

Сервис для стриминга и селфхостинга музыки. iOS-приложение

Курсовой проект на 2 курсе ПИ ФКН ВШЭ.

Выполнил: **Александр Васюков**.

Ссылки на демонстрацию:

1. YouTube: <https://www.youtube.com/watch?v=pOcgYHRTsvI>
2. Яндекс Диск: <https://disk.yandex.com/i/j0eRyUg4JNrqtg>

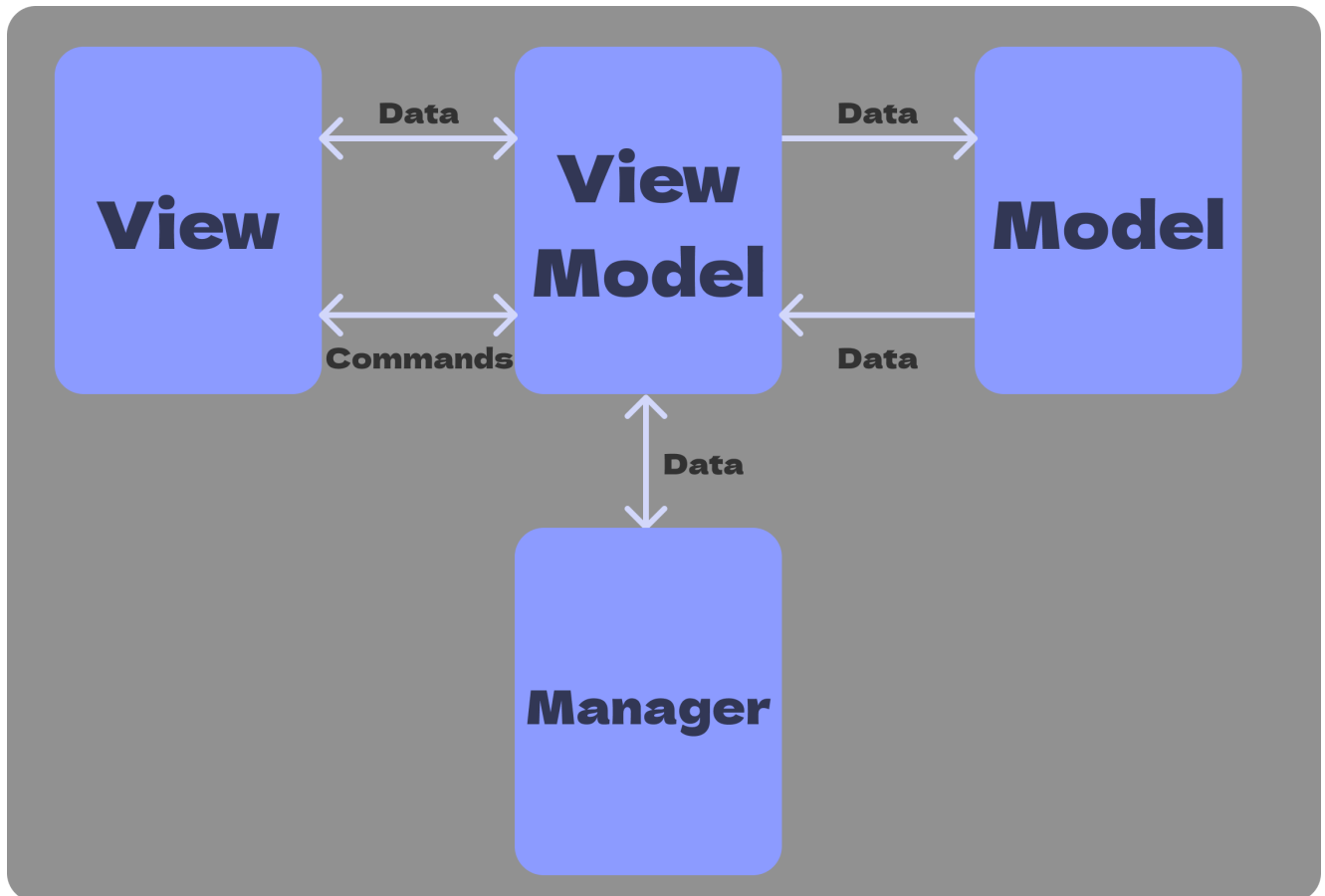
1. Функциональные преимущества

- **Хранение музыки:** Пользователи могут хранить треки как локально на устройстве, так и на сервере. Это обеспечивает гибкость и независимость от сторонних стриминговых сервисов.
- **Оффлайн-режим:** Возможность прослушивания музыки без интернет-соединения, а также локальное сохранение изменений с последующей синхронизацией при восстановлении подключения. Также есть автоматическая загрузка треков в кеш перед прослушиванием.
- **Функционал плеера:** Поддержка очереди воспроизведения, перемешивание треков, повтор трека, а также удобное управление через мини-плеер свайпами и полноценный экран плеера.
- **Удобный поиск:** Поиск треков по названию и исполнителю как среди локальных файлов, так и на сервере.
- **Управления музыкой:** Создание плейлистов, добавление треков, удаление треков с подтверждением действий, добавление треков в кеш.
- **Переключение между режимами online и offline:** Пользователь может переключаться между режимами и корректно взаимодействовать с приложением.
- **Страницы артистов:** Можно переходить на страницы артистов, где отображаются все треки данного артиста.

2. Технические преимущества

- **MVVM + Manager:** Использование архитектурного паттерна MVVM с сервисным слоем обеспечивает чистоту кода, масштабируемость и удобство работы. Разделение логики на Model, ViewModel, View и менеджеры упрощает поддержку и расширение функционала. Менеджеры (MusicManager, MusicPlayerManager, NetworkManager) позволяют отдельно работать с хранением музыки, воспроизведением треков, сервером, соблюдая принципы SOLID, поддерживая чистоту кода и масштабируемость.
- **Реактивное программирование Combine:** Делает удобной и эффективной обработку асинхронных операций, таких как сетевые запросы и обновление интерфейса.

- **UI:** Интерфейс реализован с использованием UIKit, что обеспечивает высокую производительность и кастомизацию элементов (мини-плеер, ячейки треков).
