

Offline WebView App



Group 15

Members:

Karthikheyam Suresh Kumar
Sadman Sameer
Savitaj Singh
Anuj Sevak
Nicholas Lai

Supervisor:

Dr. Reza Shahidi

Client:

Median.co



AGENDA

- Introduction
- Engineering Problem Addressed
- Project Requirements
- Design Alternatives & Selection
- Solution Design
- Project Outcomes
- Conclusion

Native vs WebView



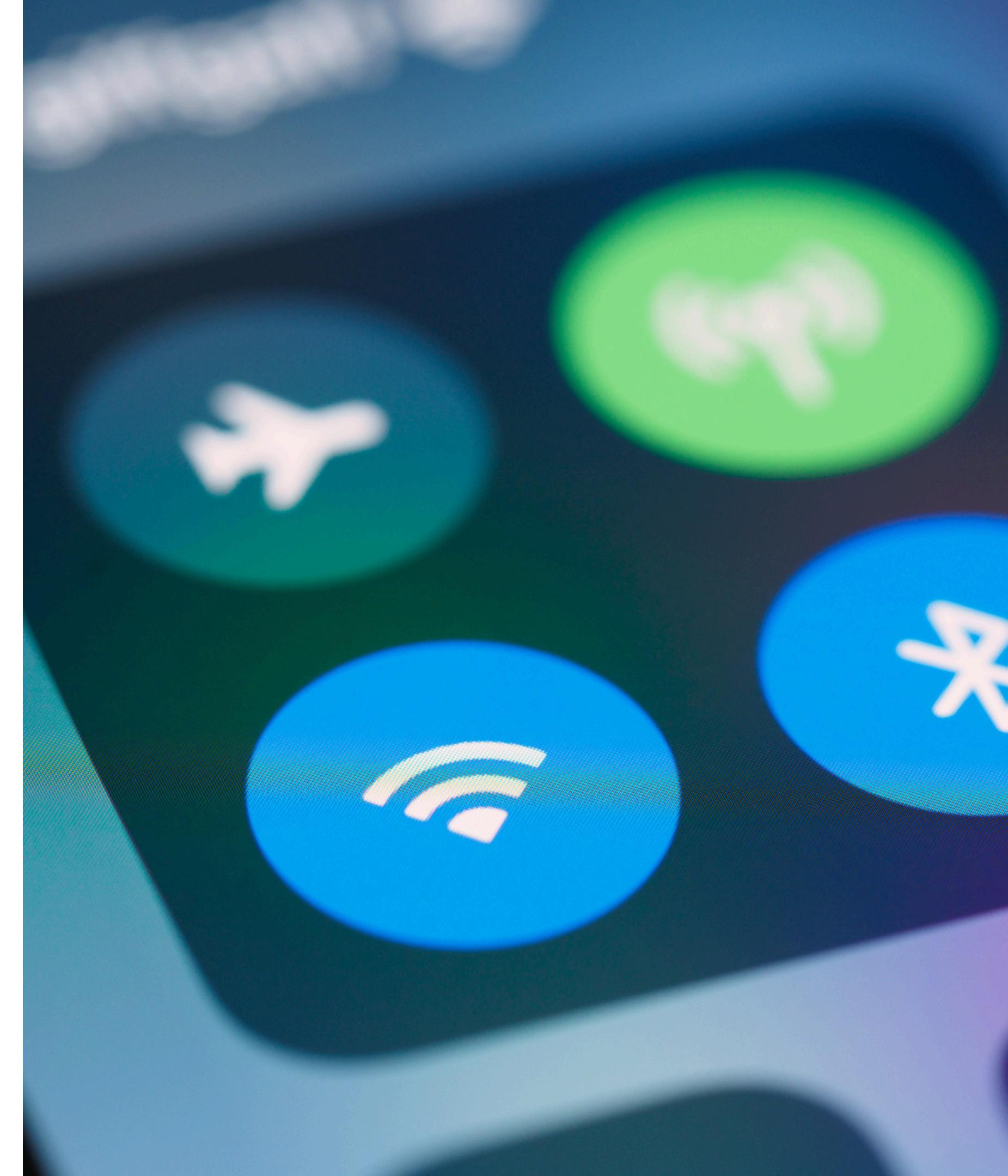


INTRODUCTION

WebView Applications

PROBLEM STATEMENT

WebView apps require a constant internet connection, making them unreliable in areas with poor or no internet connectivity.



