Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Метод та засіб агрегації новинного контенту з інформаційних електронних джерел

(тема)

Виконав:

студент II курсу, групи СПм-23-1

Могильний А.Є.
(прізвище, ініціали)

Спеціальність

123 «Комп’ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма

Системне програмування
(повна назва освітньої програми)

Керівник: ст.викл. Єрошенко О.А.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп’ютерна інженерія»
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
“____” _____ 20___ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Могильному Андрію Єгоровичу
(прізвище, ім’я, по батькові)

1. Тема роботи Метод та засіб агрегації новинного контенту з інформаційних електронних джерел

затверджена наказом по університету від “ 22 ” листопада 2024 р. № 1236 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 20 січня 2025 р.

3. Вхідні дані до роботи середовище розробки Android Studio, мова програмування Kotlin

4. Перелік питань, що потрібно опрацювати у роботі

Аналіз створення сучасних програмних застосунків.

Вибір технологій.

Прикладне програмне забезпечення "Новинного порталу".

Інструкція користувача програмного засобу.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів)

Слайд-презентація – 16 слайдів.

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Ознайомлення з літературними джерелами	26.11.24-30.11.24	
2	Розробка алгоритмів рішення та вибір засобів	02.12.24-05.12.24	
	вирішення завдань		
3	Вибір інструментальних засобів	06.12.24-10.12.24	
4	Проектування застосунку	11.12.24-21.12.24	
5	Налагодження та тестування програми	23.12.24-03.01.25	
6	Оформлення матеріалів кваліфікаційної	04.01.25-07.01.25	
	роботи		
7	Подання кваліфікаційної роботи керівникові	08.01.25-11.01.25	
	та її попередній захист		
8	Подання кваліфікаційної роботи на	13.01.25-17.01.25	
	рецензування		

Дата видачі завдання 25 листопада 2024 р.

Студент _____
(підпис)

Керівник роботи _____ ст. викл. Єрошенко О.А.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 62 с., 19 рис., 0 табл., 1 дод., 16 джерел.

НОВИННИЙ ПОРТАЛ, ІНТЕРНЕТ, ПЛАТФОРМА, СТАТТЯ, МОВА ПРОГРАМУВАННЯ, ЛОГІЧНА ЧАСТИНА, API, KOTLIN, ANDROID STUDIO.

Метою кваліфікаційної роботи є розробка мобільного застосунку для новинного порталу, який агрегує новини та має основні функції. У ході виконання кваліфікаційної роботи були досліджені технології проектування мобільних новинних застосунків, створено інтерфейс та реалізовано функціонал, що включає пошуковий механізм для знаходження новин, можливість збереження цікавих матеріалів для подальшого перегляду, а також детальний перегляд кожної новини.

Крім того, було проаналізовано потреби цільової аудиторії, що дозволило визначити ключові функції, які повинні бути впроваджені в застосунок. Це включає можливість фільтрації новин за категоріями, такими як політика, економіка, спорт, культура та технології. Такий підхід забезпечує користувачам зручний доступ до інформації, яка їх цікавить, і дозволяє швидко орієнтуватися у великій кількості контенту.

Для реалізації проекту було обрано мову програмування Kotlin, яка є сучасною та потужною платформою для розробки Android-застосунків. Використання Kotlin дозволяє скоротити час розробки завдяки простоті синтаксису та потужним можливостям інтеграції з існуючими бібліотеками Java.

ABSTRACT

Master’s thesis: 62 pages, 19 figures, 0 tables, 1 appendices, 16 sources.

NEWS PORTAL, INTERNET, PLATFORM, ARTICLE, PROGRAMMING LANGUAGE, LOGICAL PART, API, KOTLIN, ANDROID STUDIO.

The purpose of the qualification work is to develop a mobile application for a news portal that aggregates news and has basic functions. During the qualification work, mobile news application design technologies were studied, an interface was created and functionality was implemented, including a search engine for finding news, the ability to save interesting materials for later viewing, as well as a detailed view of each news item.

In addition, the needs of the target audience were analyzed, which made it possible to determine the key functions that should be implemented in the application. This includes the ability to filter news by categories such as politics, economics, sports, culture and technology. This approach provides users with convenient access to information that interests them and allows them to quickly navigate a large amount of content.

The Kotlin programming language was chosen for the project, which is a modern and powerful platform for developing Android applications. Using Kotlin allows you to reduce development time due to the simplicity of the syntax and powerful integration capabilities with existing Java libraries.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ 7
ВСТУП 8
1 АНАЛІЗ СТВОРЕННЯ СУЧАСНИХ ПРОГРАМНИХ ЗАСТОСУНКІВ 9
1.1 Аналіз створення сучасних програмних засобів 9
1.2 Аналіз існуючих рішень 11
1.2.1 Google News 12
1.2.2 CNN News 13
1.3 Огляд існуючих засобів розробки 16
1.3.1 Огляд існуючих платформ для розробки 17
1.3.2 Огляд мов програмування для розробки застосунків 18
1.4 Постановка задачі 21
2 ВИБІР ТЕХНОЛОГІЙ 22
2.1 Використані технології у розробці 22
2.2 Огляд засобів розробки 24
2.2.1 Огляд GitHub 25
2.2.2 Архітектурний шаблон MVVM 26
3 ПРОГРАМНА РЕАЛІЗАЦІЯ 29
3.1 Загальний опис 29
3.2 Інтерфейс програми 30
3.3 Програмна реалізація засобу агрегації новинного контенту з інформаційних електронних джерел 34
4 ІНСТРУКЦІЯ КОРИСТУВАЧА ПРОГРАМНОГО ЗАСОБУ 46
ВИСНОВКИ 50
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 51
ДОДАТОК А Графічний матеріал кваліфікаційної роботи 53

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Платформа – це мова програмування і відповідні інструменти та інфраструктура для підтримки процесу розробки застосунку.

БД – база даних, структурована сукупність даних, організованих відповідно до концепції, яка описує характеристики цих даних і взаємозв'язки між їх елементами.

IDE – інтегроване середовище розробки (англ. Integrated Development Environment).

SQL – мова структурованих запитів: декларативна мова програмування для взаємодії користувача з базами даних (англ. Structured Query Language).

MVVM – це архітектурний шаблон для організації компонентів в застосунку (англ. Model-View-ViewModel).

ВСТУП

У сучасну епоху цифрових технологій швидкий доступ до новин став критично важливим аспектом повсякденного життя. Користувачі прагнуть отримувати актуальну інформацію в режимі реального часу, що зумовлено стрімким розвитком інформаційних порталів та їх безперервним вдосконаленням.

В умовах величезної кількості новинних подій, що відбуваються в різних куточках планети, виникає необхідність у спеціалізованих застосунках, які структурують новини за категоріями. Це дозволяє користувачам швидше знаходити інформацію, яка їх цікавить, і відповідати на їхні інтереси. Сьогодні існує безліч новинних порталів, які можуть бути адаптовані для використання на різних пристроях і операційних системах.

Новинний портал може бути реалізований у вигляді програмного застосунку з графічним інтерфейсом, що робить його кросплатформним і доступним для широкого кола користувачів. Враховуючи різноманітність аудиторії — від молоді до літніх людей, від студентів до професіоналів — важливо створити продукт, який буде зрозумілим і зручним для всіх. Таке підходження не лише забезпечує доступ до актуальної інформації, але й відкриває можливості для комерційної реалізації проекту в майбутньому.

Метою даної роботи є детальний опис функціональної складової та графічного інтерфейсу новинного порталу. Для розробки обрано мову програмування Kotlin, яка дозволяє створювати ефективні та сучасні мобільні застосунки. Основними завданнями цього застосунку є створення інтуїтивно зрозумілого інтерфейсу, що забезпечує легкість навігації, а також реалізація зручного функціоналу для фільтрації новин за категоріями. Це допоможе користувачам швидко знаходити потрібну інформацію та залишатися в курсі подій у світі.

1 АНАЛІЗ СТВОРЕННЯ СУЧАСНИХ ПРОГРАМНИХ ЗАСТОСУНКІВ

1.1 Аналіз створення сучасних програмних засобів

Розробка мобільного застосунку для ознайомлення з новинами [1] – це складний процес, котрий вимагає детального планування та усвідомлення потреб потенційних користувачів. Однією з найважливіших задач є проведення аналізу цільової аудиторії, що включає визначення її основних інтересів, вікових і соціально-демографічних особливостей, та можливі моделі поведінки при споживанні контенту. Такий підхід допомагає у розробці функціональної частини, котра буде містити актуальність для різноманітних користувальницьких груп. Наприклад, цільова аудиторія молодшого віку може віддавати пріоритет постійному оновленню даних через пуш-сповіщення, в свою чергу, старша – простій навігації з втіленням кастомізованої стрічки новин. Додатковий аспект – це приналежність користувачів до певної географічної частини світу: застосунок, котрий надає змогу обирати регіональні новини, допоможе у забезпеченні персоналізованого інформування, особливо для тієї категорії споживачів, котра проявляє інтерес до подій в конкретних містах чи країнах.

Ключове значення у розробці застосунку для перегляду новин має вибір основного функціоналу. До основних функцій належить налаштування персоналізованої стрічки, таким чином користувачам надається змога фільтрувати категорії новин за власними побажаннями. Для значної кількості споживачів новинного контенту, важливими є можливості швидкого доступу до новин через push-сповіщення, котрі допомагають в інформуванні про важливі новини в режимі реального часу. При цьому необхідно контролювати частоту отримання таких сповіщень для уникнення «виснаження» користувача, котре в свою чергу змушує його до вимкнення даного функціоналу. Не менш важливою функцією є офлайн-режим, завдяки

чому користувачі мають змогу зберігати необхідні статті для ознайомлення з ними без інтернет-з'єднання. Інтегрування відеоконтенту – це чергова важлива функція, оскільки, відтепер, значно частіше люди надають перевагу формату у вигляді коротких відео, замість текстових матеріалів. Також змога поширювати новинні матеріали в соціальних мережах допомагає в підвищенні охоплення контенту та залучити нової користувальницької групи завдяки спільним інтересам.

Важливу роль у застосунках новинного типу відіграє UX/UI дизайн. Користувальницький інтерфейс має бути інтуїтивно зрозумілим та мати фільтрацію за категоріями, адаптивну тему до часу доби, простий пошук, а також мати велику кількість додаткового функціоналу доступності, наприклад масштабуванням тексту або можливість ознайомлення з новинами в аудіо-форматі для користувачів з обмеженими можливостями. Також важливою складовою є постійний доступ до оновлень контенту, через що необхідно провести забезпечення інтеграції з RSS-стрічками або втілити API новинних ресурсів, котрі проводять автоматичне оновлення даних.

Кросплатформенність [2] застосунку забезпечує його доступність для користувачів різноманітних пристроїв та операційних систем, таким чином розширюється охоплення аудиторію та збільшується популяризація застосунку. Важливою складовою є швидкість завантаження контенту, оскільки сучасні користувачі не готові проводити в очікуванні довше кількох секунд, через що необхідно проводити оптимізацію програмного коду, мінімізацію використання великих за розміром медіафайлів та використовувати сучасне та швидке серверне обладнання. Безпека та конфіденційність даних користувачів є доволі важливими складовими. Застосування шифрування даних є ефективною мірою у захистити персональні налаштування і переваги користувачів, а дотримання законодавчих норм, таких як GDPR, забезпечить дотримання вимог щодо зберігання і обробки даних.

