

**BỘ NÔNG NGHIỆP VÀ MÔI TRƯỜNG**  
**TRƯỜNG ĐẠI HỌC THỦY LỢI**



**BÀI TẬP LỚN**

**PHÁT TRIỂN ỨNG DỤNG CHO THIẾT BỊ DI ĐỘNG**  
**ĐỀ TÀI: ỨNG DỤNG NEWS APP**

**Giáo viên hướng dẫn: ThS. Kiều Tuấn Dũng**

**Sinh viên thực hiện:**

<b>STT</b>	<b>Mã sinh viên</b>	<b>Họ và tên</b>	<b>Lớp</b>
1	2251061858	Nguyễn Ngọc Phong	64CNTT2
2	2251061766	Trần Quý Hai	64CNTT2

**Hà Nội, năm 2025**

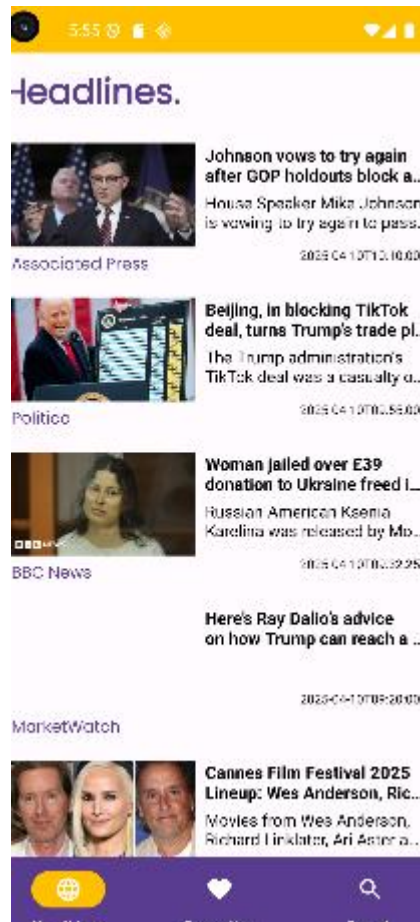
## LỜI NÓI ĐẦU

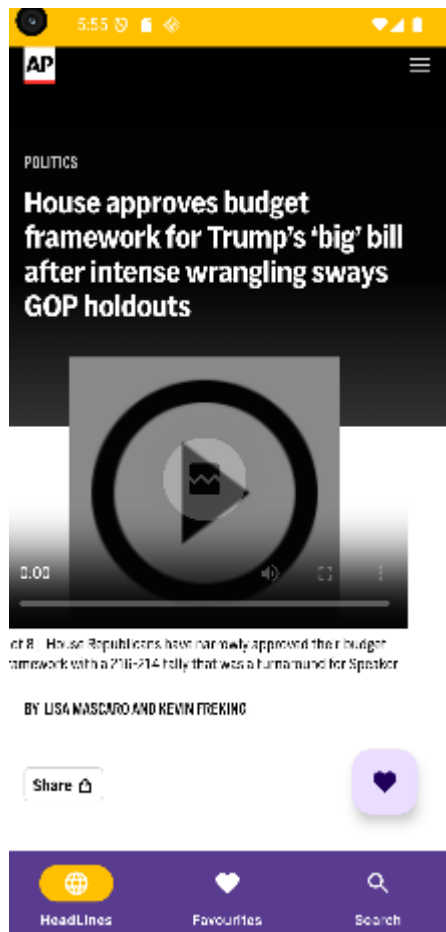
Ngày nay, với sự phát triển mạnh mẽ của công nghệ, nhu cầu đọc tin tức trên các thiết bị di động ngày càng tăng cao. Việc xây dựng một ứng dụng đọc tin tức không chỉ giúp người dùng cập nhật thông tin nhanh chóng, mà còn là cơ hội thực hành áp dụng các kiến thức về phát triển ứng dụng Android. Do đó, nhóm chúng em đã thực hiện đề tài "News App" với mục tiêu xây dựng một ứng dụng đọc tin tức đơn giản, hiệu quả và dễ sử dụng.