WEBVIEW APPS

OTHER PROBLEMS



**SMALL/ MEDIUM
SIZE BUSINESSES**



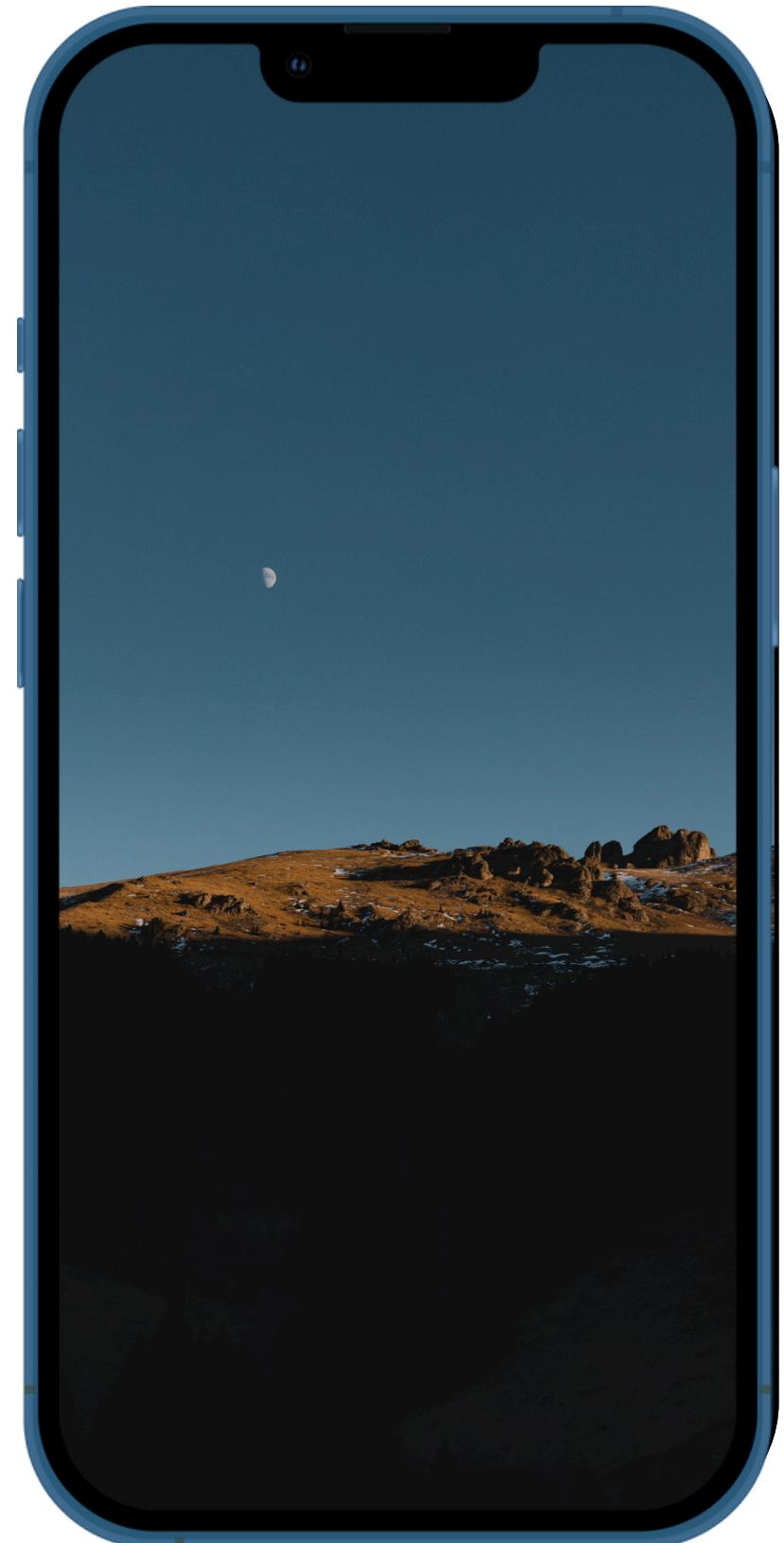
CACHING

OUR PROJECT

OFFLINE PLUGIN FOR MEDIAN

SMART
CACHING

CONTENT
SYNCHRONIZATION



GOAL

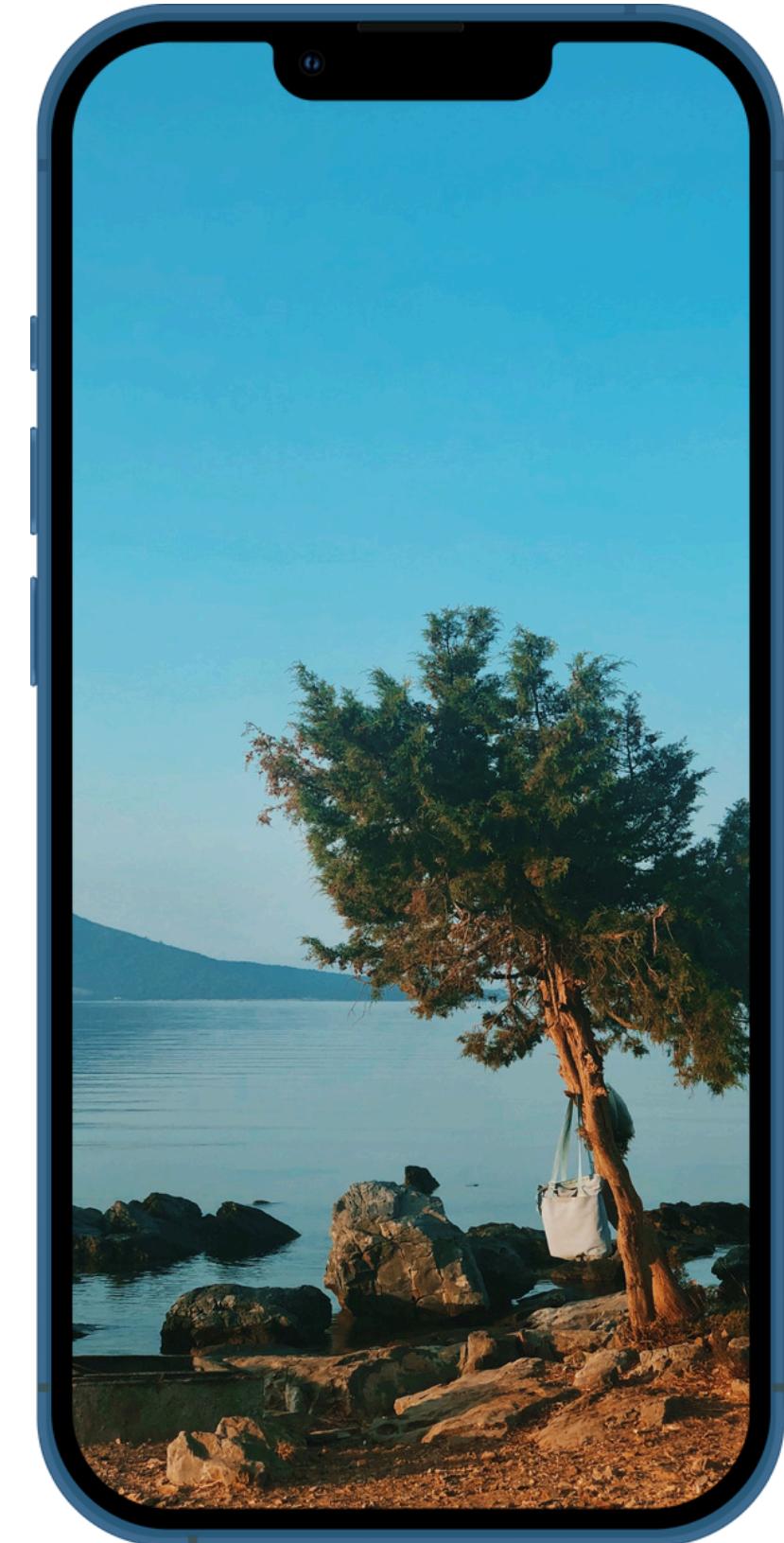
Making offline
WebView
applications
more accessible
and cost-
efficient for
businesses.

Project Requirements



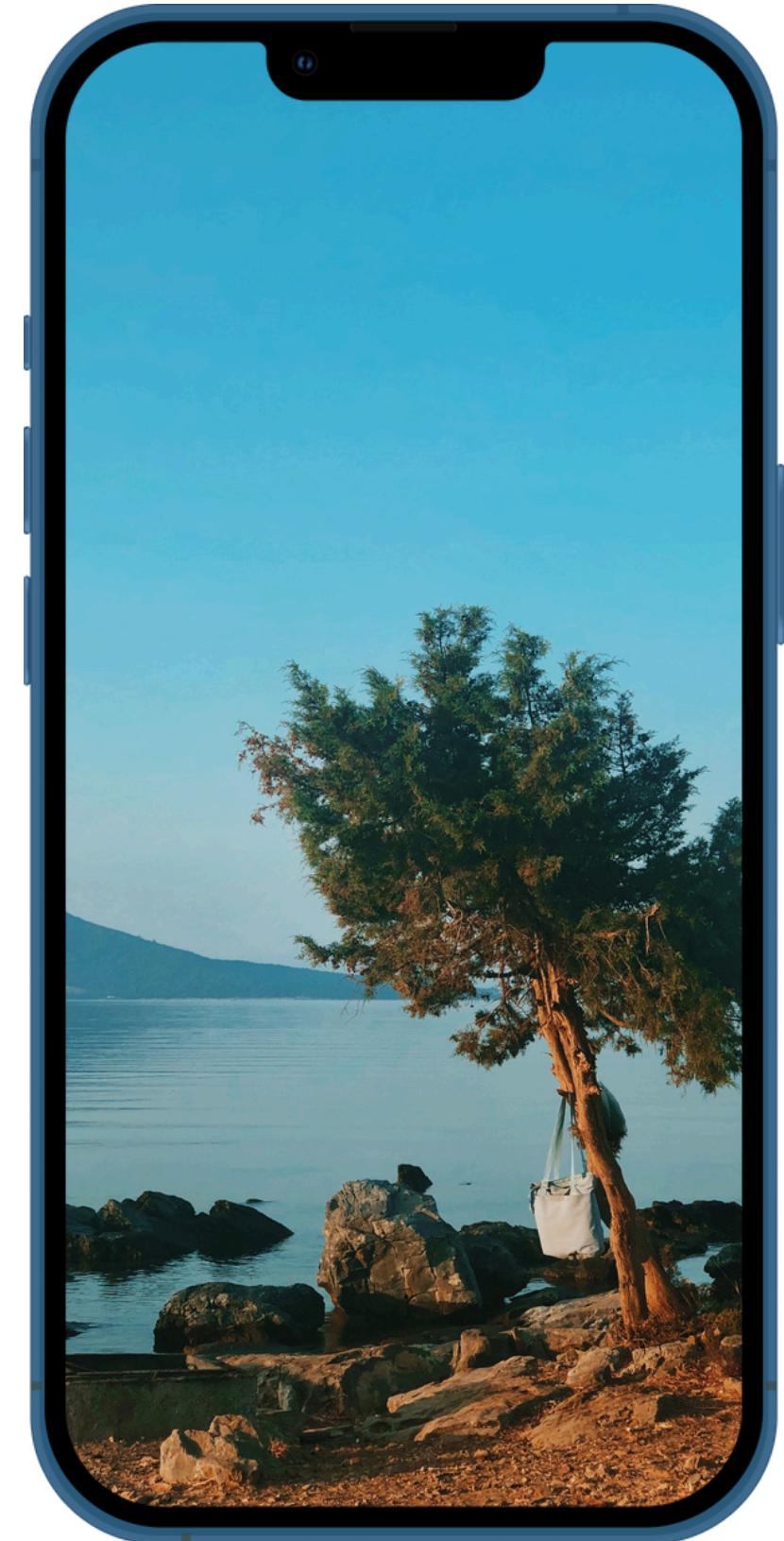
REQUIREMENTS

- Efficient Caching Mechanism
- Smart Content Synchronization
- Seamless Online-Offline Switching
- Minimal Impact on Battery Life & Storage