Ще одним важливим аспектом є монетизація застосунку, котра містить

кілька підходів. Перший, впровадження рекламних оголошень у програмному засобі, котрі забезпечуватимуть дохід, та при цьому не мають заважати при перегляді новин. Другий варіант передбачає впровадження платної підписки для доступу до додаткового контенту. Користувачі, які бажають отримати ексклюзивні статті або вимкнути рекламу, можуть оформити таку підписку. Крім того, укладення партнерських угод з медіа-компаніями та брендами відкриває можливості для взаємовигідної співпраці, що дозволяє залучати унікальний контент і підвищувати привабливість застосунку.

Підсумовуючи, створення новинного додатку – це багатогранний процес, який передбачає врахування потреб аудиторії, ретельний вибір функцій, технічну оптимізацію, захист даних і ефективну стратегію монетизації. Це дозволить створити зручний, функціональний і надійний додаток, здатний задовольнити потреби сучасних користувачів у швидкому та якісному отриманні новин.

1.2 Аналіз існуючих рішень

Для створення структури проєкту важливо провести детальний аналіз сучасних рішень, оскільки більшість застосунків містять базову функціональність, таку як фільтрація, відображення категорій та медіа-контент. Ці елементи забезпечують користувачеві зручність та легкість у навігації, дозволяючи швидко знаходити потрібну інформацію. Однак, щоб виділитися серед конкурентів, кожен продукт прагне впровадити унікальні функції, які можуть включати персоналізацію контенту відповідно до уподобань користувача, аналітичні інструменти для збору та аналізу даних про використання застосунку, а також інтеграцію з сторонніми сервісами через API для розширення функціональності.

Правильна архітектура програмного забезпечення також відіграє ключову роль у розробці сучасних застосунків. Вона визначає основні

компоненти системи та їх взаємодію, що дозволяє оцінити вартість розробки та зрозуміти, які елементи є критично важливими для впровадження. Правильно спроектована архітектура забезпечує масштабованість застосунку, що дозволяє легко додавати нові функції та адаптувати його під специфічні вимоги клієнтів. Це також спрощує обслуговування програми в майбутньому, оскільки чітка структура полегшує внесення змін і оновлень.

1.2.1 Google News

Google News надає користувачам для ознайомлення та подальшого користування персоналізовану стрічку новин, котра адаптується до інтересів користувачів за допомогою алгоритмів штучного інтелекту, котрі проводять аналіз взаємодії з контентом. Однією з ключових функцій є повне висвітлення подій, що дозволяє користувачам отримувати новинний контент з різноманітних джерел. Окрім цього застосунок пропонує вибір необхідних категорій новин та проводить підтримку географічної прив'язку, завдяки чому користувач отримує новини свого регіону. Приклад інтерфейсу новинної стрічки Google News зображено на рисунку 1.1.

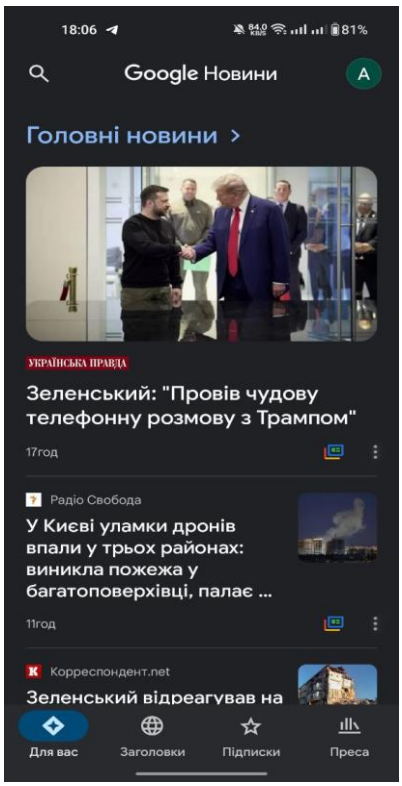

Рисунок 1.1 – Інтерфейс новинної стрічки Google News

Застосунок містить підтримку локалізації новин, також має функцію офлайн-збереження, та синхронізується з персональним акаунтом Google, чим забезпечує доступ до збереженого контенту з будь-якого пристрою.

1.2.2 CNN News

CNN News є провідною глобальна новинна платформою у світі, що належить компанії CNN (Cable News Network) та надає доступ до новин з усього світу завдяки своєму вебсайту та мобільному застосунку. Дана новинна платформа охоплює широкий тематичний спектр. CNN News за багато років зарекомендувала себе як надійне джерело актуальних та неупереджених новин, та проводить орієнтцію на швидке інформування цільової аудиторії стосовно головних подій в світі. Платформа проводить обслуговує численної аудиторію у всьому світі, та орієнтована на жителів США, надаючи новини на декількох мовах та забезпечуючи висвітлення

подій у режимі реального часу.

CNN News містить функціонал, котрий дозволяє дивитися репортажі з місця подій у пряммому ефір в мобільному застосунку та на вебсайті. Така функціональна частина надає змогу швидко відстежувати події в режимі реального часу, включно з різноманітними кризовими ситуаціями, політичними заявами, військовими конфліктами та різними конференціями. CNN також веде трансляції телевізійних програм і шоу.

Рисунок 1.2 – Головна сторінка CNN News

Важливою особливістю, котра виділяє CNN News від аналогів є розділ «Top Stories», в якому редакційна команда проводить відбір найбільш значущі або цікавих новини. Таке джерело актуальних новин охоплює як

національні, так і міжнародні події. Даний розділ постійно оновлюється новими статтями, публікуючи актуальний контент для відведеного часового проміжку. Користувачі, які заходять у додаток для швидкого перегляду новин, можуть легко переглянути всі головні події за день, навіть не потребуючи додаткових налаштувань. Приклад інтерфейсу головної сторінки CNN News зображено на рисунку 1.2.

Платформа CNN News дозволяє користувачам налаштовувати персоналізовану стрічку новин. Користувачі можуть обирати категорії за інтересами, наприклад, бізнес, спорт, подорожі або технології. Ця функція дає можливість отримувати новини, які найбільше відповідають інтересам користувача, що робить додаток більш зручним і корисним для щоденного використання. Персоналізація забезпечує більшу залученість користувачів, оскільки їм пропонуються релевантні матеріали, які можуть бути важливими у професійному або особистому контексті.

CNN News також пропонує оповіщення про екстрені новини (Breaking News Alerts), що дозволяє користувачам миттєво отримувати повідомлення про важливі події. Це особливо зручно під час надзвичайних ситуацій, природних катастроф, терактів чи важливих політичних рішень. Користувачі отримують сповіщення у вигляді пуш-повідомлень, завдяки чому вони можуть залишатися в курсі головних подій навіть тоді, коли не використовують додаток.

Ще однією значною перевагою CNN News є акцент на відеоконтенті. Додаток пропонує безліч відеоматеріалів, включаючи репортажі з місця подій, аналітичні огляди, короткі новини, а також повноформатні програми, що охоплюють найважливіші теми. Користувачі можуть переглядати відеорепортажі з ключових подій дня, наприклад, виступів державних діячів або важливих інтерв'ю. Відеоформат забезпечує не лише швидкий доступ до інформації, а й глибше занурення в контекст, завдяки живим.

Окрім цього, CNN News надає аналітичний контент і редакторські огляди. Це особливо цінне для користувачів, які хочуть не просто знати

новини, а й розуміти їхній глибший контекст і вплив на суспільство. CNN також забезпечує доступ до даних і візуалізацій, які допомагають краще зрозуміти статистику та аналітику, пов'язані з певними подіями. Наприклад, під час виборів або економічних криз CNN News публікує детальні інфографіки та дані, які допомагають користувачам глибше зануритися у суть подій.

На додаток до всього, CNN News інтегрує функцію збереження статей для подальшого перегляду. Це дозволяє користувачам створювати власну бібліотеку новинних матеріалів і повертатися до них у зручний час. Збережені статті можна читати офлайн, що є зручним для користувачів, які мають обмежений доступ до інтернету або хочуть переглядати новини під час подорожей.

1.3 Огляд існуючих засобів розробки

Для початку розробки програмного засобу необхідно визначити платформу, на якій він буде працювати. На тепершній час найпоширенішими є Android, iOS, Windows Phone, Windows, а також кросплатформні рішення. Вибір платформи на якій буде працювати застосунок прямо впливає на процес розробки. Серед існуючих засобів розробки для мобільних застосунків виділяються кілька популярних фреймворків, кожен з яких має свої унікальні особливості. React Native дозволяє створювати кросплатформні програми з використанням JavaScript, забезпечуючи високу продуктивність і нативний інтерфейс. Flutter, розроблений Google, використовує мову Dart і пропонує гаряче перезавантаження для швидкої розробки візуально привабливих додатків. Xamarin дозволяє використовувати C# для кросплатформної розробки, що сприяє повторному використанню коду. Ionic орієнтований на гібридні додатки з використанням HTML, CSS і JavaScript, спрощуючи створення інтерфейсів. NativeScript забезпечує доступ до нативних API за допомогою JavaScript або TypeScript, а

Apache Cordova (PhoneGap) дозволяє використовувати веб-технології для створення кросплатформних додатків. Вибір конкретного інструменту залежить від вимог проєкту, досвіду команди та специфіки цільової аудиторії.

1.3.1 Огляд існуючих платформ для розробки

Android – є однією з найпопулярніших платформ для розробки програмних рішень. Переважно створення застосунків відбувається з використанням мов програмування Java та Kotlin завдяки своїй простоті та ефективним рішенням. Android гарно адаптований до роботи з графікою, що значно спрощує розробку ігор та застосунків, що потребують значної графічної продуктивності. Також маючи відкриту архітектуру Android сприяє розробникам налаштовувати систему під певні потреби, проводити тестування та оптимізацію. Також для даної платформи існує велика кількість базових готових рішень, котрі суттєво спрощують початковий етап розробки.

iOS операційна система, створена компанією Apple для своїх пристроїв. Раніше, в якості основної мови програмування, для розробки застосунків, для жаної операційної системи використувувалася Objective-C. Objective-C і зараз широко використовується в розробці мобільних застосунків для IOS, проте на заміну вийшла мова програмування Swift, котра стала заміняти свого попередника у багатьох напрямках, хоча і на сьогоднійшній день також широко використовується Objective-C. Ця операційна система має вагомий рівень безпеки та високу продуктивність в комплекті до соворих вимог програмних засобів, що можуть бути встановлені на гаджети компанії Apple, й окрім їх подальшої інтеграції в дану екосистему з іншими продуктами.

Windows – є однією з ключових операційних систем, котра використовується на персональних комп’ютерах, ноутбуках, планшетах, смартфонах, фотоапаратах та інших пристроях. Дана платформа має

підтримку багатьох мов програмування, що робить її доволі гнучкою для розробки програмних засобів, зокрема завдяки і потужним інструментам розробки створених для роботи на даній платформі. Windows надає змогу реалізовувати програмні засобу широкого спектру, як для бізнес рішень, так і для більш простих задач.

Окрему нішу в розробці програмного щабезпечення займає кросплатформенність. Дана концепція дозволяє розробляти програмні засоби, котрі мажуть бути підтримуваними на кількох операційних системах одночасно. Це дозволяє значно скоротити витрати на розробку. Python одна з основних мов програмування для даної концепцій, разом з іншими фреймворками. Завдяки кросплатформенні рішенням розроблювальні програмі одночасно працюють як на Android, так і на iOS, інколи ще й на Windows, хоча в таких випадках необхідно створити універсальний дизайн, котрий не буде сильно вирізнятися поміж інших схожих рішень.

До найпоширеніших фреймфорків для кросплатформної розробки відносяться: React Native, котрий використовує JavaScript та надає нативний інтерфейс; Flutter, що базується на Dart та підтримує досить швидке оновлення коду; Xamarin від компанії Microsoft дає змогу розробникам на C# писати код; також Ionic, котрий використовує JavaScript, HTML та CSS для розробки гібридних застосунків. Такий інструментарій дає змогу створювати динамічні й ефективні програми, без додаткової необхідності писати окреми код для кожної операційної системи. За допомоги кросплатформенних рішень програмісти можуть більш серйозніше зосереджуватися на функціональній складовій та графічному дизайну для більшої привабливості та нативного розуміння програми.

