

**Министерство науки и высшего образования Российской Федерации**  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  
**(ФГБОУ ВО «АмГУ»)**

Институт компьютерных и инженерных наук  
Кафедра информационных и управляющих систем  
Направление подготовки 09.03.02 – Информационные системы и технологии  
Направленность (профиль) образовательной программы Информационные системы и технологии

ДОПУСТИТЬ К ЗАЩИТЕ

Зав. кафедрой

\_\_\_\_\_ А.В. Бушманов

« \_\_\_\_\_ » \_\_\_\_\_ 2025 г.

**БАКАЛАВРСКАЯ РАБОТА**

на тему: Разработка мини-приложения для учета индивидуальных спортивных тренировок

Исполнитель

студент группы 1104-об

\_\_\_\_\_

(подпись, дата)

С.Д. Галков

Руководитель

доцент, канд. техн. наук

\_\_\_\_\_

(подпись, дата)

И.М. Акилова

Консультант по безопасности и экологичности

доцент, канд. техн. наук

\_\_\_\_\_

(подпись, дата)

А.Б. Булгаков

Нормоконтроль

инженер кафедры

\_\_\_\_\_

(подпись, дата)

В.Н. Адаменко

Благовещенск 2025

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  
(ФГБОУ ВО «АмГУ»)

Институт компьютерных и инженерных наук  
Кафедра информационных и управляющих систем

УТВЕРЖДАЮ  
Зав. кафедрой  
\_\_\_\_\_ А.В. Бушманов  
«\_\_\_\_\_» \_\_\_\_\_ 2025 г.

### З А Д А Н И Е

К выпускной квалификационной работе студента Галкова С.Д.

1. Тема выпускной квалификационной работы: Разработка мини-приложения для учета индивидуальных спортивных тренировок (утверждено Приказом от 14.04.2025 № 980-уч)

2. Срок сдачи студентом законченной работы (проекта): 10.06.2025 г.

3. Содержание бакалаврской работы (перечень подлежащих разработке вопросов): анализ предметной области и существующих решений; Проектирование мини-приложения; разработка мини-приложения; безопасность и экологичность.

4. Перечень материалов приложения (инфологическая модель базы данных, логическая модель базы данных, таблиц, физическая модель базы данных, схем, программных продуктов, иллюстративного материала и т.п.):

5. Дата выдачи задания: 02.10.2024г.

Руководитель выпускной квалификационной работы: доцент, канд. техн. наук И. М. Акилова

Задание принял к исполнению(дата): 02.10.2024г. \_\_\_\_\_  
(подпись)

## РЕФЕРАТ

Отчет содержит 93 страницы, 41 рисунок, 26 таблиц, 4 приложения, 18 источников.

БАЗА ДАННЫХ, SQLALCHEMY, POSTGRESQL, СУЩНОСТЬ, АТТРИБУТ, FASTAPI, PYTHON, REACT, TELEGRAM MINI APPS, УЧЕТ ТРЕНИРОВОК, ОТСЛЕЖИВАНИЕ ПРОГРЕССА.

Цель работы – разработка мини-приложения для платформы Telegram Mini Apps, предоставляющего пользователям удобный инструмент для учета индивидуальных спортивных тренировок, анализа прогресса и доступа к релевантной информации.

В ходе работы проведен анализ предметной области учета спортивной деятельности, современных мобильных приложений для фитнеса и существующих решений, что обосновало актуальность разработки. Исследованы особенности платформы Telegram Mini Apps, на основе чего сформулированы функциональные и нефункциональные требования к мини-приложению.

Спроектирована архитектура клиент-серверного приложения: серверная часть на Python с использованием фреймворка FastAPI и клиентская, как Telegram Mini App, – на JavaScript. Разработана структура базы данных с использованием СУБД PostgreSQL и ORM SQLAlchemy для хранения данных о пользователях, их тренировках, персонализированных схемах, упражнениях и информационных статьях. Реализован пользовательский интерфейс и API для взаимодействия между клиентской и серверной частями.

# СОДЕРЖАНИЕ

Введение	9
1 Анализ предметной области и существующих решений	12
1.1 Особенности учета индивидуальных спортивных тренировок	12
1.1.1 Значение систематического подхода к тренировкам	12
1.1.2 Основные параметры тренировок, подлежащие учету	14
1.1.3 Анализ целевой аудитории и ее ключевых потребностей	15
1.2 Обзор и сравнительный анализ существующих решений	16
1.2.1 Анализ приложения Jefit	18
1.2.2 Анализ приложения «Фитнес тренировки в зале, дома»	25
1.2.3 Сводный сравнительный анализ	30
1.3 Особенности платформы Telegram Mini Apps	31
1.4 Формулировка требований к мини-приложению	33
2 Проектирование мини-приложения	35
2.1 Функциональные возможности и сценарии использования	35
2.2 Определение архитектуры и выбор инструментов разработки	37
2.3 Проектирование базы данных	40
2.3.1 Инфологическое проектирование	40
2.3.2 Логическое проектирование	48
2.3.3 Физическое проектирование	58
3 Разработка мини-приложения	64
3.1 Реализация клиентской части	64
3.2 Реализация серверной части	67
3.3 Демонстрация работы мини-приложения	70
4 Безопасность и экологичность	81
4.1 Безопасность	81
4.2 Экологичность	84
4.3 Чрезвычайные ситуации	87
Заключение	91

Библиографический список	92
Приложение А	94
Приложение Б	102
Приложение В	103
Приложение Г	104

## ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ

API – интерфейс программирования приложений; набор правил и протоколов, позволяющий различным программным компонентам взаимодействовать друг с другом

БД – база данных

ПО – программное обеспечение

Повторение – одно полное выполнение упражнения

Подход – серия последовательно выполненных повторений упражнения

Рабочий вес – вес отягощения, используемый для выполнения упражнения в тренировке

Технологический стек – набор программных технологий, используемых для разработки и приложения

Силовая тренировка – упражнения с сопротивлением, направленные на увеличение мышечной силы и массы

СУБД – система управления базами данных

ТМА – Telegram Mini App; веб-приложение, работающее непосредственно внутри мессенджера Telegram

UI/UX – пользовательский интерфейс/пользовательский опыт

Freemium – бизнес-модель, при которой базовый набор функций продукта предоставляется бесплатно, а расширенные возможности доступны по платной подписке

Эндпоинт – конечная точка программного интерфейса (API) для выполнения запросов

## НОРМАТИВНЫЕ ССЫЛКИ

В настоящей выпускной квалификационной работе использованы ссылки на следующие стандарты и нормативные документы:

Федеральный закон № 89-ФЗ от 24 июня 1998 г. «Об отходах производства и потребления» : (в ред. от 04.08.2023) // Собрание законодательства РФ. – 1998. – № 26. – Ст. 3009.

Федеральный закон № 123-ФЗ от 22 июля 2008 г. «Технический регламент о требованиях пожарной безопасности» : (в ред. от 25.12.2023) // Собрание законодательства РФ. – 2008. – № 30 (ч. 1). – Ст. 3579.

Приказ Министерства труда и социальной защиты Российской Федерации от 29 октября 2021 г. N 774н «Об утверждении общих требований к организации безопасного рабочего места» : (зарегистрирован в Минюсте России 07.12.2021 N 66212). – Москва, 2021.

Постановление Правительства РФ от 16 сентября 2020 г. N 1479 «Об утверждении Правил противопожарного режима в Российской Федерации» : (в ред. от 30.12.2023) // Собрание законодательства РФ. – 2020. – № 39. – Ст. 6056.

СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания» : постановление Главного государственного санитарного врача Российской Федерации от 28 января 2021 г. N 2 : (зарегистрирован в Минюсте России 29.01.2021 N 62296). – Москва, 2021.

СанПиН 2.1.3684-21 «Санитарно-эпидемиологические требования к содержанию территорий городских и сельских поселений, к водным объектам, питьевой воде и питьевому водоснабжению населения, атмосферному воздуху, почвам, жилым помещениям, эксплуатации производственных, общественных помещений, организации и проведению санитарно-противоэпиде-

мических (профилактических) мероприятий» : постановление Главного государственного санитарного врача Российской Федерации от 28 января 2021 г. N 3 : (зарегистрирован в Минюсте России 29.01.2021 N 62297). – Москва, 2021.

ГОСТ Р 50948-2001. Средства отображения информации индивидуального пользования. Общие эргономические требования и требования безопасности. – Введ. 2002-07-01. – Москва : Изд-во стандартов, 2001. – 12 с.

ГОСТ Р 52872—2019. Интернет-ресурсы и другая информация, представленная в электронно-цифровой форме. Приложения для стационарных и мобильных устройств, иные пользовательские интерфейсы. Требования доступности для людей с инвалидностью и других лиц с ограничениями жизнедеятельности. – Введ. 2020-09-01. – Москва : Стандартинформ, 2019. – 38 с.

ГОСТ Р 58751-2019. Слаботочные системы. Кабельные системы. Телекоммуникационные пространства и помещения. Рабочее место. – Введ. 2020-03-01. – Москва : Стандартинформ, 2019. – 12 с.

ГОСТ Р ИСО 9241-161-2016. Эргономика взаимодействия человек-система. Часть 161. Элементы графического пользовательского интерфейса. – Введ. 2017-06-01. – Москва : Стандартинформ, 2016. – 42 с.

Стандарт организации. Оформление выпускных квалификационных и курсовых работ (проектов) / ФГБОУ ВО «Амурский государственный университет» ; СТО СМК 4.2.3.21-2018. – Благовещенск : АмГУ, 2018. – 75 с.

## ВВЕДЕНИЕ

Развитие информационных технологий и их повсеместное проникновение во все сферы человеческой деятельности открывают новые возможности для повышения качества жизни, оптимизации различных процессов и удовлетворения индивидуальных потребностей. Одной из областей, где цифровые инструменты находят все более широкое применение, является сфера физической культуры и спорта. Регулярные занятия спортом и отслеживание индивидуального прогресса становятся важными аспектами для многих людей, стремящихся к улучшению здоровья и достижению личных спортивных целей.

В настоящей выпускной квалификационной работе рассмотрен процесс разработки мини-приложения для учета индивидуальных спортивных тренировок, предназначенного для использования в экосистеме мессенджера Telegram.

Актуальность темы исследования определяется возрастающим интересом населения к здоровому образу жизни и самостоятельным тренировкам. При этом традиционные методы фиксации тренировочных данных, такие как бумажные дневники или электронные таблицы, не всегда обеспечивают должный уровень удобства, наглядности и возможностей для анализа прогресса. Существующие на рынке мобильные приложения зачастую требуют отдельной установки, могут быть перегружены функционалом или оказаться недоступными для российских пользователей ввиду санкционных ограничений. Таким образом, создание легковесного, доступного и функционального инструмента, интегрированного в популярную в Российской Федерации коммуникационную платформу, является своевременной и востребованной задачей.

Тематика разработки программных средств для фитнеса и учета тренировок исследована достаточно широко. Существует множество коммерче-

ских и бесплатных приложений с различным набором функций. Однако специализированные исследования, посвященные разработке именно мини-приложений (Telegram Mini Apps) для детализированного учета спортивных тренировок с учетом специфики российской аудитории и текущих технологических реалий, представлены в меньшей степени. Большинство работ фокусируется на общих принципах создания мобильных приложений или на архитектуре крупных фитнес-экосистем, что оставляет пространство для разработки нишевых, интегрированных решений.

Целью выпускной квалификационной работы является разработка мини-приложения, представляющего собой легковесное веб-приложения, которое не требует установки на мобильное устройство, предназначенное для учета индивидуальных спортивных тренировок на платформе Telegram Mini Apps, предоставляющего пользователям удобный интерфейс для фиксации результатов, анализа прогресса и доступа к тренировочным материалам.

Для достижения поставленной цели необходимо решить следующие задачи:

- провести анализ предметной области учета индивидуальных спортивных тренировок и выявить ключевые потребности целевой аудитории;
- выполнить обзор и сравнительный анализ существующих программных решений для учета тренировок, определить их достоинства и недостатки;
- обосновать выбор платформы Telegram Mini Apps и технологического стека для разработки мини-приложения;
- спроектировать архитектуру клиент-серверного приложения, включая структуру базы данных и пользовательский интерфейс;
- реализовать основные функциональные модули мини-приложения в соответствии с проектными спецификациями;
- провести тестирование разработанного программного продукта для оценки его работоспособности и удобства использования;
- подготовить раздел, посвященный вопросам безопасности жизнедеятельности и охраны труда при разработке программного обеспечения.

Объектом исследования является процесс проектирования и разработки программного обеспечения для мобильных платформ в контексте задач учета спортивной деятельности.

Предметом исследования выступают архитектура, функциональные возможности и методы реализации мини-приложения для учета индивидуальных спортивных тренировок на платформе Telegram Mini Apps.

Методология исследования основывается на применении следующих методов:

- анализ и обобщение научно-технической литературы, в области разработки программного обеспечения и предметной области спортивных тренировок;

- сравнительный анализ функциональных и технических характеристик существующих программных продуктов-аналогов;

- методы системного подхода и структурного анализа при проектировании архитектуры приложения и базы данных;

- принципы объектно-ориентированного программирования при реализации программных модулей;

- эмпирические методы, включая тестирование и отладку программного кода.

Практическая значимость работы состоит в создании программного продукта, способного удовлетворить потребности пользователей в простом и эффективном инструменте для ведения дневника тренировок. Приложение решает такие проблемы, как сложность систематического отслеживания прогресса, забывание деталей предыдущих занятий и необходимость поиска информации в разрозненных источниках. Разработка ориентирована на российских пользователей Telegram и учитывает текущую доступность технологических платформ.

# 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И СУЩЕСТВУЮЩИХ РЕШЕНИЙ

Исследование предметной области индивидуальных тренировок включает анализ пользовательских потребностей, обзор существующих программных решений и обоснование выбора технологий. Результаты анализа формируют требования к мини-приложению, направленные на удовлетворение ключевых потребностей пользователей.

## 1.1 Особенности учета индивидуальных спортивных тренировок

### 1.1.1 Значение систематического подхода к тренировкам

Современный тренд на поддержание здорового образа жизни и регулярные занятия физической культурой охватывает все более широкие слои населения. Индивидуальные спортивные тренировки, будь то занятия в тренажерном зале, на уличных площадках или в домашних условиях, становятся неотъемлемой частью повседневной рутины для многих людей, стремящихся к улучшению физической формы, достижению конкретных спортивных результатов или просто поддержанию общего тонуса организма.

Эффективность тренировочного процесса во многом определяется его систематичностью и возможностью объективной оценки результатов. Систематический подход подразумевает регулярность занятий, последовательное увеличение нагрузок (принцип прогрессии), адекватное восстановление и адаптацию программ тренировок под изменяющиеся цели и текущее состояние организма. Отслеживание прогресса является ключевым элементом этого подхода, так как позволяет:

- оценивать эффективность выбранной программы тренировок. Фиксация данных о выполненных упражнениях, рабочих весах, количестве повторений и подходов дает возможность понять, приводит ли текущая методика к желаемым изменениям (например, росту силовых показателей, увеличению выносливости, изменению антропометрических данных);

– поддерживать мотивацию. Визуализация достигнутых результатов, даже небольших, служит мощным мотивирующим фактором, помогая преодолевать периоды плато или снижения энтузиазма. Наблюдение за собственным прогрессом укрепляет уверенность в своих силах и стимулирует к продолжению занятий;

– своевременно корректировать тренировочный план. Анализ данных позволяет выявлять застои в результатах, определять упражнения или методы, которые не дают эффекта, и вносить необходимые изменения в программу. Это помогает избежать травм от перетренированности или, наоборот, от недостаточной нагрузки;

– предотвращать регресс. Забывание деталей предыдущих тренировок, особенно рабочих весов или количества повторений, может привести к необоснованному снижению нагрузки и, как следствие, к потере достигнутых результатов.

Циклическая природа тренировочного процесса, где учет и анализ играют роль обратной связи, наглядно представлена на рисунке 1.



Рисунок 1 – Циклическая модель тренировочного процесса с обратной связью

Отсутствие системного учета, как видно из представленной модели, разрывает цикл обратной связи, превращая тренировки в хаотичный набор действий, что значительно снижает их результативность и может привести к разочарованию и прекращению занятий.

#### 1.1.2 Основные параметры тренировок, подлежащие учету

Для эффективного анализа и планирования тренировочного процесса, особенно в контексте фитнес-тренировок, включающих преимущественно упражнения с собственным весом и отягощениями («железом»), необходимо фиксировать ряд ключевых параметров. Данные параметры составляют основу для последующего анализа прогресса и принятия решений о корректровке тренировочной программы.

К основным параметрам, подлежащим учету, относятся:

- дата и время тренировки. Позволяет отслеживать регулярность занятий и периоды восстановления;
- тип тренировки. Например, круговая, сплит-тренировка (по группам мышц);
- список выполненных упражнений. Точные наименования упражнений (например, «Жим штанги лежа», «Приседания со штангой на плечах», «Подтягивания широким хватом»);
- рабочий вес. Используемый вес отягощения для каждого подхода в силовых упражнениях;
- количество повторений. Число выполненных движений в каждом подходе;
- количество подходов (сетов). Общее число серий выполнения упражнения;
- время отдыха между подходами. Важный параметр, влияющий на интенсивность и направленность тренировки;
- общая продолжительность тренировки;

– самочувствие и комментарии. Субъективная оценка состояния, заметки об особенностях выполнения упражнений, болевых ощущениях, уровне энергии и т.д.

Хотя для других видов физической активности (например, кардиотренировок на выносливость) могут быть важны иные параметры (дистанция, темп, ЧСС), для целевой аудитории, сфокусированной на фитнесе и работе с весами, перечисленные выше параметры являются базовыми и наиболее значимыми.

### 1.1.3 Анализ целевой аудитории и ее ключевых потребностей

Успех любого программного продукта во многом зависит от того, насколько точно он отвечает запросам и решает проблемы своей целевой аудитории.

Целевая аудитория определена как люди, самостоятельно тренирующиеся в спортивном зале или дома, от новичков до среднего уровня. Данная группа пользователей характеризуется следующими особенностями:

– самостоятельность. Пользователи преимущественно не прибегают к услугам персональных тренеров на постоянной основе, либо пользуются ими эпизодически. Они сами планируют свои тренировки и отслеживают результаты;

– уровень подготовки. От новичков, которые только начинают свой путь в фитнесе и нуждаются в базовых знаниях и простых инструментах, до атлетов среднего уровня, уже имеющих определенный опыт, но стремящихся к дальнейшему прогрессу и систематизации своих занятий;

– возраст и технологическая грамотность. Возрастной диапазон 18-40 лет предполагает активное использование современных технологий, смартфонов и мессенджеров. Пользователи Telegram, как правило, ценят удобство, скорость и интеграцию сервисов в привычную для них среду;

– мотивация. Основные мотивы – улучшение физической формы, рост силовых показателей, набор мышечной массы, похудение, поддержание здоровья.

Можно выделить следующие ключевые проблемы, с которыми сталкивается целевая аудитория и которые призвано решить разрабатываемое приложение:

- сложность отслеживания прогресса в традиционных форматах. Ведение записей в бумажных тетрадях или таблицах Excel требует дисциплины, не всегда удобно (особенно в условиях тренажерного зала), затрудняет быстрый анализ данных и визуализацию прогресса. Такие записи легко потерять или повредить;

- отсутствие готовых тренировочных схем для новичков в одном месте. Начинающие спортсмены часто испытывают трудности с составлением эффективных и безопасных программ тренировок для фитнеса. Поиск достоверной информации в интернете может быть затруднен обилием противоречивых сведений. Наличие проверенных базовых программ в приложении могло бы значительно упростить старт;

- забывание упражнений и рабочих весов предыдущих тренировок. Особенно актуально при нерегулярных занятиях или при большом разнообразии упражнений. Это приводит к потере времени на вспоминание, риску выбора неадекватной нагрузки (слишком легкой или слишком тяжелой) и, как следствие, снижению эффективности тренировочного процесса.

Таким образом, анализ особенностей учета тренировок и потребностей целевой аудитории показывает наличие явного запроса на простой, удобный и функциональный цифровой инструмент.

## **1.2 Обзор и сравнительный анализ существующих решений**

Для объективного определения рыночной ниши разрабатываемого мини-приложения и выявления его потенциальных конкурентных преимуществ первостепенное значение имеет проведение анализа существующих на рынке программных продуктов, предназначенных для учета индивидуальных спортивных тренировок. Традиционные методы, такие как ведение бумажных дневников или использование электронных таблиц, хотя и являются до-

ступными, обладают рядом существенных недостатков, включая трудоемкость, отсутствие автоматизации и сложности в анализе прогресса, что и обуславливает потребность в специализированных цифровых решениях. Особое внимание в данном обзоре будет уделено программным продуктам, актуальным для пользователей в Российской Федерации, с учетом текущей международной обстановки, доступности сервисов и потребительских предпочтений.

