TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа информационных технологий и робототехники
Направление подготовки 09.03.04 «Программная инженерия»
Отделение школы (НОЦ) информационные технологии

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Проектирование и разработка мобильного приложения для учета спортивной активности

УДК 004.451:004.455.1:796

Студент

Группа	ФИО	Подпись	Дата
8К82	Власов Никита Константинович		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель ОИТ	Дорофеев В.А.	-		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор ОСГН	Гасанов М. А.	д.э.н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Мезенцева И. Л.	-		

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Чердынцев Е.С.	к.т.н.		

Томск – 2022

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ООП

Код компетенции	Наименование компетенции
УК(У)-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач
УК(У)-2	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений
УК(У)-3	Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде
УК(У)-4	Способен осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и иностранном(-ых) языке(-ах)
УК(У)-5	Способен воспринимать межкультурное разнообразие общества в социально-историческом, этическом и философском контекстах
УК(У)-6	Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни
УК(У)-7	Способен поддерживать должный уровень физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности
УК(У)-8	Способен создавать и поддерживать безопасные условия жизнедеятельности, в том числе при возникновении чрезвычайных ситуаций
УК(У)-9	Способен проявлять предприимчивость в практической деятельности, в т.ч. в рамках разработки коммерчески перспективного продукта на основе научно-технической идеи
ОПК(У)-1	Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности
ОПК(У)-2	Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности
ОПК(У)-3	Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности
ОПК(У)-4	Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью
ОПК(У)-5	Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем
ОПК(У)-6	Способен разрабатывать алгоритмы и программы, пригодные для практического применения
ОПК(У)-7	Способен применять в практической

	деятельности основные концепции, принципы, теории и факты, связанные с информатикой
ОПК(У)-8	Способен осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий
ПК(У)-1	Владение навыками разработки требований и проектирования программного обеспечения
ПК(У)-2	Владение навыками разработки документов и стратегии тестирования программного обеспечения
ПК(У)-3	Владение навыками моделирования, анализа и использования формальных методов конструирования программного обеспечения
ПК(У)-4	Владение навыками использования операционных систем, сетевых технологий, средств разработки программного интерфейса, применения языков и методов формальных спецификаций, систем управления базами данных
ПК(У)-5	Владение концепциями и атрибутами качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роли людей, процессов, методов, инструментов и технологий обеспечения качества

TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа информационных технологий и робототехники
Направление подготовки (специальность) 09.03.04 «Программная инженерия»
Отделение школы (НОЦ) информационных технологий

УТВЕРЖДАЮ:
Руководитель ООП
_____ _____ Чердынцев Е. С.
(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

Бакалаврской работы

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
8К82	Власов Никита Константинович

Тема работы:

Проектирование и разработка мобильного приложения для учета спортивной активности	
Утверждена приказом директора (дата, номер)	

Срок сдачи студентом выполненной работы:	

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе *(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).*	Объект исследования: учет спортивной активности с помощью мобильного приложения Предмет исследования: мобильное приложение для учета спортивной активности Режим работы: непрерывный

Перечень подлежащих исследованию, проектированию и разработке вопросов *(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).*	1. Аналитический обзор 2. Проектирование серверного приложения 3. Разработка серверного приложения 4. Финансовый менеджмент 5. Социальная ответственность
Перечень графического материала *(с точным указанием обязательных чертежей)*	Презентация в формате *.pptx

Консультанты по разделам выпускной квалификационной работы
(с указанием разделов)

Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Гасанов М. А.
Социальная ответственность	Мезенцева И. Л.

Названия разделов, которые должны быть написаны на русском и иностранном языках:

-

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель ОИТ	Дорофеев В. А.	-		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К82	Власов Никита Константинович		

TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа информационных технологий и робототехники
Направление подготовки (специальность) 09.03.04 «Программная инженерия»
Уровень образования бакалавриат
Отделение школы (НОЦ) информационных технологий
Период выполнения (осенний / весенний семестр 2021 /2022 учебного года)

Форма представления работы:

Бакалаврская работа

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
	Основная часть	60
	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	20
	Социальная ответственность	20

СОСТАВИЛ:
Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель ОИТ	Дорофеев В. А.	-		

СОГЛАСОВАНО:

Руководитель ООП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Чердынцев Е. С.	к.т.н.		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа	ФИО
8К82	Власов Никита Константинович

Школа	ИШИТР	Отделение школы (НОЦ)	ОИТ
Уровень образования	Бакалавриат	**Направление/специальность**	09.03.04 Программная инженерия

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:	
1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Оклад руководителя – 35000 руб. Оклад инженера – 20000 руб.
2. Нормы и нормативы расходования ресурсов	Доплаты и надбавки руководителя 40%; Доплаты и надбавки инженера 20%; Дополнительной заработной платы 12%; Накладные расходы 16%; Районный коэффициент 30%.
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	Коэффициент отчислений на уплату во внебюджетные фонды 30,2 %
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения	Определение потенциального потребителя результатов исследования, SWOT-анализ разработанной стратегии
2. Планирование и формирование бюджета научных исследований	Определение структуры работы. Расчет трудоемкости выполнения работ. Подсчет бюджета исследования
3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	Рассчитать показатели финансовой эффективности, ресурсоэффективности и эффективности исполнения
Перечень графического материала *(с точным указанием обязательных чертежей)*:	
1. Оценка конкурентоспособности технических решений 2. Матрица SWOT 3. Альтернативы проведения НИ	

4. График проведения и бюджет НИ *5.* Оценка ресурсной, финансовой и экономической эффективности НИ

Дата выдачи задания для раздела по линейному графику	4.03.2022

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
профессор ОСГН	Гасанов М.А.	Д.э.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К82	Власов Никита Константинович		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа		ФИО	
8К82		Власов Никита Константинович	
Школа	ИШИТР	**Отделение (НОЦ)**	ОИТ
Уровень образования	Бакалавриат	**Направление/ специальность**	09.03.04 Программная инженерия

Тема ВКР:

Проектирование и разработка мобильного приложения для учета спортивной активности	
Исходные данные к разделу «Социальная ответственность»:	
Введение – Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика) и области его применения. – Описание рабочей зоны (рабочего места) при разработке проектного решения/при эксплуатации	*Объект исследования: мобильное приложение* *Область применения: спортивная деятельность, тренажерные залы* *Рабочая зона:* офис *Размеры помещения: 6*4 м* *Количество и наименование оборудования рабочей зоны: компьютер – 1 ед., мобильное устройство для тестирования приложения 1 ед.* *Рабочие процессы, связанные с объектом исследования, осуществляющиеся в рабочей зоне: проектирование и разработка программного обеспечения, контроль качества*
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Правовые и организационные вопросы обеспечения безопасности при разработке проектного решения: – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны.	1. Федеральный закон от 29 июля 2004 г. № 98-ФЗ «О коммерческой тайне» 2. Федеральный закон от 27 июля 2006 г. № 149-ФЗ «Об информации, информационных технологиях и о защите информации» 3. ГОСТ 12.1.005-88 Система стандартов безопасности труда. Общие санитарно-гигиенические требования к воздуху рабочей зоны (правовые нормы) 4. Приказ Минтруда России от 29 октября 2021 г. № 774н «Об утверждении общих требований к организации безопасного рабочего места»
2. Производственная безопасность при разработке проектного решения: – Анализ выявленных вредных и опасных производственных факторов	Вредные факторы: 1. Отсутствие или недостаток необходимого естественного освещения 2. Отсутствие или недостатки необходимого искусственного освещения 3. Повышенная пульсация светового потока 4. Нервно-психические перегрузки, связанные с напряженностью трудового процесса 5. Статические перегрузки, связанные с рабочей позой

	6. Длительное сосредоточенное наблюдение Опасные факторы: 1. Производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которого попадает работающий
3. Экологическая безопасность при разработке проектного решения	Воздействие на селитебную зону: не выявлено Воздействие на литосферу: отходы в следствие утилизации компьютера Воздействие на гидросферу: не выявлено Воздействие на атмосферу: не выявлено
4. Безопасность в чрезвычайных ситуациях при разработке проектного решения	Возможные ЧС: Пожар, наводнение, ураган, землетрясение Наиболее типичная ЧС: Пожар
Дата выдачи задания для раздела по линейному графику	**01.03.2022**

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Мезенцева Ирина Леонидовна	-		01.03.2022

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К82	Власов Никита Константинович		01.03.2022

Реферат

Выпускная квалификационная работа 95 с., 11 рис., 20 табл., 22 источников, 3 прил.

Ключевые слова: мобильное приложение, серверное приложение, Ktor, Kotlin, Android, спортивная активность, REST API

Объектом исследования является учет спортивной активности с помощью мобильного приложения

Цель работы – проектирование и разработка серверной части мобильного приложения для учета спортивной активности

В процессе выполнения работы проводился обзор современных веяний в построении архитектуры серверного приложения. Были выбраны технологии для создания серверного приложения.

В результате выполнения работы была спроектирована и разработана серверная часть приложения для учета спортивной активности

Область применения: спортивная деятельность

В будущем планируется реализация остальных функций приложения и размещение его в магазине приложений.

Обозначения и сокращения

REST API – способ создания взаимодействия клиентского и серверного приложения с помощью протокола HTTP.

SOLID – аббревиатура, каждая буква которой обозначает отдельный принцип проектирования в объектно-ориентированном программировании.

Clean Architecture – набор рекомендаций для построение архитектуры приложения. Основой рекомендаций является разделение программного кода на слои.

GitHub – веб-сервис для хостинга программных проектов, основанный на системе контроля версий git.

Heroku CLI – интерфейс командной строки, позволяющий создавать и управлять приложениями, размещенными на Heroku.

DAO – объекты позволяющие взаимодействовать с базой данных.

JDBC – стандарт взаимодействия java-приложений с базами данных.

Оглавление

Реферат 11

Обозначения и сокращения 12

Введение 16

1. Анализ предметной области 18

2. Выявление требований к приложению 20

3. Проектирование серверной части приложения 25

3.1. Проектирование сценариев использования 25

3.2. Проектирование базы данных 26

3.3. Порядок реализации функций 31

3.4. Аналитический обзор фреймворков для создания серверной части приложения. 32

3.5. Проектирование архитектуры приложения 33

3.6. Аналитический обзор решений для внедрения паттерна инъекции зависимостей 35

3.7. Аналитический обзор решений для развертки серверной части приложения 37

3.8. Аналитический обзор алгоритмов шифрования паролей 38

3.9. Используемые средства разработки 39

3.10. Результаты проектирования серверной части приложения 39

4. Разработка серверной части приложения 41

4.1. Создание базы данных PostgreSql 41

4.2. Реализация архитектуры приложения 42

4.3. Реализация авторизации в приложении 47

4.4. Реализация тестов в приложении 49

4.5. Реализация непрерывной интеграции и развертка приложения 50

4.6. Результат работы 51

5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение 53

5.1. Потенциальные потребители результатов исследования 53

5.2. Анализ конкурентных технических решений 55
5.3. SWOT-анализ 56
5.4. Планирование работ по научно-техническому исследованию 60
5.4.1. Структура работ в рамках научного исследования 60
5.4.2. Определение трудоемкости выполнения работ 61
5.4.3. Разработка графика проведения научного исследования 62
5.5. Бюджет научно-технического исследования (НТИ) 67
5.5.1. Расчет материальных затрат НТИ 67
5.5.2. Расчет затрат на специальное оборудование для научных работ 68
5.5.3. Основная заработная плата исполнителя темы 68
5.5.4. Расчет дополнительной заработной платы 72
5.5.5. Отчисления во внебюджетные фонды 73
5.5.6. Накладные расходы 74
5.5.7. Формирование бюджета затрат научно-исследовательского проекта 75
5.6. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования 76
5.7. Вывод по разделу 78
6. Социальная ответственность 79
6.1. Правовые и организационные вопросы обеспечения безопасности 79
6.2. Производственная безопасность 80
6.3.2. Нервно-психические перегрузки, связанные с напряженностью трудового процесса 83
6.3.3. Статические перегрузки, связанные с рабочей позой 83
6.3.4. Длительное сосредоточенное наблюдение 84
6.3.5. Производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которого попадает работающий 84
6.4. Экологическая безопасность 84

6.5. Безопасность в ЧС 85
6.5.1. Анализ возможных ЧС 85
6.5.2. Анализ наиболее вероятной ЧС 85
6.6. Вывод по разделу 86
Заключение 88
Список используемых источников 89
Приложение А 92
Приложение Б 93
Приложение В 94

Введение

В настоящее время, дневник тренировок — это неотъемлемая часть любого тренировочного процесса. С его помощью можно следить за результатами и прогрессировать. Дневник тренировок дает множество преимуществ. Во-первых, вы всегда знаете, что вам нужно делать и какая тренировка запланирована на сегодня. Во-вторых, можно отслеживать свои результаты и осознанно увеличивать нагрузки. В-третьих, постоянный прогресс. Так как вы знаете, какой объем тренировок был раньше, вы можете варьировать и менять их, отталкиваясь от прошлого опыта. Многие посещают фитнес зал с блокнотом и ручкой. Это неплохой вариант, но иметь приложение в смартфоне всё же гораздо удобнее.

Для простого и понятного отслеживания результатов и прогрессов было спроектировано мобильное приложение. Оценить свой прогресс и посмотреть результат каждой тренировки можно в любой момент. Необходимо лишь скачать приложение и разобраться в пользовательском интерфейсе.

Цель: Создать мобильное приложение под операционную систему Android, для ведения дневника тренировок.

Для достижения цели в данной работе выполняются следующие задачи:

1. проанализировать предметную область
2. выявить требования к приложению
3. спроектировать мобильное приложение
4. выбрать стек технологий для реализации мобильного приложения
5. спроектировать графический интерфейс мобильного приложения

6. реализовать мобильное приложение

1. Анализ предметной области

Предметная область: занятия спортом и физической культурой в тренажерном зале

В данной предметной области существует проблема учета спортивной активности, обычно учет ведется в виде дневника на бумажном носителе. В дневник записываются проведенные тренировки, обычно, каждая запись включает: время начала и завершения тренировки, список упражнений, рабочий вес для каждого из упражнений, количество повторений и подходов в каждом упражнении. Также, возможна запись заметок о тренировке. Так как записи ведутся на бумажном носителе, то в какой-то момент бумажных носителей станет несколько, поскольку в предыдущем может закончиться место для записей, бумажный носитель может прийти в негодность или спортсмен может попросту его потерять, также стоит отметить, что бумажный носитель может не оказаться рядом в нужный момент. Еще один немаловажный факт – это анализ записей, при использовании такого рода носителей он очень затруднен, так как обработка информации занимает достаточно большое количество времени.

На рисунке ниже представлена диаграмма Fish Bone, в которой рассматриваются причины неудобства анализа результатов тренировки.

