

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет Компьютерных наук

Кафедра программирования и информационных технологий

МОБИЛЬНОЕ ДЕЙТИНГ-ПРИЛОЖЕНИЕ “VIBE” С ИНТЕГРАЦИЕЙ  
МУЗЫКАЛЬНОГО ИИ

Курсовая работа

Направление: 09.03.02 Информационные системы и технологии

Зав. Кафедрой \_\_\_\_\_ д. ф.-м. н, доцент С.Д. Махортов

Руководитель \_\_\_\_\_ ст. преподаватель В.С. Тарасов

Руководитель практики \_\_\_\_\_ ассистент М.А. Савин

Обучающийся \_\_\_\_\_ Д.О. Кравченко, 3 курс, д/о

Обучающийся \_\_\_\_\_ Д.И. Щеблыкин, 3 курс, д/о

Обучающийся \_\_\_\_\_ Е.В. Баулин, 3 курс, д/о

Обучающийся \_\_\_\_\_ Д.С. Черных, 3 курс, д/о

Обучающийся \_\_\_\_\_ А.В. Кущенко, 3 курс, д/о

Воронеж 2025

## СОДЕРЖАНИЕ

|   |    |
|---|----|
| СОДЕРЖАНИЕ .....  | 2  |
| Термины и определения .....                                       | 5  |
| ВВЕДЕНИЕ.....   | 7  |
| 1 Постановка задачи.....  | 9  |
| 1.1 Цели создания приложения.....                                 | 9  |
| 1.2 Задачи приложения .....                                       | 9  |
| 2 Целевая аудитория .....   | 10 |
| 3 Обзор конкурентов.....  | 11 |
| 3.1 Tinder .....  | 11 |
| 3.2 Badoo .....   | 12 |
| 3.3 ВКонтакте Знакомства.....                                     | 13 |
| 4 Требования к приложению.....                                    | 15 |
| 4.1 Требования к приложению в целом .....                         | 15 |
| 4.2 Функциональные требования .....                               | 15 |
| 4.2.1 Регистрация и аутентификация .....                          | 15 |
| 4.2.2 Управление профилем и генерация музыки .....                | 15 |
| 4.2.3 Механизм лайков, скипов и мэтчинга .....                    | 16 |
| 4.2.4 Чат.....  | 16 |
| 4.2.5 Демонстрационный режим для неавторизованных пользователей.. | 16 |
| 4.3 Нефункциональные требования .....                             | 16 |
| 4.3.1 Производительность:.....                                    | 16 |
| 4.3.2 Надёжность и отказоустойчивость .....                       | 17 |
| 4.3.3 Безопасность.....   | 17 |
| 4.3.4 Масштабируемость .....                                      | 18 |
| 4.3.5 Юзабилити и адаптивность.....                               | 18 |
| 4.3.6 Поддержка и сопровождение.....                              | 18 |
| 4.3.7 Лицензирование и патентная чистота.....                     | 18 |
| 4.3.8 Соответствие правовым нормам .....                          | 19 |
| 5 Моделирование системы .....                                     | 20 |
| 5.1 Диаграмма прецедентов (Use Case).....                         | 20 |
| 5.2 Диаграмма состояний .....                                     | 21 |

|       |  |    |
|-------|--|----|
| 5.3   | Диаграмма классов.....                 | 21 |
| 5.4   | Диаграмма развертывания .....          | 23 |
| 5.5   | Диаграмма последовательностей .....    | 23 |
| 6     | Реализация.....                        | 27 |
| 6.1   | Средства реализации.....               | 27 |
| 6.1.1 | React (TypeScript) .....               | 27 |
| 6.1.2 | PostgreSQL .....                       | 28 |
| 6.1.3 | Spring Framework .....                 | 28 |
| 6.1.4 | Docker.....                            | 28 |
| 6.1.5 | Suno API.....                          | 29 |
| 6.2   | Архитектура.....                       | 29 |
| 6.2.1 | Компоненты серверной части.....        | 29 |
| 6.2.2 | Контейнерная среда .....               | 30 |
| 6.3   | Клиентская часть.....                  | 30 |
| 6.3.1 | Вход/регистрация.....                  | 31 |
| 6.3.2 | Лента мэтчей .....                     | 35 |
| 6.3.3 | Страница мэтчей .....                  | 36 |
| 6.3.4 | Страница чатов.....                    | 38 |
| 6.3.5 | Страница профиля пользователя.....     | 39 |
| 6.3.6 | Страница настройки приложения .....    | 40 |
| 6.3.7 | Страница редактирования описания ..... | 41 |
| 6.3.8 | Страница Vibe Premium.....             | 43 |
| 6.4   | Серверная часть.....                   | 44 |
| 6.4.1 | Хранение данных .....                  | 44 |
| 6.4.2 | Генерация музыки.....                  | 44 |
| 6.4.3 | Механизм мэтчинга .....                | 44 |
| 6.4.4 | Чаты.....                              | 44 |
| 6.4.5 | Аутентификация и авторизация .....     | 44 |
| 6.4.6 | Модерация .....                        | 45 |
| 7     | Тестирование .....                     | 45 |
| 8     | Аналитика .....                        | 47 |
| 8.1   | Методология сбора данных.....          | 47 |

|   |    |
|---|----|
| 8.2 Общие показатели посещаемости ..... | 47 |
| 8.3 Источники трафика .....             | 48 |
| 8.4 Мониторинг производительности ..... | 51 |
| ЗАКЛЮЧЕНИЕ .....                        | 53 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....  | 54 |

## Термины и определения

— Match – ситуация, когда два пользователя поставили друг другу «лайк», что открывает доступ к чату;

— Администратор – человек, имеющий доступ к расширенному функционалу веб-сервиса;

— Скип – ситуация, когда пользователь поставил «дизлайк»;

— Дейтинг (Dating) – действие пользователя, заключающееся в просмотре профиля другого пользователя и принятии решения о постановке «лайка» или «скипа»;

— Премиум-подписка – платная услуга приложения «Vibe», предоставляющая неограниченное количество действий (свайпов) и генерацию музыкальных треков;

— Suno API – внешний программный интерфейс, используемый для генерации уникальных музыкальных треков на основе пользовательских данных, интегрированный в серверную часть приложения «Vibe»;

— PWA (Progressive Web App) – прогрессивное веб-приложение, которое обеспечивает кроссплатформенную работу приложения «Vibe» в браузере и возможность установки на мобильные устройства как нативного приложения;

— Чат – функционал приложения, позволяющий пользователям, у которых произошел мэтч, обмениваться текстовыми сообщениями и фотографиями в реальном времени;

— Демонстрационный режим – режим работы приложения для неавторизованных пользователей, позволяющий просматривать 5 шаблонных профилей с демонстрационными данными и музыкой для ознакомления с функционалом;

— Профиль – набор данных пользователя, включающий имя, возраст, город, пол, описание «О себе», фотографию и связанный музыкальный

трек, используемый для представления пользователя в системе и взаимодействия с другими участниками.

## ВВЕДЕНИЕ

В современном мире, где стремительно развивается цифровизация, дейтинг-приложения стали привычным способом знакомства. Однако пользователи всё чаще ищут что-то большее, чем просто стандартный профиль, им нужна платформа для самовыражения и возможность найти партнера на более глубоком уровне. Традиционные текстовые анкеты не всегда могут передать всю индивидуальность человека, что усложняет поиск настоящей эмоциональной связи.

Приложение «Vibe» предлагает совершенно новый подход к знакомствам, фокусируясь на музыке как способе самовыражения. Вместо привычных текстов, главной особенностью профиля становится уникальный музыкальный фрагмент. С помощью специального модуля SUNO AI приложение генерирует короткий 20-секундный аудиофрагмент, который отражает личность пользователя на основе его описания «О себе». Такой инновационный подход позволяет установить более глубокую эмоциональную связь между людьми, делая процесс поиска совпадений интуитивно понятным и увлекательным.

Для тех, кто ещё не зарегистрировался, предусмотрен демонстрационный режим, где можно ознакомиться с пятью заранее подготовленными шаблонными профилями, чтобы понять, как работает приложение. Это помогает привлечь новых пользователей и стимулирует их к регистрации.

Актуальность разработки «Vibe» заключается в том, что она отвечает на растущий спрос на более индивидуализированные и эмоционально насыщенные платформы для онлайн-знакомств. В условиях, когда многие существующие сервисы предлагают схожий функционал, «Vibe» выделяется за счёт своей уникальной концепции, основанной на музыке и искусственном интеллекте.

Целью данной курсовой работы является разработка мобильного дейтинг-приложения «Vibe», которое интегрирует музыкальный

искусственный интеллект для объединения людей через сгенерированные уникальные музыкальные треки.

Задачи:

— Внедрить алгоритмы искусственного интеллекта для генерации уникального музыкального трека, который будет служить ключевым элементом профиля пользователя и поможет выразить его индивидуальность через аудио;

— Сформировать механизм мэтчинга, основанный на музыкальном «вайбе», который объединит участников для установления глубокой эмоциональной связи и позволит им общаться в чате;

— Стимулировать вовлечённость пользователей, ограничивая бесплатное использование приложения до 20 дейтингов в день, с возможностью перехода на безлимитный режим по платной подписке, создавая тем самым баланс между доступностью и монетизацией.



## **1 Постановка задачи**

### **1.1 Цели создания приложения**

- Автоматизация знакомств через уникальный музыкальный подход, повышающий эмоциональную вовлеченность пользователей;
- Персонализация поиска партнеров с помощью ИИ-генерации музыкальных треков на основе профиля пользователя;
- Монетизация через премиум-подписку, снимающую ограничения на ежедневные взаимодействия.

### **1.2 Задачи приложения**

Приложение позволяет решать следующие задачи:

- Создание персонализированного профиля с уникальным музыкальным треком, сгенерированным ИИ на основе данных пользователя (описание "О себе", имя, возраст);
- Поиск и знакомство с другими пользователями через прослушивание их музыкальных треков и систему взаимных лайков;
- Ограниченное бесплатное взаимодействие (до 20 лайков в день) с возможностью расширения функционала через премиум-подписку;
- Общение в чате после взаимного мэтча с возможностью обмена текстовыми сообщениями и фотографиями;
- Демонстрационный режим для неавторизованных пользователей с просмотром 5 шаблонных профилей;
- Управление профилем: редактирование информации, повторная генерация музыки, смена фотографии;
- Модерация контента администратором (просмотр, блокировка пользователей, контроль за нарушениями).

## 2 Целевая аудитория

Проект ориентирован на людей, ищущих знакомства и отношения, сфокусированных на общих музыкальных предпочтениях. Приложение разработано для облегчения процесса поиска партнера и предлагает следующие удобства и инструменты, соответствующие потребностям этой категории пользователей:

- Мэтчи по музыкальным предпочтениям (вайбам), что позволяет находить друг друга на основе музыкальных вкусов;

- Интуитивно понятный интерфейс, чтобы сосредоточиться на поиске людей через музыку.

Целевая аудитория Vibe включает широкую возрастную категорию от 18 до 35+ лет.

- 18-24 года: студенты и молодые специалисты, ищущие новые знакомства и возможность флирта;

- 25-34 года: карьеристы, ориентированные как на серьезные отношения, так и на интересное общение;

- 35+ лет: более опытные пользователи, ориентированные на стабильные и серьезные отношения.

Именно сужение целевой аудитории до любителей музыки и функционал, ориентированный на данную категорию пользователей, делает приложение уникальным на рынке дейтинга.

### 3 Обзор конкурентов

Среди конкурентов приложения "Vibe" были рассмотрены самые крупные игроки на рынке дейтинг-приложений: Tinder, Badoo и ВКонтакте Знакомства.

Эти сервисы, в отличие от приложения "Vibe", ориентированы на более широкий круг пользователей и различные типы знакомств. Поскольку наше приложение фокусируется на уникальном подходе к подбору (по музыкальным предпочтениям), существующие дейтинг-сервисы являются косвенными конкурентами. Ниже приведён анализ каждого из трех основных конкурентов сервиса.

#### 3.1 Tinder

Tinder — один из крупнейших игроков рынка, доступный в более чем 190 странах. Алгоритмы подбора основаны на геолокации и лайках.