# MỤC LỤC

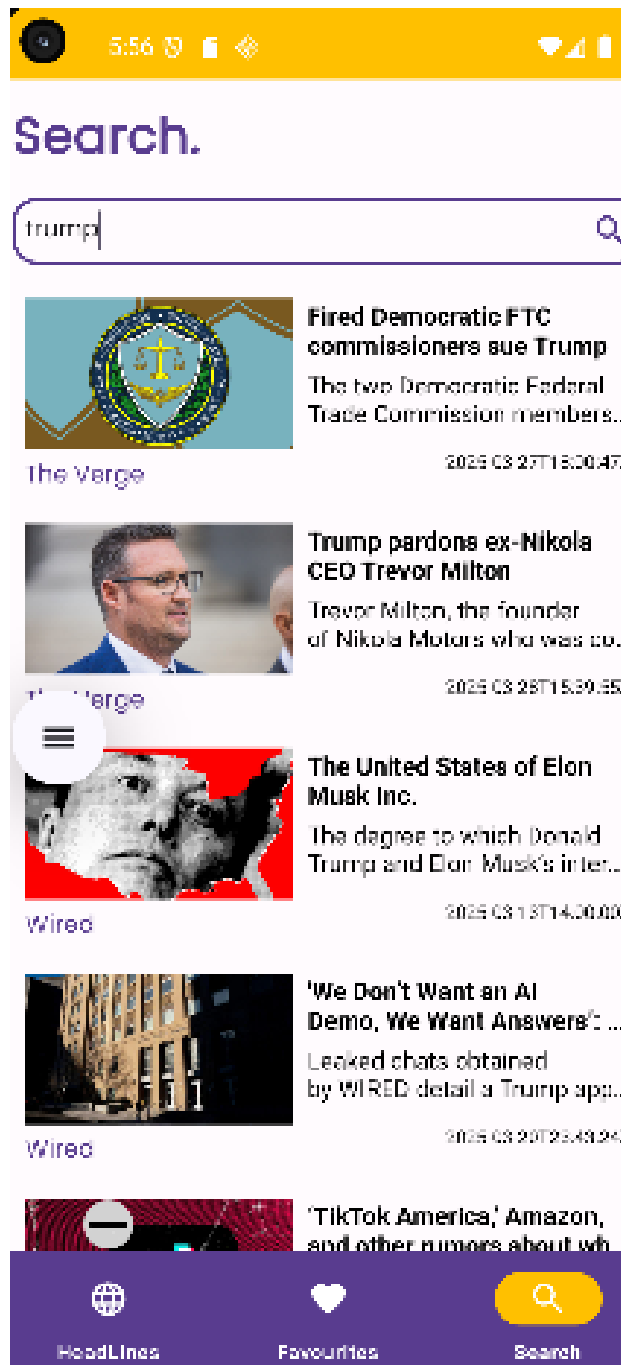
1. Tổng quan về đề tài
2. Kiến trúc và công nghệ
3. Xây dựng ứng dụng
4. Kết luận
5. Tài liệu tham khảo

## DANH MỤC HÌNH ẢNH









## DANH MỤC BẢNG BIỂU

## DANH MỤC CÁC TỪ VIẾT TẮT

STT	TỪ VIẾT TẮT	VIẾT ĐẦY ĐỦ
1	<b>RWD</b>	Responsive Web Design
2	<b>MVVM</b>	Model - View - ViewModel
3	<b>API</b>	Application Programming Interface
4	<b>UI</b>	User Interface
5	<b>UX</b>	User Experience



## Chương 1. TỔNG QUAN VỀ ĐỀ TÀI

### 1.1. Giới thiệu về đề tài

Giới thiệu về đề tài Ứng dụng News App là ứng dụng Android sử dụng kiến trúc MVVM để hiển thị danh sách tin tức theo danh mục, tìm kiếm, lưu và xem chi tiết bài viết. Ứng dụng tương tác với News API để lấy dữ liệu bài viết.

