

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

Факультет Компьютерных наук

Кафедра программирования и информационных технологий

«Дневник, который фиксирует эмоции пользователя, выявляет тренды и
помогает улучшить психологическое состояние Reflect»

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 _____ В.А. Ушаков

Обучающийся _____ Д.Д. Абдуллаев, 3 курс, д/о

Обучающийся _____ А.А. Трегубова 3 курс, д/о

Обучающийся _____ Р.А. Романин, 3 курс, д/о

Обучающийся _____ М.А. Хрячков, 3 курс, д/о

Обучающийся _____ В.И Котолевский, 3 курс, д/о

Воронеж 2025

СОДЕРЖАНИЕ

Определения, обозначения и сокращения	4
Введение.....	6
1 Постановка задачи.....	7
1.1 Цели создания системы	7
1.2 Функциональные требования к разрабатываемой системе	7
1.3 Задачи системы.....	8
2 Анализ предметной области	9
2.1 Обзор аналогов	9
2.1.1 How We Feel.....	9
2.1.2 Moodfit.....	10
2.1.3 Daylio	11
2.2 Моделирование системы	13
2.2.1 Диаграммы последовательностей.....	13
2.2.2 Диаграмма развертывания.....	15
2.2.3 Диаграмма состояний	16
2.2.4 Моделирование бизнес-части	18
3 Реализация.....	19
3.1 Средства реализации.....	19
3.1.1 Мобильное приложение (Android)	19
3.1.2 Бэкенд (Серверная часть)	19
3.1.3 Искусственный интеллект (Анализ эмоций).....	20
3.1.4 Тестирование	20
3.1.5 Дополнительные инструменты	20
3.2 Логика приложения.....	22
3.2.1 Backend разработка	22
3.2.2 Frontend разработка.....	23
3.3 Реализация интерфейса	24

3.3.1 Экран записи эмоций	24
3.3.2 Экран «Записи»	25
3.3.3 Экран «Статистика».....	27
3.3.4 Экран «Друзья»	29
3.3.5 Экран профиля пользователя	30
3.3.6 Экран чата с ИИ	32
3.3.7 Экран добавления виджетов	33
3.3.8 Экран админ-панели	34
ЗАКЛЮЧЕНИЕ	37
Список использованных источников	39

Определения, обозначения и сокращения

В настоящей курсовой работе применяют следующие термины и сокращения с соответствующими определениями:

AI (Artificial Intelligence) – Искусственный Интеллект (ИИ). В документе преимущественно используется русская аббревиатура.

API (Application Programming Interface) – Программный интерфейс приложения, набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением для использования во внешних программных продуктах.

Clean Architecture – «Чистая архитектура», архитектурный шаблон, направленный на разделение ответственностей в программном обеспечении.

DTO (Data Transfer Object) – Объект передачи данных; простой объект, используемый для передачи данных между слоями приложения.

JSON (JavaScript Object Notation) – Текстовый формат обмена данными, основанный на JavaScript.

JWT (JSON Web Token) – Веб-токен JSON; открытый стандарт (RFC 7519) для создания токенов доступа, используемых для передачи данных для аутентификации.

MVI (Model-View-Intent) – Архитектурный шаблон, используемый в разработке пользовательских интерфейсов, особенно в Android.

NLP (Natural Language Processing) – Обработка естественного языка; направление в искусственном интеллекте и математической лингвистике, изучающее проблемы компьютерного анализа и синтеза естественных языков.

ORM (Object-Relational Mapping) – Технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования.

Reflect – Название разрабатываемого мобильного приложения.

REST (Representational State Transfer) – Архитектурный стиль взаимодействия компонентов распределённого приложения в сети.

SWOT-анализ (Strengths, Weaknesses, Opportunities, Threats) – Метод стратегического планирования, заключающийся в выявлении факторов внутренней и внешней среды организации и разделении их на четыре категории: Сильные стороны, Слабые стороны, Возможности и Угрозы.

VDS (Virtual Dedicated Server) – Виртуальный выделенный сервер.

ВК – Социальная сеть «ВКонтакте».

ВОЗ – Всемирная организация здравоохранения.

XLS – Формат файла, используемый программой excel.

Введение

Современный ритм жизни, повышенные нагрузки и стрессовые ситуации оказывают значительное влияние на психологическое состояние людей. Понимание и анализ своих эмоций являются важными аспектами поддержания ментального здоровья, однако традиционные методы отслеживания эмоционального состояния часто требуют специальных знаний, времени и усилий, что делает их малодоступными для большинства людей.

В условиях повсеместного распространения мобильных технологий и развития искусственного интеллекта появилась возможность создания приложений, которые помогают пользователям фиксировать, анализировать и понимать свои эмоциональные состояния. Такие приложения обладают значительным потенциалом для улучшения психологического благополучия, предоставляя пользователям ценные инсайты об их эмоциональных паттернах. Приложение "Reflect" призвано помочь людям лучше осознавать свои эмоции, выявлять тенденции в изменении настроения и получать персонализированные рекомендации по улучшению психологического состояния.

Разрабатываемое приложение не только облегчает процесс самоанализа, позволяя пользователям легко фиксировать свое эмоциональное состояние, но и служит важным инструментом для повышения эмоционального интеллекта и самопознания. Кроме того, оно может быть полезным ресурсом для психологов и исследователей, предоставляя ценные данные об эмоциональных состояниях различных групп населения. Таким образом, данный проект направлен на создание удобного и эффективного инструмента для работы с эмоциями с использованием современных мобильных технологий.

1 Постановка задачи

1.1 Цели создания системы

Целями данной курсовой работы являются:

- реализация системы, которая позволит пользователям фиксировать свои эмоциональные состояния, анализировать тренды эмоций и получать персонализированные советы от ИИ для улучшения психологического благополучия;
- создание платформы, поддерживающей социальное взаимодействие;
- предоставление премиум-функций, которые увеличат вовлеченность пользователей и дадут заказчику дополнительный доход за счет подписок;

1.2 Функциональные требования к разрабатываемой системе

Разрабатываемое приложение должно соответствовать следующим функциональным требованиям:

- возможность регистрации через почту и пароль;
- возможность регистрации и авторизации через интеграцию с VK;
- создание и редактирование записей об эмоциональном состоянии;
- просмотр статистики эмоций;
- редактирование данных своего аккаунта после авторизации или регистрации в системе;
- редактирование профиля (изменение имени, пароля);
- добавление/удаление друзей;
- настройка уведомлений (включение/выключение);
- функционал администратора

- возможность приобретения платной подписки для возможности увеличения количества запросов на диалог с ИИ;

1.3 Задачи системы

Задачами работы являются:

- записывать свои эмоциональные ощущения в течение дня с использованием различных методов ввода (текст, выбор из списка тегов, шкалы);
- просматривать историю своих записей эмоциональных состояний в различных форматах (список, графики);
- получать анализ трендов и паттернов своих состояний с помощью ИИ;
- создавать виджеты для отображения ежедневных отметок эмоциональных состояний и визуальных индикаторов, чтобы мотивировать пользователей к регулярному использованию;
- добавлять друзей, просматривать их эмоциональное состояние и статистику;
- настраивать видимость своих записей и статистики (скрытие от всех, показ друзьям) для защиты персональных данных;
- просматривать и редактировать данные аккаунта после авторизации;
- подавать жалобы на других пользователей;
- рассматривать жалобы и принимать соответствующие меры администратором;
- блокировать социальный функционал по жалобам пользователей администратором;
- удалять пользователей, которые регулярно нарушают правила системы, администратором.

