



Skené

Developed by
Piervito Creanza & Gianluca di Paola

Design and Implementation of Mobile Applications
Professor Luciano Baresi
PoliMi, a.a. 2025-2026

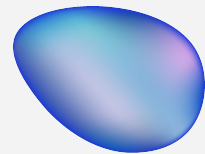


Why Skené

In **ancient Greek theatre**, the **skēnē** (σκηνή) was the backdrop, the structure behind which **actors transformed**, before which **stories came to life**.

It was the foundation of the stage, the space **where imagination met reality**.

Today, Skené brings that same transformative power to the Apple ecosystem a place where the user takes the center stage and embraces **the pàthos of cinema**.



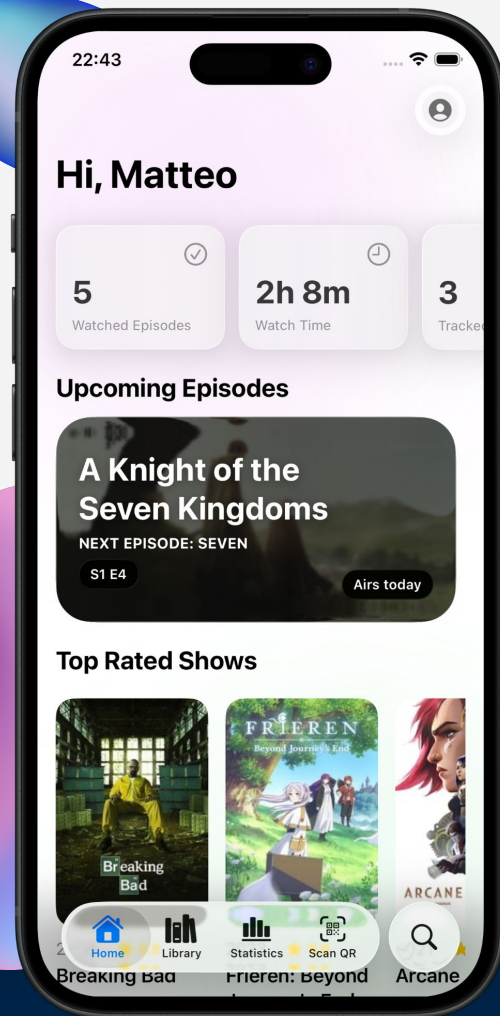
Skené today

Skené **native iOS/ipadOS/macOS 26+ application** for TV enthusiasts to discover, monitor, and manage viewing progress.

Key Goals:

- Streamline episode **tracking** and **release** monitoring.
- Leverage the **TMDB API** for a global catalog.
- Ensure data security via **iCloud** synchronization and local **SQLite** persistence.

Skené adopts a **privacy-first approach**. We empower the user, not ourselves: user data belongs exclusively to the user.



Skené Values

Privacy by
design with
iCloud storage

QRs driving
Real-world
connections

No
metrics-driven
behavior

HCI-driven,
User-Centered
Design

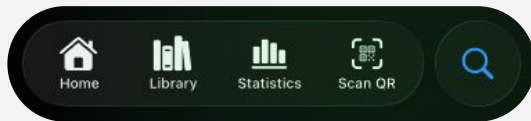
Native design
language

Responsive
Design

Skené Features



- **Library Management:** Curate personal collections and track watch states for episodes/seasons.
- **Discovery:** Search with comprehensive metadata and trending content carousels.
- **Notifications:** Alert users of upcoming air dates.
- **Social:** QR code system for peer-to-peer library sharing without centralized servers.
- **Widgets:** Home Screen access for the next scheduled episode.



Application Architecture

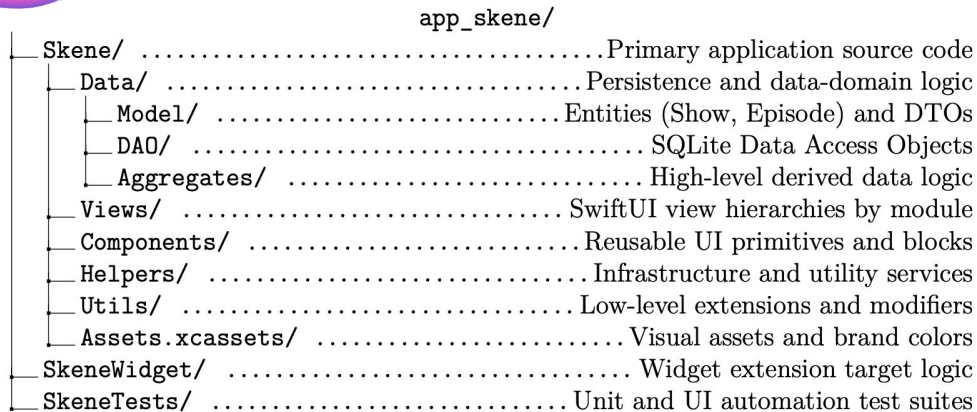


Figure 2.1: Project directory structure of the Skene application

We employed a **MVVM (Model-View-ViewModel)** design pattern, paired with a Layered Design:

- The **Presentation Layer** contains **Stateless SwiftUI views** and **reusable UI** components.
- The **Business Logic** has ViewModels **managing state** and **async flows**.
- The **Data Layer** is made of **SQLite persistence** and **iCloud/CloudKit synchronization**.
- The **Service Layer** contains dedicated **clients** for **TMDB API**, **notifications**, and **analytics**.