SUCCESS METRICS

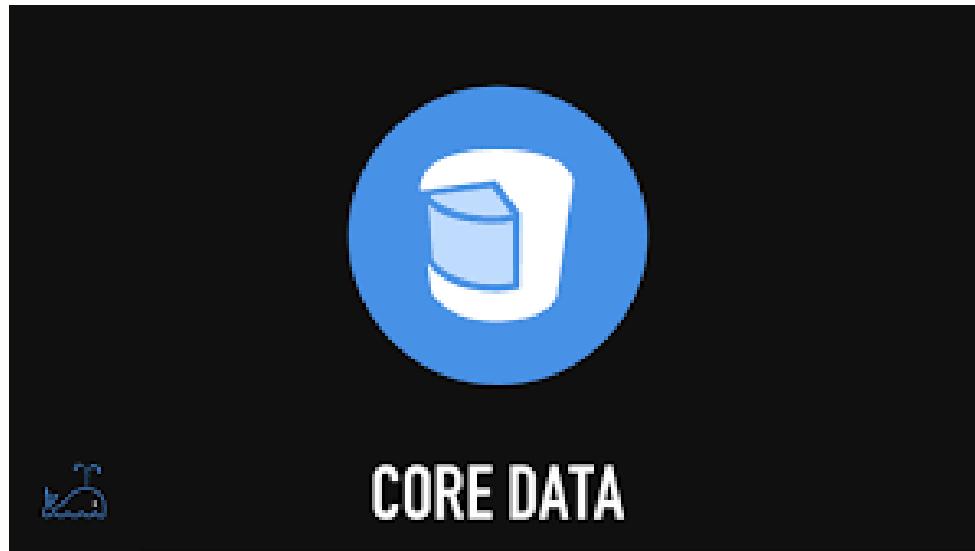
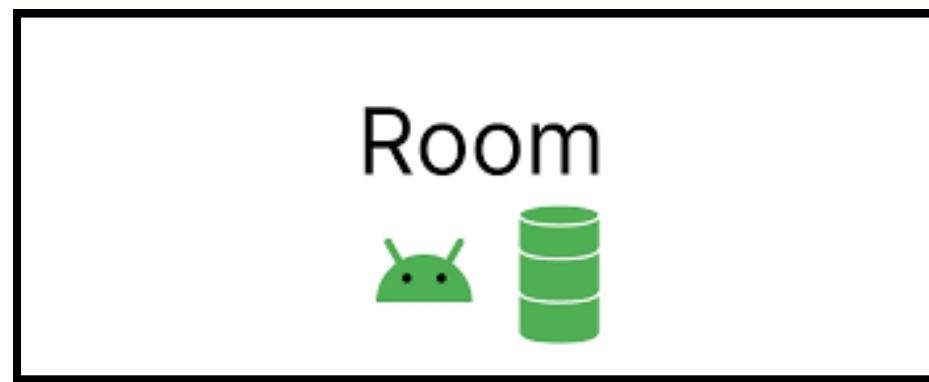
- Reduced Page Load Failures
- Cached Pages Should Load Instantly
- Efficient Background Updates
- Cross-Platform Compatibility?





Design Alternatives & Selection Process

Database Selection



Performance

Cross-Platform

Efficiency

[2], [3], [4]

Storage & Caching Mechanisms

NSURLcache (iOS)

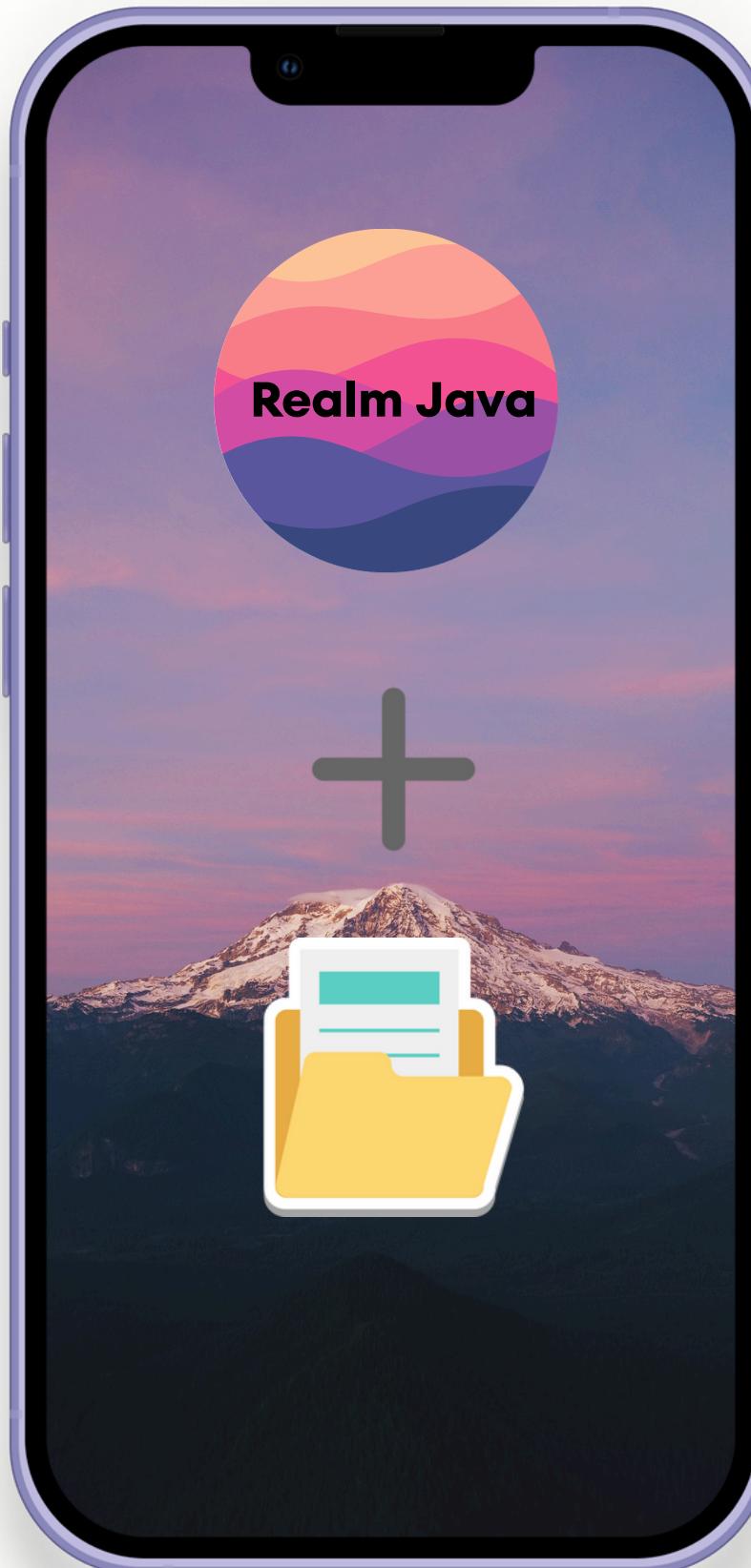
Automatic Caching but Limited Control

IndexedDB (iOS)

