МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

**КОЛЛЕДЖ МНОГОУРОВНЕВОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ РОССИЙСКОЙ АКАДЕМИИ НАРОДНОГО ХОЗЯЙСТВА И ГОСУДАРСТВЕННОЙ СЛУЖБЫ ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

**на разработку pet-проекта**

**«Музыкальная социальная сеть».**

Исполнитель: Махмудова Аминат Д.

**Москва**

**2025**

Содержание

[**1. Краткое описание проекта (суть)** 3](#_Toc209973913)

[**2. Основные требования (Функциональные)** 3](#_Toc209973914)

[2.1. Пользовательские роли 3](#_Toc209973915)

[2.2. Базовый функционал (MVP) 3](#_Toc209973916)

[2.3. Нефункциональные требования 4](#_Toc209973917)

[**3. Предложенный стек технологий (рекомендация для сдачи в сроки)** 4](#_Toc209973918)

[**4. Рекомендуемые инструменты разработки** 4](#_Toc209973919)

[**5. Архитектура и данные** 5](#_Toc209973920)

[5.1. Архитектурная схема (вкратце) 5](#_Toc209973921)

[5.2. Структура БД 5](#_Toc209973922)

[**6. API / Endpoints (пример для собственного backend)** 5](#_Toc209973923)

[**7. UI/UX: основные экраны + требования** 6](#_Toc209973924)

[**8. Тестирование** 6](#_Toc209973925)

[**9. Релиз** 6](#_Toc209973926)

[**10. Безопасность и приватность** 7](#_Toc209973927)

# **1. Краткое описание проекта (суть)**

Название мобильного приложения: FlerkthSound

Описание: Мобильное приложение, представляющее собой социальную сеть для музыкантов и меломанов. Платформа позволяет пользователям загружать и слушать музыку, создавать плейлисты, объединяться в группы по интересам (жанрам), общаться через комментарии и личные сообщения, а также получать персональные рекомендации.

Цель: создать уникальную платформу, которая объединит музыкантов и любителей музыки в одном пространстве.

Аналоги: SoundCloud, Spotify, Bandcamp.

Дата сдачи —08.12.2025

# **2. Основные требования (Функциональные)**

## 2.1. Пользовательские роли

* **Слушатель:** регистрация/вход, просмотр ленты, прослушивание музыки, создание плейлистов, комментирование, лайки, вступление в группы, личные сообщения.
* **Музыкант:** все возможности Слушателя + загрузка/удаление своих треков, управление своим артист-профилем.
* **Модератор:** все возможности Слушателя + модерация контента (треков, комментариев, групп), блокировка пользователей.
* **Администратор:** полный доступ к управлению пользователями, ролями, всем контентом и системными настройками.

## 2.2. Базовый функционал (MVP)

1. **Аутентификация:** регистрация и вход по email/паролю.
2. **Профиль пользователя:** аватарка , имя, описание, роль, плейлисты, добавленные треки), список загруженных треков.
3. **Лента/Главная:** персонализированная лента с новыми треками от подписок и рекомендациями.
4. **Система рекомендаций:** базовые рекомендации на основе прослушиваний и лайков.
5. **Плеер:** базовые функции воспроизведения (play/pause, перемотка, следующий/предыдущий трек).
6. **Плейлисты:** создание, редактирование, публикация плейлистов.
7. **Группы:** создание групп , вступление в группы, публикация контента в группе(сообщения, фото, файлы), групповой чат/обсуждения.
8. **Социальное взаимодействие:** лайки трекам, добавление плейлистов, комментарии к трекам.
9. **Личные сообщения:** обмен текстовыми сообщениями между пользователями.
10. **Поиск:** по трекам, музыкантам, плейлистам и группам.
11. **Панель модератора/администратора:** просмотр и управление пользователями, треками, комментариями, группами.

## 2.3. Нефункциональные требования

* **Платформа:** Android 8.0+ (API 26+), так как выбран Android Studio + Kotlin
* **Время отклика UI:** < 200 мс на основные экраны (при наличии сети).
* **Безопасность:** HTTPS, хранение токенов в Android Keystore, валидация данных на стороне backend.
* **Конфиденциальность:** Политика конфиденциальности, возможность удаления аккаунта и данных.
* Локализация: RU (+EN опционально).
* **Производительность:** Плавное воспроизведение аудио, кеширование часто прослушиваемых треков.
* **Доступность:** Поддержка динамического шрифта, достаточная контрастность.