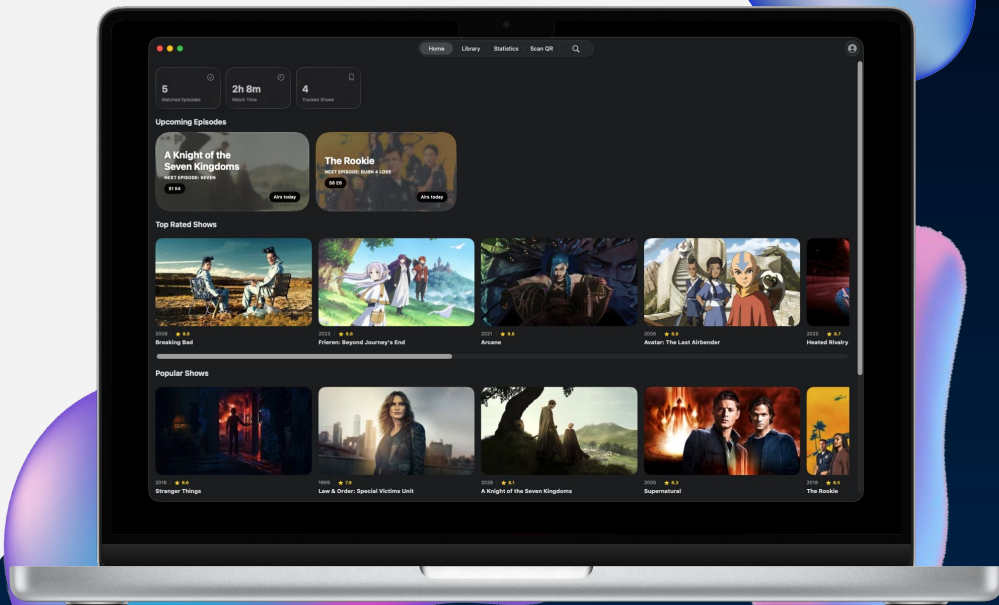
Data Management & Offline-First

Local-First Model: All primary queries target the local database for zero-latency; remote APIs is only used for the first fetch and to update metadata.

Authoritative State: Local user state (favorites, watch status) is preserved during remote metadata updates via an "Upsert" strategy.

iCloud Sync: Deliberately avoids traditional backends (Firebase/Auth0) for native, zero-knowledge privacy.

Universal Compatibility: Shared 95%+ codebase across iPhone, iPad, and Mac (iOS 26+).



RESPONSIVE

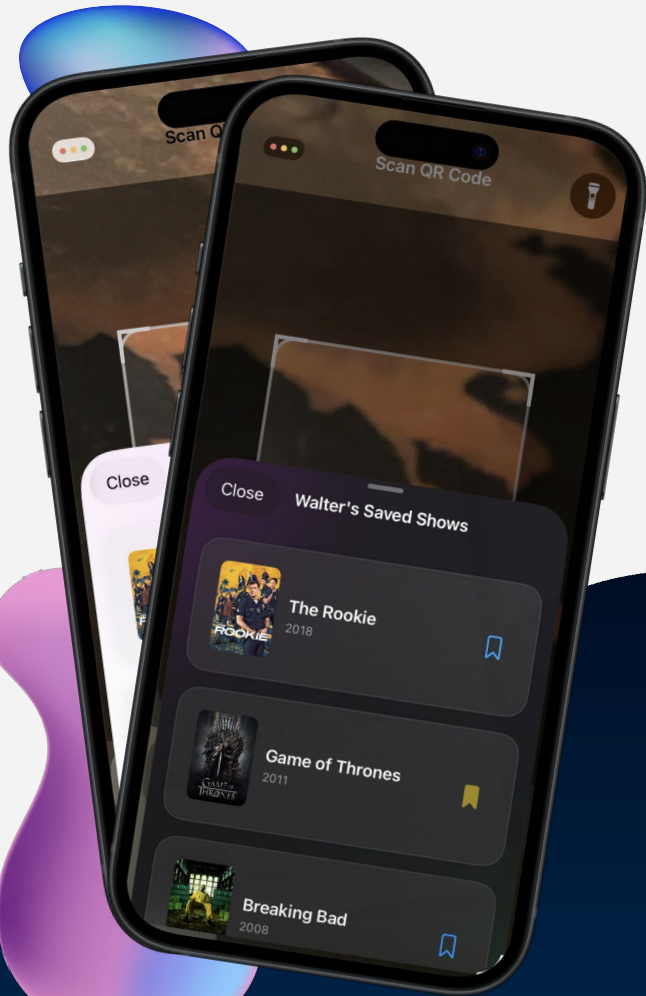


User Interface & Design Philosophy

The Problem: Traditional social features create digital footprints and distorted behavior.

