

План реализации проекта «Плейлист-Пас»

Диаграмма Гранта — План реализации

Период	Февр. 2026	Март 2026	Апр. 2026	Май 2026	Июнь 2026
Недели	6-7	8-10	11-14	15-18	19-23

Этап 0: Подготовка и анализ (Февр. 2026, недели 6-7)

Результаты:

- Видение и ценностное предложение проекта
- Бизнес-процессы (БП-1 до БП-6)
- Пользовательские истории и требования
- Глоссарий и словарь терминов
- Анализ интеграций с API

Исполнитель: Проектный менеджер, аналитик

Этап 1: Архитектор веб-системы (Март 2026, недели 8-10)

Результаты:

- Архитектура фронтенд и бэкенд компонентов
- Описание REST API интерфейсов для управления комнатами и плейлистами
- Перечень модулей системы:
 - Управление сессиями (создание/удаление комнат)
 - Поиск и добавление треков (интеграция Yandex музыка /VK музыка)
 - Управление плейлистом (CRUD операции)
 - Управление воспроизведением
 - Экспорт плейлиста в стриминг-сервисы

- Структуры данных (Room, Track, User, Playlist)
- Основные сценарии взаимодействия (QR-код сканирование, синхронизация в реальном времени)
- Описание HTTP запросов (GET, POST, PUT, DELETE)
- Диаграмма компонентов (React/Vue компоненты, Node.js сервис, базы данных)
- Диаграмма последовательности (сценарии создания комнаты, подключения гостя, добавления трека)
- Диаграмма размещения (веб-браузер, веб-сервер, API сервисы)
- Swagger OpenAPI документация для API

Исполнитель: Архитектор системы, техлид

Этап 2: Разработка (Апр. 2026, недели 11-14)

Результаты:

- Программный код фронтенда (React/Vue.js для веб-приложения)
- Программный код бэкенда (Node.js + Express для REST API)
- Интеграция с Yandex, VK API
- Инструкция по сборке и запуску
- Набор программных модулей (service слой, repository слой, компоненты UI)
- Описание переменных окружения (.env конфигурация)
- Git-репозиторий с историей коммитов

Исполнитель: Frontend разработчик, Backend разработчик (2-3 человека)

Этап 3: Тестирование (Апр.-Май 2026, недели 13-16)

Результаты:

- Описание системных тестовых сценариев (все БП-1 до БП-6)
- Программный код для Unit-тестирования (Jest, Vitest для фронта)
- Автоматические Unit-тесты (покрытие 70%+ критических функций)
- Описание действий пользователя для каждого сценария
- Описание найденных багов и их исправления
- Git-репозиторий с тестами

Тестовые сценарии:

1. Создание комнаты и генерация QR-кода
2. Сканирование QR-кода и подключение гостя
3. Поиск и добавление трека из Yandex /VK
4. Перетаскивание и изменение порядка треков
5. Удаление трека из плейлиста
6. Управление воспроизведением (play, pause, next)
7. Экспорт плейлиста хостом в Spotify
8. Сохранение плейлиста гостем в свой аккаунт

Исполнитель: QA-инженер, тестировщик

Этап 4: Развертывание и инфраструктура (Май 2026, недели 16-19)

Результаты:

- Docker конфигурация (Dockerfile для фронта и бэка)
- Docker Compose для локального запуска всей системы
- Набор скриптов для развертывания (build, deploy, rollback)
- Инструкции по установке на хостинг (Heroku/DigitalOcean/AWS)
- Настройка CI/CD пайплайна (GitHub Actions или GitLab CI)
- Документация по развертыванию в продакшн
- Конфигурация переменных окружения для разных сред

Исполнитель: DevOps/Системный администратор

Этап 5: Документирование и UX (Май 2026, недели 18-21)

Результаты:

- Руководство пользователя (как начать работу)
- Инструкция для хоста (создание комнаты, управление плейлистом)
- Инструкция для гостя (подключение, добавление треков)
- Руководство по использованию каждого музыкального сервиса
- FAQ раздел
- Описание всех ошибок и способов их решения
- Видеотutorialы (опционально)

Исполнитель: Технический писатель, UX-специалист

Этап 6: Финальная проверка и запуск (Май-Июнь 2026, недели 21-23)

Результаты:

- Полное интеграционное тестирование всех БП
- Финальная проверка документации и инструкций
- Тестирование на разных браузерах и устройствах
- Оптимизация производительности
- Развертывание в продакшн
- Официальный запуск проекта
- Сдача всех артефактов клиенту

Исполнитель: QA-инженер, проектный менеджер

Ключевые компоненты системы

Фронтенд (React/Vue.js)

- Главная страница с кнопкой «Создать комнату»
- Интерфейс хоста: просмотр QR-кода, управление плейлистом, управление воспроизведением
- Интерфейс гостя: сканирование QR-кода, поиск и добавление треков, просмотр плейлиста
- Компонент поиска треков с интеграцией Spotify/YouTube

Бэкенд (Node.js + Express)

- REST API для управления комнатами
- REST API для управления плейлистами и треками
- WebSocket или Server-Sent Events для синхронизации в реальном времени
- Интеграция с Yandex API (поиск, создание плейлиста, экспорт)
- Интеграция с VK API (поиск видео/музыки)
- Управление сессиями и авторизацией

База данных

- Таблица Rooms (комнаты)
 - Таблица Playlists (плейлисты)
 - Таблица Tracks (треки)
 - Таблица Users (пользователи/гости)
-

График разработки по неделям

Неделя	Февраль	Март	Апрель	Май	Июнь
6-7	Этап 0				
8-10		Этап 1			
11-12			Этап 2		
13-14			Этап 2		
15-16			Этап 3		
17-18				Этап 3-4	
19-21				Этап 4-5	
22-23					Этап 6

Критические пути и риски

Критические задачи (Critical Path)

1. Интеграция Yandex музыка /VK музыка API (неделя 11-12)
2. Реализация WebSocket синхронизации (неделя 12-13)
3. Тестирование функциональности экспорта (неделя 14-15)

Потенциальные риски

- Задержки в получении доступа к API Yandex музыки /VK музыки (требуется проверка)
- Сложность с синхронизацией плейлиста в реальном времени
- Совместимость QR-кода на разных мобильных браузерах

Меры по снижению рисков

- Начать процесс интеграции в начале проекта
- Использовать существующие библиотеки для QR-кода

