable tasks: 105 executed, 1 up-to-date
```

Рисунок 3.6 - Скріншот консолі терміналу

2. Генерація тестових даних

					БР.КІ-23.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		65

З метою наповнення бази та демонстрації роботи основного функціоналу було реалізовано клас FakeDataGenerator.kt, який автоматично додає до бази випадкові препарати, партії та залишки:

```
fun generateFakeData(context: Context, count: Int = 10) {
    CoroutineScope(Dispatchers.IO).launch {
        val db = DatabaseProvider.getDatabase(context)
        repeat(count) {
            val drugId = db.drugDao().insert(...)
            val batchId = db.batchDao().insert(...)
            db.inventoryDao().insert(...)
        }
    }
}
```

Ця функція викликається через пункт меню в головній активності. Після її виконання в базі з'являються тестові записи, які відображаються у списку залишків та можуть бути переміщені між складами.

3. Тестування функціональних сценаріїв

Під час тестування було перевірено сценарії, зазначені в таблиці 3.3.

Таблиця 3.3 - Сценарії тестування

Сценарій	Результат
Додавання нового препарату	Препарат зберігається, з'являється у списку при додаванні партії
Додавання партії препарату	Створюється партія та залишок у базі
Переміщення препарату між складами	Кількість змінюється, створюється/оновлюється
Перегляд журналу переміщень	Усі переміщення відображаються в окремому списку
Вивід номеру партії та терміну придатності	Дані коректно відображаються на головному екрані
Робота з LiveData та оновлення списку	Інтерфейс оновлюється автоматично при зміні бази даних

4. Аналіз стабільності та ефективності

Завдяки використанню корутин (Dispatchers.IO) всі операції з базою даних виконуються у фоновому потоці, що запобігає блокуванню основного інтерфейсу. Протягом тестування додаток не викликав помилок типу ANR (Application Not Responding) та не завершувався аварійно.

Також було перевірено поведінку додатку при:

- переміщенні нульової або надмірної кількості;
- спробі додати порожні значення;
- повторному запуску застосунку (дані зберігаються коректно через Room).

Результати тестування підтвердили, що мобільний додаток стабільно виконує свої функції, не містить критичних помилок і готовий до використання в демонстраційних цілях. Архітектура додатку дозволяє у майбутньому провести модульне тестування або інтеграційне тестування за потреби.

У третьому розділі дипломної роботи реалізовано повноцінне програмне рішення у вигляді мобільного додатку, що відповідає функціональним, архітектурним та технологічним вимогам, визначеним на етапі проектування. Розробка здійснювалась у середовищі IntelliJ IDEA мовою Kotlin із використанням бібліотек Android Jetpack, що забезпечило стабільну основу для реалізації архітектури MVVM та взаємодії з базою даних Room.

На основі клієнт-серверної моделі було реалізовано ключові компоненти додатку: основні екрани взаємодії (додавання препарату, партії, переміщення), базу даних із підтримкою обліку серій і складів, інтерфейс користувача з підтримкою Material Design, а також журнал переміщень. Окрему увагу приділено забезпеченню масштабованості та зручності розширення: структура проєкту організована логічно, із чітким поділом відповідальностей між класами, а використання шаблонів проектування (MVVM, Singleton, Adapter) підвищило якість і підтримуваність коду.

Проведено тестування додатку на етапі розробки, результати якого підтвердили стабільність роботи функціоналу, правильність виконання основних операцій, а також відповідність очікуваному результату. Використання корутин забезпечило асинхронність та уникнення помилок блокування інтерфейсу.

Таким чином, реалізований мобільний додаток повністю відповідає поставленим задачам автоматизації процесів обліку, контролю партій і логістики.

					БР.КІ-23.00.00.000 ПЗ	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дат		

ВИСНОВКИ

У результаті проведеного в розділі 1 аналізу виконано всі завдання, визначені на початковому етапі дослідження. По-перше, розглянуто специфіку бізнес-процесів фармацевтичної компанії, зокрема постачання, виробництво, контроль якості, облік, логістику, продажі та нормативне регулювання. Це дозволило виявити критичні аспекти, які необхідно враховувати при розробці інформаційних систем.

По-друге, здійснено огляд сучасних програмних рішень, включаючи мобільні додатки, ERP/CRM-системи та хмарні сервіси. Встановлено, що більшість з них не адаптовані до потреб українських фармкомпаній, що зумовлює доцільність створення власного мобільного застосунку.