1.3.2 Огляд мов програмування для розробки застосунків

Після вибору платформи для розробки в першу черпгу обирається мова програмування, оскільки вона залежить від обраної операційної системи, на

котрій буде базуватися програмний продукт. До найрозповсюдженіших мов для даної задачі відносяться: Java; Kotlin; Swift; C#; Python та Objective-C. Кожна з даних мов програмування має певні специфікації, котрі роблять її більш придатною до певних платформ або задач, чи типів розроблюваних застосунків.

Kotlin – це кросплатформна мова програмування, хоча і використовується переважно для розробки мобільних застосунків для операційної системи Android. Kotlin[3] є статистично типізованою мовою програмування. Ця мова розроблена компанією JetBrains, та працює на Java Virtual Machine. Була представлена світу в 2011 році, та вже у 2016 отримала офіційний статус мови для розробки Android. Kotlin має повну сумісність з Java, що дозволяє проводити інтеграцію без потреб переписувати код.

Java, наразі, є однією з найпопулярніших мов програмування[4] для розробки програмних засобів на Android. Ця мова програмування має гарну інтеграцію аби працювати з базами даних, наділена багатоозяжними функціональними можливостями для розробки. Зазвичай Java обирають через її універсальність, також через постійні оновлення, гарно пропрацьовану документацію[5], велику спільноту, доступність для розробників початкового рівня.

Objective-C – об'єктно-орієнтована мова програмування, котра була розроблена в першій половині 1980-х років, орієнтована для розробки програмних засобів для платформи iOS, і є однією з основних мов програмування для розробки під дану операційну систему. Зазвчиай використовується з фреймворком Cocoa Touch, котрий відповідає за інтегрування з користувальницьким інтерфейсом. На теперешній час популярність цієї мови програмування все сильніше зменшується, і основною стає Swift, завдяки своєму простішому синтаксису та кращій продуктивності.

C# – це високорівнева мова програмування, котра має широкий спектр використання, зокрема і для розробки мобільних застосунків на таких платформах як: iOS; Android; Windows. Дана мова програмування досить

часто використовуэться з фреймворками .NET MAUI та Xamarin. Xamarin забезпечує розробникам доступ до API платформ, а.NET MAUI допомагає в розширенні можливостей Xamarin. Мобільні застосунки розробляються переважно в середовищі Visual Studio, котре дає гарний інструментарій для налагодження коду та його оптимізації.

Python – універсальна мова програмування, котра має дуже простий синтаксис та читебельність коду. Python підтримує різноманітні стилі програмування, такі як: імперативне; функціональне та об'єктно-орієнтоване. Дана мова програмування має велику та активну спільноту, що ьезперечно сприяє доступу до чисельних ресурсів і спрощенню розробки програмних засобів. Для мобільної розробки використовуються фреймворки Kivy та PyQt. Ці фреймворки дають змогу швидко розроблювати прості за структурою програми, проте швидкість обробки даних не така висока порівняно з іншими конкуруючими мовами програмування.

У сфері розробки застосунків для мобільних засобів слід також звернути увагу на інтеграцію систем обробки даних. Бази даних можуть мати різні типи:

- реляційні бази даних (рисунок 1.3). Реляційні бази даних є одними з найпоширеніших й базуються на реляційній моделі даних. У таких базах дані структуровані в табличному вигляді зі зв’язками між ними; запити виконуються за допомогою мови SQL;
- об’єктно-орієнтовані бази даних структурують інформацію у вигляді об’єктів, котрі містять атрибути та методи. Даний тип бази даних має підтримку об’єктно-орієнтованого програмування та підтримує зберігання складні структури даних;
- колоночні БД. Колоночні бази даних організовують зберігання інформації в колоночному вигляді, таким чиному оптимізовуючи запити до стовпців даних, що робить їх ефективними для певних задач.

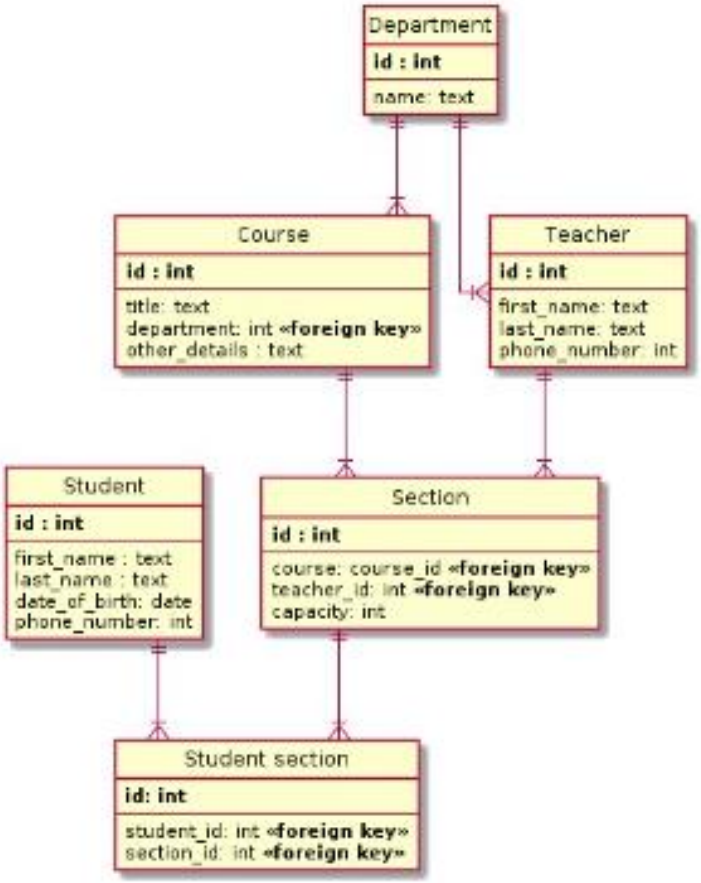

Рисунок 1.3 – Реляційні бази даних

1.4 Постановка задачі

Для створення та коректного функціонування мобільного застосунку потрібно створити декілька екранних форм та об’єктів, котрі матимуть визначений комплект функціональних характеристик, які виконуватимуть такі завдання:

- головний екран– сторінка, що містить список новин та навігаційне меню, котре надає змогу перейти на наступну екранну форму.
- сторінка пошуку – екранна форма з пошуковою стрічкою для пошку новини за певним запитом;
- збережене – екранна форма зі збереженими статтями.

2 ВИБІР ТЕХНОЛОГІЙ

2.1 Використані технології у розробці

Для розробки мобільного програмного засобу було обрано середовище розробки Android Studio (рисунок 2.1). Android Studio – це інтегроване середовище розробки IDE для створення мобільних програмних засобів на платформі Android. Надає користувачам інструментарій для розробки тестування та публікаціїї мобільних застосунків. Має підтнримку таких мовах програмування: Kotlin; C++; Java.

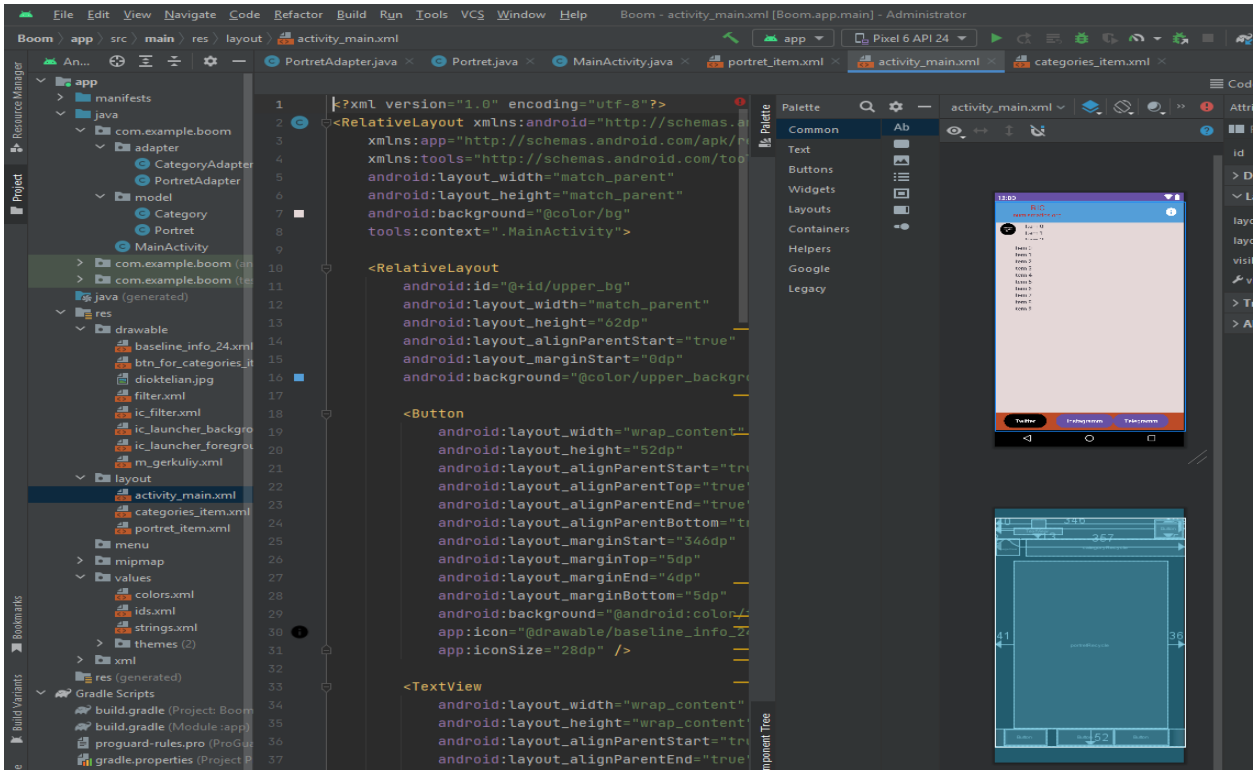

Рисунок 2.1 – Середовище розробки Android Studio

Це середовище розробки пропонує великий спектр можливостей, основний функціонал містить:

- документація та навчальні матеріали. Доступ до документації та

навчальних посібників, котрі спрощують процес розробки починаючим програмістам;

- Android емулятор – вкрай важлива функціональна складова. Допомагає тестувати програму та проводить графічне відображення з елементами логічної частини розроблювального застосунку, перед цим обравши необхідні параметри пристрою та операційної системи.

- вбудовані іструменти для тестування. IDE включає в себе засоби, котрі дозволяють проводити автоматичне тестування з відлагодженням програми, виявлення синтаксичних та логічних помилок, проблем зі зв’язками та залежностями в коді. Також дає можливість проводити аналіз продуктивності коду;

- пристрої разом з Android емулятор є можливість поідключення фізичного пристрою з запуском та тестування розроблюваного застосунку на ньому;

- інтеграція з API. Android Studio містить інтеграцію з Google Service, що допомагає в додаванні різноманітного функціоналу, зокрема підключення валютних платежів;

- інструменти для дизайну. В даному середовищі розробки є розширений редактор макетів з доволі важливою можливістю Drag-and-Drop, котра сильно спрощує розробку грачіного інтерфейсу для користувальницької аудиторії;

- контроль версій. Підтримка Git та інших різноманітних систем для управління версіями коду, що дозволяє зв’язувати різні частину коду, котрі написані на різних версіях мови програмування. Окрім цього наявний доступ до змін версій коду в роботі з іншими розробниками в командному середовищі.