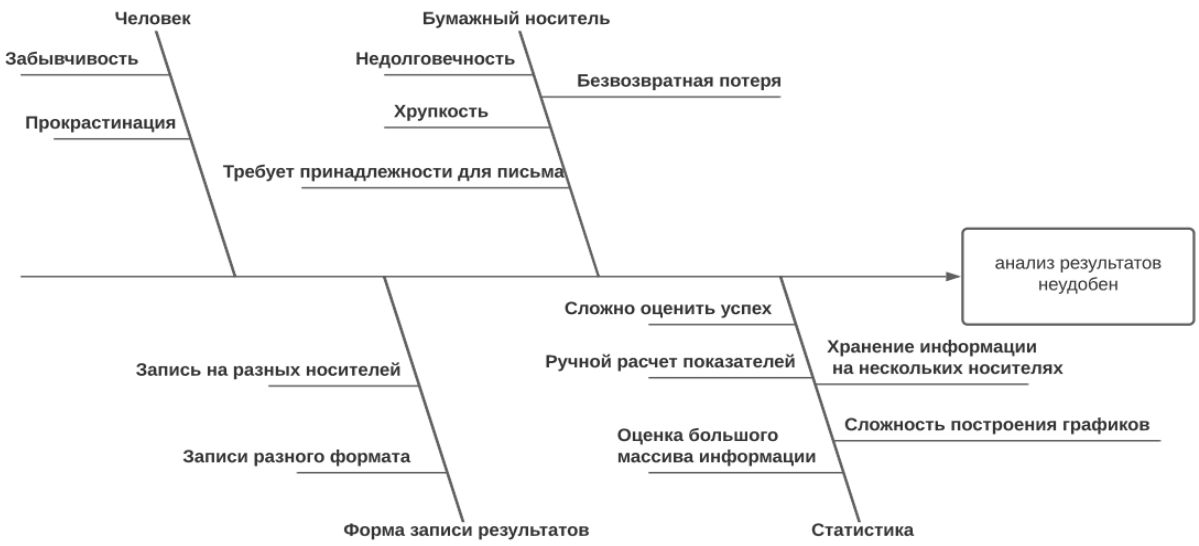


Рисунок 1 – Диаграмма Fish Bone. Анализ результатов

На следующем рисунке представлена диаграмма Fish Bone, в которой выделены причины, из-за которых сложно поделиться успехами с другими людьми.

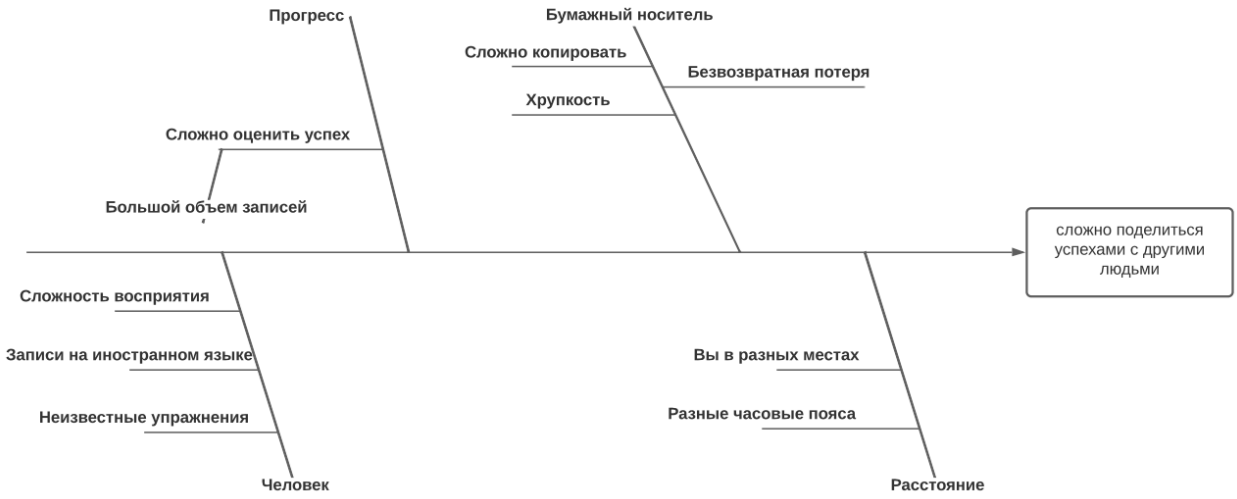


Рисунок 2 – Диаграмма Fish Bone. Поделиться

2. Выявление требований к приложению

Для выявления требований к мобильному приложению следует провести анализ конкурентных решений, которые представлены на рынке. На момент написания данной работы на рынке представлено очень много приложений для отслеживания спортивной активности, для анализа были выделены следующие решения:

- «GymRun – Журнал тренировок и фитнес-трекер»
- «GymKeeper – Дневник тренировок»

Данные решения являются одними из самых популярных в российском сегменте рынка приложений для отслеживания спортивной активности, например у «GymRun» – 1 миллион установок на устройства.

Анализ данных решений в совокупности с анализом предметной области, проведенным ранее, позволит выявить требования для приложения, проектируемого и разрабатываемого в рамках данной выпускной квалификационно работы. В анализе важно учесть недостатки и недоработки существующих решений, сделать это можно с помощью прямого их использования и отзывов пользователей, которые предоставляет магазин приложений для устройств под управлением операционной системы Android – «Google Play».

Анализ решения «GymRun – Журнал тренировок и фитнес-трекер»

В решении представлены база упражнений по группам мышц, возможность добавлять свои тренировки и упражнения, статистика по прогрессу в тренировочных программах, возможность записать замеры частей тела спортсмена. Также в решении присутствуют разнообразные калькуляторы для расчета веса штанги и веса для 1 повторного максимума в упражнении, секундомер. Решение поддерживает работу с умными часами.

Во время выполнения упражнения, когда экран мобильного устройства заблокирован, приложение отображает виджет, который содержит название выполняемого упражнения, текущий рабочий вес и кнопки для ввода количества повторений и завершения подхода.

Пример виджета представлен на рисунке ниже.

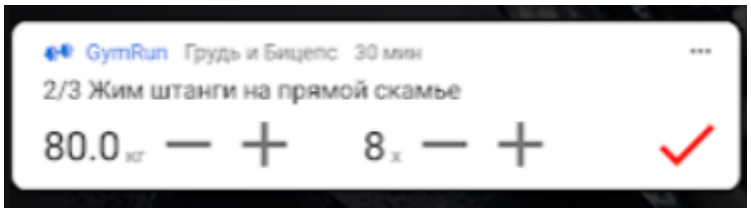


Рисунок 3 – Пример виджета

В ходе анализа данного решения также были выявлены следующие недостатки:

- есть проблемы синхронизации программ тренировок с умными часами;
- нет возможности удалять ранее добавленные упражнения;
- не предусмотрена возможность создания мышечных групп;
- нельзя заменять упражнения в процессе тренировки.

Анализ решения «GymKeeper – Дневник тренировок»

В решении есть база упражнений и бесплатные программы тренировок, калькулятор штанги, экран с личными рекордами и наградами в виде медалей, возможность фиксировать замеры тела, возможность вести несколько дневников тренировок, возможность добавлять свои упражнения. Стоит отметить, что создается резервная копия прогресса пользователя и сопутствующей информации на серверной часть приложения.

Решение отображает статистику тренировок в нескольких вариантах, а именно:

- общая статистика (количество тренировок, сколько выполнено упражнений и подходов);
- прогресс по отдельным группам мышц за определенный период;
- любимые упражнения в виде круговой диаграммы.

Минусы данного решения:

- приложение распространяется по подписке;
- подписка очень дорогая (много отзывов пользователей на эту тему);
- представлено всего десять бесплатных программ тренировок;
- нельзя добавлять свои группы мышц;
- дата выполнения упражнения хранится не более 15 дней;

Основываясь на ранее проведенном анализе предметной области и конкурентных решений, были выявлены и составлены требования к мобильному приложению, разрабатываемому в данной работе.

1. Функциональные требования

1.1. Система должна позволять неавторизованному пользователю осуществлять авторизацию по e-mail и паролю.

1.2. Система должна производить автоматическую авторизацию при включении приложения при условии недавнего входа пользователя (не более 14 дней).

1.3. Система должна отображать пользователю его профиль на вкладке «Мой профиль».

1.4. Система предоставляет интерфейс для редактирования профиля на вкладке «Мой профиль» для всех полей.

1.5. Система отображает проведенные тренировки в виде календаря или списка.

1.6. Система позволяет просматривать упражнения в каждой из тренировок.

1.7. Система позволяет сохранять прогресс выполнения тренировок, а именно количество повторений, подходов, рабочий вес, время отдыха между подходами и упражнениями.

1.8. Система позволяет пользователю планировать состав тренировки, а именно количество подходов, повторений, рабочий вес, время отдыха между подходами и упражнениями, тип упражнений.

1.9. Система позволяет пользователю создавать тренировки.

1.10. Система позволяет пользователю называть тренировки.

1.11. Система позволяет прикреплять комментарии и фотографии к тренировке.

1.12. Система позволяет пользователю просматривать информацию об упражнениях, а именно название, описание, задействованные группы мышц, технику выполнения как в текстовом формате, так и в видео.

1.13. Система позволяет пользователю заменять упражнения во время тренировки

1.14.Система позволяет пользователю менять упражнения местами во время тренировки.

1.15.Система предоставляет список готовых программ тренировок пользователю.

1.16.Система предоставляет пользователю базу упражнений.

1.17.Система позволяет пользователю добавлять свои упражнения.

1.18.Система позволяет пользователю создавать свои программы тренировок

1.19.Система напоминает пользователю о запланированной тренировке

1.20.Система позволяет пользователю сохранить замеры своего тела.

1.21.Система позволяет просматривать таблицу лидеров по упражнениям

1.22.Система позволяет пользователю подписаться на другого пользователя и следить за его тренировками.

1.23. Система позволяет пользователю делиться тренировками.

1.24.Система позволяет пользователю фильтровать упражнения по задействованных мышц, использованию тренажера, сложности и т.п.

2. Нефункциональные требования:

2.1. Система представлена мобильным приложением на базе операционной системы Android.

2.2. Минимальной версией Android API является 24.

2.3. Мобильное приложение разрабатывается на языке Kotlin.

2.4. Серверная часть приложения разрабатывается на базе фреймворка Ktor

2.5. Система должна поддерживать 200 пользователей одновременно.

2.6. Система предоставляет OAuth2-авторизацию.

2.7. Система должна делать резервную копию записей пользователя на сервере.

2.8. Система должна хранить базу упражнений.

Данные требования в дальнейшем помогут в проектировании приложения и будут опорой для реализации его функциональности.

3. Проектирование серверной части приложения

Проект разделен на две части: клиентское приложение и серверное приложение. В данном разделе представлены составляющие процесса проектирования серверной части приложения.

3.1. Проектирование сценариев использования

Исходя из вышеперечисленных требований было выделено две роли пользователя: авторизованный пользователь и неавторизованный пользователь.

Таблица 1 – Роли и их описания

Роль	Описание
Неавторизованный пользователь	Имеет только возможности зарегистрироваться или авторизоваться.
Авторизованный пользователь	Имеет доступ ко всему функционалу приложения.

Варианты использования приложения пользователем в роли «Неавторизованный пользователь» представлены на рисунке ниже.

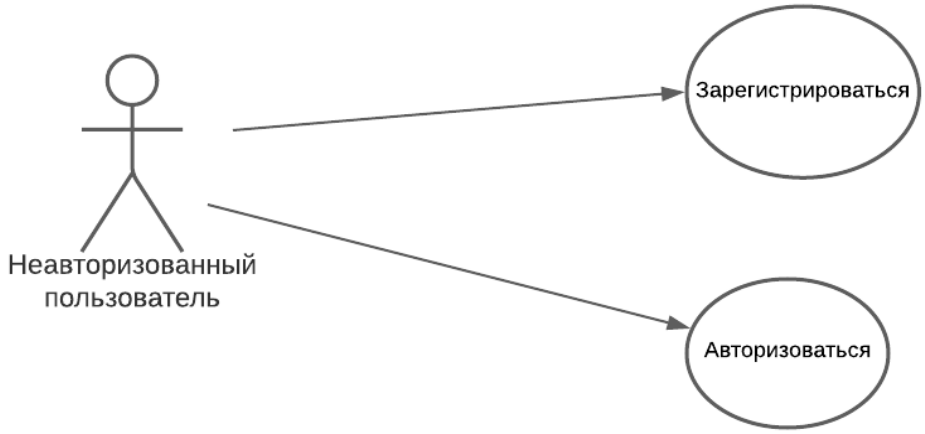


Рисунок 4 – Диаграмма вариантов использования для неавторизованного пользователя.

Варианты использования для пользователя в роли «авторизованный пользователь» представлены в приложении Б.

Из-за того, что разным ролям пользователя предоставляется различный функционал, необходимо было сделать распознавание роли пользователя. В данном случае достаточно лишь проверять авторизован ли пользователь. Подробнее процесс аутентификации будет описан далее.

3.2. Проектирование базы данных

В соответствии с требованиями и выявленными сценариями использования была спроектирована структура базы данных.

Сущности базы данных:

С1. Пользователь

1. Идентификатор
2. Email
3. Имя пользователя
4. Пароль

Сущность, необходимая для хранения основных данных пользователя.

С2. Показатели пользователя

1. Идентификатор пользователя
2. Дата записи
3. Вес
4. Рост

Сущность, хранящая физические показатели пользователя. Составной ключ необходим для того, чтобы пользователь мог делать множество замеров в разное время.

С3. Упражнение

1. Идентификатор
2. Название
3. Описание

Сущность, хранящая информацию об упражнениях.

С4. Материалы к упражнению

1. Идентификатор упражнения
2. URI материала

Сущность хранит различные материалы (видео, фото) об упражнениях. Составной ключ в этой сущности необходим из-за того, что может быть несколько материалов связанных с одним упражнением

С5. Отзыв к упражнению

1. Идентификатор упражнения

2. Идентификатор пользователя
3. Оценка
4. Текст отзыва

Сущность хранит отзывы пользователей на упражнения.

С6. Тренировка пользователя

1. Идентификатор
2. Идентификатор пользователя
3. Время тренировки
4. Название
5. Дата

Сущность, хранящая информацию о тренировках пользователя.

С7. Упражнение в тренировке с весом и подходами

1. Идентификатор тренировки
2. Идентификатор упражнения
3. Номер подхода
4. Вес
5. Повторений
6. Фактический вес
7. Фактическое количество повторений

Сущность, хранящая информацию о выполненных упражнениях в тренировке. Для типа упражнений с весами и подходами.

С8. Программа тренировок

1. Идентификатор
2. Название
3. Описание

Сущность, хранящая основную информацию о программах тренировок.

С9. Тренировка в программе тренировок

1. Идентификатор
2. Идентификатор программы
3. Название
4. URI изображения
5. Описание

Сущность, хранящая информацию о тренировках, входящих в программу тренировок.

С10. Упражнение в тренировке из программы с весом и подходами

1. Идентификатор тренировки
2. Идентификатор упражнения
3. Номер подхода
4. Вес
5. Повторений

Сущность, хранящая информацию об упражнениях в программе тренировок. Для типа упражнений с весом и подходами.

С11. Отзыв к программе

1. Идентификатор упражнения
2. Идентификатор пользователя
3. Оценка
4. Текст отзыва

Сущность, хранящая отзывы пользователей к программам тренировок.