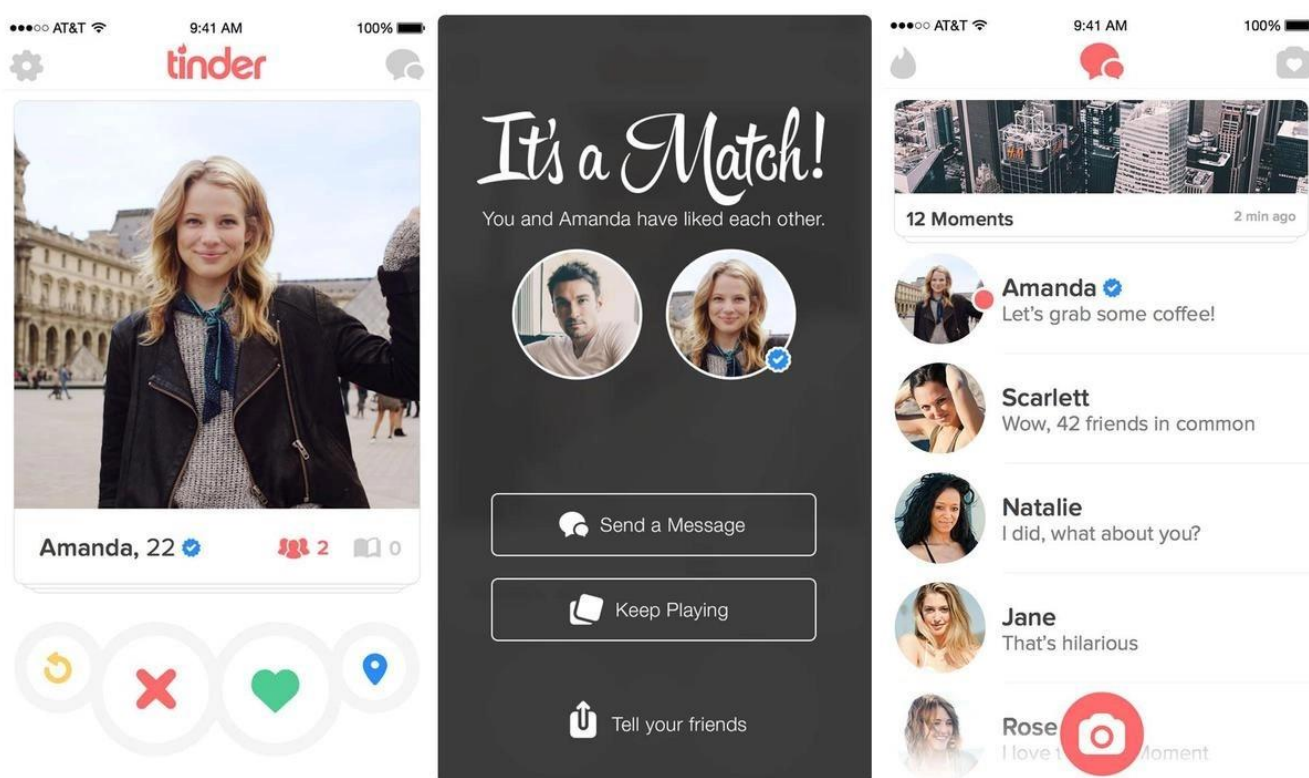


Рисунок 1 – Пример интерфейса приложения Tinder

Преимущества Tinder:

- Широкая известность и огромная база пользователей;
- Удобное мобильное приложение;
- Внедрение видеозвонков и усиленных мер безопасности.

Недостатки Tinder:

- Высокая конкуренция внутри сервиса;
- Ориентация на более поверхностные знакомства, основанные на фотографиях и коротких описаниях;
- Отсутствие основного акцента на глубоких общих интересах, таких как музыкальные предпочтения.

### 3.2 Badoo

Badoo — один из крупнейших дейтинг-сервисов, особенно популярный в Европе и Латинской Америке. Он предлагает возможность поиска по интересам и близости.

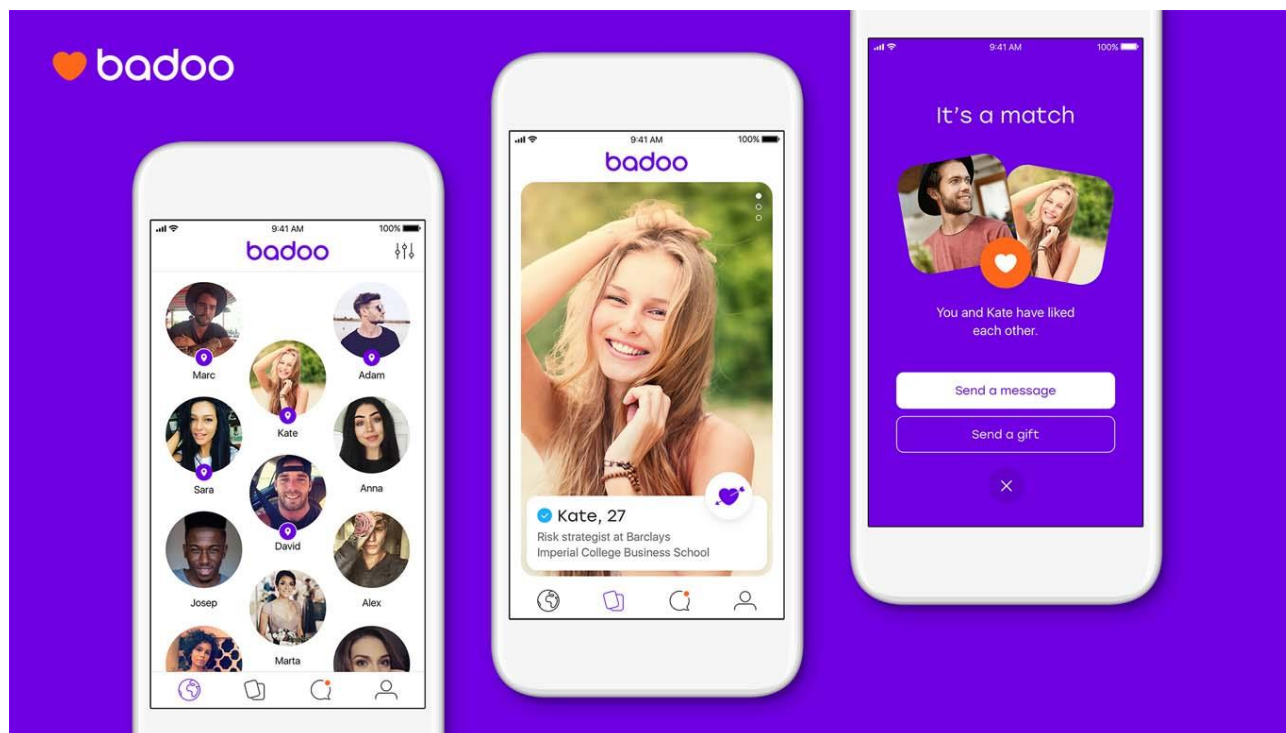


Рисунок 2 – Пример интерфейса приложения Badoo

Преимущества Badoo:

- Большая аудитория, подходит для широкой аудитории;

- Проверка профилей вручную для повышения безопасности;
- Возможность поиска по интересам и близости.

Недостатки Badoo:

- Может уступать по качеству подбора матчей, поскольку не всегда фокусируется на глубоких общих интересах;
- Ориентация на более широкий спектр знакомств, без уникального фокуса на музыкальных предпочтениях.

### 3.3 ВКонтакте Знакомства

ВКонтакте Знакомства — это сервис, интегрированный с социальной сетью ВКонтакте. Он использует встроенные алгоритмы рекомендаций на основе друзей и интересов.

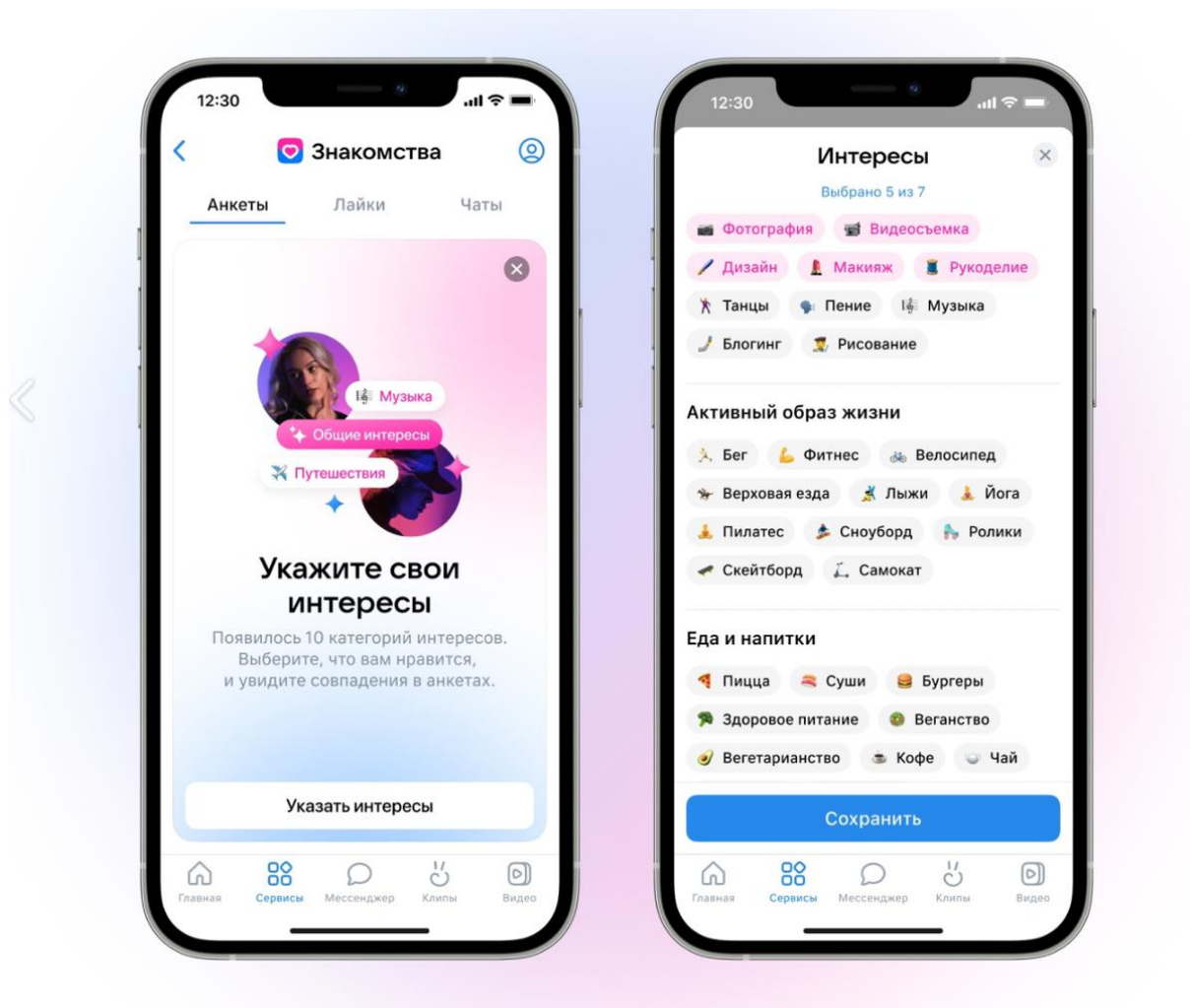


Рисунок 3 – Пример интерфейса приложения ВКонтакте Знакомства

Преимущества ВКонтакте Знакомства:

- Отличный охват за счет огромной аудитории ВКонтакте;
- Интеграция с социальной сетью, что может упростить начало общения.

Недостатки ВКонтакте Знакомства:

- Низкая монетизация;
- Сложность выхода за рамки экосистемы VK;
- Отсутствие уникального механизма подбора, такого как музыкальные предпочтения, который является основой "Vibe"

Эти сервисы, в отличие от разработанного проекта "Vibe", ориентированы на широкий спектр пользователей и различные типы знакомств. Исходя из анализа, "Vibe" обладает следующими преимуществами:

- Мэтчи по музыкальным предпочтениям (вайбам), что позволяет находить друг друга на основе своих музыкальных предпочтений;
- Более глубокое и эмоциональное соединение благодаря общим музыкальным интересам, в отличие от знакомств через фотографии и текстовые профили;
- Интуитивно понятный интерфейс, сосредоточенный на нахождении людей через музыку.

## **4 Требования к приложению**

### **4.1 Требования к приложению в целом**

- Кроссплатформенное PWA-приложение, работающее на Android (10+) и iOS (11.3+);
- Поддержка русского и английского языков;
- Минималистичный дизайн с адаптивным интерфейсом.

### **4.2 Функциональные требования**

#### **4.2.1 Регистрация и аутентификация**

- Пользователь проходит регистрацию через форму, содержащую обязательные поля: логин, пароль (не менее 6 символов), «О себе», возраст, пол, имя, город, загрузка фотографии;
- Верификация не предусмотрена на данном этапе;
- Вход в систему осуществляется с использованием логина и пароля.

#### **4.2.2 Управление профилем и генерация музыки**

- Пользователь может редактировать профиль, при этом изменение фотографии допускается отдельно от текстового описания «О себе»;
- При первом заполнении или изменении поля «О себе» система обращается к Suno API для генерации двух вариантов 20-секундных аудиотреков на основе данных профиля (описание, имя, возраст);
- Пользователь выбирает один из представленных треков.

### **4.2.3 Механизм лайков, скипов и мэтчинга**

- Пользователь просматривает карточки других участников, включающие фотографию, имя, возраст и аудиотрек;
- Предоставляются кнопки «Лайк» и «Скип»;
- Бесплатный режим ограничен 20 дейтингами (просмотрами) в день; при взаимном «лайке» между двумя пользователями возникает мэтч, открывающий доступ к чату;
- Платная подписка (299 руб./мес или 2700 руб./год) снимает ограничение и предоставляет неограниченный доступ.

### **4.2.4 Чат**

- После мэтча пользователи могут обмениваться текстовыми сообщениями и фотографиями;
- История переписки доступна пользователям.

### **4.2.5 Демонстрационный режим для неавторизованных пользователей**

- Гостям (неавторизованным пользователям) показывается 5 заранее подготовленных шаблонных профилей с демонстрационными данными и музыкой для ознакомления с функционалом приложения с дальнейшим предложением зарегистрироваться.

## **4.3 Нефункциональные требования**

### **4.3.1 Производительность:**

- Время отклика сервера не должно превышать 2 секунд при стандартной нагрузке;



- Генерация аудиотрека должна происходить быстро и без значительных задержек;
- Запросы к базе данных должны выполняться за минимальное время за счет оптимизации индексов и кэширования (Redis).

#### **4.3.2 Надёжность и отказоустойчивость**

- Система должна быть устойчивой к сбоям с автоматическим резервным копированием данных;
- Реализуется централизованный мониторинг с использованием инструментов, таких как Grafana и Prometheus.