В Android Studio ще є безліч функіональних можливостей для покращення процесу написання коду, його редагування чи оптимізації. Дане середовище розробки приваблює користувачій своює зручністю та постійними оновленнями, котрі сприяють швидкості розроблення мобільних

засточунків на платформі Android. Також розробник має змогу відкрити меню базових шаблонів, котрі заготовлені під застосунки різноманітних направлень та обрати вже заготовлений і необхідних шаблон для подальшої роботи з ним та створенням проєкту (рисунок 2.2).

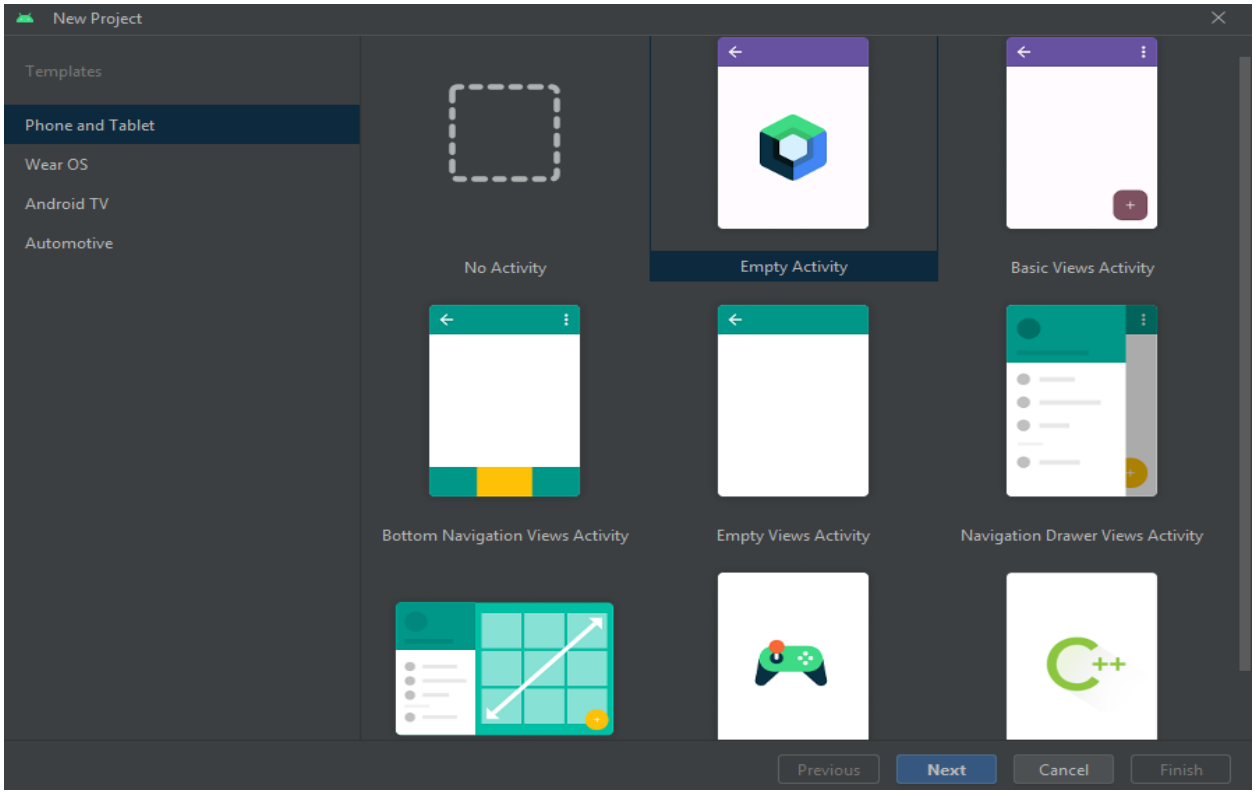

Рисунок 2.2– Середовище розробки Android Studio

Після обрання потрібного шаблону програміст починає працювати з попередньо налаштованою архітектурою проєкту та вносить, за потреби, додаткові корективи.

2.2 Огляд засобів розробки

Для розробки програмного засобу «Новинний портал» була обрана мова програмування Kotlin. На сьогоднішній день Kotlin є однією з найпопулярніших та найбільш підтримуваних мов програмування для

Android розробки. Маючи велику спільноту та постійні оновлення Kotlin є доволі зручною мовою для роботи над реалізацією обраної задачі. В процесі підготовки та програмної реалізації проєкту було використано архітектурну модель MVVM. Для роботи з програмним кодом застосовувалося офіційне середовище розробки Android Studio, а також GitHub для управління версіями проєкту.

2.2.1 Огляд GitHub

GitHub – це платформа, котра забезпечує розробників необхідним інструментарієм для спільної роботи над різноманітними проєктами, дає можливість зберігання кодової бази та управління версіями проєкту з вносенням заміток з поточними змінами. GitHub пропонує наступні можливості:

- соціальна взаємодія. Платформа працює як форум або ж соціальна мережа для розробників, де користувачі можуть ділатися досягненнями, допомагати одне одному, отримувати відгуки та використовувати чужі готові рішення для спрощення розробки власних проєктів;

- репозиторії. GitHub дає створювати репозиторії для зберігання проєкту, та проводити його постійне оновлення з веденням замітоко про поточні правки. Ропозиторії бувають як публічними так і приватними, також вони можуть бути спільними, у випадку якщо розробка проєкту ведеться командою розробників;

- GitHub Actions. Інструмент для автоматизації, котрий проводить підтримку безперервної інтеграції та розгортанняя. Це дозволяє провести автоматизацію процесів тестування, налаштуванню збирання та розгортання проєкту. Таким чином цей інструмент допомагє в автоматизації розробки.

- інструменти для управління проектами. Доступна можливість структурувати свою діяльність і моніторити розвиток проєкту.

- спільна робота. Власники проєктів можуть запрошувати інших

розробників, де користувачі можуть коментувати код, вносити правки та редагувати окремі файли через веб-інтерфейс. Такий інструментарій сильно сприєю поліпшенню розробки та ефективній роботі команди в обміні знаннями.

2.2.2 Архітектурний шаблон MVVM

Архітектура MVVM (Model-View-ViewModel) – це один з найбільш поширеніших архітектурних шаблонів проєктування, першочергово в розробці застосунків, котрі мають графічний інтерфейс. Даний архітектурний шаблон MVVM [6] (рисунок 2.3) часто використовується в роботі з Xamarin для розробки мобільних застосунків на основі мови програмування C#, а також Java чи Kotlin.

Модель є основною частиною архітектури і відповідає за логічну складову та обробку й управлінням даних. Основні функції моделі включають:

- зберігання даних. Модель складається з структурних одиниць, котрі представляють собою інформацію, з котрою працює мобільний застосунок.
- логічна складова. У моделі реалізуються всі алгоритми та правила, які визначають, як інформація може бути створена, змінена або видалена;
- взаємодія з базою даних. Модель може містити код для доступу до сховища даних або зовнішніх сервісів для отримання та зберігання інформаці.

Модель не має мати будь-яких залежностей від представлення. Таким чином це забезпечує її незалежність та можливість для повторного використання в разі необхідності.

Представлення являє собою складову, котра відповідає за інтерфейс, а саме відображення інформаційної складової, котра надходить з ViewModel. Надходжуюча фнформація може бути текстом, зображенням, кнопками або іншими елементами інтерфейсу. Також представлення відповідає за обробку

дій та зв’язування даних, що допомагає автоматизовано оновлювати інтерфейс у разі зміни даних в програмі.

Модель-представлення слугує посередником поміж моделлю і представленням, забезпечуючи обробку даних, отримуючи дані з моделі і перетворює їх у формат, котрий може бути відображеним в представленні, наприклад: агрегація даних; сотрування або фільтрація. Також модель-представлення відповідає за реалізацію команд, таких як: збереження даних; завантаження нової інформації. Окрім цього також може зберігати стан програми між змінами в представленні, що особливо корисно в роботі мобільних застосунків, де потірбно зберігати інформацію під час переходу між екранними формами.

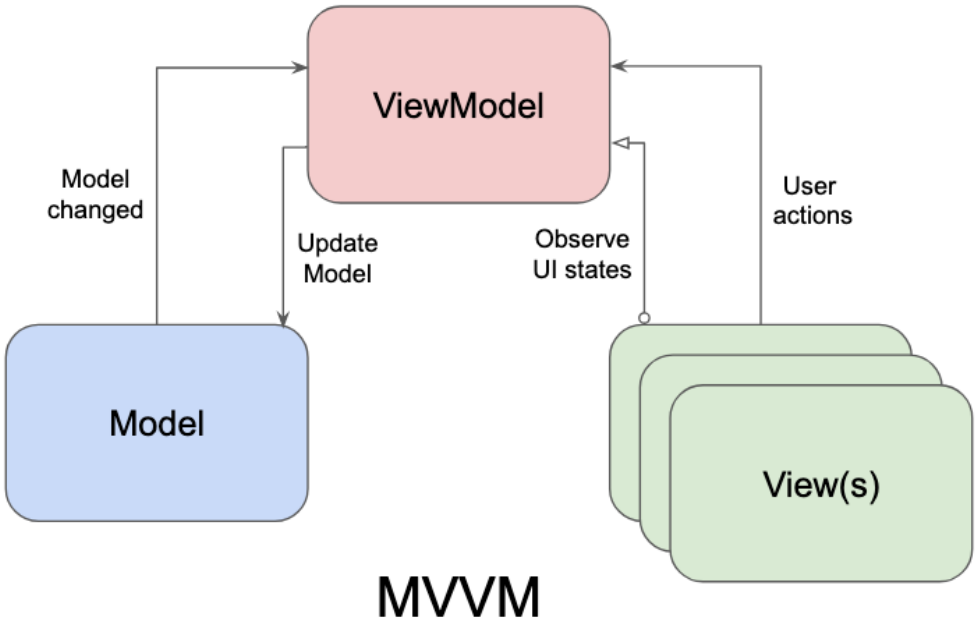

Рисунок 2.3 – Архітектурний шаблон MVVM

MVVM (Model-View-ViewModel) є популярним шаблоном проектування, який часто застосовується в розробці мобільних застосунків, зокрема на платформах Android та iOS, а також у десктопних програмах на

основі WPF або UWP. У контексті новинного додатку модель може представляти новинні статті та їх категорії, включаючи дані про авторів, дати публікацій та зміст статей. ViewModel обробляє запити на отримання новин, здійснює фільтрацію за категоріями та управляє станом інтерфейсу, забезпечуючи зручний доступ до даних. Представлення відповідає за показ списку статей, а також за деталі конкретної статті під час вибору користувачем. Завдяки чіткому розділенню відповідальностей між цими компонентами, MVVM спрощує тестування, масштабування та підтримку додатків, дозволяючи командам працювати над різними частинами коду незалежно. Це підвищує ефективність розробки і забезпечує легкість в інтеграції нових функціональних можливостей у майбутньому.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Загальний опис

Розроблений мобільний застосунок поділяється на дві частини: візуальна та логічна складові. Інтерфейс програми відповідає за зручність користувальницької діяльності у всіх її проявах, тоді як логічна частина забезпечує коректне відображення даних та роботу з ними.

Архітектура застосунку (рисунок 3.1.) була реалізована з використанням шаблону MVVM.