Structured Data, but complexity in file handling

Service Workers Cache

Limited Storage Size



Efficient Cache Management

Storage Scalability

Fast Querying & Retrieval

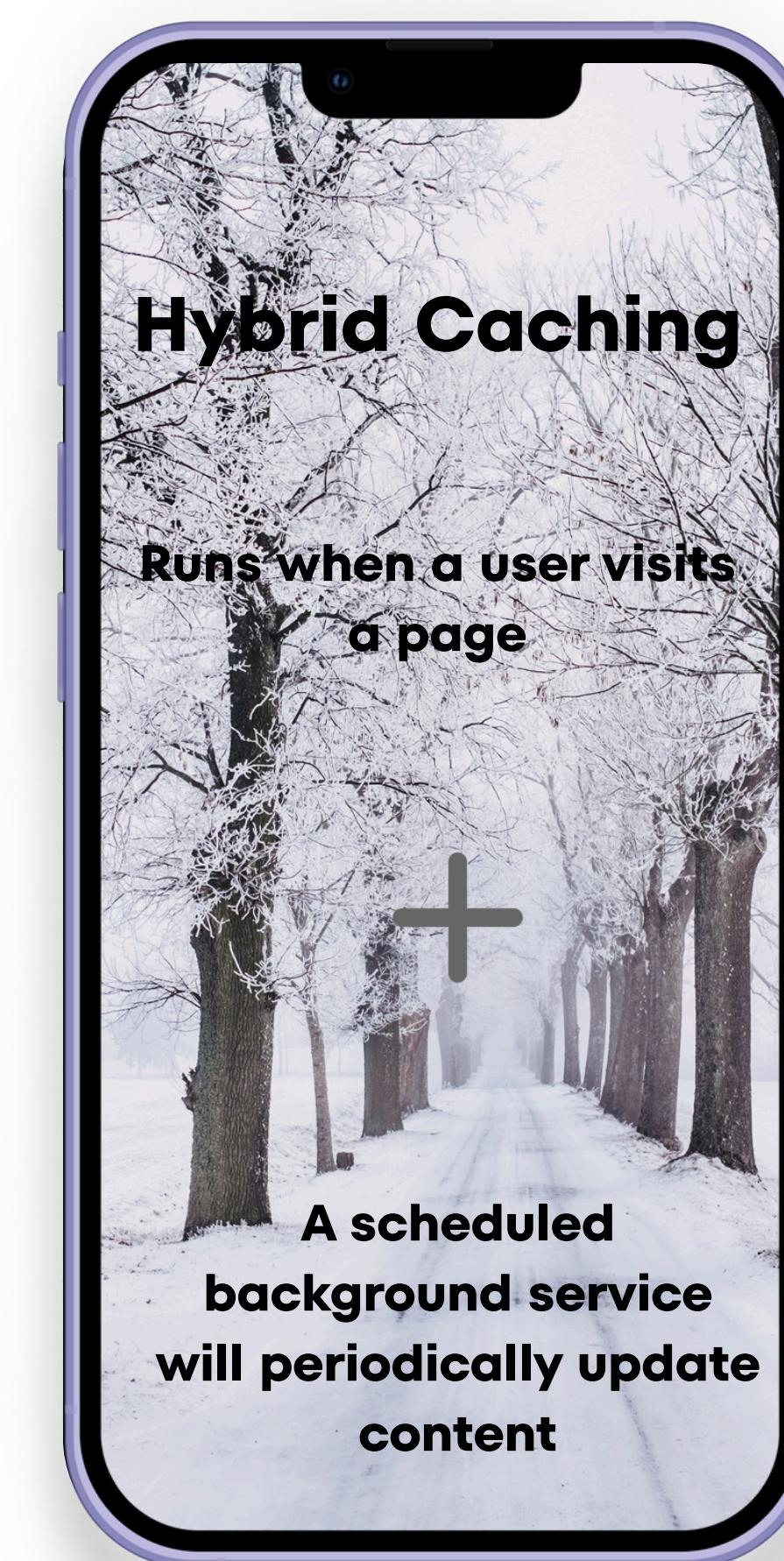
Background Service Selection

Temporary Caching

Runs Only when specifically called for

Persistent Caching

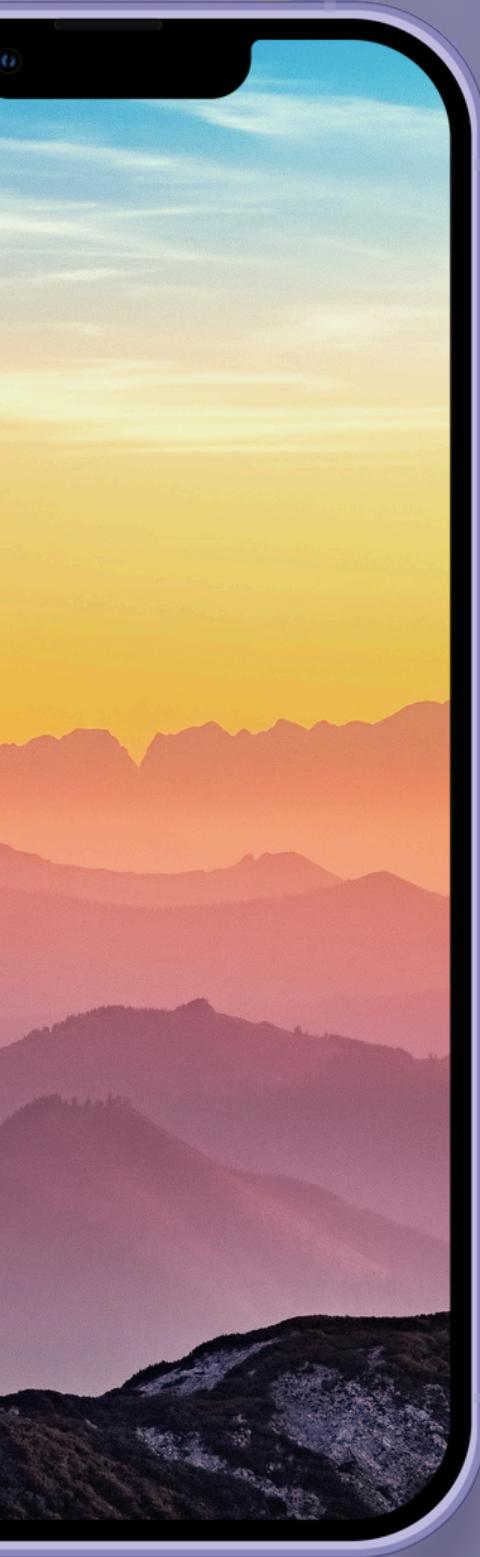
Always running - Ensures content stays updated