С12. Программы, выбранные пользователем

1. Идентификатор пользователя
2. Идентификатор программы
3. Дата выбора

Сущность, хранящая информацию о программах, выбранных пользователями.

С13. Группы мышц

1. Идентификатор
2. Название

Сущность, хранящая возможные группы мышц.

С14. Задействованные в упражнении группы мышц

1. Идентификатор группы
2. Идентификатор упражнения

Сущность, хранящая информацию о группах мышц, задействованных в упражнении.

С15. Подписки пользователей

1. Идентификатор Подписчика
2. Идентификатор Пользователя

Сущность, хранящая информацию о подписках пользователей друг на друга.

После выявления и нормализации сущностей была сформирована логическая модель базы данных, на основе которой в дальнейшем была создана физическая модель. ER-диаграмма получившейся структуры представлена в приложении Б.

3.3. Порядок реализации функций

В рамках проектирования был определен порядок реализации функций приложения с помощью с помощью метода «MosCoW».

Заглавные буквы в названии метода обозначают:

- Must – должны сделать это;
- Should – должны сделать, если возможно;
- Could – можно сделать, если будет время;
- Wouldn’t – хотелось бы сделать, но не будет достаточно времени.

Приоритет задач представлен ниже.

- Must:
  - Просмотр упражнений
  - Просмотр программ тренировок

- Создавать записи о тренировке, включающую список выполненных упражнений
- Отображать инфографику, используя данные из проведенных тренировок

- Should:
  - Создавать свои упражнения
  - Создавать свои программы тренировок
  - Вносить замеры тела спортсмена (обхват частей тела, вес, рост)
- Could:
  - Прикреплять комментарии к записи
  - Отслеживать тренировки других пользователей
- Wouldn’t:
  - Прикреплять фотографии к записи

3.4. Аналитический обзор фреймворков для создания серверной части приложения.

На сегодняшний день существует множество различных решений для ускорения разработки серверных приложений. Отличаются эти решения используемыми языками программирования, архитектурой, гибкостью и сложностью освоения. К примеру:

- Spring boot – один из самых популярных java веб-фреймворков;
- Django REST Framework – легковесный python веб-фреймворк;
- Laravel – php-фреймворк, основанный на более «тяжеловесном» фрейморке Symfony;
- Express.js – веб-фреймворк для Node.js;

- Ktor – гибкий и легковесный фреймворк построенный на Kotlin и корутинах.

Результатом рассмотрения представленных вариантов был выбран фреймворк Ktor. Так как мобильное приложение под android разрабатывается на языке Kotlin, одной из главных причин такого выбора стало использования одинакового языка в разных частях приложения. Благодаря этому разработчик без особых проблем может переключатся и участвовать в разработке как серверной, так и клиентской части.

Также важными аспектами фреймворка являются легковесность и отсутствие жесткой привязки к какой-либо архитектуре. Они являются как плюсами, так и минусами. С одной стороны, благодаря этому есть свобода в реализации и низкий порог входа, с другой же стороны в фреймворке нет различных готовых вещей, которые упрощают разработку, как пример панель администратора в Django REST Framework, и из-за отсутствия опыта архитектура приложения может получиться проблемной.

3.5. Проектирование архитектуры приложения

Для взаимодействия клиента с сервером был выбран стиль REST API, исходя из этого была спроектирована архитектура. Для понимания того, как устроена архитектура создаваемого серверного приложения, была построена диаграмма компонентов приложения, представленная на рисунке ниже.

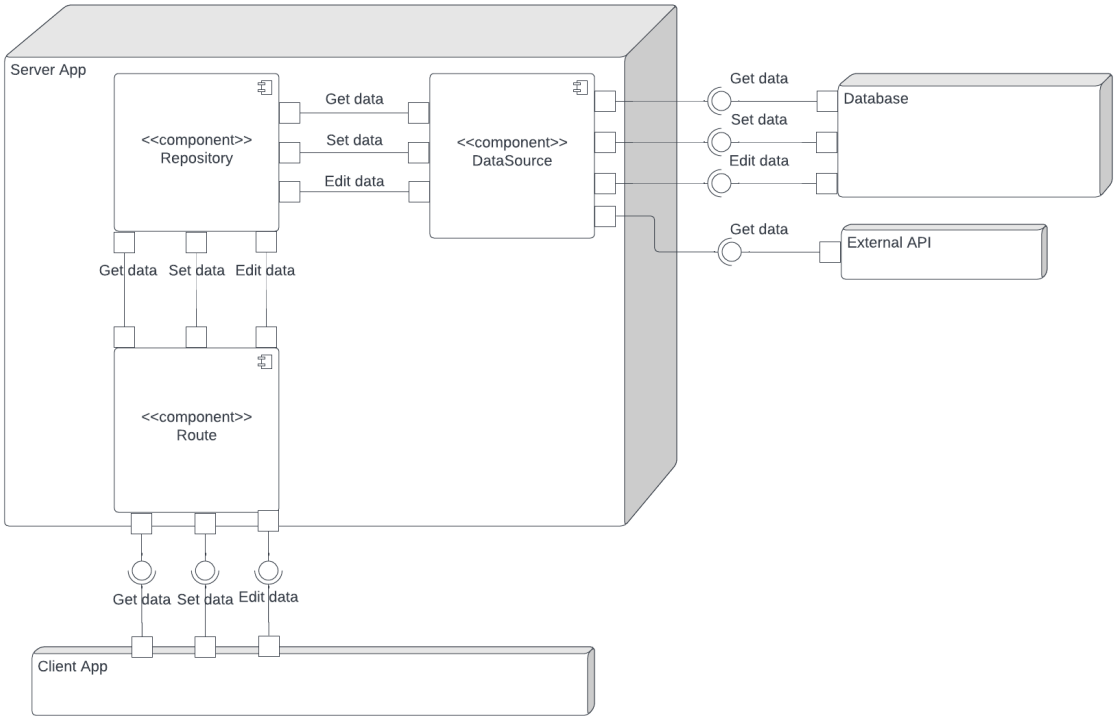


Рисунок 5 – Диаграмма компонентов приложения

Архитектура приложения основана на принципах «Clean architecture» и наборе принципах SOLID.

Рассмотрим представленную архитектуру. Код программы разделен на три ключевых слоя: data, domain и routes. Компоненты, представленные на диаграмме, взаимодействуют между собой с помощью похожих запросов, каждый запрос отражает конкретное намерение для взаимодействия с данными. Хоть запросы и похожи, сильно отличается внутренняя логика и ответственность каждого компонента.

Для описания предметной области используются принципы DDD (domain driven design), таким образом код поделен на отдельные бизнес области, которые отделены друг от друга структурой пакетов.

Компонент DataSource отвечает за взаимодействие с источником данных и имеет всю необходимую для этого логику. DataSource-компоненты являются ключевыми звеньями слоя data.

Компонент Repository необходим для отделения бизнес-логики от деталей реализации работы с данными. Repository-компонент может объединять в себе несколько DataSource-компнентов для реализации необходимой бизнес-логики. Repository-компоненты являются ключевыми звеньями слоя Domain.

Компонент Route за один из маршрутов REST API. Компонент обращается за данными в Repository и после передает их в ответе на HTTP-запрос.

Помимо представленных на диаграмме компонентов в структуре программы могут быть различные вспомогательные классы для конвертации данных, классы сущности и так далее. Но вся основная логика приложения находится в описанных компонентах.

3.6. Аналитический обзор решений для внедрения паттерна инъекции зависимостей

Одним из SOLID принципов является принцип инверсии зависимостей (DIP – Dependency Inversion Principle). Смысл этого принципа заключается в том, что код высокоуровневых слоев не должен зависеть от реализации низкоуровневых слоёв, он должен зависеть от абстракций. Применяя этот принцип, можно с легкостью подменять одну реализацию на другую, не боясь, что что-то может сломаться.

Для реализации этого принципа можно использовать паттерн инъекции зависимостей. Сутью принципа является то, что зависимый класс знает лишь об абстракциях и работает только с ними, не создавая объекты реализаций. Ответственность за создание и внедрение конкретных зависимостей на себя берёт внешний механизм. Чаще всего в роли этого механизма выступает специальный DI (dependency injection) фреймворк. Ниже представлен рисунок,

как может выглядеть дерево зависимостей без использования инъекции зависимостей и с использованием инъекции зависимостей.

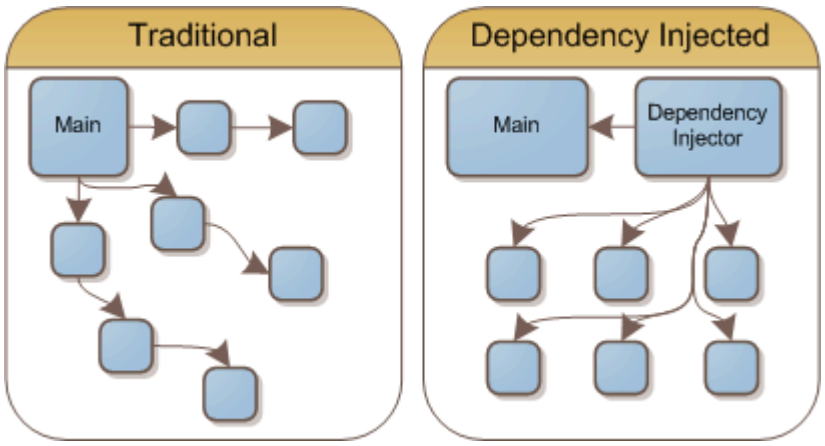


Рисунок 6 – Сравнение деревьев зависимостей

Кроме DI фреймворков существуют также SL (service locator) фреймворки. В целом они выполняют похожие функции, отличия часто лишь в функциональности, разнице реализации и сложности применения. В сообществе разработчиков ведётся множество споров о том какой фреймворк чем именно является. Для упрощения, в данном случае, все представленные для сравнения фреймворки будут считаться DI фреймворками. Также важно отметить, что на официальных сайтах этих фреймворков они именуются именно как DI фреймворки.

Для рассмотрения было выбраны три фреймворка:

- Dagger 2;
- Kodein;
- Koin.

Dagger 2, хоть и является мощным инструментом, был отвергнут почти сразу из-за того, что не имеет специальной поддержки для фреймворка Ktor. В оставшихся вариантах эта поддержка присутствует. Также Dagger 2 написан

на Java и использует аннотации для внедрения зависимостей, из-за чего его не так удобно использовать в Kotlin проекте.

Koin и Kodein довольно похожи. Оба имеют полную поддержку языка Kotlin, из-за чего их применение очень удобно в Kotlin проекте. Kodein более гибок и имеет больший функционал за счёт чего больше подходит для крупных проектов, но при этом внедрить и использовать его в проекте сложнее. Также Kodein имеет меньшую популярность, чем Koin и появился позже него, из-за чего найти решения возникающих с ним проблем может быть намного сложнее.

По итогам анализа в качестве DI фреймворка для проекта был выбран Koin.

3.7. Аналитический обзор решений для развертки серверной части приложения

В настоящее время для упрощения и ускорения разработки и поставки нового функционала пользователям можно использовать различные облачные сервисы, вместо покупки и самостоятельной настройки выделенного сервера.

Для рассмотрения были выбраны следующие сервисы:

- Azure;
- Google Cloud Platform;
- AWS;
- Heroku.

Сервисы Azure, Google Cloud Platforms и AWS имеют существенно большее количество возможностей по сравнению с Heroku. Google Cloud Platforms имеет хорошие и удобные возможности для интеграции аналитики в мобильные приложения. Azure имеет очень удобную документацию и

бесплатные курсы обучения на русском языке, что удобно. AWS является надежным и зарекомендовавшим себя решением, имеющем различные дополнительные возможности, упрощающие размещение приложения в облаке.

Но не смотря на всё это выбор пал на Heroku, по причине того, что эта платформа имеет привлекательную ценовую политику: есть возможность бесплатного использования минимального тарифа без ограничений по времени. Сервис не имеет как-либо существенных недостатков, крайне удобен и прост в использовании. Для существующих целей Heroku подходит больше остальных вариантов, хотя не исключено, что в будущем, при развитии приложения, платформа для развертки будет изменена.

3.8. Аналитический обзор алгоритмов шифрования паролей

Авторизацию пользователя в приложение было решено реализовывать с помощью связки адреса электронной почты и пароля.

Хранить пароли в базе данных в «чистом виде» довольно рискованно, поэтому распространенной практикой является хранение паролей в зашифрованном виде. Существует множество криптографических алгоритмов, позволяющих это делать.

На тему криптографии существует множество споров в сообществе программистов. В ходе анализа было рассмотрено множество различных типов алгоритмов. Главными критериями для выбора стало:

- Актуальность;
- Реализация на языке Kotlin или Java;
- Умеренные требования к вычислительным мощностям.

Итогом стал выбор алгоритма Bcrypt. Для более подробного рассмотрения алгоритмов нужно довольно много времени и возможно неплохим вариантом было бы проконсультироваться на этот счёт со специалистами в сфере информационной безопасности. При дальнейшем развитии проекта алгоритм может быть изменен, также, для улучшения уровня безопасности, планируется добавления способа двухфакторной аутентификации.

3.9. Используемые средства разработки

В ходе реализации данного проекта используется список таких средств разработки:

Intellij IDEA – среда разработки, созданная компанией JetBrains. Имеет огромный функционал с возможностью расширения с помощью плагинов. Является одним из лучших решений для разработки на Kotlin и имеет поддержку фреймворка Ktor;

Postman – инструмент для тестирования REST API;

PgAdmin – платформа для создания и проектирования баз данных PostgreSql;

Lucidchart – веб-приложения для работы с различными видами диаграммами.

Heroku CLI – приложение командной строки, позволяющее создавать и администрировать проекты на платформе Heroku.

3.10. Результаты проектирования серверной части приложения

На основе результатов проектирования был выбран следующий стек технологий:

1. Ktor – гибкий и легковесный фреймворк построенный на Kotlin и корутинах;
2. Gson – библиотека, необходимая для конвертации JSON;
3. JUnit5 – библиотека для модульного тестирования;
4. Mockk – библиотека для подмены реализации классов заглушками с целью тестирования;
5. Koin – фреймворк внедрения зависимостей, необходим для реализации паттерна проектирования Dependency Injection;
6. Exposed – библиотека, позволяющая взаимодействовать с базой данных в ORM стиле. Написана на Kotlin и позволяет использовать все преимущества этого языка во время разработки;
7. PostgreSQL – свободная объектно-реляционная система управления базами данных. За счёт способа распространения считается одним из самых гибких инструментов для хранения данных в реляционном стиле;
8. Ktor JWT Auth – расширение для Ktor, позволяющее легко осуществлять авторизацию с помощью стандарта JSON Web Token;
9. Bcrypt – адаптивная криптографическая хеш-функция формирования ключа, используемая для защищенного хранения паролей;
10. Ktor client – HTTP-клиент, написанный на языке Kotlin.

4. Разработка серверной части приложения

В этом разделе будут представлены основные этапы разработки серверной части приложения.

4.1. Создание базы данных PostgreSql

В качестве СУБД для проекта была выбрана PostgreSql. Удаленный сервер базы данных был настроен с помощью ресурсов Heroku, о котором подробнее написано в дальнейших разделах.