#### **4.3.3 Безопасность**

Система должна обеспечивать надёжную защиту персональных данных пользователей и предотвращать несанкционированный доступ к конфиденциальной информации. Для этого реализуются следующие меры:

- Все данные передаются через защищённые соединения (SSL/TLS);
- Аутентификация и авторизация реализуются с помощью Spring Security 6 и JWT; пароли хранятся в зашифрованном виде (например, с использованием BCrypt);
- Механизмы защиты от CSRF, XSS, SQL-инъекций и ограничение количества неудачных попыток входа (brute-force защита);
- Доступ к конфиденциальной информации предоставляется только авторизованным пользователям, а все действия логируются для аудита.

#### **4.3.4 Масштабируемость**

- Система должна поддерживать горизонтальное масштабирование для работы с большим числом одновременных пользователей, с использованием контейнеризации Docker и Docker Compose.

#### **4.3.5 Юзабилити и адаптивность**

- Приложение должно обладать интуитивно понятным и адаптивным интерфейсом;
- Разработка происходит в формате PWA, что обеспечивает возможность работы в браузере и установки на мобильные устройства под управлением Android (с версии 10) и iOS (с версии 11.3);
- Интерфейс должен поддерживать мультиязычность (русский и английский).

#### **4.3.6 Поддержка и сопровождение**

- Система должна быть легко обновляемой и поддерживаемой, обеспечивая оперативное внесение изменений и устранение неисправностей без значительного простоя.

#### **4.3.7 Лицензирование и патентная чистота**

- Проект должен соответствовать нормам патентного законодательства;
- Не допускается использование лицензионно ограниченных компонентов без соответствующих разрешений;
- Все открытые библиотеки должны использоваться в соответствии с их лицензиями (MIT, Apache 2.0 и т. д.);

- Вся интеллектуальная собственность (код, музыка) принадлежит проекту или лицензируется корректно.

#### **4.3.8 Соответствие правовым нормам**

- Соблюдение GDPR и законов о персональных данных (для европейских пользователей);
- Учет требований локального законодательства в странах, где будет доступно приложение;
- Хранение данных пользователей в соответствии с международными стандартами.

## 5 Моделирование системы

### 5.1 Диаграмма прецедентов (Use Case)

Диаграмма прецедентов – диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей.

На рисунке 4 представлена диаграмма прецедентов для авторизованного пользователя.



Рисунок 4 – Диаграмма прецедентов. Авторизованный пользователь.

На рисунке 5 представлена диаграмма прецедентов для неавторизованного пользователя.

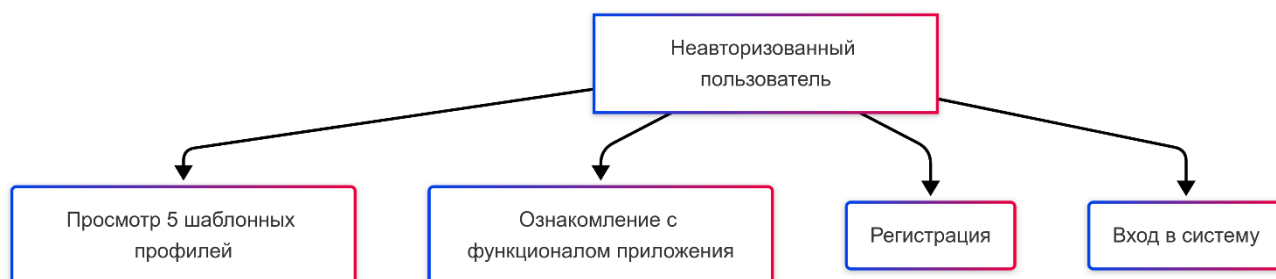


Рисунок 5 – Диаграмма прецедентов. Неавторизованный пользователь.

На рисунке 6 представлена диаграмма прецедентов для администратора.

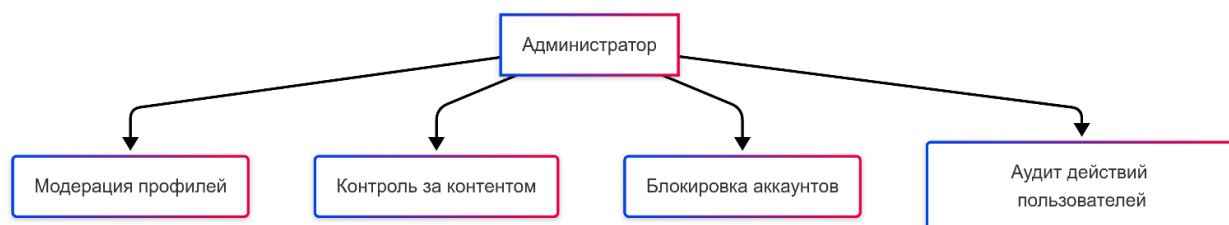


Рисунок 6 – Диаграмма прецедентов. Администратор.

## 5.2 Диаграмма состояний

Диаграмма состояний определяет последовательность состояний объекта, вызванных последовательностью событий. Данная диаграмма полезна при моделировании жизненного цикла объекта.

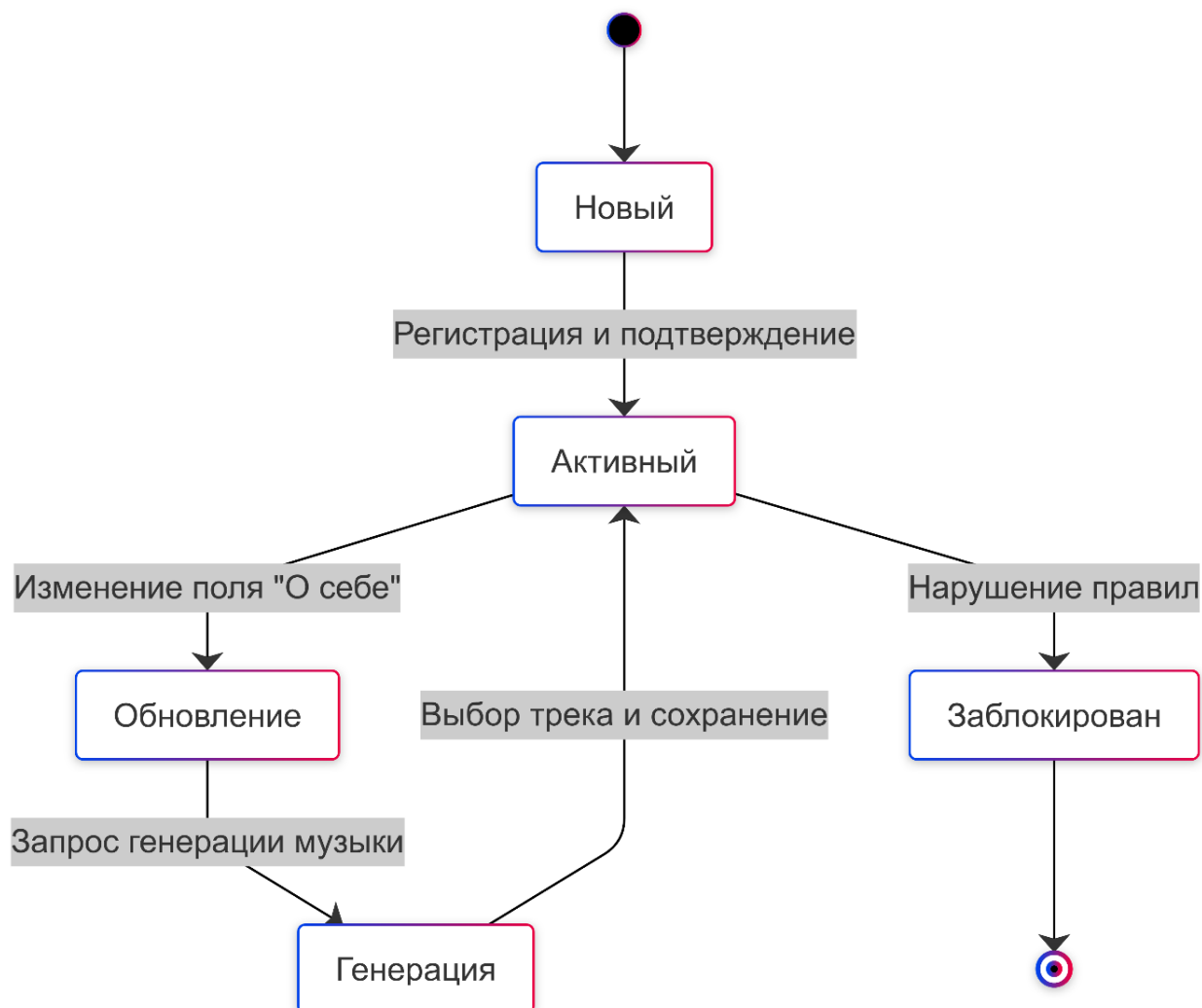


Рисунок 7 – Диаграмма состояний.

## 5.3 Диаграмма классов

Диаграмма классов – структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей (отношений) между ними.

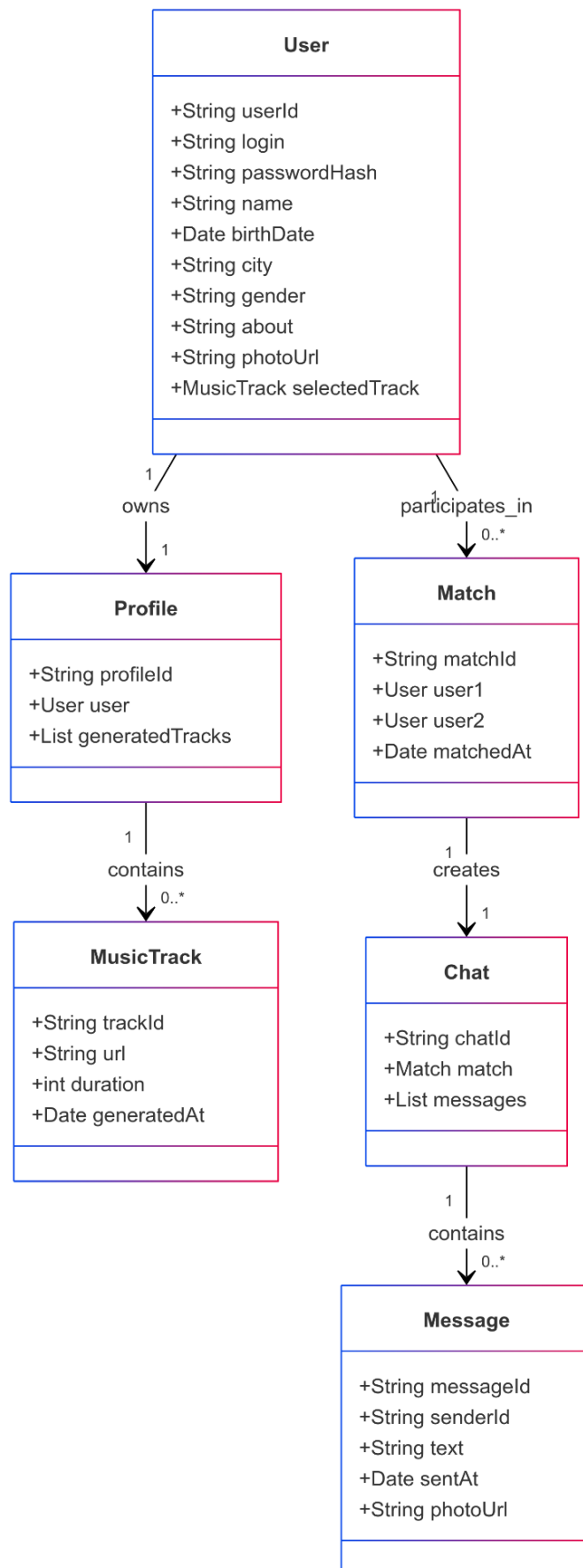


Рисунок 8 – Диаграмма классов.

## 5.4 Диаграмма развертывания

Диаграммы развертывания обычно используются для визуализации физического аппаратного и программного обеспечения системы. Она моделирует физическое развертывание артефактов на узлах.

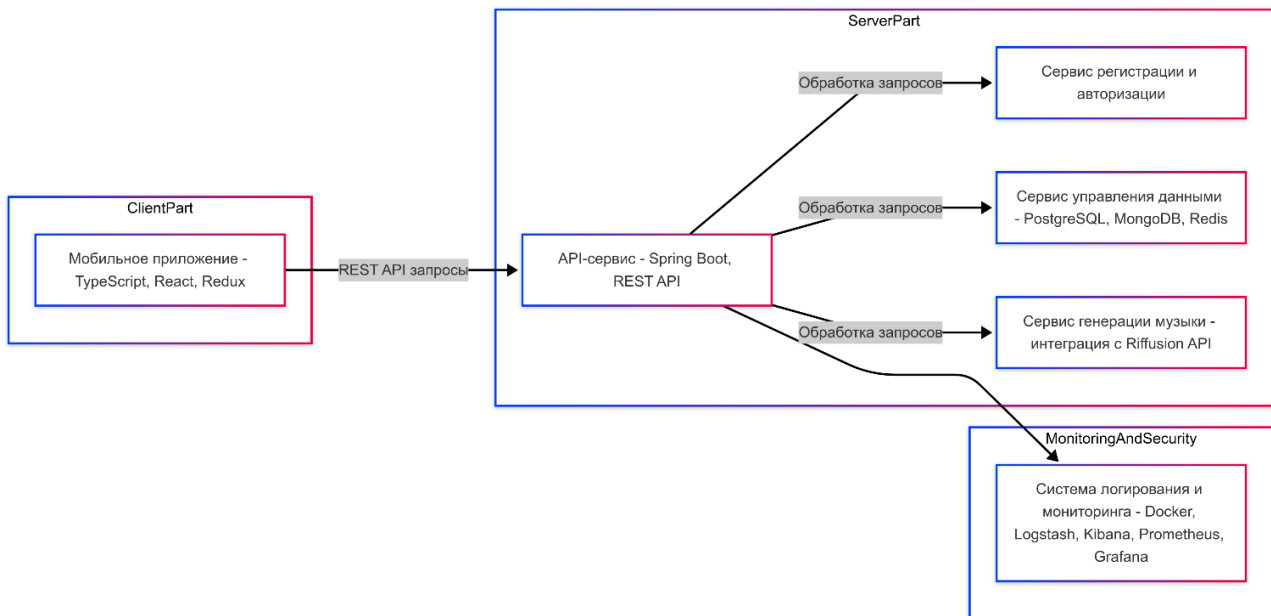


Рисунок 9 – Диаграмма развертывания

## 5.5 Диаграмма последовательностей

Диаграмма последовательности – UML-диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие акторов (действующих лиц) информационной системы в рамках прецедента.