2 Анализ предметной области

В современном мире, где уровень стресса и психоэмоциональных нагрузок постоянно растет, все больше людей сталкиваются с тревожностью, депрессией и эмоциональным выгоранием. Согласно исследованиям ВОЗ [11], за последние 10 лет количество людей с диагностированными депрессивными расстройствами увеличилось на 20%, а каждый пятый человек хотя бы раз в жизни испытывал симптомы тревожного расстройства. При этом многие не обращаются за профессиональной помощью из-за страха стигматизации, высокой стоимости терапии или просто непонимания своего состояния.

В таких условиях цифровые инструменты для самопомощи становятся важным решением. Они позволяют отслеживать эмоциональное состояние, выявлять закономерности и принимать меры до того, как ситуация усугубится. Однако большинство существующих приложений предлагают лишь базовый функционал – запись настроения и простые графики, без глубокой аналитики и персонализированных рекомендаций.

2.1 Обзор аналогов

Среди конкурентов приложения для того, чтобы фиксировать эмоции, анализировать тренды и получать персонализированные советы от ИИ были рассмотрены три конкурента: How We Feel, Moodfit и Daylio, которые предлагают схожие возможности в нише трекинга настроения. Рассмотрим SWOT анализ каждого конкурента.

2.1.1 How We Feel

Сильные стороны:

- социальные функции позволяют делиться эмоциями с друзьями в реальном времени, что поддерживает сообщество;
- опциональная ИИ-поддержка предоставляет персонализированные советы;
- научная основа (разработано с участием Yale University) добавляет доверия;
- простой интерфейс и высокие оценки пользователей (4.9 из 5 на App Store);

Слабые стороны:

- проблемы с интерфейсом после обновлений, например, элементы UI могут быть недоступны на устройствах с широкими статус-барами;
- точность данных зависит от честности и самосознания пользователя;

Возможности:

- общественное сознание о важности ментального здоровья растет, что создает благоприятную среду для приложений, направленных на его поддержку;
- интеграция с другими приложениями здоровья для комплексного опыта;

Угрозы:

- риски утечки данных, что может повлиять на репутацию;
- необходимость следить за технологическими трендами;
- застой в эмоциональной оценке из-за отсутствия у пользователя постоянной внешней мотивации фиксировать своё состояние;

2.1.2 Moodfit

Сильные стороны:

- широкий набор инструментов: трекинг настроения, медитация, благодарностный журнал;
- гибкость в настройке целей и трекинга;

Слабые стороны:

- подписка требуется для полного доступа, что может отпугнуть пользователей;
- эффективность Moodfit зависит от готовности пользователя регулярно использовать приложение и выполнять упражнения. Недостаток самодисциплины может снизить результаты;

Возможности:

- улучшение премиум-функций для оправдания стоимости подписки;
- общественное сознание о важности ментального здоровья растет, что создает растущий спрос на решения для самопомощи и приложения для ментального здоровья;

Угрозы:

- конкуренция с бесплатными альтернативами;
- ужесточение регулирования в сфере здравоохранения и конфиденциальности данных может потребовать дополнительных усилий для соответствия и повлиять на бизнес-модель;
- изменения в регулировании приложений для психического здоровья;

2.1.3 Daylio

Сильные стороны:

- простота использования, подходит для широкой аудитории;

- бесплатная версия с достаточным функционалом (трекинг настроения, статистика, стрики);
- поддерживает долгосрочное использование, помогая анализировать привычки;
- гибкость в настройке настроений и активностей;

Слабые стороны:

- отсутствие социальных функций, что ограничивает взаимодействие с друзьями;
- нет ИИ-поддержки;
- премиум-версия может не оправдывать ожидания по сравнению с конкурентами;

Возможности:

- добавление социальных функций для привлечения пользователей, ищущих сообщество;
- внедрение ИИ для анализа данных и персональных рекомендаций;

Угрозы:

- сложности с удержанием пользователей из-за отсутствия новизны;
- вызовы монетизации без отпугивания бесплатных пользователей;

2.2 Моделирование системы

2.2.1 Диаграммы последовательностей

Диаграмма последовательности иллюстрирует взаимодействия между пользователем/администратором, приложением, сервером, базой данных и модулем искусственного интеллекта. Сначала пользователь регистрируется через email, отправляя данные через приложение на сервер для сохранения в базе данных. Затем пользователь может записывать свое эмоциональное состояние, выбирая эмоцию и интенсивность; эти данные также отправляются на сервер и сохраняются. Для анализа тренда эмоций пользователь запрашивает отчет за период; приложение обращается к серверу, который извлекает данные из базы, передает их ИИ для анализа и возвращает результат пользователю. Диаграмма также показывает процесс добавления друга, где пользователь отправляет запрос через приложение на сервер, который фиксирует его в базе данных. Наконец, администратор может заблокировать аккаунт пользователя через админ-панель: приложение передает команду на сервер, который обновляет статус пользователя в базе. Данная диаграмма представлена на Рисунке 1 - Диаграмма последовательностей (ключевые функции).