Для всестороннего и структурированного сравнения существующих приложений и сервисов будут использованы следующие ключевые критерии:

- возможность детализированного учета тренировок. Оценивается глубина фиксации данных по упражнениям, включая запись подходов, повторений, используемых весов, а также времени отдыха между подходами;

- наличие и качество базы упражнений. Анализируется объем встроенной базы упражнений, наличие описаний техники выполнения, иллюстраций или видеоматериалов, а также возможность добавления пользователем собственных, кастомных упражнений;

- работа с тренировочными программами. Рассматривается наличие готовых программ тренировок, предназначенных для различных уровней подготовки и спортивных целей, а также функционал для создания, редактирования и сохранения персональных тренировочных планов;

- инструменты отслеживания прогресса. Оцениваются возможности для визуализации динамики результатов, такие как построение графиков изменения рабочих весов или количества повторений по упражнениям, ведение статистики выполненных тренировок и общей активности;

- наличие дополнительных функций. Учитывается присутствие сопутствующего полезного функционала, расширяющего пользовательский опыт, например, информационных статей по тематике фитнеса и здорового образа жизни, калькуляторов (индекса массы тела, суточной нормы калорий и т.п.) или социальных взаимодействий (если применимо);

– платформенная доступность и локализация. Определяется, на каких платформах доступно приложение (мобильное приложение для iOS/Android, веб-сервис, мини-приложение), его фактическая доступность для загрузки и использования на территории РФ, а также наличие качественного русскоязычного интерфейса и поддержки;

– стоимость и модель монетизации. Анализируется ценовая политика: является ли приложение полностью бесплатным, платным с единовременной покупкой, распространяется ли по модели freemium (с бесплатным базовым набором функций и платной подпиской на расширенные возможности) или содержит встроенную рекламу;

– ключевые особенности, преимущества и недостатки. Выделяются уникальные характеристики каждого рассматриваемого решения, его сильные стороны, а также явные ограничения или неудобства с точки зрения целевого пользователя.

### 1.2.1 Анализ приложения Jefit

Одним из наиболее известных и функционально насыщенных международных приложений, предназначенных для учета силовых тренировок, является Jefit. Данное приложение позиционируется разработчиками как комплексный инструмент, подходящий для атлетов любого уровня подготовки – от новичков до профессионалов, благодаря наличию как готовых тренировочных планов, так и широких возможностей для их персонализации.

Функциональные возможности Jefit охватывают все ключевые аспекты тренировочного процесса.

В части учета тренировок, приложение позволяет осуществлять детализированную запись и изменение параметров каждого упражнения, включая рабочий вес, количество выполненных повторений в каждом подходе, а также время отдыха между подходами, для которого предусмотрен встроенный таймер. Пример интерфейса ввода данных о выполненном упражнении представлен на рисунке 2. Для выбранного упражнения, доступны GIF-анимации или видеоролики, демонстрирующие правильную технику. Однако,

по результатам практического ознакомления с приложением, было выявлено, что функция просмотра видео-инструкций по технике выполнения упражнений является платной и доступна только пользователям с Elite-подпиской. Это является значительным недостатком для новичков в бесплатной версии, особенно учитывая сложности с оплатой подписки из РФ.

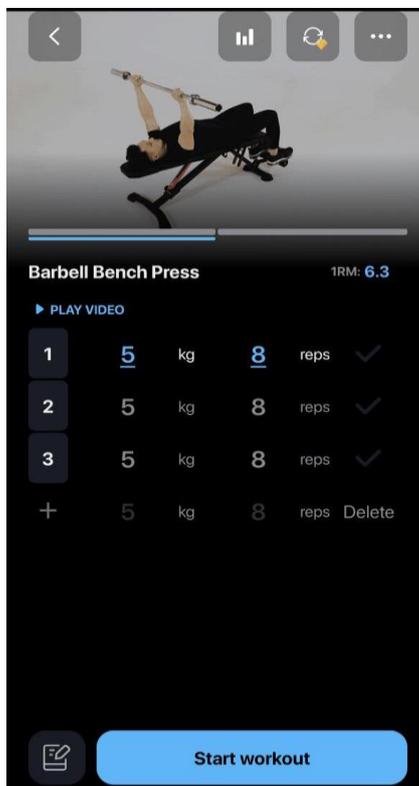


Рисунок 2 – Интерфейс ввода данных о выполненном упражнении в приложении Jefit

Приложение обладает обширной базой упражнений, насчитывающей, по данным интерфейса приложения, 1386 упражнений, что иллюстрируется на рисунке 3. База включает упражнения с различными видами отягощений и для всех основных мышечных групп. Предусмотрен поиск и фильтрация упражнений по целевым мышцам и используемому оборудованию. Пользователь имеет возможность добавлять собственные упражнения, отсутствующие в стандартной базе. Каждое упражнение в базе, как правило, сопровождается текстовым описанием техники выполнения и указанием задействованных мышечных групп.

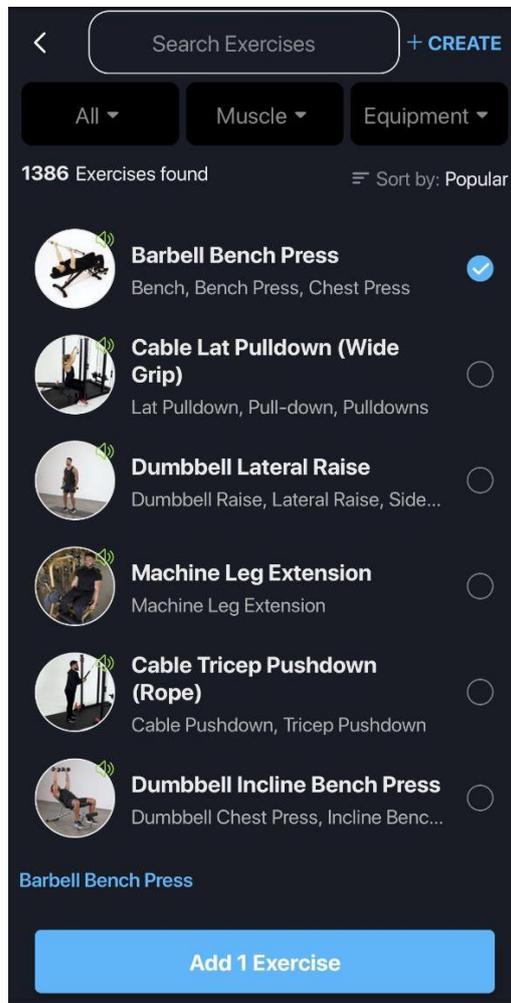


Рисунок 3 – Экран поиска упражнений в приложении JEFIT

В разделе с тренировочными программами, JEFIT предлагает пользователям доступ к большой библиотеке готовых планов, созданных как командой JEFIT, так и сообществом, что можно видеть на рисунке 4. Программы можно фильтровать по различным критериям, таким как цель (набор мышечной массы, похудение, развитие силы), уровень подготовки и доступное оборудование. Однако стоит отметить, что значительная часть наиболее проработанных и комплексных готовых программ тренировок доступна преимущественно в рамках платной Elite-подписки. Бесплатным пользователям может быть предложен ограниченный набор базовых планов или программ.

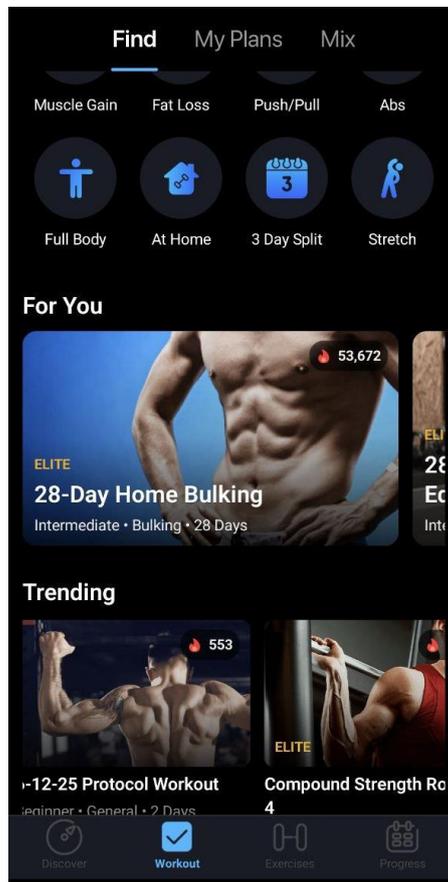


Рисунок 4 – Экран выбора готовых тренировочных программ в Jefit

Важной функцией Jefit, доступной как в бесплатной, так и в платной версии, является гибкий инструментарий для создания и кастомизации собственных тренировочных планов. Пользователи могут формировать свои программы тренировок «с нуля», добавляя упражнения из обширной базы данных или создавая собственные. Как видно на рисунке 5, интерфейс отображает список запланированных на день упражнений с их параметрами и визуальными подсказками, а также предоставляет возможность оперативно дополнять тренировочный план новыми упражнениями непосредственно в процессе его просмотра или выполнения. Jefit также рассчитывает ориентировочную продолжительность запланированной тренировки. Существующие программы, как загруженные из библиотеки, так и созданные ранее, также поддаются редактированию, что дает пользователю полный контроль над

своим тренировочным процессом. Эта возможность делает Jefit привлекательным для пользователей, предпочитающих индивидуальный подход к составлению тренировок, а не только следование готовым шаблонам.

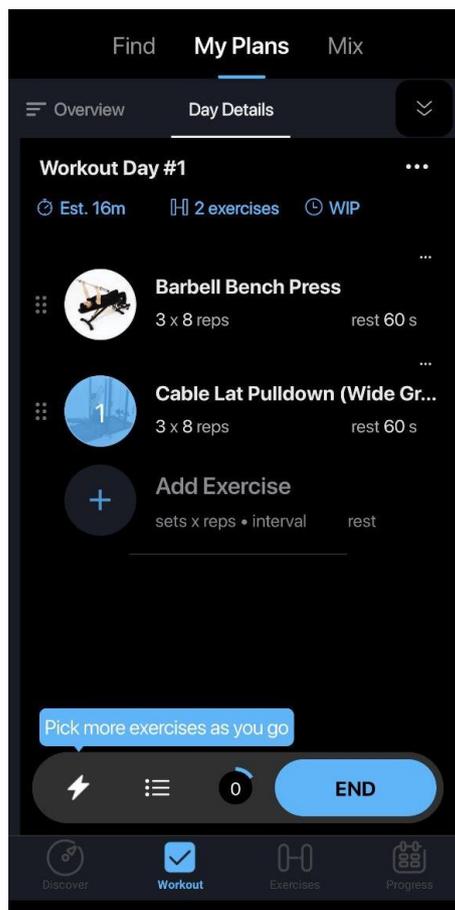


Рисунок 5 – Интерфейс плана тренировочного дня в Jefit

Касательно отслеживания прогресса, Jefit сохраняет историю выполненных тренировок, рабочих весов и повторений по каждому упражнению. На основе этих данных приложение позволяет анализировать динамику силовых показателей. Пользователи могут просматривать графики изменения, например, расчетного одноповторного максимума или общего тренировочного объема (последний, доступен в Elite-версии). На рисунке 6 представлен пример такого графика, где отображается динамика одноповторного максимума за выбранный период, с указанием максимального и минимального достигнутых значений. Приложение позволяет фильтровать данные по различным временным интервалам, например, за последние 14 дней, месяц, год, для наглядной оценки долгосрочного или краткосрочного прогресса.



Рисунок 6 – Пример графика отслеживания прогресса по упражнению

Из дополнительных функций Jefit стоит отметить наличие калькуляторов для расчета одноповторного максимума и индекса массы тела, а также социальные функции для взаимодействия с международным сообществом пользователей. Заявлена интеграция с сервисами Google Fit, Apple Health и некоторыми умными часами.

Касательно платформенной доступности, Jefit функционирует на платформах iOS, Android и имеет веб-интерфейс. Несмотря на формальную доступность приложения для загрузки в РФ, его использование для большинства российских пользователей значительно осложняется полным отсутствием русскоязычной локализации: весь интерфейс и контент представлены исключительно на английском языке.

Приложение Jefit использует модель монетизации freemium: базовый функционал предоставляется бесплатно с показом рекламы, тогда как расширенные возможности (углубленная аналитика, премиум-программы, видеоинструкции, отсутствие рекламы) требуют платной подписки. Однако для пользователей из Российской Федерации возможность оформления такой

подписки стандартными способами в настоящее время фактически отсутствует из-за действующих ограничений международных платежных систем.

Ключевые особенности, преимущества и недостатки:

К преимуществам Jefit можно отнести:

- обширную базу упражнений (более 1300) с текстовыми описаниями и GIF-анимациями (для некоторых упражнений);
- высокую степень детализации при учете параметров силовых тренировок;
- гибкие возможности по созданию и настройке индивидуальных тренировочных программ;
- развитые (хотя преимущественно платные) инструменты для отслеживания прогресса и аналитики.

Основными недостатками Jefit, являются:

- полное отсутствие русскоязычной локализации интерфейса;
- невозможность официальной оплаты Elite-подписки из РФ, что блокирует доступ к важным функциям, включая видео-инструкции к упражнениям и продвинутую аналитику;
- интерфейс, который может показаться перегруженным и сложным для освоения новичками или пользователями, ищущими максимально простое решение;
- наличие рекламы в бесплатной версии.

В целом, Jefit представляет собой мощный и многофункциональный инструмент, ориентированный преимущественно на опытных атлетов, которые ценят детализированный учет, обширную базу упражнений и готовы платить за расширенные возможности, а также комфортно себя чувствуют в англоязычной среде. Однако из-за полного отсутствия русскоязычной локализации интерфейса, фактической невозможности для российских пользователей оплатить подписку и получить доступ к полному функционалу, а также из-за потенциально перегруженного для новичков интерфейса, данное приложение не в полной мере закрывает потребности сегмента пользователей,

ищущих простое, интуитивно понятное и полностью доступное на русском языке решение для базового и среднего уровня учета фитнес-тренировок.

### 1.2.2 Анализ приложения «Фитнес тренировки в зале, дома»

В качестве второго примера для анализа рассмотрим приложение «Фитнес тренировки в зале, дома» от разработчика Fitness Online LLC (далее – Fitness Online App). Данное приложение широко представлено на российском рынке и позиционируется как комплексное решение, предлагающее пользователям не только функции учета тренировок, но и обширный дополнительный контент и сервисы, что смещает его фокус от простого дневника в сторону многофункциональной фитнес-платформы.

При анализе функциональных возможностей Fitness Online App, в сравнении с Jefit, можно выделить ряд особенностей. Приложение позволяет фиксировать упражнения, подходы, повторения и вес, а также использовать таймер для отдыха, что иллюстрирует рисунок 7, демонстрирующий интерфейс выполнения подхода.

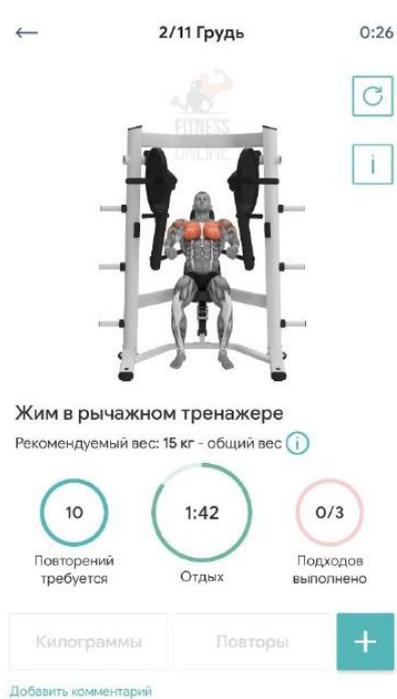


Рисунок 7 – Интерфейс выполнения подхода в упражнения в Fitness Online App

База упражнений в приложении является обширной и включает визуальные материалы для демонстрации техники. Пользователям также представлена возможность добавления собственных упражнений. Важной особенностью является наличие большого выбора готовых тренировочных программ, однако, как показано на рисунке 8, доступ к полным версиям многих из них (кроме первого тренировочного дня) ограничен в бесплатной версии и требует оформления платной подписки.

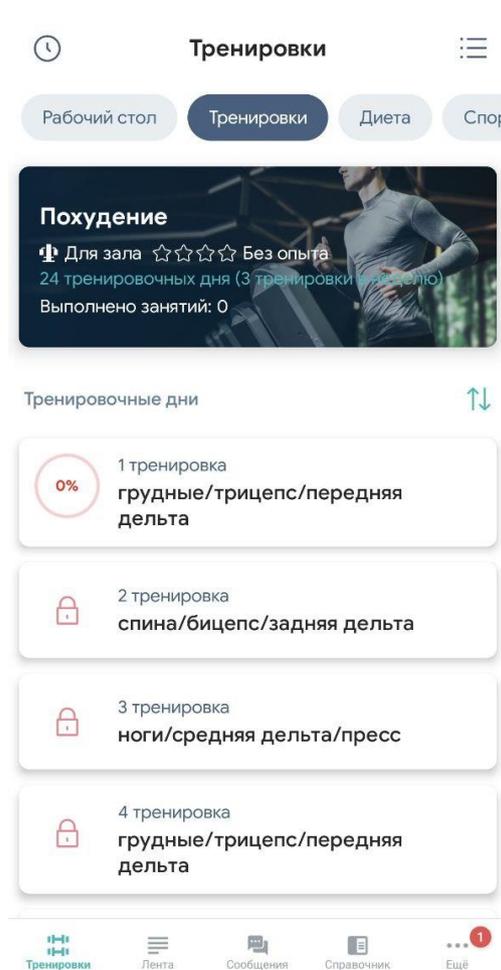


Рисунок 8 – Экран тренировочной программы «Похудение» в Fitness Online App

Инструменты отслеживания прогресса в Fitness Online App включают статистику по проработке мышечных групп, типам упражнений, общей длительности занятий и расходу калорий. Однако, в отличие от Jefit, явных и легкодоступных графиков, демонстрирующих долгосрочную прогрессию

именно силовых показателей (рост рабочих весов или повторений) по конкретным упражнениям, в ходе ознакомления с приложением выявлено не было, что может быть недостатком для пользователей, сфокусированных на силовом тренинге.

Отличительной чертой Fitness Online App является акцент на дополнительные функции и социальную составляющую. Присутствует «Спортивная лента», где пользователи могут публиковать свои результаты и фотографии, что придает приложению характер социальной сети.



Рисунок 9 – Пример «Спортивной ленты» в приложении Fitness Online App.

Кроме того, в приложение интегрированы обширные справочные материалы по спортивному питанию, калорийности продуктов. Примечательно, что функция учета питания («Диета») интегрирована непосредственно в ос-

новой раздел тренировок и доступна бесплатно, позволяя фиксировать приемы пищи, как показано на рисунке 10. Такое обилие дополнительного, хотя и бесплатного, функционала может как привлекать пользователей, ищущих комплексное решение, так и отвлекать тех, кому нужен исключительно дневник тренировок.

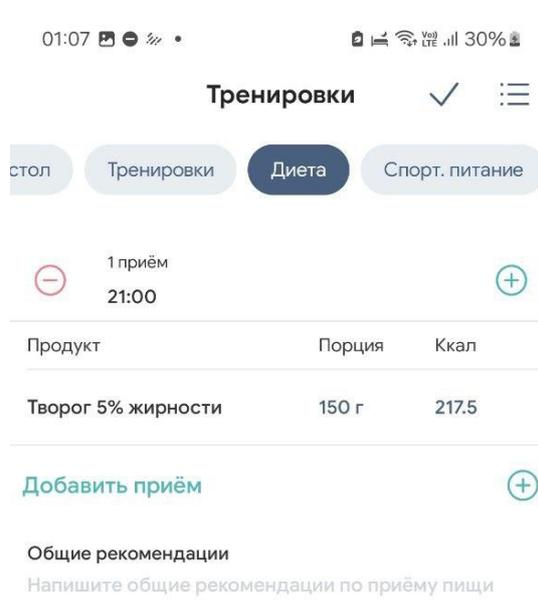


Рисунок 10 – Раздел учета питания («Диета») в приложении Fitness Online App.

Платформенная доступность Fitness Online App обеспечивается для устройств на базе Android, также вероятно наличие версии для iOS. Приложение полностью русифицировано и ориентировано на рынок РФ.

Модель монетизации основана на freemium-подходе: базовые функции, включая запись тренировок и доступ к части контента, предоставляются бесплатно, но с существенными ограничениями на доступ к полным тренировочным программам. Платная подписка снимает эти ограничения и открывает доступ к дополнительным премиум-функциям.