По-третє, обґрунтовано вибір інструментів і технологій для реалізації програмного продукту. Основною мовою програмування обрано Java, як надійне та поширене рішення для Android-розробки. У якості середовища розробки використано IntelliJ IDEA з відповідним набором бібліотек і компонентів. Таким чином, розділ 1 сформував теоретичну й технологічну основу для подальшого проектування та реалізації мобільного додатку для фармацевтичної компанії.

У другому розділі реалізовано етап проектування мобільного застосунку, що безпосередньо відповідає завданню проектування архітектури та моделі бази даних, визначеному у вступі дослідження.

По-перше, на основі проаналізованих бізнес-потреб фармацевтичної компанії було сформульовано мету мобільного застосунку, визначено основні функціональні та нефункціональні вимоги, а також уточнено роль мобільного інтерфейсу в загальній інформаційній системі підприємства.

По-друге, розроблено клієнт-серверну архітектуру системи з урахуванням специфіки фармацевтичного обліку, де мобільний додаток виконує роль інтерактивного клієнта, інтегрованого з ERP через REST API. Побудовано

					БР.КІ-23.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		68

схему взаємодії компонентів системи та UML-діаграму, яка відображає логіку обміну даними та функціональний поділ відповідальностей.

По-третє, спроектовано логічну та фізичну структуру бази даних. Визначено ключові сутності (препарати, серії, склади, рухи, користувачі тощо), атрибути, типи зв'язків і обмеження. Створено ER-діаграму та DBML-опис, що відповідають вимогам до нормалізації, масштабованості та інтеграційної сумісності з ERP/WMS-модулями.

По-четверте, описано механізми захисту даних, забезпечення цілісності, журналювання подій (audit trail) та резервного копіювання. Окрему увагу приділено підтримці офлайн-режиму на мобільному клієнті, що реалізується через локальну БД SQLite з подальшою синхронізацією з центральним сервером на PostgreSQL.

Загалом, у межах розділу 2 виконано повноцінне системне проектування мобільного застосунку для фармацевтичної компанії, що забезпечує технологічну готовність до реалізації наступного етапу – програмної розробки та впровадження основного функціоналу засобами Java.

В розділі 3 реалізовано мобільний додаток відповідно до затвердженої архітектури. Створено інтерфейси для додавання препаратів, управління партіями, перегляду залишків і виконання переміщень між складами. Застосовано архітектурний шаблон MVVM, шаблони Singleton та Adapter, Room як ORM-рішення для SQLite, а також засоби асинхронної обробки даних на базі Kotlin Coroutines. Проведено тестування додатку із застосуванням генерації тестових даних, перевірено всі ключові функції та підтверджено стабільність роботи програмного продукту.

Результати дипломної роботи демонструють можливість ефективного застосування сучасних мобільних технологій для автоматизації процесів у фармацевтичному бізнесі. Створений додаток може слугувати основою для подальшого розширення функціональності, інтеграції з зовнішніми ERP-системами та впровадження в реальні бізнес-процеси.

					БР.КІ-23.00.00.000 ПЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дат		

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Як мобільні додатки трансформують галузь охорони здоров'я URL: <https://stfalcon.com/uk/blog/> (дата звернення: 05.04.2025).
2. Методологія наукових досліджень URL: <https://www.wunu.edu.ua/opp/> (дата звернення: 05.04.2025).
3. Методологія наукових досліджень URL: <https://www.wunu.edu.ua/opp/> дата звернення: 05.04.2025).
4. Наукове дослідження URL: <https://uk.wikipedia.org/wiki> (дата звернення: 05.04.2025).
5. Всесвітня організація охорони здоров'я. Essential medicines: Fact sheet URL: <https://www.who.int/news-room/fact-sheets/detail/essential-medicines> (дата звернення: 05.04.2025).
6. Структура фармацевтичного ринку України URL: <https://blog.youcontrol.market/struktura-farmatsievtichnogo-rinku-ukrayini/> (дата звернення: 05.04.2025).
7. Державна служба України з лікарських засобів та контролю за наркотиками: офіц. вебсайт URL: <https://www.dls.gov.ua/en/state-service-of-ukraine-on-medicines-and-drugs-control/> (дата звернення: 05.04.2025).
