

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «РОЗРОБКА МОБІЛЬНОГО ЗАСТОСУНКУ  
ДЛЯ ВИВЧЕННЯ ШКІЛЬНОГО КУРСУ ХІМІЇ»

Виконав: студент \_\_\_\_\_ 5 \_\_\_\_\_ курсу, групи \_\_\_\_\_ 6.1229-з  
спеціальності \_\_\_\_\_ 122 Комп'ютерні науки \_\_\_\_\_  
(шифр і назва спеціальності)

освітньої програми \_\_\_\_\_ Комп'ютерні науки \_\_\_\_\_  
(назва освітньої програми)

В.О. Махмутов

(ініціали та прізвище)

Керівник \_\_\_\_\_ старший викладач кафедри комп'ютерних наук,  
Циммерман Г.А. \_\_\_\_\_

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент \_\_\_\_\_ професор кафедри програмної інженерії,  
к.ф.-м.н., доцент Кудін О.В. \_\_\_\_\_

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Запоріжжя 2024

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

Факультет математичний

Кафедра комп'ютерних наук

Рівень вищої освіти бакалавр

Спеціальність 122 Комп'ютерні науки

(шифр і назва)

Освітня програма Комп'ютерні науки

**ЗАТВЕРДЖУЮ**

Завідувач кафедри комп'ютерних наук,  
Д.Т.Н., доцент

Шило Г. М.

(підпис)

“ 22 ” грудня 2023 р.

## ЗАВДАННЯ

### НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Махмутову Владиславу Олександровичу

(прізвище, ім'я та по-батькові)

1. Тема роботи Розробка мобільного застосунку для вивчення шкільного курсу хімії  
керівник роботи Циммерман Геннадій Анатолійович, старший викладач

(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

кафедри комп'ютерних наук

затверджені наказом ЗНУ від « 21 » грудня 2023 року № 2181-с

2. Строк подання студентом роботи 15.05.2024

3. Вихідні дані до роботи 1. Постановка задачі.

2. Перелік літератури.

3. Додаток.

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

1. Постановка задачі. Розробити мобільний застосунок для вивчення шкільного курсу хімії.

2. Основні теоретичні відомості. Проектування та використання мобільних застосунків навчального призначення

3. Назвати розділи у відповідності з темою

Основи розробки мобільного застосунку з хімії на платформі android, Створення інтерактивного мобільного додатку для вивчення хімії, Тестування та впровадження мобільного застосунку з хімії

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

презентація

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та по-сада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 22.12.2023**КАЛЕНДАРНИЙ ПЛАН**

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	28.02.2024	
2.	Збір вихідних даних.	05.03.2024	
3.	Обробка методичних та теоретичних джерел.	11.04.2024	
4.	Розробка першого та другого розділу.	25.04.2024	
5.	Розробка та тестування моб застосунку	02.05.2024	
6.	Розробка третього розділу.	17.05.2024	
7.	Оформлення та нормоконтроль кваліфікаційної роботи бакалавра.	26.05.2024	
8.	Захист кваліфікаційної роботи.	30.05.2024	

Студент \_\_\_\_\_  
(підпис)**В.О. Махмутов** \_\_\_\_\_  
(ініціали та прізвище)Керівник роботи \_\_\_\_\_  
(підпис)**Г.А. Циммерман** \_\_\_\_\_  
(ініціали та прізвище)**Нормоконтроль пройдено**Нормоконтролер \_\_\_\_\_  
(підпис)**О.Г. Спиця** \_\_\_\_\_  
(ініціали та прізвище)

## РЕФЕРАТ

Кваліфікаційна робота бакалавра «Розробка мобільного застосунку для вивчення шкільного курсу хімії»: 46 с., 12 рис., 5 табл., 17 джерел, 1 додаток.

ІНТЕРАКТИВНЕ НАВЧАННЯ, МОБІЛЬНИЙ ЗАСТОСУНОК, ОСВІТНІ ТЕХНОЛОГІЇ, РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, ТЕСТУВАННЯ ЗАСТОСУНКІВ, ХІМІЯ, ANDROID STUDIO, JETPACK COMPOSE, KOTLIN.

Об'єкт роботи – проектування мобільних застосунків навчального призначення.

Мета роботи: створення мобільного застосунку для вивчення хімії на платформі Android, з використанням сучасних технологій програмування, як Kotlin та Jetpack Compose, та принципів Material Design. Розробка такого застосунку була спрямована на підвищення інтерактивності та ефективності навчального процесу за допомогою використання інтерактивних елементів.

Методи використані при виконанні роботи – аналіз проблеми створення та використання моб застосунків, розробка архітектурних рішень з використанням патернів проектування програмного забезпечення, синтезування інтерфейсу та процедур обробки даних засобами програмування на Kotlin.

Застосунок був порівняний з іншими існуючими рішеннями, що дозволило виявити його особливості в плані користувацького досвіду та навчальної ефективності.

Було створено повнофункціональний застосунок, який інтегрує інтерактивність в навчальний процес.

## SUMMARY

Bachelor's Qualifying Theses « Development of the mobile application for studying a school chemistry course»: 46 pages, 12 figures, 5 tables, 17 references, 1 supplements.

INTERACTIVE LEARNING, MOBILE APP, EDUCATIONAL TECHNOLOGIES, SOFTWARE DEVELOPMENT, APPLICATION TESTING, CHEMISTRY, ANDROID STUDIO, JETPACK COMPOSE, KOTLIN.

The object of the work – Designing mobile applications for educational purposes

Aim of the study: create a mobile application for studying chemistry on the Android platform, using modern programming technologies such as Kotlin and Jetpack Compose, and the principles of Material Design. The development of this application was aimed at enhancing the interactivity and efficiency of the learning process through the use of interactive elements.

Methods used in the work – Analysis of the problem of creating and using mobile applications, development of architectural solutions using software design patterns, synthesis of the interface and data processing procedures using Kotlin programming tools.

The application was compared with other existing solutions, which allowed the identification of its advantages in terms of user experience and educational efficiency.

A fully functional application was created that integrates interactivity into the educational process.

## ЗМІСТ

Завдання на кваліфікаційну роботу.....	2
Реферат .....	4
Summary .....	5
Зміст.....	6
Вступ.....	7
1 Основи розробки мобільного застосунку з хімії на платформі android ....	9
1.1 Огляд мови програмування Kotlin і середовища Android Studio .....	9
1.2 Архітектурні принципи розробки мобільних застосунків з використанням Jetpack.....	12
1.3 Проектування інтерфейсу користувача за допомогою Jetpack Compose.....	15
1.4 Огляд існуючих застосунків для вивчення хімії .....	18
2 Створення інтерактивного мобільного додатку для вивчення хімії.....	23
2.1 Вибір інтерактивних елементів для застосунку .....	23
2.2 Технологічний процес створення застосунку .....	28
3 Стратегії тестування та методи впровадження мобільного застосунку з хімії.....	31
3.1 Стратегії та методи тестування мобільних застосунків.....	31
3.2 Розгортання додатку та залучення користувачів.....	36
Висновки .....	39
Перелік посилань.....	41
Додаток А Код застосунку .....	43

## ВСТУП

У сучасному світі значення освітніх технологій стрімко зростає, відповідаючи на потреби суспільства у більш гнучких, доступних та індивідуалізованих формах навчання. Особливо це стосується точних наук, таких як хімія, де розуміння складних концепцій може бути значно поліпшене за допомогою візуалізації та інтерактивності. Аналіз сучасного стану освітніх програм показує, що існуючі мобільні застосунки для вивчення хімії часто обмежені статичним контентом та не використовують повний потенціал сучасних технологій для залучення студентів. Така ситуація підкреслює актуальність розробки нових рішень, які могли б зробити процес навчання більш ефективним та мотивуючим.

Мета цієї дипломної роботи полягає у розробці мобільного застосунку для вивчення хімії на платформі Android, який би використовував сучасні інструменти та методології програмування для створення інтерактивного та залучаючого навчального середовища. Задля досягнення цієї мети були поставлені наступні завдання: аналіз існуючих застосунків, розробка архітектури та інтерфейсу з використанням Kotlin, Android Studio, Jetpack Compose, і Material Design, а також впровадження ефективних стратегій тестування та залучення користувачів.

Важливість роботи підкріплюється широкими можливостями застосування розробленого застосунку. Новий застосунок може бути використаний у школах, коледжах та університетах для покращення процесу навчання, а також для самостійного вивчення хімії бажаними. Крім того, застосунок може бути інтегрований у більш широкі освітні платформи, що надають інтерактивні курси з різних предметів. Основні проектні рішення, прийняті під час розробки, базуються на глибокому аналізі потреб користувачів та найновіших трендах у розробці програмного забезпечення. Використання Kotlin та Jetpack Compose дозволяє забезпечити високу якість коду та гнучкість інтерфейсу, тоді як

Material Design вносить елементи сучасного дизайну, що робить застосунок не тільки функціональним, але й візуально привабливим.

Розробка мобільного застосунку для вивчення хімії є не тільки актуальною, але й важливою задачею, яка спрямована на підвищення ефективності освітнього процесу і може мати значний вплив на якість хімічної освіти. Результати дослідження мають теоретичне та практичне значення, оскільки вносять вклад у розвиток освітніх технологій і методів викладання хімії, а також забезпечують базу для подальших розробок в цій галузі.

Процедура створення мобільних застосунків для вивчення наукових дисциплін, зокрема хімії, вимагає не тільки технічної компетенції, але й глибокого розуміння освітніх потреб та навчальних цілей. Розробка таких застосунків зазвичай починається з детального аналізу існуючих освітніх методик і підходів до викладання предмету, щоб забезпечити їхнє ефективне втілення у цифровий формат. Важливим етапом є створення інтерактивних та залучаючих матеріалів, які можуть використовувати графічні та відеоелементи для кращого засвоєння складних концепцій. Після розробки архітектури та дизайну застосунку настає фаза тестування, де основна увага приділяється виявленню технічних помилок та забезпеченню зручності користувацького інтерфейсу. Фінальний етап включає збір відгуків від користувачів, що дозволяє подальше удосконалення застосунку.

Завдяки систематичному підходу до розробки, з орієнтацією на освітні цілі та користувацький досвід, можливо створити мобільний застосунок, який не тільки покращує розуміння хімії, але й стимулює інтерес та заохочує до глибшого вивчення предмету. Такий застосунок може слугувати зразком для майбутніх розробок у галузі освітніх технологій та відігравати ключову роль у підвищенні якості освіти.