К преимуществам Fitness Online App можно отнести:

– полную русскоязычную локализацию и ориентацию на российского пользователя;

- наличие обширной базы упражнений с визуализациями и описаниями;
- большой выбор готовых тренировочных программ (при условии наличия платной подписки);
- интегрированный справочный материал по питанию, упражнениям и другим смежным темам;
- наличие социальных функций (лента, сообщество) для пользователей, ищущих такой тип взаимодействия.

Основными недостатками Fitness Online App, с точки зрения пользователя, ищущего простой и эффективный дневник тренировок, являются:

- существенные ограничения функциональности в бесплатной версии, особенно в части доступа к полным версиям готовых тренировочных программ;
- фокус на широком спектре фитнес-услуг (питание, социальная сеть, справочники), что приводит к перегруженности интерфейса и отвлекает от основной задачи учета тренировок;
- отсутствие инструментов для наглядного отслеживания долгосрочной прогрессии нагрузок (рост весов/повторений) по конкретным упражнениям, что важно для силового тренинга;
- позиционирование скорее, как многофункциональной фитнес-платформы, нежели специализированного дневника тренировок.

В заключение анализа данного приложения можно отметить, что Fitness Online App предлагает широкий набор инструментов для пользователей, заинтересованных не только в учете тренировок, но и в получении готовых программ, справочной информации и участии в социальной жизни фитнес-сообщества. Однако для пользователей, чьей основной целью является простой и сфокусированный учет собственных силовых тренировок и отслеживание прогрессии нагрузок без значительных платных ограничений на базовый функционал, данное решение может оказаться избыточным или не в полной мере удовлетворяющим их специфические потребности.

### 1.2.3 Сводный сравнительный анализ

Проведенный анализ двух различных по своей концепции приложений – Jefit и «Фитнес тренировки в зале, дома» от Fitness Online LLC, а также учет особенностей традиционных методов ведения тренировочных дневников, позволяет провести их комплексное сравнение по ключевым критериям. Такое сопоставление необходимо для выявления сильных и слабых сторон каждого подхода, а также для более четкого понимания существующих на рынке предложений. Результаты данного сравнительного анализа, сфокусированные на наиболее значимых аспектах для целевой аудитории, представлены в таблице 1.

Таблица 1 – Сравнение методов учета тренировок

Критерий	Традиционные методы	Jefit	«Фитнес тренировки в зале, дома» (Fitness Online App)
Основной функционал учета тренировок	ручной, гибкий, но трудоемкий	высокая детализация, таймеры, заметки	детализированный учет, таймеры
Доступность и создание программ (в бесплатной версии)	только ручное создание	ограниченный доступ к готовым, создание своих возможно	сильно ограниченный доступ к готовым (большинство дней платно), создание своих возможно
Отслеживание прогрессии нагрузок (по упражнениям)	ручной анализ, сложная визуализация	детальные графики (часть платно)	отсутствуют явные/легкодоступные графики прогрессии по весам/повторам для конкретных упражнений
Локализация и доступность в РФ (включая оплату премиум-функций)	неприменимо	полностью на английском языке, оплата премиум-функций из РФ фактически невозможна	полностью на русском языке, оплата премиум-функций возможна
Общее удобство использования и модель монетизации (для основной ЦА)	неудобно для анализа, бесплатно	потенциально сложный интерфейс, freemium с ограничениями и проблемами оплаты для пользователей из РФ	потенциально перегруженный интерфейс, freemium с существенными ограничениями на базовые программы для бесплатных пользователей

Анализ данных, представленных в таблице 1, а также текстовых описаний рассмотренных приложений и традиционных методов, позволяет сделать ряд обобщающих выводов.

Традиционные методы ведения дневника тренировок, несмотря на бесплатность и гибкость, характеризуются высокой трудоемкостью, отсутствием автоматизации и сложностями в анализе прогресса.

Среди программных решений, международное приложение Jefit выделяется детализированным учетом и обширной базой упражнений, однако его использование в России сопряжено с отсутствием русскоязычной локализации и фактической невозможностью оплаты премиум-функций. Отечественное приложение «Фитнес тренировки в зале, дома» (Fitness Online App) решает проблему языка и доступности платежей, но его модель монетизации существенно ограничивает бесплатный доступ к готовым тренировочным программам, а фокус на широком спектре дополнительных услуг может приводить к перегруженности интерфейса для пользователей, ищущих именно трекер тренировок. Важным упущением обоих рассмотренных программных продуктов, с точки зрения силового тренинга, является либо платный доступ к детализированным графикам прогрессии (Jefit), либо их отсутствие (Fitness Online App).

Таким образом, на российском рынке сохраняется потребность в специализированном программном обеспечении, которое бы предлагало пользователям удобный русскоязычный интерфейс, достаточный базовый функционал для учета тренировок и отслеживания прогресса на бесплатной основе, и было бы легко доступно, например, через интеграцию с популярными платформами.

### **1.3 Особенности платформы Telegram Mini Apps**

Выбор платформы для реализации приложения является важным стратегическим решением, влияющим на процесс разработки, доступность для пользователей и возможности интеграции. Платформа Telegram Mini Apps

представляет собой веб-приложения, запускаемые непосредственно внутри мессенджера Telegram.

Рассмотрим ключевые аспекты данной платформы применительно к задаче разработки мини-приложения для учета индивидуальных спортивных тренировок.

Преимущества платформы ТМА:

- нативная интеграция и доступность для пользователя. Пользователям не требуется искать и устанавливать отдельное мобильное приложение, так как доступ к ТМА осуществляется непосредственно из интерфейса Telegram. Это значительно снижает барьер для начала использования и обеспечивает ощущение привычной среды мессенджера;

- широкий охват аудитории. Telegram является одним из наиболее популярных мессенджеров в Российской Федерации, что открывает доступ к многомиллионной аудитории потенциальных пользователей без необходимости значительных затрат на маркетинг, характерных для традиционных мобильных приложений;

- упрощенная авторизация и работа с данными пользователя. ТМА позволяют легко и безопасно получать основные данные пользователя из его Telegram-профиля (например, ID, имя) без реализации сложных процедур регистрации;

- использование стандартных веб-технологий. Разработка ТМА базируется на HTML, CSS и JavaScript, что позволяет применять существующие веб-фреймворки и инструменты, ускоряя и упрощая процесс создания;

- легкость обновления. Обновление ТМА происходит аналогично обновлению веб-сайтов – достаточно изменить файлы на сервере, и пользователи при следующем запуске получат актуальную версию;

- взаимодействие с интерфейсом Telegram. Предоставляемый Software Development Kit (далее – SDK) позволяет мини-приложению взаимодействовать с нативным интерфейсом Telegram: адаптировать цветовую схему, управлять стандартными кнопками, получать данные пользователя.

### Ограничения платформы TMA:

- работа в изолированном окружении. TMA функционирует как веб-страница, загружаемая во встроенном в Telegram Iframe, что может накладывать некоторые ограничения на доступ к API устройства по сравнению с нативными приложениями;
- зависимость от SDK Telegram. Взаимодействие с функциями Telegram осуществляется через предоставляемую библиотеку, что требует ее корректного использования;
- требования к UI/UX. Для органичной интеграции в экосистему Telegram рекомендуется следовать определенным инструкциям по дизайну, хотя разработчик и сохраняет значительную свободу;
- производительность. Для очень сложных приложений с интенсивной графикой или обработкой больших данных на клиенте могут возникнуть ограничения производительности, однако для задач разрабатываемого приложения (формы, списки, базовые графики) производительность достаточная;

В целом, для разрабатываемого мини-приложения, нацеленного на предоставление удобного инструмента для учета индивидуальных спортивных тренировок, платформа Telegram Mini Apps представляет собой сбалансированное решение. Она сочетает удобство доступа и широкий охват аудитории с эффективностью разработки на основе веб-технологий, предоставляя при этом достаточный набор функций для реализации запланированной функциональности. Выявленные ограничения не являются критичными для данного проекта и могут быть учтены на этапе проектирования и разработки.

### **1.4 Формулировка требований к мини-приложению**

На основе анализа предметной области, потребностей пользователей и существующих решений формулируются основные требования к разрабатываемому мини-приложению, служащие основой для его дальнейшего проектирования и реализации. Более детальная спецификация требований представлены в приложении А.

### Функциональные требования:

- система должна обеспечивать идентификацию и автоматическую регистрацию пользователя при первом входе на основе его данных из Telegram;
- пользователь должен иметь возможность создавать, просматривать, редактировать и удалять записи о своих тренировках (которые могут быть сформированы вручную или автоматически на основе тренировочных схем), указывая дату, тип, продолжительность, заметки, а также добавлять в тренировку упражнения с указанием названия, количества подходов, повторений, времени отдыха, рабочего веса, дистанции и продолжительности;
- система должна предоставлять доступ к списку информационных статей (базовая теоретическая информация по тренировкам и смежным темам) с возможностью их просмотра;
- пользователь должен иметь возможность отслеживать прогрессию нагрузок на графиках (например, «вес/дата»);
- в профиле пользователя должна отображаться его основная информация из Telegram;
- должен быть реализован календарь для отображения отмеченных пользователем проведенных тренировок;
- должна быть обеспечена интуитивно понятная навигация между основными разделами.

### Нефункциональные требования:

- приложение должно демонстрировать высокую производительность, обеспечивая быстрый отклик на действия пользователя и оперативную загрузку данных;
- система должна быть надежной, гарантируя стабильную работу и сохранность пользовательских данных о тренировках и прогрессе;
- интерфейс должен быть интуитивно понятным, простым в освоении, эстетически приятным и соответствовать общей стилистике платформы Telegram;
- приложение должно быть полностью локализовано на русский язык;

## 2. ПРОЕКТИРОВАНИЕ МИНИ-ПРИЛОЖЕНИЯ

На основе требований, сформулированных в предыдущей главе, в рамках данной главы осуществляется детальное проектирование разрабатываемого мини-приложения. Процесс проектирования охватывает ключевые аспекты создания программного продукта: определение общей архитектуры системы и выбор соответствующего технологического стека, формализацию функциональных возможностей и сценариев взаимодействия пользователя с приложением, а также подробное проектирование структуры базы данных. Целью данного этапа является создание всесторонней технической основы, необходимой для последующей реализации и тестирования мини-приложения. В последующих разделах будут последовательно рассмотрены каждый из этих этапов проектирования.

### **2.1 Функциональные возможности и сценарии использования**

На основе анализа предметной области, потребностей целевой аудитории, были определены ключевые функциональные возможности разрабатываемого мини-приложения. Данные возможности направлены на предоставление пользователю удобного инструмента для учета индивидуальных спортивных тренировок, анализа прогресса и доступа к релевантной информации.

Разрабатываемое мини-приложение должно предоставлять Пользователю следующие возможности:

– управлять своими тренировками. Создавать новые записи о проведенных тренировках (которые могут быть сформированы вручную или автоматически на основе персонализированных тренировочных схем), указывая дату, тип, продолжительность и заметки. Детализировать каждую тренировку добавлением выполненных упражнений (название, количество подходов, повторений, время отдыха, рабочий вес, дистанцию и продолжительность). Просматривать историю сохраненных тренировок, а также редактировать или удалять их;

– планировать тренировки. Создавать персонализированные тренировочные схемы, состоящие из последовательности тренировочных дней с набором упражнений и их целевыми параметрами (подходы, повторения, рабочий вес, дистанция, продолжительность и т.п.). Активировать выбранную схему в качестве текущего тренировочного плана. Автоматически формировать записи о будущих тренировках на основе активной схемы;

– отслеживать прогресс по упражнениям. Фиксировать и просматривать историю изменения ключевых показателей по выбранным упражнениям, а также визуализировать эту динамику в виде графиков;

– просматривать информационные статьи. Получать доступ к библиотеке материалов по темам фитнеса, методик тренировок, питания и здорового образа жизни, с возможностью навигации и чтения;

– просматривать свой профиль. Отображать основную информацию, автоматически полученную из его учетной записи Telegram (имя, аватар), а также просматривать календарь тренировок (с отметками проведенных и запланированных занятий).

На рисунке 11 представлена диаграмма вариантов использования мини-приложения, для отображения сценариев взаимодействия пользователя и мини-приложения.



Рисунок 11 – Диаграмма вариантов использования мини-приложения

Данная диаграмма иллюстрирует, как пользователь взаимодействует с ключевыми функциями мини-приложения, описанными выше. Определенные таким образом функциональные возможности и сценарии использования мини-приложения.

## **2.2 Определение архитектуры и выбор инструментов разработки**

Выбор архитектуры и технологического стека является фундаментальным этапом проектирования, определяющим масштабируемость, производительность, надежность и удобство дальнейшей разработки и поддержки программного продукта. Для разрабатываемого мини-приложения была выбрана клиент-серверная архитектура, которая является стандартным и наиболее эффективным решением для приложений, требующих хранения и обработки пользовательских данных, а также предоставления интерактивного пользовательского интерфейса.

Данная архитектура предполагает разделение системы на две основные взаимодействующие части:

- клиентская часть (Frontend). Реализуется в виде Telegram Mini App. Отвечает за отображение пользовательского интерфейса, взаимодействие с пользователем, сбор вводимых данных и отправку запросов на серверную часть;

- серверная часть (Backend). Отвечает за реализацию бизнес-логики приложения, обработку запросов от клиента, взаимодействие с базой данных (хранение, извлечение и модификация данных), а также за аутентификацию и авторизацию пользователей.

Такое разделение обеспечивает ряд преимуществ, включая независимость разработки и обновления клиента и сервера, возможность масштабирования серверной части при росте нагрузки, а также более четкое разделение ответственности между компонентами системы. Общая схема взаимодействия компонентов представлена на рисунке 12.

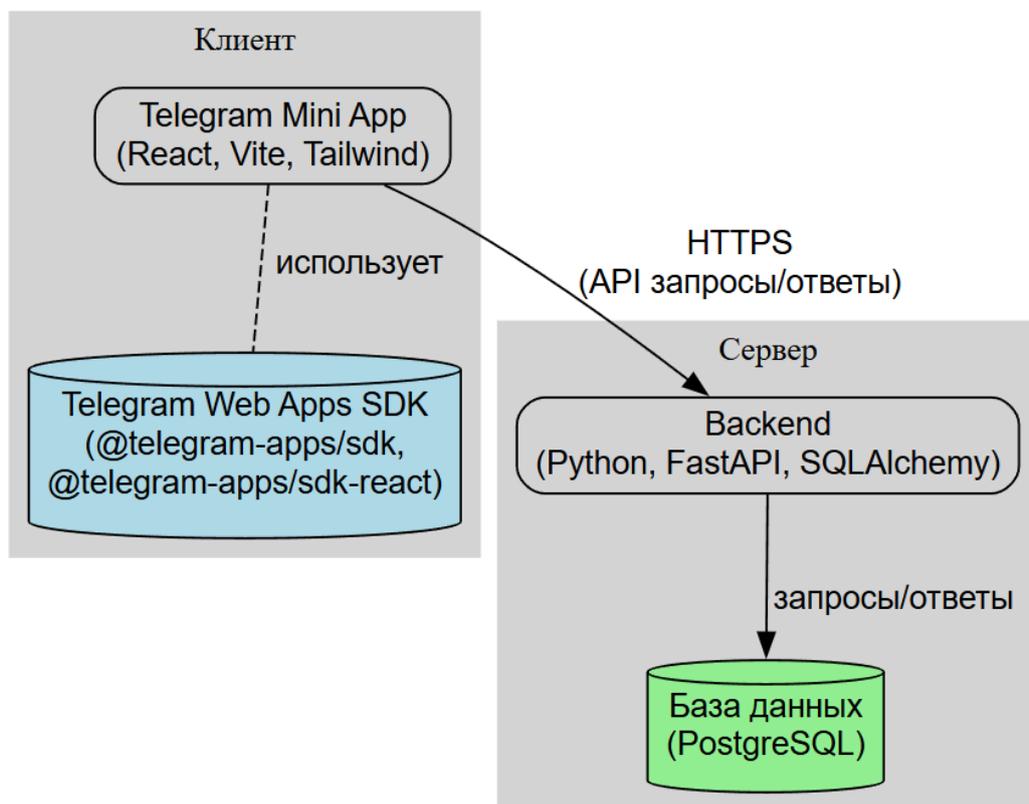


Рисунок 12 – Общая архитектура мини-приложения

Для реализации каждой из частей архитектуры, а также для обеспечения их взаимодействия и хранения данных, необходимо подобрать подходящие инструменты и технологии.

Инструменты и технологии для Frontend (клиентской части):

- React. Выбор данной JavaScript-библиотеки обусловлен ее популярностью, компонентным подходом к построению интерфейсов и эффективной работой с виртуальным DOM, что способствует созданию отзывчивых пользовательских интерфейсов;

- Vite. В качестве инструмента сборки и сервера разработки выбран Vite благодаря его высокой скорости холодной загрузки и горячей замены модулей, что ускоряет процесс разработки;

- React Router DOM. Для организации навигации и маршрутизации в одностраничном приложении используется данная библиотека, позволяющая декларативно определять маршруты и управлять переходами между разделами приложения;

– Tailwind CSS. Для стилизации интерфейса выбран утилитарный CSS-фреймворк, позволяющий быстро создавать кастомные дизайны с использованием низкоуровневых утилитарных классов;

– Telegram Web Apps SDK (Software Development Kit). Официальный набор средств разработки, предоставляемый Telegram для создания Mini Apps. Он необходим для взаимодействия клиентской части приложения с нативным окружением мессенджера Telegram, таким как получение данных пользователя, управление элементами интерфейса платформы и другие специфические функции.

Инструменты и технологии для Backend (серверной части):

– Python. В качестве основного языка программирования для серверной части выбран Python благодаря его простоте, читаемости и большому количеству доступных библиотек и фреймворков;

– PostgreSQL. В качестве реляционной СУБД для серверной части выбрана PostgreSQL, объектно-реляционная система управления базами данных с открытым исходным кодом, известная своей надежностью, расширяемостью и поддержкой сложных типов данных (включая JSON);

– FastAPI. Современный, высокопроизводительный веб-фреймворк для создания API на Python, обеспечивающий автоматическую валидацию данных (с Pydantic), генерацию документации API и поддержку асинхронного программирования;

– SQLAlchemy. SQL-инструментарий и Object-Relational Mapper (ORM) для Python, используемый для определения моделей данных и взаимодействия с реляционными базами данных;

– Pydantic. Библиотека для валидации данных и управления настройками, широко используемая FastAPI для определения схем запросов и ответов API;

– Uvicorn. ASGI (Asynchronous Server Gateway Interface) сервер, используемый для запуска FastAPI-приложений, способный обрабатывать большое количество асинхронных запросов.

Другие инструменты разработки и вспомогательные технологии:

– Git. Распределенная система контроля версий, используемая для управления исходным кодом проекта и совместной работы;

– Alembic. Инструмент для миграций схемы базы данных, интегрированный с SQLAlchemy, позволяющий управлять изменениями в структуре БД по мере развития приложения.

Выбранный технологический стек обеспечивает гибкость, производительность и удобство разработки, необходимые для создания современного и функционального мини-приложения, отвечающего поставленным требованиям.

### **2.3 Проектирование базы данных**

Проектирование базы данных представляет собой один из наиболее ответственных и фундаментальных этапов в жизненном цикле разработки любой информационной системы, и разрабатываемое мини-приложение не является исключением. От того, насколько корректно, продуманно и эффективно будет спроектирована структура хранения данных, напрямую зависят критически важные характеристики будущего приложения. К таким характеристикам относятся его общая производительность, особенно при выполнении операций чтения и записи данных, способность обеспечивать целостность и непротиворечивость хранимой информации, а также возможности для дальнейшего масштабирования системы при увеличении объема данных или расширении функциональности. Ошибки, допущенные на этапе проектирования базы данных, могут привести к значительным трудностям на последующих стадиях разработки, тестирования и эксплуатации, а их исправление зачастую требует существенных временных и ресурсных затрат.

#### **2.3.1 Инфологическое проектирование**

Проектирование базы данных представляет собой один из наиболее ответственных и фундаментальных этапов в жизненном цикле разработки любой информационной системы, и разрабатываемое мини-приложение не

является исключением. От того, насколько корректно, продуманно и эффективно будет спроектирована структура хранения данных, напрямую зависят критически важные характеристики будущего приложения. К таким характеристикам относятся его общая производительность, особенно при выполнении операций чтения и записи данных, способность обеспечивать целостность и непротиворечивость хранимой информации, а также возможности для дальнейшего масштабирования системы при увеличении объема данных или расширении функциональности. Ошибки, допущенные на этапе проектирования базы данных, могут привести к значительным трудностям на последующих стадиях разработки, тестирования и эксплуатации, а их исправление зачастую требует существенных временных и ресурсных затрат.