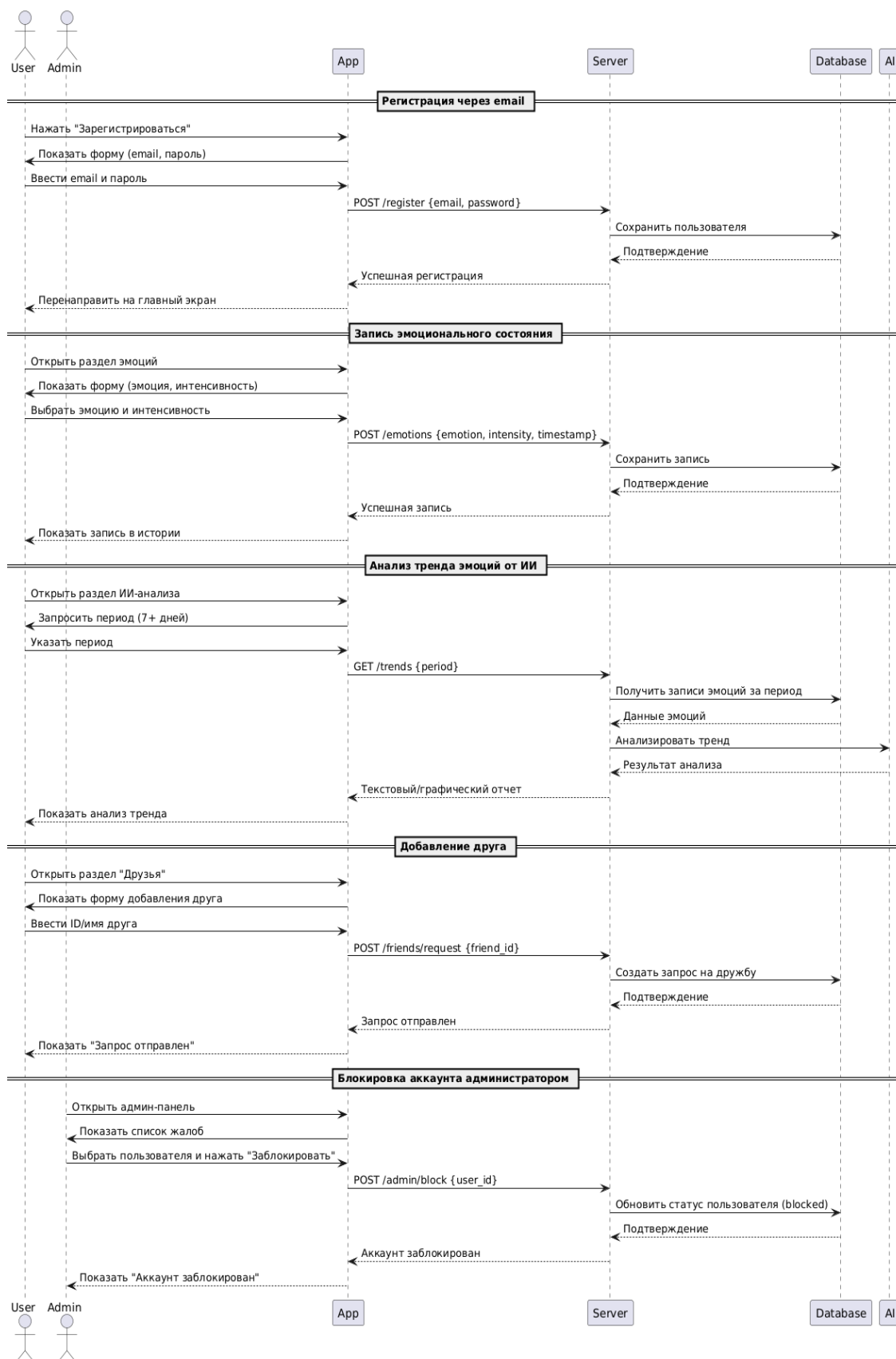


Рисунок 1 - Диаграмма последовательностей (ключевые функции)

2.2.2 Диаграмма развертывания

Диаграмма архитектуры показывает, как взаимодействуют различные компоненты системы. На верхнем уровне находятся: Web Административная панель и Android Мобильное приложение. Оба этих устройства взаимодействуют с Backend Server через REST API. Backend Server, в свою очередь, использует SQL (обернутый в Django ORM, что является дополнительной защитой от SQL инъекций) для обмена данными с базой данных PostgreSQL. Кроме того, Backend Server обращается к нейронной сети DeepSeek (также через REST API) для выполнения задач, которые используют искусственный интеллект. Таким образом, Backend Server выступает центральным узлом, обрабатывающим запросы от клиентских устройств, управляющим данными в базе и использующим возможности нейронной сети. Данная диаграмма представлена на Рисунок 2 - Диаграмма развертывания.

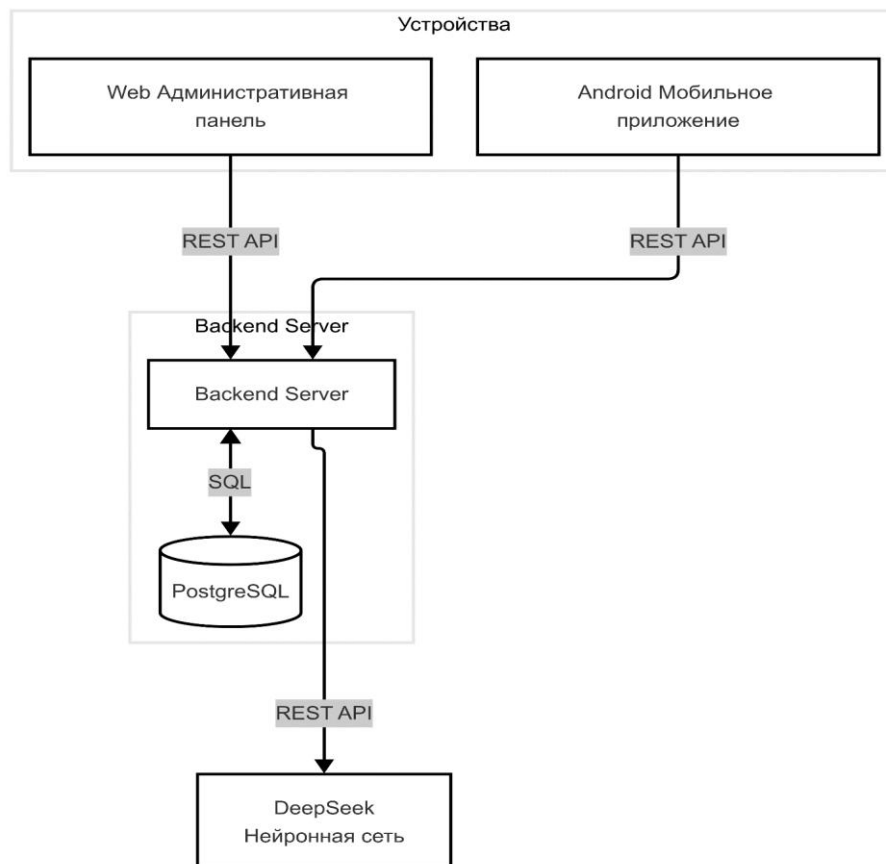


Рисунок 2 - Диаграмма развертывания

2.2.3 Диаграмма состояний

Эта диаграмма состояний описывает потоки взаимодействия пользователя и администратора с приложением.

Начинается всё с состояния "НеАвторизован". Отсюда пользователь может перейти к "Регистрации" (включающей ввод данных и подтверждение, ведущие к "Успешной регистрации" и переходу в авторизованное состояние) или попытаться авторизоваться, что приводит к состоянию "Авторизован".

В состоянии "Авторизован" пользователь попадает на "ОсновнойЭкран". С этого экрана доступны переходы в различные разделы:

- "РазделПрофиля": включает просмотр профиля, редактирование, покупку премиум-доступа, удаление профиля (с подтверждением);
- "РазделНастроек": позволяет просматривать настройки, настраивать уведомления и видимость, обращаться в поддержку (с отправкой сообщения) и создавать виджеты (с подтверждением создания);
- "РазделИИАнализа": здесь можно просматривать ИИ, запрашивать советы и тренды (с последующим получением совета или анализа);
- "РазделЭмоций": включает просмотр эмоций, запись новых эмоций (создание и сохранение), экспорт статистики (с просмотром статистики и завершением экспорта), просмотр "стрика" (серии), а также редактирование и удаление записей эмоций;
- "РазделДрузей": содержит список друзей, возможность добавления друга (с отправкой запроса), просмотр эмоций друга, удаление друга (с подтверждением) и отправку жалобы на друга (с подтверждением отправки);

- Из любого из этих разделов основного экрана или из самого основного экрана пользователь может "Выйти из аккаунта", возвращаясь в состояние "НеАвторизован".

