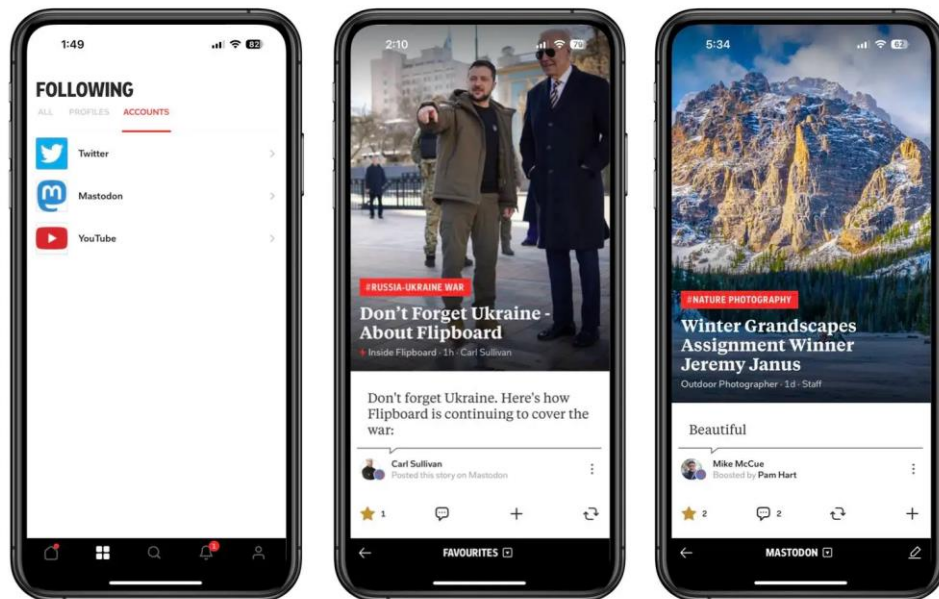


Specifications EpiFlipBoard

Yassin SAID - Guillyann FERRERE



{EPITECH}

General Presentation

Description:

EpiFlipBoard will be a **social magazine mobile application** that allows users to discover, follow, and share content about news and trending topics.

The app will combine content aggregation with social interaction to create an engaging and personalized reading experience.

Key points:

- Users can browse curated articles, images, and videos presented in a magazine-style layout.
- The interface includes a “flip” navigation system inspired by Flipboard’s tactile browsing experience.
- Personalized feeds will adapt to users’ interests, selected topics, or followed sources.
- Social features will allow users to save, share, and recommend content to others.
- The app will focus on simplicity, visual appeal, and smooth interaction.

Context:

This project is part of the **Professional Work module**, designed to simulate the full lifecycle of developing a real-world software product inspired by existing applications.

Key aspects:

- The goal is to apply **Agile methodology** and **software engineering practices** in a professional context.
- The project encourages the use of **modern technologies**, **version control**, and **collaboration tools**.
- The outcome will be both **a functional prototype** and **a technical documentation**, demonstrating software development competencies.
- The reference model is Flipboard — an application that merges social media and journalism in a visual, interactive format.

Objective:

To **design and develop a mobile content aggregation platform** that gathers **news, articles, videos, and images** from multiple online sources and presents them in an interactive magazine-style interface.

Specific objectives:

- Build an intuitive and dynamic interface using modern mobile development technologies.
- Implement an aggregation system capable of fetching data from external APIs (e.g., NewsAPI, RSS feeds).
- Enable users to **create and customize personal magazines** by saving or sharing their favorite content.
- Provide a **smooth flip animation** and clean user experience consistent across mobile devices.
- Integrate a **user authentication system** (login, signup, logout) with personalized data management.
- Ensure responsive performance and scalability for future improvements.

Inspiration:

<https://flipboard.com/>

Elements inspired by Flipboard:

- The concept of a **digital magazine interface**.
- The ability to **curate and share** personalized content.
- The **“flip” browsing animation** between articles.
- The focus on **visual storytelling** and immersive reading.