Для взаимодействия с базой из приложения используется стандарт JDBC и библиотека Exposed. Ниже представлена автоматически сгенерированная, с помощью pgAdmin, диаграмма физической модели.

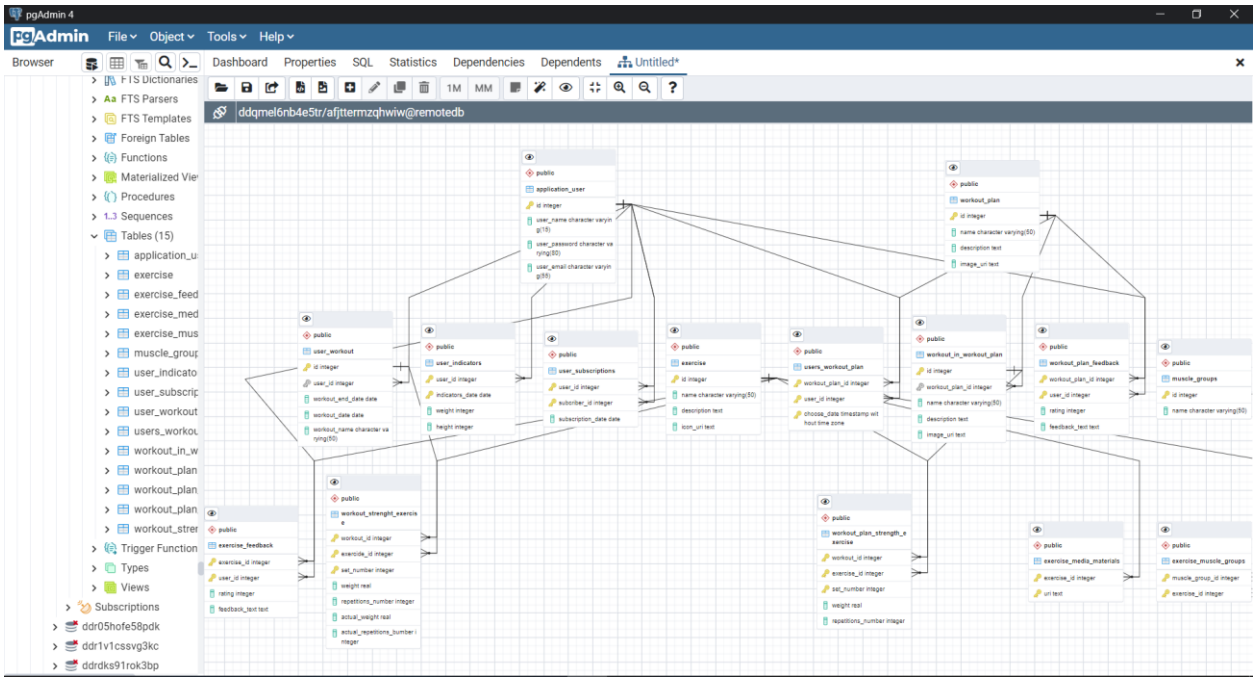


Рисунок 7 – Физическая модель базы данных.

Пример взаимодействия с базой данных будет представлен в следующем разделе.

4.2. Реализация архитектуры приложения

Архитектура приложения была реализована согласно составленной ранее диаграммой компонентов. На рисунках ниже представлена структура пакетов и классов приложения.

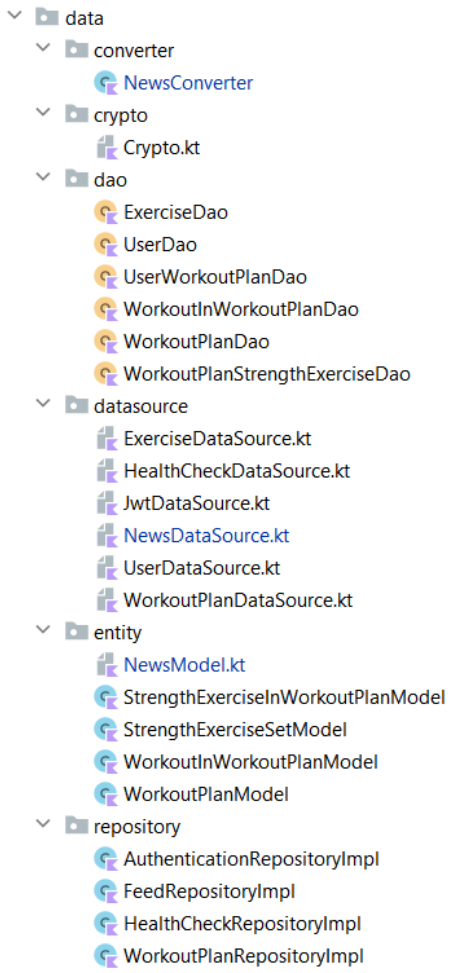


Рисунок 8 – структура слоя data

На слое data ключевыми являются DataSource классы, они отвечают за получения данных из различных источников, в данном случае из базы данных и стороннего REST API.

Также здесь присутствуют DAO объекты, каждый из которых отражает определенную таблицу из базы данных. Они необходимы для получения

данных из определенной таблицы с помощью библиотеки Exposed. На листинге ниже приведен пример получения списка упражнений.

```
interface ExerciseDataSource {

    fun getList(): List<Exercise>
}

class ExerciseDataSourceImpl : ExerciseDataSource {

    override fun getList(): List<Exercise> = transaction {
        ExerciseDao.selectAll().map { row ->
            Exercise(
                row[ExerciseDao.id],
                row[ExerciseDao.name],
                row[ExerciseDao.description],
                row[ExerciseDao.iconUri] ?: ""
            )
        }
    }
}
```

Также для получения спортивных новостей было использовано открытое REST API, запросы к которому производятся с помощью Ktor Client. На листинге ниже представлен DataSource класс, который отвечает за получения списка новостей.

```
interface NewsDataSource {

    suspend fun getList(): NewsResponse
}

class NewsDataSourceImpl(
    private val client: HttpClient
) : NewsDataSource {

    override suspend fun getList(): NewsResponse =
        client.get<NewsResponse> {
            url.encodedPath = "/top-headlines"
            parameter("country", "ru")
            parameter("category", "sport")
        }
}
```

Также на данном слое хранятся классы реализации абстракций слоя бизнес-логики Repository. Ниже представлен пример класса, который отвечает за получение данных из DataSource классов для передачи информации для главного экрана мобильного приложения.

```
class FeedRepositoryImpl(
    private val newsDataSource: NewsDataSource,
```

```
    private val exerciseDataSource: ExerciseDataSource,
    private val workoutPlanDataSource: WorkoutPlanDataSource,
    private val newsConverter: NewsConverter,
): FeedRepository {

    override suspend fun get(count: Int): Feed {
        val workoutPlans = workoutPlanDataSource.getList().take(count)
        val exercises = exerciseDataSource.getList().take(count)
        val news = try {
            newsConverter.convert(newsDataSource.getList()).take(count)
        } catch (e: Exception) {
            emptyList()
        }

        return Feed(
            workoutPlans,
            exercises,
            news
        )
    }
}
```

Помимо этого, на этом слое хранятся различные сущности, конвертеры и класс-абстракция над библиотекой реализующей алгоритм шифрования Bcrypt.

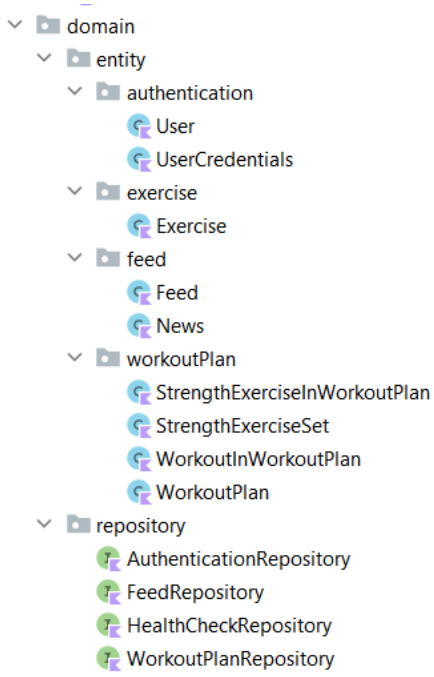


Рисунок 9 – структура слоя domain.

На данном слое хранятся различные бизнес-сущности и интерфейсы Repository. Бизнес-сущности были обозначены на этапе проектирования для совместного их использования как в серверном, так и в клиентском приложении, это сильно упростило взаимодействие этих частей проекта.

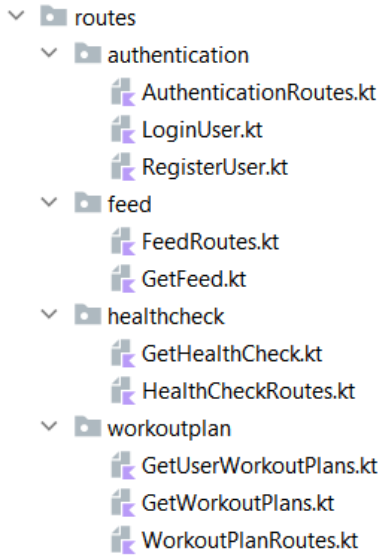


Рисунок 10 – структура слоя routes

На слое routes находится реализация различных маршрутов REST API. В файлах Routes объединены функции-запросов для различных областей. К примеру, в WorkoutPlanRoutes содержаться запросы для получения всех планов тренировок и для получения планов тренировок, выбранных определенным пользователем. Ниже представлен листинг функции с запросов для получения данных для главного экрана мобильного приложения.

```
fun Route.getFeed() {
    val feedRepository: FeedRepository by inject()

    route("/feed") {
        authenticate("auth-jwt") {
            get {
                try {
                    val count =
call.request.queryParameters["count"]?.toInt()
                    val feed = count?.let {
                        feedRepository.get(count)
                    } ?: feedRepository.get()
                    call.respond(feed)
                } catch (e: PSQLException) {
                    call.respondText(DATABASE_UNAVAILABLE_MESSAGE +
e.message, status = HttpStatusCode.InternalServerError)
                } catch (e: IllegalArgumentException) {
                    call.respondText(ILLEGAL_ARGUMENT_MESSAGE + e.message,
status = HttpStatusCode.BadRequest)
                } catch (e: Throwable) {
                    call.respondText(INTERNAL_SERVER_ERROR_MESSAGE +
e.message, status = HttpStatusCode.InternalServerError)
                }
            }
        }
```

```
    }
}
```

В данной функции можно увидеть объявление маршрута, в данном случае /feed, то что информацию из данного запроса может получить только авторизованный пользователь и также видно, что сама информация берется с помощью обращения к классу Repository, который помещается в функцию с помощью фреймворка внедрения зависимостей Koin. Пример ответа на данный запрос есть в приложении В.

Также в стуктуре проекта присутствует пакет с модулями фреймворка внедрения завимостей. Ниже представлен листинг с модулем, отвечающем за зависимости для получения списка новостей.

```
private const val BASE_URL = "https://newsapi.org/v2"

val newsModule = module {
    val client = HttpClient(CIO) {
        install(JsonFeature) {
            serializer = GsonSerializer()
        }

        install(Logging) {
            logger = Logger.DEFAULT
            level = LogLevel.HEADERS
        }

        defaultRequest {
            url.takeFrom(URLBuilder().takeFrom(BASE_URL).apply {
                encodedPath += url.encodedPath
            })
            parameter("apiKey",
envConfig.property("news.apiKey").getString())
        }
    }

    single<NewsDataSource> { NewsDataSourceImpl(client) }
    single { NewsConverter() }
}
```

В данном модуле создается объект HTTP-клиента и класс-конвертер, которые будут помещаться в DataSource при его создании. Оба класса объявлены с помощью оператора single, это означает, что экземпляр класса будет создан в единственном экземпляре и не будет пересоздаваться каждый раз, когда он будет нужен.

Помимо представленных слоев в проекте также присутствует слой util, в котором используются различные вспомогательные классы и глобальные функции, которые используются во всём проекте. Также в проекте есть различные файлы конфигураций: файл конфигурации расширений, файл объявления gradle-зависимостей приложения и файл с объявлением используемых переменных среды. Кроме этого есть отельный пакет с тестами, о которых будет рассказано подробнее в другом разделе.

4.3. Реализация авторизации в приложении

Для авторизации в приложение было решено использовать связку пароля и адреса электронной почты и токен доступа стандарта JWT.

При первичной авторизации от пользователя передаются адрес электронной почты и пароль, которые проходят проверку соответствия с данными, хранящимися в базе данных. В случае успешного прохождения всех проверок, клиентскому приложению в ответе сервера приходит сгенерированный токен, который используется в дальнейших запросах для проверки авторизации. В листинге кода ниже представлен класс отвечающий за создание токена.

```
interface JwtDataSource {

    fun generateToken(userEmail: String, userId: Int): String
}

class JwtDataSourceImpl : JwtDataSource {

    private companion object {

        const val EXPIRED_HOURS = 12L
        const val EMAIL = "email"
        const val ID = "id"
    }

    private val expiresTime = Date(System.currentTimeMillis() +
TimeUnit.HOURS.toMillis(EXPIRED_HOURS))
    private val secret = envConfig.property("jwt.secret").getString()
    private val issuer = envConfig.property("jwt.issuer").getString()
    private val audience = envConfig.property("jwt.audience").getString()
    private val algorithm = Algorithm.HMAC256(secret)

    override fun generateToken(userEmail: String, userId: Int): String =
```

```
        JWT.create()
            .withAudience(audience)
            .withIssuer(issuer)
            .withClaim(EMAIL, userEmail)
            .withClaim(ID, userId.toString())
            .withExpiresAt(expiresTime)
            .sign(algorithm)
}
```

В коде видно, что в токене помимо необходимых частей, которые отвечают за распознавание того, что токен был создан именно этим приложением и секретного набора символов, который также используется при генерации токена, зашифрованы ещё данные об адресе электронной почты и идентификатора пользователя. Это необходимо для распознавания того, от какого именно пользователя пришёл запрос на сервер.

Все пароли пользователя хранятся в базе данных в зашифрованном виде. Ниже представлен листинг класс-абстракции, в который отвечает за это.

```
interface Crypto {

    fun hash(password: String): String
    fun verify(password: String, hash: String): Boolean
}

class BCryptoImpl : Crypto {

    private companion object {
        const val BCRYPT_COST = 6
    }

    private val salt = envConfig.property("crypt.cryptSalt").getString()

    override fun hash(password: String): String =
        BCrypt.withDefaults().hash(
            BCRYPT_COST,
            salt.toByteArray(Charsets.UTF_8),
            password.toByteArray(Charsets.UTF_8)
        ).toString(Charsets.UTF_8)

    override fun verify(password: String, hash: String): Boolean =
        BCrypt.verifyer()
            .verify(password.toByteArray(Charsets.UTF_8),
hash.toByteArray(Charsets.UTF_8))
            .verified
}
```

У данного класса есть два метода: метод, который отвечает за шифрование пароля и метод, который отвечает за сверку чистого и

зашифрованного пароля, это необходимо для авторизации пользователя, когда от клиентского приложения передается пароль в «чистом» виде.

4.4. Реализация тестов в приложении

В серверном приложении было реализовано два вида тестов: модульные тесты и тесты маршрутов REST API.