Параллельно существует поток для администратора. После "Авторизации администратора" он попадает в "АдминПанель". Основное действие здесь - "ПросмотрЖалоб". Из этого состояния администратор может:

- "Заблокировать пользователя", что ведет к состоянию "БлокировкаАккаунта" с последующим "Подтверждением блокировки";
- "Ограничить функции", что ведет к состоянию "ОграничениеДоступа" с "Подтверждением ограничения";
- "Выйти из аккаунта".

Диаграмма показывает начальные и конечные состояния для каждого из основных модулей (например, разделов приложения или админ-панели).

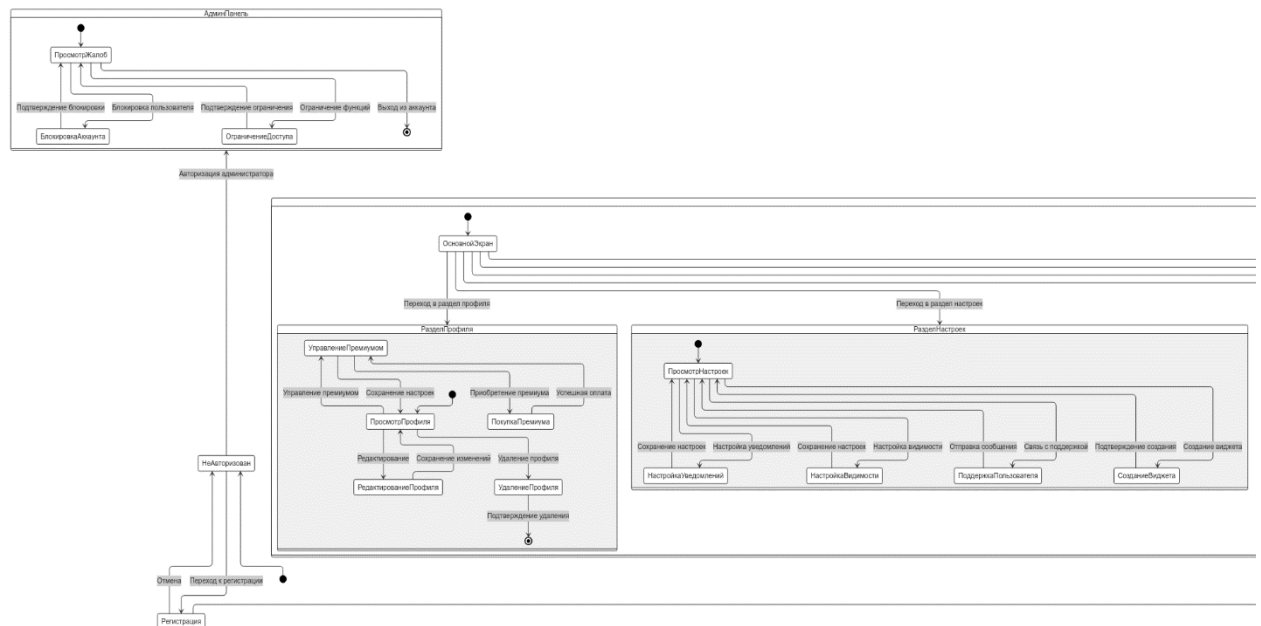


Рисунок 3 - Диаграмма состояний (1 часть)

3 Реализация

3.1 Средства реализации

3.1.1 Мобильное приложение (Android)

Основные технологии:

- Язык программирования: Kotlin;
- Среда разработки: Android Studio Koala.

Архитектура:

- MVI (Model-View-Intent) – управление состоянием UI;
- Clean Architecture – разделение на слои (Presentation, Domain, Data).

Библиотеки:

- Coroutines + Flow – асинхронность;
- Dagger Hilt – внедрение зависимостей;
- Retrofit – REST API клиент;
- Room – локальная база данных (кеширование);
- WorkManager – фоновые задачи (уведомления);

3.1.2 Бэкенд (Серверная часть)

Основные технологии:

- Язык программирования: Python
- Фреймворк: Django (Django REST Framework для API)
- База данных: PostgreSQL
- Аутентификация: JWT (JSON Web Tokens)
- API: REST (JSON)
- Хостинг: VDS (Sprintbox)
- Контейнеризация: Docker
- CI/CD: GitHub Actions

3.1.3 Искусственный интеллект (Анализ эмоций)

Модель: DeepSeek-R1:Free (7B параметров)

API для взаимодействия: RabbitMQ

3.1.4 Тестирование

Во время работы проводились следующие типы тестирования:

- Функциональное тестирование;
- UI тестирование;
- Тестирование скорости;
- Тестирование удобства использования;

Проводилось ручное тестирование с составлением чек-листов, тест кейсов. В ходе тестирования приложения было выявлено и исправлено около 20 багов средней и высокой серьезности.

3.1.5 Дополнительные инструменты

В процессе разработки данного проекта использовался следующий набор инструментов:

- Для версионного контроля и совместной работы над кодовой базой применялся Git, с размещением репозитория на платформе GitHub.
- Управление задачами и отслеживание прогресса разработки осуществлялось с помощью таск-менеджера YouGile
- Разработка пользовательского интерфейса (UI) и пользовательского опыта (UX) проходила в Figma. Этот инструмент использовался для создания интерактивных макетов, прототипов и дизайн-системы.

- Для документирования API использовался Swagger (OpenAPI).

3.2 Логика приложения

3.2.1 Backend разработка

Проект построен на основе Django с использованием модульной архитектуры, где каждое логическое направление (профиль, эмоции, друзья, жалобы, админка и т.д.) оформлено в виде отдельных Django-приложений. Основное API реализовано через Django REST Framework, с авторизацией на базе JWT (SimpleJWT). Приложение разделено на backend (API, WebSocket, AI-процессинг) и отдельную административную панель, каждая из которых работает как отдельный Django-проект внутри Docker.

В качестве базы данных используется PostgreSQL, развёрнутый в отдельном контейнере внутри Docker Compose-среды. Django подключается к базе через стандартный psycopg2 адаптер, а все взаимодействия с БД реализованы через ORM

PostgreSQL используется для хранения всех пользовательских данных, включая профили, эмоциональные состояния, теги, дружбы, жалобы, административные действия и другие сущности

Для асинхронной обработки запросов к ИИ используется RabbitMQ: когда пользователь отправляет сообщение, оно попадает в очередь, где потребитель (AI consumer) в отдельном контейнере обрабатывает его с помощью внешнего AI-провайдера (через OpenRouter API). Веб-сокеты реализованы с использованием Django Channels и Redis. Уведомления (например, о входящих заявках в друзья) отправляются пользователям в реальном времени через WebSocket. Вся архитектура развёрнута с помощью Docker Compose, отдача статики и проксирование трафика осуществляется через Nginx

3.2.2 Frontend разработка

В качестве базовой архитектуры Frontend-части мобильного приложения Reflect выбрано сочетание Clean Architecture и MVI (Model-View-Intent), что обеспечивает четкое разделение ответственности между компонентами, предсказуемость состояния интерфейса и легкость тестирования. Проект организован в виде многоуровневой модульной структуры, включающей основные модули data, domain, presentation и вспомогательный модуль common.