Таблица 2 – Основные сущности

Название сущности	Описание сущности
Пользователь	Содержит информацию о зарегистрированных пользователях приложения, идентифицируемых по их Telegram ID.
Тренировка	Содержит информацию о конкретной тренировочной сессии пользователя, включая дату, тип, длительность и набор выполненных упражнений.
Статья	Содержит информационные материалы по тематике фитнеса и тренировок.
Вид_Упражнения	Содержит справочную информацию о различных видах упражнений (например, "Жим штанги лежа", "Беговая дорожка"), их описания, категории и задействованные группы мышц.
Тренировочная_Схема	Содержит описание персонализированных тренировочных планов, созданных пользователями, определяющих последовательность дней и упражнений.
День_Схемы	Представляет собой шаблон конкретного дня в составе тренировочной схемы, описывающий набор упражнений и их целевые параметры, запланированные для выполнения в этот день.
Упражнение_Схемы	Представляет собой шаблон конкретного упражнения с его целевыми параметрами (подходы, повторения, вес, дистанция, длительность), включенный в состав определенного Дня_Схемы.
Активная_Схема	Служит для фиксации, какая из Тренировочных_Схем в данный момент является активной для конкретного Пользователя, служащей основой для автоматического формирования будущих тренировок.

Далее приведена спецификация атрибутов для каждой из выделенных сущностей в таблицах 3-10.

Таблица 3 – Спецификация атрибутов сущности «Пользователь»

Название атрибута	Описание атрибута	Пример
Код_Пользователя	Уникальный идентификатор пользователя (Telegram ID)	5084376925
Имя	Имя пользователя из Telegram	Иван
Фамилия	Фамилия пользователя из Telegram	Петров
Логин_Telegram	Логин пользователя в Telegram (username)	@ivan_petrov
URL_Фото	URL-адрес фотографии профиля пользователя в Telegram	https://t.me/photo...
Код_Языка	Код предпочитаемого языка пользователя в Telegram (например, ru)	ru
Дата_Регистрации	Дата и время создания записи о пользователе	2025-05-16 10:00:00

Таблица 4 – Спецификация атрибутов сущности «Тренировка»

Название атрибута	Описание атрибута	Пример
Код_Тренировки	Уникальный идентификатор тренировки.	1
Дата_Проведения	Дата и время проведения тренировки.	2025-05-17 18:30:00
Название_Тренировки	Тип или название тренировки (например, «День ног»).	Силовая на грудь
Длительность_мин	Длительность тренировки в минутах.	60
Заметки	Общие заметки или комментарии к тренировке.	Тяжелая тренировка, хорошее самочувствие
Список_Упражнений	Структурированный список упражнений, выполненных в тренировке, с их фактическими параметрами.	[{"name": "Жим", "sets": 3, ...}]
Код_Схемы_Источника	Идентификатор Тренировочной_Схемы, из которой была сгенерирована данная тренировка (если применимо).	42 (nullable)
Код_Дня_Схемы_Источника	Идентификатор Дня_Схемы, из которого был сгенерирован данный тренировочный день (если применимо).	5 (nullable)

Таблица 5 – Спецификация атрибутов сущности «Статья»

Название атрибута	Описание атрибута	Пример
1	2	3
Код_Статьи	Уникальный идентификатор статьи.	12
Заголовок	Заголовок статьи.	Основы правильного питания для набора массы

1	2	3
Содержание	Полное содержание статьи.	Для набора мышечной массы необходимо...
Дата_Публикации	Дата и время создания (публикации) статьи.	2025-05-10 12:00:00

Таблица 6 – Спецификация атрибутов сущности «Вид\_Упражнения»

Название атрибута	Описание атрибута	Пример
Код_Вида_Упражнения	Уникальный идентификатор вида упражнения.	1
Название	Название упражнения (например, «Жим штанги лежа», «Бег»).	Жим штанги лежа
Описание	Подробное описание упражнения и техники выполнения.	Базовое упражнение для развития мышц груди...
Группа_Мышц	Основная группа мышц, на которую направлено упражнение.	Грудь, Спина, Ноги, Кор
Необходимый_Инвентарь	Оборудование, необходимое для выполнения упражнения.	Штанга, гантели, турник, собственный вес
Категория	Категория упражнения (например, strength, cardio, duration_based, bodyweight_reps).	strength

Таблица 7 – Спецификация атрибутов сущности «Тренировочная\_Схема»

Название атрибута	Описание атрибута	Пример
1	2	3
Код_Схемы	Уникальный идентификатор тренировочной схемы.	1
Название_Схемы	Название тренировочной схемы (например, «Фулбоди для новичков»).	Фулбоди для новичков
Описание	Описание или цель схемы.	Программа для роста силы и массы
Длительность_(недель)	Предполагаемая длительность схемы в неделях.	4 (nullable)
Дата_Создания	Дата и время создания схемы.	2025-06-01 09:00:00
Дата_Обновления	Дата и время последнего обновления схемы.	25-06-05 14:00:00

1	2	3
Код_Создателя	Идентификатор Пользователя, создавшего данную схему.	5084376925

Таблица 8 – Спецификация атрибутов сущности «День\_Схемы»

Название атрибута	Описание атрибута	Пример
Код_Дня_Схемы	Уникальный идентификатор шаблона дня в схеме.	10
Порядок_Дня	Порядковый номер дня в рамках тренировочной схемы (1, 2, 3...).	1
Название_Дня	Название дня в схеме (например, «День 1: Ноги», «Верх тела»).	День 1: Ноги
Заметки	Заметки или комментарии к данному дню схемы.	Фокус на квадрицепсы
Код_Схемы	Идентификатор Тренировочной_Схемы, к которой относится данный день.	1

Таблица 9 – Спецификация атрибутов сущности «Упражнение\_Схемы»

Название атрибута	Описание атрибута	Пример
1	2	3
Код_Упражнения_Схемы	Уникальный идентификатор шаблона упражнения в дне схемы.	100
Порядок_Упражнения	Порядковый номер упражнения в рамках данного дня схемы.	1
Подходы	Целевое количество подходов.	3
Повторения	Целевое количество повторений или диапазон.	8-12
Отдых_(сек)	Целевое время отдыха между подходами в секундах.	60
Целевой_Вес	Целевой вес отягощения для упражнения (если применимо).	75.5 (nullable)
Ед_Веса	Единица измерения целевого веса.	кг
Целевая_Дистанция	Целевая дистанция для упражнения (если применимо).	5.0 (nullable)
Ед_Дистанции	Единица измерения целевой дистанции.	км
Целевая_Длительность_(сек)	Целевая продолжительность выполнения упражнения в секундах (если применимо).	1800 (nullable)
Заметки	Заметки или особенности выполнения для данного упражнения в схеме.	Медленно опускать
Код_Дня_Схемы	Идентификатор Дня_Схемы, к которому относится этот шаблон упражнения.	10

1	2	3
Код_Вида_Упражнения	Идентификатор Вида_Упражнения, к которому относится этот шаблон.	1

Таблица 10 – Спецификация атрибутов сущности «Активная\_Схема»

Название атрибута	Описание атрибута	Пример
Код_Активной_Схемы	Уникальный идентификатор записи об активной схеме пользователя.	1
Дата_Активации	Дата и время активации данной схемы пользователем.	2025-06-10 08:00:00
Код_Пользователя	Идентификатор Пользователя, для которого активна данная схема.	5084376925
Код_Схемы	Идентификатор Тренировочной_Схемы, которая активна для пользователя.	1

После определения сущностей и их атрибутов, необходимо спроектировать связи между ними для обеспечения целостности данных и их корректного функционирования в системе.

В результате анализа были спроектированы следующие основные связи:

– связь «Пользователь – Тренировка», представлена на рисунке 13.



Рисунок 13 – Связь «Пользователь – Тренировка»

Каждый пользователь может создать множество тренировок, но каждая тренировка создается ровно одним пользователем.

– связь «Пользователь – Тренировочная\_Схема», представлена на рисунке 14.

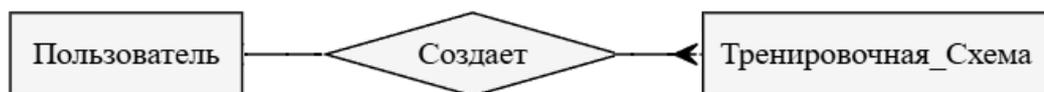


Рисунок 14 – Связь «Пользователь – Тренировочная\_Схема»

Каждый пользователь создает множество схем, но каждая схема создается ровно одним пользователем.

– связь «Тренировочная\_Схема – День\_Схемы», представлена на рисунке 15.



Рисунок 15 – Связь «Тренировочная\_Схема – День\_Схемы»

Одна схема содержит множество дней, но день принадлежит одной схеме.

– связь «День\_Схемы – Упражнение\_Схемы», представлена на рисунке 16.



Рисунок 16 – Связь «День\_Схемы – Упражнение\_Схемы»

Каждый день может включать множество упражнений из схемы, но каждое упражнение из схемы принадлежит ровно одному дню этой схемы.

– связь «Вид\_Упражнения – Упражнение\_Схемы», представлена на рисунке 17.



Рисунок 17 – Связь «Вид\_Упражнения – Упражнение\_Схемы»

Каждый вид упражнения может определять множество упражнений в схеме, но каждое упражнение в схеме основано ровно на одном виде упражнения.

– связь «Пользователь – Активная\_Схема», представлена на рисунке 18.



Рисунок 18 – Связь «Пользователь – Активная\_Схема»

Каждый пользователь может иметь одну активную в данный момент схему, и каждая активная схема принадлежит ровно одному пользователю.

– связь «Активная\_Схема – Тренировочная\_Схема», представлена на рисунке 19.



Рисунок 19 – Связь «Активная\_Схема – Тренировочная\_Схема»

Каждая активная схема соответствует ровно одной тренировочной схеме, и каждая тренировочная схема может быть связана одной активной схемой.

– связь «Тренировка – Тренировочная\_Схема», представлена на рисунке 20.

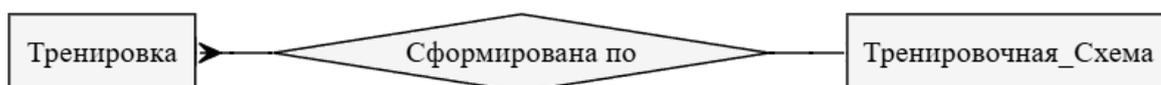


Рисунок 20 – Связь «Тренировка – Тренировочная\_Схема»

Каждая тренировка может быть сформирована одной тренировочной схеме. Одна тренировочная схема может быть источником для множества тренировок.

– связь «Тренировка – День\_Схемы», представлена на рисунке 21.



Рисунок 21 – Связь «Тренировка – День\_Схемы»

Каждая тренировка может быть сформирована по одному дню схемы. Один день схемы может быть источником для множества тренировок.

Результатом этапа инфологического проектирования является концептуальная модель данных, которая наглядно отражает выявленные сущности, их атрибуты и установленные между ними связи. Комплексная инфологическая

модель базы данных для разрабатываемого мини-приложения, выполненная в нотации Чена, представлена в приложении Б.

### 2.3.2 Логическое проектирование

На основе концептуальной модели, разработанной на предыдущем этапе, были определены таблицы, которые составят основу реляционной базы данных. Для каждой сущности, выявленной на инфологическом этапе, была создана соответствующая таблица. Детальная спецификация каждой таблицы, включая перечень полей, их логические типы данных, а также обозначение первичных (PK) и внешних (FK) ключей, представлена в таблицах 11-18.

Таблица 11 – Спецификация таблицы «Пользователь»

Имя поля	Тип данных (логический)	PK	FK	Описание
1	2	3	4	5
Код_Пользователя	BIGINT	Да	–	Уникальный идентификатор Пользователя (Telegram ID)
Имя	VARCHAR(255)	–	–	Имя пользователя (получено из Telegram)
Фамилия	VARCHAR(255)	–	–	Фамилия пользователя (получено из Telegram)
Логин_Telegram	VARCHAR(255)	–	–	Логин пользователя в Telegram
URL_Фото	VARCHAR(1024)	–	–	URL фото профиля пользователя Telegram
Код_Языка	VARCHAR(10)	–	–	Код языка пользователя (получено из Telegram)
Дата_Регистрации	DATETIME	–	–	Дата и время создания записи о Пользователе в системе

Таблица 12 – Спецификация таблицы «Тренировка»

Имя поля	Тип данных (логический)	PK	FK	Описание
1	2	3	4	5
Код_Тренировки	INTEGER	Да	–	Уникальный идентификатор Тренировки
Код_Пользователя	BIGINT	–	«Пользователь» (id)	Идентификатор Пользователя, которому принадлежит Тренировка

1	2	3	4	5
Дата_Проведения	DATETIME	–	–	Дата и время проведения Тренировки
Название_Тренировки	VARCHAR(255)	–	–	Тип или название Тренировки
Длительность_мин	INTEGER	–	–	Длительность Тренировки в минутах
Заметки	TEXT	–	–	Текстовые заметки к Тренировке (опционально)
Список_Упражнений	JSON	–	–	Структурированные данные о выполненных упражнениях (список объектов)
Код_Схемы_Источника	INTEGER	–	«Тренировочная_Схема» (id)	Идентификатор тренировочной схемы, по которой была сгенерирована тренировка (опционально)
Код_Дня_Схемы_Источника	INTEGER	–	«День_Схемы» (id)	Идентификатор дня схемы, по которому была сгенерирована тренировка (опционально)

Таблица 13 – Спецификация таблицы «Статья»

Имя поля	Тип данных (логический)	PK	FK	Описание
Код_Статьи	INTEGER	Да	–	Уникальный идентификатор Статьи
Заголовок	TEXT	–	–	Заголовок Статьи
Содержание	TEXT	–	–	Содержание Статьи
Дата_Публикации	DATETIME	–	–	Дата и время создания Статьи

Таблица 14 – Спецификация таблицы «Вид\_Упражнения»

Имя поля	Тип данных (логический)	PK	FK	Описание
1	2	3	4	5
Код_Вида_Упражнения	INTEGER	Да	–	Уникальный идентификатор вида упражнения

1	2	3	4	5
Название	VARCHAR(255)	–	–	Название упражнения
Описание	TEXT	–	–	Описание упражнения
Группа_Мышц	VARCHAR(100)	–	–	Основная группа мышц, на которую направлено упражнение
Необходимый_инвентарь	VARCHAR(255)	–	–	Инвентарь, необходимый для выполнения упражнения
Категория	VARCHAR(50)	–	–	Категория упражнения (например, strength, cardio, bodyweight)

Таблица 15 – Спецификация таблицы «Тренировочная\_Схема»

Имя поля	Тип данных (логический)	PK	FK	Описание
Код_Схемы	INTEGER	Да	–	Уникальный идентификатор тренировочной схемы
Название_Схемы	VARCHAR(255)	–	–	Название тренировочной схемы
Описание_Схемы	TEXT	–	–	Подробное описание схемы
Длительность	INTEGER	–	–	Примерная длительность схемы в неделях (опционально)
Код_Создателя	BIGINT	–	«Пользователь» (id)	Идентификатор пользователя, создавшего схему
Дата_Создания	DATETIME	–	–	Дата и время создания схемы
Дата_Обновления	DATETIME	–	–	Дата и время последнего обновления схемы (автоматически обновляется)

Таблица 16 – Спецификация таблицы «День\_Схемы»

Имя поля	Тип данных (логический)	PK	FK	Описание
1	2	3	4	5
Код_Дня_Схемы	INTEGER	Да	–	Уникальный идентификатор шаблона дня схемы
Код_Схемы	INTEGER	–	«Тренировочная_Схема» (id)	Идентификатор тренировочной схемы, к которой относится этот день
Порядок_Дня	INTEGER	–	–	Порядковый номер дня в цикле схемы (1, 2, 3...)

1	2	3	4	5
Название_Дня	VARCHAR(100)	–	–	Название дня тренировки (например, "День 1: Ноги")
Заметки	TEXT	–	–	Дополнительные заметки к дню схемы

Таблица 17 – Спецификация таблицы «Упражнение\_Схемы»

Имя поля	Тип данных (логический)	PK	FK	Описание
1	2	3	4	5
Код_Упражнения_Схемы	INTEGER	Да	–	Уникальный идентификатор шаблона упражнения в дне схемы
Код_Дня_Схемы	INTEGER	–	«День_Схемы» (id)	Идентификатор дня схемы, к которому относится это упражнение
Код_Вида_Упражнения	INTEGER	–	«Вид_Упражнения» (id)	Идентификатор определения вида упражнения
Порядок_Упражнения	INTEGER	–	–	Порядок упражнения в дне схемы
Подходы	INTEGER	–	–	Рекомендуемое количество подходов
Повторения	VARCHAR(50)	–	–	Рекомендуемое количество повторений (например, "8-12")
Отдых	INTEGER	–	–	Рекомендуемое время отдыха между подходами в секундах
Целевой_Вес	FLOAT	–	–	Целевой вес для выполнения упражнения
Ед_Веса	VARCHAR(10)	–	–	Единица измерения целевого веса (например, 'кг')
Целевая_Дистанция	FLOAT	–	–	Целевая дистанция для выполнения упражнения (например, 'км')
Ед_Дистанции	VARCHAR(10)	–	–	Единица измерения целевой дистанции

1	2	3	4	5
Целевая_Длительность	INTEGER	–	–	Целевая длительность выполнения упражнения в секундах
Заметки	TEXT	–	–	Заметки к конкретному упражнению в схеме

Таблица 18 – Спецификация таблицы «Активная\_Схема»

Имя поля	Тип данных (логический)	PK	FK	Описание
Код_Активной_Схемы	INTEGER	Да	–	Уникальный идентификатор активной схемы пользователя
Код_Пользователя	BIGINT	–	«Пользователь» (id)	Идентификатор пользователя, у которого активна схема
Код_Схемы	INTEGER	–	«Тренировочная_Схема» (id)	Идентификатор тренировочной схемы, которая активирована
Дата_Активации	DATETIME	–	–	Дата и время активации схемы

Следующим важным шагом логического проектирования является анализ полученной структуры таблиц на соответствие нормальным формам. Нормализация направлена на устранение избыточности данных и предотвращение аномалий при их модификации (вставке, обновлении, удалении). Основой для процесса нормализации служит выявление и анализ функциональных зависимостей между атрибутами в каждой таблице.



Рисунок 22 – Функциональные зависимости для таблицы «Пользователь»



Рисунок 23 – Функциональные зависимости для таблицы «Тренировка»



Рисунок 24 – Функциональные зависимости для таблицы «Статья»



Рисунок 25 – Функциональные зависимости для таблицы «Вид\_Упражнения»



Рисунок 26 – Функциональные зависимости для таблицы «Тренировочная\_Схема»

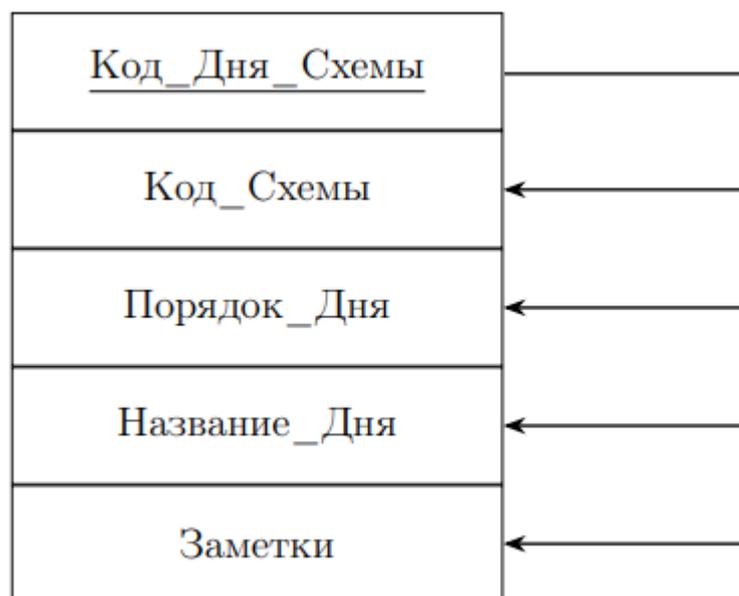


Рисунок 27 – Функциональные зависимости для таблицы «День\_Схемы»

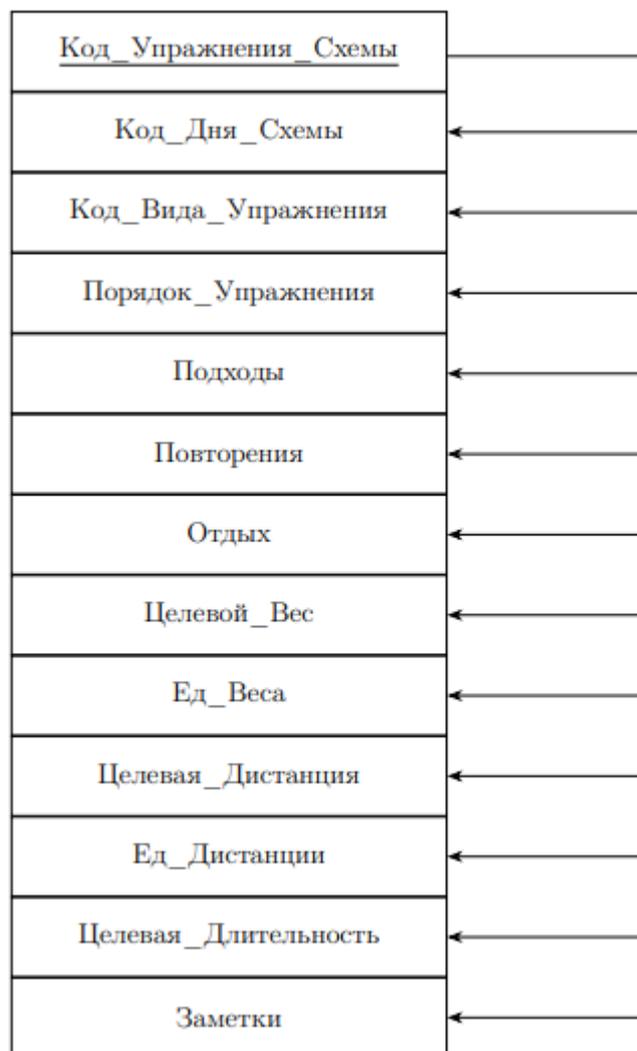


Рисунок 28 – Функциональные зависимости для таблицы «Упражнение\_Схемы»



Рисунок 29 – Функциональные зависимости для таблицы «Активная\_Схема»

Представленные на рисунках 22-29 диаграммы показывают, что во всех таблицах первичный ключ выступает детерминантом для всех остальных неключевых атрибутов данной таблицы. Отсутствие частичных или транзитивных зависимостей неключевых атрибутов от первичного ключа будет рассмотрено далее при анализе нормальных форм.