Модульные тесты отвечают за проверку корректности поведения каждого отдельного метода в приложении. Ниже представлен листинг кода тестирующего методы, отвечающие за регистрацию и авторизацию.

```
class AuthenticationRepositoryImplTest {

    private val userDataSource = mockk<UserDataSource>()
    private val jwtDataSource = mockk<JwtDataSource>()
    private val authenticationRepositoryImpl =
AuthenticationRepositoryImpl(userDataSource, jwtDataSource)

    @Test
    fun `register EXPECT add user with data source`() {
        val user = User("test", "test", "test")
        val userId = 1

        every { userDataSource.add(user) } returns userId

        authenticationRepositoryImpl.register(user)

        verify { userDataSource.add(user) }
    }

    @Test
    fun `login EXPECT jwt token`() {
        val userCredentials = UserCredentials("test", "test")
        val userId = 1
        val expected = "jwt token"

        every { userDataSource.getId(userCredentials) } returns userId
        every { jwtDataSource.generateToken(userCredentials.email, userId) }
returns expected

        val actual = authenticationRepositoryImpl.login(userCredentials)

        assertEquals(expected, actual)
    }
}
```

Тесты маршрутов отвечают за проверку корректности работы запросов при определенных заданных параметра. Особенность этих тестов заключается в том, что для них необходим запуск приложения. Из-за этого тесты

используют тестовую версию конфигурационного файла расширений, ведь он необходим для запуска приложения. Ниже представлен пример теста, проверяющего запрос авторизации при корректных данных.

```
class LoginUserTest : BaseRoutesTest {

    private val authenticationRepository = mockk<AuthenticationRepository>()

    private val module = module {
        single { authenticationRepository }
    }

    override val koinModules = listOf(module)

    private val gson = Gson()

    private val userCredentials = UserCredentials("test", "test")
    private val jwtToken = "jwt token"
    private val requestBody = gson.toJson(userCredentials)
    private val responseBody = gson.toJson(mapOf("token" to jwtToken))

    @Test
    fun `route login with valid credentials EXPECT response token`() =
withBaseTestApplication {
        with(handleRequest(HttpMethod.Post, "/login") {
            every { authenticationRepository.login(userCredentials) } returns
jwtToken

            addHeader(HttpHeaders.ContentType,
ContentType.Application.Json.toString())
            setBody(requestBody)
        }) {
            val actual = response.content
            val expected = responseBody

            assertEquals(expected, actual)
        }
    }
}
```

4.5. Реализация непрерывной интеграции и развертка приложения

В качестве платформы для развертывания был выбран Heroku. Способов развертывания на этом сервисе несколько:

- «Ручной», с помощью Heroku CLI;
- Привязка изменений в удаленном репозитории к Heroku;
- Развертка с помощью инструментария непрерывной интеграции.

Проект был размещен в удаленном репозитории на GitHub, поэтому самым удобным был второй вариант.

До определенного момента развертывание работало корректно, но в какой-то момент на стороне Heroku появилась ошибка, из-за которой выбранный способ перестал работать. В связи с этим были предприняты попытки автоматизировать развертывание с помощью инструмента непрерывной интеграции GitHub Actions, но это вызвало множество проблем. После обращения в службу поддержки Heroku оказалось, что автоматическую развертку на Heroku поддерживают не все сервисы непрерывной интеграции. В итоге, в последние недели разработки, приходилось развертывать проект с помощью Heroku CLI.

Но непрерывная интеграция в виде GitHub Actions всё же была подключена к проекту. С помощью неё собирается проект и запускаются тесты при каждом изменении удаленного репозитория на GitHub.

4.6. Результат работы

В ходе выполнения практической части работы была реализована серверная часть приложения для ведения дневника тренировок. Также было реализовано взаимодействие с клиентской частью приложения.

Итоги работы содержат следующие результаты:

- Создана база данных PostgreSql;
- Реализована архитектура приложения;
- Реализована авторизация в приложение;
- В приложении реализовано тестирование;
- Приложение развернуто на Heroku.

5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

Разработка НИ производится группой работников, состоящей из двух человек – студента-инженера и научного руководителя.

Цель данной выпускной квалификационной работы заключается в проектировании и разработке серверной части мобильного приложения для учета спортивной активности, людей, занимающихся в тренажерном зале.

Целью раздела «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение» является определение перспективности и успешности НИ, оценка его эффективности, уровня возможных рисков, разработка механизма управления и сопровождения конкретных проектных решений на этапе реализации.

Для достижения обозначенной цели необходимо решить следующие задачи:

1. Оценить коммерческий потенциал и перспективность разработки НИ;
2. Осуществить планирование этапов выполнения исследования;
3. Рассчитать бюджет затрат на исследования;
4. Произвести оценку научно-технического уровня исследования и оценку рисков.

5.1. Потенциальные потребители результатов исследования

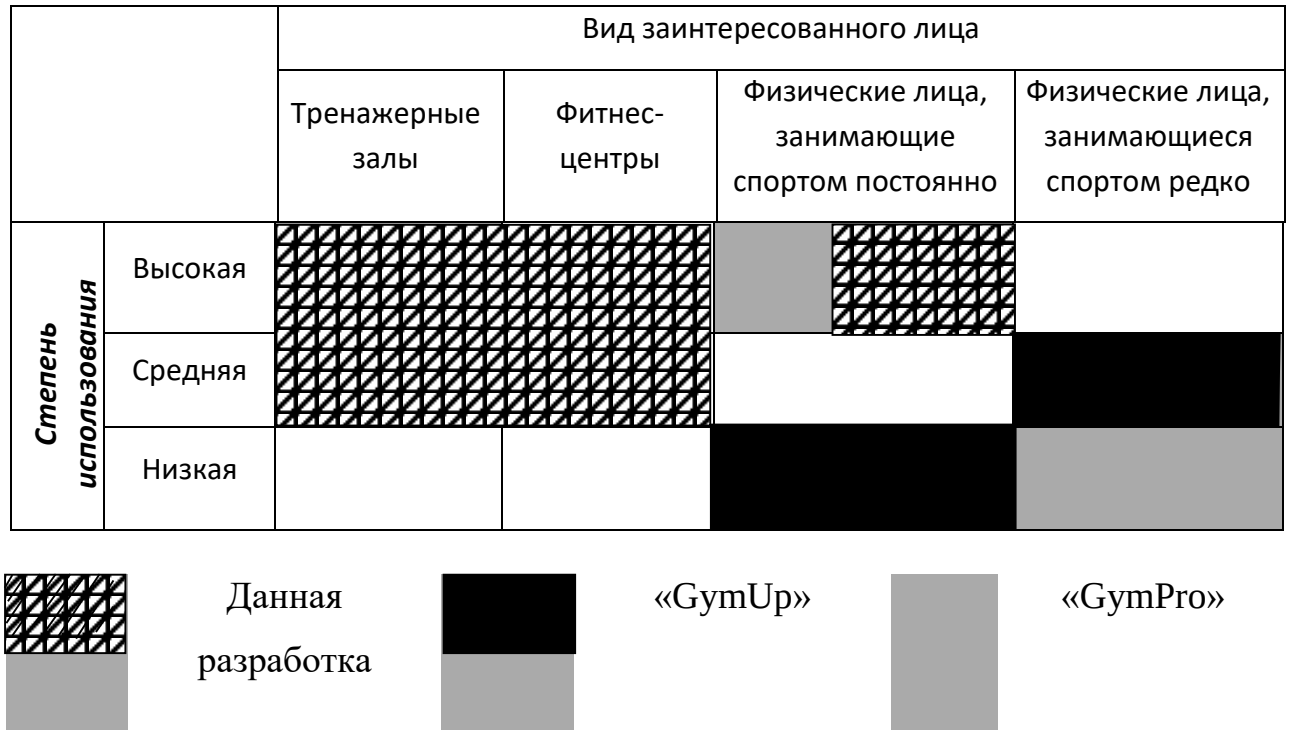
Продуктом разработки, которой посвящена работа, является мобильное приложение для учета спортивной активности.

Исходя из особенностей мобильного приложения, можно судить о круге лиц, которые потенциально будут заинтересованы в разработке. Целевым рынком текущей разработки являются физические лица и юридические лица, так или иначе занимающиеся деятельностью в области спорта, например, спортсмены или фитнес-центры. В данной области есть проблема с учетом спортивной активности, чаще всего она выполняется на

бумажных носителях, существуют даже специальные дневники тренировок, подготовленные для учета активности подобного рода. Все компании и физические лица будут заинтересованы в текущем проекте, так как он позволяет уйти от использования бумажных носителей и в некоторой мере автоматизировать процесс учета спортивной активности.

Сегментировать рынок услуг по данной тематике можно по степени использования подобных разработок и виду заинтересованных лиц.

Таблица 2 – Карта сегментирования рынка услуг по учету спортивной активности



На карте сегментирования рынка услуг по учету спортивной активности указано текущее положение дел для продуктов-конкурентов «GymUp» и «GymPro». Заштрихованными областями показаны области, в которые данный продукт будет интегрироваться, так как они на данный момент в большинстве своем никем не заняты и там преобладает использование бумажных носителей.

5.2. Анализ конкурентных технических решений

Разрабатываемое проектное решение не является уникальным в своем роде, но имеет отличия от конкурентов. В качестве конкурентов было выбрано несколько похожих проектов, позиционирующихся как приложения для учета спортивной активности.

Для анализа были выбраны следующие конкурентные решения:

- «GymUp – Дневник тренировок» (К1)
- «GymKeeper – Дневник тренировок» (К2)

Целесообразно проводить данный анализ с помощью оценочной карты. Оценочная карта для сравнения конкурентных технических решений представлена в таблице 3.

Таблица 3 – Оценочная карта для сравнения конкурентных технических решений

Критерии оценки	Вес критерия	Баллы			Конкурентоспособность		
		$Б_{ни}$	$Б_{к1}$	$Б_{к2}$	$К_{ни}$	$К_{к1}$	$К_{к2}$
1	2	3	4	5	6	7	8
Технические критерии оценки ресурсоэффективности							
1. Понятность интерфейса	0,1	4	2	3	0,4	0,2	0,3
2. Внешний вид	0,1	4	3	4	0,4	0,3	0,4
3. Удобство использования	0,2	5	3	5	1	0,6	1
4. Резервное копирование данных	0,15	3	0	4	0,45	0	0,6
5. Анализ тренировок	0,25	4	4	4	1	1	1
Экономические критерии оценки эффективности							
6. Стоимость подписки	0,1	4	3	1	0,4	0,3	0,1
7. Уровень проникновения на рынок	0,1	1	4	3	0,1	0,4	0,3

Итого	**1**	**25**	**19**	**24**	**3,75**	**2,8**	**3,7**

Анализ конкурентных технических решений определяется по формуле:

$$К = \sum В_i \times Б_i \quad (5.1)$$

где К – конкурентоспособность вида;

$В_i$– вес критерия (в долях единицы);

$Б_i$ – балл i-го показателя.

По данным оценочной карты можно увидеть, что разрабатываемое проектное решение является достаточно конкурентоспособным. Сильными сторонами является: удобство использования, внешний вид и понятность интерфейса; слабыми сторонами: резервное копирование и уровень проникновения на рынок. Так как проект только начинает свой путь, ему будет сложно проникнуть на рынок и занять там устойчивую позицию, поэтому необходимо предпринимать действия для продвижения проекта.

5.3. SWOT-анализ

Произведем также в данном разделе SWOT – анализ НИ, позволяющий оценить факторы и явления, способствующие или препятствующие продвижению проекта на рынок.

Сильные стороны — это факторы, которые положительно сказываются на развитии проекта. Сюда обычно включают все, что превращает функционирование в успешную и конкурентную работу.

Слабые стороны– это недостаток, упущение или ограниченность научно-исследовательского проекта, которые препятствуют достижению его целей. Это то, что плохо получается в рамках проекта или где он располагает недостаточными возможностями или ресурсами по сравнению с конкурентами.

Возможности включают в себя любую предпочтительную ситуацию в настоящем или будущем, возникающую в условиях окружающей среды проекта: тенденцию, изменение или предполагаемую потребность, которая поддерживает спрос на результаты проекта и позволяет руководству проекта улучшить свою конкурентную позицию.

Угроза представляет собой любую нежелательную ситуацию, тенденцию или изменение в условиях окружающей среды проекта, которые имеют разрушительный или угрожающий характер для его конкурентоспособности в настоящем или будущем. В качестве угрозы может выступать барьер, ограничение или что-либо еще, что может повлечь за собой проблемы, разрушения, вред или ущерб, наносимый проекту.

На первом этапе SWOT анализа в таблице 4 были описаны сильные и слабые стороны проекта, выявлены возможности и угрозы реализации НИ.

Таблица 4 – Матрица SWOT анализа

Сильные стороны	Возможности во внешней среде
С1. Удобство С2. Функциональность С3. Собственная расширяемая база упражнений	В1. Нетрудоемкая адаптация мобильного приложения под иностранные языки; В2. Повышение качества приложения В3. Сотрудничество с фитнес-клубами и тренажерными залами
Слабые стороны	Угрозы внешней среды
Сл1. Мобильное приложение доступно только на Android-устройствах Сл2. Отсутствие технической поддержки после реализации	У1. Отсутствие спроса; У2. Отсутствие обратной связи от пользователей У3. Нехватка финансирования

Второй этап состоит в выявлении соответствия сильных и слабых сторон научно-исследовательского проекта внешним условиям окружающей среды. Это соответствие или несоответствие должны помочь выявить степень

необходимости проведения стратегических изменений. В рамках данного этапа была построена интерактивную матрица проекта. Интерактивная матрица проекта представлена в табл. 5 и 6.

Таблица 5 – Интерактивная матрица сильных и слабых сторон и возможностей

	Сильные стороны				Слабые стороны	
Возможности проекта		С1	С2	С3	Сл1	Сл2
	В1	0	+	+	0	-
	В2	+	+	+	-	-
	В3	+	+	+	-	+

Таблица 6 – Интерактивная матрица сильных сторон и слабых сторон и угроз

	Сильные стороны				Слабые стороны	
Угрозы проекта		С1	С2	С3	Сл1	Сл2
	У1	+	+	+	-	-
	У2	+	+	+	-	+
	У3	+	+	+	-	-

Анализ интерактивных таблиц представляется в форме записи сильно коррелирующих сильных сторон и возможностей или слабых сторон и возможностей:

- В1В2В3С2С3
- В3Сл2
- У1У2У3С1С2С3
- У2Сл2

Из анализа можно сделать следующие выводы:

- Приложение имеет перспективны для расширения вне РФ. Сотрудничество с фитнес-клубами и тренажерными залами позволит возможности приложения;
- Для повышения качества приложения необходимо осуществлять техническую поддержку после реализации;
- Развивать приложение невозможно без роста спроса и обратной связи от пользователей;
- Большой угрозой для приложения является отсутствие спроса из-за отсутствия технической поддержки после реализации, в приложении могут быть критические ошибки, мешающие его нормальной работе.

В рамках третьего этапа была составлена итоговая матрица SWOT-анализа, представленная в таблице 7.