### 1.2. Mục tiêu của đề tài

- Xây dựng ứng dụng đọc tin tức trên nền tảng Android.
- Hiển thị danh sách bài viết, hỗ trợ tìm kiếm và lưu bài viết yêu thích.
- Áp dụng các công nghệ hiện đại như Retrofit, Room, Navigation Component.

### 1.3. Phạm vi của đề tài

- Phát triển ứng dụng News App với các chức năng cơ bản: đọc, tìm kiếm, lưu bài viết.
- Giao diện đơn giản, trực quan.
- Sử dụng nguồn tin từ NewsAPI.

### 1.4 Phân chia nhiệm vụ

<<Bảng phân chia nhiệm vụ>>

STT	Mục công việc	Gợi ý mô tả
1	Thiết kế giao diện	DùngFigma/Sketch tạo mockup
2	Lập trình View(UI)	Dùng XML và RecyclerView
3	Lập trình ViewModel và xử lý logic	Dùng LiveData,ViewModel
4	Kết nối API	Sử dụng Retrofit,Json
5	Thiết kế cơ sở dữ liệu cục bộ với room	Tạo Entity,DAO,Database
6	Hiển thị bài viết bằng WebView	Tạo Fragment,WebView

7	Tối ưu hiệu năng,xử lí lỗi	Phân trang,debounce,lỗi mạng
8	Tài liệu và Báo cáo	Soạn thảo tài liệu,Viết báo cáo

Trần Quý Hai : Phần 1 và làm Đăng nhập,Đăng ký .

Nguyễn Ngọc Phong :Làm Phần 2-8

## Chương 2. KIẾN TRÚC VÀ CÔNG NGHỆ

### 2.1. Kiến trúc hệ thống Ứng dụng sử dụng kiến trúc MVVM:

- Model: Định nghĩa các đối tượng Article, Source.
- ViewModel: Tương tác giữa Repository và View.
- View: Hiển thị dữ liệu bằng RecyclerView và Fragment.

### 2.2. Giới thiệu về Công nghệ phát triển

- Kotlin: ngôn ngữ chính để phát triển ứng dụng Android.
- Retrofit: thư viện để gọi API.
- Room: thư viện để lưu dữ liệu cục bộ.
- Navigation Component: quản lý chuyển đổi giữa các màn hình.
- ViewModel + LiveData: quản lý vòng đời và dữ liệu hiển thị.
- RecyclerView: hiển thị danh sách các bài viết.