The Skené Solution:

- **QR Sharing:** Requires physical proximity, turning show recommendations into genuine social moments.
- **No Backend:** No persistent social graph or server mediates the exchange.
- **Resistance to Metrics:** No "likes," follower counts, or leaderboards.



Home Screen Widgets

Single-Episode Focus: Intentionally displays only the most imminent episode to reduce cognitive load.

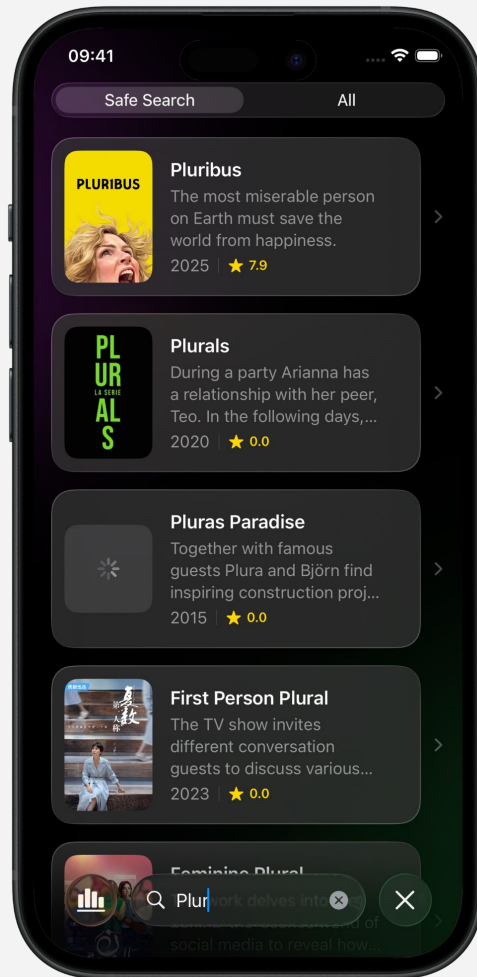
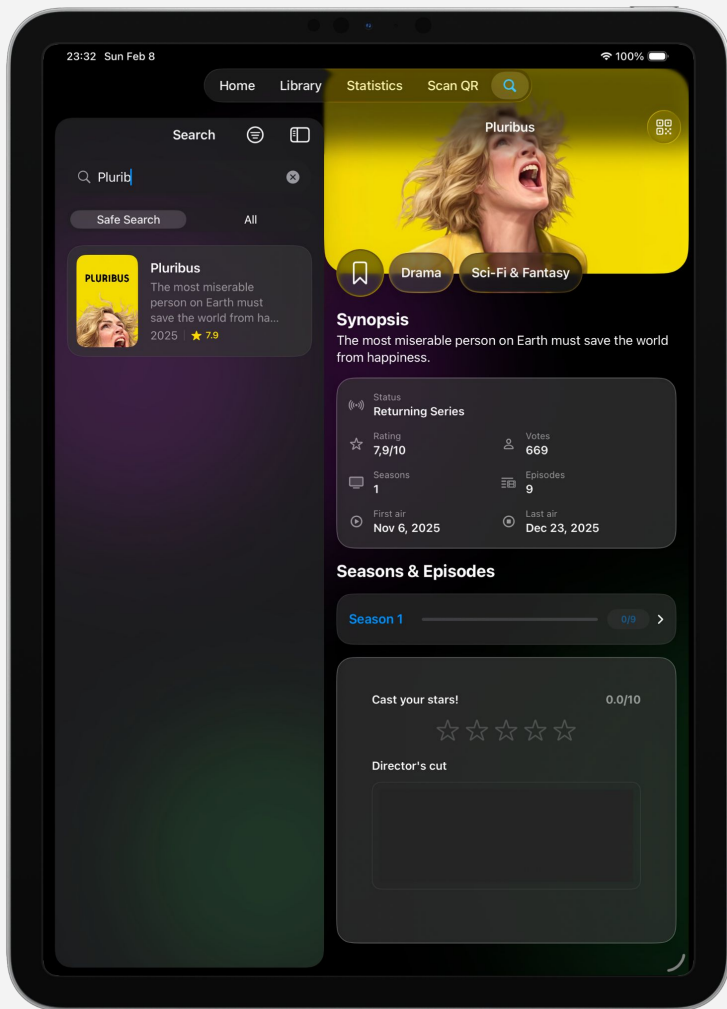
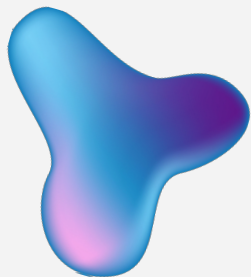
Supported Sizes: Small (compact info) and Medium (adds episode titles).

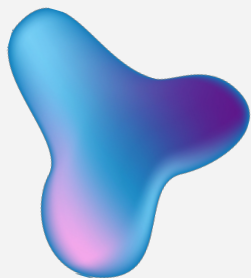
Integration:

- **Shared App Group:** Reads directly from the main app's SQLite DB for zero-latency.
- **Deep Linking:** Tapping the widget opens the app directly to the specific episode.