# 1 ОСНОВИ РОЗРОБКИ МОБІЛЬНОГО ЗАСТОСУНКУ З ХІМІЇ НА ПЛАТФОРМІ ANDROID

## 1.1 Огляд мови програмування Kotlin і середовища Android Studio

Основою розробки мобільних додатків на платформі Android є вибір правильних інструментів і методів для ефективною реалізації всіх запланованих функцій. Одним з ключових рішень є вибір мови програмування Kotlin. Розроблена компанією JetBrains, мова програмування Kotlin – це статично типізована мова програмування, яка повністю сумісна з Java та оптимізована спеціально для платформи Android [1]. Вона пропонує чистіший і виразніший синтаксис, ніж Java, що дозволяє розробникам писати код, який легше читати і підтримувати, Kotlin також безпечніше обробляє нульові значення і має ряд інших функцій, які підвищують продуктивність і надійність додатків.

При розробці мобільних додатків для Android важливо вибрати ефективне середовище розробки, і Android Studio, офіційне середовище розробки додатків для Android від Google, є хорошим вибором. Заснована на IntelliJ IDEA від JetBrains, Android Studio інтегрується з Kotlin і пропонує широкий спектр можливостей. Середовище надає прості інструменти для проектування користувацького інтерфейсу, аналізу продуктивності додатків, управління залежностями та налагодження, що робить його незамінним інструментом для всіх розробників додатків для Android. Особливої уваги заслуговує вбудований емулятор, який дозволяє розробникам тестувати додатки в контрольованому середовищі без необхідності використання реального пристрою.

Окрім Kotlin та Android Studio, існує багато інших мов програмування та середовищ розробки, доступних для створення мобільних додатків для Android. Наприклад, Java залишається популярною мовою завдяки своїй стабільності та багатим API, а середовище Eclipse з плагіном ADT (Android Development Tools) широко використовувалося до того, як фокус змістився на

Android Studio. Існують також альтернативні платформи, такі як Xamarin від Microsoft, яка дозволяє розробникам використовувати C# для створення мобільних додатків, що працюють на різних платформах, включаючи Android та iOS. React Native, розроблений Facebook, дозволяє писати додатки на JavaScript і React для швидкої розробки додатків з широкою взаємодією з користувачем.

Однак є кілька переваг вибору Kotlin та Android Studio для розробки мобільних додатків, які роблять його особливо привабливим варіантом. Kotlin був чітко визначений Google як офіційна мова для розробки додатків для Android, що свідчить про те, що мова була оптимізована для потреб цієї платформи. Крім того, Kotlin має значні переваги над Java з точки зору безпеки та простоти коду, зокрема, завдяки більш жорсткому управлінню нулями та більш виразному синтаксису. Ці особливості зменшують ймовірність помилок у застосунках і спрощують обслуговування коду. Як офіційне інтегроване середовище розробки (IDE) для Android, Android Studio глибоко інтегрована з Kotlin та іншими компонентами екосистеми Android, включаючи не тільки редактор коду, але й інструменти для візуалізації користувацького інтерфейсу, управління залежностями та налагодження. Ці інструменти значно прискорюють процес розробки та тестування, забезпечуючи при цьому високу якість та стабільність нашого додатку.

При розробці мобільного додатку на платформі Android також важливо зосередитися на інтеграції сучасних технологій дизайну та користувацького інтерфейсу. Цей процес можна значно покращити за допомогою таких технологій, як Jetpack Compose – нової бібліотеки користувацького інтерфейсу, яка дозволяє розробникам створювати більш модульні та адаптивні інтерфейси, а також Material Design. Бібліотека використовує декларативний підхід до опису користувацького інтерфейсу, що робить його більш чутливим до змін даних та стану програми. Використання Jetpack Compose значно зменшує кількість коду, необхідного для реалізації складних інтерфейсів, і збільшує швидкість розробки додатків.

Jetpack Compose є ключовою бібліотекою в процесі розробки хімічних мобільних додатків і являє собою сучасний фреймворк для створення користувацьких інтерфейсів на платформі Android. Бібліотека використовує мову програмування Kotlin, що дозволяє розробникам виконувати більш чистий і виразний код. Відмінною рисою Jetpack Compose є декларативний підхід до дизайну інтерфейсу користувача, що означає, що розробники можуть описувати зовнішній вигляд інтерфейсу, а не керувати окремими змінами інтерфейсу користувача. Це забезпечує більшу гнучкість і швидкість реагування програми, оскільки інтерфейс користувача автоматично реагує на зміни в даних або стані програми. Використовуючи Android Studio, Jetpack Compose було інтегровано в наш додаток, що надало багато можливостей для ефективного використання бібліотеки. Android Studio підтримує Kotlin і Jetpack Compose, надаючи розробникам потужні інструменти для візуального редагування, попереднього перегляду користувацького інтерфейсу та налагодження. Це дозволяє легко розгортати, тестувати і налагоджувати інтерфейси, розроблені за допомогою Jetpack Compose, що скорочує час розробки і підвищує продуктивність розробника.

Jetpack Compose також дозволяє використовувати інноваційні методи дизайну, такі як Material Design, щоб зробити додатки не тільки потужними, але й візуально привабливими. Модульна природа Jetpack Compose дозволяє створювати динамічні, адаптивні інтерфейси, оптимізовані для широкого діапазону розмірів і орієнтації екранів. Така гнучкість є критично важливою для мобільних додатків, які працюють на широкому спектрі пристроїв.

Одним з ключових аспектів при розробці мобільних додатків для навчання є оптимізація взаємодії користувача з додатком. Щоб процес навчання був зручним та ефективним, важливо продумати управління станами та обробку подій у додатку. Такі функції, як збереження прогресу користувача, адаптація навчального матеріалу до індивідуальних потреб і здібностей учня та інтеграція інтерактивних елементів для залучення учнів до активного навчання, мають вирішальне значення для успіху додатку. Інтеграція бази даних

для зберігання даних користувачів, включаючи їхні відповіді, оцінки та прогрес у навчанні, вимагає використання ефективного інструменту управління даними. Room, бібліотека персистентності в складі компонентів архітектури Android, є хорошим рішенням для цього завдання. Вона дозволяє створити локальну базу даних, яка використовує об'єктно-реляційне відображення для спрощення доступу до даних та управління ними, забезпечуючи високу продуктивність та узгодженість даних [2]. Гейміфікація також може бути інтегрована для підвищення залученості користувачів, використовуючи такі елементи, як досягнення, медалі та таблиці лідерів, щоб додати елемент змагання та мотивації до процесу навчання. Ці елементи можна ефективно реалізувати за допомогою Firebase Analytics і Google Play Services, які дозволяють збирати дані про те, як користувачі взаємодіють з додатком, і адаптувати навчальний контент відповідно до їхніх потреб і успіхів.

Розробка мобільних додатків для організації навчання вимагає комплексного підходу, який включає не лише технічну реалізацію на основі потужних інструментів і мов програмування, а й глибоке розуміння педагогічних потреб і психології навчання. Інтеграція використання інтерактивності, адаптивності для користувача та гейміфікації може значно підвищити ефективність додатку та його привабливість для учнів.

## **1.2 Архітектурні принципи розробки мобільних застосунків з використанням Jetpack**

Розробка мобільного застосунку для навчання на платформі Android вимагає від розробників не тільки глибоких знань у програмуванні, але й розуміння архітектурних принципів, які забезпечують стабільність, масштабованість та легкість управління кодом. Однією з найновіших та найбільш ефективних бібліотек для досягнення цих цілей є Android Jetpack, зокрема його компоненти, що підтримують архітектурні принципи розробки.

Jetpack є набором бібліотек та інструментів, який надає консистентну та чітко визначену архітектуру для розробки Android-додатків. Використання архітектурних компонентів Jetpack дозволяє розробникам зосередитися на створенні якісного користувацького інтерфейсу та логіки застосунку, замість турботи про базову інфраструктуру програми. Серед ключових компонентів можна виділити LiveData, ViewModel, Room, та Data Binding. LiveData використовується для спостереження за даними, які можуть змінюватися. Вона дозволяє компонентам інтерфейсу реагувати на зміни даних без додаткового коду для керування цими змінами, що значно спрощує роботу з асинхронними даними та покращує відповідність користувацького інтерфейсу актуальному стану даних.

ViewModel відповідає за керування даними для користувацького інтерфейсу, ізолюючи управління даними від користувацького інтерфейсу, що допомагає вирішити проблеми з управлінням станом UI і пов'язаних з ним життєвим циклом додатку. ViewModel також забезпечує дані для інтерфейсу, що потребують складних обчислень або асинхронної обробки, зменшуючи ризик виникнення помилок і збільшуючи продуктивність додатку.

Room є бібліотекою персистентності, яка надає абстракцію над SQLite для більш легкої роботи з базою даних [3]. Це забезпечує безпечний і простий у використанні інтерфейс для зберігання довгострокових або складних даних, не жертвуючи продуктивністю. Data Binding, зі свого боку, дозволяє зв'язувати користувацькі інтерфейси у макеті XML безпосередньо з даними у додатку, що зменшує необхідність написання "клеючого" коду, який вручну з'єднує ці два шари.

Для розробки мобільного додатку для навчання хімії у школі на платформі Android важливо правильно обрати архітектурний патерн, який визначатиме структуру та спосіб взаємодії між компонентами додатку. Одним з ефективних патернів для створення додатків, які масштабуються і прості в обслуговуванні, зарекомендував себе патерн Model-View-ViewModel (MVVM).

Цей патерн допомагає відокремити бізнес-логіку від користувацького інтерфейсу, що забезпечує більшу гнучкість і тестування компонентів.

В середовищі Android MVVM використовує клас ViewModel для управління та зберігання даних користувацького інтерфейсу, забезпечуючи відокремлення від представлення даних. ViewModel взаємодіє з моделлю, яка обробляє бізнес-логіку та доступ до даних, зокрема через базу даних Room Database, що дозволяє виконувати операції з базою даних в окремому потоці, уникає перевантажень інтерфейсу та пропонує кращий відгук програми. Ці аспекти є критично важливими для додатків, які мають справу з великими обсягами даних або складною логікою обробки даних, як-от освітні програми з хімії. Бібліотеки зв'язування даних використовуються для зв'язування даних між компонентами інтерфейсу користувача і даними, які вони відображають. Це дозволяє автоматично оновлювати елементи інтерфейсу при зміні даних у ViewModel без додаткового втручання розробника, що призводить до більш чистого та зручного для читання коду. Прив'язка даних також знижує ймовірність помилок при ручному управлінні інтерфейсом і підвищує продуктивність, усуваючи необхідність постійного пошуку представлень компонентів у коді.