Модуль common выполняет роль инфраструктурного слоя, содержащего общие для всего приложения компоненты. В него входят:

- интерсепторы для OkHttpClient (папка interceptor), обрабатывающие сетевые запросы и ответы;
- реализации работы с SharedPreferences (папка prefs) для хранения локальных данных;
- вспомогательные утилиты и расширения, используемые во всех модулях.

Модуль data отвечает за работу с данными и включает:

- настройки dependency injection через Hilt (папка di);
- data Transfer Objects (папка dto) для сериализации/десериализации;
- реализацию сетевого слоя через Retrofit (папка remote);
- конкретные реализации репозитория (папка repository), разбитые по функциональным доменам.

Модуль domain содержит ядро бизнес-логики приложения:

- модели предметной области (папка model);
- интерфейсы репозитория (папка repository);
- Use Cases (папка usecase), инкапсулирующие бизнес-правила.

Модуль presentation реализует визуальную часть на основе MVI:

- адаптеры для RecyclerView и ViewPager (папка adapter);
- общие UI-компоненты (папка common);
- диалоговые окна (папка dialog);
- компоненты главного Activity (папка mainActivity);
- экранные компоненты (папка screens), организованные по принципу feature-модулей.

3.3 Реализация интерфейса

3.3.1 Экран создания записи эмоций

Назначение экрана

- Фиксация текущего эмоционального состояния
- Интеграция с ИИ для получения советов

Элементы экрана:

- Шкала эмоций (от 1 до 10 меняется ползунком)
- Кнопки действий
- "Уточнить" – переход к расширенному выбору эмоций.
- "Сохранить без подробностей" – запись только уровня настроения (без тегов/описания).



Рисунок 5 - Экран записи эмоций

3.3.2 Экран «Записи»

Экран "Записи" является центральным элементом приложения, предоставляя пользователям полный обзор их эмоциональной истории. Этот экран выполняет несколько ключевых функций:

Структура экрана:

- Верхняя панель:
 - Заголовок "Записи"
 - Кнопка для открытия "стрика"

- Основная область:
- Хронологический список всех записей
- Каждая запись содержит:
 - Время создания
 - Основную эмоцию
 - Краткое описание
 - Используемые теги
- Нижняя панель навигации:
 - Быстрый доступ к основным разделам приложения

Функциональные возможности

- Просмотр записей
 - Возможность прокрутки всей истории

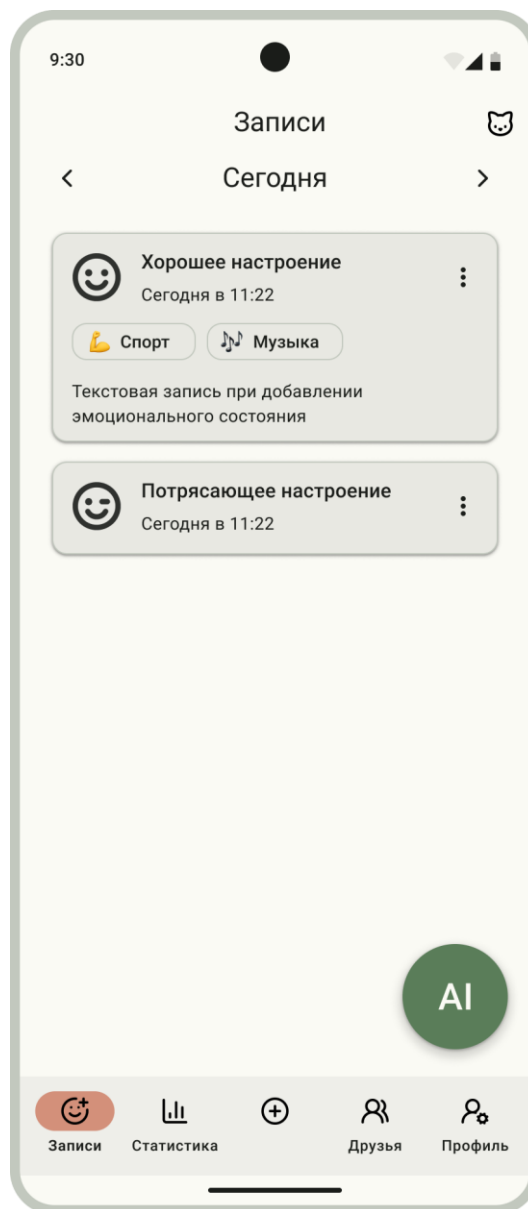


Рисунок 6 - Экран информации о растении

3.3.3 Экран «Статистика»

Экран статистики предоставляет пользователям детализированную аналитику их эмоционального состояния за различные периоды времени, помогая выявлять закономерности и тенденции.

Структура экрана:

- Верхняя панель управления
 - Выбор периода: "Неделя", "Месяц", "Год"

- Кнопка экспорта данных (xls)
- Переключатель между видами статистики
- Основная аналитическая панель
 - График эмоционального состояния
 - Круговая диаграмма распределения эмоций
- Детализированные показатели
 - Средний показатель настроения
 - Самые частые эмоции
 - Теги эмоций



Рисунок 7 - Экран статистики

3.3.4 Экран «Друзья»

Экран "Друзья" позволяет пользователям управлять своим социальным окружением в приложении, просматривать эмоциональное состояние друзей и взаимодействовать с ними.

Основные элементы интерфейса

- Верхняя панель
 - Заголовок "Друзья"
 - Кнопка "Найти друзей" (переход в поиск)
- Основное содержание
 - Вертикальный список друзей (как в приложенном изображении)
 - Для каждого друга отображается:
 - Имя и фамилия
 - Индикатор текущего эмоционального состояния (цветной круг/иконка)

Функциональные возможности:

- Просмотр друзей
 - Скроллируемый список всех добавленных друзей
 - При тапе на друга - переход в его профиль

