ДОДАТОК А

Графічний матеріал кваліфікаційної роботи

Харківський національний університет радіоелектроніки

Кафедра ЕОМ

Метод та засіб агрегації новинного контенту з електронних інформаційних джерел

Кваліфікаційна робота

Другий (магістерський) рівень

Автор:
Могильний А.Є.
Ст. гр. СПм-23-1

Керівник:
Єрошенко О. А.

Харків 2025

Актуальність роботи

Актуальність розробки застосунку для агрегації новин на основі API у 2025 році зростає завдяки кільком ключовим трендам. По-перше, впровадження 5G забезпечує швидкий доступ до контенту, що підвищує вимоги до продуктивності новинних платформ. По-друге, тенденція до персоналізації досвіду користувача за допомогою штучного інтелекту дозволяє пропонувати релевантні новини, що підвищує залученість. Зростаючий попит на мобільні додатки для зручного доступу до інформації робить продукт конкурентоспроможним. Врахування цих тенденцій допоможе задовольнити потреби сучасних користувачів у новинній інформації.

Мета та задачі роботи

Метою кваліфікаційної роботи є розробка мобільного застосунку для новинного порталу, який агрегує новини та має основні функції, такі як зручний доступ до інформації з різних джерел, інтуїтивно зрозумілий інтрефейс, пошукова стрічка.

ЗАДАЧІ РОБОТИ

- Аналіз існуючих рішень
- Аналіз технологій для розробки
- Створення методу агрегації новинного контенту з інформаційних електронних джерел
- Практичне використання засобу агрегації новинного контенту з інформаційних електронних джерел

3

Об’єкт, процес та методи дослідження

Об’єкт розробки – процес агрегації новин.

Предмет дослідження – Retrofit для роботи з API, Room для управління локальною базою даних, а також алгоритми машинного навчання для персоналізації контенту. Дослідження також охоплює моделі даних, які представляють новинні статті.

Методи дослідження. В роботі застосовуються методи аналізу збору даних, методи обробки даних, методи об’єктно-орієнтованого програмування.

4


Аналоги
На сучасний момент існують аналоги, котрі частково або повністю виконують поставлені задачі
Google News
CNN News
FOX News
Financial Time
5


НЕДОЛІКИ ІСНУЮЧИХ РІШЕНЬ
Консервативна орієнтація
Наявність великої кількості реклами
Платний доступ в окремих випадках
Окрім вже вказаних неділіків можна ще вказати наступні: висока вартість підписки, що може бути недоступним для багатьох читачів, особливо студентів. Більшість контенту доступна лише підписникам, що обмежує можливості безкоштовних користувачів.
6

НЕДОЛІКИ ІСНУЮЧИХ РІШЕНЬ

- Фокус на бізнез новинах
- Складний інтерфейс з великою кількістю сторінок
- Обмежена інформація через цензурування небежанихполітичних матеріалів

Аналітичні матеріали часто мають складний стиль, що може бути важким для розуміння без відповідної освіти. Деякі читачі також вважають, існує певна політичнв або економічна упередженість, що може впливати на подачу новин.

7

ПЕРЕВАГИ РОЗРОБЛЕНОГО ЗАСТОСУНКА

- Простий користувацький інтерфейс
- Відсутність реклами
- Інтерфейс оптимізований під швидку орієнтацію у функціях

8

ПЕРЕВАГИ РОЗРОБЛЕНОГО ЗАСТОСУНКА

- Відсутність політичної цензури
- Одночасна доступність до кількох джерел інформації
- Присутність допоміжного функціоналу

Розроблений програмний продукт виконаний у мінімалістичному дизайнні з доволі простим інтерфейсом для кращої орієнтації користувача у застосунку. Окрім цього повністю відсутня реклама.

9

Аналіз збору даних

Збір даних відбувається завдяки News API. News API надає доступ до актуальних новинних статей з понад 150,000 джерел по всьому світу, що дозволяє швидко отримувати інформацію за допомогою простих HTTP-запитів. Дані передаються у форматі JSON, що забезпечує легкість обробки та інтеграції з іншими компонентами застосунку.