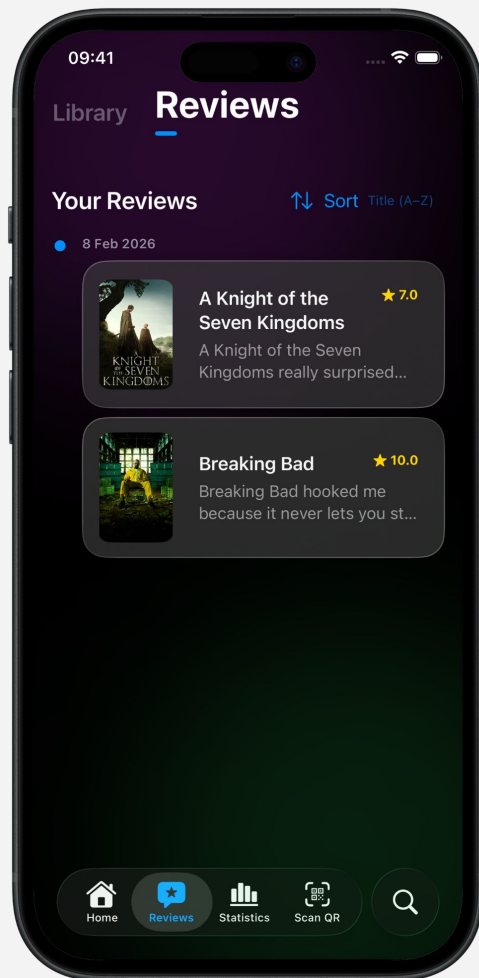
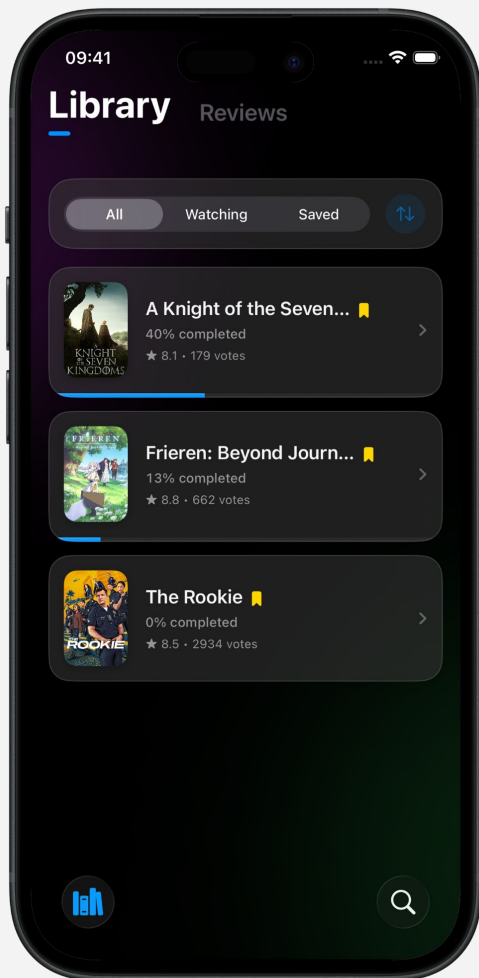