При розробці мобільного додатку для вивчення хімії необхідно подбати про оптимізацію користувацького досвіду та забезпечення високої продуктивності програми на широкому спектрі пристроїв. Одним із кроків для досягнення цього є правильне використання асинхронного програмування для обробки даних, що особливо важливо, коли додаток вимагає значних обчислювальних ресурсів або доступу до мережі.

Асинхронні операції в Jetpack виконуються за допомогою бібліотеки Coroutines на мові Kotlin. За допомогою Coroutines можна писати код, який легко читати і підтримувати так само, як і синхронний код, навіть якщо він насправді виконує асинхронні операції. Це важливо для додатків, які обробляють наукові дані, використовуючи інтенсивні мережеві запити або складні обчислювальні завдання [4]. Використання процедур забезпечує більш плавну

роботу користувача, оскільки основний потік програми не блокується під час виконання важких завдань.

Інтеграція сучасних методів зберігання даних також важлива. За допомогою DataStore від Jetpack ми можемо замінити SharedPreferences для більш надійного та гнучкого зберігання користувацьких налаштувань. DataStore використовує необроблені буфери або налаштування для зберігання даних і підтримує транзакційні та узгоджені дані, що є критично важливим для додатків, які повинні зберігати важливі навчальні дані та користувацькі налаштування. Окрім технічних аспектів, необхідно звернути увагу на реалізацію функцій доступності, що є ключовим для створення інклюзивного навчального середовища. Android надає низку утиліт для забезпечення доступності, таких як TalkBack, які допомагають людям з інвалідністю користуватися додатками. Впровадження цих функцій не лише розширює аудиторію користувачів, але й підкреслює соціальну відповідальність розробників.

### **1.3 Проектування інтерфейсу користувача за допомогою Jetpack Compose**

Розробка користувацького інтерфейсу мобільного додатку призначеного для навчання на платформі Android з використанням Jetpack Compose є критично важливим аспектом забезпечення не тільки естетики, але й функціональності та інтуїтивно зрозумілої взаємодії. Jetpack Compose – це сучасна бібліотека для створення декларативних користувацьких інтерфейсів в Android-додатках, яка використовує мову програмування Kotlin, що значно спрощує процес розробки інтерфейсів [5]. Вона використовує Kotlin як мову програмування, що значно спрощує процес розробки інтерфейсу.

Однією з головних переваг Jetpack Compose є те, що він дозволяє розробникам описувати інтерфейс як функцію, яка повертає макет користувацького інтерфейсу, замість традиційного XML-підходу. Цей декларативний підхід

дозволяє розробникам більш чітко визначити, як користувацький інтерфейс повинен реагувати на різні стани та події, таким чином спрощуючи код і забезпечуючи кращу підтримку. Замість того, щоб вручну керувати складним життєвим циклом об'єктів і компонентів інтерфейсу користувача, Compose автоматично оновлює інтерфейс користувача, коли змінюються підписані дані.

Jetpack Compose для ефективної реалізації різноманітних навчальних матеріалів, таких як інтерактивні елементи, анімації реакцій або молекулярні моделі, які потребують динамічної взаємодії. Наприклад, за допомогою Compose можна легко створювати картки з реагентами і продуктами реакції, які візуально відповідають змінам в реакції в режимі реального часу, що дає учням глибше розуміння процесу реакції. Jetpack Compose інтегрується з іншими компонентами Jetpack, такими як навігація, життєвий цикл і моделі представлення, щоб створити більш структуровану і керовану архітектуру додатків [6]. Наприклад, за допомогою навігації Compose ми можемо легко реалізувати складні навігаційні сценарії у додатку, такі як переходи між різними темами в курсі хімії або докладні описи конкретних хімічних елементів, забезпечуючи плавний і інтуїтивно зрозумілий користувацький досвід.

Таблиця 1.1 не лише відображає технічні можливості Jetpack Compose, але й підкреслює її важливість у створенні інтуїтивно зрозумілого та інтерактивного навчального середовища. Використання цієї технології в навчальних програмах з хімії може підвищити рівень залученості та задоволеності користувачів, ефективно впливаючи на їхню зацікавленість та мотивацію до навчання.

Використання Jetpack Compose при розробці мобільного додатку для вивчення хімії для Android – це інвестиція в майбутнє додатку, що забезпечує адаптивність, швидкість розробки та простоту обслуговування, які є критично важливими для успішної освітньої технології.



Таблиця 1.1 – Використання Jetpack Compose у дизайні інтерфейсу користувача

Аспект	Властивості
<b>Декларативний UI</b>	Jetpack Compose дозволяє описувати UI через функції Kotlin, що забезпечує простий та чистий код, де логіка візуалізації тісно пов'язана з даними.
<b>Інтеграція з Jetpack</b>	Compose легко інтегрується з іншими компонентами Jetpack, такими як Navigation, Lifecycles та ViewModels, що забезпечує стабільну архітектуру та високу продуктивність додатку.
<b>Інтерактивні елементи</b>	Підтримка динамічних компонентів, таких як анімації та інтерактивні навчальні матеріали, робить навчання більш захоплюючим та ефективним.
<b>Доступність</b>	Jetpack Compose сприяє легкій імплементації функцій доступності, таких як TalkBack, що робить застосунок доступнішим для користувачів з обмеженими можливостями.

Розробка користувацьких інтерфейсів у мобільних додатках для Android з використанням Jetpack Compose є стратегічно важливою для досягнення головної мети – створення інтуїтивно зрозумілого та ефективного навчального середовища. Враховуючи, що Jetpack Compose є основним інструментом для реалізації користувацьких інтерфейсів, ми зосередилися на використанні його декларативних можливостей для створення динамічних і чуйних інтерфейсів, які відображають складні наукові дані в легкій для розуміння формі.

Важливою частиною процесу проектування є забезпечення доступності та врахування потреб усіх користувачів, в тому числі з обмеженими можливо-

стями, і Jetpack Compose дозволяє легко реалізувати різноманітні функції доступності, такі як підтримка зчитувачів з екрану та форматування великого тексту, що є критично важливими для освітніх програм.

#### **1.4 Огляд існуючих застосунків для вивчення хімії**

Аналіз існуючих мобільних додатків для вивчення хімії є важливим кроком у процесі розробки нових додатків, оскільки це дозволяє виявити сильні та слабкі сторони існуючих рішень і знайти шляхи для їх інновацій та вдосконалення [7]. Різні навчальні програми з хімії пропонують широкий спектр функцій, від простих довідників і періодичних таблиць до більш складних інтерактивних симуляторів, які дозволяють користувачам проводити віртуальні хімічні експерименти.

Більшість існуючих програм для вивчення хімії використовують традиційний підхід, надаючи користувачам текстовий контент, зображення та пояснення ключових хімічних понять і процесів. Деякі додатки включають інтерактивні елементи, такі як 3D-моделі молекул або інтерактивні картки, що допомагають користувачам запам'ятовувати формули та хімічні властивості речовин. Ці функції можуть покращити навчання завдяки візуалізації та активній участі учнів у навчальному процесі.

Однак однією з головних проблем багатьох додатків є брак персоналізації. Хоча деякі додатки можна налаштувати відповідно до рівня знань користувача, більшість з них не пристосовані до індивідуальних навчальних потреб чи конкретних стилів навчання. Це створює простір для розробки більш адаптивних і гнучких рішень, які можуть використовувати штучний інтелект або машинне навчання для налаштування контенту на основі прогресу і здібностей користувача. Багато додатків не враховують соціальний аспект навчання. Впровадження таких функцій, як спільне навчання або міжкористувацькі змагання, може значно підвищити мотивацію та залученість

учнів. Інтеграція цих функцій може перетворити вивчення хімії з ізольованої діяльності на цікавий спільний досвід.

Дослідження ринку мобільних додатків для навчання виявило деякі цікаві тенденції та потреби, які ще не повністю задовольняються існуючими рішеннями [8]. Більшість додатків зосереджені на базовому вивченні хімічних елементів, реакцій і періодичної таблиці, але є ще багато можливостей для інновацій, особливо у використанні доповненої реальності (AR) і віртуальної реальності (VR) для створення більш інтерактивного і захоплюючого навчального процесу. Доповнена реальність дозволяє студентам візуалізувати хімічні структури в трьох вимірах безпосередньо через камеру мобільного пристрою. Це може значно покращити розуміння складних молекулярних зв'язків і стереохімії, які часто важко візуалізувати за допомогою 2D-зображень. Такі додатки можуть включати функції молекулярних маніпуляцій, які дозволяють користувачам змінювати орієнтацію і спостерігати за реакціями в реальному часі.

З іншого боку, використання віртуальної реальності дозволяє учням зануритися у віртуальну лабораторію, безпечно проводити експерименти і досліджувати хімічні процеси. Такий підхід не лише робить вивчення хімії більш цікавим, але й надає досвід, який багато шкіл не можуть запропонувати через обмеження безпеки або відсутність необхідного обладнання. Існуючі додатки часто не забезпечують належної підтримки для практичних завдань з хімії, таких як балансування рівнянь або обчислення молярних співвідношень [9]. Розробка додатків, які включають більш просунуті інструменти, що допомагають виконувати ці завдання, значно сприятиме підвищенню рівня знань учнів.

Важливість аналізу існуючих мобільних додатків полягає не лише у визначенні їхніх сильних і слабких сторін, а й у виявленні можливостей для застосування передових технологій і методик, які можуть перетворити традиційне навчання хімії на більш інтерактивний і захоплюючий процес. Основним недоліком багатьох існуючих додатків є те, що вони зосереджені на статичних візуальних презентаціях і не використовують всі переваги сучасних

функцій для персоналізації користувацького досвіду та залучення учнів. Якщо існуючі рішення надають базові інструменти, такі як періодична таблиця або текстовий контент, що пояснює хімічні процеси, можливо значно підвищити цінність навчальних додатків, впровадивши технології доповненої і віртуальної реальності. Ці технології дозволяють учням не лише бачити зображення, але й взаємодіяти з хімічними структурами та процесами. Наприклад, доповнена реальність може перетворити звичайний клас на захопливу віртуальну лабораторію, яка дозволяє учням досліджувати молекулярні зв'язки та реакції в режимі реального часу, роблячи навчання більш інтуїтивним та практичним.