# **3. Предложенный стек технологий (рекомендация для сдачи в сроки)**

* **Frontend (мобильное):** Kotlin + Android Studio.
* **Backend:** Node.js (Express) + MySQL. (Соответствует моему выбору, но требует больше времени на настройку)
* **Альтернатива (для ускорения разработки):** **Firebase (Authentication, Firestore, Cloud Functions, Cloud Storage)**. Позволяет быстро собрать MVP без написания backend-кода, обеспечивает готовые решения для аутентификации, базы данных и хранения файлов.
* **Хостинг:** Heroku, DigitalOcean или аналоги для backend (если не Firebase).
* **CI/CD:** GitHub Actions для автоматических сборок APK.

# **4. Рекомендуемые инструменты разработки**

* **IDE:** Android Studio.
* **Дизайн:** Figma — прототипы и дизайн-макеты или Android Studio.
* **Контроль версий:** GitHub.
* **Backend-менеджмент:** Firebase Console (если выбран Firebase) или Postman для тестирования API.
* **База данных:** MySQL Workbench или phpMyAdmin.

# **5. Архитектура и данные**

## 5.1. Архитектурная схема (вкратце)

Мобильное приложение (Kotlin) ↔ Backend API (Node.js/Express) ↔ База данных (MySQL). Аутентификация через JWT-токены. Для хранения аудиофайлов и аватарок — облачное хранилище (например, AWS S3, Firebase Storage или локальный диск сервера).

## 5.2. Структура БД

* **users:** { id, email, password\_hash, name, bio, avatar\_url, role (listener/artist/moder/admin), is\_verified, created\_at }
* **tracks:** { id, artist\_id, title, description, genre, audio\_file\_url, cover\_art\_url, duration, plays\_count, is\_public, created\_at }
* **playlists:** { id, user\_id, title, description, cover\_art\_url, is\_public, created\_at }
* **playlist\_tracks:** { playlist\_id, track\_id, order } // Связь многие-ко-многим
* **groups:** { id, name, description, owner\_id, avatar\_url, is\_public, created\_at }
* **group\_members:** { group\_id, user\_id, role (member/moderator) }
* **group\_posts:** { id, group\_id, user\_id, content (text), track\_id (optional), created\_at } // Посты в группе
* **comments:** { id, user\_id, track\_id, text, created\_at }
* **likes:** { user\_id, track\_id (optional), playlist\_id (optional), created\_at }
* **follows:** { follower\_id, following\_id } // Подписки
* **conversations:** { id, created\_at }
* **participants:** { conversation\_id, user\_id }
* **messages:** { id, conversation\_id, sender\_id, text, created\_at }

# **6. API / Endpoints (пример для собственного backend)**

* POST /auth/register
* POST /auth/login
* GET /users/profile
* GET /tracks/feed // Лента
* GET /tracks/search?q=...
* POST /tracks/upload (для музыкантов)
* GET /tracks/:id
* POST /tracks/:id/like
* POST /tracks/:id/comment
* GET /playlists
* POST /playlists
* GET /groups
* POST /groups (для модераторов/админов)
* POST /groups/:id/join
* GET /conversations
* POST /conversations/:id/messages

# **7. UI/UX: основные экраны + требования**

1. **Экран входа/регистрации.**
2. **Главная (Лента):** список новых треков от подписок и рекомендаций.
3. **Поиск:** вкладки "Треки", "Исполнители", "Плейлисты", "Группы".
4. **Плеер:** полноэкранный плеер с обложкой, кнопками управления, прогресс-баром и секцией комментариев.
5. **Библиотека:** вкладки "Мои треки" (для музыкантов), "Мои плейлисты", "Подписки".
6. **Профиль:** личные данные, аватар, список треков/плейлистов, кнопка редактирования.
7. **Экран группы:** описание, участники, лента постов группы, кнопка "Вступить".
8. **Чаты:** список диалогов и экран самого чата.
9. **Панель модератора/админа:** списки для модерации (пользователи, треки, комментарии).
10. **Wireframe:** Минималистичный дизайн с акцентом на контент. Крупные обложки альбомов, читаемые списки. Темная тема как предпочтительная для музыкальных приложений.