## Chương 3. XÂY DỰNG ỨNG DỤNG

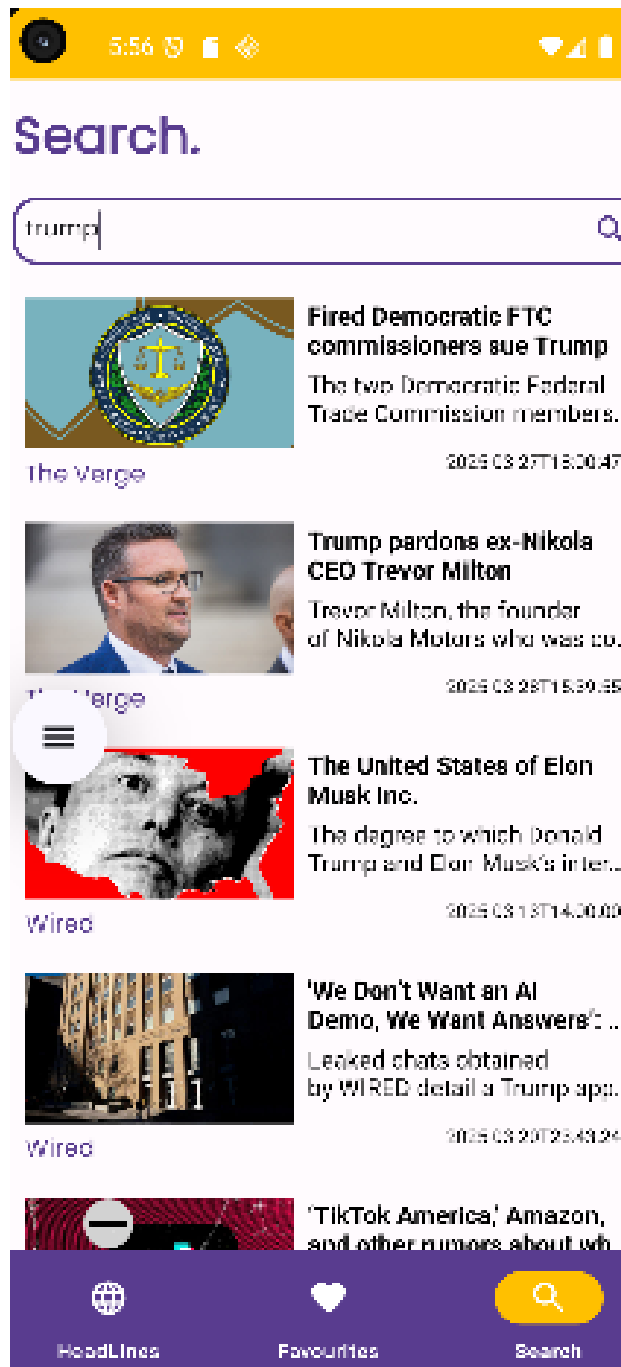
### 3.1. Thiết kế Figma

<<Link Github và ảnh kết xuất Figma các màn hình>>


- Link git ; <https://github.com/phong41302/News-App>
- Link Figma  
<https://www.figma.com/design/myyv7D1BGbeAiRoey6eLAy/News-App?node-id=0-1&p=f&t=d52qzOjm57YQhgEs-0>







### 3.2. Thiết kế CSDL

articles	
id 	int
title	varchar
description	varchar
url	varchar
urlToImage	varchar
publishedAt	varchar
source	varchar

- **Bảng ArticleEntity** lưu trữ thông tin các bài viết mà người dùng đã lưu vào danh sách yêu thích.
  - Mỗi dòng trong bảng tương ứng với một bài báo.
  - Kiểu dữ liệu String được sử dụng cho các trường văn bản, Int cho khoá chính id.
- **Mối quan hệ và đặc điểm:**
- Đây là **cơ sở dữ liệu đơn bảng**, vì ứng dụng hiện tại chỉ lưu trữ danh sách bài viết yêu thích cục bộ, không có quan hệ phức tạp giữa nhiều bảng.
  - **ArticleEntity** là **entity class** được annotate bởi Entity, sử dụng làm đối tượng để ánh xạ dữ liệu giữa Room và SQLite.

Code định nghĩa bảng **ArticleEntity**:

```
@Entity(tableName = "articles")
data class ArticleEntity(
    @PrimaryKey(autoGenerate = true)
    val id: Int = 0,
    val title: String,
    val description: String,
    val url: String,
    val urlToImage: String,
    val publishedAt: String,
    val source: String
)
```

- **Dao tương ứng:**

```

@Dao
interface ArticleDao {
    @Insert(onConflict = OnConflictStrategy.REPLACE)
    suspend fun upsert(article: ArticleEntity)

    @Delete
    suspend fun delete(article: ArticleEntity)

    @Query("SELECT * FROM articles")
    fun getAllArticles(): LiveData<List<ArticleEntity>>
}

```

Database

```

@Database(entities = [ArticleEntity::class], version = 1)
abstract class ArticleDatabase : RoomDatabase() {
    abstract fun getArticleDao(): ArticleDao
}

```

### 3.3. Giao diện ứng dụng

#### 3.3.1. Màn hình HeadlinesFragment

- RecyclerView hiển thị danh sách tin tức.
- Hiện loading khi tải dữ liệu.
- Hiện lỗi nếu có lỗi mạng hoặc API.