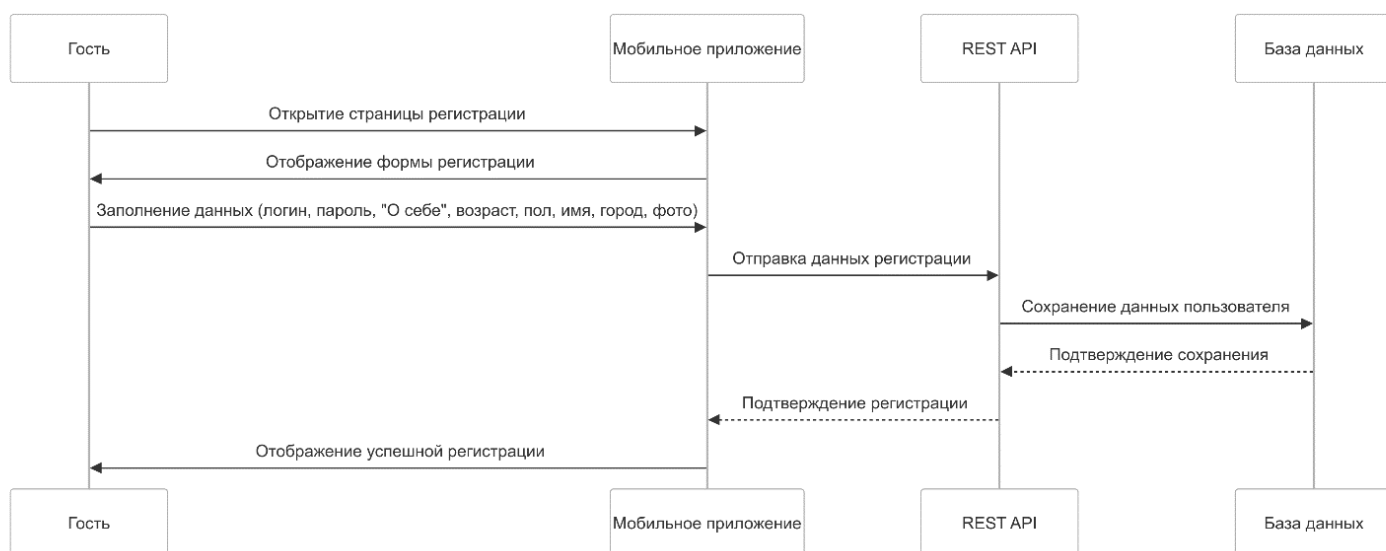


Рисунок 10 – Диаграмма последовательностей. Процесс регистрации

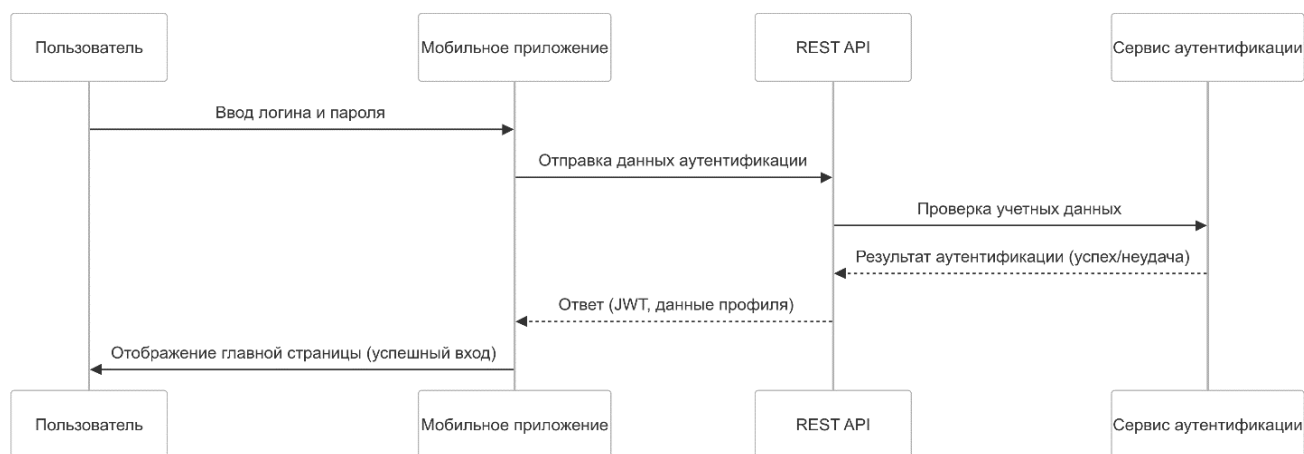


Рисунок 11 – Диаграмма последовательностей. Процесс входа в систему



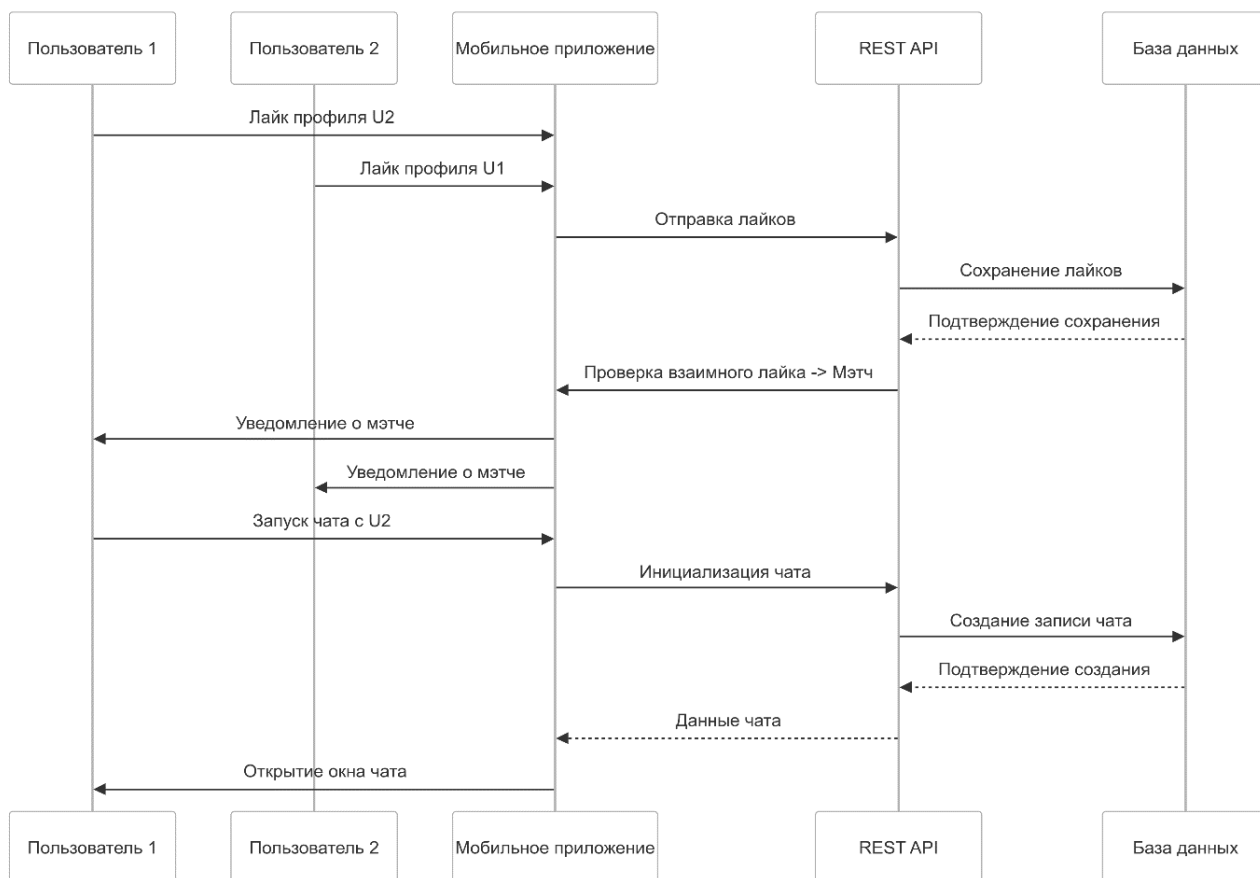


Рисунок 12 – Диаграмма последовательностей. Процесс мэтчинга и начала чата.

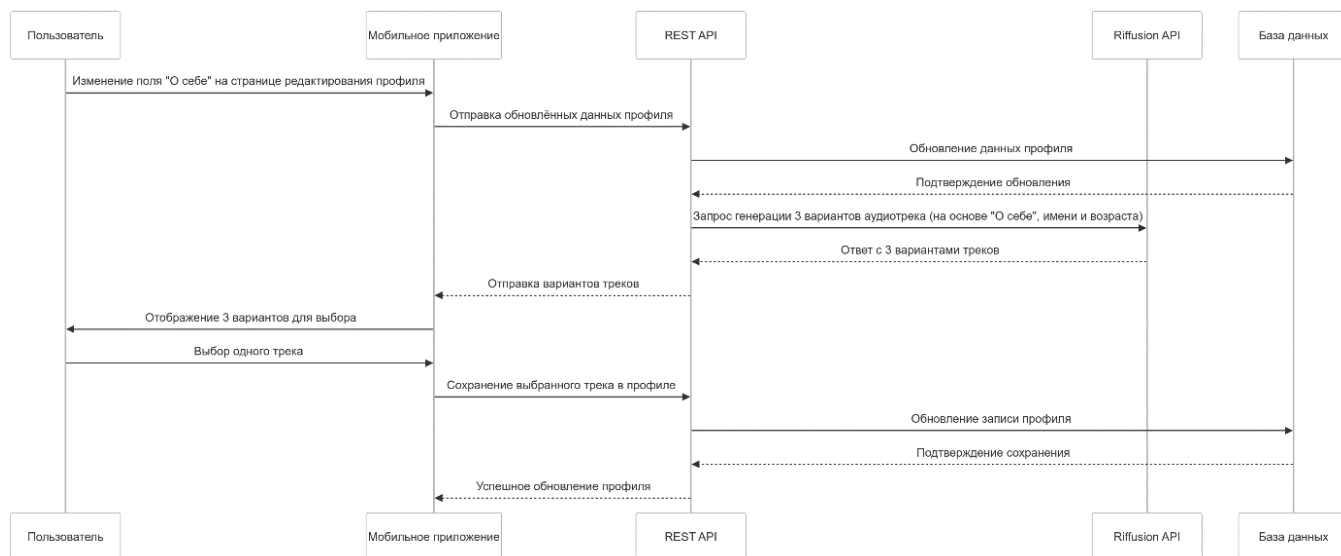


Рисунок 13 – Диаграмма последовательностей. Процесс обновления профиля (генерации музыки).

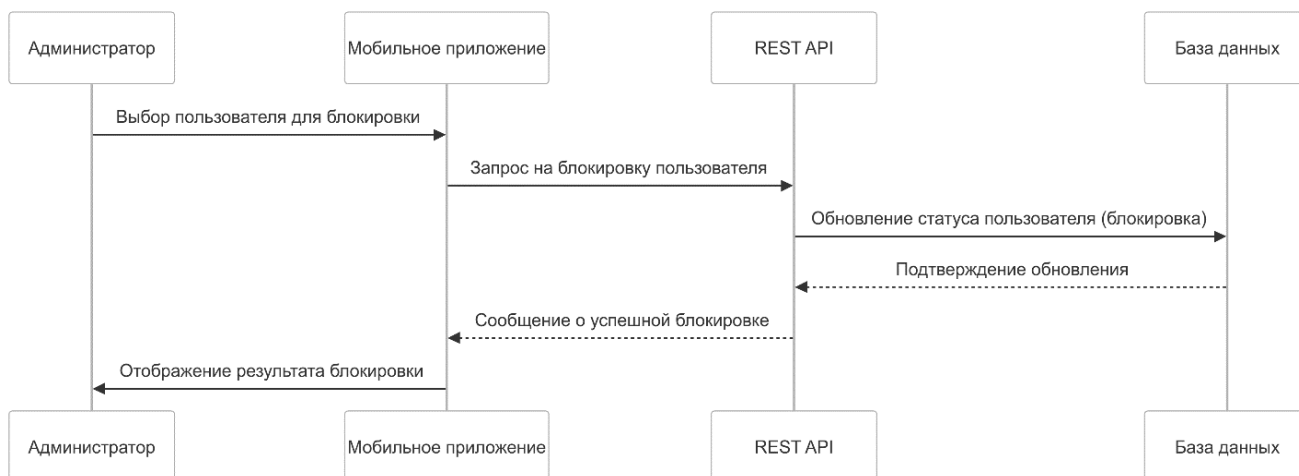


Рисунок 14 – Диаграмма последовательностей. Процесс блокировки пользователей (Для администратора).

