

CSE535: Mobile Computing

Final Project

Ananya Garg -2022068

Ishika Gupta - 2022222

Swarnima Prasad - 2022525



INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI



TABS – A News Reading App



Introduction

Tabs is a **accessible** news application designed to provide users with access to news articles from various **genres**, fetched from online sources using hyperlinks or **APIs** through engaging and **interactive** user interface .



Users

Tabs is intended for a diverse audience, including general readers seeking convenient access to news, professionals staying informed, and students researching current events.



MVVM Architecture



Model - View - ViewModel Architecture

1. Model (Data Layer)

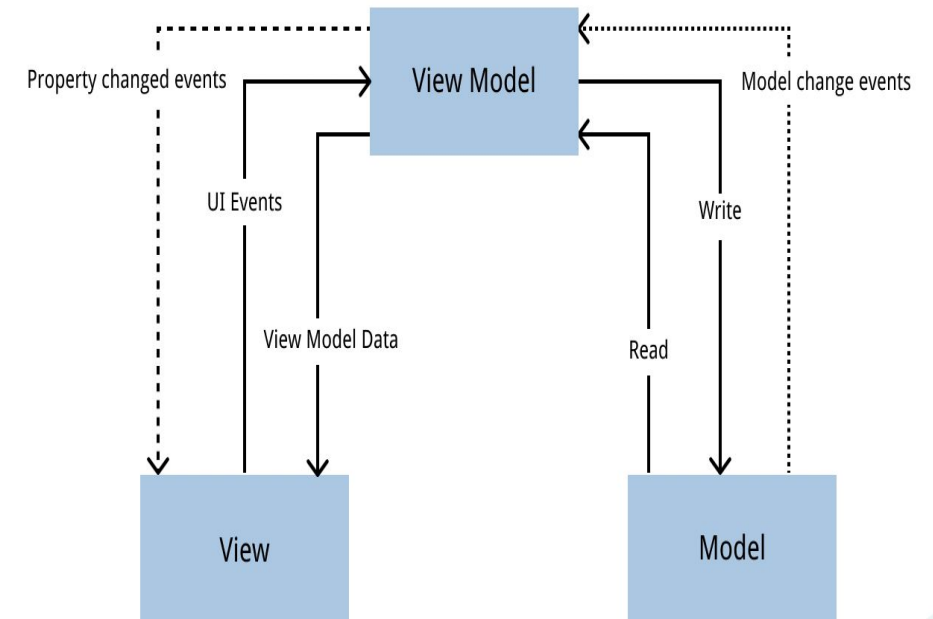
- Manages data (API, DB, cache) & handles errors.
- Uses **Retrofit** (API), **Room/PostgreSQL** (local storage), **WorkManager** (background tasks).

2. View (UI Layer)

- Displays data & handles user interactions.
- Uses **Activities** (screens), **RecyclerView** (news list), **Navigation Component** (transitions).

3. ViewModel (Logic Layer)

- Connects View & Model, manages UI logic, survives config changes.
- Uses **LiveData** (UI updates), **Coroutines** (async tasks).



Basic Functionalities



News Discovery & Personalization

- **Smart Search:** Quickly find relevant articles using keyword-based search.
- **Category Filters:** Tailor your news feed by selecting preferred topics (e.g., Sports, Tech, Business).
- **Saved Articles:** Bookmark articles to read at your convenience.

Interactive News Experience

- **News Cards:** Browse summarized headlines; tap to explore full articles seamlessly.
- **Engage & Save:** Instantly save articles that matter to you with a single tap.



Additional Functionalities



Accessibility

Talkback: Read aloud news content.

Speech Recognition (Sensor-Based Feature)

Voice Search: Users can search for news content using voice commands.

Energy Efficiency

Theme Toggle (Light/Dark Mode): Enables users to switch between light and dark themes for comfort and energy efficiency.

Accessibility Feature – TalkBack

- **Initialize TTS:** When the screen loads, it starts setting up the Text-to-Speech engine.
- **Wait for Ready:** The app waits for the engine to signal it's ready (via onInit). The "Speak" button is disabled until then.
- **Enable Button:** Once ready, the "Speak" button (fabSpeak) is enabled.
- **User Taps Speak:** The user presses the enabled "Speak" button.
- **Read Content:** The app grabs the text from the Title, Source, and Description fields on the screen.
- **Speak:** It sends this combined text to the TTS engine to be read aloud.
- **Clean Up:** When the screen closes, the app shuts down the TTS engine to save resources.

Sensor-Based Feature - Voice Search

- **User Action:** User taps the microphone icon in the toolbar (*R.id.action_voice_search*).
- **Permission & Activation:** *checkAndStartVoiceRecognition()* handles permission; if granted, *startVoiceRecognition()* starts listening via *SpeechRecognizer*.
- **Speech to Text:** On speech completion, *onResults()* receives transcribed text.
- **UI Update & Search:** Text is inserted into *searchView* using *setQuery()*, and *newFilterItems()* filters articles based on the recognized query.

Energy Efficiency Feature - Theme Toggle (Light/Dark Mode)

User Action: Taps the theme switch (in Toolbar or Saved News Fragment).

Trigger: `setOnCheckedChangeListener` detects the change.

Logic: Determines the selected theme mode (Light/Dark), saves it in `SharedPreferences`, applies it using `AppCompatActivity.setDefaultNightMode()`, and calls `recreate()` to refresh the UI.

Result: The UI updates to the selected theme and the preference is retained for future launches.

Live Demo

Experience the App in Action