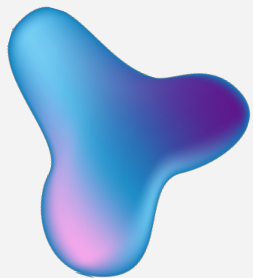
SEARCH



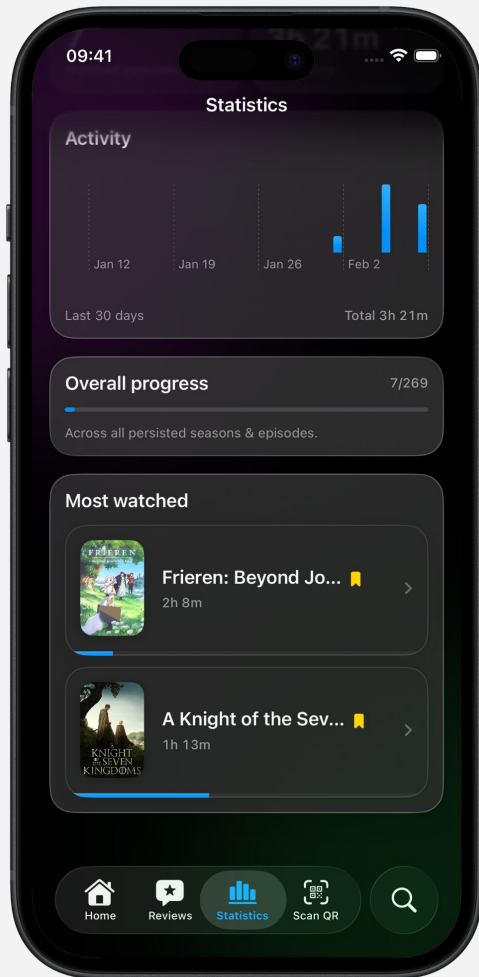
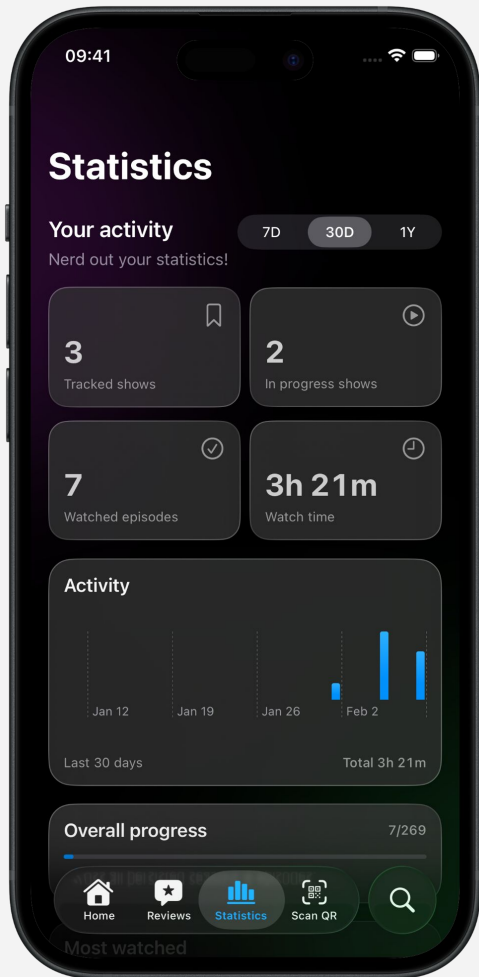


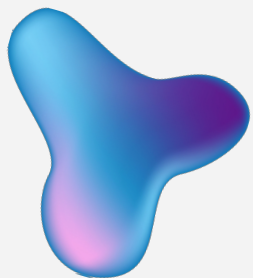
LIBRARY



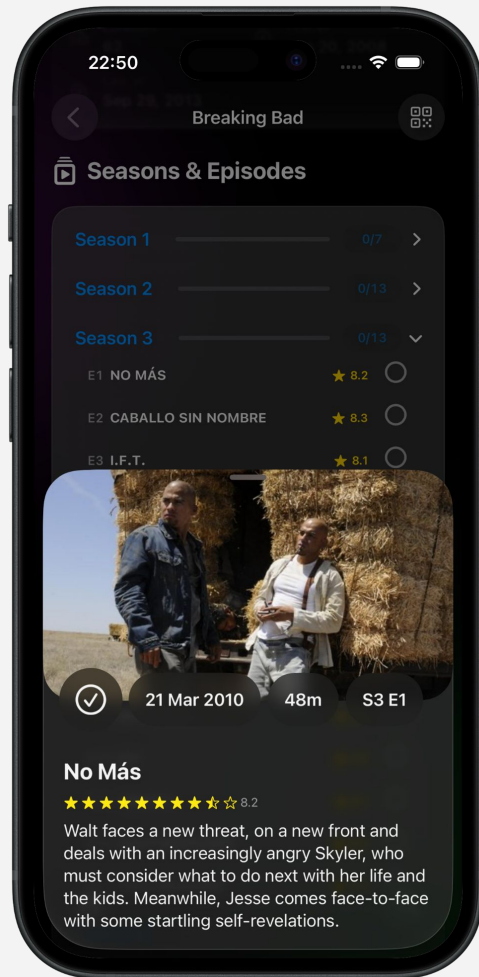
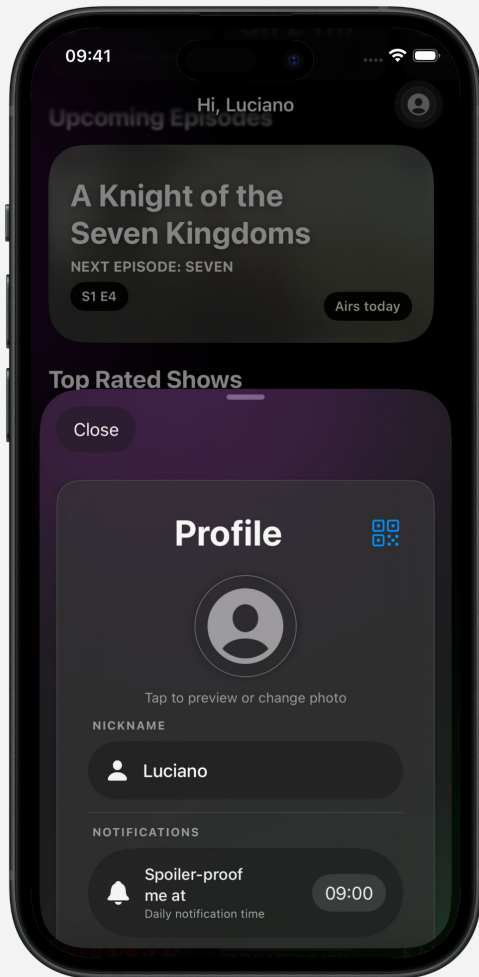


STATISTICS





APP SHEETS



Testing & Quality Assurance

Dual-Pronged Strategy: Automated Unit Testing + Human-Computer Interaction (HCI) testing.