## **6 Реализация**

Настоящая глава описывает процесс реализации мобильного приложения «Vibe», инновационного дейтинг-приложения с интеграцией музыкального ИИ, детализируя выбранные средства разработки, архитектурные решения и особенности серверной и клиентской частей. Приложение построено по клиент-серверной архитектуре с четким разделением функционала между back-end и front-end, взаимодействующими посредством REST API.

### **6.1 Средства реализации**

#### **6.1.1 React (TypeScript)**

Для разработки клиентской части приложения «Vibe» был выбран фреймворк React с использованием языка TypeScript. React обеспечивает создание динамического и отзывчивого пользовательского интерфейса, а TypeScript добавляет строгую типизацию, что повышает надежность и упрощает поддержку кода. Основные преимущества выбора React и TypeScript:

- Кроссплатформенность: React позволяет создавать прогрессивные веб-приложения (PWA), которые работают в браузере и могут быть установлены как нативные приложения на Android (с версии 10) и iOS (с версии 11.3);

- Компонентный подход: Модульная структура React упрощает разработку и повторное использование компонентов, таких как карточки профилей, чаты и навигационное меню;

- Быстрая разработка: Горячая перезагрузка (hot reloading) ускоряет процесс разработки, позволяя мгновенно видеть изменения в интерфейсе;

- Типобезопасность: TypeScript предотвращает множество ошибок на этапе компиляции благодаря строгой типизации.

### **6.1.2 PostgreSQL**

Для хранения и управления данными приложения «Vibe» выбрана реляционная система управления базами данных (СУБД) PostgreSQL. Эта СУБД обеспечивает:

- Надежность и ACID-совместимость: Гарантирует целостность и согласованность данных, что критично для хранения пользовательских профилей, мэтчей и истории чатов;

- Масштабируемость: Поддерживает высокую нагрузку и большие объемы данных, что позволяет приложению обрабатывать множество одновременных пользователей;

### **6.1.3 Spring Framework**

В качестве фреймворка для разработки серверной части (back-end) приложения "Vibe" использован Spring Framework. Основные преимущества:

- Быстрая разработка: Встроенные компоненты Spring, такие как Spring Security и Spring Data, ускоряют создание серверной логики;

- REST API: Spring Framework предоставляет мощные инструменты для создания масштабируемых RESTful API, обеспечивающих взаимодействие между клиентской и серверной частями;

- Безопасность: Встроенные механизмы защиты от CSRF, XSS, SQL-инъекций и других угроз обеспечивают надежность системы;

- Масштабируемость: Spring поддерживает горизонтальное масштабирование, что позволяет обрабатывать большое количество запросов.

Java используется как основной язык программирования для серверной части, обеспечивая высокую производительность и совместимость с экосистемой Spring.

### **6.1.4 Docker**

Для упаковки и развертывания компонентов приложения использован инструмент контейнеризации Docker. Преимущества:

— Изолированность: Каждый компонент (сервер, база данных, API) запускается в отдельном контейнере, что исключает конфликты зависимостей;

— Портируемость: Контейнеры легко переносятся между средами разработки, тестирования и продакшена;

— Упрощенное развертывание: Docker Compose позволяет управлять всеми сервисами с помощью единого конфигурационного файла.

### **6.1.5 Suno API**

Для генерации уникальных музыкальных треков на основе пользовательских данных (имя, возраст, описание «О себе») использован Suno API. Этот модуль позволяет создавать 20-секундные аудиофрагменты, которые становятся ключевым элементом профиля пользователя, обеспечивая уникальный «музыкальный вайб».

## **6.2 Архитектура**

Приложение «Vibe» построено по классической клиент-серверной архитектуре, обеспечивающей четкое разделение обязанностей между клиентской и серверной частями. Клиентская часть отвечает за пользовательский интерфейс и взаимодействие, а серверная часть управляет бизнес-логикой, данными и интеграцией с Suno API.

### **6.2.1 Компоненты серверной части**

Серверная часть приложения "Vibe" состоит из следующих ключевых компонентов:

— API (Spring REST): Обеспечивает маршрутизацию HTTP-запросов, валидацию данных и возврат ответов в формате JSON. Основные эндпоинты включают управление профилями, мэтчинг, чаты и генерацию музыки;

— Core (Инициализационный слой Spring): Отвечает за запуск сервисов, соединение с PostgreSQL, применение миграций и инициализацию логирования;

— Business Logic Modules: Модули Spring для управления пользователями (users), профилями (profiles), мэтчами (matches), чатами (chats) и аналитикой (analytics);

— DB (Spring Data JPA): Интерфейс для взаимодействия с PostgreSQL через Java-объекты, обеспечивающий управление моделями данных (User, Profile, Match, Chat);

— Logger: Центральная система логирования для мониторинга событий, ошибок и аудита;

— Utils: Вспомогательные функции для обработки данных, валидации и интеграции с Suno API.

### **6.2.2 Контейнерная среда**

Развертывание осуществляется с использованием Docker и Docker Compose. Контейнерная среда включает:

— Контейнер web: Spring-приложение с сервером (например, Tomcat) для обработки HTTP-запросов;

— Контейнер db: PostgreSQL для хранения данных;

— Контейнер nginx: Обратный прокси для обслуживания статических файлов и балансировки нагрузки;

— Контейнер swagger-ui: Интерфейс для документации API.

### **6.3 Клиентская часть**

Клиентская часть приложения «Vibe», разработанная на React с TypeScript, обеспечивает интуитивно понятный и адаптивный интерфейс. Основные аспекты реализации:

— Пользовательский интерфейс (UI): Дизайн выполнен в минималистичном стиле с использованием цветовой палитры, включающей бледно-розовый (#FFE8F4), мягкий красный (#FE6D87), черный (#1E1E1E) и белый (#FFFFFF). Основной шрифт — Roboto, дополнительный — Emilys Candy для логотипа;

— Управление состоянием: Используется библиотека (например, Redux или Context API) для управления состоянием приложения, обеспечивая предсказуемое поведение при работе с профилями, мэтчами и чатами;

— Взаимодействие с REST API: Клиентская часть отправляет HTTP-запросы к серверу для получения/отправки данных. Обработка ошибок сети реализована для стабильной работы приложения.

### **6.3.1 Вход/регистрация**

Страница доступна всем пользователям.

На странице «Вход» расположены:

- Название приложения;
- Поля для ввода почты и пароля, указанных при регистрации;
- Кнопка «Войти»;
- Кнопка «Войти как гость»;
- Кнопка «Зарегистрироваться»;

На странице «Зарегистрироваться» расположены: поля для ввода E-mail, пароля, подтверждения пароля и кнопка «Отправить код», которая перенаправляет пользователя на страницу «Ввод кода регистрации».

На странице «Ввод кода регистрации» расположено: информация об отправке кода на почту с указанием почты, на которую отправлен код, поле ввода 4-значного кода, кнопка «Отправить повторно». После успешного ввода кода идет перенаправление на страницу «Анкета» для завершения регистрации профиля.

На странице «Анкета» расположено:

- Поле для ввода имени;
- Поле для выбора даты рождения;
- Поле для выбора города;
- Поле для ввода информации «О себе»;
- Кнопки для выбора пола «Мужской/Женский»;

- Кнопка «Загрузить фото»;
- Кнопка «Отправить», которая перенаправляет на страницу «Окно загрузки».

На экране «Окно загрузки». в центре располагаются тематическая картинка и небольшая информация с просьбой подождать генерацию музыки.

Vibe

## Вход

E-mail

Введите логин

Обязательное поле

Пароль

Введите пароль

Войти

Войти как гость

Зарегистрироваться

Рисунок 15 – Страница входа





## Регистрация

E-mail

Введите логин

Пароль

Введите пароль

Подтверждение пароля

Введите пароль

Отправить код

Рисунок 16 – Страница регистрации



Мы отправили код для регистрации на вашу почту **user2004@gmail.com** .

Введите 4-значный код, указанный в электронном письме



Не получили электронное письмо?

[Отправить повторно](#)

Рисунок 17 – Страница подтверждения почты

**Анкета**

Имя

Дата рождения

Город

О себе

Пол

☐ Мужской ☐ Женский

+ Загрузить фото

Отправить

Рисунок 18 – Страница анкеты

### 6.3.2 Лента мэтчей

Страница доступна всем пользователям.

На странице расположено:

- Фото пользователя, попавшегося в ленте;
- Имя пользователя;
- Город пользователя;
- Кнопка для прослушивания трека пользователя, сгенерированного

ИИ по описанию профиля;

- Кнопка «Крестик» для отклонения анкеты;
- Кнопка «Сердечка» для оценки анкеты;

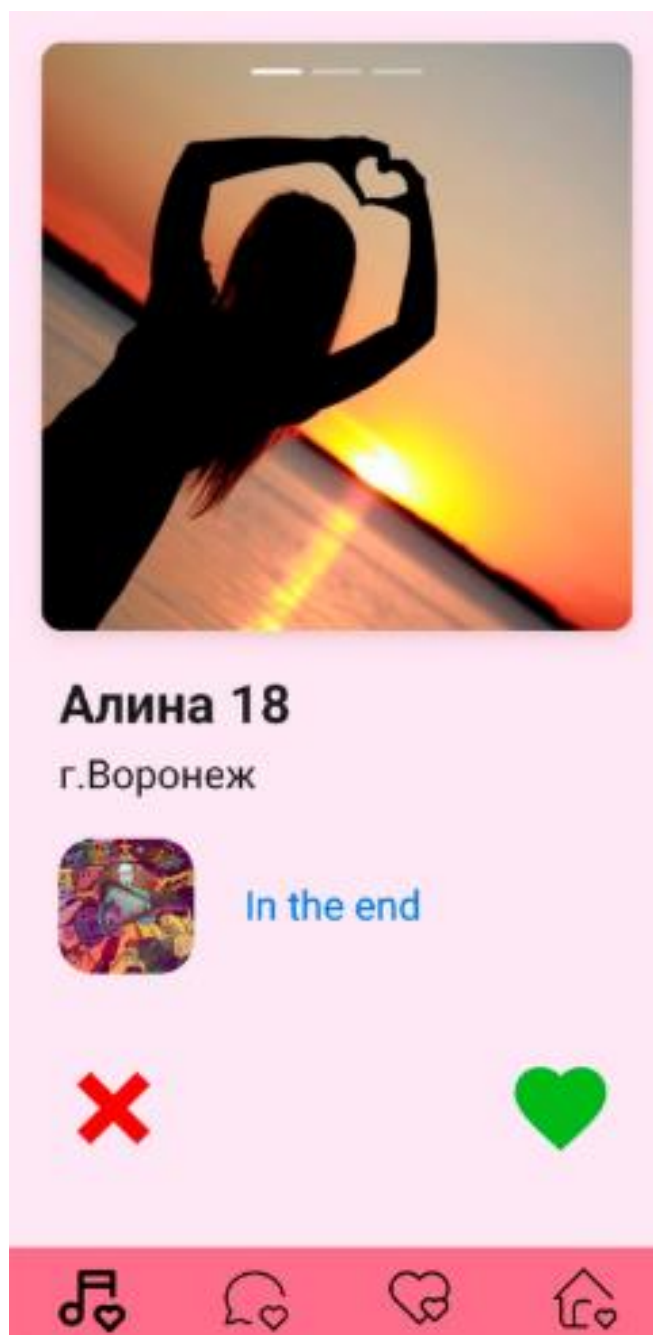


Рисунок 19 – Страница ленты мэтчей

### 6.3.3 Страница мэтчей

Страница доступна только авторизованным пользователям.

В шапке страницы расположен логотип приложения.

На странице расположен список пользователей, с которыми у вас произошел мэтч: фото пользователя, которое является также кнопкой для перехода в чат с пользователем, имя, кнопка трека, сгенерированного ИИ по описанию пользователя, кнопка «Крестик» для отмены мэтча с определенным пользователем.

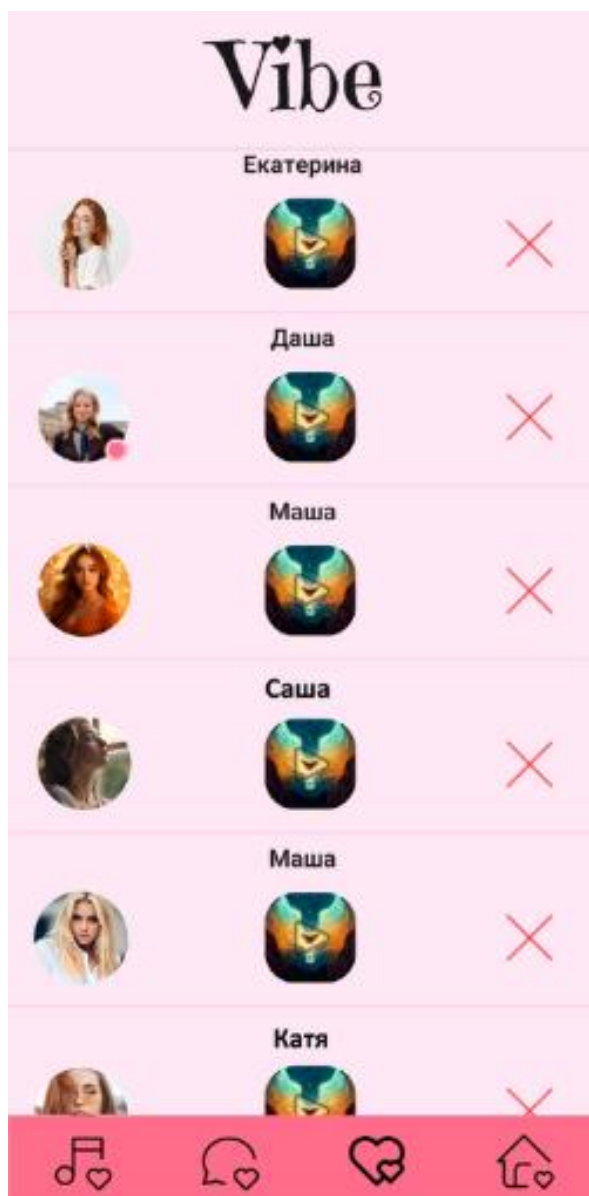


Рисунок 20 – Страница мэтчей

### 6.3.4 Страница чатов

Страница доступна только авторизованным пользователям.