On-Demand
Caching

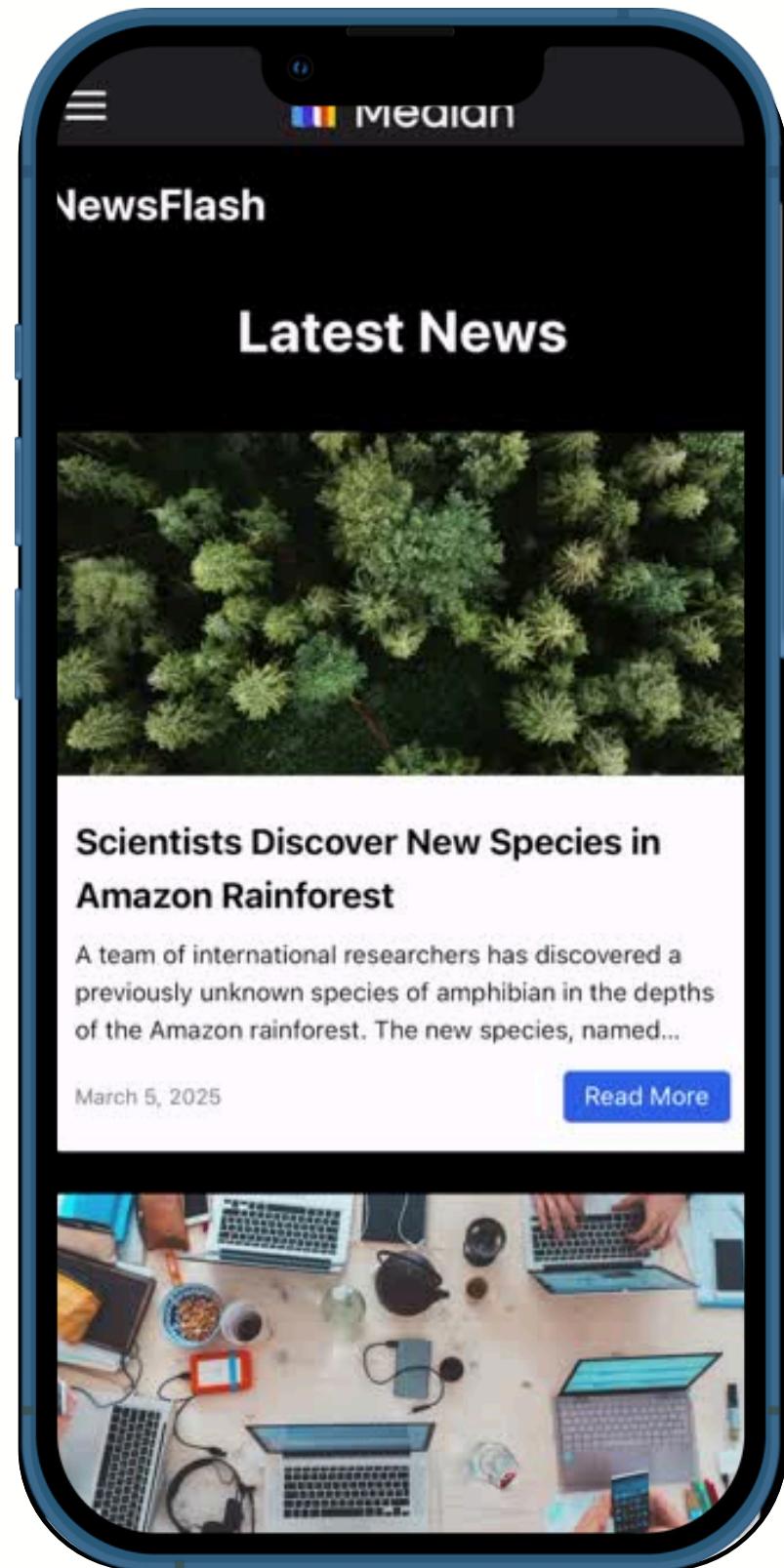
Periodic Sync

Battery
Efficiency

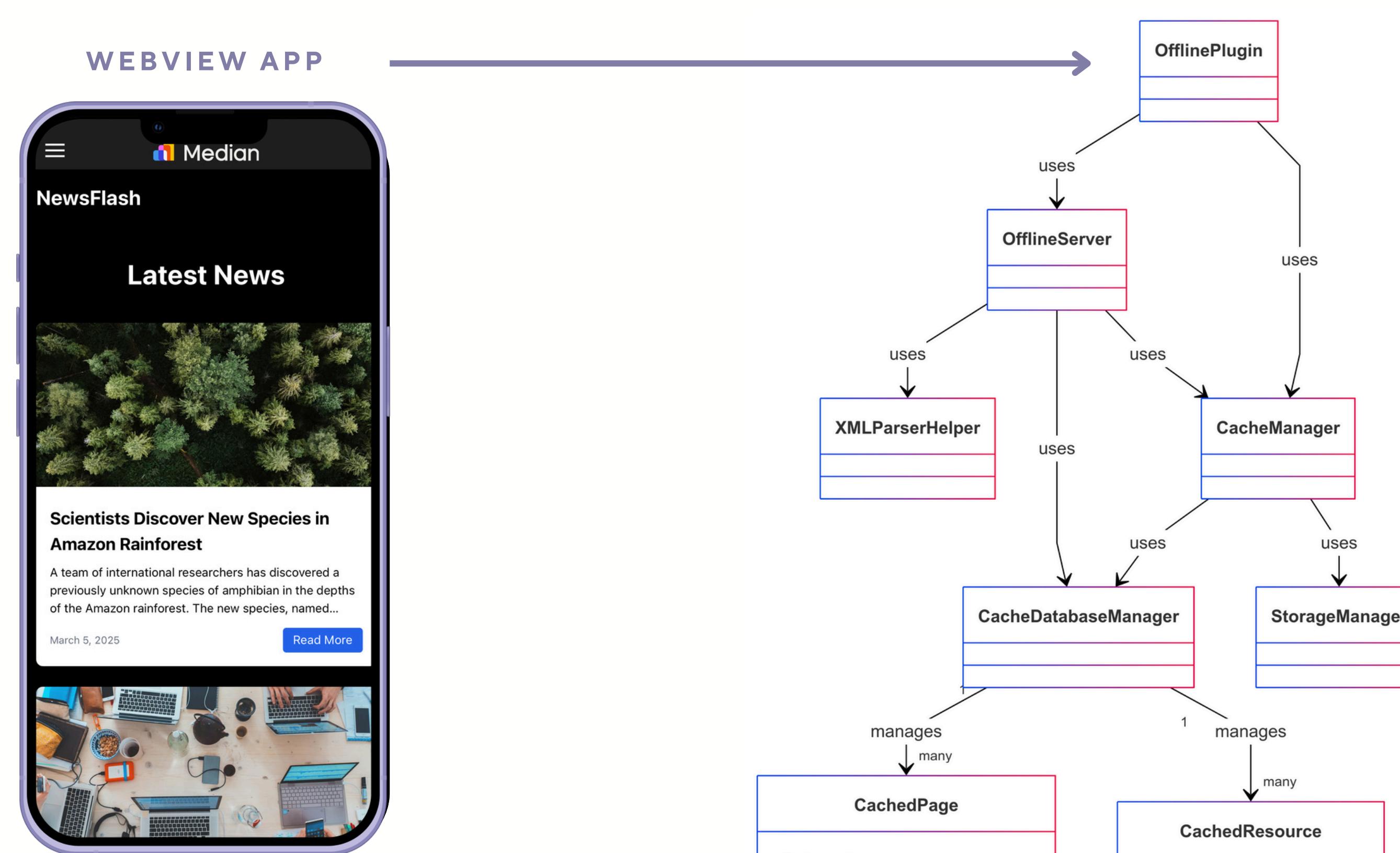


Design

Simple Webview App Demo



High Level Design



CLASS

CacheManager

As the name suggests



Download web pages and resources

using HTML parsing libraries like SwiftSoup and Jsoup

Save cache content records in local database

using CacheDatabaseManager

Cache management

Make caching decisions using DB and StorageManager

Cache cleanup

Cleanup cache based on available storage

CLASS

CacheDatabase Manager

CRUD operations manager



Manage CRUD operations using Realm DB

Exposes CRUD methods

Saves pages and resources info in DB

Uses CachedPage and CachedResource Model

CLASS

CachedPage

Database Model



INCLUDES:

URL

URL of the requested page

Status

Whether the page is downloading or downloaded

Last modified by server date

Last modified date of the page by its server

File path

Path of the cached page in local storage

Saved by user

Boolean value indicating whether this was cached by device user

Expiry date

Date when the cache is expired and cleaned up

Priority

Priority value of the cached page

Resources

List of cachedResources this page uses

CLASS

CachedResource

Database Model



INCLUDES:

URL

URL of the requested page

File path

Path of the cached page in local storage

Name

Boolean value indicating whether this was cached by device user

Saved by user

Boolean value indicating whether this was cached by device user

Expiry date

Date when the cache is expired and cleaned up

Priority

Priority value of the cached page

Used by pages list

List of cached page URLs this resource is used by

CLASS

StorageManager

Device Storage Manager



Manages storage space and limits for cached content

By calculating available disk space

CLASS

OfflineServer

Manage offline page access and caching



Handles page requests

Determines online load vs cached page load

Enforces caching decisions

Using CacheManager, determines what and when to cache based on web developers' provided configurations.



CLASS

OfflineServer

Manage offline page access and caching

EXAMPLE CONFIGURATION:

```
"landingPage": "https://simple-news-app-three.vercel.app",
"prefetchUrlsRegexes": [
  {
    "regex": "https://simple-news-app-three.vercel.app/article1.html",
    "priority": 0.9
  },
  {
    "regex": "https://simple-news-app-three.vercel.app/article2.html",
    "priority": 1
  }
]
```