8. Закон України «Про лікарські засоби» від 04.04.1996 р. № 123/96-ВР URL: <https://zakon.rada.gov.ua/laws/show/123/96-вр> (дата звернення: 06.04.2025).
9. Ministry of Health of Ukraine. Good Manufacturing Practice Guidelines URL: <https://moz.gov.ua/gmp-guidelines> (дата звернення: 06.04.2025).
10. Commission of the European Communities. EudraLex – Volume 4 Good Manufacturing Practice (GMP) Guidelines URL: https://health.ec.europa.eu/eudralex/eudralex-volume-4_en (дата звернення: 06.04.2025).
11. Law of Ukraine «On State Control of Compliance with Legislation on Medicinal Products» від 03.10.2019 р. № 85-IX URL: <https://zakon.rada.gov.ua/laws/show/85-19> (дата звернення: 06.04.2025).

					БР.КІ-23.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		70

12. Закон України «Про захист персональних даних» від 01.06.2010 р. № 2297-VI URL: <https://zakon.rada.gov.ua/laws/show/2297-17> (дата звернення: 06.04.2025).

13. FIFO, FEFO, LIFO: What is the meaning? URL: <https://www.gmp-compliance.org/gmp-news/fifo-fefo-lifo-what-is-the-meaning> (дата звернення: 06.04.2025).

14. TrackTraceRx. Cloud-based Pharmaceutical Track & Trace URL: <https://www.tracktracerx.com> (дата звернення: 07.04.2025).

15. Veeva Systems. Cloud Software for the Global Life Sciences Industry [Сайт] URL: <https://www.veeva.com> (дата звернення: 07.04.2025).

16. TraceLink. Digital Supply Chain for the Life Sciences URL: <https://www.tracelink.com> (дата звернення: 07.04.2025).

17. Tabletki.ua – пошук ліків онлайн URL: <https://tabletki.ua/> (дата звернення: 07.04.2025).

18. Apteki.ua – замовлення ліків в аптеках України URL: <https://apteki.ua/uk> (дата звернення: 07.04.2025).

19. Ліки Контроль – перевірка інструкцій та аналогів URL: <https://likicontrol.com.ua/> (дата звернення: 07.04.2025).

20. Як працюють мобільні додатки аптек, АЗС, ДРЦ і кінотеатрів. URL: <https://rau.ua/novyni/mobilni-dodatki-aptek-azs-drc/> (дата звернення: 07.04.2025).

21. Загальні технічні вимоги до медичних та фармацевтичних (аптечних) інформаційних систем для підключення до інформаційної системи електронної охорони здоров'я (ЕОЗ) URL: https://ehealth.gov.ua/wp-content/uploads/2022/07/Zagalni_tehnichni_vymogy_do_Beezpeky_MIS.pdf

22. Pharmaceutical Quality System Q10: ICH guideline URL: <https://database.ich.org/sites/default/files/Q10%20Guideline.pdf> (дата звернення: 21.04.2025).

23. Guidance on Data Integrity / PIC/S URL: <https://picscheme.org/en/publications?tri=16> (дата звернення: 21.04.2025).

					БР.КІ-23.00.00.000 ПЗ	Арк.
						71
Змн.	Арк.	№ докум.	Підпис	Дат		

24. Data Integrity and Compliance with CGMP: Guidance for Industry / U.S. Food and Drug Administration URL: <https://www.fda.gov/media/119267/download> (дата звернення: 21.04.2025).

25. Good Distribution Practice (GDP) : Questions & Answers / European Medicines Agency URL: <https://www.ema.europa.eu/en/human-regulatory/research-development/compliance/good-distribution-practice> (дата звернення: 21.04.2025).

26. Android Developers. Room Persistence Library URL: <https://developer.android.com/jetpack/androidx/releases/room> (дата звернення: 21.04.2025).

27. Android Developers. Lifecycle: ViewModel та LiveData URL: <https://developer.android.com/jetpack/androidx/releases/lifecycle> (дата звернення: 21.04.2025).

28. Android Developers. LiveData overview URL: <https://developer.android.com/topic/libraries/architecture/livedata> (дата звернення: 21.04.2025).

29. Gradle Build System URL: <https://docs.gradle.org/current/userguide/userguide.html> (дата звернення: 21.04.2025).

					БР.КІ-23.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		72