#### 3.3.2. Màn hình SearchFragment

- EditText cho phép nhập từ khóa tìm kiếm.
- RecyclerView hiển thị kết quả.
- Debounce input tránh spam API.
- Xử lý lỗi và hiển thị ProgressBar khi tìm kiếm.

#### 3.3.3. Màn hình FavouritesFragment

- RecyclerView hiển thị danh sách bài viết đã lưu.
- Cho phép xóa bài viết khỏi danh sách yêu thích.

#### 3.3.4. Màn hình ArticleDetailFragment

- WebView hiển thị nội dung chi tiết bài viết.
- FloatingActionButton để lưu bài viết vào danh sách yêu thích.

### 3.4. Code minh họa các chức năng cốt lõi

Gọi API bằng Retrofit:

```
interface NewsApi {
    @GET("v2/top-headlines")
    suspend fun getTopHeadlines(
        @Query("country") country: String = "us",
        @Query("page") page: Int = 1,
        @Query("pageSize") pageSize: Int = 20,
        @Query("apiKey") apiKey: String = API_KEY
    ): Response<NewsResponse>
}
```

ViewModel xử lý dữ liệu:

```
class HeadlinesViewModel(private val repository: NewsRepository) : ViewModel() {
    val newsLiveData = MutableLiveData<Resource<List<Article>>>()

    fun getHeadlines(page: Int = 1) {
        viewModelScope.launch {
            newsLiveData.postValue(Resource.Loading())
            val response = repository.getTopHeadlines(page)
            newsLiveData.postValue(response)
        }
    }
}
```

Fragment hiển thị dữ liệu:

```
viewModel.newsLiveData.observe(viewLifecycleOwner) { response ->
    when (response) {
        is Resource.Success -> adapter.submitList(response.data)
        is Resource.Error -> showError()
        is Resource.Loading -> showLoading()
    }
}
```



# KẾT LUẬN

## 1. Kết quả đạt được

- Hoàn thành ứng dụng đọc tin tức đơn giản.
- Áp dụng thành công kiến trúc MVVM, Retrofit, Room.
- Giao diện đẹp, dễ sử dụng.

## 2. Nhược điểm

- Chưa có chức năng phân loại theo chuyên mục.
- Giao diện chưa tùy biến theo chế độ Dark/Light.

## 3. Hướng phát triển

- Thêm chức năng lọc tin theo chuyên mục.
- Hỗ trợ đa ngôn ngữ.
- Tùy biến giao diện nâng cao.

## TÀI LIỆU THAM KHẢO

- NewsAPI. *Developer Documentation*. Truy cập tại: <https://newsapi.org/docs>
- AndroidDevelopers. *ViewModelOverview*.  
<https://developer.android.com/topic/libraries/architecture/viewmodel>
- AndroidDevelopers. *LiveData*.  
<https://developer.android.com/topic/libraries/architecture/livedata>
- AndroidDevelopers. *RoomPersistenceLibrary*.  
<https://developer.android.com/jetpack/androidx/releases/room>
- Google. *Navigation Component*. Truy cập tại:  
<https://developer.android.com/guide/navigation>
- Kotlin Documentation. *Coroutines Guide*. Truy cập tại:  
<https://kotlinlang.org/docs/coroutines-overview.html>
- Square Inc. *Retrofit: A type-safe HTTP client for Android and Java*. Truy cập tại: <https://square.github.io/retrofit/>
- AndroidDevelopers. *RecyclerView*.  
<https://developer.android.com/guide/topics/ui/layout/recyclerview>
- Mark L. Murphy. *The Busy Coder's Guide to Android Development*. CommonsWare, 2020.
- Android Weekly. *Android Architecture - MVVM Pattern*. Truy cập tại:  
<https://androidweekly.net>