Unit Testing: Focus on high-value logic (ViewModels, DataSources) with **90%+ coverage on core components**.

HCI Testing: Task-based scenarios (e.g., "Find a show") to validate Views, intuitiveness and visual hierarchy.

Synergy: Unit tests ensure data integrity; HCI tests ensure "snappy" feel and usability.

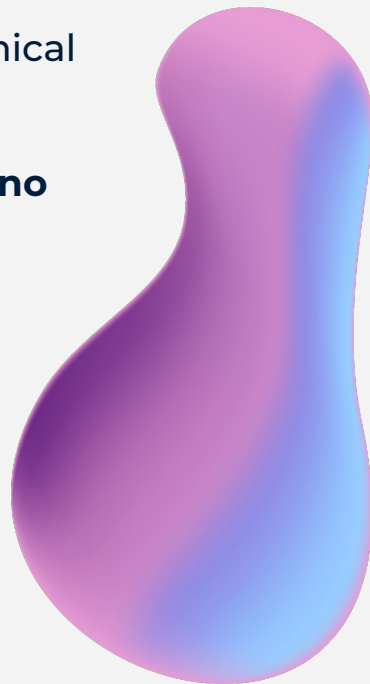
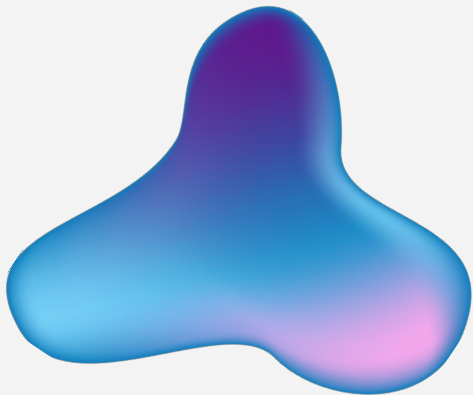
Name	Coverage	Executable Lines
> Show.swift	100,0%	133
> GenreDataSource.swift	100,0%	70
> ShowPoster.swift	100,0%	68
> StatCard.swift	100,0%	66
> DateAndRating.swift	100,0%	34
> ReviewDataSource.swift	100,0%	26
> TVRecommendations.swift	100,0%	22
> TVSearch.swift	100,0%	22
> StatisticsService.swift	100,0%	21
> Genre.swift	100,0%	13
> MainDataSource.swift	100,0%	11
> TMDbClient.swift	100,0%	6
> TVSeason.swift	100,0%	3
> Episode.swift	99,0%	100
> SearchViewModel.swift	98,6%	71
> QRScannerViewModel.swift	98,2%	55
> SkeneApp.swift	97,7%	43
> SettingsViewModel.swift	97,2%	71
> DatabaseManager.swift	96,8%	277
> StatisticsCalculator.swift	96,7%	122
> NotificationManager.swift	95,7%	92
> ImageURLTransform.swift	95,2%	21
> HomeView.swift	93,8%	581
> SeasonDataSource.swift	93,3%	224
> TVDetails.swift	92,3%	65
> LibraryViewModel.swift		139
> StatisticsViewModel.swift		
> Review.swift		
> ImageLoader.swift		