Target Audience:

EpiFlipBoard aims to attract users who value information accessibility and visual reading comfort.

Main target groups:

- **Students** — who want to stay informed about topics related to their studies or interests.
- **Professionals** — who need a centralized and customizable feed of industry news.
- **General users** — who prefer visual, modern, and social ways of consuming news.

User expectations:

- Centralized, personalized, and real-time access to relevant content.
- Aesthetic and ergonomic interface designed for daily use.
- The ability to engage socially (share, comment, or recommend).
- Lightweight and responsive mobile experience.

Content

Functionality:

Code	Functionality	Description	Priority
F1	Connection	Connection with google in auth2	High
F2	Account creation	Account creation either as publisher/creator or reader	High
F3	Post creation	Publisher and creator can make a post with a title and content text, image and video and categorize that post with different tags	High
F4	Flip navigation	Flip navigation transition while you scroll through the posts	High
F5	Research	The readers can search through specific subject and add tags into their filters	Medium
F6	Post Interaction	Readers can like, comment, save and share a post	Medium
F7	Offline access	Create a cache system to work offline	Medium
F8	Account settings	The users can manage application parameters such as background colors, notification policies...	Medium
F9	Notification	Notification will be sent to reader when a post is related to what they liked	Low
F10	Profil Page	The readers can navigate through their saved magazines	Low
F11	Account interaction	The users can share and block an account as well as following it to get a notification at every update	Low

Technical Stack

The EpiFlipboard project will be developed as a **cross-platform mobile application**, designed to work on both **Android** and **iOS** devices.

Frontend (Mobile App Interface):

Framework: Flutter

- Provides a single codebase for Android and iOS with excellent performance and near-native rendering.
- Ideal for building smooth animations, especially the magazine “flip” effect.

Language: Dart

- Flutter’s official programming language, offering strong typing and fast compilation.

UI Design: Flutter Widgets + Material Design 3

- Ensures a consistent and responsive interface with elegant transitions and flexible layouts.

Animation Library: Flutter Animation Controller / PageView / Rive

- Used to implement interactive “flip” effects and smooth page transitions between articles.

State Management: Provider or Riverpod

- Simplifies data flow and state handling within the app.

Backend (Server and API Layer):

Framework: FastAPI (Python) or Node.js (Express)

- Handles user authentication, database operations, and external API communication.

API Type: REST API (JSON)

- The mobile app communicates with the backend using standard REST endpoints.

Database

Database System:

PostgreSQL (hosted on Supabase or Neon.tech)

→ Stores user accounts, preferences, and saved articles.

ORM:

For FastAPI → SQLAlchemy

For Node.js → Prisma

External APIs

NewsAPI.org, RSS Feeds, or other open news sources

→ Used to fetch news articles, images, and metadata dynamically.

Storage and Hosting

Backend Hosting: Render, Railway, or Supabase Functions

Database Hosting: Supabase (PostgreSQL)

Media Storage: Supabase Storage or Firebase Storage (for user avatars or saved media)

Testing and Deployment

Testing:

Frontend → **Flutter Test / Widget Test** for frontend

Backend → Pytest (FastAPI) or Jest (Node.js)

Continuous Integration / Deployment:

GitHub Actions for automated testing and deployment

Mobile Distribution:

- **Android Build:** Generated using **Flutter Build APK / App Bundle** commands.
→ Example: flutter build apk --release or flutter build appbundle
- **iOS Build:** Generated with **Xcode** or **Codemagic** (cloud CI/CD platform for Flutter).
→ Example: flutter build ios --release
- **App Distribution Platforms:**
- **Google Play Console** for Android
- **Apple App Store** for iOS
- **Firebase App Distribution** (optional) for internal testing before release

Project Management and Collaboration

- **Version Control:** Git + GitHub
- **Project Management:** GitHub Projects (Kanban: To Do / In Progress / Done)
- **Communication:** Discord / Slack
- **Documentation:** README + Wiki (GitHub)