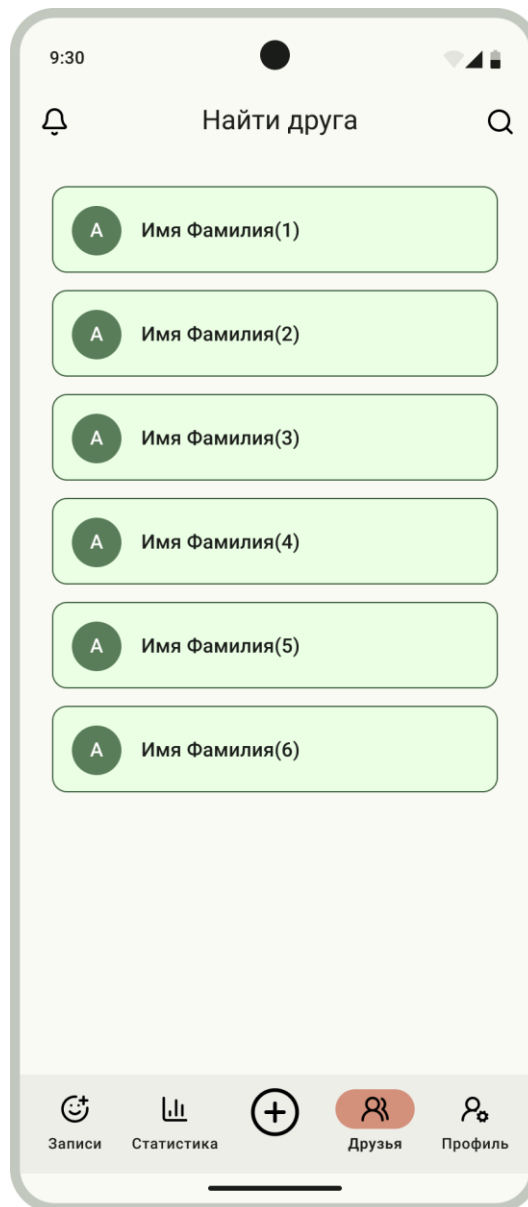


Рисунок 8 - Экран профиля пользователя

3.3.5 Экран профиля пользователя

Экран профиля предоставляет пользователю доступ к персональным настройкам аккаунта, информации о подписке и функциям управления учетной записью.

Структура экрана:

- Основные элементы:
 - Аватар пользователя (круглый элемент)
 - Имя пользователя (крупным шрифтом)
 - Статус аккаунта (обычный/премиум)

- Функциональные возможности:
 - Управление подпиской
 - Создание виджетов
 - Выход из системы

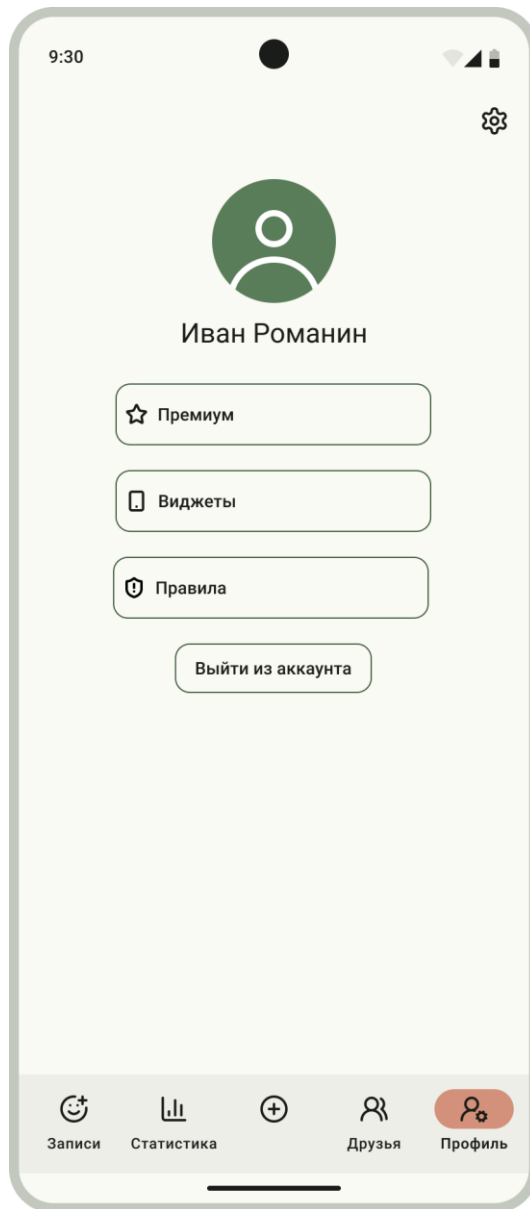


Рисунок 9 - Экран профиля пользователя

3.3.6 Экран чата с ИИ

Экран "Чат с ИИ" представляет собой чат для общения пользователя с искусственным интеллектом, разработанную для анализа эмоционального состояния и предоставления персонализированных рекомендаций.

Структура экрана:

- Верхняя панель:
 - Заголовок чата
 - Кнопка возврата в предыдущее меню
- Основная область чата:
 - Приветственное сообщение
 - Блок предустановленных сообщений

Функциональные возможности:

- Отправка текстовых сообщений ИИ
- Выбор предустановленных вопросов
- Прикрепление текущего эмоционального состояния к сообщению
- Просмотр истории диалога

Пользовательский сценарий:

- Пользователь открывает чат
- Выбирает один из предустановленных вопросов или вводит свой
- ИИ анализирует запрос + историю эмоций пользователя
- Формируется персонализированный ответ.