Таблица 7 – Итоговая матрица SWOT-анализа

	Сильные стороны научно-исследовательского проекта: С1. Удобство С2. Функциональность С3. Собственная расширяемая база упражнений	**Слабые стороны научно-исследовательского проекта:** Сл1. Мобильное приложение доступно только на Android-устройствах Сл2. Отсутствие технической поддержки после реализации
Возможности: В1. Нетрудоемкая адаптация мобильного приложения под иностранные языки; В2. Повышение качества приложения	Приложение имеет большой потенциал при дальнейшем расширении вне РФ. Сотрудничество с фитнес-клубами и тренажерными залами позволит расширять базу упражнений и функциональность.	Повышение качества приложения невозможно без технической поддержки после реализации.

В3. Сотрудничество с фитнес-клубами и тренажерными залами		
Угрозы: У1. Отсутствие спроса; У2. Отсутствие обратной связи от пользователей У3. Нехватка финансирования	Отсутствие спроса и обратной связи влияет на удобство и функциональность приложения для пользователей, без этих факторов развивать приложение будет невозможно.	Большой угрозой для приложение является отсутствие спроса из-за отсутствия технической поддержки после реализации, в приложении могут быть критические ошибки, мешающие его нормальной работе.

5.4. Планирование работ по научно-техническому исследованию

5.4.1. Структура работ в рамках научного исследования

Одним из начальных этапов проведения научно-исследовательских работ является необходимость планирования работ, которое включает в себя определение полного перечня работ и их распределения по исполнителям проекта. Над данным научным исследованием работают студент и научный руководитель. Студент, далее инженер, занимается разработкой мобильного приложения. Научный руководитель, контролирует работу студента, определяет цели и оценивает результаты проделанной работы.

Перечень этапов и работ, распределение исполнителей по данным видам работ приведен в таблице 8.

Таблица 8 – Перечень этапов, работ и распределение исполнителей

Основные этапы	№ раб	Содержание работ	Должность исполнителя
Разработка технического задания	1	Составление и утверждение технического задания	Инженер, Руководитель
Аналитическое	2	Анализ предметной области	Инженер

исследование предметной области	3	Анализ конкурентных решений	Инженер
	4	Календарное планирование работ	Руководитель, Инженер
Проектирование серверной части	5	Проектирование структуры базы данных	Инженер
	6	Проектирование архитектуры серверного приложения	Инженер
Разработка серверной части	7	Создание базы данных	Инженер
	8	Разработка серверной части приложения	Инженер
Обобщение и оценка результатов	9	Оценка эффективности полученных результатов	Научный руководитель
Оформление отчета по НИР	10	Составление пояснительной записки	Инженер

5.4.2. Определение трудоемкости выполнения работ

Трудовые затраты в большинстве случаях образуют основную часть стоимости разработки, поэтому важным моментом является определение трудоемкости работ каждого из участников научного исследования.

Трудоемкость выполнения научного исследования оценивается экспертным путем в человеко-днях и носит вероятностный характер, который зависит от множества трудно учитываемых факторов. Для определения ожидаемого (среднего) значения трудоемкости $t_{ожi}$ используется следующая формула:

$$t_{ожi} = \frac{3t_{\min i} + 2t_{\max i}}{5}, \quad (5.2)$$

где $t_{ожi}$ – ожидаемая трудоемкость выполнения i-ой работы чел.-дн.;

$t_{\min i}$– минимально возможная трудоемкость выполнения заданной i-ой работы, чел.-дн.;

$t_{\max i}$– максимально возможная трудоемкость выполнения заданной i-ой работы, чел.-дн.;

Исходя из ожидаемой трудоемкости работ, определяется продолжительность каждой работы в рабочих днях T_p, учитывающая параллельность выполнения работ по нескольким исполнителями.

$$T_{pi} = \frac{t_{\text{ожi}}}{\text{Ч}_i}, \quad (5.3)$$

где T_{pi} – продолжительность одной работы, раб.дн.;

$t_{\text{ожi}}$ – ожидаемая трудоемкость выполнения одной работы, чел.-дн.;

Ч_i – численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе, чел.

5.4.3. Разработка графика проведения научного исследования

Наиболее удобным и наглядным представлением проведения научных работ является построение ленточного графика в форме диаграммы Ганта.

Диаграмма Ганта – горизонтальный ленточный график, на котором работы по теме представляются протяженными во времени отрезками, характеризующимися датами начала и окончания выполнения данных работ.

Для удобства построение графика, длительность каждого из этапов работ из рабочих дней следует перевести в календарные дни. Для этого необходимо воспользоваться следующей формулой:

$$\text{T}_{\text{ki}} = \text{T}_{\text{pi}} \cdot \text{k}_{\text{кал}}, \quad (5.4)$$

где T_{ki} – продолжительность выполнения i-й работы в календарных днях;

T_{pi} – продолжительность выполнения i-й работы в рабочих днях;

$k_{\text{кал}}$– коэффициент календарности.

Коэффициент календарности определяется по следующей формуле:

$$k_{\text{кал}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - (T_{\text{вых}} + T_{\text{пр}})}, \quad (5.5)$$

где $T_{\text{кал}}$ – количество календарных дней в году;

$T_{\text{вых}}$ – количество выходных дней в году;

$T_{\text{пр}}$ – количество праздничных дней в году.

Расчет коэффициента календарности:

$$\text{k}_{\text{кал}} = \frac{\text{T}_{\text{кал}}}{\text{T}_{\text{кал}} - (\text{T}_{\text{вых}} + \text{T}_{\text{пр}})} = \frac{365}{365 - 118} = 1{,}48$$

Таблица 9 – Временные показатели проведения научного исследования

Название работы	Трудоемкость работ						Длительность работ в рабочих днях T_{pi}		Длительность работ в календарных днях T_{ki}	
	T_{min}, чел–дни		T_{max}, чел–дни		$T_{ожi}$, чел– дни		И	НР	И	НР
	И	НР	И	НР	И	НР				
Составление и утверждение технического задания	3	3	4	5	3,4	3,8	2	2	3	3
Анализ предметной области	4	-	8	-	5,6	-	6	-	8	-
Анализ конкурентных решений	2	-	4	-	2,8	-	3	-	4	-

Календарное планирование работ	2	2	3	3	2,4	2,4	1	1	2	2
Проектирование структуры базы данных	3	-	5	-	3,8	-	4	-	6	-
Проектирование архитектуры серверного приложения	15	-	22	-	17,8	-	18	-	27	-
Создание базы данных	2	-	5	-	3,2	-	3	-	4	-
Разработка серверной части приложения	30	-	44	-	35,6	-	36	-	53	-
Оценка эффективности полученных результатов	-	3	-	5	-	3,8	-	4	-	6

Составление пояснительной записки	10	-	20	-	14	-	14	-	21	-
Итого							87	7	128	11

Составлен план научного исследования, в котором разработан календарный план выполнения работ. Для построения таблицы временных показателей проведения НИ был рассчитан коэффициент календарности. С помощью показателей в таблице 9 был разработан календарный план-график проведения НИ по теме. Для иллюстрации календарного плана была использована диаграмма Ганта, указывающая на целесообразность проведения данного исследования.

Рисунок 11 – Диаграмма Ганта

5.5. Бюджет научно-технического исследования (НТИ)

1. Материальные затраты.
2. Затраты на специальное оборудование
3. Основная и дополнительная ЗП.
4. Социальные отчисления.
4. Прямые затраты.
5. Накладные расходы.

5.5.1. Расчет материальных затрат НТИ

Данная статья затрат включает в себя затраты на приобретение сырья, материалов, полуфабрикатов и комплектующих со стороны, используемых

при разработке проекта. Также в эту статью включаются транспортно-заготовительные расходы, равные 15% от общей стоимости материальных затрат.

Специфика данного проекта заключается в проектировании и разработке программного обеспечения на персональном компьютере из-за чего данный проект не подразумевает никаких материальных расходов.

5.5.2. Расчет затрат на специальное оборудование для научных работ

В данную статью включают все затраты, связанные с приобретением специального оборудования (приборов, контрольно-измерительной аппаратуры, стендов, устройств и механизмов), необходимого для проведения работ по конкретной теме. Определение стоимости спецоборудования производится по действующим прейскурантам, а в ряде случаев по договорной цене. При приобретении спецоборудования необходимо учесть затраты по его доставке и монтажу в размере 15% от его цены. Расчет затрат по данной статье представлен в таблице 10

Таблица 10 – Расчет бюджета затрат на приобретение спецоборудования для научных работ

Наименование	Единица измерения	Количество	Цена за ед., тыс. руб.	Затраты на материалы, тыс. руб.
ПК MSI GP63 с ПО	Шт.	1	75	86,25
Итого:				86,25

5.5.3. Основная заработная плата исполнителя темы

В настоящую статью включается основная заработная плата научных и инженерно-технических работников, рабочих макетных мастерских и опытных производств, непосредственно участвующих в выполнении работ по

данной теме. Величина расходов по заработной плате определяется исходя из трудоемкости выполняемых работ и действующей системы окладов и тарифных ставок. В состав основной заработной платы включается премия, выплачиваемая ежемесячно из фонда заработной платы в размере 20-30 % от тарифа или оклада. Расчет основной заработной платы приводится в таблице 11.

Таблица 11 – Расчет основной заработной платы

№ п/п	Наименование этапов	Исполнители по категориям	Трудоемкость, чел.-дн.	Заработная плата, приходящаяся на один чел-дн.	Всего заработная плата по тарифу (окладам), руб.
1.	Составление и утверждение технического задания	И, НР	2	4409,39	8818,77
2.	Анализ предметной области	И	6	1567,00	9401,97
3.	Анализ конкурентных решений	И	3	1567,00	4700,99
4.	Календарное планирование работ	И, НР	1	4409,39	4409,39
5.	Проектирование структуры базы данных	И	4	1567,00	6267,98
6.	Проектирование архитектуры серверного приложения	И	18	1567,00	28205,92
7.	Создание базы данных	И	3	1567,00	4700,99

8.	Разработка серверной части приложения	И	36	1567,00	56411,84
9.	Оценка эффективности полученных результатов	НР	4	2842,39	11369,56
10.	Составление пояснительной записки	И	14	1567,00	21937,94
Итого					156225,3

Статья включает основную заработную плату работников, непосредственно занятых выполнением проекта, (включая премии, доплаты) и дополнительную заработную плату и рассчитывается по формуле:

$$З_{зп} = З_{осн} + З_{доп} \quad (5.7)$$

где $З_{осн}$ – основная заработная плата;

$З_{доп}$ – дополнительная заработная плата (12–20 % от $З_{осн}$).

Основная заработная плата руководителя рассчитывается по следующей формуле:

$$З_{осн} = З_{дн} \cdot Т_р \quad (5.8)$$

где $З_{осн}$– основная заработная плата одного работника;

$Т_р$ – продолжительность работ, выполняемых научно-техническим работником, раб. дн.;

$З_{дн}$ – среднедневная заработная плата работника, руб.

Среднедневная заработная плата рассчитывается по формуле:

$$З_{дн} = \frac{З_м \cdot М}{F_д} \quad (5.9)$$

где $З_м$ – месячный должностной оклад работника, руб.;

М – количество месяцев работы без отпуска в течение года:

при отпуске в 24 раб. дня М =11,2 месяца, 5–дневная неделя;

при отпуске в 48 раб. дней М=10,4 месяца, 6–дневная неделя;

$F_д$ – действительный годовой фонд рабочего времени научно–технического персонала, раб. дн.

Таблица 12 – Баланс рабочего времени

Показатели рабочего времени	**Руководитель**	**Исполнитель**
Календарное число дней	365	365
Количество нерабочих дней	66	118
- выходные дни	52	104
- праздничные дни	14	14
Потери рабочего времени	48	24
- отпуск	48	24
- невыходы по болезни	-	-
Действительный годовой фонд рабочего времени	251	223

Месячный должностной оклад работника (руководителя):

$$З_м = З_{тс} \cdot (1 + k_{пр} + k_д) \cdot k_р \quad (5.10)$$

где $З_{тс}$ – заработная плата по тарифной ставке, руб.;

$k_{пр}$ – премиальный коэффициент, равный 0,3 (т.е. 30 процентов от $З_{тс}$);

$k_д$ – коэффициент доплат и надбавок составляет примерно 0,2 – 0,5;

$k_р$ – районный коэффициент, равный 1,3 (для Томска).

Тарифная заработная плата $З_{тс}$ находится из произведения тарифной ставки работника 1-го разряда Tci = 600 руб. на тарифный коэффициент $k_т$ и учитывается по единой для бюджетной организации тарифной сетке.

Тарифный коэффициент для НР = 1,866; для С = 1,407.

Расчет основной заработной платы представлен в таблице 13

Таблица 13 – Расчет основной заработной платы

Исполнители	Разряд	$k_Т$	$З_{тс}$, руб.	$k_{пр}$	$k_д$	$k_р$	$З_м$, руб.	$З_{дн}$, руб.	$Т_р$, раб. дн.	$З_{осн}$, руб.
Научный руководитель	Старший преподава-тель	1,866	35000	-	0,4	1,3	63700	2842,39	7	19896,73
Студент	Инженер	1,407	20000	-	0,2	1,3	31200	1567	87	136328,6
Итого										156225,3

5.5.4. Расчет дополнительной заработной платы

Дополнительная заработная плата учитывает величину предусмотренных Трудовым кодексом РФ доплат за отклонение от нормальных условий труда, а также выплат, связанных с обеспечением гарантий и компенсаций (при исполнении государственных и общественных обязанностей, при совмещении работы с обучением, при предоставлении ежегодного оплачиваемого отпуска и т.д.).

Расчет дополнительной заработной платы рассчитывается по формуле:

$$З_{доп} = k_{доп} \cdot З_{осн}\ , \qquad (5.11)$$

где $k_{доп}$ – коэффициент дополнительной заработной платы, принятый на стадии проектирования за 0,12. В таблице 14 приведена форма расчёта основной и дополнительной заработной платы.

Таблица 14 – Заработная плата исполнителей проекта

Заработная плата	**Руководитель**	**Исполнитель**
Основная зарплата	19896,73	136328,6
Дополнительная зарплата	2387,85	16359,43
Итого по статье $C_{зп}$	174972,61	

5.5.5. Отчисления во внебюджетные фонды

В данной статье расходов отражаются обязательные отчисления по установленным законодательством Российской Федерации нормам органам государственного социального страхования (ФСС), пенсионного фонда (ПФ) и медицинского страхования (ФФОМС) от затрат на оплату труда работников.

Величина отчислений во внебюджетные фонды определяется исходя из формулы:

$$З_{внеб} = k_{внеб} \cdot (З_{осн} + З_{доп}) \qquad (5.12)$$

где $k_{внеб}$ – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и пр.).

В соответствии с Федеральным законом от 24.07.2009 №212-ФЗ установлен размер страховых взносов равный 30,2%.

Отчисления во внебюджетные фонды представлены в таблице 15.