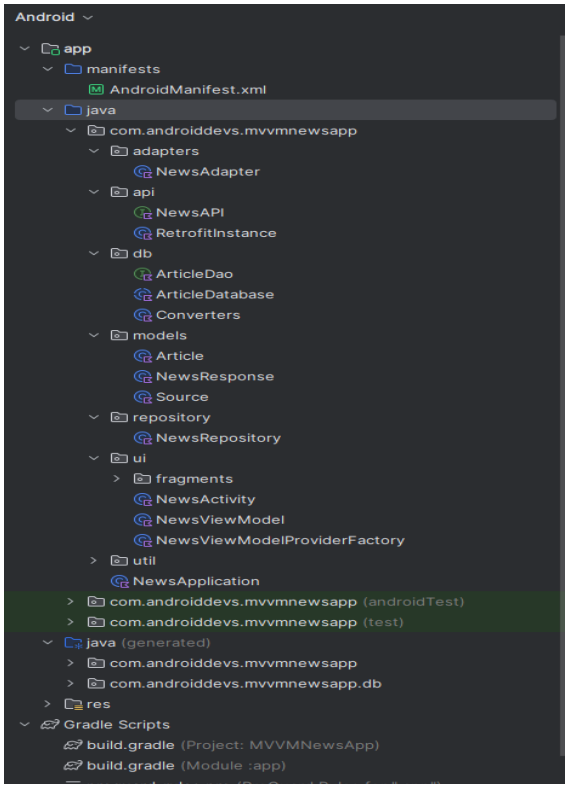

Рисунок 3.1 – Архітектура застосунку

Архітектура MVVM є шаблоном, який забезпечує чітке розділення між графічним інтерфейсом користувача та логічною частиною програми.

Архітектурна складова програмного засобу можу піддаватися змінам,

враховуючи вимоги замовників, окремі ідеї візуальної складової, також переваги архітектурних шаблонів одне над одним та вкладених функцій. Та зазвичай основна компонентна частина у більшость випадків є однаковою, до незмінних компонентів водносяться: модель даних, представлення, контролер або адаптер.

В розробленому мобільному застосунку створено наступні компоненти і розділи: головна сторінка; збережене; пошукова стрічка; картка об’єкту; сторінка об’єкту. На сторінці об’єкту наведено наступні зведення: загаловок; фото; текст статті;.

Мобільний застосунок складається з наступних елементів: головна сторінка; збережене; пошукова сторінка. Після того як користувачем буде запущено програму, то на екрані відобразиться головна сторінка з зберегти якусь статтю з поточної стрічки аби прочитати її пізніше або ж перейти на сторінку пошуку чи збережених новин.

Метою створення програмного засобу слугує агрегація оновлюваних новинних матеріалів та відображення їх для ознайомлення користувачам. В програмі було реалізовано можливість здійснювати пошукові запити задля отримання бажаних новинних матеріалів.

3.2 Інтерфейс програми

Інтерфейс відповідає за графічне представлення програмного коду з відображення отримуваної інформації з баз даних у доступному для перегляду виді. Інтерфейс слугує для взаємодії з користувачем, і відображає йому дані. Зовнішній вигляд інтерфейсу впливає на користування програмою, шляхом спрощення зрозумілості користування певними функціями, а звідси і йде вплив на зручність та задоволення використання продукту користувачами. Зазвичай реалізація інтерфейсу залежить від вимог проекту та побажань замовника.

Головна сторніка демонструє елементи вбудованого конструктуру

середовища Android Studio, завдяки якому створюється макет на якому базується логічна та візуальна складові. До елементів констуктору: кнопки; віджети; картки; зображення; кнопки-зображення. Інтерфейс головної сторінки зображено на рисунку 3.2.

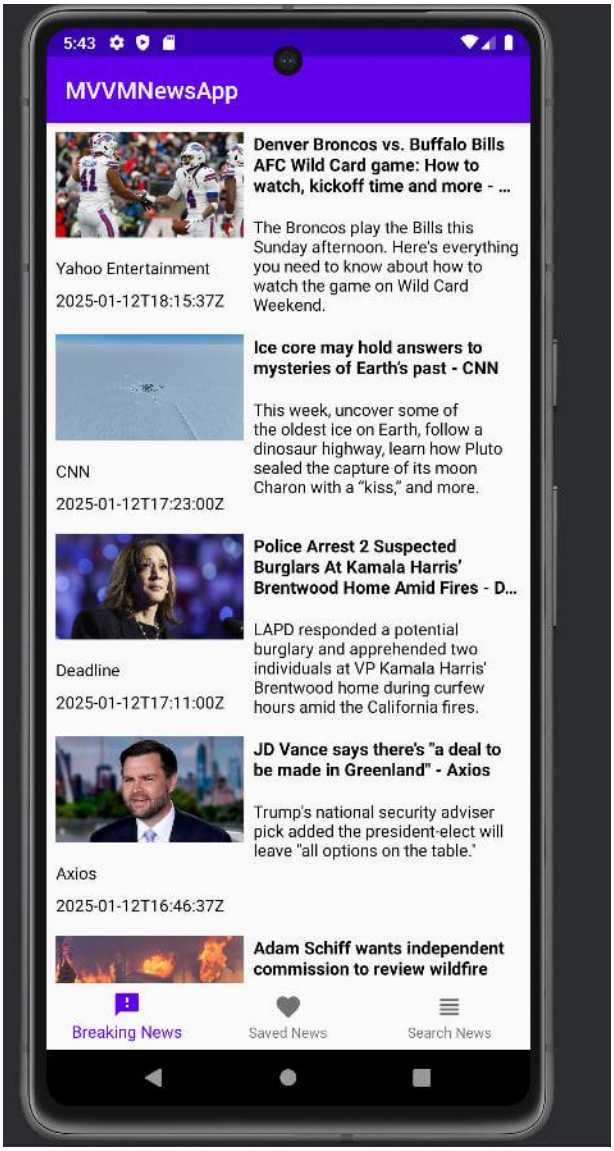

Рисунок 3.2 – Інтерфейс головної сторінки

На головній сторінці застосунку розташована нижня навігаційна панель, яка забезпечує зручний доступ до основних розділів програми. Завдяки інтуїтивно зрозумілому дизайну, користувачі можуть легко знаходити потрібну інформацію та отримувати доступ до всіх можливостей

застосунку без зайвих зусиль. Нижню навігаційну панель зображено на рисунку 3.3.

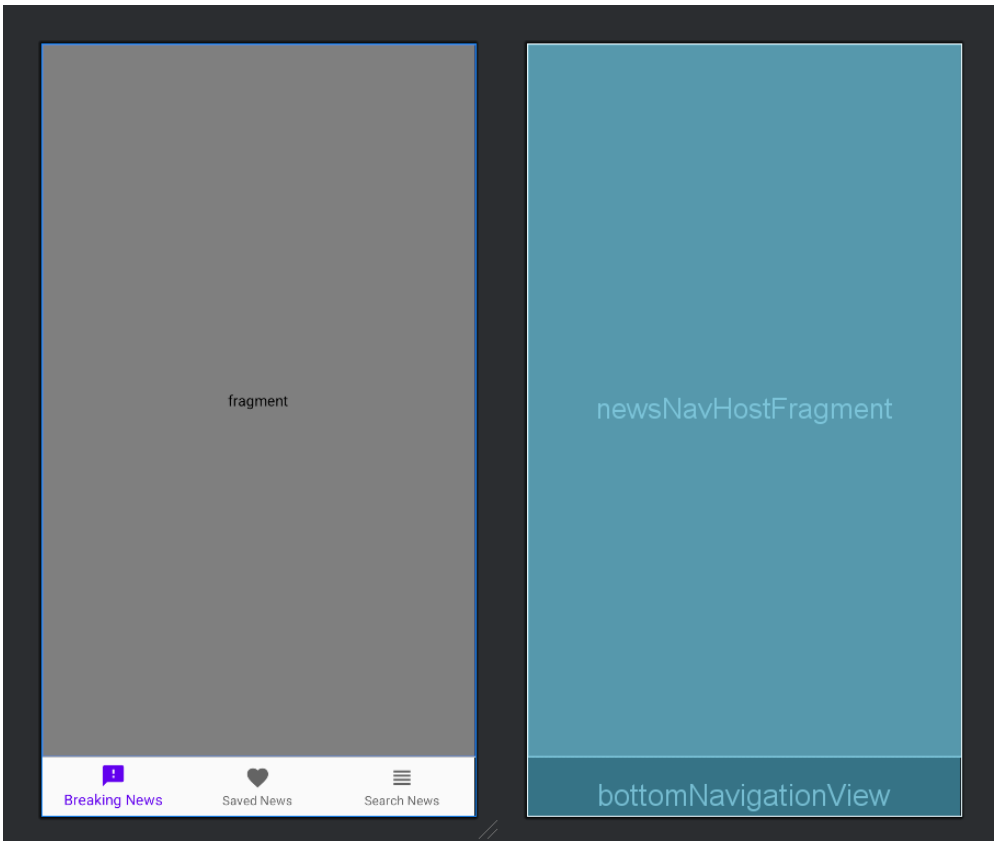

Рисунок 3.3 – Нижня навігаційна панель

На основній частині екранної форми розташоване вікно зі списком, яке відображає новинні статті. Цей список формує візуальне представлення останніх новин, зібраних з різноманітних засобів масової інформації. Кожна стаття в цьому списку містить заголовок, короткий опис та дату публікації, що дозволяє користувачам швидко ознайомлюватися з основними темами. Також передбачено можливість натискання на заголовок статті для переходу до детального перегляду. У цьому режимі користувачі можуть ознайомитися з повним текстом статті, переглянути супутні зображення або відео та дізнатися більше про автора. Режим детального перегляду зображено на рисунку 3.4.

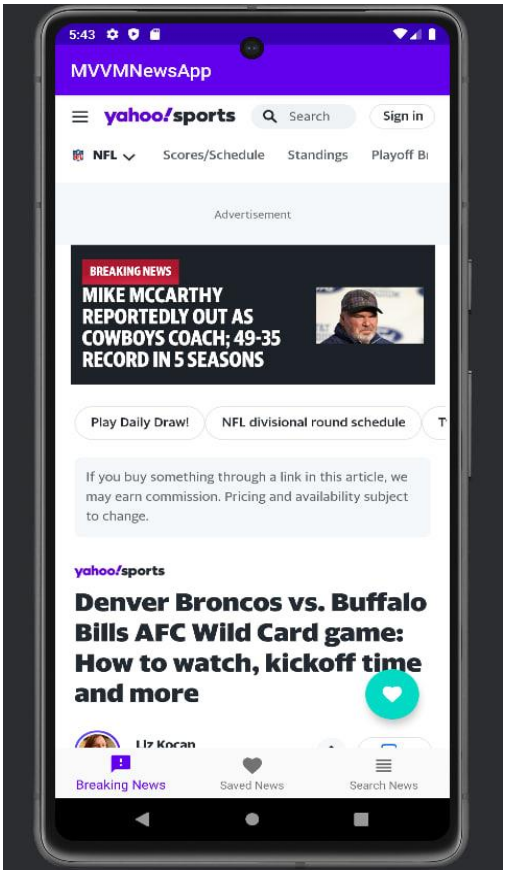

Рисунок 3.4 – Режим детального перегляду

Головна сторінка застосунку створена таким чином, щоб забезпечити максимальний комфорт і зручність для користувачів. Вона дає простоту навігації для перегляду новинних статей з різних джерел, що робить її потужним інструментом для отримання актуальної інформації.

На основній частина екранної форми головної сторінки розташовано список, котрий містить в собі макет картки для ознайомлювального відображення новини. Макет відображення та розмітки картки зображено на рисунку 3.5.