Інтеграція News API також дозволяє зменшити час на ручний збір даних, автоматизуючи процес отримання новин і знижуючи витрати на обробку інформації. Це робить застосунок більш конкурентоспроможним і здатним швидко реагувати на зміни в інформаційному середовищі. Загалом, використання News API підвищує ефективність застосунку завдяки доступу до актуальної інформації у зручному форматі.

10

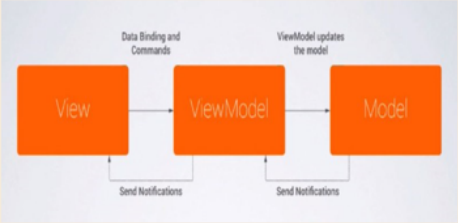
СТРУКТУРА ІНТЕРФЕЙСА

11

Структура розробленої системи

Модель (Model): Відповідає за управління даними додатку, збереження, отримання та маніпуляцію даними.

Вигляд (View): Відображає дані користувачу та обробляє взаємодію з ними. Отримує оновлення від моделі і відображає їх користувачу. Вигляд не має прямого доступу до моделі.



Модель представлення (ViewModel): Виступає як посередник між моделлю та виглядом. Він отримує оновлення від моделі і передає їх вигляду, а також отримує взаємодію користувача з виглядом і передає її моделі для обробки.

12

Взаємодія фрагментів з *ViewMidel* та керування компонентами інтерфейсу

```
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, text: "Article saved successfully", Snackbar.LENGTH_SHORT).show()
        }
    }
}
```


Взаємодія фрагментів з *ViewMidel* та керування компонентами інтерфейсу

У цьому лістингу ViewModel ініціалізується шляхом приведення активності до NewsActivity, що дозволяє отримувати доступ до спільних даних і функцій. Фрагмент отримує деталі статті через navArgs, що забезпечує безпечну передачу аргументів між фрагментами. Для відображення вмісту статті використовується WebView, який завантажує URL статті, а також налаштовано кнопку дії (FAB) для збереження статті при натисканні, з показом Snackbar для зворотного зв'язку користувача про успішне збереження. Ця структура демонструє, як фрагменти можуть ефективно взаємодіяти з ViewModel та керувати компонентами інтерфейсу в модульному вигляді в Android-додатку.

Налаштування відображення категорій та монет за обраним імператором.

```
class RetrofitInstance {
    companion object {

        private val retrofit by lazy {
            val logging = HttpLoggingInterceptor()
            logging.setLevel(HttpLoggingInterceptor.Level.BODY)
            val client = OkHttpClient.Builder()
                .addInterceptor(logging)
                .build()
            Retrofit.Builder()
                .baseUrl(BASE_URL)
                .addConverterFactory(GsonConverterFactory.create())
                .client(client)
                .build()
        }

        val api by lazy {
            retrofit.create(NewsAPI::class.java)
        }
    }
}
```

Клас RetrofitInstance реалізує синглтон для створення єдиного екземпляра Retrofit, що забезпечує централізований доступ до API новин. Використовується лінива ініціалізація через by lazy, що дозволяє створювати об'єкт retrofit лише при першому зверненні, підвищуючи ефективність. Для відстеження HTTP-запитів і відповідей застосовується HttpLoggingInterceptor з рівнем логування BODY, що надає детальну інформацію про запити. Створюється клієнт OkHttp з інтерсептором, а також налаштовується Retrofit через Retrofit.Builder(), де вказується базовий URL (BASE_URL), конвертер JSON (Gson) та клієнт OkHttp. Змінна api ініціалізується і створює реалізацію інтерфейсу NewsAPI, що містить методи для виконання запитів до API.

Автор:
Могильний А. Є.
E-Mail:
andrii.mohylnyi@nure.ua

ВИСНОВКИ

- Розроблено зручний користувацький інтерфейс.
- Розроблено логічну частину застосунку.
- Розроблене програмне рішеня має інтуїтивно зрозумілий інтерфейс та виконує усі необхідні функції.
- Проведено функціональне тестування на кількох пристроях.