В шапке страницы расположен логотип приложения.

На странице расположен список пользователей, с которыми есть диалог. Фотографии пользователей, имя -на которые можно нажать для переходы в личные сообщения.

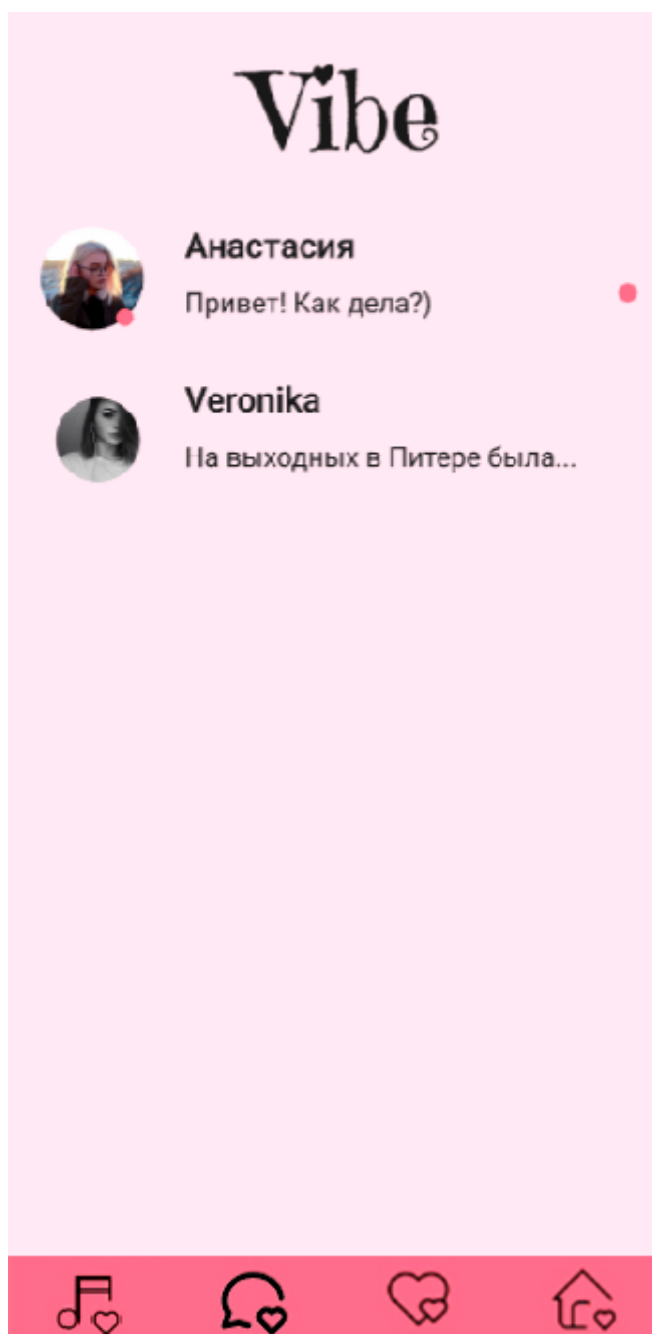


Рисунок 21 – Страница чатов

### 6.3.5 Страница профиля пользователя

Страница доступна только авторизованным пользователям.

На странице расположено: фото профиля, информация о пользователе (имя, город), кнопка для прослушивания своего трека с его названием, а также кнопка «Настройки», кнопка «Изменить описание», кнопка «Изменить фото».

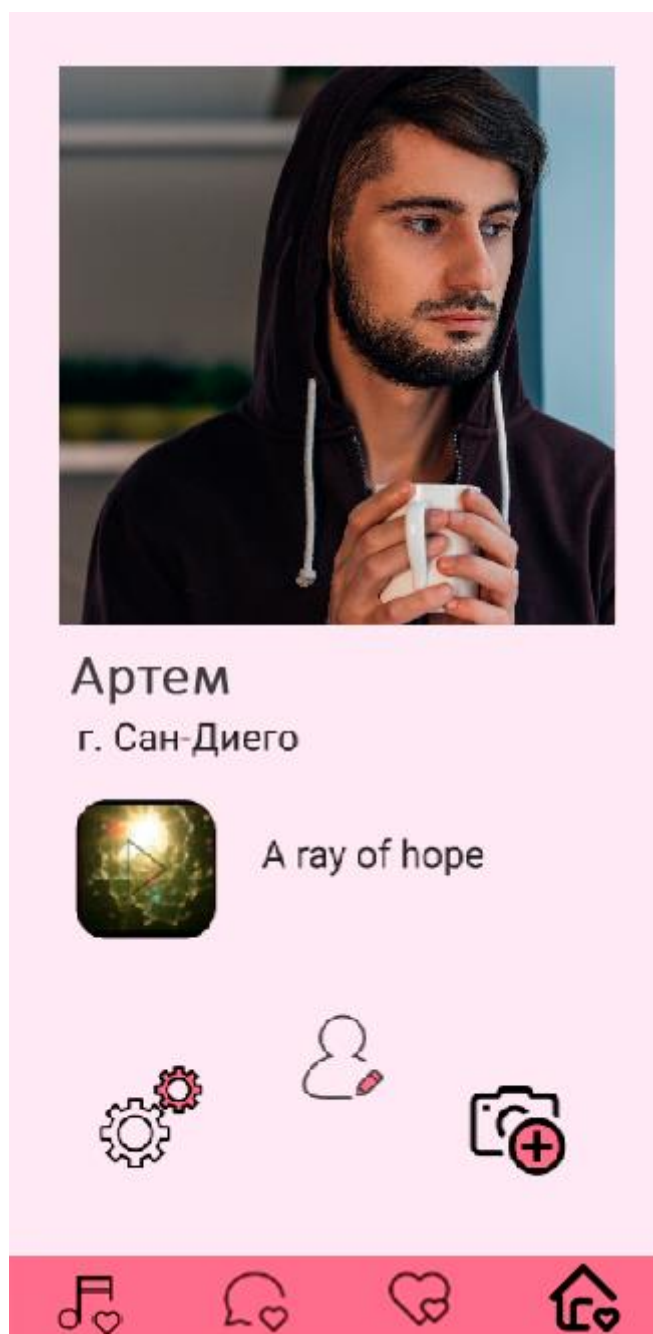


Рисунок 22 – Страница профиля

### **6.3.6 Страница настройки приложения**

Страница доступна только авторизованным пользователям.

В шапке страницы расположен логотип приложения. На странице расположена кнопка для включения/отключения автопроигрывания трека, кнопки для изменения языка приложения «en/ru», кнопка для смены геолокации, ползунок для выбора возрастного диапазона, кнопка для перехода на страницу «Покупка Vibe Premium», а также кнопка «Выйти из аккаунта» внизу страницы.



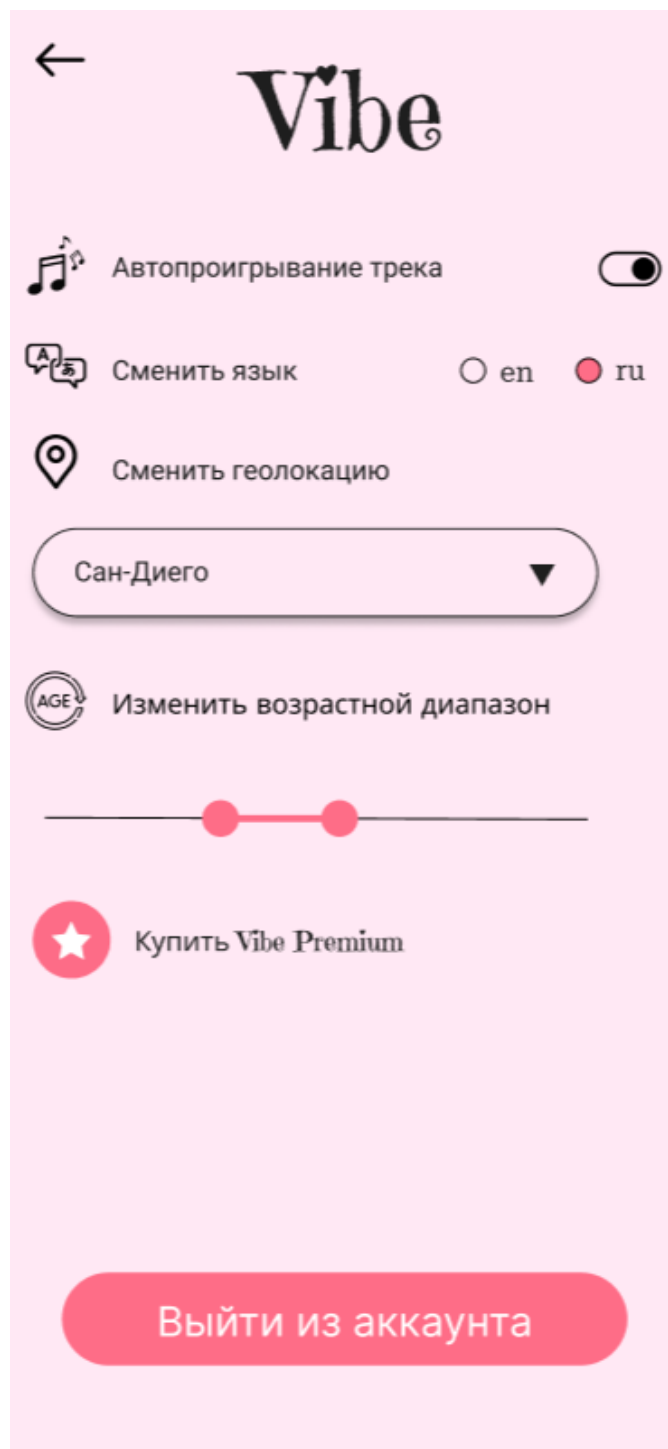


Рисунок 23 – Страница настройки приложения

### 6.3.7 Страница редактирования описания

Страница доступна только авторизованным пользователям.

В шапке страницы расположен логотип приложения. А также на странице расположено поле для ввода описания «О себе», 2 трека с

возможностью прослушивания, 2 кнопки для выбора трека, кнопка «Сгенерировать».

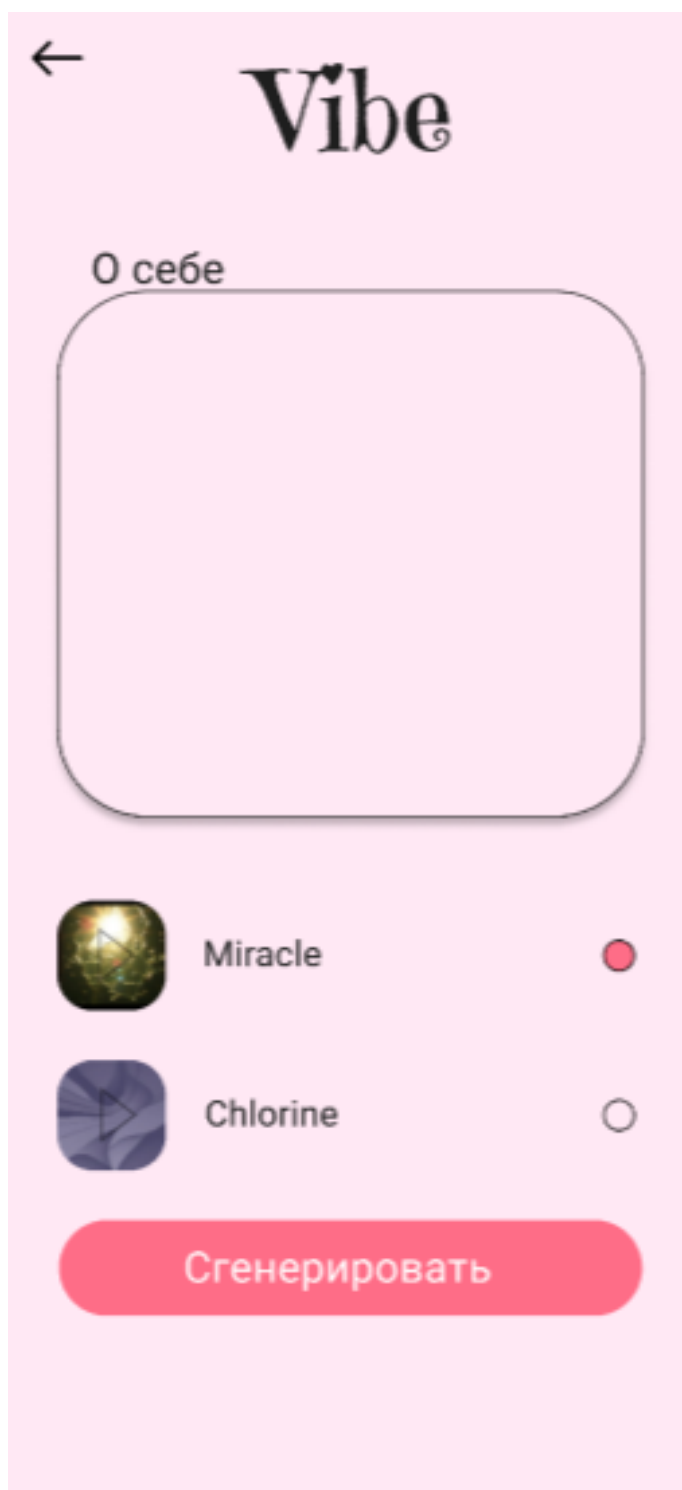


Рисунок 24 – Страница редактирования описания

### 6.3.8 Страница Vibe Premium

Страница доступна только авторизованным пользователям. На странице написана информация о подписке, кнопки для выбора подписки, кнопка для оформления покупки.