Незважаючи на велику кількість пропозицій на ринку мобільних додатків, все ще існує значний потенціал для інноваційних рішень, які використовують новітні технології для покращення результатів навчання та роблять процес навчання більш цікавим і доступним.

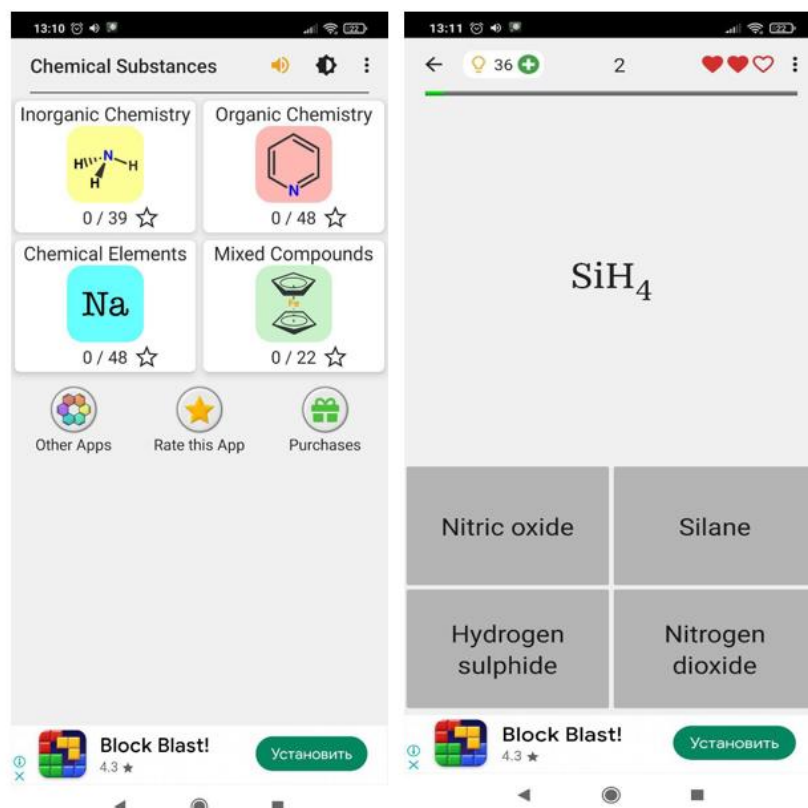


Рисунок 1.1 – «Chemical substances»

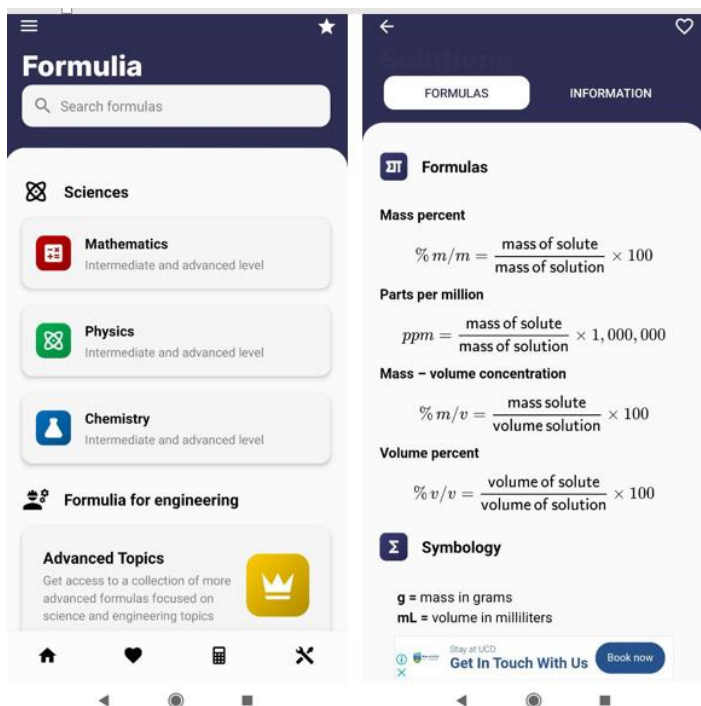


Рисунок 1.2 – «Formulia»



Рисунок 1.3 – «Chemistry Lab»

У розділі 1.4 було проаналізовано багато існуючих мобільних додатків для вивчення хімії. Як приклади, деякі з них представлені на рис. 1.1, 1.2, 1.3. Всі вони відрізняються своїм наповненням та рівнем складності. Але у всіх

застосунках, які були розглянуті, був суттєвий недолік: вони не були українською. Це ускладнює подальше вивчення хімії та здачу ЗНО як основного тесту для перевірки знань через різницю в номенклатурах та термінах.

## 2 СТВОРЕННЯ ІНТЕРАКТИВНОГО МОБІЛЬНОГО ДОДАТКУ ДЛЯ ВИВЧЕННЯ ХІМІЇ

### 2.1 Вибір інтерактивних елементів для застосування

Інтеграція інтерактивних елементів та мультимедійних ресурсів у мобільні додатки є важливим компонентом, який може значно підвищити залученість та мотивацію учнів. Створення навчального контенту з використанням інтерактивних візуалізацій, анімації, відео та інших мультимедійних засобів може допомогти учням краще зрозуміти складні хімічні процеси і концепції, які може бути важко висвітлити за допомогою традиційних письмових матеріалів.

Інтерактивність мобільних додатків може бути досягнута завдяки таким функціям, як віртуальні лабораторії, де користувачі можуть проводити експерименти, змішуючи хімічні речовини і спостерігаючи за реакціями в безпечному і контрольованому цифровому середовищі. Це допомагає учням відчувати автентичність наукових досліджень без ризиків і обмежень реальної лабораторії.

Інтеграція мультимедійних відео, що демонструють хімічні процеси або наукові лекції експертів, може збагатити навчальний процес, поглиблюючи розуміння матеріалу. Відео особливо корисні для візуального зображення таких процесів, як перенесення електронів в окисно-відновних реакціях, що може допомогти учням краще зрозуміти, як ці реакції відбуваються на молекулярному рівні [10]. Іншим важливим аспектом інтеграції мультимедійних ресурсів є використання аудіо-візуальних підказок для пояснення складних понять. Наприклад, звукові ефекти можуть бути використані, щоб підкреслити моменти реакції, або анімація може бути використана для імітації змін на атомному і молекулярному рівнях, а інтерактивні тести і вікторини можуть допомогти перевірити розуміння і закріпити знання.

Ключем до підвищення ефективності навчального процесу в мобільних додатках для вивчення хімії є включення інтерактивних інструментів, які сприяють глибшому зануренню в матеріал і заохочують активну участь користувача. Використання інтерактивних схем, моделювання хімічних процесів та графічних симуляцій може значно покращити розуміння складних хімічних явищ. Наприклад, інтеграція модулів, які дозволяють користувачам самостійно моделювати різні типи хімічних реакцій, включаючи екзотермічні та адсорбційні реакції, допомагає візуалізувати енергетичні зміни, які відбуваються під час реакції. Особливо корисними є інтерактивні інструменти, такі як віртуальні лабораторії, де користувачі можуть проводити віртуальні експерименти з різними хімічними речовинами і спостерігати за результатами без реального ризику. Ці інструменти не лише забезпечують безпечне навчальне середовище, але й дозволяють користувачам експериментувати з умовами реакції, такими як температура і концентрація, що дає їм змогу краще зрозуміти, як ці фактори впливають на процес реакції.

Ще одним важливим фактором є використання анімації для демонстрації молекулярних структур і механізмів реакцій. Анімація може показати, як атоми і молекули взаємодіють у реальному часі, що допомагає краще зрозуміти такі поняття, як валентність, молекулярна геометрія і міжмолекулярна взаємодія [11]. Така візуалізація особливо корисна в дослідженнях з органічної хімії, де структура і форма молекул відіграють ключову роль у розумінні хімічної активності та властивостей речовин. Впровадження елементів гейміфікації (наприклад, рівнів, балів і нагород) також може значно підвищити мотивацію та залученість учнів. Гейміфікація може перетворити процес навчання на захопливу гру, в якій користувачі отримують не лише нові знання, а й відчуття досягнення з кожним успішно пройденим етапом навчання. Ці елементи можна поєднувати з системами відстеження прогресу та особистих досягнень, щоб надати користувачам детальний зворотній зв'язок про їхній прогрес та сфери для подальшого навчання.



Таблиця 2.1 є частиною ширшої стратегії розробки застосунку, що прагне забезпечити інтерактивне та залучаюче навчальне середовище. Використання цих елементів дозволяє не тільки покращити розуміння наукових понять, але й робить процес навчання більш захоплюючим та веселим для користувачів. Кожен елемент розробляється з метою максимізувати навчальний потенціал і забезпечити глибоке занурення в матеріал, що в кінцевому підсумку призводить до кращого освітнього результату.

Таблиця 2.1 – Види інтерактивних елементів і мультимедійних ресурсів в освітніх мобільних застосунок

<b>Інтерактивний елемент або мультимедійний ресурс</b>	<b>Опис</b>
Віртуальні лабораторії	Дозволяють користувачам проводити віртуальні хімічні експерименти, маніпулюючи з речовинами та умовами реакції у безпечному середовищі. Це стимулює практичне розуміння хімічних процесів.
Анімації молекулярних структур	Відтворюють динамічні візуалізації молекул і атомів у реакціях, що допомагає зрозуміти молекулярну геометрію та хімічні зв'язки.
Гейміфікація	Включення ігрових елементів, таких як рівні, бали та нагороди, для підвищення мотивації і залучення учнів через конкурентні та навчальні виклики.
Відео лекції та демонстрації	Мультимедійні відео, які пояснюють складні хімічні концепції та демонструють експерименти, забезпечуючи більш глибоке розуміння теми.
Інтерактивні квізи та тести	Дозволяють користувачам перевіряти свої знання та отримувати зворотний зв'язок, що сприяє кращому засвоєнню інформації і самооцінці.