На основе структуры таблиц и выявленных функциональных зависимостей проведен анализ на соответствие нормальным формам:

Все таблицы находятся в первой нормальной форме (1НФ), так как все их атрибуты являются атомарными. Атрибут Список\_Упражнений в таблице «Тренировка», реализуемый как JSON, на уровне СУБД рассматривается как единое атомарное значение, что соответствует требованиям 1НФ.

Таблица находится во второй нормальной форме (2НФ), если она находится в первой нормальной форме и каждый неключевой атрибут полностью функционально зависим от всего первичного ключа. Во всех спроектированных таблицах первичный ключ является простым (состоит из одного атрибута), что гарантирует полную функциональную зависимость всех неключевых атрибутов от него. Следовательно, все таблицы находятся во второй нормальной форме.

Таблица находится в третьей нормальной форме (3НФ), если она находится во второй нормальной форме и не содержит транзитивных зависимостей неключевых атрибутов от первичного ключа. Анализ семантики атрибутов и функциональных зависимостей в каждой из таблиц («Пользователь», «Тренировка», «Статья», «Вид\_Упражнения», «Тренировочная\_Схема», «День\_Схемы», «Упражнение\_Схемы», «Активная\_Схема») показывает, что неключевые атрибуты зависят только от первичного ключа (например, Код\_Пользователя, Код\_Тренировки, Код\_Статьи и т.д. в зависимости от таблицы) и не зависят от других неключевых атрибутов. Таким образом, транзитивные зависимости отсутствуют, и все спроектированные таблицы удовлетворяют требованиям третьей нормальной формы.

Итоговая логическая модель базы данных, представленная в приложении В детально отражающая структуру всех спроектированных таблиц, их атрибуты, определенные первичные и внешние ключи, а также установленные реляционные связи между таблицами. Разработанная в результате логического проектирования и нормализации, данная модель формирует прочную основу для последующего физического проектирования и обеспечивает эффективное, структурированное хранение и целостную обработку всего объема данных, необходимого для корректного и полнофункционального функционирования мини-приложения.

### 2.3.3 Физическое проектирование

Завершающим и критически важным этапом в комплексном процессе проектирования базы данных для разрабатываемого мини-приложения является физическое проектирование. Именно на данной стадии абстрактная логическая модель данных, созданная ранее, трансформируется в конкретную, готовую к реализации схему, полностью адаптированную под специфику и возможности выбранной системы управления базами данных – в данном случае, PostgreSQL. Этот переход требует тщательного рассмотрения множества технических деталей.

Основные задачи, решаемые в рамках физического проектирования, включают не только окончательный выбор наиболее подходящих и эффективных типов данных PostgreSQL для каждого атрибута каждой таблицы, но и точное определение всех первичных и внешних ключей с настройкой соответствующих ограничений целостности. Кроме того, на этом этапе детально анализируется предполагаемая нагрузка на базу данных, характер наиболее частых запросов, и на основе этого анализа принимается решение о необходимости и стратегии создания дополнительных индексов для ключевых полей, что напрямую влияет на оптимизацию выполнения запросов и, как следствие, на общую производительность и отзывчивость всей системы. Правильное физическое проектирование также учитывает особенности хранения данных в СУБД, потенциальные узкие места и возможности для их предотвращения.

Физическое определение атрибутов (полей) для каждой таблицы, их типы данных в PostgreSQL, а также условия и индексация представлены в таблицах 19-26.

Таблица 19 – Физическая структура таблицы «Пользователь»

Название поля	Тип данных (PostgreSQL)	Условия	Формат данных	Индексация
Код_Пользователя	BIGINT	PRIMARY KEY	Целое число (64-бит)	PRIMARY KEY
Имя	VARCHAR(255)	NULL	Текст (до 255 символов)	–
Фамилия	VARCHAR(255)	NULL	Текст (до 255 символов)	–
Логин_Telegram	VARCHAR(255)	NULL	Текст (до 255 символов)	–
URL_Фото	VARCHAR(1024)	NULL	Текст (до 1024 символов)	–
Код_Языка	VARCHAR(10)	NULL	Текст (до 10 символов)	–
Дата_Регистрации	TIMESTAMPTZ	NOT NULL, DEFAULT NOW()	Дата и время с часовым поясом	–

Таблица 20 – Физическая структура таблицы «Тренировка»

Название поля	Тип данных (PostgreSQL)	Условия	Формат данных	Индексация
1	2	3	4	5
Код_Тренировки	SERIAL	PRIMARY KEY	Целое число (автоинкремент)	PRIMARY KEY
Список_Упражнений	JSONB	NULL	JSON (бинарный формат)	–
Код_Схемы_Источника	INTEGER	NULL, FOREIGN KEY REFERENCES «Тренировочная_Схема»(Код_Схемы)	Целое число	FOREIGN KEY
Код_Дня_Схемы_Источника	INTEGER	NULL, FOREIGN KEY REFERENCES «День_Схемы»(Код_Дня_Схемы)	Целое число	FOREIGN KEY

1	2	3	4	5
Код_Пользователя	BIGINT	NOT NULL, FOREIGN KEY REFERENCES «Пользователь»(Код_Пользователя)	Целое число (64-бит)	FOREIGN KEY
Дата_Проведения	TIMESTAMPTZ	NOT NULL	Дата и время с часовым поясом	–
Название_Тренировки	VARCHAR(255)	NOT NULL	Текст (до 255 символов)	–
Длительность_мин	INTEGER	NOT NULL	Целое число	–

Таблица 21 – Физическая структура таблицы «Статья»

Название поля	Тип данных (PostgreSQL)	Условия	Формат данных	Индексация
Код_Статьи	SERIAL	PRIMARY KEY	Целое число (автоинкремент)	PRIMARY KEY
Заголовок	TEXT	NOT NULL	Текст произвольной длины	–
Содержание	TEXT	NOT NULL	Текст произвольной длины	–
Дата_Публикации	TIMESTAMPTZ	NOT NULL, DEFAULT NOW()	Дата и время с часовым поясом	–

Таблица 22 – Физическая структура таблицы «Вид\_Упражнения»

Название поля	Тип данных (PostgreSQL)	Условия	Формат данных	Индексация
Код_Вида_Упражнения	INTEGER	PRIMARY KEY	Целое число	PRIMARY KEY
Название	VARCHAR(255)	NOT NULL	Текст (до 255 символов)	–
Описание	TEXT	NULL	Текст произвольной длины	–
Группа_Мышц	VARCHAR(100)	NULL	Текст (до 100 символов)	–
Необходимый_инвентарь	VARCHAR(255)	NULL	Текст (до 255 символов)	–
Категория	VARCHAR(50)	NULL	Текст (до 50 символов)	–

Таблица 23 – Физическая структура таблицы «Тренировочная\_Схема»

Название поля	Тип данных (PostgreSQL)	Условия	Формат данных	Индексация
Код_Схемы	INTEGER	PRIMARY KEY	Целое число	PRIMARY KEY
Название_Схемы	VARCHAR(255)	NOT NULL	Текст (до 255 символов)	–
Описание_Схемы	TEXT	NULL	Текст произвольной длины	–
Длительность	INTEGER	NULL	Целое число	–
Код_Создателя	BIGINT	NOT NULL, FOREIGN KEY REFERENCES «Пользователь»(Код_Пользователя)	Целое число (64-бит)	FOREIGN KEY
Дата_Создания	TIMESTAMPTZ	NOT NULL, DEFAULT NOW()	Дата и время с часовым поясом	–
Дата_Обновления	TIMESTAMPTZ	NOT NULL, DEFAULT NOW()	Дата и время с часовым поясом	–

Таблица 24 – Физическая структура таблицы «День\_Схемы»

Название поля	Тип данных (PostgreSQL)	Условия	Формат данных	Индексация
1	2	3	4	5
Код_Дня_Схемы	INTEGER	PRIMARY KEY	Целое число	PRIMARY KEY
Код_Схемы	INTEGER	NOT NULL, FOREIGN KEY REFERENCES «Тренировочная_Схема»(Код_Схемы)	Целое число	FOREIGN KEY
Порядок_Дня	INTEGER	NOT NULL	Целое число	–
Название_Дня	VARCHAR(100)	NULL	Текст (до 100 символов)	–
Заметки	TEXT	NULL	Текст произвольной длины	–

Таблица 25 – Физическая структура таблицы «Упражнение\_Схемы»

Название поля	Тип данных (PostgreSQL)	Условия	Формат данных	Индексация
Код_Упражнения_Схемы	INTEGER	PRIMARY KEY	Целое число	PRIMARY KEY
Код_Дня_Схемы	INTEGER	NOT NULL, FOREIGN KEY REFERENCES «День_Схемы»( Код_Дня_Схемы)	Целое число	FOREIGN KEY
Код_Вида_Упражнения	INTEGER	NOT NULL, FOREIGN KEY REFERENCES «Вид_Упражнения»( Код_Вида_Упражнения)	Целое число	FOREIGN KEY
Порядок_Упражнения	INTEGER	NOT NULL	Целое число	–
Подходы	INTEGER	NULL	Целое число	–
Повторения	VARCHAR(50)	NULL	Текст (до 50 символов)	–
Отдых	INTEGER	NULL	Целое число	–
Целевой_Вес	REAL	NULL	Число с плавающей точкой	–
Ед_Веса	VARCHAR(10)	NULL	Текст (до 10 символов)	–
Целевая_Дистанция	REAL	NULL	Число с плавающей точкой	–
Ед_Дистанции	VARCHAR(10)	NULL	Текст (до 10 символов)	–
Целевая_Длительность	INTEGER	NULL	Целое число	–
Заметки	TEXT	NULL	Текст произвольной длины	–

Таблица 26 – Физическая структура таблицы «Активная\_Схема»

Название поля	Тип данных (PostgreSQL)	Условия	Формат данных	Индексация
1	2	3	4	5
Код_Активной_Схемы	INTEGER	PRIMARY KEY	Целое число	PRIMARY KEY

1	2	3	4	5
Код_Пользователя	BIGINT	NOT NULL, FOREIGN KEY REFERENCES «Пользователь»( Код_Пользова- теля)	Целое число (64- бит)	FOREIG N KEY
Код_Схемы	INTEGER	NOT NULL, FOREIGN KEY REFERENCES «Тренировочная _Схема»(Код_С хемы)	Целое число	FOREIG N KEY
Дата_Активации	TIMESTAMPTZ	NOT NULL, DEFAULT NOW()	Дата и время с ча- совым поясом	–

Физическое проектирование также предусматривает построение структуры физической модели данных, которая наглядно иллюстрирует реализованные таблицы, их поля с физическими типами данных и связи. Физическая модель базы данных представлена в приложении Г.

## 3 РАЗРАБОТКА МИНИ-ПРИЛОЖЕНИЯ

### 3.1 Реализация клиентской части

Разработка клиентской части мини-приложения осуществлялась как интерактивное веб-приложение, функционирующее в среде Telegram Mini Apps, с использованием JavaScript-библиотеки React. В основу реализации положен компонентный подход, обеспечивающий модульность, переиспользование элементов и удобство поддержки кода. Для ускорения процесса разработки и оптимизации сборки проекта использовался инструмент Vite. Управление всеми программными зависимостями проекта выполнялось посредством пакетного менеджера npm. Для обеспечения единообразной и адаптивной стилизации элементов пользовательского интерфейса применялся утилитарный CSS-фреймворк Tailwind CSS, что позволило создавать дизайн, органично вписывающийся в эстетику платформы Telegram, за счет использования низкоуровневых утилитарных классов и системных переменных темы мессенджера (например, `--tg-theme-bg-color`, `--tg-theme-text-color`).

В число ключевых технологий и библиотек, задействованных на клиентской стороне, вошли:

- `react-router-dom`. Данная библиотека использовалась для организации навигационной системы и маршрутизации в одностраничных приложениях, что позволяет динамически отображать различные страницы в зависимости от URL без полной перезагрузки;

- `Axios`. Применялся для выполнения HTTP-запросов к программному интерфейсу (API) серверной части. Для унификации всех сетевых взаимодействий и централизованного управления параметрами запросов был создан специализированный клиентский модуль `apiClient`, через который производятся все обращения к бэкенду;

- `@heroicons/react/24/solid`. Этот набор векторных иконок интегрирован в интерфейс для повышения его визуальной привлекательности и интуитивной понятности;

– @telegram-apps/sdk. Официальный комплект для разработки программ (SDK) Telegram, который является критически важным для обеспечения взаимодействия клиентской части с нативной средой мессенджера. Через данный SDK осуществляется получение данных пользователя (с помощью `WebApp.initDataUnsafe` или `WebApp.initData`), управление встроенными элементами интерфейса платформы Telegram (например, заголовком окна, стандартными кнопками `WebApp.MainButton` и `WebApp.BackButton`), а также адаптация цветовой схемы приложения под текущую тему пользователя (`WebApp.themeParams`, `WebApp.colorScheme`). Информирование о готовности приложения осуществляется посредством `WebApp.ready()`.

Центральным элементом клиентской архитектуры является корневой компонент `App`, выполняющий функции инициализации глобального состояния приложения. Его задачи охватывают управление процессом аутентификации пользователя, координацию первоначальной загрузки пользовательских данных, таких как список тренировок, и конфигурирование основной системы маршрутизации. Для обеспечения бесшовной авторизации и доступности информации об аутентифицированном пользователе в любой точке приложения используется компонент `AuthProvider`, построенный на базе `React Context API`. Этот подход позволяет эффективно распространять состояние аутентификации (включая данные пользователя, флаги процесса аутентификации и информацию об ошибках) по всему дереву компонентов без необходимости их явной передачи через пропсы. Механизм аутентификации предусматривает получение пользовательских данных из объекта `WebApp.initDataUnsafe` или `WebApp.initData` в реальной среде Telegram Mini App. Для обеспечения непрерывности разработки и отладки приложения вне мессенджера, в случае отсутствия нативных данных, предусмотрено использование моковых пользовательских данных.

Основные функциональные разделы мини-приложения реализованы в виде отдельных `React`-компонентов-страниц, каждый из которых предназначен для выполнения конкретных задач:

– список тренировок (Trainings.jsx). Данный компонент отображает хронологически упорядоченный перечень всех зарегистрированных тренировок пользователя, предоставляя интерфейс для их просмотра и базового управления;

– создание и редактирование тренировок (CreateTraining.jsx). Компонент предназначен для формирования записей о новых тренировках, позволяя пользователю детализировать каждое занятие путем добавления выполненных упражнений с их параметрами. Данный компонент также адаптирован для модификации уже существующих тренировок;

– детализация тренировки (TrainingDetail.jsx). При выборе конкретной тренировки компонент отображает исчерпывающую информацию о ней, включая подробный список выполненных упражнений и предоставленные пользователем заметки. Функциональность компонента включает возможности для редактирования отдельных параметров упражнений и удаления записи о тренировке;

– информационные статьи (Home.jsx). Предоставляет пользователям доступ к базе данных информационных статей, охватывающих широкий спектр тем, связанных с фитнесом и здоровым образом жизни, с возможностью их просмотра;

– библиотека упражнений (ExerciseLibrary.jsx). Реализует функциональность просмотра и поиска по справочной базе определений упражнений, предоставляя информацию об их описаниях, задействованных мышечных группах и необходимом оборудовании;

– управление тренировочными схемами (TrainingSchemesList.jsx, SchemeEditor.jsx). Данные компоненты позволяют пользователям создавать персонализированные тренировочные схемы, состоящие из последовательности дней и упражнений с целевыми параметрами. Реализована функциональность активации выбранной схемы, что делает её текущим тренировочным планом пользователя, а также деактивации и редактирования схем;

– профиль пользователя (Profile.jsx). На данной странице отображается основная информация о пользователе, полученная из его аккаунта Telegram (имя, фамилия, юзернейм, аватар), а также включает интерактивный календарь для визуализации дней с проведенными и запланированными тренировками;

– отслеживание прогресса (Progress.jsx, ProgressDetail.jsx). Компоненты данного раздела служат для отслеживания динамики результатов пользователя. Компонент Progress.jsx предоставляет интерфейс для выбора конкретного упражнения, а ProgressDetail.jsx затем визуализирует временные ряды ключевых показателей (например, рабочего веса, дистанции или времени) по выбранному упражнению в виде линейных графиков, используя библиотеку recharts, а также отображает историю результатов в табличном формате.

Управление локальным состоянием отдельных компонентов и их жизненным циклом реализовано с использованием встроенных хуков React (useState и useEffect). Передача данных между компонентами осуществляется преимущественно через механизм пропсов, что обеспечивает предсказуемый и однонаправленный поток информации по всему приложению.

### **3.2 Реализация серверной части**

Серверная часть приложения, обеспечивающая реализацию бизнес-логики, управление данными и предоставление программного интерфейса (API), разработана на языке программирования Python с использованием веб-фреймворка FastAPI версии 0.115.12. В качестве реляционной системы управления базами данных (СУБД) выбрана PostgreSQL, а для объектно-реляционного отображения (ORM) используется библиотека SQLAlchemy версии 2.0.40. Соединение с PostgreSQL обеспечивается адаптером psycopg2-binary версии 2.9.10.

Архитектура данных на сервере базируется на объектно-реляционных моделях, определенным в модуле моделей. Эти модели являются программным представлением сущностей и их взаимосвязей, разработанных на этапах

инфологического и логического проектирования. Каждая модель определяет поля таблицы, их типы данных, ограничения целостности (такие как первичные и внешние ключи, уникальные ограничения) и реляционные связи между сущностями, используя инструментарий ORM. Например, поле для хранения списка упражнений в сущности «Тренировка» реализовано как тип данных JSONB, что обеспечивает гибкое и структурированное хранение детальных параметров выполненных упражнений.

Для обеспечения строгой валидации входящих HTTP-запросов и стандартизации структуры исходящих ответов API применяются схемы Pydantic версии 2.11.3 (с использованием `pydantic_core` версии 2.33.1 и `typing_extensions` версии 4.13.2), определенные в модуле схем. Эти схемы гарантируют автоматическую проверку соответствия данных заданным типам и форматам, что существенно повышает надежность, безопасность и предсказуемость взаимодействия между клиентской и серверной частями приложения.