- **Работа с музыкой:** Надежное воспроизведение аудиофайлов, поддержка потокового стриминга и оффлайн-режима.
- **Обработка изображений через CoreImage:** Динамическое определение основного цвета обложки трека для создания эстетичного интерфейса для фона страницы исполнителя.



3. Преимущества перед аналогами

- **Отсутствие рекламы и ограничений:** В отличие от многих стриминговых сервисов (Яндекс.Музыка, Spotify), проект не содержит рекламы и позволяет бесплатно прослушивать музыку без ограничений.
- **Селфхостинг и приватность:** Пользователи полностью контролируют свои данные, что отличает проект от популярных сервисов.
- **Удобный интерфейс:** По сравнению с аналогами для селфхостинга (Navidrome, Ampache), приложение предлагает интуитивно понятный и стильный интерфейс.
- **Поддержка iOS:** У большинства приложений для селфхостинга музыки отсутствует поддерживаемое iOS-приложение, а данный проект как раз предназначен под операционную систему iOS.

Функция	Яндекс Музыка	Spotify	ВК Музыка	YouTube Music	Apple Music	Our Project
Скачивание треков на устройство	-	-	-	-	-	+
Нет рекламы	-	-	-	-	+	+
Бесплатное прослушивание без ограничений	-	-	-	-	-	+
Возможность использования в России	+	-	+	-	+	+
Добавление своих треков	+	-	+	-	+	+
Итого	2	0	2	0	3	5

Таблица 1. Сравнение функциональных характеристик со стриминговыми сервисами

Функция	Black Candy	Navidrome	Ampache	mStream	Our Project
Удобный интерфейс	-	+	-	-	+
Кеширование	-	+	+	-	+
Поиск треков	-	+	+	-	+
Страницы артистов	-	+	+	-	+
Наличие приложения под iOS	-	-	-	-	+
Итого	0	4	3	0	5

Таблица 2. Сравнение функциональных характеристик с сервисами для селфхостинга музыки

4. Экономические и социальные преимущества

- **Независимость:** Пользователи из России, как основные потребители, могут беспрепятственно слушать музыку, даже если треки удалены или заблокированы в стриминговых сервисах.
- **Персонализация:** Возможность загружать собственные треки и создавать плейлисты без ограничений.

5. Качество реализации

- **Чистая и модульная структура:** Код разделен на логические модули (модели, ViewModel, менеджеры), что упрощает чтение и поддержку.
- **Примеры реализаций:**
 - Использование Combine для обработки сетевых запросов.
 - Реализация плеера с поддержкой очереди и управления треками.
 - Динамическая загрузка и кеширование треков и изображений.