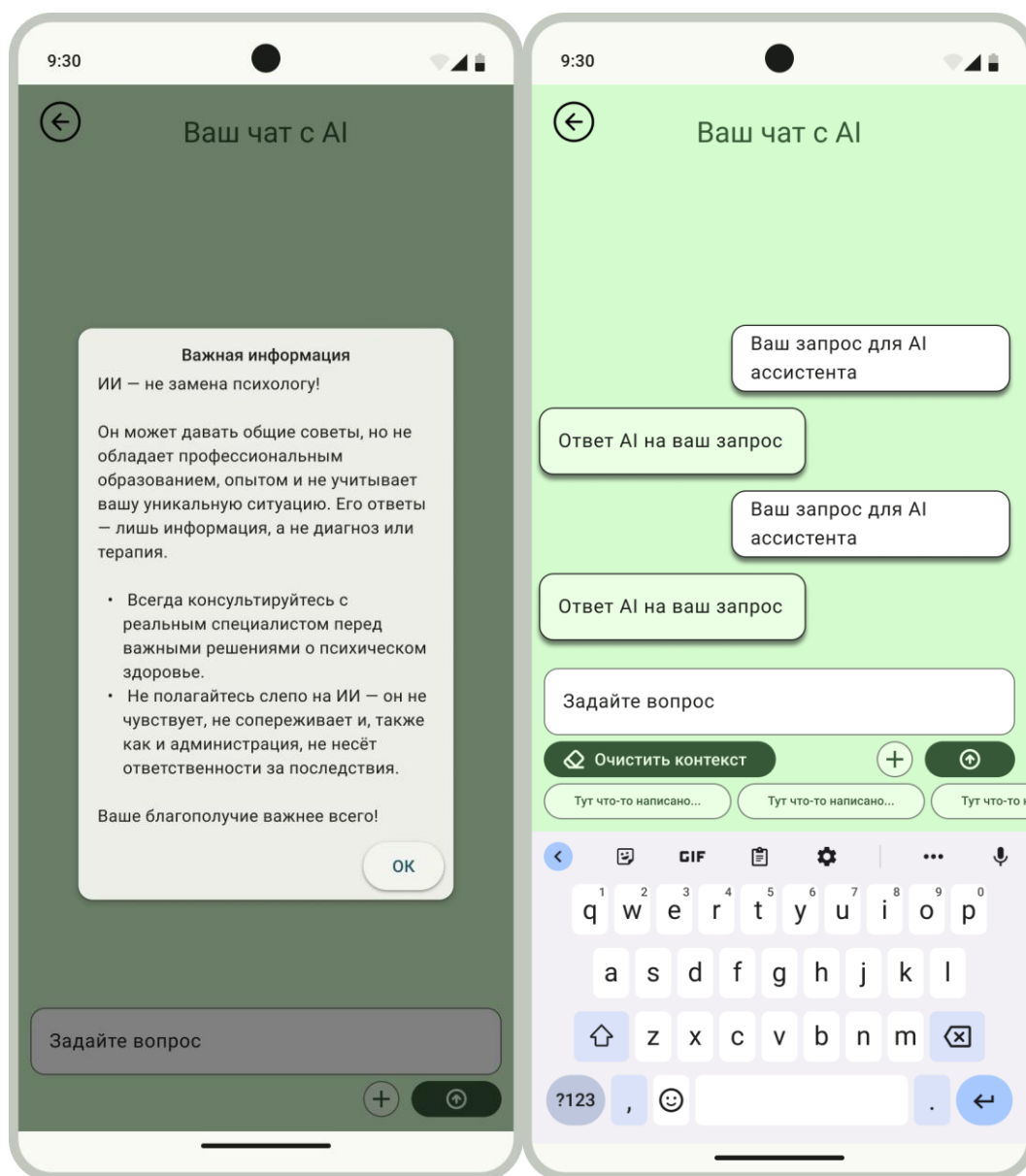


Рисунок 10 - Экран чата с ИИ

3.3.7 Экран добавления виджетов

Экран позволяет пользователям настраивать и добавлять на главный экран своего устройства компактные информационные блоки, отображающие ключевые данные из приложения. Виджеты обеспечивают быстрый доступ к важной информации без необходимости открывать приложение.

Список доступных виджетов:

- «Стрик пользователя»:

Отображает текущую серию дней подряд с записями эмоций

– «Эмоциональное состояние»:

Показывает последнюю зафиксированную эмоцию.



Рисунок 11 - Экраны добавления виджетов

3.3.8 Экран админ-панели

Экран админ-панели предоставляет администраторам приложения инструменты для управления пользователями и обработки жалоб.

Меню навигации:

- Пользователи
- Жалобы

Функциональные возможности:

- Управления пользователями
 - Просмотр списка всех аккаунтов
 - Фильтрация
 - Удаление
 - Просмотр информации
- Управление жалобами:
 - Список всех обращений
 - Фильтрация
 - Просмотр информации о жалобе
 - Блокировка пользователя
 - Блокировка Социального взаимодействия