Test Coverage
> 90%

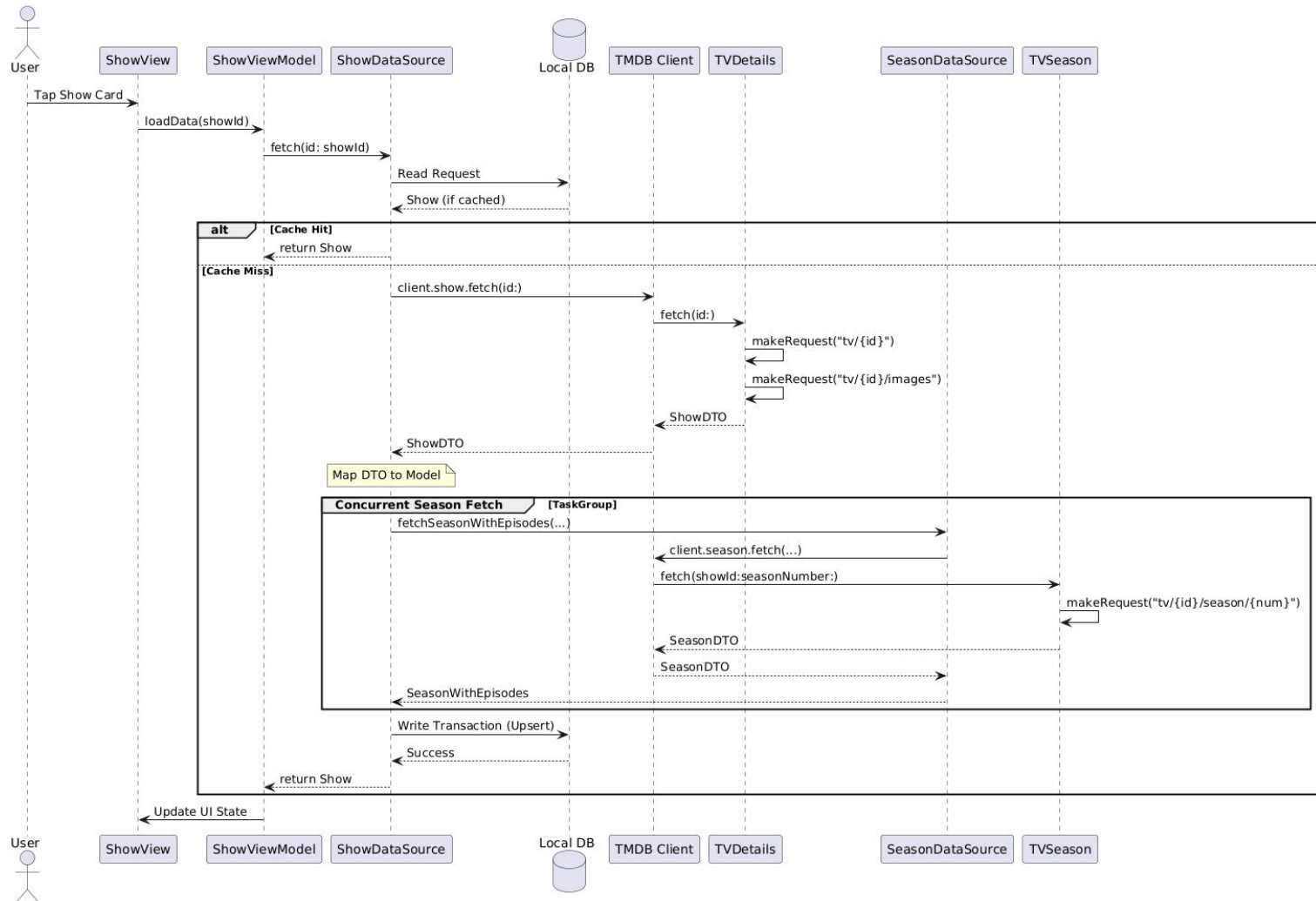
Conclusion & Acknowledgments

Special thanks to **Bending Spoons** and **Giancarlo Sorrentino** for technical supervision and architectural guidance.

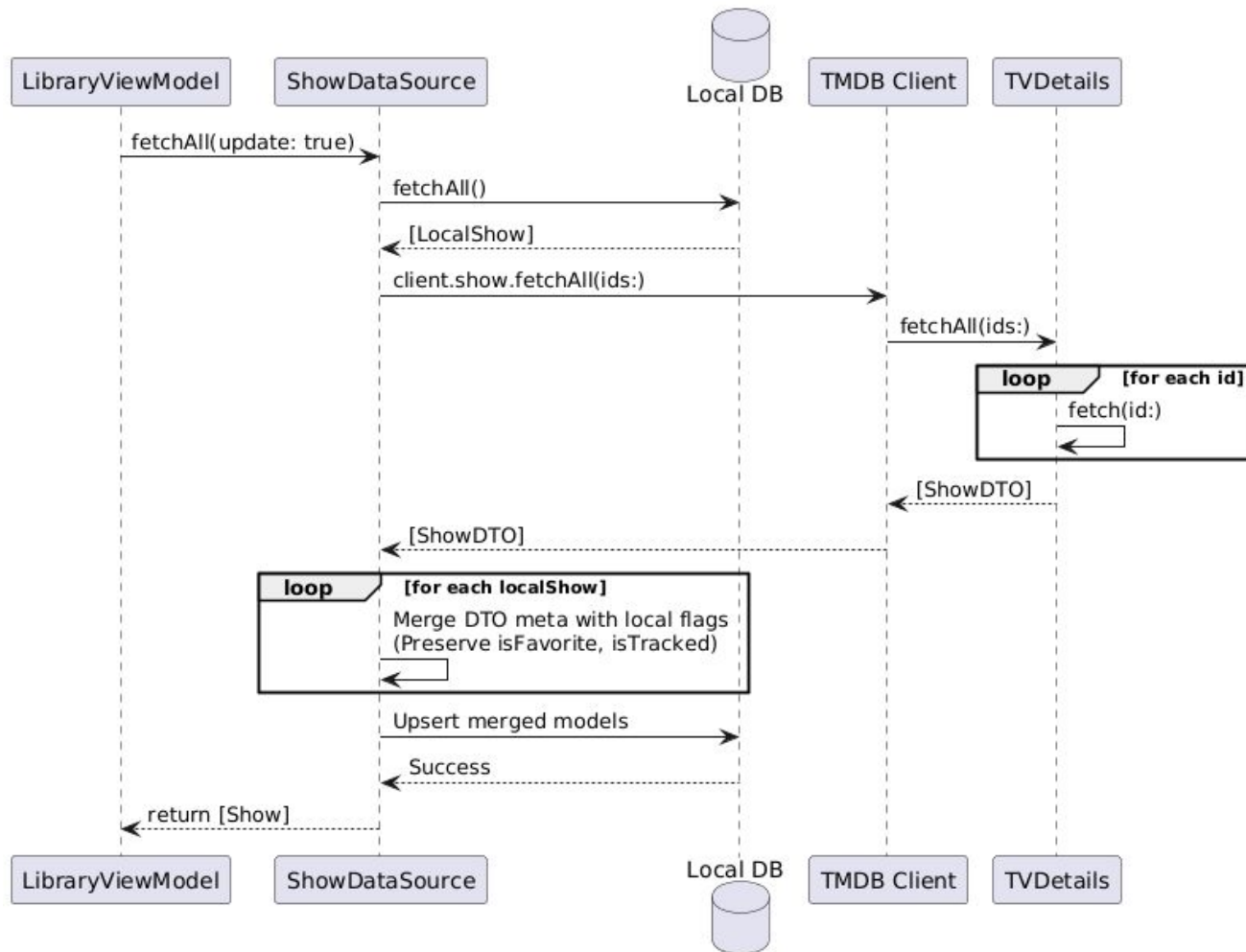
We would also like to express our sincere gratitude to **Professor Luciano Baresi**, who made this project possible.