Використання інтерактивних елементів і мультимедійних ресурсів у мобільних додатках для вивчення хімії може поглибити розуміння наукових принципів і теорій та зробити процес навчання більш живим і захоплюючим.

Інтеграція інтерактивних елементів у мобільні додатки для вивчення хімії має вирішальне значення для забезпечення більш цікавого та ефективного навчального процесу. Щоб продемонструвати, як інтерактивність може трансформувати навчання хімії, ми використаємо приклад інтеграції одного з ключових інтерактивних елементів – тестових завдань – у структуру мобільного додатку.

Інтеграція тестових модулів у мобільний додаток для вивчення хімії з можливістю відкриття тестів на сайті [zno.osvita.ua](http://zno.osvita.ua) значно підвищує якість навчального процесу. Така інтеграція рис.7 (див. Додаток А) забезпечує доступ до реальних тестових завдань ЗНО, максимально наближених до умов справжнього іспиту, та надає можливість відразу бачити правильні відповіді і пояснення.

#### Процес інтеграції тестів

##### а) розробка тестового модуля:

- використання WebView: У мобільний додаток інтегрується компонент WebView, який дозволяє користувачам безпосередньо відкривати веб-сторінки з тестами на сайті [zno.osvita.ua](http://zno.osvita.ua);
- налаштування WebView: Забезпечується коректне відображення контенту з сайту, включаючи навігацію між тестами, введення відповідей та отримання результатів;

##### б) реалізація інтерактивних можливостей:

- інтерактивні тести: Завдання з хімії, представлені на сайті, дозволяють студентам виконувати їх в інтерактивному режимі. Кожне завдання супроводжується описом правильного виконання та схемою оцінювання;

- реальні умови тестування: Спосіб виконання завдань та форма відповідей максимально наближені до реальних умов ЗНО, що допомагає студентам ефективніше готуватися до іспитів;
- в) обробка результатів:
- автоматичне оцінювання: Після виконання всіх тестів студент отримує правильні відповіді на всі завдання та обчислений результат у тестових і рейтингових балах;
  - аналіз часу виконання: Зазначається час, витрачений на виконання тесту, що дозволяє оцінити свою швидкість та ефективність;
- г) додаткові функції:
- можливість дострокового завершення: Студенти можуть завершити тест достроково, натиснувши відповідну кнопку, що дозволяє гнучко керувати своїм часом;
  - повний список тестових завдань: Доступ до повного списку всіх тестових завдань зовнішнього незалежного оцінювання з хімії, з можливістю отримати правильну відповідь на кожне завдання відразу після його виконання.

#### Переваги інтеграції з [zno.osvita.ua](http://zno.osvita.ua)

- доступ до реальних тестів: Студенти мають можливість виконувати завдання, максимально наближені до реальних умов ЗНО, що підвищує ефективність підготовки;
- миттєвий зворотний зв'язок: Після завершення кожного завдання студент отримує правильну відповідь і пояснення, що сприяє кращому засвоєнню матеріалу;
- аналіз результатів: Результати виконаних тестів включають оцінки у тестових та рейтингових балах, а також час виконання, що допомагає студентам оцінити свої знання та вдосконалити навички.

Інтеграція тестових модулів з можливістю відкриття сайту [zno.osvita.ua](http://zno.osvita.ua) створює комплексний навчальний інструмент, що підвищує ефективність та інтерактивність навчання хімії.

## 2.2 Технологічний процес створення застосунку

Основні моменти розробки застосунку які описані нижче показані на рис. 1-8 (див. Додаток А).

Процес розробки освітнього мобільного додатку для вивчення хімії поєднує в собі методи програмування та педагогічні методи для створення ефективного навчального інструменту. Процес розпочався з використання середовища розробки Android Studio, яке є основним інструментом для налаштування та програмування додатку рис.1 (див. Додаток А) . Android Studio надає широкі можливості редагування коду, управління залежностями та налагодження, що дозволяє розробникам ефективно реалізовувати заплановану функціональність.

Основною мовою програмування, що використовується для написання коду, є Kotlin, яка була обрана завдяки своїй безпеці, виразності та сумісності з Java, що робить її ідеальною для розробки сучасних Android-додатків. Програмування мовою Kotlin в Android Studio включає розробку користувацького інтерфейсу з використанням Jetpack Compose – оновленої бібліотеки для створення декларативних інтерфейсів, які спрощують візуалізацію компонентів інтерфейсу та забезпечують високу швидкість відгуку додатку рис.2 (див. Додаток А).

Розробка мобільного додатку починається з детального плану, який включає визначення ключових освітніх цілей та технічних вимог. Перш ніж приступити до програмування, ми досліджуємо потреби учнів та вчителів, щоб переконатися, що додаток буде не лише насиченим контентом, але й відповіда-

тиме сучасним освітнім стандартам та очікуванням користувачів. Процес розробки поділяється на кілька ключових етапів, кожен з яких вимагає високого рівня взаємодії та координації між командою розробників. Першим кроком було налаштування середовища розробки. Ми використали Android Studio, яке є одним з провідних середовищ розробки додатків для Android, що надає всі необхідні інструменти для ефективної роботи. Android Studio дозволила нам використовувати Kotlin як основну мову програмування, що було вдалим вибором завдяки його виразності, безпеці та простоті використання.

Кодування додатку розпочалося з розробки архітектури додатку. Ми обрали патерн проектування MVVM (Model-View-View Model), який допомагає нам організувати код таким чином, що користувацький інтерфейс залишається відокремленим від бізнес-логіки рис.4 (див. Додаток А). ViewModel взаємодіє з моделлю даних для отримання інформації, яка потім відображається в інтерфейсі користувача за допомогою декларативних методів Jetpack Compose рис.6 (див. Додаток А). Ключовою особливістю нашого додатку є використання інтерактивних елементів, таких як інтерактивні тести. Ці елементи запрограмовані за допомогою компонентів Jetpack, щоб забезпечити плавний користувацький досвід.

Завершальним етапом розробки є тестування додатку. Цей етап передбачає ручне та автоматизоване тестування всіх функцій для виявлення та виправлення помилок, що дозволяє значно підвищити стабільність та надійність продукту. Після успішного тестування додаток готовий до розгортання та використання в навчальному процесі.

Таблиця 2.2 систематизує всі ключові етапи розробки мобільного додатку для вивчення хімії, починаючи з технічної підготовки і закінчуючи випуском та вдосконаленням продукту, забезпечуючи структурований підхід до створення якісного освітнього інструменту.

Таблиця 2.2 – Алгоритм розробки мобільного застосунку для вивчення хімії

Етап	Діяльність	Інструменти/Технології
Налаштування середовища	Встановлення та конфігурація Android Studio, включення необхідних плагінів та бібліотек.	Android Studio, Kotlin, Jetpack Compose
Кодування	Створення модулів для обробки даних (Model), управління даними (ViewModel) та представлення (View).	Kotlin, Jetpack Compose
Інтеграція зовнішніх служб	Підключення до API для наукових даних, інтеграція баз даних для збереження інформації.	RESTful API, бази даних
Тестування	Виконання модульних, інтеграційних та прийомних тестів.	Android Studio
Випуск та оцінка	Розгортання додатку, збір зворотного зв'язку, внесення виправлень.	Google Play Store, зворотний зв'язок від користувачів

Використання такого додатку для викладання хімії має великий сенс, оскільки сучасні студенти звикли до цифрових технологій, і традиційні методи навчання здаються їм менш цікавими. Додатки з інтерактивними елементами та мультимедійними ресурсами можуть зробити процес навчання хімії більш цікавим та ефективним, підвищити мотивацію учнів і дати їм можливість глибше зрозуміти предмет через практичний досвід та візуалізацію.

Інтеграція цих технік та інструментів дозволила нам створити освітній мобільний додаток, який є не лише технологічно досконалим, але й глибоко залучає учнів до процесу навчання хімії, роблячи його інтерактивним, простим у використанні та ефективним.

### 3 СТРАТЕГІЇ ТЕСТУВАННЯ ТА МЕТОДИ ВПРОВАДЖЕННЯ МОБІЛЬНОГО ЗАСТОСУНКУ З ХІМІЇ

#### 3.1 Стратегії та методи тестування мобільних застосунків

Піраміда тестування (рис.2.1) — це модель, яка ілюструє різні рівні тестування програмного забезпечення та їх розподіл у загальному процесі розробки.

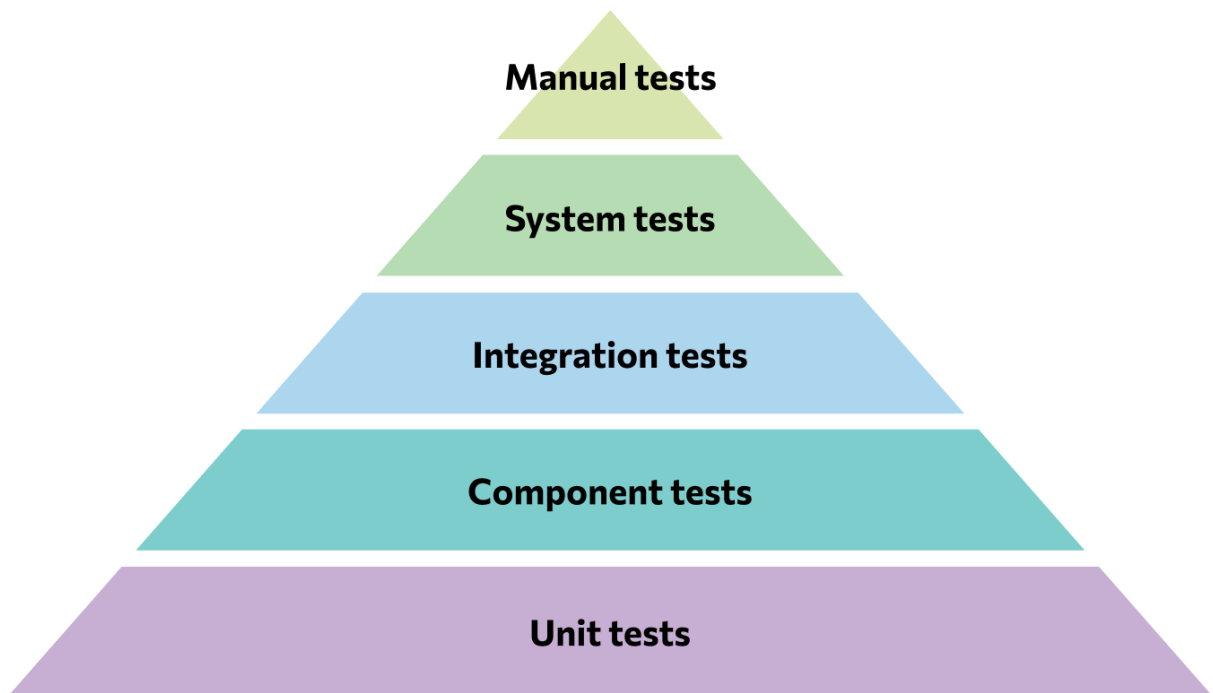


Рисунок 2.1 – Піраміда тестування

Кожен рівень піраміди має своє місце та роль у забезпеченні якості програмного забезпечення. Більша частина тестування виконується на нижніх рівнях піраміди, де тести простіші та дешевші у виконанні, тоді як верхні рівні призначені для більш комплексних і дорогих у виконанні тестів.