Таблица 15 – Отчисления во внебюджетные фонды

Исполнитель	Основная заработная плата, руб.	Дополнительная заработная плата, руб.
Руководитель проекта	19896,73	2387,85
Студент	136328,6	16359,43
Коэффициент отчислений во внебюджетные фонды	0,302	
Итого		52841,73

5.5.6. Накладные расходы

Накладные расходы учитывают прочие затраты организации, не попавшие в предыдущие статьи расходов. Их величина определяется по формуле:

$$З_{накл} = (\sum статей) \cdot k_{нр} \qquad (5.13)$$

где $k_{нр}$ – коэффициент, учитывающий накладные расходы.

Величину коэффициента накладных расходов можно взять в размере 16%.

Накладные расходы для исполнения1 составили:

$$З_{накл} = (0 + 86250 + 156225{,}3 + 18747{,}28 + 52841{,}73) \cdot 0{,}16$$
$$= 50250{,}29 \text{ руб.}$$

5.5.7. Формирование бюджета затрат научно-исследовательского проекта

Рассчитанная величина затрат научно–исследовательской работы является основой для формирования бюджета затрат проекта. Определение бюджета затрат на научно–исследовательский проект приведено в таблице 16.

Таблица 16 –Расчет бюджета затрат НТИ

Наименование статьи	Сумма, руб.	Примечание
1. Материальные затраты НТИ	0	Пункт 4.5.1
2. Затраты на специальное оборудование для научных (экспериментальных) работ	86250	Пункт 4.5.2
3. Затраты по основной заработной плате исполнителей темы	156225,3	Пункт 4.5.3
4. Затраты по дополнительной заработной плате исполнителей темы	18747,28	Пункт 4.5.4
5. Отчисления во внебюджетные фонды	52841,73	Пункт 4.5.5
6. Затраты на научные и производственные командировки	-	Отсутствуют
7. Контрагентские расходы	-	Отсутствуют
8. Накладные расходы	50250,29	Пункт 4.5.6
9. Бюджет затрат НТИ	364314,6	

5.6. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования

Определение эффективности происходит на основе расчета интегрального показателя эффективности научного исследования. Его нахождение связано с определением двух средневзвешенных величин: финансовой эффективности и ресурсоэффективности.

Интегральный показатель финансовой эффективности научного исследования определяется как:

$$I_{\text{фин.р}}^{\text{исп.}i} = \frac{\Phi_{pi}}{\Phi_{max}} \quad (5.14)$$

где $I_{\text{фин.р}}^{\text{исп.}i}$ – интегральный финансовый показатель разработки;

Φ_{pi} – стоимость i-го варианта исполнения;

Φ_{max} – максимальная стоимость исполнения научно-исследовательского проекта.

$$I_{\text{фин.р}}^{\text{исп1}} = \frac{364314{,}6}{364314{,}6} = 1;$$

Интегральный показатель ресурсоэффективности вариантов исполнения объекта исследования можно определить следующим образом:

$$I_{pi} = \sum_{i=1}^{n} a_i \times b_i \quad (5.15)$$

где I_{pi} – интегральный показатель ресурсоэффективности для i-го варианта исполнения разработки;

a_i – весовой коэффициент i-го варианта исполнения разработки;

b_i^a, b_i^p – бальная оценка i-го варианта исполнения разработки, устанавливается экспертным путем по выбранной шкале оценивания;

n – число параметров сравнения.

Таблица 17 – Сравнительная оценка характеристик вариантов исполнения проекта

Объект исследования / Критерии	Весовой коэффициент параметра	Исп.1

1. Способствует росту производительности труда пользователя	0,1	4
2. Удобство в эксплуатации (соответствует требованиям потребителей)	0,2	5
3. Помехоустойчивость	0,15	5
4. Энергосбережение	0,15	4
5. Надежность	0,2	5
6. Материалоемкость	0,2	5
Итого	1	4,75

$$I_{\text{р}-\text{исп1}} = 0{,}1 \cdot 4 + 0{,}2 \cdot 5 + 0{,}15 \cdot 5 + 0{,}15 \cdot 4 + 0{,}2 \cdot 5 + 0{,}2 \cdot 5 = 4{,}75;$$

Интегральный показатель эффективности вариантов исполнения разработки ($I_{\text{исп}i}$) определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формуле:

$$I_{\text{исп1}} = \frac{I_{\text{р}-\text{исп1}}}{I_{\text{фин.р}}^{\text{исп1}}} = \frac{4{,}75}{1} = 4{,}75$$

Сравнение интегрального показателя эффективности вариантов исполнения разработки позволит определить сравнительную эффективность проекта и выбрать наиболее целесообразный вариант из предложенных.

Сравнительная эффективность проекта ($Э_{ср}$):

$$Э_{\text{ср}} = \frac{I_{\text{исп2}}}{I_{\text{исп1}}} \qquad (5.16)$$

Таблица 18– Сравнительная эффективность разработки

№	Показатели	Исп.1
1	Интегральный финансовый показатель разработки	1
2	Интегральный показатель ресурсоэффективности разработки	4,75
3	Интегральный показатель эффективности	4,75
4	Сравнительная эффективность вариантов исполнения	1

Исполнение проекта возможно только в одном варианте, так как проект задается руководителем.

5.7. Вывод по разделу

В рамках раздела «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение» был оценен коммерческий потенциал разработки, потенциальные потребители результатов исследования, приведен анализ конкурентных решений. Основываясь на результатах проведенного в данном разделе анализа, можно сделать вывод, что проект является конкурентоспособным и перспективным. Стоимость разработки НИ составила 364314,6 тыс. руб.

6. Социальная ответственность

В рамках выполнения работы выполнялось проектирование и разработка серверной части мобильного приложения для отслеживания спортивной активности. Потенциальными потребителями являются люди, занимающиеся спортом.

Рабочей зоной является офисное помещение площадью $24\ м^2$. В помещении располагается компьютер и мобильный телефон для разработки и контроля качества разрабатываемого решения. В рамках рабочего процесса осуществляется проектирование и разработка программного обеспечения.

6.1. Правовые и организационные вопросы обеспечения безопасности

В трудовом кодексе РФ содержатся основные положения отношений между организацией и сотрудниками, включая оплату и нормирование труда, выходных, отпуска и так далее [12].

Работа в офисе относится ко второй категории тяжести труда – работы выполняются при оптимальных условиях внешней производственной среды и при оптимальной величине физической, умственной и нервно-эмоциональной нагрузки. Продолжительность рабочего дня работников не должна превышать 40 часов в неделю. Возможно, сокращение рабочего времени. Для работников, возраст которых меньше 16 лет – не более 24 часов в неделю, от 16 до 18 лет – не более 35 часов, как и для инвалидов I и II групп [13].

Согласно закону об информации, информационных технологиях и о защите информации работодатель не вправе собирать, хранить, использовать и распространять информацию о частной жизни работников без их письменного согласия [14]. Результаты труда работников, а именно исходный текст проектного решения является коммерческой тайной. Согласно закону о коммерческой тайне, работодатель обязан обеспечить работника условиями для соблюдения установленного режима коммерческой тайны [15].

Рабочее место должно соответствовать приказу Минтруда России об организации безопасного рабочего места [16]. Рабочее место должно быть оборудовано креслом с возможность регулировки его высоты и наклона спинки. Работник должен поддерживать порядок на рабочем месте в целях обеспечения собственной безопасности и безопасности окружающих.

6.2. Производственная безопасность

В данном пункте анализируются вредные и опасные факторы, которые могут возникать при проведении при разработке или эксплуатации проектируемого решения.

Для идентификации потенциальных факторов необходимо использовать ГОСТ 12.0.003-2015 «Опасные и вредные производственные факторы. Классификация» [22]. Перечень опасных и вредных факторов, характерных для проектируемой производственной среды представлены в таблице 19.

Таблица 19 – Возможные опасные и вредные производственные факторы на рабочем месте инженера-программиста

Факторы (ГОСТ 12.0.003-2015)	Нормативные документы
Вредные факторы	
Отсутствие или недостаток необходимого естественного освещения	СП 52.13330.2016
Отсутствие или недостатки необходимого искусственного освещения	
Повышенная пульсация светового потока	

Нервно-психические перегрузки, связанные с напряженностью трудового процесса	Трудовой кодекс Российской Федерации от 30.12.2001 №197-ФЗ (ред. от 01.03.2022)
Статические перегрузки, связанные с рабочей позой	Приказ Минтруда России от 29 октября 2021 г. № 774н «Об утверждении общих требований к организации безопасного рабочего места»
Длительное сосредоточенное наблюдение	
Опасные факторы	
Производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которого попадает работающий	ГОСТ 12.1.038-82

6.3. Анализ опасных и вредных производственных факторов

6.3.1. Отсутствие или недостаток необходимого естественного или искусственного освещения и повышенная пульсация светового потока

Недостаточная освещенность рабочей зоны помещения, оборудованной ПК, также является одной из причин нарушения зрительной функции, а также влияет на общее самочувствие и эффективность труда.

Искусственные источники света, питаемые переменным или импульсным током, обладают характеристикой пульсации светового потока. Превышение норм пульсации может при краткосрочном влиянии привести к

усталости органов зрения, снижению внимания, утомляемости организма, замедлению активности мозга, тошноте и нарушению пищеварения. При долгосрочном влиянии последствия более серьезные: депрессия, бессонница, патология сердечно-сосудистой системы.

Искусственное освещение в помещениях для эксплуатации ПК должно осуществляться системой общего равномерного освещения. В случаях преимущественной работы с документами, следует применять системы комбинированного освещения (к общему освещению дополнительно устанавливаются светильники местного освещения, предназначенные для освещения зоны расположения документов). Окна в помещениях, где эксплуатируется вычислительная техника, преимущественно должны быть ориентированы на север и северо-восток. Коэффициент естественного освещения должен быть равен 4,2 и 1,5 при верхнем или комбинированном освещении и при боковом освещении соответственно. Требования к показателям искусственного освещения в соответствии с СП 52.13330.2016 [21] указаны в таблице 20.

Таблица 20 – Требования к освещению рабочего места

Характеристика зрительной работы	Наименьший или эквивалентный размер объекта различения, мм	Искусственное освещение			
		Средняя освещенность на рабочей поверхности от системы общего освещения, лк, не менее	Цилиндрическая освещенность, лк	Объединенный показатель UGR, не более	Коэффициент пульсации освещенности K_n%, не более
Очень высокой точности	От 0,15 до 0,30	750	150	16	10

6.3.2. Нервно-психические перегрузки, связанные с напряженностью трудового процесса

Нервно-психические перегрузки – совокупность таких сдвигов в психофизиологическом состоянии организма человека, которые развиваются после совершения работы и приводят к временному снижению эффективности труда. Состояние утомления (усталость) характеризуется определенными объективными показателями и субъективными ощущениями.

Нервно-психические перегрузки подразделяются на следующие:

- умственное перенапряжение;
- перенапряжение анализаторов;
- монотонность труда;
- эмоциональные перегрузки.

При первых симптомах психического перенапряжения необходимо:

- дать нервной системе расслабиться;
- рационально чередовать периоды отдыха и работы;
- начать заниматься спортом;
- ложиться спать в одно и то же время;
- в тяжелых случаях обратиться к врачу.

Естественно, что полностью исключить провоцирующие факторы из жизни вряд ли удастся, но можно уменьшить их негативное воздействие, давая нервной системе необходимый отдых.

6.3.3. Статические перегрузки, связанные с рабочей позой

Статические перегрузки, связанные с рабочей позой, возникают из-за несоответствия рабочего места требованиям к его организации. Чтобы

компенсировать этот фактор, необходимо руководствоваться приказом Минтруда России об организации безопасного рабочего места. Рабочее место должно быть оборудовано креслом с возможность регулировки его высоты и наклона спинки и рабочей поверхностью требуемой высоты.

6.3.4. Длительное сосредоточенное наблюдение

Работа с использованием компьютеров предполагает длительное наблюдение за происходящим на экране монитора, это влечет за собой постоянное напряжение зрительного анализатора. Чтобы снизить нагрузку необходимо обеспечить дополнительные перерывы для работников.

6.3.5. Производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которого попадает работающий

Во время работы над проектным решением работник может получить травму от воздействия электрического тока. Оно может привести к ожогам частей тела или даже к смерти. Поражение током, может произойти в результате прикосновения к токоведущим частям техники, на которых появилось напряжение или остался заряд. Для переменного тока частотой 50 Гц допустимое значение напряжения прикосновения составляет 2 В, а силы тока – 0,3 мА, для тока частотой 400 Гц, соответственно – 2 В и 0,4 мА, для постоянного тока – 8 В и 1 мА [17].

В качестве мер защиты нужно использовать оградительные устройства, устройства автоматического контроля и сигнализации, изолирующие устройства и покрытия, устройства защитного заземления.

6.4. Экологическая безопасность

Во время работы над проектным решением может произойти загрязнение литосферы.

Вышедшие из строя компьютеры и мобильные телефоны, использующиеся во время разработки, являются опасными отходами. Согласно приказу Минприроды России «Об утверждении требований при обращении с группами однородных отходов I-V классов опасности» компаниям и индивидуальным предпринимателям запрещено выбрасывать компьютерную технику в мусорные баки [18]. Поэтому, чтобы свести воздействие на литосферу к минимуму, необходимо передавать компьютерную технику на утилизацию в специализированные центры.

6.5. Безопасность в ЧС

6.5.1. Анализ возможных ЧС

Во время работы над проектным решением есть вероятность возникновения чрезвычайных ситуаций, таких как пожар, наводнение, ураган и землетрясение.

6.5.2. Анализ наиболее вероятной ЧС

Поскольку для разработки проектного решения используется компьютерное оборудование, то наиболее вероятной ЧС является пожар, возникший вследствие неисправности оборудования или нарушения техники безопасности. В ГОСТ 12.1.004–91 «Пожарная безопасность. Общие требования», регламентированы нормы пожарной безопасности, которые необходимо соблюдать при работе за компьютером [19]:

- работы за компьютером должны проводиться только при исправном состоянии оборудования и электропроводки;
- в помещении должны быть средства для тушения пожаров, а именно огнетушители;
- должны быть предусмотрены эвакуационные выходы и обеспечено беспрепятственное движение людей к ним.

В случае возникновения пожара работникам необходимо:

- немедленно сообщить по телефону о пожаре в пожарную охрану;
- оповестить о пожаре всех работников;
- принять меры по эвакуации людей, тушению пожара, сохранению материальных ценностей.

На основании Федерального закона от 22.07.2008 N 123-ФЗ (ред. От 30.04.2021) «Технический регламент о требованиях пожарной безопасности», возникший пожар будет относиться к классу А – пожары твердых горючих веществ и материалов. Помещение необходимо обеспечить следующими первичными средствами пожаротушения: огнетушитель, пожарный кран (на этаже), покрывало для изоляции очага возгорания [20].

6.6. Вывод по разделу

В рамках раздела были рассмотрены организационные и правовые аспекты обеспечения безопасности при работе над проектным решением, проанализированы возможные опасные и вредные факторы производства, с приведением мер по снижению воздействия рассмотренных факторов, в подразделе по экологической безопасности было проанализировано влияние на окружающую среду. В разделе был проведен анализ возможных ЧС и было выявлено наиболее вероятное из них.