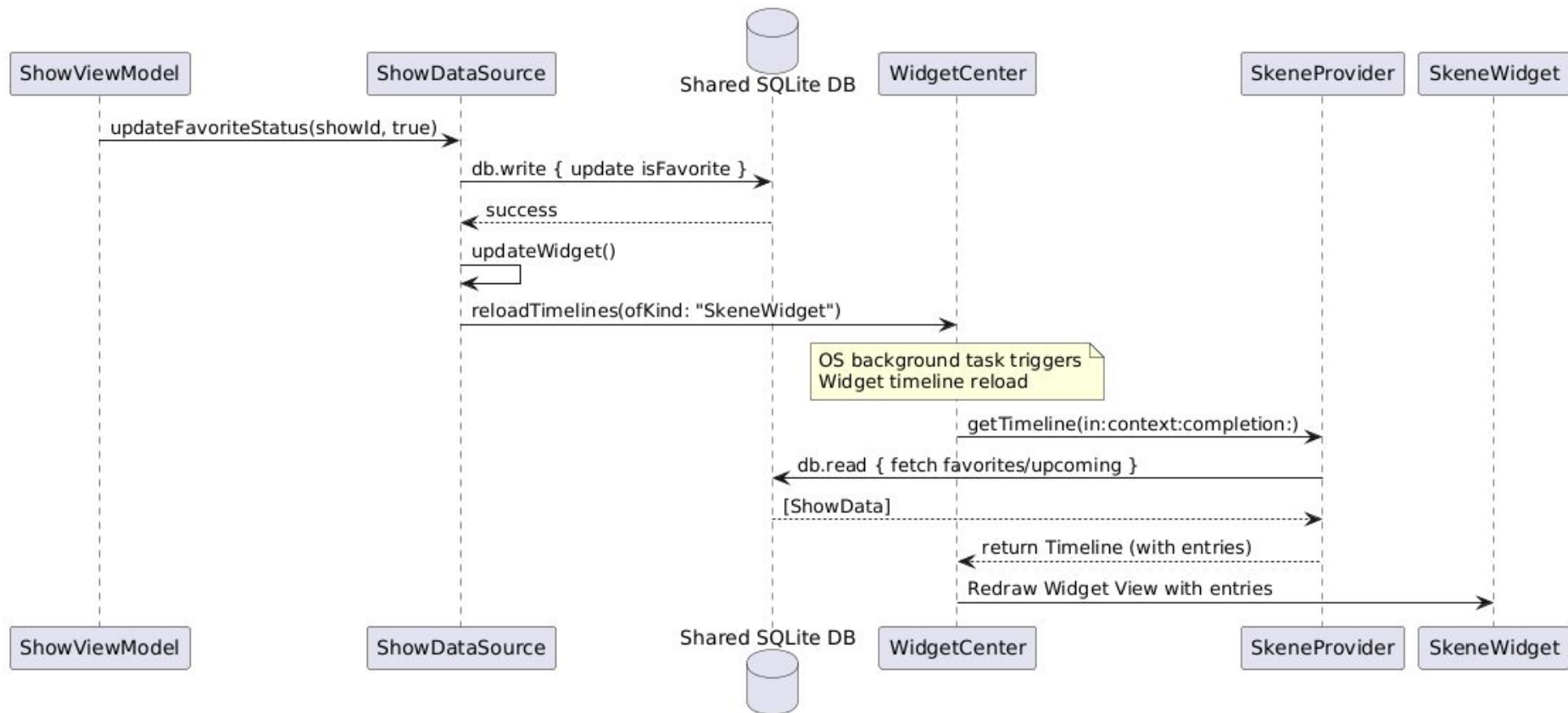
Sequence: Fetching TV Show Details



Sequence: Library Update & Metadata Sync



Sequence: Widget Update Trigger



Sequence: Sharing Library (User-to-User)