Тестування мобільних хімічних додатків є важливим етапом у процесі розробки, оскільки воно гарантує виявлення та виправлення помилок, покращення якості продукту і, зрештою, задоволення кінцевого користувача. Прийняття систематичної стратегії і підходу до тестування під час мобільної розробки, особливо при використанні Kotlin, Android Studio, Jetpack Compose і Material Design, має фундаментальне значення для створення надійних і ефективних додатків.

Першим кроком у тестуванні мобільного додатку є перевірка компонентів додатку за допомогою модульних тестів, а інтеграція Kotlin з такими бібліотеками, як JUnit, забезпечує зручне середовище для модульного тестування, яке розробники можуть використовувати для перевірки логіки без запуску додатку [14]. Ці тести допомагають переконатися, що кожен метод в коді працює коректно за незалежних умов. Наступним кроком є інтеграційне тестування, яке вивчає взаємодію між компонентами програми. Android Studio полегшує цей процес, надаючи інструменти налагодження та профілювання для пошуку проблем в інтеграції компонентів. Цей тип тестування важливий для виявлення помилок, які можуть бути неочевидними на рівні юніт-тесту.

Щоб забезпечити високу якість користувацького інтерфейсу, тестування користувацького інтерфейсу включає використання Jetpack Compose. Тестування користувацького інтерфейсу в Android Studio автоматично імітує взаємодію користувача з додатком, таку як кліки, свайпи та інші жести. Це гарантує, що користувацький інтерфейс реагує ефективно і відповідно до плану, а елементи Material Design візуально відповідають очікуванням. Важливо проводити тестування залучення користувачів, щоб зрозуміти, як реальні користувачі взаємодіють з додатком. Це включає групове тестування, опитування та інтерв'ю з потенційними користувачами. Цей тип тестування може виявити неочікувані помилки в користувацькому інтерфейсі або проблеми зі зрозумілістю контенту, що важливо для освітніх додатків. Завершальним етапом тестування є перевірка продуктивності та безпеки додатку [15]. Це включає в себе перевірку того, що додаток стабільно працює під високим наванта-



женням, не споживає надмірних ресурсів пристрою і відповідає всім стандартам безпеки даних. Цей тип тестування гарантує, що додаток є надійним і безпечним у використанні, що є критично важливим для додатків, які використовуються в освітньому процесі.

Оптимізація процесу тестування мобільного додатку з хімії передбачає не лише перевірку його функціональності, але й забезпечення досягнення освітніх цілей на широкому спектрі пристроїв з різною конфігурацією. Використання Kotlin у поєднанні з Android Studio, Jetpack Compose та Material Design надає розробникам потужний інструментарій для створення мобільних додатків, які також потребують передових стратегій тестування для забезпечення їхньої якості та стабільності.

Мова програмування Kotlin має вбудовані механізми модульного тестування, які дозволяють розробникам швидко та ефективно тестувати окремі функції та методи програми. Це особливо важливо, коли в додатку реалізовано багато складних алгоритмів обробки даних, які відіграють ключову роль у процесі навчання. З іншого боку, Android Studio надає розширені можливості інтеграційного тестування для перевірки взаємодії між різними компонентами програми. Це особливо важливо для перевірки того, що компоненти інтерфейсу користувача, розроблені за допомогою Jetpack Compose, правильно взаємодіють з внутрішньою логікою, яка керує даними та бізнес-логікою додатку.

Оскільки Jetpack Compose дозволяє створювати декларативні користувацькі інтерфейси, розробники можуть легше реалізовувати динамічні взаємодії в інтерфейсі користувача, що є критично важливим для освітніх додатків, де взаємодія студента з підручником повинна бути максимально цікавою та інтуїтивно зрозумілою. Однак це також створює труднощі для тестувальників, які повинні забезпечити узгодженість і адаптивність користувацького інтерфейсу на різних пристроях. Методи Material Design допомагають створювати візуально привабливі інтерфейси, але вони також потребують ретельної перевірки на адаптивність до різних розмірів і орієнтації екрану [16]. Тестування візуального дизайну інтерфейсів на відповідність стандартам Material Design

важливо для того, щоб переконатися, що користувацький досвід залишається незмінним, незалежно від пристрою, на якому запущено додаток.

Таблиця 3.1 є інструментом для планування та організації процесу тестування, допомагаючи розробникам зосередити зусилля на ключових аспектах кожного типу тестування. Інтеграція цих стратегій в розробку мобільного застосунку дозволить забезпечити його високу якість, надійність і користувацьку привабливість, а також сприятиме ефективності навчального процесу через застосунок.

Таблиця 3.1 – Стратегії та методи тестування мобільного застосунку з хімії

<b>Тип тестування</b>	<b>Методи тестування</b>	<b>Особливості тестування</b>
Юніт-тестування	Використання Kotlin та JUnit для перевірки окремих функцій та методів.	Перевірка внутрішньої логіки компонентів програми на коректність обробки даних і виконання алгоритмів.
Інтеграційне тестування	Застосування Android Studio для тестування взаємодії між компонентами додатку.	Перевірка зв'язків між UI компонентами Jetpack Compose та бекенд сервісами застосунку.
Випробування користувачів	Організація групових тестувань, анкетувань та інтерв'ю з потенційними користувачами.	Оцінка загального досвіду користувачів, зручності навігації та залученості освітнього контенту.
Тестування продуктивності	Аналіз виконання застосунку під високим навантаженням і перевірка споживання ресурсів пристрою.	Забезпечення стабільної роботи застосунку без зайвого споживання батареї та ресурсів пам'яті.

Зважаючи на ці особливості, стратегія тестування мобільних додатків з хімії повинна включати не лише тестування коду та інтерфейсу, але й оцінку загального користувацького досвіду та взаємодії з навчальними матеріалами.

Це включає тестування на реальних пристроях, тестування з реальними користувачами для отримання зворотного зв'язку та оцінку продуктивності додатку з плином часу.

Тестування мобільних додатків для вивчення хімії є важливою частиною процесу розробки, оскільки цей етап забезпечує надійність та ефективність кінцевого продукту. Процес тестування організований таким чином, що всі аспекти програми ретельно перевіряються, починаючи з тестування базової функціональності і закінчуючи комплексним тестуванням користувацького досвіду. Всі ці етапи вимагають глибоких знань не тільки програмування, але й специфіки досліджуваної теми (в нашому випадку – хімії).

По-перше, розробка і тестування додатків починається зі створення стабільного середовища в Android Studio, де програмісти пишуть код на мові Kotlin. Це середовище надає зручні інструменти для налагодження та профілювання коду, що дозволяє швидко виявляти та виправляти помилки на ранніх стадіях розробки. Код програми побудований за моделлю MVVM, яка допомагає відокремити бізнес-логіку від користувацького інтерфейсу, забезпечуючи більшу гнучкість і легкість тестування.

Кожен модуль додатку проходить модульне тестування за допомогою фреймворку JUnit. Ці тести зосереджені на перевірці окремих функцій і класів, щоб переконатися, що логіка програми відповідає специфікації. Наступним кроком є інтеграційне тестування, яке забезпечує коректну взаємодію компонентів додатку. Наприклад, важливо переконатися, що модуль віртуальної лабораторії правильно взаємодіє з базою даних хімічних елементів.

Особливу увагу ми приділяємо тестуванню користувацького інтерфейсу, що особливо важливо для освітніх додатків. Для створення інтерактивних та візуально привабливих інтерфейсів ми використовуємо Jetpack Compose. Цей інструмент дозволяє нам імітувати дії користувача, такі як кліки, свайпи та інші жести для тестування користувацького інтерфейсу, що має вирішальне значення для забезпечення зручності та інтуїтивної зрозумілості додатку.

Важливою частиною нашої стратегії є використання тестування залучення користувачів, яке передбачає випробування додатку в навчальному процесі та збір відгуків від реальних студентів та викладачів. Це не лише допомагає виявити технічні помилки, але й покращити викладання та навчання, адаптуючи контент до потреб користувачів. Заключний етап – тестування продуктивності, де ми гарантуємо, що додаток ефективно працює на різних пристроях і конфігураціях, не викликаючи проблем з продуктивністю або споживанням ресурсів. На цьому етапі ми гарантуємо, що додаток є надійним і що широке коло студентів зможе використовувати додаток незалежно від обладнання, яке вони використовують.

Кожен крок чітко задокументований з процедурами та результатами, включаючи скріншоти тестування та детальні звіти про хід роботи, які дозволяють нам наочно продемонструвати ефективність наших методів тестування та їх важливість для створення якісного освітнього продукту.

### **3.2 Розгортання додатку та залучення користувачів**

Розгортання мобільного застосунку з хімії та залучення користувачів є завершальними, але дуже важливими етапами у процесі його розробки. Цей процес вимагає детального планування та виконання низки кроків, які забезпечують не тільки технічну справність застосунку, але й його популярність та ефективність у освітньому процесі.

Після завершення тестування та впевненості у стабільності та продуктивності застосунку, перший крок у процесі розгортання полягає в його публікації на платформах, таких як Google Play. Використання середовища розробки Android Studio спрощує підготовку пакета застосунку (APK або App Bundle) та його подальшу дистрибуцію. Важливо забезпечити, що всі метадані застосунку, включаючи описи, знімки екрану, та відео-демонстрації, якісно

відображають функціональність та переваги застосунку, були точно та привабливо представлені, щоб привернути увагу потенційних користувачів.