CLASS

OfflinePlugin

Main plugin entry point for WebView App

Serve cached resources

via OfflineServer

Provides JavaScript Bridge

For web developers to access and manage cache via their websites



CLASS

OfflinePlugin

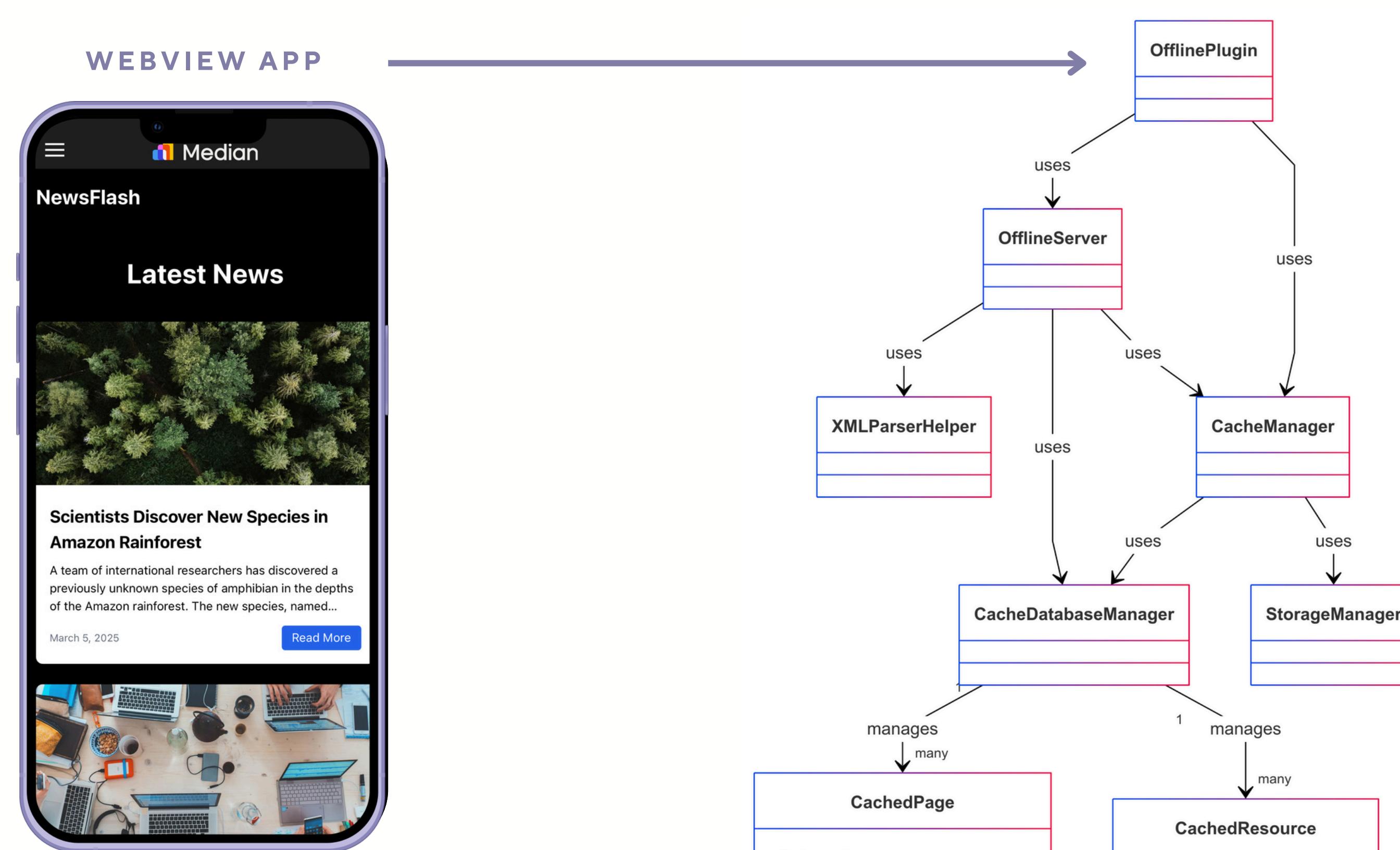
Main plugin entry point for WebView App



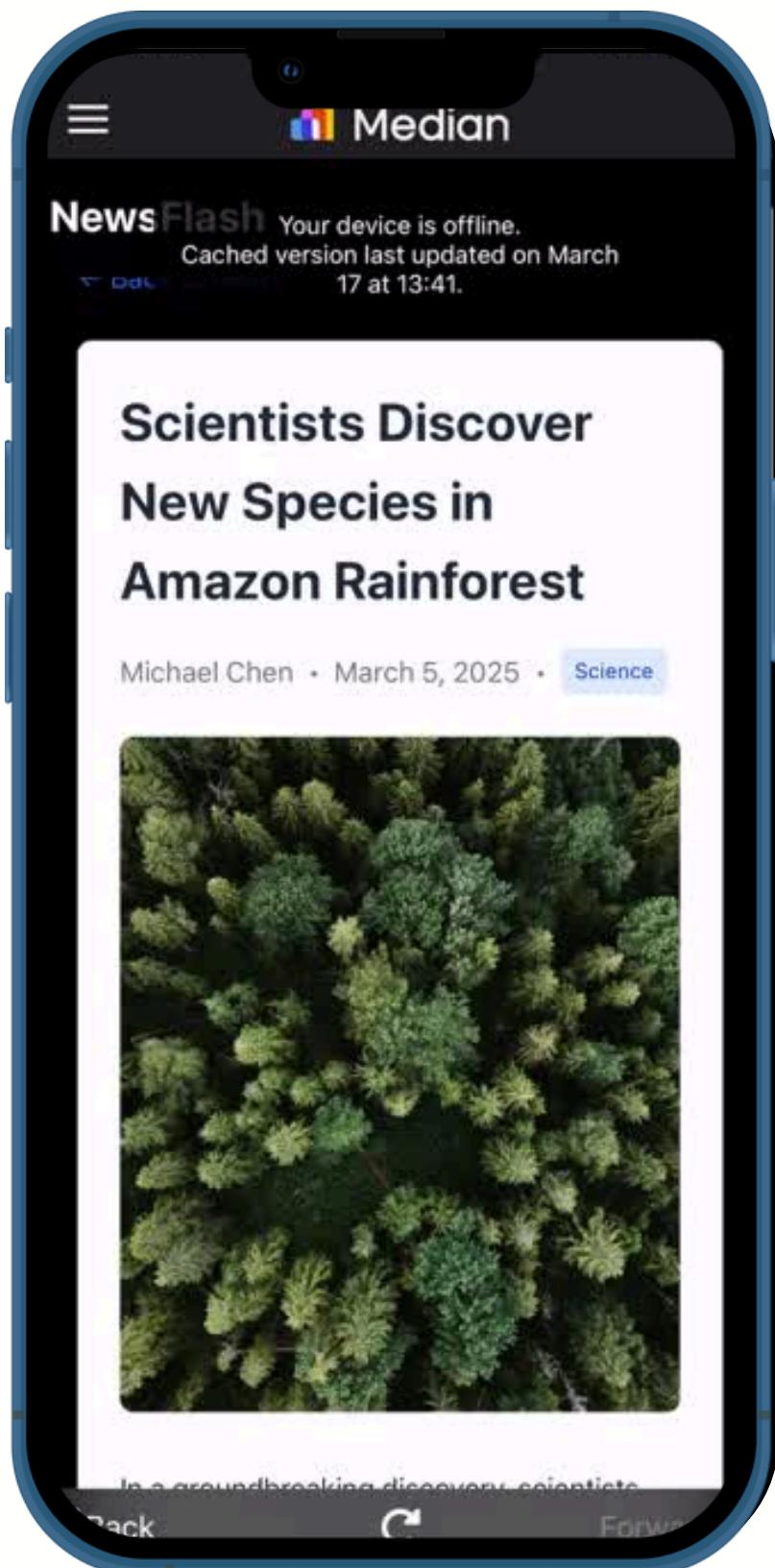
JAVASCRIPT BRIDGE EXAMPLE

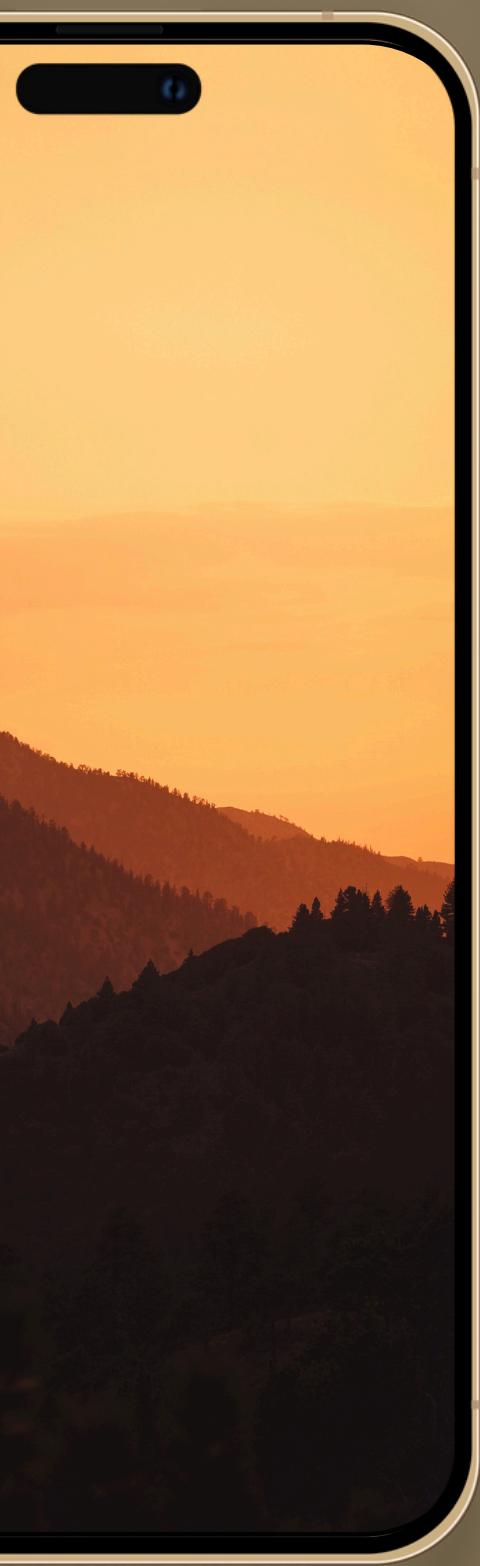
```
<script>
  async function downloadArticle() {
    let response = await median.offlineAppSync.cachePage({
      url: 'https://simple-news-app-three.vercel.app/article5.html',
      priority: 1,
      name: 'International Arts Festival Celebrates Global Cultural Heritage',
      saveForUser: true,
    });
    if (response.success) {
      console.log('Article cached for offline reading!');
    } else {
      console.error('Failed to download article');
    }
  }
</script>
```

Design Diagram revisit



Offline Webview App Demo





Project Outcomes

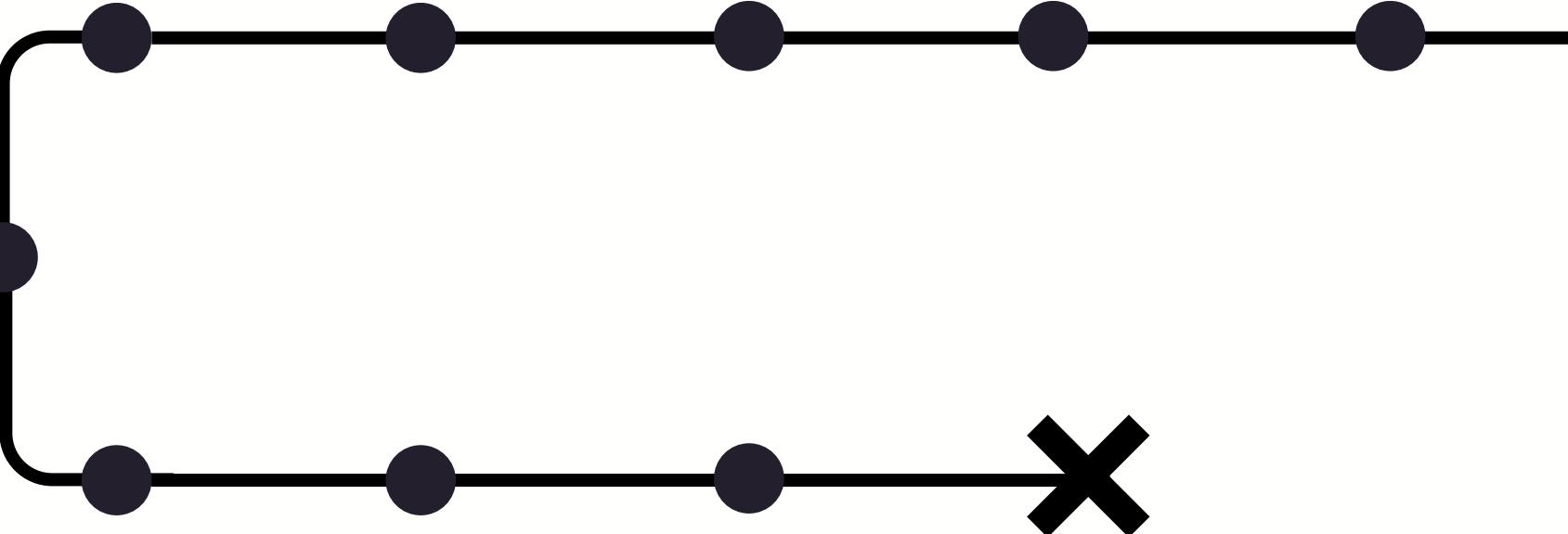
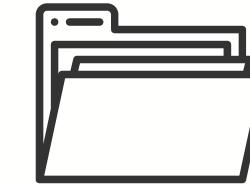
Changes in Scope