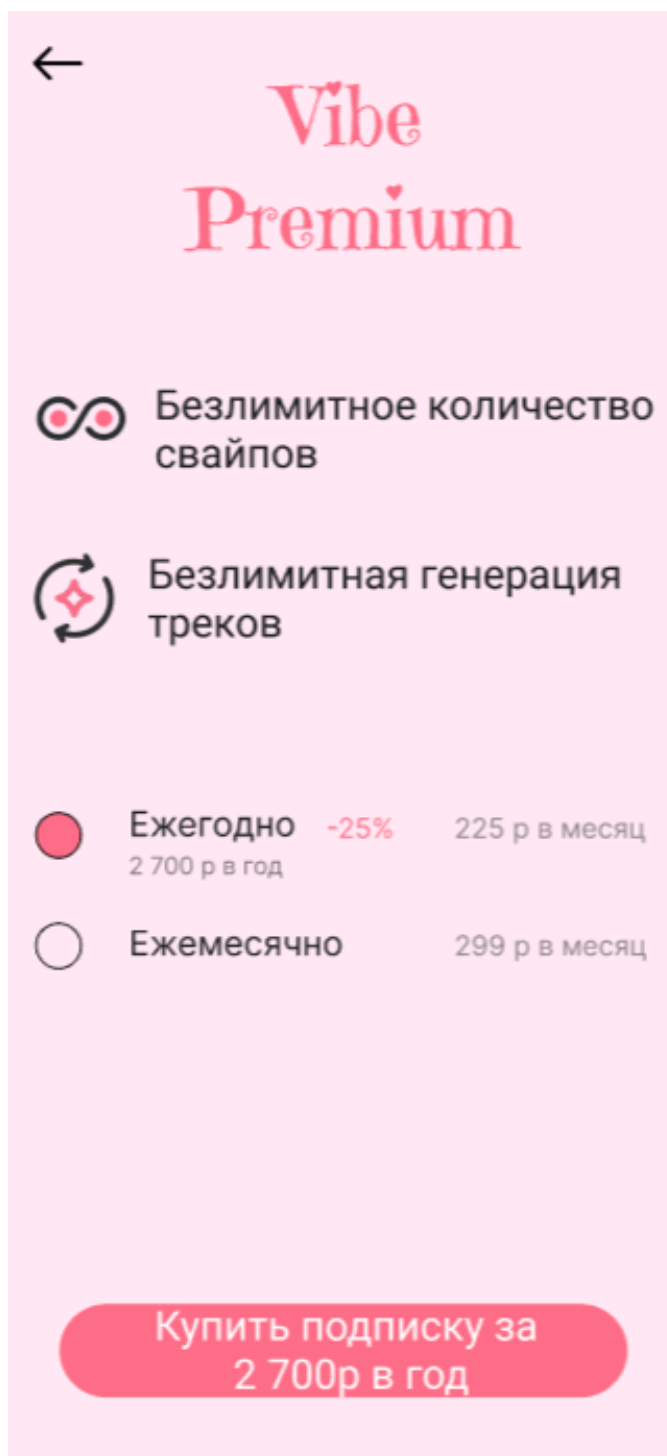


Рисунок 25 – Страница Vibe Premium

## **6.4 Серверная часть**

Серверная часть приложения «Vibe» управляет бизнес-логикой, хранением данных и интеграцией с Suno API. Основные функции:

### **6.4.1 Хранение данных**

PostgreSQL используется для хранения пользовательских профилей, мэтчей, чатов и логов. Spring Data JPA обеспечивает взаимодействие с базой данных через Java-объекты, упрощая управление моделями (User, Profile, Match, Chat).

### **6.4.2 Генерация музыки**

При заполнении или редактировании поля «О себе» сервер отправляет запрос к Suno API с данными пользователя (имя, возраст, описание). API возвращает два 20-секундных аудиотрека, которые сохраняются и связываются с профилем пользователя.

### **6.4.3 Механизм мэтчинга**

Система мэтчинга регистрирует «лайки» и «дизлайки» пользователей. При взаимном «лайке» создается мэтч, открывающий доступ к чату. Бесплатный режим ограничивает пользователей 20 действиями в сутки, что контролируется сервером.

### **6.4.4 Чаты**

После мэтча пользователи могут обмениваться текстовыми сообщениями и фотографиями. История чатов сохраняется в PostgreSQL, а сервер обеспечивает доставку сообщений в реальном времени.

### **6.4.5 Аутентификация и авторизация**

Реализована с использованием Spring Security и JWT. Пароли хранятся в зашифрованном виде (например, с использованием BCrypt). Защита от CSRF, XSS, SQL-инъекций и brute-force атак реализована на уровне сервера с использованием Java и Spring Security.

## 6.4.6 Модерация

Администраторы имеют доступ к панели управления, позволяющей просматривать, модерировать и блокировать профили пользователей. Все действия логируются для аудита.

## 7 Тестирование

Было проведено интеграционное тестирование приложения Vibe в соответствии со стратегией тестирования.

Интеграционное тестирование – это фаза тестирования, при которой проверяется взаимодействие между модулями системы. В рамках проекта Vibe были проверены следующие ключевые сценарии:

- Регистрация и авторизация пользователей (модуль Auth);
- Генерация музыкальных треков (интеграция с Riffusion API);
- Функционал лайков/дизлайков (модуль Likes/Dislikes);
- Чат при взаимной симпатии (модуль Matches/Chat).

|    | A     | B                                      | C                                  | D   | E   | F         | G                  |
|----|-------|--|------------------------------------|---|---|-----------|--------------------|
| 1  | ID    | Название теста                         | Предусловие                        | Шаги  | Ожидаемый результат                                     | Приоритет | Категория          |
| 2  | TC-01 | Регистрация                            | Экран регистрации открыт           | Ввести валидный email и пароль, нажать 'Зарегистрироваться'                 | Пользователь успешно зарегистрирован                    | P0        | Функционал         |
| 3  | TC-02 | Регистрация с пустыми полями           | Экран регистрации открыт           | Нажать 'Зарегистрироваться' без ввода данных                                | Сообщение 'Пожалуйста, заполните все обязательные поля' | P0        | Negative           |
| 4  | TC-03 | Верификация email                      | Экран регистрации открыт           | Ввести корректный код, присланный на введенную при регистрации почту        | Успешна верификация email                               | P0        | Функционал         |
| 5  | TC-04 | Верификация email с некорректным кодом | Экран регистрации открыт           | Ввести любой другой код отличный от того который прислан на введенный email | Сообщение 'Неверный код'                                | P0        | Negative           |
| 6  | TC-05 | Регистрация с некорректным email       | Экран регистрации открыт           | Ввести 'user@example.com', нажать 'Зарегистрироваться'                      | Сообщение 'Неверный формат email'                       | P0        | Negative           |
| 7  | TC-06 | Вход в приложение                      | Пользователь зарегистрирован       | Ввести корректные email и пароль, нажать 'Войти'                            | Пользователь заходит в приложение                       | P0        | Функционал         |
| 8  | TC-07 | Вход с неверным паролем                | Пользователь существует            | Ввести верный email и неверный пароль, нажать 'Войти'                       | Сообщение 'Неправильный пароль'                         | P0        | Negative           |
| 9  | TC-08 | Сохранение предпочтений                | Пользователь авторизован           | Выбрать 3 жанра/настройки, нажать 'Сохранить' и перезайти                   | Предпочтения сохранены и отображаются                   | P1        | Функционал         |
| 10 | TC-09 | Ограничение выбора предпочтений        | Пользователь на экране Preferences | Попытаться выбрать более 5 жанров/настроек                                  | Нельзя выбрать больше 5; Save доступен при выборе 1-5   | P1        | Negative           |
| 11 | TC-10 | Генерация трека по описанию            | Пользователь в профиле             | Ввести текст и нажать 'Генерировать трек'                                   | Отображается spectroграмма и аудиоплеер                 | P0        | Функционал         |
| 12 | TC-11 | Обработка ошибки AI-сервиса            | AI-сервис недоступен               | Нажать 'Генерировать трек', инициировать ответ 503                          | Сообщение 'Сервис недоступен, попробуйте позже'         | P1        | Negative           |
| 13 | TC-12 | Юзажирование результатов генерации     | Сгенерирован трек                  | Вернуться на экран генерации  | Последний трек загружается из кэша                      | P2        | Производительность |
| 14 | TC-13 | Переход на Matches                     | Есть сгенерированный трек          | Нажать 'Matches'  | Отображаются профили с треками                          | P0        | Функционал         |
| 15 | TC-14 | Лайк профиля                           | Экран Matches открыт               | Нажать 'Лайк'   | Профиль перемещается в listMatch                        | P0        | Функционал         |
| 16 | TC-15 | Дизлайк профиля                        | Экран Matches открыт               | Нажать 'Дизлайк'  | Профиль исключается                                     | P1        | Функционал         |
| 17 | TC-16 | Ограничение лайков в день              | Пользователь авторизован           | Поставить 20 лайков, попытаться 21  | Сообщение 'Лимит лайков на сегодня исчерпан'            | P0        | Negative           |
| 18 | TC-17 | Чат при взаимном лайке                 | Взаимная симпатия                  | Перейти в чат и отправить сообщение   | Сообщение отображается у обоих                          | P1        | Функционал         |
| 19 | TC-18 | Прослушивание трека                    | Есть профиль с треком              | Нажать 'Воспроизвести'  | Трек воспроизводится                                    | P1        | Функционал         |
| 20 | TC-19 | Граничные значения в чате              | Чат открыт                         | Отправить пустое и >1000 символов   | Пустое не отправляется; длинный текст обрезается        | P2        | Negative           |
| 21 | TC-20 | Адаптивность UI                        | Разные устройства                  | Открыть на разных экранах   | UI корректен без обрезок                                | P0        | UI/UX              |
| 22 | TC-21 | Проверка CORS                          | DevTools Network                   | Анализ заголовков запросов  | CORS настроен корректно                                 | P1        | Безопасность       |
| 23 | TC-22 | Защита API ключа                       | DevTools Application               | Проверить localStorage  | Ключ хранится защищенно                                 | P1        | Безопасность       |
| 24 | TC-23 | Производительность навигации           | Приложение запущено                | Быстро переключаться между экранами и                                       | Переходы плавные  | P2        | Производительность |
| 25 | TC-24 | Логирование запросов                   | DevTools/серверные логи            | Сгенерировать треки, проверить логи   | Запросы записаны с таймстампом и статусами              | P1        | Integration        |
| 26 | TC-25 | Регрессия после перезагрузки           | На любом экране                    | Перезагрузить приложение  | Состояние сохраняется                                   | P2        | Regression         |
| 27 | TC-26 | Mock режим при ошибке                  | AI-сервис возвращает 500           | Нажать 'Генерировать трек'  | Используются mock-данные                                | P2        | Negative           |
| 28 | TC-27 | Ограничение генерации                  | Пользователь авторизован           | Генерировать >5 раз за минуту   | Сообщение 'Слишком частые запросы, подождите'           | P1        | Negative           |

Рисунок 26 – Ручные тест-кейсы

Результаты:

- Успешно пройдены 22 из 27 тест-кейсов;
- Обнаружены критические ошибки;
- TC-03 – верификация не выполняется (Blocker);
- TC-14 – профиль не перемещается в раздел listMatch (Critical);

— TC-17 – сообщения не отправляются (Critical).

## 8 Аналитика

### 8.1 Методология сбора данных

Для сбора данных о пользовательском поведении использовался сервис Яндекс.Метрика. Отслеживание активности настроено по следующим ключевым показателям: посетители и просмотры, источники трафика, показатель отказов, глубина просмотра, время на сайте, цели, вебвизор.

Для мониторинга производительности системы применяются инструменты Grafana и Prometheus, которые обеспечивают сбор и визуализацию данных о работе приложения в реальном времени. Эти инструменты используются для анализа системных метрик, таких как загрузка CPU, использование памяти и время отклика сервера, что позволяет оперативно выявлять и устранять потенциальные проблемы.

### 8.2 Общие показатели посещаемости

— Количество уникальных посетителей за период: 39;

Вывод: Умеренное количество уникальных посетителей указывает на начальный этап привлечения аудитории или ограниченный доступ к приложению.

— Количество визитов за период: 128;

— Вывод: Соотношение визитов к посетителям (3.2 визита на уникального пользователя) свидетельствует о некоторой повторной заинтересованности пользователей и их возвращаемости.

— Количество просмотров за период: 2899.

Вывод: Высокое количество просмотров страниц (средняя глубина просмотра 22.65) указывает на активное взаимодействие пользователей с контентом и глубокое его изучение.

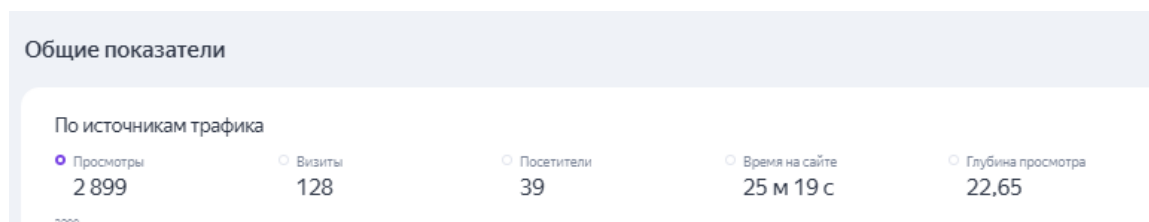


Рисунок 27 – Общие показатели посещаемости

### 8.3 Источники трафика

Основные источники трафика:

- Переходы с сохраненных страниц: 1763 просмотра;
- Внутренние переходы: 558 просмотров;
- Прямые заходы: 548 просмотров;
- Переходы по ссылкам на сайтах: 15 просмотров;
- Переходы из поисковых систем: 4 просмотра.