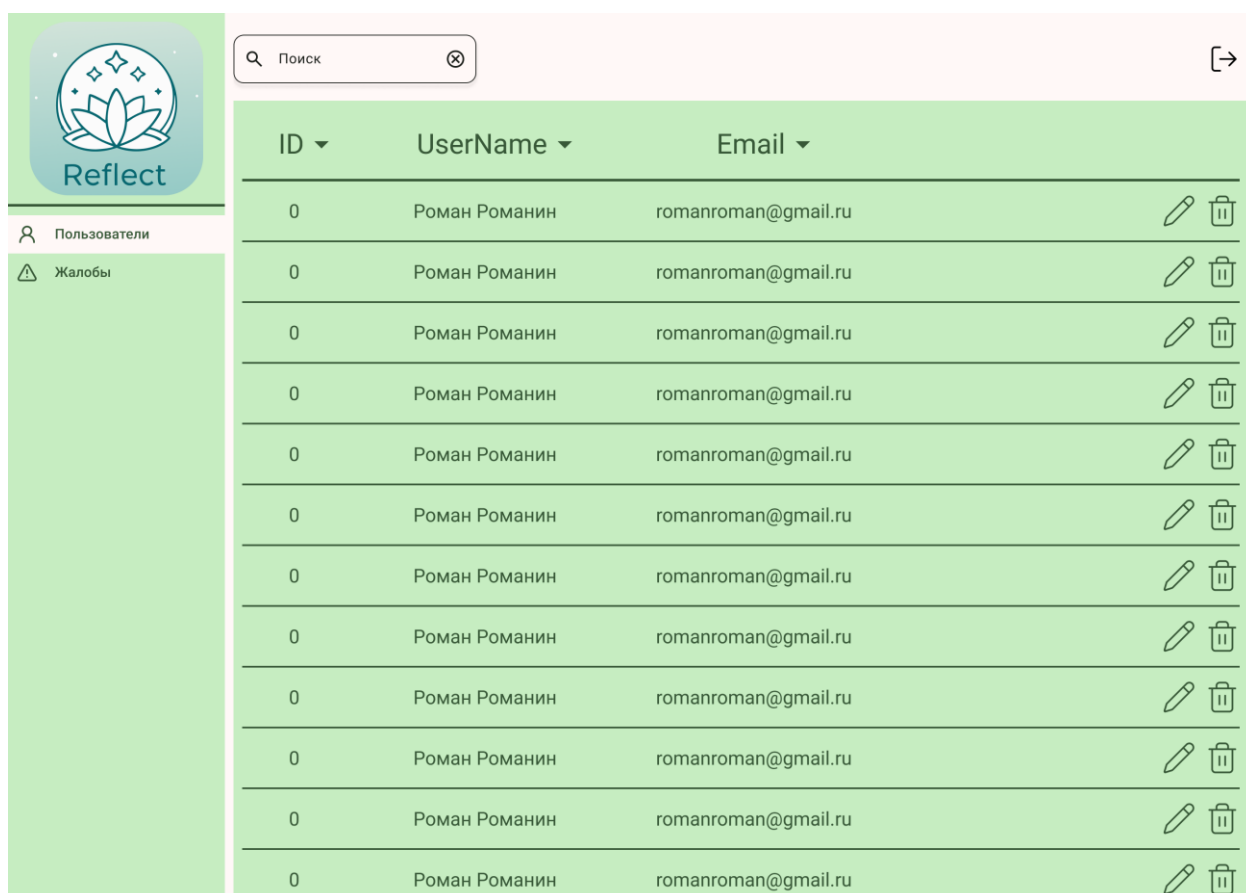


Рисунок 12 - Экран админ-панели – раздел Пользователи

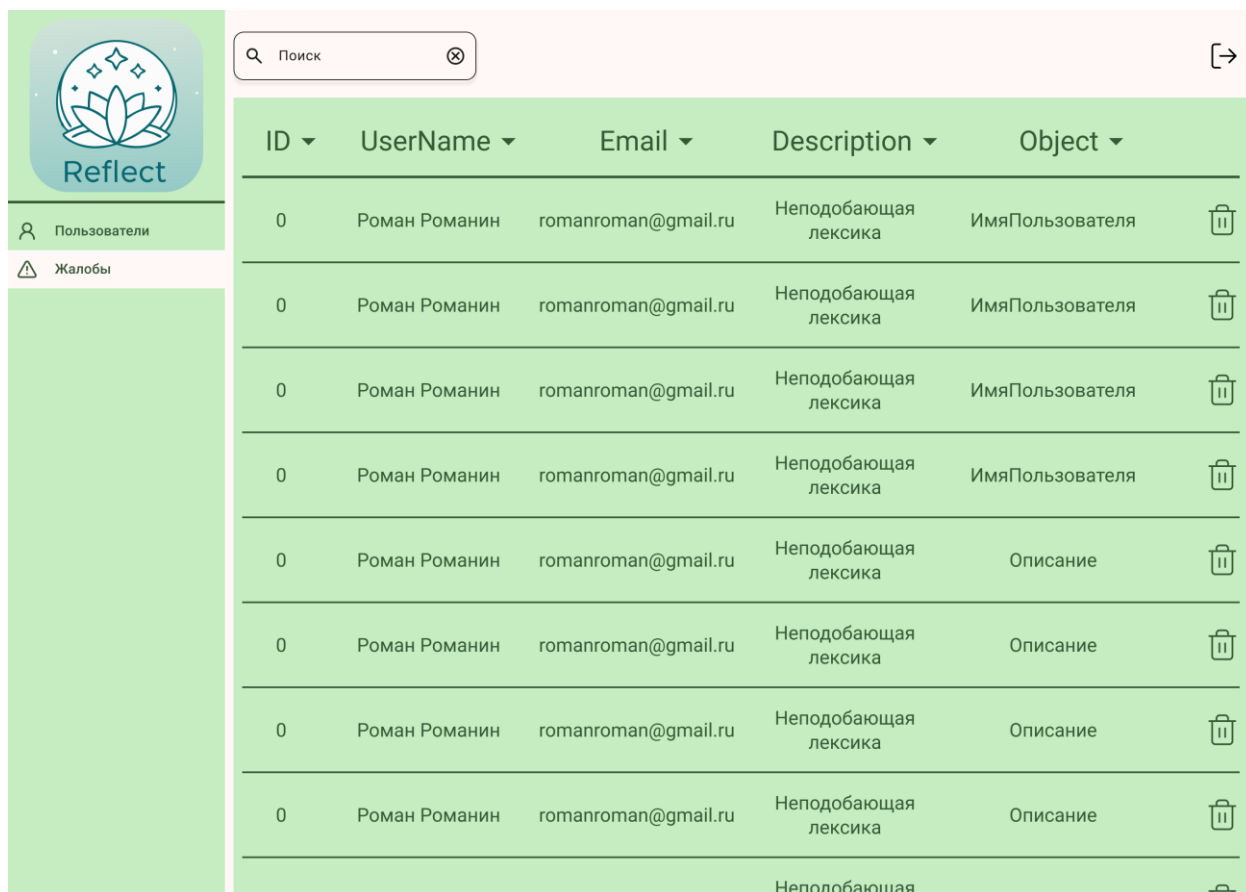


Рисунок 13 - Экран админ-панели – раздел Жалобы

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы над проектом было разработано мобильное приложение Reflect, предназначенное для улучшения самопонимания и эмоционального благополучия пользователей. Приложение предоставляет функционал для ежедневной фиксации эмоций: пользователи могут выбирать настроение по шкале от 1 до 10, уточнять его с помощью детализированного списка эмоций, добавлять текстовые описания и релевантные теги (например, «работа», «семья», «отдых»).

На основе этих записей Reflect формирует детальную статистику, отображаемую в виде графиков изменения настроения за выбранный период и круговых диаграмм частоты различных эмоций – для премиум пользователей. Ключевой особенностью является интегрированный ИИ-ассистент (на базе модели DeepSeek), который анализирует историю записей пользователя и через диалоговый интерфейс предлагает персонализированные рекомендации и техники для улучшения психологического состояния или преодоления сложных эмоциональных периодов.

Приложение также поддерживает социальное взаимодействие: пользователи могут добавлять друзей, просматривать их эмоциональную активность (при наличии соответствующего разрешения) и таким образом оказывать взаимную поддержку. Для повышения вовлеченности реализованы виджеты для главного экрана смартфона, отображающие «стрик» (непрерывную серию дней с записями) и последнюю зафиксированную эмоцию. Разработан также функционал администратора для модерации контента и управления жалобами пользователей.

Техническая реализация на платформе Android с применением архитектурных паттернов Clean Architecture и MVI (Model-View-Intent) обеспечила четкое разделение логики, удобство тестирования и предсказуемое поведение пользовательского интерфейса. Серверная часть,

разработанная на Python с использованием фреймворка Django и базы данных PostgreSQL, отвечает за хранение пользовательских данных, аутентификацию и обработку API-запросов, включая взаимодействие с ИИ-моделью.

Таким образом, разработанное приложение Reflect представляет собой конкретный инструмент, позволяющий пользователям не просто пассивно отслеживать, но и активно работать со своим эмоциональным состоянием через механизмы саморефлексии, анализа накопленных данных и получения конструктивных советов от искусственного интеллекта. Реализованная архитектура и выбранный технологический стек обеспечивают стабильность, производительность приложения и создают прочную основу для его дальнейшего развития и масштабирования.

Список использованных источников

1. Психическое здоровье: доклад ВОЗ о глобальной ситуации – [электронный ресурс]. – URL: <https://news.un.org/ru/story/2022/06/1426022> (дата обращения: 24.03.2025). – Текст: электронный.
2. Clean Architecture для Android: принципы и реализация – [электронный ресурс]. – URL: <https://developer.android.com/topic/architecture> (дата обращения: 15.05.2025). – Текст: электронный.
3. MVI (Model-View-Intent) в Android-разработке – [электронный ресурс]. – URL: <https://proandroiddev.com/mvi-architecture-in-android-c02a1a1c2ba4> (дата обращения: 15.05.2025). – Текст: электронный.
4. Официальная документация Kotlin – [электронный ресурс]. – URL: <https://kotlinlang.org/docs/home.html> (дата обращения: 16.05.2025). – Текст: электронный.
5. Jetpack Compose: современный UI для Android – [электронный ресурс]. – URL: <https://developer.android.com/jetpack/compose> (дата обращения: 16.05.2025). – Текст: электронный.
6. Retrofit и OkHttp: работа с API в Android – [электронный ресурс]. – URL: <https://square.github.io/retrofit/> (дата обращения: 17.05.2025). – Текст: электронный.
7. DeepSeek-R1: документация NLP-модели – [электронный ресурс]. – URL: <https://deepseek.com/docs/models/r1> (дата обращения: 18.05.2025). – Текст: электронный.
8. JWT (JSON Web Tokens): аутентификация в мобильных приложениях – [электронный ресурс]. – URL: <https://jwt.io/introduction> (дата обращения: 19.05.2025). – Текст: электронный.

9. Jetpack Glance: создание виджетов для Android – [электронный ресурс]. –

URL: <https://developer.android.com/develop/ui/views/appwidgets/overview>
(дата обращения: 20.05.2025). – Текст: электронный.