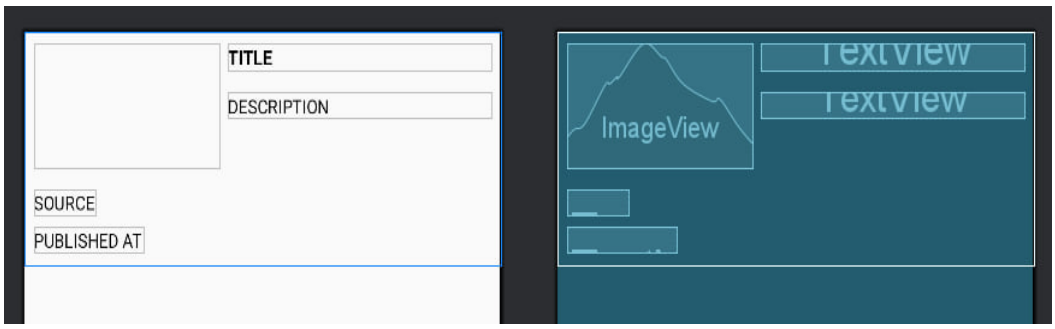

Рисунок 3.5 – Макет відображення та розмітки картки

Картка виконує роль накладного елемента на головному екрані. За кожною карткою закріплені текстові поля з фіксованими та змінними даними, які вставляються на основну сторінку вибраного користувачем об'єкта. Крім того, картка є шаблоном, з якого інформація вставляється в основну сторінку обраної користувачем статті під час переходу на її детальну сторінку.

3.3 Програмна реалізація засобу агрегації новинного контенту з інформаційних електронних джерел

Для взаємодії з програмою з боку користувача необхідно аби був створений інтерфейс з кнопками, саме завдяки кнопкам і відбувається взаємодія користувачів з основним функціоналом програми, а саме її логічною частиною. Кожна кнопка має свій окремий функціонал, котрий в дечому може бути схожим. Розташування кнопок відбувається в контейнорних лейаутах, котрі використовуються для розміщення візуальних елементів програми. В розробленому застосунку наявна нижня навігаційна панель, робота якої забезпечується спеціальним навігаційним контролером. Програмна реалізація відображення інтерфейсу наведена в лістингу 3.1.

Лістинг 3.1 – Програмна реалізація відображення інтерфейсу

```
override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_news)
        val newsRepository =
NewsRepository(ArticleDatabase(this))
        val viewModelProviderFactory =
NewsViewModelProviderFactory(application, newsRepository)
        viewModel = ViewModelProvider(this,
viewModelProviderFactory).get(NewsViewModel::class.java)

bottomNavigationView.setupWithNavController(newsNavHostFragment.
findNavController())}
```

Цей код є частиною методу onCreate класу NewsActivity, де спочатку викликається метод суперкласу для коректного виконання життєвого циклу активності, а потім задається макет інтерфейсу користувача через setContentView(R.layout.activity_news). Далі створюється об'єкт NewsRepository, який відповідає за доступ до даних статей, отримуючи екземпляр бази даних ArticleDatabase. Ініціалізується фабрика для створення ViewModel NewsViewModelProviderFactory, яка передає контекст додатку та репозиторій як параметри. Після цього за допомогою ViewModelProvider створюється екземпляр NewsViewModel, що забезпечує доступ до даних і бізнес-логіки, пов'язаної з новинами. Нарешті, нижня панель навігації BottomNavigationView налаштовується для роботи з контролером навігації NavController, що дозволяє автоматично обробляти переходи між фрагментами при натисканні на елементи меню.

В процесі створення логічної функціональної частина та інтерфейсної складової було використано середовище розробки Android Studio. Структура програмного засобі втілена з адаптерів, фрагментів та моделі, й xml файлів інтерфейсу. Структура застосунку зображена на рисунку 3.6.

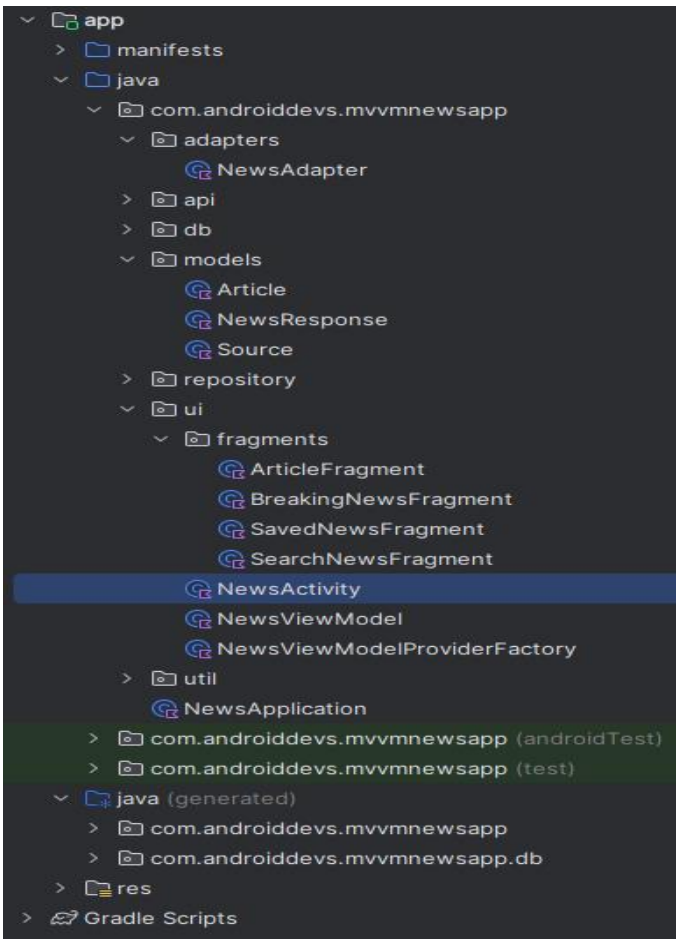

Рисунок 3.6 – Структура застосунку

В структурі розробленого мобільного застосунку містяться вікна; фрагменти; адаптери; моделі; класи; файли зображень; файл рядків; файл палітри кольорів; ресурси застосунку. Ці компоненти взаємодіють один з одним, забезпечуючи інтеграцію та синергію, що сприяє розробці функціональності додатку. Кожен з елементів, таких як елементи інтерфейсу, адаптери, моделі даних та ресурси, виконує свою специфічну роль у загальному процесі. Наприклад, елементи інтерфейсу відповідають за візуальне представлення даних і взаємодію з користувачем, тоді як адаптери забезпечують зв'язок між даними та інтерфейсом, перетворюючи дані в зручний для відображення формат. Моделі даних організовують і структурують інформацію [7], а ресурси, такі як кольори та текстові рядки, сприяють створенню єдиного стилю та локалізації додатку. Завдяки цій

взаємодії компоненти можуть працювати разом, створюючи зручний і функціональний досвід для користувачів.

Інтерфейс головної сторінки розроблений таким чином аьи користувачами було максимально легко сприймати інформації та розуміти як працює застосунок. Через такий підхід було обрано втілити мінімалістичний дизайн. Розроблено основні елементи користування: нижня навігаційна панель, вікно для назви застосунку та додаткової інформації, поле для списку, в в якому будуть відображатися новинні матеріали. Макет головної сторінки зображено на рисунку 3.7.

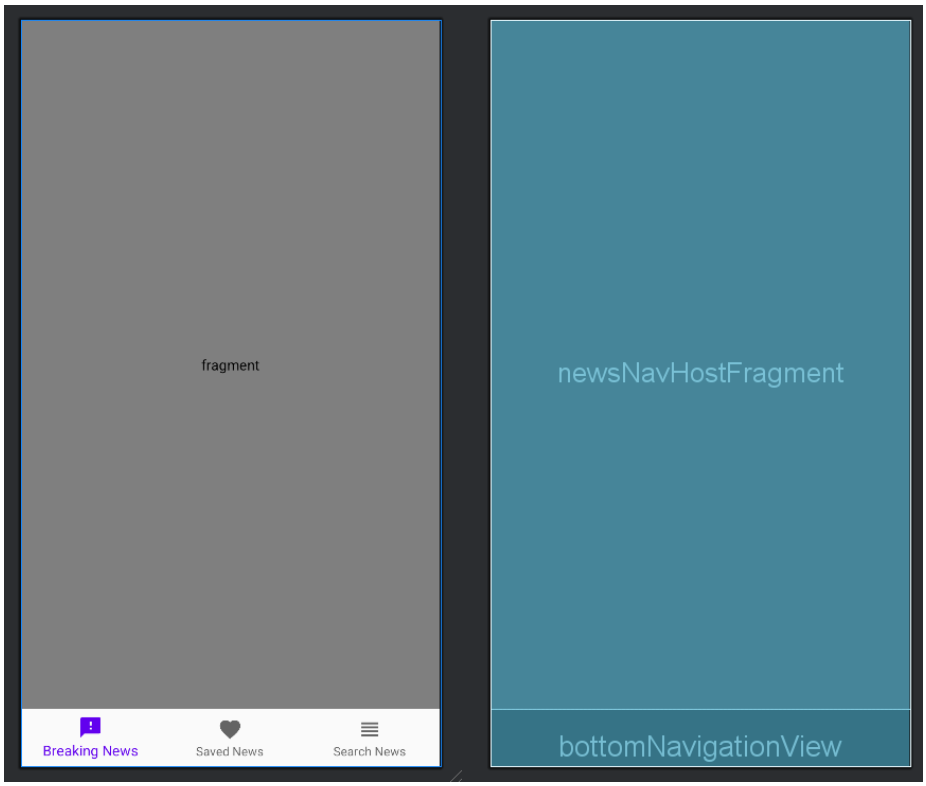

Рисунок 3.7 – Макет головної сторінки

На представленому макеті втілено розмітка для головної сторінки в розробленому програмному засобі на операційній системі Android. Дана розмітка втілює базову концепцію структури інтерфейсу. Інтерфес-макет визначає структуру для активності Android, використовуючи ConstraintLayout як кореневий елемент, що дозволяє гнучке позиціонування дочірніх

елементів на основі обмежень. Усередині ConstraintLayout розміщено FrameLayout з ID flFragment, який слугує контейнером для фрагментів і заповнює простір над BottomNavigationView, висота якого встановлена на 56dp. Всередині FrameLayout знаходиться NavHostFragment, відповідальний за управління навігацією в додатку, з атрибутами, які вказують на граф навігації. BottomNavigationView надає користувачеві можливість перемикатися між різними секціями додатку через меню, визначене в ресурсах. Цей макет забезпечує організовану архітектуру для новинного додатку Android, дозволяючи плавну навігацію та динамічне відображення контенту через фрагменти.

На головній екранній формі також розташована кнопка для збереження новин аби мати змогу швидко віднайти необхідну статтю та прочитати її пізніше. Програмна реалізація збереження та видалення статті наведена в лістингу 3.2.

Лістинг 3.2 – Програмна реалізація збереження та видалення статті

```
// Отримання збережених новин
    fun getSavedNews(): LiveData<List<Article>> {
        return articleDao.getAllSavedArticles() // Метод DAO для
отримання статей}
    // Збереження або оновлення статті
    suspend fun upsert(article: Article) {
        articleDao.insertOrUpdate(article) // Метод DAO для
вставки або оновлення
    }
    // Видалення статті
    suspend fun deleteArticle(article: Article) {
        articleDao.delete(article) // Метод DAO для видалення
статті}
```

Цей код реалізує методи в класі NewsRepository, які дозволяють працювати зі збереженими новинами в базі даних через DAO. Метод getSavedNews() повертає LiveData зі списком статей, отриманих з бази даних, що дозволяє спостерігати за змінами в даних. Метод upsert(article: Article) використовується для додавання або оновлення статті в базі даних. Він

викликає метод insertOrUpdate(article) DAO. Метод deleteArticle(article: Article) [8] видаляє статтю з бази даних, використовуючи метод delete(article) DAO.