Реализация бизнес-логики и определение API-эндпоинтов осуществлены в специализированных модулях-роутерах FastAPI, логически сгруппированных по функциональному назначению. Каждый роутер содержит набор функций-обработчиков HTTP-запросов, реализующих операции создания, чтения, обновления и удаления (CRUD), а также более сложную бизнес-логику:

- управление пользователями. Данный роутер отвечает за процессы аутентификации и регистрации пользователей. Предусмотрено идиоматичное поведение: при первом обращении пользователя создается новая запись в базе данных с данными, полученными из Telegram; при последующих обращениях данные пользователя обновляются при необходимости;

- управление тренировками. Роутер предоставляет полный набор операций CRUD для тренировок, включая создание, получение списков, детализацию отдельных записей, а также их обновление и удаление;

– управление тренировочными схемами. Роутер реализует комплексное управление персонализированными тренировочными схемами, позволяя создавать новые схемы, добавлять шаблоны дней и упражнений в них, а также осуществлять просмотр, редактирование и удаление схем. Особое внимание уделено логике активации схем, при которой для пользователя устанавливается одна активная схема. Важной частью бизнес-логики является также генерация тренировок на неделю на основе выбранной активной схемы и механизм деактивации схемы, который приводит к удалению всех будущих тренировок, запланированных по данной схеме, для предотвращения несогласованности данных;

– управление справочниками. Роутер обеспечивает предоставление API для работы со справочником определений упражнений и информационных статей, позволяя клиенту получать их списки и содержание;

– отслеживание прогресса. Роутер реализует функциональность сбора и агрегации данных о прогрессе пользователя по выбранным упражнениям из исторического журнала тренировок. Формируются временные ряды данных, необходимые для построения графиков прогресса.

Взаимодействие с базой данных в функциях-обработчиках роутеров осуществляется через сессии ORM, которые внедряются в качестве зависимостей, что обеспечивает инкапсуляцию логики работы с базой данных, управление транзакциями и эффективное использование пула соединений.

Настройка окружения, инициализация веб-фреймворка и подключение всех определенных роутеров осуществляются в главном файле приложения. Здесь же конфигурируются механизмы межсайтовой безопасности (CORS) для обеспечения безопасного взаимодействия между клиентской и серверной частями, которые могут быть развернуты на различных доменах. Конфигурационные параметры приложения, такие как URL базы данных, управляются через модуль настроек, который загружает их из переменных окружения. Запуск серверной части приложения выполняется с помощью ASGI-сервера. При инициализации приложения также вызывается функция для первичного

заполнения базы данных справочной информацией, что обеспечивает наличие базового контента для демонстрации функциональности приложения.

### 3.3 Демонстрация работы мини-приложения

При первом запуске мини-приложения инициируется процесс автоматической аутентификации пользователя на основе его данных из Telegram. После успешной идентификации и верификации, пользователь получает доступ к основным разделам приложения через нижнюю навигационную панель, что обеспечивает бесшовный вход в систему.

На странице «Профиль» отображается основная информация о пользователе, автоматически полученная из его аккаунта Telegram. К этим данным относятся аватар пользователя и имя пользователя (Username), что персонализирует взаимодействие.

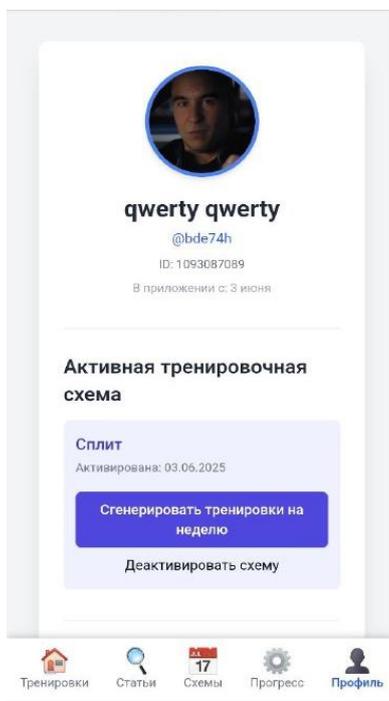


Рисунок 30 – Интерфейс страницы «Профиль пользователя»

Ниже расположен интерактивный календарь, предназначенный для визуализации тренировочной активности. На данном макете календаря за май 2025 года видно, как отмечены дни с проведенными тренировками, а также выведена информация о следующей запланированной тренировке. Этот

функционал позволяет пользователям наглядно отслеживать свою тренировочную активность.

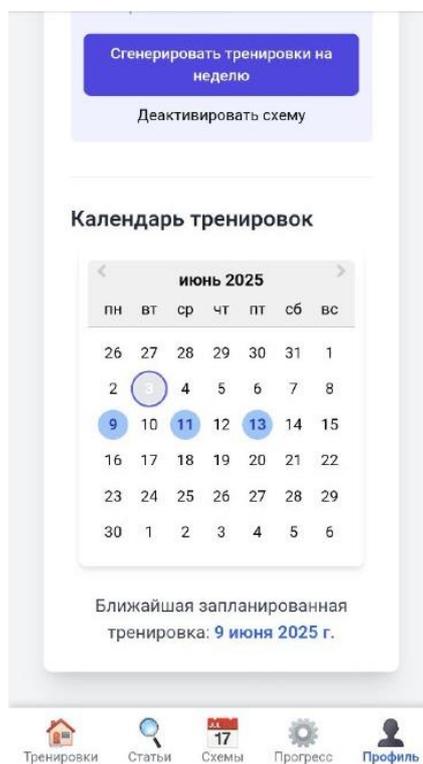


Рисунок 31 – Интерфейс страницы «Профиль пользователя» с календарем

Раздел «Статьи» предоставляет пользователю доступ к библиотеке информационных материалов по темам фитнеса, методик тренировок и здорового образа жизни. На главном экране раздела отображается перечень статей с их заголовками и краткими аннотациями, позволяя быстро ознакомиться с содержанием и выбрать интересующий материал. Прокрутка ленты статей демонстрирует разнообразие доступных тем, таких как «Разминка и заминка», «Питание для фитнеса» и «Повторения, подходы, отдых».

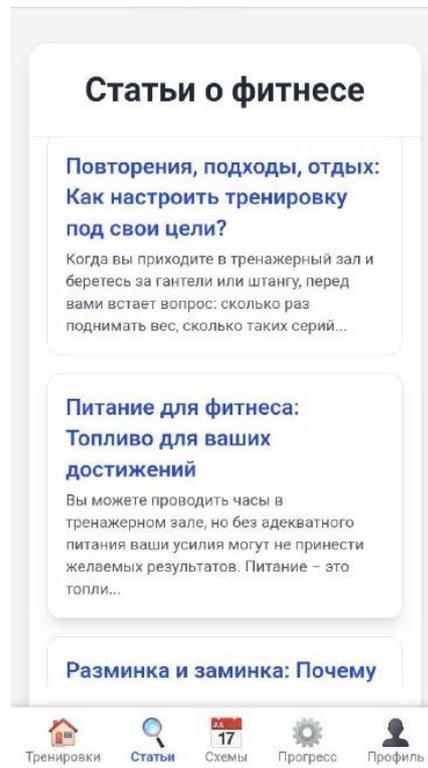


Рисунок 32 – Интерфейс раздела «Статьи о фитнесе» со списком материалов

При выборе конкретной статьи открывается экран с её полным текстом. Статьи структурированы для удобного чтения, могут содержать подзаголовки и списки. Раздел призван предоставить пользователям, особенно новичкам, необходимую теоретическую базу.

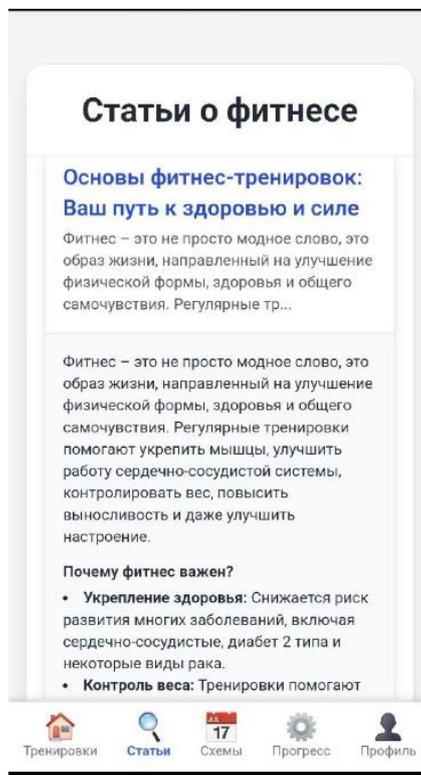


Рисунок 33 – Просмотр полного текста статьи «Основы фитнес-тренировок»

На странице «Мои тренировки» отображается перечень всех ранее сохраненных занятий. Каждая запись в списке содержит ключевую информацию: название или тип тренировки, дату проведения, длительность, а также краткий перечень выполненных упражнений и заметки, если они были добавлены. Это позволяет пользователю быстро ориентироваться в своей тренировочной истории. В верхней части экрана расположена кнопка «Добавить», предназначенная для перехода к форме создания новой тренировки.



Рисунок 34 – Интерфейс страницы «Мои тренировки» со списком занятий

При выборе конкретной тренировки из списка открывается экран с её полной детализацией. На этом экране представлена исчерпывающая информация о выбранном занятии, включая его название, точную дату и общую длительность. Основную часть экрана занимает детализированный список выполненных упражнений с указанием количества подходов, повторений и времени отдыха для каждого из них. Также отображаются все заметки, которые пользователь оставил к данной тренировке. Для навигации предусмотрена кнопка «К списку тренировок», а также функция «Удалить» и «Сохранить» для управления сохраненными записями.

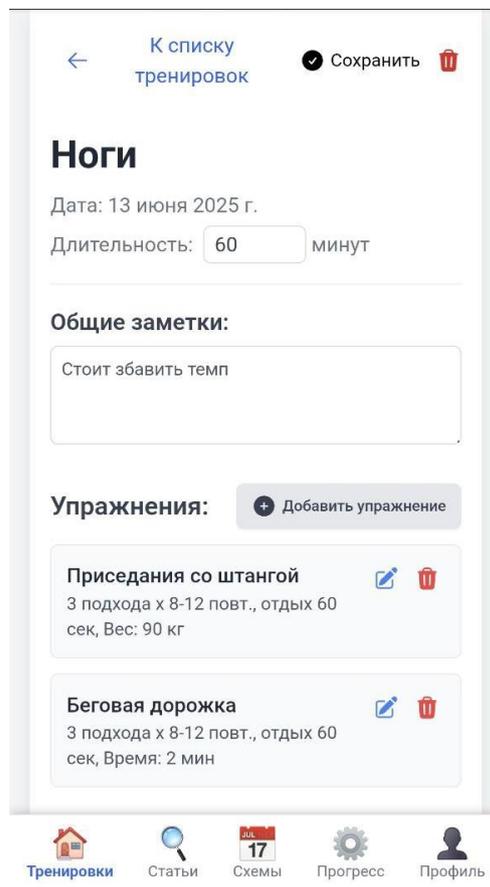


Рисунок 35 – Интерфейс страницы с детализацией выбранной тренировки

Для анализа динамики своих результатов пользователь может перейти в раздел «Прогресс по упражнениям». На этой странице представлен список упражнений, по которым ведётся или может вестись отслеживание силовых показателей или других метрик. Пользователь выбирает интересующее его упражнение для просмотра детальной информации.

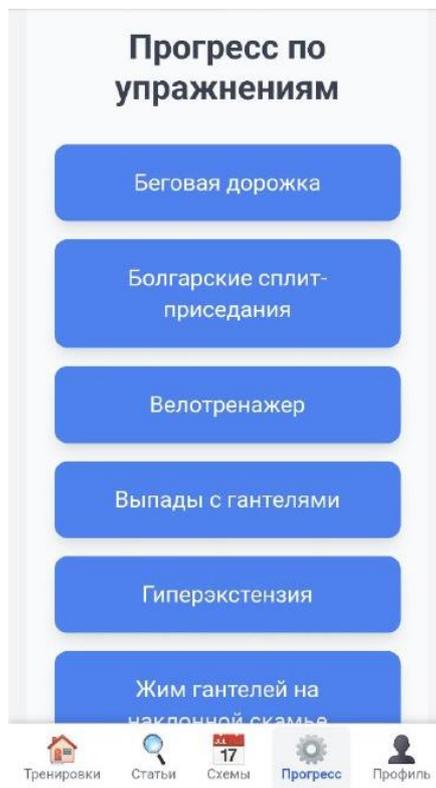


Рисунок 36 – Интерфейс выбора упражнения для отслеживания прогресса

После выбора упражнения, например, «Приседания со штангой», открывается страница с детальным отображением прогресса. Центральным элементом является график, наглядно демонстрирующий изменение рабочего веса (ось Y) в зависимости от даты выполнения упражнения (ось X). На представленном рисунке виден рост рабочего веса в приседаниях со штангой с 79 кг (12.05.2025) до 85 кг (16.05.2025). Под графиком приводится «История тренировок» в табличном виде, где указаны конкретные даты и достигнутые в эти дни результаты, что дополняет визуализацию графика точными данными.

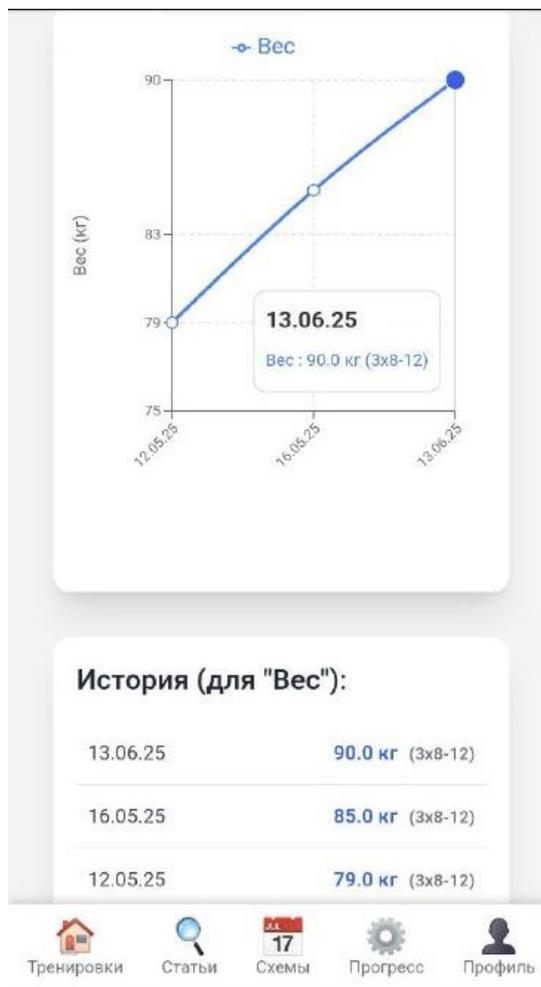


Рисунок 37 – График прогресса и история результатов по упражнению «Приседания со штангой»

Раздел «Тренировочные схемы» позволяет пользователям управлять персонализированными тренировочными схемами. На соответствующей странице отображается список созданных или доступных схем. Каждая запись содержит название, краткое описание и длительность схемы в неделях, а также количество дней в ней. Пользователь может просмотреть детали схемы, активировать её как текущую, что позволяет автоматически генерировать тренировки на будущие даты, редактировать или удалить схему.

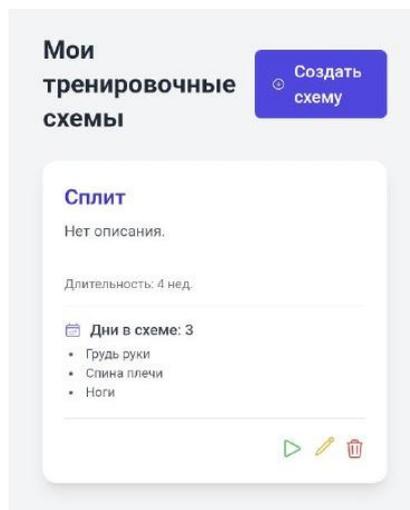


Рисунок 38 – Список тренировочных схем пользователя

Интерфейс для создания новой схемы или редактирования существующей позволяет задавать её название, описание, предполагаемую длительность. В рамках схемы пользователь может определять тренировочные дни, присваивая им порядок и название, а также добавлять в каждый день упражнения из справочника, указывая для них целевые параметры: подходы, повторения, время отдыха, целевой вес, дистанцию или продолжительность.

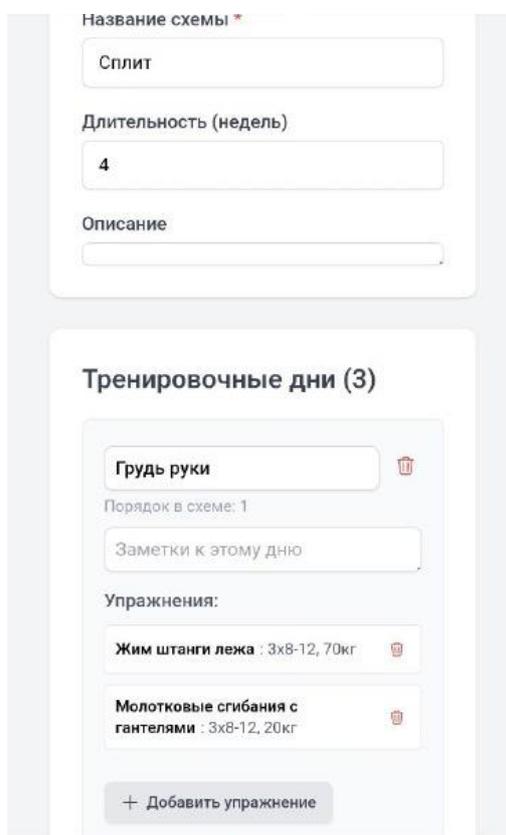


Рисунок 39 – Создание тренировочной схемы

В разделе «Библиотека упражнений» представлен полный каталог определений упражнений, доступных в системе. Пользователь может просматривать список упражнений, использовать функцию поиска по названию и фильтрацию по категории или мышечной группе.

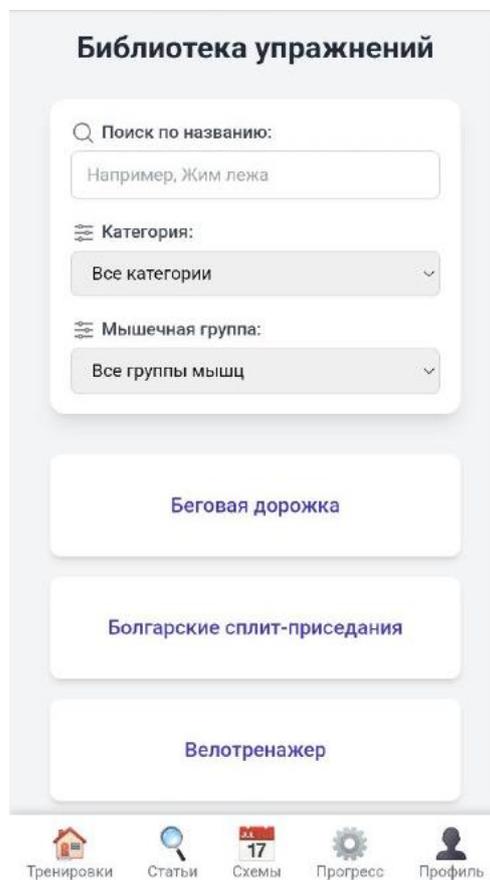


Рисунок 40 – Интерфейс библиотеки упражнений

При выборе любого упражнения отображается его подробное описание, включая инструкцию по технике выполнения, указание задействованных мышечных групп и необходимого инвентаря.

## Болгарские сплит-приседания

**Описание:**

Усложненный вариант выпадов, где задняя нога располагается на возвышении (скамье или специальной опоре). Это упражнение отлично прорабатывает мышцы передней ноги (квадрицепс, ягодицы, бицепс бедра) в одностороннем режиме, что способствует устранению мышечного дисбаланса, а также активно развивает баланс, координацию и стабильность.

**Техника выполнения:**

1. Встаньте спиной к скамье (или возвышению) на расстоянии примерно 2-3 шагов. Возьмите гантели в руки (если используете отягощение).
2. Положите подъем стопы одной ноги на край скамьи. Передняя нога стоит на полу, обеспечивая основную опору.
3. На вдохе, сгибая переднюю ногу в колене, опуститесь вниз до тех пор, пока бедро

передней ноги не станет параллельно полу.



Тренировки Статьи Схемы Прогресс Профиль

Рисунок 41 – Детализация упражнения из библиотеки

Представленная информация позволяет пользователю углубить свои знания о технике выполнения и задействованных мышцах. Это способствует более осознанному и эффективному планированию тренировок.

## 4 БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ

### 4.1 Безопасность

Безопасность труда разработчика программного обеспечения, занятого созданием мини-приложения, определяется условиями работы с персональным компьютером (ПК), включая организацию рабочего места, эргономику и принципы взаимодействия с графическим пользовательским интерфейсом. Длительная, монотонная работа в сидячем положении, а также интенсивная зрительная нагрузка могут приводить к развитию ряда профессиональных заболеваний и общему утомлению. Поэтому приоритетной задачей является создание условий, минимизирующих эти риски и поддерживающих высокую работоспособность.