Помещение, в котором разрабатывается проектное решение в отношении опасности поражения людей электрическим током относится к помещению без повышенной опасности. Персонал относится к I группе по электробезопасности. Категория тяжести труда при разработке проектного решения – Iа. В офисном помещении не производится работа с горючими материалами или газами, поэтому категория помещения по взрывопожарной и пожарной опасности, соответствует умеренной пожароопасности (Г). Объект не оказывает значительное негативное воздействие на окружающую среду и

соответствует критериям IV категории объектов, оказывающих негативное воздействие на окружающую среду.

Заключение

В ходе выполнения работы была проанализирована выбранная предметная область, спроектирована и реализована серверная часть приложения для ведения дневника тренировок. Был реализован функционал из блока must порядка реализации задач.

Благодаря использованию современных веяний в построении архитектуры кода, с добавлением нового функционала в приложение не будет возникать критичных проблем. Тестирование кода повышает его общее качество и надежность.

В дальнейшем, после расширения функционала приложения, планируется разместить его в магазине приложение Play Market.

Список используемых источников

1. Ktor Documentation [Электронный ресурс]. - URL: https://ktor.io/docs/welcome.html (Дата обращения: 20.05.2022).
2. Kotlin with Ktor [Электронный ресурс]. - URL: https://www.baeldung.com/kotlin/ktor (Дата обращения: 20.05.2022).
3. Exposed wiki [Электронный ресурс]. - URL: https://github.com/JetBrains/Exposed/wiki (Дата обращения: 20.05.2022).
4. Руководство по работе с фреймворком Kotlin Exposed [Электронный ресурс]. - URL: https://habr.com/ru/post/464185/ (Дата обращения: 20.05.2022).
5. PostgreSql documentation [Электронный ресурс]. - URL: https://www.postgresql.org/docs/ (Дата обращения: 20.05.2022).
6. Koin for Ktor [Электронный ресурс]. - URL: https://insert-koin.io/docs/reference/koin-ktor/ktor/ (Дата обращения: 20.05.2022).
7. The Heroku CLI [Электронный ресурс]. - URL: https://devcenter.heroku.com/articles/heroku-cli (Дата обращения: 20.05.2022).
8. Mocking is not rocket science: Basics [Электронный ресурс]. - URL: https://blog.kotlin-academy.com/mocking-is-not-rocket-science-basics-ae55d0aadf2b (Дата обращения: 20.05.2022).

9. MockK — библиотека для mocking-a в Kotlin [Электронный ресурс]. - URL: https://habr.com/ru/post/341202/ (Дата обращения: 20.05.2022).

10. Kodein — интересная альтернатива Dagger 2 для внедрения зависимостей в Kotlin [Электронный ресурс]. - URL: https://habr.com/ru/company/sberbank/blog/347208/ (Дата обращения: 20.05.2022).

11. Коротко о главном: Clean Architecture, Robert C. Martin [Электронный ресурс]. - URL: https://habr.com/ru/post/464185/ (Дата обращения: 20.05.2022).

12. Трудовой кодекс Российской Федерации от 30.12.2001 №197-ФЗ (ред. от 01.03.2022) – Текст: непосредственный

13. ГОСТ 12.1.005–88 Система стандартов безопасности труда. Общие санитарно-гигиенические требования к воздуху рабочей зоны (правовые нормы) – Текст: непосредственный

14. Федеральный закон от 27 июля 2006 г. № 149-ФЗ «Об информации, информационных технологиях и о защите информации» – Текст: непосредственный

15. Федеральный закон от 29 июля 2004 г. № 98-ФЗ «О коммерческой тайне» – Текст: непосредственный

16. Приказ Минтруда России от 29 октября 2021 г. № 774н «Об утверждении общих требований к организации безопасного рабочего места» – Текст: непосредственный

17. ГОСТ 12.1.038–82 ССБТ. Электробезопасность. Предельно допустимые уровни напряжений прикосновения и токов (с Изменением N 1) – Текст: непосредственный.

18. Приказ Минприроды России (Министерства природных ресурсов и экологии РФ) от 11 июня 2021 г. № 399 «Об утверждении требований при обращении с группами однородных отходов I–V классов опасности» – Текст: непосредственный

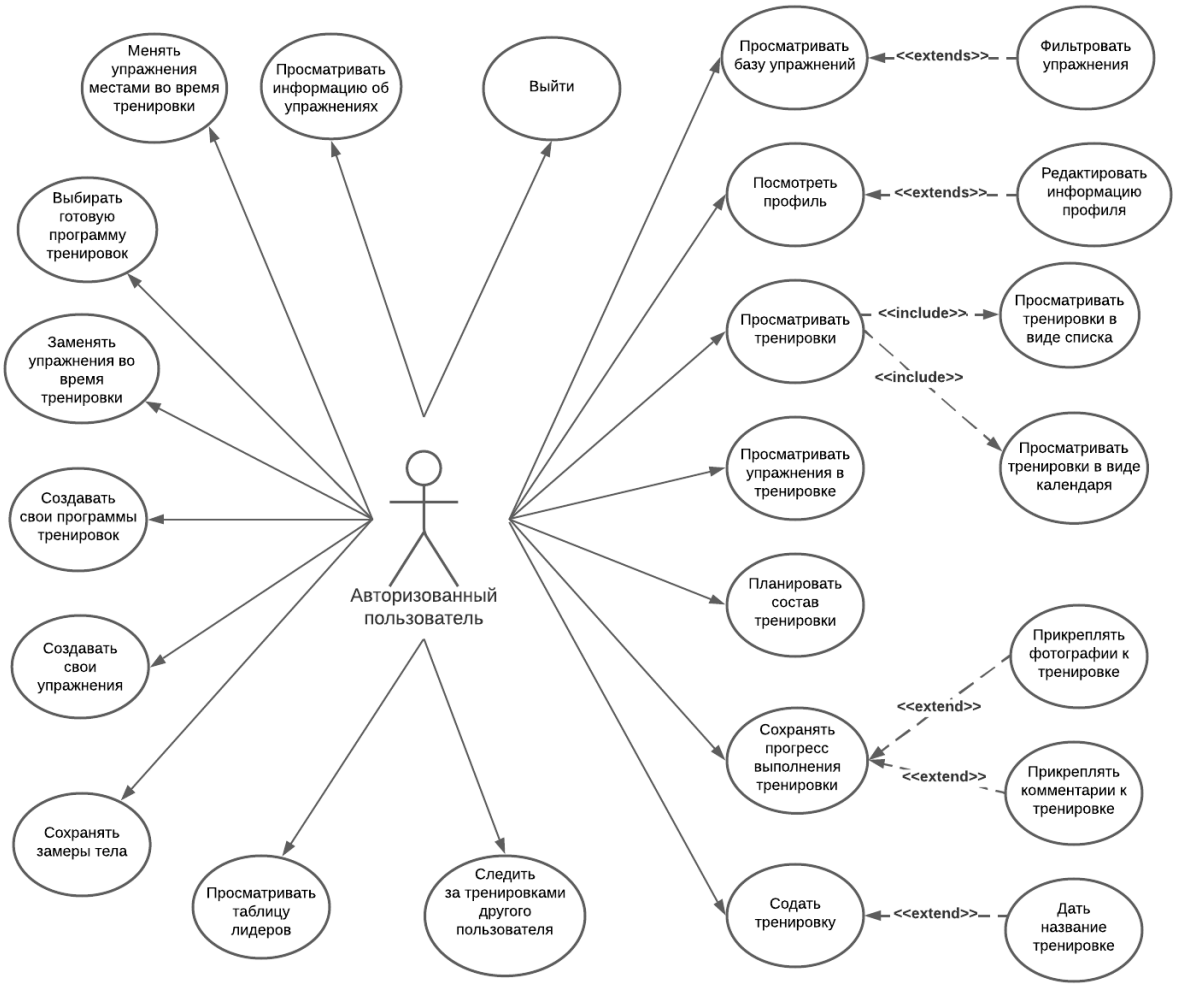
19. ГОСТ 12.1.004–91 Система стандартов безопасности труда (ССБТ). Пожарная безопасность. Общие требования (с Изменением N 1) ГОСТ от 14 июня 1991 г. № 12.1.004-91 – Текст: непосредственный

20. Федеральный закон от 22.07.2008 N 123-ФЗ (ред. От 30.04.2021) «Технический регламент о требованиях пожарной безопасности» – Текст: непосредственный

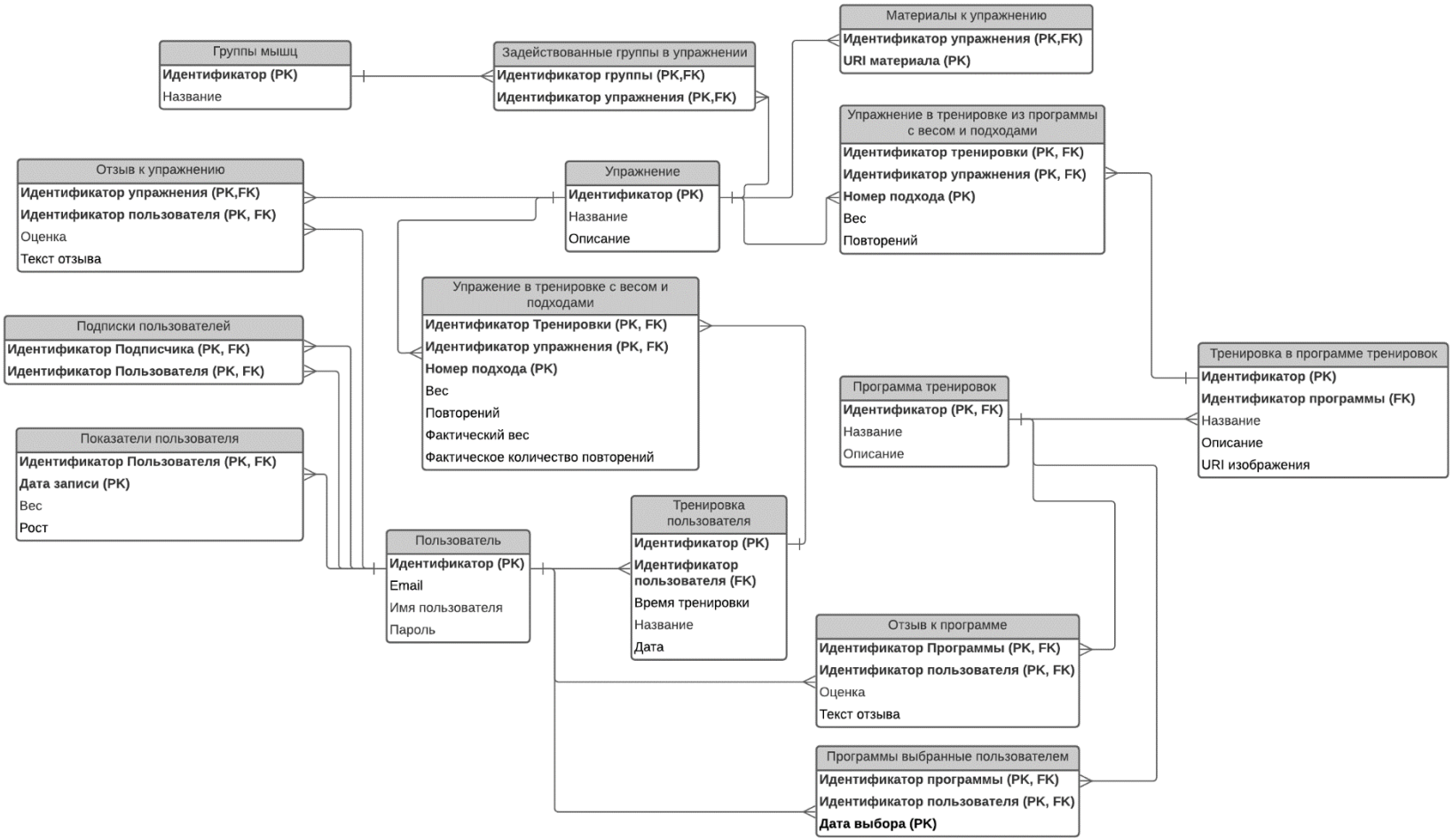
21. СП 52.13330.2016. Естественное и искусственное освещение: утвержден Приказом Министерства строительства и жилищно-коммунального хозяйства Российской Федерации от 7 ноября 2016 г. N 777/ введен с 8 мая 2017 г. – Текст: непосредственный.

22. ГОСТ 12.0.003–2015 Система стандартов безопасности труда (ССБТ). Опасные и вредные производственные факторы. Классификация (с Поправками) ГОСТ от 09 июня 2016 г. № 12.0.003-2015 – Текст: непосредственный

Приложение А



Приложение Б



Приложение В


```
{
    "workoutPlans": [
        {
            "id": 1,
            "title": "First plan",
            "description": "awfawfawf",
            "imageUri": "https://www.google.com/search?q=%D0%BA%D0%B0%D1%80%D
1%82%D0%B8%D0%BD%D0%BA%D0%B0+%D1%82%D1%80%D0%B5%D0%BD%D0%B8%D1%80%D0%BE%D0%B2
%D0%BA%D0%B8&sxsrf=APq-
WBsUJ0Sb696wMg404YStl8LmVoJepA:1648892387838&source=lnms&tbm=isch&sa=X&sqi=2&
ved=2ahUKEwio4LLoivX2AhUpg3IEHR3mBOgQ_AUoAXoECAEQAw&biw=1536&bih=722&dpr=1.25
#imgrc=Ke5Gph__nk7v-M",
            "workouts": [
                {
                    "id": 1,
                    "title": "1",
                    "description": "first workout",
                    "imageUri": "23123",
                    "exercises": [
                        {
                            "id": 1,
                            "title": "exercise1",
                            "description": "The exercise1",
                            "imageUri": "",
                            "sets": [
                                {
                                    "index": 12,
                                    "weight": 123.0,
                                    "repetitionsNumber": 1
                                },
                                {
                                    "index": 13,
                                    "weight": 232.0,
                                    "repetitionsNumber": 2
                                }
                            ]
                        },
                        {
                            "id": 2,
                            "title": "exercise2",
                            "description": "The exercise2\n",
                            "imageUri": "",
                            "sets": [
                                {
                                    "index": 2,
                                    "weight": 232.0,
                                    "repetitionsNumber": 123
                                }
                            ]
```

```
                        }
                    ]
                }
            ]
        }
    ],
    "exercises": [
        {
            "id": 1,
            "title": "exercise1",
            "imageUri": "The exercise1",
            "description": "",
            "sets": []
        }
    ],
    "news": [
        {
            "title": "Андрей Василевский установил рекорд НХЛ по количеству с
ухих матчей в решающих играх серии - Чемпионат",
            "imageUri": "https://img.championat.com/news2/social/b/d6/4704301
.jpg",
            "description": "27-летний российский голкипер «Тампа-
Бэй Лайтнинг» Андрей Василевский стал рекордсменом Кубка Стэнли по количеству
 сухих матчей в решающих играх серии плей-офф. Об этом сообщает пресс-
служба НХЛ в «Твиттере».",
            "sourceUrl": "https://www.championat.com/hockey/news-4704301-
andrej-vasilevskij-ustanovil-rekord-nhl-po-kolichestvu-suhih-matchej-v-
reshayuschih-igrah-serii.html",
            "time": 1653358458000
        }
    ]
}
```