Ключову роль в роботі мобільного застосунку відіграє база даних. Вона зберігає отримувану інформацію, управляє нею та організовує. База даних забезпечує доступ до інформації, а також підтримує синхронізацію між пристроями. Також верхній частині екрану розміщено кнопку для фільтрації монет за категоріями. Ствоерння класу бази даних наведено у лістингу 3.3.

Лістинг 3.3 – Ствоерння класу бази даних

```
@Database(
    entities = [Article::class],
    version = 1)
@TypeConverters(Converters::class)
abstract class ArticleDatabase : RoomDatabase() {
    abstract fun getArticleDao(): ArticleDao
    companion object {
        @Volatile
        private var instance: ArticleDatabase? = null
        private val LOCK = Any()
        operator fun invoke(context: Context) = instance ?:
synchronized(LOCK) {
            instance ?: createDatabase(context).also { instance
= it }}
        private fun createDatabase(context: Context) =
            Room.databaseBuilder(
                context.applicationContext,
                ArticleDatabase::class.java,
                "article_db.db"
            ).build()}}
```

Клас ArticleDatabase є абстрактною реалізацією бази даних у додатку Android, створеною за допомогою бібліотеки Room. Він містить анотації, які визначають сутності типу Article і версію бази даних, а також вказує на використання класу Converters для перетворення нестандартних типів даних. Метод getArticleDao() повертає об'єкт ArticleDao[9], що містить методи для роботи з таблицею статей. Реалізовано патерн синглтон для забезпечення єдиного екземпляра бази даних, використовуючи змінну instance, синхронізацію через об'єкт LOCK[10] та оператор invoke, який створює базу

даних за допомогою методу createDatabase(context), що викликає Room.databaseBuilder для ініціалізації бази даних.

Для того аби інформація зберігалася в базі даних необхідно для кожного типу даних визначити формат зберігання. Програмна реалізація визначення формату зберігання даних наведена лістингу 3.4.

Лістинг 3.4 – Програмна реалізація визначення формату зберігання даних

```
data class Article(
    @PrimaryKey(autoGenerate = true)
    var id: Int? = null,
    val author: String?,
    val content: String?,
    val description: String?,
    val publishedAt: String?,
    val source: Source?,
    val title: String?,
    val url: String?,
    val urlToImage: String?
) : Serializable
```

У даному лістингу клас Article є моделлю даних, яка представляє статтю новини у вашому Android-додатку, містячи поля для зберігання важливої інформації, такої як автор, вміст, опис, дата публікації, джерело, заголовок, URL-адреса статті та URL-адреса зображення. Поле id є первинним ключем, яке автоматично генерується базою даних при додаванні нової статті. Клас реалізує інтерфейс Serializable[11], що дозволяє об'єктам цього класу бути серіалізованими та передаватися між компонентами Android через Bundle. Таким чином, клас Article забезпечує структуру для організації та управління даними статей новин у додатку.

Після переходу на головну сторінку (рисунок 3.3) користувачу демонструється вся інформація з бази даних в обробленому вигляді. Програмна реалізація отримання актуальних новин наведена у лістингу 3.5.

Лістинг 3.5 – Програмна реалізація отримання актуальних новин

```
    private fun handleBreakingNewsResponse(response:
```

```
Response<NewsResponse>) : Resource<NewsResponse> {
        if(response.isSuccessful) {
            response.body()?.let { resultResponse ->
                breakingNewsPage++
                if(breakingNewsResponse == null) {
                    breakingNewsResponse = resultResponse
                } else {
                    val oldArticles =
breakingNewsResponse?.articles
                    val newArticles = resultResponse.articles
                    oldArticles?.addAll(newArticles) }
                return Resource.Success(breakingNewsResponse ?:
resultResponse)}}
        return Resource.Error(response.message())}
    private fun handleSearchNewsResponse(response:
Response<NewsResponse>) : Resource<NewsResponse> {
        if(response.isSuccessful) {
            response.body()?.let { resultResponse ->
                if(searchNewsResponse == null || newSearchQuery
!= oldSearchQuery) {searchNewsPage = 1
                    oldSearchQuery = newSearchQuery
                    searchNewsResponse = resultResponse} else {
                    searchNewsPage++
                    val oldArticles =
searchNewsResponse?.articles
                    val newArticles = resultResponse.articles
                    oldArticles?.addAll(newArticles)}
                return Resource.Success(searchNewsResponse ?:
resultResponse)}}
        return Resource.Error(response.message())}
    private suspend fun safeSearchNewsCall(searchQuery: String){
        newSearchQuery = searchQuery
        searchNews.postValue(Resource.Loading())
        try {if(hasInternetConnection()) {
val response = newsRepository.searchNews(searchQuery,
searchNewsPage)
searchNews.postValue(handleSearchNewsResponse(response))} else {
                searchNews.postValue(Resource.Error("No internet
connection"))}} catch(t: Throwable) {when(t) {is IOException ->
searchNews.postValue(Resource.Error("Network Failure"))else ->
searchNews.postValue(Resource.Error("Conversion Error"))}}}
    private suspend fun safeBreakingNewsCall(countryCode:
String) {
        breakingNews.postValue(Resource.Loading())try {
            if(hasInternetConnection()) {
                val response =
newsRepository.getBreakingNews(countryCode, breakingNewsPage)
breakingNews.postValue(handleBreakingNewsResponse(response))}}}}
```

Даний лістинг є частиною класу ViewModel, який відповідає за отримання актуальних новин та їх пошук у мобільному додатку. При

ініціалізації викликається метод getBreakingNews, що завантажує новини для Сполучених Штатів. Методи getBreakingNews і searchNews запускають асинхронні корутини для виконання запитів до API через безпечні виклики safeBreakingNewsCall і safeSearchNewsCall, які обробляють відповіді, перевіряють наявність Інтернету та обробляють можливі помилки, такі як відсутність мережі або проблеми з конвертацією даних. Відповіді на запити обробляються в методах handleBreakingNewsResponse і handleSearchNewsResponse, які оновлюють список статей у відповідності до отриманих даних, а також зберігають інформацію про старі та нові статті для подальшого використання.

Режим детального перегляду статті в мобільному додатку дозволяє користувачам отримувати повну інформацію про новину, включаючи дату та час публікації, повний текст статті, супроводжуючий текст, додаткові посилання, ознайомлювальний матеріал, відео та фото. При натисканні на заголовок, текст або супроводжуюче зображення користувач переходить на екран детального перегляду, де всі ці елементи відображаються у зручному форматі. Для реалізації цього процесу використовується навігаційний компонент, який передає дані про вибрану статтю до нового фрагмента або активності. У детальному фрагменті дані заповнюють відповідні UI-елементи, а також передбачені можливості для взаємодії з користувачем, такі як ділення статтею або збереження її у обрані матеріали. Реалізація режиму детального перегляду у працюючому застосунку зображена на рисунку 3.8.

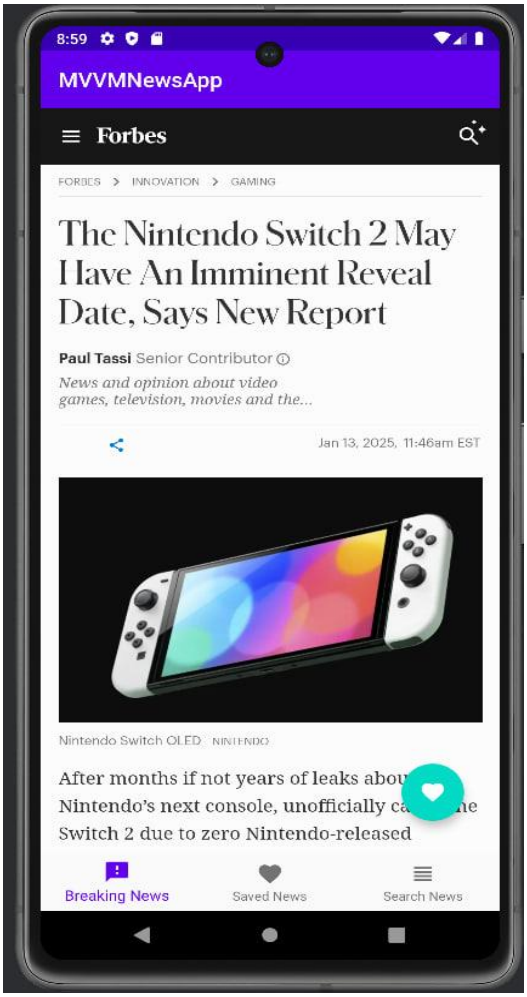

Рисунок 3.8 – Реалізація метода onBindViewHolder у працюючому застосунку

Для відображення повних та детальних даних про статтю було чтворена клас ArticleFragment. Програмна реалізація класу ArticleFragment наведена у лістингу 3.6.

Лістинг 3.6 – Програмна реалізація класу ArticleFragment

```
class ArticleFragment : Fragment(R.layout.fragment_article) {
    lateinit var viewModel: NewsViewModel
    val args: ArticleFragmentArgs by navArgs()
    override fun onViewCreated(view: View, savedInstanceState:
Bundle?) {
        super.onViewCreated(view, savedInstanceState)
        viewModel = (activity as NewsActivity).viewModel
        val article = args.article
        webView.apply {
            webViewClient = WebViewClient()
```

```
            loadUrl(article.url)}
        fab.setOnClickListener {
            viewModel.saveArticle(article)
            Snackbar.make(view, "Article saved successfully",
Snackbar.LENGTH_SHORT).show()}}}
```

Клас ArticleFragment є важливим компонентом мобільного додатку, що реалізує функціональність детального перегляду статей новин. Він відповідає за відображення повної інформації про обрану статтю, використовуючи компонент WebView для завантаження та перегляду вмісту статті за її URL-адресою. При ініціалізації фрагмента, у методі onViewCreated, здійснюється ініціалізація ViewModel, яка отримується з активності NewsActivity, що дозволяє фрагменту взаємодіяти з репозиторієм новин. Стаття, передана через аргументи з попереднього екрану, зберігається в змінній article, що дозволяє зручно отримувати доступ до її властивостей.

У WebView встановлюється клієнт веб-переглядача WebViewClient, що забезпечує належне відображення вмісту статті. Завантажуючи URL-адресу статті, користувач може переглядати повний текст новини безпосередньо в додатку. Додатково, фрагмент надає можливість зберігати статтю у "обрані" через кнопку дії (Floating Action Button). При натисканні на цю кнопку викликається метод saveArticle у ViewModel[12], який зберігає статтю в базі даних, а після успішного виконання операції показується повідомлення Snackbar про те, що стаття була успішно збережена.

Таким чином, ArticleFragment забезпечує інтерактивний та зручний інтерфейс для користувачів, дозволяючи їм не лише переглядати деталі новин, але й зберігати їх для подальшого читання, що підвищує загальний досвід використання додатку. Цей фрагмент є важливим елементом архітектури додатку, оскільки він поєднує в собі функціональність перегляду контенту та управління даними.

Для коректного функціонування мобільного застосунку необхідно мати перевірки на підключення до мережі інтернет, оскільки вся загальна концепція роботи програмного засобу базується на парсингу інформаційних даних. Програмна реалізація перевірки на підключення до мережі інтернет у

лістингу 3.7.

Лістинг 3.7 – Програмна реалізація перевірки на підключення до мережі інтернет

```
is Resource.Error -> {
    hideProgressBar()
    response.message?.let { message ->
        Toast.makeText(activity, "An error occurred: $message",
Toast.LENGTH_LONG).show()
        showErrorMessage(message)}}
```

У даному лістингу код реалізує обробку ситуацій, коли під час отримання новин виникає помилка. Він приховує індикатор завантаження, відображає коротке спливаюче повідомлення для користувача з інформацією про помилку та може також показувати більш детальну інформацію про проблему в інтерфейсі додатку. Це забезпечує кращий досвід користувача, оскільки користувачі отримують зворотний зв'язок про стан запиту та можливі проблеми.