Обеспечение безопасных условий труда базируется на гигиенических нормативах и требованиях, а также общих требованиях к организации рабочего места, установленных в Российской Федерации. Ключевые аспекты условий труда включают:

- микроклимат. В помещениях, где осуществляется работа с ПК, должны поддерживаться оптимальные параметры температуры, относительной влажности и скорости движения воздуха, согласно СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания»;

- освещение. Рабочее место должно иметь как естественное, так и искусственное освещение, равномерное и достаточное, без создания бликов на экране монитора;

- шум и вибрация. Уровни шума на рабочем месте не должны превышать допустимых значений, а вибрация, как правило, отсутствует или находится в пределах допустимых норм;

- электромагнитные поля (ЭМП). Современные ПК и мониторы генерируют ЭМП на уровнях, не превышающих гигиенические нормативы, что под-

тверждается ГОСТ Р 50948-2001 «Средства отображения информации индивидуального пользования. Общие эргономические требования и требования безопасности».

Правильная организация рабочего места разработчика имеет решающее значение для предотвращения утомляемости и развития заболеваний опорно-двигательного аппарата, а также зрительного анализатора. Общие требования к организации безопасного рабочего места утверждены Приказом Министерства труда и социальной защиты РФ от 29 октября 2021 г. N 774н «Об утверждении общих требований к организации безопасного рабочего места». Основные аспекты эргономики рабочего места включают:

- рабочий стол. Высота стола должна обеспечивать удобное положение рук и ног, а глубина и ширина – достаточное пространство для размещения оборудования и рабочей документации;

- рабочее кресло. Кресло должно быть подъемно-поворотным, регулируемым по высоте сиденья и спинки, с поддержкой поясничного отдела и подлокотниками;

- размещение монитора. Экран монитора следует располагать на расстоянии 60-70 см от глаз пользователя, а его верхний край – на уровне глаз или немного ниже, чтобы минимизировать напряжение мышц шеи и глаз;

- клавиатура и мышь. Клавиатура и мышь должны располагаться так, чтобы при работе предплечья были параллельны полу, а кисти рук находились в нейтральном положении;

- пространство и планировка. Рабочее место должно обеспечивать достаточное пространство для движений и смены рабочей позы, что соответствует принципам ГОСТ Р 58751-2019 «Слаботочные системы. Кабельные системы. Телекоммуникационные пространства и помещения. Рабочее место».

Режим труда и отдыха. Длительная непрерывная работа за компьютером приводит к значительному утомлению, поэтому необходимо строго соблюдать регламентированные перерывы, например, каждые 45-60 минут работы за ПК

делать 5-10-минутные перерывы. Во время перерывов рекомендуется выполнять упражнения для глаз и легкую физическую разминку. Это способствует психофизиологическому восстановлению и поддержанию высокой продуктивности.

Организация графического интерфейса пользователя (GUI) разработанного мини-приложения. Эргономика и безопасность при взаимодействии с программным обеспечением не ограничиваются лишь условиями труда и аппаратным обеспечением. Сам графический интерфейс разработанного мини-приложения для учета спортивных тренировок также должен соответствовать эргономическим требованиям, чтобы минимизировать зрительное и когнитивное утомление пользователя и обеспечить удобство его эксплуатации. В соответствии с ГОСТ Р ИСО 9241-161-2016 «Эргономика взаимодействия человек-система. Часть 161. Элементы графического пользовательского интерфейса», при проектировании интерфейса мини-приложения учтены следующие аспекты:

- визуальное оформление. Интерфейс адаптирует цветовую схему под текущую тему Telegram, обеспечивая чёткое и контрастное отображение информации, с достаточными размерами шрифтов для комфортного чтения;

- информативность и предсказуемость. Элементы управления интуитивно понятны и логично расположены, их реакция на действия пользователя предсказуема, а система предоставляет своевременную обратную связь о выполняемых операциях и ошибках, используя стандартные уведомления Telegram SDK;

- доступность. Приложение наследует нативные механизмы доступности Telegram, что соответствует принципам ГОСТ Р 52872—2019 «Интернет-ресурсы и другая информация, представленная в электронно-цифровой форме. Приложения для стационарных и мобильных устройств, иные пользовательские интерфейсы. Требования доступности для людей с инвалидностью и других лиц с ограничениями жизнедеятельности», делая его удобным для широкого круга пользователей.

Соблюдение перечисленных требований к условиям труда, организации рабочего места и эргономике разработанного графического интерфейса позволяет создать безопасную и продуктивную среду как для разработчика на этапе создания программного продукта, так и для конечных пользователей мини-приложения в процессе его эксплуатации, способствуя их здоровью, эффективности и положительному опыту взаимодействия с программным продуктом.

## **4.2 Экологичность**

Деятельность по разработке программного обеспечения, такая как создание мини-приложения для учета спортивных тренировок, сама по себе не является прямым источником значительных промышленных загрязнений или выбросов. Однако использование персонального компьютера, периферийного оборудования, потребление электроэнергии и образование различных видов отходов неизбежно оказывают косвенное воздействие на окружающую среду. Ответственный подход к минимизации этого воздействия является важным аспектом современной профессиональной деятельности в сфере информационных технологий.

Основными источниками отходов, которые могут образовываться в процессе разработки и эксплуатации программного обеспечения (включая как оборудование разработчика, так и серверную инфраструктуру), являются:

- вышедшая из строя или устаревшая оргтехника. Включает персональные компьютеры, мониторы, принтеры, сетевое оборудование и серверы, содержащие драгоценные и цветные металлы, а также опасные вещества, требующие специальной утилизации;

- отработанные расходные материалы. Это картриджи для принтеров и различные типы батареек и аккумуляторов, содержащие токсичные компоненты и тяжелые металлы, требующие отдельного сбора для переработки;

– макулатура. Образуется от использования бумаги для печати документов, черновиков и упаковочных материалов, чье производство требует значительных природных ресурсов и которое при неправильной утилизации увеличивает объем полигонов;

– люминесцентные лампы. Используются для освещения помещений и содержат пары ртути, которая является высокотоксичным веществом, опасным для здоровья человека и окружающей среды при нарушении целостности ламп;

– пластиковые отходы. Включают упаковку от оборудования и канцтоваров, а также изношенные пластиковые компоненты, которые разлагаются на протяжении сотен лет и способствуют образованию микропластика, загрязняющего экосистемы.

В соответствии с действующим законодательством Российской Федерации, в частности, Федеральным законом № 89-ФЗ «Об отходах производства и потребления» и СанПиН 2.1.3684-21 «Санитарно-эпидемиологические требования к содержанию территорий городских и сельских поселений, к водным объектам, питьевой воде и питьевому водоснабжению населения, атмосферному воздуху, почвам, жилым помещениям, эксплуатации производственных, общественных помещений, организации и проведению санитарно-противоэпидемических (профилактических) мероприятий», отходы, образующиеся в процессе деятельности организаций и граждан, подлежат сбору, накоплению, транспортированию, обработке, утилизации, обезвреживанию и размещению таким образом, чтобы минимизировать их негативное воздействие на окружающую среду и здоровье человека.

Для организации-разработчика и организации, эксплуатирующей серверную инфраструктуру приложения, а также для индивидуального разработчика, обращение с отходами должно быть организовано следующим образом:

– накопление. Вышедшие из строя компьютеры, мониторы, принтеры и другое электронное оборудование временно складываются в специально предназначенных безопасных зонах, исключающих несанкционированный доступ и потенциальное загрязнение;

– хранение. До момента передачи специализированным предприятиям, отходы должны храниться в условиях, которые предотвращают их повреждение и возможную утечку вредных веществ (например, ртути из люминесцентных ламп или тяжелых металлов из печатных плат);

– передача специализированным организациям. Электронные отходы, а также другие потенциально опасные отходы, такие как люминесцентные лампы или батарейки, должны передаваться исключительно лицензированным компаниям, имеющим разрешение на работу с отходами соответствующих классов опасности;

– учет и отчетность. Организации обязаны вести строгий учет всех образующихся отходов и оформлять документацию, подтверждающую факт их передачи на утилизацию или переработку, в соответствии с требованиями действующего законодательства.

Хотя непосредственная разработка мини-приложения не генерирует большого количества промышленных отходов, разработчик может способствовать снижению общего экологического следа, придерживаясь принципов экологически ответственного поведения, как на этапе создания программного продукта, так и косвенно влияя на экологичность его эксплуатации. Ключевые меры включают:

– энергосбережение. Выключать компьютер и периферийные устройства по окончании рабочего дня, а также применять энергосберегающие режимы работы, что уменьшает расход электроэнергии и сокращает выбросы парниковых газов;

– экономия бумаги. По возможности использовать электронный документооборот, избегать излишней печати, а при необходимости печати использовать двустороннюю печать и выбирать бумагу из вторсырья;

– продление срока службы оборудования. Бережное отношение к технике, своевременное проведение профилактического обслуживания и ремонт могут значительно отсрочить момент её списания и образования электронных отходов;

– участие в программах раздельного сбора. Активно участвовать в сборе макулатуры, пластика, батареек и электронных отходов, если такие программы организованы на территории (например, учебного заведения или муниципального района);

– оптимизация программного кода. Разработка эффективного программного обеспечения, требующего меньших вычислительных ресурсов, способствует снижению энергопотребления на пользовательских устройствах и серверной инфраструктуре, что косвенно снижает «цифровой углеродный след» проекта.

Таким образом, экологичность разработки мини-приложения косвенно связана с общей экологической культурой разработчика и его способностью применять принципы минимизации отходов и ответственного потребления ресурсов в своей повседневной деятельности. Соблюдение требований законодательства по обращению с отходами, образующимися от использования техники и других материалов, является ключевым фактором минимизации негативного воздействия на окружающую среду, а оптимизация самого программного продукта вносит свой вклад в устойчивое развитие IT-индустрии.

### **4.3 Чрезвычайные ситуации**

Работа над мини-приложением, осуществляемая на персональном компьютере в рабочем помещении, сопряжена с потенциальными рисками возникновения чрезвычайных ситуаций (ЧС), преимущественно техногенного характера. Наиболее вероятной и опасной из них является пожар. Поэтому в любом рабочем пространстве должны быть предусмотрены и реализованы меры по предупреждению и ликвидации таких ЧС, направленные на обеспечение безопасности находящихся там людей и сохранности имущества.

Возможные причины возникновения пожара в рабочем помещении или в здании, где оно расположено, включают:

- неисправность электрооборудования и электропроводки. Включает короткие замыкания из-за поврежденной изоляции, перегрузку сети при одновременном включении нескольких мощных электроприборов, а также использование несертифицированных или самодельных устройств;

- неосторожное обращение с огнем. Это курение в неустановленных местах, оставление без присмотра электронагревательных приборов, использование открытого огня (свечей, спичек) вблизи горючих материалов или небрежное обращение с легковоспламеняющимися жидкостями;

- нарушение правил пожарной безопасности. Выражается в самовольном подключении к электросетям, использовании неисправных или несертифицированных электроприборов, а также в загромождении эвакуационных путей и выходов;

- поджог. Это умышленное действие третьих лиц, направленное на причинение ущерба имуществу или нанесение вреда людям, что может привести к масштабным пожарам;

- внешние факторы. Такие как пожары в соседних помещениях или зданиях, распространяющиеся на рабочее пространство, а также природные явления, например, удары молнии, способные вызвать возгорание.

Меры по предупреждению пожара в рабочем помещении:

Для предотвращения возникновения пожара и минимизации его последствий в любом помещении, где ведется разработка программного обеспечения, должен быть реализован комплекс организационных и технических мероприятий в соответствии с Федеральным законом № 123-ФЗ «Технический регламент о требованиях пожарной безопасности» и Правилами противопожарного режима в Российской Федерации, утверждёнными Постановлением Правительства РФ.

Организационные мероприятия. Эти меры направлены на обучение и информирование всех лиц, находящихся в помещении, а также на установление строгих правил и контроль за их соблюдением:

- инструктажи и обучение. Для всех лиц, использующих помещение, должны проводиться регулярные инструктажи по пожарной безопасности, включая ознакомление с правилами поведения при пожаре, расположением эвакуационных выходов и первичных средств пожаротушения;

- разработка документации. В помещении или здании должны быть разработаны и утверждены планы эвакуации людей при пожаре, схемы расположения средств пожаротушения, которые должны быть размещены на видных местах;

- контроль за электрооборудованием. Должен быть введен строгий запрет на использование неисправных электроприборов, самодельных удлинителей, а также на перегрузку электрической сети;

- содержание территории и помещений. Регулярная уборка рабочих мест и общих зон от горючих материалов и мусора, а также запрет на загромождение путей эвакуации.

Технические мероприятия. Эти меры направлены на оснащение помещения специальными средствами и системами, предназначенными для обнаружения, локализации и тушения пожара:

- системы автоматической пожарной сигнализации (АПС) и оповещения и управления эвакуацией людей при пожаре (СОУЭ). Все помещения должны быть оборудованы исправными системами АПС и СОУЭ для своевременного оповещения и указания путей эвакуации;

- первичные средства пожаротушения. Наличие и исправное состояние огнетушителей и пожарных кранов (если предусмотрено), достаточных для тушения возможных классов пожаров, включая электрооборудование;

- эвакуационные пути и выходы. Обеспечение достаточного количества эвакуационных выходов, их свободное состояние, отсутствие замков, препятствующих свободному открытию дверей изнутри;

– система видеонаблюдения. Установка камер в общих коридорах и на входах в здание для мониторинга ситуации и оперативного реагирования на происшествия.

Действия лиц, находящихся в помещении, при возникновении пожара.

При обнаружении пожара или признаков горения (задымление, запах гари, повышение температуры) каждый находящийся в помещении обязан действовать четко и оперативно, следуя установленным правилам пожарной безопасности:

– немедленно сообщить о пожаре. Следует вызвать пожарную охрану по телефону «101» или «112», указав точный адрес, место возникновения пожара и что горит, а также сообщить ответственному лицу или администрации;

– принять меры по эвакуации людей. После сообщения о пожаре необходимо немедленно покинуть помещение, используя ближайший эвакуационный выход, и сообщить о пожаре другим людям, помогая тем, кто нуждается в помощи;

– приступить к тушению пожара первичными средствами. Если это безопасно и очаг возгорания небольшой, попытаться потушить пожар имеющимися первичными средствами пожаротушения, но при угрозе для жизни немедленно эвакуироваться;

– организовать встречу подразделений пожарной охраны. После эвакуации, если это возможно и безопасно, встретить пожарные подразделения и указать им точное место возгорания, наличие людей в здании и другую важную информацию.

Соблюдение всех перечисленных мер по предупреждению пожара и готовность находящихся в помещении лиц к правильным действиям в случае его возникновения являются залогом обеспечения безопасности и минимизации возможных последствий чрезвычайной ситуации в любом рабочем пространстве, где ведется разработка программного обеспечения.

## ЗАКЛЮЧЕНИЕ

В результате выполнения работы было разработано и описано мини-приложение для учета индивидуальных спортивных тренировок, предназначенное для платформы Telegram Mini Apps.

В ходе исследования проведен анализ предметной области, выявлены потребности целевой аудитории и изучены существующие решения, что подтвердило актуальность создания доступного и функционального инструмента для тренирующихся. На основе этого анализа были сформулированы требования к приложению.

Проектирование системы включало выбор клиент-серверной архитектуры с использованием современных технологий: React для клиентской части и Python с FastAPI для серверной, с базой данных PostgreSQL. Были разработаны функциональные спецификации, структура базы данных и пользовательский интерфейс.

В процессе реализации были созданы основные модули клиентского и серверного приложений, обеспечивающие ключевую функциональность: аутентификацию пользователей, создание и редактирование тренировок и тренировочных схем, отслеживание прогресса по упражнениям и доступ к информационным статьям и базе упражнений. Продемонстрирована работоспособность разработанного мини-приложения.

Таким образом, цель выпускной квалификационной работы – разработка мини-приложения для учета индивидуальных спортивных тренировок – была достигнута. Созданный программный продукт решает поставленные задачи, предоставляя пользователям удобный инструмент для ведения тренировочного процесса и анализа результатов, и имеет потенциал для дальнейшего развития и расширения функциональности.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Бэнкс, А. React и Redux: функциональная веб-разработка / А. Бэнкс, Е. Порселло; пер. с англ. – Санкт-Петербург : Питер, 2021. – 336 с. – ISBN 978-5-4461-1781-0.

2 Васкес, М. Python для разработки веб-приложений с FastAPI: создание высокопроизводительных API / М. Васкес; пер. с англ. – Москва : ДМК Пресс, 2023. – 432 с. – ISBN 978-5-93700-131-9.

3 Вейнгартнер, С. PostgreSQL. Основы и практика / С. Вейнгартнер, Г. Хазе; пер. с нем. – Санкт-Петербург : БХВ-Петербург, 2022. – 816 с. – ISBN 978-5-9775-1065-5.

4 Документация FastAPI [Электронный ресурс]. – Режим доступа : <https://fastapi.tiangolo.com/>. – 03.02.2025.

5 Документация React [Электронный ресурс]. – Режим доступа : <https://react.dev/>. – 16.02.2025.

6 Документация SQLAlchemy [Электронный ресурс]. – Режим доступа : <https://www.sqlalchemy.org/>. – 17.03.2025.

7 Документация Tailwind CSS [Электронный ресурс]. – Режим доступа : <https://tailwindcss.com/docs>. – 13.04.2025.

8 Документация Telegram Mini Apps [Электронный ресурс]. – Режим доступа : <https://core.telegram.org/bots/webapps>. – 08.02.2025.

9 Кауфман, Л. PostgreSQL для начинающих. Проектирование и администрирование баз данных / Л. Кауфман, П. Крейг; пер. с англ. – Москва : Вильямс, 2021. – 960 с. – ISBN 978-5-907114-73-0.

10 Кардаш, Т. А. Эргономика рабочих мест служащих и инженерно-технических работников, оснащенных ПЭВМ : учеб. пособие / Т. А. Кардаш ; АмГУ, ИФФ. – Благовещенск : Изд-во Амур. гос. ун-та, 2002. – 60 с.

11 Лутц, М. Изучаем Python. Программирование игр, визуализация данных, веб-приложения / М. Лутц; пер. с англ. – 5-е изд. – Санкт-Петербург : Диалектика, 2021. – Том 2. – 880 с. – ISBN 978-5-907365-34-7.

- 12 МакГрат, М. Python. Программирование для начинающих / М. МакГрат; пер. с англ. – Москва : Эксмо, 2023. – 192 с. – ISBN 978-5-04-119015-2.
- 13 Пособие по безопасной работе на персональных компьютерах / разработ. В. К. Шумилин. – Москва : НЦ ЭНАС, 2005. – 28 с.
- 14 Рамбо, Д. Изучаем React. Современная разработка веб-приложений / Д. Рамбо, Э. Фридман; пер. с англ. – Санкт-Петербург : Питер, 2022. – 512 с. – ISBN 978-5-4461-1830-5.
- 15 Саммерфилд, М. Python на практике. Создание качественных приложений / М. Саммерфилд; пер. с англ. – Санкт-Петербург : БХВ-Петербург, 2020. – 576 с. – ISBN 978-5-9775-6630-0.
- 16 Скляр, А. В. Создание чат-ботов на Python / А. В. Скляр. – Санкт-Петербург : БХВ-Петербург, 2021. – 256 с. – ISBN 978-5-9775-4089-2.
- 17 Фримен, Э. React. Библиотека профессионала / Э. Фримен; пер. с англ. – Санкт-Петербург : Питер, 2023. – 784 с. – ISBN 978-5-4461-1910-8.
- 18 Шумилин, В.К. ПЭВМ. Защита пользователя / В.К. Шумилин. – Москва : Охрана труда и социальное страхование, 2001. – 214 с.

# ПРИЛОЖЕНИЕ А

## Техническое задание

### 1 ОБЩИЕ СВЕДЕНИЯ

#### 1.1 Полное наименование программного продукта

Мини-приложение для учета индивидуальных спортивных тренировок.

Условное обозначение: FitTrack App.

#### 1.2 Плановые сроки:

- начало работ: 02.10.2024;
- окончание работ: 10.06.2025.

### 2 НАЗНАЧЕНИЕ И ЦЕЛИ ПРОГРАММНОГО ПРОДУКТА

#### 2.1 Назначение приложения

Приложение предназначено для предоставления пользователям Telegram удобного цифрового инструмента для ведения дневника индивидуальных спортивных тренировок, систематического отслеживания прогресса и доступа к релевантной информации по фитнесу. Оно направлено на упрощение учета занятий для самостоятельно тренирующихся людей и повышение их мотивации за счет наглядности результатов.