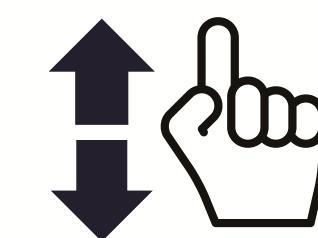
Javascript Bridge



Storage Limit



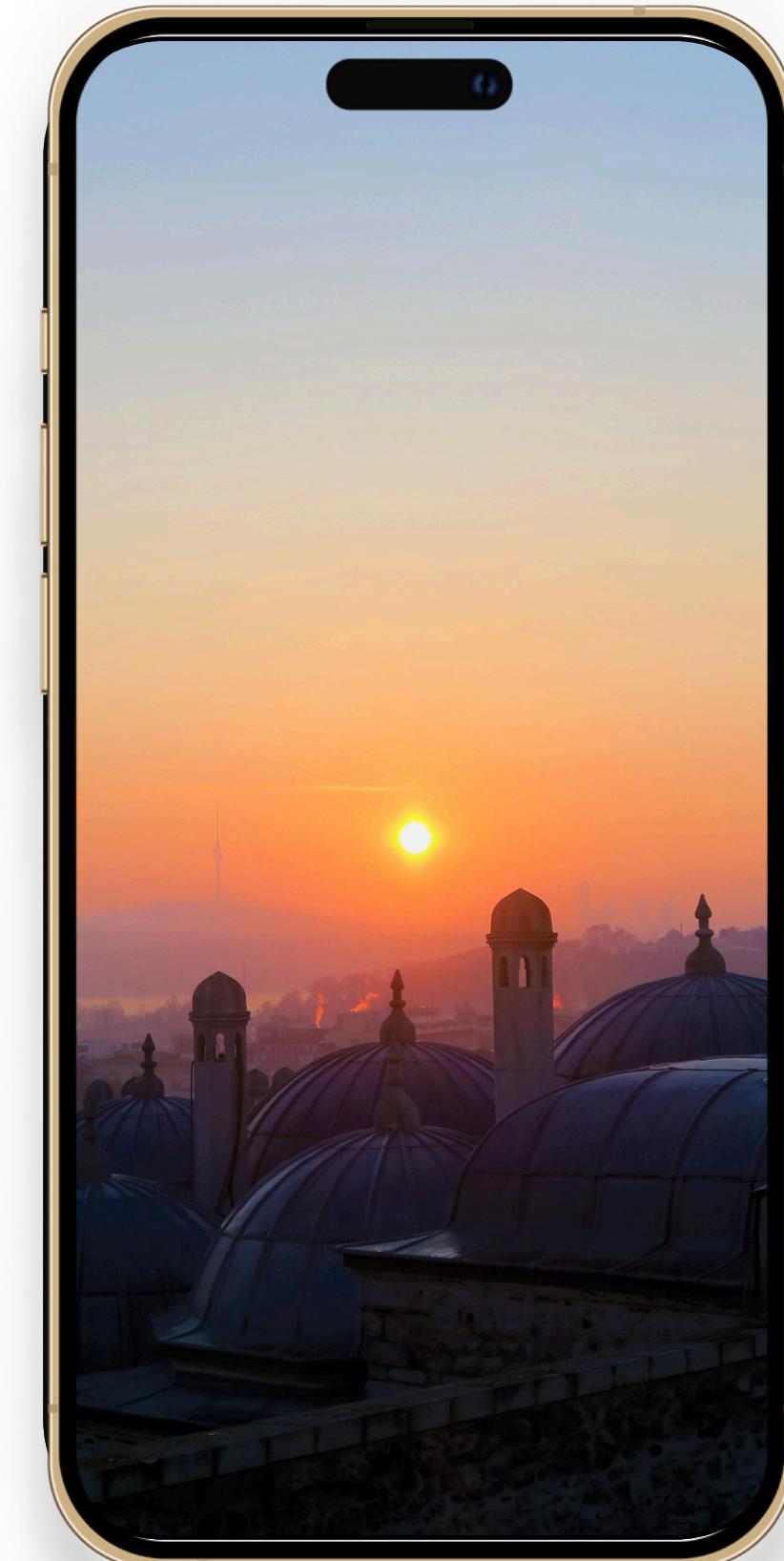
Google Analytics



Infinite Scrolling

Goals Fulfilled

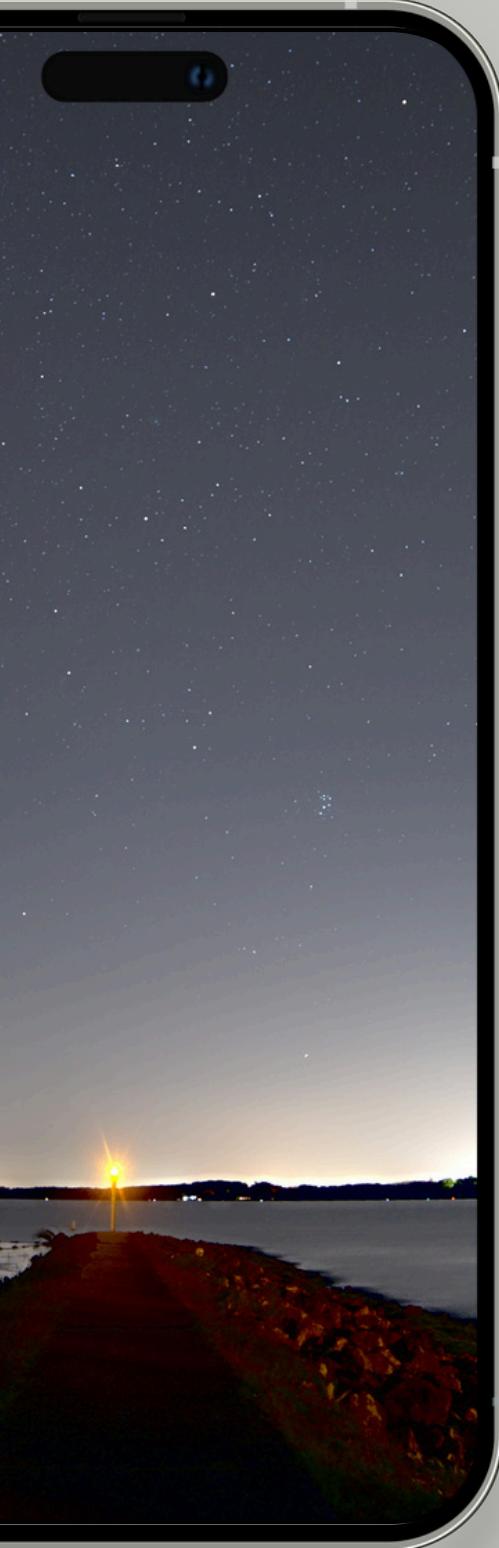
- Bridged WebView and native performance with offline browsing
- Developed and integrated MVP with core offline features
- Built scalable, cross-platform solution for future expansion
- Limit impact on user device with battery and storage optimization