# **8. Тестирование**

* **Unit-тесты:** для критической бизнес-логики (например, валидации данных).
* **Интеграционные тесты:** ключевые сценарии ("зарегистрироваться → загрузить трек → найти трек через поиск → оставить комментарий").
* Регрессионное тестирование перед сдачей.
* **Тестирование на реальных устройствах** (Android)**.**

# **9. Релиз**

* Репозиторий: Git (GitHub), feature-branch workflow.
* CI/CD: GitHub Actions для сборки APK при пуше в главную ветку.
* Автоматизированные сборки для тестовых релизов (Internal testing).
* Финальный релиз: Google Play / RuStore.

# **10. Безопасность и приватность**

* Все запросы по HTTPS.
* Пароли хэшируются (bcrypt).
* Возможно – аутентификация через JWT.
* Все запросы по HTTPS.
* Ограничение загрузки файлов (size/type), сканирование расширений.
* Возможность удаления аккаунта / данных.

**11. Роадмап / план по дням с конкретными датами**

**1 спринт (7 дней): 22.09.2025 – 28.09.2025**

**Задачи:**

1. Разработка технического задания (ТЗ) для приложения "Музыкальная социальная сеть".
2. Разработка структуры базы данных.
3. Разработка прототипов интерфейсов в Figma: экраны авторизации, регистрации, главной ленты, профиля и тд.

**Доставка:** Утвержденное ТЗ, схема БД, набор прототипов экранов в Figma.

**2 спринт (7 дней): 29.09.2025 – 05.10.2025**

**Задачи:**

1. Реализация экранов авторизации и регистрации в приложении.
2. Наполнение базы данных начальными тестовыми данными (пользователи, жанры).
3. Реализация MainActivity с базовой навигацией (например, нижняя панель вкладок).

**Доставка:** Приложение с рабочим процессом регистрации/входа, основной навигацией и локальной БД с тестовыми данными.

**3 спринт (14 дней): 06.10.2025 – 24.10.2025**

**Задачи:**

1. Создание репозитория GitHub и выгрузка текущего проекта.
2. Связь проекта Android Studio с GitHub.
3. Реализация экрана профиля пользователя (просмотр (и может редактирование)).
4. Реализация экрана загрузки трека (для музыкантов).
5. Разработка экрана главной ленты для отображения треков.

**Доставка:** Проект на GitHub, работающие экраны профиля, загрузки треков и ленты.

**4 спринт (14 дней): 27.10.2025 – 09.11.2025**

**Задачи:**

* 1. Реализация базового аудиоплеера (воспроизведение, пауза).
  2. Добавление функционала "Лайков" к трекам.
  3. Реализация системы подписок на других пользователей.
  4. Создание функционала плейлистов (создание, добавление трека).

**Доставка:** Приложение с работающим плеером, системой лайков, подписками и плейлистами.

**5 спринт (14 дней): 10.11.2025 – 23.11.2025**

**Задачи:**

1. Реализация системы комментариев к трекам.
2. Добавление возможности поиска треков и пользователей.
3. Улучшение интерфейса на основе принципов юзабилити (улучшение отображения ленты, плеера).
4. Можно сделать интеграцию с Firebase для аналитики или уведомлений (опционально).

**Доставка:** Приложение с комментариями, поиском и улучшенным интерфейсом.

**6 спринт (14 дней): 24.11.2025 – 07.12.2025**

**Задачи:**

1. Тестирование приложения, исправление критических ошибок.
2. Подготовка презентации / демонстрации (скриншоты, видео).
3. Подготовка пакета сдачи: zip-архив с проектом, APK-файл, инструкции по запуску, тестовые учетные записи.
4. Финальная полировка и подготовка к защите.

**Доставка:** Стабильная версия приложения, полный пакет документов для сдачи, презентация.

**7 спринт (7 дней): 08.12.2025 – 14.12.2025**

**Задача:** Защита индивидуального проекта.

**Доставка:** Успешная защита проекта.

**12. Уникальные функции проекта (если останется время)**

* Пуш-уведомления с настройками (типы уведомлений).
* Темная тема.
* Донаты музыкантам или подписка.
* Темная/светлая тема.