#### 2.2 Цели программного продукта

Цели программного продукта включают в себя:

- создать легковесное, интуитивно понятное и функциональное мини-приложение, интегрированное в экосистему Telegram;
- обеспечить возможность детализированного учета выполненных тренировок и планирования занятий на основе персонализированных схем;
- предоставить инструменты визуализации прогресса по ключевым показателям упражнений;
- сформировать базу информационных материалов по фитнесу и упражнениям для пользователей.

Критерии оценки успеха:

## Продолжение ПРИЛОЖЕНИЯ А

- полная реализация функциональных требований;
- время отклика на типовые операции (например, сохранение тренировки, загрузка списка статей) не превышает 5 секунд;
- Успешная автоматическая идентификация пользователя Telegram при первом входе.

### 3 ТРЕБОВАНИЯ К ПРОГРАММНОМУ ПРОДУКТУ

#### 3.1 Требования к программному продукту в целом

Структура и функционирование. Приложение представляет собой клиент-серверное решение.

- клиентская часть (Frontend). Telegram Mini App, реализующая пользовательский интерфейс и взаимодействующая с Telegram Web Apps SDK;
- серверная часть (Backend), реализующая бизнес-логику, взаимодействие с базой данных и предоставление API-интерфейсов;
- база данных. Для хранения данных и обеспечения их целостности;
- основные функциональные модули. Модуль управления тренировками, модуль планирования тренировок (схемы), модуль отслеживания прогресса, модуль просмотра статей и справочников, модуль профиля пользователя.

Численность и квалификация персонала:

- персонал по разработке. Разработчик, обладающий знаниями в области Python (FastAPI, SQLAlchemy), JavaScript (React), PostgreSQL, принципов работы с API и Telegram Mini Apps;
- персонал по эксплуатации. Конечные пользователи с базовыми навыками работы со смартфоном и мессенджером Telegram. Специальное обучение не требуется благодаря интуитивно понятному интерфейсу.

Надежность:

## Продолжение ПРИЛОЖЕНИЯ А

- приложение должно обеспечивать стабильное функционирование и сохранность пользовательских данных при штатных нагрузках;
- допустимая частота ошибок на типовые операции (например, сохранение/загрузка данных) не более 1% от общего числа операций;
- время доступности серверной части не менее 90% в период эксплуатации.

### Безопасность:

- использование безопасных протоколов для передачи данных (HTTPS);
- авторизация пользователей на основе данных, предоставляемых Telegram Web Apps SDK.

### Эргономика:

- интерфейс должен быть интуитивно понятным, простым в освоении и использовании для целевой аудитории (от новичков до среднего уровня) в соответствии с ГОСТ Р ИСО 9241-161-2016;
- соответствие дизайна и навигации рекомендациям Telegram по стилистике Mini Apps;
- полная русскоязычная локализация интерфейса и контента;
- соответствие требованиям доступности для пользователей, в том числе с ограничениями жизнедеятельности, согласно ГОСТ Р 52872—2019;
- требования к средствам отображения информации индивидуального пользования должны соответствовать ГОСТ Р 50948-2001.

### 3.2 Требования к функциям приложения

- идентификация и авторизация пользователя. Приложение должно обеспечивать идентификацию и автоматическую регистрацию пользователя при первом входе на основе его данных из Telegram;

## Продолжение ПРИЛОЖЕНИЯ А

– управление тренировками: Пользователь должен иметь возможность создавать, просматривать, редактировать и удалять записи о своих тренировках, указывая дату, тип, продолжительность, заметки, а также добавлять в тренировку упражнения с указанием названия, количества подходов, повторений, времени отдыха, рабочего веса, дистанции и продолжительности;

– просмотр информационных статей. Приложение должно предоставлять доступ к списку информационных статей с возможностью их просмотра;

– отслеживание прогресса. Пользователь должен иметь возможность отслеживать прогрессию нагрузок на графиках (например, «вес/дата»);

– профиль пользователя. В профиле пользователя должна отображаться его основная информация из Telegram и календарь для отображения отмеченных проведенных тренировок;

– планирование тренировок. Пользователь должен иметь возможность создавать персонализированные тренировочные схемы, состоящие из последовательности тренировочных дней с набором упражнений и их целевыми параметрами. Должна быть реализована функциональность активации выбранной схемы в качестве текущего тренировочного плана;

– генерация тренировок из схем. Приложение должно обеспечивать автоматическое формирование записей о будущих тренировках на основе активной схемы;

– навигация. должна быть обеспечена интуитивно понятная навигация между основными разделами;

– библиотека упражнений. пользователь должен иметь возможность просматривать и искать информацию в справочной базе определений упражнений.

## Продолжение ПРИЛОЖЕНИЯ А

### 3.3 Требования к видам обеспечения

Математическое обеспечение включает в себя: алгоритмы для агрегации данных о прогрессе (например, выбор максимального значения по метрике для конкретной даты), сортировки тренировок и упражнений, расчета длительности тренировок.

Информационное обеспечение:

- база данных. Реляционная СУБД PostgreSQL (версия не ниже 12);
- структура данных. Модели данных, определенные с использованием SQLAlchemy ORM, включая хранение упражнений в формате JSONB;
- схемы данных. Валидация входных/выходных данных с использованием Pydantic.

Программное обеспечение:

- клиентская часть. Разработка на JavaScript с использованием библиотеки React, инструмента сборки Vite, роутера React Router DOM, CSS-фреймворка Tailwind CSS и официального Telegram Web Apps SDK;
- серверная часть. Разработка на языке Python (версия не ниже 3.9) с использованием веб-фреймворка FastAPI, ORM SQLAlchemy, ASGI-сервера Uvicorn;
- вспомогательное ПО. Система контроля версий Git, менеджер пакетов npm (для Frontend) и pip (для Backend).

Техническое обеспечение:

- для пользователя. Смартфон или компьютер с установленным мессенджером Telegram (iOS, Android, Windows, macOS, Linux, Web-версия) и стабильным интернет-соединением;
- для серверной части. Сервер (виртуальный или физический) с установленной операционной системой Linux, доступом к PostgreSQL и возможностью развертывания Python/FastAPI приложения;

## Продолжение ПРИЛОЖЕНИЯ А

### 4 СОСТАВ И СОДЕРЖАНИЕ РАБОТ ПО СОЗДАНИЮ ПРОГРАММНОГО ПРОДУКТА

#### 4.1 Этапы работ

– анализ предметной области и существующих решений. Изучение потребностей целевой аудитории, обзор существующих решений и обоснование выбора платформы и технологий;

– проектирование мини-приложения. Разработка архитектуры приложения, детализация функциональных возможностей, проектирование логической и физической структуры базы данных;

– разработка мини-приложения. Реализация клиентской и серверной частей, интеграция модулей, создание API;

– тестирование и отладка. Проверка работоспособности, соответствия требованиям, выявление и исправление ошибок.

### 5 ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ ПРОГРАММНОГО ПРОДУКТА

#### 5.1 Контроль качества.

– функциональное тестирование. Проверка реализации всех заявленных функций;

– тестирование производительности. Оценка скорости отклика приложения на типовые операции;

– тестирование безопасности. Проверка защиты данных и устойчивости к несанкционированному доступу;

– UI/UX аудит. Оценка удобства использования и соответствия интерфейса эргономическим требованиям.

### 6 ТРЕБОВАНИЯ К ПОДГОТОВКЕ ОБЪЕКТА АВТОМАТИЗАЦИИ

#### 6.1 Подготовка персонала

## Продолжение ПРИЛОЖЕНИЯ А

Для использования мини-приложения конечными пользователями не требуется специальной подготовки, кроме базовых навыков работы со смартфоном/компьютером и мессенджером Telegram. Интуитивно понятный интерфейс и подсказки в приложении минимизируют потребность в дополнительном обучении.

### 6.2 Техническая подготовка

– на стороне пользователя. Наличие актуальной версии мессенджера Telegram на устройстве (смартфон/планшет/компьютер) и стабильного интернет-соединения;

– на стороне сервера. Обеспечение работоспособности хостинга и настроенной среды для развертывания Python/FastAPI приложения с подключением к базе данных PostgreSQL.

## 7 ТРЕБОВАНИЯ К ДОКУМЕНТИРОВАНИЮ

### 7.1 Перечень документов

– настоящее Техническое задание;

– пояснительная записка к выпускной квалификационной работе (согласно СТО СМК 4.2.3.21-2018).

## 8 ИСТОЧНИКИ РАЗРАБОТКИ

### 8.1 Нормативные документы

– ГОСТ 2.105-95. Единая система конструкторской документации. Общие требования к текстовым документам.

– ГОСТ 7.32-2001. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления.

– СТО СМК 4.2.3.21-2018. Стандарт организации. Оформление выпускных квалификационных и курсовых работ (проектов) / ФГБОУ ВО «Амурский государственный университет».

## Продолжение ПРИЛОЖЕНИЯ А

### 8.2 Литература:

- Васкес, М. Python для разработки веб-приложений с FastAPI: создание высокопроизводительных API.
- Вейнгартнер, С. PostgreSQL. Основы и практика.
- Рамбо, Д. Изучаем React. Современная разработка веб-приложений.
- Материалы по теории и методике спортивных тренировок, физиологии, питанию.

# ПРИЛОЖЕНИЕ Б

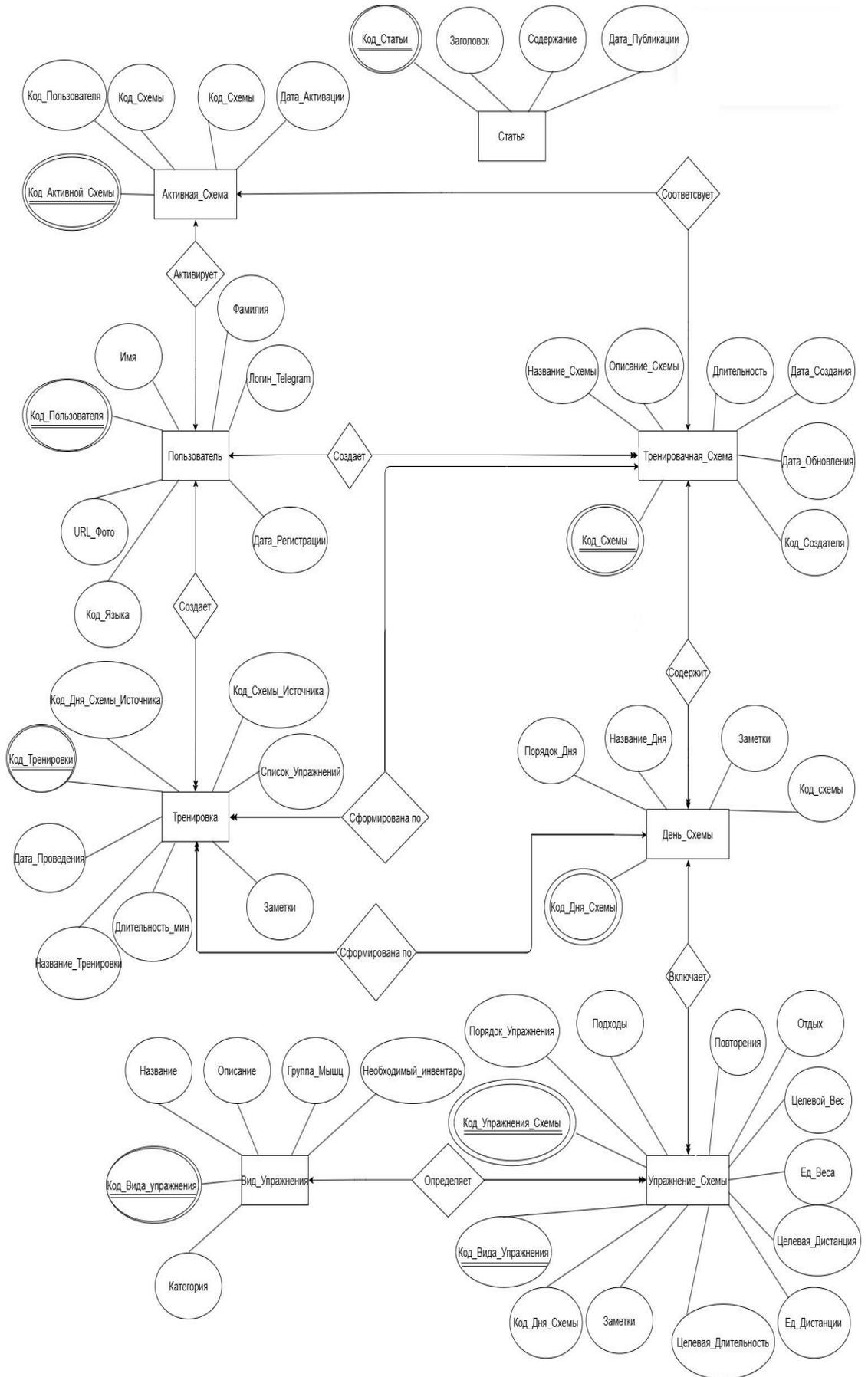


Рисунок Б.1 – Инфологическая модель базы данных

# ПРИЛОЖЕНИЕ В

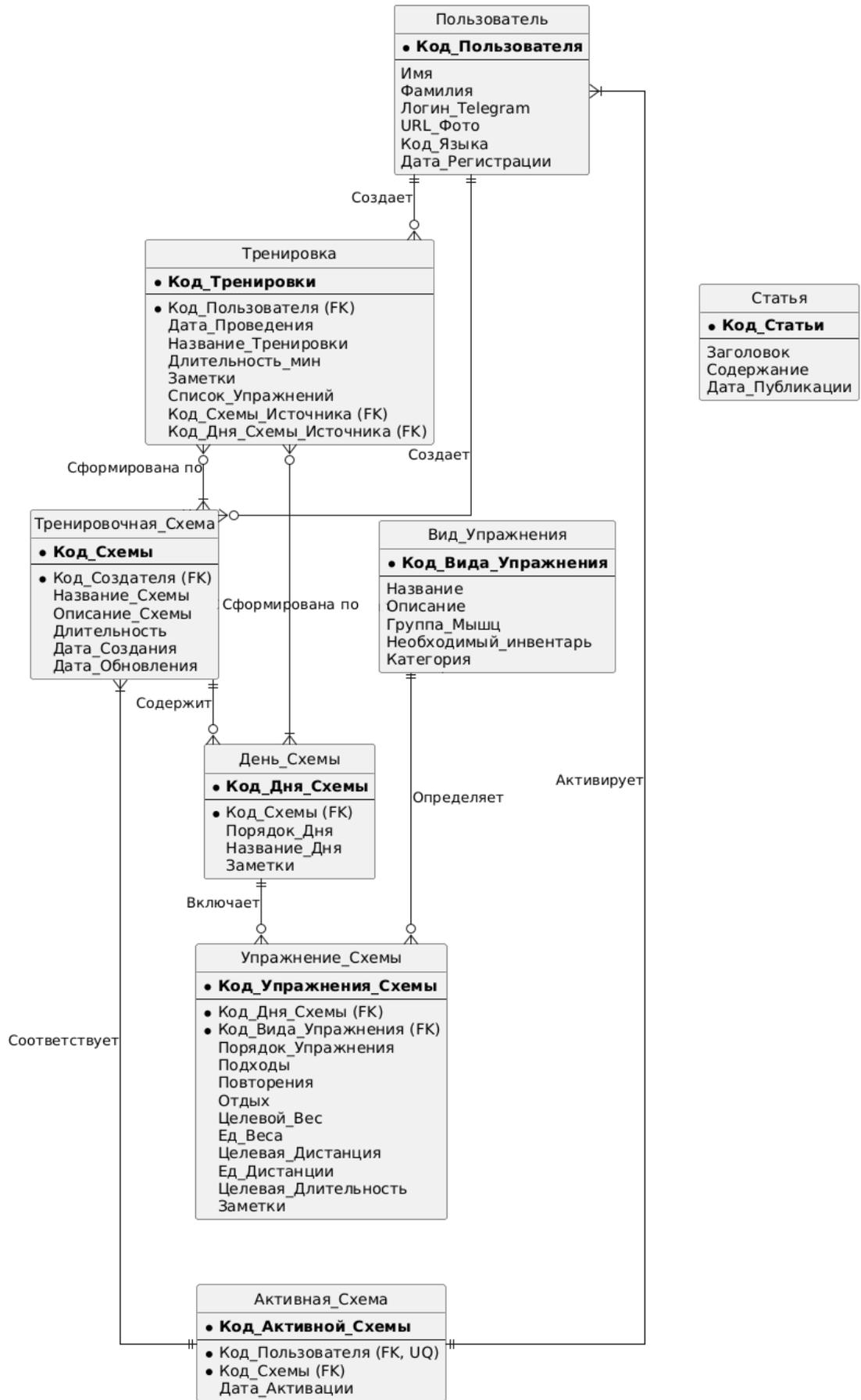


Рисунок В.1 – Логическая модель базы данных

# ПРИЛОЖЕНИЕ Г

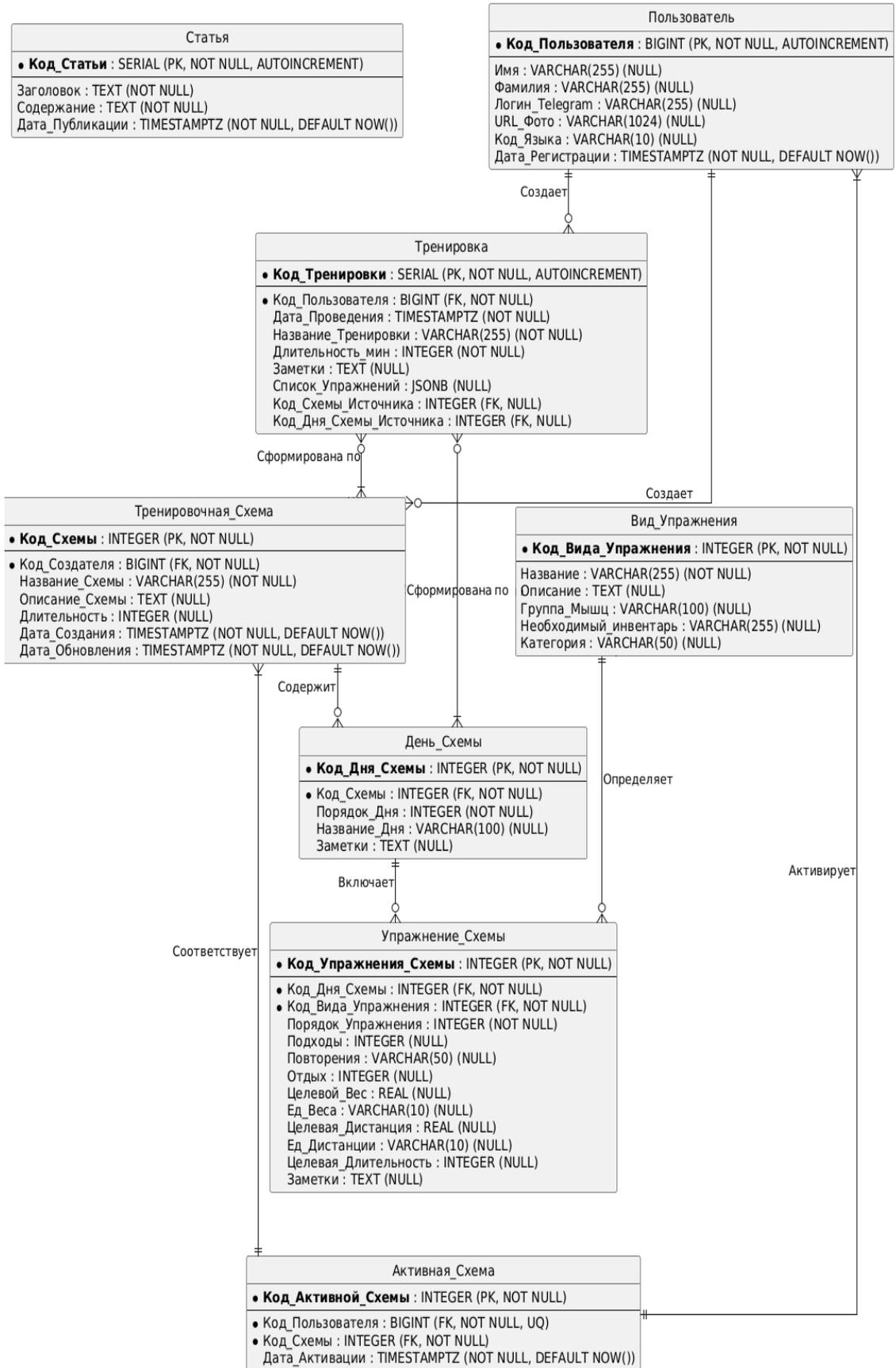


Рисунок Г.1 – Физическая модель базы данных