Паралельно з розгортанням застосунку, важливо розробити стратегію маркетингу та залучення користувачів. Оскільки застосунок орієнтований на студентів та викладачів хімії, ефективним кроком може бути співпраця з освітніми інституціями, науковими спільнотами та освітніми блогерами для проведення вебінарів, семінарів та інших заходів, які демонструють можливості застосунку [17]. Також корисним може бути розміщення інформації про застосунок на освітніх платформах, форумах та у соціальних мережах. Використання технологій Jetpack Compose та Material Design не тільки сприяє створенню зручного та естетично привабливого інтерфейсу, але й може бути важливою частиною маркетингової стратегії.

Таблиця 3.2 є інструментом для планування та координації впровадження застосунку, який включає всі необхідні заходи для забезпечення його успіху. Використання цих кроків допоможе не тільки забезпечити технічну готовність продукту, але й забезпечити його прийняття серед цільової аудиторії, підвищуючи шанси на комерційний успіх і визнання в освітньому секторі.

Таблиця 3.2 – Ключові кроки розгортання та залучення користувачів для мобільного застосунку з хімії

<b>Крок</b>	<b>Опис діяльності</b>
Публікація на платформах	Реєстрація та розміщення застосунку на платформах, таких як Google Play, включаючи підготовку всіх необхідних метаданих та маркетингових матеріалів.
Маркетингова стратегія	Розробка та реалізація маркетингової стратегії, яка включає співпрацю з освітніми блогерами, проведення вебінарів та розміщення реклам у соціальних мережах.
Тестування користувацького інтерфейсу	Використання інструментів Android Studio для автоматизації тестування користувацького інтерфейсу, забезпечуючи, що UI відповідає очікуванням користувачів.

## Продовження таблиці 3.2

<b>Крок</b>	<b>Опис діяльності</b>
Збір зворотного зв'язку	Організація збору зворотного зв'язку від користувачів через анкети, форуми та прямі відгуки в застосунку для виявлення потенційних проблем та покращень.
Оновлення застосунку	Регулярне оновлення застосунку з новими функціями та виправленнями, заснованими на зворотному зв'язку користувачів та аналізі використання.

Для того, щоб підтримувати інтерес користувачів до додатку, важливо регулярно оновлювати його, додаючи нові функції, навчальні модулі та покращуючи функціональність на основі відгуків користувачів.

## ВИСНОВКИ

Розробка мобільного застосунку з хімії на платформі Android, заснована на глибокому розумінні сучасних технологій та інструментів програмування, представляє собою комплексне завдання, що вимагає інтеграції багатьох аспектів від планування до впровадження. Використання мови програмування Kotlin у поєднанні з середовищем розробки Android Studio стало фундаментом для створення стабільного та ефективного застосунку. Kotlin надає розробникам сучасні можливості та інструменти для реалізації якісного коду, що забезпечує високу продуктивність і безпеку застосунку.

Архітектурні принципи, використовуючи Jetpack, дозволили впровадити модульний підхід до структури програми, що полегшило тестування, підтримку та розширення функціональності застосунку. Jetpack Compose, у свою чергу, відіграв ключову роль у проектуванні інтерфейсу користувача, дозволяючи створювати інтуїтивно зрозумілі та візуально привабливі UI, що відповідають сучасним стандартам дизайну.

Огляд існуючих застосунків для вивчення хімії підкреслив необхідність інновацій та додавання унікальної вартості, що мотивувало нас на створення більш залучаючого та інтерактивного досвіду навчання. Це було реалізовано через інтеграцію мультимедійних ресурсів та інтерактивних елементів, які збагачують навчальний процес та роблять його більш ефективним і захоплюючим.

Продовжуючи процес розробки мобільного застосунку з хімії, значну увагу було приділено стратегіям та методам тестування, що є ключовими для забезпечення високої якості кінцевого продукту. Використання модульних та інтеграційних тестів дозволило глибоко аналізувати кожен аспект застосунку, від логіки обробки даних до коректної взаємодії між компонентами. Це забезпечує надійність і стабільність системи. Застосування UI тестів з інструментами, як Espresso, в Android Studio забезпечило додаткову перевірку зручності

користувачів, відповідності інтерфейсу стандартам Material Design та забезпечення інтуїтивно зрозумілої взаємодії. Останнім кроком у розробці було розгортання додатку та стратегії залучення користувачів. Публікація застосунку в Google Play та інших платформах була супроводжувана масштабною маркетинговою кампанією, що включала співпрацю з освітніми блогами, проведення вебінарів та активну присутність у соціальних медіа. Залучення користувачів також стимулювалось через оновлення контенту, введення нових функцій та постійний збір та аналіз відгуків користувачів, що дозволяло швидко реагувати на їхні потреби та покращувати застосунок.

Застосунок не тільки задовольняє академічні потреби студентів і викладачів, але й надає мотивуюче та залучаюче середовище для вивчення хімії. Використання передових технологій програмування та дизайну, ретельно продуманих стратегій тестування і залучення користувачів допомогли створити застосунок, який вирізняється на ринку освітніх технологій та вносить істотний вклад у сучасну хімічну освіту.



## ПЕРЕЛІК ПОСИЛАНЬ

1. Андреев В. О., Лебідь Н. П. Розробка мобільних застосунків з використанням Kotlin: монографія. Львів : Видавництво ЛНУ, 2018. С. 34–45.
2. Бережна І. П., Галаган Л. І. Методи інтерактивного навчання у вищій освіті. *Освіта України*. №11. 2017. С. 21–30.
3. Василенко В. А., Мельник Л. Г. Сучасні підходи до викладання хімії в школах. *Науковий світ*. №4. 2019. С. 88–95.
4. Гончаренко С. У., Кравчук Л. Т. Освітні технології у школах України : аналіз ефективності: монографія. Харків : Факт, 2016. С. 77–85.
5. Дем'яненко М. В., Петрова О. В. Методи тестування програмного забезпечення : підручник. Київ : Каравела, 2020. С. 102–118.
6. Жук Ю. О., Петриченко М. Ф. Android Studio як середовище розробки мобільних застосунків. *Вісник XIII*. №7. 2018. С. 14–23.
7. Захарчук В. Б., Литвин О. П. Принципи використання Material Design в мобільних застосунках. *Техніка*. №3, 2017. С. 53-62.
8. Карпенко Л. О., Шевченко В. К. Інновації в освіті: використання мобільних застосунків для навчання: монографія. Одеса: Астропринт, 2019. С. 134-145.
9. Ковальчук В. Я., Шрамко Я. Б. Історія розвитку мобільних застосунків. *Історія науки і техніки*. №2, 2016. С. 112-120.
10. Литвинова С. Г., Петрова О. В. Інтерактивні методи навчання в школах. *Наука і освіта*. №6, 2021. С. 74-82.
11. Мельник Л. Г., Васильєв Ю. М. Мобільні технології в освіті. *Вісник педагогічних наук*. №4, 2018. С. 47-55.
12. Назаренко Л. А., Сергієнко В. П. Основи програмування на Kotlin: підручник. Київ: Либідь, 2019. С. 66-79.
13. Олійник В. О., Кузьменко В. К. Застосування Jetpack Compose в розробці інтерфейсів. *Вісник ЛПУ*. №9, 2020. С. 35-43.

14. Петренко О. М., Сідоров Б. І. Використання мультимедіа у викладанні хімії. *Освіта і наука*. №3, 2017. С. 92-101.
15. Рибальченко В. К., Ткачук В. В. Технології тестування мобільних застосунків. *Техніка*. №5, 2020. С. 23-30.
16. Савченко В. О., Кулішов В. В. Новітні технології у викладанні природничих наук. *Наука і життя*. №11, 2016. С. 17-25.
17. Сидорова Л. Е., Грищенко В. К. Практичне використання Material Design у мобільних застосунках. *Вісник науки*. №8. 2018. С. 55–63.