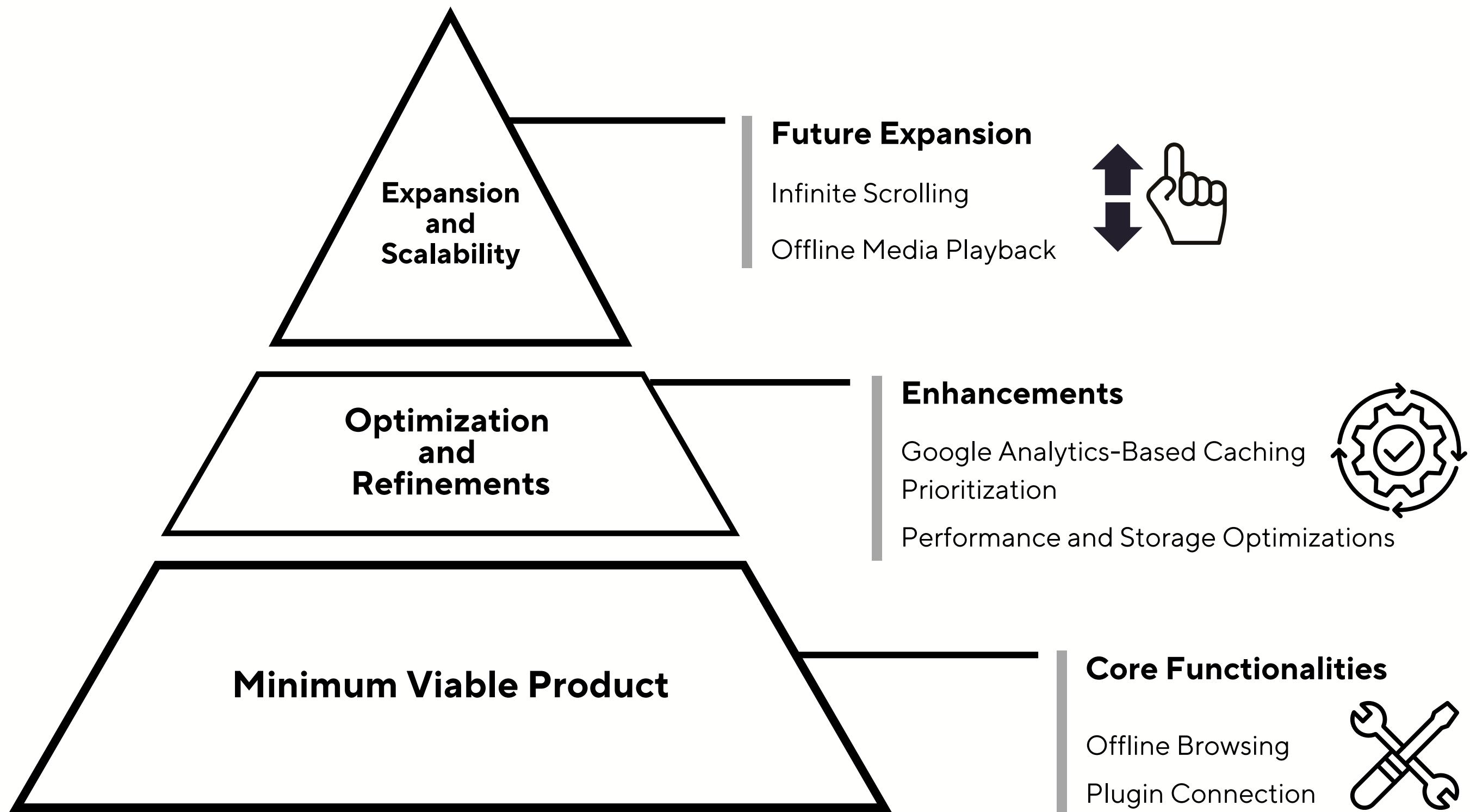
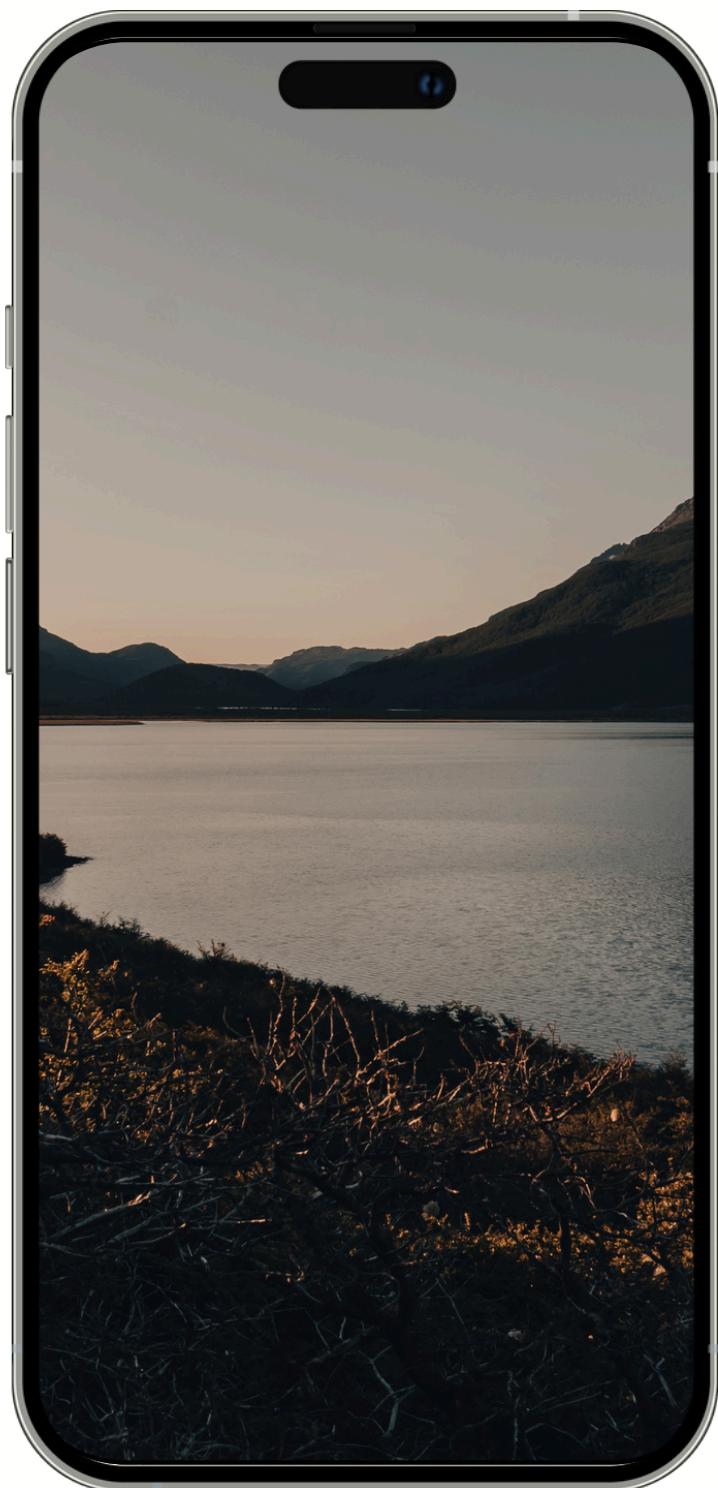
Core Functionalities

- Smart Caching
- JavaScript Bridge
- Connectivity Awareness
- Prefetching
- Storage and Battery Optimization

Future Work



Feature Pyramid



Conclusion

- Project successfully addresses the challenge of enabling offline functionality in WebView applications
- We have provided a cost-effective alternative to Native apps for Small/Medium sized businesses
- While core objectives have been achieved, opportunities for further improvements remain



Thank You!

Median.co
Group 15

References

- [1] Median.co, "Native apps vs. webview apps: What's the best choice?," Native apps vs. webview apps: What's the difference?, <https://median.co/blog/native-app-vs-webview-app-what-is-the-difference> (accessed Mar. 17, 2025).
- [2] "Core Data," Apple Developer Documentation. [Online]. Available: <https://developer.apple.com/documentation/coredata> (accessed: March 16, 2025).
- [3] "Save data in a local database using Room," Android Developers, [Online]. Available: <https://developer.android.com/training/data-storage/room>. [accessed: March 16, 2025].
- [4] M. G. R. Batista, "Realm Is the Best Android Database Solution," Toptal, [Online]. Available: <https://www.toptal.com/android/realm-best-android-database-solution>. [accessed: March 16, 2025].
- [5] "IndexedDB API," Mozilla Developer Network (MDN) Documentation. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API (accessed: March 16, 2025).
- [6] "NSURLCache," Apple Developer Documentation. [Online]. Available: <https://developer.apple.com/documentation/foundation/nsurlcache> (accessed: March 16, 2025).
- [7] "Service workers ,," web.dev, [Online]. Available: <https://web.dev/learn/pwa/service-workers> (accessed March 16, 2025).

References

[8] "jsoup: Java HTML Parser," jsoup. [Online]. Available: <https://jsoup.org> (accessed: March 18, 2025).

[9] "SwiftSoup: Pure Swift HTML Parser," GitHub. [Online]. Available: <https://github.com/scinfu/SwiftSoup> (accessed: March 18, 2025).