4 ІНСТРУКЦІЯ КОРИСТУВАЧА ПРОГРАМНОГО ЗАСОБУ

Для того аби скористатися функціоналом мобільного застусунку його необхідно запустити. Після запуску користувачу головна сторінка новинного порталу, на якій відображено різні за категоріями новини. Також на головному екрані зображено нижнє навігаційне меню з модливістю перейти до збережених статей або до пошукової форми. Головна сторінка новинного порталу відображена на рисунку 4.1.

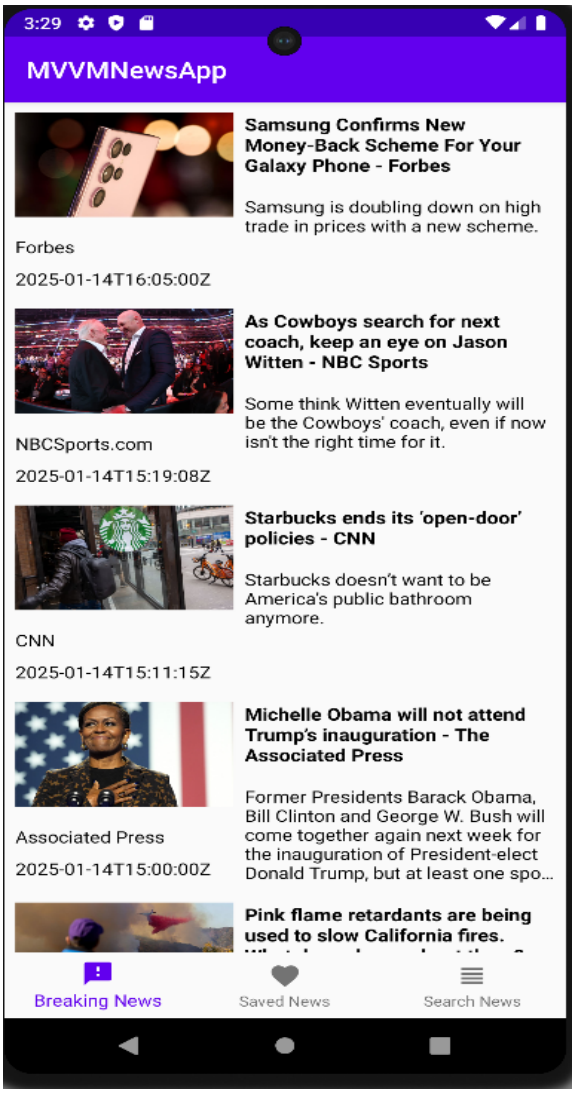

Рисунок 4.1 – Головна сторінка новинного порталу

Після відкриття головної сторінки користувачу надається можливість обрати новинний матеріал та дізнатися інформацію більш детальну інформацію про новину, котра його зацікавила. Шляхом натискання на поле з новиною користувач переходить на екранну форму з повним відображенням новинного матеріалу. Екранна форма з повним відображенням новинного матеріалу зображена на рисунку 4.2.

Рисунок 4.2 – Екранна форма з повним відображенням новинного матеріалу

Зокрема, також є можливость зберегти переглянуту статтю для її

подальшого перегляду завдяки натисканню на спеціальну зберігаючу кнопку. Кнопка збереження зображена на рисунку 4.3.

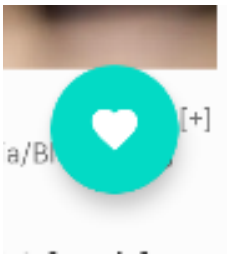

Рисунок 4.3 – Кнопка збереження

Окрім функціональної частини для збереження статей, також присутня можливість видалити непотрібний новинний матеріал зі списку збереженого. Для цього необхідно зробити свайп. Після чого буде відображено спливаюче повідомлення зі спеціальною кнопкою, в котрому буде запропоновано повернути статтю назад, якщо вона була видалена помилково Спливаюче повідомлення зображено на рисунку 4.4.

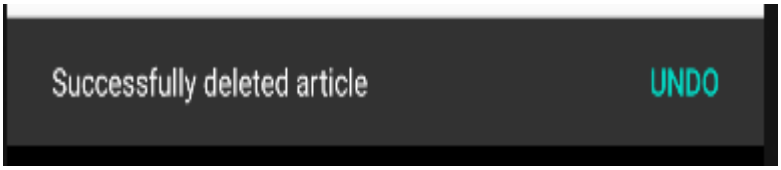

Рисунок 4.4 – Спливаєче повідомлення

Також натиснувши на кнопку пошуку в нижньому меню навігації буде здійснено перехід на чергову екранну форму з пошуковою стрічкою, де шляхом введення запиту з клавіатури можна знайти необхідний новинний матеріал. Сторінка пошуку зображена на рисунку 4.5.

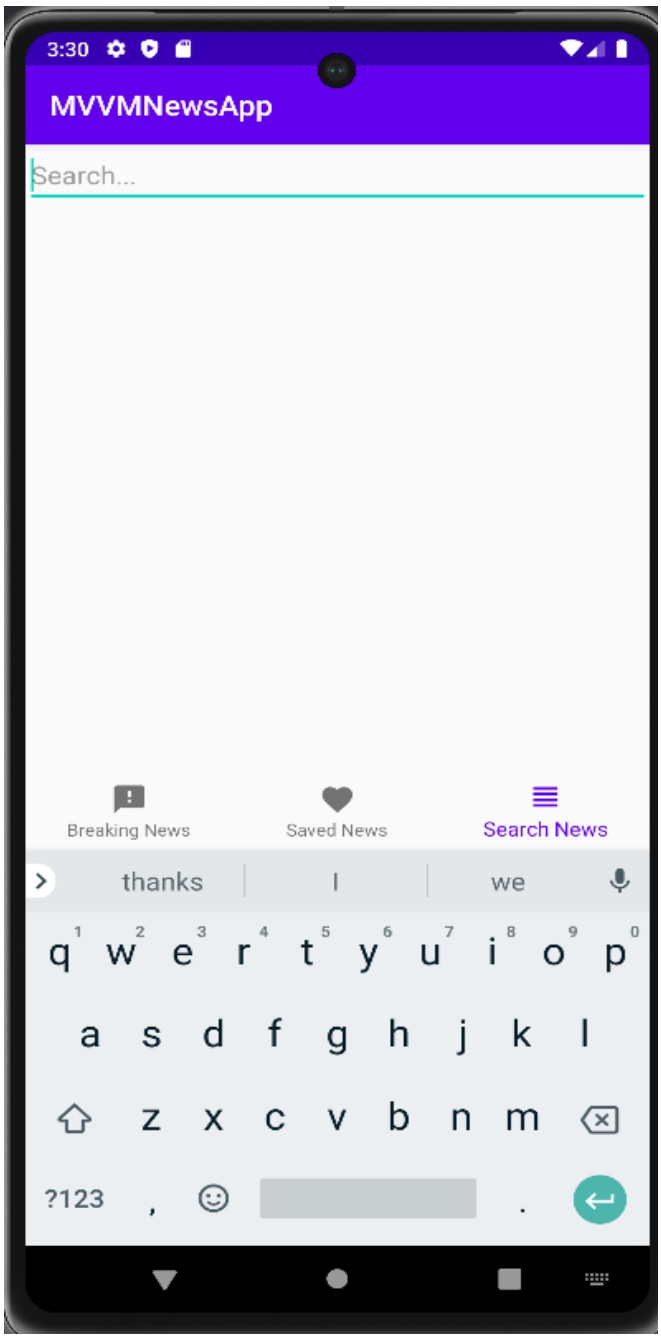

Рисунок 4.5 – Сторінка «Пошуку»

Мобільний застосунок розроблений з простим та мінімалістичним дизайном задля кращого інтуїтивного сприйняття та розуміння усього функціоналу і його принципу роботи.

ВИСНОВКИ

У процесі розробки мобільного застосунку новинного порталу було враховано ключові аспекти, що стосуються сучасних вимог до інформаційних технологій та користувацького досвіду. Для реалізації проекту використовувалося середовище Android Studio з мовою програмування Java, що забезпечило стабільність та продуктивність програми. Було створено інтуїтивно зрозумілий інтерфейс, що дозволяє користувачам легко орієнтуватися в додатку та швидко знаходити необхідні новини. Основна функціональність включає можливість перегляду актуальних новин, пошуку за ключовими словами та збереження статей.

Додаток реалізує архітектуру Model-View-ViewModel (MVVM), що дозволяє відокремити логіку програми від інтерфейсу користувача, забезпечуючи легкість в обслуговуванні та подальшому розвитку. Завдяки цьому підходу вдалося створити програмний продукт, який відповідає вимогам користувачів і передбачає можливість подальшого вдосконалення. Серед можливостей для покращення роботи додатку можна виділити: розширення фільтрів для відображення новин за категоріями, інтеграцію функції збереження статей до списку обраного, а також впровадження системи сповіщень про нові статті в обраних категоріях. Поставлені задачі були виконані успішно з потенціалом для подальшого розвитку, що робить наш новинний портал корисним інструментом для користувачів, які шукають актуальну інформацію в зручному форматі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Новини [Електронний ресурс] – Режим доступу: www/ URL: https://uk.wikipedia.org/wiki/Новини/ – 23.12.2024 г. – Загол. з екрану.

2. Посібник з технологій по програмуванню [Електронний ресурс] – Режим доступу: www/ URL: https://metanit.com/java/tutorial/1.1.php/ – 23.12.2024 г. – Загол. з екрану.

3. Посібник з мови програмування Kotlin [Електронний ресурс] – Режим доступу: https://metanit.com/java/tutorial/ – Дата звернення: 28.04.2023 – Заголовок з екрану.

4. Java Documentation [Електронний ресурс] – Режим доступу: https://docs.oracle.com/en/java/ – Дата звернення: 23.12.2024 – Заголовок з екрану.

5. Java [Електронний ресурс] – Режим доступу: https://www.java.com/en/ – Дата звернення: 23.12.2024 – Заголовок з екрану.

6. Архітектурний паттерн MVVM [Електронний ресурс] – Режим доступу: https://sappro.sapland.ru/material/19180/ – Дата звернення: 23.12.2024 – Заголовок з екрану.

7. Розробка на Kotlin для Android [Електронний ресурс] – Режим доступу: https://avada-media.ua/services/kotlin-android/ – Дата звернення: 23.12.2024 – Заголовок з екрану.

8. Розробка кросплатформного IoT рішення з використанням Kotlin Multiplatform [Електронний ресурс] – Режим доступу: https://pnn.com.ua/ua/portfolio/detail/cross-platform-apps-for-iot-solutions-using-kotlin-multiplatform – Дата звернення: 23.12.2024 – Заголовок з екрану.

9. Kotlin – мова Full Stack розробника [Електронний ресурс] – Режим доступу: https://itvdn.com/ua/webinars/description/kotlin-fullstack – Дата звернення: 23.12.2024 – Заголовок з екрану.

10. Андроїд розробка з використанням мови програмування Kotlin

[Електронний ресурс] – Режим доступу: https://gurt.org.ua/news/keywords/Kotlin/ – Дата звернення: 23.12.2024.

11. Основи Kotlin для Android [Електронний ресурс] – Режим доступу: https://career.softserveinc.com/uk-ua/technology/course/kotlin-fundamentals – Дата звернення: 23.12.2024.

12. Особливості розробки мультиплатформних застосунків [Електронний ресурс] – Режим доступу: http://journals.khnu.km.ua/vestnik/?p=16772 – Дата звернення: 23.12.2024 – Заголовок з екрану.