## ДОДАТОК А

## Код застосунку

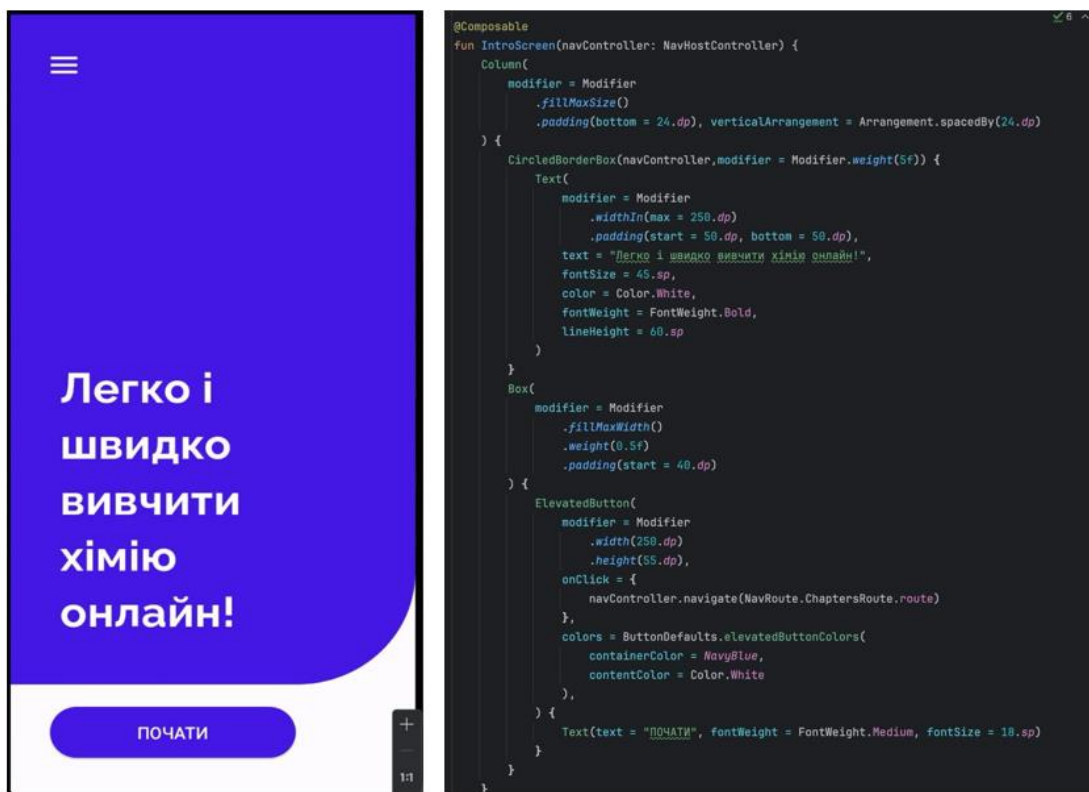


Рисунок А.1 – Головний екран



Рисунок А.2 – Екран з розділами

## Розділ 1

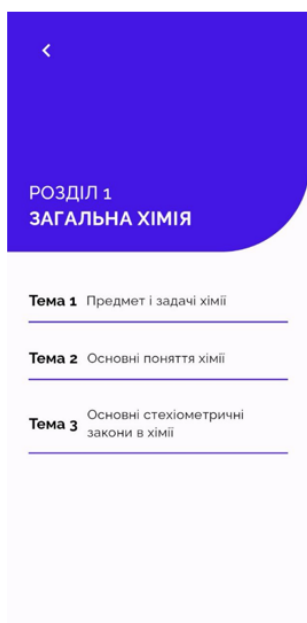
### Загальна Хімія

```

@Composable
fun ChapterFile(index: Int, chapterInfo: ChapterInfo, onClick : () -> Unit) {
    Box(modifier = Modifier
        .clip(RoundedCornerShape(30.dp))
        .fillMaxWidth()
        .height(100.dp)
        .background(color = if (index % 2 == 0) NavyBlue else Purple)
        .clickable {
            onClick()
        }) {
        Row(modifier = Modifier
            .fillMaxWidth()
            .padding(horizontal = 20.dp, verticalAlignment = Alignment.CenterVertically, horizontalArrangement = Arrangement.SpaceBetween){
                Column(modifier = Modifier.fillMaxHeight(), verticalArrangement = Arrangement.Center){
                    Text(text = "Позитив ${index + 1}", color = Color.White, fontWeight = FontWeight.Bold, fontSize = 22.sp)
                }
                Text(modifier = Modifier.widthIn(max = 250.dp), text = chapterInfo.title, color = Color.White, fontWeight = FontWeight.Medium, fontSize = 20.sp)
            }
        )
    }
}

```

Рисунок А.3 – Розділ



```

@Composable
fun ChapterScreen(navController: NavHostController, viewModel: MainViewModel, chapterId: String?) {
    val currentChapter = viewModel.currentChapter
    Column(modifier = Modifier.fillMaxSize()) {
        PageHeader(navController = navController, title = "Позитив $chapterId", subtitle = currentChapter?.title?.uppercase() ?: "Загальна хімія", weight = ??)
        currentChapter?.let {
            LazyColumn(
                modifier = Modifier.weight(1f),
                verticalArrangement = Arrangement.spacedBy(32.dp),
                contentPadding = PaddingValues(horizontal = 30.dp, vertical = 50.dp),
            ) {
                itemsIndexed(it.topicsList) { index, topicInfo ->
                    TopicFile(index, topicInfo) {
                        viewModel.changeCurrentTopic(topicInfo)
                        navController.navigate(NavRoute.TopicRoute("${index+1}").route)
                    }
                }
            }
        }
    }
}

```

Рисунок А.4 – Екран з підрозділами

# Тема 1 Предмет і задачі хімії

```
@Composable
fun TopicTile(index: Int, topicInfo: TopicInfo, onClick: () -> Unit) {
    Box(modifier = Modifier
        .fillMaxWidth().clickable {
            onClick()
        }) {
        Column(modifier = Modifier.fillMaxSize()){
            Row(modifier = Modifier.fillMaxWidth(), verticalAlignment = Alignment.CenterVertically, horizontalArrangement = Arrangement.spacedBy(12.dp)) {
                Text(text = "Тема ${index + 1}", color = Color.Black, fontWeight = FontWeight.Bold, fontSize = 20.sp)
                Text(text = topicInfo.title, color = Color.Black, fontWeight = FontWeight.Normal, fontSize = 18.sp)
            }
            Spacer(modifier = Modifier.height(16.dp))
            Divider(color = NavyBlue, thickness = 2.dp)
        }
    }
}
```

Рисунок А.5 – Підрозділ



```
@SuppressLint("SetJavaScriptEnabled")
@Composable
fun TopicScreen(navController: NavHostController, viewModel: MainViewModel, topicId: String?) {
    val context = LocalContext.current
    val currentTopic = viewModel.currentTopic

    Scaffold(
        floatingActionButton = {
            ExtendedFloatingActionButton(
                containerColor = NavyBlue,
                onClick = {
                    navController.navigate(NavRoute.TestRoute(topicId = "$topicId").route)
                },
                icon = { Icon(Icons.Filled.Star, contentDescription: "Пройти тест", tint = Color.White) },
                text = { Text(text = "Пройти тест", color = Color.White) },
            )
        }
    ) {
        Column(
            modifier = Modifier
                .padding(it)
                .fillMaxSize()
        ) {
            PageHeader(
                navController = navController,
                title = "Тема $topicId",
                subtitle = currentTopic?.title?.uppercase() ?: "",
                weight = 7f
            )
            CustomWebView(url = "http://zno.academia.in.ua/mod/book/tool/print/index.php?id=${currentTopic?.description}?: 3748")
        }
    }
}
```

Рисунок А.6 – Екран з текстом підрозділу

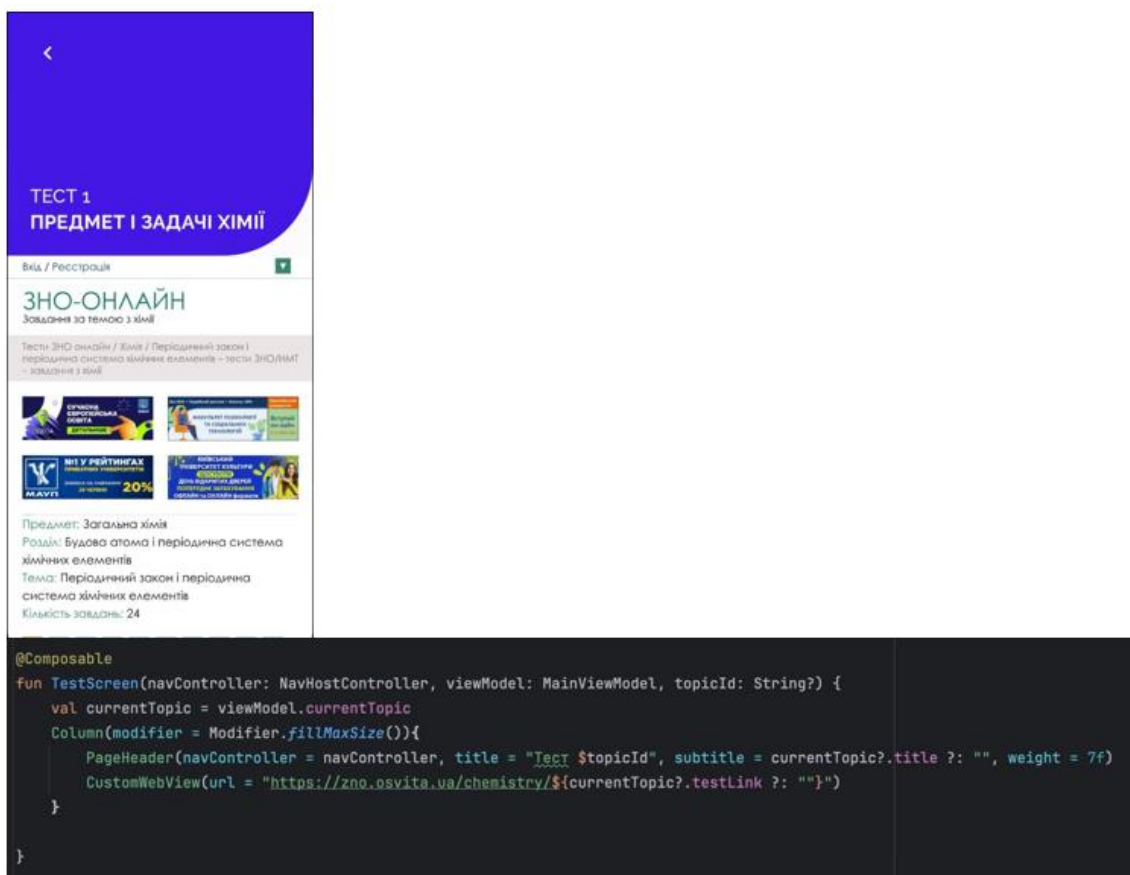


Рисунок А.7 – Екран з тестом

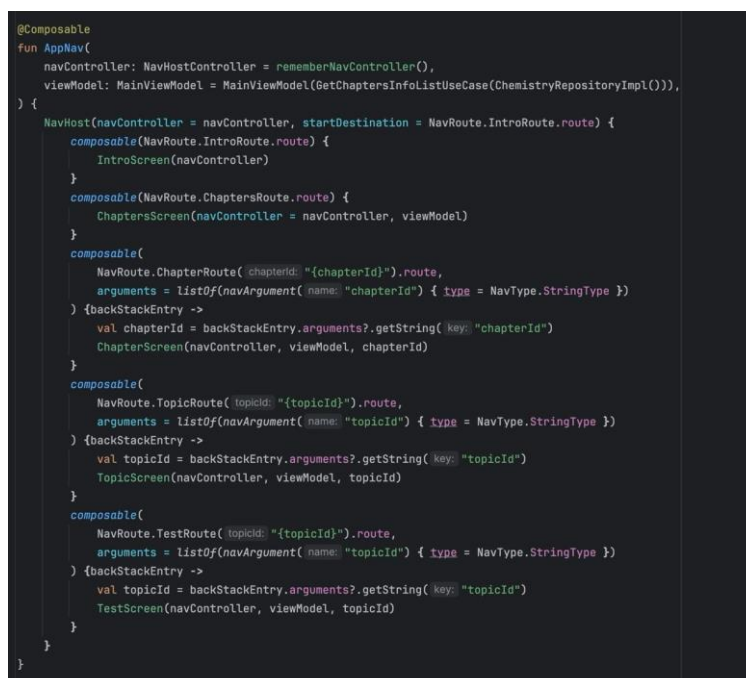


Рисунок А.8 – Навігація застосунку