Вывод: Доминирование "Переходов с сохраненных страниц" и "Прямых заходов" может указывать на то, что пользователи либо сохраняют приложение на свои устройства (что характерно для PWA), либо активно возвращаются к нему после первого взаимодействия. Это также может свидетельствовать о распространении информации о приложении через "сарафанное радио" или прямую рекламу. Низкий процент переходов по ссылкам на сайтах и крайне низкое количество переходов из поисковых систем говорят о необходимости усиления внешнего продвижения и SEO-оптимизации.

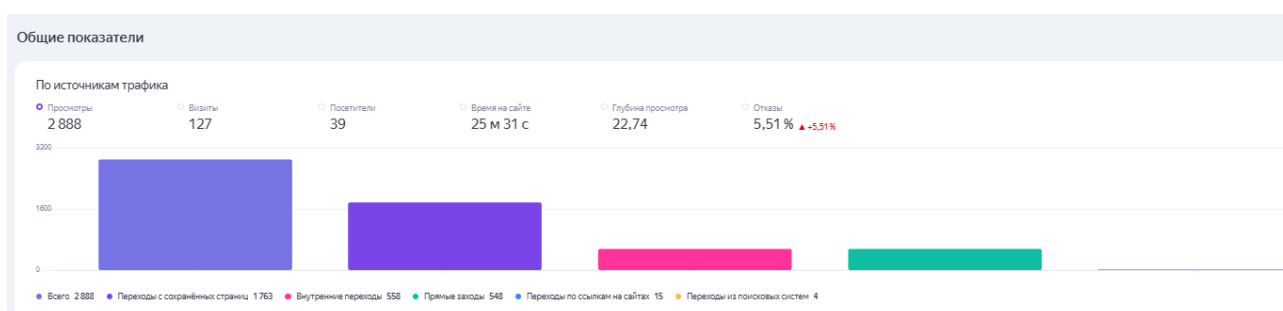


Рисунок 28 – Источники трафика

### 6.4 Вовлеченность пользователей

- Показатель отказов: 5.51%;

Вывод: Низкий показатель отказов является положительным результатом, указывающим на то, что большинство пользователей



просмотрели как минимум две страницы, что свидетельствует о высокой релевантности контента и/или удобстве первого взаимодействия с PWA.

— Средняя глубина просмотра: 22.74 страниц;

Вывод: Чрезвычайно высокая средняя глубина просмотра подтверждает, что пользователи активно изучают различные разделы приложения и взаимодействуют с большим количеством контента. Это очень хороший показатель вовлеченности.

— Среднее время на сайте: 25 минут 31 секунда.

Вывод: Длительное среднее время, проведенное пользователями в приложении, является ярким индикатором его полезности, интересности и удобства использования. Пользователи задерживаются в приложении, что говорит о его ценности.

## **6.5 Достижение целей**

Настроены следующие цели и зафиксированы следующие показатели:

— Цель "Автоцель: отправка формы": Конверсия 56.3%, достижение цели 398 раз, целевых визитов 72;

Вывод: Очень высокая конверсия по отправке формы указывает на ее эффективность и сильную мотивацию пользователей к заполнению. Большое количество достижений свидетельствует о частом использовании этой функции.

— Цель "Автоцель: заполнил контактные данные": Конверсия 44.9%, достижение цели 57 раз, целевых визитов 57;

Вывод: Высокая конверсия в 44.9% является отличным показателем для заполнения контактных данных, что свидетельствует о заинтересованности пользователей в дальнейшем взаимодействии.

— Цель "Автоцель: отправил контактные данные": Конверсия 43.1%, достижение цели 56 раз, целевых визитов 56.

Вывод: Этот показатель подтверждает, что большинство пользователей, заполнивших данные, успешно их отправляют, что указывает на отсутствие технических барьеров или проблем с завершением этого действия.

---

**1. Автоцель: отправка формы** 📄

|                 |       |
|-----------------|-------|
| Конверсия       | 56,3% |
| Достижения цели | 398   |
| Целевые визиты  | 72    |

**2. Автоцель: заполнил контактные данные** 📄

|                 |       |
|-----------------|-------|
| Конверсия       | 44,5% |
| Достижения цели | 57    |
| Целевые визиты  | 57    |

**3. Автоцель: отправил контактные данные** 📄

|                 |       |
|-----------------|-------|
| Конверсия       | 43,8% |
| Достижения цели | 56    |
| Целевые визиты  | 56    |

Рисунок 29 – Достижение целей

## 6.6 Анализ популярных страниц и точек входа

Популярные страницы по просмотрам URL:

- localhost:5173/profile: 569 просмотров;
- localhost:5173/listChat: 491 просмотр;
- localhost:5173/matchFeed: 490 просмотров;
- localhost:5173/listMatch: 350 просмотров;
- localhost:5173/login: 219 просмотров;
- vibedating.ru/login: 196 просмотров;
- localhost:5173/chat/6838381f13c7eb77dc67e39f: 188 просмотров;

- vibedating.ru/listChat: 185 просмотров;
- vibedating.ru/profile: 168 просмотров;
- vibedating.ru/listMatch: 147 просмотров.

| Просмотры URL                                |  |             |
|--|--|-------------|
| Адрес страницы                               |  | Просмотры ↓ |
| localhost:5173/profile                       |  | 569         |
| localhost:5173/listChat                      |  | 491         |
| localhost:5173/matchFeed                     |  | 490         |
| localhost:5173/listMatch                     |  | 350         |
| localhost:5173/login                         |  | 219         |
| vibedating.ru/login                          |  | 196         |
| localhost:5173/chat/6838381f13c7eb77dc67e39f |  | 188         |
| vibedating.ru/listChat                       |  | 185         |
| vibedating.ru/profile                        |  | 168         |
| vibedating.ru/listMatch                      |  | 147         |
| Итого  |  | 3 777       |

Рисунок 30 – Просмотры URL

## 8.4 Мониторинг производительности

Для мониторинга производительности приложения использовались Grafana и Prometheus. Prometheus собирает метрики в реальном времени, включая загрузку CPU, использование памяти (heap и non-heap), количество открытых файлов и время отклика сервера. Эти данные визуализируются в Grafana с использованием дашборда (dashboard ID 10280), что позволяет оперативно анализировать состояние системы.

В ходе исследования зафиксированы следующие показатели:

- Uptime: 2.5 часа (система запущена с 2025-06-04 18:13:03);
- Heap Used: 13.4%;
- Non-Heap Used: 12.2%;
- CPU Usage: 0.06–0.08% (стабильно низкая нагрузка);
- Load Average: Mean 1.87, Last 0.32 (система не перегружена);
- G1 Eden Space (heap): 62 MiB used из 90 MiB max;

- G1 Old Gen (heap): 48 MiB used из 978 MiB max;
- G1 Survivor Space (heap): 3.58 MiB used из 4 MiB max;
- Classes Loaded: 2 (стабильное значение).

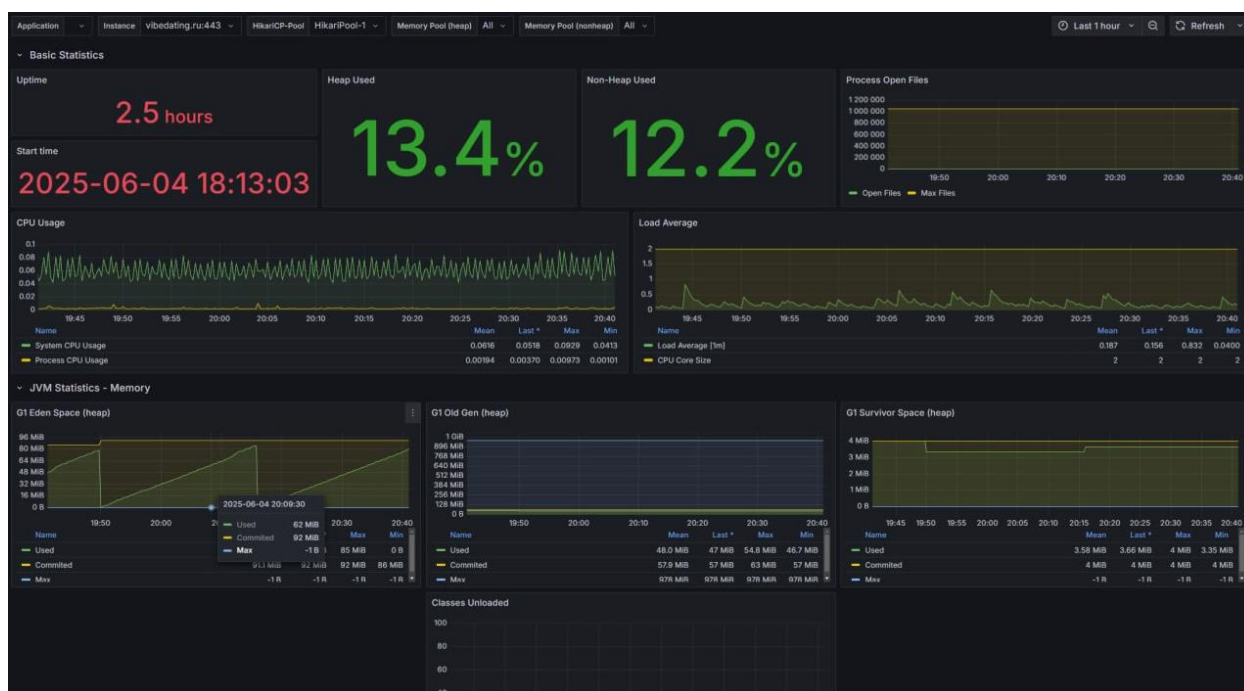


Рисунок 31 – Дашборд мониторинга производительности приложения

Вывод: Низкая загрузка CPU и памяти, а также стабильные показатели нагрузки указывают на то, что система работает эффективно при текущем уровне нагрузки. Использование Grafana Prometheus позволяет оперативно отслеживать состояние приложения и принимать меры при необходимости, обеспечивая высокую отказоустойчивость и производительность.

## ЗАКЛЮЧЕНИЕ

В ходе данной работы были выполнены все поставленные цели. Мы изучили предметную область и рассмотрели существующие решения поставленной проблемы.

В результате было реализовано мобильное дейтинг-приложение «Vibe», основная функциональность которого включает:

- Регистрация и аутентификация пользователя;
- Управление профилем пользователя и генерация уникального музыкального трека на основе описания "О себе", имени и возраста;
- Механизм лайков, скипов и мэтчинга для поиска потенциальных партнеров и открытия чата при взаимной симпатии;
- Чат для обмена текстовыми сообщениями и фотографиями между пользователями, у которых произошел мэтч;
- Демонстрационный режим для неавторизованных пользователей с 5 шаблонными профилями для ознакомления с функционалом;
- Администрирование: просмотр, модерация и блокировка профилей пользователей для поддержания корректной работы сервиса.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Фленаган, Д. JavaScript: Полное руководство / Д. Фленаган; пер. с англ. — 6-е изд. — М.: Вильямс, 2012. — 1088 с. — ISBN 978-5-8459-1717-2.
2. Фаулер, М. Архитектура корпоративных программных приложений / М. Фаулер; пер. с англ. — М.: Вильямс, 2006. — 544 с. — ISBN 5-8459-0857-4.
3. Шилдт, Г. Java: Полное руководство / Г. Шилдт; пер. с англ. — 10-е изд. — М.: Вильямс, 2018. — 1376 с. — ISBN 978-5-8459-2147-5.
4. Гама, Э. Шаблоны проектирования / Э. Гама, Р. Хелм, Р. Джонсон, Дж. Влиссидес; пер. с англ. — СПб.: Питер, 2019. — 448 с. — ISBN 978-5-4461-1356-9.
5. Кнут, Д. Э. Искусство программирования / Д. Э. Кнут; пер. с англ. — М.: Вильямс, 2018. — Т. 1. Основные алгоритмы. — 720 с. — ISBN 978-5-8459-2014-0.
6. Троелсен, Э. Язык программирования C# 7 и платформы .NET и .NET Core / Э. Троелсен, Ф. Джепикс; пер. с англ. — М.: Вильямс, 2018. — 1328 с. — ISBN 978-5-8459-2159-8.
7. Официальная документация React. [Электронный ресурс] // React. — URL: <https://react.dev/> (дата обращения: 04.06.2025).
8. Официальная документация PostgreSQL. [Электронный ресурс] // PostgreSQL Global Development Group. — URL: <https://www.postgresql.org/docs/> (дата обращения: 04.06.2025).
9. ГОСТ 7.1-2003. Библиографическая запись. Общие требования и правила составления. — Введ. 2004-01-01. — М.: ИПК Издательство стандартов, 2003. — 48 с.
10. Регламент (ЕС) 2016/679 Европейского парламента и Совета от 27 апреля 2016 года о защите физических лиц в связи с обработкой персональных данных (GDPR). [Электронный ресурс] // Официальный журнал Европейского Союза. — URL: <https://eur-lex.europa.eu/eli/reg/2016/679/oj> (дата обращения: 04.06.2025).

11. Suno API Documentation. [Электронный ресурс] // Suno. — URL: <https://docs.suno.ai/> (дата обращения: 04.06.2025).